Machine Learning for Fast Data Transfers

Nikhil Krishnan
California Institute of Technology
nkrishna@caltech.edu

Dorian Kcira*
California Institute of Technology
dkcira@caltech.edu

Maria Spiropulu†
California Institute of Technology
smaria@caltech.edu

Abstract

The Large Hadron Collider (LHC) is the most powerful particle collider in the world, and the Caltech Compact Muon Solenoid (CMS) group produces tens of petabytes of data in its experiments at the LHC. As a result, the CMS group has its own infrastructure for data transfers, implementing grid-based data analysis and global-scale networking. Because the CMS experiment generates voluminous amounts of data, there needs to be a way to improve the current data management tools to optimize speed of transfers and facilitate fast I/O access. Machine learning offers numerous methods for such an optimization, as the field allows us to look at the present state of the network and use information about recorded transfers to predict features about future data transfers, thereby reducing the number of variables the data management tool has to consider. Using logged information from PhEDEx, the CMS data transfer system, we created a large sample matrix of data transfers, partitioned it into testing and training samples, and conducted Principal Component Analysis, a machine learning method that projects a large multivariate dataset onto the two variables that most explain variance.

I. Background

The Caltech Compact Muon Solenoid (CMS) group has been at the forefront of the Large Hadron Collider (LHC) computing and software efforts aimed at enabling grid-based data analysis for the last 15 years, as well as global-scale networking for the last 30 years [8]. Based on the experience with LHC data taking, central processing, and user analysis, it has become clear that fast data transfers and distributed file systems that allow fast I/O access are of crucial importance. Caltech has made progress towards implementation of both systems and interfaces between them at Caltech and within the CMS grid infrastructure. Work is ongoing to bring these systems at a level of maturity to be used in production at CMS and elsewhere.

One of CMS's requirements for sites this year, was to demonstrate that after their respective 100 Gbps links were upgraded, the sites would be able to transfer data through the Wide Area Network (WAN) at rates higher than 20 Gbps. Caltech was the first site to achieve that milestone, and was invited to share its expertise with other sites. Caltech is currently coordinating the effort among all US CMS grid sites to scale up various aspects of the configuration, locally and on the central services, in order to find the best operating point for each site.

2015 will be especially important, as the Large Hadron Collider will resume operation at higher energy and luminosity, and thus with extended discovery will reach the spring of 2016, following an extended shutdown during which the accelerator, the ATLAS (A Toroidal

*Principal Mentor, Research Scientist
†Project Supervisor, Professor of Physics

LHC ApparatuS) and CMS detectors and their data acquisition systems have been upgraded. Judging by the data flows across the Atlantic and Research and Education Networks in 2013 and 2014, the resumption of LHC operations with the prospect of new discoveries in the wake of the Higgs boson discovery in 2012. This will bring a new level of demand and challenge in terms of worldwide grid computing, data storage at rates well above 100 petabytes per year, and data transfers of a few hundred petabytes during 2015.

The ANSE (Advanced Network Services for Experiments) project started at Caltech, which plans its first production deployments starting in the next quarter, will have a key role in supporting the new level of data operations, and enabling the LHC experiments to realize their discovery potential. ANSE's goal is to integrate network monitoring and network provisioning capabilities with the software stacks of the CMS and ATLAS experiments at the LHC. This is achieved through enabling more deterministic time to complete a designated set of data transfers. Furthermore, 2014 has been a time of rapid evolution in Software Defined Networking (SDN) and the ANSE team has adapted to and remains at the forefront of these developments, working on the Floodlight (2013-14) and subsequently in the OpenDaylight framework. They are developing intelligent path selection methods across complex networks supporting multiple data transfer requests, each of which is carried out through the use of multipath Transmission Control Protocol (TCP) and Caltech's Fast Data Transfer (FDT) application.

II. Introduction

The CMS experiment generates several tens of petabytes worth of data annually and has a grid infrastructure with more than 70,000 cores spread globally in the LHC grid. The Caltech CMS group in particular is constructing an architecture that will facilitate the LHC experiments' ability to assign priorities of data transfers between institutes in the CMS experiment. The architecture guiding the flow of data is comprised of the OpenDaylight controller, a multi-protocol infrastructure, and a plugin for the File Transfer Service (FTS) [5]. With these powerful data management tools in place, the goal is to optimize the completion time for transferring groups of data (datasets) between endpoints (grid sites or nodes).

Machine learning is critical to the solution of this problem, as its methods will allow the OpenDaylight controller to make intelligent decisions about sending data between nodes. The field itself involves developing algorithms for recognizing patterns in large datasets, known as "training sets," and applying them to make predictions on future data, known as "testing sets." This concept is instrumentally applicable to the CMS network infrastructure, as the present state of the network can be treated as a training set that will allow the software to make predictions about future datasets [7]. With this information, the software can learn to take decisions that will optimize the speed at which data is transferred across the network.

All of the different data management tools, described in detail below, record an extensive logging of analytics about the transfers that occur between the CMS grid sites. The logged data holds important information about the physics datasets, including the number of files transferred and each of their sizes, the source and destination of the file transfers, the rate at which data is transferred, the number of errors and expired files, etc. The aim of this project is to implement machine learning methods on the logged information from PhEDEx, the data transfer management system for CMS [8].

Machine learning problems can be grouped into two major categories: supervised learning, where the data considered has additional features that can be predicted by the algorithm, and unsupervised learning, where training sets can be analyzed to discover patterns in the data. We conducted Principal Component Analysis (PCA), a supervised machine learning method that projects a multivariate set onto two dimensions. This explains the variance in the data, visualizes the logged transfers and finds the correlations between the features to reduce the

number of random variables the data management software has to consider. Furthermore, after completing PCA, we used Pearson correlation to interpret the results. Ultimately, we are left with a greater understanding of the utility of the features about our data transfers, and these methods can be implemented in the CMS data management software to further its efficiency.

III. CMS Data Management and Transfer Tools

In order for the CMS experiment to be able to produce many petabytes of data every year, the infrastructure behind the network must be strong. The CMS group uses multiple key tools that comprise this immense infrastructure consisting of numerous grid sites, and a description of each is included below, which shows how LHC data is collected and transferred and the use of machine learning.

- **PhEDEx (Physics Experiment Data Export)**: the Data Transfer Management System for CMS. It manages the high level aspects of the transfers starting from datasets and transfer endpoints, allowing other software to perform the actual file transfers. PhEDEx is designed to handle this task with minimum operator effort, automating the workflows from large scale distribution of High Energy Physics (HEP) experiment datasets down to reliable and scalable transfers of individual files. PhEDEx handles virtually all CMS production data transfers, and we get the logged data used for our machine learning analysis from this system.
- **FTS**: the Grid Data Transfer Service is a data movement service that aims to reliably copy one Storage URL to another. It uses a third party copy (e.g. gsipftp) to achieve this, but will retry if the transfer fails. It also schedules the copy processes along network channels to ensure that the bandwidth is used properly.
- **SRM**: the storage resource manager, is a Grid storage service providing interfaces to storage resources, as well as advanced functionality such as dynamic space allocation and file management on shared storage systems. It calls on transport services to bring files into their space transparently and provide effective sharing of files. At a CMS grid site, SRM will delegate the file transfer to the GridFTP servers.
- **GridFTP**: a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks. The GridFTP protocol is based on FTP, the highly-popular Internet file transfer system.

A schematic representation of the data management and flow for CMS grid sites is given in Figure 1. One can see that the transfers are triggered by PhEDEx, which are then processed by FTS and SRM. Ultimately the transfers are started as GridFTP processes, running on dedicated GridFTP servers with appropriate computing resources and networking capabilities.

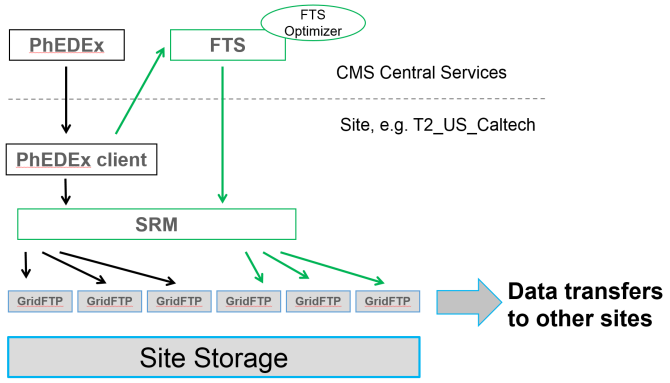

Figure 1: A diagram showing the path of CERN data transfers through PhEDEx.

IV. Data Transfers and Extraction

In order to conduct a proper analysis on all of the logged data collected, it is crucial to know what kind of data is collected by PhEDEx and the analytics of transfers themselves. The Caltech CMS group, fueled by the discovery of the

Higgs-like particle, works on collecting photon measurements needed to find the Higgs or other particles that could also decay to photons [1]. These calculations, taken at the LHC, will be used to search for and characterize new particles. As such, a critical component of the Caltech approach is controlling the flow of information from the LHC collisions, all of which is recorded and managed by PhEDEx. PhEDEx itself records information about the data transfers, and an example spreadsheet of such information is provided in Figure 2 [8].

Last hour

To		From	Files	Total Size	Rate	Errors	Expired	Avg. Est. Rate	Avg. Est. Latency
,*	or	,*							
T2_KR_KNU		T1_US_FNAL_Disk	354	1.4 TB	384.4 MB/s	-	-	173.6 MB/s	5h28
T1_DE_KIT_MSS		T1_DE_KIT_Buffer	665	1.1 TB	315.9 MB/s	-	1	47.9 MB/s	2h11
T2_KR_KNU		T2_RU_JINR	193	649.5 GB	180.4 MB/s	-	-	94.1 MB/s	2d17h28
T1_DE_KIT_Buffer		T1_US_FNAL_Disk	229	429.7 GB	119.3 MB/s	-	-	42.9 MB/s	3d19h28
T1_DE_KIT_Buffer		T1_IT_CNAF_Disk	239	393.0 GB	109.2 MB/s	-	-	32.4 MB/s	1d12h43

Figure 2: An example of information collected by PhEDEx about the CMS data transfers.

The first two columns of data represent the various grid sites that the transfers get sent to, and while they are relevant to finding a machine learning based solution, they were not considered in this paper. Rather, among the variables above, the features considered for PCA were the number of files, the total size of the files in GB, the rate of the file transfers in MB/s, the number of errors occurred during the transfer, the number of expired transfers and the average estimated rate of the transfers in MB/s before they take place.

Having established the recorded information about the data transfers, the data was then extracted and converted to a useful form. One of the most important tools needed to conduct machine learning analysis upon the recorded data is scikit-learn, a Python module based on NumPy and SciPy integrating novel machine learning algorithms for both supervised and unsupervised learning problems [6]. In order to conduct these methods on the logged data, it needs to be converted to an ndarray, a multidimensional homogeneous matrix native to NumPy that can be processed by scikit-learn [6]. To do this, the data was first extracted from PhEDEx in a JavaScript Object Notation (JSON) file and converted to a Comma Separated Value (CSV) file [9]. This was accomplished through the use of Python scripts, Microsoft Excel and the CSV python module. Once the CSV file was obtained, methods contained within NumPy were used to transform the CSV file into a readable ndarray. With the ndarray obtained, the next step of the project became to conduct PCA.

V. Principal Component Analysis

Principal Component Analysis (PCA) is a ubiquitous tool in the fields of data analysis and machine learning, and a discussion of the method as well as an example are provided to contextualize the project. Subsequently, Exact and Approximate PCA are conducted on the logged PhEDEx data, and the results are interpreted.

PhEDEx collects information regarding the transfer of large datasets. In order to find a pattern in the data, we use PCA, which provides us with a simple, non-parametric mechanism to extract relevant information. This methods uses an orthogonal decomposition to convert these variables into principal components, a smaller amount of linearly uncorrelated variables [3]. By doing so, dimensionality reduction is carried out on the data, and the dataset can be reduced to two dimensions, allowing the logged information to be visualized on a scatter plot, for instance. By performing PCA, a large multivariable dataset can be projected onto a plot where the x and y axes represent the principal components, allowing us to visualize large clusters of data and find patterns in them [10]. One of the fundamental procedures to carrying out PCA is the Singular Value Decomposition (SVD), a factorization of a matrix into a product of matrices. By altering the way SVD is conducted, we can find out multiple interesting patterns about the data, leading to two different types of PCA [3]. Two particu-

lar types of PCA were carried out, Exact PCA and Approximate PCA. The former uses linear dimensionality reduction through SVD of the data, preserving only the most significant singular vectors to project the data onto a lower dimensional space while the latter uses the same reduction but where the Singular Value Decomposition is computed randomly [11]. Figures 3 and 4 show the results after each of the PCA methods was carried out on the data.

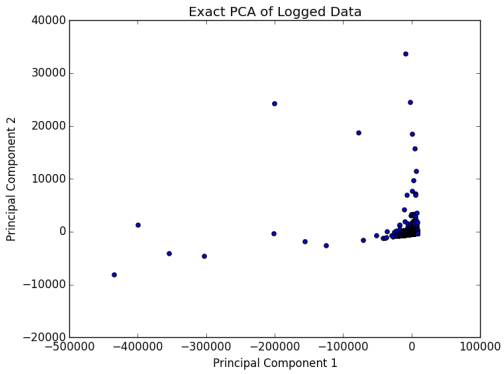

Figure 3: A graph representing Exact PCA carried about on the logged data transfers. Each dot represents the logged data transformed onto the prinicipal components, where the x-axis is Principal Component 1 and the y-axis is Prinicipal Component 2.

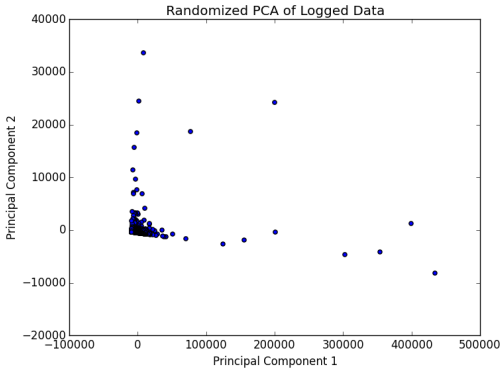

Figure 4: A graph representing Randomized PCA carried about on the logged data transfers. The clustering of data points in corners for both this and FIgure 3 represent correlations between features of our logged data, corroborating subsequent analysis.

VI. Interpretation and Results

While the scatter plots are visually interesting, it is necessary to formalize a method for interpreting the results of the PCA plots. The features considered were number of files, total file size, rate, number of errors, number of expired files and average estimated rate of transfer. A critical tool in being able to interpret the results of these graphs is the Pearson-product-moment correlation coefficient, simply a measure of the linear correlation between two variables. By computing the Pearson correlations between each feature of the logged dataset and each principal component, we can figure out which of the variables are most powerfully correlated with which component [2]. We set a benchmark correlation value of 0.5 between the principal components and the features of the logged PhEDEx dataset - if the correlation between the feature and the principal component is at 0.5 or greater, we determine it to be strong.

Table 1: *Measure of Pearson Coefficients*

Principal Component		
Features	PC1	PC2
# Files	**-0.9887**	0.0080
Total File Size (GB/s)	**-0.9998**	-0.0024
Rate (MB/s)	**-0.9998**	-0.0027
# Errors	-0.0884	0.1736
# Expired	-0.2201	**0.9668**
Avg. Est. Rate	-0.4047	-0.0512

Each number represents the Pearson Coefficient calculated between the feature and the prinicipal component. A bolded number indicates that the Pearson Coefficient between that particular feature and principal component is significant.

Calculating a Pearson correlation coefficient between two variables involves taking the covariance of the variables and dividing it by the product of their standard deviations [2]. To compute the Pearson coefficients in Table 1, a vectorized numPy script was written to

take the ndarray of logged PhEDEx data, get an ndarray of the principal components using scikit-learn's PCA method and return a matrix of the correlation coefficients as shown in Table 1. The first Principal Component (PC1) is strongly correlated with number of files, total file size and rate of the data transfer, indicating that these three features all vary together. Because these coefficients are negative, this shows that the first principal component increases as the number of files, file size and rate decrease. This corresponds strongly with the scatter plots of the PCA and the Approximate PCA, explaining the clustering of the data points that occurs in both.

VII. Discussion

I. Results

By completing a machine learning based analysis of the logged PhEDEx data, we can determine that there is a strong correlation between the number of files, total file size and rate, since each of their Pearson coefficients is very high. What this means is that all of these features explain a substantial amount of the variance in the transfer data, and they essentially give the same information on the logged data. When implementing a machine learning algorithm into the OpenDaylight controller, there will no longer be a need to consider all three of these variables at once; rather, the controller will be able to optimize transfers through consideration of only a few variables.

II. Future Plans

The next step is to classify the logged data through the K-means clustering algorithm. In order to get a better handle on the PhEDEx data, it is imperative to cluster the unlabeled data and use the results of the PCA to be able to predict certain aspects of them.

Another approach currently being pursued by the CMS group is to create a neural network between a small number of grid sites in the CMS group. Our goal is to train a machine learning algorithm on the grid sites and once it is sufficiently trained, apply the resulting algorithm to the OpenDaylight controller.

VIII. Acknowledgements

This research was supported and fully funded by the Edward C. and Alice Stone Fellowship. We would like to thank both Dr. and Mrs. Stone for their contributions to the CMS experiment and the pursuit of groundbreaking research.

We also would like to thank Dr. Maria Spiropulu and Mr. Dorian Kcira at the California Institute of Technology for their assistance and guidance, without whom this project would not be possible.

References

[1] GL Bayatian, A Korablev, A Soha, O Sharif, M Chertok, W Mitaroff, F Pauss, V Genchev, M Wensveen, V Lemaitre, et al. Cms physics. *J. Phys. G*, 34(CMS-TDR-8-2):995–1579, 2007.

[2] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[3] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[4] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.

[5] Jan Medved, Robert Varga, Anton Tkacik, and Ken Gray. Opendaylight: Towards a model-driven sdn controller architecture. In *2014 IEEE 15th International Symposium on*, pages 1–6. IEEE, 2014.

[6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss,

Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

[8] J Rehn, T Barrass, D Bonacorsi, J Hernandez, I Semeniouk, L Tuura, and Y Wu. Phedex high-throughput data transfer management system. *Computing in High Energy and Nuclear Physics (CHEP) 2006*, 2006.

[9] Yakov Shafranovich. Common format and mime type for comma-separated values (csv) files. 2005.

[10] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[11] B Walczak and DL Massart. Wavelet packet transform applied to a set of signals: A new approach to the best-basis selection. *Chemometrics and intelligent laboratory systems*, 38(1):39–50, 1997.