Detecting Anomalies in the LCLS Workflow

Tal Shachaf
Dept. of Mathematics and Dept. of Computer Science
University of California, Berkeley
Berkeley, California
tsshachaf@berkeley.edu

Alexander Sim
Computational Research Division
Lawrence Berkeley National Laboratory
Berkeley, California
asim@lbl.gov

Kesheng Wu
Computational Research Division
Lawrence Berkeley National Laboratory
Berkeley, California
kwu@lbl.gov

Wilko Kroeger
SLAC National Accelerator Laboratory
Stanford University
Menlo Park, California
wilko@slac.stanford.edu

***Abstract*—The Linac Coherent Light Source (LCLS) located at SLAC National Accelerator Laboratory has been essential to over 1023 publications since 2009. The LCLS produces vast quantities of data - thousands of gigabytes per experiment. The data must be analyzed and stored at large data centers to be available to the world-wide user community. Due to the vast quantities of data flowing through the network, many abnormal data transfers remain unnoticed. This work focuses on identifying network failures that could slow down the data transfer process. This work aims to develop a diagnostic tool to detect when network transfers become anomalously slow. The tool uses an algorithm based on the hampel filter to detect poor performance and alert SLAC administrators to bottlenecks in each phase of the workflow. We will describe our experience of preparing the data and modifying the hampel filter to enhance its effectiveness. We found that applying a heuristic to the algorithm in conjunction with parsing the data along key features improved performance.**

***Index Terms*—LCLS, Linac Coherent Light Source, Hampel Filter, Network Anomaly Detection, Data Transfers**

I. Introduction

The Linac Coherent Light Source (LCLS) is a linear accelerator at SLAC National Accelerator Laboratory that has been essential to over 1023 publications since 2009. Each experiment produces thousands of gigabytes, and SLAC expects the upgraded LCLS-II will produce 10000 times as much data in the same time frame, reaching exascale workflow[1]. LCLS data must be transferred to NERSC supercomputers for analysis before it can be used by the PI and world-wide community. Network failure can bottleneck the workflow, decreasing data quantity and quality. Data shows that LCLS network transfers slow to as much as 1/30th of their peak rates, significantly affecting processing times. Without analyzed data for feedback, PIs cannot adjust initial laser settings, which can be inaccurate and result in undesired measurements. The vast quantity of data makes failure difficult to detect through human monitoring, and with the LCLS-IIs increased data flow network limitations will have more severe impacts on the workflow. Therefore, we are researching a scalable diagnostic tool based on the hampel filter to send alerts of anomalously slow transfers in the LCLS workflow.

A typical LCLS data goes through three phases we call FFB, ANA and NERSC. First, during the FFB phase, the seven LCLS instruments record images in fast feedback (FFB) storage. Data movers then transfer the images to one of four analysis (ANA) file systems. Lastly, in the NERSC phase, the images are uploaded through ESnet to NERSC storage for a more thorough analysis.

For each phase, we study the attributes that affect the data transfer rate. This allows us to contextualize our anomaly detection algorithm and enhance its precision. For the algorithm we investigated approaches based on the z-score [2], interquartile range [3], and hampel filter [4]. To account for performance shifts over time, we applied a sliding window over the average data transfer rates. However, we found that under equivalent parameters, the z-score and interquartile range schemes had difficulty detecting anomalies in the highly-varying FFB dataset.

To alter the hampel filter's analysis for a specific dataset, at least half of the data must be altered. The interquartile range is only half as robust, while one alteration will affect the z-score. Therefore, we focused on the more robust hampel filter for further investigation. Furthermore, we began researching an adaptive anomaly detection algorithm that heuristically determines the parameters for the hampel filter. Our goal is to make the algorithm flexible against future alterations in data procurement methods or LCLS performance with minimal human adjustment. We design a tool to detect the most severe

This work was prepared in partial fulfillment of the requirements of the Berkeley Lab Undergraduate Research (BLUR) Program, managed by Workforce Development & Education at Berkeley Lab. 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.

[1]An exascale workflow is computationally intensive enough to make effective use of exascale computers, which will compute 10^{18} floating point operations per second.

anomalies, so as to prevent major bottlenecks and ensure SLAC has the resources to respond to the algorithm's alerts.

Previous work [1] investigated errors in the LCLS data and identified the slowest 1% of all transfers. We furthered the work by studying data characteristics and applying statistical techniques to detect anomalies. Preparing the data and modifying the technique according to correlated features enhanced the alert system's precision. Improving the algorithm minimizes usage of human resources while improving detection of workflow bottlenecks resulting from network failure.

II. Materials & Methods

SLAC provided statistics on data transfers from June, 2017 to January, 2018 spanning the FFB, ANA and NERSC phases. The FFB and ANA datasets include information on the LCLS instrument, ANA filesystem, data mover, time began, time elapsed and byte size per entry. The NERSC transfers note the time began, time elapsed and data quantity per entry. The transfer length includes checksum timing, which skews transfer rates to be slower, as we will discuss further in the results section of the paper. Each experiment is assigned a LCLS instrument and linked to an ANA filesystem. For each data transfer, the total time elapsed and data quantity allows us to calculate the average transfer rate. We isolated transfers from 1 GB to 100 GB in size, as smaller files are relatively insignificant and configuration settings limit files to less than 100 GB. [1]

We focused on the hampel filter because its robustness makes it more effective. The hampel filter, the pseudocode for which is depicted in Alg. 1, utilizes the median of a dataset to estimate the standard deviation. It relates the median distance from the median to the standard deviation through a scalar proposed by its developer, Ronald K. Pearson. Due to making estimates from medians, its breakdown point [6] is 0.5, a quantification indicating half of a dataset must be altered to alter the filter's calculations. The z-score and interquartile range methods have respective breakdown points of 0 and 0.25, indicating they are less robust. We attribute the robustness to why the hampel filter managed to detect anomalies in the FFB data, which was too scattered for the other methods to detect anomalies. Accordingly, we chose to focus on the hampel filter for our research.

As seen in Alg. 1, the anomaly detection schemes make use of 2 parameters: an integer window length, half-window, and an aggressiveness value. The window size determines how many data points in either direction to compare each data transfer with, while the aggressiveness determines how many standard deviations from the median a transfer must be to be considered anomalous. The lower the aggressiveness parameter, the higher the minimum threshold for transfer rates will be and more data will be detected.

The algorithm iterates through the data, and for each data entry it examines the half-window data points before and after it. For each such window of data points, the algorithm calculates the median and median absolute deviation from the median (MAD). The MAD is multiplied by 1.4826 to estimate the standard deviation.

Algorithm 1 Pseudocode for calculating anomaly thresholds with the hampel filter.

Precondition: dataframe $data$ with column $transfer\_rate$, positive integer $half\_window$

function HAMPEL($data, half\_window, aggressiveness$)
  $file \leftarrow data.transfer\_rate.copy()$
  initialize $threshold$
  for $i \leftarrow 1$ to $length(file)$ **do**
    $initial \leftarrow i - half\_window$
    $terminal \leftarrow i + half\_window$
    $win \leftarrow file[initial : terminal]$
    $median \leftarrow median(win)$
    $deviations \leftarrow [|rate - median|$ for $rate$ in $win]$
    $MAD \leftarrow median(deviations)$
    $standard\_dev \leftarrow MAD * 1.4826$
    $tolerance \leftarrow standard\_dev * aggressiveness$
    $threshold\{i\} \leftarrow median - tolerance$
    if $|median - file\{i\}| >= tolerance$ **then**
      $file\{i\} \leftarrow median$
  return $threshold$

To improve scalability, we implement the hampel filter to internally store its sliding window as a sorted list [5]. Doing so allows for constant median lookup, linear threshold calculation and linear value replacement.

The aggressiveness parameter specifies the number of standard deviations from the median that will be considered acceptable. If the transfer is slower than expected, then it is detected as an anomaly. Additionally, if it is faster or slower than the normal range, then it is substituted by the median for sliding windows of other file transfers. Doing so prevents outliers from affecting calculations, further making the hampel filter more robust than alternative techniques.

However, despite the hampel filter's robustness, its parameters must still be adjusted for different datasets, hindering autonomous anomaly detection. Adjusting parameters to maintain anomaly alerts within meaningful yet manageable levels requires human intervention. Therefore, we investigated the effect of applying heuristics to adapt to the datasets. Heuristically determining the aggressiveness parameter allows the algorithm to parse datasets by correlated variables and run separate analyses. We expect contextualizing datasets in such a fashion to lower the volume of alerts while preserving precision.

III. Results

A. Data Characteristics

To prepare the data, we began investigating the characteristics of the FFB, ANA and NERSC datasets for correlations. We began by investigating the FFB dataset, as the hampel filter showed significant difficulty detecting anomalies in the FFB transfers. The FFB data showed much higher standard

deviation estimates than the ANA data even though its range was lower. While investigating the FFB dataset, we coalesced each data point by average transfer rate and measured its frequency of occurrence, as can be seen in Fig. 1. We observed that, as we anticipated from having distinct LCLS instruments, the dataset concentrates at a couple frequent transfer rates. However, the transfer rates were not concentrated temporally, (Fig. 2) as they ought to be if the instruments were the bottlenecks. Typically, only a few of the seven LCLS instruments are run simultaneously, so the data should have shown horizontal strips. Indeed, preparing the data by instrument preserved multiple frequent transfer rates, such as instrument CXI's data transfers in Fig. 3. The frequent transfer rates were preserved, indicating the bottlenecks are independent of the instrument in use. Therefore, We hypothesize the FFB transfers are bottlenecked by an unrecorded feature. Specifically, we suspect that swappable detectors used in conjunction with the instruments are creating bottlenecks at different average transfer rates.

To verify our hypothesis, we need data on any correlations between the average transfer rate and detector. Otherwise, we cannot properly distinguish slow transfers caused by network failure from transfers limited by slower detectors. Thus, we cannot effectively apply the algorithm to create an alert system for FFB transfers.

The ANA dataset contains multiple features, of which we concluded the ANA filesystem to be most meaningful. On the other hand, the NERSC dataset records fewer features, so we could not prepare the dataset by a correlated feature. Instead, we prepared the data to more accurately depict the network transfer rate. NERSC data transfers are fast enough that checksums significantly affected average transfer rates, so we extrapolated a function to estimate checksum timing from the data files size. We anticipated removing the time spent checksumming would result in a more accurate distribution of the transfer rates. Indeed, the change can be observed in Fig. 4. Originally, 99.9% of the data transfers were slower than 500 MB/s. After removing checksums though, only 45.8% of transfers were slower than 500 MB/s. The range must be extended as high as 1 GB/s to encompass 96% of data transfers. And since slower transfers rates were less affected by checksums[2] the data distribution is noticeably more scattered than before. As a result, the hampel filters standard deviation estimates significantly increased, demonstrating the difficulty in optimizing the anomaly detection algorithms parameters for different data distributions.

B. Anomaly Detection Algorithm

As seen from the effect checksums had on the NERSC dataset, the algorithm can be significantly affected by changes to data collection procedures. Investigation showed that different LCLS datasets have different optimal aggressiveness parameter values.[3] For an incorrectly set parameter, the algorithm

[2]The longer total time elapsed made the checksum times less significant.

[3]We use optimal to refer to the integer parameter value that will only detect the most significant anomalies in the dataset.

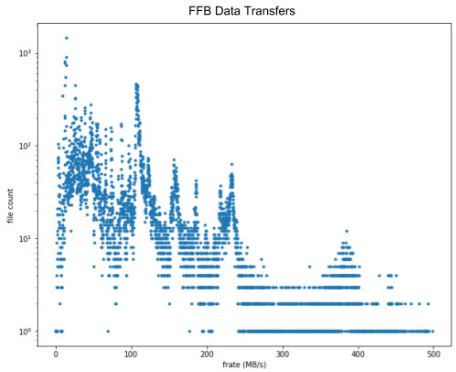

Fig. 1. Shows the FFB dataset plotted by average transfer rate versus number of occurrences. The data shows multiple peaks, indicating there are distinct transfer rates at which FFB transfers bottleneck.

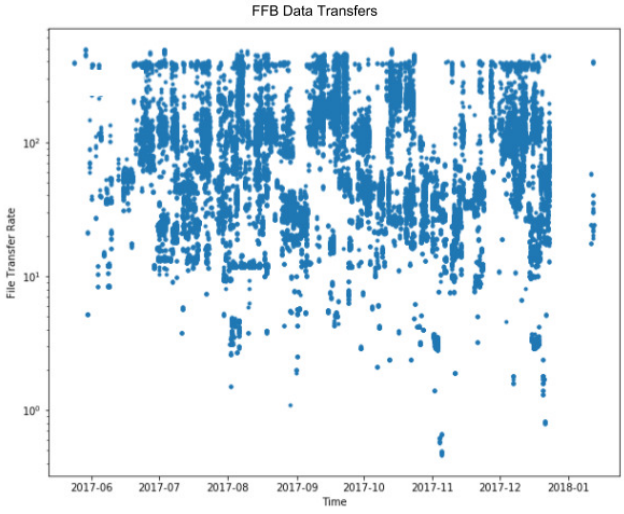

Fig. 2. Shows the FFB dataset graphed by average transfer rate against time. The data is scattered with no clear grouping or correlation, resulting in high standard deviations in the hampel filter.

will detect extended periods of transfers as anomalies. For the 4 ANA filesystems, the optimal parameter value ranges from 4 to 7, while for the NERSC dataset it is as low as 2. Applying a heuristic to estimate this optimal value allows the algorithm to detect only the most statistically significant anomalies of each dataset, thus decreasing alerts while maintaining precision. We applied a rudimentary heuristic, with pseudocode in Alg. 2, to determine the aggressiveness parameter for each of the 4 ANA filesystems separately, with the results shown in Fig. 5. In comparison to analyzing the 4 filesystems' dataset with the same parameters, the heuristic detects a group of previously missed slow transfers in August and ignores a high-performing group of transfers in September. Additionally, when there is sustained network failure, the heuristic picks a smaller subset of the transfers, resulting in fewer yet more precise alerts.

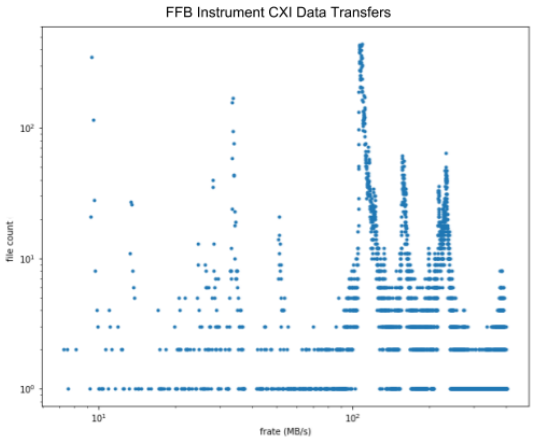

Fig. 3. Depicts the portion of the FFB dataset from instrument CXI. The dataset plotted by average transfer rate versus number of occurrences. The data still shows multiple peaks, indicating the FFB instruments are not the cause of the bottleneck when transferring data from the LCLS to FFB storage.

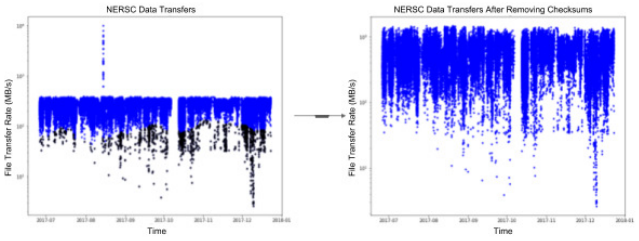

Fig. 4. The left graph shows the results of running the hampel filter on the NERSC data with aggressiveness value 2. The black points are transfers that were detected as anomalously slow. In the right graph, the NERSC data's average transfer rates were recalculated to exclude the time spent checksumming the transfer. Under identical parameters, no anomalously slow transfers were detected due to the change in the distribution.

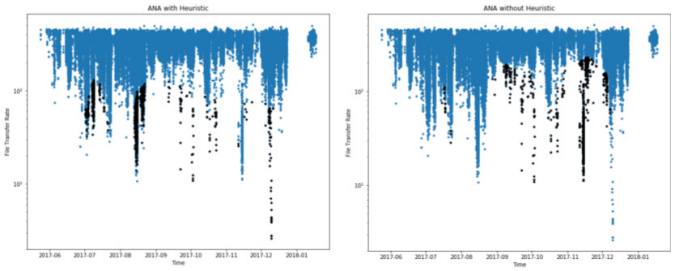

Fig. 5. Contrasts analyzing the ANA dataset with the adaptive algorithm, depicted on the left, and the hampel filter depicted on the right. The heuristic results in fewer total detections and detects slower data transfers.

Algorithm 2 Rudimentary heuristic to calculate aggressiveness parameter

Precondition: dataframe $data$ with column $transfer\_rate$, positive integer $half\_window$. Can duplicate $data$ to include the $half\_window$ earliest and latest points.

function HEURISTIC($data, half\_window$)
  $file \leftarrow data.transfer\_rate.copy()$
  $aggressiveness \leftarrow 0$
  for $i \leftarrow 1$ to $length(file)$ **do**
    $initial \leftarrow i - half\_window$
    $terminal \leftarrow i + half\_window$
    $win \leftarrow file[initial : terminal]$
    $median \leftarrow median(win)$
    $deviations \leftarrow [|rate - median|$ for $rate$ in $win]$
    $MAD \leftarrow median(deviations)$
    $standard\_dev \leftarrow MAD * 1.4826$
    $tol \leftarrow |median - file\{i\}|/standard\_dev$
    $aggressiveness \leftarrow \lceil max(aggressiveness, tol) \rceil$
  return $aggressiveness - 1.5$

IV. Discussion

Enhancing anomaly detection will improve the quantity and quality of data produced and released by principal investigators. Applying statistical techniques makes anomaly detection less dependent on human input for qualitatively determining what transfer rates qualify as network failure. Additionally, it ignores fast network transfers, as opposed to alternative schemes such as detecting the slowest 1% of transfers.

Two areas for future research are the datasets and heuristic. If data collection incorporates more features, such as the FFB instruments' detectors, then one might find other prominent features. Preparing the data using those features can enhance analysis. Furthermore, the current heuristic iterates over the aggressiveness parameter to choose the most deviated anomalies. Improving the heuristic and applying it to the other parameter, the length of the data window, may enhance the algorithm

V. Conclusion

Currently, network failure often remains undetected in the LCLS workflow. The parallel dataflows will bury the bottleneck and hinder human detection. The hampel filter-based algorithm described in the paper is meant to alert SLAC administrators to network failure so they can diagnose and fix bottlenecks, thus speeding up the workflow. Currently, we plan to periodically run the algorithm on newly generated LCLS data and generate alerts for anomalous transfers.

References

[1] M. Yang, X. Liu, W. Kroeger, A. Sim, K. Wu. 2018. Identifying Anomalous File Transfer Events in LCLS Workflow. 1-4. 10.1145/3217197.3217203.
[2] C. Aggarwal. Outlier Analysis. Second ed., Springer, 2017.
[3] J. Peat, B. Barton. 2014. Medical statistics : a guide to spss, data analysis and critical appraisal. Retrieved from https://ebookcentral.proquest.com

[4] R. K. Pearson, Y. Neuvo, J. Astola, M. Gabbouj. EURASIP J. Adv. Signal Process. (2016) 2016: 87. https://doi.org/10.1186/s13634-016-0383-6

[5] M. Suomela. Median Filtering is Equivalent to Sorting. arXiv:1406.1717 [cs.DS]

[6] F. Hampel. 1974. The Influence Curve and Its Role in Robust Estimation. Journal of the American Statistical Association, 69(346), 383-393. doi:10.2307/2285666