

AN INVESTIGATION INTO HTC PERFORMANCE OF THE WORLDWIDE  
LHC COMPUTING GRID

by

Jerrod Dixon

A THESIS

Presented to the Faculty of  
The Graduate College at the University of Nebraska  
In Partial Fulfilment of Requirements  
For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor David Swanson

Lincoln, Nebraska

May, 2018

AN INVESTIGATION INTO HTC PERFORMANCE OF THE WORLDWIDE  
LHC COMPUTING GRID

Jerrod Dixon, M. S.

University of Nebraska, 2018

Adviser: David Swanson

The Worldwide LHC Computing Grid (WLCG) has been used to great effect to process the data produced at the Large Hadron Collider (LHC). The data is distributed internationally, and accessed from across geographic locations. This paper looks to investigate performance hits because of network issues.

## ACKNOWLEDGMENTS

# Contents

<b>Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>3</b>
2.1 HTCondor ClassAd data . . . . .	4
2.2 perfSONAR . . . . .	4
2.3 Important Values . . . . .	4
<b>3 Related Work</b>	<b>6</b>
<b>4 Experiment on CMS performance</b>	<b>7</b>
4.1 Relating Datasets . . . . .	7
4.2 Execution site vs Data site . . . . .	8
4.3 Processing . . . . .	8
<b>5 Results</b>	<b>9</b>
5.1 Affect on performance . . . . .	9

5.2 Inverse relationship occurrence between <code>CpuEff</code> and <code>EventRate</code> . . . .	11
<b>6 Conclusions and Future Work</b>	<b>13</b>
<b>Bibliography</b>	<b>15</b>

# List of Figures

5.1	Packet Loss in workflow of DisplacedJet . . . . .	10
5.2	Latency in workflow of BTagCSV . . . . .	10
5.3	Latency in workflow of BTagMu . . . . .	10

# List of Tables

2.1	Typical HTCondor properties . . . . .	5
2.2	CMS-specific HTCondor properties . . . . .	5
5.1	Percent of time a regional location is accessed as a data location . . . . .	10
5.2	Percent of time a regional location is attributed to negative slope . . . . .	11
5.3	Percent of time Workflows have inverse slopes with prefix fabozzi_Run2016B	12

# Chapter 1

## Introduction

The Worldwide LHC Computing Grid (WLCG)[2] manages the processing of computing centers around the world. They cooperate in processing the large amounts of data ( 50 Petabytes in 2016 alone) produced by the Large Hadron Collider (LHC) at CERN.

Over the years, the processing methods have improved, and the implementation of the WLCG itself has grown in such a way that more data is produced than can be processed in a reasonable amount of time. This has resulted in backlogs of data requiring processing. Coordination of work is at issue as well, deciding when and where the transfer of the data to offsite locations should occur and where it is to go.

Depending on which resources require the dataset, the data can be replicated in multiple geographically distant locations in such a way to allow easier access. Also, load balancing of work causes issues with users submitting work to be done to the same processing center. This can cause certain locations to become overloaded with requests for processing while others go almost unused. By extension the data also runs into the issue of being requested to do the calculations against getting called to a processing center that is not geographically close, for example the job is processed



in Italy but the data exists in Spain, slowing the acquisition time.

It has been an interesting subject but an investigation into how this work is done and split across sites has yet, to the authors knowledge, been done yet. For the purposes of this paper the data processed by the WLCG is investigated, with special attention paid to work done where the data for the job itself exists offsite from where the work is being done. Due to limitations of knowing how much data was pulled from where, only jobs with one data location are used.

## Chapter 2

# Background

A grid network is classically defined as a software solution, implementing general commodity hardware, to enable the creation of a series of machines to achieve High Performance Computing (HTC). The benefit here is that by implementing such a solution, allows for cheaper access to HTC style computing without requiring investment into designing and implementing specialized hardware to improve scientific computations.

In High Throughput Computing solutions, there are times when the cluster is not fully in use by the institution that purchased it. Because of this, institutions across the globe have come up with methods to connect their personal clusters to a international grid. In this way, these HTC solutions are better utilized to achieve a higher usage saturation point, taking work from excessively busy systems and giving work to idle ones.

## 2.1 HTCondor ClassAd data

The data is gathered first using a python script to spider across the WLCG to pull ClassAd information about the work done on their respective clusters from the HTCondor[1] database. This data is then stored on an elasticsearch server to store for long term access. This gathers the statistics about the jobs themselves, such as duration of job, event rate, and efficiency of processing defined by the field `CpuEff` in Table 2.1.

## 2.2 perfSONAR

PerfSONAR[7] servers are setup at different regional locations to record different statistics about the network communications across links to other WLCG locations. This data includes throughput, packet loss, latency, and traceroute data (though for this investigation the traceroute data remains unused). This data is also stored to a MWT2 elasticsearch service at preset 'pulse' intervals.

## 2.3 Important Values

In the HTCondor ClassAds there are a few fields in particular we are interested in, as can be seen in Table 2.1. There are however, CMS experiment specific values we are interested in as well, defined in Table 2.2. Both of these sets of values are stored and reported by the HTCondor databases, recorded as ClassAds. On their own they can be interesting, but we are primarily concerned with their relation to networking information. Correlation with the perfSONAR data and HTCondor values are done in this experiment.

Table 2.1: Typical HTCondor properties

Field	Description
CpuEff	Percentage value of CPU time spent processing data over $X$ amount of time.
JobCurrentStartDate	Date and time the job started processing, not when it was submitted.
JobFinishedHookDone	Date and time the job finished processing, not when it left the system.
WallClockHr	Integer value of time spent processing, invariant of number of cores.
RequestCpus	Number of cores used in processing work.

Table 2.2: CMS-specific HTCondor properties

Field	Description
KEvents	Number of events processed by the job in thousands.
InputGB	Size of input data for the job to process.
ChirpCMSSWEventRate	Clock rate of CPU as measured during the last 15 minutes of run time.
Workflow	Name of the workflow the job is working on in human readable format.

## Chapter 3

### Related Work

The ability of this work to make investigations into the performance of jobs against network related data is directly related to the work of [6]. As stated, the collaborations for the LHC rely heavily upon network technologies for the infrastructure but it is incredibly difficult to locate issues. It is from this inspiration that this work draws to investigate the work done utilizing the WLCG grid network while accounting for the affect network communications have on the processing and transference of data.

In addition, the primary gathering of HTCondor ClassAds is performed via a script designed and executed by [3]. Without this work there would be no easy method to investigate the HTCondor ClassAds distributed around the world, and read their values.

Of particular concern is the CRAB3 scheduling system, a layer of abstraction thrown on the WLCG implementation to give ease of access and submission methodology to allow distribution of work across the network. As described in [5], the CRAB3 method also is important to the method of data distribution. It works by allowing users to provide ad-hoc methods of processing, and distributing work to sites that have agreed to processing, depending on their tier level.

## Chapter 4

# Experiment on CMS performance

The data processed is gathered and interpreted from July 16th, 2016 to July 23rd, 2016.

### 4.1 Relating Datasets

These two datasets are unrelated beyond location data (that is, whether it was recorded at the 'source' location or the 'destination' location). What the network performance data records as 'source' and 'destination' is considered in the htcondor ClassAd data as 'processing location' and 'data location', respectively. To relate these two datasets, values stored in the MWT2 elasticsearch server are considered to be relevant if the 'source' location is the 'processing location' and the 'destination' location is the 'data location' where the dataset for the workflow to be processed is stored. In addition, the store date must be within the range of the processing time for the job itself.

The relating data is stored on a staging elasticsearch server to hold the processed data. Each record in this third storage server relates to a ten minute period consid-

ering all work and network traffic during that period. Per a given 10 minute period values relating to the mean, median, min, max, and standard deviation ( $\sigma$ ) of the distribution of each value considered is stored to maintain statistical qualities of the data the record represents.

## 4.2 Execution site vs Data site

While correlating the data, we tie the `src` property of the perfSONAR record value to the job execution site, indicating the location of recording the network state is the same as where the work is performed.

Alternatively, we tie the `dest` property of the perfSONAR record value to the data location, indicating the recording of the network state from where the data exists.

## 4.3 Processing

Once this data is correlated it is stored, as previously stated, on a staging elasticsearch server. Graphs of these values are produced via python script, separating the graphs so that each is specific for a particular site, over individual links, on specific days, and per workflow. A line of best fit is calculated per value relationship. For interpreting this line of fit, the  $r$  value is generally ignored as it is uncertain what a best fit  $r$  value would be. The relationship we are primarily concerned with is if the slope of the line is positive or negative, to indicate an increasing or decreasing value. As such the p-value is used to determine if the line is valid results. Keeping with common practices, only lines with a value below 0.05 are observed.

# Chapter 5

## Results

In the processing of data to calculate the relationships between HTCondor work data and perfSONAR networking data such as what is seen in Figure 5.1, 5.2, and 5.3, a line of linear regression is calculated for the relationship to best guess the quality of comparison. It is uncertain whether the r-value fitting the regression to the data is a valid comparison, so the calculated p-value is used to assume quality. The line of regression is not used to predict future values, only ensure the data is closely related. Since we are only curious about the relationship, whether both sets follow similar trends, we only need to know if the slope of the line follows a positive or a negative trend. As such, the p-value is assumed sufficient in qualifying the fit. Only relationships with a p-value that is less than 0.05 are accepted a valid set.

### 5.1 Affect on performance

As a metric of performance on a system, we will primarily be observing the `CpuEff` property, as defined in Table 2.1. We're interested primarily in the relation of how often `CpuEff` has a negative slope value when compared to network performance.



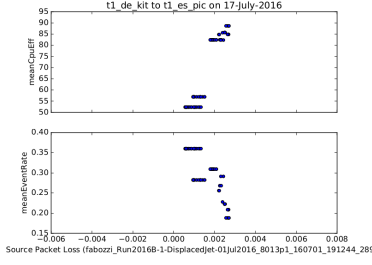


Figure 5.1: Packet Loss in workflow of DisplacedJet

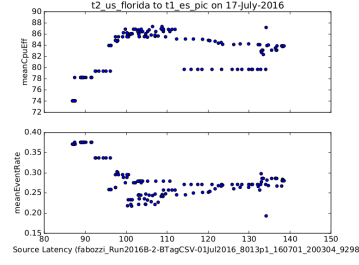


Figure 5.2: Latency in workflow of BTagCSV

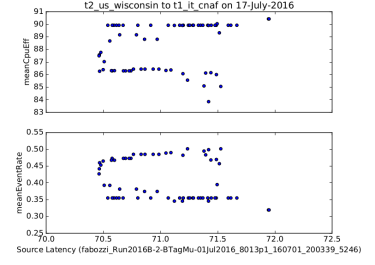
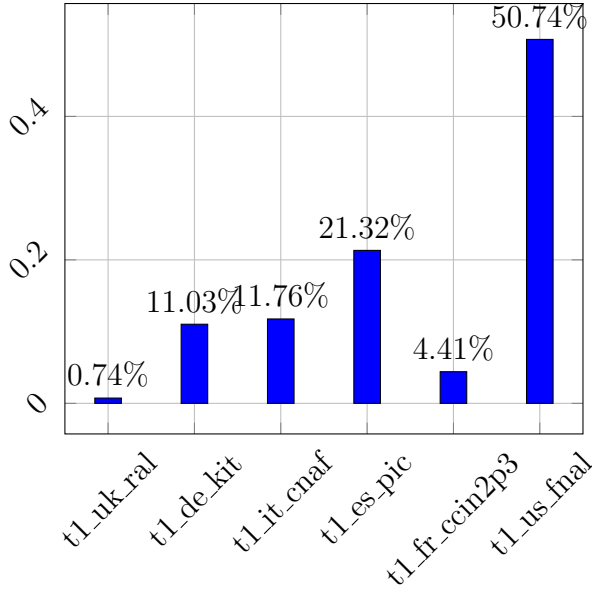


Figure 5.3: Latency in workflow of BTagMu

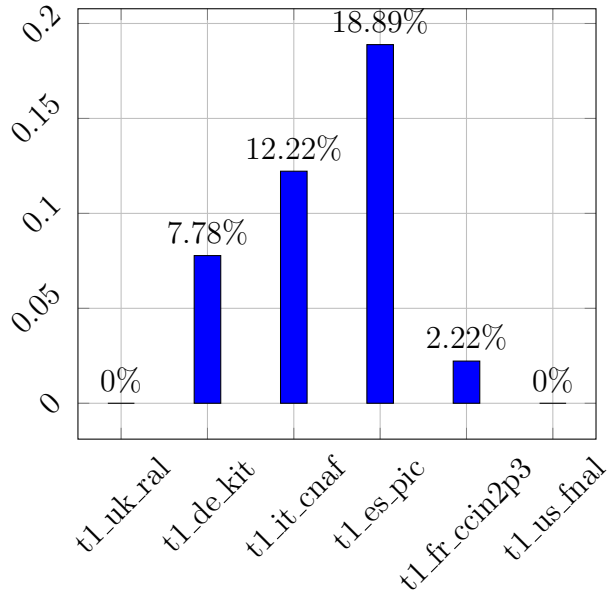
Table 5.1: Percent of time a regional location is accessed as a data location



Across all the sites noted as execution locations, it is first important to determine where the data is being pulled from, when the data exists offsite from the current location. As shown in Table 5.1, only 6 locations are used when accessing offsite data. Of the most concern, is the `t1_us_final` data location, which is accessed 50.7% of the time to pull data from.

Taking into account how often a calculated linear regression line results with a negative slope when accounting `CpuEff` versus the calculated network values, maintaining a calculated p-value of less than 0.05, we can observe in Table 5.2 that during

Table 5.2: Percent of time a regional location is attributed to negative slope



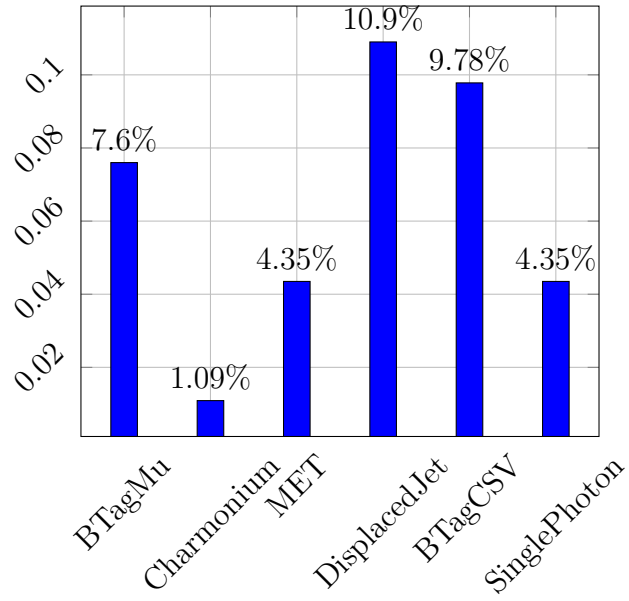
the calculated period that `t1_es_pic` exhibits the highest amount of work where efficiency is lost due to network performance.

## 5.2 Inverse relationship occurrence between CpuEff and EventRate

For the data collected, when comparing `EventRate` and `CpuEff` to networking information, 48.9% of the time the two values have an inverse relationship. That is, while one improves the other declines. As seen in Table 5.3, the distribution of which workflows show this relationship are displayed.

It is uncertain what exactly is causing this relationship. Given the nature of work for the CMS experiment, a potential implication is the luminosity of the given data. For Scattering Theory, the concept of luminosity[4] is the relationship between the number of events generated in the experiment over time. The field `KEvents` contains

Table 5.3: Percent of time Workflows have inverse slopes with prefix fabozzi\_Run2016B



the number of events processed total, but there is no job data holding over how much time that number of events existed.

## Chapter 6

# Conclusions and Future Work

From the observed period we can see that, as shown in Table 5.1, the regional locations used to store data are very limited in number, and are very specific sites. Despite `t1_us_fnal` being the majority storage location of processable data, `t1_es_pic` seems to be the regional location where performance drops when accessing data stored there. The processed values are limited though, as work where multiple data locations are used cannot be calculated. Currently there is no mechanism or property which indicates how much data is pulled from where when data exists in multiple locations.

However, as shown in Table 5.3, the data processed from this given period has an inverse relationship between performance of the work, defined by `CpuEff`, and the rate of CPU usage, defined by `EventRate`. Further processing is required to resolve exactly why this is, potentially relating to the luminosity[4] of the data being processed. In order to fully determine if this is the source, a method of reverse-calculating the luminosity record of the processing job will be required.

Currently, the value of fit of the data is done via a linear regression. For future work better statistical modeling utilizing Data Mining processing methods, such as a priori modeling, would be a valuable method of investigation into the job performance

dataset. In addition, utilization of streaming methodology would be an interesting method of processing data live rather than through batch work.

# Bibliography

- [1] *HTCondor*.
- [2] Worldwide lhc computing grid.
- [3] Brian Bockelman. cms-htcondor-es, September 2016.
- [4] CERN. Concept of luminosity, March 2017.
- [5] collaboration CERN. Cms distributed data analysis with crab3, 2015.
- [6] Shawn McKee. Wlcg network monitoring using perfsonar-ps, April 2013.
- [7] perfSonar. perfsonar motivation, March 2017.