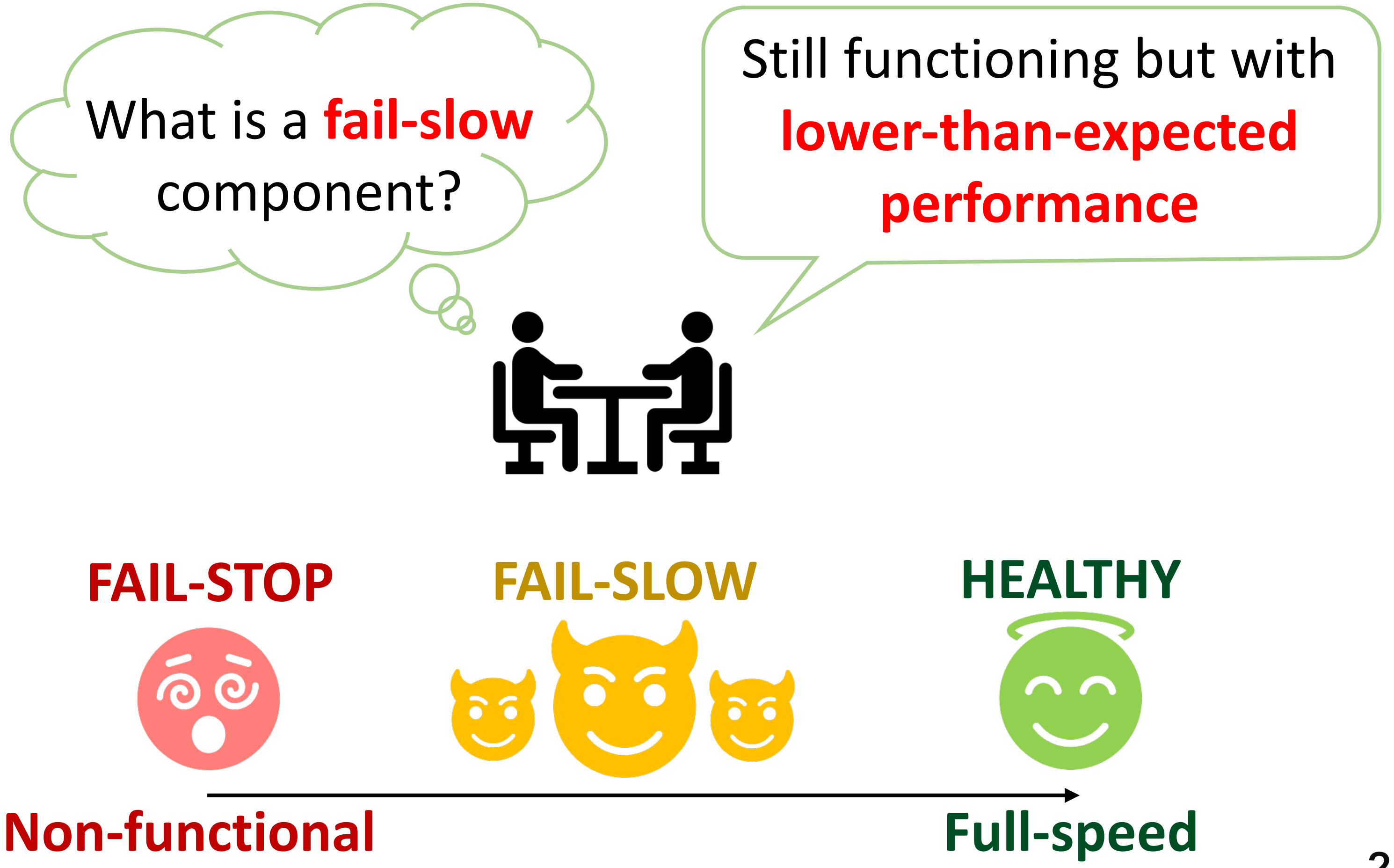
21st USENIX Conference on File and Storage Technologies FAST'23

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
  - …



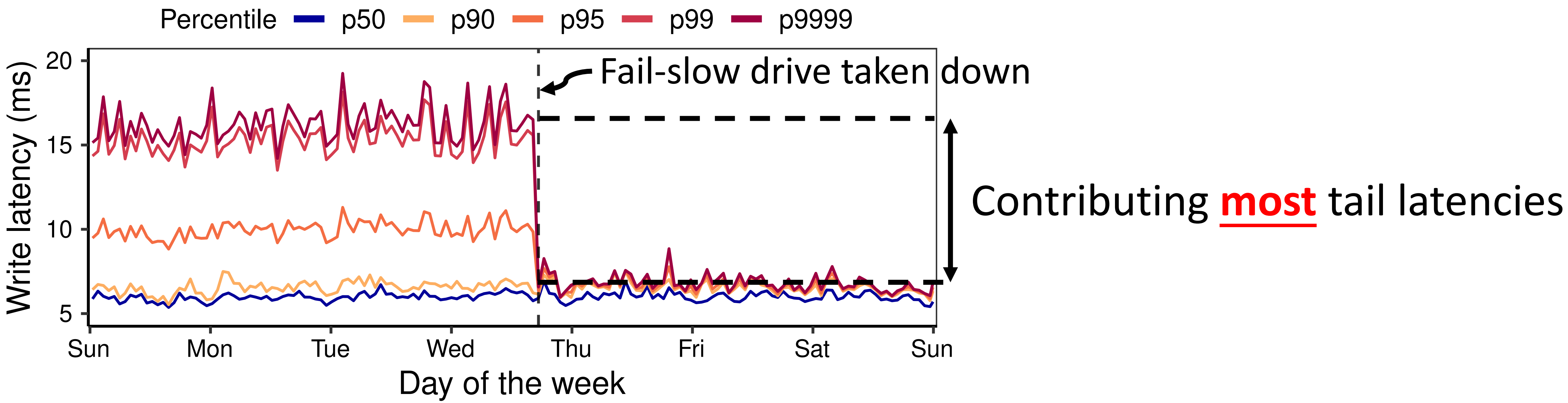
Not A Problem?

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.

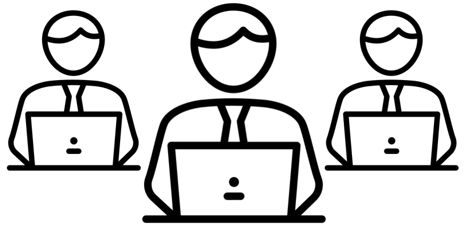
Not A Problem?



FAIL-SLOW

Slow! Slow! Slow! Silent performance degradation!

Let's "dig" them out!



Hard To Detect

- **No Ground Truth in <u>Identifying Fail-Slow</u>**

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

FAST'23

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~~

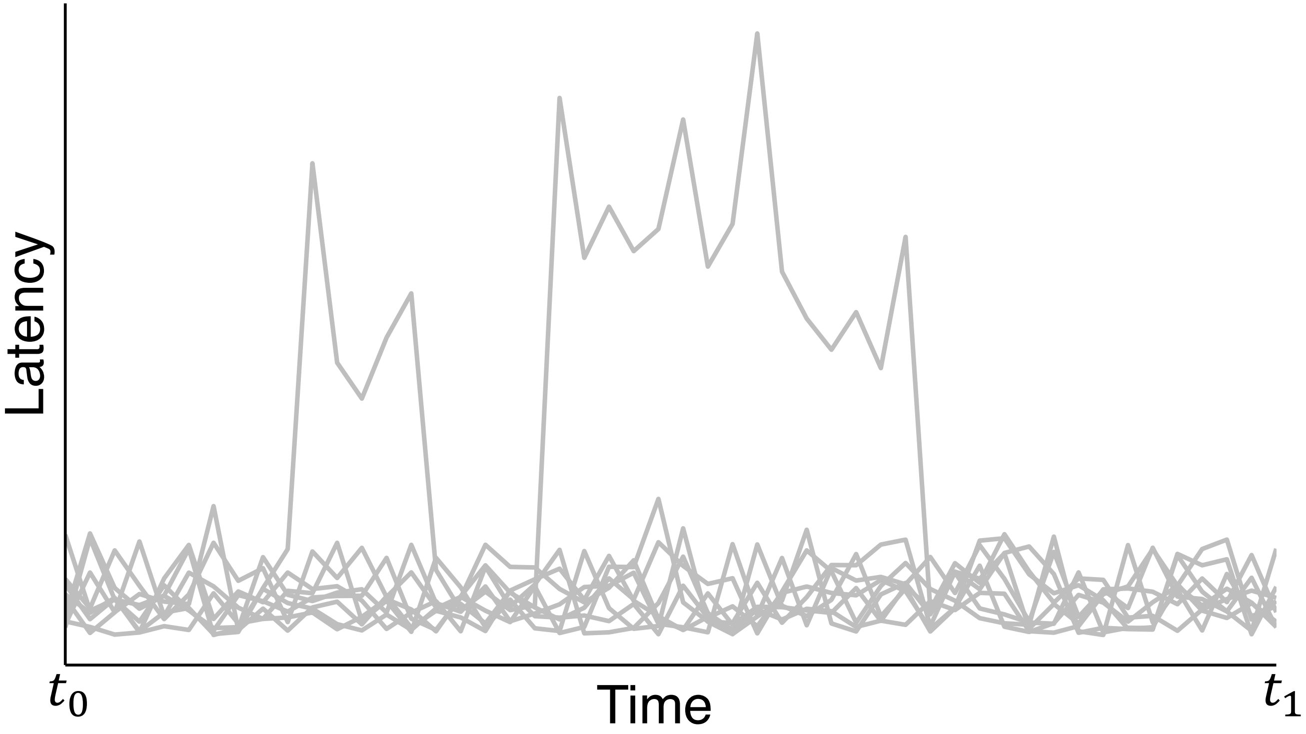
~~DATASET~~

FAILED ATTEMPTS

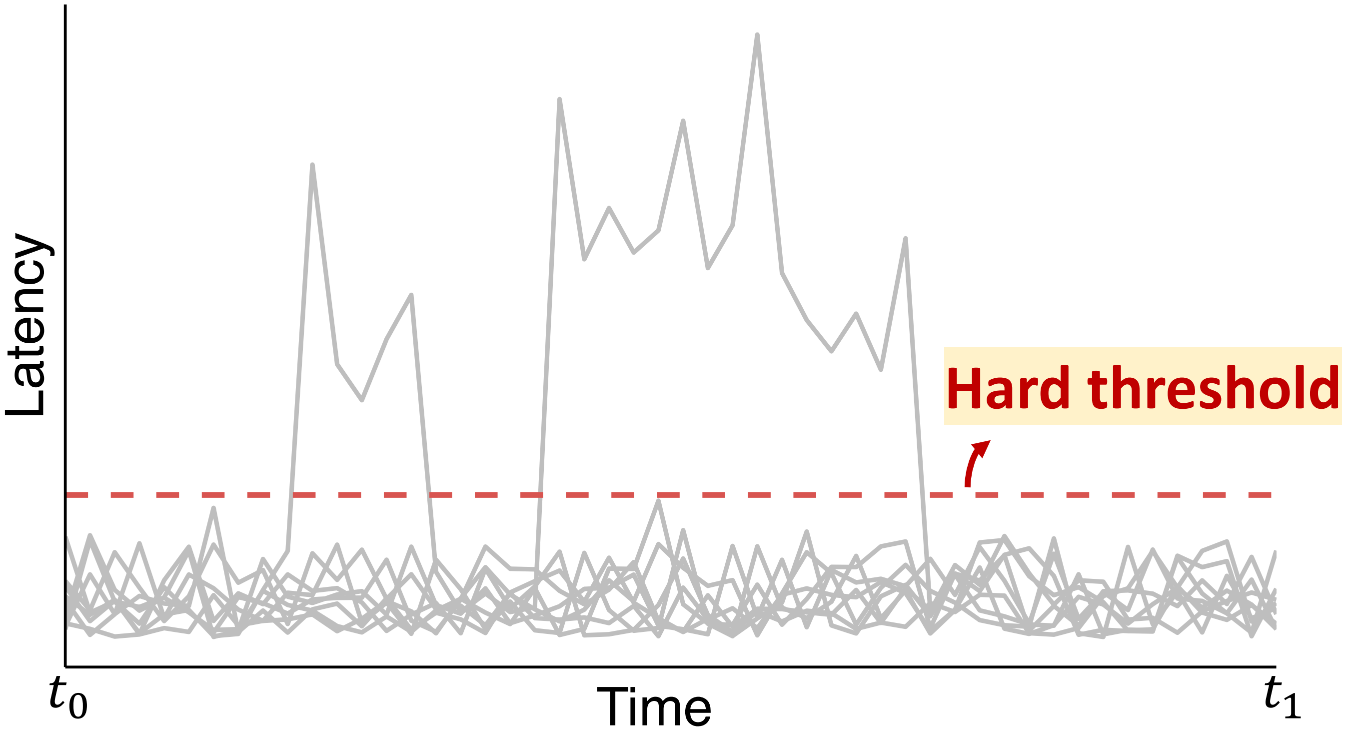
PERSEUS

EVALUATION & CONCLUSION

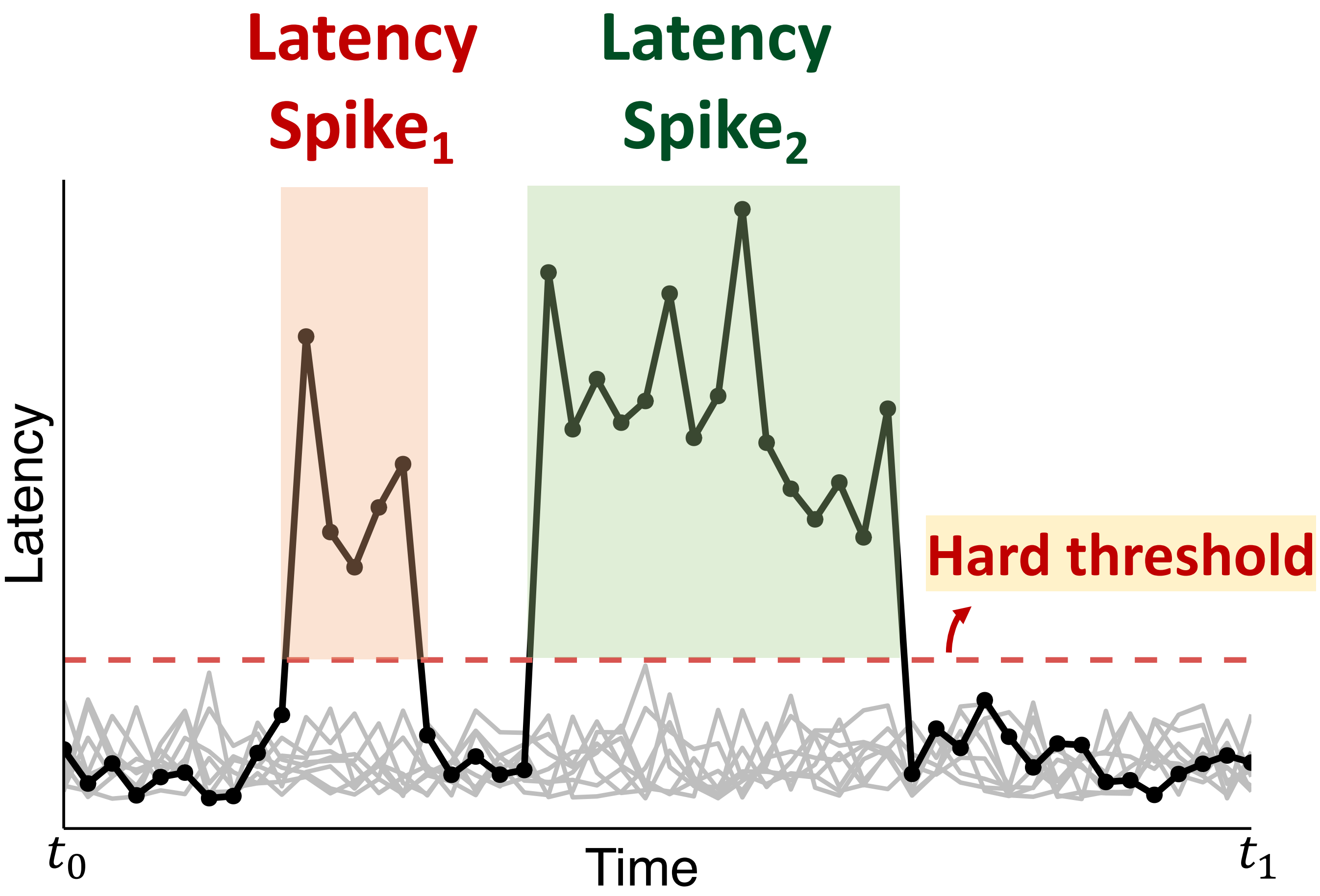
Failed Attempt: Threshold Filtering



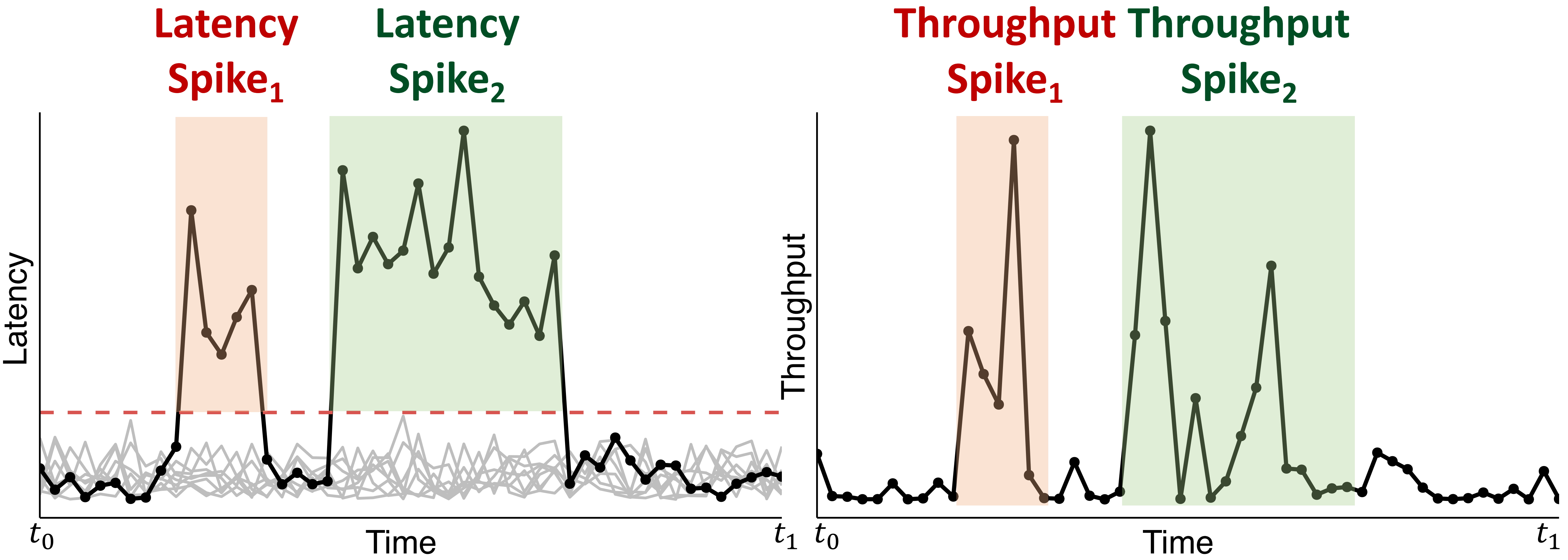
Failed Attempt: Threshold Filtering



Failed Attempt: Threshold Filtering

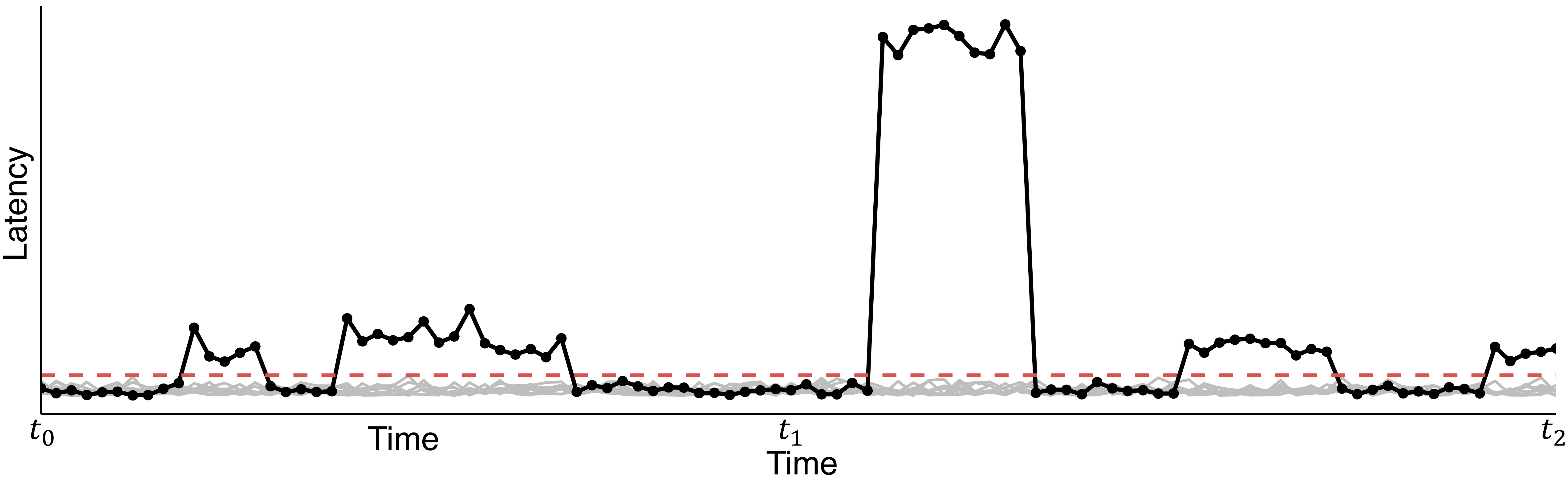


Failed Attempt: Threshold Filtering



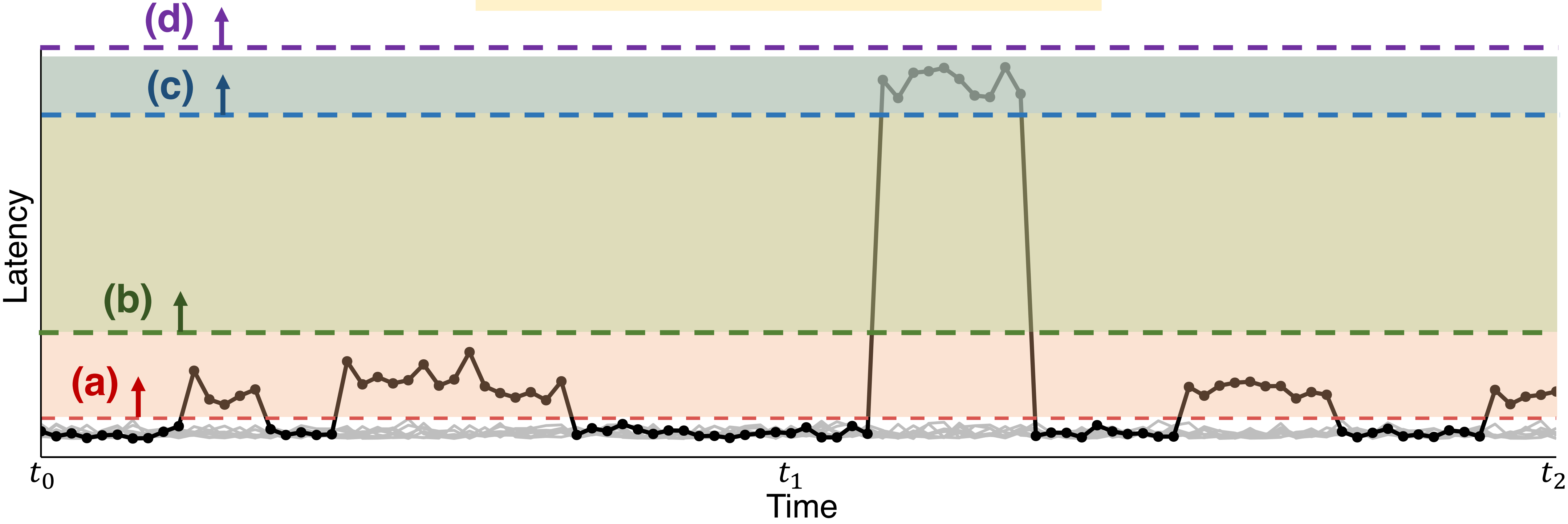
Workload bursts are common causes of latency variations

Failed Attempt: Threshold Filtering



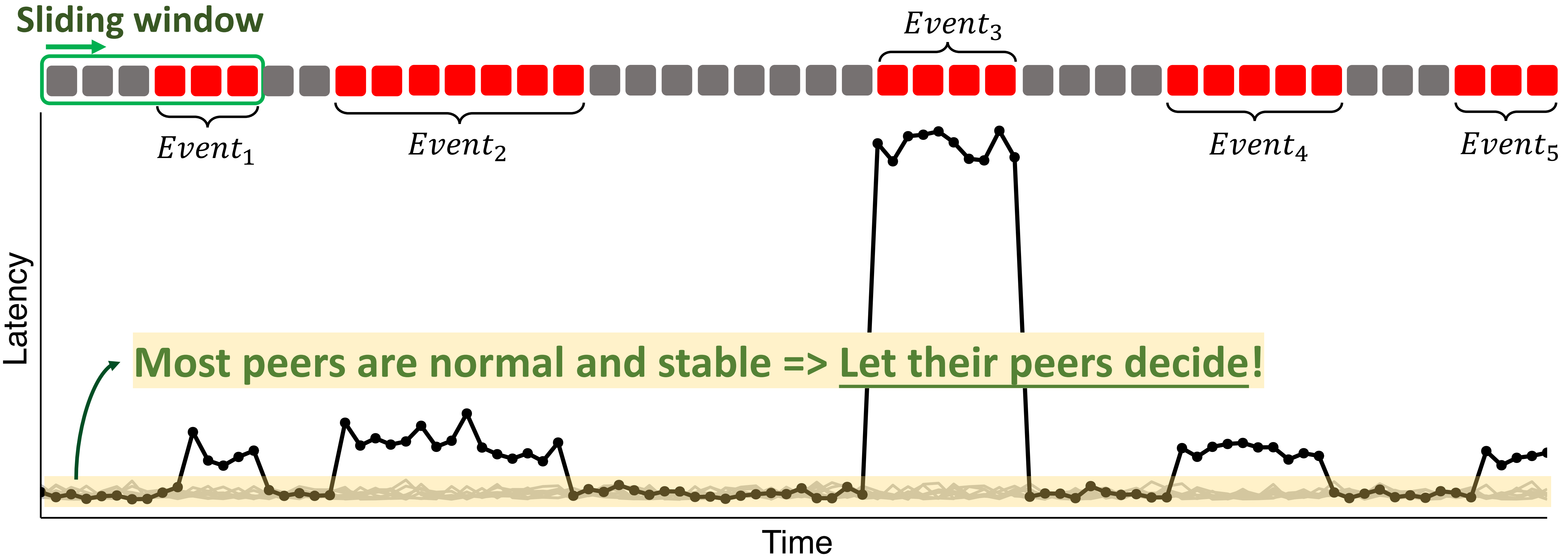
Failed Attempt: Threshold Filtering

Threshold are hard to decide



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

Failed Attempt: Peer Evaluation



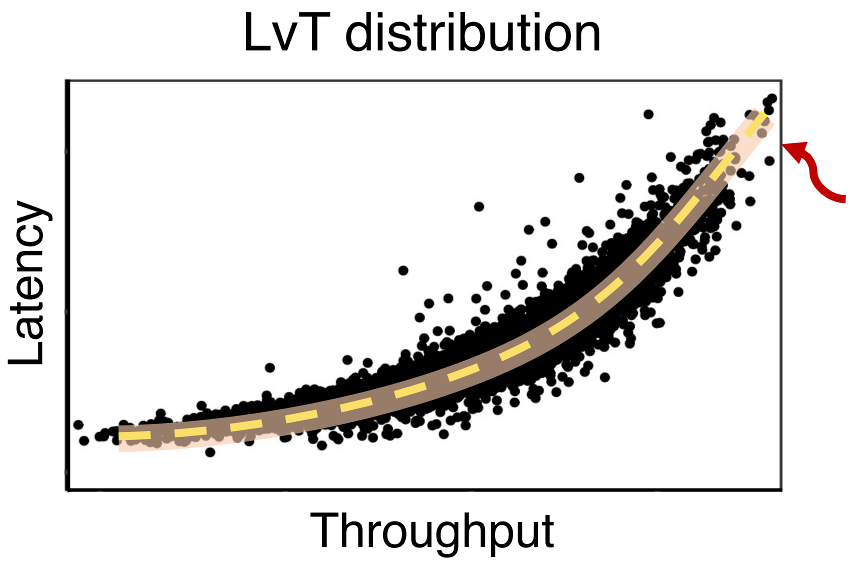
Time-consuming to fine-tune

Design Guidelines

Insight: "Workload pressure can affect latency variations"

- How to model such a positive correlation?

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



Linear regression?

=> Define a *statistically normal* drive

Design Guidelines

Insight: "No golden standards to identify fail-slow "

Guideline: Non-binary output

- Model the likelihood of fail-slow

Outline

~~INTRODUCTION~~

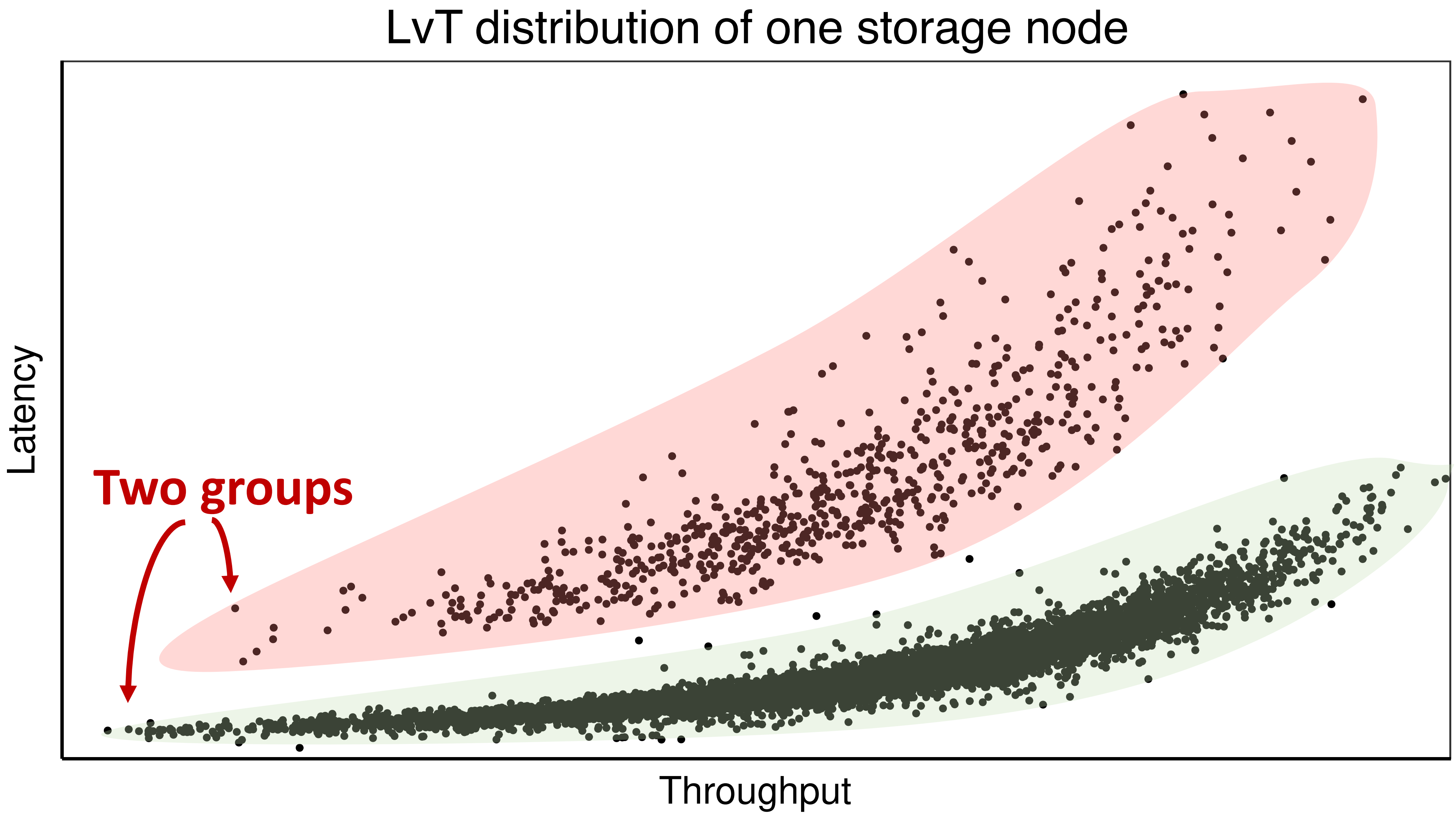
~~DATASET~~

~~FAILED ATTEMPTS~~

PERSEUS

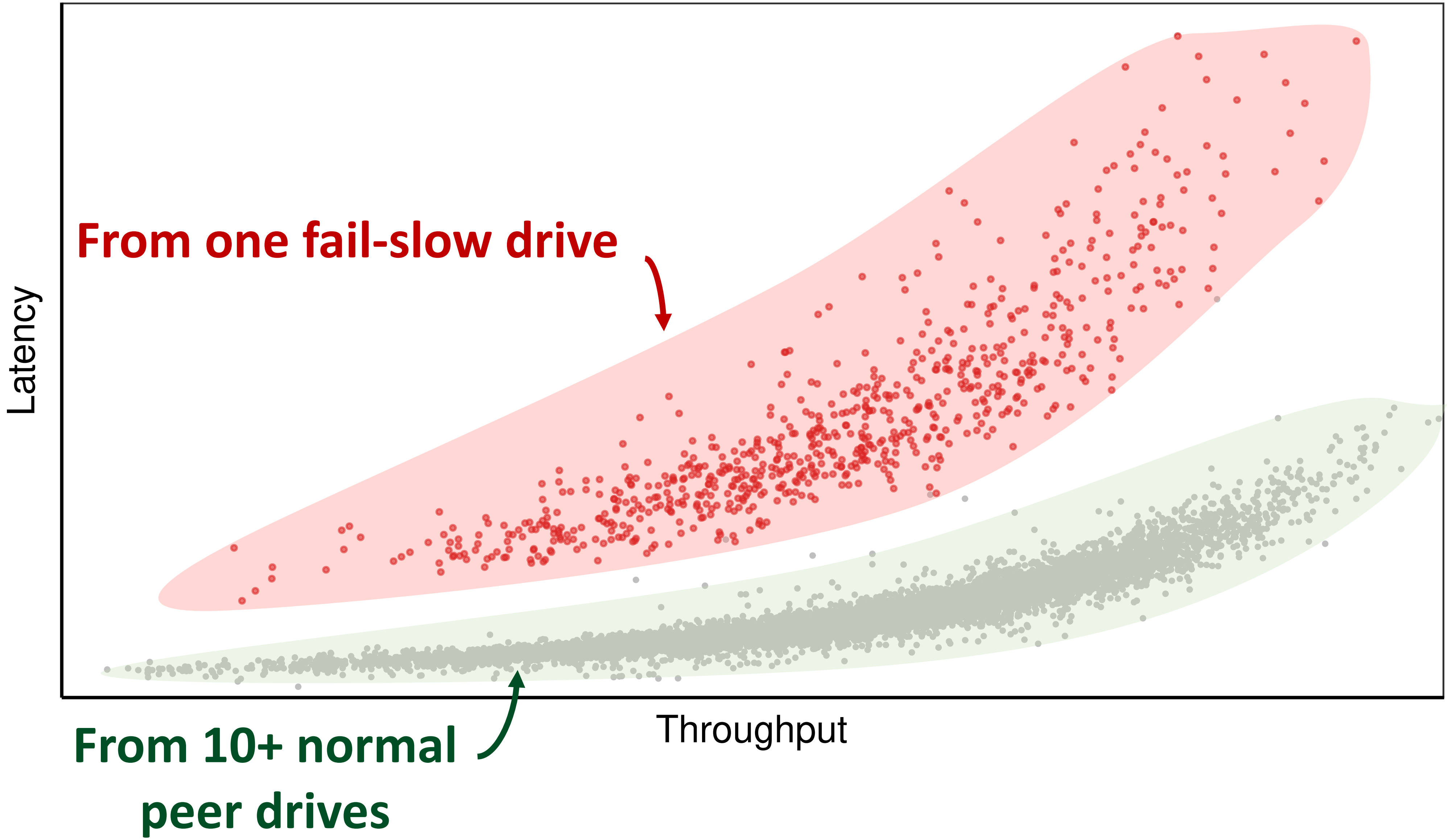
EVALUATION & CONCLUSION

Raw Data



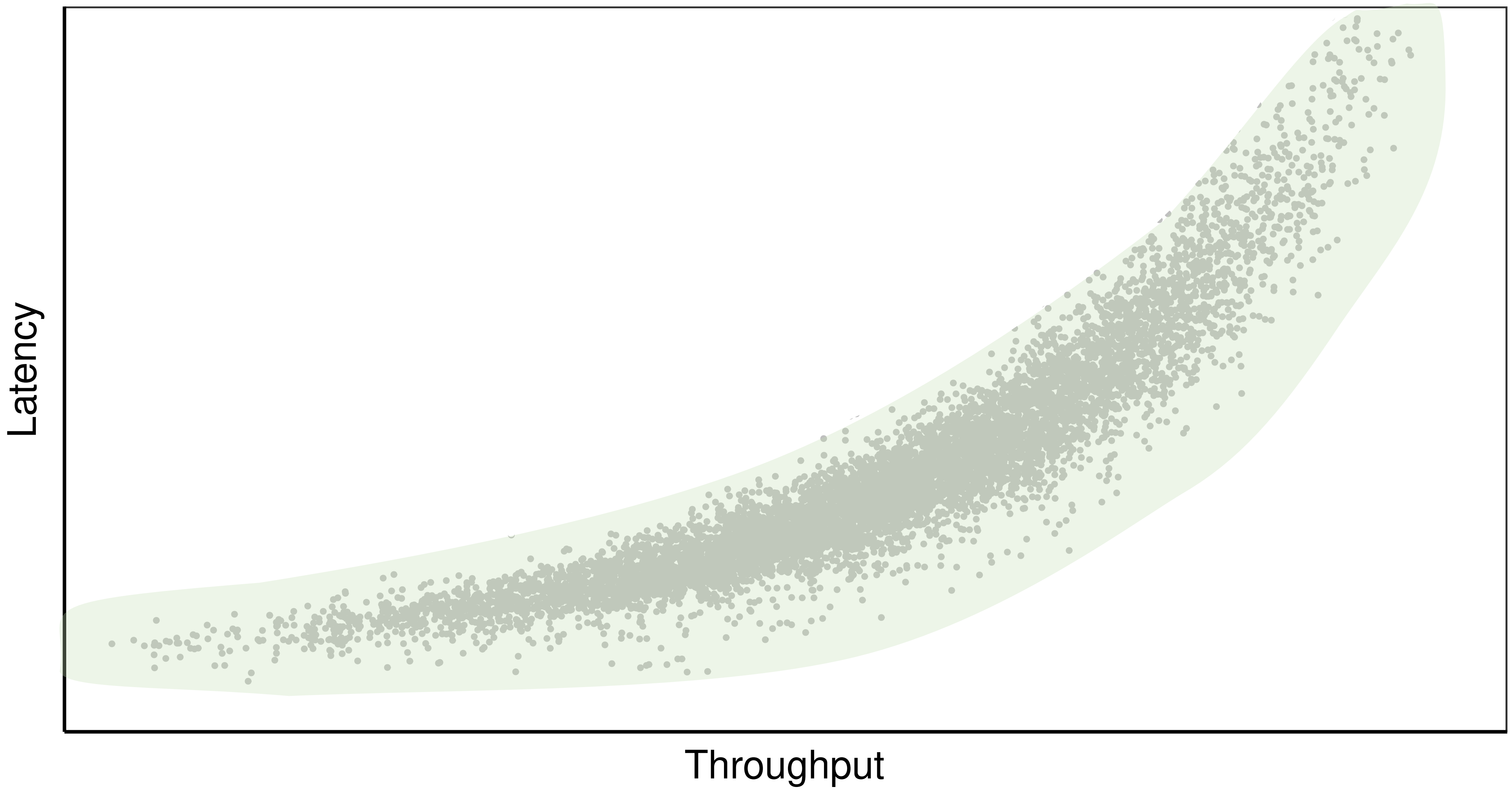
LvT distribution of one storage node
Latency
Throughput
Two groups

Raw Data

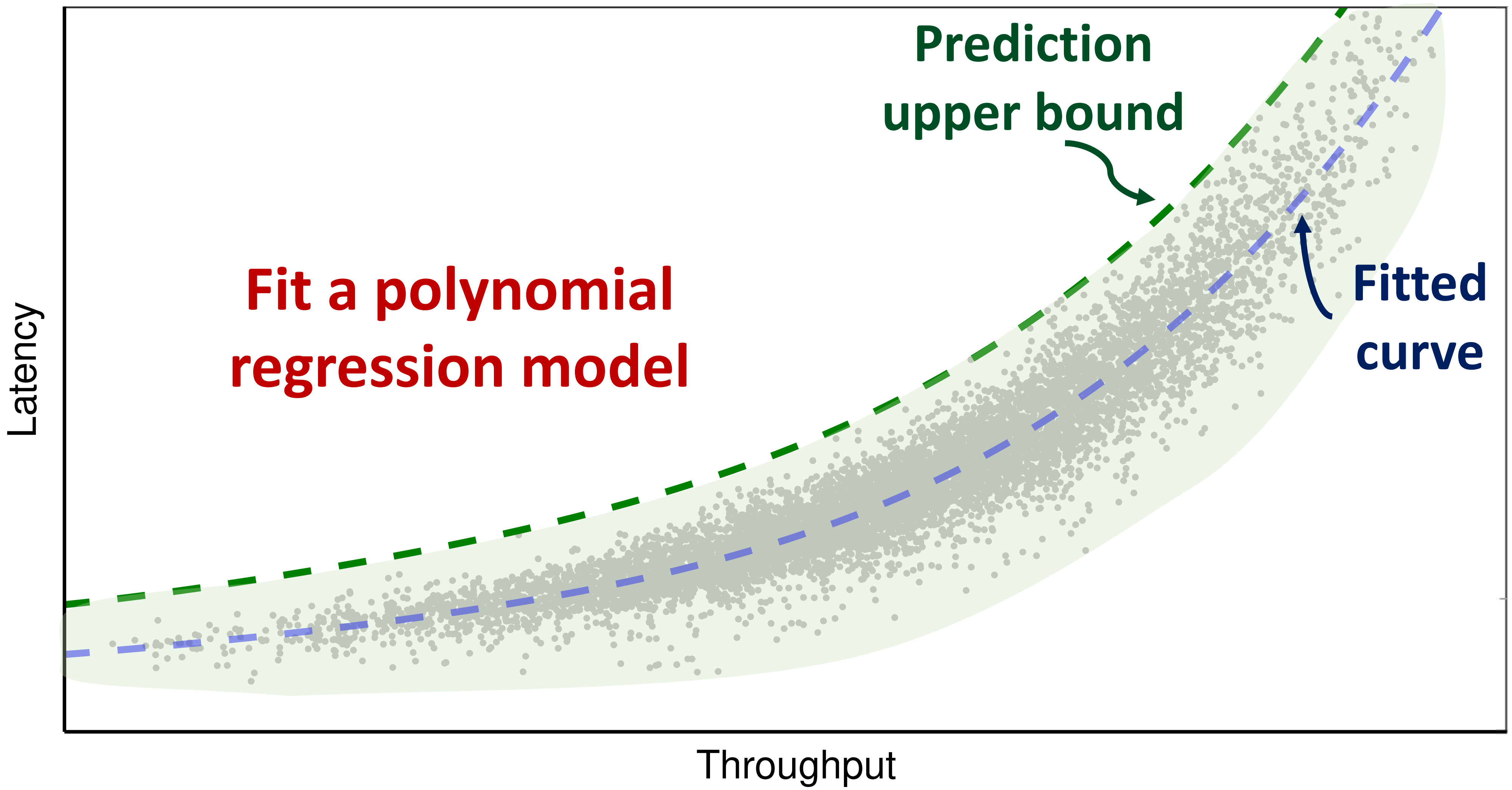


From one fail-slow drive
Latency
Throughput
From 10+ normal peer drives

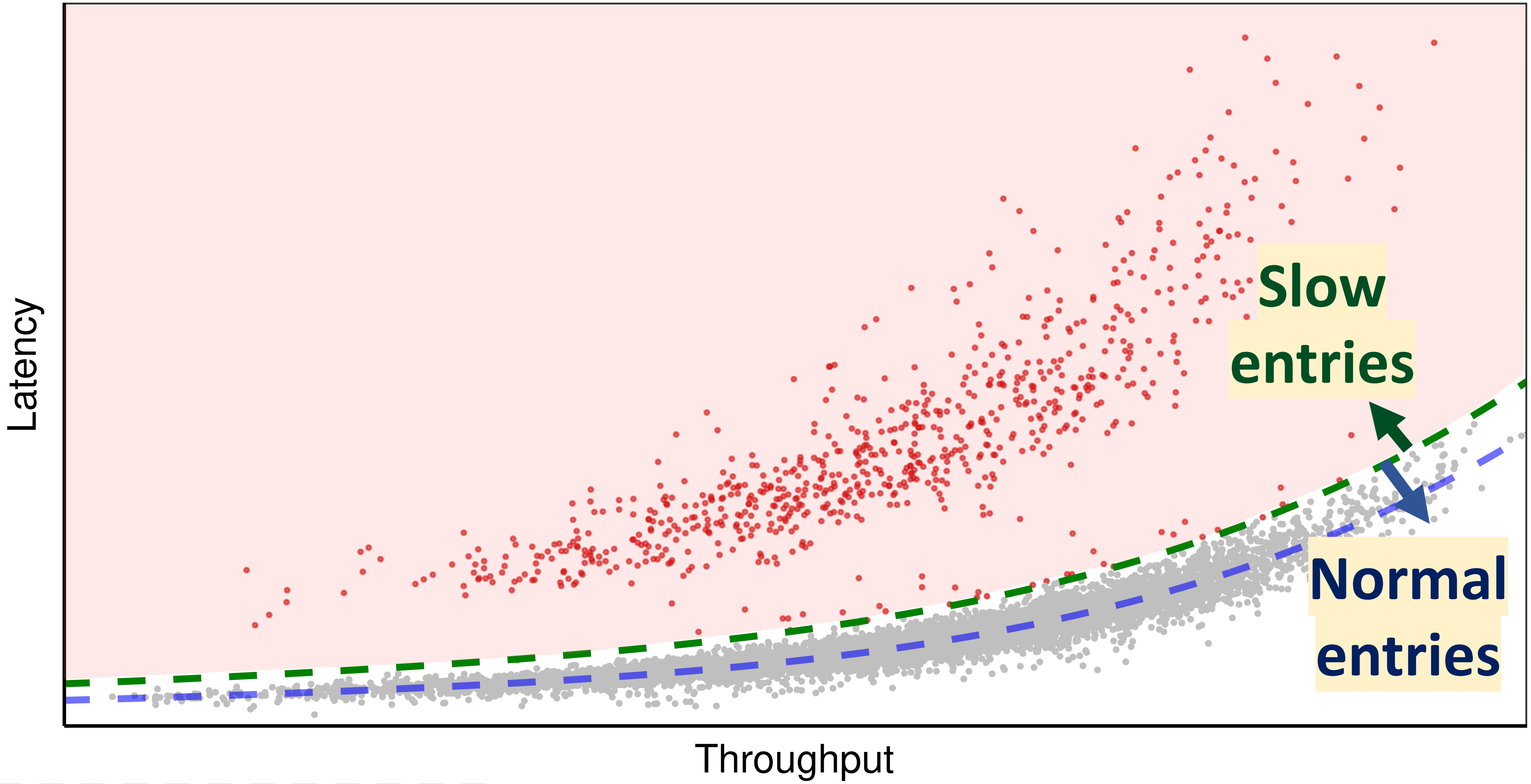
Step 1: Outlier Detection



Step 2: Building Regression Model

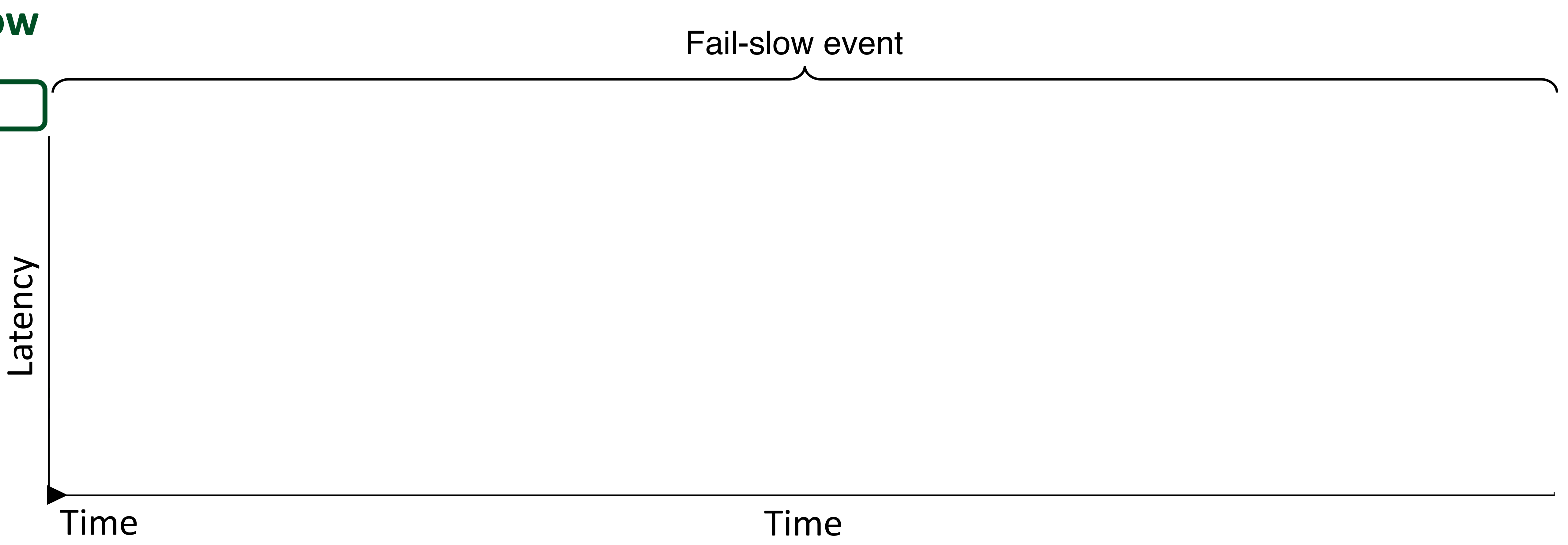


Step 2: Building Regression Model



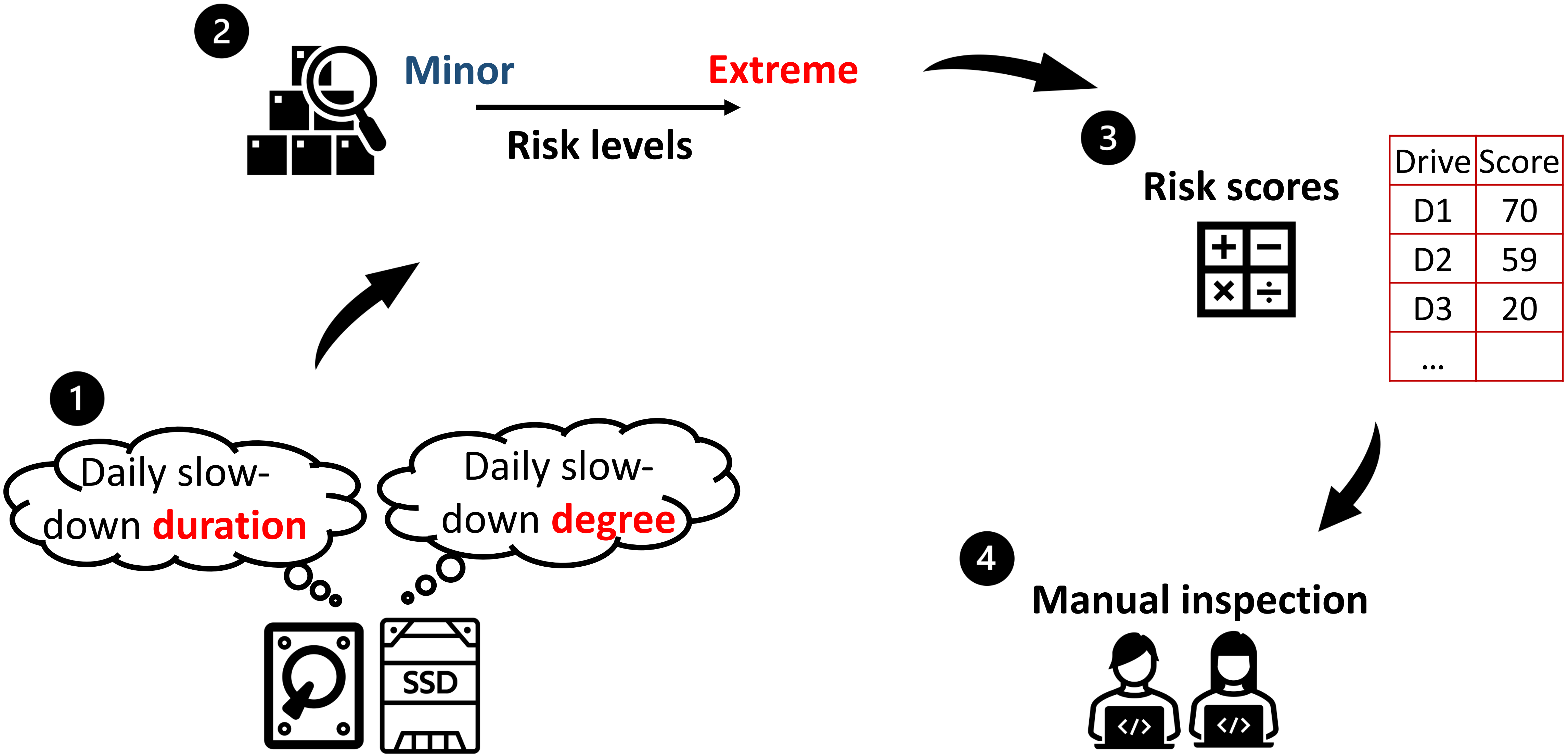
Prediction upper bounds as adaptive latency thresholds without fine-tuning

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~~

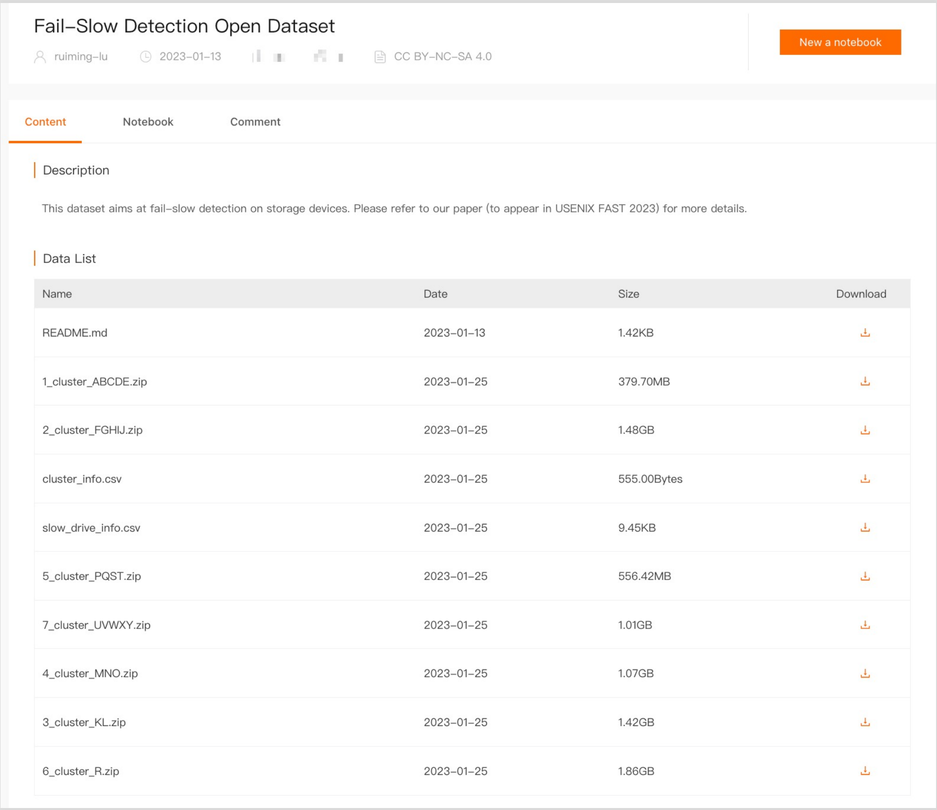
~~FAILED ATTEMPTS~~

~~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



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

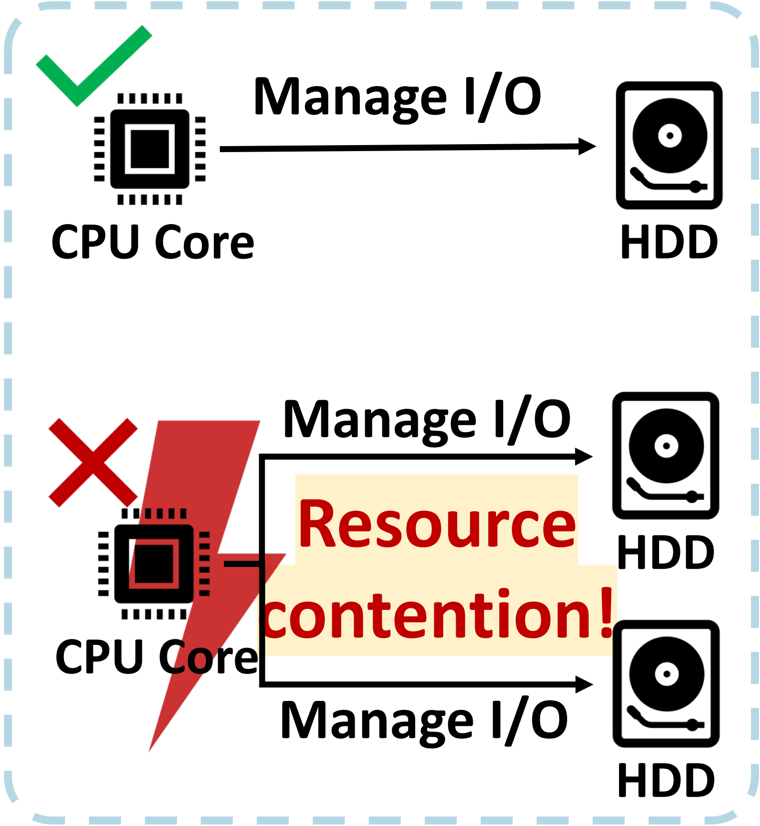
- **Perseus outperforms all previous attempts (§5.4)**
- **Effectiveness of Perseus's Design Choices (§5.5)**
- **Reduce Tail Latency By 31-48% (§5.6)**
- **Root Cause Analysis (§6)**

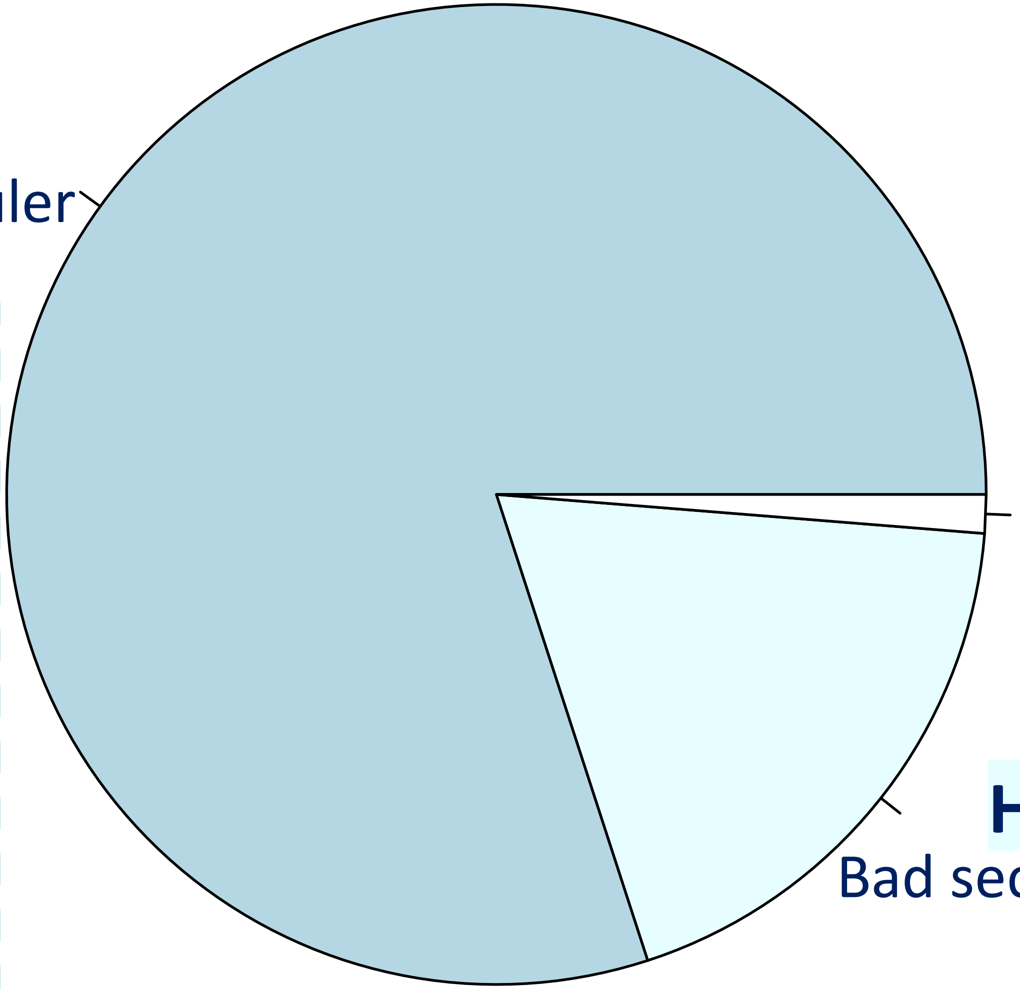
More details in the paper!

Root Cause Distribution

Among 315 Confirmed Fail-Slow Drives:

Software (252):
Ill-implemented scheduler





Environment (4):
Power & temperature

Hardware (59):
Bad sector, bad capacitors, etc.

Summary

Perseus

Detection Framework

Efficient Fail-Slow Detection

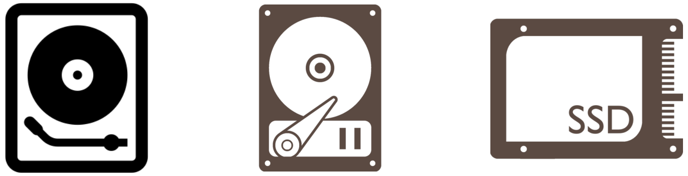
Non-intrusive
(Performance) log-based
No source code altering

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