38th International Conference on Massive Storage Systems and Technology
MSST 2024 (Research Track)
June 6-7, 2024 at Santa Clara University, Santa Clara, California, USA

# Dissecting I/O Burstiness in Machine Learning Cloud Platform: A Case Study on Alibaba's MLaaS

Qiang Zou, *Guangxi Minzu University, China*

Yuhui Deng, *Jinan University, China*
**Yifeng Zhu**, *University of Maine, USA*
Yi Zhou, *Columbus State University, USA*
Jianghe Cai, *Jinan University, China*
Shuibing He, *Zhejiang University, China*
yifeng.zhu@maine.edu
https://web.eece.maine.edu/~zhu/

# Outline

- Background
- Motivation
- Burstiness & Heavy-tailed Property
- Auto-correlation & Self-similarity
- Synthesis
- Conclusion

# Background

- Why Alibaba's MLaaS (Machine-Learning-as-a-Service)?
  - ✓ Alibaba Cloud launched **PAI** – the ML **P**latform for **A**rtificial **I**ntelligence
  - ✓ Representative - one of the leading MLaaS platforms in China
- For PAI:
  - ✓ Over 6500 GPUs across 1800 machines
  - ✓ See Fig. 1 for the architecture overview
- The scalable storage solutions rely on:
  - ✓ Four key components – Alibaba Cloud's object storage, distributed file system, database solutions, and elastic block storage

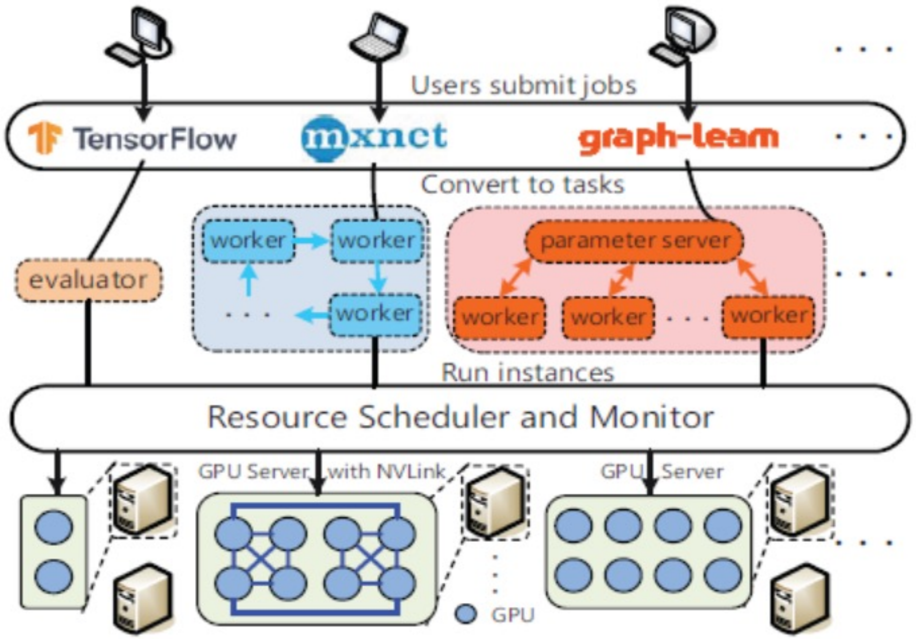



Fig. 1. PAI architecture overview.

# Background

- PAI traces collected on machines of GPU clusters
  - ✓ The PAI traces at the job, task, and instance levels provide launch information including status, start_time, etc.
  - ✓ The machine-level PAI trace contains information, such as timestamps, I/O waiting times (iowait), execution times in user and kernel modes, etc.
  - ✓ See the referenced literature [1] for the details
- This study aims at:
  - ✓ The machine-level trace
  - ✓ Timestamp information (in seconds for about two months)

TABLE I
SUMMARY OF PAI TRACE AND MACHINE SPECS OF GPU CLUSTERS [1].

| #Machines | 1800 | | Duration | 2 months | |
|---|---|---|---|---|---|
| Memory (GiB) | 512 | 512 | 512 | 384 | 512/384 |
| GPU type | P100 | T4 | Misc. | V100M32 | V100 |
| #GPUs | 2 | 2 | 8 | 8 | 8 |
| #Nodes | 798 | 497 | 280 | 135 | 104 |

# Motivation

**Observations**

Traditional methods, such as uniform [4] and exponential [5] distributions, are used to describe request behaviors in typical ML workloads

Burst phenomena are prevalent in both ML and DL workloads [1], [2], [6]

The traditional distributions mentioned above fail to specialize in capturing the burst behavior

Inspire us to think

**Issues**

Is it appropriate to use traditional methods to describe the burst behaviors in MLaaS I/O workloads?

Whether do the request activities in MLaaS workloads also present the self-similarity observed in other workloads [12], [13]?

How to address these issues?

# Burstiness & Heavy-

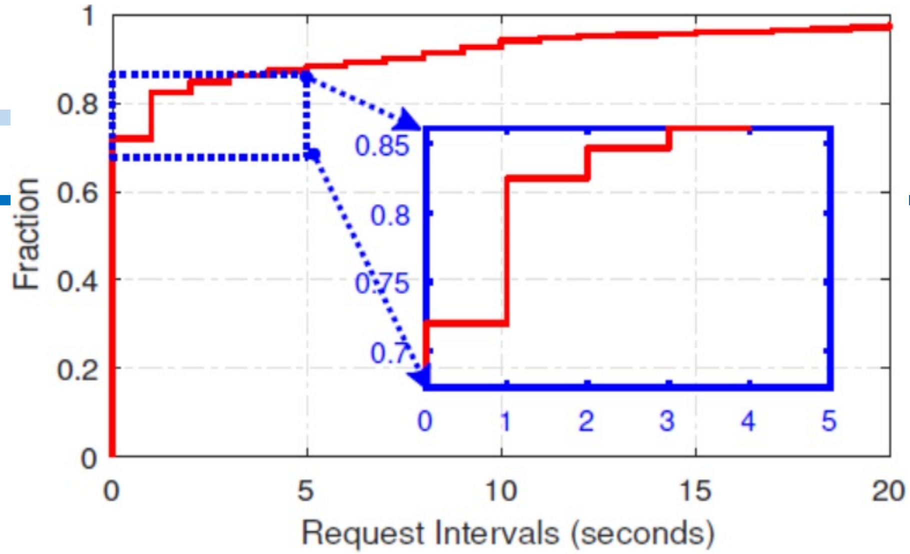



Fig. 2. Empirical CDF of request arrival intervals in the PAI workload.

- ➢ To show the burstiness quantitatively,
  - ✓ By concept: ***non-stationary***, a large ***variance***;
  - ✓ Approach – empirical study
- ➢ Non-stationary:
  - ✓ 83% of I/O requests arrive within an interval of no more than 1 second
  - ✓ up to 72% of requests arrive simultaneously at certain moments (in seconds)
- ➢ Variance: as high as 8892
- ➢ To measure the strength of burstiness,
  - ✓ Using the *__i__ndex of __d__ispersion for __i__ntervals* (IDI) [20]
  - ✓ A larger value of the index of dispersion indicates stronger burstiness
  - ✓ We calculate the IDI for I/O arrivals in the PAI workload as 1519 (significant bursty)

# Burstiness & Heavy-tailed Property

- Gaussianity Test:
  - ✓ helps accurately describe the tail trend in the distribution of access characteristics
  - ✓ can be conducted using a quantile-quantile (**QQ**) **plot**
- **For PAI** (see Figure 3):
  - ✓ The corresponding scatter points clearly do not fall on a straight line
  - ✓ Instead, the curve is concave upward, indicating *a heavy-tailed trend*
  - ✓ Suggesting that the I/O behaviors in the PAI workload are non-Gaussian

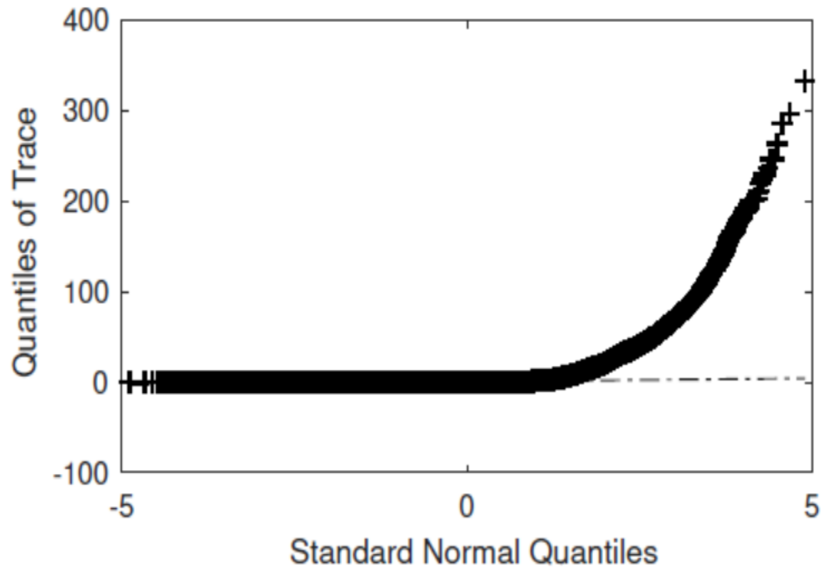



Fig. 3. Examine the Gaussianity of I/O request activities in the PAI workload through QQ plot of the PAI trace data versus standard normal, respectively.

# Auto-correlation & Self-similarity

- Tool: Auto-Correlation Function (ACF)
  - ✓ For a time series $Y = \{Y_t: t = 1,\ 2, \dots, n\}$, $\theta = E[Y_t]$, $y_t = Y_t - \theta$,
  - ✓ Correlation coefficients: $R(k) = \frac{E[y_t \cdot y_{t+k}]}{E[{y_t}^2]}$, for $k \geq 0$
  - ✓ A correlation coefficient forms a mapping relationship with a time interval (also called *lag*) *k*
- How is the auto-correlation curve *related to request activities*?
  - ✓ **If** the correlation coefficients of arrival intervals decrease rapidly with the increase of *lag* and approach 0, there is almost no correlation.
  - ✓ **Otherwise**, there is a certain degree of correlation for requests

# **Auto-correlation** & Self-similarity

- For I/O requests in the PAI workload, see Figure 4,
  - ✓ As the lag increases from 0 to 100, the correlation coefficients of I/O requests do not approach zero sharply; instead, they exhibit a gradual declining trend
  - ✓ There is *a noticeable degree of correlation* between request arrivals in the PAI workload
- Therefore, exploring self-similarity in the PAI workload becomes essential to accurately understand request behaviors

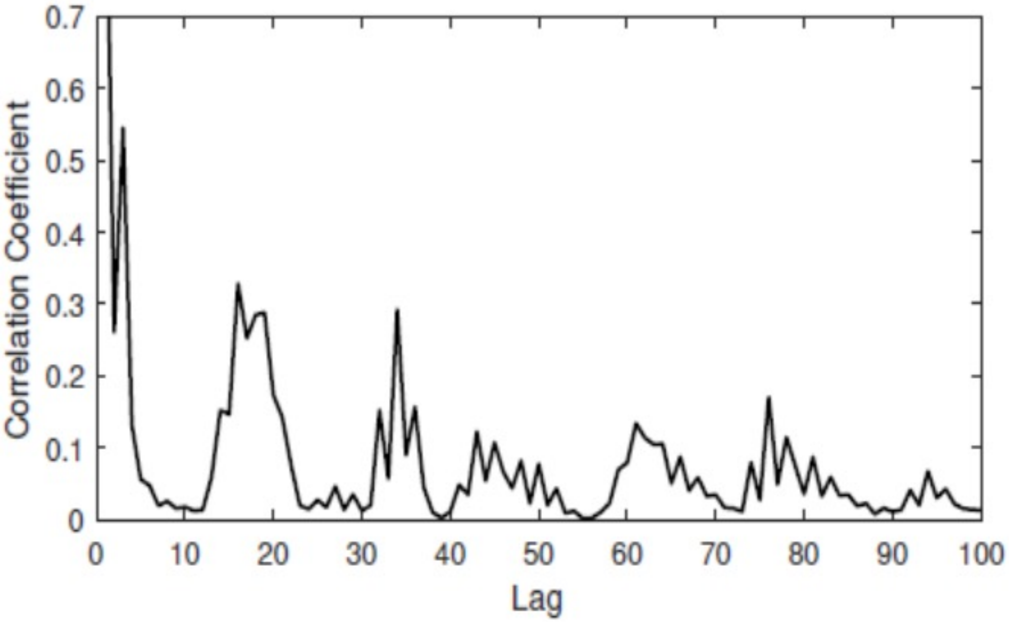



Fig. 4. Auto-correlation function of I/O request arrivals in the PAI workload.

# Auto-correlation & Self-similarity

- What is self-similarity?
  - ✓ In brief, the characteristics of a certain process are similar from different time scales
- How to explore the self-similarity in system workloads?
  - ✓ Showing the visualization
  - ✓ Providing theoretical evidence
  - ✓ Estimating the *Hurst* parameter ($0.5<H<1$)
- The well-known tools to estimate the *Hurst* Parameter:
  - ✓ Variance-time plot [12]
  - ✓ R/S (rescaled adjusted range) analysis (also called *Pox plot*) [26]

# Self-similarity (Visualization)

- Main trait: the persistence of bursts and burst aggregations at various timescales
- For PAI, see Figure 5:
  - ✓ Three different timescales in subplots (a)-(c);
  - ✓ each subsequent timescale being ten times larger than the previous one
  - ✓ Each subplot is derived from a subinterval randomly selected from the time range depicted in the following subplot and it enhances the temporal resolution by a factor of 10
- **Finding**: The time range characterized by bursty requests consists of nested subintervals, each is made of even smaller subintervals with similar burst behaviors.

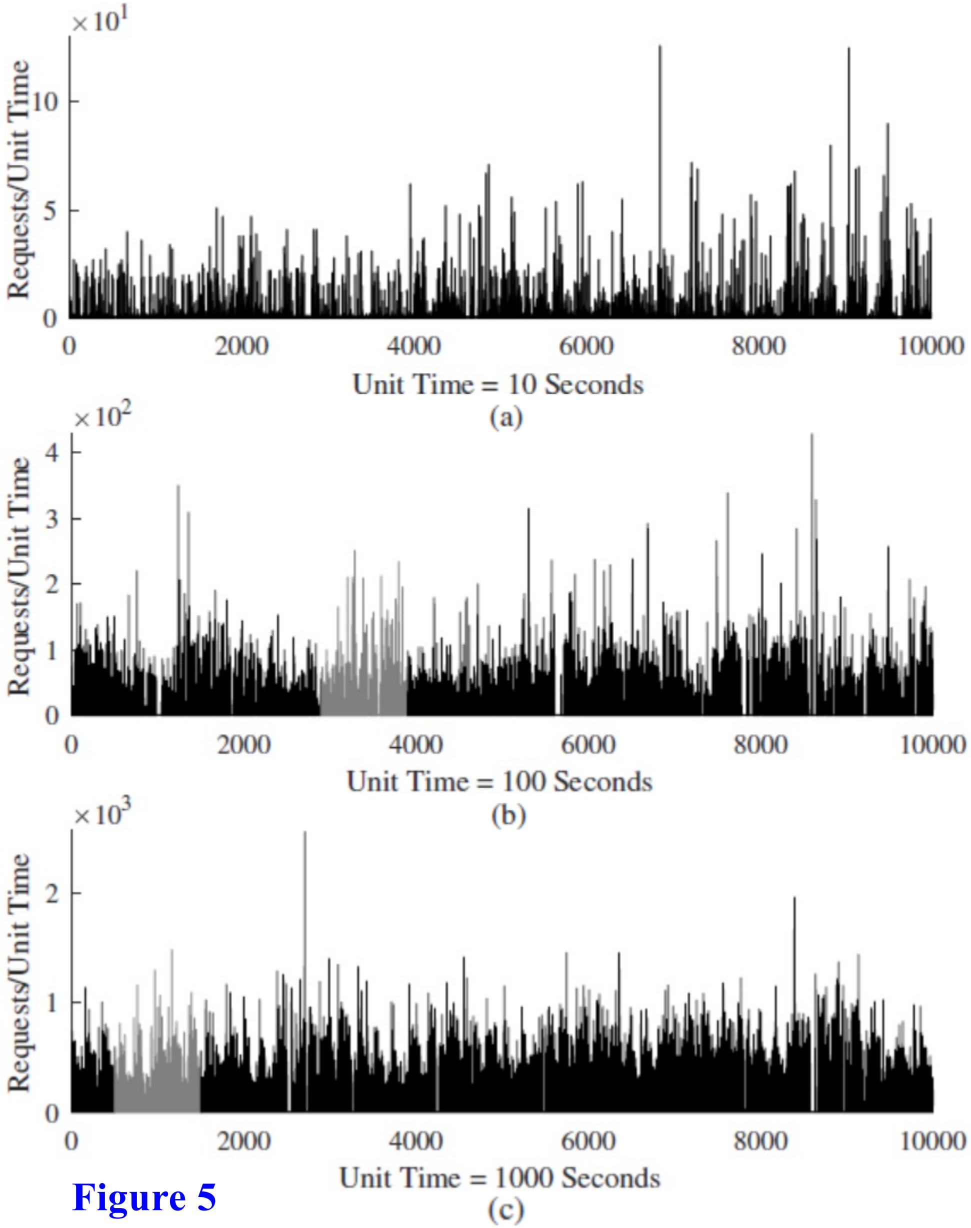



**Figure 5**

# Self-similarity (Theoretical evidence)

- Theoretical basis: see the statements regarding the structure of $R^{(m)}(k)$ in Section IV-A
- Examining the auto-correlation functions of the aggregated time series of the request sequence at multiple aggregation levels
- For PAI, see Figure 6:
  - ✓ Plot (a) depicts the ACFs of the aggregated time series of the request sequence at multiple aggregation levels, that appear to converge to a similar function structure;
  - ✓ Plot (b) demonstrates that the auto-correlation coefficients of Poisson workload at each aggregation level are generally very small and almost equal to zero
- Quite different from Poisson, request activities in the PAI workload behave like a self-similar process

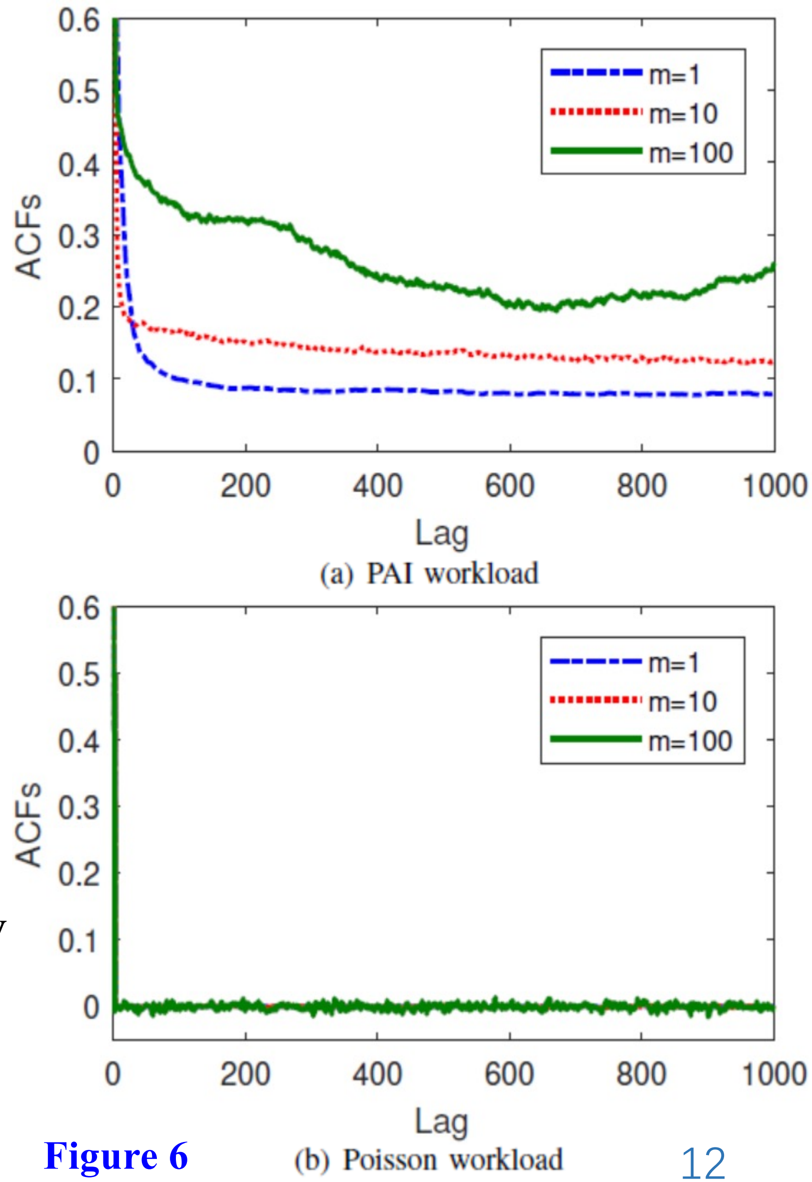



**Figure 6**

# Self-similarity (Hurst Parameter)

- For the I/O request sequence in the PAI workload, the Hurst parameter can be estimated by:
  - ✓ The variance-time plot: see Fig. 7(a);
  - ✓ The Pox plot: see Fig. 7(b)
- Finding:
  - ✓ All Hurst parameter estimates are greater than 0.5;
  - ✓ Quantitatively confirming the existence of self-similarity

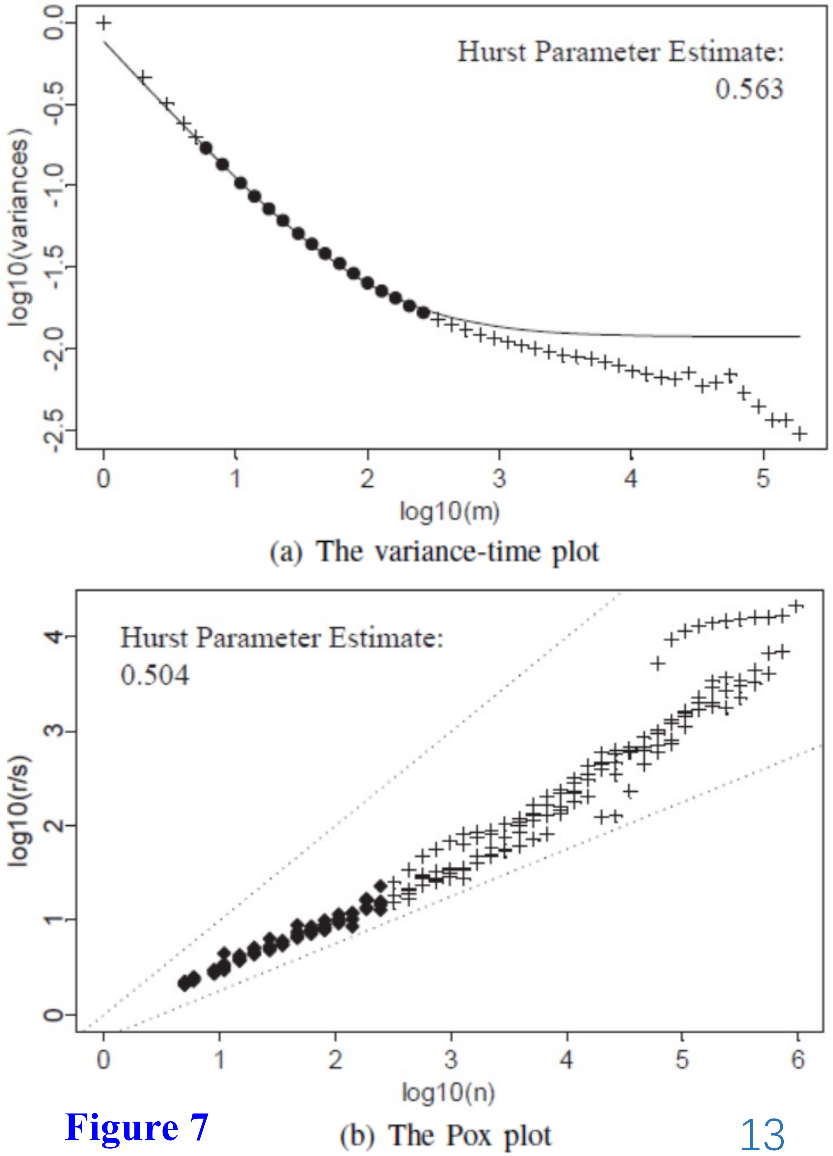



**Figure 7**

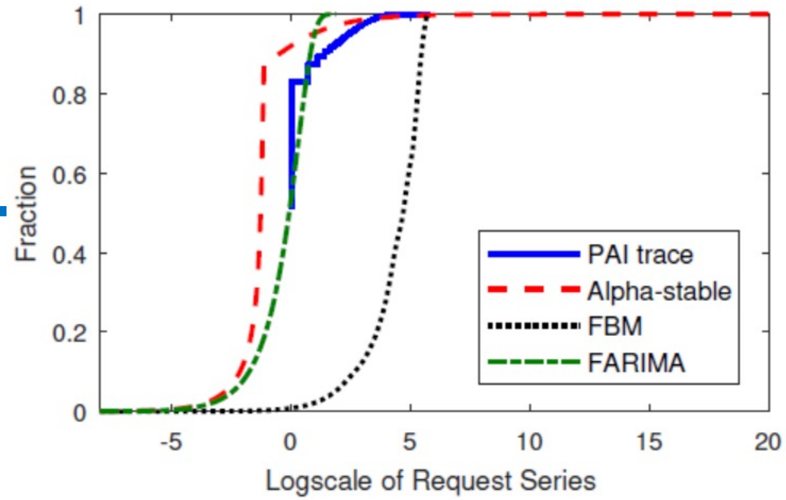
# Synthesis

- We have made the following findings:
  - ✓ The arrival process of I/O requests is highly bursty
  - ✓ Traditional methods struggle to accurately characterize the PAI workload, as the I/O arrivals show a certain degree of correlation
  - ✓ There seems to be self-similarity in the PAI workload
  - ✓ The I/O request activities in PAI appear to be non-Gaussian
- These findings inspire us to use several methods to synthesize I/O request series for the **self-similar** PAI workload

# Synthesis

- Two typical self-similar workload models are chosen:
  - ✓ Fractional Brownian motion (FBM) [27] – is adept at characterizing self-similarity under Gaussian conditions
  - ✓ Fractional autoregressive integrated moving average (FARIMA) [28] – is well-known for its ability to describe both long-range and short-range dependences
- The versatile alpha-stable model [8] is also extended,
  - ✓ **by** redefining its model parameters to synthesize request series for PAI
  - ✓ **to** faithfully describe the bursts and heavy-tailed properties under non-Gaussian conditions

# Synthesis




- To evaluate the accuracy of these models, we adopt:
  - ✓ The trimmed mean of errors [8];
  - ✓ The cumulative distribution functions (CDFs)
- The trimmed mean of errors for FBM, FARIMA, and alpha-stable models:

**Figure 8**

  - ✓ is 109.46, 1.26, and 0.78, respectively;
  - ✓ the trimmed mean of the errors for the alpha-stable synthetic sequence (i.e., 0.78) is very close to that for the FARIMA synthetic one (i.e., 1.26)
- For the CDFs of the actual series and the synthetic ones, as shown in Figure 8:
  - ✓ Both the FARIMA synthetic sequence and the alpha-stable synthetic one exhibit convincing matching degrees
  - ✓ One advantage of the latter over the former is its ability to better capture the heavy-tailed feature

# Conclusion

- Characterizing the request behaviors in MLaaS workloads is crucial for scheduling and managing the I/O subsystem in GPU clusters.
- This paper studies the burstiness of the I/O requests in a representative and real-world MLaaS workload – the PAI workload, and shows the existence of self-similarity in the PAI workload.
- Based on the inputs measured from real trace data, we deploy self-similar workload models to synthesize I/O request sequences for the PAI workload.

38th International Conference on Massive Storage Systems and Technology
MSST 2024 (Research Track)
June 6-7, 2024 at Santa Clara University, Santa Clara, California, USA

# Dissecting I/O Burstiness in Machine Learning Cloud Platform: A Case Study on Alibaba's MLaaS


Qiang Zou (qiangzou@hotmail.com), Yuhui Deng (tyhdeng@jnu.edu.cn),
**Yifeng Zhu (yifeng.zhu@maine.edu)**, Yi Zhou (zhou_yi@columbusstate.edu),
Jianghe Cai (761571151@qq.com), Shuibing He(heshuibing@zju.edu.cn)


# *Thank you!*