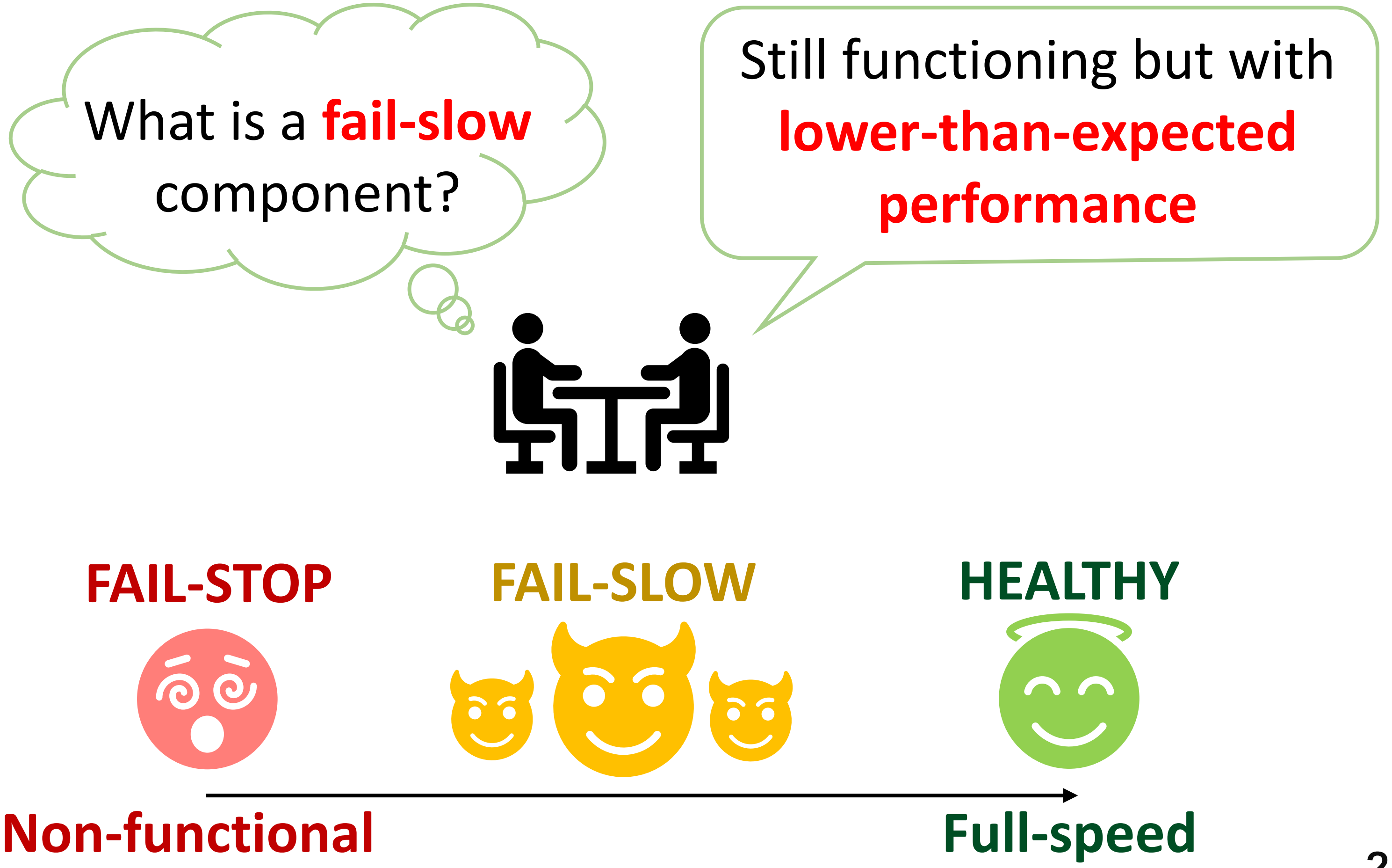
21st USENIX Conference on File and Storage Technologies FAST'23

Perseus: A Fail-Slow Detection Framework for Cloud Storage Systems

Ruiming Lu, Erci Xu, Yiming Zhang,
Fengyi Zhu, Zhaosheng Zhu, Mengtian Wang,
Zongpeng Zhu, Guangtao Xue, Jiwu Shu, Minglu Li, Jiesheng Wu

Data Center Instability

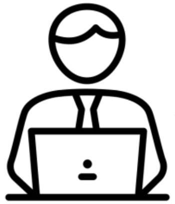
- **Failures in The Wild**
  - **Fail-Slow**
  - Fail-Stop
  - Byzantine
  - …



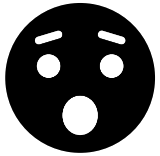
Not A Problem?

FAIL-SLOW

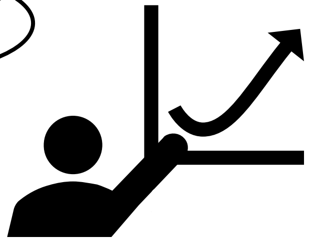
- Severe
  - "Fail-slow NVMe SSDs can **degrade to SATA SSD or HDD-level performance**[1]"
- Not Uncommon



  - Annual fail-slow failure rate is **1-2%**[2]!
  - **As frequent as fail-stop** incidents!



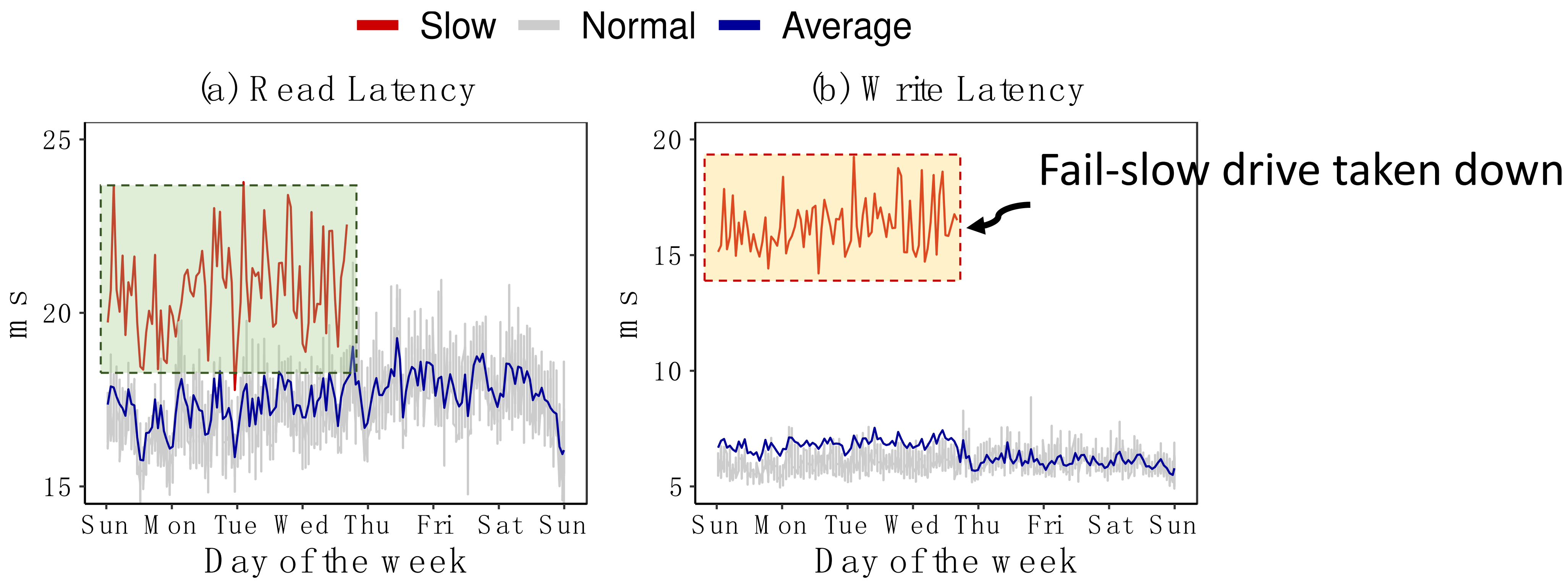
- Confusing
  - Fail-slow or just **normal variations**?



[1] NVMe SSD Failures in the Field: the Fail-Stop and the Fail-Slow, Lu et al.
[2] IASO: A Fail-Slow Detection and Mitigation Framework for Distributed Storage Services, Panda et al.

Not A Problem?

Fail-Slow in The Field:

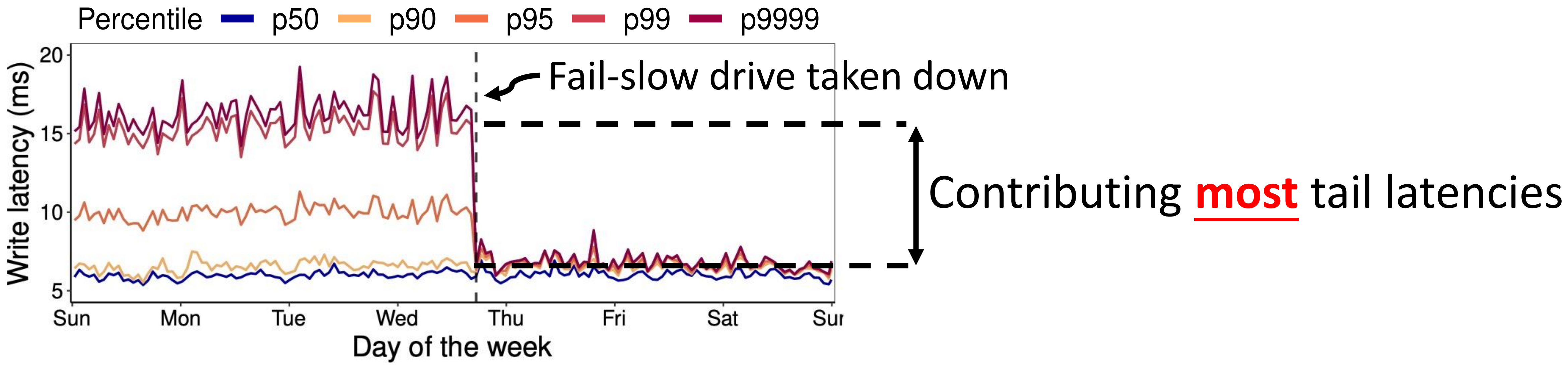


1.01-1.50X higher for read

2.06-3.65X higher for write

Not A Problem?

Fail-Slow in The Field:



FAIL-SLOW

Slow! Slow! Slow! Silent performance degradation!

Let's "dig" them out!

Hard To Detect

- **No Ground Truth in Identifying Fail-Slow**

How slow is a drive to be considered fail-slow?

>100us?

>500us?

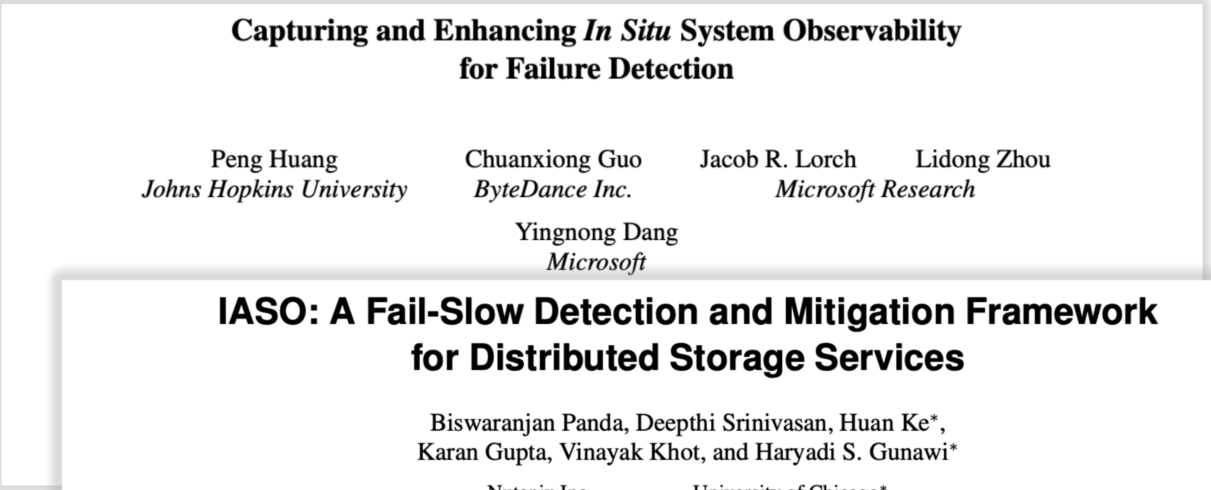
>1ms?

⋮

A “thousand” ways to define!

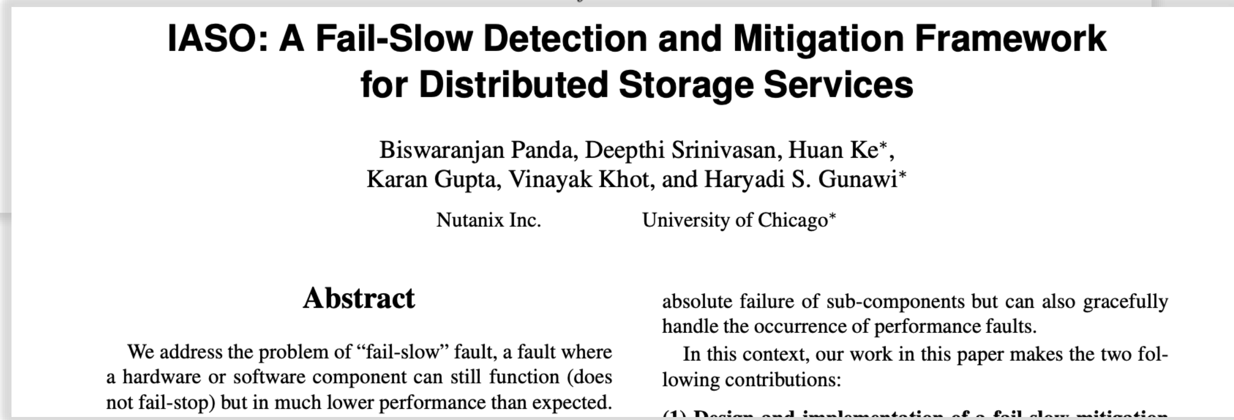
Fail-Slow Detection (FSD)

- **Previous FSD Studies Are**
  - Intrusive
    - **Source Code Accessing/Altering**
  - Coarse-grained
    - **Node-Level Detection**



Capturing and Enhancing *In Situ* System Observability for Failure Detection

Peng Huang, *Johns Hopkins University*; Chuanxiong Guo, *ByteDance Inc.*; Jacob R. Lorch, Lidong Zhou, *Microsoft Research*; Yingnong Dang, *Microsoft*



IASO: A Fail-Slow Detection and Mitigation Framework for Distributed Storage Services

Biswaranjan Panda, Deepthi Srinivasan, Huan Ke*, Karan Gupta, Vinayak Khot, and Haryadi S. Gunawi*

Nutanix Inc. University of Chicago*

Abstract

We address the problem of "fail-slow" fault, a fault where a hardware or software component can still function (does not fail-stop) but in much lower performance than expected.

absolute failure of sub-components but can also gracefully handle the occurrence of performance faults.

In this context, our work in this paper makes the two following contributions:

Fail-Slow Detection (FSD)

- **Our Work Shares**
  - Years of Experiences in FSD
  - A Practical FSD Framework named **Perseus**
  - Root Cause Analysis

Outline

~~INTRODUCTION~~

DATASET

FAILED ATTEMPTS

PERSEUS

EVALUATION & CONCLUSION

Our Dataset

- **248K+ drives**
  - 55% NVMe SSD + 45% SATA HDD
  - 4 manufacturers
  - 9 major drive models
  - Diverse cloud services:
    - Log service, big data, E-commerce, table storage, stream processing, database, object storage, data warehouse, block storage

Our Dataset

- **248K+ drives**
- **10-month performance logs (iostat)**
  - Latency/throughput time series
- **Test dataset released**
  - https://tianchi.aliyun.com/dataset/144479

FAST'23

Outline

~~INTRODUCTION~~

~~DATASET~~

FAILED ATTEMPTS

PERSEUS

EVALUATION & CONCLUSION

Ideal Fail-Slow Detection Should Be ...?

Efficient Fail-Slow Detection

Non-intrusive

No source code altering
External performance log-based

Accurate

High precision/recall rate

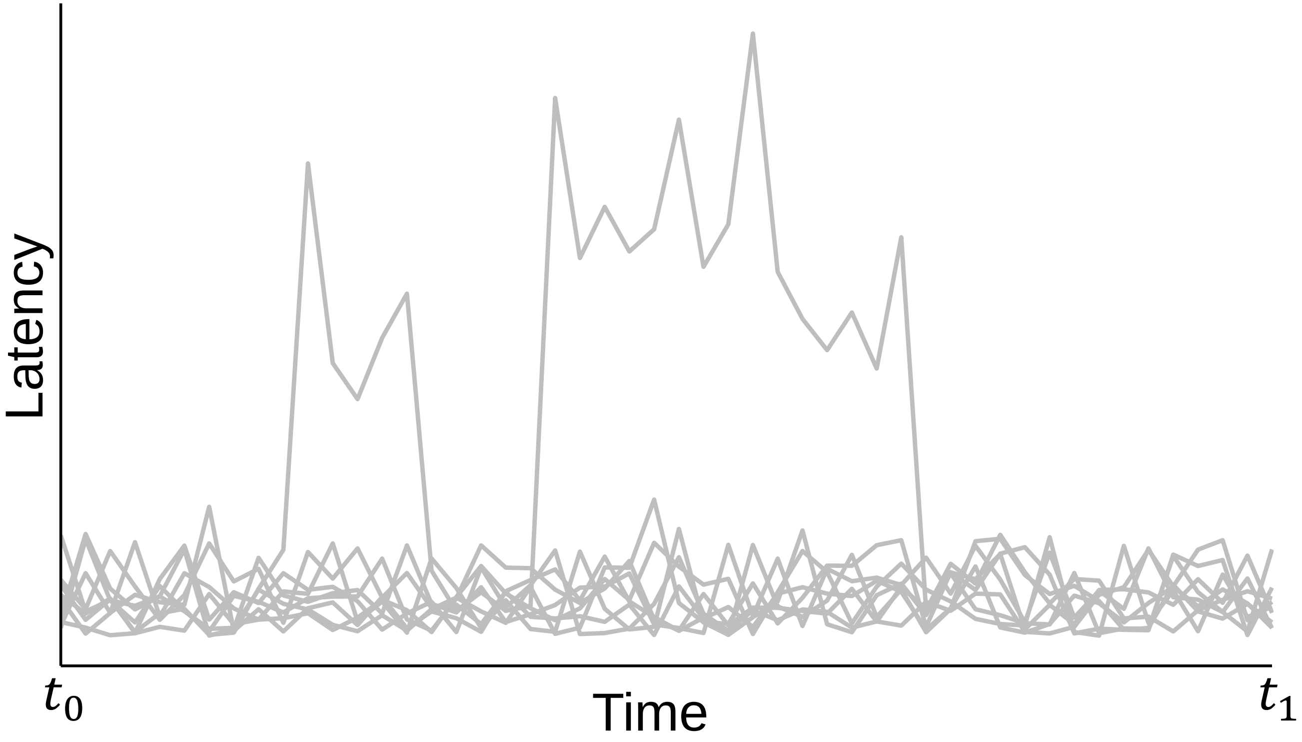
Fine-grained

Device-level detection

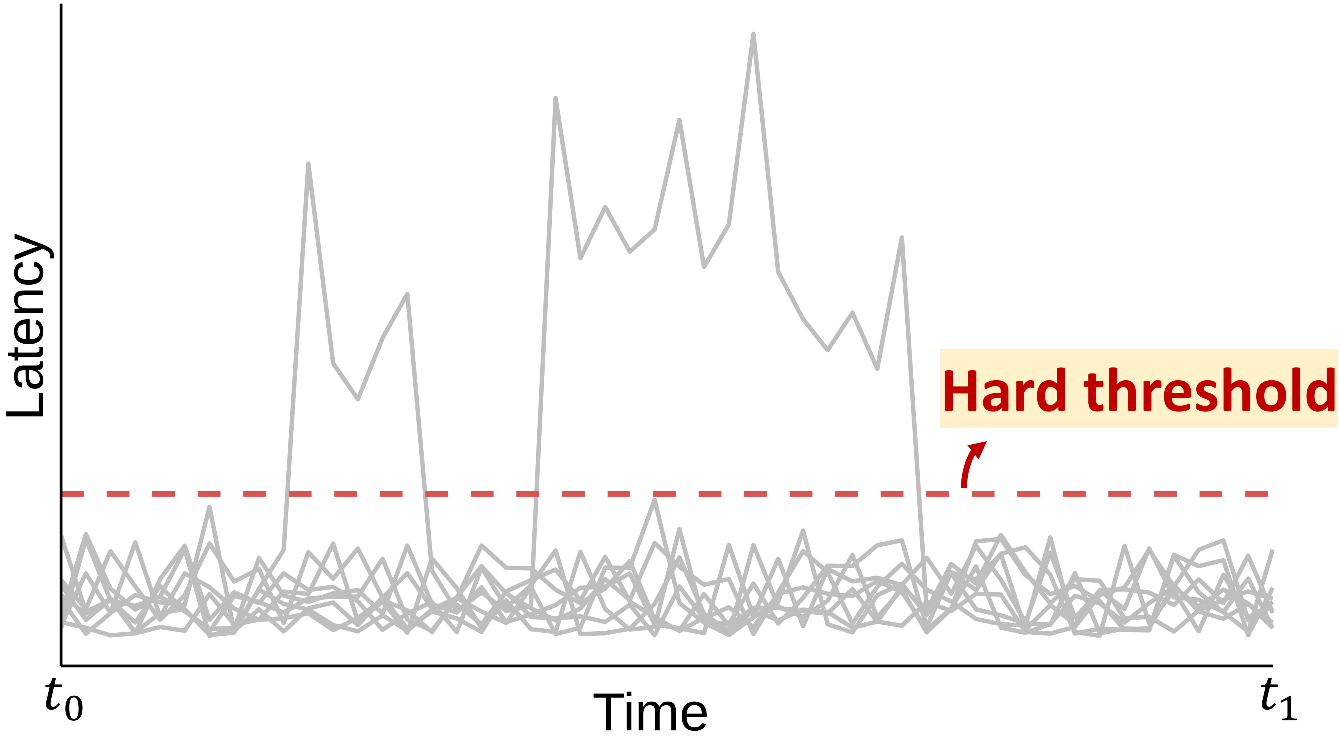
General

Adaptable to
- SSD/HDD clusters
- Various cloud services

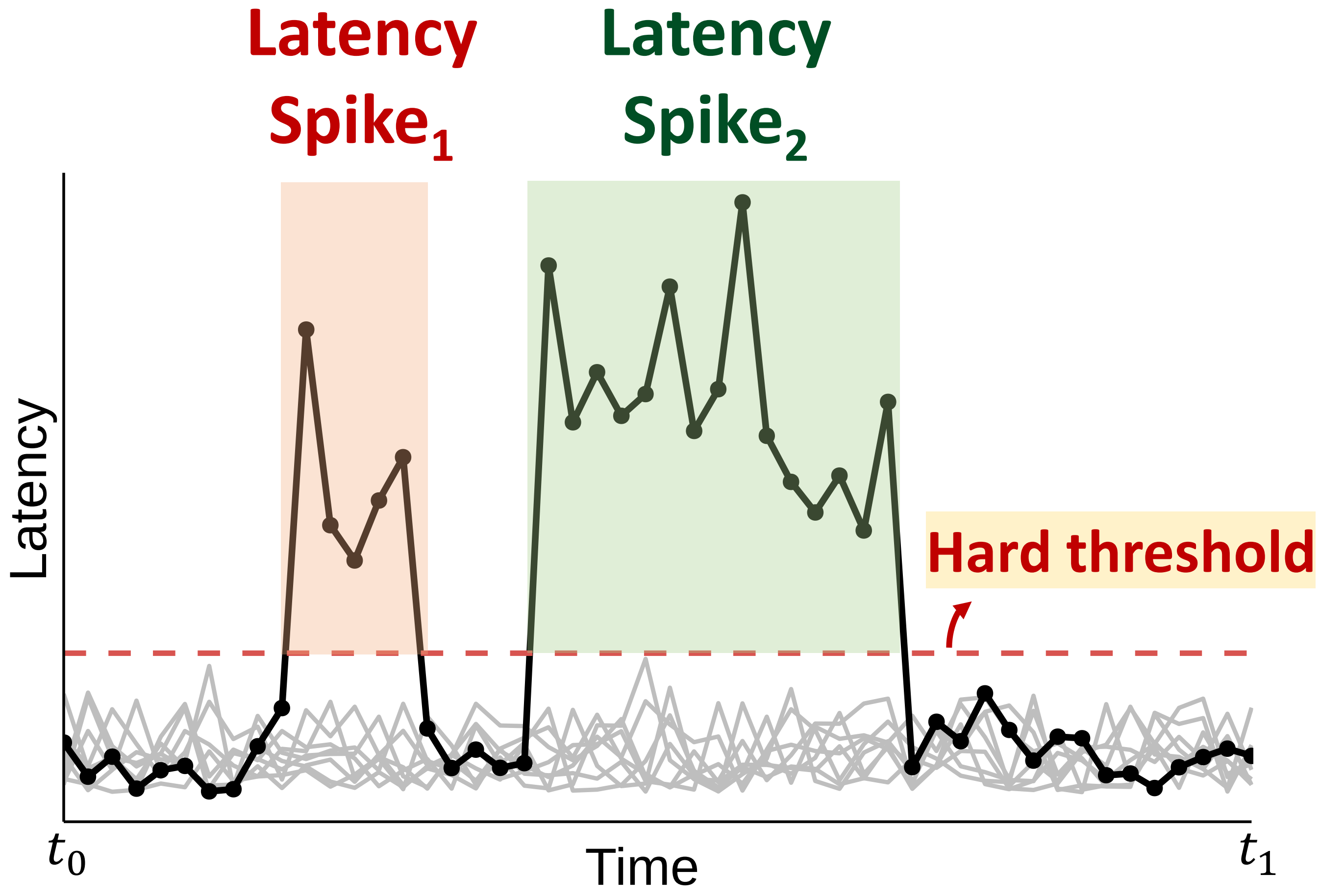
Failed Attempt: Threshold Filtering



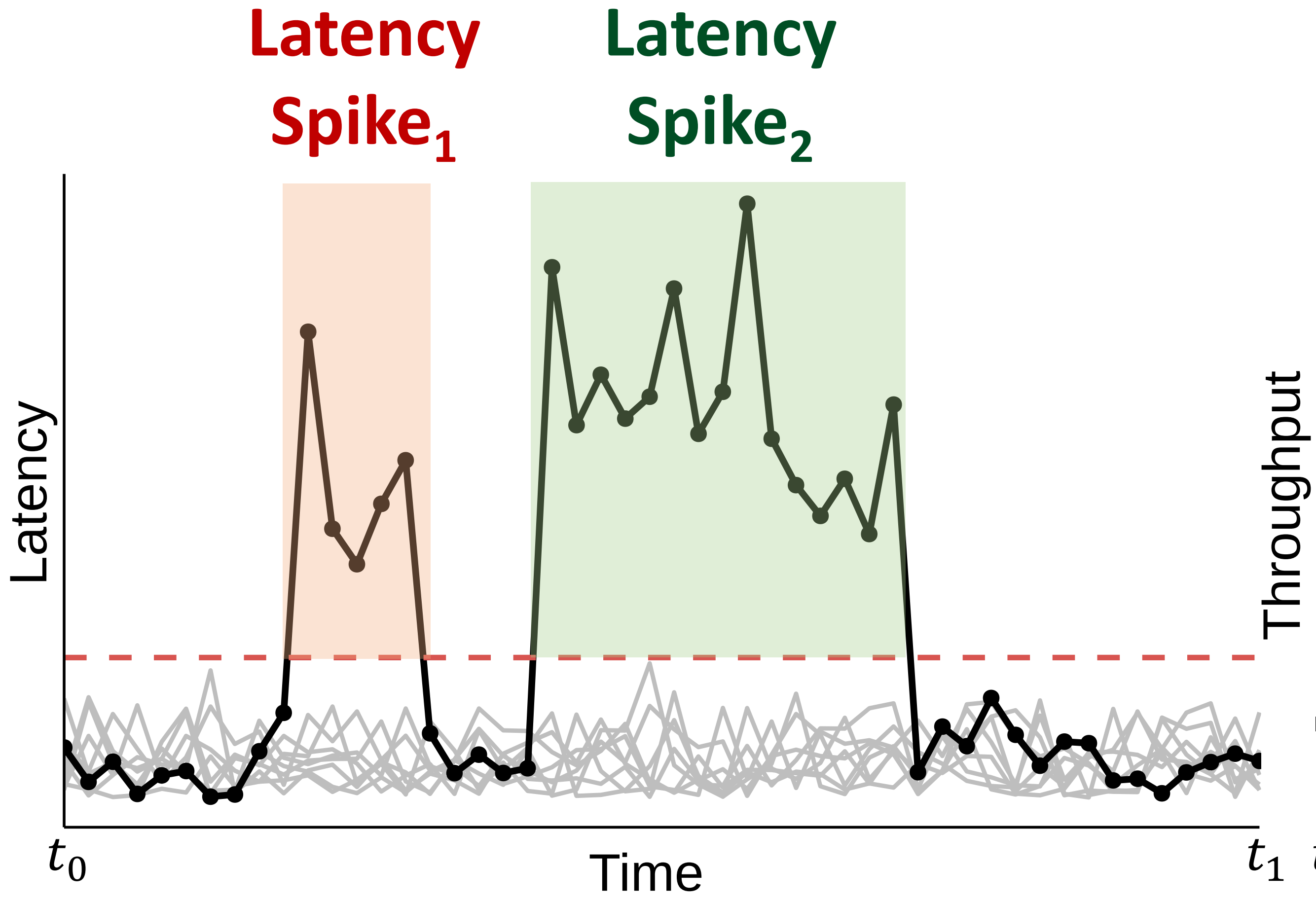
Failed Attempt: Threshold Filtering

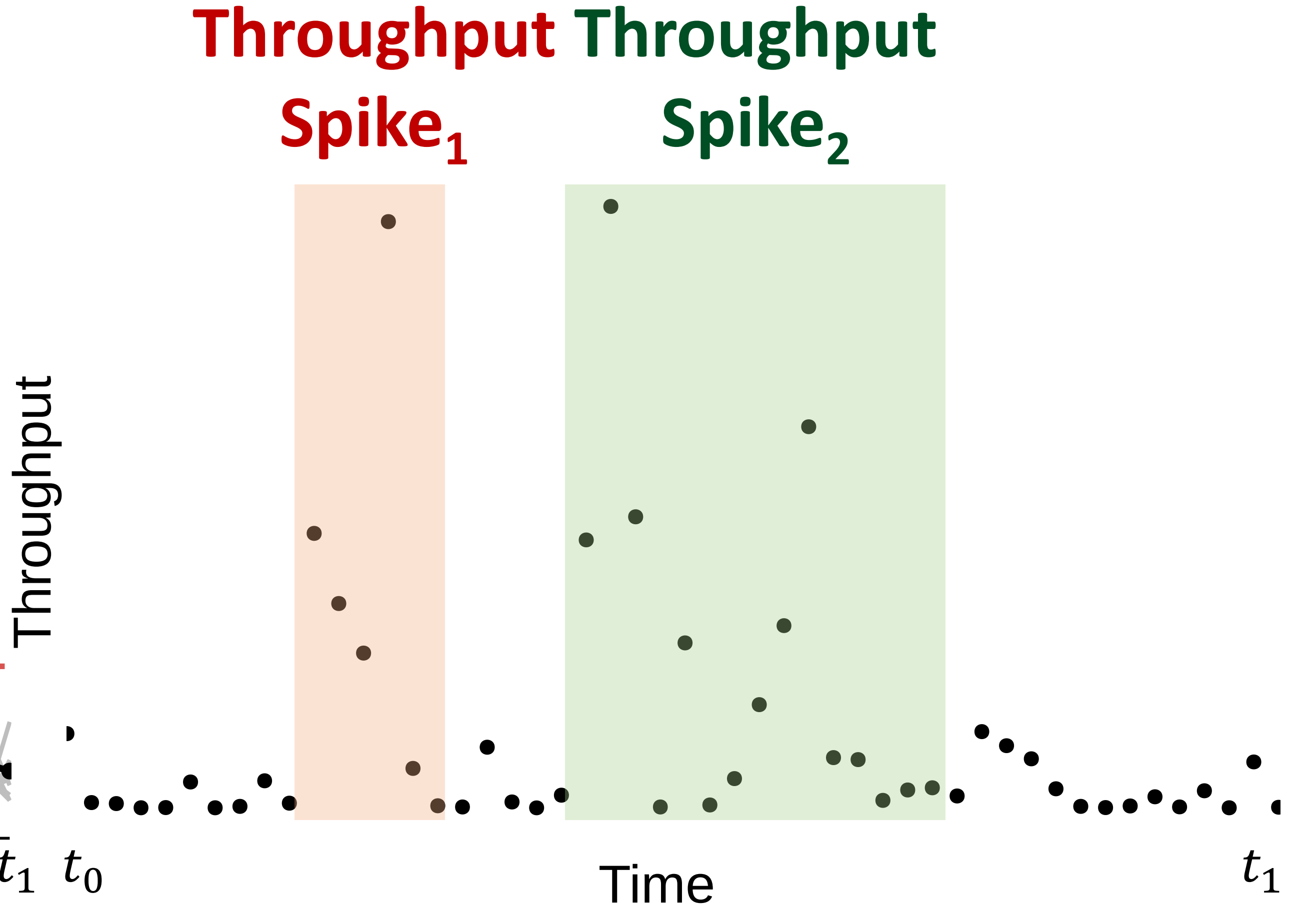


Failed Attempt: Threshold Filtering



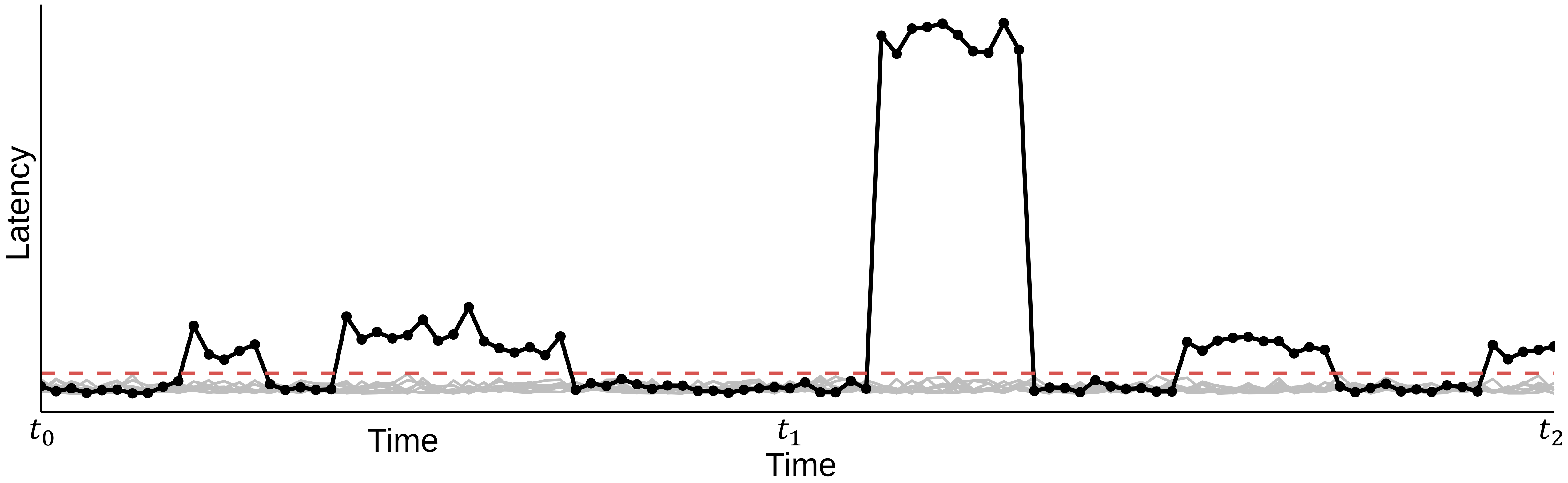
Failed Attempt: Threshold Filtering





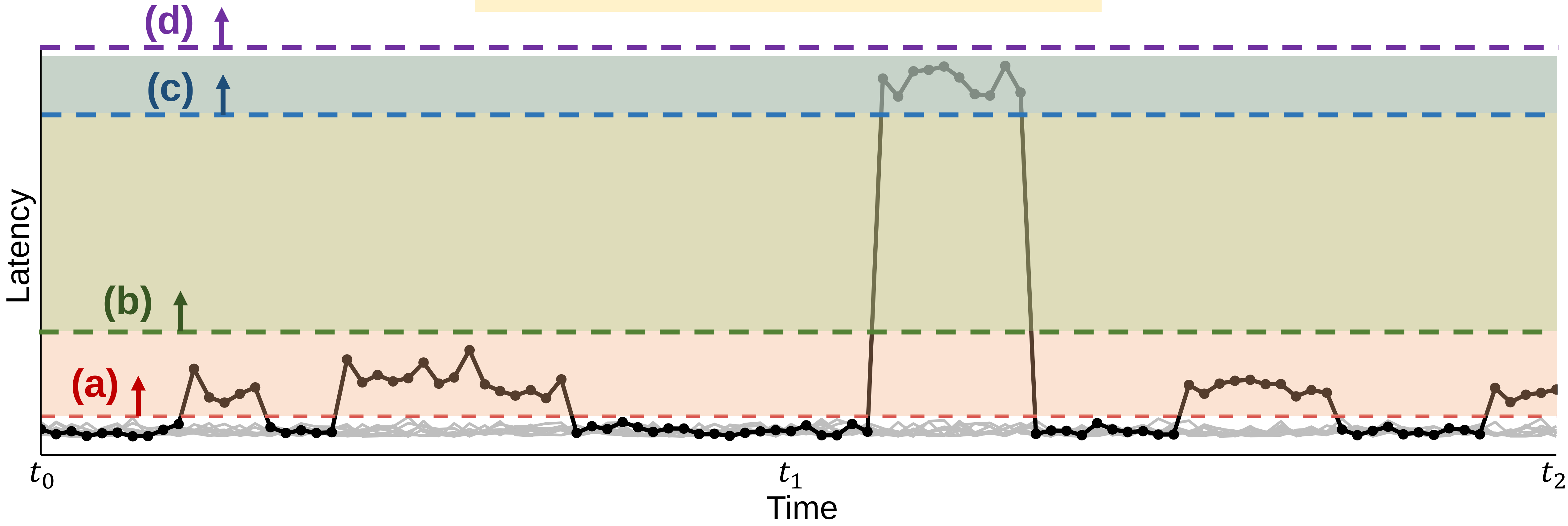
Workload bursts are common causes of latency variations

Failed Attempt: Threshold Filtering



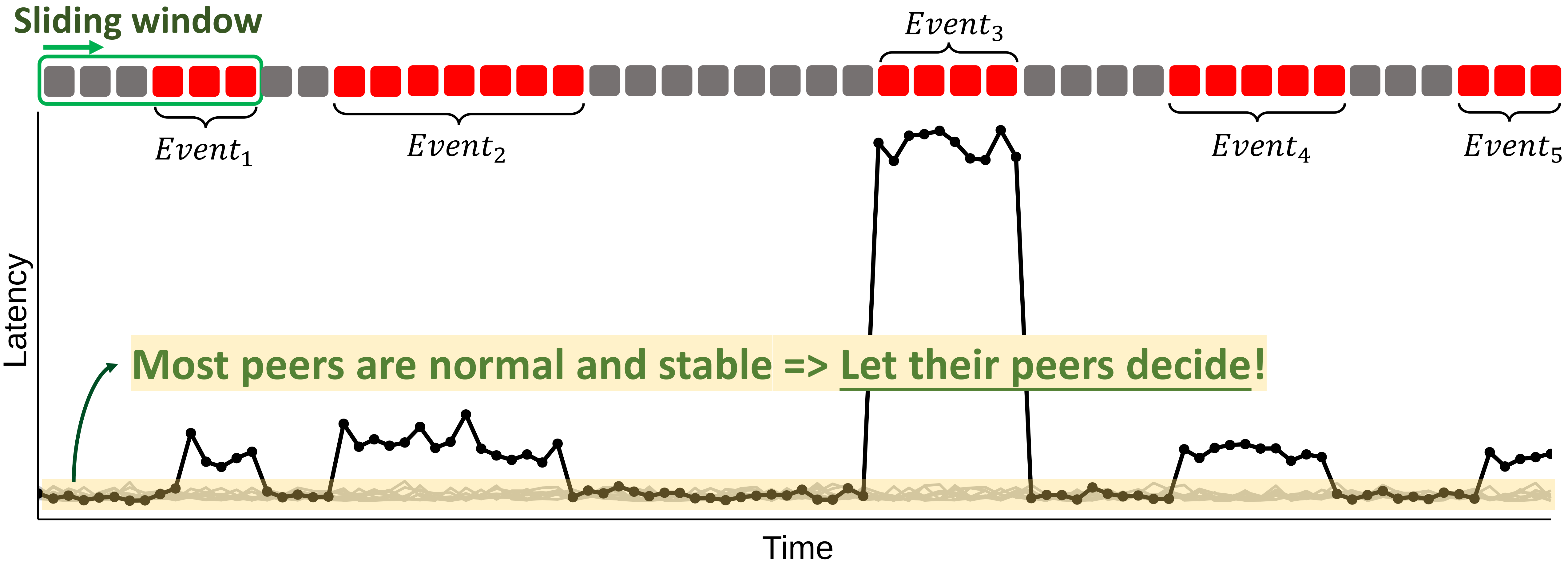
Failed Attempt: Threshold Filtering

Threshold are hard to decide



Dilemma {
- **Relaxed => More False Positives**
- **Strict => Many Fail-Slow Undiscovered**

Failed Attempt: Peer Evaluation



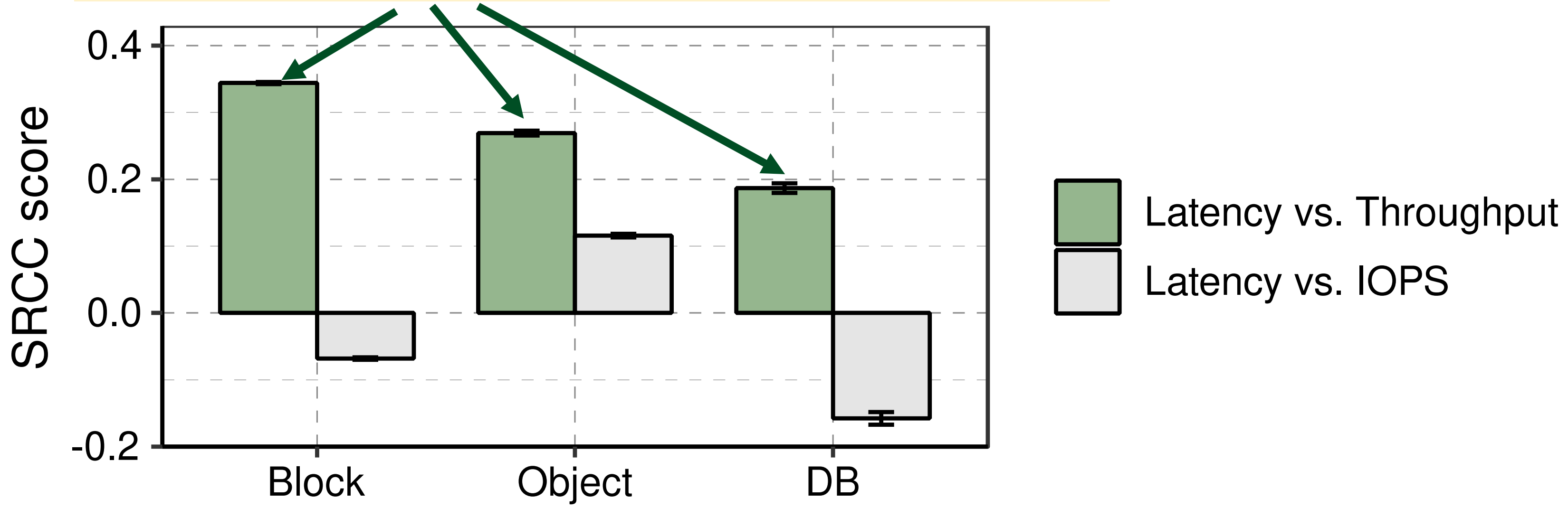
Time-consuming to fine-tune

Design Guidelines (I)

Insight: "Workload pressure can affect latency variations"

- Throughput or IOPS?

Correlation: Throughput >> IOPS



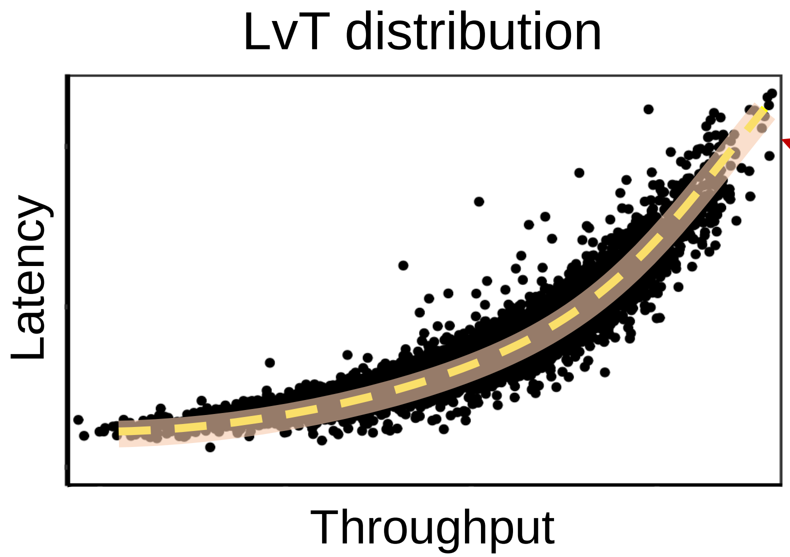
Guideline 1: Use throughput to model the workload pressure

Design Guidelines (II)

Insight: "Workload pressure can affect latency variations"

- How to model such a positive correlation?

Guideline 2: Model the latency-vs-throughput (LvT) distribution

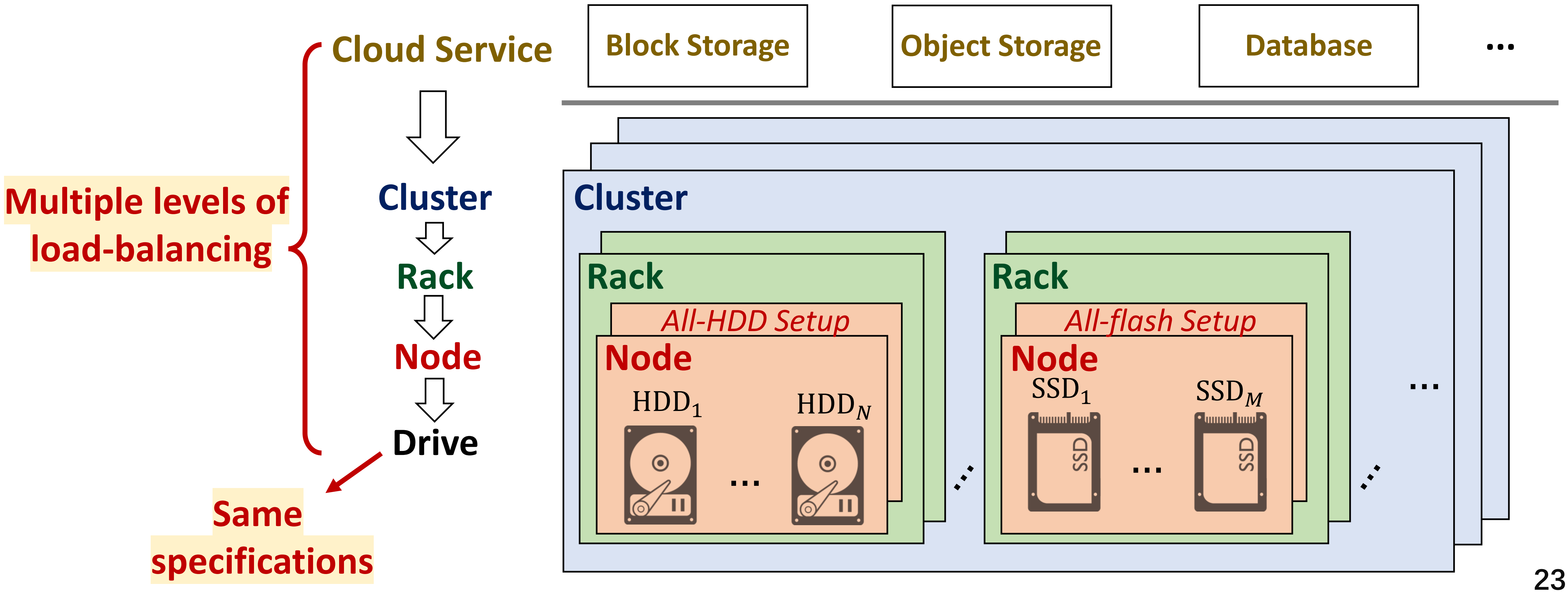


Linear regression?

=> Define a *statistically normal* drive

Design Guidelines (III)

Insight: "Homogeneous setup: similar workload + same drive spec"



Design Guidelines (III)

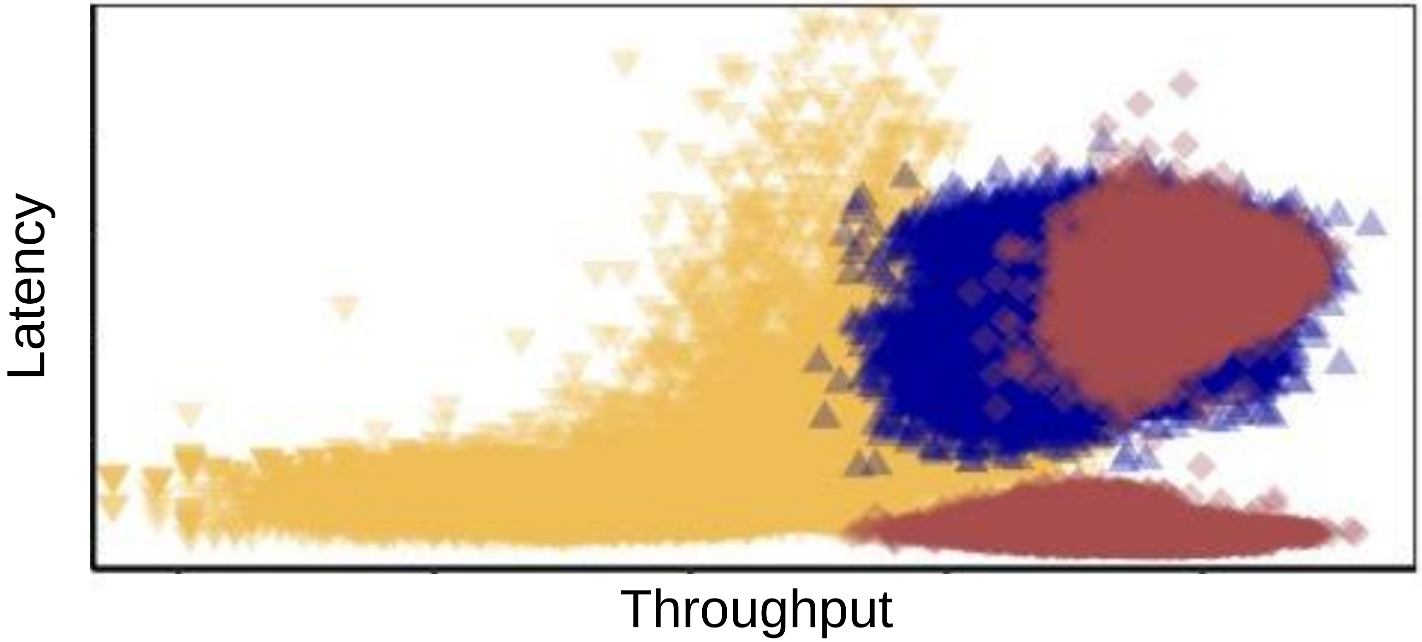
Insight: "Homogeneous setup: similar workload + same drive spec"

- **Need to determine the scope of drives to model**
  - **Drives from the same <u>service</u>?**
  - **Drives from the same <u>cluster</u>?**
  - **Drives from the same <u>node</u>?**

FAST'23 Design Guidelines (III)

Insight: "Homogeneous setup: similar workload + same drive spec"

(a) Same service, different clusters

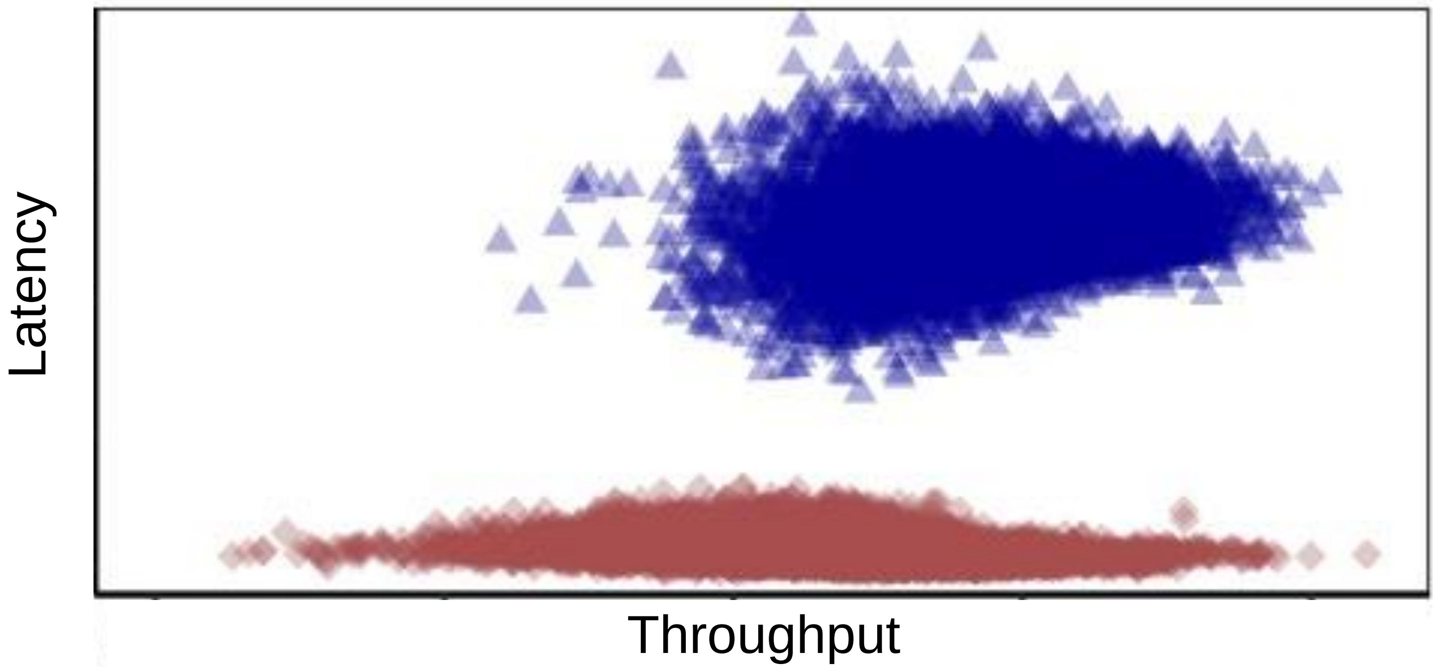


❌ Distinct LvT distributions

Design Guidelines (III)

Insight: "Homogeneous setup: similar workload + same drive spec"

(b) Same cluster, different nodes

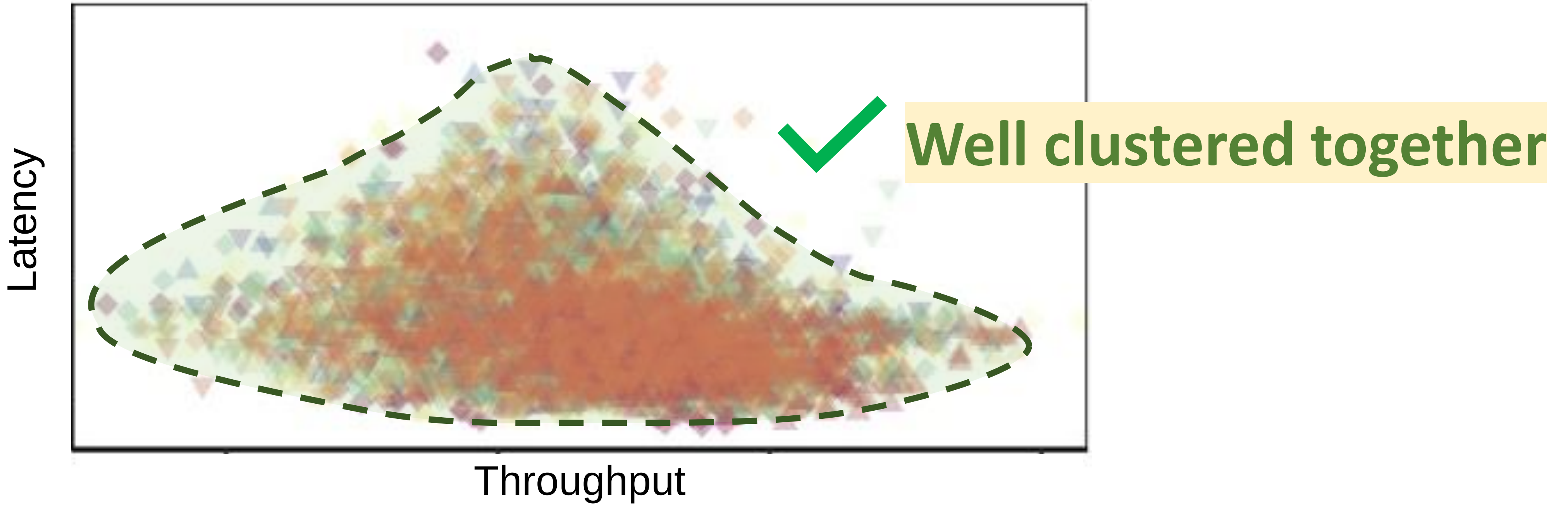


❌ **Distinct LvT distributions**

Design Guidelines (III)

Insight: "Homogeneous setup: similar workload + same drive spec"

(c) Same node, different drives



Guideline 3: Use node-wise samples to model LvT distribution

Design Guidelines (IV)

Insight: “No golden standards to identify fail-slow ”

Guideline 4: Non-binary output

- Model the likelihood of fail-slow

Outline

~~INTRODUCTION~~

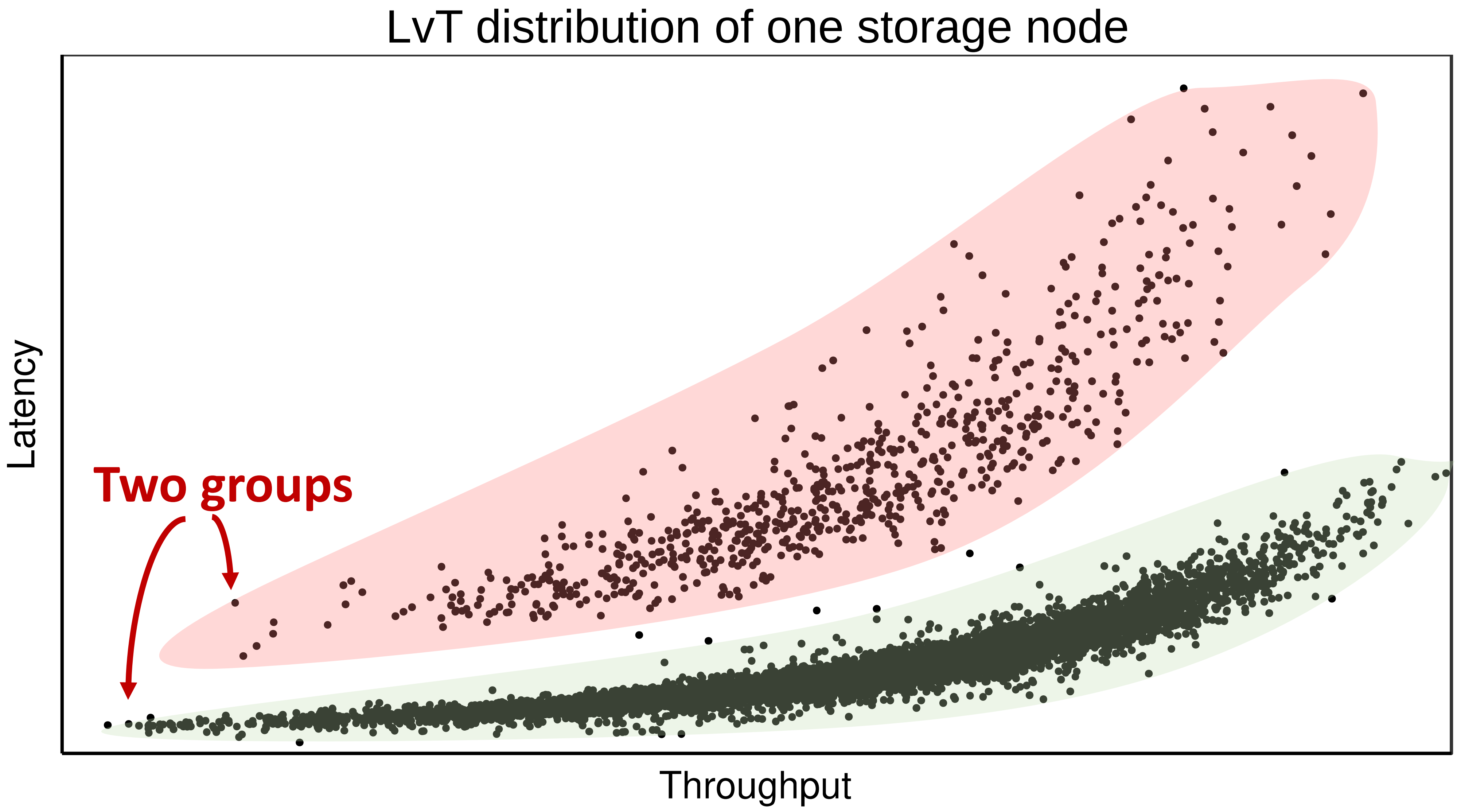
~~DATASET~~

~~FAILED ATTEMPTS~~

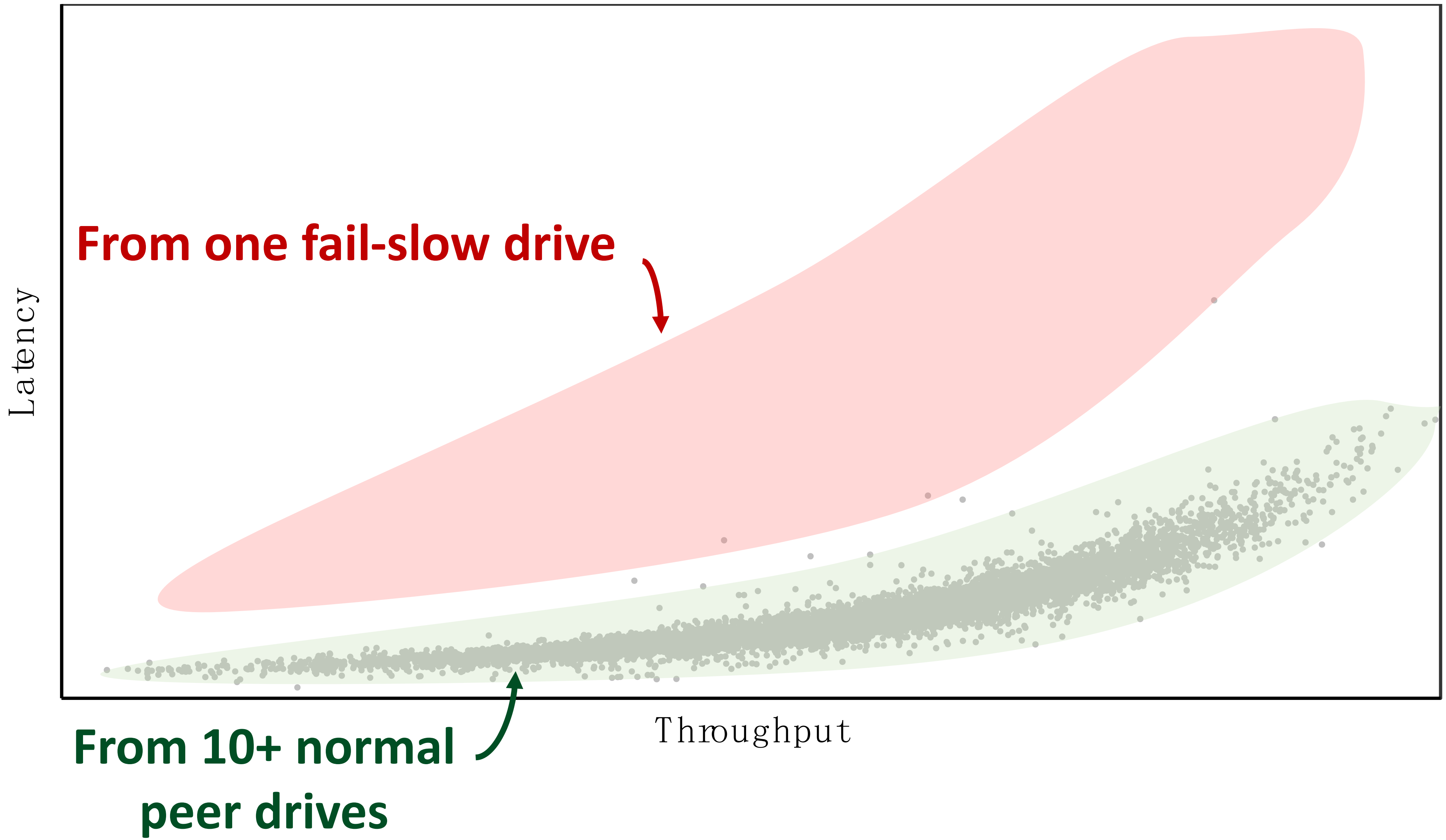
PERSEUS

EVALUATION & CONCLUSION

Raw Data

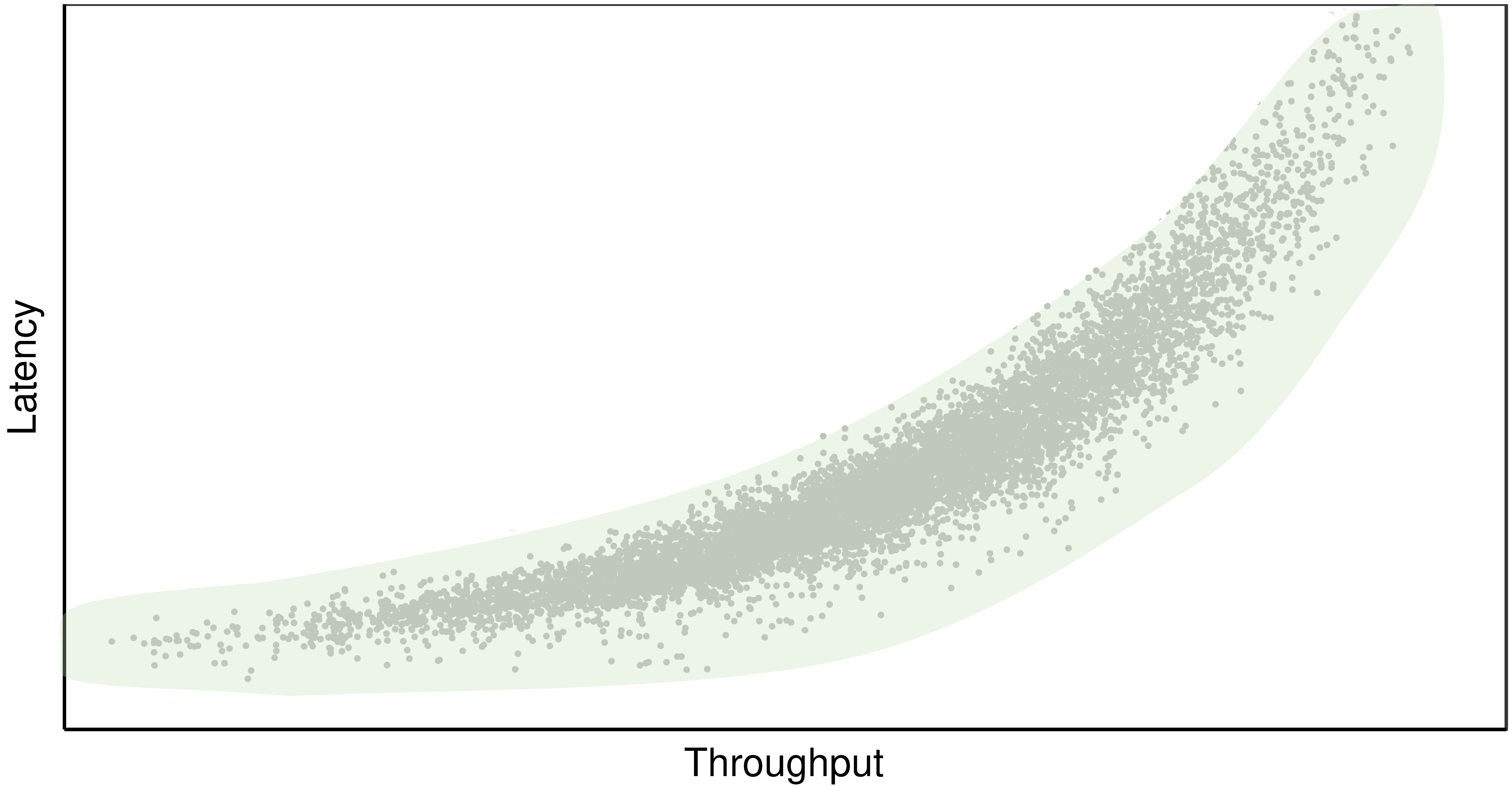

LvT distribution of one storage node
Latency
Throughput
Two groups

Raw Data

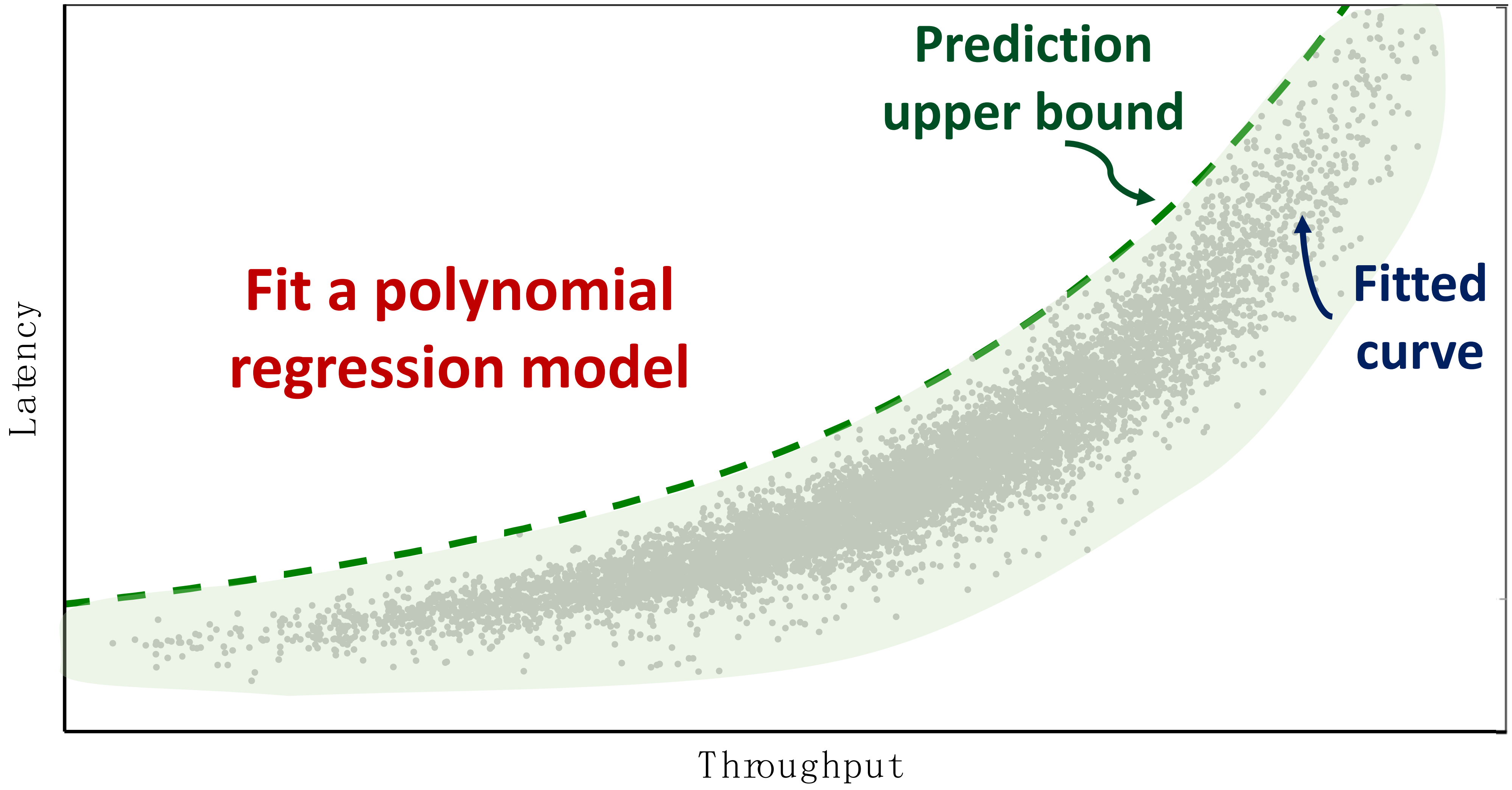


From one fail-slow drive
Latency
Throughput
From 10+ normal peer drives

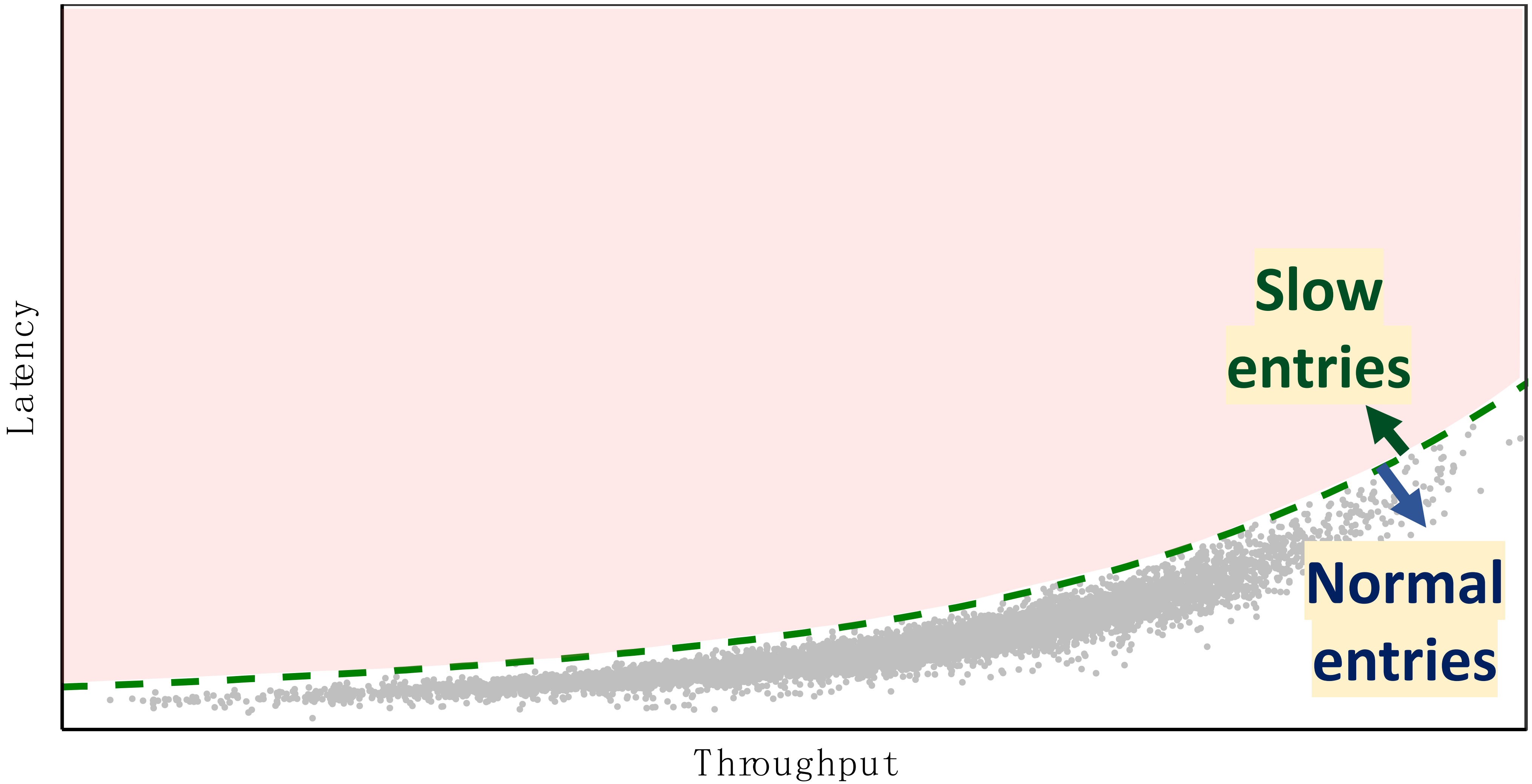
Step 1: Outlier Detection



Step 2: Building Regression Model

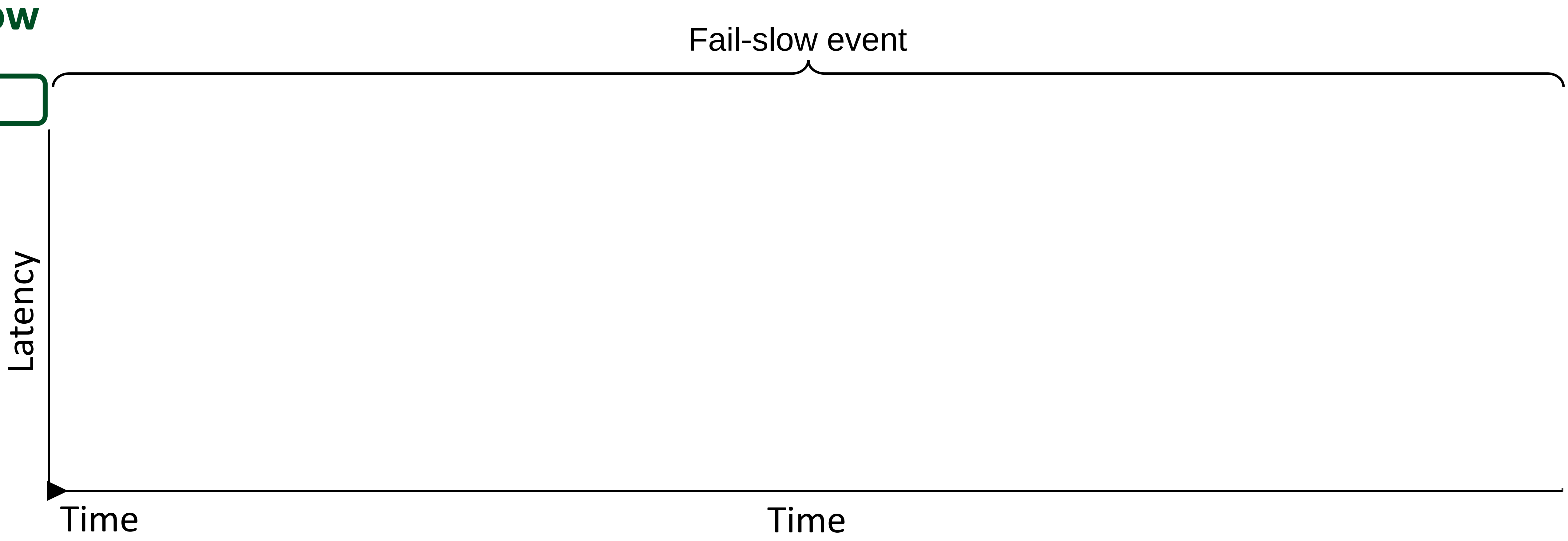


Step 2: Building Regression Model



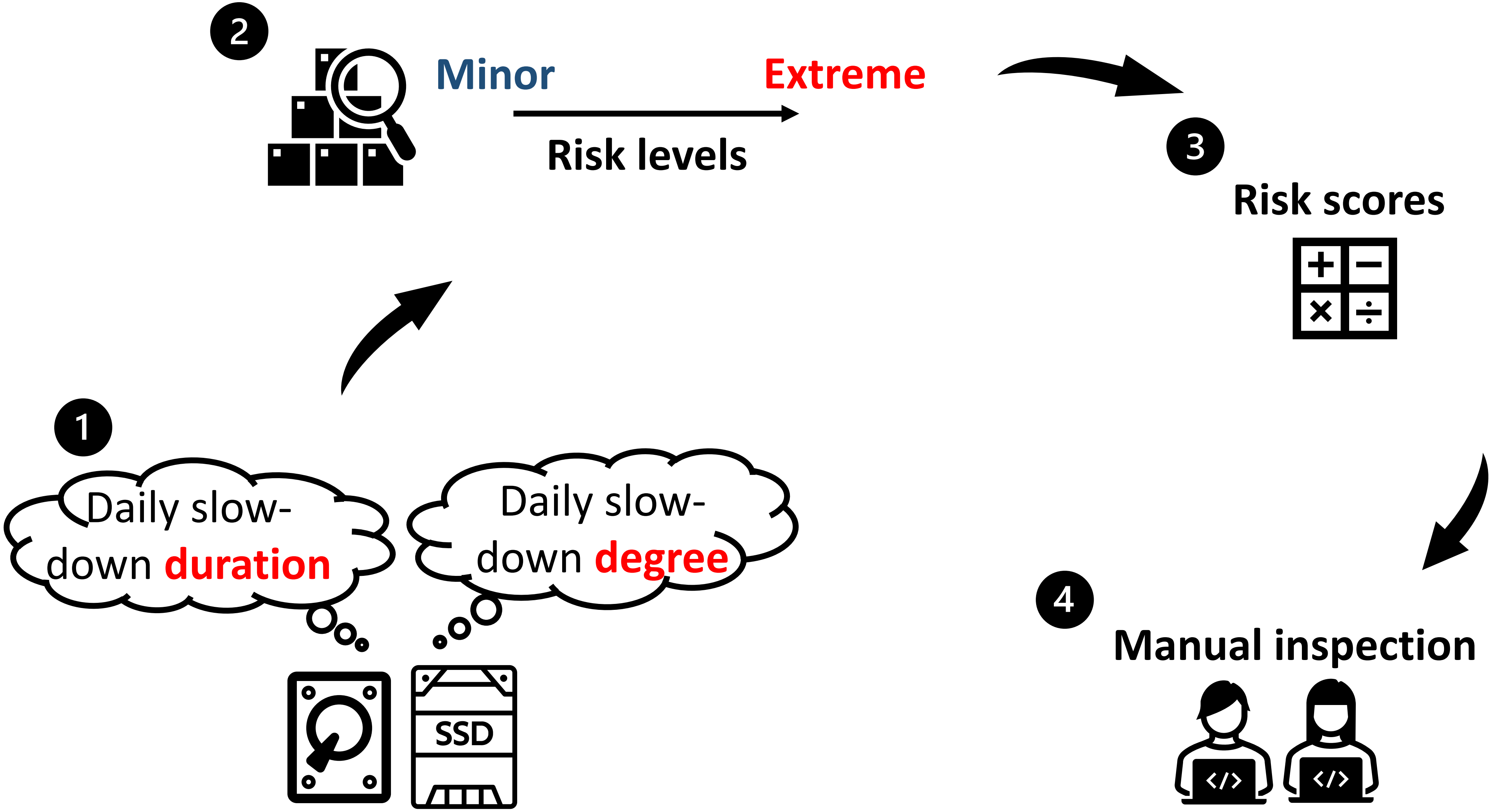
Prediction upper bounds as adaptive latency thresholds without fine-tuning

Step 3: Identifying Fail-Slow Event



Revisit the temporal dimension

Step 4: Evaluating Risk



Drive	Score
D1	70
D2	59
D3	20
…	

Quantify the slowness of drives

Outline

~~INTRODUCTION~~

~~DATASET~~

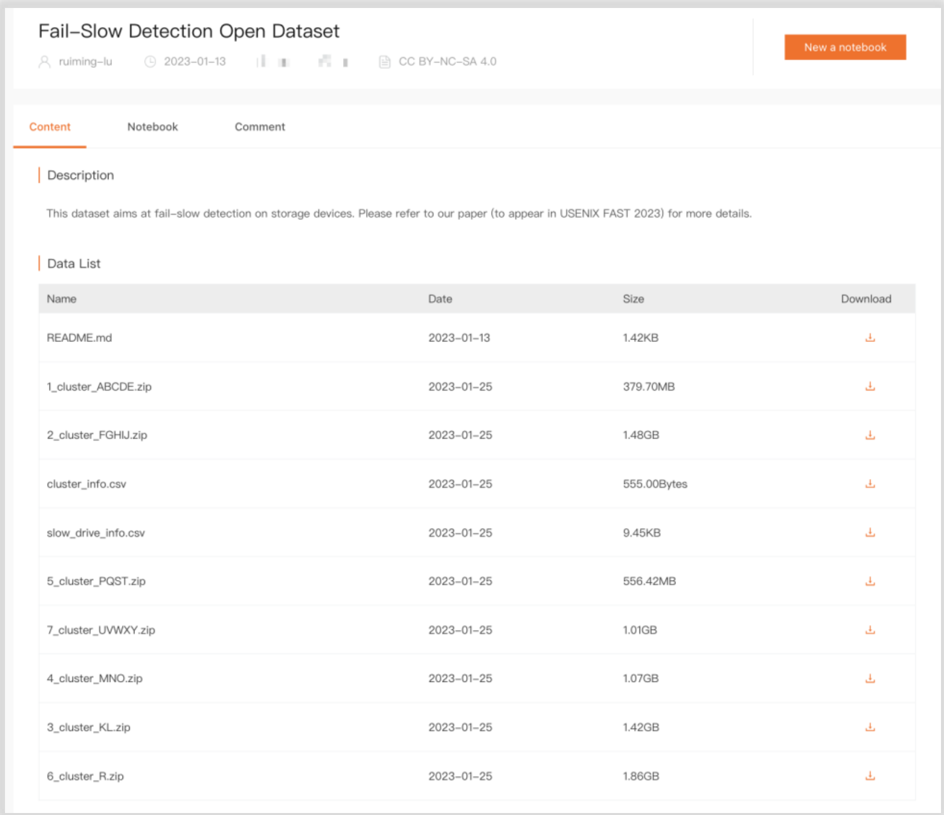
~~FAILED ATTEMPTS~~

~~PERSEUS~~

EVALUATION & CONCLUSION

Evaluation Benchmark

- **Built and released our self-assembled test dataset**
  - Clear labels (fail-slow or not)
  - 15 days of operational traces
  - 41K drives
  - ~300 fail-slow drives


Fail-Slow Detection Open Dataset

ruiming-lu 2023-01-13 CC BY-NC-SA 4.0

New a notebook

Content Notebook Comment

Description

This dataset aims at fail-slow detection on storage devices. Please refer to our paper (to appear in USENIX FAST 2023) for more details.

Data List

Name	Date	Size	Download
README.md	2023-01-13	1.42KB	
1_cluster_ABCDE.zip	2023-01-25	379.70MB	
2_cluster_FGHIJ.zip	2023-01-25	1.48GB	
cluster_info.csv	2023-01-25	555.00Bytes	
slow_drive_info.csv	2023-01-25	9.45KB	
5_cluster_PQST.zip	2023-01-25	556.42MB	
7_cluster_UVWXY.zip	2023-01-25	1.01GB	
4_cluster_MNO.zip	2023-01-25	1.07GB	
3_cluster_KL.zip	2023-01-25	1.42GB	
6_cluster_R.zip	2023-01-25	1.86GB	

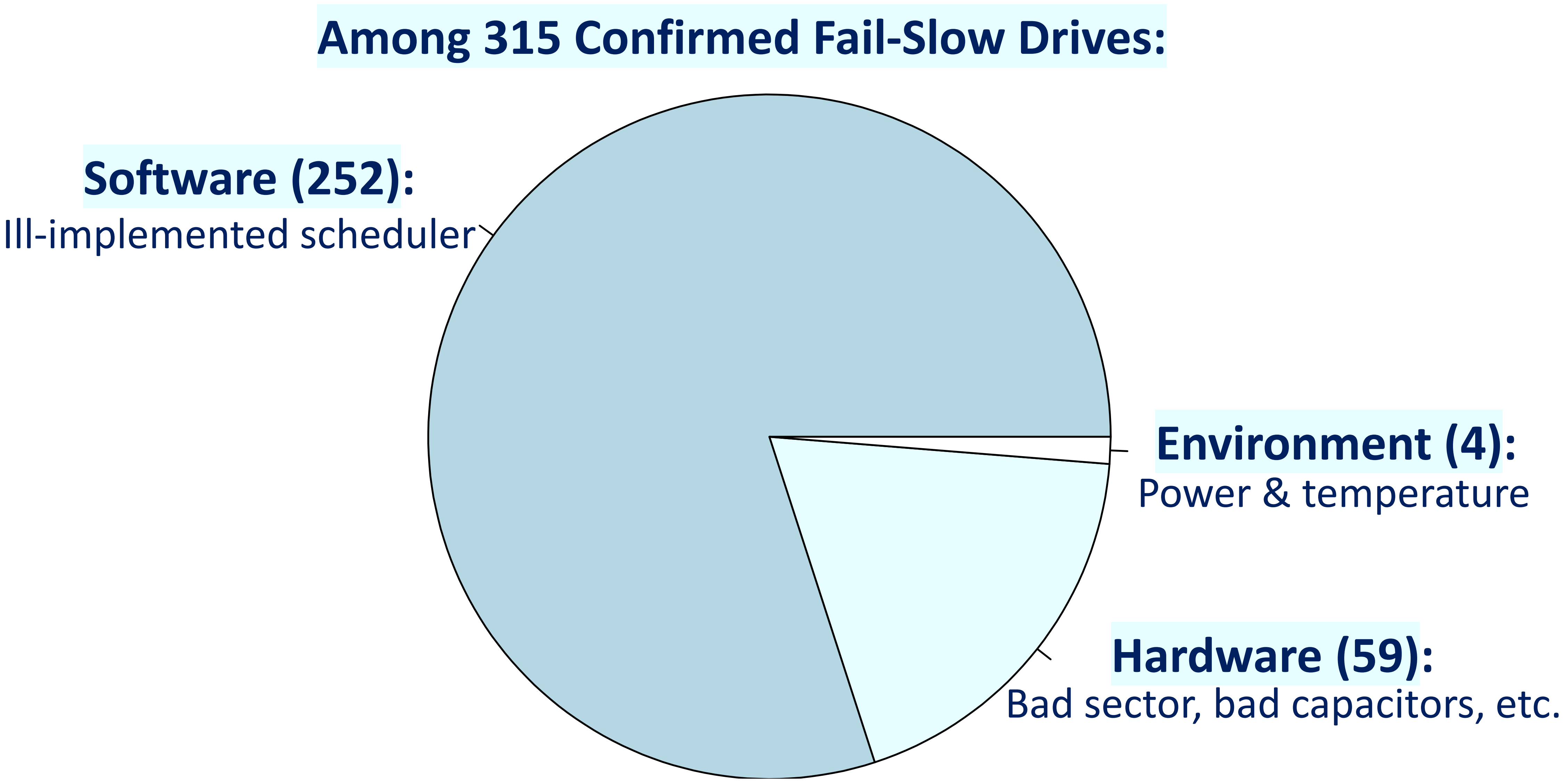
https://tianchi.aliyun.com/dataset/144479

Evaluations

- **Perseus outperforms all previous attempts (§5.4)**
- **Effectiveness of Perseus's Design Choices (§5.5)**
- **Reduce Tail Latency By 31-48% (§5.6)**
- **Root Cause Analysis (§6)**

More details in the paper!

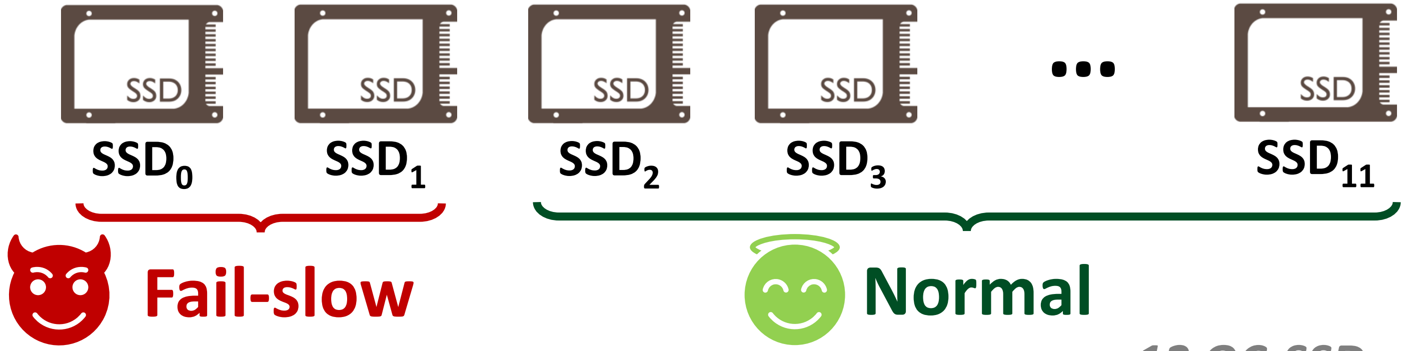
Root Cause Distribution



Root Cause: Software

Case I: In Open-Channel SSD Cluster

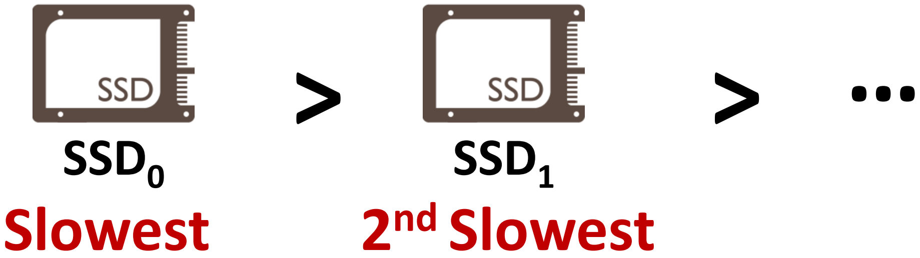
1. Every node always has 2 fail-slow drives



12 OC-SSDs per node

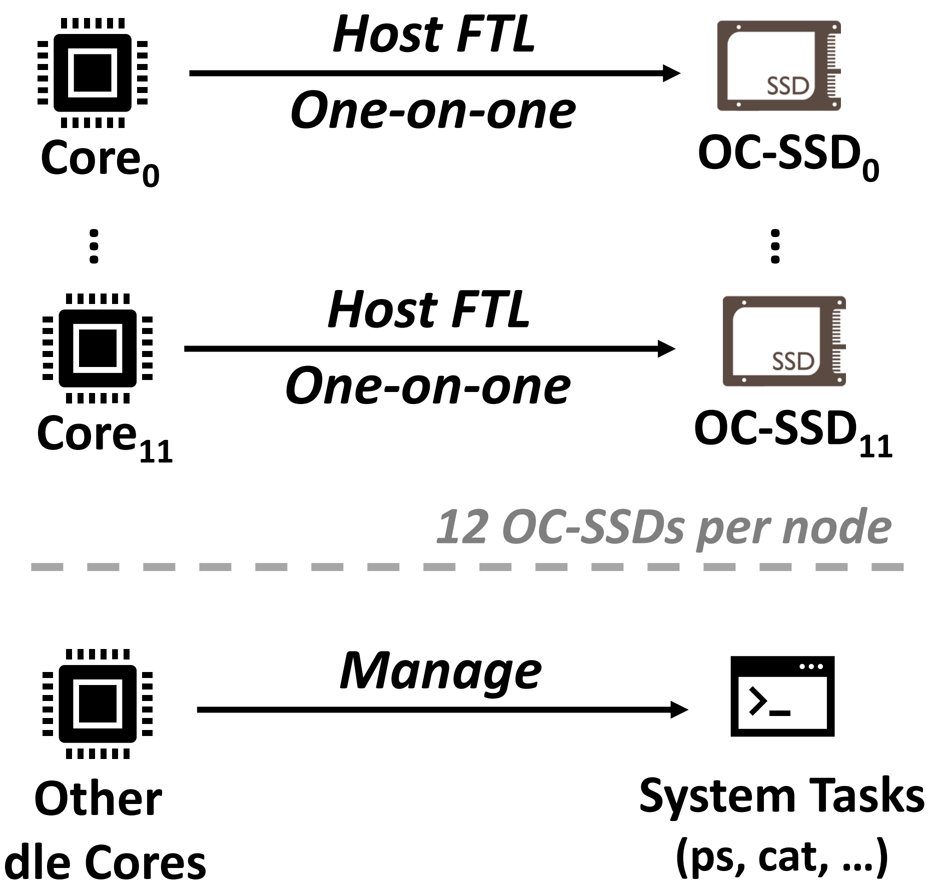
2. Latency levels follow ascending order of logical IDs

Latency levels:

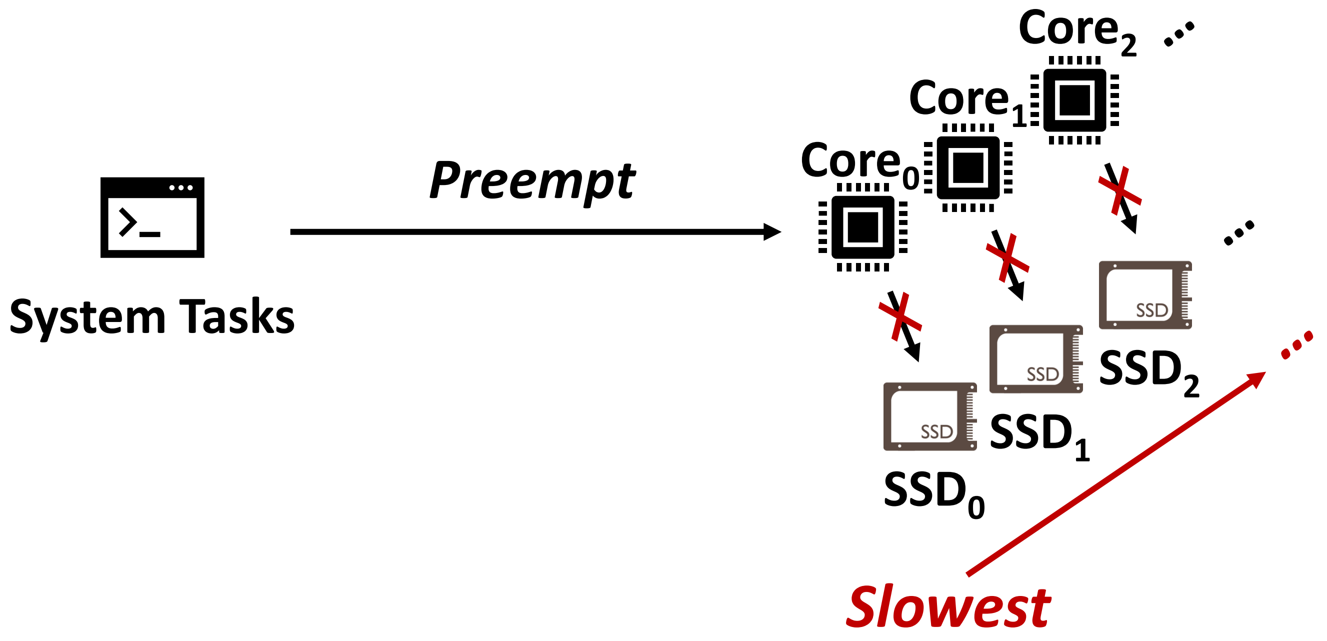


Root Cause: Software

Case I: In Open-Channel SSD Cluster



If No Idles Cores Available…



=> Fail-Slow OC-SSDs in Ascending Order

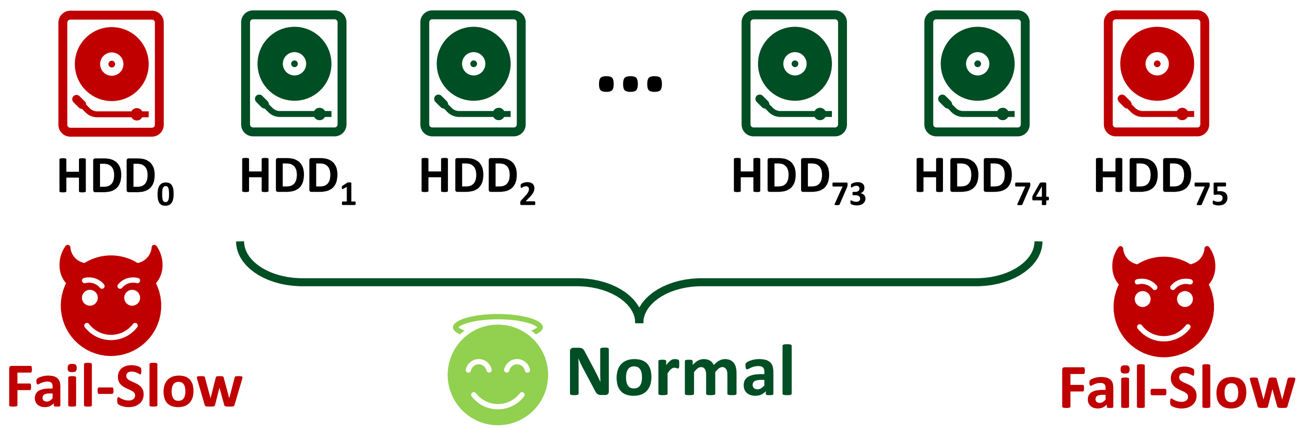
Root Cause: Software

Case II: In All-HDD Cluster

1. Fail-slow drives always appear in fixed pairs

76 HDDs per node

Two fail-slow disks in a node:



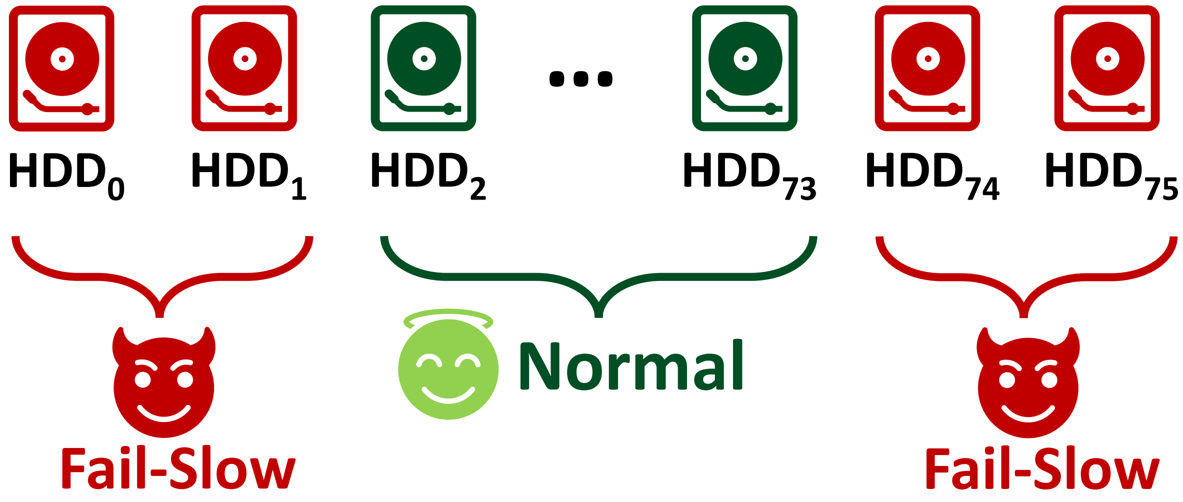
Root Cause: Software

Case II: In All-HDD Cluster

1. Fail-slow drives always appear in fixed pairs

76 HDDs per node

Four fail-slow disks in a node:



Root Cause: Software

Case II: In All-HDD Cluster

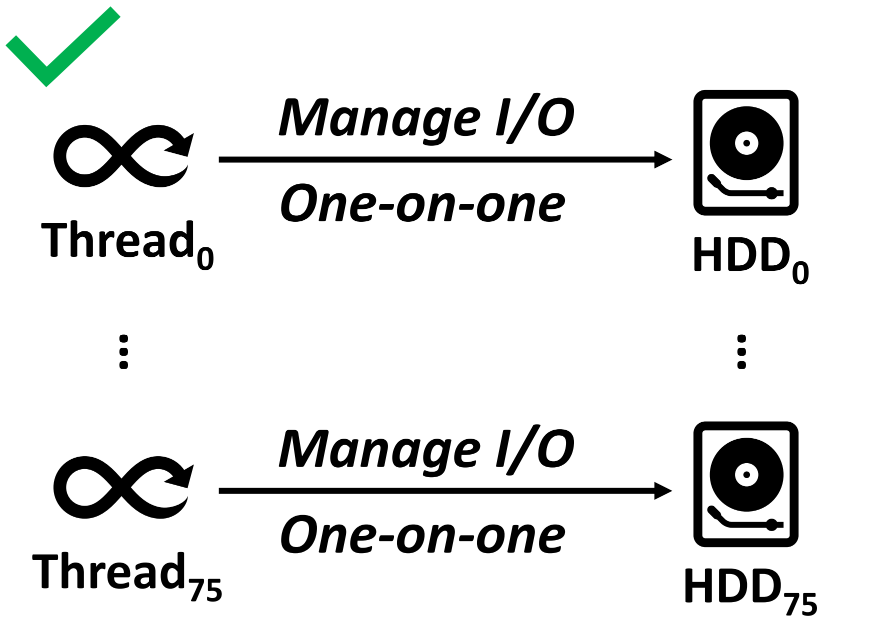
1. Fail-slow drives always appear in fixed pairs

2. All fail-slow drives are experiencing similar slowdown

3. $\#\textit{Fail-slow} = 2 \times \#\textit{Offline}$

Root Cause: Software

Case II: In All-HDD Cluster

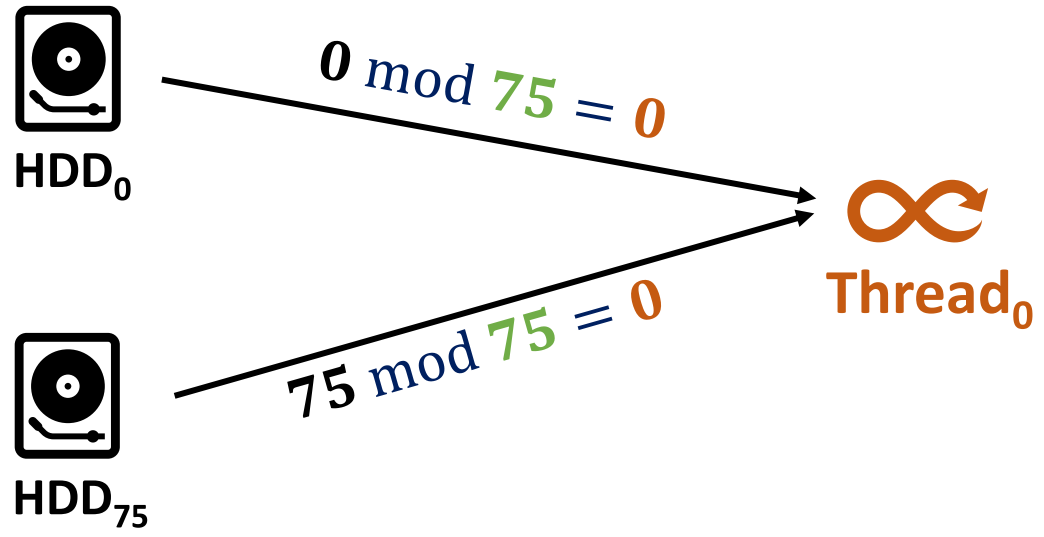


76 HDDs per node

$$Thread_{ID} = Disk_{ID} \bmod \#Drives$$

When Disks Taken Down For Repair...

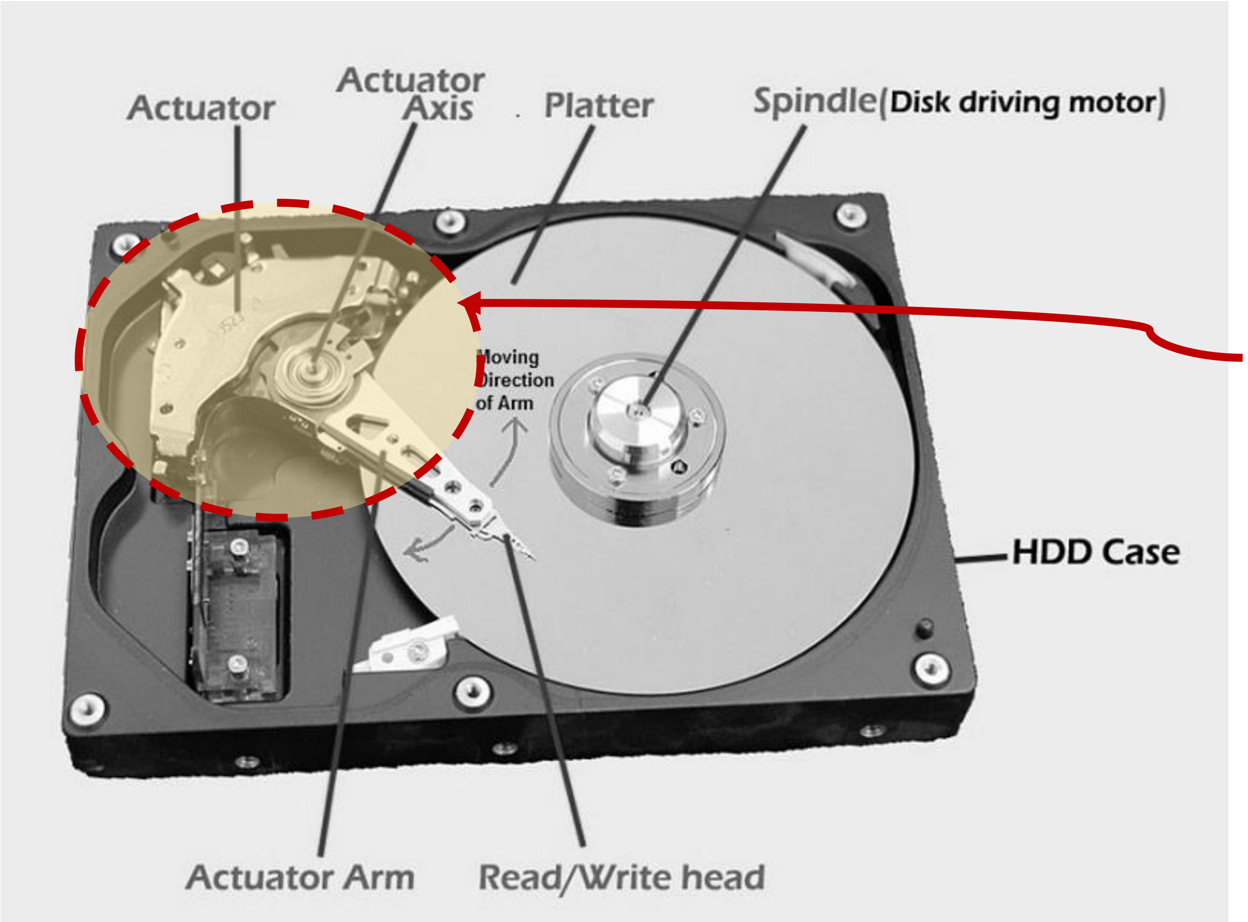
e.g., HDD$_{20}$ taken down $\#Drives = 75$



=> Resource Contention-Induced Fail-Slow

Root Cause: Hardware

- **Rotor Eccentricity**



Source: https://www.techintangent.com/hard-disk-description/

Frequently miss targeted positions

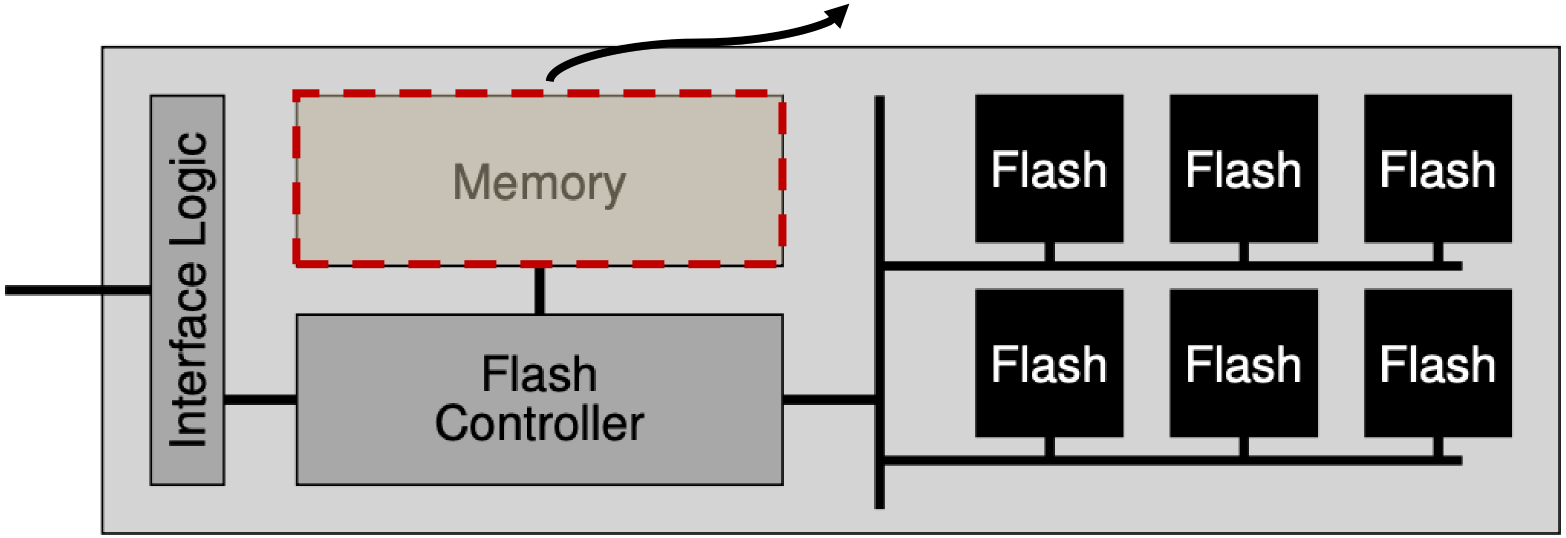
⇓

I/O delay

Root Cause: Hardware

- Rotor Eccentricity
- <u>Bad Capacitors</u>

DRAM as an internal write-back cache



Source: Operating Systems: Three Easy Pieces

DRAM capacitors failed => Delayed writes

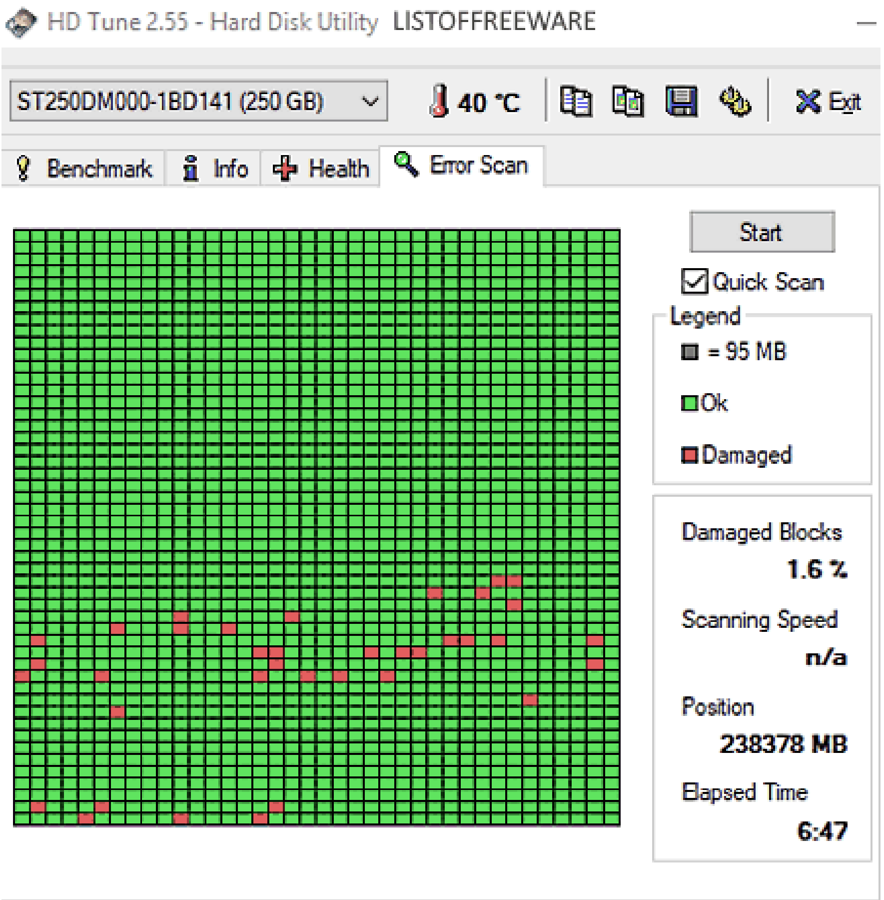
Root Cause: Hardware

- Rotor Eccentricity
- Bad Capacitors
- <u>Bad Sectors</u>:

Data reallocate to spare sectors

⇓

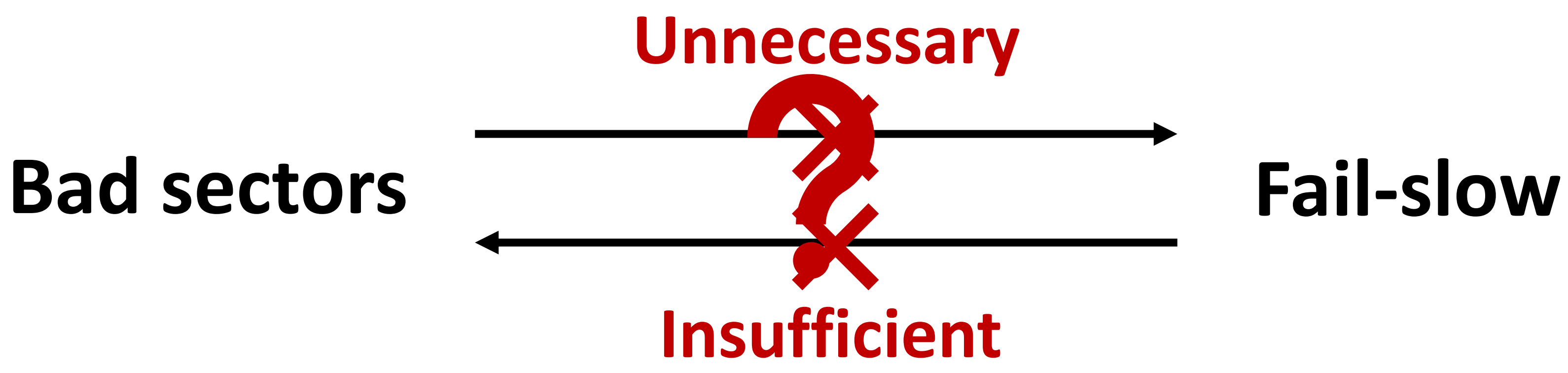
Long seek time



Source: https://listoffreeware.com/best-free-software-to-check-hard-drive-bad-sectors-windows/

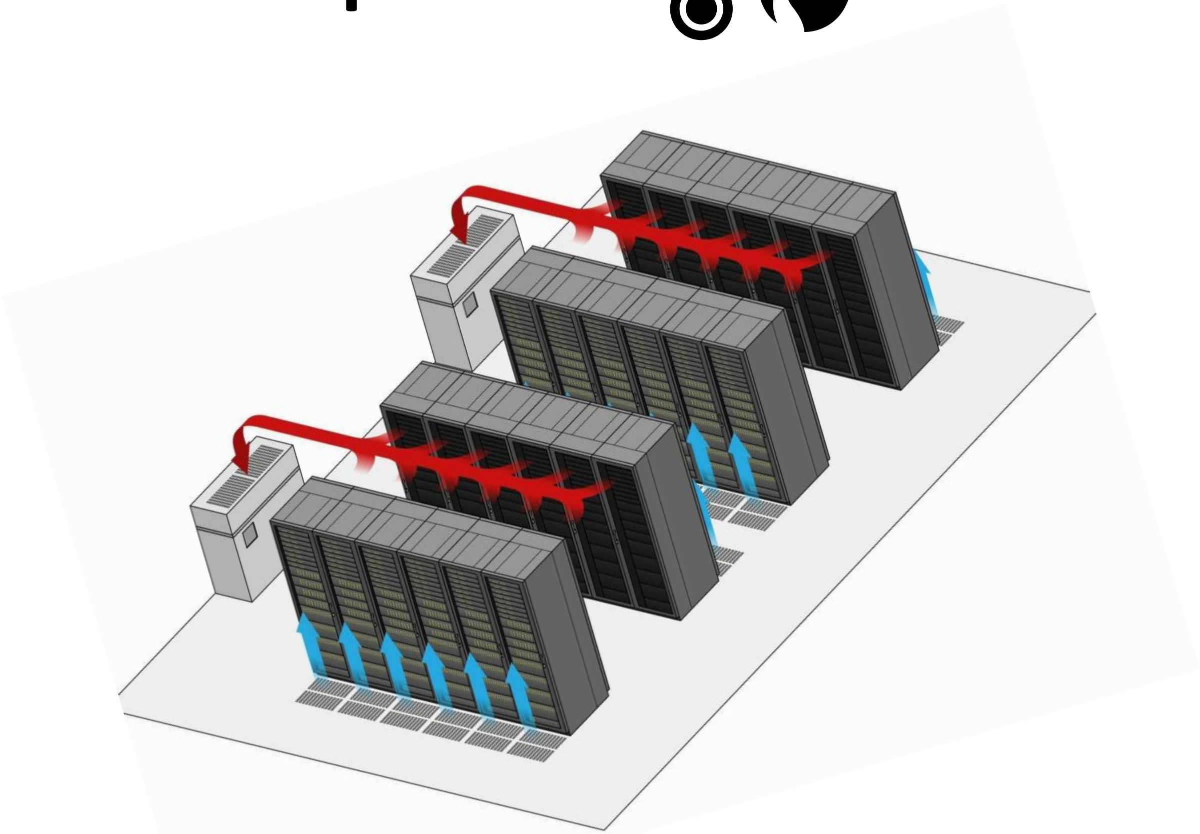
Root Cause: Hardware

- Rotor Eccentricity
- Bad Capacitors
- <u>Bad Sectors</u>:



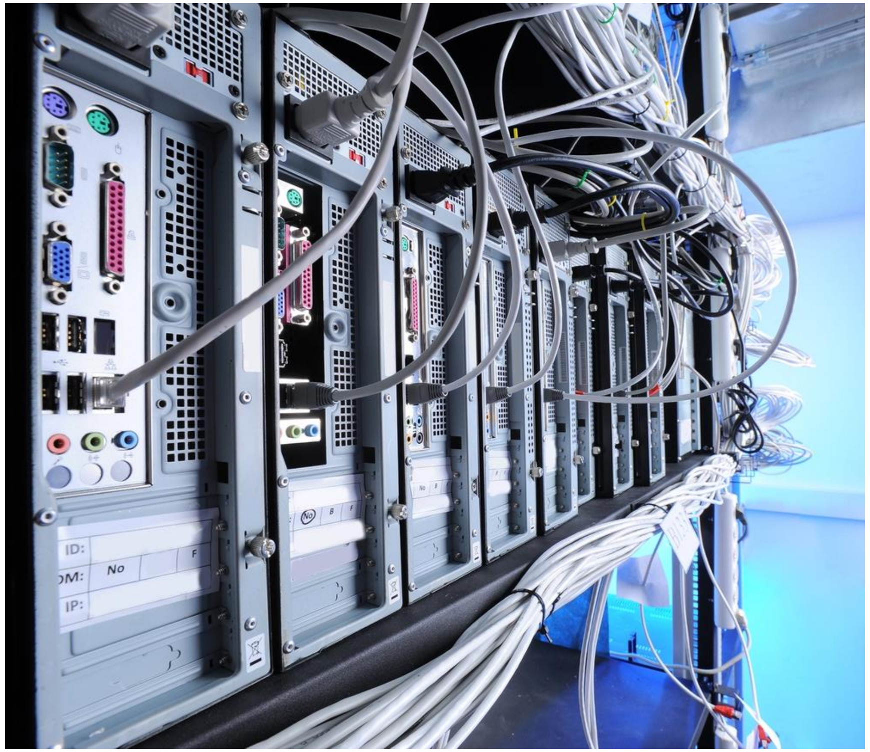
Root Cause: Environment

- **Temperature**



Source: https://www.upsite.com/blog/helping-your-data-center-breath-easier-with-good-air-flow-management/

- **Power**



Source: https://www.ecsintl.com/how-data-centers-can-effectively-manage-power-surges/

Summary

Perseus

Detection Framework

Efficient Fail-Slow Detection

Non-intrusive
(Performance) log-based
No source code altering

Accurate
Recall/precision rate > 0.99

Fine-grained
Device-level detection

General
One set of parameters fits all scenarios

Storage Devices

…

Adaptable to Other Problem Domains

Thank you!

Perseus: A Fail-Slow Detection Framework
for Cloud Storage Systems

Ruiming Lu, Erci Xu, Yiming Zhang, Fengyi Zhu, Zhaosheng Zhu,
Mengtian Wang, Zongpeng Zhu, Guangtao Xue, Jiwu Shu, Minglu Li, Jiesheng Wu

Contact email: lrm318@sjtu.edu.cn

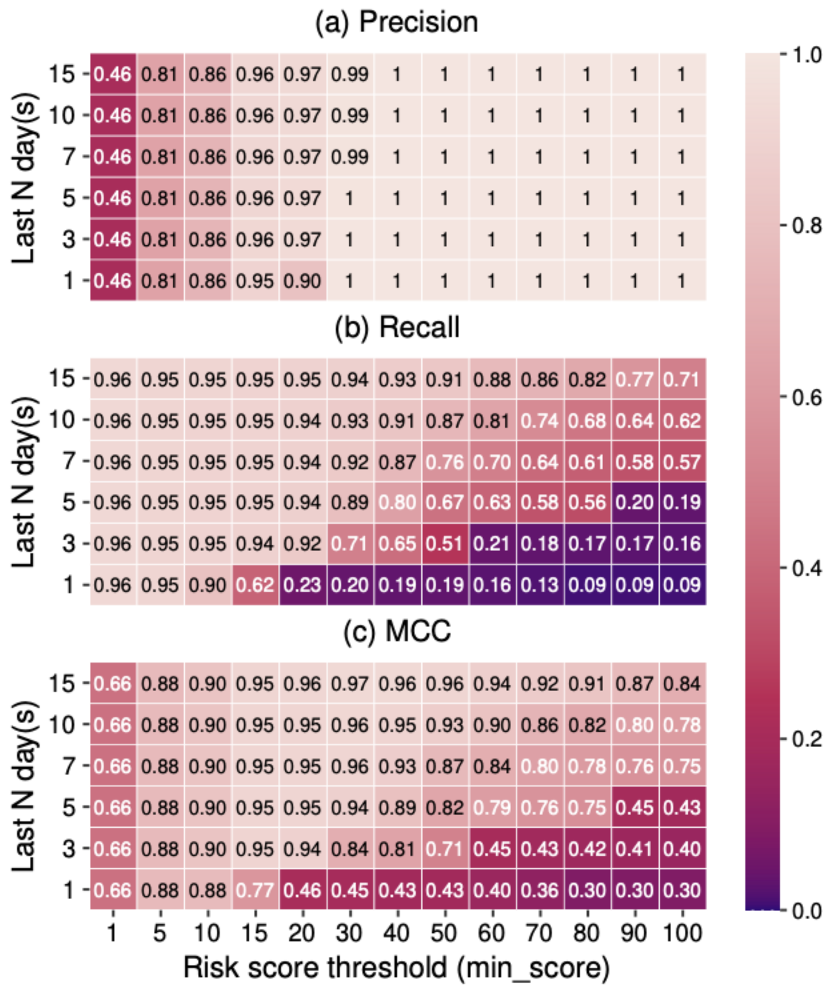
FAST'23 Evaluations

Effectiveness of Perseus's design choices (§5.5)

Perseus's design tradeoffs (§5.5)

Parameter	Range	Description
S1: Outlier detection (§4.2)		
PCA	On/Off	Transform the coordinates w.r.t. the principal components.
DBSCAN	On/Off	Density-based outlier detection.
S3: Identifying fail-slow event (§4.4)		
X	95∼99.9	Use the *X*% prediciton upper bound as the latency upper bound.
S4: Evaluating risk (§4.5)		
min_score	1∼100	Risk score threshold.
N	1∼15	Evaluate the risk score of the most recent *N* days.

Metric	w/o Outlier	w/o PCA	p95	p99	p999	Deployed
Full-set						
Precision	0.98	0.55	0.99	1.00	1.00	0.99
Recall	0.51	0.43	0.99	0.93	0.93	1.00
MCC	0.71	0.49	0.99	0.96	0.96	0.99
Subset (excluding software-induced)						
Precision	0.95	0.36	0.94	0.98	1.00	0.94
Recall	0.82	0.91	0.95	0.92	0.95	1.00
MCC	0.88	0.57	0.95	0.95	0.98	0.97



Metric	Thresh-Stat	Thresh-Emp	Peer Eval	IASO-Based	PERSEUS-Deployed
Full-set					
Precision	1.00	1.00	0.98	0.48	0.99
Recall	0.52	0.02	0.57	0.24	1.00
MCC	0.72	0.14	0.74	0.32	0.99
Subset (excluding software-induced)					
Precision	1.00	1.00	1.00	0.45	0.94
Recall	0.71	0.09	0.65	0.61	1.00
MCC	0.84	0.30	0.80	0.52	0.97