# Neural Network Analysis in Higgs Search using $t\bar{t}H, H \rightarrow b\bar{b}$

***and***

# TAG Database Development for ATLAS

Helen Marie McGlone

University of Glasgow

*Department of Physics and Astronomy*
*University of Glasgow*

Thesis submitted for the degree of
Doctor of Philosophy

HM McGlone, 2009

CERN-THESIS-2009-193
//2009

For My Dad and His Da,
George and Patrick
And my Granda, Alec
with love

A Long December

It's been a long day this Thursday
Got a long long way to go
Been a long time since you've come by
And I only know the things that I know

Had a bad month in September
October scared the hell out of me
I get lonely in November
But December is where I want to be

A Long December and there's reason to believe
Maybe this year wil be better than the last
I can't remember the last thing that you said as you were leaving
Now the days go by so fast

And it's one more day up in the canyons
And it's one more night in Hollywood
If you think that I could be forgiven
I wish you would

The smell of hospitals in winter and the feeling
That it's all alot of oysters and no pearls
All at once you look across a crowded room
And the way the light attaches to a girl

And it's one more day up in the canyons
And it's one more night in Hollywood
If you think you might come to California
I think you should

Drove up to hillside manor sometime after two am
And talked a little while about the year

I guess the winter makes you laugh a little slower
Makes you talk a little lower about the things you could not show her

It's been a long December and there's reason to believe
Maybe this year will be better than the last
I can't remember all the times I've tried to tell my myself
To hold on to these moments as they pass

And it's one more day up in the canyon
And it's one more night in Hollywood
It's been I long time since I've seen the ocean
I guess I should

*Counting Crows*



*To 28 beautiful souls*, affectionate and beautiful children of *Tam Ky Baby Orphanage*, for their *love and inspiration*, for giving me hope and teaching me to survive








*Chi, Chi, Kien, Xuan, Tuan, Ngoc, Gai, Hieu, Trang, Chun, Chi, Lan, Tuay, Loc, Ngan, Nhu, Tuan, Ke, Tu, Choi, C'an, Tai, Giang*

A survivor is a triumphant person who lives with, after, or in spite of a traumatic event. Survivors refuse to assume the identity of their adversity. They are not imprisoned by constructs of a label. Instead, survivors use ther brush with mortality as a catalyst for creating *a better self*. We transform our experience in order to further evolve *spiritually, emotionally, physically* and mentally. *Our reality challenges us to go deeper*. Survivors cultivate an essence that will never be victim to a word

*Kris Carr inspired by Beth Villandry*

**Abstract**

The Large Hadron Collider, LHC, at Conseil Européen pour la Recherche Nucléaire, CERN, in Geneva, Switzerland, is an international physics project of unprecedented scale. First proton beams were circulated in the LHC in 2008. The ATLAS Collaboration, an international group of 2000 analysts, scientists, software developers and hardware experts, seeks to push the boundaries of our current understanding of the Universe, and our ability to undertake such studies.

A central physics focus of the ATLAS experiment is study of a Higgs boson, a theoretically predicted particle, as yet unobserved in nature. In this thesis, a Neural Network is adopted and developed as an analysis method in a study of a Standard Model Higgs boson in the low mass Higgs range, using the physics channel $t\bar{t}H, H \rightarrow b\bar{b}$ and Higgs mass $m_H = 120$ GeV. The Neural Network analysis shows that a neural network method can give an improvement in sensitivity of the $t\bar{t}H, H \rightarrow b\bar{b}$ channel. A set of Event Characteristics, associated with a topology where the existence of a reconstructed Higgs boson is not required in each event are defined and it is demonstrated that these characteristics, when used in a neural network, can improve the sensitivity of the channel by improving separation of signal and background events. The neural network analysis uses a collection of *Generic* Event Characteristics, a neural network of layout $36 : 8 : 4 : 1$, 1000 learning cycles and 734033 $t\bar{t}H, H \rightarrow b\bar{b}$ simulated signal and background events, for an integrated luminosity of $1 fb^{-1}$, to give an output sensitivity of 4.74. We see that the neural network analysis method as described in this analysis improves the sensitivity of the channel from that of the Cuts-Based Analysis performed in previous studies.

In the quest for new and multipurpose physics searches and studies, ATLAS will produce data of unprecedented volume and rate in Particle Physics. As analysts are internationally located, data must be accessible across worldwide collaborating institutions. A significant challenge for the ATLAS collaboration lies in developing the capacity in computing terms to manage an unprecedented data challenge in a fluid, sound and transparent way.

The ATLAS Event Level Metadata System, TAG Database, is a central part of the ATLAS Computing system. The Event Level Metadata system captures information about ATLAS physics events on an event by event basis, and offers later access to the events for analysis. In this thesis, developments and implementation of the Event Level Metadata system are presented in terms of three studies, these are Feasability, Scalability and Accessibility. Feasibility studies demonstrate that an Event Level Metadata system can operate within the larger ATLAS software system and gathered information on the implications for Event Level Metadata system development. Scalability studies present implementation and performance of a realistic terabyte scale relational TAG Database and demonstrate that an Event Level Metadata system at terabyte scale is achievable. Accessibilty studies present the development of a web interface to the Event Level Metadata system. Studies in this thesis therefore demonstrate that an Event Level Metadata can be integrated with the ATLAS software system, develop solutions for integration, prove that an Event Level Metadata relational

database can scale to ATLAS terabyte size, present performance results for a realistic ATLAS scale system and develop a user interface to the Event Level Metadata system.

# Contents

**Author's Declaration** **VII**

**1 Introduction** **1**

**2 The Standard Model** **3**
2.1 Introduction . . . 3
2.2 Particles and Forces . . . 4
2.3 Fermions . . . 5
2.4 Bosons . . . 7
2.5 Extending the Standard Model . . . 8
2.6 Grand Unification Theories . . . 8
2.7 Supersymmetric partners . . . 9
2.8 Gauge Theory and Symmetries . . . 10
2.9 Field theory for Electromagnetic interactions, QED, U(1) abelian group . . . 11
2.10 Gauge symmetry in non-abelian case . . . 14
2.11 Gauge theory for strong force, SU(3) . . . 15
2.12 Gauge theory for electroweak force, SU(2) . . . 15
2.13 The Higgs Mechanism . . . 15
2.14 Electroweak Lagrangian and Higgs Mechanism . . . 20
2.15 Fermion masses . . . 21
2.16 The Higgs Boson . . . 21
2.17 Constraints on the Higgs Boson mass . . . 21
2.18 Higgs Production and Decay at ATLAS . . . 22
2.18.1 Production . . . 22
2.18.2 Decay . . . 23
2.18.3 $m_H < 130$ GeV . . . 24
2.18.4 130 Gev $< m_H <$ 180 GeV . . . 24
2.18.5 180 GeV $< m_H <$ 1 TeV . . . 25
2.19 Discovery Potential . . . 25
2.20 Summary . . . 26

**3 Large Hadron Collider** **28**
3.1 Introduction . . . 28
3.2 The LHC . . . 28
3.3 The LHC Accelerator . . . 30
3.4 The LHC Accelerator Complex . . . 32
3.5 LHC Startup 2008 . . . 33

3.6 LHC Experiments . . . 36
3.7 LHC Physics Motivation . . . 37
3.8 Particle Physics Experiments leading to the LHC . . . 39
3.8.1 CERN . . . 39
3.8.2 Fermilab . . . 40
3.9 Summary . . . 41

**4 ATLAS** **42**
4.1 Introduction . . . 42
4.2 ATLAS Detector . . . 42
4.3 The ATLAS Co-ordinate System . . . 44
4.4 The Inner detector . . . 45
4.4.1 The Pixel Detector . . . 46
4.4.2 Semiconductor Tracker . . . 47
4.4.3 Transition Radiation Tracker . . . 47
4.5 Calorimetry . . . 48
4.5.1 The Electromagnetic calorimeter . . . 48
4.5.2 The Hadronic Calorimeter . . . 49
4.6 The Muon system . . . 50
4.7 Magnet Systems . . . 50
4.8 Trigger and DAQ . . . 50
4.8.1 Event Selection . . . 51
4.8.2 Menus, Signatures and Configuration . . . 54
4.9 The First Data seen ATLAS . . . 55
4.10 Summary . . . 56

**5 ATLAS Computing** **57**
5.1 Introduction . . . 57
5.2 The ATLAS Computing Model . . . 58
5.3 ATLAS Data . . . 58
5.4 ATLAS Event Data types . . . 59
5.4.1 RAW data . . . 59
5.4.2 ESD data . . . 59
5.4.3 AOD data . . . 60
5.4.4 TAG data . . . 60
5.4.5 Simulated data . . . 60
5.4.6 Derived Physics Data . . . 60
5.5 ATLAS Event Data Production . . . 62
5.5.1 First Pass ESD Production . . . 62
5.5.2 First Pass AOD Production . . . 62
5.5.3 TAG Production . . . 63
5.5.4 Reprocessing . . . 63
5.5.5 Processing times . . . 64
5.6 Distributed Computing . . . 64
5.7 ATLAS Tier Structure . . . 65
5.8 ATLAS Tiers . . . 65
5.8.1 Tier 0 . . . 65

5.8.2 Tier 1 . . . 66
5.8.3 Tier 2 . . . 67
5.8.4 Tier 3 . . . 68
5.9 ATLAS Data Flow . . . 68
5.10 ATLAS Data Analysis . . . 70
5.11 ATLAS Data Management . . . 70
5.11.1 The Distributed Data Management system . . . 71
5.11.2 Datasets . . . 71
5.11.3 Files . . . 72
5.11.4 Data Movement . . . 73
5.11.5 Dataset Catalogs . . . 74
5.11.6 Site Services . . . 76
5.11.7 User Tools . . . 76
5.12 ATLAS Data Persistency . . . 76
5.12.1 POOL . . . 77
5.12.2 POOL File Catalog . . . 78
5.12.3 POOL Collections . . . 78
5.13 ATLAS Data Storage . . . 80
5.13.1 Files . . . 80
5.13.2 Relational databases . . . 81
5.13.3 Coexistence of Files and Relational Databases . . . 81
5.14 ATLAS Data Navigation . . . 82
5.15 Event and Non Event data . . . 82
5.15.1 ATLAS Non Event Data . . . 83
5.15.2 Non event data Databases . . . 84
5.16 Summary . . . 86

**6 ATLAS Event Level Metadata** **87**
6.1 Introduction . . . 87
6.2 ATLAS Event Level Metadata . . . 88
6.3 TAG Database System . . . 89
6.4 TAG Use Cases . . . 91
6.5 TAG Data Volume . . . 92
6.6 TAG Data Rate . . . 92
6.7 TAG Database Distribution model . . . 93
6.8 TAG Writing . . . 93
6.9 TAG Attributes . . . 94
6.10 TAG Back Navigation . . . 95
6.11 TAGs and POOL Collections . . . 95
6.12 TAG size . . . 96
6.13 User Interaction with the TAG Database . . . 96
6.14 TAG Database at LHC Startup . . . 97
6.15 Conclusions and Future Directions . . . 97

**7 ATLAS TAG Database Feasibility Study** **99**
7.1 Introduction . . . 99
7.2 Merging a TAG Database with ATLAS Distributed Data Management . . 100
7.3 Datasets in DQ2 vs Files in TAG Database . . . 101
7.3.1 Important terms . . . 102
7.4 Understanding the DQ2 Catalog Schema when implementing TAGs . . . 102
7.5 Adding a Dataset attribute to a TAG . . . 103
7.6 VUID in TAG Tests . . . 103
7.6.1 VUID lookup vs VUID in TAGs . . . 103
7.6.2 VUID DQ2 lookup vs VUID in TAG, tests throughout day . . . 105
7.6.3 Impact on overall time as a result of VUID lookup . . . 106
7.7 Subscription methods . . . 107
7.7.1 Complete Subscription . . . 108
7.7.2 Incomplete Subscription . . . 110
7.8 Comparing performance of subscription methods . . . 110
7.8.1 Results . . . 113
7.8.2 The hidden parameters . . . 114
7.9 Catalog Schema . . . 114
7.9.1 Schema tests . . . 115
7.9.2 t_pfn vs t_files for a single query . . . 115
7.9.3 Single vs Bulk query . . . 116
7.9.4 LFN vs LFN and VUID . . . 117
7.9.5 Increasing number of rows in the DQ2 Catalogs . . . 119
7.9.6 IN clause performance . . . 120
7.9.7 Production background query rates . . . 120
7.10 Distributed Data Management and Event Level Metadata Studies Outcomes 123
7.11 Trigger and TAGs . . . 123
7.11.1 Event and Run Level Metadata . . . 123
7.11.2 Time Varying Trigger Menus . . . 124
7.11.3 Size of Trigger Data . . . 124
7.11.4 Trigger Decoding in TAGs . . . 125
7.12 Conclusions and Future Directions . . . 126

**8 Scalability and Performance of a Terabyte TAG database** **127**
8.1 Introduction . . . 127
8.2 Terabyte TAG Database Performance and Scalability . . . 128
8.3 Demands on a TAG Database . . . 129
8.3.1 A Challenging Environment . . . 129
8.3.2 A Challenging User . . . 130
8.3.3 Challenging query patterns . . . 130
8.4 A Terabyte TAG Database . . . 130
8.4.1 Test Architecture . . . 131
8.5 Challenges of 1TB data . . . 131
8.6 Partitioning . . . 134
8.6.1 Partition keys . . . 134
8.6.2 Horizontal Partitioning . . . 135
8.6.3 Vertical Partitioning . . . 135

8.6.4 Horizontal Partitioning Solution for 1TB . . . 136
8.6.5 Vertical Partitioning of 1TB database . . . 137
8.6.6 Complete Partitioning Solution for 1TB . . . 137
8.7 Indexing . . . 137
8.7.1 Indexing solutions and experience for 1TB . . . 138
8.8 The Oracle Optimiser . . . 139
8.8.1 Optimising the Oracle Optimiser . . . 139
8.8.2 Optimiser Hints . . . 140
8.8.3 Query Hints Solution for 1TB . . . 140
8.9 Assessing performance . . . 141
8.9.1 Defining queries to assess performance of terabyte scale database . . . 141
8.9.2 1. What does the query return? . . . 141
8.9.3 2. What percentage of the table does the query return? . . . 141
8.9.4 3. Which attributes are defined in the query? . . . 142
8.10 Assessment of 1TB Performance . . . 142
8.11 An Extreme Performance Case . . . 148
8.12 Stress tests . . . 149
8.13 Performance Tests of Terabyte TAG Database at Tier One . . . 154
8.14 Conclusions and Future Directions . . . 160

**9 An Event Level Selection Service Interface - ELSSI 162**
9.1 Introduction . . . 162
9.2 ELSSI . . . 163
9.3 ELSSI Design Principles . . . 164
9.4 ELSSI design . . . 165
9.5 ELSSI to Relational TAG Database . . . 167
9.6 ELSSI Output . . . 168
9.7 ELSSI and AMI . . . 169
9.8 ELSSI and Trigger Decisions . . . 169
9.9 ELSSI Security . . . 169
9.10 ELSSI Query Creation . . . 170
9.11 Counts and Retrieves . . . 171
9.12 ELSSI Optimal Query Processing . . . 172
9.13 ELSSI Query Monitoring . . . 173
9.14 ELSSI saved sessions . . . 173
9.15 ELSSI and TAG Value Distibutions . . . 173
9.16 ELSSI at Tier 0 and Tier 1 . . . 173
9.17 ELSSI Streams version . . . 174
9.18 ELSSI FDR version . . . 178
9.19 Conclusions and Future Directions . . . 179

**10 Neural Net Analysis of $t\bar{t}H, H \rightarrow b\bar{b}$ 181**
10.1 Introduction . . . 181
10.2 Events in the $t\bar{t}H, H \rightarrow b\bar{b}$ channel . . . 184
10.2.1 Signal events . . . 184
10.2.2 Background Events . . . 184
10.3 $t\bar{t}H, H \rightarrow b\bar{b}$ Recent Analysis . . . 185

10.4 Event Characteristics Collection One . . . 188
10.5 Event Characteristics Collection Two . . . 189
10.6 Datasets . . . 191
10.7 Event Preselection . . . 193
10.8 Selection and Reconstruction . . . 194
10.8.1 Higgs Input Variables . . . 194
10.8.2 Generic Input Variables . . . 194
10.9 Neural Network Analysis . . . 195
10.9.1 Learning . . . 196
10.9.2 Layout . . . 197
10.9.3 Selection of Event Characteristic Input Collection . . . 198
10.9.4 Variable Distributions . . . 202
10.9.5 Importance of Variables . . . 206
10.9.6 Results . . . 209
10.10Conclusions and Future Directions . . . 211

**11 Conclusions** **212**

**A Analysis using Artifical Neural Nets** **215**
A.1 Neural Net . . . 215

**B Preliminary Studies for TAG Database Scalability** **219**
B.1 Horizontal Partitioning Studies . . . 219
B.2 Vertical Partitioning Studies . . . 222

# Author's Declaration

This thesis presents work performed from 2005-2008 in the Experimental Particle Physics Group in the Department of Physics and Astronomy at the University of Glasgow and at Conseil Européen pour la Recherche Nucléaire, CERN, in Geneva, Switzerland. The TAG Database system described in Chapter 5 is a project in which I am a central member and developer as part of the Event Level Metadata Team, along with David Malon[1], Jack Cranshaw[1], Julius Hrivnac[3], Qizhi Zhang[1], Marco Mambelli[2], Florbela Tique Aires Viegas[5], Elizabeth Gallas[4], Caitriana Nicholson[6] and Mike Kenyon[6]. Chapter 6 describes work on the Distributed Data Management system performed with the Event Level Metadata team, in particular Caitriana Nicholson, and work on the Trigger system performed with the Event Level Metadata team, in particular Qizhi Zhang. Chapter 7 describes Scalability work performed with the Event Level Metadata team, in particular Florbela Tique Aires Viegas and Luc Goosens[5]. Chapter 8 describes work performed in the development of the ELSSI browser with the Event Level Metadata team. The analysis in chapter 9 uses a common analysis code developed by Chris Collins-Tooth[6], with my own developments to suit a neural net analysis, and an analysis framework developed by Stan Thompson[6] and Rick St-Denis[6], again with developments for my own analysis. Other than where specifically referenced, all results presented throughout this thesis are my own.

[1]Argonne National Laboratory, Chicago, USA

[2]University of Chicago, Chicago, USA

[3]Laboratoire de l'Accélérateur Linéaire, Orsay, Université Paris Sud 11, Paris, France

[4]University of Oxford, Oxford, UK

[5]Organisation Européenne pour la Recherche Nucléaire, CERN, Geneva, Switzerland

[6]University of Glasgow, Glasgow, UK

# Chapter 1

# Introduction

In 2008 the LHC collider at CERN, Geneva, circulated first beams and the ATLAS Collaboration saw first beam data. One of the central physics goals of the ATLAS experiment is to search for the Higgs boson and to understand the mechanism through which particles aquire mass. In the current understanding of particle physics, the Standard Model, the mechanism by which this happens leads to the existence of the Higgs boson. The Higgs particle is theoretically postulated but is as yet to have its existence confirmed by experimental observation. The LHC will search for the Higgs boson over all potential masses up to 1 $TeV$. This thesis presents analysis of a potential discovery channel in the low mass range, $t\bar{t}H, H \rightarrow b\bar{b}$, using a neural net method to investigate improvements to the discovery potential.

In the search for new and rare processes such as Higgs events, the ATLAS experiment will produce data at rate and volume unprecedented in high energy particle physics. Analysts studying these data are internationally located, so the data is to be accessible to physicists at international collaborating research organisations. A central challenge for the ATLAS collaboration lies in development of the computing capacity to manage an unprecedented data challenge, an anticipated annual data volume of the order of ten petabytes, to handle this information and to make data available to analysts across the globe in a fluid, sound and transparent way.

The Event Level Metadata System is a means for analysts to interact with globally

distributed data and to perform studies and selection of physics events for study and analysis. The Event Level Metadata system is terabyte is size, globally distributed and must perform accurately and respond rapidly to the needs of analysts. This thesis presents the Event Level Metadata system and studies and developments performed with respect to the system within the ATLAS computing and analysis environment.

An overview of the Standard Model is given in Chapter Two with a theoretical introduction to the Higgs Mechanism. Higgs production and decay channels across the potential Higgs mass range and the discovery potential for a Standard Model Higgs at ATLAS are reviewed. The LHC project, accelerator and physics motivations are described in Chapter Three. The LHC Startup in 2008 is presented, the four LHC experiments and experiments leading to the LHC are introduced. Chapter Four focuses on the ATLAS Collaboration, the detector and the first experimental data of 2008. The ATLAS Computing System is introduced in Chapter Five and the Event Level Metadata System, the focus of the developments and studies in this thesis, is presented in Chapter Six. Feasibility studies for merging an Event Level Metadata system with the ATLAS Distributed Data Management system and the ATLAS Trigger performed in early Event Level Metadata developments are presented in Chapter Seven. Scalability studies and developments for the Event Level Metadata system, a central challenge in system development, are presented in Chapter Eight. The Event Level Metadata Interface, ELSSI, is presented in Chapter Nine. A neural net analysis of the channel $t\bar{t}H, H \rightarrow b\bar{b}$, Higgs production with an associated top quark pair, where the Higgs decays to $b$ quarks, is presented in Chapter Ten. The current understanding of the channel is presented and a neural net is investigated for identification of signal from background events, to improve significance of the low mass Higgs channel.

# Chapter 2

# The Standard Model

## 2.1 Introduction

The Standard Model is the current theory describing the universe in particle physics terms. A fundamental aim of the LHC and ATLAS is to study the Standard Model, to improve understanding of the model and to complete the picture by studying as yet unobserved particles in the model, namely the Higgs boson. This chapter describes the Standard Model in its current form, the Higgs mechanism and describes Higgs searches at the LHC, [1], [2], [3] and [5] and the theory and phemomonology of the Standard Model are introduced. The fundamental particles and forces that make up the universe within the Standard Model are first described in terms of fermions and bosons, then extentions to the Standard Model, Supersymmetry and Gauge Coupling Unification, are discussed. Field Theories of the Standard Model, Quantum Electro Dynamics for Electromagnetism, Quantum Chromodynamics for the Strong force and Quantum Field Theory for the Electroweak force will be described and used to motivate Spontaneous Symmetry Breaking of the Electroweak theory. It is this symmetry breaking that leads to the Higgs Mechanism within the Standard Model. The Higgs Mechanism, the Higgs potential and the Higgs boson are then discussed. The potential methods for discovery of a Standard Model Higgs boson at the ATLAS experiment at the LHC are introduced, in particular the low mass Higgs discovery methods and potential.

## 2.2 Particles and Forces

The Standard Model is the theory describing our current understanding of the universe in terms of its constituent fundamental particles and the interactions, or forces, between them. The Standard Model describes almost all experimentally observed results. All matter and interactions in the universe are described in terms of fundamental point like particles, fermions and bosons, with internal angular momentum characterised by spin quantum number. Fermions are particles of half integer spin. Bosons are particles of integer spin and mediate the interactions between fermions. Antiparticles are fundamental particles of equal mass and opposite charge of a particle, the positron is the antiparticle of the electron. Fermion and antifermions can only be created or annihilated in pairs, as antiparticle existence holds for fermions and bosons, but a conservation law holds for fermions. Antiparticles are denoted by the bar notation, for example an antiproton is denoted $\overline{p}$.

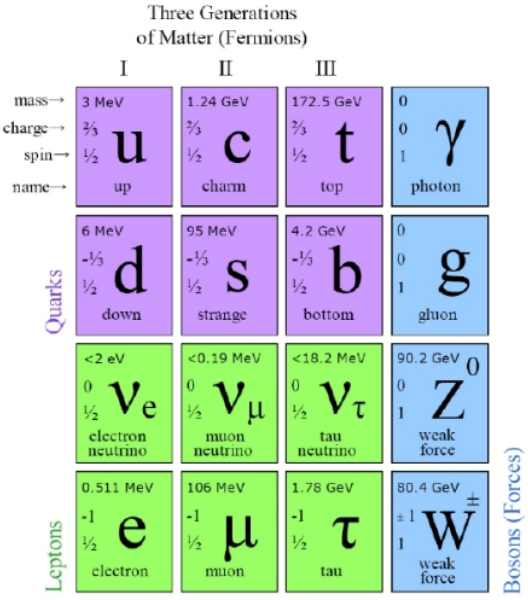



*Figure 2.1: Fundamental particles in the Standard Model, Fermions and Bosons, Mass, Charge and Spin [6]. Particle masses are described in more detail in tables 2.1 and 2.2*

## 2.3 Fermions

All matter is made of fundamental (pointlike) particles called fermions. Fermions have half-integer spin, $\frac{1}{2}h$, $\frac{3}{2}h$, $\frac{5}{2}h$..... In the fermion group are six quarks $q$ = d,u,s,c,b,t and six leptons $l = e, \nu_e, \mu, \nu_\mu, \tau, \nu_\tau$. Figure 2.2 shows the fundamental fermions of the Standard Model, grouped into three families of doublets.

$$\begin{pmatrix} e \\ \nu_e \end{pmatrix}_L \begin{pmatrix} \mu \\ \nu_\mu \end{pmatrix}_L \begin{pmatrix} \tau \\ \nu_\tau \end{pmatrix}_L$$
$$\begin{pmatrix} u \\ d \end{pmatrix}_q \begin{pmatrix} c \\ s \end{pmatrix}_q \begin{pmatrix} t \\ b \end{pmatrix}_q$$

*Figure 2.2: Fundamental particles in the Standard Model, Quarks and Leptons. Both quarks and leptons are grouped into three families of doublets*

Leptons have positive, negative or neutral charge. Neutral leptons denoted by $\nu$ are called neutrinos and are paired by flavour with the corresponding charged lepton. Quarks carry charges of $+\frac{2}{3}$ or $-\frac{1}{3}$. Ordinary matter is made of first generation quarks and leptons, second and third generations are seen in high energy physics experiments and cosmic ray events. Second and third generation quarks and leptons decay rapidly. Z boson decay studies at LEP found that there are three generations of light neutrinos, with mass less than half the Z mass, suggesting that there are exactly three generations [7].

Leptons are found in individual particle states whilst quarks are seen confined in bound states. Protons and neutrons are made of $u$ and $d$ quarks in the states $uud$ and $ddu$ respectively. Quarks are bound in states of three quarks and quark antiquark pairs, Bayrons are $qqq$ and Mesons are $q\bar{q}$, collectively baryons and mesons are called Hadrons. Quarks are not seen in nature outside of confined states. In high energy collisions such as those at the LHC, quarks can temporarily be separated from the gluons binding them into hadrons.

Tables 2.1 and 2.2 show the fundamental quarks and leptons of the Standard Model,

mass, charge, spin and family, values taken from [8]

| Symbol | Name | Mass | Charge | Spin | Family |
|---|---|---|---|---|---|
| u | up | 1.5-3.3 MeV | $\frac{2}{3}$ | $\frac{1}{2}$ | 1 |
| d | down | 3.5-6.0 MeV | $-\frac{1}{3}$ | $\frac{1}{2}$ | 1 |
| c | charm | $1.27^{+0.07}_{-0.11}$ GeV | $\frac{2}{3}$ | $\frac{1}{2}$ | 2 |
| s | strange | $104^{+26}_{-34}$ MeV | $-\frac{1}{3}$ | $\frac{1}{2}$ | 2 |
| t | top | $171.2 \pm 2.1 GeV$ | $\frac{2}{3}$ | $\frac{1}{2}$ | 3 |
| b | bottom | $4.20\ ^{+0.17}_{-0.07}$ GeV | $-\frac{1}{3}$ | $\frac{1}{2}$ | 3 |

*Table 2.1: The Fundamental Quarks of the Standard Model, Mass, Charge, Spin and Family*

| Symbol | Name | Mass | Charge | Spin | Family |
|---|---|---|---|---|---|
| $e$ | electron | $0.510998910 \pm 0.000000013 MeV$ | $e$ | $\frac{1}{2}$ | 1 |
| $\nu_e$ | electron-neutrino | $< 2$ eV at 95 % CL | 0 | $\frac{1}{2}$ | 1 |
| $\mu$ | muon | $105.658367 \pm 0.000004 MeV$ | $e$ | $\frac{1}{2}$ | 2 |
| $\nu_\mu$ | muon-neutrino | 0.19 MeV at 90 % CL | 0 | $\frac{1}{2}$ | 2 |
| $\tau$ | tau | $1776.84 \pm 0.17 MeV$ | $e$ | $\frac{1}{2}$ | 3 |
| $\nu_\tau$ | tau-neutrino | $< 18.2$ MeV at 95 % CL | 0 | $\frac{1}{2}$ | 3 |

*Table 2.2: The Fundamental Leptons of the Standard Model, Mass, Charge, Spin and Family*

## 2.4 Bosons

Interactions between fermions are described as exchange of integral spin boson particles, said to mediate the interactions. Bosons are particles of integer spin, $0, h, 2h...$ and called force carriers. There are four fundamental interactions of forces, these are Strong, Weak, Electromagnetic and Gravitational.

Strong interactions bind quarks into bound states and are mediated by massless gluons. Electromagnetic interactions are mediated through exchange of photons. Weak interactions are mediated by massive $W$ and $Z$ bosons and Gravitational force is mediated by a spin 2 boson, the graviton, as yet unobserved.

| Symbol | Name | Mass | Charge | Spin | Interaction |
|---|---|---|---|---|---|
| $\gamma$ | photon | $< 10^{-18}$ eV | 0 | 1 | electromagnetism |
| $W^{\pm}$ | W-boson | 80.398 $\pm 0.025 GeV$ | $\pm 1$ | 1 | weak nuclear |
| $Z$ | Z-boson | 91.1876 $\pm 0.0021 GeV$ | 0 | 1 | weak nuclear |
| $g$ | gluon | 0 | 0 | 1 | strong nuclear |

*Table 2.3: The Fundamental Force Carriers of the Standard Model, Mass, Charge, Spin and Interaction*

Table 2.3 shows the fundamental force carriers of the Standard Model, Mass, Charge, Spin and Interaction, values taken from [8], and an unobserved Spin 0 Higgs boson, with mass ranges set by current experimental observations. The range of each force is proportional to the mass of the mediating boson. The electromagnetic force, mediated by massless photons, has an infinite range, the weak force acts over a range of $10^{-18}$m and the strong force, although mediated by massless gluons, acts over a $10^{-15}$m range, an effect of the self interactions of gluons causing the strength of the force to increase with distance.

## 2.5 Extending the Standard Model

The Standard Model successfully and succinctly describes experimental data gathered in recent years and is generally thought to be a useful and sucessful model. There are increasing thoughts and evidence that fundamental particle physics extends beyond the Standard Model or that at the least there are missing pieces within the model [4]. The Standard Model has many parameters which are chosen to fit the data. Gravitational force is not included in the model, neutrinos are assumed to be massless yet recent results suggest that in fact neutrinos are massive [11]. Grand Unification Theories attempt to address such questions within the Standard Model.

The Standard Model theory does not explain how fundamental particles aquire mass. In 1964 a theory explaining the existence of massive particles, as interactions mediated by bosons, was proposed [9], [10]. The bosons were named Higgs bosons after the physicist who proposed the theory. The LHC experiments and ATLAS in particular aim to confirm or disprove experimentally the existence of such a mechanism.

## 2.6 Grand Unification Theories

Grand Unification Theories attempt to unify the electroweak, strong, electromagnetic and gravitational fields into a single theory, suggesting that although the fields lack symmetry and appear distinct at low energies, the four seemingly different forces or fields are in fact aspects of the same unified field and are seen to be a single unified field at high energies.

Grand unification is based on the idea that at high energies, all symmetries have the same gauge coupling strength, which is consistent with the speculation that they are really different manifestations of a single overarching gauge symmetry. Figure 2.3 shows Gauge Coupling for the Standard model on the left, where fields do not unify as energy increases and Gauge Coupling for the Minimal Supersymmetric Standard Model, MSSM, an supersymmetric extension to the Standard Model, right, taken from [8]. The Minimal Supersymmetric Standard Model is the minimal extension to the Standard Model to

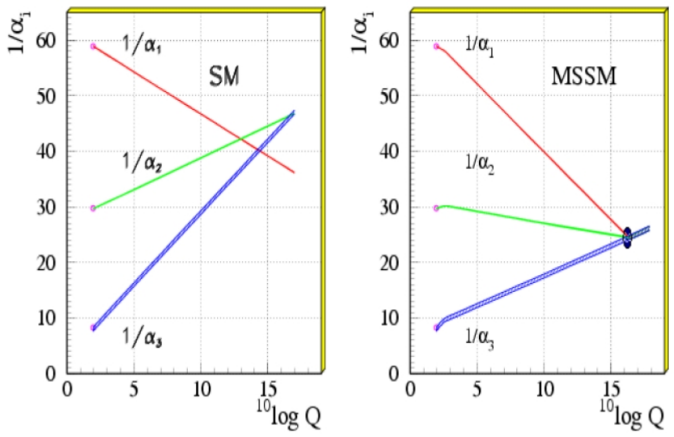



*Figure 2.3: Gauge Coupling Unification for non-SUSY, left, and SUSY Gauge Coupling Unification, right, using LEP data, showing the unification of fields when minimal supersymmetric extentions are added to the Standard Model changing the slope of the $\beta$ function, relating the coupling constant and strength of an interaction, so that fields converge at high energies*

include supersymmetry.

The slope of the lines in figure 2.3 represent the $\beta$ function. A $\beta$ relates a coupling constant $g$ and the energy scale $\mu$ of an interaction. It is defined by the relation $\beta(g) = \mu\frac{\partial g}{\partial \mu}$. The coupling constant $g$ represents the strength of an interaction. The $\beta$ function is a result of the virtual particles in an interaction. In a supersymmetric model more particles are available to contribute as particles have supersymmetric partners, the $\beta$ function changes and the slopes of the $\beta$ function changes so that the forces converge. So in MSSM theory, when supersymmetric extentions are made to the Standard Model, it can be seen that the fields converge at high energies.

## 2.7 Supersymmetric partners

Supersymmetry theory extends beyond the Standard Model and postulates that at high energies of TeV order there is a symmetry between fermions and bosons, so that every

fermion has a supersymmetric boson partner and each boson a supersymmetric fermion partner. Fermions obey Fermi-Dirac statistics and bosons obey Bose-Einstein statistics, the wavefunction $\Psi$ describes an ensemble of particles and the statistics obeyed by particles describe how the wavefunction $\Psi$ behaves in exchange of identical particles. $\Psi$ is symmetric in exchange of identical bosons and antisymmetric in exchange of identical fermions.

This thesis focuses on physics within the Standard Model and the search for a Standard Model Higgs boson. Grand Unification Theories, Supersymmetry and other extentions to the Standard Model are discussed in more detail in [8].

## 2.8 Gauge Theory and Symmetries

The Standard Model describes interactions as gauge theories to motivate the need for force carriers. A gauge theory is one that has invariance under local space-time dependent transformations. By requiring invariance of a system under a set of local transformations, gauge theories require that transformations depend on position in space and time. The Lagrangian describes the equations of motion of a system. In gauge theory, the Lagrangian of the system is locally as well as globally invariant. Invariance of the Lagrangian under local transformation gives rise to conserved quantities. The Standard Model Lagrangian is invariant under local gauge transformations of the symmetry groups $\mathrm{SU}(2)_L \otimes U(1)_Y$ and $\mathrm{SU}(3)_C$. The Electroweak Lagrangian corresponds to a $\mathrm{SU}(2)_L$ symmetry describing rotations in weak isospin space and $\mathrm{U}(1)_Y$ representing hypercharge transformations. The Strong Force is based on $\mathrm{SU}(3)_C$.

The Standard Model has three quantum field gauge theories, one for each interaction. For Elecromagnetism, the field theory is Quantum Electro Dynamics, QED, for the Strong force, the field theory is Quantum Chromo Dynamics, QCD. For the Electroweak force, the field theory is called Electoweak Field theory, and includes Spontaneous Symmetry Breaking in order to allow $W$ and $Z$ bosons to have mass. Electromagnetic, Weak and Strong interactions, when treated as gauge theories, give rise to the photon, whose ex-

change mediates electromagntic interactions, to three bosons, $W^{\pm}$ and $Z$ that mediate weak interactions, and to eight gluons that mediate strong interations.

We now discuss the field theories for Electromagnetism, Strong and Electroweak forces and Spontaneous Symmetry Breaking. Spontaneous Symmetry Breaking is a theory that allows a system to break symmetry in a vacuum state whilst symmetry is held through the rest of the system. The theory leads to the existence of a Standard Model Higgs boson and the Higgs mechamism, a theory that explains why matter in the universe has mass.

| Interaction | Field Theory | Boson(s) |
|---|---|---|
| Electromagnetic | Quantum Electro Dynamics | $\gamma$ |
| Strong | Quantum Chromo Dynamics | $g$ |
| Electroweak | Electroweak Field Theory | $W^{\pm}, Z^0$ |

*Table 2.4: The Quantum Field Theories of the Standard Model and the bosons associated with each field*

Table 2.4 shows the Quantum Field Theories of the Standard Model and the bosons associated with each field.

## 2.9 Field theory for Electromagnetic interactions, QED, U(1) abelian group

For the Electromagnetic interaction the field theory is Quantum Electro Dynamics, QED[1]. The Lagrangian of a free Dirac fermion, where the fermion field is $\psi$ is

$$L = \overline{\psi}(i\gamma^{\mu}\partial_{\mu} - m)\psi \tag{2.9.1}$$

[1]Descriptions of Field Theories taken from [5]

where $m$ is the mass of the fermion and $\gamma^\mu$ are gamma matrices. This Lagrangian density is invariant under transformation of the fermion field

$$\begin{aligned} \psi &\rightarrow e^{iQw}\psi \\ \overline{\psi} &\rightarrow e^{iQw}\overline{\psi} \end{aligned} \tag{2.9.2}$$

where $Q$ is the charge operator and $w$ is a real constant. The U(1) group is a group of all unitary matrices (where inverse equals adjoint), and the group is abelian as any two members of the group commute. In the transformation of the fermion field, we see that the set of all numbers $e^{-iw}$ are an abelian group, as $e^{-iw_1}e^{-iw_2} = e^{-iw_2}e^{-iw_1}$. So by saying that the Lagrangian density is invariant under transformation of the fermion field, we are saying that the Lagrangian is invariant under global U(1) transformations.

For invariance under gauge invariance to be a fundamental property of nature, the Lagrangian must also be invariant under local transformations. A local transformation is one in which $w$ has space time dependence. Local space time dependent transformations are called Gauge transformations. When $w$ depends on a space-time variable $x$, the field transformation is

$$\begin{aligned} \delta\psi(x) &= iw(x)\psi(x) \\ \delta\overline{\psi}(x) &= iw(x)\overline{\psi}(x) \end{aligned} \tag{2.9.3}$$

The Lagrangian however is not invariant under these transformations, due to the partial derivative, which causes the Lagrangian to vary by

$$\delta L = -\overline{\psi}(x)\gamma^4[\partial_\mu Qw(x)]\psi(x) \tag{2.9.4}$$

To restore gauge invariance we assume that the fermion field interacts with a vector field

$A_\mu$, a gauge field with interaction term

$$-e\overline{\psi}\gamma^\mu A_\mu Q\psi \tag{2.9.5}$$

so the Lagrangian is

$$L = \overline{\psi}(i\gamma^\mu(\partial_\mu + ieQA_\mu) - m)\psi \tag{2.9.6}$$

Assuming that the gauge field, like the fermion field, changes according to

$$\begin{aligned} -eQA_\mu &= -eQ(A_\mu + \partial A_\mu(x)) \\ &= -eQA_\mu + Q\partial_\mu w(x), \end{aligned} \tag{2.9.7}$$

this change cancels with $\delta L$, gauge invariance is restored. $L$ is now the fermion part of the QED Lagrangian and $A_\mu$ is a photon field.

Finally we introduce a kinetic term for the field $A_\mu$, field strength $F_{\mu\nu} = \delta_\mu A_\nu - \delta_\nu A_\mu$ and the Lagrangian density is

$$L = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \overline{\psi}(i\gamma^\mu(\partial_\mu + ieQA_\mu) - m)\psi \tag{2.9.8}$$

We cannot add a mass term to the Lagrangian as this would make the Lagrangian not invariant under gauge transformations and make the photon massive. As this Lagrangian density does not contain a mass term we correctly predict a massless mediating gauge boson, the photon.

When we demand invariance under the local gauge transformation (that is, demanding invariance under U(1) gauge transforms) we must introduce a vector or gauge field $A_\mu$. The gauge field is then associated with the photon field coupling to the fermion. To hold onto U(1) symmetry we introduce this massless field $A_\mu$, that we can interpret as a photon.

## 2.10 Gauge symmetry in non-abelian case

A non-abelian group is one for which transformation are non-commuting. To extend gauge theory to a non-abelian case, it is useful to define a covariant derivative

$$D_\mu = \partial_\mu + ieA_\mu \tag{2.10.1}$$

The Lagrangian then becomes

$$L = \overline{\psi}(i\gamma^\mu D - m)\psi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} \tag{2.10.2}$$

Abelian gauge theory can be extended to a nonabelian case by considering $n$ fermion fields $\Phi_i$. The Lagrangian density is

$$\begin{aligned} L &= \overline{\psi}^i(i\gamma^\mu\partial_\mu - m)\psi_i \\ &= \overline{\psi}^1(i\gamma^\mu\partial_\mu - m)\psi_1 + \overline{\psi}^2(i\gamma^\mu\partial_\mu - m)\psi_2 + ..... \end{aligned} \tag{2.10.3}$$

The Lagrangian is now invariant under transformations of a group of $n$ x $n$ matrices called SU(n), where $\psi \to U\psi, \overline{\psi} \to \overline{\psi}U^\dagger$ is an internal symmetry and $UU^\dagger$, $\det U = 1$. Matrices satisfying this are SU(n). An arbitrary SU(n) matrix is

$$U = e^{-i}\sum_{a=1}^{n^2-1} w^a T^a \tag{2.10.4}$$

and $T^a$ are the generators of the group. U(1) has a single generator.

Extending abelian gauge theory in this way extends the Lagrangian to be invariant under SU(n) transformations. A non-abelian gauge theory is one in which the Lagrangian is invariant under local transformations to a non-abelian group, this is achieved by introducing a gauge boson for each generator of the group. The partial derivative in the Lagrangian for the fermion field is replaced by a covariant derivative.

In a non-abelian case gauge bosons have self interaction, as observed for gluons in

QCD and $W^{\pm}$, $Z$ and photons. The difference to the abelian case is that cross terms appear in the derivative, indicating that the vector bosons are self interacting.

## 2.11 Gauge theory for strong force, SU(3)

For the Strong force, we use Quantum Chromo Dynamics, QCD, to describe interactions between quarks as a (non-abelian) gauge theory with the group SU(3). In QCD quarks are described by a field $\psi_i$ with $i = 1, 2, 3$ as the color quantum number. Eight gauge bosons are introduced to preserve local invariance. These are the eight gluons of the Standard Model. For QED and QCD, unbroken gauge theory predicts that gauge bosons be massless.

## 2.12 Gauge theory for electroweak force, SU(2)

The Electroweak Lagrangian corresponds to SU(2) symmetry. In extending gauge theory to electroweak interactions, the four gauge bosons required to maintain local invariance are massless. We know however that the force carriers in electroweak interactions, the $W^{\pm}$ and $Z$ bosons, are massive. So, to extend gauge theory to electroweak interactions and unify the three interactions of the Standard Model, we are led to spontaneous symmetry breaking and to extend electroweak theory, we have to break symmetry of gauge theories to allow the $W$ and $Z$ bosons to have mass. Spontaneous symmetry breaking allows existence of massive gauge bosons whilst maintaining local gauge invariance, and occurs when the Lagrangian of a system is invariant under a symmetry group but the symmetry breaks in the ground state.

## 2.13 The Higgs Mechanism

The Higgs mechanism is an extension of spontaneous symmetry breaking to allow the creation of massive vector bosons in a gauge invariant theory. In the Standard Model

Higgs theory, non-abelian electroweak theory is broken. Spontaneous symmetry breaking allows gauge bosons to have mass and at the same time local gauge invariance is held. We start with abelian theory and then extend this to a non-abelian theory.

The Higgs mechanism is demonstrated in terms of gauge theory by introducing a complex scalar field

$$\phi = \frac{(\phi_1 + i\phi_2)}{\sqrt{2}} \tag{2.13.1}$$

for which the Lagrangian is

$$L = (\partial_\mu \phi)^*(\partial_\mu \phi) - (\mu^2 \phi^* \phi + \lambda(\phi^* \phi)^2) \tag{2.13.2}$$

The first term is the kinetic part of the Lagrangian, the second is the potential $V(\phi)$. The Lagrangian is invariant under a local gauge transformation for U(1) if we apply

$$\phi \rightarrow e^{-i\alpha(x)}\phi \tag{2.13.3}$$

and replace $\delta_\mu$ by a covariant derivative

$$D = \partial_\mu - ieA_\mu \tag{2.13.4}$$

where the gauge field transforms as

$$A_\mu \rightarrow A_\mu + \frac{1}{e}\partial_\mu \alpha \tag{2.13.5}$$

The gauge invariant Lagrangian is then

$$L = (D_\mu \phi)^*(D_\mu \phi) - \mu^2 \phi^* \phi - \lambda(\phi^* \phi)^2 - \frac{1}{4} F_{\mu\nu} F^{\mu\nu} \tag{2.13.6}$$

where

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu \tag{2.13.7}$$

As $\lambda \leq 0$ would imply an unbounded potential, we assume $\lambda > 0$. When $\mu^2 > 0$, the minimum of the potential is uniquely at zero. When $\mu^2 < 0$ the potential is zero at $\phi = 0$ and has a minimum described by a circle in the $\phi_1\phi_2$ plane, with circle radius $\mu$. When $\mu^2 < 0$, there are an indefinite number of lowest energy states around this circle.

$$\begin{aligned} \phi_1 + \phi_2 &= \mu^2 \\ \mu^2 &= -\tfrac{\mu^2}{\lambda} \end{aligned} \tag{2.13.8}$$

The Higgs potential holds an electroweak symmetry and has a vacuum state in which the symmetry is broken along the lowest energy states. The Higgs potential $V(\phi)$ for a complex scalar field, assuming $\lambda > 0$ is shown in figure 2.4. On the left is $\mu^2 > 0$ and on the right is $\mu^2 < 0$ [5]. Figure 2.5 shows the shape of the Higgs potential for $\lambda > 0$, $\mu^2 < 0$ in three dimensions. The potential is tall in the centre and dips around the center before increasing as $\phi$ increases. The zero field configuration at the central point is unstable, so the system will fall into lower energy state. The lowest energy state, that is, the vacuum, is not empty and is permeated by the Higgs potential field. The Higgs theory says that if this field couples to other particles, it inhibits the motion of the particle to give it mass.

We can translate the field $\phi$ to a minimum energy position at $\phi_1 = \mu, \phi_2 = 0$ and define new fields, $\eta$ and $\xi$

$$\phi(x) = \sqrt{\frac{1}{2}}[\nu + \eta(x) + i\xi(x)] \tag{2.13.9}$$

where

$$\begin{aligned} \phi_1 &= \nu + \eta(x) \\ \phi_2 &= \xi(x) \end{aligned} \tag{2.13.10}$$

The Lagrangian can then be expanded about the vacuum in terms of these fields

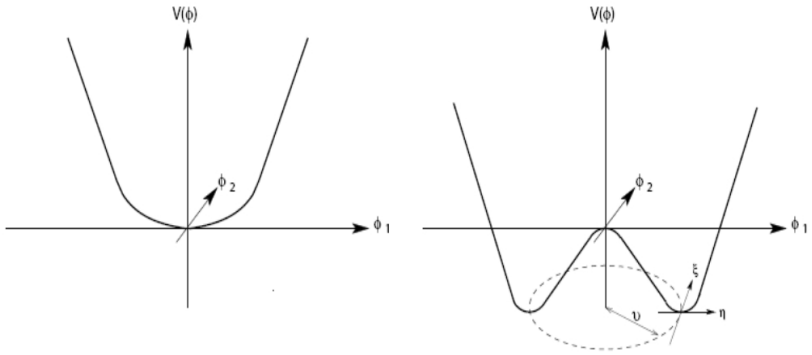



*Figure 2.4: The potential $V(\Phi)$ for a complex scalar field, $\lambda > 0$, $\mu^2 > 0$ on the left, $\mu^2 < 0$ on the right*

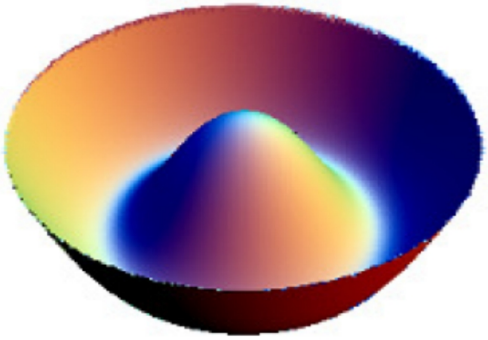


*Figure 2.5: The Higgs Potential in three dimensions, referred to as a Mexican hat shape, with a peak in the centre and dip all around, the distance from the centre represents the strength of the Higgs field and the zero field state, at the top of the peak, is unstable to perturbations, so that the system will move to lower energy state and the lowest energy state, the vacuum, is not empty*

$$L = \frac{1}{2}(\partial_\mu \xi)^2 + \frac{1}{2}(\partial_\mu \eta)^2 - \nu^2 \lambda \eta^2 + \frac{1}{2} e^2 \nu^2 A_\mu A^\mu$$
$$- e\mu A_\mu \delta^\mu \xi - \frac{1}{4} F_{\mu\nu} F^{\mu\nu} + ..... \qquad (2.13.11)$$

where $L$ also includes interaction terms. The Lagrangian now contains terms for a massive vector boson $A_\mu$, a massive scalar $\eta$ and a massless boson $\xi$. The massless boson is called a Goldstone boson.

$A_\mu$ now has an extra degree of freedom, as it has mass. Translating variables should not give extra degrees of freedom, and this suggests that the fields in the equation above are not distinct particles.

We therefore choose a gauge transformation to eliminate $\xi = \phi_2(x)$, taking the U(1) transformation in real and imaginary parts

$$\begin{aligned}
\phi \to \phi &= e^{-i\theta(x)}\phi \\
\phi \to \phi &= (\cos\theta(x) - i\sin\theta(x))(\phi_1 + i\phi_2) \\
&= (\phi_1\cos\theta(x) - \phi_2\sin\theta(x)) + i(\phi_1\sin\theta(x) - \phi_2\cos\theta(x))
\end{aligned} \tag{2.13.12}$$

so that the transform is

$$\theta = -\tan^{-1}\frac{\phi_2}{\phi_1} \tag{2.13.13}$$

and combined with the approximation

$$\begin{aligned}
\phi &= \sqrt{\frac{1}{2}}(\nu + \eta + i\xi) \\
&\approx \sqrt{\frac{1}{2}}(\nu + \eta)e^{-\frac{i\xi}{\nu}}
\end{aligned} \tag{2.13.14}$$

to the lowest order in $\xi$, we have a different set of real fields, $h, \theta, A_\mu$

$$\begin{aligned}
\phi &\to \sqrt{\frac{1}{2}}(\nu + h(x))e^{i\frac{\theta(x)}{\nu}} \\
A_\mu &\to +\frac{1}{e\nu}\partial_\mu\theta
\end{aligned} \tag{2.13.15}$$

so that we get a Lagrangian of two massive interacting particles, a vector gauge boson $A_\mu$ of mass $m_A = e$ and a massive scalar boson $h$ of mass $m_h = \sqrt{2\lambda\nu^2}$

$$L = \frac{1}{2}(\partial)\mu h)^2 - \lambda\nu^2 h^2 + \frac{1}{2}e^2\nu^2 A_\mu^2 - \lambda\nu h^3$$
$$-\frac{1}{4}\lambda h^4 + \frac{1}{2}e^2 A_\mu^2 h^2 + \nu e^2 A_\mu^2 h - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} \tag{2.13.16}$$

The Goldstone boson becomes an extra degree of freedom for the original gauge boson, allowing it to have mass. The Higgs mechanism is an introduction of a complex scalar field with two additional degrees of freedom, one goes to the $A_\mu$ boson so it may have mass and the other appears as a scalar boson $h$ with mass $m_h$, a Higgs boson.

## 2.14 Electroweak Lagrangian and Higgs Mechanism

The SU(2) group has three generators corresponding to gauge bosons $W_\mu^1, W_\mu^2, W_\mu^3$ and coupling $g$ and the U(1) group has a boson $B_\mu$ and coupling $g$. $\theta_W$ is a weak mixing angle. $W^\pm$, $Z$ and $\lambda$ are linear superpositions of gauge fields

$$W^\pm = \frac{W_\mu^1 \pm iW_\mu^2}{\sqrt{2}}$$
$$Z_\mu = \cos\theta_W W_\mu^3 - \sin\theta_W B_\mu$$
$$A_\mu = cos\theta_W B_\mu + sin\theta_W W_\mu^3 \tag{2.14.1}$$

$W^\pm, Z$ aquire mass through the Higgs mechanism. To break $SU(2)_L \otimes U(1)_Y$ symmetry, we introduce a doublet of complex fields and four degrees of freedom

$$\phi = \begin{pmatrix} \phi^+ \\ \phi^- \end{pmatrix} = \sqrt{\frac{1}{2}} \begin{pmatrix} \phi_1 + i\phi_2 \\ \phi_3 + i\phi_4 \end{pmatrix}$$

In spontaneous symmetry breaking three degrees of freedom are given to $W^\pm, Z$ to allow them to have mass, and the other to a scalar Higgs boson. The photon remains massless as the elecroweak lagrangian remains invariant under local U(1) transformations.

## 2.15 Fermion masses

An explicit mass term cannot be present in the Lagrangian for fermions as this would mix right handed and left handed states that must be treated separately for the weak interaction. It is possible to have interaction for a left handed fermion doublet, a right handed fermion singlet and a scalar doublet $\Phi$, the Higgs field.

These interactions are Yukawa interactions and are of the form

$$G_f = (\psi_L \Phi \psi_R + \psi_R \Phi \psi_L) \tag{2.15.1}$$

$G_f$ is the coupling constant of the interaction. Quarks and leptons aquire mass through coupling to the Higgs field, with mass proportional to the coupling.

## 2.16 The Higgs Boson

A Higgs boson observation is a central focus of the LHC. The properties of the Higgs boson are fixed by its mass. Once the Standard Model Higgs mass is known, all decay widths and production processes follow. In order to search for a Higgs boson experimentally, it is important to consider the constraints placed on the mass of the Higgs by previous experiments and the production and decay mechanisms within the mass range expected for the Higgs and within the reach of the LHC.

## 2.17 Constraints on the Higgs Boson mass

The mass range of the Higgs boson has been determined theoretically and experimentally in previous collider experiments.

LEP excluded a Higgs boson with a mass of less than 114 GeV at 95 percent probability, figure 2.6, [28].

The Tevatron combined results from CDF and D0 to suggest exclusion of Higgs mass of 170 GeV at 95 percent probability, figure 2.7 [6].

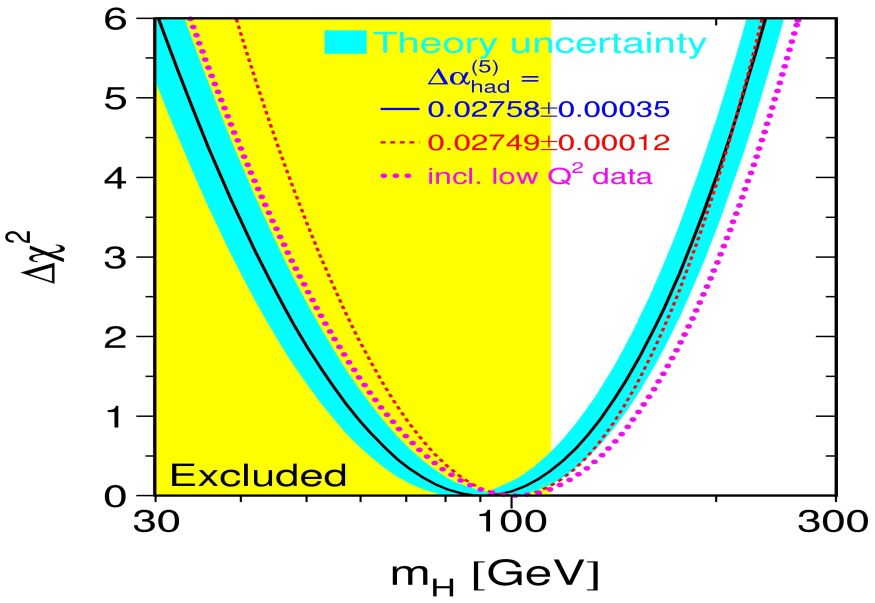



*Figure 2.6: Standard Model Higgs data at LEP, showing that gathered data excluded the existance of a Higgs boson up to a mass of 114 GeV at 95% probability*

## 2.18 Higgs Production and Decay at ATLAS

### 2.18.1 Production

There are multiple production mechanisms and decay processes for the discovery potential of the mass range for Higgs at LHC [29]. Many discovery scenarios are possible, depending on the mass of the Standard Model Higgs.

Higgs production mechanism at the LHC with the potential to lead to observable cross sections are

- Gluon-gluon fusion
- WW and ZZ fusion
- Associated production with W and Z
- Associated production with $t\bar{t}$

Figure 2.8 [29] shows the dominant production method at LHC is gluon fusion with the largest production cross section for the full mass range. At lower masses, associated

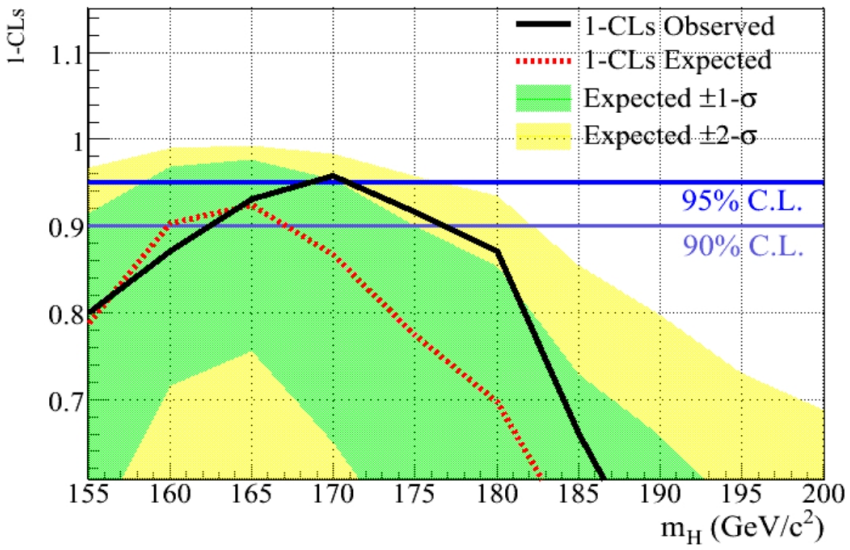



*Figure 2.7: Standard Model Higgs data at Tevatron, combining data from the CDF and D0 experiments to exclude a Higgs mass of 170 GeV at 170 GeV*

production with top quarks and $W/Z$ bosons are interesting and at higher masses $W$ and $Z$ boson fusion processes are interesting.

In this thesis, the $t\bar{t}H, H \rightarrow b\bar{b}$ channel is studied for a Higgs mass $m_H = 120$ GeV. At this mass value for the Higgs boson, associated production with top quarks is an interesting process, as the second dominant process after gluon fusion.

### 2.18.2 Decay

After Higgs production, the decay of the Higgs boson depends on its mass. The Higgs couples preferentially to heavier particles so it decays primarily to the highest mass particles allowed.

Figure 2.9 [29] shows Higgs branching ratios as a function of mass. For Higgs masses $m_H < 140GeV$, the preferred decay mode is $b\bar{b}$. At $m_H \approx 140GeV$, $WW$ and $ZZ$ decay are dominant. For a Higgs mass $m_H = 120$ GeV studied in this thesis, $b\bar{b}$ is the preferred decay process.

As the decay mechanism likely to be seen varies across the Higgs mass range, different search strategies are used for different Higgs masses. There are three mass ranges that

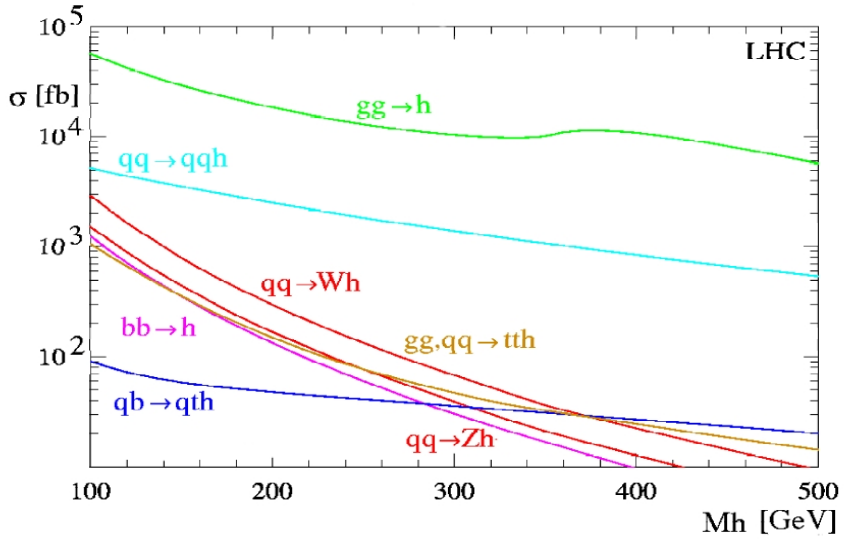



*Figure 2.8: Standard Model Higgs Production at the LHC, the dominant production method is gluon fusion with the largest production cross section over the full mass range, at lower masses associated production with top quarks and $W/Z$ bosons are interesting and at higher masses $W$ and $Z$ boson fusion processes are interesting*

can be defined for ATLAS in terms of differing search strategies accounting for varying dominant decay processes with varying Higgs mass.

### 2.18.3 $m_H < 130$ GeV

For the low Higgs mass range $m_H < 130$ GeV, the Higgs decays primarily to $b\bar{b}$. A large QCD background makes the signal from direct Higgs production difficult to extract, so when a Higgs is produced with a $t\bar{t}$ pair the search can require an isolated lepton from top decay in the event to identify signal events over background. The $t\bar{t}H, H \rightarrow b\bar{b}$ events are in this category. In the low mass range the channels $H \rightarrow \gamma\gamma$ direct production and $H \rightarrow \tau\tau$ vector boson fusion are the other events of interest.

### 2.18.4 130 Gev $< m_H <$ 180 GeV

In the medium Higgs mass range 130 GeV $< m_H <$ 180 GeV, the $WW$ decay mode dominates over $ZZ$. $H \rightarrow WW^* \rightarrow l^+l^-\nu\overline{\nu}$ has significance of $5\sigma$ over the full mass range.

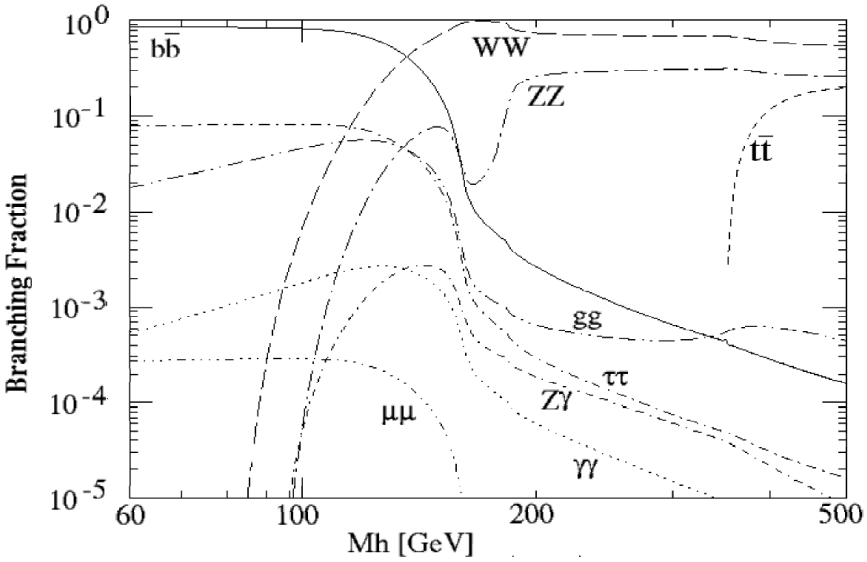



*Figure 2.9: Standard Model Higgs Decay showing Higgs brancing ratios as a function of mass, for $m_H < 140GeV$ the preferred decay mode is $b\bar{b}$, for $m_H \approx 140GeV$, $WW$ and $ZZ$ decay are dominant and for $m_H = 120$ $GeV$ $b\bar{b}$ is the preferred process*

### 2.18.5 $180$ GeV $< m_H < 1$ TeV

For the upper Higgs boson mass range 180 GeV $< m_H <$ 700 GeV, the $H \rightarrow ZZ \rightarrow 4l$ is the most reliable signal for potential Higgs discovery. For this channel, the signal is smaller than the background, a continuum production of ZZ boson pairs. The leptons in the final state have high momenta and detection does not require demanding performance by the detector. Available integrated luminosity will define the discovery potential of the channel. At $m_H >$ 800 GeV, the rate of $H \rightarrow ZZ \rightarrow 4l$ becomes too low to be used, so instead Higgs searches at this mass range look for neutrinos and jets in the final state and $H \rightarrow ZZ \rightarrow ll\nu\nu$ and $H \rightarrow WW \rightarrow l\nu jj$ are the processes of most interest.

## 2.19 Discovery Potential

Discovery potential is determined by expected significance, $\sigma$, expected number of signal events divided by the square root of expected number of background events. Significance

is calculated using Monte Carlo data.

$$\sigma = \frac{\text{number of signal events}}{\sqrt{\text{number of background events}}} \tag{2.19.1}$$

A $\sigma$ value of 5 or above is classed as a discovery. In the analysis of the $t\bar{t}H, H \rightarrow b\bar{b}$ channel in this thesis we aim to increase the $\sigma$ value of the channel to improve discovery potential.

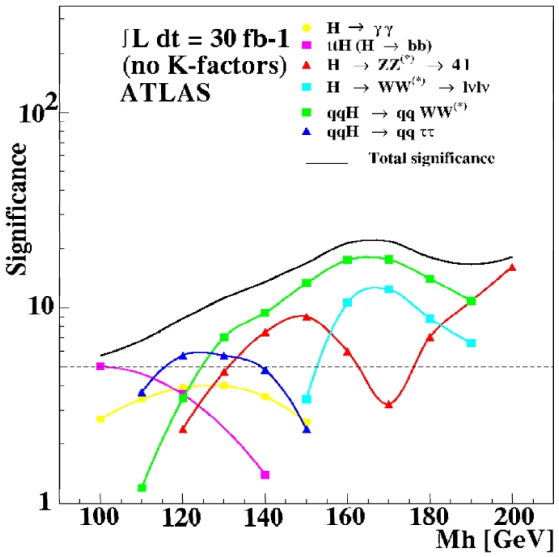



*Figure 2.10: The Significance of Higgs Search Channels in the LHC mass range, showing significance of interesting channels and total combined significances, demonstrating that Higgs boson searches are most challenging in the lower mass range*

Figure 2.10 shows that Higgs searches are most challenging in a lower mass range [18], as the significance of potential discovery channels in the lower mass range are lower, as is the total significance. For $m_H = 120$GeV, the channels of interest are $H^* \rightarrow 4l$, $qqH \rightarrow qqWW^*$, $H \rightarrow \gamma\gamma$ and $qqH \rightarrow qq\tau\tau$. $t\bar{t}H, H \rightarrow b\bar{b}$ is a $qqH \rightarrow qqWW^*$ process.

## 2.20 Summary

The Standard Model has been presented in this chapter. The fundamental particles and forces that make up the universe within the Standard Model have been introduced with

details of the fermions and bosons in the model. Limitations of the Standard Model, Supersymmetric theory and Gauge Coupling Unification have been introduced. Field Theories of the Standard Model, Quantum Electro Dynamics for Electromagnetism, Quantum Chromodynamics for the Strong force and Quantum Field Theory for the Electroweak force have been described and this has been used to motivate Spontaneous Symmetry Breaking of the Electroweak theory leading to the Standard Model Higgs Mechanism. The Higgs Mechanism, the Higgs potential and the Higgs boson have been detailed.

The potential methods for discovery of a Standard Model Higgs boson at the ATLAS experiment at the LHC have been introduced, with emphasis on the low mass Higgs discovery methods and potential, as an analysis for a Higgs mass $m_H = 120$ GeV is the focus of the analysis in this thesis.

We now move on to the Large Hadron Collider, LHC, the latest particle physics collider, at CERN in Geneva, where extensive studies of the Standard Model are planned.

# Chapter 3

# Large Hadron Collider

## 3.1 Introduction

The Large Hadron Collider at CERN, Geneva is the latest and most powerful collider in particle physics. First circulating beams were seen in 2008. In this chapter, the LHC is introduced, luminosity and collisions rate are discussed and the physics motivation and potential are outlined. The first beams are presented. The experiments and context of the developments of the LHC are outlined.

## 3.2 The LHC

The Large Hadron Collider, LHC, is a proton beam collision particle accelerator [12]. Using a 27km underground circular tunnel inherited from the LEP accelerator on the Swiss-French border at the Conseil Européen pour la Recherche Nucléaire, CERN [13] in Geneva, figure 3.1, the LHC accelerator produces, accelerates and collides proton beams. The accelerator feeds four experimental detectors, ATLAS, CMS, LHCb and ALICE, located at four intersection points throughout the LHC tunnel. Each of the LHC experiments has an international collaboration of scientists, supporting detector development, data gathering, computing infrastructure and physics analysis. The LHC project, through these four experiments, will facilitate new and unprecedented developments in detector

technologies, computing systems and physics discoveries.

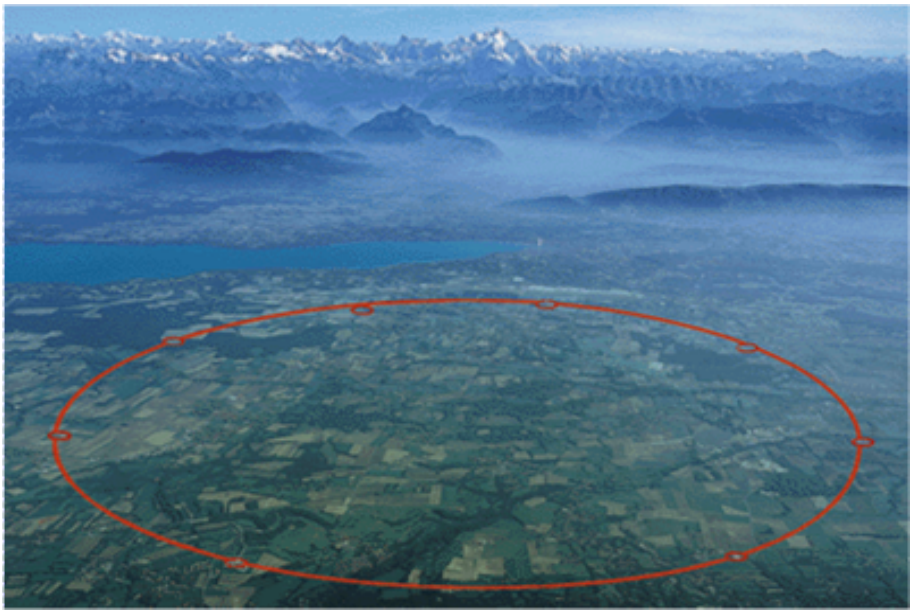


*Figure 3.1: The LHC in Geneva with Mont Blanc and Lac Leman, the circular tunnel runs for 27 km and particles are accelerated around the tunnel to speeds close to the speed of light before being collided at centre of mass energies of 14 TeV*

Physics events, where protons collide at an interaction point inside an experimental detector, allow physicists to search for signatures of as yet unseen processes, as well as measure and study known processes. The LHC offers an opportunity for new physics discovery due to a high energy and luminosity.

Accelerated proton beams are counter rotated and accelerated throughout the tunnel, ultimately with energy 7 TeV, so at beam collision point centre of mass energy is 14 TeV. Each beam is made of bunches of particles, with billions of protons in each bunch. When bunches of protons in beams cross, the majority of particles bypass each other, but some will collide, or interact, in a physics event, producing new particles and energy. A high centre of mass energy allows production of heavy particles.

Luminosity, a measure of the intensity of the accelerator, or the rate at which collisions or events take place, is an important feature in the search for rare processes, as a very high number of collisions are needed for rare processes to be seen and statistically verified and studied. Luminosity is increased by increasing the number of particles in each bunch, the rate of bunch crossings as well as by making the bunches as compact as possible at

the interaction point.

| LHC Beam Parameter | Value |
|---|---|
| Design Luminosity | $10^{34}$ cm$^{-2}$s$^{-1}$ |
| Particles Per Bunch | $1.15 \times 10^{11}$ |
| Bunches Per Beam | 2808 |
| Beam Turns Per Second | 11245 |
| Bunch Spacing | 25 ns |
| Bunch Crossing Rate | 40 MHz |
| Beam (Top) Energy | 362 MJ |
| Proton Injection Energy | 450 GeV |
| Proton Collision Energy | 7000 GeV |

*Table 3.1: Vital Statistics for LHC Beam*

Table 3.1 shows the vital statistics for the LHC beam. When the accelerator switches on for data taking, luminosity will be lower, allowing initial physics studies, detector calibration and accelerator improvements, before luminosity is increased to design luminosity in the subsequent months and years.

The graph in figure 3.2 shows the energy and luminosity for proton (anti)proton colliders, [14], left to right, Intersecting Storage Rings, ISR, the Super Proton AnitProton Synchrotron, Sp$\overline{p}$S, the Tevatron and the LHC, with a startup date of 2008. The graph demonstrates the significant increase in both energy and luminosity of hadron colliders in recent years, motivated by improving technology and the search for more complex are rare physics.

## 3.3 The LHC Accelerator

The LHC experiments require high energy proton beams travelling at almost the speed of light to collide at interaction points within the experimental detectors. At the LHC,

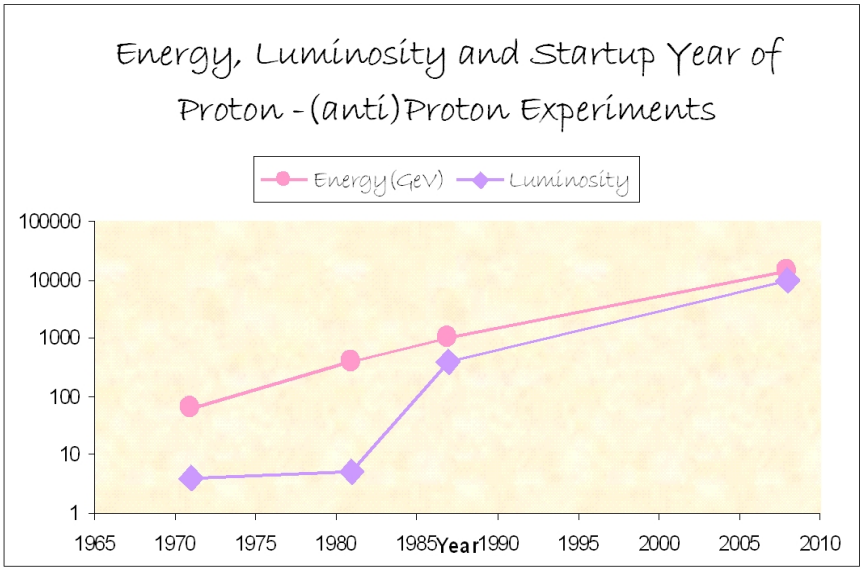



*Figure 3.2: Energy, Luminosity and Start Up year of proton and antiproton experiments, showing the significant increase in energy and luminosity of colliding beam accelerators in recent years*



*Figure 3.3: The LHC Accelerator Beampipe throughout which protons are circulated and accelerated, the beampipe is cooled to 1.9 K and superconducting magnets control the trajectory of the beam around the LHC circle*

figure 3.3 [12], protons are produced in a linear accelerator and particle accelerators are used to boost proton energy for injection into the LHC loop. Protons are produced at 50 MeV and are fed into a Proton Synchrotron Booster, a Proton Synchrotron and a Super Proton Synchrontron in turn, where proton energies are boosted to 1.4 GeV, 25 GeV and 450 GeV respectively.

Once proton energy has been increased, the particles are then injected into the LHC beampipe, a vacuum held at temperatures as low as 1.9 K, sufficiently low to allow superconducting magnets to operate at a magnetic field of 8 Tesla, capable of controlling and deflecting the direction of the 7 TeV proton beams around the LHC loop. Superconducting Radio Frequency cavities accelerate the proton beams by transfering the energy of radio frequency waves to the protons, which travel through a series of cavities, tuned to transfer maximum energy. In this way the superconducting cavities accelerate the 7 TeV beams to almost the speed of light, then maintain the 7 TeV beams in bunches of $10^{11}$ protons at 25 nanosecond intervals needed for design luminosity. The accelerator pushes the boundaries of engineering. The vital statistics of the LHC accelerator are shown in table 3.2.

## 3.4 The LHC Accelerator Complex

The LHC accelerator loop consists of eight arcs and eight insertions. An insertion is a line section in the accelerator, for beam collisions, beam injection, beam dumping or beam cleaning. The straight sections are centered on accelerator pits, where the experiments, radiofrequency, beam dumps are located and the arcs are arched and contain mainly dipoles. Each arc has 154 bending dipole magnets. A sector is a one eighth part of the LHC, starting in the center of a straight part and ending in the center of the next one. The eight sectors are the working units of the the LHC, powering of each sector is independent. An octant spans a straight section, from the middle of the arc on the left to the middle of the arc on the right.

The LHC Accelerator Complex is shown in figure 3.4 [12] and eight arcs/octants and

intersection points and the position of the experiment detectors in figure 3.5. As well as the arcs and intersection points, the accelerator complex includes the PS Booster, Proton Synchroton and Super Proton Synchrotron, the four experiment areas and control centers from which the accelerator and experiments are controlled. The Accelerator Complex runs through both Swiss and French territory.

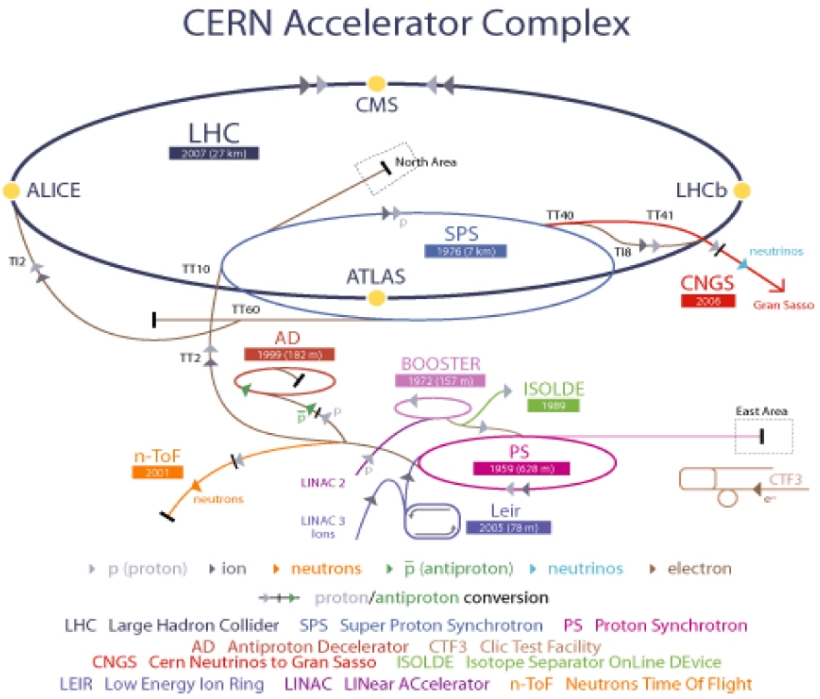



*Figure 3.4: The LHC Accelerator Complex, showing the LHC accelerator loop, the PS Booster, Proton Synchroton and Super Proton Synchrotron and the position of ATLAS, CMS, LHCb and ALICE*

## 3.5 LHC Startup 2008

In September 2008, a first full circulated beam was successfully guided around the full LHC loop. Beams were passed in both clockwise and anticlockwise directions. Several hundred orbits were successfully achieved. On day two, a beam was captured and circulated in an anticlockwise direction for thirty minutes.

In preparation for the LHC start, extensive commisioning tests were undertaken. Commisioning tests involved testing and ensuring successful working of the accelerator in terms

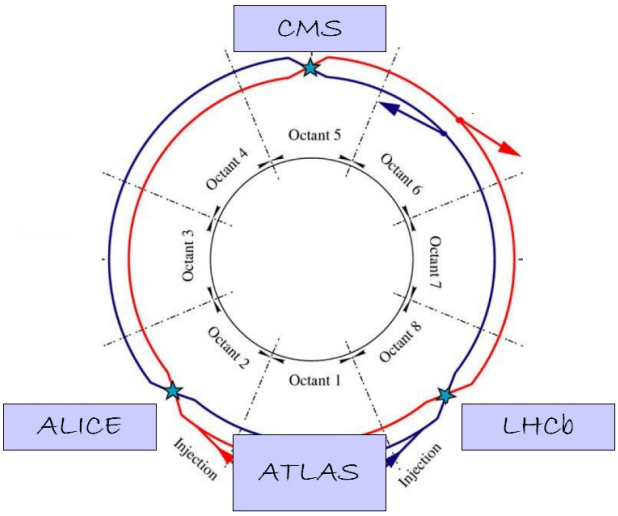



*Figure 3.5: The LHC Accelerator loop, showing the eight arcs/octants and intersection points and the position of the experiment detectors*

of cooling to working temperatures, testing of the magnets to ensure that they withstand working levels of current and Synchronisation tests with the Super Proton Synchrotron accelerator. These conditions are achieved before an attempt at a full revolution of beam through the LHC circle is realistic.

Synchronisation tests were performed in August in two stages, in early August synchronisation of a clockwise beam through the transfer system and into the accelerator was achieved and in late August the process was repeated for an anticockwise beam. On each occassion the beam, consisting of a single bunch of protons, was guided around 3km of the LHC loop, several hundred times. Beams were injected in the clockwise direction in the first week of August and clockwise in the last week of August. Point five, where CMS is located, was not crossed in the days before the first beam.

The first beams in the LHC circulated at injection energy 450GeV. The first beam can be seen in a cross-section of the beampipe [15], one dot is the beam being injection into the ring and the second is a recording of the beam returning to the start location after a full circulation.

After the first beams are achieved, the LHC is now prepared for circulation of higher

| LHC Accelerator Parameter | Value |
|---|---|
| Accelerator Circumference | 26659 m |
| Proton Injection Energy | 50 MeV |
| Proton Energy after PS Booster | 1.4 GeV |
| Proton Energy after Proton Synchrotron | 25 GeV |
| Proton Energy after Super Proton Synchrotron | 450 GeV |
| Proton Energy after LHC acceleration | 7 TeV |
| Acceleration time in LHC | 20 minutes |
| Average Depth Tunnel | 100 metres |
| Highest Depth Tunnel | 50 metres (towards Lac Leman) |
| Lowest Depth Tunnel | 175 metres (under Jura mountains) |
| Temperature of Accelerator | 1.9 K (-271.3 degrees Celcius) |
| Number of magnets | 9593 |
| Number of main dipoles | 1232 |
| Number of main quadrupoles | 392 |
| Vacuum in Beam pipe | $10^{-10}$ - $10^{-11}$ mbar |

*Table 3.2: Vital Statistics for the LHC Accelerator*

energy beams. Originally the aim was for beams of energy 5TeV in 2008 as well as collisions of beams travelling in opposite directions around the LHC circle. However a failure in part of the LHC occurred, thought likely as result of a fault in a connection between two magnets, leading to a helium leak, an increase in temperature and damage to the accelerator. As a result, plans for the next circulating beams have been moved to 2009, to give time for investigation of the fault and repairs, then cooling to operating temperatures.

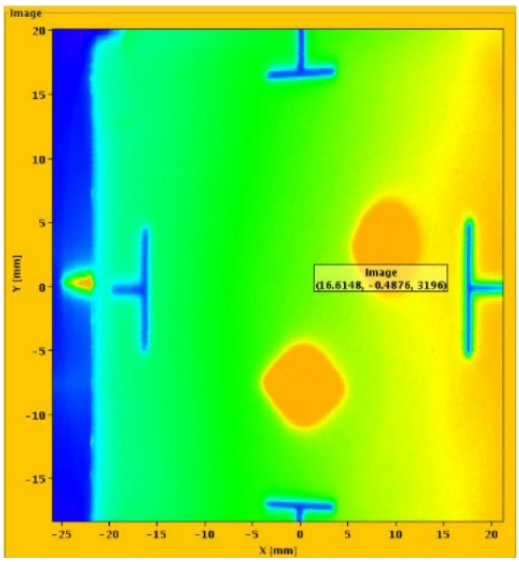



*Figure 3.6: A First Beam seen in the LHC in September 2008, a cross section of the beampipe shows the 0.45 TeV beam being injected into the LHC ring and returning to the start after a full circulation*

## 3.6 LHC Experiments

The LHC beam is used for four experiments and proton collision bunch crossings occur at four interaction points in the detectors across the LHC loop. ATLAS [15], [16], [17], [18] and CMS [19] are large general purpose detectors, whose aim is to detect and study a large range of physics processes, both within and beyond the Standard Model, including a search for and study of the postulated Higgs boson. By studying similar physics independently, ATLAS and CMS can verify and confirm physics results and discoveries. LHCb [20] and ALICE [21] are smaller experiments with detectors designed to study more specific physics. LHCb is motivated by study of the $b$ quark and to ultimately understand the apparent matter anti-matter discrepency in the universe. ALICE is a heavy ion experiment, studying quark-gluon plasma through lead ion collisions. At LHC startup, the first beams were seen in all of the experiment detectors.

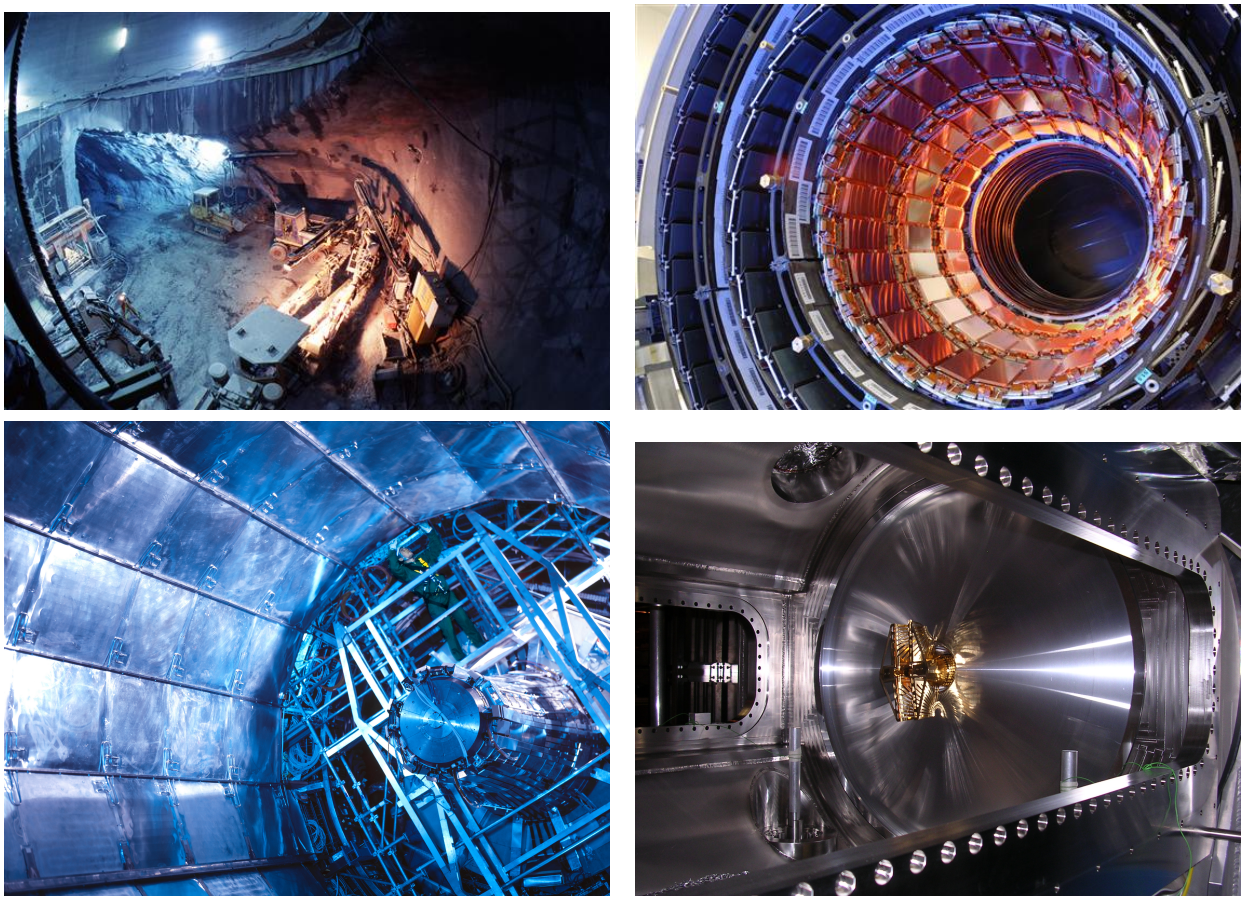


*Figure 3.7: Left to Right, the ATLAS Cavern, CMS inner tracker barrel showing three layers of silicon, inside the ALICE Magnets, the radio frequency boxes in LHCb [13]*

## 3.7 LHC Physics Motivation

The LHC has an unprecedented capacity for new physics, both within and beyond the Standard Model, due to its high energies, improved cross sections and high luminosity. LHC experiments aim to study new physics, as well as improving accuracy and making previously inaccessable measurements within the Standard model.

The LHC experiments aim to establish values for previously unmeasured quantities within the Standard Model, where the energies required for measurements have been until now unattainable. Higgs boson searches are a major feature of such physics studies. ATLAS and CMS have extensive Higgs boson physics search programmes covering the mass range from 114.4 GeV, excluded at 95% by LEP to several hundred GeV where theoretical predictions become restrictive, and across many production and decay modes.

Should discovery of the Higgs boson occur, the experiments will then attempt to study and measure properties of the newly detected particle. A discovery, identification and confirmation of the existence of the Higgs boson will enhance understanding the Standard Model.

Higher precision measurements of known quantities are also expected, as higher cross sections improve the ability for measurements. For ATLAS and CMS these include precise measurements of mass, coupling and decay properties of the top quark, measurements of the $W$ boson mass and gauge boson triple coupling, while LHCb will focus on the properties of B mesons and Charge Parity violation, the mechanism by which matter anitmatter asymmetry is understood to occur.

The Standard Model has been well tested up to an 100 GeV energy scale. Beyond this scale to the 1 TeV energies within the reach of LHC experiments, may lie many more discoveries beyond the standard model. The Hierarchy Problem, why the Higgs mass is so much smaller than the Planck scale, may be explained by supersymmetry. Supersymmetry theory describes a symmetry between fermions of half integer spin with bosons of integer spin, so that each fermion has a bosonic supersymmetric partner and each boson a femionic supersymmetric partner. There are supporting arguments for the case that supersymmetric particles weigh around 1TeV, making them within the potential reach of the LHC experiments [30].

As string theory predicts extra dimentions in space, it may be possible to see evidence of new dimensions at the LHC. Supersymmetry allows many possibilities for differences between matter and antimatter than allowed by the Standard Model, so evidence and study of supersymmetry may go some way towards explaining the discrepancy between matter and antimatter in the universe. LHCb will study potential discrepancies in the standard model through the decays of mesons containing bottom and strange quarks and CP violation.

## 3.8 Particle Physics Experiments leading to the LHC

### 3.8.1 CERN

#### Super Proton AntiProton Synchrotron, UA1 and UA2

The Super Proton Synchrotron accelerator at CERN was used as a proton antiproton accelerator to provide beams for the UA1 and UA2 experiments between 1981 and 1984. In 1983 signatures of $W$ bosons were observed, followed shortly after by evidence of $Z$ bosons. The SPS accelerator is used as the final stage energy boost for protons at the LHC, accelerating beams from 26 GeV to 450 GeV. When the LHC is upgraded to increase luminosity in 2015, the SPS will also be upgraded to the Super SPS, capable of 1 TeV energies.

#### LEP

The Large Electron-Positron Collider, LEP, operated at CERN between 1989 and 2000. The tunnel used in the LEP accelerator is currently used for the LHC experiments. LEP created $e^+e^-$ interaction for study at four experiments, Aleph, Delphi, Opal and L3 [31]. When electrons and positrons collide, as in LEP, they annihilate and produce photons or $Z$ and $W$ bosons. $Z$ and $W$ bosons, already detected at CERN in UA1 and UA2, were then studied more precisely at LEP. LEP 1 studied $Z$ and LEP 2 studied $W$.

LEP began by accelerating electrons and positrons at energies of 45 GeV each so that $Z$ bosons could be produced and the $Z$ mass be measured. After energy upgrade, $W$ boson pairs were produced. By 2000, when the experiments shut down, the accelerator was capable of energies of 209 GeV. LEP led to many precision measurements within the Standard model, most notaby the mass of the $Z$ and $W$ bosons, as well as placing a lower limit on the mass of the Higgs boson $H$. The LEP results for $Z$,$W$ and $H$ are shown in table 3.3, values taken from [8]

| Observed at LEP | Mass |
|---|---|
| Z | 91.1876 ± 0.0021 GeV |
| W | 80.398 ± 0.025 GeV |
| H | > 114.4 GeV |

*Table 3.3: W and Z masses and H mass limits observed at LEP*

### 3.8.2 Fermilab

The Tevatron, Trillion eV producing Synchrotron, is a particle accelerator at Fermilab, the Fermi National Accelerator Laboratory in Batavia, Illinois, USA. Until the LHC begins operation, the Tevatron is the highest energy particle accelerator in the world. The Tevatron uses a 6 km circular ring to accelerate protons and antiprotons to energies of up to 1 TeV. By 2008 the Tevatron was capable of centre of mass energies of 1.96 GeV and has begun the Higgs boson searches. These Higgs searches are to be continued at the LHC.

#### CDF and DO

The Collider Detector at Fermilab, CDF [32], and the D0 experiment [33] use proton antiproton collisions produced by the Tevatron accelerator at alternative interaction points. CDF and D0 studies focus primarily on Standard Model searches and measurements, and together shared responsibility for discovery and measurement of the top quark [34]. In 1995 the top quark was discovered and by 2007 precision measurements of top quark mass had been made, shown in table 3.4, values taken from [8]

| Observed at CDF and D0 | Mass |
|---|---|
| $t$ | 171.2 ± 2.1 GeV |

*Table 3.4: Top mass observed at CDF and D0*

## 3.9 Summary

The LHC is a proton proton beam particle accelerator at CERN in Geneva, Switzerland. In this chapter, the LHC beam and luminosity has been discussed and the accelerator described. Vital Statistics for the LHC accelerator and beam have been presented. The aims of the LHC have been introduced, as has the innovative and complex engineering structure of the collider. The LHC Startup in September 2008 and data from the first beams to circulate the LHC loop has been presented. The LHC accelerator complex and the four international collaborations that use the LHC beams have been introduced, as well as the experiments and the notable physics results leading to the development of the LHC projects. We now focus on one of the international collaborations at the LHC, the ATLAS collaboration.

# Chapter 4

# ATLAS

## 4.1 Introduction

ATLAS, [15], [16], [17], [18] is a multipurpose physics experiment at the LHC. The ATLAS collaboration consists of 2000 internationally based scientists, in 151 universities and institutions, from 134 countries worldwide.

ATLAS hopes to shed light on new theories beyond the Standard Model and maximise the discovery potential for new physics within the Standard Model, such as supersymmetry and the Higgs boson. ATLAS aims also to make improved measurements of particles known to exist within the Standard Model, such as heavy quarks and gauge bosons.

ATLAS is now introducted and the aims of the collaboration are discussed. The detector is presented in terms of its components, their structure and purpose within the experiment. As the first LHC beams circulate through the LHC tunnel, the first ATLAS events were seen. Data from the first LHC beam in the ATLAS detector are described.

## 4.2 ATLAS Detector

The ATLAS, A Large ToroidaL ApparatuS, detector is 45 metres long, 23 metres in diameter and cylindical in shape. ATLAS weighs 7000 tonnes and is designed to accurately detect and measure features of physics interactions, which can then be used to study each

event. Since many of the physics processes ATLAS is searching for are rare and the particles involved are shortlived, it is not possible to directly witness an event of interest. Instead physicists study a series of detector responses to a collision and use the detector output to rebuild the full physics event to study, in a process known as reconstruction.

The ATLAS detector is shown in figure 4.1. Vital Statistics for the ATLAS detector are shown in table 4.1. Particles from the accelerator pass through the central beam pipe in the centre of the detector and collide at the Interaction Point. Particles produced in collisions then propagate out from the primary vertex.

The detector is a series of subdetectors, these are the Inner detector [22], Electromagnetic calorimeter [23], Hadron Calorimeter [23] and Muon system [24]. Each subdetector system detects features of an event, occuring throughout the full detector volume.

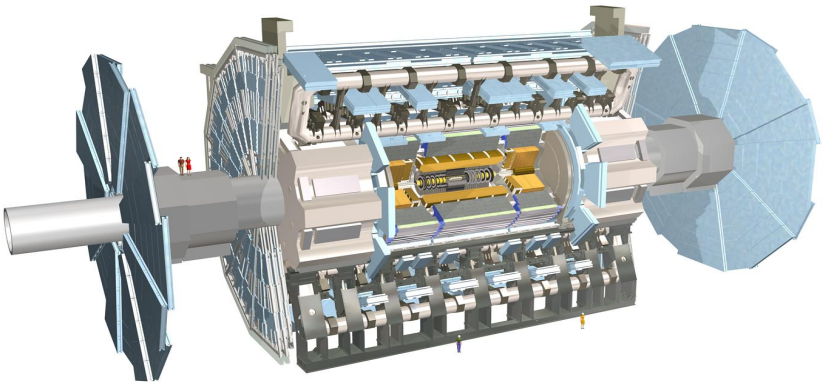


*Figure 4.1: The ATLAS Detector, 45 metres in length, 23 metres in diameter and cylindrical in shape, the diagram shows the subdetector layers, the Inner Detector, Electromagnetic Calorimeter, Hadronic Calorimeter, Muon system and Magnet system, that make up the full detector*

As the detector is a series of subdetector layers each designed to detect specific particles and their properties, an event can be seen in a subset of the subdetector layers, depending on the particles being detected. The Inner Detector measures the paths and therefore momenta of charged particles, the calorimeters measure the energy of charged particles

and the Muon spectrometer identifies and measures muons, which reach the outer parts of the detector. Neutrinos are not seen in the detector and are instead inferred from the presence of missing energy in an event.

ATLAS also has two magnet systems, the first containing the Inner Detector subsystem and the second the Muon system.

Figure 4.2 shows a simulation of a physics event using the Atlantis Event Display [25]. The image shows the cascade of information propegating from a proton proton collision and the hits and tracks in the ATLAS Inner Detector. Event features are seen through layers of the detector, depending on the characteristics of the physics objects in the event.

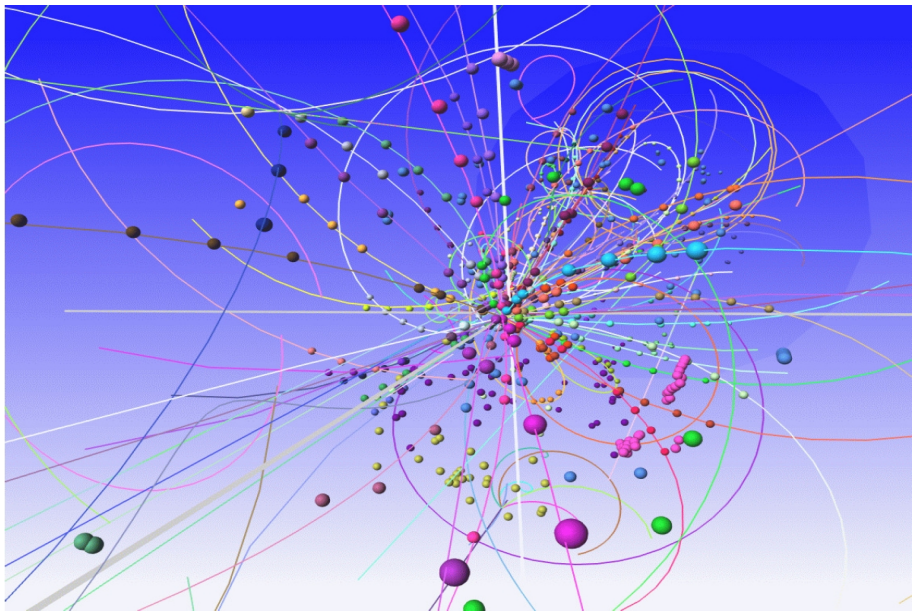


*Figure 4.2: An ATLAS event created using the visual event display Atlantis, the event shows the instance just after a collision where output of an event as hits and tracks in the Inner Detector can be seen*

## 4.3 The ATLAS Co-ordinate System

In Cartesian co-ordinates, the interaction point in the detector is the origin, the $x$ axis is horizontal and is directed towards the centre of the LHC loop, the $z$ axis is directed in the anticlockwise beam direction, viewing the LHC loop from above, and the $y$ axis is directed upwards with respect to the $x$ and $z$ axes.

Transforming to spherical co-ordinates gives the polar angle $\theta$ and azimuthal angle $\phi$, where $\theta$ is measured from the beam axis in the $z-y$ plane and $\phi$ is measured around the beam axis in the $x-y$ plane. ATLAS adopts the convention by which $+\theta$ values refer to the positive $z$ direction and $+\phi$ values refer to an anticlockwise angle measurement.

Pseudorapidity $\eta$ is a valuable quantity as particle separation in $\eta$ space is Lorentz invariant and particle production is uniform viewed in $\eta$.

The distance measurement $\Delta R$ measures a separation between objects in pseudorapidity azimuthal space.

$$\phi = \tan^{-1}(\frac{y}{x}) \tag{4.3.1}$$

$$\theta = \cos^{-1}(\frac{z}{\sqrt{x^2+y^2+z^2}}) \tag{4.3.2}$$

$$\eta = -\ln\tan(\frac{\theta}{2}) \tag{4.3.3}$$

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{4.3.4}$$

## 4.4 The Inner detector

The Inner Detector is a cylinder 7 metres in length and 1.15 metres in radius and is the innermost part of the ATLAS detector. The Inner Detector is held within a solenoid which gives the subsystem a 2 Tesla magnetic field. By combining discrete high resolution semiconductor pixel and strip detectors in the inner radii part and continuous tracking elements, strawtube tracking detectors capable of detecting transition radiation, in the outer radii part, the inner detector measures the paths and momenta of charged particles. The inner detector must provide good b-tagging performance throughout LHC active data taking.

The magnetic field surrounding the inner detector causes charged particles to follow a curved path, the direction of the curve shows the charge of the particle and the angle of trajectory/degree of curvature gives the particles momenta. Electron recognition takes place in the Transition Radiation Tracker in the outer radii of the Inner Detector. The Inner Detector is primarily designed therefore to detect charged particles and to allow determination of charge and momenta. The Inner Detector operates within the $|\eta| < 2.5$ region of the detector.

Accuracy is crucial in providing useful measurements. The Inner Detector therefore records on average 43 position measurements for each charged particle between the beam and electromagnetic calorimeter. Position measurements are reconstructed into tracks providing high precision momentum and charge information. Secondary vertex identification using the reconstructed tracks can be used to indicate the presence of short lived particles such as $\tau$ leptons and $b$ quarks. The inner detector construct is composed of three parts, the inner barrel $\pm$ 80 cm along the $z$ axis and two end caps and is designed to withstand relatively high levels of ionising radiation. In the barrel region detectors are mounted in concentric cirles around the beam pipe, in the end caps detectors are perpendicular to the direction of the beam. The Inner Dector subsystem can be subdivided into three components, Pixel Detector, Semi Conductor Tracker and Transition Radiation Tracker.

### 4.4.1 The Pixel Detector

The inner most part of the Inner Detector is the Pixel Detector. The Pixel Detector is a grouping of pixel cells, each measuring 50 $\mu$m in $\phi$ and 400 $\mu$m in $z$. In total there are 2.3 square metres of these fine resolution detectors. The pixel detector is assembled in three layered modules, the layers are at $\eta$ =5.05, $\eta$ =8.85 and $\eta$ =12.25 from the beam line. The pixel detector is capable of precise measurement of positions and provides three precision measurements as close to the interaction point as possible in the detector. This ability is important for secondary decay measurements, identification of B hadrons and

therefore tagging of $b$ jets.

### 4.4.2 Semiconductor Tracker

The SemiConductor Tracker detectors, SCT, are mounted on detector barrel layers at $\eta$ =30, $\eta$ =37.3, $\eta$ =44.7 and $\eta$ =52 cm, along with nine end cap discs, comprising 61 $m^2$ of silicon detectors in total. The SemiConductor Tracker is made of 6.4 $cm^2$ silicon wafers bonded in pairs to make strips, then joined again in pairs of two back to back at a 40 mrad angle, into a module.

The SCT has fewer read out channels and less material than the pixel detectors, so track density is lower in the SemiConductor Tracker than in the pixel detector, but, given the wider spacing, can still provide precise momentum measurements.

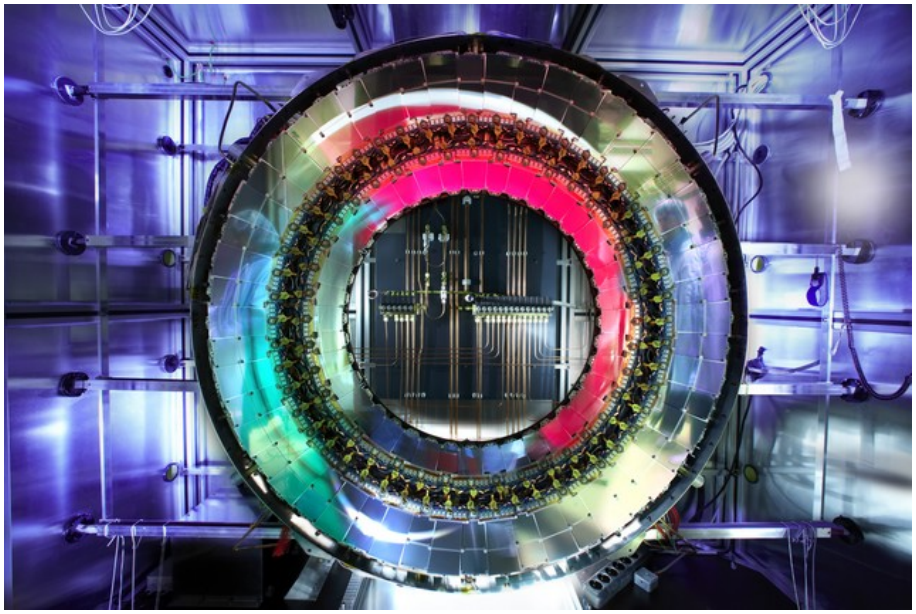


*Figure 4.3: The ATLAS SemiConductor Tracker, in the Inner Detector, measures paths and momenta of charged particles*

### 4.4.3 Transition Radiation Tracker

The Transition Radiation Tracker, TRT, provides tracking in the 56 cm to 107 cm detector radii range using straw tube detectors. The TRT contains 370000 aluminium straws, each 4 mm in diameter with length up to 150 cm. Each straw tube contains a wire

and a mixture of gases, including Xenon. Transition radiation is emitted when particles traverse the boudaries between materials of different dielectric properties. Xenon gas allows electron identification through detection of transition radiation photons that pass a higher threshold in the read out electronics than the charge liberated by a minimum ionising particle.

## 4.5 Calorimetry

The ATLAS Calorimetry system has two main sections, the Electromagnetic Calorimeter and the Hadronic Calorimter. The Calorimetry system covers the pseudorapidity range $|\eta| < 3.2$ in the Electromagnetic Calorimeter, $|\eta| < 1.7$ in the barrel Hadronic Calorimeter, $1.5 < |\eta| < 3.2$ in the Hadronic end cap Calorimeter and $3.1 < |\eta| < 4.9$ in the forward Calorimeters. Calorimeters absorb and measure the energies of electrons, photons and hadrons and are therefore responsible for accurate measurement of the energy and position of electrons and photons, energy and direction of jets and missing transverse momentum of an event and particle identification. Calorimeter resolution improves with energy. Quantities measured in the calorimters are used online, in real time. The Trigger system uses Calorimeter output to identify events to be passed to the offline system.

### 4.5.1 The Electromagnetic calorimeter

The Electromagnetic Calorimeter measures the energy of particles by absorbing energy from those particles which interact electromagnetically, so is sensitive to charged particles and photons. The Electromagnetic Calorimeter is intended to detect and allow calculation of the energies of charged particles.

The Electromagnetic Calorimeter covers the pseudorapidity range $|\eta| < 3.2$. The Electromagnetic Calorimeter is a lead/liquid argon, LAr, detector and consists of a barrel and two end caps. Lead is an energy absorbing material and liquid argon is a sampling material. The Electromagnetic Calorimeter is a sampling detector, this means particle absorption and active signal readout are handled separately. In the Electromagnetic

Calorimeter are layers of lead interleaved with liquid argon in an accordeon geometry for complete azimuthal symmetry without cracks, optimised for high sampling rate. The lead creates an electromagnetic shower and absorbing particle energy and the liquid argon allows a sampling measurement of the energy deposited in the detector. Energy is absorbed and periodically the shape of the resulting particle shower is sampled, from this particle energy can be measured. Cyrostats are placed around the Electromagnetic Calorimeter to keep it cool, at a temperature of 89 K.

The EM Calorimeter has a presampler layer of lead, at $|\eta| < 1.8$, intended to correct for losses in the inner detector and solenoid. After the presampler there are three sampling layers, varying in granularity or resolution with $\eta$. The Electromagetic Calorimeter focuses on high granularity in the low pseudorapidity region $|\eta| < 2.5$.

### 4.5.2 The Hadronic Calorimeter

The Hadronic Calorimeter, like the Electromagnetic Calorimeter, consists of a barrel and end caps and is developed for a specific pseudorapidity range, higher $|\eta|$ values than the Electromagnetic Calorimeter. Particles which pass through the Electromagnetic Calorimeter and do not interact by Strong force are detected in the Hadronic Calorimeter.

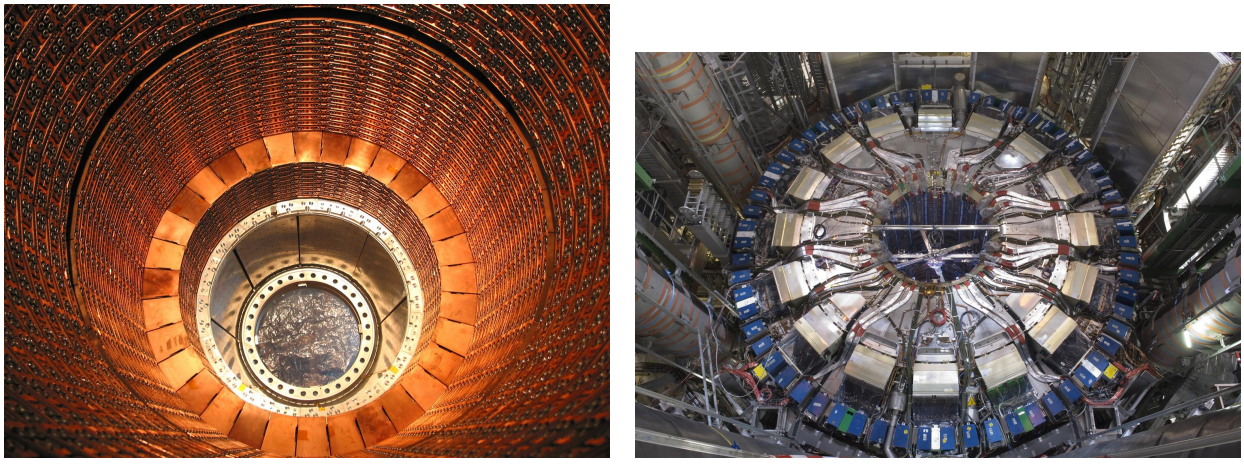


*Figure 4.4: Left to right - Assembly and installation of the ATLAS Hadronic endcap Liquid Argon Calorimeter and Insertion of Calorimeter into ATLAS Detector*

## 4.6 The Muon system

The Muon spectrometer detects and measures the mass of muons in an event, using magnetic deflection of muon tracks in a magnetic field. The muon system consists of superconducting aircore toroid magnets and high precision tracking chambers. Muons are the only particles, with neutrinos which pass through the Muon system and are seen as missing energy in an event, to pass through the Inner Detector and Calorimeters and reach the outer parts of the detector. The Muon system is used in the online Trigger system for many important physics channels and as a high precision Muon spectrometer for measuring track momenta.

## 4.7 Magnet Systems

ATLAS has two magnet systems, these surround the Inner Detector and the Muon system and bend the paths of charged particles so that momenta can be measured. The Inner detector is encompassed in a solenoidal 2 Tesla field. The field strength ensures that all particles, including more energetic particles, are caused to take a curved path in response to the magnetic field. The second outer toroidal magnetic field is created using eight aircore superconducting magnets shown in figure 4.5, and two end cap toroidal magnets. The eight superconducting magnets are the shape of a round cornered rectangle, with dimensions 5 metres by 25 metres and each weighs 100 tonnes, together creating a magnetic field with circular field lines in a direction perpendicular to the beam.

## 4.8 Trigger and DAQ

The ATLAS Trigger and Data Acquisition system, TDAQ, [26], [27], is a hardware and software system that acts as a bridge between the detector during data taking and physics study, translating across online detector response and making data available for offline data analysis and processing.

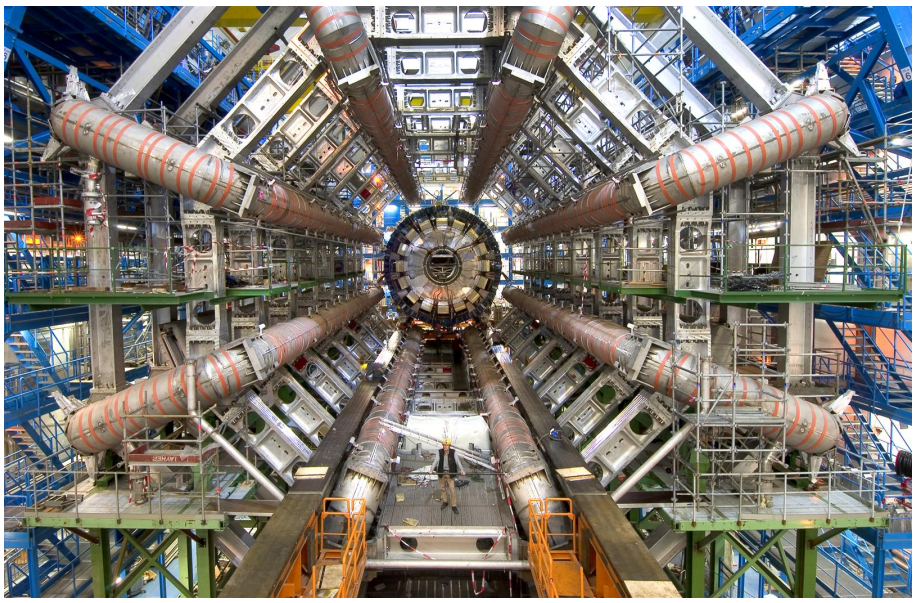


*Figure 4.5: The ATLAS Outer Magnet System showing the eight superconducting magnets in the ATLAS Cavern, outside the Calorimeters and inside the Muon system, together creating a magnetic field perpendicular to the beam*

The Trigger must provide efficient rejection of high rate backgrounds as well as efficient selection of the rare signal events for which ATLAS is searching. Trigger decisions are made online as the information is first seen in the detector. The time latency is very short so that the computing system can maintain a decision rate comparable with the event rate. The ATLAS Trigger and DAQ system has three levels of event selection. Level One selections are to be made within 2 $\mu$s, Level Two selections within 10 ms and Event Filter selections within 1 second.

The three levels of Trigger selection are shown in figure 4.6. The aim is to reduce the event rate to a rate managable for offline processing. Each event seen in the detector is passed to the online Trigger system to undergo a series of tests to ascertain its usefulness to future offline analysis. The selection levels refine the selection made at previous levels and apply increasingly complex selection algorithms and criteria.

### 4.8.1 Event Selection

The Level One (LV1) Trigger reduces an initial event rate of 40 MHz in the detector to 75 kHz. Level One is a hardware Trigger based on calorimeter and muon information from

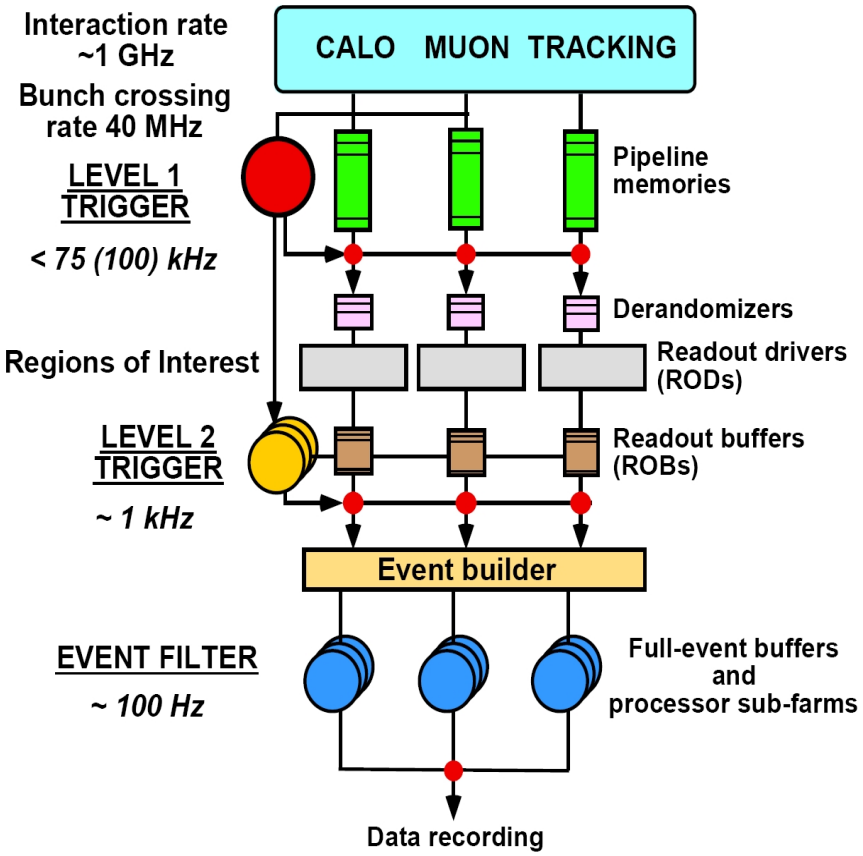



*Figure 4.6: The ATLAS Trigger system showing the three levels of selection, the Level One Trigger, the Level Two Trigger and the Event Filter, together reducing the online event rate from ≈ 1 GHz to ≈ 200 Hz for offline analysis [26], [27]*

| ATLAS Detector Parameter | Value |
|---|---|
| ATLAS length | 45 m |
| ATLAS diameter | 23 m |
| ATLAS weight | 7000 tonnes |
| Temperature in ATLAS | 1.9 K |
| Magnetic Field in Inner Detector | 2 Tesla |
| Magnetic Field of Outer Superconducting Magnets | non-uniform |
| Inner Detector length | 7 m |
| Inner Detector radius | 1.15 m |
| Inner Detector range | $\|\eta\| < 2.5$ |
| Electromagnetic Calorimeter range | $\|\eta\| < 3.2$ |
| Hadronic Calorimeter range | $\|\eta\| < 1.7$ |
| Hadronic Endcap Calorimeter range | $1.5 < \|\eta\| < 3.2$ |
| Forward Calorimeter range | $3.1 < \|\eta\| < 4.9$ |
| Pixel Channels | $140 \times 10^6$ |
| Transition Radiation Tracker Channels | $6.2 \times 10^6$ |
| Silicon Strip Channels | $0.42 \times 10^6$ |
| Pixel Layers crossed per track | 3 |
| Pixel Strip Layers crossed per track | 8 |
| Transition Radiation Tracker tracking points | 36 |

*Table 4.1: Vital Statistics for the ATLAS Detector*

the detector. Level One uses general physics criteria, such as high transverse energy in the calorimeters, to meet the requirements of most physics channels. Events that pass the Level One Trigger are stored in readout buffers to be considered by the Level Two Trigger (LV2).

Level Two is a software Trigger and applies selection algorithms to further test the event, reducing the event rate to 2 kHz. Event data in regions flagged as interesting by

Level One as regions of candidate jets, electrons, photons are unpacked from the readout buffers to be filtered by Level Two algorithms, not the full event. These regions are called Regions of Interest and are described geometrically in terms of $\eta$ and $\phi$.

Events passing the Level Two Trigger are passed by the DAQ system from readout buffers to the Event Filter (EF). The Level Two Trigger and Event Filter are collectively called the High Level Trigger (HLT). The Event Filter performs further selection algorithms, using access to further offline data such as alignment data. Events passing the Event Filter are then stored, with the Event Filter output data appended to the event, at a rate reduced from 40 MHz to 200 Hz.

### 4.8.2 Menus, Signatures and Configuration

The full online event selection using the Trigger is described by a Trigger Configuration. The Configuration is the Trigger Menu plus prescale values and forced accept rates.

A Trigger Menu is a series of Trigger Signatures, where Trigger Signatures are a logical combination of Trigger Elements. The Trigger Signature e25i is a combination of three Elements, e, 25 and i, and refers to an isolated electron of transverse energy greater than 25 GeV. The Signature 2e15i combines the same three Elements to create a signature for two isolated electrons, each with transverse energy greater than 15 GeV.

The Trigger signatures of interest in the $t\bar{t}H, H \rightarrow b\bar{b}$ analysis channel, used to identify semileptonic signal events, are

- e25i - an isolated electron of energy of at least 25 $GeV$
- e60 - an electron of energy of at least 60 $GeV$
- mu20i - an isolated muon of energy of at least 20 $GeV$

This Trigger selection ensures that there is an isolated high $p_T$ lepton in the event. The $t\bar{t}H, H \rightarrow b\bar{b}$ analysis channel is addressed in Chapter 10.

If one or more of the Trigger Signatures in a Trigger Menu is activated, the event passes selection. The reason for passing the selection, the signature activated, is stored

in the event data, for later identification of usefulness to a specific analysis. In this way a Trigger Menu can be created so that the initial event identification steps of multiple distinct physics searches are taken at the same time. Trigger Configuration is fixed across data runs and can only be changed at run boundaries. In Chapter 7, the change of active Trigger Menu and the consequences for the ways the TAG Database handles Trigger decisions and menus are discussed.

## 4.9 The First Data seen ATLAS

In September 2008 the first full beams of protons were circulated through the LHC tunnel, first in a clockwise direction, then anticlockwise. Beams were seen in the ATLAS detector. The beams were directed at a target near ATLAS, a collimator used to focus or block the beam, and the detector systems lit up as a cascade of muon particles passed through the detector, providing an opportunity to test the detector systems when real beams are circulated.

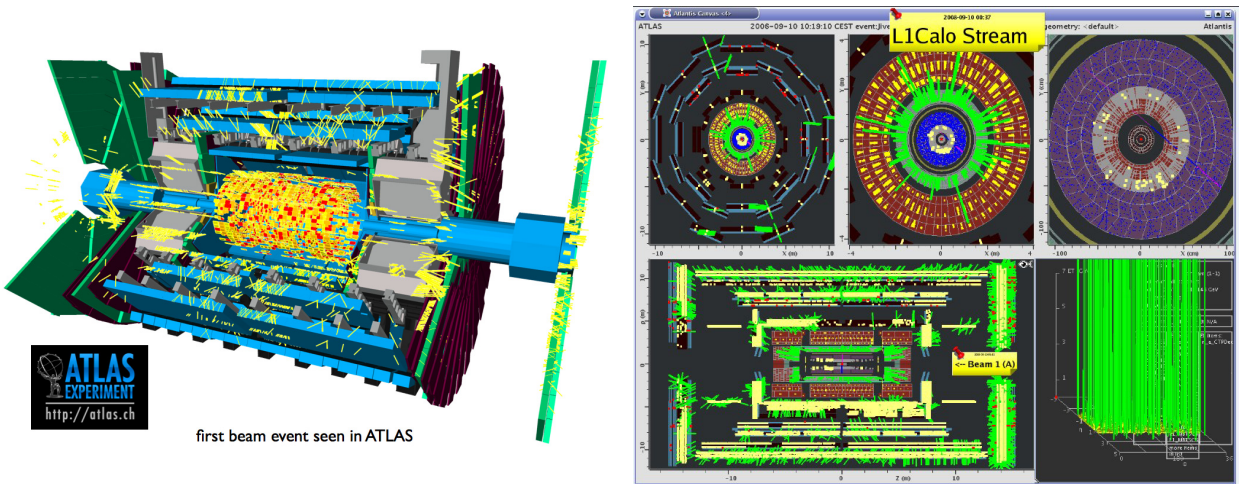



*Figure 4.7: The First ATLAS Beam Events seen in the detector in September 2008, as proton beams were directed at a target near the detector, a cascade of muons was seen throughout the detector and in the detector systems [15]*

## 4.10 Summary

We have now introduced the ATLAS collaboration and discussed the experiment in terms of its physics purpose and detector system. Vital Statistics for the ATLAS detector have been presented. ATLAS is a set of subdetector layers, the detector layers and purpose within the experiment described. The ATLAS co-ordinate system has been presented. ATLAS Trigger and Data Aquisition are a complex on and offline event selection systems, the selection system has been presented and the levels of selection explained, with the Trigger selections for $t\bar{t}H, H \rightarrow b\bar{b}$, the potential Higgs discovery channel studied in this thesis, presented as an example. The first LHC beam data seen in the ATLAS detector at Startup, muon particles passing through the detector after the first LHC beams were directed towards a target near ATLAS, has been presented. We now move on to the ATLAS Computing systems, an important, challenging and integral part of the ATLAS collaboration.

# Chapter 5

# ATLAS Computing

## 5.1 Introduction

In the quest for new physics and multipurpose physics searches and studies, ATLAS will produce data of volume and rate unprecedented in Particle Physics. As physicists studying ATLAS data are scattered internationally, ATLAS data must be accessible to physicists at internationally located collaborating institutions. A significant challenge for the ATLAS collaboration lies in developing the capacity to manage an unprecedented data rate and an anticipated yearly data volume of the order of petabytes in a fluid, sound and transparent way. This chapter describes the ATLAS Computing Model and the design and performance of the system and environment, using the themes of data type, access, creation, storage, persistency, navigation and management. ATLAS event data is introduced and event data types and data production are detailed. ATLAS adopts Grid Computing shaped by a hierarchial tier model, the ATLAS tiers and tier roles within the collaboration are described. The ATLAS Distributed Data Management system oversees the movement of all data within the tier model, the data management system, its concepts and uses are discussed. ATLAS non event data is introduced along with its role in the collaboration. The ATLAS Computing Model is an innovative and comprehensive system and an integral and important part of the ATLAS collaboration.

## 5.2 The ATLAS Computing Model

The design, architecture and performance demands of the ATLAS software and computing system is described within the ATLAS Computing Technical Design report [17], in the ATLAS Computing Model [35]. The Computing Model covers data lifecycle from online trigger selection through processing, distribution, storage and analysis by a physicist. These steps in data lifecycle are referred to as offline computing. Online steps are the detector, Trigger and DAQ systems.

The Computing Model describes ATLAS offline computing in terms of the multiple roles data will play in a running, high data rate and large data volume experiment. The central computing and data themes are storage, access, processing, format, analysis and management. The Computing Model adopts Grid Computing [36], where decentralised distributed resources and data are shared throughout the collaboration.

## 5.3 ATLAS Data

For ATLAS, $10^{15}$ bytes of data are expected annually. Table 5.1 shows the data rates from the High Level Trigger for ATLAS and the other LHC experiments. As each interaction produces a large number of particles, and ATLAS has a high rate of interactions, ATLAS has a large event size for the RAW data selected by the Trigger to be written to files for storage and processing. The RAW data is added to by reconstruction, analysis and simulation data.

| Experiment | Data Rate from High Level Trigger |
|---|---|
| ATLAS | 200 Hz |
| LHCb | 2000 Hz |
| CMS | 150 Hz |
| ALICE | 100 Hz |

*Table 5.1: ATLAS and other output data rates from the High Level Trigger at the LHC experiments [17], [19], [20], [21]*

## 5.4 ATLAS Event Data types

The ATLAS Computing Model defines ATLAS event data as a series of event representations. The data may be of a variety of formats, created for differing purposes. Data of varying types will have varying distribution models, different access patterns and access frequency.

### 5.4.1 RAW data

RAW data is the bytestream event data passed to the offline event store, where event data is stored and handled offline, from the online event filter, the last stage of online selection by the Trigger system. RAW data is detector output and is yet to undergo any reconstruction. The RAW data must be processed to produce event data in an object oriented format which can be used by an ATLAS analyst. The Computing Model assumes a RAW event data size of 1.6-2 MB at an output rate of 200 Hz from the online selection. RAW events are written to files of maximum size 2 GB and transferred in files from the event filter to offline resources for reconstruction. Events are grouped in RAW data files by detector run, but are not ordered by any physics selection criteria, or time within a run.

### 5.4.2 ESD data

ESD is Event Summary Data and is the result of performing a reconstruction process on RAW detector output data, producing event data in a first object format. Physics objects such as tracks, vertices, jets, electrons, muons and physics criteria are described by event data in object oriented format. The Computing Model assumes an ESD event size of 500 KB, reduced from the 2 MB RAW event size. ESD event data is stored in POOL ROOT files, discussed later in the chapter. Events are grouped in files by detector run but no time or physics selection criteria, as the mappings from RAW to ESD files are one to one.

### 5.4.3 AOD data

AOD, Analysis Object Data, is a further reduced event representation produced from ESD data. AOD is an object oriented event representation containing physics objects and is intended for physics analysis. The AOD event data size assumed by the Computing Model is 100 kB per event and AOD is stored in POOL ROOT files. AOD events are grouped in files by a physics selection implemented through Physics Streams.

### 5.4.4 TAG data

TAG data are event level metadata, thumbnail information about an event. An event TAG is a small summary of event characteristics intended to facilitate identification and selection of events for an analysis without having to open and search through larger AOD files. The Computing Model assumes a target size for a TAG of 1 kB per event. TAGs are stored in both POOL ROOT files and relational databases. TAGs are initially written to files at time of AOD creation and are later imported into relational tables.

### 5.4.5 Simulated data

Simulated data describes all the data produced in the process of simulating ATLAS events. Simulation of event data is a process involving generating events by some physics signal criteria, simulating the interaction of particles with the detector and the detector response. As simulated data are simulated event data from different stages of processing, the data is a range of data types. Simulated data are stored in POOL ROOT files. Simulated events in bytestream format are 2 MB and are larger than RAW events, as a simulated event will also include Monte Carlo truth information.

### 5.4.6 Derived Physics Data

Derived Physics Data is ntuple type representation of data in a format useful for analysis, histogramming and visualisation by a physicist. Derived Physics Data is created by a

physicist using AOD as input. Physicists selects the physics objects of interest to an analysis and creates a new event representation containing only these objects, so creating a smaller event representation that can be more easily moved and analysed. Derived Physics Data is expected to be an order of ten times larger than TAG data, depending on content selected by a physicist to be included for an event.

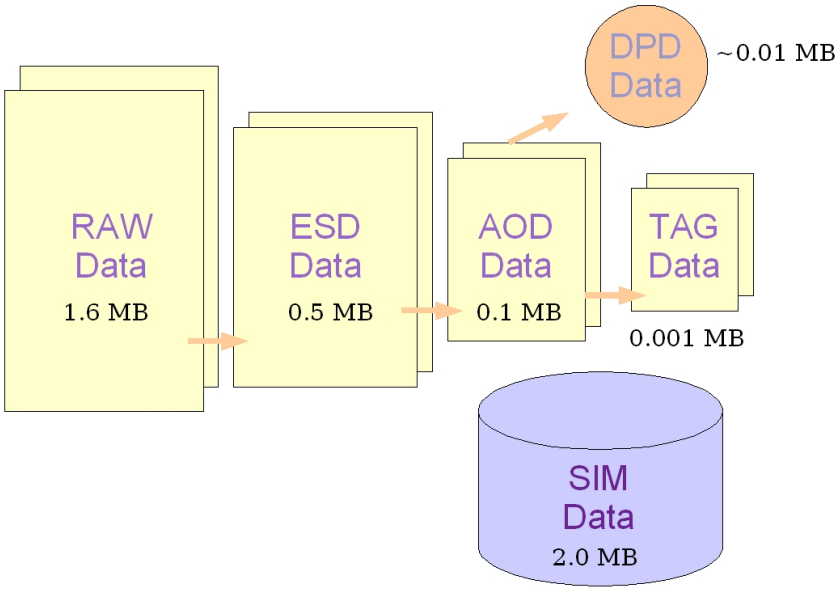



*Figure 5.1: ATLAS Data types, order of production and size*

| | RAW | ESD/RECO/DST | AOD/rDST | TAG | SIM |
|---|---|---|---|---|---|
| ATLAS (MB) | 1.6 | 0.5 | 0.1 | 0.001 | 2 |
| ALICE(p-p) (MB) | 1 | 0.2 | 0.05 | 0.01 | 0.4 |
| ALICE(heavy ion) (MB) | 12.5 | 2.5 | 0.25 | 0.01 | 300 |
| CMS (MB) | 1.5 | 0.25 | 0.05 | 0.01 | 2 |
| LHCb (MB) | 0.025 | 0.075 | 0.025 | 0.001 | - |

*Table 5.2: Data sizes for ATLAS and LHC experiments [37]*

## 5.5 ATLAS Event Data Production

Data Production refers to processing of event data and can take place both as First Pass Production to produce Primary data, initial processing of data, and then subsequent re-processing as software versions used to process the data improve with detector experience. Processing takes place on all data types.

### 5.5.1 First Pass ESD Production

Event data in RAW bytestream detector output is reconstructed to a first object representation as ESD. ESD production begins as soon as RAW data files and any simultaneously required calibration and conditions data arrive at Tier 0 from the Event Filter. RAW to ESD files map one to one. Each ESD first pass processing production job takes one RAW bytestream event data file as input and produces one file of reconstructed events in object oriented ESD format in a POOL ROOT file as output. Events are grouped in RAW data files by run number and as ESD files are produced in a one to one mapping from RAW files, events in ESD files are grouped only by run number, making the unit of ESD production a detector run.

### 5.5.2 First Pass AOD Production

First pass AOD production takes place immediately after first pass ESD production. AOD production creates more detailed physics objects from the ESD event data. Both AOD and ESD are event data in object oriented format but it is the smaller AOD that is most suited to analysis.

As AOD event data are intended to support analysis, streaming is introduced at AOD production whereby events are selected as belonging to one of many predefined physics streams. Streams reflect anticipated data access patterns of ATLAS analysis, to streamline access to data likely to be accessed by a physicist for specific analyses, improving access times by grouping data in files, and also acting as an event selection criteria. AOD events are grouped therefore into files by stream.

Each AOD production job takes multiple ESD files as input and produces multiple AOD files as output, reflecting the streams for which events qualify. AOD events in a file always share a common stream criteria and run number and are selected from ESD as such. As AOD event representation is much smaller than ESD, AOD production produces small AOD files, so AOD files are merged into larger composite files, each containing events from many input ESD files.

### 5.5.3 TAG Production

Event level metadata TAGs are created at AOD production, specifically as AOD files are merged. TAGs are written firstly to POOL ROOT files as explicit collections. The file resident collections are later imported into relational databases at a managed and controlled rate, so avoiding contention in writing to databases tables. Event level metadata are intended to support selection of events across stream boundaries. TAG collections corresponding to the AOD streams and collections spanning stream boundaries will be built at first pass AOD production. The purpose of such event collections is to support event selection both within and across streams.

### 5.5.4 Reprocessing

The output of First Pass processing are Primary data. The latency of first pass processing is a function of the online computing system, as first pass processing takes place as data arrives at computing resources from the detector. Reprocessing will use the same software version as the original first pass processing. Reprocessing takes place subsequently at computing resources distributed throughout collaborating institutions. Reprocessing has a longer latency than first pass processing and so can use calibration and alignment data. Reprocessed data is therefore an improvement on first pass processing as a study of the calibration stream data has been undertaken in the intermediate time.

| | Reco time/event | Sim time/event | Analysis time/event |
|---|---|---|---|
| ATLAS | 15 | 100 | 0.5 |
| LHCb | 2.4 | 50 | 0.3 |
| CMS | 25 | 45 | 0.25 |
| ALICE (p-p) | 5.4 | 35 | 3 |
| ALICE (heavy ion) | 675 | 15000 | 350 |

*Table 5.3: Processing times for LHC experiments*

### 5.5.5 Processing times

Table 5.3 shows the processing times in kSI2000-s for reconstruction, simulation and analysis for ATLAS and LHC experiments [37].

## 5.6 Distributed Computing

ATLAS uses Grid Computing. Grid, or Distributed, Computing is a system whereby computing resources are physically separated and management of resources is decentralised. ATLAS Computing is not decentralised in the strictest sense, as a Tier system features in the model, but the collaboration does use Grid computing technologies across a set of hierarchial levels. ATLAS shares computing resources and responsibilities in a system distributed across collaborating institutions and offers a standardised interface to the computing system using middleware, special software designed for distributed computing systems, as an interface independant of location. As such ATLAS uses the main features of a grid computing system.

The LCG project, LHC Computing Grid, develops and provides much of the middleware needed to implement and use a computing Grid. Physicists and ATLAS software developers both perform analysis and develop ATLAS specific software within a Grid Computing environment.

## 5.7 ATLAS Tier Structure

The ATLAS Computing Model is shaped by Grid computing coupled with a hierarchial Tier system. The Tier system is afforded by the varying computing resources, both in terms of hardware and people power, that may be hosted at ATLAS participant institutions. All ATLAS contibuting sites will perform a varying role within the system based on an assigned status, or Tier.

ATLAS has four Tier levels. ATLAS has one primary or central Tier at CERN and this is called Tier 0. ATLAS then has ten globally distributed Tier 1 sites forming an umbrella structure over smaller and more abundant globally distributed Tier 2 sites. Each Tier 2 has a regional Tier 1 site to which it is in the first instance associated by geographical association although communication and transfer of data is supported between any Tier 1 and Tier 2 site. Tier 3 is a local ATLAS environment with storage which may be more hetrogeneous between Tier 3s.

It is not necessary that the set of Tier 1 or 2 sites support identical resources, but it is assumed that each tier set will support, by Tier definition, comparable storage ratio of CPU, disk and tape. The four tiered grid architecture is developed from the MONARC model, a project initiated in 1998 to develop a computing model for LHC experiments [37].

## 5.8 ATLAS Tiers

### 5.8.1 Tier 0

The central role of the Tier 0 centre at CERN is to process, store and distribute both the RAW data received from the Event Filter and subsequent processed data. Tier 0 operations can therefore be described in terms of processing, storage and distribution.

Tier 0 processing roles are First pass ESD Production, First pass AOD Production and reconstruction of the calibration and express stream data. Tier 0 storage roles are the archiving of primary RAW data, first pass ESD data, first pass AOD data and file and relational storage of TAG data. A copy of all reconstructed data is stored at Tier 0.

Tier 0 operations also encompass data distribution. The ESD, primary AOD and TAG data resulting from the first pass reconstruction performed at Tier 0 are distributed from Tier 0 to each Tier 1 centre. Reconstructed calibration data is distributed to the CERN Analysis Facility. Physically the Tier 0 consists of the Castor mass storage system and local replica catalog, a CPU farm, the Conditions database, the TAG database, a Tier 0 management system, a Tier 0 Production database and a Data Management system.

As the central point for ATLAS storage and primary processing, the Tier 0 centre must be high performance in terms of availability, response and reliability. In the event of downtime, the responsibilities for first pass processing and calibration are passed to Tier 1 centres. Two disk buffers, one providing 5 days of data protection for data flowing from Event Filter to Tier 0 to allow for any error or network outage and a second smaller buffer for protection in the event of failure during transfer of data from Tier 0 to Tier 1, are incorporated into the Tier 0 model. The Tier 0 is accessible to those directly involved in processing, not to individual physicists.

### 5.8.2 Tier 1

ATLAS has ten Tier 1 centres worldwide. Tier 1 sites host a subset of ATLAS data and have responsibility for a subset of reprocessing, the data from which is for use across the collaboration. A Tier 1 must provide access to and support analysis of all the data hosted at the site and also support the overall experiment calibration processing ability. A Tier 1 acts as a regional centres for a number of geographically located Tier 2 sites. The Rutherford Appleton Laboratory (RAL) is the regional centre for the UK.

Each Tier 1 stores and provides access to a subset, around one tenth, of the RAW data, which can be reprocessed at the hosting Tier 1 site following first pass processing at Tier 0. Subsequently the ESD, AOD and TAG datasets resulting from this reprocessing is made available to all ATLAS sites from the Tier 1 where the reprocessing is performed. The most recent copy of the data is available and accessible on disk with low latency, and a previous version is available on tape with a longer latency. Each Tier 1 will also store a

copy of the most recent version of data processed at an alternative Tier 1 site as a backup and the simulated data from Tier 2 sites.

As a Tier 1 centre plays a crucial role in data reprocessing, receiving RAW data and supporting data access and analysis, a Tier 1 must have a high performance in terms of availability and recovery from failure. Data stored at Tier 1 is accessible across the collaboration, not necessarily with short latency for RAW data but at least a fraction of the data should be on fast disk storage for calibration and algorithm development. Access to ESD, AOD and TAG is short latency for the most recent version of processing.

### 5.8.3 Tier 2

Tier 2 centres take a range of roles involving simulation, analysis, providing calibration constants and hosting of data, the nature of which depends on the resources available at the site. The central role shared by Tier 2 centres is production of Monte Carlo simulation, with simulated data copied to Tier 1 after it is produced. Tier 2 centres share the ATLAS simulation responsibility.

A Tier 2 hosts one third of the current primary AOD and all the TAG data. Tier 2's may also host a small set of RAW and ESD data for development of code. Some Tier 2's may take a role in calibration depending on local detector interests. If this is the case then the Tier 2 will host some calibration data. The simulated data produced at Tier 2 is sent to a Tier 1 centre, unless the Tier 2 can provide good performance in terms of access to the simulated data. A Tier 2 has lower demands on performance and availability than higher levels, unless the Tier 2 chooses to host the simulated data rather than passing this to a Tier 1 site. The resources must support simulation of ATLAS data.

Tier 2 sites will support a geographical area. Each Tier 2 has a preferred Tier 1 for data access but this is not fixed so if a Tier 1 is unavailable or data required for study on analysis is stored at an alternative Tier 1, Tier 2 centres can communicate with an alternative Tier 1 site. The Tier 1 centres are depicted as a cloud in some representations to show this model. All members of the ATLAS collaboration have access to Tier 2

centres. ATLAS policy may however determine priority to some users dependent on the data hosted at a Tier 2 site. A Memorandum of Understanding mapping the Tier 2 centres is in place between CERN and participating sites.

### 5.8.4 Tier 3

A Tier 3 site refers to a resource of a local nature. Tier 3s host user data needed for an analysis, such as ntuples, and support access to analysis using the Grid. A Tier 3 may be a local cluster or a collection of user desktop machines. The size and capability of a Tier 3 is likely to be driven by the needs of local users and the locally available resources, so while Tier 1s are homogeneous at least in capability (although not necessarily in the means by which the capability is provided), Tier 3s are likely to be a diverse set. The central role of a Tier 3 is analysis support and a Tier 3 does not have any responsibilty for collaboration wide data storage. Tier 3s are Grid enabled so that users can access to the Grid but Tier 3s are not part of the LCG Project. The resources may be used for simulation or analysis of data for a physics working group, but as a Tier 3 is locally and not centrally managed the required role is not defined beyond that of providing local user access to local storage and a capacity for local analysis.

## 5.9 ATLAS Data Flow

The ATLAS data flow takes events selected by the online trigger system through a series of offline steps. As part of the data flow, data is both produced and distributed.

The main input to the offline Computing Model data flow is a primary event stream containing all physics events in a series bytestream RAW data files. RAW data is transferred to the Tier 0 site, CERN, for storage and first pass processing. A subset of RAW data is copied to each Tier 1 site so that all RAW data is available at Tier 0 and at least one Tier 1.

First pass reconstruction is then run at Tier 0, producing ESD data. ESD data is divided into subsets and distributed to Tier 1 sites, each Tier 1 assumes responsibility for

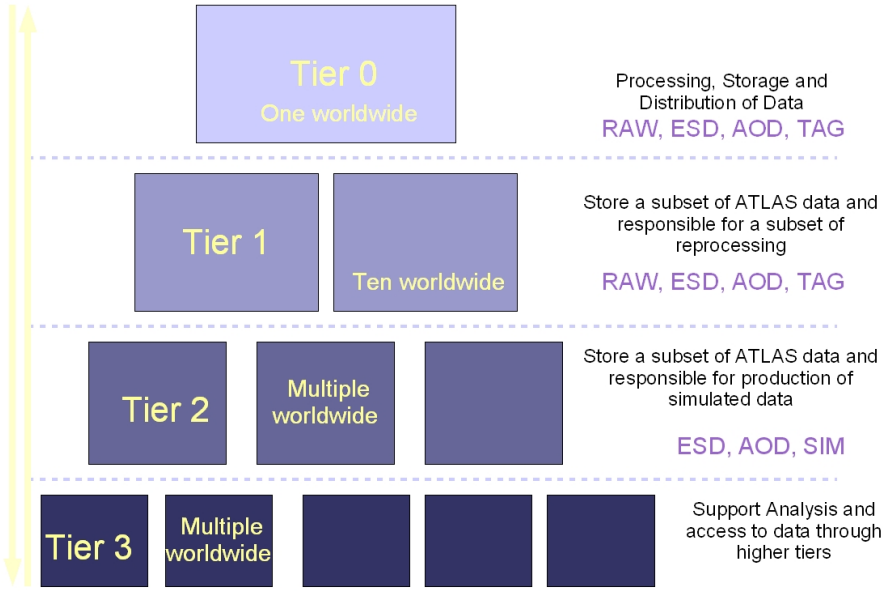



*Figure 5.2: ATLAS Tier Model with main roles of tiers and number worldwide*

an ESD set and holds a copy of a further subset of another Tier 1 site for the purposes of backup. As backstream navigation from ESD to RAW data may be needed, a subset of ESD data files will be sent to the Tier 1 holding the corresponding RAW data.

AOD files are then created from ESD at Tier 0, the AOD is archived at Tier 0, then replicated and a full set of all AOD is sent to each Tier 1 site, so that Tier 0 and every Tier 1 holds a complete set of primary AOD. The AOD at Tier 1 is copied and distributed to all associated Tier 2 sites.

The Distributed Database services in the Database Project provide physical database services and supporting software. The Distributed Deployment of Databases, 3D, project is an LCG project responsible for the facilities and software needs to establish the database service, replication and scalable access. ATLAS collaborates with the 3D project and ATLAS services are based on centralised writing and distributed reading. Database replication developed by 3D will be used for replication to Tier 1 and Tier 2. Tier 1 sites will offer Oracle and MySQL support, Tier 2 support is less encompassing, so distribution and replication must be as automated as possible. Mechanisms for data distribution are selective replication out of Oracle masters into MySQL and SQLite replicas, and dataset

subscriptions for file based data.

## 5.10 ATLAS Data Analysis

ATLAS analysts use Athena [36], a common analysis framework, to perform analysis of ATLAS data. Athena is based on Gaudi, an analysis framework designed for LHCb. Athena supports analysis of all event processing types, including simulation, reconstruction and physics analysis. Athena allows developers to attach analysis code to a general outline, providing common analysis functionalities and communication between software components needed for an analysis. Analysts configure Athena using a python script, jobOptions, allowing development of a specific analysis within the commonality of the analysis framework.

Athena JobOptions specifies features of an Athena analysis at runtime. Analysts specify input and output collections, the number of events to be processed in the analysis, the type of processing to be performed, objects to be saved within the analysis, message outputs, and analysis specific algorithms to run as part of the analysis. Athena is used extensively in the analysis presented in Chapter 10 of this thesis, a neural net analysis of the Higgs physics channel $t\bar{t}H, H \rightarrow b\bar{b}$

## 5.11 ATLAS Data Management

The ATLAS Distributed Data Management system [38], [39] and [40], is described in the ATLAS Computing Technical Design Report. The ATLAS Distributed Data Management system manages the movement and bookkeeping for all types of ATLAS file based data, including event data, non event conditions data and user defined groupings of data as sets of files, within a Grid computing environment. The environment and demands placed on the Distributed Data Management system are described in the Computing Model.

The project aims to integrate all Grid Data Management for ATLAS into one system, to manage all ATLAS file based data and to implement all ATLAS data flow as defined in

the Computing Model. This is achieved by developing an ATLAS specific software layer to interact with the grid middleware. The software layer is called DQ2. The DQ2 software provides a common interface for all ATLAS file based data management interactions

### 5.11.1 The Distributed Data Management system

The Distributed Data Management system has three main components. These are Dataset Catalogs, Site Services and User Client tools. The Dataset Catalogs are a bookeeping catalog interface implemented using Grid specific replica catalog middleware. Site Services are a file transfer service that operates at each distributed data management site and uses a database and Grid middleware, gridftp and srm, to manage file transfers. User client tools are a python client used by physicists for lookup and replication of data.

The system is based on interaction between these components. Users interact with the Catalogs through end user tools to search for datasets, define new datasets and to place subscription requests, the site services search the catalogs for new entries and transfer datasets accordingly, while the dataset catalogs collectively allow a full record to be kept of datasets within the system.

### 5.11.2 Datasets

The Distributed Data Management system centres on the concept of the Dataset. Datasets are an ensemble of file based data and some corresponding dataset metadata. Files are usually grouped by some common characteristic, such as detector run, physics criteria or usefulness to an analysis. Datasets are the unit of data interaction, manipulation and transfer within the Distributed Data Management system. As there are by definition many less datasets than files, datasets are intended to afford performance and scalability within the system. Data lookup at dataset granularity is inherently preferable in terms of performance than lookup of files.

Datasets are implemented with versioning and mutability. Versioning is intended to support small changes to the content of a dataset by adding new files. The mutability

state of a dataset defines whether files can be added to a dataset (open dataset), a new version must be created to add data (closed dataset) or if no data can be added and no new versions can be created (frozen dataset).

Datasets are identified in the Distributed Data Management system using three identifiers, these are Dataset Unique IDentifier DUID, Version Unique IDentifier VUID and Dataset name.

#### Dataset Unique IDentifier

Dataset Unique IDentifier, DUID, is a unique identifier assigned to each dataset by the distributed data management system. Each dataset has one and only one unique DUID identifier.

#### Version Unique IDentifier

A dataset may have many Version Unique IDentifiers, VUIDs. A dataset is issued with a new VUID for each new version of the dataset that is created. Information about the previous dataset version VUID is retained in the system when a new version is written.

#### Dataset name

Dataset name is a human readable name assigned by a user or by the production system to a dataset. The system requires that the dataset name be unique.

### 5.11.3 Files

Files are the unit of the ATLAS production system. Data from the detector, event data, conditions data and any other data are written initially to files. Files are then grouped together into datasets. Users can later create new datasets in addition to those created by the production system through an analysis. Any access, manipulation or movement of data using the distributed data management system must use datasets, not files.

Files can be contained in many datasets. The dataset is a concept of a grouping of files not necessarily implying or demanding physical co-location, although it is likely files in a dataset will be located in a common site. One file therefore when contained in many datasets on a single site may exist on the site in one physical instance only. Allowing datasets to contain common files does not imply redundancy of file data.

Files in the Distributed Data Management are not intended for user interaction, due to the scale of ATLAS data. Users are to interact with data in the unit of the dataset.

Files are identified by globally unique identifier GUID, logical file name LFN and physical file name PFN.

#### Global Unique Identifier

Every ATLAS file is assigned a Global Unique Identifier, GUID. The GUID is assigned by software and is a randomly generated 16 digit number.

#### Logical File Name

Every ATLAS file has a Logical Name, LFN, a readable logical file name. LFN can be assigned by a software system or by a user. The LFN refers to the file as a concept and has no information about any physical file location.

#### Physical File Name

An ATLAS file is assigned a Physical File Name, referring to any physical location of a file at a site. A file can have many PFNs if it is stored as many physical replicas at many sites. An identical file with many PFNs will still have a single LFN.

### 5.11.4 Data Movement

The Distributed Data Management system oversees and manages data movement between Tier 0, Tier 1 and Tier 2 sites by interacting with grid middleware and global and local dataset and file catalogs. Data movement can happen as part of the production system,

as well as a result of user requests, although the majority of data movement takes place as part of shared ATLAS data production. Data is moved in units of datasets, data should not be moved on a file by file basis.

As the system uses both global and local file catalogs as part of the data movement process, global and local catalogs need to interface in a consistent way. The content of local file catalogs are therefore managed by the Distributed Data Management system, although local storage implementation details and system support are managed at a local site.

The mechanism of data transfer is a Data Subscription. All data movement is triggered by the creation of a subscription. A Data Subscription is a transfer request, set by a production task or individual user.

### 5.11.5 Dataset Catalogs

The Distributed Data Management bookkeeping system centres on a set of dataset catalogs, some are global in scope and hosted at Tier 0 and others are local catalogs hosted at each site. The Catalogs collectively allow a full record to be held of all data. The Central Catalogs are global in scope and are divided by content into a Dataset Repository Catalog, a Dataset Content Catalog and a Dataset Location Catalog. The system also involves a global Dataset Subscription Catalog, a global Dataset Selection Catalog and Local File Catalogs.

#### Dataset Repository

The Dataset Repository is a catalogue of datasets, each dataset represented by one entry in the catalog and information about all versions of a dataset are stored. All datasets in the system are registered in the dataset repository. The Dataset Repository is the central ATLAS lookup for datasets, although for individual physics analysis and searches for datasets by dataset metadata users are expected to use the Dataset Selection Catalog.

### Dataset Content Catalog

The Dataset Content Catalog holds information about the logical file constituents of datasets. As this implies all ATLAS files are listed at least once in the Content Catalog, a central catalog with global scope, so scalability is an issue.

Organising files in groupings of datasets is a means of addressing scalability and the Content Catalog, by its nature, does not benefit in scalability through the dataset concept. The Content Catalog is however available in global scope to support some global file lookup with some reasonable efficiency, although the system in general is optimised for datasets.

The Event Level Metadata system, by its nature, demands a file level lookup. In chapter 7, implementations of a file based Event Level Metadata system with the dataset concepts of the Distributed Data Management catalogs are studied.

### Dataset Location Catalog

The Location Catalog holds information about the sites where a dataset is located.

### Dataset Subscription Catalog

All dataset transfer requests, complete and not yet complete, are subscriptions and are stored in the Subscription Catalog.

### Dataset Selection Catalog

The Dataset Selection Catalog has details of datasets and associated metadata. Users interact with the Dataset Selection Catalog. The Dataset Selection Catalog is global in scope and is not managed by the Distributed Data Management project. It is however a user interface to datasets so is relevant to the Distributed Data Management system. For ATLAS, the Dataset Selection Catalog is the ATLAS Metadata Interface, AMI [56].

#### Local File Catalogs

The global catalogs store file information at logical level, so by GUID and LFN, not physical level, PFN. The information about where a file is physically located on a Storage Element at a site is held locally in a Local File Catalog.

### 5.11.6 Site Services

Site services are software that run locally at each site. Site services are intended to manage movement of datasets and are not for use directly by end users. Site services run locally and are an interface between global and local systems. The interface to storage at a site is SRM, each DQ2 site points to a SRM storage area.

### 5.11.7 User Tools

Users interact with the Distributed Data Management system through end user tools, designed to support lookup of dataset information, definition of new datasets, requests for transfer of datasets by creation of a subscription. The user interface software is referred to as DQ2.

## 5.12 ATLAS Data Persistency

Persistence is an ability of an object to exist beyond the lifetime of the process that creates it. ATLAS implements object persistency using a transient data store, Storegate, a system to define when and by what means data will be written to transient and persistent storage, ItemLists and OutStreams, and a persistent storage project, the Pool Of persistent Objects for LHC, POOL. Persistent objects may be saved to files and relational databases using the persistence system.

#### StoreGate

As a transient memory store, StoreGate acts as an insulation layer between Athena and persistent data storage. Athena insulates physics code acting on event data from persitence technologies in an approach inherited from Gaudi on which Athena is based. In the model, physics algorithms read and write data objects from a transient in memory data sharing store, sometimes described as a blackboard of memory. In ATLAS the transient memory store is called StoreGate. Objects can be written from transient memory in StoreGate to persistent storage. StoreGate identifies objects as type and key and retrieval of an object from StoreGate is transparent in terms of storage technology, files or relational database.

#### ItemLists and OutStreams

ItemLists and OutStreams define when and by what means data will be written to transient and persistent storage. ItemList specifies which objects are to be persistified, items in the ItemList specify the values needed to retrieve an object from StoreGate. OutStreams define the writing of event to persistent storage and specify the output technology. An outstream is associated with an item list and optional event selection criteria. A job may have many OutSreams each with its own selection criteria, Itemlist and output technology, allowing a job to write events of interest to different streams for different physics groups with different policies about which event data objects are written to a stream.

### 5.12.1 POOL

ATLAS uses POOL, Pool Of persistent Objects for LHC [41], as a persistency project to provide a common persistency framework for LCG. POOL can store multipetabytes of distributed data and metadata in a grid enabled way. POOL can be used with both file and relational database data types, as POOL is a hybrid technology store, meaning C++ object streaming technology such as ROOT I/O are combined with relational database technologies. POOL is a distributed store and supports navigation between individual

data objects in files and relational databases. CORAL provides a layer of abstraction between POOL and software applications outside. POOL is made of components, a Storage Hierarchy, File Catalog, Storage service and Conversion, Object cache and references and Collections.

### 5.12.2 POOL File Catalog

The File catalog is a record of all POOL databases, where a POOL database is usually a file that stores objects, to resolve file references into physical file names which are then used to access file contents for processing and analysis. In the grid environment a File Catalog component is based on Replica Location Service RLS.

### 5.12.3 POOL Collections

The POOL Collection package is the user interface to an infrastructure for defining, creating, populating, using and maintaining ensembles of objects stored in the POOL persistency framework. A POOL Collection is a variable length list of references to objects whose states are made persistent in POOL storage.

The Event Collection is a central motivating factor for POOL collections. An Event Collection is an ensemble of event objects. Analysis typically use groups of events sharing some characteristics, rather than individual event objects. Analysis jobs using POOL therefore must be able to specify an event collection as input and equally to create and populate POOL collections as output. Collections, rather than individual objects, files and tables that contain the event objects, are the POOL unit of input and output.

POOL provides support for selection of objects within a collection without demanding the objects be restored or navigation within the collection. POOL supports extension of a Collection to include a number of attributes that may be queried. The attribute metadata is in the form of attribute value pairs to support user queries. POOL has an AttributeList component which together with the Collection infrastructure provides a system to support queryable object and Event Collections.

POOL supports Collections implemented in relational databases and ROOT files. Collections can be defined explicitly or implicitly. All POOL Collection implementations have a common POOL Collection interface, so Collections based on POOL ROOT files and on relational tables can be used in coexistence.

### Transient and Persistent Collections

A POOL collection is described as transient or persistent form depending on the way the data is stored. In a transient collection, the collection is an ensemble of the data objects, in a persistent collection the collection is an ensemble of references to the data objects. In a persistent collection, the objects are stored in ROOT or MySQL database. The metadata associated with the objects in a persistent collection are stored in the collection rather than the persistent storage with the objects.

### Explicit and Implicit Collections

Explicit Collections are ensembles of persistent objects where references to the objects and the metadata associated with the objects are stored using POOL Collection interfaces. Implicit collections meanwhile are ensembles of persistent data objects which are not stored using the POOL Collection interfaces. POOL can interact with implicit collections through an interface called ImplicitCollection.

### Collection types

POOL supports different collection types, where the type refers to the storage infrastructure used to store the persistent collection. POOL uses two types of databases for persistent collection storage, MySQL and ROOT, described in the following sections, corresponding to collection type MySQLCollection and RootCollection. If the collection is a collection of persistent data objects not stored using the POOL collection interface, the collection type is an ImplicitCollection. POOL supports multiple collection types at once.

## 5.13 ATLAS Data Storage

ATLAS implements two storage technologies for ATLAS data, file based storage and relational database storage. All event data in the Event Store can be described as file or relational data. The storage method selected for a set of data depends on the data and access patterns to be supported.

Event Level Metadata is a special case of event data in terms of storage as it is described in the Computing Model as being stored both as file based data and as relational data. The studies in this thesis focus on the development of the relational Event Level Metadata system.

### 5.13.1 Files

Files are a simple means of data storage and are used extensively throughout ATLAS, for event and non event data, in data production and in analysis. File storage is useful for large and small data and is inexpensive to implement.

ATLAS files interact with the ATLAS software environment, therefore ATLAS files map to C++ and object representations, as the online and offline software systems are object based, event data used by analysis and processing is object oriented and the ATLAS analysis software system, Athena, uses C++. The file based data system is implemented within the Distributed Data Management system.

Files are refered to as POOL ROOT files, as ROOT is accessed through the POOL persistency framework, for all ATLAS files. ROOT I/O is part of the ROOT project, [42], and allows C++ objects to be stored in files, through use of a C++ dictionary to all movement of C++ objects to and from files. ROOT is the bridge from transient data objects to files. POOL supports persistent file and object references which allow navigation to files that contain ATLAS event data objects and to objects within files. POOL File Catalogs allow ATLAS event data files to interface to the ATLAS computing environment, as POOL allows files to be used within a catalogued, navigable, distributed file system.

### 5.13.2 Relational databases

A relational database is a useful means of data storage when a system must operate in an environment of concurrent writing and consistent reading of data and when reading and writing of data are distributed, as relational databases can support central writes and distributed reads. Relational databases also support indexing of data and so fast structured queries can be supported. A relational database is also appropriate when the data and queries on the data would benefit from being stored in some structured fashion.

A relational database demands higher support overheads than a flat file system and some relational databases require licences, therefore can be supported at Tier 0 and 1 centres but may not be available at lower tiers or an individual user laptop analysis environment. For ATLAS, relational database systems are SQL type systems, Oracle [43] at Tier 0, Oracle and MySQL [44] at Tier 1 depending on licences and support, and MySQL and SQLite, a system combining SQL relational databases with a local file based storage [45], at lower tiers.

As database technology implemented at Tiers varies, a technology neutral database interface is needed to ensure that other ATLAS software systems can interact with relational databases without dependancies on technologies. CORAL, a COmmon Relational Abstraction Layer, replacing the Relational Access Layer, is a project developed within POOL providing an interface to relational databases which is not dependant on database technology. CORAL is an insulation layer which can be used to access Oracle, MyDQL and SQLite databases without knowledge of the database technology, so allows development of software to access data in a database independent of database technology to be developed on top of the CORAL layer. CORAL provides functionality for accessing data in relational databases without knowledge of database technology specific characteristics.

### 5.13.3 Coexistence of Files and Relational Databases

An Event Level Metadata system is unique in ATLAS as it uses files and relational databases. ATLAS data is stored as both files and relational database, depending on the

needs of the collaboration. All data is held in files while only some, such as Event Level Metadata and Conditions data, translated into relational tables.

## 5.14 ATLAS Data Navigation

In the ATLAS Event Store, where all event data is held, it is possible to navigate between data types for an event. The Event Store holds and makes available information that will allow access to upstream data. Upstream data refers to event data of a previous processing step, so for example an event in AOD format has upstream data in the form of the corresponding ESD and RAW event data. Upstream data can be useful to an analysis when a physicist wants to study in more detail the event objects and details of detector output. In the Event Level Metadata system, the TAG database, navigation to upstream data, in this case from TAG to AOD, is central to the usefulness of the system.

The mechanism for upstream navigation is a DataHeader. As an event is written using an OutStream, each event object in the associated ItemList is written. A master object, a DataHeader, is written. A DataHeader is a reference to where each individual physics object has been written, with its StoreGate identifier (type and key) and any information needed to restore the state of StoreGate. The DataHeader serves implicitly as the entry point to an event, if one retains a refererence to a persistent event it is in fact a reference to a DataHeader. A DataHeader also has references to DataHeaders from upstream processing steps which allows back navigation to upstream data.

## 5.15 Event and Non Event data

ATLAS data can be described as Event and Non Event data. Event data is RAW, ESD, AOD and event level metadata TAG data. All ATLAS Event Data is stored in the Event Store. The Event Store is a multipetabyte distributed system that uses file and relational database storage. The Event Store aims to be a scalable and performant system which can be easily navigated and accessed.

The Event Store is responsible for writing event data to and reading event data from the Event Store, for developing a navigational infrastructure, implementing suitable storage technologies and for the interfaces between systems and for users to interact with the Event Store. The ATLAS Event Store also encompasses an Event Level Metadata system, a TAG database, described in Chapter 6 and central to the developments in this thesis.

ATLAS non event data is Configuration or setup data, Geometry data, Detector Control System DCS data, Monitoring data and Calibration and Alignment data. Non event data are held in Configuration database and Conditions database. ATLAS offline computing will access the Conditions database, as will the Event Level Metadata system.

### 5.15.1 ATLAS Non Event Data

ATLAS non event data is used in data taking, reconstruction and processing [46], [47]. Non event data is used therefore by the online and offline computing system. All data accessed offline, for example by reconstruction or analysis, is stored in the Conditions database.

Non event data is Configuration or setup data, Geometry data, Detector Control systems DCS data, monitoring data, Calibration and alignment data and Conditions data.

Configuration or setup data is used in the online system including subdetectors, TDAQ, Event Filter and DCS system. Configuration data consists of all data needed to setup and operate the experiment. Data is collected for each data run.

Geometry data gives the physical geometry parameters and location of the components of the detector including survey information. The data is not expected to change except during installation or major changes to the detector. The data is used by the high level Trigger, offline reconstruction and detector simulation.

Detector Control System produces DCS data, digitised analogue readings of temperatures, pressures, high voltages and state transitions such as systems changing mode and switching on and off. DCS data comes from the subdetectors, is accessed by time stamp

and is not synchronous with data taking. It is used by the high level Trigger, the Event Filter and offline reconstruction.

Monitoring data is data derived from the event stream to monitor the performance of system components. Monitoring is intended to verify data quality. Monitoring data is derived from the real event stream and is noted by time stamp.

Calibration and Alignment data includes all the constants needed to run reconstruction and offline analysis other than fixed numbers in the detector geometry.

The offline reconstruction software only accesses the conditions database not the online configuration database. The data comes with a timestamp and a version, many versions may exist for same event data for improvements in calibration calculations.

Conditions data is produced by online monitoring and calibration, high level Trigger, Event Filter and is sent to the Conditions database by the Configuration database. Conditions data is needed for prompt reconstruction and offline computing and some data will be fed back to the high level Trigger. Non event data are held in the Configuration database and Conditions database. Offline computing accesses the Conditions database and data is sent to the Conditions database from the Configuration database.

### 5.15.2 Non event data Databases

Non event data is stored in the Configuration and Conditions Database. The Configuration database stores data needed for the current and next run of the ATLAS DAQ including all relevant setup data and a subset of currently valid conditions data. The configuration data acts as a source for conditions data and used the Conditions database as an archive.

The Conditions database stores all data needed for offline reconstruction and analysis of event data, all calibration and alignment data, geometry information and setup information. It is used as an archive for DCS and monitoring data which may also be used for offline analysis.

The Conditions database is a heterogeneous structure incorporating many database

technologies and is accessed through a common interface, the COOL API. The Conditions database is as a result referred to the COOL Conditions database.

### COOL

COOL is a C++ API for reading and writing conditions data and implements a common persistency solution to store and manage conditions data. Conditions data is calibration, alignment and slow control data and are non event data describing the detector at data taking. The COOL Conditions database is responsible for storing almost all non event data needed to operate the detector and perform reconstruction and analysis, stores DCS detector control system data or slow controls data, data such as voltages, currents, temperature and status information produced by the control and monitoring systems.

The COOL Conditions database stores online bookeeping data, types of run, number of events and files, detector and software configuration, used by offline reconstruction and data management, online configuration and calibration data, parameters needed to operate detector online system, calibration constants and Trigger thresholds, offline calibration data, used to generate calibration constants to be used in later reconstruction, and monitoring and histogram data.

COOL implements an Interval of Validity, objects stored and referenced in COOL have a start and end valid time. COOL data are stored in folders arranged in hierarchial folder sets, times are specified as run and event, or as a timestamp. COOL has SingleVersion where an object is valid at a time and Multiversion where many objects can be valid for the same time and are distinguished by version, for example calibration data where several calibration sets are valid for the same run each corresponding to a different processing pass or calibration algorithm. COOL implements each folder as a relational database table and each stored object is a row in the table. COOL can reference data stored elsewhere, for example a POOL object which allows an external object to be associated with an Interval of Validity, useful for calibration data which may be large and have complex structure and be created and processed as a C++ object.

COOL is implemented using CORAL which allows database applications to be written independently of the underlying database technology, so COOL databases can be Oracle, MySQL and SQLite. The master conditions database at CERN is Oracle as are most Tier 1 replicas, Tier 2's will make available subsets of conditions database using MySQL, so a user interacting with alternative technologies needs only a different connection string.

COOL provides a C++ API and an underlying database schema to support the data model. Once a COOL database has been created and populated users can interact with the database directly using database tools.

## 5.16 Summary

This chapter has described the ATLAS Computing Model. We have discussed ATLAS event data types and sizes, data production, processing and reprocessing, ATLAS Distributed Computing, the ATLAS Tier Model and flow of ATLAS data throughout Tiers. The ATLAS Distributed Data Management concepts and system has been introduced and ATLAS data storage, persistency, navigation, and non event data described. We have addressed the data themes of type, access, creation, storage, persistency, navigation and management. The POOL persistency project has been presented as has COOL, the non event conditions data system. Files and relational databases have been introduced as means of storing ATLAS data, and the benefits and uses of each described. We have presented Athena, the ATLAS analysis software for data analysis. We now move on to discuss the Event Level Metadata software system of the ATLAS Computing Model, a system in which event data in the form of TAG data are captured during ATLAS data processing and offered to users for event selection and analysis in both files and relational databases. The Event Level Metadata system interacts with all the components of the ATLAS Computing Model system presented in this chapter, as the Event Level Metadata System is impacted by, interacts with or directly uses all of the components described. Developments and studies of the Event Level Metadata system are the focus of the development studies in this thesis.

# Chapter 6

# ATLAS Event Level Metadata

## 6.1 Introduction

The ATLAS Event Level Metadata system, referred to as the ATLAS TAG Database, is a multi terabyte system within the ATLAS software, analysis and distributed computing environment. The Event Level Metadata system is introduced in only general terms in the Computing Model. The realistic design and implementation of the system, the feasibility of an Event Level Metadata system at ATLAS scale and the realistic and practical use of a TAG system to analysts within the collaboration was studied and developed in the years leading to ATLAS start up. This chapter presents developments in the understanding of a realistic Event Level Metadata implementation in the time leading to startup and presents the outcome of these studies, the current Event Level Metadata system design.

Event Metadata TAGs are defined and their purposes in the ATLAS experiment are discussed. The TAG Database system, its structure, interaction with other relevant components of the ATLAS software and analysis system, both on and offline are outlined. Interactions with users, in both a current and planned context, are introduced. Use cases for Event Level Metadata and the TAG Database are discussed. After discussing the state of the system at startup, current ongoing developments and future plans for development of the system are outlined.

## 6.2 ATLAS Event Level Metadata

The ATLAS Computing Model describes an Event Level Metadata system called the ATLAS TAG Database. The role of the TAG Database is to support seamless discovery, identification, selection and retrieval of ATLAS event data held in the multipetabyte distributed ATLAS Event Store. The Event Level Metadata system captures information about ATLAS physics events on an event by event basis and offers later access to the events through an event metadata search. In the model, every ATLAS physics event selected by the online Trigger system as potentially interesting has Event Level Metadata, and event TAG, or information about each event, written to correspond to the event data. As events are selected by the online ATLAS Computing system, event data is written and stored in increasingly reduced formats. Event Level Metadata is constructed using AOD event data at AOD production, or merging for smaller AOD files. At 1kB per event, an event TAG is the most concise event data to be created.

Each event TAG contains event level metadata attributes defined by the Physics Analysis Groups and TAG Database development group. The attributes are chosen on account of their potential to support selection of events and navigation within the system.

Event Level Metadata attributes are grouped into event identification and global event properties, Trigger information, quality information, temporal information and some high level physics object information. The content is intended to support efficient and useful selections across a large data sample, not direct analysis on TAGs. Tests have shown that there can be considerable advantage to selecting events from the TAG Database and using these as input to analysis, compared to running over a full AOD sample [53].

Events returned to the physicist will be those events which satisfy the query, based on the TAG attributes defined therein, and no others. The result set will include pointers to event data, which can then be used as input to analysis. The pointer is the GUID of the file and the Dataheader to locate an event within a file. TAGs contain sufficient navigational information to allow direct navigation to the event data at all upstream processing stage, currently AOD and ESD, as RAW data is bytestream rather that object

format, therefore the data header object method used to locate events in a file cannot be used. Event Level Metadata also has information to allow retrieval of qualifying Event Level Metadata itself.

Event Level Metadata are held in both files and relational databases. Collectively the file and relational database resident collections are known as the TAG Database. As Event Metadata is created using AOD event data, TAGs are written to ROOT files. These files are then used to populate a queryable relational database. Initially file based TAGs were a means of introducing latency to the TAG creation stages of production, and were intended for use only to populate relational tables, but file based TAGs have proven useful for the physicist too. For this reason the lifetime and use of file based TAGs persists beyond population of relational tables.

In order to be useful and reliable, the ATLAS TAG Database must support fast, efficient and accurate querying, massive data volume, a demanding update rate and efficient navigation from event metadata to event data itself - this in essence is the challenge of the TAG Database. In the Event Level Metadata systems the very different challenges of fast and efficient data access, up to date and reliable data storage, fast data upload, accurate and reliable database management and seamless navigation to upstream data must coexist.

## 6.3 TAG Database System

An outline of the ATLAS TAG Database system is shown in figure 6.1. At the stage of AOD production or AOD merging by the Tier 0 production system, POOL Collection Utilities are used to create TAG files. The TAG files are then used to populate relational tables. The file to relational database loading steps are managed so that the relational database tables are populated in a controlled way. As AOD files are copied to Tier 1 locations, relational TAG collections can be populated at Tier 1 sites. At Tier 0 the relational database management system is Oracle, at Tier 1 it is Oracle or MySQL, depending on whether the Tier 1 site has support for Oracle databases. A user interface,

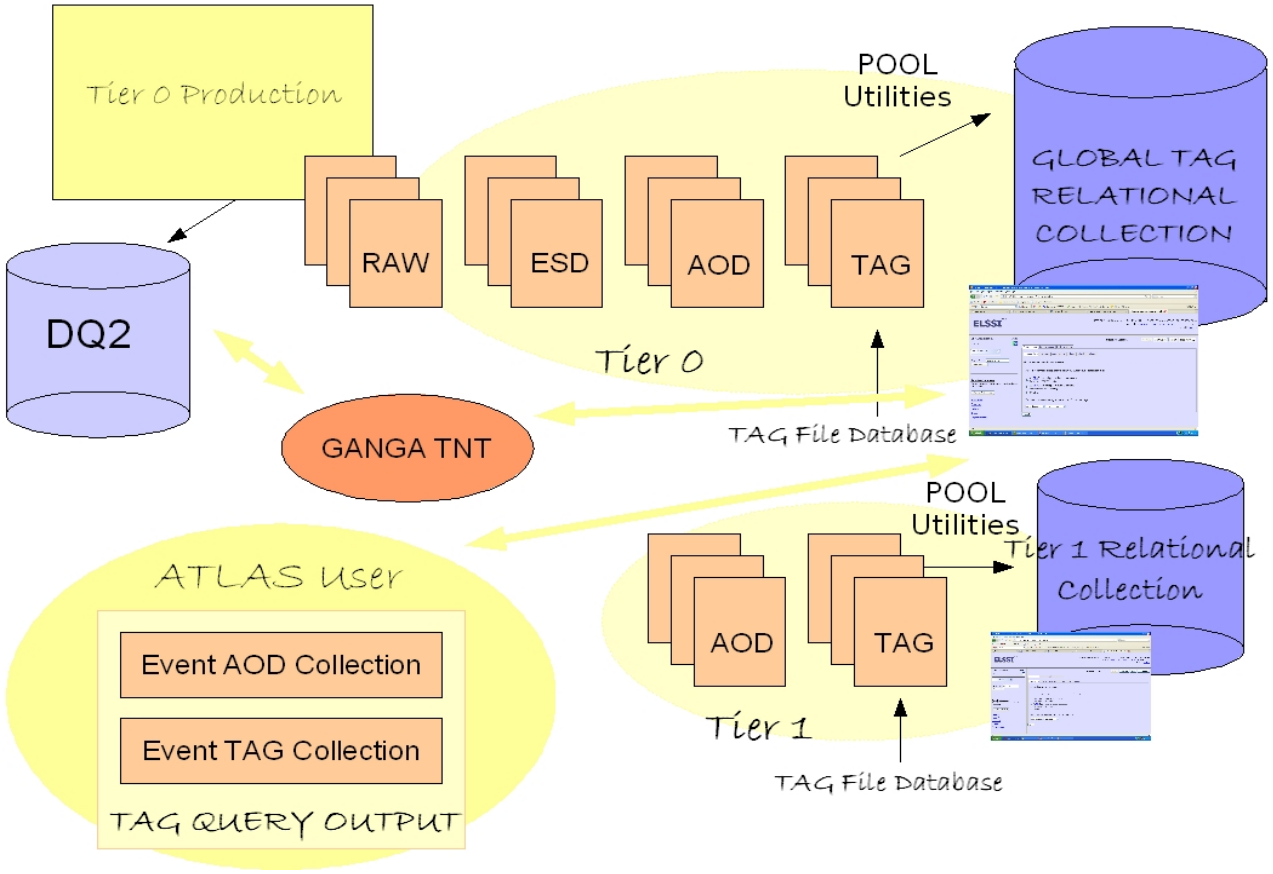



*Figure 6.1: ATLAS TAG Database System, showing Data Production at Tier One, population of a Global Relational TAG Collection and the User Web Interface to the Relational Collections. The system allows the ATLAS analyst to access the relational collections and use the web interface and GANGA-TNT to create TAG query outputs, collections of event TAGs and event data*

ELSSI, described in chapter 8, is the central access point for users to access the relational TAG Database Collections. ELSSI instances can be installed on servers at Tier 0 and Tier 1 sites and may point to central or local TAG Collections. When a user submits a query to the TAG database through ELSSI, the ELSSI server contacts GANGA-TNT [53], then in turn DQ2, to locate the TAG and AOD files corresponding to events satisfying the user query. A result set is then grouped together and returned to the user for analysis.

## 6.4 TAG Use Cases

Event Level Metadata TAGs are intended to support efficient and useful analysis for physicists. A number of use cases for event TAG queries are planned for ATLAS, based on early ATLAS Event Level Metadata plans [48] [49] and ongoing learning as the system develops. The central use cases are

- Information and statistics without opening data files
- Cross stream selections
- Cut refinement without opening data files
- Create physics group skims
- Access to ESD or RAW data for an event selection
- Event selection with quality information

A query on the TAG database allows a user to gather statistics about event data without opening the larger AOD files. It is possible to query across a large data sample to ask how many ATLAS events satisfy some general characteristic, where the query would realistically return too many events, or require too many AOD files to be opened if the query were attempted on event data. Using event Level Metadata however, larger scale statistic queries are both possible and useful. TAG queries allow queries across ATLAS stream boundaries, due to the fact that if an event is only written to a higher priority

physics stream, event metadata in the TAG will store information about all streams for which an event qualifies.

A query on event TAGs also allow a user to count how many events will satisfy a given cut, allowing refinement of the cut should the query return too many or too few events for the data to be useful. Physics working group leaders can use the TAG database system to create samples for working groups, containing preselected events useful to shared analysis. Accessing upstream data, ESD and RAW data, is also possible through a TAG query, as references to the upstream data is held in TAGs. Event selection is also possible with updated quality information, added at reprocessing stages.

## 6.5 TAG Data Volume

The ATLAS Computing Model assumes 200 days of data taking per year, 50000 active seconds per day (58% efficiency per day), ATLAS expects an event rate from the HLT of 200 Hz, $10^7$ events per day. As the current budget for TAGs is 1 kB per event, the TAG Database is a terabyte scale system in volume. Anticipated TAG Database storage requirements are shown in table 6.1 [55]. The scale is small compared to the Event Store and other Event Data types, but unlike other event data types, TAGs must be readily queryable, to provide both statistical information about events and produce event collections for analysis.

| Year | Percentage of Year for Data Taking | Amount Data |
|---|---|---|
| 2008 | 40 | 1.42 TB |
| 2009 | 60 | 3.65 TB |
| each additional | 60 | 6.09 TB |

*Table 6.1: TAG Storage Requirements*

## 6.6 TAG Data Rate

As well as supporting the terabytes of data volume and allowing reading of the data, the TAG Database must also allow writing of events on a large scale. Data are produced by

the detector at a rate of 200 Hz, therefore during active data taking the database must accept on average 200 new entries per second. In order to avoid contention between read and write operations in the database, the files into which event TAGs are first loaded are later used to populate the relational database in a controlled and managed way. In this way, TAG files introduce latency to the system.

## 6.7 TAG Database Distribution model

The distribution model for Event Level Metadata follows that of AOD data. A global TAG relational database implemented in Oracle will be built at Tier 0. A series of duplicates of the relational global collection will be sent to Tier 1 centres, implemented in Oracle or MySQL depending on the resources available at each site. The replicas will serve as both backup and support, as user queries may be shared between database instances. A full set of file based TAGs will be held at Tier 0, with copies being sent to each Tier 1. As the file based TAGs may be used to populate a relational TAG Database, it is foreseeable that relational database instances can be created at lower tiers using file based TAGs.

## 6.8 TAG Writing

The LCG POOL Collection infrastructure is used to implement the system and both file and database resident tags use POOL Collections. The ROOT tree is the fundamental unit of file based TAGs, providing a simple means of capturing Event Level Metadata in files, and the POOL relational collection structure provides a foundation for relational database based TAGs.

TAGs are written at the AOD merging stage of AOD data production. Event Level Metadata are written initially to POOL files as Explicit Collections. TAGs are then imported into relational database tables at a later time in a controlled way, so concurrency and write access to relational database tables can be managed. Initially file based TAGs were a means of introducing a latency to the system, but TAG files emerged as useful,

so are made part of the TAG database system. TAGs therefore emerge firstly as file resident TAGs in POOL Collections and are later imported into a relational database. TAG building algorithms are developed by the ATLAS Physics Analysis Tools group and are used to write TAG attributes.

## 6.9 TAG Attributes

An Event TAG is a grouping of event metadata, called TAG Attributes. Attributes can be grouped into event identification and global event properties, trigger information, quality information, temporal information and some high level physics object information. The content is intended to support efficient and useful selections across a large data sample, not direct analysis on TAGs.

A TAG, as well as containing event level metadata describing an event, contains a pointer to the event AOD data in POOL resident files. The pointer is the GUID of the file to which the event is written and the DataHeader information about where in the file the event can be found, since many events are written to a single file. The GUID is the key to navigation from each event TAG to the corresponding event data.

Physics properties describing each event are stored in the TAG for later identification by a Physicist for analysis and the Event Level Metadata attributes to be contained in the TAG to support querying for events of interest are selected by Physics Analysis Groups.

The attributes can be defined in terms of

- File handling (GUID, AOD Data header)
- Event Basics (Event number, Run number)
- Trigger Criteria
- Overall Physics Criteria
- Physics Group Criteria

It is the Physics Criteria that will be used by Physicists for event selection, file handling and event basics attributes are used for system specific operations and may eventually be hidden from a Physicist.

## 6.10 TAG Back Navigation

The Event Level Metadata system incorporates the ability for back navigation. Collections are created after each processing step so selection of data from upstream processing stages is made possible through an Event Level Metadata query. A selection on the TAG database can provide a list of references to corresponding ESD events for input to a later job, without any need to open the AOD data files to find the ESD references, so making the process of creating an ESD event list significantly more efficient.

## 6.11 TAGs and POOL Collections

POOL Collections are a central part of the TAG Database system. POOL Collection Utilities are used to write relational TAGs to relational database tables using TAG files as input and to create user output TAG event collections to user queries performed on the TAG database.

An event of interest to multiple analyses will be written once and the event TAG can be written to many collections. As TAGs are much smaller than AOD event data, this is an efficient means of accessing event data across many users and analysis. Alternatively a TAG can be written to a single collection and subsequent queries can be used to build multiple collections corresponding to many analysis specific selections. The small TAG size and the POOL Collection structure make these strategies realistic and useful. POOL Collections provide a useful and efficient means of selecting objects within a collection, using the Attributelist within file TAG event data.

## 6.12 TAG size

The size allocated by the Computing Model to a TAG is 1KB per event, many times smaller than typically used for analysis AOD data. It is the relatively small size of a TAG that allows creation of a Global TAG database and, as a consequence, fast and efficient preselection of AOD files for analysis.

## 6.13 User Interaction with the TAG Database

Users interact with the TAG relational database primarily through the Event Level Selection Service, ELSSI, [50], a web based interface that allows users to browse available Event Leve Metadata, create an Event Level Metadata query, perform a query and return result sets for subsequent analysis. The development and features of the ELSSI system is described in chapter 9.

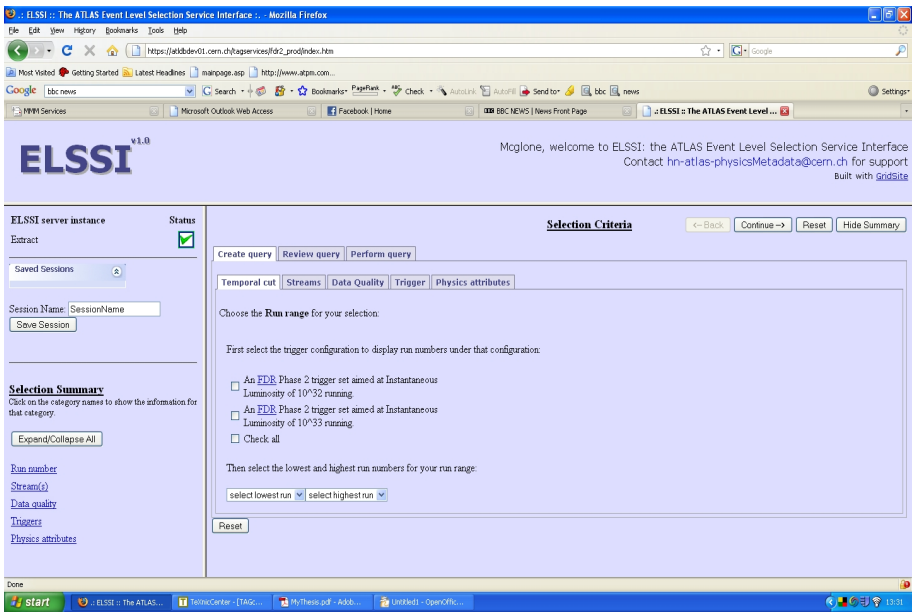



*Figure 6.2: The ATLAS ELSSI Interface, the User Web Interface to the Relational TAG Collection allowing access to ATLAS TAGs*

## 6.14 TAG Database at LHC Startup

At LHC startup in 2008, the TAG database system was a well developed and integrated part of the ATLAS software and analysis system. Scalability tests, presented in Chapter 8, demonstrated that a relational TAG database at ATLAS scale is a feasible outcome. The TAG system has been integrated with the Distributed Data Management and Trigger systems, described in Chapter 7. The ELSSI interface system has been presented to ATLAS users at collaboration meetings and tutorials and users have been introduced increasingly to the use of the TAG system. Event TAGs have been sucessfully created in TAG files at Tier 0 Production and imported into relational tables using the POOL Collection Utilities. Data for Streams Tests and two FDR data runs have been imported into relational tables and are available to users through ELSSI interfaces.

## 6.15 Conclusions and Future Directions

The Event Level Metadata system is introduced in the Computing Model. This chapter has presented developments in the understanding of a realistic Event Level Metadata implementation in the time leading to startup and presents the current Event Level Metadata system design. Event Level Metadata has been introduced in the context of physics analysis and the Computing Model. The TAG Database system has been presented and its components and their interactions described. Use Cases that have developed in implementation and understanding through development and study of the realistic system have been presented. Metrics for Volume and Data Rate of TAG data have been presented, setting out the environment in which a TAG system must function. The TAG distribution model is outlined in the Computing Model and has been studied in practical and realistic terms in the time leading to start up, the learning from this study and the current realistic distribution model has been presented. TAG writing has been described as has TAG content, again merging the system outlined in the Computing Model with the realistic implementation of a TAG system. Interactions with users, in both a current

and planned context, have been introduced. Use cases for Event Level Metadata and the TAG Database in a current and future context have been discussed.

In the next three chapters we discuss in more detail the developments and studies that lead to the understanding and implementation of the current Event Level Metadata system, which started with the general description of a system in the Computing Model and now is a well understood and performant terabyte scale system, merged with the components of the ATLAS software system and used by analysts to access and study ATLAS data. Chapter 7 focuses on the feasibility of merging the TAG system with two central features of ATLAS software, distributed computing and analysis, the Distributed Data Management system and the ATLAS Trigger system. Chapter 8 presents studies on implementation and performance of a realistic terabyte scale relational TAG Database system. Chapter 9 presents the development of a user interface to the TAG Database, the Event Level Selection Service Interface.

# Chapter 7

# ATLAS TAG Database Feasibility Study

## 7.1 Introduction

An Event Level Metadata system for ATLAS poses a number of challenges. At the start of system development, it is necessary to firstly ascertain whether a TAG Database system is feasible within the larger ATLAS software system, as an Event Level Metadata system is not an independnt analysis system. There are major, established and important components of the ATLAS system with which the TAG Database must operate smoothly, otherwise the TAG Database system itself will be an unfeasible prospect in its form described in Chapter 6. This chapter discusses feasibility studies undertaken in the primary steps of TAG Database development, considering the feasibility of function of a TAG Database within the ATLAS Distributed Data Management system and the ATLAS Trigger system. The concepts and design of the Distributed Data Management system have been presented in Chapter 5, and the Trigger system in Chapter 4. The impact of the operating environment on the TAG Database system and the necessary developments that must be undertaken so that the TAG Database can operate in the ATLAS environment, with a focus on the Distributed Data Management system and the Trigger system, are presented in this chapter. As we establish feasibility, steps taken to implement inte-

gration the TAG database with the ATLAS Distributed Data Management and Trigger systems are presented.

For merging the Event Level Metadata system with the Distributed Data Management system, central challenges are implementing the dataset concept used in the Distributed Data Management system with the file concept used in the Event Level Metadata system. To study this challenge and develop workable solutions, we study the DQ2 catalog schema and consider how to work within this system with Event Level Metadata. We also consider adding a dataset attribute to Event Level Metadata to implement a bridge to the DQ2 dataset based system. We then study the impact of implementing file lookup within the DQ2 system and we study and develop a subscription method, a Distributed Data Management process of data collection. We then develop and optimise a subscription method that we can use in the Event Level Metadata system to return Event Level Metadata query output to users within the Distributed Data Management scheme.

For merging the Event Level Metadata system with the Trigger system, we consider the challenges of implementing time varying Trigger menus with Event Level Metadata in relational tables, and we propose a relational solution, which is later implemented to implement time varying Trigger menus in the Event Level Metadata Interface, described in Chapter 9, available to analysts.

## 7.2 Merging a TAG Database with ATLAS Distributed Data Management

The Distributed Data Management system for ATLAS manages all ATLAS event data. For the TAG Database to be a feasible system in ATLAS, it must operate within the Distributed Data Management system. As the Distributed Data Management system is a dataset based system and the TAG database uses a file based lookup, the process of investigating the feasibility of a TAG Database in a dataset environment and of developing the TAG database infrastructure within the dataset environment is an important step in

the development of a feasible TAG database.

In this chapter the feasibility and praticalities of file based lookup in the Distributed Data Management system are investigated. Understanding the Catalog schema for TAG queries is an important feature of the TAG database development. Adding a dataset attribute to TAG is considered as a means of bridging file based lookup in TAGs and dataset based Distributed Data Management catalogs. The usefulness of the dataset attribute in TAGs strategy is considered. The Distributed Data Management Subscription method for data movement is considered in relation to TAG queries and a method for gathering of output data is developed and tested.

A means of merging TAG files with DQ2 concepts of dataset, data movement and cataloging is needed. This is facilitated by adopting the dataset concept and developing a method for dataset creation and transfer between sites

The DQ2 Catalog schema evolves as the Distributed Data Management system evolves and the impact on the Event Level Metadata system varies with this evolution. It is important to understand the ways in which Event Level Metadata can be implemented with DQ2, the performance impacts on Event Level Metadata and Distributed Data Management in doing so and the feasibility of merging the two systems.

## 7.3 Datasets in DQ2 vs Files in TAG Database

In the Distributed Data Management system, data is grouped, transfered and tracked in units of dataset. The dataset is a means of ensuring scalability with a system which must catalog and manage all ATLAS event data. File based lookup was discouraged, as there were concerns as to the impact on scalability and performance as the system is not optimised for file based lookup.

A query to the TAG database returns a pointer to each event by file GUID. File based lookup is therefore central to the TAG database system. Whilst the Distributed Data Management is an established and central part of ATLAS, the TAG Database is in its infancy at the time of the studies presented in this chapter, therefore the development

of the TAG database from the start must be driven by an understanding of the Data Management system and an effort to function within the Distributed Data Management dataset environment, in a way optimal to a TAG database system, making optimal use of the Distributed Data Management system and without putting unacceptable demands of the Distributed Data Management system, [51], [52].

### 7.3.1 Important terms

The concepts and terms of the Distributed Data Management system are discussed in Chapter 5. The important terms for the studies in this chapter are

- GUID - *Global Unique Identifier, a unique file identifier*
- LFN - *Logical File Name, a user readable file identifier*
- VUID - *Version Unique Identifier, a unique dataset version identifier*

## 7.4 Understanding the DQ2 Catalog Schema when implementing TAGs

The TAG Database must use the Distributed Data Management system to translate file GUID from a TAG query into locally gathered event data files. So the input to DQ2 from the event Level Metadata system is GUID of a file. To translate the file GUID from a TAG into output query results in files and datasets using the Distributed Data Management system, a set of information about the files of interest is needed from the Distributed Data Management Catalogs. To register a file in a new dataset, GUID and LFN are needed. To locate the file, with a view to copying the file to a local site or send jobs to the event, the file location is needed.

The catalogs of interest are the Dataset Content and Dataset Location catalogs. We attempt to assess the feasibility of file lookup in these catalogs and improve the lookup performance where possible.

## 7.5 Adding a Dataset attribute to a TAG

An attempt to optimise the file lookup is considered by searching not just by GUID, but by GUID:VUID pair. The VUID could also be used to identify Dataset and therefore file location information in the Dataset Location Catalog. This would mean capturing some dataset information at TAG writing time. This is not without performance overheads, so we first assess how much time is saved in a TAG process if the VUIDs are known in advance.

## 7.6 VUID in TAG Tests

A python script was written to perform all the necessary DQ2 steps in the TAG process. The script performs a query on the Rome TAG Database MySQL Collections and returns a number of GUIDs. The GUIDs are used to lookup all the necessary information in the DQ2 Catalogs, then perform a data subscription using an Incomplete Subscription method, identified as optimal later in these studies, to deliver the files to the local site. The performance implications of providing a VUID and GUID in advance as if the VUID was a TAG attribute, compared with providing only GUIDs, are considered. In the test model the DQ2 Catalog schema and the client API were used as they are, so without any adapted methods. Only the client methods already available in the DQ2 client were used.

### 7.6.1 VUID lookup vs VUID in TAGs

Figure 7.1 shows the results of comparing the performance of DQ2 TAG steps with and without the VUID lookup. The plots in Figure 7.1 shows identical tests performed independently using the DQ2 central catalogs and file lookup queries on five days, comparing lookup when VUID is known prior to lookup, so mimicking a system where the VUID is available as Event Level Metadata in the TAG, and lookup when the VUID information is not known in advance, so not held in the TAG.

To create the results in Figure 7.1, queries were repeated for many iterations and

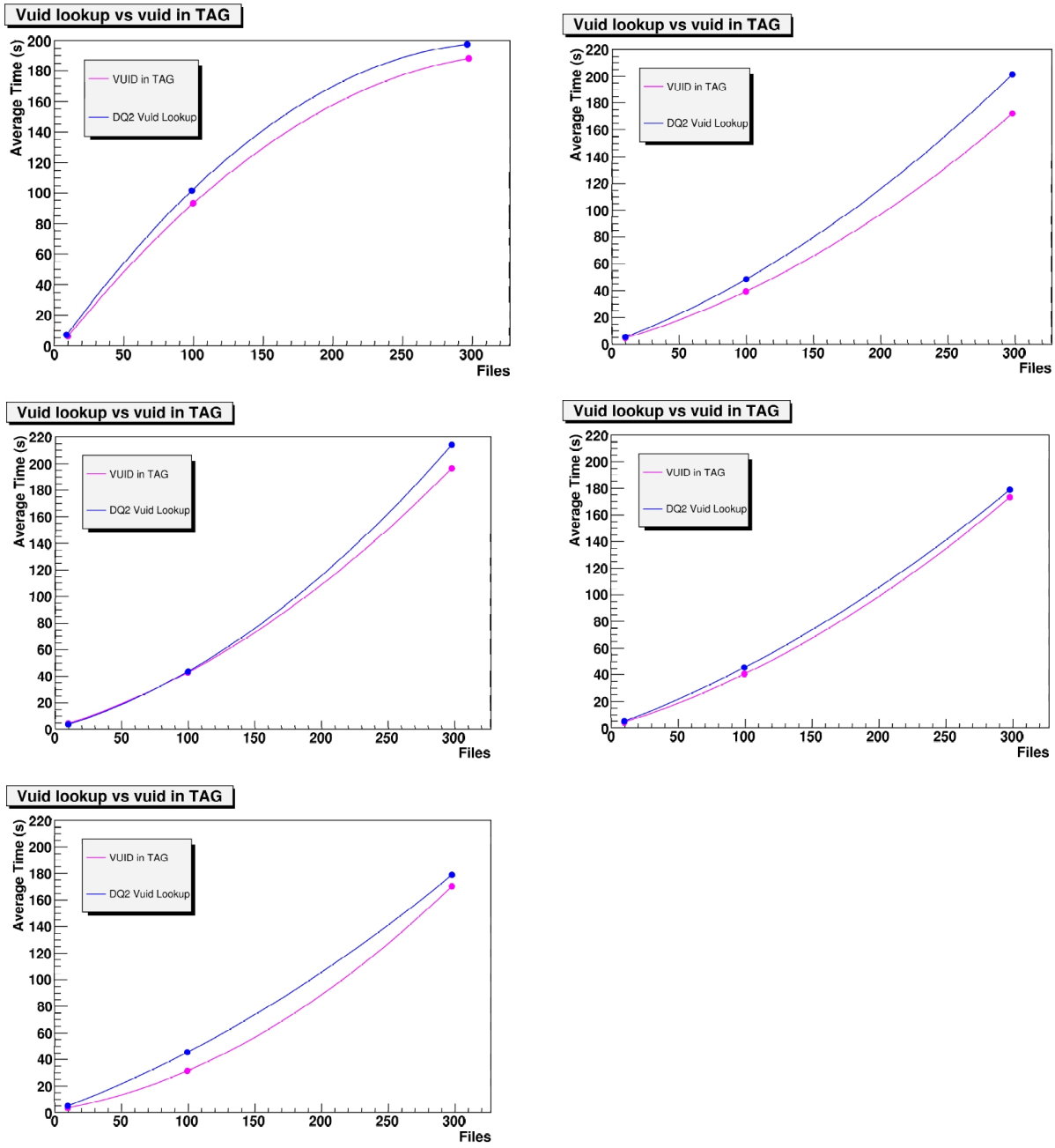



*Figure 7.1: Comparing the performance of a file lookup in the DQ2 Catalogs when the dataset containing the file is known in advance and stored in the Event TAG and when the dataset is not known in advance. Tests were performed on five independent days to assess performance over a varying outside server load. The graphs show that there is not a significant advantage in performance when a containing dataset is known in advance*

averaged. Tests are then performed to study the lookup process timing across single days, to see whether variation in response times with and without a VUID attribute simulated in TAGs throughout a day of testing. The results show that at the time of

tests there is no significant advantage in file lookup in the cental DQ2 catalogs when the VUID is known in advance.

### 7.6.2 VUID DQ2 lookup vs VUID in TAG, tests throughout day

The graphs in Figure 7.1 represent queries that are repeated for many iterations then averaged. Investigating closely we see that the response time varies during testing. The fluctuations are shown in Figure 7.2.

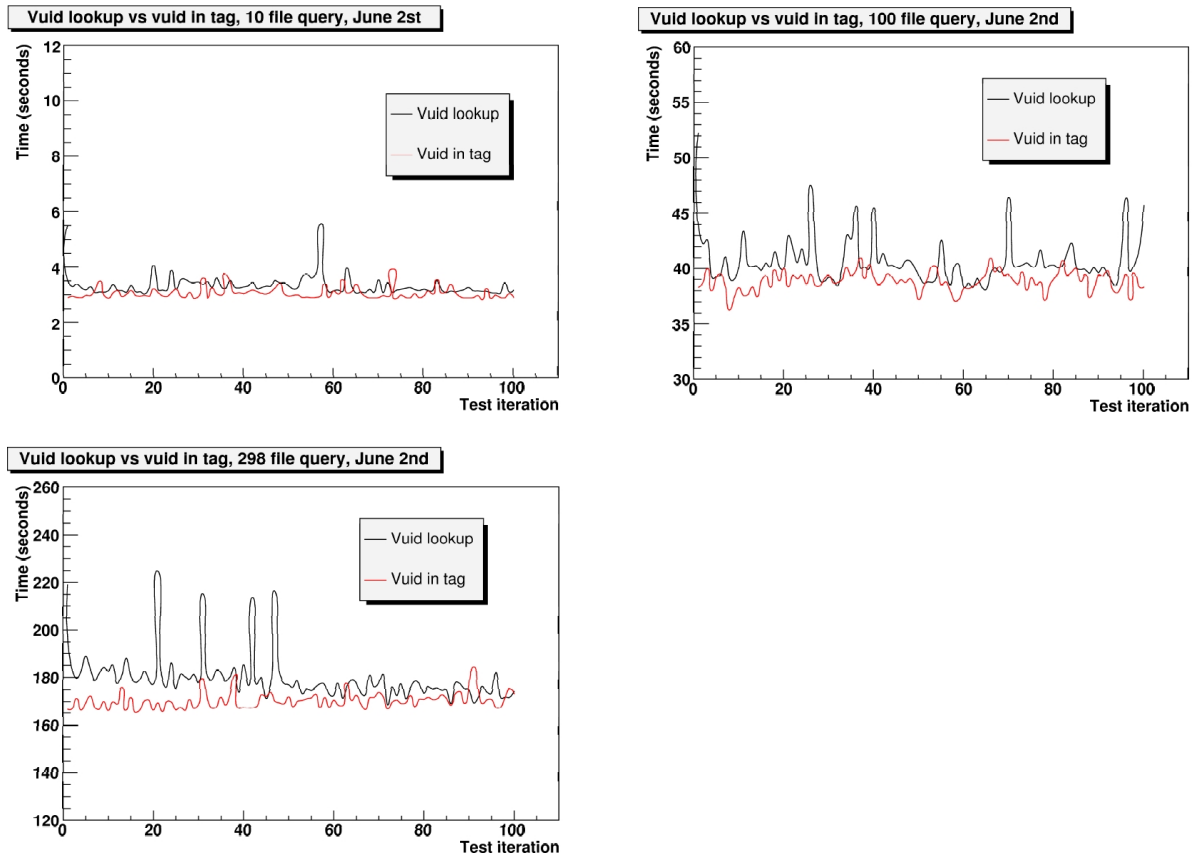



*Figure 7.2: Comparing the performance of a file lookup in the DQ2 Catalogs when the dataset containing the file is known in advance and stored in the Event TAG and when the dataset is not known in advance, individual lookup times, we see a larger fluctuation in response time when a dataset is not known in advance*

It can be seen in Figure 7.2 that the fluctuations in response times are larger in the case where a lookup of VUID is involved, suggesting that the VUID lookup introduces a

degree of variation. These results indicate that a VUID attribute in an Event Metadata may be a useful feature, as it allows a more consistent and therefore predictable response time from the DQ2 central catalogs when a file lookup is performed.

### 7.6.3 Impact on overall time as a result of VUID lookup

The response times measured in the comparison of VUID lookup with VUID in TAG include steps other than just the VUID lookup. An assessment is made of the overall impact of providing a VUID in TAG and removing the VUID lookup step by considering the percentage of time saved in the overall process. Figure **??** shows a comparison of a TAG database query, in this case a query on the MySQL collections with the DQ2 VUID lookup and the total time taken for a TAG query through to DQ2 interaction and subscription of TAG query result files to site, for the case where one hundred files are returned by the query to the TAG database.

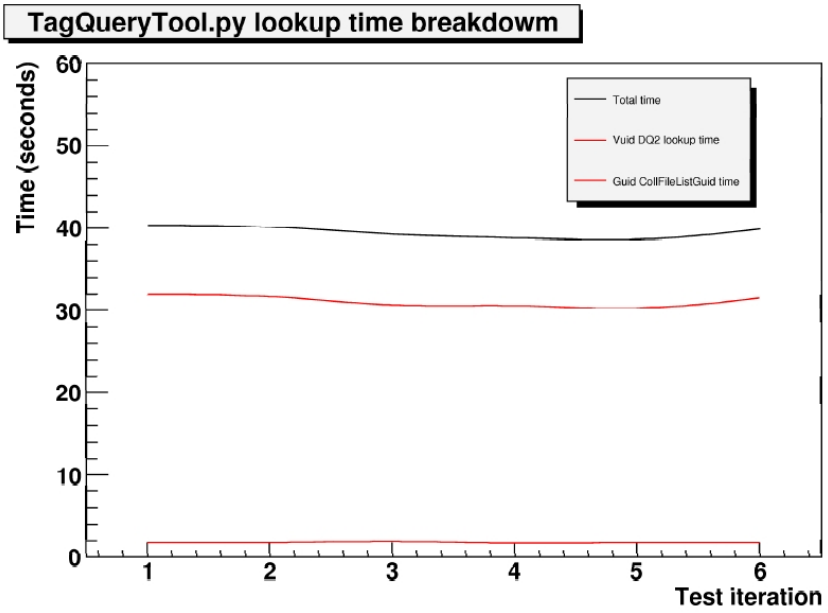



*Figure 7.3: A breakdown of the lookup steps performed when querying file information in the DQ2 catalogs, we see that the process of collecting the dataset information when this is not known in advance is a small percentage of the overall process time*

Figures 7.1, 7.2 and 7.3 show that as it is consistently the case that if VUIDs are provided in the TAG, negating GUID to VUID lookup in the Catalogs, the overall process

is faster, although we see that only a small fraction of the time for the complete process is saved by removing the VUID lookup. So if the Catalogs and API are to be used as they are, it may not be worth the performance implications of capturing a VUID at TAG writing time, as this saves only a relatively small overhead later.

It is however the case that the impact on the DQ2 central catalogs is less when fewer lookup steps are required, so we decided based on these studies to include a VUID metadata attribute in event TAGs, due to the potential advantages and as the impact in the overall TAG writing process is small, as the system must already write many attributes, including metadata at time of production, so capture of VUID is not a significant extention to the system.

## 7.7 Subscription methods

The TAG Database system demands that useful output in the shape of collections of Event TAGs and of AOD Event data can be returned to a user for analysis. In the ATLAS software environment, gathering of data in this way requires use of the DQ2 subscription facilities, in order to gather output data from a TAG Database either locally or at a selected ATLAS computing Grid site.

Subscription is a means of moving data between sites using DQ2 and we aim to develop an efficient subscription method that functions in the ATLAS dataset environment, when the output data is first given by references to ATLAS files. Two potential implementations of Dataset Subscription for the Event Level Metadata system are designed and studied.

Implementing use of datasets and subscription in the TAG Database system requires that steps are taken to create a new dataset containing the files returned by a query to the TAG database, to record the existence of this dataset in the DQ2 Catalogs and to then move the new dataset and TAG files to a chosen site.

To implement creation of a new dataset, a dataset must first be registered in the DQ2 Catalogs as existing within the system, and the files contained in the dataset specified by GUID and LFN. The files must be present in LFC ATLAS, an ATLAS wide Local File

Catalog, if DQ2 is to later collect these files together in new datasets. A dataset can be registered as existing at a location, but the files must be already present at the location. Registering a dataset at a location will not implement any movement of files or datasets, information will simply be placed in the relevant DQ2 Catalogs.

A dataset can then be moved between sites using a Subscription. Placing a dataset subscription at a site will prompt DQ2 to collect all the files in the dataset locally at the site, using information in the Content (which files are needed) and Location (where copies of the dataset are already located) Catalogs.

For a query on the TAG Database, it can be assumed that the AOD files identified by a query will be present at one or more DQ2 sites in datasets created at Production. The DQ2 Catalogs will therefore be aware of the existence of the files and datasets. Since the files will be registered in LFC ATLAS, it follows that the subscription method for creation and movement of new datasets can be adapted to suit the TAG model. Two methods are developed and considered, Complete and Incomplete Subscription.

In the tests that follow we assume that all the information needed to create and subscribe datasets, that is file GUIDs and LFNs and pre existing file locations, are already known, as we aim to compare subscription methods in an environment where other lookups cannot effect the results. In reality the Catalogs must be queried for all the information needed.

### 7.7.1 Complete Subscription

The first method developed is a Complete Subscription Method. A set of datasets, each containing a subset of the files identified by a TAG query, are registered as existing in the DQ2 Catalogs. Each subset of files are defined by a common location. The TAG datasets are registered as being present at the site where the particular file subset is present. All the new datasets are then subscribed to the local site, so that the new TAG datasets, and therefore TAG query result files, are collected locally.

In the TAG Database system, a user performs a query on a TAG database and identifies

a number of events of interest to an analysis, the TAG database returns pointers to the files containing the events, where files may be local or remote. In the subscription method development, the process is modelled where one file is local and three are remote to the user. DQ2 methods are then used to collect the files and therefore events locally for analysis. The Complete Subscription method is shown in figure 7.4

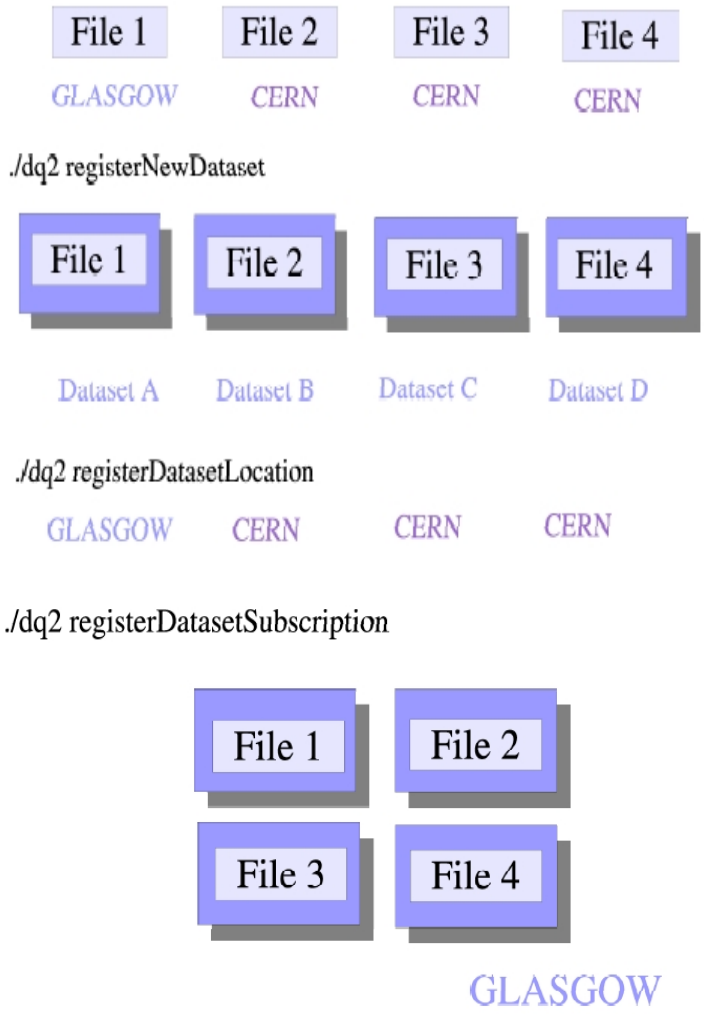



*Figure 7.4: The Complete Subscription Method, files are identified by a TAG query as containing events of interest, the files are registered in DQ2 as existing in a dataset at sites where the files are located, the datasets are then subscribed using DQ2 to the site where files are required for analysis, so collecting the files to the desired site for analysis*

### 7.7.2 Incomplete Subscription

The second method considered uses Incomplete Subscription. An Incomplete subscription method take advantages of a DQ2 method of registering a dataset at a site when the set of files contained in the dataset are only partially present at the site, so registering a dataset as incomplete at a site. A single new dataset is registered as existing, and registered as being present and incomplete at each site where a subset of TAG query result files are present. The new TAG dataset is then subscribed to the local site.

Registering one rather than many new datasets in this way is an optimal approach for the Global Catalogs, as it involves fewer new catalog entries. It is also thought that perhaps since it involves fewer calls to the Global Catalogs and less demand on local site services, the incomplete subscription method may prove optimal too in terms of TAG performance.

Added to the potential performance benefits that an Incomplete Subscription method brings to the TAG system, it may also be the case that creating one new dataset rather than many will better assist repeated use of a TAG dataset, should a user subsequently require the files corresponding to a specific query.

A user again performs a query on a TAG database, returning pointers to one local file and three remote files. The aim is again to collect all the files locally for analysis. The Incomplete Subscription method is shown in 7.5

## 7.8 Comparing performance of subscription methods

Early in the TAG database development effort, tests were developed to study and compare the Complete and Incomplete Subscription models, to study and compare performance and to select the optimal method to implement in the TAG Database system.

The tests were undertaken firstly using a set of fake AOD generated dummy data files, and repeated using AOD data produced for an ATLAS data workshop held in Rome, referred to as the Rome data, corresponding to a realistic query on the current TAG

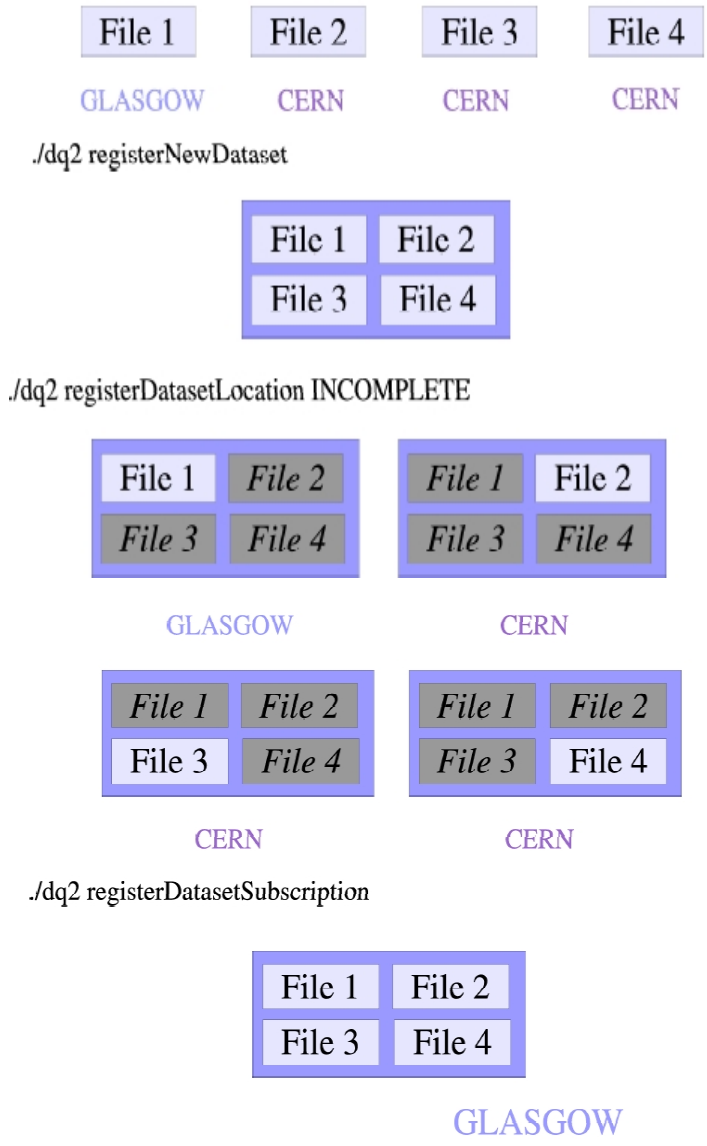



*Figure 7.5: An Incomplete Subscription Method, a single new dataset is defined, the dataset contains all the files identified as interesting by a TAG query. The dataset is registered as being present and incomplete at the sites where the files of interest are located and the new dataset is then subscribed using DQ2 to the site where files are required for analysis, so collecting the files to the desired site for analysis*

database at the time. Results that follow correspond to the Rome data. The tests took place using DQ2 version 0.2.11, the current and most recent version at time of testing, and the locations used to store, locate and transfer data were CERN and the Tier 2 at Glasgow. An FTS channel was established between these sites in preparation for testing, so that data tranfer was possible.

To prepare an environment, the following steps were taken

- The Rome AOD files were first registered in LFC ATLAS, the ATLAS wide file catalog which also acts as a LFC for the CERN site (CASTOR), where the AOD POOL files were stored, as DQ2 requires files to be registered in this catalog.
- The files were registered as being on CASTOR SRM, as this is the location of the Rome AOD following Rome Production.
- The LFNs and GUIDs assigned to the Rome AOD were read in from the output of the registration with LCG in LFC ATLAS and passed to DQ2, as these are needed as parameters in the new dataset registration stages.
- Care was taken to ensure that the LFNs are consistent in LFC ATLAS, CASTOR SRM and DQ2.
- A 25% subset of the test files were copied to and registered on the local site

Consequently both DQ2 and LFC ATLAS are aware that the AOD files in the study exist on grid storage and a GUID is assigned to the file by the registration process. Placing a subset of files on the local site creates a more realistic environment as in any real situation it is likely that some of the required files, if not all, are already present locally.

An instance of the DQ2 Global Catalogs is required in any subscription process. A development instance of the DQ2 Global Catalogs at CERN were used for the tests involving simulated AOD files. The development Catalogs were also used by other ATLAS groups for DQ2 related testing, so the subscription tests were repeated over several days

to allow for variations in the external load on the Catalog Server. In the tests involving the Rome AOD, the Tier 2 instance of the Global Catalogs were used. As these Catalogs were intended for TAG development studies alone, it was assured that there may be no other demands on the Catalog server to impact on the results.

The test process was as follows

- The DQ2 client and Catalogs were accessed through one of two python scripts designed to perform and time the Complete and Incomplete Subscription methods.
- The site services at the local site were monitored for activity related to moving the files to site and the subscription process, defined as the time from specifying which files were required locally to delivery of the files to site, was timed.
- After each complete test, the files were deleted from local storage, both physically and in the Local File Catalog. The new dataset and file entries in DQ2 were erased, as it was important to ensure a clean environment for repeated tests.
- Complete and Incomplete timing results were compared

### 7.8.1 Results

As hoped with the view of minimising new Catalog entries and simplifying subsequent access to the TAG query dataset, it was seen that the Incomplete Subscription method was consistently preferable in time to the Complete Subscription method. When the development DQ2 Global Catalogs were used it was possible to see the effects of the varying load on the Catalogs on the subscription process, but despite this, it was always the case that the Incomplete method is faster. The results of the Subscription Model tests are shown in Figure 7.6.

The Incomplete Subscription Method is therefore adopted by the Event Level Metadata system and implemented in GANGA-TNT, as it is seen to be a consistently optimal and preferable approach.

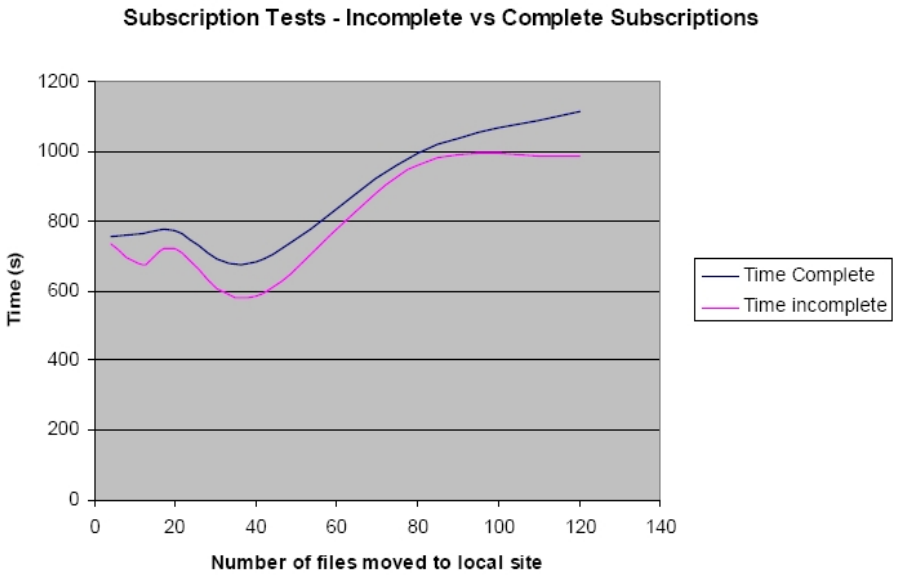



*Figure 7.6: Comparison of the Complete and Incomplete Subscription Methods, we see that the Incomplete Subscription Method performs consistently faster than the Complete Method, the rise and fall of response times can be seen as the outside load on the DQ2 Catalog varies throughout*

### 7.8.2 The hidden parameters

Concurrent users, server load, CPU, DQ2 structure are parameters which may effect the performance of a subscription. Variations in subscription response times are seen in the results of figure 7.6. The subscription tests compare performance time for an increasing number of files requested by a user. The hidden parameters may vary as the DDM system and ATLAS experiment progress but as the Incomplete method is shown to have consistently better performance.

## 7.9 Catalog Schema

The Distributed Data Management project catalog schema may change and in fact underwent schema change at the time of feasibility testing and early TAG developments, creating a dynamic testing environment. It is important that the TAG system can respond to changes in the catalog schema and will not be adversely effected by changes, to ensure an Event Level Metadata system that can perform and adapt to a dynamic catalog

system. A method of testing file lookup based on two catalog schema was developed and performed.

### 7.9.1 Schema tests

Using the Global Catalogs instance at Glasgow, a table t_pfn, representing an early schema, and t_files, reflecting a newer schema, were populated with a million entries. The tables mimic DQ2 tables representing earlier and newly developed DQ2 schema in the DQ2 system.

A series of identical queries were performed on the tables to compare the performance of each table in returning identical information. The queries were performed both using MYSQL queries directly on the catalogs, and using a query submitted through the client API. The entries in the table were scaled up to two, four and eight million and tests repeated. Eight million files is thought to be a reasonable estimate at an ATLAS scale for a table containing ATLAS file information.

### 7.9.2 t_pfn vs t_files for a single query

The performance of t_pfn and t_files are first compared in terms of the response time for a query structured so that one hit is performed on the table for each file GUID, this will be referred to as a SINGLE structured query. The tests are repeated for an increasing number of files in the full query, for an increasing number of rows in the table, and the response time noted.

The first set of graphs in figure 7.7 show the performance response time in terms of returning a result for all files, while the second set show an average time per file. Figure 7.7 shows that t_pfn and t_files perform comparably for a query structured as a single query per file. The response time per file is 7.5 ms per file consistently as both size of database and total number of files in the query are increased, for both tables, so the total query time increases directly with the number of files, independent of size of table or number of files.

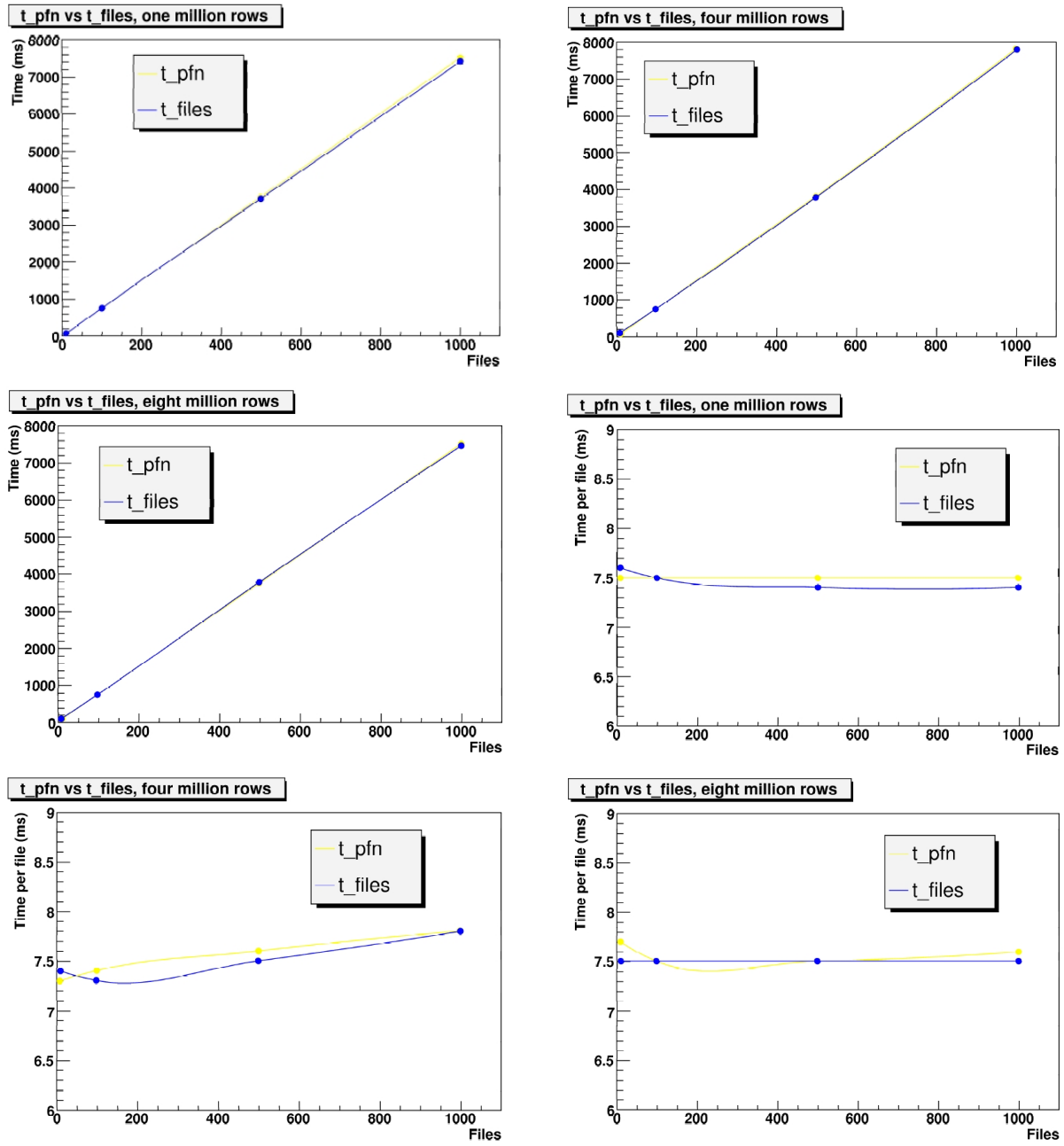



*Figure 7.7: Comparison of performance of queries performed on two alternative DQ2 schemas, t_pfn vs t_files, for queries where a single hit is performed on the database for each file involved, for an increasing number of rows in the test database. The first three graphs show the time for an increasing number of files in the query, the second three show the time per file.*

### 7.9.3 Single vs Bulk query

The t_files schema separates the attributes so that a wildcard search is not needed in a query by file GUID. This allows a query to be structured for t_files using a MySQL IN

clause, so that all files are queried in one hit on the table. We refer to this as a BULK query. A set of queries, firstly a SINGLE query, then a BULK query, are performed on t_files to assess whether the IN clause can be used to improve the response time and hence performance for a resulting query for an equivalent number of files. Tests are repeated on t_files for an increasing number of files in the query and an increasing number of rows in the table.

Figure 7.8 shows that while a SINGLE query becomes expensive for an increasing number of files, and increases directly with number of files, the BULK query is both much less expensive and does not increase rapidly with number of files. This is consistent as the number of rows in the table and the number of files in the query are increased.

As an IN clause is possible with the t_files schema but cannot be implemented with the wildcard element required by t_pfn, t_files schema is a favourable schema for improved response time performance. Figure 7.8 shows that 1000 files can be queried in a bulk query on t_files compared with the same response time for 10 files in a single query on t_pfn. At the time this study was performed, the Distributed Data Management system independently opted for a schema using the t_files table structure, so the BULK query method developed in this study can be and is adopted by the Event Level Metadata system when interacting with the Distributed Data Management system.

### 7.9.4 LFN vs LFN and VUID

Assessing the performance implications of varying the number of attributes returned by a query is significant, as a query on the TAG Database may require LFN, VUID or many other attributes to be returned in order for the output to be useful to a user. At the time of feasibility studies, it was not yet determined whether a TAG Database could realistically return a full event TAG. Later chapters in this thesis study development of the system where user output is well defined. Users can output a subset of event metadata, all event metadata, or the event data itself, using the pointers to the event stored in the event level metadata. At feasibility testing stages the potential for user output was still to be

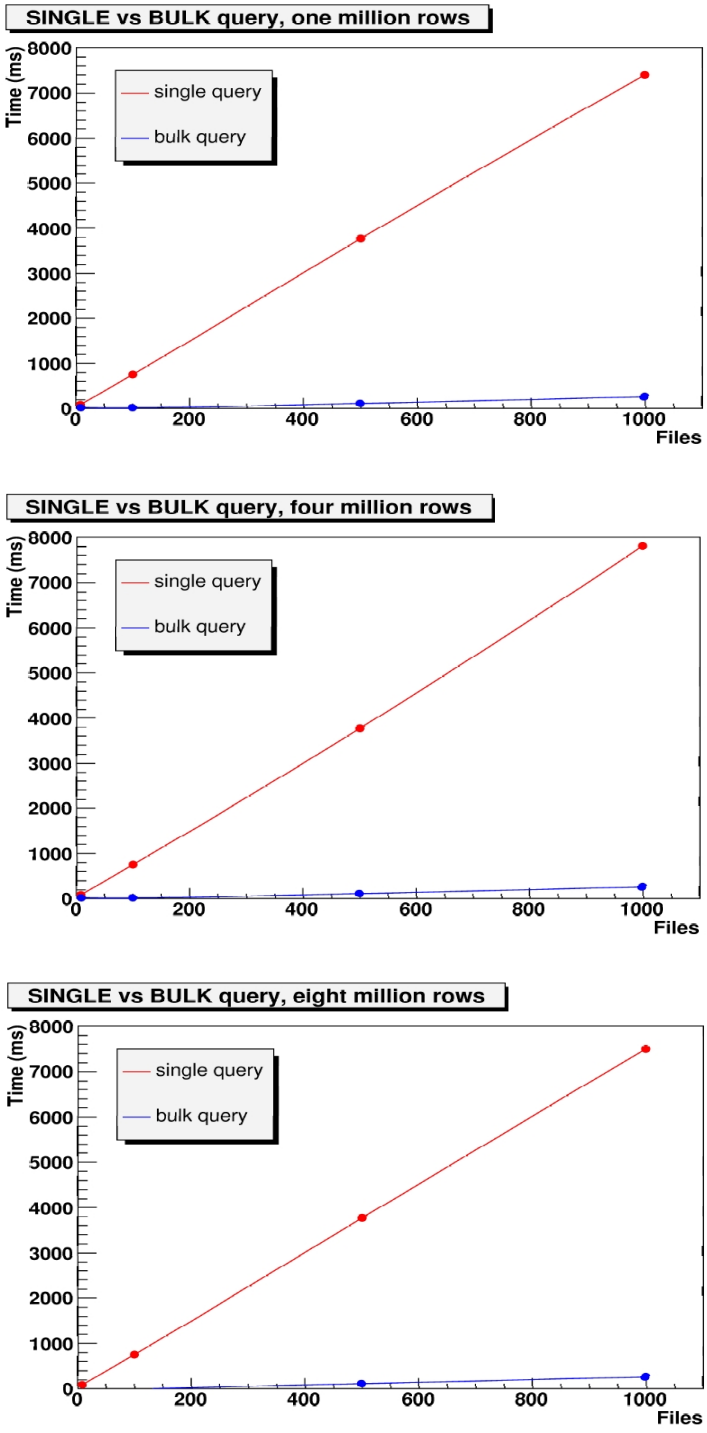



*Figure 7.8: A comparison of SINGLE and BULK query performance times, where a SINGLE query performs a hit on the database per file and a BULK query uses an IN clause, for an increasing number of rows in the database. We see that a SINGLE query is expensive as number of files increases, a BULK query is less expensive and does not increase with number of files*

determined, and is determined in early stages in this study.

To assess the impact of varying the attributes returned, queries are performed on t_files for eight million rows and for an increasing number of files in the query. The implications of varying the number of attributes returned by the query for each file are assessed by comparing queries asking to return VUID with equivalent queries that return LFN and VUID.

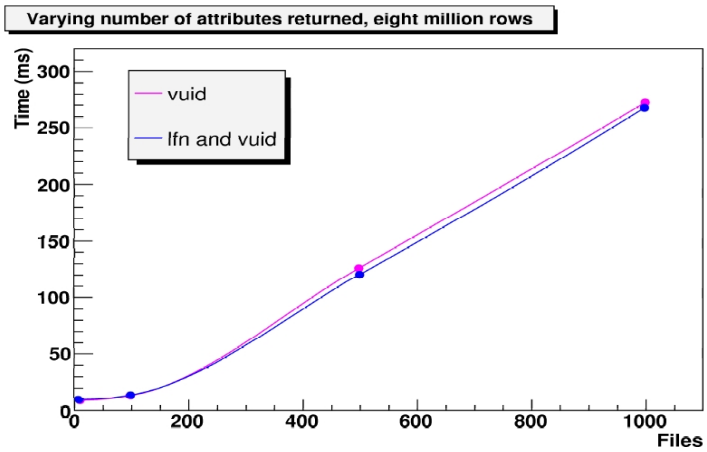



*Figure 7.9: The effect of varying the attributes returned by a query on the DQ2 Catalogs, looking up LFN and looking up LFN and VUID, for an increasing number of files in the query. We see the effect of changing and increasing the number of attributes returned is negligible and performance times are comparable*

The results in Figure 7.9 show that the effects of varying the number of attributes asked for in a query are negligible therefore it safe to assume that the performance results apply to a situation where one or more attributes are returned.

### 7.9.5 Increasing number of rows in the DQ2 Catalogs

As the ATLAS experiment progresses, the content catalog tables will be populated with more and more ATLAS files. The effects of increasing the size of the content catalog table t_files in terms of a TAG query are assessed by performing a series of equivalent queries on t_files for an increasing number of files in the query for an increasingly populated t_files.

Figure 7.10 shows that the effect of increasing the number of rows or equivalently ATLAS files in t_files from one to two, four then eight million rows is a relatively small

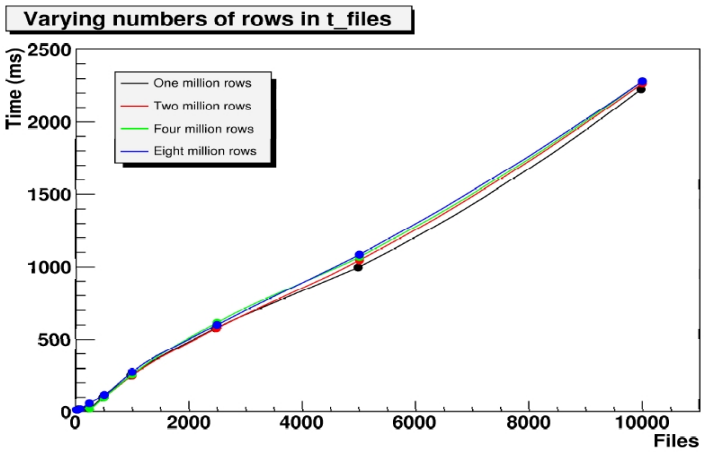



*Figure 7.10: Performance of file look ups in the DQ2 catalogs when we increase the number of rows in the catalog tables, then perform a series of queries for an increasing number of files. We see only a very small increase in performance times of queries identifying the same number of files in tables with increasing number of rows in the Catalogs, two, four and eight million.*

increase in response time with a two fold increase of number of files in the table.

### 7.9.6 IN clause performance

As an IN clause BULK query on t_files is identified as the optimal method versus an equivalent single query structure on t_files or t_pfn, it is possible to assess the performance of an IN clause as the number of files in the query is increased further. A t_files table with eight million rows is queried for a series of steps up to 20000 file GUIDs in the query.

Figure 7.11 shows that the performance of an IN clause responds steadily to an increasing number of file GUIDs in the query up to 20000 file GUIDs. 20000 file GUIDs are thought to be a realistic high end limit to the number of files a user would identify in a query to the TAG database.

### 7.9.7 Production background query rates

As the described tests are performed on an instance of DQ2 Global Catalogs installed at Glasgow dedicated to TAG testing, there are no concurrent non TAG queries as there are on the production catalogs and as there will be in a realistic case. To assess the

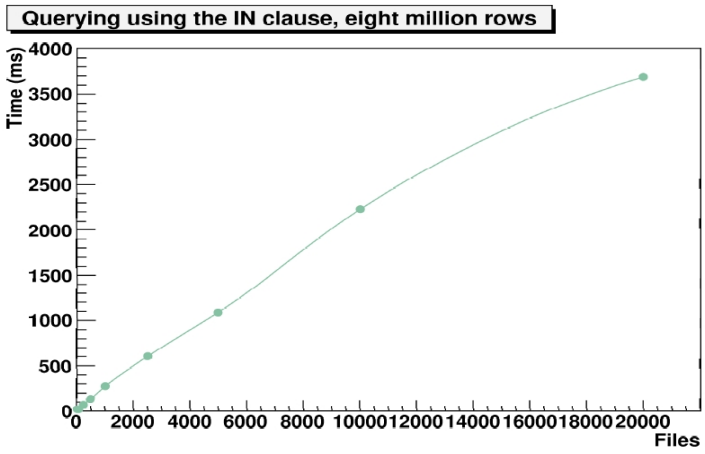



*Figure 7.11: Performance of an IN clause query look up method for a DQ2 Catalog table with eight million rows, we increase the number of files returned by the query to 20000. We see a steady and predictable response time increasing with the number of files in the query, demonstrating that file look ups of the scale of up to 20000 files in a catalog of eight million rows are feasible*

performance implications of having concurrent queries taking place on the table, the current ATLAS DQ2 Production Catalogs, are studied through the Apache log showing activity in the DQ2 catalogs and MySQL statistics to identify a realistic background query rate. This is assessed as being 60 hits per second on average and 100 hits per second peak query rate.

Scripts were written to perform both average and peak background query rates on the Catalogs used for TAG tests, and a series of queries were performed on t_files populated with eight million rows for an increasing number of files in the query for a case both with and without background queries. The plots in Figures 7.12 and 7.13 show the effect of simulating average and peak concurrent non-TAG queries on the catalogs. An increase in response test of approximately 10% is seen consistently across an increasing number of files in the query for both an average and peak background query rate.

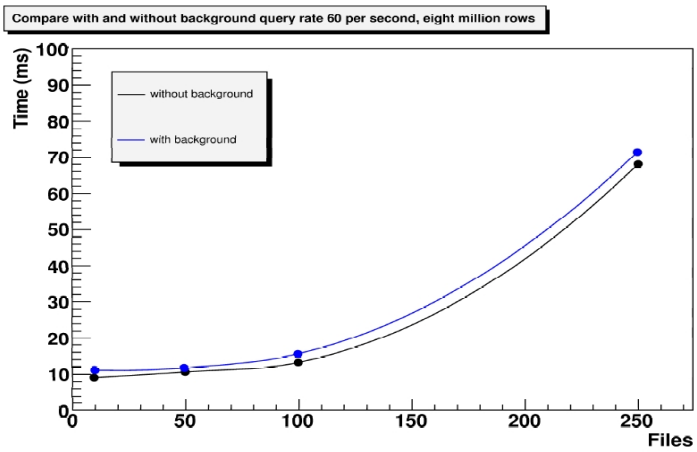



*Figure 7.12: Response times when querying a catalog of eight million rows for an increasing number of files with and without an average background rate of 60 outside hits per second, an average background query rate. We see that performance and response times are not notably effected by an average background query rate*

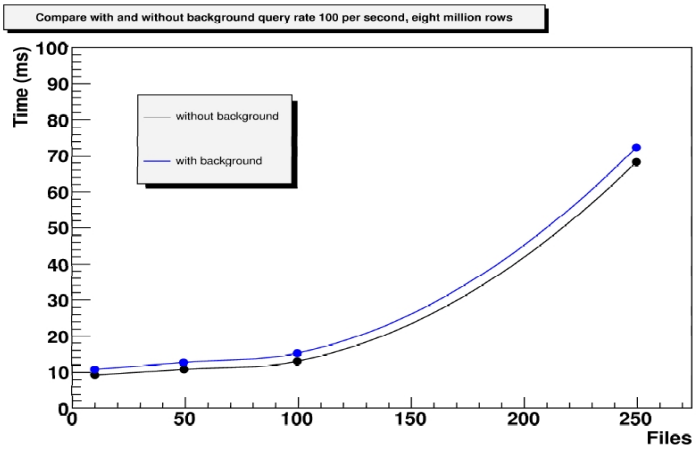



*Figure 7.13: Response times when querying a catalog of eight million rows for an increasing number of files with and without an average background rate of 90 outside hits per second, a peak backround query rate. We see that performance and response times are not notably effected by a peak background query rate*

## 7.10 Distributed Data Management and Event Level Metadata Studies Outcomes

In the studies of merging the Distributed Data Management and Event Level Metadata systems, we have studied the challenges of merging the data management dataset concept with the event metadata file units. Through the studies, we have achieved the following

- identified an optimal subscription method
- shown that we can return full event TAG content, or a subset of attributes
- a BULK query lookup using an IN clause is favourable to a SINGLE lookup
- the Event Level Metadata system can respond to dynamic changes in the DQ2 catalogs

We now move on to merging the Event Level Metadata system with the ATLAS Trigger system.

## 7.11 Trigger and TAGs

The ATLAS Trigger is a fundamental and central part of the ATLAS software and analysis system. The software system and analysts alike rely on the Trigger decisions to guide data selection, for storage and for analysis. For a TAG Database event selection system to be useful to users, it is imperative that Trigger decisions are available to users within each event TAG, presenting a significant design and implementation challenge for the development of the TAG database.

### 7.11.1 Event and Run Level Metadata

- Trigger Decision - Event metadata
- Trigger Configuration - Run metadata

Two pieces of information are needed for Trigger information to be meaningful, the Trigger Decision, whether or not the individual trigger passed, a yes or no, 1 or 0 decision, an event by event piece of information, and the Trigger Configuration, run by run information describing the meaning of each signature. The Trigger Configuration is needed to interpret the Trigger Decision.

### 7.11.2 Time Varying Trigger Menus

The Trigger menus used for ATLAS data are time varying in size and content. The Signatures which when combined constitute a Trigger Menu will vary on a run level. The space and format allocated in a relational TAG database may not be time varying on a run level as this is not practical in a relational context. A solution for Trigger implemented on event TAGs must convert the time varying Trigger Menus to an implementation in which a time steady interpretation of Trigger Decisions can be made. To achieve this, a fixed size of Trigger data to be held in an event TAG is established.

### 7.11.3 Size of Trigger Data

The Level One Trigger has a limit of 256 Trigger decisions per event, so the bit pattern needed in a TAG has a well defined size. The Level Two and High Level Trigger has an order of magnitude higher Trigger Decisions, of which a subset will make up an active Trigger Menu for run ranges. Should the full range of potential higher level Trigger Decisions be contained as bits in an event TAG, the Trigger Decision part of the TAG could be bigger in size than the full intended TAG size. A limit is therefore placed on the number of active higher Level Trigger Decisions at 1024, so the full bit pattern needed in the TAG for Trigger Decisions is well defined.

An event TAG has 1 kB size assigned by the Computing Model, and it is important to stay within this size range to ensure the system is managable and performant. For 256 Level One Trigger Decisions and 1024 High Level Trigger Decisions, storing each decision as a boolean would use more space than is necessary. Instead the information can be

compressed into bitmasks to save space and make the Trigger information a managable size.

### 7.11.4 Trigger Decoding in TAGs

In order to interpret Trigger Decisions in TAGs, where many Trigger Decisions are compressed into a Trigger word to usefully utilize available space without using more TAG space than is practically available in a TAG, a decoding strategy is needed. The decoding process takes the user selected Trigger Signature, decodes the Trigger words to see which part of the Trigger mask corresponds to the signature of interest, then reads the Trigger decision to see whether the Trigger was satisfied for a given event. The data required to translate Trigger bits to active signatures is stable on a run to run basis and is stored in the TAG database alongside the TAG event data. The size of the configuration data is small compared to overall TAG data size, so there is no notable impact in storing the Trigger translation data in this way.

The Trigger decoding software is written in PL/SQL. In order to ensure the Trigger lookups are performed in a reasonable query time, it is assumed that the name of the Trigger signature and the bit position, or chain counter, which determines the position of the corresponding Trigger decision bit in a Trigger word, is unique across a queried data range. In order to determine the bit corresponding to a Trigger decision, the system maps run to Trigger Configuration, to Trigger Decision and finally to the chain counter identifying the bit of interest. This mapping can change across run boundaries and in reality will not change often and is likely to be consistent over queried data. In assuming the uniqueness of the mapping, performance impacts in terms of query time are significant [54]. So it is possible to translate time varying Trigger Menus and incorporate both event and run level metadata into a data format that can be practically and usefully queried through the TAG Database in an efficient and performant way.

## 7.12 Conclusions and Future Directions

In the studies in this chapter, we have developed a TAG model where file based query results can be incorporated with the Distributed Database Management dataset environment. We explore the possibility of adding a dataset attribute to a TAG, to merge the two models, we develop a subscription model that uses the datasets of Distributed Data Management with the files of TAG queries and we assess potential impact of schema changes in the catalogs, plus query strategies that perform optimally when using the Distributed Data Management Catalogs. We can also develop a TAG Trigger model where Trigger decisions are incorporated into Event TAGs in a way efficient to both the TAG Database system and to ATLAS analysts. We conclude that file based lookup and operations needed by the TAG Database system are feasible and can be performant and optimal in the Distributed Data Management and Trigger environments, and suggest an Event Level Metadata system architecture that adopts the concepts and methods found to be feasible and optimal in these studies.

The next stages in ATLAS TAG Database development is to establish that the system can realistically scale to ATLAS needs. Having established the feasibility of merging the TAG Database with the ATLAS Distributed Data Management and Trigger software systems, we now attempt to study and develop an ATLAS TAG Database at realistic scale and in doing so attempt to assess the query times that a user may expect when using a relational Event Level Metadata database.

# Chapter 8

# Scalability and Performance of a Terabyte TAG database

## 8.1 Introduction

The ATLAS TAG Database is a multi terabyte Event Level Metadata selection system. An Oracle hosted global TAG relational database, containing all ATLAS events, implemented in Oracle, will exist at Tier 0, as defined in the Computing Model. Implementing a system that is both performant and manageable at this scale is a challenge and is the focus of this chapter. We present studies on implementation and performance of a realistic terabyte scale relational TAG Database [55]. The aims of the studies are to create a useful terabyte TAG database, assess the analysis environment in which a realistic relational global TAG collection must perform, investigate strategies for organisation of data within a relational stucture so that data can be both written and read in a useful and realistic way, and give a performance assessement of a realistic terabyte scale relational TAG Database.

The studies begin by creating a terabyte scale TAG Database using simulated TAG data, the terabyte database, its structure and creation will be described. We then assess the challenges of the environment in which the relational database will operate. We investigate many partitioning and indexing strategies and select a strategy optimal to the TAG database environment. We create a set of realistic analysis queries and perform these

on the database to assess performance. The queries are described in this chapter. We then present the performance results for the optimally performing relational database. The performance results are important in providing information to analysts about the optimal way to interact with the database and the performance that can be expected and in leading system analysts in management of the database system, so that optimal performance can be provided. Tests are then extended to Tier One sites, to assess and compare performance at this level and the results of the tests presented.

This chapter deals with the scalability challenges of the relational TAG Database, a unique and demanding challenge within ATLAS due to ATLAS' unprecedented data rate and volume and the high performance query demands for ATLAS users. The ATLAS Database and the challenging environment in which it must operate are introduced, terabyte scale relational database scalability tests performed in early 2007 are described, the experience and learning from the scalability tests are shared and performance results are presented.

## 8.2 Terabyte TAG Database Performance and Scalability

In early 2007, it was decided that a large scale realistic test of a terabyte scale TAG Database was needed, to demonstrate a capability to manage a realistic terabyte scale TAG database and to uncover challenges brought with scale. The scalability and performance tests are also an opportunity to optimise and measure performance. The tests began with the creation of a 1TB TAG Database, hosted on a development Oracle server at CERN. A set of realistic and useful test queries were developed. Indexing, partitioning strategies, Oracle Optimiser behavior, query processing strategies, Oracle Hints, parallel processing and multi client environments were explored. The data was queried and performance assessed for a series of schema iterations, each development in schema influenced by the learning and knowledge gained from previous iterations.

A 1 TB relational TAG Database is deployed at Tier 0 using simulated TAG data for terabyte scalability testing. The database contains one billion events, each described by two hundred event metadata attributes and is intended for extensive testing in terms of queries, population and manageability. The 1 TB tests aim to demonstrate and optimise the performance and scalability of an Oracle TAG database on a global scale.

Partitioning and indexing strategies are crucial to well performing queries and manageability of the database and have implications for database population and distribution, so these are investigated. Physics query patterns are anticipated, but a crucial feature of the system must be to support a broad range of queries across all attributes.

At the time of scalability studies and developments, event TAGs from ATLAS Computing System Commissioning, CSC, distributed simulations were available, and so were accumulated in an Oracle hosted database at CERN, to provide an event level selection service valuable for user experience and gathering information about physics query patterns. The outcomes of the terabyte scale studies were implemented in the structure and presentation of the CSC TAG data.

## 8.3 Demands on a TAG Database

We assess the input to a TAG database, the environment in which the database needs to perform and the performance requirements a terabyte scale database for ATLAS needs to meet. The database we create in the scalability studies must perform to these demands in order to demonstrate that ATLAS scale TAG database is a realistic project and to provide meaningful performance results.

### 8.3.1 A Challenging Environment

The TAG database will realistically support high data volume and an incoming data rate of 200 Hz. Volume is terabyte scale, increasing as ATLAS continues to produce data and there will be 200 new event TAGs per second for data taking. Assume 50 K active seconds

per day and 58% efficiency for each active day, there will be $10^7$ new event TAGs every day.

### 8.3.2 A Challenging User

An ATLAS analyst expected to use the TAG database, by the nature of physics studies an analyst typically performs, is anticipated to be a challenging user. An analyst will demand

- Fast, efficient, accurate queries
- Reliable navigation to event data
- Seemless integration with analysis

The terabyte scale database must perform in these terms if it is to be a realistic and useful feature of the ATLAS analysis system.

### 8.3.3 Challenging query patterns

Typical queries expected to be submitted to the relational TAG database by analysts create a challenging query environment, as queries are likely to vary, notably in terms of the attributes on which event selection is made, the attributes in the query predicate, as analysts and physics analyses at ATLAS perform a wide range of studies.

## 8.4 A Terabyte TAG Database

The distributions of values in the attributes created for the terabyte scale database mimic the types of columns and the distributions expected for event level metadata. Each row is approximately 1 kB, the expected TAG size. The test table is shown in figure 8.1

To create a billion test rows, one million rows were first created, then replicated to create a test table of one billion rows. A full set of TAG attributes in the initial million rows is repeated throughout the billion rows, were attribute sets are distinguishable by

numbers. The attributes in set 01 are indexed and the remainder in set 02 to 08 are not, so that indexed versus non indexed attributes can be compared. Both Btree and Bitmap indexes are assigned throughout, depending on the distribution and cardinality of values in the attribute, so a variety of attribute distributions, including flat, random, boolean, exponential and gaussian distributions are available to be queried. A variety of Oracle datatypes were used for the TAG data, NUMBER, CHAR, VARCHAR2 and BINARYFLOATs, to mimic realistic event level metadata TAG values and to provide realistic test conditions where queries on and administration of varying datatypes can be compared.

A number of globally identifiable variables are created throughout the billion rows, so each row is unique. An attribute with 10 distinct values is included to represent ten potential ATLAS physics streams by which AOD data are grouped.

### 8.4.1 Test Architecture

The architecture used for the tests, figure 8.2 is an Oracle development server, INT8R, at CERN, with two Oracle instances, each with 2 CPUs and 2 GB memory, and 2 TB shared storage.

## 8.5 Challenges of 1TB data

A terabyte scale was selected for development and testing as it is a realistic order of data for a TAG Database and as we expect that important phase transitions in performance, behaviour and management demands are crossed as we scale the number of events from millions to billions. We anticipate that query processing on data at this scale has four possible processing patterns,

- A complete set of the data fits into memory/cache

  In this case a query is limited by CPU and memory

- A complete set of indices fits into memory/cache

| Name | Datatype | Size | Scale | Nulls? | Index ? |
|---|---|---|---|---|---|
| ID | NUMBER | 36 | 0 | Yes | B*Tree |
| RUNNR | NUMBER | 12 | 0 | Yes | Bitmap |
| EVENTNR | NUMBER | 12 | 0 | Yes | |
| GOLDEN1 | NUMBER | 12 | 0 | Yes | Bitmap |
| GOLDEN2 | NUMBER | 12 | 0 | Yes | |
| AODFILEFK | NUMBER | 12 | 0 | Yes | Bitmap |
| ESDFILEFK | NUMBER | 12 | 0 | Yes | |
| BOOL500CHAR01 | CHAR | 1 | | Yes | Bitmap |
| BOOL100CHAR01 | CHAR | 1 | | Yes | Bitmap |
| BOOL10CHAR01 | CHAR | 1 | | Yes | Bitmap |
| BOOL1CHAR01 | CHAR | 1 | | Yes | Bitmap |
| BOOL500NUM01 | NUMBER | 1 | 0 | Yes | Bitmap |
| BOOL100NUM01 | NUMBER | 1 | 0 | Yes | Bitmap |
| BOOL10NUM01 | NUMBER | 1 | 0 | Yes | Bitmap |
| BOOL1NUM01 | NUMBER | 1 | 0 | Yes | Bitmap |
| ENUMUNI100VC01 | VARCHAR2 | 10 | | Yes | Bitmap |
| ENUMUNI10VC01 | VARCHAR2 | 10 | | Yes | Bitmap |
| ENUMEXP1000NUM01 | NUMBER | 5 | 0 | Yes | Bitmap |
| ENUMEXP100NUM01 | NUMBER | 5 | 0 | Yes | Bitmap |
| ENUMEXP10NUM01 | NUMBER | 5 | 0 | Yes | Bitmap |
| ENUMUNI1000NUM01 | NUMBER | 5 | 0 | Yes | Bitmap |
| ENUMUNI100NUM01 | NUMBER | 5 | 0 | Yes | Bitmap |
| ENUMUNI10NUM01 | NUMBER | 5 | 0 | Yes | Bitmap |
| ENUMUNI5NUM01 | NUMBER | 5 | 0 | Yes | Bitmap |
| NOR100BF01 | BINARY_FLOAT | 4 | | Yes | B*Tree |
| NOR10BF01 | BINARY_FLOAT | 4 | | Yes | B*Tree |
| NOR1BF01 | BINARY_FLOAT | 4 | | Yes | B*Tree |
| NOR100NUM01 | NUMBER | 12 | 5 | Yes | B*Tree |
| NOR10NUM01 | NUMBER | 12 | 5 | Yes | B*Tree |
| NOR1NUM01 | NUMBER | 12 | 5 | Yes | B*Tree |
| UNINUM01 | NUMBER | 12 | 0 | Yes | B*Tree |
| UNI10KNUM01 | NUMBER | 12 | 0 | Yes | Bitmap |
| UNI1KNUM01 | NUMBER | 12 | 0 | Yes | Bitmap |
| UNI100NUM01 | NUMBER | 12 | 0 | Yes | Bitmap |
| UNI10NUM01 | NUMBER | 12 | 0 | Yes | Bitmap |

*Figure 8.1: The Scalability and Performance studies test table, showing the attributes, datatypes, size and index types used as a basis to create a billion row table, each row of estimated TAG size 1 kB*

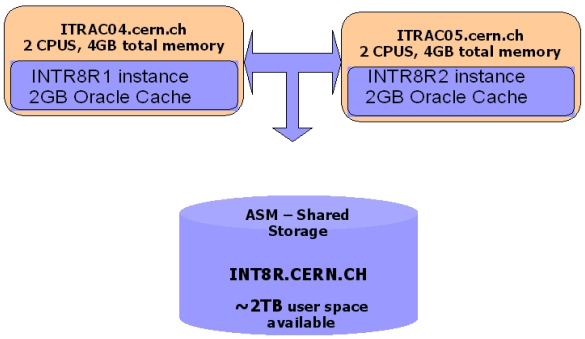



*Figure 8.2: The Test Architecture for Scalability and Performance tests, an Oracle development server at CERN with two Oracle instances, each with 2 CPUs and 2 GB memory, and 2 TB of shared storage*

In this case identification of data is fast with slower retrieval from disk, we estimate that for contiguous parts an order of 100 MB/s and random parts an order of 1000 IO/s, 1 MB/s where contiguous parts degrade with parallelism

- Indices and data do not fit in memory

  Indices must be read from disk, usually contiguously

- Intermediate results and final results do not fit in memory

  In this case disk must be used for sorts, intersections and joins

The terabyte data we create will not fit into memory. As each index is of the order of 10 GB per GB of data, indexes will not all easily fit in memory either. We anticipate and aim to support queries that are open ended and will select on a variable set of attributes, so index cache turnover will be high, limiting any caching advantage. Queries are potentially unselective, returning a large percentage of available data, up to an order of 10%. This means there may not be enough memory to store intermediate results and processing multiple queries in parallel could be difficult.

The terabyte data scale therefore presents challenges beyond those of a smaller relational database. Strategies must be adopted to optimise and facilitate administration and query performance in this challenging environment.

## 8.6 Partitioning

Partitioning is a strategy used in relational databases to divide data from larger whole units to smaller ones. Partitioning is often referred to as a Divide and Conquer strategy and involves splitting data into smaller composite parts to improve the way it can be managed and queried. For the terabyte scalability and performance studies, both Horizontal and Vertical partitioning are considered. The motivation for partitioning the TAG Database are described in terms of two mutually important dimensions - query performance and database manageability. We aim to improve both in a realistic ATLAS TAG analysis environment, without any performance improvement on one at the expense of the other.

### 8.6.1 Partition keys

A partition key is an attribute in a relational table by which data is divided. A partition key needs to appear in a query for partitioning to be a direct positive benefit to a query. TAG user queries are expected to vary in content and we aim to support all various query possibilities and attributes, so we want to add benefit to as many queries as possible, ideally all, without cost to any.

We choose RUNNR, STREAM and GOLDEN attributes as potential partition keys, were RUNNR is detector run number, STREAM is physics stream and GOLDEN attributes represent attributes expected to be extensively used in a query. RUNNR is interesting because it is the unit of Tier 0 production, and STREAM is interesting because it is likely most analysis can identify a preselection based on events written to a predefined physics stream. GOLDEN attributes are placeholders for other attributes that may potentially identify themselves are useful for partition keys as information about

query patterns is gathered through experience of analyst query patterns on Event Level Metadata databases.

### 8.6.2 Horizontal Partitioning

Horizontal partitioning involves subdividing data by rows, into a set of smaller tables with a subset of the event TAGs in each. Any query that uses the attribute by which data is partitioned, the partition key, in its predicate will benefit in performance, as only partitions of interest to the query are then considered. This is known as partition elimination or partition pruning. There is no performance overhead meanwhile for queries that do not specify partition key, or that require data from multiple horizontal partitions. Oracle allows partitioning by Range, List and Hash, as well as Composite Range-List and Range-Hash - all were considered at small scale to understand the functionality and the potential benefits, tests and results described in Appendix B. Performance was seen to improve directly with the amount of data removed from consideration, and Range-List and Range were identified as the optimal schema choices.

TAG attributes suitable for partitioning by within these partitioning strategies were considered. It is important to select a partition key which users can often and easily define in their queries, or we can reasonably expect them to be required to do so, so that partition elimination will improve performance. If the partition key does not appear in the query, no partition elimination will take place and no performance benefit is made.

### 8.6.3 Vertical Partitioning

Vertical partitioning involves dividing data along vertical lines, so each event TAG would be split across vertical partitions. Such partitioning of data can improve query performance by removing data irrelevant to the query from consideration. There is however a management overhead for this schema, as POOL collection tools used to input data into relational tables would have to be adapted, and potential performance disadvantage, as should a query require data from many partitions, joins, expensive database operations

especially is the data does not all fit into memory, become necessary.

### 8.6.4 Horizontal Partitioning Solution for 1TB

A useful candidate for TAG partition key is Run number, as it is the unit of Tier 0 data production. Data can be written to the database in units of runs, grouped to create reasonably sized partitions. Once a run is complete we can be certain that no more write operations will be needed, the partition can then be declared complete and read only. Read and write operations can be separated. Equally, it is thought reasonable to ask a user to define some temporal quality in their query. Physics Stream is also a candidate, as an event attribute that physicists are anticipated to define queries by.

Using Run and Stream, Range and Range-List partitioning schemas were tested. Performance benefits were seen with both, but Range-List was found at this scale to increase the management overhead as the schema becomes more complex. As a result, an alternative means of composite partitioning is developed. Stream is used to create ten separate tables, each stream table is partitioned by run. Query performance is enhanced when one or both partition keys are included in the query, ten independent stream tables allow significant improvement in query performance with or without run number specified in the query. When a query involves more than one stream, queries can be easily divided in a preprocessing step and performed in parallel.

The schema does not add any significantly increased management overhead, in fact administration tasks are simplified as the ten Stream tables can be managed independently. Each partition is 1GB, we have 100 partitions per run in ten Stream tables. The load method for tables is to put data into a WRITER table, once the run is complete we copy the partition into memory, indexes are rebuilt and the data copied into the READER table. The schema has considerable benefits for indexing strategies, as discussed later in this chapter.

### 8.6.5 Vertical Partitioning of 1TB database

At the scale of smaller tests, the performance impact of joins across vertical partitions was low and only a disadvantage when querying from all partitions. The tests and results are described in Appendix B. At terabyte scale using larger realistic TAG queries, join operations are costly and are likely to require use of disk. The POOL Collection Infrastructure does not currently support disjoint import of data in its current state, so developments in the import system would be needed. As it is not clear how attributes should be grouped in vertical partitions for effective data elimination, as vertical partitions increase management and as joins across partitions are potentially costly, vertical partitioning was not used in the terabyte scale database. Vertical partitioning of a TAG Database at scale will be restudied in future, once a better understanding of query patterns has been gathered from deployment of TAG Databases for use by ATLAS physicists. It may be that attributes can be classified as Hot or Cold, depending whether they are often or seldom queried. Study of query patterns is therefore an ongoing project.

### 8.6.6 Complete Partitioning Solution for 1TB

The partitioning strategy adopted and found to be optimal in terms of performance and manageability is a separation of data into ten Physics Stream tables using the Stream attribute, each table is horizontally range partitioned by Run number.

The terabyte data partitioning strategy developed in this study and used in the relational TAG database is shown in Figure 8.3.

## 8.7 Indexing

Indexing attributes can potentially improve query performance by avoiding the need for table reads and therefore speeding up queries. Indexes allow table lookups by rowid, a fast operation where the index is used to identify the row satisfying the query, then the data is taken directly from the rows, without needing to scan the table. Indexes however

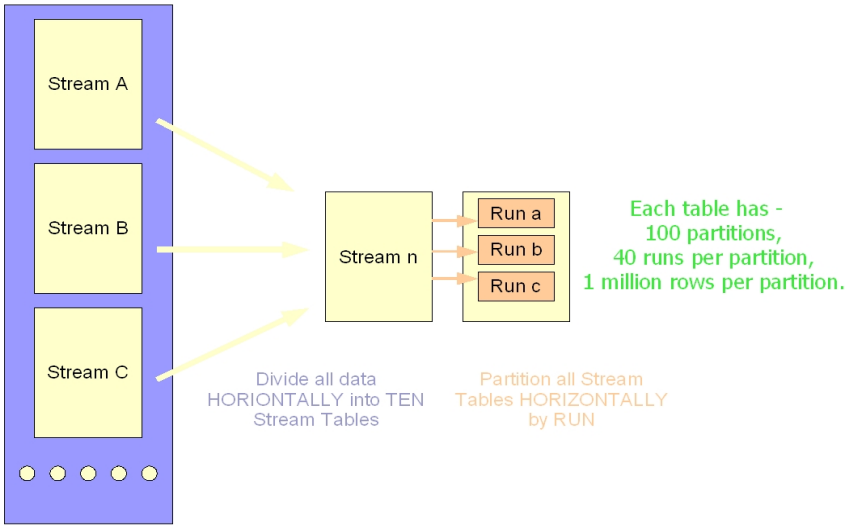



*Figure 8.3: 1TB data Partitioning Strategy, data is divided into ten Stream tables, each Stream table is then partitioned horizontally by run*

require storage space and involve an overhead of creation. We use Btree indexes, Bitmap indexes and non indexed attributes to assess query performance and optimal query paths. Btree indexes are suited to attributes with many distinct values, bitmap to those with fewer. Btree indexes are more costly in maintenance and storage than bitmap indexes, due to their larger size.

### 8.7.1 Indexing solutions and experience for 1TB

Initially some attributes were not indexed, to study the usefulness of unindexed attributes in a table of this scale. It was seen that without indexing on attributes in the query predicate, a query was forced into a full table scan and this is much more expensive in terms of performance than a query in which all attributes in the WHERE clause are indexed. Indexing an attribute which appears only in a SELECT clause does not impact performance, as the table lookup mechanism is performed on the attributes in the WHERE and not SELECT.

As indexing has such a drastic performance effect and as it is difficult to say which attributes are more likely to appear in a predicate, it was decided that indexing all at-

tributes should be attempted. This is feasible when considered in combination with the partitioning strategy adopted for the table, in which events are partitioned horizontally by Run. Without this, indexing all attributes would be impossible. Indexes are partitioned with the table, so for building indexes, we can force Oracle to hold the index in memory, meaning the time to build indexes is seconds, rather than hours. Assuming partitioning of the table is done by Runs, indexes will be rebuilt only when we finish loading the data of the run in the WRITER table, will we load the partition into memory, rebuild all indexes, then put this in the READER table.

Btree and bitmap indexes were tested, to understand the optimal query plans and management overhead of each index type at this scale. Bitmap indexes have an average size of 2 MB per partition, and for Btree 20 MB is the average, where each partition is 1GB, we have 100 partitions per Stream table. After extensive testing we see that Btree and Bitmap indexes perform optimally under distinct operations, and a strategy for addressing query processing in terms of these index types must be developed, this is achieved by studying the behaviour of the Oracle Optimiser.

## 8.8 The Oracle Optimiser

Oracle has an Optimiser which evaluates each SQL query, assesses the possible execution plans and selects the most efficient based on a number of criteria. We use the Cost Based Optimiser, which selects an execution plan based on estimated lowest Cost.

### 8.8.1 Optimising the Oracle Optimiser

In query testing and execution plan comparison of queries on 1TB scale data, it was seen that often the optimiser would select a non optimal query plan. Often an index would not be used, a full table scan would be selected when a better choice existed, partitions would not be used to the fullest or parallel processing would not be used with indexes. A method was developed over the course of thorough query testing to implement SQL queries using Oracle Hints, so that the Optimiser is guided into adopting an optimal query plan.

### 8.8.2 Optimiser Hints

An Oracle Optimiser Hint is a suggested given to the Oracle Optimiser at query creation time, recommending the optimal way for Oracle to perform the query. Queries were divided into sets based on their features and optimal query plans, and Oracle Hints applied accordingly. The optimising preprocess is a necessary feature of the TAG Database at scale, also demanding monitoring, as the system extends and more is understood about usage and query patterns, it may be necessary to adapt the hint strategy in response.

The Oracle hints found to potentially improve performance for the 1TB scale data are

- PARALLEL - needed for parallelization of full table scan or partition range scan
- PARALLEL_INDEX - needed for parallelization of index access
- INDEX_JOIN - for hash joins with b-tree indexes
- INDEX_COMBINE - for bitmap indexes
- opt_param(_INDEX_JOIN_ENABLED , false) - Can enable and disable session parameters for a single SQL

### 8.8.3 Query Hints Solution for 1TB

All the hints described above were seen to improve query performance by influencing the query processing plan across various query types. After extensive testing we saw that parallel processing is desirable as processing time is reduced when a query can be processed in parallel. Btree indexes perform optimally when an INDEX_JOIN operation is performed and bitmaps when INDEX_COMBINE influences processing. As user queries may filter on both types of indexes, we develop a query processing hint strategy where the SQL query is reduced into two separate SQLs for btree and bitmap indexes. Queries are processed separately, allowing Oracle to implement an optimal processing plan for each.

## 8.9 Assessing performance

### 8.9.1 Defining queries to assess performance of terabyte scale database

Queries will be impacted by scale depending on the features of the query. To performance tune to the demands of varying queries, a set of query features are defined, leading to a three dimensional query description. Queries across all three dimensions were studied.The three dimensions are output content, output size and input predicate.

### 8.9.2 1. What does the query return?

A realistic TAG query can select

- COUNT query - count how many events satisfy a given query predicate
- SELECT ID, FILEID query - return sufficient event and file information to retrieve event data from files
- SELECT ALL - return all attributes for events satisfying a given predicate to perform a direct study eg. to build a root file

In the studies or queries and implementation plans, we establish that there is negligible impact by increasing the attributes returned in a select query, as the performance overhead is in location of the row rather that reading the output data.

### 8.9.3 2. What percentage of the table does the query return?

As the percentage of the table entries returned by a query increases, the overhead to process the query increases. Queries of a similar nature in all but the number of qualifying rows were studied in order to understand the relationship between performance and selectivity.

### 8.9.4 3. Which attributes are defined in the query?

Attributes selected by a user for the WHERE clause query predicate are likely to vary, although there may be some that appear more frequently than others. Queries involving a varying number of attributes, attributes of different index type, Btree, Bitmap, none, and attributes of different data type, distribution and cardinality were studied to see the effect and performance impact.

## 8.10 Assessment of 1TB Performance

To assess performance, two general queries are used

- *Count the events with at least two electrons and missing $E_T$ greater than 10 GeV that are good for physics* - a SUMMARY query
- *Select the events with at least two electrons and missing $E_T$ greater than 10 GeV that are good for physics* - a CONTENT query

Queries are optimised and performed on the partitioning and indexing schema discussed. Query predicates in the tests are based on both index types, the attributes are separated by index type, we use INDEX_JOIN for btrees and INDEX_COMBINE for bitmaps, then INTERSECT the results. The buffer cache is flushed between queries, so no cache advantage is allowed. We increase number of partitions involved as we increase the number of rows returned, holding a consistent percentage rows from each partition, to allow comparison.

For SUMMARY queries, when events are counted, we see in Figures 8.4 and 8.5 that time increases with number of partitions. The increase is linear, so we can predict the time a query will take based on the number of partitions involved. We note that while time is related to the number of partitions, it is not so directly related to the amount of data returned in that n times data from a set number of partitions does not take n times as long. We can therefore predict time with some bounds. We observe that times are

in order of seconds, an encouraging and positive result, as a response time in the order of seconds can be considered an online respone, a considerable advantage to an analysis system and an attractive and useful feature for analysts.

For CONTENT queries, where we select data and return output, we see a linear increase in time with number of partitions and again time overhead is in number of partitions accessed, not data returned from within, this is shown in Figures 8.6 and 8.7. Times are in order of seconds, again a very positive result. Without partitioning, indexing and Oracle Hint strategies developed, these query times are seen to be of the order of hours, demonstrating the significant improvements that can be achieved with schema and query performance tuning.

Count queries perform much better than queries that select and return attributes from the table, shown in comparing the count results in Figures 8.4 and 8.5 with the select results in 8.6 and 8.7. Count queries are performed purely on the index, there is no need to use the table. Select queries perform comparably, regardless of whether the query returns a subset of or all attributes, as a query of this type has overhead in locating and accessing the row, rather than reading of selected attributes. This is similar to the results of Chapter 7, seen when querying tables in the Distributed Data Management system.

Analysts will therefore be encouraged to filter their query using counts, performing adapted queries iteratively to see how many events will be returned, before returning events. It is anticipated that this will both improve an analyst session and minimise unnecessary more costly queries on the database.

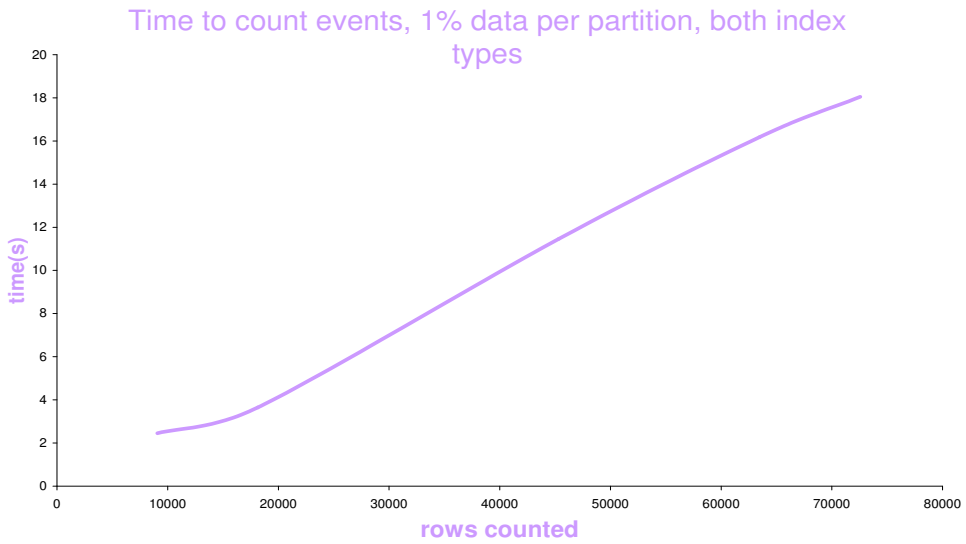



*Figure 8.4: Time to count events when 1% of data is selected from an increasing number of partitions in the query and both index types are in the query predicate. We see that time increases linearly with number of partitions*

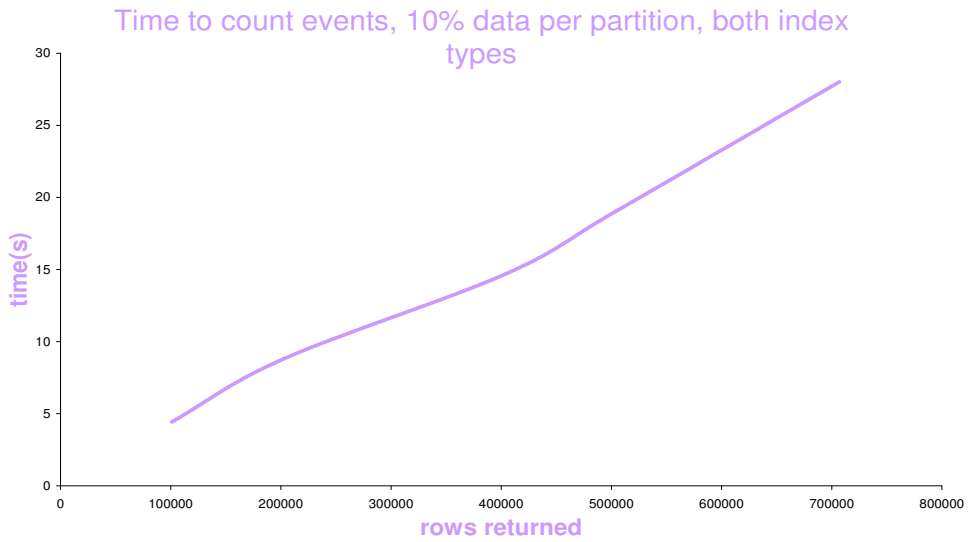



*Figure 8.5: Time to count events when 10% of data is selected from an increasing number of partitions in the query and both index types are in the query predicate. We see that time again increases linearly with number of partitions and returning ten times the data does not take ten times as long*

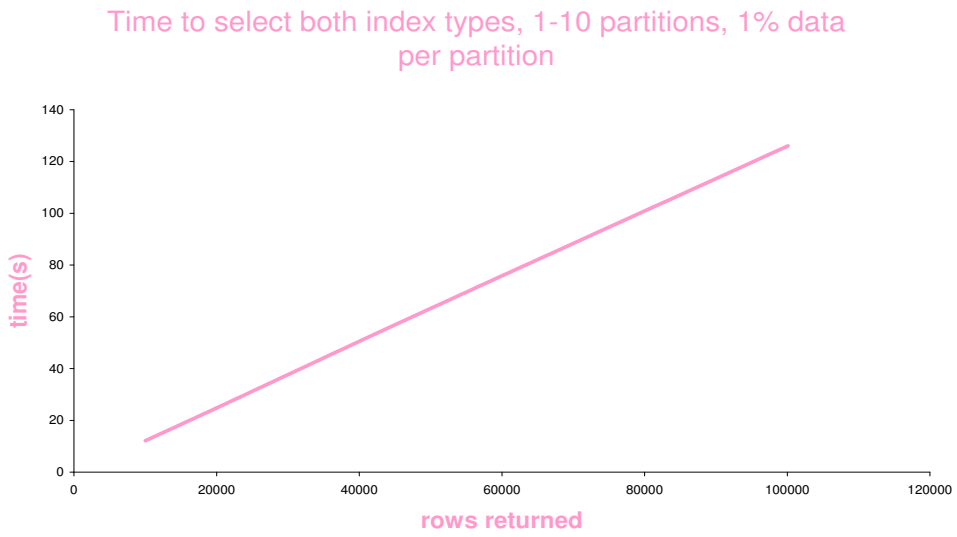



*Figure 8.6: Time to select events when 1% of data is selected from an increasing number of partitions in the query and both index types are included in the query predicate. We see that time increases linearly with number of partitions and times are in the order of seconds*

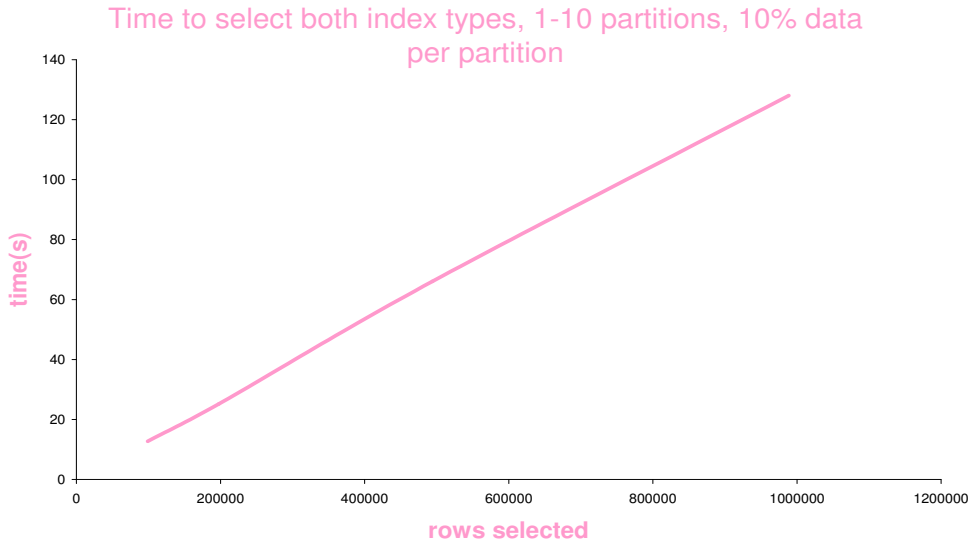



*Figure 8.7: Time to select events when 10% of data is selected from an increasing nmber of partitions in the query and both index types are included in the query predicate. We see that time again increases linearly with number of partitions, selecting ten times the data does not take ten times as long and times are again in the order of seconds*

## 8.11 An Extreme Performance Case

We extend test queries to an extreme case, to understand if the observed linear relations seen in Figures 8.4 and 8.5 for counts and 8.6 and 8.7 for selects, extrapolate indefinitely. The results are shown in Figures 8.8 and 8.9, for COUNT and SELECT queries respectively.

We note that for select queries in figure 8.9, if the observed linear relation is constant and roughly proportional to number of partitions in query, then a query from all 100 partitions would take 20 minutes, but this is not the case in practice, as seen in the plot. In reality we are seeing a threshold case where the sorts required for the query move from memory to disk, resulting in a higher performance overhead. The same query plan is used, but with use of disk. We note however that in all extreme cases, which we do not anticipate will be often performed, optimised performance is still notably faster using the strategies developed than a full table scan.

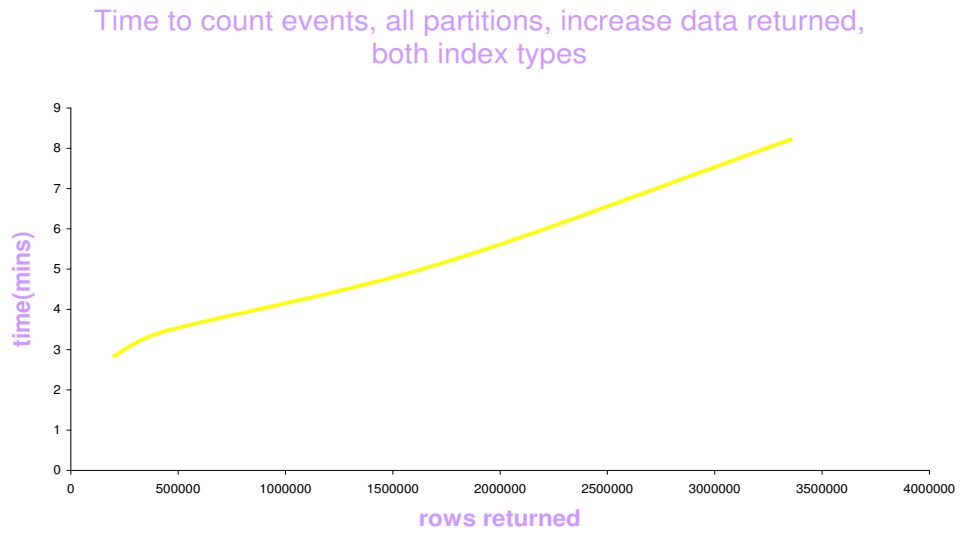



*Figure 8.8: Time to count events in an extreme case when an increasing percentage of data is selected from all partitions in the query. Both index types are in the query predicate. We see that time increases linearly with the amount of data returned and times are in the order of a few minutes*

## 8.12 Stress tests

Stress tests were performed to assess performance of the database in a multi client environment. Expected user query patterns were simulated by creating a sample job of nine optimised queries with a selection of count and retrieve queries across a selection of attributes. Each query scans 1 GB of data, at 220 Hz event rate this corresponds to one hour and thirty minutes in logical units. The session runs on one node of the INT8R cluster, with two CPUs and 2 GB memory.

We establish firstly that the query job described, running alone in a single session environment, would take ten minutes. Stress tests increase the number of concurrent sessions to see the impact on performance and determine the level of multiple clients running optimised TAG queries that can be supported on the system.

Each job divides its time between CPU and I/O, with some cluster time, when satu-

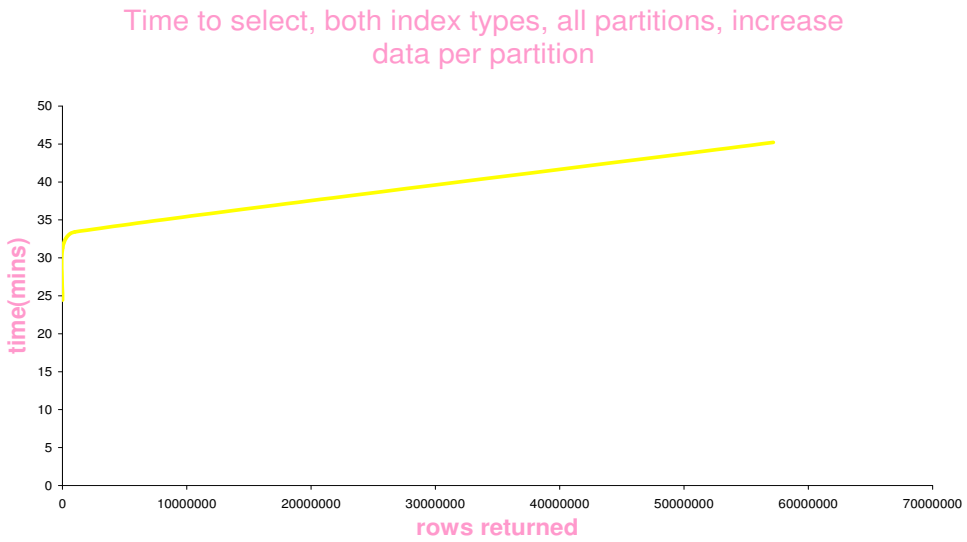



*Figure 8.9: Time to select events in an extreme case when an increasing percentage of data is selected from all partitions in the query. Both index types are in the query predicate. We see that time increases linearly with the amount of data returned and times are in the order of thirty to forty five minutes*

ration is not almost reached, Figure 8.10, Saturation of the machine was seen at one job per minute, Figure 8.11

| Interval(s) | I/O(average/max) | Concurrent Jobs(average/max) |
|---|---|---|
| 60 | 0/35 | 15/20 |
| 90 | 0/25 | 4/6 |

*Table 8.1: Stress test results*

One job per 90 seconds is the equivalent of approximately 9000 queries a day. Each Tier 1 site will have 2 nodes, although upgrades are expected, this rate would occupy one node for TAG queries. Tier 0 production database has 6 nodes, TAG queries of these rates could be supported on one. A hardware upgrade was planned for April 2008, after the stress testing was performed, so increased performance is expected. Once new machines are available, TAG queries can again be stress tested, in an ongoing process of monitoring performance. The results show that there is a need to manage and limit concurrent client sessions, as the application is resource intensive. The results show that a high number of TAG queries can be realistically supported each day, especially as analysts are spread across the globe in multiple time zones, further distributing queries by time.

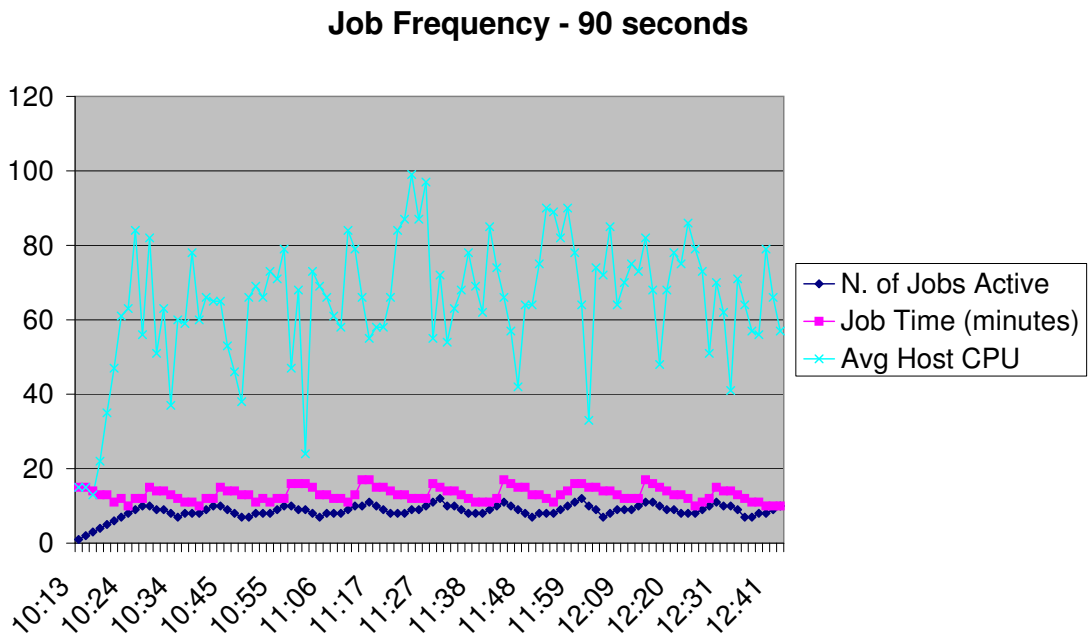



*Figure 8.10: Output from the Database monitoring system when a sample job of nine optimised TAG queries, counts and selects and using both index types, is sent every 90 seconds, we see that the database can process jobs at this rate without saturation*




*Figure 8.11: Output from the Database monitoring system when a sample job of nine optimised TAG queries, counts and selects and using both index types, is sent every 60 seconds, we see that the database saturates in terms of CPU at this rate*

[1]Output graphs for database stress tests provided by Florbela Viegas

## 8.13 Performance Tests of Terabyte TAG Database at Tier One

Performance tests were then extended to Tier 1 sites, to compare results at Tier 0 with similar tests at Tier 1. BNL and TRIUMF were used as Tier 1 test sites.

At each Tier 1 site where TAG queries are tested, a test environment was created similar to the test environment at Tier 0. Ten tables were created, with 100 partitions each. The partitions are 1 GB in size. The indexes are built on 10% of the columns and the queries will try to use these indexes in a similar way to queries used to assess performance at Tier 0. SQL queries, COUNT and SELECTs are performed at BNL and TRIUMF Tier 1s to compare performance with that of Tier 0 CERN. The tests are based on optimised TAG queries used to develop query processing strategy and assess performance of TB oracle database at CERN. Tier 0 has 1 dual-core, Tier 1 BNL has 2 dual-core 3 GHz CPUs and Tier one TRIUMF has 1 dual-core 1.6 GHz CPU. The performance tests will therefore also show how different CPU configurations affect performance.

We see that performance is comparable with Tier 0 performance in two ways

- time for SELECT and COUNT queries increases linearly with the number of partitions
- time for SELECT and COUNT queries is approx constant for an increasing number of rows returned when the number of partitions involved in the query is constant

These are important results as they allow the time for a particular query on a given hardware configuration to be predicted and also confirm that performance behaviour is comparable for Tier 0 and Tier 1 sites.

In general the results in figures 8.12, 8.13, 8.14 and 8.15 show that BNL Tier 1 performs fastest, then TRIUMF Tier 1, then CERN Tier 0 for queries the count events and results show similar patterns select queries that return event, with Tier 1s outperforming Tier 0. Times to return query results when a select query is performed are relatively similar. It

was not anticipated that CERN would have slowest performance so tests were repeated to confirm CERN performance, both with and without flush of buffer cache, as the first set of results for CERN flush buffer cache before each query as a standard of the test environment, so that there will be no advantage in repeat queries due to caching of data in the database, but at Tier 1 we do not have a way of implementing this remotely. The results when the buffer cache is not flushed prior to queries at CERN are shown in Figures 8.16, 8.17, 8.18 and 8.19.

The results demonstrate the effect of flushing the buffer cache between queries and ensuring no caching advantage. We again confirm that performance behaviour is comparable for Tier 0 and Tier 1 sites using the performance test queries developed for the terabyte database performance and scalability tests.

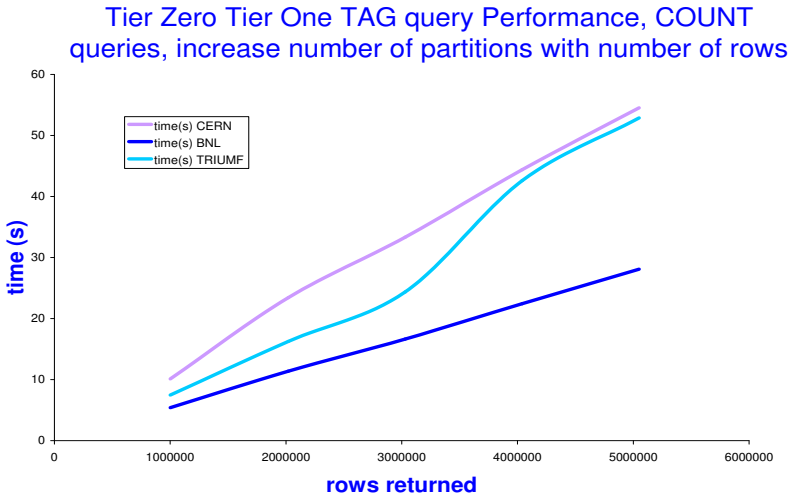



*Figure 8.12: Time to count a constant % of rows per partition, we see a linear increase in time with rows counted and times in the order of seconds. Performance is comparable for Tier 0 and Tier 1, although Tier 1 outperforms Tier 0*

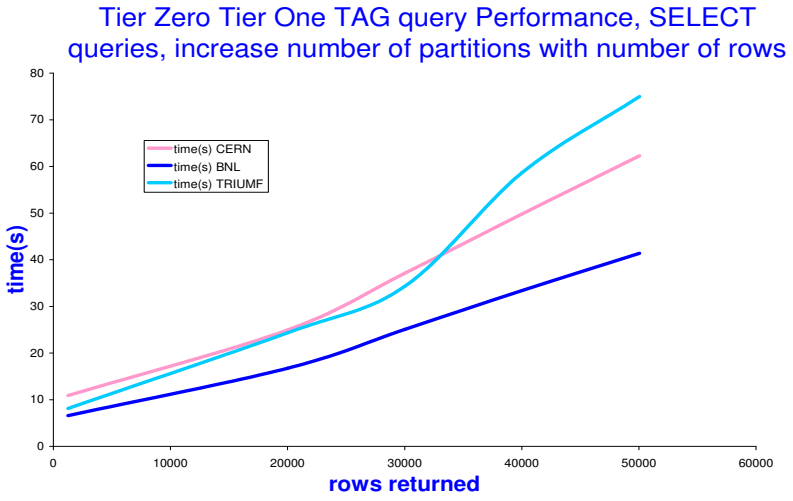



*Figure 8.13: Time to select a constant % of rows returned per partition, we see a linear increase in time with rows selected and times in the order of seconds. Performance is comparable for Tier 0 and Tier 1, although Tier 1 outperforms Tier 0*

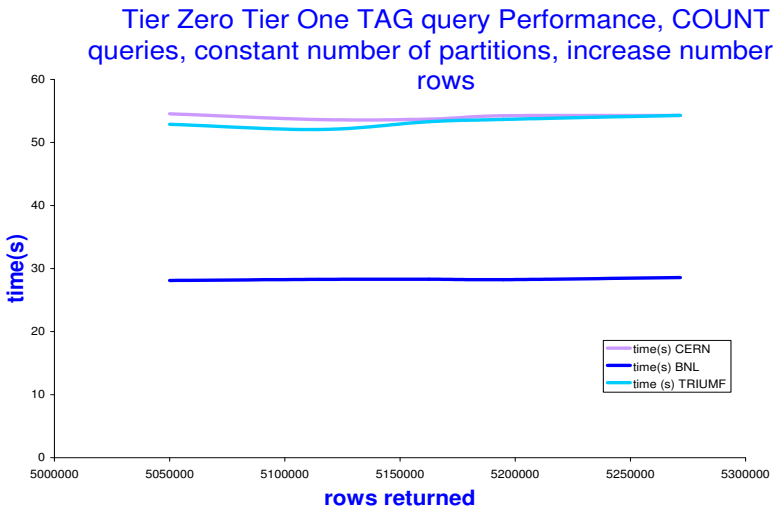



*Figure 8.14: Time to count rows in a constant number of partitions, we see a constant response as the rows counted are increased. Time is of the order of seconds. Performance is comparable for Tier 0 and Tier 1, although Tier 1 outperforms Tier 0*

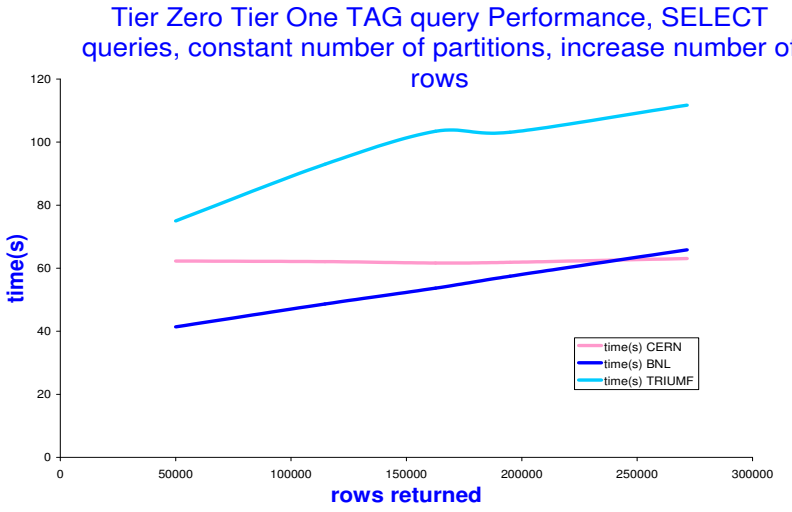



*Figure 8.15: Time to select rows in a constant number of partitions, we see a constant response as the rows selected are increased. Time is of the order of seconds. Performance is comparable for Tier 0 and Tier 1, although Tier 1 outperforms Tier 0*

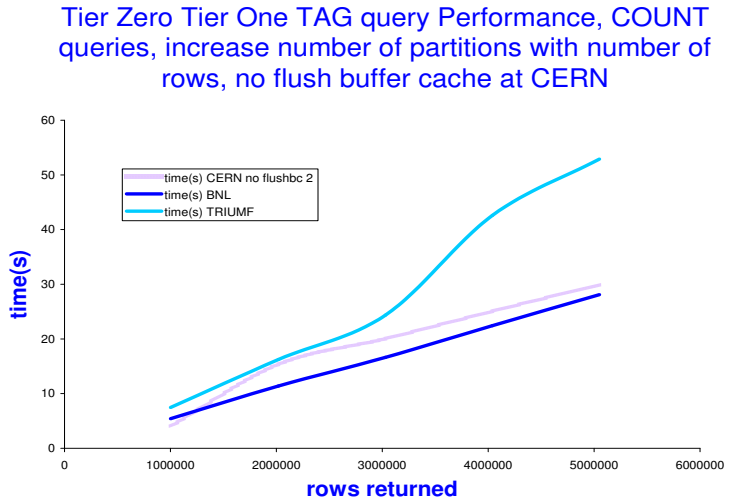



*Figure 8.16: Time to count a constant % of rows per partition, we see a linear increase in time with rows counted and times in the order of seconds. Performance is comparable for Tier 0 and Tier 1, when buffer cache is not flushed, Tier 0 outperforms Tier 1*

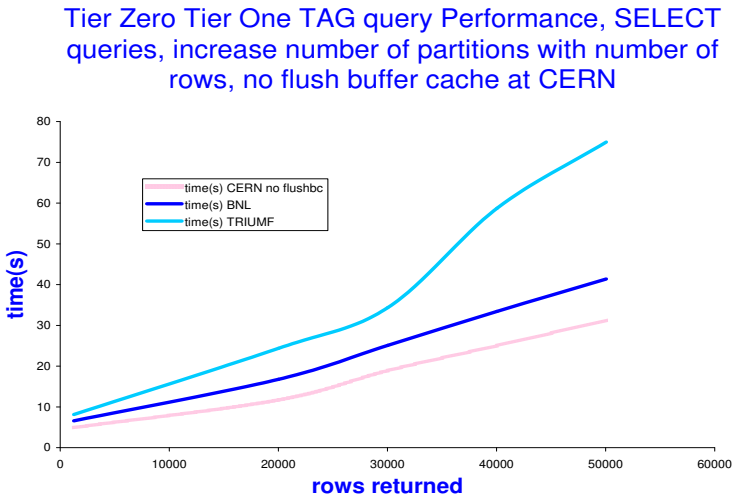



*Figure 8.17: Time to select a constant % of rows per partition, we see a linear increase in time with rows selected and times in the order of seconds. Performance is comparable for Tier 0 and Tier 1, when buffer cache is not flushed, Tier 0 outperforms Tier 1*

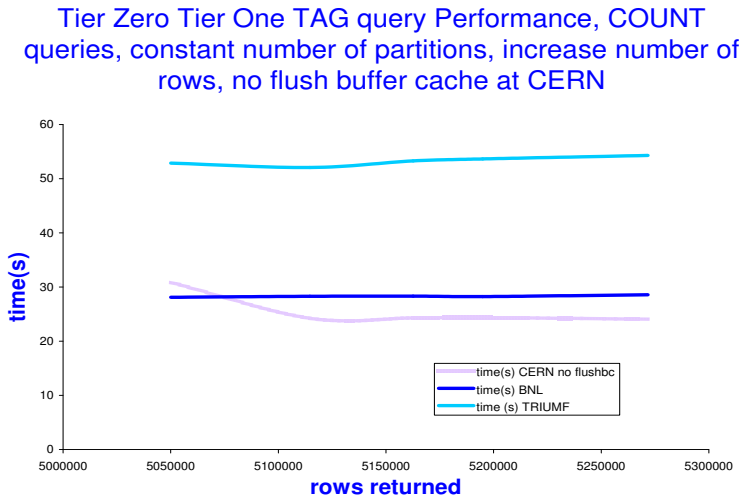



*Figure 8.18: Time to count rows in a constant number of partitions, we see a constant response as the rows counted are increased. Time is of the order of seconds. Performance is comparable for Tiers, when the buffer cache is not flushed Tier 0 outperforms Tier 1*

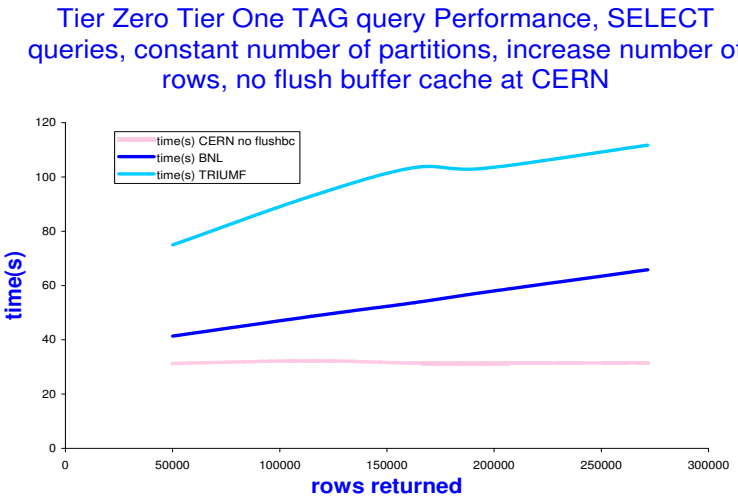



*Figure 8.19: Time to select rows in a constant number of partitions, we see a constant response as the rows selected are increased. Time is of the order of seconds. Performance is comparable for Tiers, when the buffer cache is not flushed Tier 0 outperforms Tier 1*

## 8.14 Conclusions and Future Directions

The ATLAS Event Level Metadata system encompasses data at petabyte scale. Implementing a system that is both performant and manageable at ATLAS scale is a central challenge in the ATLAS TAG database effort. The performance and scalability tests of a relational TAG database presented in this chapter have demonstrated that a relation database can scale to ATLAS terabyte scale. A Global TAG relational database, implemented in Oracle, was sucessfully created at Tier 0. We have presented studies on implementation and performance of a realistic terabyte scale relational TAG Database [55], created a useful terabyte TAG database, assessed the analysis environment in which a realistic relational global TAG collection must perform and investigate strategies for organisation of data within a relational stucture so that data can be both written and read in a useful and realistic way. Using a set of queries defined as useful and likely analyst access patterns we have presented a performance assessement of a realistic terabyte scale relational TAG Database.

The results of the studies in this chapter are important in the development of a relational TAG Database for ATLAS in many ways. Firsly we have demonstrated that it is possible to create, manage and perform useful queries on a relational TAG database at ATLAS scale and within the ATLAS environment. We have studied indexing and partitioning strategies and identified the optimal structure for a database at this scale for ATLAS data. The structure developed, tested and assessed in this chapter has been adopted as the standard for structure of relational TAGs by the Event Level Metadata development effort. We have also defined a realistic set of analysis queries that we can assure will perform optimally and can then implemement an element of control of the queries performed on the database by giving guidance to analysts as to the optimal query patterns to perform in order to return useful and meaningful results. We implement both the query pattern structure and the schema developed in this chapter in the user interface to relational TAG data, described in the next chapter.

We have also presented the performance results for the optimally performing relational

database. The performance results are important as they can be used to provide information to analysts about the optimal way to interact with the database and the performance that can be expected for analysts using a realistic, useful and performant relational TAG database at ATLAS scale.

# Chapter 9

# An Event Level Selection Service Interface - ELSSI

## 9.1 Introduction

A user interface to a relational ATLAS TAG database, the Event Level Selection Service for ATLAS, ELSSI, is presented in this chapter. The ELSSI interface is a web interface intended as a central way for analysts to interact with a relational TAG database. ELSSI is intended to manage the complexity of the relational system and present the user with an intuitive and useful means of creating an event level metadata query and return a result set. ELSSI adopts the concepts of query patterns developed in the scalability and performance tests in Chapter 8, so that analysts create queries which are well performing and anticipated and so that performance can be predicted and impact on the database can be managed. In this chapter we present the development and implementation of an ELSSI user interface. The design concepts and implementation of the interface are described. In terms of design concepts we present efforts to create an intuitive interface, optimal queries on the database and useful user output. In terms of implementation we discuss the components of the ELSSI system and interaction and integration with the wider ATLAS Computing Model. We present the query creation process followed by an ELSSI user and the output options for query results. We discuss security of the ELSSI

system and present experience in TAG attribute distribution and query patterns gained through ELSSI as the interface is released to analysts. ELSSI is impacted by the studies and results of hapters 7 and 8, as the system interacts with the ATLAS Distributed Data Management and Trigger systems guided by the studies and learning of Chapter 7, and the relational database and ELSSI query pattern creation is led by the studies and outcomes of Chapter 8. We present in this chapter interaction of the ELSSI interface with the ATLAS Trigger, the ATLAS Distributed Data Management system and the ATLAS Metadata Interface, AMI. As the TAG schema is intended to be dynamic and can potentially respond to query patterns by changing to suit likely patterns in data queried and query types, through ELSSI we implement a system of monitoring query patterns so that analyst query patterns can be monitored and studied, some preliminary results from query monitoring are presented. We present the ELSSI instances released to ATLAS users, the ELSSI Streams Tests version and the ELSSI FDR version, highlighting the developments of the system up to ELSSI at ATLAS start up in 2008.

## 9.2 ELSSI

The Event Level Selection Service Interface for ATLAS, ELSSI, is the central way ATLAS users can interact with the TAG Database, covering all the steps in the TAG interaction process. Through ELSSI, users can browse events available and their content, construct a useful and optimised query, perform a query on the TAG Database and extract results, both event TAGs and the AOD corresponding to selected events for analysis.

ELSSI is a php, OCP and javascript based web interface, intended to allow physicists to create and perform TAG event selection queries on a relational TAG database. ELSSI intends to be a useful and usable interface that will create an optimised query using simple user inputs. ELSSI versions are named by the data available through the interface. The relational TAG data accessible through ELSSI is organised according to the schema and loading strategies developed in Chapter 8, in the ATLAS scale TAG Database Scalabiity and Performance tests. The first ELSSI version release was ELSSI for ATLAS Streams

Model Tests in 2007. In 2008, ELSSI versions for Full Data Run, FDR, data were released to users.

## 9.3 ELSSI Design Principles

The interface design is lead by the following central concepts

#### ELSSI User Input

An ELSSI user is required to have some knowledge of the data they wish to query and the data they wish to collect in an output for analysis. A user need also meet some security requirements in the shape of a valid ATLAS grid certificate loaded into the browser from which they access ELSSI.

#### Intuitive Query Creation

ELSSI is intended to be self-explanatory, intuitive, that is a user should not require extensive preparation or instruction in order to use the interface and build a TAG query. In the TAG Database Scalability testing described in Chapter 8, sample test queries were performed using SQL commands directly on the database. Users are not expected to know SQL, or how to write an optimised query which takes into account the learning of the scalability tests and the schema of the database however, so the ELSSI web interface design aims to allow users to create queries without anyknowlede of SQL. An optimised SQL query based on the studies of Chapter 8 is constructed by the interface behind the scenes using user ELSSI inputs.

#### Optimal Queries on Database

The ELSSI interface should guide a user through query building in such a way that the eventual query performed is an optimised query based on the outcomes of the scalability tests. It is important to users that queries performed on the database both in their own analyses and in the analyses of others are the optimal queries possible, to ensure a

responsive and efficient database response as multiple queries are performed. Inefficient queries result in slow response from the database and slow access to the database as a consequence. Users are therefore encouraged through the structure and appearance of the interface to use to their advantage the partitioning strategy and schema adopted in relational TAGs, for example users are encouraged to perform count queries before performing retrieval and to filter query by run and stream.

A physicist does not need any knowledge of SQL or query optimisation, the SQL query is built, and optimised, behind the scenes, based on user input to the interface. Furthermore the user should learn something of the structure of TAGs and the TAG database from the physical structure of the interface, so that they may develop sophisticated and useful queries.

#### Useful Output

ELSSI aims to return useful, accurate and meaningful result sets to users. The output has three forms, count of events satisfying a query, retrieval of event TAGs satisfying a query and retrieval or extraction of AOD event data for events satifying an event query. The third output, called Extraction, is the most complex of the three, as it involves an SQL query on the relational database, interaction with POOL Collection Utilities, DQ2 and Ganga tools. The process of creating an AOD collection of events corresponding to an event TAG query is called Skimming.

## 9.4 ELSSI design

The ELSSI system, shown in Figure 9.1 uses and interacts with numerous software tools and ATLAS software systems - php, OCP, javascript, AMI, ATLAS Conditions data, Gridsite for security, the ATLAS Trigger, a retrieval mechanism, POOL Collections, DQ2, and GANGA. PHP is used for web programming, while the javascript code allows an interaction element to be incorporated. OCP allows calls to be made to the Oracle database from within the php web interface.

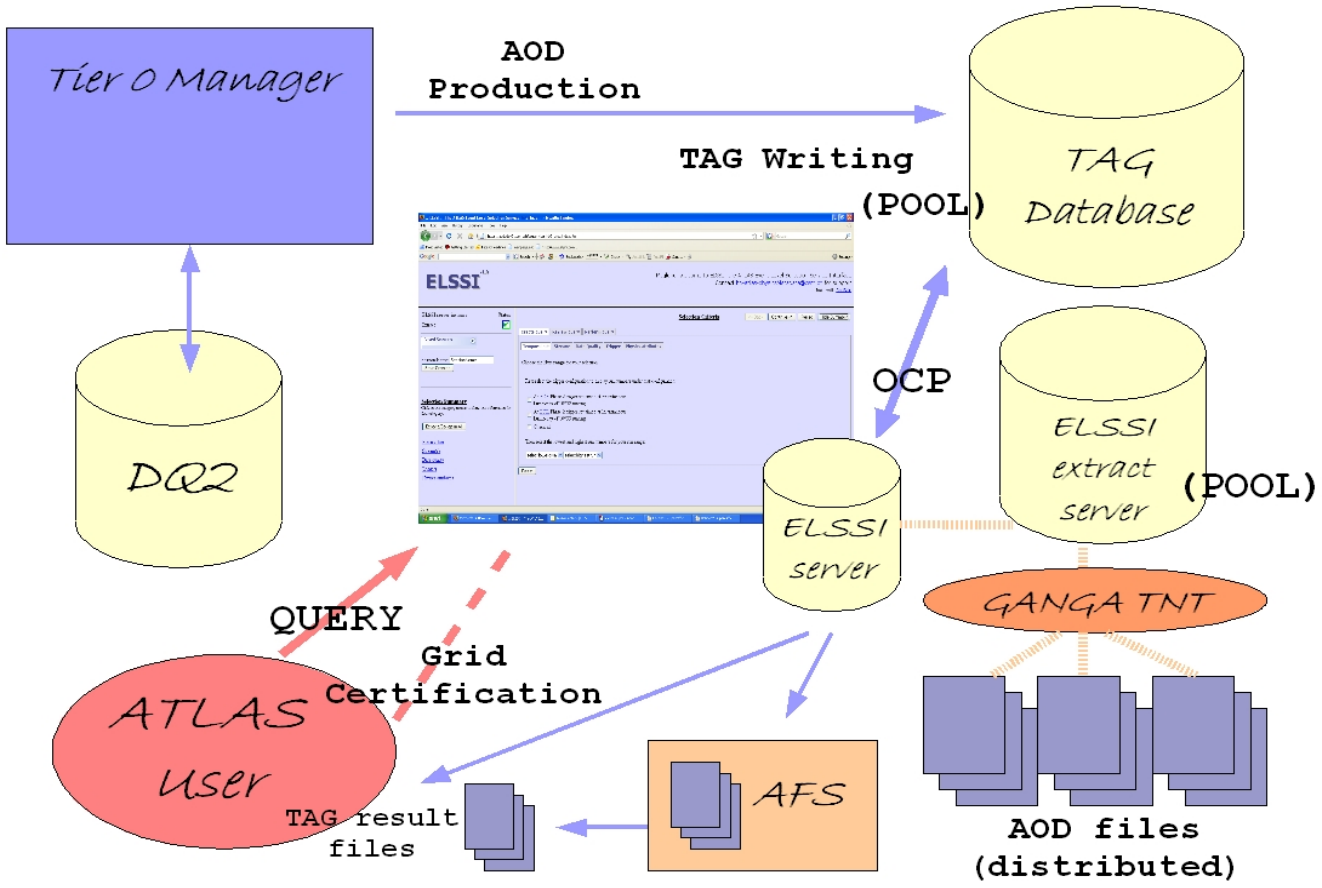



*Figure 9.1: The ELSSI System, an overview of the connections between the ATLAS data production process, TAG Production, the ELSSI Interface, an ATLAS user and further components of the ATLAS and ELSSI system*

As AOD files are written, TAGs are written to POOL ROOT files. Files are registered in DQ2 by the Tier 0 system. TAGs are then imported from POOL files into relational tables, using POOL collection utilities. Once TAGs are in relational tables, they are then available through ELSSI, which is held on a server at Tier 0. A user creates a query using the ELSSI web interface, then performs the query. At this point the browser connects to the relational TAG database using OCP and, if requested, creates an output collection of TAGs. The creation of an output is performed on a designated server, separate from the ELSSI server, and uses POOL collection tools to create an output collection by returning a collection of TAGs. The ELSSI interface browser performs a check on loading and displays a check box to confirm that the extraction server is working properly. The output is then made available in files to a user on AFS space. If a user requests the event data rather than the event metadata for an output to a query, the ELSSI server communicates with GANGA and uses a GANGA tool called GANGA-TNT to collect the event data. GANGA-TNT uses the steps developed in Chapter 7 to interact with the DQ2 system, by looking up AOD files corresponding to a query, creating a new event collection of files corresponding to the results of a query, then creating a new output collection using the lookup and subscription methods developed in Chapter 7. The output collection can then be used as input to an athena job for analysis. The system also has a security component, using gridsite to ensure ELSSI users have a valid grid certificate loaded into the browser from which ELSSI is accessed.

## 9.5 ELSSI to Relational TAG Database

ELSSI communicates with the Oracle relational tag database using OCP, Oracle calls embedded in the php code. Some database queries are performed in response to user selections, some are performed when ELSSI is opened and some are performed on a predefined time elapse basis. This is to ensure that the interface provides up to date and accurate information to the user without performing expensive queries repeatedly when an update every hour or more is enough information.

The interface is intended to be dynamic, so calls are sent to the database based on user input. For the Streams test data fewer dynamic calls are necessary due to the consistent nature of the TAG data, so many query are performed on load for the Streams test data rather than on user input, but the interface is designed for a realistic case and dymanic queries will be performed once the data so requires. Dynamic and on loading queries include which attributes are available to query on for a given stream run selection and which runs are available within a stream selection. Time scheduled queries are event counts per stream, these queries are performed at set time intervals so there is no unnecessary load on the database created by repeating the query every time the ELSSI interface page loads or refreshes.

## 9.6 ELSSI Output

ELSSI is designed to manage all the stages of user interaction with the TAG Database, allowing a user to create a useful query on the TAG database, perform the query, then return useful output for input to analysis. User output for analysis can be statistical data about the result set of a TAG query, a collection of event TAGs, or a collection of event data corresponding to the event TAGs returned by a query.

FDR versions of the ELSSI browser onwards include an extraction feature, allowing a user to create and return an event collection containing the AOD event data corresponding to the output of a TAG query. The server used to process the extraction is distinct to the ELSSI server. The ELSSI homepage had a visual check, to show that the extraction server is available at the time. The process of creating an event collection of AOD event data corresponding to events returned by a TAG query is referred to as Skimming. The Skim process uses GANGA-TNT to create an output event collection.

## 9.7 ELSSI and AMI

The ATLAS Metadata Interface, AMI, [56], [57], is a bookkeeping interface where metadata for ATLAS datasets can be stored in a generic way. ELSSI connects to AMI allowing a user to store Collection metadata, that is, information about an output collection created by a user through a TAG database query. Information is captured about the query used to create the collection, the collection name and the user who created the query. AMI can then be used to keep record of collections created using ELSSI.

## 9.8 ELSSI and Trigger Decisions

For the first ELSSI Interface released to users, the Triggers available within a data sample were constant throughout and so were coded directly into the ELSSI interface code. For FDR data samples, a more realistic Trigger Implementation, where the Trigger attributes available in the data sample are read from the TAG data and a Trigger decoding strategy described in Chapter 7 is implemented for Trigger selections in the interface.

## 9.9 ELSSI Security

Access to the ELSSI browser requires a Grid Certificate. This security is designed to control access to the browser and the database underneath and allow only trusted users to access the data. Implementing security in this way protects the system from both accidental and intentional misuse. An ELSSI user must load a valid grid certificate into the browser from which ELSSI is accessed. ELSSI identifies users by the Distinguished Name, DN, contained in the certificate identification and this name is used in the welcome banner on the ELSSI page, in the saved session features and in the ELSSI bookkeeping facilities in which users and queries are identified and stored.

## 9.10 ELSSI Query Creation

To create a query, ELSSI leads the user through five create query event selection steps, shown in Figure 9.2




*Figure 9.2: Creating a query using ELSSI takes a user through the five selection steps*

### Stream selection

For the Streams test TAG data, five inclusive and six exclusive streams of data are available. In the relational database, each stream is a table in the database, following the model for TAG data in which each ATLAS physics stream will be stored in a table. The user may select one or more streams in the interface. In doing so the user applies a first filter to the data.

### Temporal selection

After Stream selection, the database is queried to retrieve the temporal conditions available for the selected stream. In the Streams test data, the temporal condition is run number, as it will be for real data. With real data there may be more details about each run, for the Streams test data, the ten runs are simply numbered. Each Stream table in the relational TAG database is partitioned horizontally by run, as is the model for real data daveloped in Chapter 8. The user may select one or more runs, either by number or by range. This applies a further filter to the data of interest.

#### Data Quality selection

The user may select data by Data Quality conditions. In the Streams test data, data is marked Good when a file is deemed complete. The information detailing the Data Quality is imported from the Conditions database into the TAG database and the interface queries the new table in the TAG database to select data by quality. In real data there will be many more Data Quality options.

#### Trigger selection

In the Streams test data the Trigger configuration is fixed and all available Triggers are active through all runs. The user is offered a realistic two step selection, is the Trigger ative, then did the trigger fire. As the menu is fixed for Streams test data, there is no need for translation of Trigger configuration or bitmap compression of Trigger attributes as there will be for real data. So although the Streams test user interaction mimics the realistic situation, the interface and database operate in a simplified way. In the FDR ELSSI interface onwards, compression and decoding of available Triggers are implemented. The user is offered all the available Trigger Menus for the selected temporal and stream selections. Triggers are a central way of filtering and selecting data for data analysis, so it is important that the ELSSI Trigger selections are performant.

#### Physics selection

The physics attributes offered to a physicist for selection are read from the TAG database and are based on the stream and temporal selections a user has made. As with the Trigger selections, physics attributes are a central means of filtering data for data analyss.

## 9.11 Counts and Retrieves

Count queries are much less expensive on the relational database, demonstrated at ATLAS scale in the Scalability and Performance tests of Chapter 8, so encouraging analysts to

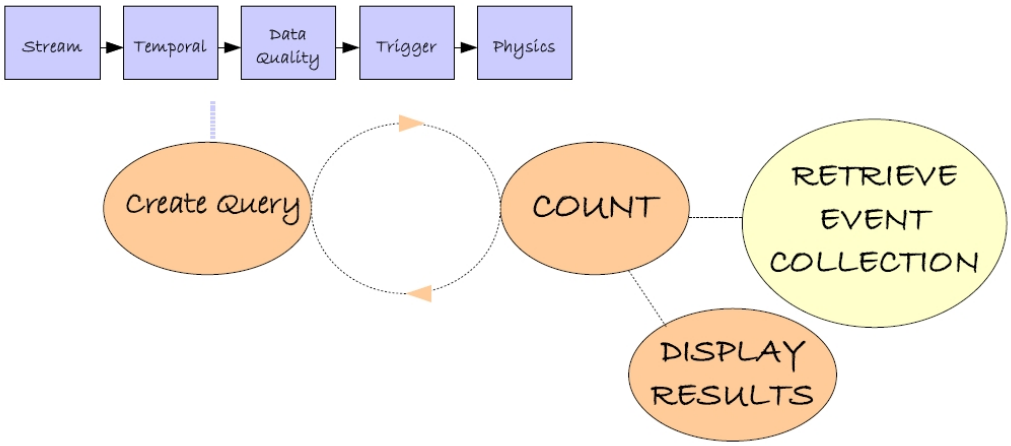



*Figure 9.3: The ELSSI process of creating a query, a user is encouraged to iteratively count events returned by a query, display results in tabular and graphical form and then select an event collection as output*

use counts before retrieves will allow for optimal performance of the database by reducing unneccessary retrieve queries. A user is encouraged to create a query using the steps presented, then to perform a count query to check the number of events that the query will return. The count step is both helpful to an analyst, as it allows queries which are too strict and return no or too few results and queries which are too broad returning too many or all events to be filtered before retrieving results. Count queries also allow a user to understand the effect of filters and see how many events satisfy a given query.

## 9.12 ELSSI Optimal Query Processing

For the Streams test data, the number of event tags and therefore database size are small and the query processing strategy developed in the scalability is not needed, as Oracle manages queries on a database of smaller size without need for optimal approaches. Later when ELSSI accessed larger amounts of data as in the FDR data, the query processing strategy developed in Chapter 8 is written into the php interface code, so a fully optimised query is sent to the database. The user need not know the details of optimisation as ELSSI handles the SQL optimisation.

## 9.13 ELSSI Query Monitoring

A query monitoring system is incorporated into ELSSI in order to store and study queries submitted to the database. A table was created within the relational TAG database and user queries and corresponding hostname is stored as queries are performed. In this way it is possible to gather some general statistics about how many users are connecting to ELSSI, which attributes are popular in queries and which are not and general query patterns, such as frequency of counts vs selects. This information can be used to influence future relational database schema developments and to ensure recommended query patterns are being adopted so that queries are optimal.

## 9.14 ELSSI saved sessions

ELSSI has a saved session feature, allowing users to store a record of previous queries created using the interface. The saved session feature uses browser cookies to store information.

## 9.15 ELSSI and TAG Value Distibutions

ELSSI uses Java SQLtuple, [58], to allow users to plot value distributions of event TAG attributes satisfying a query. A user can use this feature to gather statistics about value distributions for events satisfying a TAG query, allowing further modification to a query if required, or gathering of basic statistics about an output result set, asshown in Figure 9.4.

## 9.16 ELSSI at Tier 0 and Tier 1

At Tier 0 ELSSI is hosted on atldbdev01, a development server at CERN. The extraction server is on a separate server. Results from event TAG queries are stored on AFS for a nominal amount of time. It is expected that as demand for the system increases during startup and data taking, users will be requested to copy the event TAG result set to some

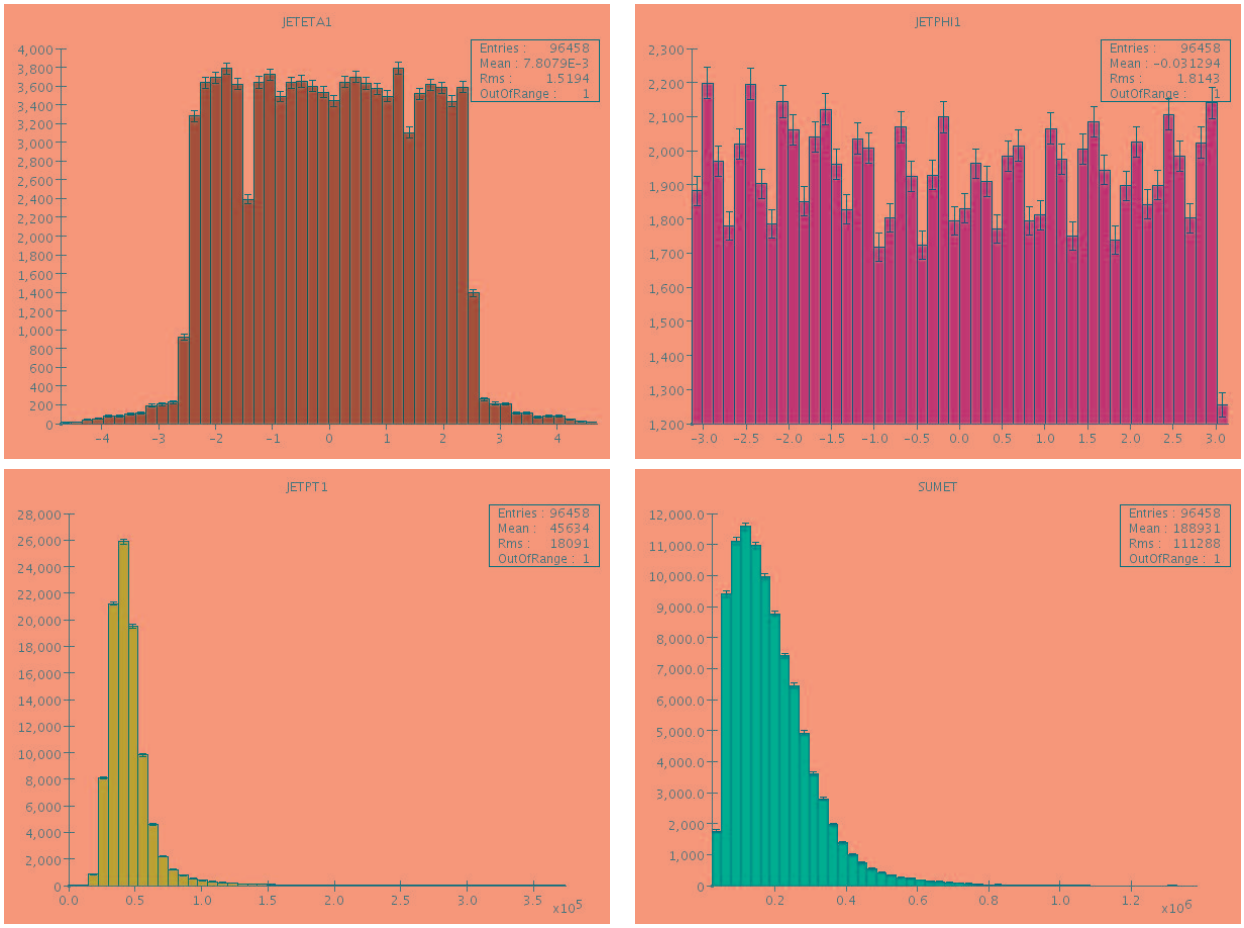



*Figure 9.4: Value distributions for the exclusive electron stream, created using SQLtuple with the relational TAGs attributes in ELSSI for the Streams tests, left to right, $\eta$, $\phi$, $p_T$ of the first jet in an event and the sum of missing $E_T$*

Grid or local storage, as the initial AFS area will be cleared periodically in response to demand for space. ELSSI can also be installed at Tier 1 and a local ELSSI instance can point either to the Global TAG Collection at CERN, or to a local instance of a relational TAG database.

## 9.17 ELSSI Streams version

In May 2007 a TAG Database was created using data from the ATLAS Streams Tests. An interface was created to allow users to create SQL queries and query the database, and the database and interface were made available to users. During this time a system of monitoring user queries was established in order to develop learning about query patterns,

as these may then influence the development of the relational database schema. Some monitoring system will be necessary as the database expands to scale, so that performance can be monitored and optimising strategies can be developed in response to query patterns and increased data volume.

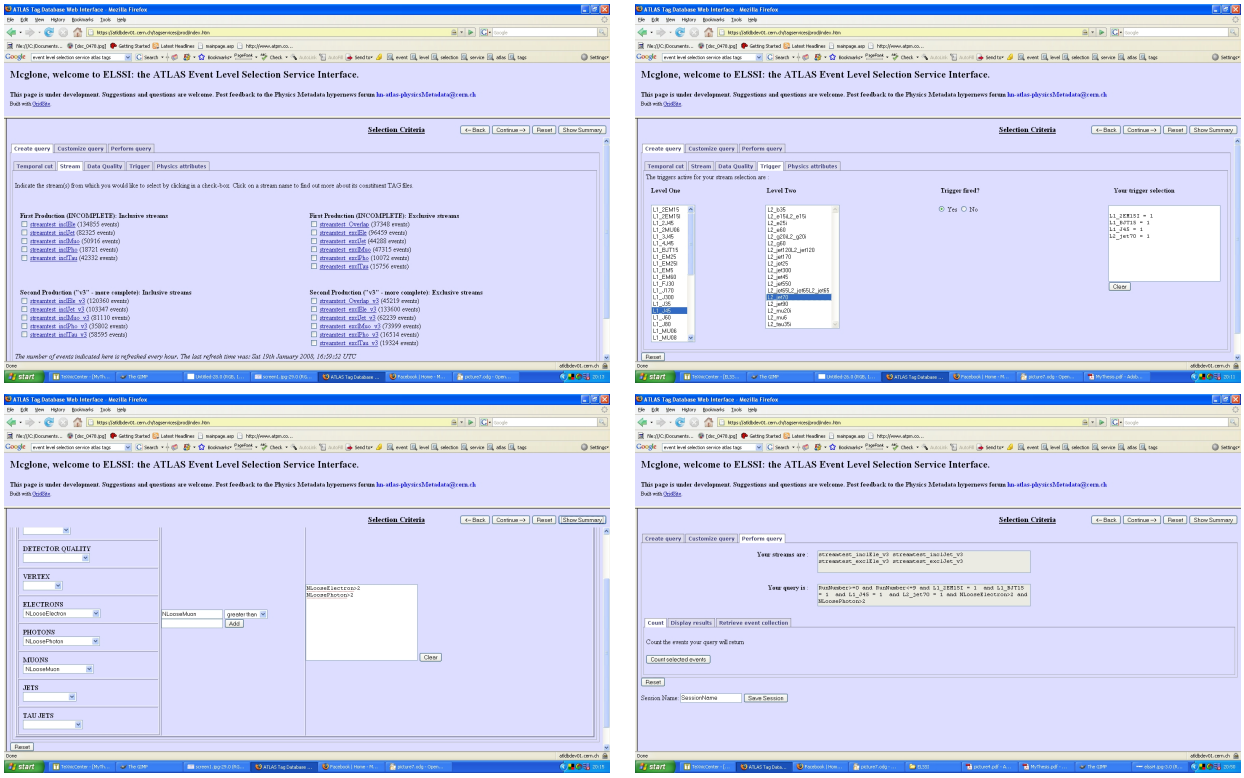



*Figure 9.5: ELSSI for Streams Tests, left to right, Create Query Stream selection, Level One and Level Two Trigger selection, Physics Attribute selection and the Perform query page*

### ATLAS Streams tests

In 2007 ATLAS performed a set of tests of the Streaming model to be adopted by the experiment, where Streams are a method of populating files by some first instance selection criteria. ATLAS is expected to define around ten streams each described by some physics, Trigger or analysis criteria or some combination of these. Stream definitions may evolve as experience of analysis is gained. AOD production introduces streaming by physics criteria in this way to loosely represent analysis access patterns. The intention is to group events into AOD files by likely analysis to minimise the number of files which have to

be accessed in an analysis. Streams are not a definitive set of events for analysis. As streaming takes place at AOD production and the input ESD files are defined by runs, output streams, by dividing events from the same run into many streams, are a disjoint partition of a run.

The Computing Model defined the model to be one of Exclusive Streaming, where each event is written only once, but there were suggestions that an Inclusive or Overlapping model, where an event is written to every stream to which it qualifies would be more useful without adding cost. The Streams tests were intended to study both models and identify that most useful to the collaboration. The Streams tests proved a timely opportunity to implement and introduce physicists to a relational TAG Database.

#### Exclusive Streams

The Streaming model defined in the Computing Model is that each event is written to one single stream, an Exclusive Streams model. Many events may qualify for multiple streams, but an event is written to only one stream to avoid event overlap in an analysis of events that crosses stream boundaries. As many events may qualify for more than one stream and as replication of AOD data would be an unnecessary use of resources when the aim of streaming is optimisation of access to AOD data, physics groups define 10 mutually exclusive and maximally balanced streams.

#### Inclusive Streams

Alternatively an Inclusive streams model is considered, where events are written to all streams to which they qualify. The Streams models are compared in the Streams test. ELSSI aims to support access to the Streams tests TAG data so that analysts can compare the streaming models.

#### The Streams Test Relational TAG Database

The Streams tests produced around eight million event tags, with 3600000 events each put into an inclusive and exclusive stream, plus an overlap stream for an exclusive stream

model. Data production happened twice, with updated software versions, giving sixteen million event TAGs. ATLAS data production produces file based event TAGs at AOD merging, then relational TAG loading was performed by hand using POOL Collection utilities to produce a relational TAG database.

The structure of the database is based on the outcome of scalability tests in Chapter 8. The relational TAGs are implemented in Oracle, although the database is small in size compared to the scalability tests, the optimal schema is adoped as it is the TAG database model. So each stream is a table and within each stream table events are partitioned by run. The TAGs have approximately 200 attributes and each attribute is indexed. The trigger implementation in the Streams test TAGs is not as sophisticated as it will be for realistic data or as it later for FDR data, so no Trigger bitmap compression or translation of Trigger menus was needed, instead a fixed trigger menu is used throughout.

**Query Monitoring for ELSSI Streams Test**

Some of the information gathered from the SQL query monitoring for ELSSI in the first months of release are shown below. Users were introduced to ELSSI as the gateway to ATLAS data analysis using TAGs through the Streams Test ELSSI version, the first ELSSI release. Many tutorials were presented to users on the use of ELSSI and event level metadata TAGs in analysis. The ELSSI system captures some metadata about the submitted query, including the query string, a user identification and an output collection name an output collection is created.

In statistics gathered for the Streams test relational database, accessed by users through ELSSI, we see a little over 5000 submitted queries, around 80% of which are count queries, showing that users are complying with the pattern of performing repeated count queries and select queries after a query has been refined, minimising the impact on the database and streamlining the user interaction process.

## 9.18 ELSSI FDR version

The ELSSI FDR version was released in 2008 and was based on the ELSSI Streams test version with additional features available to users. Most significantly these are a more realistic translation of Trigger menus and the ability to use the extraction server and GANGA-TNT to return a collection of AOD events corresponding to a TAG query performed using ELSSI, as well as a number of new features designed to improve the experience of using ELSSI for analysts.

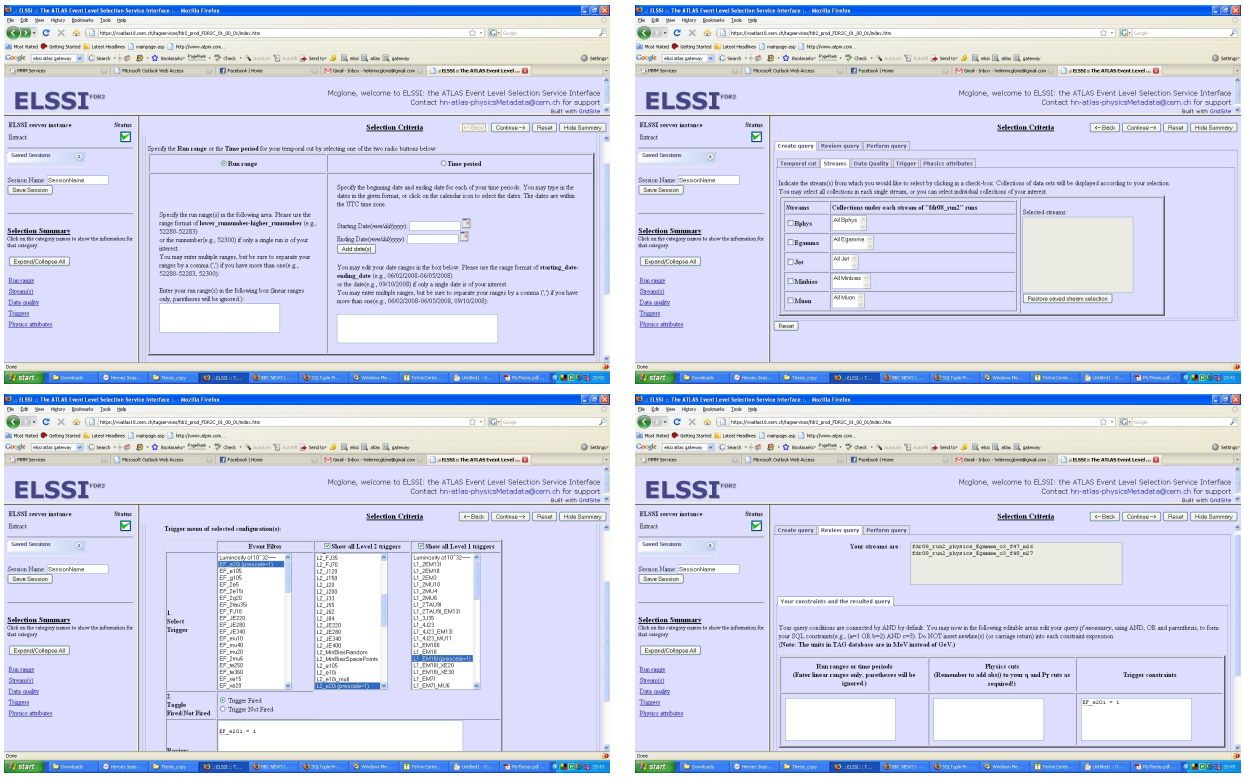


*Figure 9.6: ELSSI FDR run 2 interface, left to right, the temporal selection by run and time, the stream selection, the trigger selection with e20i selected and the query review page*

The ELSSI Interface for FDR data is shown in Figure 9.6. ELSSI FDR uses the Temporal - Streams - Data Quality - Trigger - Physics query creation pattern designed for the Streams Tests ELSSI instance. For the temporal selection, users can select by run number or time period, selecting data production start and end dates, a more detailed selection available than in ELSSI Streams Test instance. Alternatively for ELSSI FDR an

analyst can select an FDR Trigger configuration, FDR phase 2 Trigger set for $10^{32}$ or $10^{33}$ integrated luminosity, to automatically select data runs using the selected configuration. As additional Trigger information available in FDR data, this option in the ELSSI FDR version is an addition to the ELSSI Streams test instance, when only a single Trigger configuration was available. For Stream selection, ELSSI FDR offers the five physics streams of the FDR data, Bphys, Egamma, Jet, Minbias and Muon, and a user can select some or all of the data collections in each stream. The Data Quality selection for ELSSI FDR links to a table in the TAG database with conditions data imported from the Conditions database, allowing selection of complete luminosity blocks or all data. The Trigger selection for ELSSI FDR is improved from the ELSSI Streams version, in ELSSI FDR the Trigger configuration offered in the create query stages is based on the luminosity range selected, where the Trigger menus are read from Trigger information tables held in the TAG database. The Trigger selections in ELSSI are encoded as in the model developed in Chapter 7. The Triggers offered are Event Filter, Level One and Level Two selections. For ELSSI FDR the physics attributes are read from the TAG database, using a metadata type attribute in the TAG database to group the attributes by type. In the ELSSI FDR perform query, users may perform counts, display results in a table and create a file of event metadata corresponding to the query output as for ELSSI Streams. In addition a user in ELSSI FDR may also create histograms and graphic displays of selected data, as shown in Figure 9.4, and use GANGA-TNT to create an output event collection of event data outputed by a query.

## 9.19 Conclusions and Future Directions

ELSSI, a web interface intended as a central way for analysts to interact with a relational TAG database, has been presented in this chapter. ELSSI manages the complexity of the relational system and presents the user with an intuitive and useful means of creating an event level metadata query and return a result set. The ELSSI interface is a dynamic system and is under continuous development as more ATLAS is available and relational

TAGs are created. We have presented the design concepts, development and implementation of the ELSSI interface and shown that ELSSI adopts the concepts of query patterns developed in the scalability and performance tests in Chapter 8, so that analysts create queries which are well performing and anticipated and so that performance can be predicted and impact on the database can be managed. We have demonstrated a system of monitoring query patterns through ELSSI, so that the TAG system and ELSSI can perform dynamically, responding the demands of analysts. ELSSI interacts with multiple components of the ATLAS Computing Model and software environment, these interacts have been laid out. ELSSI interface developments were shaped and impacted by the studies in Chapters 7 and 8 in this thesis, as the system interacts with the ATLAS Distributed Data Management and Trigger systems guided by the studies and learning of Chapter 7, and ELSSI query pattern creation and the underlying schema of the relational TAG database available through ELSSI are led by the studies and outcomes of Chapter 8. The ELSSI system at ATLAS start up in 2008 has been presented in this chapter. ELSSI is an ongoing development project and continues to be introduced to analysts and developed dynamically in response to analyst and data needs.

# Chapter 10

# Neural Net Analysis of $t\bar{t}H, H \rightarrow b\bar{b}$

## 10.1 Introduction

In the Standard Model, a light Higgs boson is described as being in the mass range, $m_H \leq 135GeV$. For this range, $H \rightarrow b\bar{b}$ is the leading decay mode at the Large Hadron Collider. $t\bar{t}H$ production is primarily via gluon-gluon interactions at 90%, the remaining 10% are quark antiquark interactions. We study the potential for discovery of a light Higgs using a semi-leptonic final state, where a top quark is used as a lepton trigger, with $30fb^{-1}$ integrated luminosity. Events where a Higgs boson is produced with associated $t\bar{t}$ have a distinct signature due to the presence of two $W$s and four $b$ jets. In this chapter, a neural network is used to analyse a Monte Carlo simulated data sample of $t\bar{t}H, H \rightarrow b\bar{b}$ signal and $t\bar{t}j\bar{j}$, $t\bar{t}b\bar{b}$ QCD and Electroweak background events, to assess events passed in to the network and to identify signal and background events. A neural net is used to separate $t\bar{t}H, H \rightarrow b\bar{b}$ signal from background events, to potentially improve the significance of a light Higgs mass channel.

In this study two neural net methods are developed. A neural net uses a collection of event characteristics, called input variables, to distinguish signal events from background. The input variables when considered alone are not sufficient to classify events but when combined and correlated can be potentially useful to classify events. Two sets of input variables are considered in this study and the performance of the corresponding neural

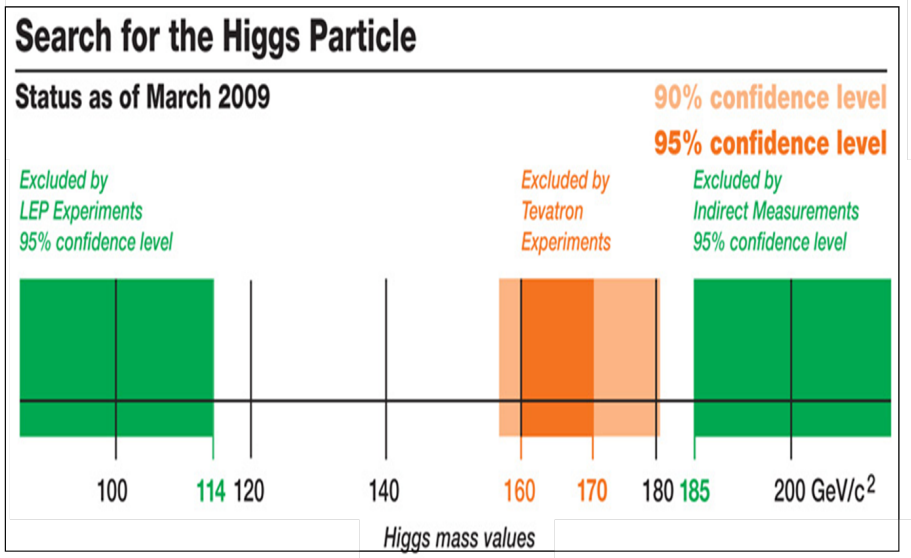



*Figure 10.1: Search for the Higgs Boson, Higgs mass values status as of March 2009*

networks are compared.

The first set of event characteristics are variables that assume a Higgs boson can be reconstructed in each event in the Monte Carlo data sample. There are three scenarios where a Higgs boson can be reconstructed in an event. Firstly, the Higgs boson may be correctly reconstructed in a signal event. Secondly, a Higgs boson may have been incorrectly reconstructed in a signal event, events of this type are combinatoric background events. Lastly, a Higgs boson may have been reconstructed in a background event where in fact no Higgs boson is present.

The first set of event characteristics considered for a neural network are called the *Higgs Input Variables* as they are derived from events in which a Higgs boson is present. In this case there are two concerns in the performance of the neural network. Firstly, the network may be misled by the input of event characteristics that describe incorrectly reconstructed Higgs bosons, both combinatoric background and background events. Secondly, when the reconstruction of a Higgs boson is required for an event to be considered for input to a neural network, there are fewer events available to be assessed by the neural net, as events which do not meet the preselection criteria for reconstruction of a Higgs boson are not used. The $t\bar{t}H, H \rightarrow b\bar{b}$ analysis described is already statistically limited by the number of Monte Carlo events simulated by the experiment and available for analysis, so it is

preferable not to reduce this sample any further than necessary and to make best use of the limited statistics available.

A second collection of event characteristics is proposed for which the preselection does not require that a Higgs boson is reconstructed in the event. This increases the number of events available to the neural network. In an effort to reduce the combinatoric background, the second set of event characteristics does not assign a single combination pattern of jets or assume that the assignments are accurate, instead the combination of possible jet pairings and the corresponding reconstructed values for the top quarks in the event are passed to the neural network. In this way the neural network can be responsible for identification of subtle distinguishing characteristics in signal and background events without the result being potential affected by combinatoric background events in the input Monte Carlo simulated data sample. The second set of event characteristics are called *Generic Input Variables* as they are derived from the generic characteristics of an event.

In this chapter we introduce the $t\bar{t}H, H \rightarrow b\bar{b}$ channel with an aim to identify and understand potential distinguishing variables for use in a neural network. Recent analysis and results for the channel are presented. In general previous studies use event characteristics similar to the *Higgs Input Variables* used in this study to potentially improve the signal to background ratio in a Monte Carlo sample [60], [61], this study is the first to develop a neural network based on the *Generic Input Variables*. The Monte Carlo simulated samples used in this study are described and the neural networks for both sets of event characteristics are presented. The event preselections for the analysis are presented and the differences between the preselection for the *Higgs Input Variables* and the looser set of cuts for the *Generic Input Variables* are described. The *Higgs Input Variables* and the *Generic Input Variables* used in this study as input for neural network analysis are presented and described. Distributions for each input variable with comparison of values for signal and background processes are presented. The results of the study are presented and the neural network methods are compared.

## 10.2 Events in the $t\bar{t}H, H \rightarrow b\bar{b}$ channel

### 10.2.1 Signal events

$t\bar{t}H, H \rightarrow b\bar{b}$ events are described as having one of three possible final states, these are fully leptonic, fully hadronic and semi-leptonic states. Top quarks in the events decay almost exclusively to $b$s and $W$s, so $t\bar{t}H, H \rightarrow b\bar{b}$ final states can be identified in this way by the pattern of W final states in each event. $\tau$s are excluded from analysis in this study in order to be consistent with earlier analysis [61].

The fully leptonic state is the easiest to trigger as it involves two isolated leptons, however with a low branching fraction at 10% and two neutrinos in the event, reconstruction of the top quarks is not possible. The fully hadronic state has a high branching fraction at 46%, however a large QCD multijet crosssection makes triggering using jets difficult. The semi-leptonic state has a branching fraction at 44% and the single isolated lepton in the event is a useful trigger, along with high jet multiplicity, many $b$s and missing transverse energy originating from a neutrino. A semi-leptonic final state is considered for the analysis performed in [61] and is adopted as the signal in this chapter. The semi leptonic final state for $t\bar{t}H, H \rightarrow b\bar{b}$ is shown in Figure 10.2.

### 10.2.2 Background Events

The Physics backgrounds to $t\bar{t}H, H \rightarrow b\bar{b}$ are $t\bar{t}b\bar{b}$ and $t\bar{t}jj$ events. Only $t\bar{t}$ events with six jets are present in the preselected sample and requiring that four of these jets be $b$ jets reduces the $t\bar{t}jj$ background. In $t\bar{t}$ events, most of the extra jets in the event are light jets, so $t\bar{t}jj$ reduction requires a well performing $b$ tagging algorithm with good rejection of light jets. $t\bar{t}jj$ is the forms part of the reducible background for $t\bar{t}H, H \rightarrow b\bar{b}$.

$t\bar{t}b\bar{b}$ production occurs through either QCD or Electroweak interaction processes. The QCD interactions are reducible and the Electroweak interactions are irreducible. The QCD production cross section is ten times the size of the Electroweak background cross section. The Electroweak and QCD background are shown in Fsigure 10.3.

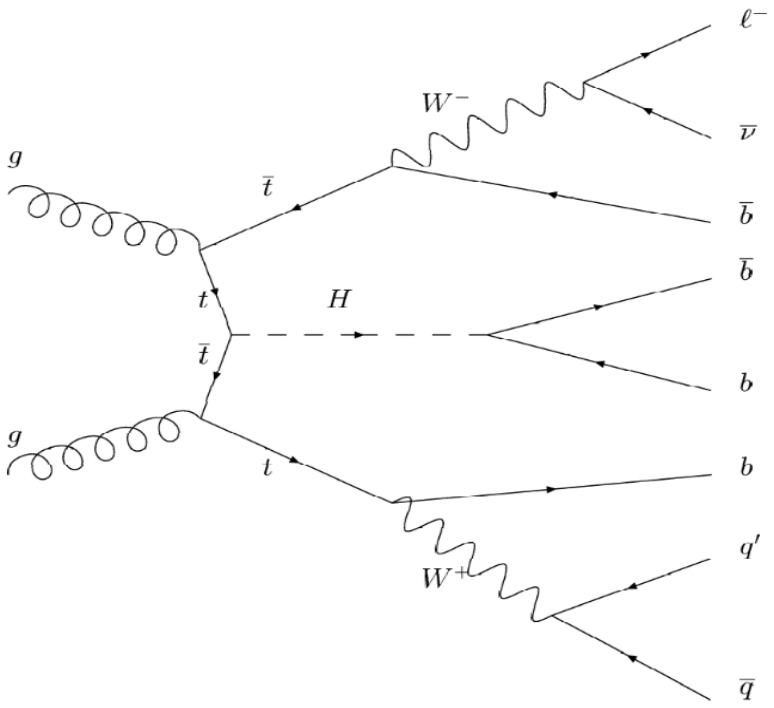



*Figure 10.2: $t\bar{t}H, H \rightarrow b\bar{b}$ semileptonic signal event*

Combinatorial background for $t\bar{t}H, H \rightarrow b\bar{b}$ occurs when the reconstructed objects in the final state are misassigned.

## 10.3 $t\bar{t}H, H \rightarrow b\bar{b}$ Recent Analysis

The $t\bar{t}H, H \rightarrow b\bar{b}$ channel was studied in the ATLAS Technical Design Report, [18], by Cammin, [60], and most recently as part of a Physics Study performed as part of the ATLAS Computing System Commissioning, CSC, exercise, the results of which were published in a $t\bar{t}H, H \rightarrow b\bar{b}$ CSC note, [61]. The CSC note presents the most recent state of the art analysis performed in respect to the $t\bar{t}H, H \rightarrow b\bar{b}$ channel.

In the CSC note, three analysis methods are presented, a Cut-Based Analysis, a Pairing Likelihood Analysis and a Constrained Fit Analysis, each based on 30 $fb^{-1}$ of $t\bar{t}H, H \rightarrow b\bar{b}$ simulated data and Higgs boson mass 120 GeV. The Cuts Based Analysis begins with $W$ reconstruction, a Leptonic $W$ is reconstructed using the lepton that caused the event to trigger and a neutrino solution calculated using missing transverse energy

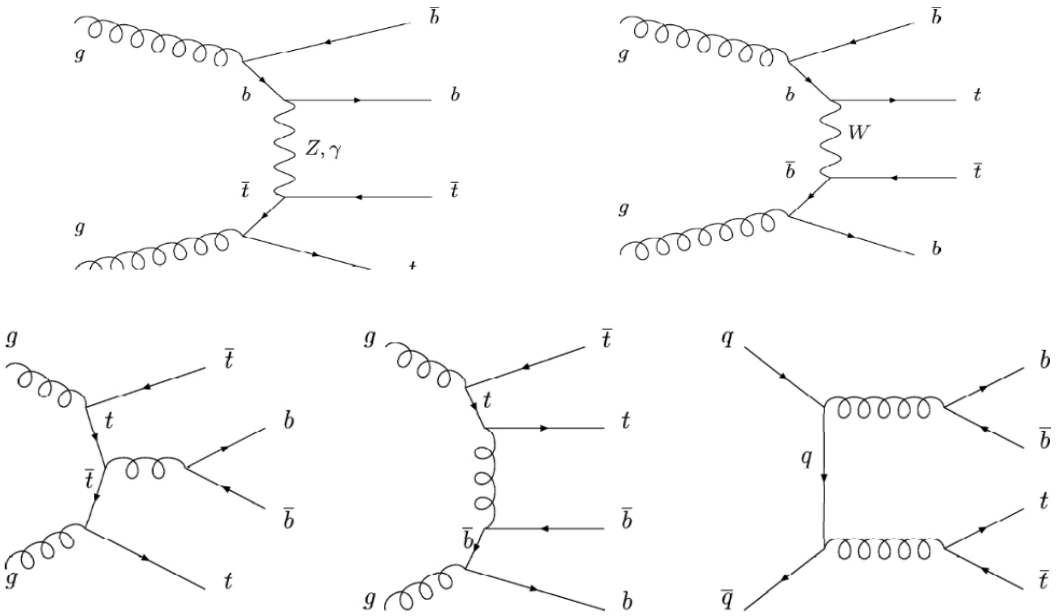



*Figure 10.3: Electroweak and QCD backgrounds*

in the event. A Hadronic $W$ is then reconstructed using the two jets in the event least likely to be $b$ jets based on $b$jetweights. A $W$ mass cut of $\pm 25$ GeV is applied and only events in which a Leptonic and Hadronic $W$ within the $W$ window can be reconstructed are kept. The top quarks are then reconstructed by pairing the four jets identified as $b$ jets with the $W$ solutions and minimizing a $\chi$ squared function so that a single solution is selected. A top mass cut is also applied, where top candidates must lie within a $\pm 25 GeV$ window of the top mass. After the tops are reconstructed, the remaining $b$ jets are used to create a Higgs boson solution.

$$\chi^2 = (\frac{m_{jjb} - m_t}{\sigma_{m_{jjb}}})^2 + (\frac{m_{lvb} - m_t}{\sigma_{m_{lvb}}})^2 \tag{10.3.1}$$

The Pairing Likelihood analysis is similar to the Cut-Based analysis method, with additional information about jet pairs to make the $W$ and $t$ solutions. The Pairing Likelihood method uses both the masses of the jets and the distance between them to create jet pair candidate solutions. The Constrained Fit analysis attempts to address jet

combinatorics in event reconstruction by using further available event information. Jet charge is used when creating jet pairs. Measured jet charges are used together with the charge of the lepton that acts as trigger for an event to assist in the assignment of jets to reconstructed $t$, $\bar{t}$ and $W$s. The Constrained fit used a sliding jet momentum scale, resulting in variation in neutrino energy. Hadronic $W$ and $t$ solutions were forced to be on a mass shell and energy rescaled with this in mind.

The findings of the $t\bar{t}H, H \rightarrow b\bar{b}$ CSC note are shown in table 10.1. It is clear that the $t\bar{t}H, H \rightarrow b\bar{b}$ will prove difficult to accurately detect. For an integrated luminosity of $30fb^{-1}$ calculated significance $\sigma$ for the Cut-Based analysis is 1.82, for the Pairing Likelihood it is 1.95 and for the Constrained Fit the significance is 2.18, where $\sigma$ is the number of signal events divided by the square root of the number of background events and is a measurement of the discovery potential of the channel. Each of these values could be significantly reduced by systematic uncertainties in the detector, although it is thought the significances can be improved with improvement in $b$ tagging methods and algorithms.

| Method | Significance |
|---|---|
| Cuts Based Analysis | 1.82 |
| Pairing Likelihood | 1.95 |
| Constrained Fit | 2.18 |

*Table 10.1: Significance results for the CSC studies based on 30 $fb^{-1}$ of $t\bar{t}H, H \rightarrow b\bar{b}$ Monte Carlo simulated data for a Higgs mass of 120 GeV*

## 10.4 Event Characteristics Collection One

The first set of event characteristics, *Higgs Input Variables*, identified as potentially useful for input to a neural network are

- $m_{bb}$ - Invariant mass of the $bb$ pair associated with the Higgs boson
- ${p_T}^H$ - Transverse momentum of the reconstructed Higgs boson
- $\cos\theta(b,b)$ - Cosine of the angle between the reconstructed Higgs in the lab frame and the nearest $b$ jet in $\Delta R$ in the rest frame of the Higgs
- $\Delta\eta(t,H)$ - Difference in $\eta$ between the reconstructed Higgs and the top quark nearest to the Higgs in $\Delta R$
- $\Delta\eta(b,b)$ - Difference in $\eta$ between the $b$ jets associated with the reconstructed Higgs
- ${m_{bb}}^1$ - Lowest invariant mass when two of all the possible $b$ jets associated with the Higgs are combined
- ${m_{bb}}^2$ - Second lowest invariant mass when two of all the possible $b$ jets associated with the Higgs are combined
- $\Delta\phi(t,t)$ - Difference in $\phi$ between the top quarks associated with the Higgs
- ${p_T}^t + {p_T}^t$ - Sum of $p_T$ of the top quarks associated with the Higgs

This collection of event characteristics are associated with a topology where the existence of a reconstructed Higgs boson is imposed on each event. Any event for which a Higgs boson cannot be reconstructed is not selected for use in this neural net. For each event there is a single solution for each event characteristic. In this way there are 9 input variables available to the neural network for each event in the Higgs set.

## 10.5 Event Characteristics Collection Two

The second set of event characteristics, *Generic Input Variables* identified as potentially useful for input to a neural network are

- $m_{b_{n1}b_{n2}}$ - Mass of all pair combinations of potential $b$ candidates in an event, all jets considered, ordered by $b$ jet weight, select top **6** values
- $p_T^{b_{n1}b_{n2}}$ - $p_T$ of all pair combinations of $b$ candidates in an event, all jets considered, ordered by $b$ jet weight, select top **6** values
- $E_T^{t_{n1}} + E_T^{t_{n2}}$ - Sum of $E_T$ for all pair combinations of potential $t$ candidates in an event, $t$ pairs ordered by $\chi^2$, select top **6** values
- $\Delta\phi(t_{n1}, t_{n2})$ - Difference in $\phi$ of all pair combinations of potential $t$ candidates in an event, $t$ pairs ordered by $\chi^2$, select top **6** values
- $\Delta\eta(t_{n1}, t_{n2})$ - Difference in $\eta$ between $t$ pair candidates in an event, $t$ pairs ordered by $\chi^2$, select top **6** values
- $b_n$ likelihood - Likelihood that a $b$ candidate is a $b$, ordered by $b$ jet weight, select top **6** values

This collection of event characteristics are associated with a topology where the existence of a reconstructed Higgs boson is not required in each event. For each event characteristic $n$ goes from 1 to $n_{high}$, where $n_{high}$ represents the highest integer index value assigned, so that all possible solutions of event variables indexed by $n$ are considered. Each event characteristic is then ordered by a criteria driven by attempts to create a vector with the most likely correct value for the event characteristic assigned position 1, the second most likely position 2 and so on. For event characteristics involving $b$ candidates, values are ordered using $b$ jet weights, and for event characteristics involving $t$ candidates, values are ordered using $\chi^2$, where $\chi^2$ is a statistical measurement of how well a measurement or measurements agree with experimentally known values [8]. $\chi^2$ is given

in equation 10.5.1 where $y_l$ are the values being assessed, $y$ is the known experimental result and $\sigma_l$ is the standard deviation of $y$.

In this case, for two $t$ candidates, $\chi^2$ is a measurement of how accurately we have assigned a pair of top quark candidates in a reconstructed event, based on the known top mass $m_t = 175GeV$. For two $t$ candidates, $t_1$ and $t_2$, with masses $m_{t_1}$ and $m_{t_2}$, $\chi^2$ is given by equation 10.5.2

$$\chi^2 = \sum_{l=1}^{N} \frac{(y_l - y)^2}{{\sigma_l}^2} \tag{10.5.1}$$

$$\chi^2 = (m_{t_1} - m_t)^2 + (m_{t_2} - m_t)^2 \tag{10.5.2}$$

For each event characteristic, the best 6 values are then selected for input into a neural network. In this way there are 36 input variables available for the neural network in the *Generic* set, compared with 9 input variables for the *Higgs* set.

Looser cuts are applied to the input Monte Carlo data sample used to create the input characteristic variables than in the creation of the first set of event characteristics. The preselection for the second set of event characteristics is adapted from that for the first set of event characteristics. The requirement for a jet to qualify as a $b$ jet is lowered and all jets in the event are considered as possible $b$ jets, ordered by the likelihood that a jet is a $b$ jet. $W$ mass cuts are applied for the first set of event characteristics but not for the second set. The preselection is described in detail in section 10.8. As a result of the looser preselection, more events are available as input to the neural network. So in summary, more statistics and more general event characteristics are then available as input to a neural network for the *Generic* set and the input variables are translated into a vector of potential values.

## 10.6 Datasets

The signal and background events used in this study are Monte Carlo events, simulated using the generators MC@NLO [62], AcerMC33, AcerMC34 [63] and Pythia 6.4 [64]. The signal datasets are simulated using Pythia 6.4, the $t\bar{t}jj$ datasets are simulated using MC@NLO, the $t\bar{t}b\bar{b}$ Electroweak datasets are simulated using AcerMC33/Pythia 6.4 and the $t\bar{t}b\bar{b}$ QCD datasets are simulated using AcerMC34/Pythia 6.4.

The datasets used are the same data as used in the $t\bar{t}H, H \rightarrow b\bar{b}$ CSC Analysis Note [61], in which the $t\bar{t}H, H \rightarrow b\bar{b}$ channel is studied for discovery potential at ATLAS using Cut-based and Likelihood Analysis methods.

The selection of a generator for simulation is based on ability to accurately model a particular process, for example AcerMC uses matrix elements rather than parton showers and therefore is thought to create more accurate $b$ jet momentum in each event. In the case of the $t\bar{t}b\bar{b}$, AcerMC is used to simulate the hard process and then Pythia is used to add extra jets using initial/final state radiation if needed.

The CSC Analysis Note [61] describes the simulated data in detail and the same samples are used in this neural network analysis. The information is summarized as follows. The Monte Carlo datasets use leading order cross-sections for the signal and $t\bar{t}b\bar{b}$ events and next-to-leading order cross-section simulations for $t\bar{t}jj$ background events. There were no next-to-leading order signal events available at the time of this analysis.

The signal sample events are generated using Higgs boson mass 120 GeV using the process $pp \rightarrow t\bar{t}HX \rightarrow l\nu bq\bar{q}b\bar{b}X$ where $l = e$ or $\mu$. Both signal and $t\bar{t}b\bar{b}$ background are generated requiring at least one lepton, electron or muon, of $|\eta| < 2.7$ and $p_T > 10$ GeV. The leading order production cross-section is $\sigma(t\bar{t}H) = 537$ fb and branching ratios $H \rightarrow b\bar{b}$ of 67.5%, $W \rightarrow l\nu$ of 10.66%, $W \rightarrow hadrons$ of 67.6% and lepton filter efficiency $\epsilon = 0.953$ are applied. The resulting cross section is 100 fb.

The $t\bar{t}b\bar{b}$ QCD and EW background sample events are generated using the process $gg \rightarrow t\bar{t}b\bar{b}X \rightarrow l\mu bq\bar{q}b\bar{b}X$ where $l = e$ or $\mu$. For $t\bar{t}b\bar{b}$ QCD AcerMC 3.4 is used and interfaced to PYTHIA 6.403 for the simulation of the initial and final state radiation,

hadronisation and decay. For $t\bar{t}b\bar{b}$ EW AcerMC 3.3 and PYTHIA 6.403 are used. The leading order $t\bar{t}b\bar{b}$ QCD cross-section is $\sigma(pp \rightarrow t\bar{t}b\bar{b}) = 8.2(gg)(+0.5(q\bar{q}))$ pb, lepton filter efficiency $\epsilon = 0.946$ and for $t\bar{t}b\bar{b}$ EW, the leading order cross-section is $\sigma(pp \rightarrow t\bar{t}b\bar{b}) = 0.90(gg)(+0.04(q\bar{q}))$ pb, lepton filter efficiency $\epsilon = 0.943$. The reducible $t\bar{t}jj$ background events are generated by MC@NLO 3.1 interfaced to HERWIG 6.510, [65], and Jimmy [66]. The process used is $pp \rightarrow t\bar{t} \rightarrow (l\nu q\bar{q})bq\bar{q}b$ where $l = e, \mu, \tau$ and for the inclusive $t\bar{t}$ cross-section a NLO+NLL calculation $\sigma(pp \rightarrow t\bar{t}) = 833$ pb. A filter is applied to the $t\bar{t}jj$ sample requiring that each event has an electron or muon of psuedorapidity $|\eta| < 2.7$ and transverse momentum $p_T > 14$ GeV and that the jets in the events are six in number, $p_T > 14$ GeV and $|\eta| < 5.2$ with four of $p_T > 14$ GeV and $|\eta| < 2.7$. The jets in the generated events are reconstructed using a fixed-cone algorithm with cone size $\Delta R = 0.4$, [67]. Within the $t\bar{t}jj$ sample, around 10% [61] of the events are $t\bar{t}b\bar{b}$ events and are removed using a method described in [68].

The cross-sections for each processes calculated using the generators, the number of events generated and the equivalent integrated luminosity are shown in table 10.2. All branching fractions and filter efficiencies are included.

| Process | $\sigma(fb)$ | Events | $L(fb^{-1})$ | Factorisation and Renormalisation Scales |
|---|---|---|---|---|
| $ttH$(LO) | 100 | 92750 | 931 | $Q^2 = {m_t}^2 + max({p_T}^2{}_t, {p_T}^2{}_t^-)$ |
| $t\bar{t}b\bar{b}$ QCD(LO) | 2371 | 98350 | 42 | $Q = \frac{m_H}{2} + m_t$ =235 GeV |
| $t\bar{t}b\bar{b}$ EW(LO) | 255 | 24750 | 97 | $Q\frac{m_H}{2} + m_t$ =235 GeV |
| $t\bar{t}$ filtered(NLO) | 109487 | 710321 | 6.5 | $Q^2 = m_t^2 + \frac{1}{2}({p_T}^2{}_t + {p_T}^2{}_t^-)$ |

*Table 10.2: Cross-sections, branching fraction, number of events, integrated luminosity and factorisation and normalisation scale for all the processes used in the CSC and this analysis for $t\bar{t}H, H \rightarrow b\bar{b}$, from [61]*

## 10.7 Event Preselection

In the analysis presented in the CSC note a set of event preselections are performed on the generated data. In the neural network analysis, for both sets of proposed event characteristics, a shared event preselection based on the preselection applied in the studies in the CSC note are performed on the Monte Carlo simulated datasets in order to select events that are useful to the neural network analysis. After the shared preselection, a further event selection specific to each set of event characteristics is performed.

The shared preselection requires that, in every event, a single lepton is present. The lepton is the trigger for the event. For the event, an electron or muon is accepted. The selection requires an electron $p_t \geq 25$, $|\eta| < 2.5$ or muon $p_t \geq 20$, $|\eta| < 2.5$. The event preselection also requires that and there are at least six jets in each event. The semileptonic signal for $t\bar{t}H, H \rightarrow b\bar{b}$ has high jet multiplicity. For the CSC analysis, four of the six jets are required to be $b$ jets. For the neural network analysis, for both *Higgs* and *Generic* variable sets, the preselection cuts are loosened and only two of the six jets are required to be tagged as $b$ jets. The threshold $b$jetweight used to tag a jet as a $b$ jet in the CSC analysis is 5.5.

| Preselection Conditions |
|---|
| An electron $p_t \geq 25$, $\|\eta\| < 2.5$ |
| OR |
| A muon $p_t \geq 20$ $\|\eta\| < 2.5$ |
| AND |
| At least six jets in each of the event |
| WHERE |
| At least four/two of the jets are tagged as $b$ jets |

*Table 10.3: Preselection for $t\bar{t}H, H \rightarrow b\bar{b}$ used in the CSC and neural network analysis, the distinction between the two being the number of jets required to be tagged as b jets, for the CSC analysis it is 4 and for the neural network only 2 are required*

## 10.8 Selection and Reconstruction

### 10.8.1 Higgs Input Variables

The event selection and reconstruction method for creation of the first *Higgs* set of input variables follows the shared preselection, so that each event has at least one lepton, six jets, two of which are $b$ jets.

The reconstruction then takes place as follows. Neutrino solutions are first calculated using missing transverse energy, $W$ Leptonic solutions are created using a neutrino and the lepton in the event. Hadronic $W$ solutions are then reconstructed using combinations of all jets other than four most likely to be $b$ jets. A $W$ mass cut is applied so that only $W$ candidates within $\pm 25$ GeV of the true $W$ mass are accepted. There is required to be at least one Leptonic $W$ and at least one Hadronic $W$ in an event, otherwise an event is not selected for further reconstruction and selection. Leptonic $t$ solutions and hadronic $t$ solutions are created using the reconstructed $W$ candidates, a top mass window is applied so that only $t$ candidates within $\pm 25$ GeV of the true top mass are accepted. Top pairs are then created. The top pair with a $\chi^2$ value suggesting the best top pair has been identified is then selected and from this the Leptonic $W$, Leptonic $t$, Hadronic $W$ and Hadronic $t$ are assigned. Each event then has *one* top combination, *one* leptonic $W$, *one* leptonic $t$, *one* hadronic $W$ and *one* hadronic top, with light jets assigned to the Hadronic $W$. A Higgs candidate is then created from the left over $b$ jets. Each event therefore has a reconstructed Higgs. The first set of event characteristics are then gathered.

### 10.8.2 Generic Input Variables

The event selection and reconstruction method for creation of the second *Generic* set of input variables follows the shared preselection, so that each event has at least one lepton, six jets, two of which are $b$ jets. However, for the *Generic* event characteristic analysis, all jets in an event are considered as $b$ jet candidates.

The reconstruction then takes place as follows. Again neutrino solutions are calculated

using missing transverse energy, leptonic and hadronic top candidates are created using neutrino solutions, the event lepton and all available jets in the event. No $W$ mass cut is applied. An ordered jet collection is then created, where jets are ordered by $b$ jet weight. Jets are then combined into pairs and the mass and $p_T$ of each pair is calculated. Top pairs are then created using leptonic $W$, hadronic $W$ and all jet solutions. Again a top mass window is applied, so that only $t$ candidates within $\pm 25$ GeV of the true top mass are accepted. All remaining top pairs are then ordered by $\chi^2$. An $E_T$ sum, $\phi$ difference and $\eta$ difference is calculated for all top pairs. In this way the second set of event characteristics are gathered.

## 10.9 Neural Network Analysis

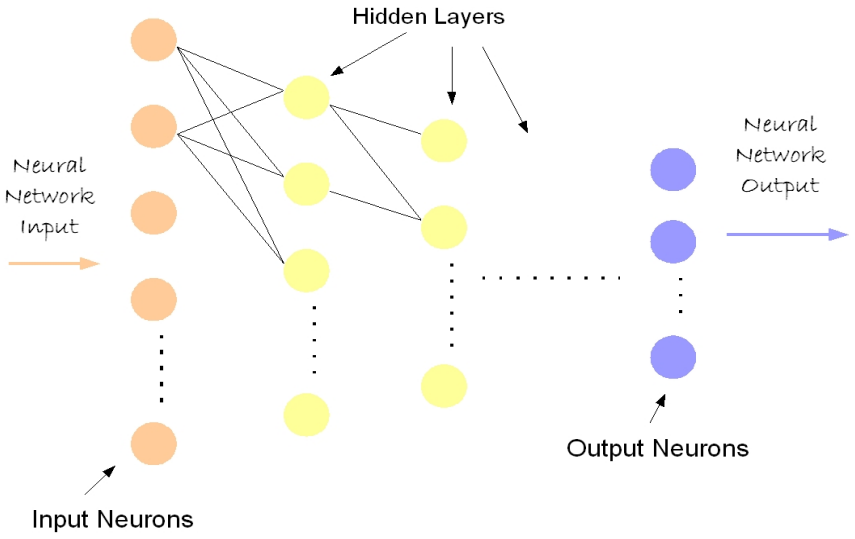



*Figure 10.4: neural network showing Input, Output and Hidden Layers*

A neural network is an analysis tool used to identify patterns in data. In the $t\bar{t}H, H \rightarrow b\bar{b}$ analysis, we use a neural network to attempt to improve the significance and sensitivity of the channel by developing a neural network to identify signal from background events. A MultiLayer Perceptron in used and the layout of a neural network is shown in figure 10.4.

Sensitivity allows us to exclude, in a particular mass region and at 95% confidence

level, the existence of a Higgs boson if it were to be produced at some multiple times the standard model prediction. Exclusion is first achieved in a mass region where the sensitivity of an experiment is at its highest. A 95% confidence level exclusion means that there is only a 5% chance that the observation has been mislabelled as background when it is signal

The sensitivity results from the neural network analysis in this study describe how much the standard model cross-section needs to be scaled by in order to achieve exclusion. When sensitivity = 1, the standard model cross-section is excluded. For sensitivity $> 1$ we have excluded, at 95% confidence level, the given multiple of the standard model cross-section. For sensitivity $< 1$ we have exclusion above 95% confidence level and there is no doubt of exclusion. Sensitivity therefore gives a measure of the usefulness of a neural network analysis.

The inputs to the analysis are the *Higgs* event characteristics and the *Generic* event characteristics. The output is an assessment of the neural networks interpretation of an event, as signal or background.

Performance of the neural net is assessed by noting a sensitivity measurement produced by the analysis output. Events are weighted according to the relative cross-sections used in the CSC analysis to reflect realistic proportions of signal and background events within the simulated data sample. Among the ATLAS systematics included in the neural net analysis are Jet Energy Scale and $b$ tagging efficiency.

### 10.9.1 Learning

The analysis requires that the neural network is first trained and tested using a data sample, where events are already identified as signal or background. Events are assigned an event type flag, 1 for signal and 0 for background. The data is split into two parts, a part for training and a part for testing, and both are passed into the neural network for assessment. Events in the training sample are used to develop pattern recognition and event recognition strategies, events in the testing sample are used to improve and assess

the performance of the neural network. The process is known collectively as *Learning*. Learning takes place over a number of cycles, or the number of times data is passed through the network. Too many cycles results in over learning, where the network recognises individual events in place of event patterns, too few cycles and the network will not develop optimal identification patterns.

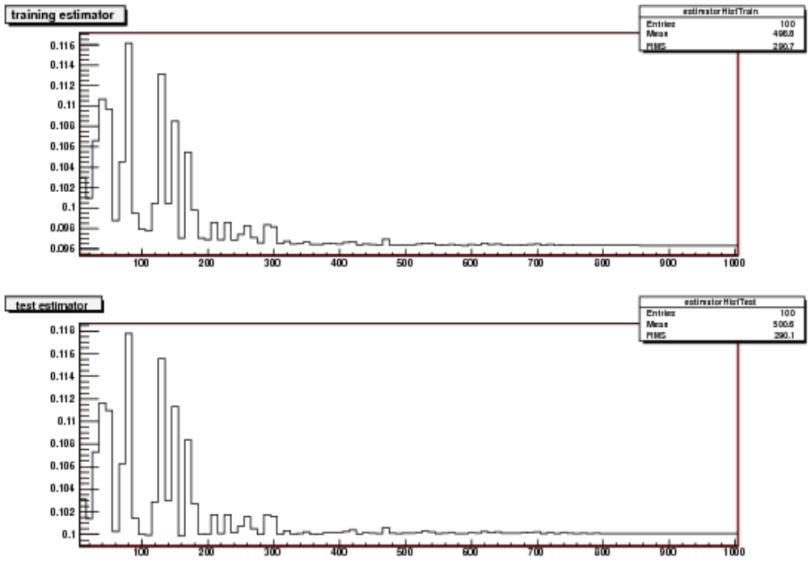



*Figure 10.5: Training and Testing estimators, convergence after 700 cycles*

Figure 10.5 shows the training and testing process. The neural network improves its ability to differentiate signal and background events as the error reduces. Learning is stopped after 1000 cycles, as convergence of the error value is seen at around 700 cycles, signifying that the network has learned as much as is possible from the input data sample and many more cycles may lead to over learning in the network.

### 10.9.2 Layout

The neural network layout can be varied in terms of the number of hidden layers and number of nodes in each layer. The neural network used in the analysis has the layout $36:8:4:1$, two hidden layers, the first with 8 nodes and the second with 4.

### 10.9.3 Selection of Event Characteristic Input Collection

In order to determine which of the Event Characteristic Collections allow a better performing neural network, we compare the output of each analysis method. The analysis outputs a value for sensitivity of the $t\bar{t}H, H \rightarrow b\bar{b}$ channel when a neural network analysis is performed. Table 10.4 show the sensitivity results for a neural network using the *Generic* and *Higgs* event characteristic collections for integrated luminosity $1fb^{-1}$. The Generic set give a better sensitivity output, 4.74, compared to the Higgs set, 8.69, showing that the *Generic* Event Characteristic set, when used in a neural network analysis, out-perform the *Higgs* event Characteristic set by a factor of 2.

The analysis output also includes an assessment of the neural network ability to separate signal and background events. Figure 10.6 shows the signal and background separation of events performed by the neural network for the *Generic* set. Input events in the testing sample are assigned a value between 0 and 1 based on the previous learning of the network on the training event sample. Signal events are coloured in blue, background in white. The plot shows a clear and distinct separation of events and demonstrates the ability of a trained neural network to distinguish signal events from background. Figure 10.7 shows signal efficiency against background event rejection for the *Generic* set. As signal efficiency increases, background rejection decreases.

Figure 10.8 shows the separation of signal and background events for a neural network that uses a *Higgs* variable set. Figure 10.9 shows the signal efficiency against background event rejection for the *Higgs* set.

Comparing the separation and efficiency figures shows that while the *Generic* variables allow for a clear separation of signal and background events and a smooth curve for signal efficiency against background rejection, an analysis that uses the *Higgs* variable set produces a less distinct separation of signal and background events and a less smooth curve for signal efficiency against background rejection.

Table 10.5 shows the number of events after shared preselection and event characteristic selection and reconstruction.

| Variables | Sensitivity |
|---|---|
| Generic | 4.74 |
| Higgs | 8.69 |

*Table 10.4: Sensitivity for $t\bar{t}H, H \rightarrow b\bar{b}$ for integrated luminosity $1fb^{-1}$ using a neural net with Generic and Higgs Event Characteristics*

| Data | Events Preselected | Events Generated |
|---|---|---|
| $t\bar{t}H, H \rightarrow b\bar{b}$ | 13938 | 92750 |
| $t\bar{t}b\bar{b}$(QCD) | 7651 | 98350 |
| $t\bar{t}b\bar{b}$(EW) | 2123 | 24750 |
| $t\bar{t}j\bar{j}$ | 23831 | 710321 |

*Table 10.5: Number of events in analysis, after preselection and generated, for $t\bar{t}H, H \rightarrow b\bar{b}$ signal and $t\bar{t}b\bar{b}$, $t\bar{t}b\bar{b}$ and $t\bar{t}j\bar{j}$ backgrounds*

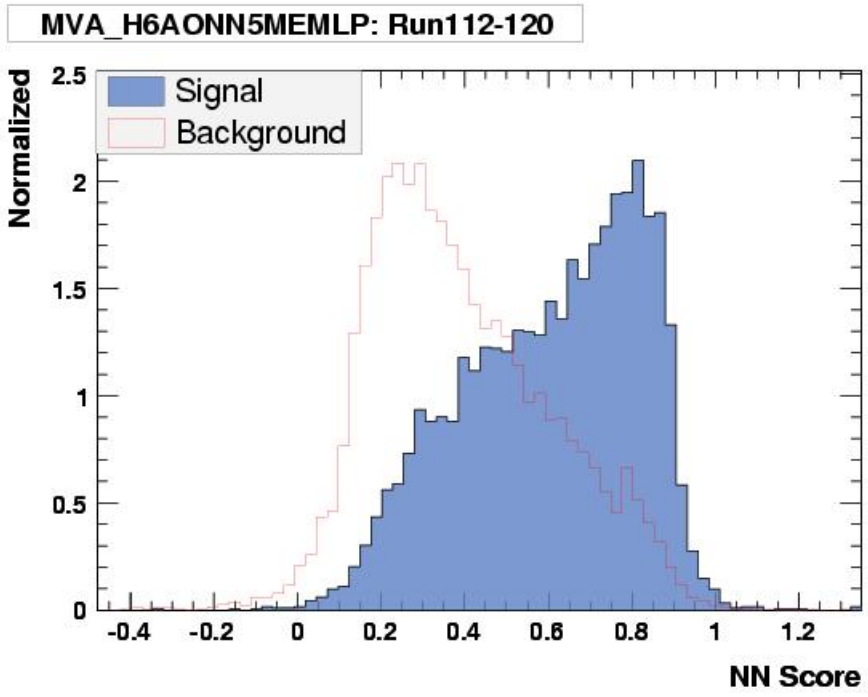



*Figure 10.6: Signal and Background Separation for Generic Event Characteristics*

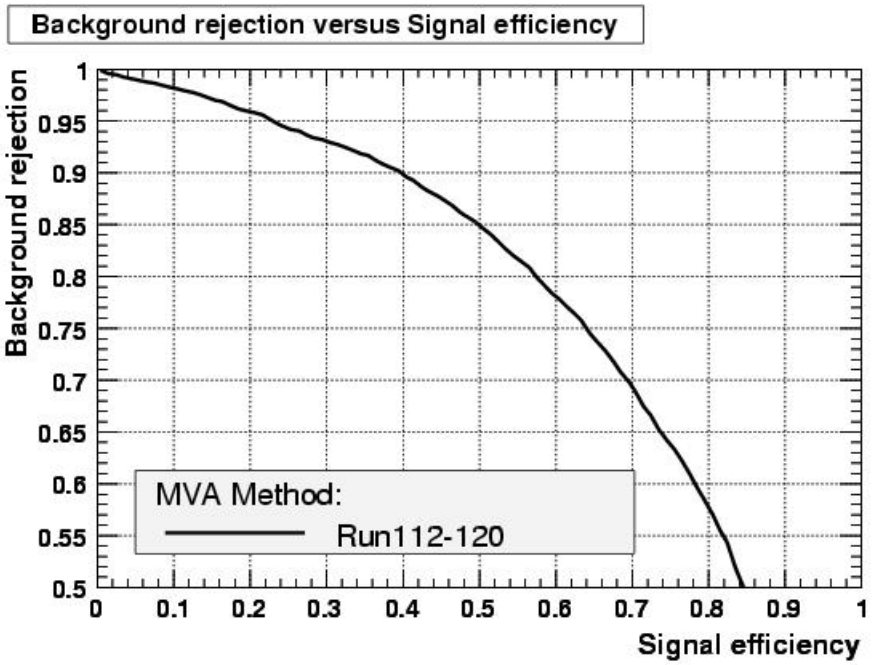



*Figure 10.7: Background Rejection to Signal Efficiency for Generic Event Characteristics*

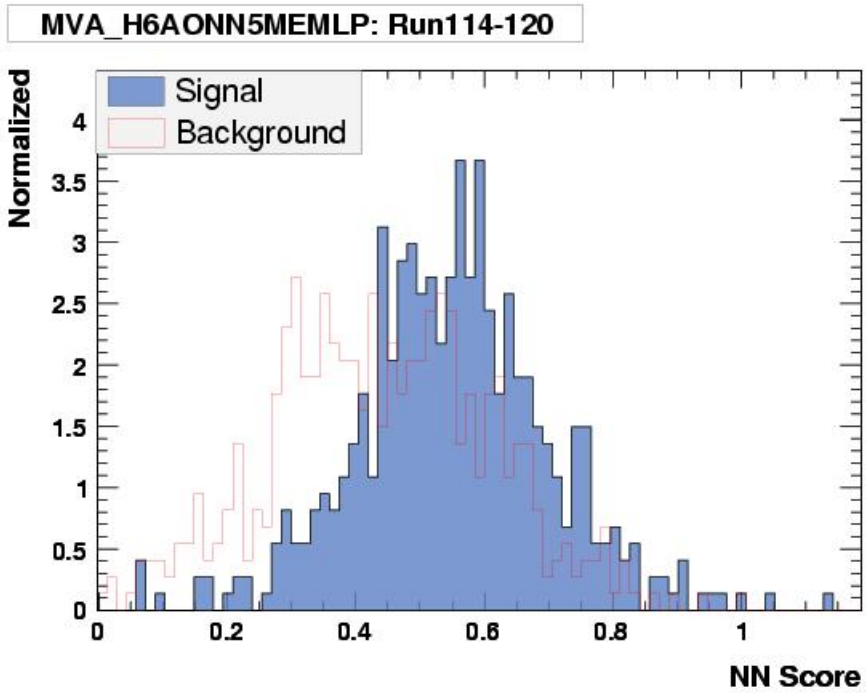



*Figure 10.8: Signal and Background Separation for Higgs Event Characteristics*

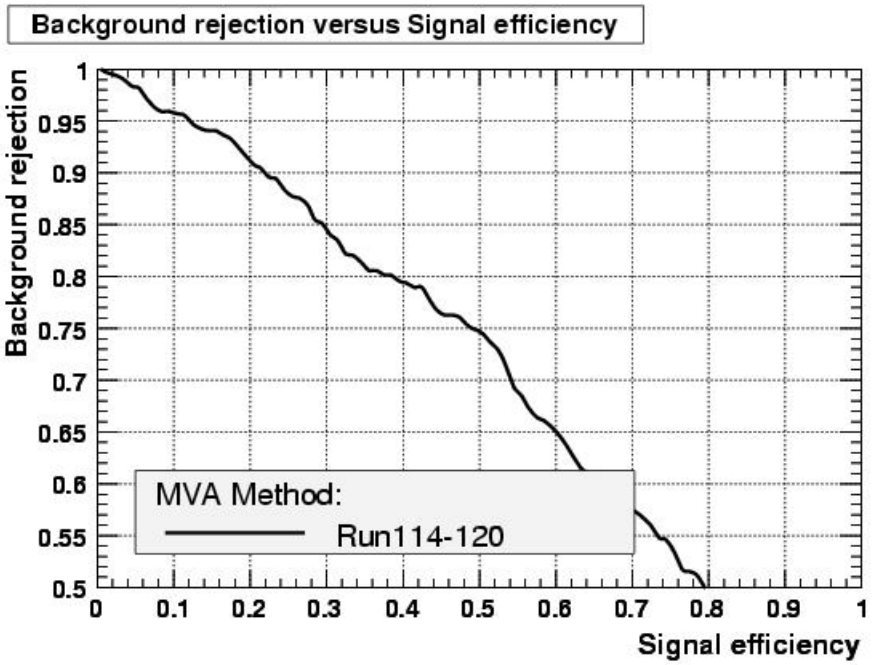



*Figure 10.9: Background Rejection to Signal Efficiency for Higgs Event Characteristics*

### 10.9.4 Variable Distributions

Figures 10.10 and 10.11, 10.12 and 10.13 show the distribution of the input variables for the *Higgs* and *Generic* set respectively, showing both signal and background events. There are differences in the distributions for signal and background events. Alone, these differences would not allow for a powerful method of event separation as the differences are small. The neural network analysis combines small differences and the correlations between variables to create a more powerful method of signal and background event separation.

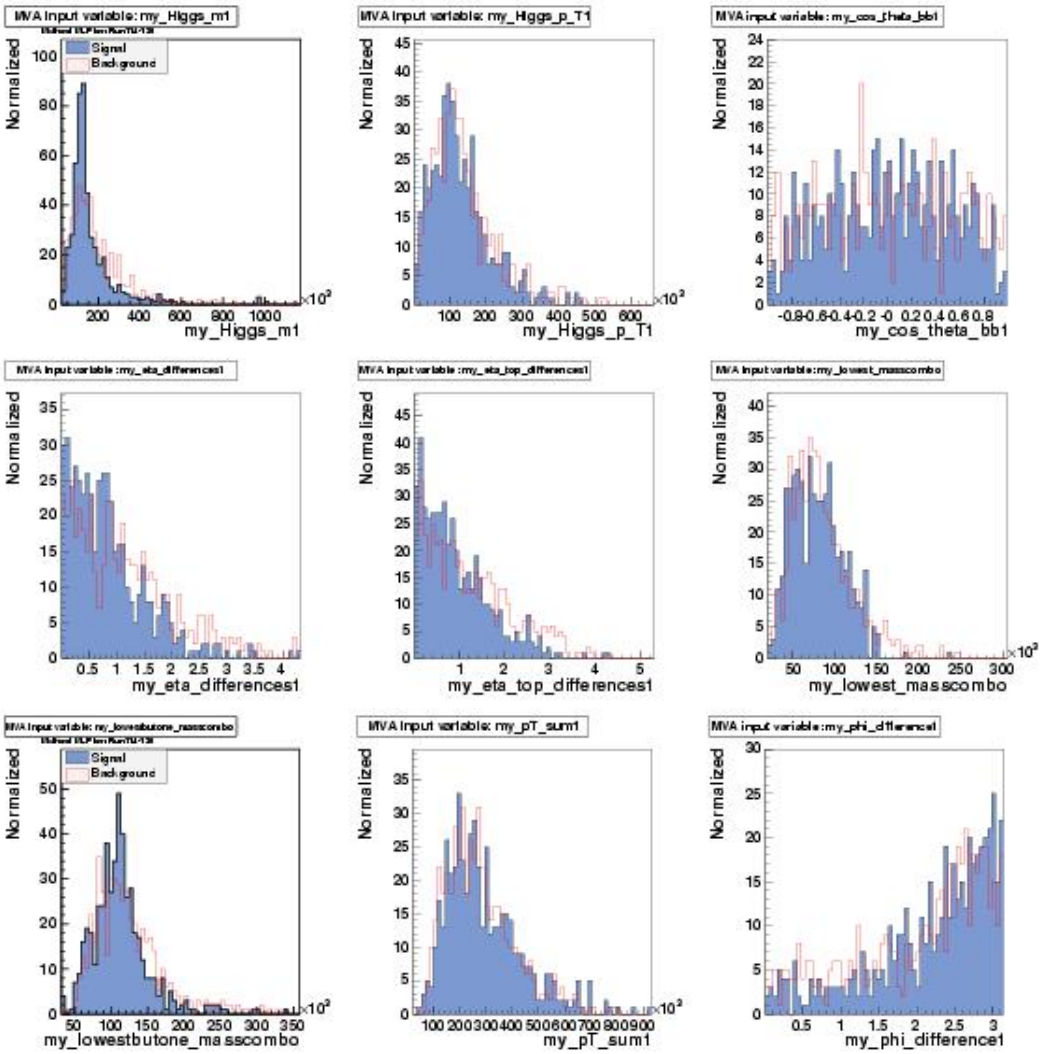



*Figure 10.10: Distributions for the Higgs Event Characteristics*

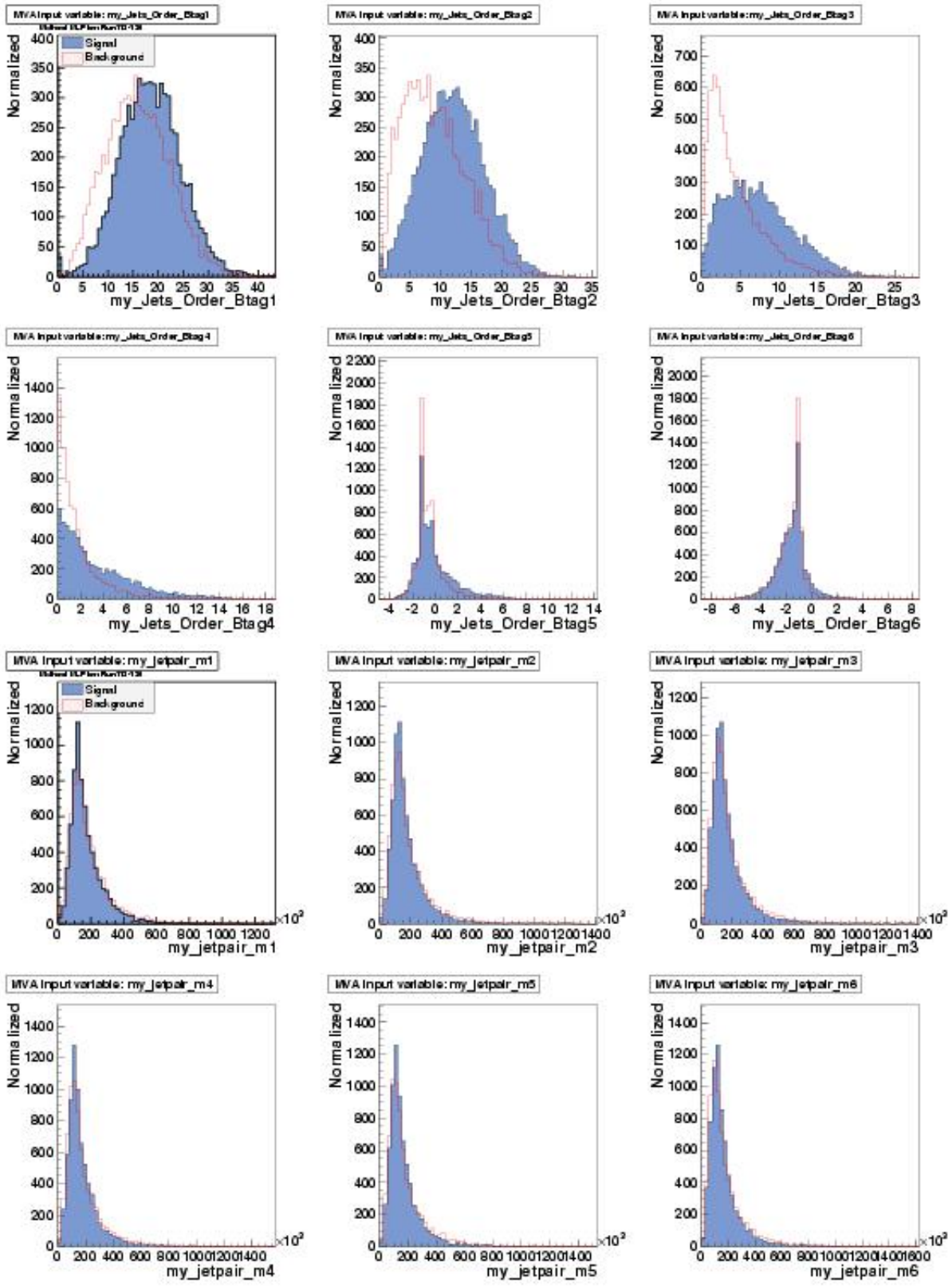



*Figure 10.11: Distributions for the Generic Event Characteristics, bLikelihood (6) and $m_{bb}$ (6)*

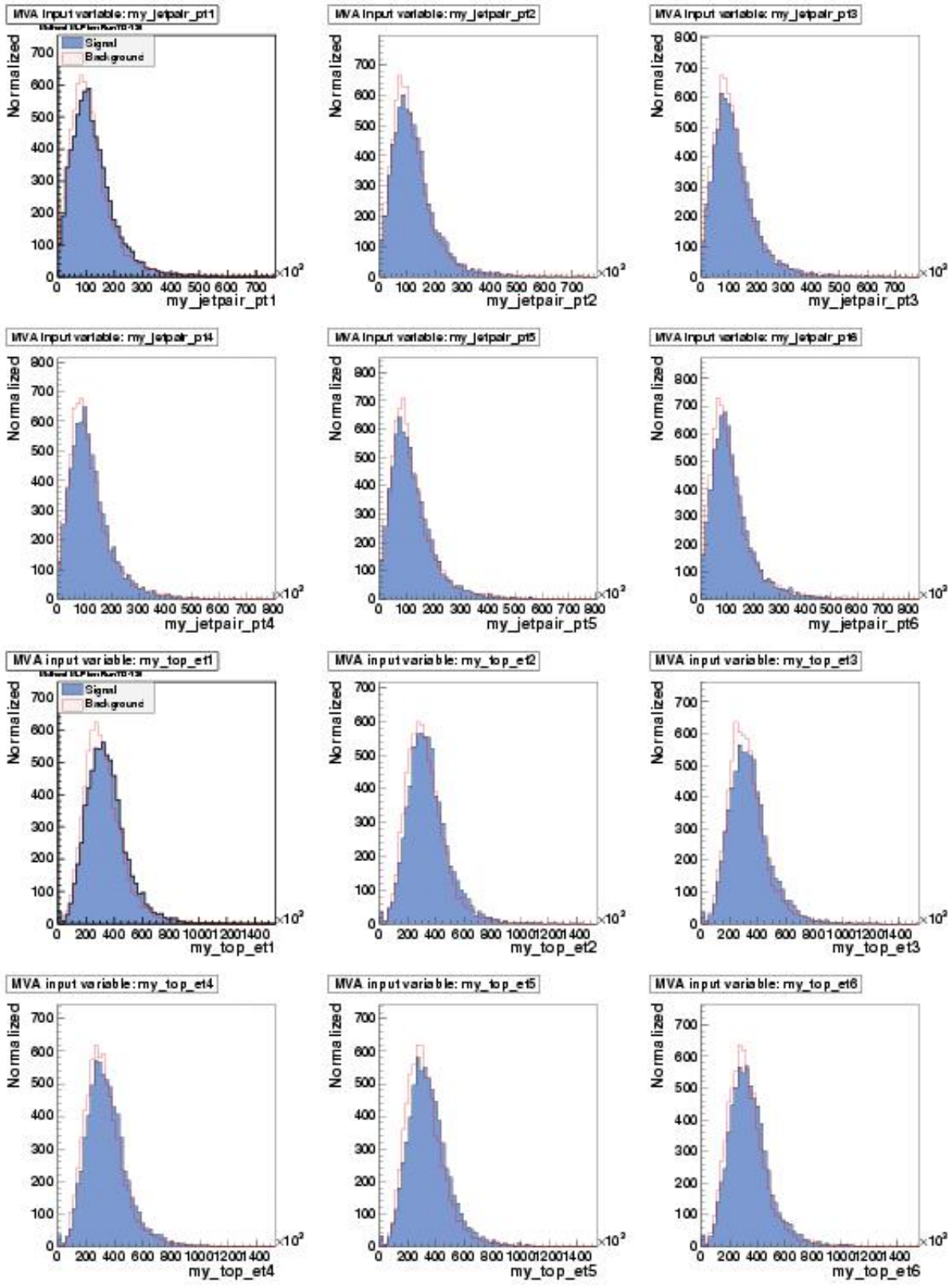



*Figure 10.12: Distributions for the Generic Event Characteristics, $p_T^{bb}$ (6) and $\Delta\eta(t,t)$ (6)*

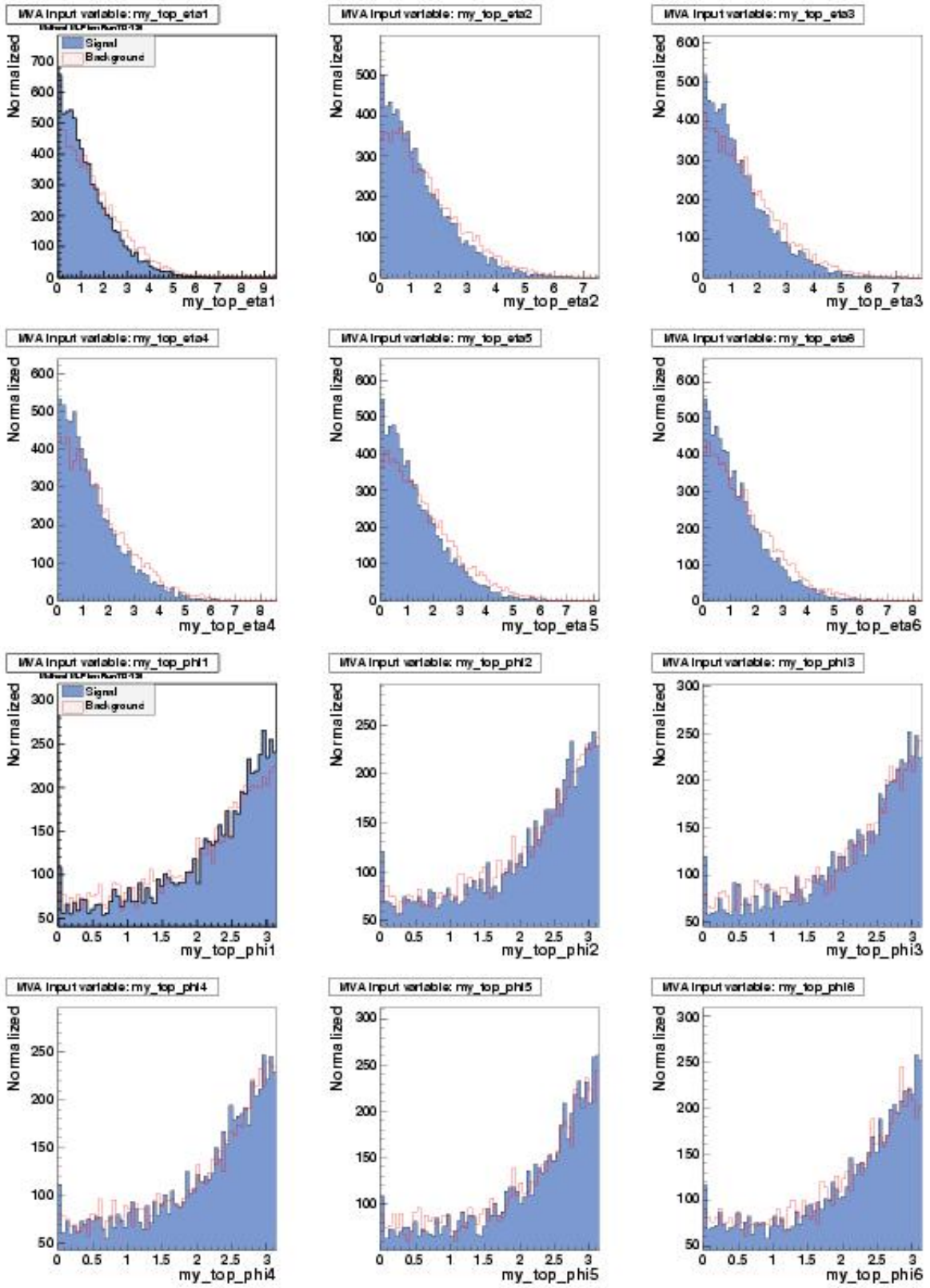



*Figure 10.13: Distributions for the Generic Event Characteristics, $\Delta\eta(t,t)$ (6) and $\phi_t^1 - \phi_t^2$ (6)*

### 10.9.5 Importance of Variables

The neural network analysis assesses the performance and usefulness of each event characteristic to the analysis. Once learning is complete, the analysis outputs a list of input event characteristics in order of usefulness and importance to analysis. Each event characteristic is given a rank. Table 10.6 show the input variables used by a $t\bar{t}H, H \rightarrow b\bar{b}$ *Generic* event characteristics neural network. Figure 10.14 shows each variable rank plotted against the natural logarithm of importance. A separation of the *Generic* variable set is seen, with the first half of the event characteristics in order of rank seen as being of significantly more importance to the analysis.

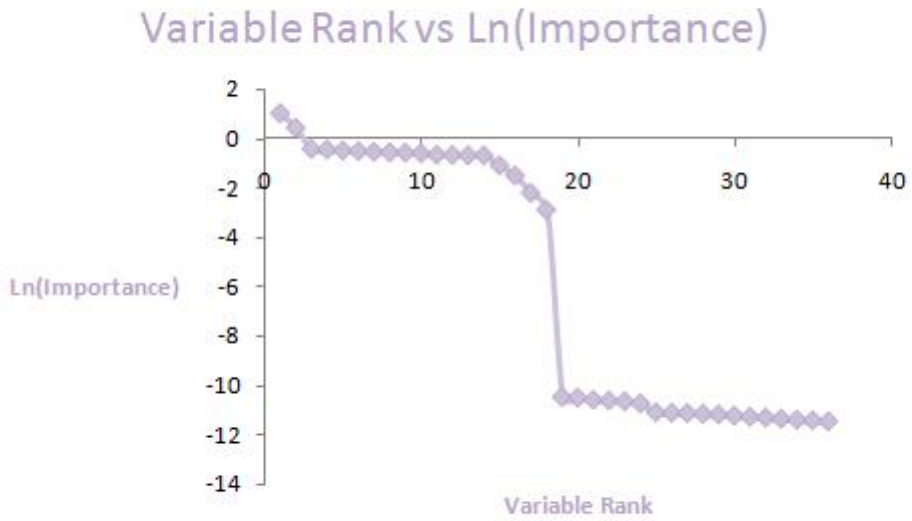



*Figure 10.14: Variable Rank vs Ln Importance for Generic Event Characteristics*

| Rank | Input Variable | Importance |
|---|---|---|
| 1 | $bLikelihood_6$ | 1.032e+01 |
| 2 | $bLikelihood_5$ | 2.666e+00 |
| 3 | $\Delta\eta(t,t)_2$ | 3.698e-01 |
| 4 | $\Delta\eta(t,t)_4$ | 3.469e-01 |
| 5 | $\Delta\eta(t,t)_6$ | 3.191e-01 |
| 6 | $\Delta\eta(t,t)_3$ | 3.032e-01 |
| 7 | $\Delta\eta(t,t)_5$ | 2.835e-01 |
| 8 | $\Delta\phi(t,t)_6$ | 2.656e-01 |
| 9 | $\Delta\phi(t,t)_2$ | 2.623e-01 |
| 10 | $\Delta\eta(t,t)_1$ | 2.506e-01 |
| 11 | $\Delta\phi(t,t)_4$ | 2.144e-01 |
| 12 | $\Delta\phi(t,t)_3$ | 2.108e-01 |
| 13 | $\Delta\phi(t,t)_5$ | 2.045e-01 |
| 14 | $\Delta\phi(t,t)_1$ | 2.042e-01 |
| 15 | $bLikelihood_4$ | 7.991e-02 |
| 16 | $bLikelihood_3$ | 3.100e-02 |
| 17 | $bLikelihood_2$ | 5.986e-03 |
| 18 | $bLikelihood_1$ | 1.259e-03 |
| 19 | $p_{T\,2}^{bb}$ | 3.249e-11 |
| 20 | $p_{T\,6}^{bb}$ | 3.019e-11 |
| 21 | $p_{T\,1}^{bb}$ | 2.466e-11 |
| 22 | $p_{T\,3}^{bb}$ | 2.321e-11 |
| 23 | $p_{T\,4}^{bb}$ | 2.161e-11 |
| 24 | $p_{T\,5}^{bb}$ | 1.772e-11 |
| 25 | $E_T^t + E_{T\,2}^t$ | 7.645e-12 |
| 26 | $E_T^t + E_{T\,1}^t$ | 7.356e-12 |
| 27 | $E_T^t + E_{T\,5}^t$ | 7.180e-12 |
| 28 | $m_{bb1}$ | 6.704e-12 |
| 29 | $E_T^t + E_{T\,5}^t$ | 6.350e-12 |
| 30 | $E_T^t + E_{T\,3}^t$ | 5.682e-12 |
| 31 | $E_T^t + E_{T\,4}^t$ | 5.055e-12 |
| 32 | $m_{bb4}$ | 4.724e-12 |
| 33 | $m_{bb2}$ | 4.223e-12 |
| 34 | $m_{bb3}$ | 3.848e-12 |
| 35 | $m_{bb5}$ | 3.672e-12 |
| 36 | $m_{bb6}$ | 3.285e-12 |

*Table 10.6: Input Variables for a $t\bar{t}H, H \to b\bar{b}$ after training, ordered by rank*

When *Generic* Event Characteristics are listed by Importance to a neural network, we can make the following observations

- Variables are separated into two clear sets, the first 18 are of high importance, the second 18 of low importance
- In the variables of high importance, we see all the values for $bLikelihood$, $\Delta\phi(t,t)$ and $\Delta\eta(t,t)$
- In the variables of low importance, we see all the values for $p_T^{bb}$, $E_T^t + E_T^t$ and $m_{bb}$
- The variables of most importance are $bLikelihood$ of the 5th and 6th $b$ jet candidates, that is, the $b$ jets with the lowest likelihood of being $b$ jets

The importances of the event characteristics are assigned by the analysis based on the usefulness of a variable in recognising signal from background events. The separation of variables into two sets in terms of importance may suggest that there are correlations between the variables of high importance and those of low importance that result in duplicate information being passed to the analysis so that variables which provide largely repeated information are given low importance.

### 10.9.6 Results

Table 10.7 shows the resulting sensitivities for a neural network and the sensitivity of the channel in the cuts based analysis. The neural network analysis uses the *Generic* Event Characteristic collection, a neural network of layout $36:8:4:1$, 1000 learning cycles and 13938 $t\bar{t}H, H \rightarrow b\bar{b}$ signal events, 7651, 2123 and 23831 background events for $t\bar{t}b\bar{b}$(QCD), $t\bar{t}b\bar{b}$(EW) and $t\bar{t}j\bar{j}$, for $1fb^{-1}$ integrated luminosity, where events are weighted according to the relative cross-sections used in the CSC analysis to reflect realistic proportions of signal and background events within the simulated data sample. The neural network analysis method as described in this analysis improves the sensitivity of the channel from that of the Cuts-Based Analysis performed in the CSC analysis, where sensitivity of the channel is 14[1], to a sensitivity of 8.69 when *Higgs* event characteristics are used and 4.74 when *Generic* event characteristics are used.

| Method | Sensitivity |
|---|---|
| Neural Net Analysis *Generic* Inputs | 4.74 |
| Neural Net Analysis *Higgs* Inputs | 8.69 |
| Cuts Based Analysis | 14.65 |

*Table 10.7: Sensitivity for $t\bar{t}H, H \rightarrow b\bar{b}$ using a Neural Net and Cuts-Based Analysis*

Figure 10.15 shows the neural network output for all $t\bar{t}H, H \rightarrow b\bar{b}$ signal and background processes, the inset shows the area around the concentration of signal events. We see a clear separation of $t\bar{t}H, H \rightarrow b\bar{b}$ signal and background events.

[1]Catherine Wright, Private Communication

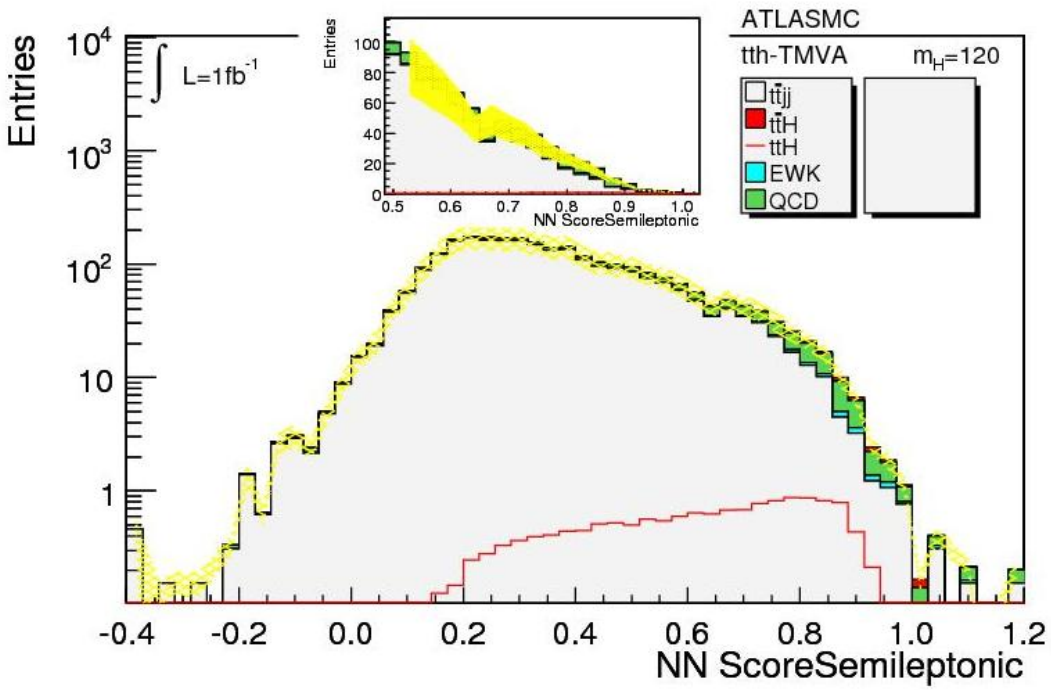



*Figure 10.15: neural network output for all $t\bar{t}H, H \rightarrow b\bar{b}$ signal and background processes, the inset shows the area around the concentration of signal events, for a neural network of Generic Event Characteristic collection inputs, with* 36 *Generic input variables, a neural network of layout* $36:8:4:1$, 1000 *learning cycles for integrated luminosity of* $1fb^{-1}$, *giving an output sensitivity of* 4.74

## 10.10 Conclusions and Future Directions

In this chapter, signal and background events for the $t\bar{t}H, H \rightarrow b\bar{b}$ channel are introduced as a way to identify and understand potential distinguishing variables for use in a neural network analysis. The neural network method can give an improvement in sensitivity for the channel $t\bar{t}H, H \rightarrow b\bar{b}$. A new set of event variables, intended to allow recognition of signal and background events, are defined. This *Generic* set of variables are associated with a topology where the existence of a reconstructed Higgs boson is not required in each event. Event Characteristic input values are translated into a vector of potential values, allowing the neural network access to more information per event. The analysis establishes that the *Generic* set of variables, where Event Characteristics are treated as multiple potential values, has the best potential for improving the sensitivity of the channel.

The neural network developed in the analysis is a neural network of *Generic* Event Characteristic collection inputs, layout $36 : 8 : 4 : 1$, 1000 learning cycles for $t\bar{t}H, H \rightarrow b\bar{b}$ signal and corresponding background events. The analysis gives an output sensitivity of 4.74 for the channel, an improvement on the sensitivity value of the most recent Cuts-Based analysis of the CSC analysis.

Further study of the input variables used in this analysis and the correlations between these variables is a useful direction for further study. Equally, there may be further distinguishing variables of potential use to a neural network, this possibility also merits further study.

The analysis in this chapter has served as a proof of concept for the neural network analysis method for $t\bar{t}H, H \rightarrow b\bar{b}$. The analysis has also used the CDF analysis tools that used to exclude the existence of a Higgs boson in the mass range $160 - 170GeV$, for simulated ATLAS data, demonstrating the use of CDF analysis methods in an ATLAS context. The analysis has demonstrated that the neural network method can give a notable improvement in sensitivity for the $t\bar{t}H, H \rightarrow b\bar{b}$ channel.

# Chapter 11

# Conclusions

In this thesis, research, development studies and results in the development of an Event Level Metadata Analysis software system, a TAG Database, for the ATLAS experiment, part of the LHC collaboration at CERN in Geneva, have been presented, as well as a physics analysis of the Higgs boson channel $t\bar{t}H, H \rightarrow b\bar{b}$ for Higgs mass $m_H = 120$ GeV for a Neural Network analysis.

The Event Level Metadata system research has been presented in terms of three studies, these are Feasability, Scalability and Accessibility.

Feasibility studies are the initial steps in the development of an Event Level Metadata system. The studies demonstrated that an Event Level Metadata system can operate within the larger ATLAS software system and gathered information on the implications for Event Level Metadata system development. Specifically, interactions with the ATLAS Distributed Data Management system and the ATLAS Trigger system are studied as these are the software systems with which the Event Level Metadata system must closely interface. For merging the Event Level Metadata system with the Distributed Data Management system, central challenges are implementing the dataset concept used in the Distributed Data Management system with the file concept used in the Event Level Metadata system. This study develops workable solutions and in doing so demonstrates the feasibility of implementing a bridge between systems. Solutions are adding a dataset attribute to Event Level Metadata, studying the impact of implementing file lookup within

the DQ2 system and demonstrating that the impact is acceptable and managable, and by developing an optimising a method to return Event Level Metadata query output to users. For merging the Event Level Metadata system with the Trigger system, the challenge centers on implementing time varying Trigger menus with Event Level Metadata in relational tables. We propose a relational solution to implement time varying Trigger menus in the Event Level Metadata Interface.

Scalability studies follow proof of feasibility. We present studies on implementation and performance of a realistic terabyte scale relational TAG Database and demonstrate that an Event Level Metadata system at terabyte scale is achievable. A database schema and an indexing and partitioning strategy for a relational TAG database and a strategy for upload of and access to data is developed and presented. Performance of a terabyte scale relational database assessed. In these studies we investigated strategies for organisation of data within a relational stucture so that data can be both written and read in a useful and realistic way, and give a performance assessement of a realistic terabyte scale relational TAG Database.

A query in which a user counts events returned by a query is returned in an order of seconds, 100000 events are counted in 4.5 seconds, 400000 events are counted in 14 seconds. This can be considered an online respone, a considerable advantage to an analysis system and an attractive and useful feature for analysts. Performance time can be predicted within some bounds as the increase of performance time with data queried is seen to be linear. A query in which a user returns events for analysis are again in the order of seconds. 100000 events are selected in 10 seconds, 400000 events are selected in 50 seconds. The increase in time with events selected is linear, so performance can be predicted.

Accessibilty studies present the development of a web interface to the Event Level Metadata system. The ELSSI interface is intended to manage the complexity of the relational system and present the user with an intuitive and useful means of creating an Event Level Metadata query and return a result set. ELSSI adopts the concepts of query patterns developed in the Scalability studies and interacts with the Distributed Data

Management and Trigger systems in the ways developed and presented in the Feasibility Studies.

The research studies in this thesis have therefore demonstrated that an Event Level Metadata can be integrated with the ATLAS software system and developed solutions for integration, proven that an Event Level Metadata relational database can scale to ATLAS terabyte size and presented performance results for a realistic ATLAS scale system, and developed a user interface to the Event Level Metadata system.

As real data is collected from ATLAS, there will be a need to calibrate and align this data for analysis, through detector specific calibration and quality studies. The real data will then be compared with the Monte Carlo Truth data. The TAG Database described in this thesis will allow event selection using the newly calibrated detector measurements, allowing events to be identified for both real and truth data independant of the Monte Carlo truth information and for the relative efficiency of trigger selections to be calculated.

In the physics analysis of this thesis, a neural network method is developed for the channel $t\bar{t}H, H \rightarrow b\bar{b}$ for a Higgs mass of $m_H = 120$ GeV. The analysis shows that a neural network method can give an improvement in sensitivity for the channel. A set of event variables, associated with a topology where the existence of a reconstructed Higgs boson is not required in each event are defined and it is demonstrated that these variables when used in a neural network can improve the sensitivity of the channel by improving separation of signal and background events. The neural network analysis uses the *Generic* Event Characteristic collection, a neural network of layout $36 : 8 : 4 : 1$, 1000 learning cycles and 734033 $t\bar{t}H, H \rightarrow b\bar{b}$ signal and background events for an integrated luminosity of $1fb^{-1}$ to give an output sensitivity of 4.74. We see that the neural network analysis method as described in this analysis improves the sensitivity of the channel from that of the Cuts-Based Analysis performed in the CSC analysis, where sensitivity of the channel is 10. Sensitivity of the channel is then improved from the from the value in recent analysis, from 10.0 to 4.74, demonstrating that the Neural Network analysis method can give a notable improvement in sensitivity for the $t\bar{t}H, H \rightarrow b\bar{b}$ channel.

# Appendix A

# Analysis using Artifical Neural Nets

## A.1 Neural Net

An Artificial Neural Network, ANN, is an analysis tool modelled on biological neural network systems. Neural networks consist of a series of neurons, connected in a system of layers. An example of a simple neural network is shown in figure A.1. A neural network has multiple input neurons, multiple hidden layers and multiple outputs. Neurons are interconnected through synapses.

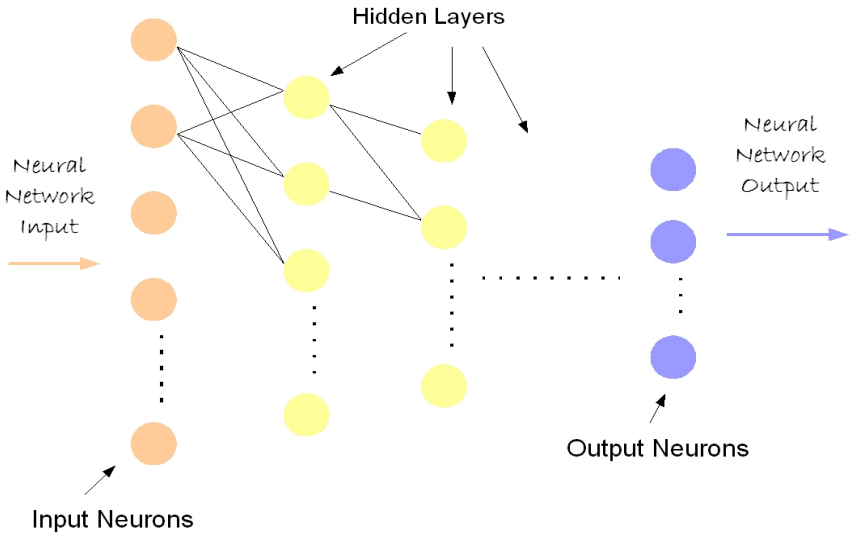



*Figure A.1: Neural network showing Input, Ouput and Hidden Layers*

In a biological neural network, a neuron is a special type of cell, capable of transmitting signals to neighbouring neurons via connecting synapses. Each neuron creates an output based on input recieved from other neurons. A neural network is adaptive, responding to learning and information gained from processing of input data. The biological system is a complex network of interconnecting layers and neurons, capable of learning and pattern recognition in situations of massive data input.

In an artificial neural network, a biological system is simulated and used for statistical analysis and modelling of data, pattern recognition and modelling of complex relations between inputs and outputs. Each neuron is connected to all neurons in connecting layers, with no connection between neurons in common layers. Each connection, or synapses, has a weight, or strength, in the network, determined through learning gained in the system as data is passed through. Weights are adapted as the system learns. Input to any neuron in hidden or output layers is a weighted sum of inputs from neurons in the preceding layer. Inputs are assigned as they identified as useful variables in the input data sample.

Neural networks are used in data analysis, prediction, recognition and classification and are used in the Financial data analysis, study and prediction of market and financial data, image analysis, through pattern recognition and other fields where complex analysis of a large data sample is useful. In particle physics neural networks can be used for data analysis in studies of signal and background processes.

A neural network is trained to be useful using a data sample in which inputs and outputs are known. A network takes input data and passes this through the network many times, adaptive and learning to create an accurate output, which in the training sample, is known. After training a network can then be used to assess data samples based on given inputs when the output is not known in advance.

Each connection, or synapses, has a strength or weight that can be varied. The input to a node is the weighted sum of the inputs in a previous layers. The input to the $j$th

node in the first hidden layer is

$$a_j = \sum_i w_{ij} z_i \tag{A.1.1}$$

where $z_i$ is the output from the $i$th node in the input layer and $w_{ij}$ is the weight of the connection between the nodes. The output from the node is

$$z_j = f(a_j + W_{j0}) \tag{A.1.2}$$

where $f$ is some sigmoid function, for example the logistic function

$$f(x) = \frac{1}{1 + e^{-x}}$$

and $W_{j0}$ is the bias or threshold of the node. The output of a node is a non linear function of the inputs to the node. During training, a neural network calculates an output based on the weights between nodes, compares the output to the expected output and adjusts the weights to create an output closer to the correct value. The process is repeated multiple times. Network training is defined as Supervised Learning when both inputs and outputs of a data sample are presented to the network for training. The intention is to produce a network mapping that can then be used for analysis on an unseen data sample, where the output is not known.

An error function minimisation method is the basis of learning in a neural network. Learning can be described in two steps, evaluation of the derivative of the error with respect to the each weight in the network using a back propegation method, then use of the derivatives to adjust the weights to minimise the error function.

A number of learning methods for minimisation of an error function are available for implementation in artificial neural networks. For example

- Robbins Monro method
- Steepest descent with fixed step size

- Steepest descent algorithm with line search
- Conjugate gradients with the Polak Ribiere updating formula
- Conjugate gradients with the Fletcher Reeves updating formula
- Broyden Fletcher Goldfarb Shanno method

The analysis in this thesis uses the ROOT artificial neural network multilayer perceptron, MLP, within the Toolkit for Multivariate Analysis, TMVA, data analysis software, and the Broyden Fletcher Goldfarb Shanno, BFGS, learning method. Selection of a learning method, as with structure of the artificial neural network, is based on experimentation with learning methouds and outputs on training data.

A detailed description of Neural Networks can be found in [69] and [70]. A description of the multilayerperceptron and learning methods can be found in [71].

# Appendix B

# Preliminary Studies for TAG Database Scalability

## B.1 Horizontal Partitioning Studies

Study of Horizontal partitioning strategies. Five methods are available

- Range - partition by values in a given range
- List - partition by values defined in a list
- Hash - partition by some hash value allocation
- Range-List - partition first by range, then by list
- Range-Hash - partition first by range, then by hash

Performance is seen to improve directly with the amount of data removed from consideration for all methods.

Figure B.1 shows the time taken to locate an increasing number of rows for a database of one million rows, ten percent of the scale of the proposed terabyte scale database. The tests aim to study database behaviour patterns an identify stategies of interest and potential usefulness to a larger scale database. The graph shows that all horizontal partitioning

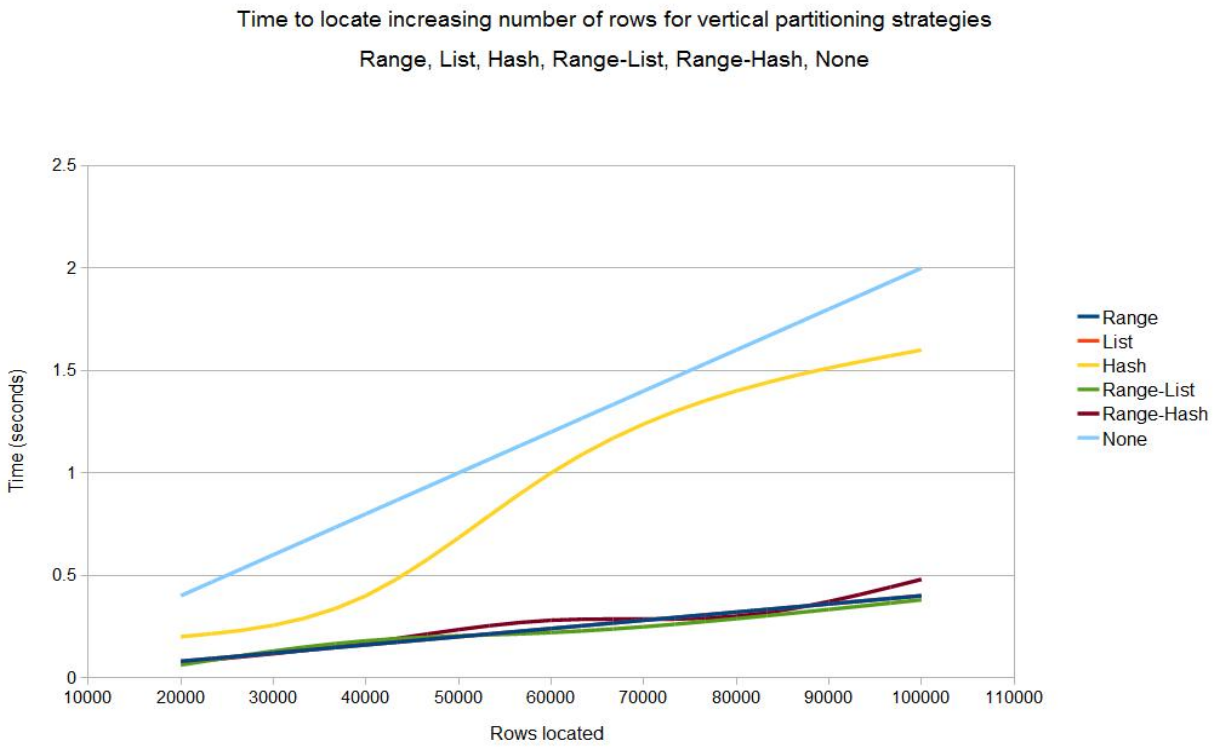



*Figure B.1: Horizontal partition testing results*

strategies perform consistently better than an unpartitioned table for an increasing number of rows. The strongest performance is seen by Range and List partitioning, where the results are identical to the nearest millisecond and as a result overlapped on the graph. Hash partitioning performs similarly to Range and List, but has a larger fluctuation and therefore inconsistency in performance. Of the composite partitioning schemes Range-List and Range-Hash, the former offers a more consistent performance, the latter introducing an inconsistency associated with the Hash partitioning.

As Range and List perform best, we identify these as methods of interest. In fact the methods differ only in the way the partitions are defined in the database and not in resulting structure and are therefore considered as a single method. Defining partitions by Range rather than List is preferable in a TAG Database where values selected for each partition could be multiple and extend to accuracies of multiple distinct decimal places, so Range partitioning is selected as being the strategy of interest. Hash partitioning is judged to introduce fluctuations without significant performance benefit over other single partitioning strategies, so is not selected over the other single partitioning strategies.

The composite partitioning strategies, while performing more slowly than single schemes, do perform better than no partitioning at all. It may be that at larger scale, a single partitioning strategy is difficult to implement and a composite strategy more appropriate, so for this reason Range–List, more predictable that Range-Hash, is also identified as interesting to implementation of a TAG Database.

## B.2 Vertical Partitioning Studies

Performance impact of joins across vertical partitions are low and only a disadvantage when querying across all partitions. Figure B.2 shows the results for time taken to locate 100000 rows in a table partitioned into ten vertical partitions compared to returning the same rows in a table without vertical partitions. The number of attributes in the query is increased in order to expand the reach of the query across an increase number of partitions in the vertically partitioned table. Figure B.2 shows that vertical partitioning offers performance improvement directly proportion to the amount of data relevant to the query, which is less when only a subset of partitions need to be accessed to locate the attributes given in the query. This is the case for queries that include up to ninety percent of partitions. However, when all partitions are involved, the results show that in fact a query on a partitioned table performs more slowly than the same query on an unpartitioned table.

Looking at the query execution plan in the case of the queries on the partitioned table, it is seen that joins of tables are necessary for partitioned tables and while this is not so expensive to make the query less performant when only some of the partitioned tables are needed for the query, when all partitions need to be accessed the cost of the joins leads to a query taking longer on a partitioned table than on an unpartitioned one.

Despite the performance benefits when only a subset of vertical partitions need to be accessed, it is the case that queries on the TAG Database may include a number of varying attributes and therefore may need to access all partitions. For this reason the vertical partitioning strategy is noted as potentially interesting, but not implemented in the first terabyte scale databases.

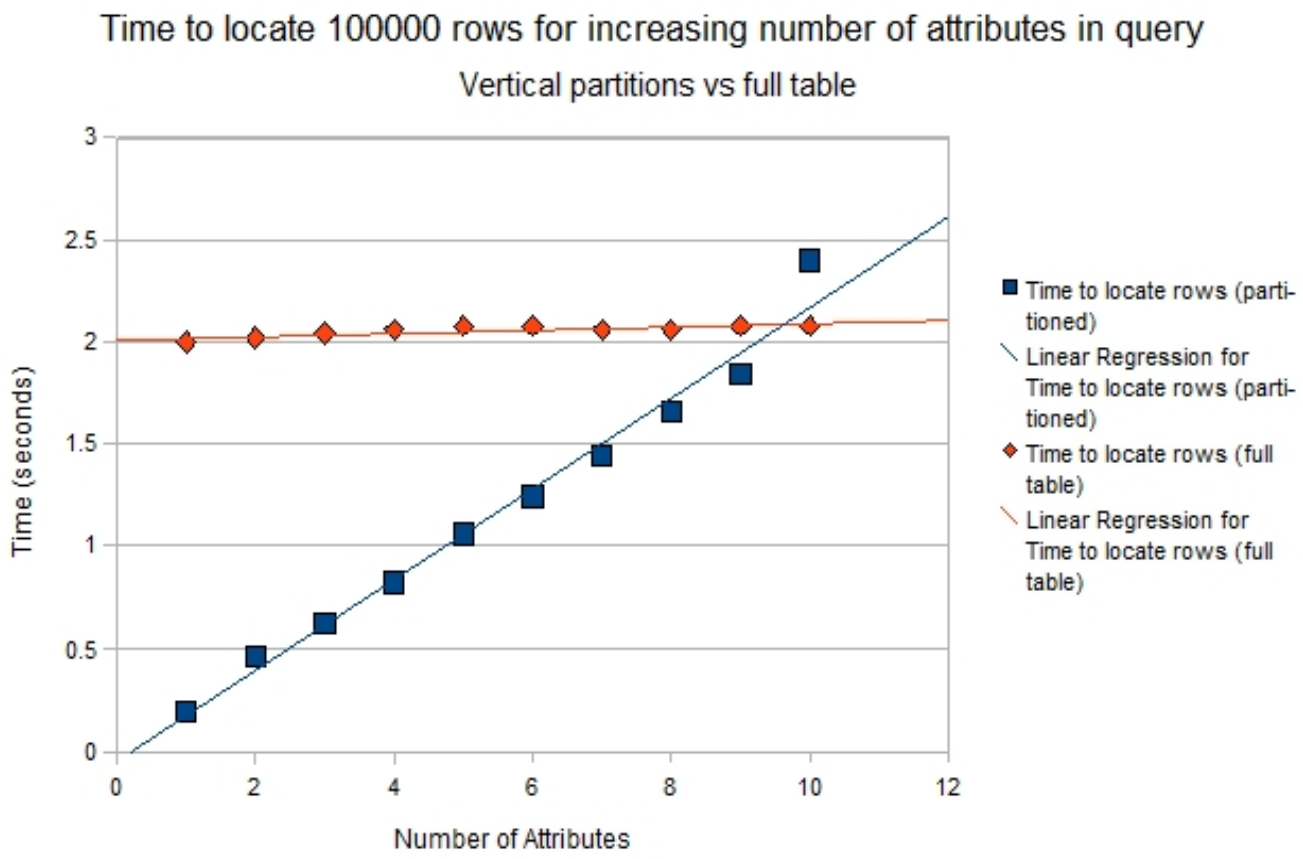



*Figure B.2: Vertical partition testing results*

# Bibliography


[1] IJR Aitchison and AJG Hey *Gauge Theories in Particle Physics, Volume 1: From Relativistic Quantum Mechanics to QED* IoP Publishing 2003

[2] IJR Aitchison and AJG Hey *Gauge Theories in Particle Physics, Volume 2:QCD and the Electroweak Theory* IoP Publishing 2004

[3] Donald H Perkins *Introduction to High Energy Physics* 4th Edition, Cambridge University Press

[4] Pierre Raymond *Journeys Beyond the Standard Model* Westwood Press

[5] CCLRC *Proceedings of the School for Experimental High Energy Physics Students held 5 to 17 September 2004* Technical Report, April 2005, RAL-TR-2005-02

[6] http://www.fnal.gov

[7] http://public.web.cern.ch/Public/en/Research/LEPExp-en.html

[8] C. Amsler et al *The Review of Particle Physics*, Physics Letters B667, 1, 2008

[9] P Higgs, *Broken symmetries, massless particles and gauge fields*, Phys Lett, 12:132-133, 1964

[10] P Higgs, *Broken symmetries and the mass of gauge bosons*, Phys Rev Lett, 13:508-509, 1964

[11] R.N. Mohapatra et al, APS neutrino theory working group, *Theory of Neutrinos: A White Paper, Reports on Progress in Physics 70: 1757, 2007*

*[12] LHC webpage* http://lhc.web.cern.ch/lhc

*[13] CERN webpage* www.cern.ch

*[14] Janice Drohan* An Investigation of the Higgs Boson Production Channel $t\bar{t}H, H \rightarrow b\bar{b}$ with the ATLAS Detector at the LHC

*[15] ATLAS webpage* http://atlas.web.cern.ch/Atlas/index.html

*[16] T Lagouri* ATLAS : A General Purpose p-p Experiment at the Large Hadron Collider (LHC) at CERN, *November 1999, MPI-PhE-99-19*

*[17] ATLAS Collaboration* ATLAS Detector and Physics Performance Volume 1 *Technical Report, CERN, May 1999, CERN-LHC-99-14*

*[18] ATLAS Collaboration* ATLAS Detector and Physics Performance Volume 2 *Technical Report, CERN, May 1999, CERN-LHC-99-14*

*[19] CMS Collaboration* CMS, The Compact Muon Solenoid Technical Proposal, *Technical Report, CERN, CERN-LHCC-94-38*

*[20] LHCb Collaboration* LHCb Technical Proposal, *Technical Report, CERN, CERN-LHCC-98-04*

*[21] ALICE Collaboration* ALICE Technical Proposal, *Technical Report, CERN, CERN-LHCC-95-71*

*[22] ATLAS Collaboration* ATLAS Inner Detector Technical Design Report *Technical Report, CERN, April 1997, CERN-LHCC-97-16*

*[23] ATLAS Collaboration* ATLAS Calorimeter Performance Technical Design Report *Technical Report, CERN, December 1996, CERN-LHCC-96-40*

*[24] ATLAS Collaboration,* ATLAS Muon Spectrometer Technical Design Report *Technical Report, CERN, June 1997, CERN-LHCC-97-22*

*[25] Atlantis Event Display* http://www.hep.ucl.ac.uk/atlas/atlantis

*[26] ATLAS Collaboration,* ATLAS Level 1 Trigger Technical Design Report *Technical Report, CERN, June 1998, ATLAS-TDR-12*

*[27] ATLAS Collaboration,* ATLAS High Level Trigger, Data Aquisition and Controls Technical Design Report *Technical Report, CERN, October 2003, CERN-LHCC-2003-022*

*[28] The ALEPH Collaboration, the DELPHI Collaboration, the L3 Collaboration, the OPAL Collaboration, the SLD Collaboration, the LEP Electroweak Working Group, the SLD electroweak, heavy flavour groups* Precision Electroweak Measurements on the Z Resonance *Phys. Rept. 427:257, 2006*

*[29] Christopher Collins-Tooth* The Higgs Discovery Potential of ATLAS *December 2007, GLAS-PPE/2007-19*

*[30] G Kane* The Dawn of Physics Beyond the Standard Model *Scientific American, June 2003*

*[31] LEP webpage* http://public.web.cern.ch/Public/en/Research/LEPExp-en.html

*[32] CDF webpage* http://www-cdf.fnal.gov

*[33] D0 webpage* http://www-d0.fnal.gov

*[34] F. Abe et al* Observation of Top Quark Production in pp Collisions with the Collider Detector at Fermilab *Physical Review Letters 74, 1997*

*[35] ATLAS Collaboration* ATLAS Computing Technical Design Report *Technical Report, CERN, June 2005, CERN-LHC-2005-22*

*[36] LHC Computing Grid webpage* http://lcg.web.cern.ch/lcg

*[37] Caitriana Nicholson* File Management for HEP Data Grids *March 2006*

[38] *Miguel Branco, David Cameron, Pedro Salgado* DDM Design and Implementation *November 2006*

[39] *Mario Lassnig, Miguel Branco, David Cameron, Benjamin Gaidioz, Vincent Garonne, Birger Koblitz, Massimo Lamanna* Managing ATLAS data on a petabyte-scale with DQ2 *Computing in High Energy Physics, September 2007*

[40] *ATLAS Distributed Data Management webpage* https://twiki.cern.ch/twiki/bin/view/Atlas/Distrib

[41] *Pool Of Persistent Objects for LHC webpage* http://pool.cern.ch

[42] *The ROOT System webpage* http://root.cern.ch

[43] *Oracle webpage* www.oracle.com

[44] *MySQL webpage* www.mysql.com

[45] *SQLite webpage* www.sqlite.org

[46] *Richard Hawkings, Florbela Viegas, Gancho Dimitrov* Relational databases for conditions data and event selection in ATLAS, *Computing in High Energy Physics, September 2007*

[47] *Pedro, L and Perus, A and Badescu, E and Burckhart-Chromek, Doris and Caprini, M and Flammer, J and Dobson, M and Kazarov, A and Kolos, S and Kotov, V and Liko, D and Klose, D and Mapelli, L and Mineev, M and Ryabov, Yu and Soloviev, I and Alexandrov, Yu A and Amorim, A and Barros, N and Cook, J and Franco, T and Hawkings, R and Malon, D and Schaffer, A and Jones, R* Conditions Databases, The Interfaces between the Different ATLAS Systems, *Computing in High Energy Physics, Interlaken, Switzerland, September 2004*

[48] *Jack Cranshaw, Ketevi Assamagan, Kyle Cranmer, Amir Farbin, Julius Hrivnac, David Malon* Requirements and Use Cases for ATLAS Event Level Metadata, *November 2005*

*[49] Assamagan, K A and Barberis, D and Bentvelsen, Stanislaus Cornelius Maria and Brooijmans, G and Cranmer, K and Cranshaw, J and Dell'Acqua, A and Farbin, A and Froidevaux, D and Gianotti, F and Hinchliffe, Ian and LeCompte, T and Maeno, T and Malon, D and Paige, F and Polesello, G and Quarrie, D and Rousseau, D and Schaffer, R D and Smizanska, M and Unal, G and Voss, K and Wielers, M* Report of the Event Tag Review and Recommendation Group*, April 2006, CERN, ATL-SOFT-PUB-2006-002. ATL-COM-SOFT-2006-003*

*[50] The Event Level Selection Service webpage* https://atldbdev01.cern.ch/tagservices/index.htm

*[51] D Malon1, P van Gemmeren1, R Hawkings2 and A Schaffer3* An inconvenient truth: file-level metadata and in-file metadata caching in the (file-agnostic) ATLAS event store *Computing in High Energy Physics, September 2007*

*[52] Caitriana Nicholson, Helen McGlone* Experience using DDM*, Event Store Developers Workshop, February 200*

*[53] Caitriana Nicholson, Jack Cranshaw, Anthony Doyle, Mike Kenyon, David Malon, Helen McGlone* Integration of the ATLAS Tag Database with Data Management and Analysis Components*, Computing in High Energy Physics, September 2007*

*[54] Qizhi Zhang's webpage* http://www.hep.anl.gov/qzhang

*[55] Helen McGlone, Jack Cranshaw, Luc Goosens, David Malon, Florbela Viegas* Building a Scalable Event-Level Metadata System for ATLAS*, Computing in High Energy Physics, September 2007*

*[56] Solveig Albrand, Jerome Fulachier* ATLAS Metadata Interfaces (AMI) and ATLAS Metadata Catalogs *Computing in High Energy Physics, Interlaken, Switzerland, September 2004*

*[57] AMI web interface* http://ami.in2p3.fr/opencms/opencms/AMI/www

*[58] Julius Hrivnac* http://hrivnac.home.cern.ch/hrivnac/Activities/Packages/SQLTuple

*[59] D Costanzo, J Cranshaw, S Gadomski, S Jzquel, A Klimentov, G Lehmann-Miotto, D Malon, G Mornacchi, P Nmethy, T Pauly, H von der Schmitt, D Barberis, F Gianotti, I Hinchliffe, L Mapelli, D Quarrie, S Stapnes* Metadata for ATLAS *April 2007, CERN, CERN-ATL-COM-GEN-2007-001*

*[60] Jochen Cammin, Markus Schumacker* The ATLAS Discovery Potential for the channel $t\bar{t}H, H \to b\bar{b}$

*[61] The ATLAS Collaboration* Search for $t\bar{t}H, H \to b\bar{b}$ with the ATLAS detector *ATL-PHYS-PUB-2006-000, June 2008*

*[62] S Frixione and BR Webber,* MC@NLO *JHEP 0206 (2002) 029 and updates*

*[63] B Kersevan and E Richter-Was,* Acer MC *Comp. Phys. Comm.149 (2003) 142 and updates*

*[64] T Sjostrand, S Mrenna and P Skands,* Pythia *JHEP 0605 (2006) 26*

*[65] G Corcella et al,* Herwig *JHEP 0101 (2001) 010*

*[66] J Butterworth et al,* http://projects.hepforge.org/jimmy

*[67] The ATLAS Collaboration,* Jet Reconstruction Performance, ATL-PHYS-PUB-2006-000*, June 2008*

*[68] The ATLAS Collaboration,* Cross-Sections, Monte Carlo Simulations and Systematic Uncertainties *ATL-PHYS-PUB-2006-000, June 2008*

*[69] C Bishop* Neural Networks for Pattern Recognition *Oxford University Press, 1995*

*[70] K Gurney* Introduction to Neural Networks *University College London Press, 1997*

*[71] C Delaere* The MultiLayerPerceptron *http://www.fynu.ucl.ac.be/users/c.delaere/MLP*