How Do ML Jobs Fail in Datacenters? Analysis of a Long-Term Dataset from an HPC Cluster

Xiaoyu Chu
Vrije Universiteit Amsterdam
the Netherlands
x.chu@vu.nl

Sacheendra Talluri
Vrije Universiteit Amsterdam
the Netherlands
s.talluri@vu.nl

Laurens Versluis
Vrije Universiteit Amsterdam
the Netherlands
l.f.d.versluis@vu.nl

Alexandru Iosup
Vrije Universiteit Amsterdam
the Netherlands
a.iosup@vu.nl

ABSTRACT

Reliable job execution is important in High Performance Computing clusters. Understanding the failure distribution and failure pattern of jobs helps HPC cluster managers design better systems, and users design fault tolerant systems. Machine learning is an increasingly popular workload for HPC clusters are used for. But, there is little information on machine learning job failure characteristics on HPC clusters, and how they differ from the previous workload such clusters were used for. The goal of our work is to improve the understanding of machine learning job failures in HPC clusters. We collect and analyze job data spanning the whole of 2022, and over 2 million jobs. We analyze basic statistical characteristics, the time pattern of failures, resource waste caused by failures, and their autocorrelation. Some of our findings are that machine learning jobs fail at a higher rate than non-ML jobs, and waste much more CPU-time per job when they fail.

CCS CONCEPTS

• **Computer systems organization** → **Reliability**.

KEYWORDS

job failure, machine learning, HPC datacenters, failure characterization, time correlation failures, reliability

ACM Reference Format:
Xiaoyu Chu, Sacheendra Talluri, Laurens Versluis, and Alexandru Iosup. 2023. How Do ML Jobs Fail in Datacenters? Analysis of a Long-Term Dataset from an HPC Cluster. In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE '23 Companion), April 15–19, 2023, Coimbra, Portugal*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3578245.3584726

1 INTRODUCTION

Reliable job execution is a cornerstone of managing High Performance Computing (HPC) facilities [5, 12]. Failed jobs in HPC datacenters waste researchers' time, compute resources, and can delay research outcomes. Understanding the pattern and distribution of failures helps design resilient systems [4, 7] and understand the impact of failure on existing applications. Machine learning applications are an important part of an HPC system's workload [3]. But, there is little information available on their execution and failures in HPC datacenters. Towards solving this paucity, we present the first study that characterizes ML job failures in an HPC cluster.

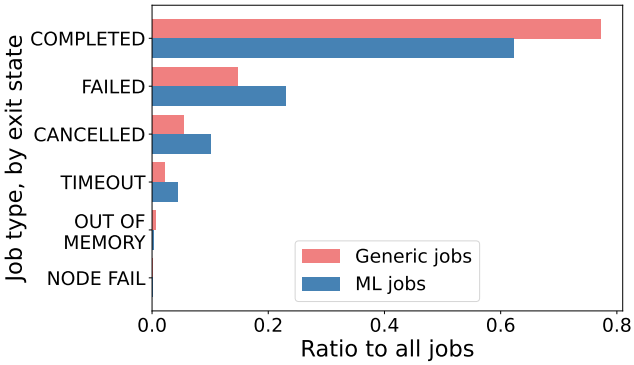


Figure 1: Distribution of generic and ML jobs by their final exit state.

Wasted machine time due to unsuccessful executions accounts for an astonishing 65% of the total machine time, and wasted used CPU, RAM, and DISK demands are roughly 56%, 67%, 70% of the total used demands [10]. Previous failures characterization studies [1, 2, 9–11, 14, 15] try to analyze unsuccessful jobs observed in datacenters, using cluster data collected by companies such as Google [8]. Researchers conduct statistical analysis on metrics such as the number of jobs, failure rate[15], CPU/Disk/Memory usage[1]. However, these studies do not consider the special characteristics of different kinds of workloads such as machine learning workloads.

Machine learning workloads are fast-emerging workloads in HPC datacenters. Table 1 shows machine learning jobs account for 13.32% of all kinds of jobs in our dataset. However, machine learning jobs have a 10% lower completion rate compared to generic jobs, which is different from other types of jobs (see Figure 1). So it is important to understand machine learning job failure characteristics in order to mitigate their negative impact on system resource usage and application performance.

ICPE '23 Companion, April 15–19, 2023, Coimbra, Portugal

ACM ISBN 979-8-4007-0072-9/23/04…$15.00
https://doi.org/10.1145/3578245.3584726

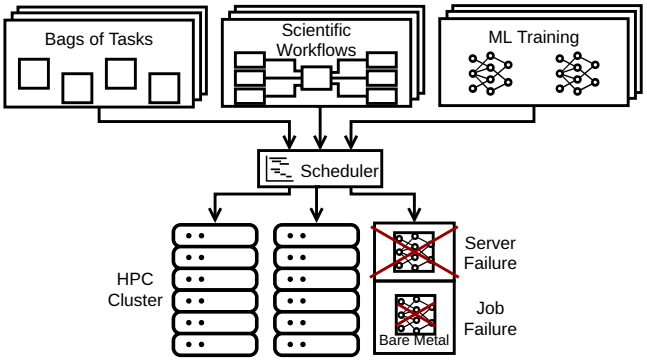


Figure 2: System model of fail-stop HPC failures.

In this paper, we aim to answer the following main research question: What are the characteristics of failed machine learning jobs in HPC datacenters? Our study is based on traces collected from the Lisa cluster at the Dutch National Supercomputing Center, SURF. The traces contain job information for 12 months, for the whole period from January to December 2022.

Towards understanding the characteristics of machine learning job failures, we make a threefold contribution in this work:

(1) We implement a toolbox for collecting, cleaning, and analyzing job data from SLURM scheduler logs. We propose a method to compare the failures of ML jobs with non-ML jobs (in Section 3).
(2) We identify differences in submit rate, job duration, time pattern, and resource use between ML jobs and non-ML jobs (in Section 4).
(3) We analyze the time correlation of job failure events observed in machine learning workloads (in Section 5).

2 SYSTEM MODEL

We characterize fail-stop failures in a national scale HPC system in this work. Figure 2 depicts the system model we use. The jobs we characterize were submitted to the Lisa HPC cluster. The HPC cluster is composed of many racks, each of which accommodates multiple servers. The servers are connected to each other through a high performance network interconnect.

The HPC cluster is used for different kinds of jobs including bags of tasks, workflows, and machine learning training jobs. The jobs are submitted to a SLURM scheduler which then schedules them onto the servers of the cluster. A job can use a single server or multiple servers.

We use a fail-stop model for job failures. A job is considered to have failed when it fails with an error, is cancelled by the user, or runs out its reserved time. Jobs can also fail if the servers they are running on fail. We do not consider ML models with bad accuracy, or simulations with incorrect results as failures.

We collect the data we analyze in this work from the SLURM scheduler logs. We collect job submission information such as number of machines, users, etc. We also collect job scheduling information such as the nodes that were allocated and the completion status.

3 METHOD FOR ANALYZING JOB FAILURES IN HPC CLUSTERS

3.1 Toolbox

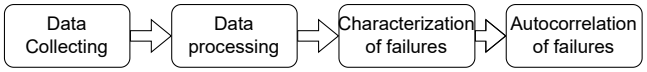


Figure 3: Overview of the toolbox.

The toolbox consists of four parts (see Figure 3): (1) Collect job data from the workload scheduler in HPC clusters. (2) Clean up the job data. (3) Characterize the failure data in the aspects of failure characteristics, arrival patterns, and CPU usage. (4) Present the failure rate autocorrelations in machine learning job failures at multiple time granularities.

3.2 Data collection and clean up

We collect the job data from the workload scheduler SLURM used in the SURF LISA cluster. The clean-up process includes parsing the data format, processing missing values, unifying the unit, splitting nodes, and converting timestamp data. To understand the characteristics of ML job failures and explore the difference between different jobs, we divide the job traces dataset into generic completed jobs, generic failed jobs, ML completed jobs, and ML failed jobs. After that, we get a clean dataset used for analysis.

The job data we analyzed was collected from the scheduler in SURF LISA. LISA is a large HPC cluster consisting of several hundreds of multi-core nodes running the Debian Linux operating system. It provides computing powers for users from different Dutch research and academic institutes. Table 1 summarizes the statistics of the job traces we collected. We observe machine learning jobs account for 13.32% of all jobs.

3.3 Characterization and auto-correlation of failures

We conduct three types of characterization: (1) Failures statistics. Do ML jobs exhibit more failures than generic jobs? When do ML jobs fail, relative to their runtime? To answer these questions, we explore the fraction of jobs per job states, and the duration of jobs.

Table 1: Overview of our job traces dataset.

Dataset	Description
Source	SURF Lisa
Timespan	12 months
Year	2022
#Nodes	348
#Users	2662
#Jobs	2,301,128
%ML Jobs	13.32%
%Generic Jobs	86.68%

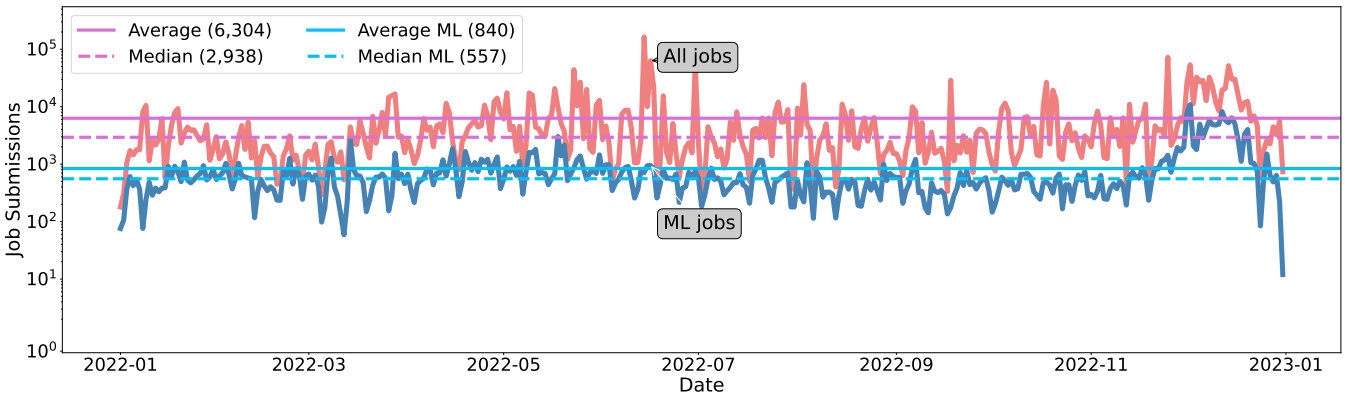


Figure 4: Jobs submitted by date. The horizontal axis contains ticks at the start of every other month. The number of ML jobs is overlaid on the

(2) Failures arrival patterns. How many generic and ML jobs are submitted per day? Are there any patterns in the occurrence of failures? To answer these questions, we visualize submitted jobs by date, and the daily and hourly numbers of failures. (3) Failures CPU usage. To find out how many CPUs are used in different jobs, we present the distribution of the number of CPUs for each kind of job.

We use the autocorrelation function (ACF) to measure the degree of correlation based on the temporal failure data. ACF can assess the degree of correlation between failures at a series of time logs. We consider the failure rate process, which is the number of failure events per unit. We compute the autocorrelation of the failure rate for different time lags including weeks, days, and hours.

4 CHARACTERIZATION OF FAILURES

We characterize the failures in this section, with the method described in 3.3. We present our observations and discussions for each kind of characterization.

4.1 Failures statistics

O-1. Machine learning jobs have a higher failure ratio (23.06%) than generic jobs (14.72%).

O-2. Most machine learning jobs fail early: 86.95% of failed jobs have a runtime less than 6 minutes.

Table 2: Fraction of jobs per job state.

State Type	Generic Jobs	ML Jobs
COMPLETED	77.15%	62.17%
FAILED	14.72%	23.06%
CANCELLED	5.47%	10.15%
TIMEOUT	2.12%	4.38%
OUT OF MEMORY	<1%	<1%
REQUEUED	<1%	<1%
NODE FAILURE	<1%	<1%

O-3. The median runtime of machine learning failures is 18 seconds longer than generic failures.

O-4. 90% of failed machine learning jobs are executed in 0.2 hours, while 90% successful jobs are executed in 3.79 hours.

To explore the basic statistical characteristics of generic and machine learning jobs, we compare the terminal execution state for both job types, ML and generic. Table 2 shows a lot of jobs have unsuccessful job outcomes.

We observe that machine learning jobs have a higher failure rate (23.06%) than generic jobs (14.72%) (**O-1**). Machine learning jobs that are in the "CANCELLED", and "TIMEOUT" states are double the amount of generic jobs. However, jobs that end in "OUT OF MEMORY", "REQUEUED", and "NODE FAILURE" of both types are less than 1%.

Runtime is an important feature to understand how much time the jobs consume. Is there any difference between the duration of generic jobs and machine learning jobs? And how about completed and failed jobs? To answer these questions, we inspect the runtime of generic and machine learning completed and failed jobs as shown

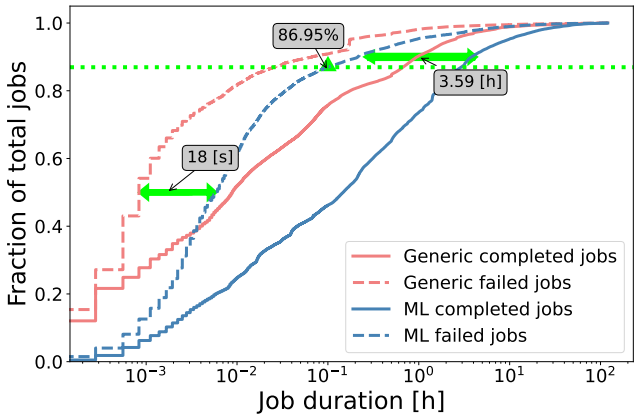


Figure 5: Duration of jobs, CDF plot.

in Figure 5. The figure is an empirical cumulative distribution function (ECDF) of job duration in hours. The majority of completed jobs take up to 1 hour from start to complete. In contrast, most failed jobs are very short: 86.95% of machine learning failed jobs have a runtime of 6 minutes or less (**O-2**). Compared to machine learning failed jobs, 90.54% generic failed jobs are less than 6 minutes.

A failed machine learning job runs for 18 seconds longer than a failed generic job at the median (**O-3**). Above the median, the difference decreases. As for the completed and failed ML jobs, 90% failed jobs are executed in 0.2 hours, while 90% successful jobs are executed in 3.79 hours, which is 3.59 hours longer than unsuccessful jobs (**O-4**). On average, unsuccessful machine learning jobs last for 27 minutes, which is about two times of generic jobs for 14 minutes.

4.2 Failures arrival patterns

O-5. Arrival and demand are highly variable. The number of all submitted jobs per day varies by up to three orders of magnitude.

O-6. Machine learning failures have a diurnal pattern, e.g. 9 to 6 per day.

Many jobs are submitted to a scientific datacenter every day. We make a timeline plot of job submissions to visualize the number of machine learning and generic arrival jobs (see Figure 4). To explore the arrival amounts and patterns of failures, we present the number of failures per day in a week and per hour in a day (see Figure 6).

We observe that job arrival and demand are highly variable per day. Fig 4 shows that 6,304 jobs are submitted per day on average, with a maximum of 163,786 (**O-5**). Compared to the analysis results of the jobs data collected in 2020 [13], the median amount of machine learning jobs increased from 329 to 557, reflecting the upward trend of machine learning research and application.

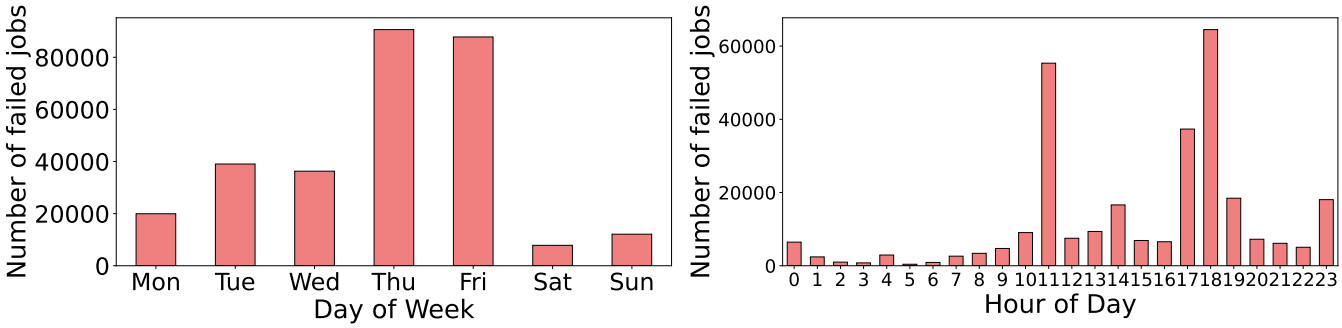
Machine learning exhibits a correlation between work hours and days. In the graphs depicted in Figure 6, while machine learning job failures exhibit a diurnal pattern, generic job failures exhibit irregular/erratic fail behavior, with anomaly peaks at certain days and hours (**O-6**). We conjecture that generic job failures are irregular because general jobs have more unspecified noise due to the operation of the system.

4.3 CPU usage of failures

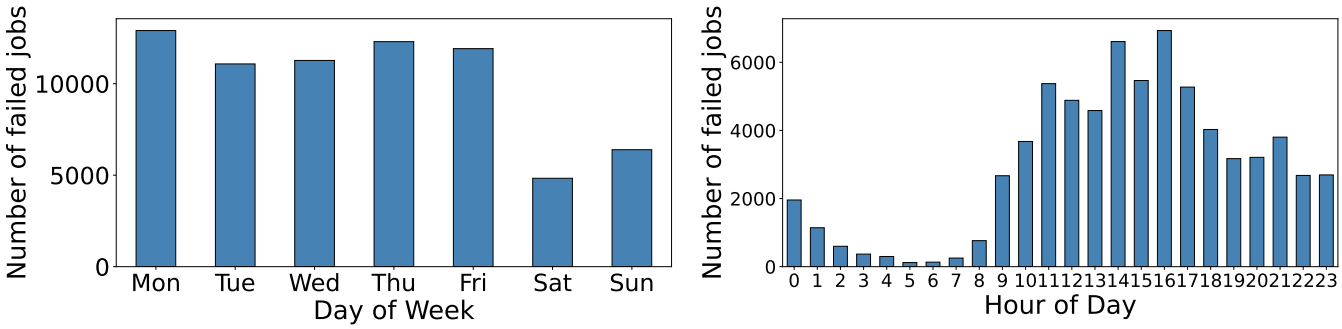
O-7. Failed ML jobs consume a larger fraction (10.6%) of resources consumed by completed ML jobs, compared to generic jobs (7.6%).

O-8. There is no significant difference between completed and failed ML jobs in the distribution of the number of CPUs used per job.

We quantify the resource waste by failed jobs. We define resource waste as the CPU-time consumed by failed jobs as a fraction of the total CPU-time consumed by completed jobs of that kind. We observe that failed machine learning jobs consume 10.6% of the total CPU time consumed by completed ML jobs. This is much higher than the 7.6% resource waste by generic jobs.



(a) Generic jobs.



(b) Machine learning jobs.

Figure 6: Daily and hourly failures for generic and machine learning jobs.

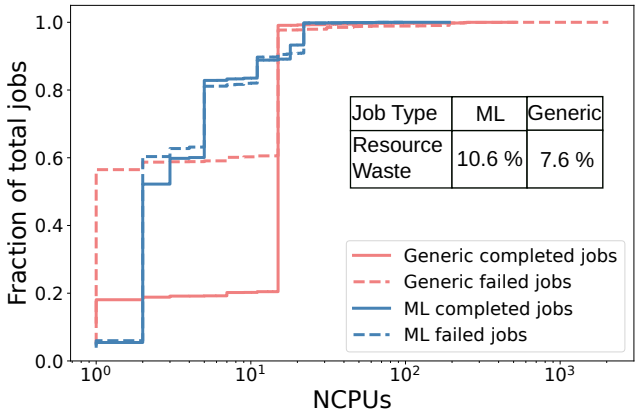


Figure 7: The number of CPUs used in jobs, CDF plot.

We investigate this further by analyzing the distribution of CPU cores allocated to jobs. Figure 7 depicts the ECDF of the number of CPUs (NCPUs) allocated per job, grouped by different categories. We observe that both failed and completed ML jobs exhibit a similar distribution of CPUs allocated. In contrast, failed generic jobs allocate much fewer CPUs per job. Over 50% of failed generic jobs use a single CPU. This is lower than the 3 CPUs per job used by over 50% of failed ML jobs. It is also much lower than the 16 CPUs used by 78.65% of completed generic jobs.

We conjecture that failed generic programs run by users are exploratory programs run during the program development process. Allocating a single CPUs might be sufficient to test these programs. We do not have a theory as to why machine learning jobs require 3x the amount of CPU as generic jobs in the development phase.

5 AUTOCORRELATION OF FAILURES

In this section, we investigate if the occurrence of failure will occur again in the next hour, day, or week. We use the autocorrelation of the trace, computed at different aggregation levels and time lags for our investigation. We use the method described in Section 3.3 to compute the autocorrelation.

O-9. There is high autocorrelation at the week granularity for small time lags, and none thereafter.
O-10. The autocorrelation at the day granularity declines steadily.
O-11. There is little autocorrelation at the hour granularity, but whatever exists follows a diurnal pattern.

Figure 8 depicts the autocorrelation functions of the data aggregated at the week, day, and hour time granularities. We depict the autocorrelation up to 30 units of time lag at every granularity. The horizontal axis represents the time lag, and the vertical axis depicts the autocorrelation at a particular time lag.

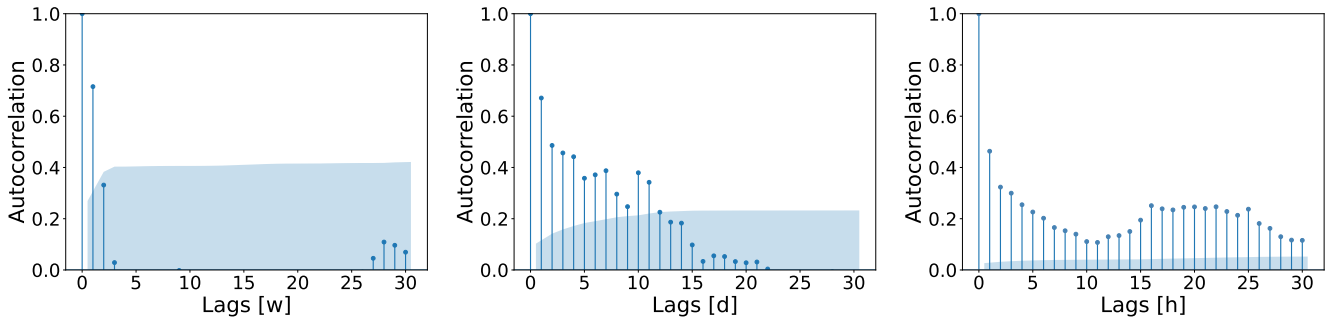
At the week granularity, we observe a strong correlation at time lag 1, a medium correlation at time lag 2, and almost no correlation thereafter. At the day granularity, we observe a medium correlation at low time lags which steadily declines. At the hour granularity, we observe little autocorrelation. But, whatever we observe follows reflects the diurnal pattern which we already found.

The autocorrelation results imply that the number of failures in the previous days and hours reflective of failures in the subsequent days and hours, but only to a little extent. At the week level, while the number of failures in a week is predictive of failures in the next week, it has no predictive power thereafter.

6 LIMITATIONS

Our work has several limitations. We discuss 3 main limitations in this section: the bias inherent to our traces, the limited number of data sources, and the lack of an example of the direct use of the main findings.

First, we only use a single dataset from one datacenter. It is possible that our results, albeit valid for the traces used in this work, are not representative of the workload in other HPC datacenters. Second, the bias inherent in trace use is the classification of machine learning jobs and generic jobs. In this work, jobs on GPU nodes are seen as machine learning jobs. Although most machine learning jobs are running in GPU nodes, some machine learning jobs are sent to CPU nodes by users, which probably affects the results. Finally, there is a lack of an example of the direct use of the main findings. Our understanding of machine learning job failures could not be used for tuning a component of the system yet.



(a) Autocorrelation at week granularity. (b) Autocorrelation at day granularity. (c) Autocorrelation at hour granularity.

Figure 8: Autocorrelation functions with the data aggregated at different time granularities.

7 RELATED WORK

Much work has been dedicated to characterizing, modeling, and predicting job failures in HPC and other clusters [1, 2, 9–11, 14, 15]. These failure characterization works try to analyze unsuccessful jobs observed in datacenters, using cluster data collected by organizations such as Google [8] and Los Alamos National Labs (LANL) [6]; However, the datasets were published many years ago and their data spans relatively short periods of time. Previous failure analysis studies focus mostly on job-related metrics such as the number of jobs, failure rate [15], and CPU/disk/memory usage[1]. Although the time correlation of failure events deserves a detailed investigation due to its practical importance [14], relatively little attention has been given to characterizing the time correlation of failures in HPC clusters.

However, the studies do not consider the special characteristics of different kinds of workloads such as machine learning workloads. In contrast, our work is the first to investigate machine learning job failures in a large-scale HPC distributed system. We perform a detailed investigation using a long-term dataset for 12 months. We also consider the time correlation of failures.

8 CONCLUSION AND FUTURE WORK

In this work, we conduct an analysis of machine learning job failures in an HPC cluster. We propose a method to compare the failures of ML jobs with non-ML jobs. We present 11 observations in our characterization and autocorrelation study of failures.

First, we collected long-term HPC datacenter job traces of temporal and spatial metrics. Second, we characterized failure states, runtime, arrival patterns, and CPU usage. We found that machine learning jobs have a higher failure ratio and present daily patterns in contrast to generic jobs. We also found that failed ML jobs consumed more CPU resources. Last but not least, we investigated in this work the time correlations of ML failure events. We found that the autocorrelation varies at different time granularities: In the week and day granularity, there are obvious correlations but only at small time lags; While at the hour granularity there is little autocorrelation, but whatever exists follows the diurnal pattern as we observed in the characterization.

We see several possible extensions to this work. First, we only consider failed machine learning jobs. However, ML jobs that are cancelled and timeout also accounts for a large fraction of all states. We will investigate the characteristics of those jobs in the future. Second, we will collect more diverse job traces from different HPC or other datacenters to make our results representative. Finally, we want to use the time correlation between day and hour to predict failures before it happens, and design a method to help reduce the impact of failures.

ACKNOWLEDGMENT

We thank the Dutch national supercomputing center SURF for providing us the data. We thank the China Scholarship Council (CSC) for supporting Xiaoyu Chu. We thank the projects NWO Top2 OffSense, EU H2020 GraphMassivizer, and EU MCSA-RISE CLOUDSTARS for co-funding this project.

REFERENCES

[1] Xin Chen, Charng-Da Lu, and Karthik Pattabiraman. 2014. Failure analysis of jobs in compute clouds: A google cluster case study. In *2014 IEEE 25th International Symposium on Software Reliability Engineering*. IEEE, 167–177.

[2] Nosayba El-Sayed, Hongyu Zhu, and Bianca Schroeder. 2017. Learning from failure across multiple clusters: A trace-driven approach to understanding, predicting, and mitigating job terminations. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1333–1344.

[3] Geoffrey C. Fox, James A. Glazier, J. C. S. Kadupitiya, Vikram Jadhao, Minje Kim, Judy Qiu, James P. Sluka, Endre T. Somogyi, Madhav V. Marathe, Abhijin Adiga, Jiangzhuo Chen, Oliver Beckstein, and Shantenu Jha. 2019. Learning Everywhere: Pervasive Machine Learning for Effective High-Performance Computation. In *IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2019, Rio de Janeiro, Brazil, May 20-24, 2019*. IEEE, 422–429. https://doi.org/10.1109/IPDPSW.2019.00081

[4] Rohan Garg, Tirthak Patel, Gene Cooperman, and Devesh Tiwari. 2018. Shiraz: Exploiting System Reliability and Application Resilience Characteristics to Improve Large Scale System Throughput. In *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018, Luxembourg City, Luxembourg, June 25-28, 2018*. IEEE Computer Society, 83–94. https://doi.org/10.1109/DSN.2018.00021

[5] Saurabh Gupta, Tirthak Patel, Christian Engelmann, and Devesh Tiwari. 2017. Failures in large scale systems: long-term measurement, analysis, and implications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2017, Denver, CO, USA, November 12 - 17, 2017*, Bernd Mohr and Padma Raghavan (Eds.). ACM, 44. https://doi.org/10.1145/3126908.3126937

[6] LLC LANS. [n. d.]. Operational data to support and enable computer science research.

[7] Adam Moody, Greg Bronevetsky, Kathryn Mohror, and Bronis R. de Supinski. 2010. Design, Modeling, and Evaluation of a Scalable Multi-level Checkpointing System. In *Conference on High Performance Computing Networking, Storage and Analysis, SC 2010, New Orleans, LA, USA, November 13-19, 2010*. IEEE, 1–11. https://doi.org/10.1109/SC.2010.18

[8] Charles Reiss, John Wilkes, and Joseph L Hellerstein. 2011. Google cluster-usage traces: format+ schema. *Google Inc., White Paper* 1 (2011).

[9] Andrea Rosa, Lydia Y Chen, and Walter Binder. 2015. Predicting and mitigating jobs failures in big data clusters. In *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE, 221–230.

[10] Andrea Rosa, Lydia Y Chen, and Walter Binder. 2015. Understanding the dark side of big data clusters: An analysis beyond failures. In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 207–218.

[11] Andrea Rosà, Lydia Y Chen, and Walter Binder. 2016. Failure analysis and prediction for big-data systems. *IEEE Transactions on Services Computing* 10, 6 (2016), 984–998.

[12] Bianca Schroeder and Garth A. Gibson. 2010. A Large-Scale Study of Failures in High-Performance Computing Systems. *IEEE Trans. Dependable Secur. Comput.* 7, 4 (2010), 337–351. https://doi.org/10.1109/TDSC.2009.4

[13] Laurens Versluis, Mehmet Cetin, Caspar Greeven, Kristian Laursen, Damian Podareanu, Valeriu Codreanu, Alexandru Uta, and Alexandru Iosup. 2021. A Holistic Analysis of Datacenter Operations: Resource Usage, Energy, and Workload Characterization–Extended Technical Report. *arXiv preprint arXiv:2107.11832* (2021).

[14] Nezih Yigitbasi, Matthieu Gallet, Derrick Kondo, Alexandru Iosup, and Dick Epema. 2010. Analysis and modeling of time-correlated failures in large-scale distributed systems. In *2010 11th IEEE/ACM International Conference on Grid Computing*. IEEE, 65–72.

[15] Yulai Yuan, Yongwei Wu, Qiuping Wang, Guangwen Yang, and Weimin Zheng. 2012. Job failures in high performance computing systems: A large-scale empirical study. *Computers & Mathematics with Applications* 63, 2 (2012), 365–377.