Identifying Anomalous File Transfer Events in LCLS Workflow

Mengying Yang, Xinyu Liu
University of California, Berkeley
Berkeley, California
mengying_yang,xinyu_liu@berkeley.edu

Wilko Kroeger
SLAC Nat'l Accelerator Laboratory
Stanford, California
wilko@slac.stanford.edu

Alex Sim, Kesheng Wu
Lawrence Berkeley Nat'l Laboratory
Berkeley, California
asim,kwu@lbl.gov

ABSTRACT

This short paper reports our on-going work to study and identify anomalous file transfers for a large scientific facility known as Linac Coherent Light Source (LCLS). We identify the anomalies based on the statistical models extracted from the recent observations of the file transfer events. This data-driven approach could be used in different use cases to identify unusual events. More specifically, we propose two different identification strategies based on the different properties of the observed file transfers. Because these methods capture key aspects of the two different segments of the data transfer pipeline, they are able to make accurate identifications for their respective workflow components. The current anomaly detection algorithms only make use of the file sizes as the primary feature. We anticipate that integrating more information will improve the prediction accuracy. Additional work is planned to validate the identification algorithms on more data and in different use cases.

KEYWORDS

Network; Storage; File Transfer; Workflow Anomaly Detection; Autonomic Management

1 INTRODUCTION

Many recent scientific discoveries such as Higgs boson and gravitational waves are produced from analyzing a huge amount of data collected from large scientific facilities. These scientific facilities typically require a high-speed data network to support the effort to distribute the data to the thousands of scientists around the world. It is important for the infrastructure operators to be able to monitor the health of the system and anticipate potential failure, so as to avoid catastrophic interruption of the operations of the expensive shared facility.

This short paper reports our exploration of the data transfer performance of the LCLS files, and develops algorithms for identifying the unusually slow transfers. The key contribution of this work is the development of the two algorithms for identifying the unusually slow file transfers. Being able to identify abrupt changes during file transfer process has been crucial to administrators because they can investigate on the cause of the issue in a timely manner. We aimed to take statistical approaches to help identify unusual behavior during the sequential data generative process in an automated fashion. These algorithms take advantages of some basic characteristics of the network data transfers, and do not make other assumptions about behavior of the data. We anticipate these anomaly detection algorithms are useful in other cases involving file transfers to help administrators monitor file transfer process, even though in this work, we exercise these algorithms only with a set of monitoring data from the LCLS system.

2 LCLS

The Linac Coherent Light Source (LCLS) [1] at the SLAC National Accelerator Laboratory provides a X-ray source that enables the study of fundamental processes of chemistry, physics, biology and technology. LCLS has seven different instrumental stations each with different detectors and X-ray beam characteristics allowing for diverse types of experiments. An Experiment lasts typically for five days split into five 12 hour shifts of beam time. During data collection it is critical for the experimenters to process the data quickly in order to check the quality and make decisions about the ongoing collection process. Access to the data is facilitated by two storage systems. The first one is the fast feedback storage (FFB) which provides fast, low latency access to the collected data. It is only used by the active experiments. The second system is the analysis storage system (ANA). It is large in size (4PB), shared between all experiments and holds the experimental data for many month. It is used for data analysis after experiment finished but also by active experiment for non time critical data processing.

Fig. 1 shows an overview of the data flow for the LCLS data. The data acquisition (DAQ) distributes detector data to multiple nodes (DSS) and writes them to files on these nodes. The data mover copies the files from the DSS nodes to the FFB (FFB transfer) and in a subsequent step from the FFB to the analysis file system (ANA transfer). The DAQ writes to multiple files in parallel (one per DSS nodes) and therefore the FFB and ANA transfers also copy files in parallel. Typically 5-6 files are transferred in parallel. For the FFB transfer the files are transferred while they are written to the DSS and the transfer rate is limited by the data acquisition rate (< 200MiB/s per file). The ANA transfer copies a file only after it has been completely written to the FFB. The maximum rate for the ANA transfer is limited to about 400-450 MB/s due to checksum calculations.

The LCLS file transfer dataset contains 258,765 observations with 10 variables. The variables used in our study are the start and stop time of a file transfer (epoch time in seconds), file transfer rate (MiB/sec), file size (gigabytes), a boolean variable of whether it is an FFB or ANA transfer (ffbtrans). Variables that were ignored included the name of a file, LCLS instrument the data was collected with, file system the data were written to, and more. The FFB transfers accounts for 131,274 observations and the ANA transfer for 127,491

AI-Science'18, June 11, 2018, Tempe, AZ, USA

ACM ISBN 978-1-4503-5862-0/18/06…$15.00
https://doi.org/10.1145/3217197.3217203

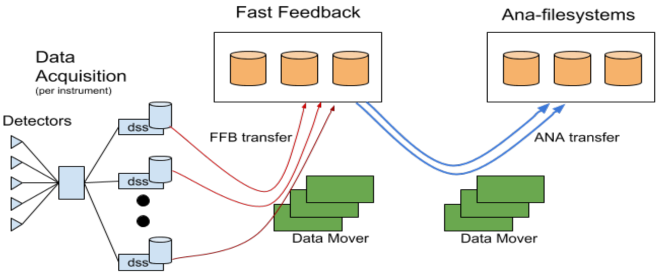


Figure 1: LCLS Data flow. The red (dss->ffb) and blue (ffb->ana) lines show the data transfers by the LCLS data system.

STATS	size (gigabyte)	transfer rate(MB/s)
median	4.0	47.9
mean	13.8	81.2
std	22.1	91.0
min	0	0
max	1304.14	498.2

(a)

STATS	size (gigabyte)	transfer time (MB/s)
median	4.0	342.7
mean	13.29	301.8
std	21.49	109.7
min	0	0
max	1304.14	522.2

(b)

Table 1: (a)DSS to FFB (b)FFB to ANA

observations. Table 1 shows the distribution of transfer rate and file size for both FFB and ANA transfers.

These general descriptive statistics lead us to identify unusual transfer behaviors; extremely large files, zero file size, and slow transfer rate. It shows that both file sizes distribution from FFB and ANA are extremely right skewed because the median file size is considerably smaller than the mean file size. Based on our investigation, the skewness is caused by 12 files (6 files in each dataset) whose file size is much larger than the 100 GiB that is enforced by the data acquisition control. These transfers happened within the same day and are due to a configuration error. For our analysis, we removed these files to have a more accurate analysis.

While exploring the file transfer records, we found 76 files with zero file size and zero transfer rate. These files were likely produced by some failure in the data acquisition system, but are likely not useful for identifying other types of slow file transfers. Thus in the latter discussions, we removed these data records.

For both the FFB or ANA transfers, the definition of the slow file transfer rate in our project is the transfer rate that is lower than 1 percentile of all the transfer rates for files that is larger than 1GB in the dataset of FFB or ANA respectively. We chose to focus on files that are larger than 1 GB since large files usually transfer faster, and the slow transfers would therefore be an indicator for unusual system performance. The 1 percentile transfer rates for FFB and ANA are 4.3MB/s and 45.7MB/s respectively. There are 912 and 872 observations of the very slow transfer rate data from FFB and ANA respectively.

3 ANOMALY DETECTION ALGORITHM

In this section, we discuss the methods we proposed to detect the slow data transfers. Our work relies heavily on the file transfer rate reported by the data transfer engine, bbcp, as the indicator of the system performance. Our algorithm is designed to detect slow transfers in real time, which could be used to generate alerts to administrators so that they can check if there is any problems within the system. The threshold of the slow transfer rate threshold is not constant. Many uncertainty element will influence the file transfer rate in the future and change the threshold. Thus, instead of setting a certain threshold, we hope to detect the anomaly in the real time.

Section 3.1 and 3.2 introduce two methods we developed to systematically and dynamically detect the slow transfers. Method 1 in Section 3.1 builds a model that predicts the minimum transfer time based on file transfer size plus a stochastic model for the congestion caused by interference from other programs that are accessing some part of the storage, file system, CPU and networking system that are serving the current file transfer. This model takes advantage of the fact that congestion only increase the data transfer time and is therefore specifically for predicting file transfer time. Method 2 in 3.2 finds the unusual slow transfer rate based on the past 5000 data transfers, but with some adjustments to keep the historical data information. Both methods have its pros and cons under different situation. The next section compares those results.

3.1 Model-Based Detection Method

We started with the idea of using the transfer size and transfer time to detect slow transfers. If given a file size, the actual transfer time takes much longer than the predicted transfer time, we would consider this file transfer as an anomaly. Here the predicted transfer time corresponds to the minimum transfer time (base time) and we can use quantile regression with size as covariate to predict its base time [2]. Then we can check whether the percentage of change between predicted time and actual transfer time is above certain threshold.

Since the sizes spread across many orders of magnitudes and are highly skewed, we applied *log*2 transformation of both size and transfer time in an attempt to reduce the impact of large files. At this initial stage, the algorithm worked as follows, we firstly used the first a file as training points for the B-spline minimum quantile regression with *log*2 of size as covariate and *log*2 of transfer time as response. [3] We used the trained model to predict this a numbers of files to get the first segment of base time. Then the difference between the files' actual transfer time and predicted base time will be the error, and the error percentage is simply the error divided by the predicted base time. We predicted the next segment with h number of files. We appended the error percentage of this segment to the initial error percentage and get the q percentile of error. Any files with error percentage larger than q would be declare as anomaly points. We refitted the B-spline Quantile regression at every segment with new data appended to the old. The choice of those parameters varied by the data set being studied. For FFB transfers, we set a to be 1000, which means we accumulated first 1000 points as training points, h to be 200, which means each segment contains

Algorithm 1 Model-Based Detection

```
$y\{0j\}$ ← log transfer time for file $j$ *for first a files*
$s\{0j\}$ ← log transfer size for file $j$ for first $a$ files
Apply BSpline quantile regression with $s\{0\}$ as covariate and $y\{0\}$ as response to get $\hat{y}_0$
$e\{0\} \leftarrow (2^{y\{0\}} - 2^{\hat{y}_0})/2^{\hat{y}_0}$ *j for first a files*
**for** *each segment i of length h* **do**
    $y\{ij\}$ ← log transfer time for file $j$ in Segment $i$
    $s\{ij\}$ ← log transfer size for files $j$ in Segment $i$
    Apply BSpline quantile regression to estimate to get $\hat{y}\{ij\}$in Segment $i$
    $e\{ij\} \leftarrow 2^{y\{ij\}} - 2^{\hat{y}_{\{ij\}}}/2^{\hat{y}_{\{ij\}}}$ $j$ in Segment $i$
    Append $e\{ij\}toe\{0\}$
    $threshold$ ← *q percent quantile of e{0}*
    **if** $e\{ij\} \geq threshold$ **then**
        State file $i$ as an anomaly
    **end if**
**end for**
```

Algorithm 2 Distribution-Based Detection

```
$data$ ← first m file transfer data
$threshold\{0\}$ ← 0.1 percentile of first m points collected
$mean\{0\}$ ← mean of the first m file transfer rate
$data\_copy$ ← first m file transfer rate data
**for** *each new file record i* **do**
    $mean\{i\}$ ← mean of file transfer rate from record i-m to i
    $dis\{i\}$ ← mean{i} - mean{i-1}
    **if** $dis\{i\} \geq (75 percentile of all dis)$ **then**
        $threshold\{i\}$ ← mean of all previous threshold
        $data\_copy\{i\}$ ← mean of all previous threshold
    **else**
        $threshold\{i\}$ ← q percentile of data_copy from i-m to i
        $data\_copy\{i\}$ ← new file transfer rate i
    **end if**
    **if** $threshold\{i\} \geq newfiletransferrate\{i\}$ **then**
        State file $i$ as an anomaly
    **end if**
**end for**
```

200 files, and q to be 99, which means the threshold is set to be above 99 percent quantiles of errors.

For ANA transfers, we set a to be 4000, which means we accumulated first 4000 points as training points, h to be 200, which means each segment contains 200 files, and q to be 99.95, which means the threshold is set to be above 99.95 percent quantiles of errors.

3.2 Distribution-Based Detection Method

The computation cost of the prediction in method 1 is high, to decrease the cost we developed method 2. Since the baseline in method 1 is consistent we can interpret the slope of the base line as the inverse of the maximum file transfer rate. Thus, the distribution of the error percentage is highly correlated with the distribution of the file transfer rate. Based on this interpretation, the method 2 directly uses the file transfer rate distribution of the past 5000 file transfers and sets the 0.2 percentile as the threshold. Based on our investigation, 5000 data will let the distribution of file transfer rate become relatively stable. To avoid the loss of historical information due to the influence of the large fluctuations of transfer rates, we kept tracking the mean of the 5000 transfer rates. When adding a new observation increases the new mean by more than the 75 percentile of the mean difference distribution, we set this observation's rate as the mean of the all threshold . This way, we not only keep tracking the changing of the file transfer rate, but also keep the historical data information and make comparison.

As for parameters, we set m to be 5000, which means we use the previous 4999 file transfer rate and the new transfer rate to find the threshold of slow transfer. We set q = 0.2, which means the threshold is the 0.2 percentile of the data_copy for both ANA and FFB transfers.

4 RESULT DISCUSSION

We applied the two algorithm separately to the FFB and ANA transfer. Since we do not want to send alert so frequently, we aggregated the result by hour. In a given hour, if there is one or more than one anomaly point detected, we will send an alert this hour. To test whether the algorithms have predictive power, we checked whether the hour we send alerts contain one percent slow transfer rate. There are two metrics we used to evaluate the model performance[4]: recall and precision. Recall is the proportion of hours that actually contain slow transfer rates and were detected by the model as an hour containing slow transfer rates. Precision is the proportion of hours that were predicted as an hour containing slow transfer rates and actually were containing slow transfer rates.

Files transfer to FFB:. Figure 2 is the comparison plots for files transfer to FFB. The first plot shows the position of the actual slowest one percent rate. The second and third are the anomalies detected from the model-based method and distribution-based method predicted respectively. Visually, the plots are almost identical.

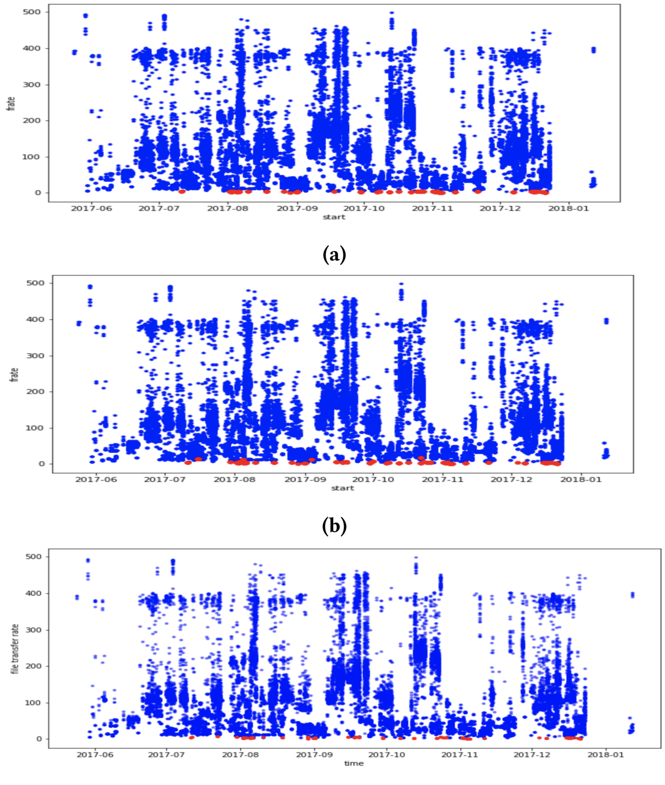
There are 2526 hours in FFB transfers and 120 hours that contain slow transfer rates. The model-based detection algorithm detected 129 hours containing slow transfer rates and 109 hours were matched with the hours actually containing slow transfer rates. Thus, the precision for this algorithm is 84.5%. The recall is 90%.

The distribution-based detection algorithm detected 71 hours for which an alert should be sent and there are 64 hours that contain slow transfer rates. Thus the precision for this algorithm is 90.14%. However, the recall is only 53.3%.

Files transfer to ANA:. Figure 3 is the comparison plots for the ANA transfers. Visually, the plots are similar but not as identical as the other dataset.

There are 2598 hours in total and 97 hours contain the slow transfer rates. The model-based detection algorithm detected 20 hours with anomaly points and 18 hours matched with the slowest one percent rate. Thus, the precision is 90%, The recall is 18.6%

The distribution-based detection algorithm detected 50 hours for which an alert should be sent and 47 hours that actually contain slow transfer rates. Thus, the precision is 94% and the recall is only 48.45%.

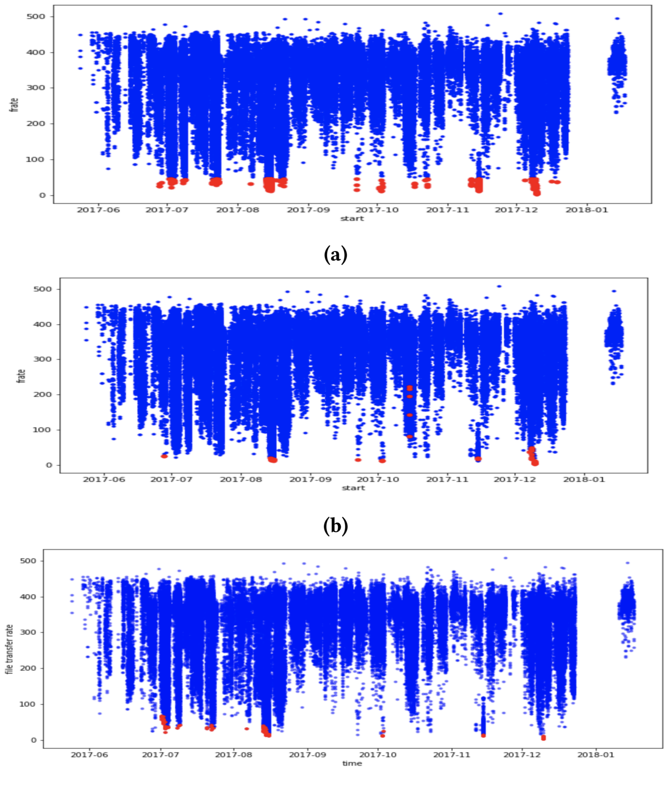


(a)

(b)

(c)

Figure 2: (a)Slowest one percent rate in red with real data in blue (b)Anomalies predicted in red with real data in blue for model-based method(c)Anomalies predicted in red with real data in blue for distribution-based method



(a)

(b)

(c)

Figure 3: (a)Slowest one percent rate in red with real data in blue (b)Anomalies predicted in red with real data in blue using model-based method (c)Anomaly points predicted in red with real data in blue using distribution-based model

Comparison: Model-based method performs better for the FFB transfers and worse for the ANA transfers since this method works better when the slowest transfer rate is stable and the transfer behavior is unstable for consecutive transfers. Distribution-based method performs better for files transfer to ANA since this dataset has a stable consecutive transfer behavior but unstable lowest rate. In addition this distribution-based method is computationally less expensive.

5 SUMMARY

Our key objective is to identify unusual file transfers in the LCLS data system. The initial data exploration helped us identify files with zero size and extremely large files being transfered. We explored the events involving slow file transfers as a symptom of failures that might require the attention of system administrators. We proposed two methods to detect slow transfers, one based on a performance model and another based on the observed distribution of file transfer rates. From the tests, we observed that model-based method works better for transfers to FFB, while the distribution-based method works better for transfers to ANA.

We plan to combine the two methods so that the detection algorithm can be generalized and have a consistent behavior. Another challenge is to incorporate other variables that can potentially influence the transfer rate. For example, the instrument the file transfer belonged to may influence the transfer rate. We also plan to add new variables to the data set in particular performance parameters of the file systems the transfers read from and write to. This will require further statistical testing to determine that usefulness in detecting anomalous events.

ACKNOWLEDGMENT

This work was supported by the Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center. Use of the Linac Coherent Light Source (LCLS), SLAC National Accelerator Laboratory, is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515.

REFERENCES

[1] "Data systems for the linac coherent light source," *Advanced Structural and Chemical Imaging*, vol. 3, no. 3, 2017.

[2] Ł. Komsta, "Comparison of several methods of chromatographic baseline removal with a new approach based on quantile regression," *Chromatographia*, vol. 73, no. 7, pp. 721–731, Apr 2011.

[3] D. Suhubdy, "An online algorithm for a low cost real-time sensor-based fenceline leak detection system," https://www.researchgate.net/publication/299172324_An_Online_Algorithm_For_A_Low_Cost_Real-Time_Sensor-Based_Fenceline_Leak_Detection_System, 03 2016.

[4] J. H. M. Kamber, *Data Mining concept and techniques.* San Francisco: Morgan Kaufmann Publishers, 2001.