



HAL
open science

Tracing-Based I/O Diagnosis for Performance Optimization of Software-Defined Storage Systems

Lina Sadi, Islam Chakib Kedadsa, Vincent Lannurien, Stéphane Rubini, Alexandre Skrzyniarz, Jalil Boukhobza

► To cite this version:

Lina Sadi, Islam Chakib Kedadsa, Vincent Lannurien, Stéphane Rubini, Alexandre Skrzyniarz, et al.. Tracing-Based I/O Diagnosis for Performance Optimization of Software-Defined Storage Systems. Per3S Performance and Scalability of Storage Systems, May 2025, Paris, France. <hal-05115969v2>

HAL Id: hal-05115969

<https://hal.science/hal-05115969v2>

Submitted on 23 Jun 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Tracing-Based I/O Diagnosis for Performance Optimization of Software-Defined Storage Systems

Lina Sadi^{*1,2}, Islam Chakib Kedadsa^{*1,2}, Vincent Lannurien¹,
Stephane Rubini⁴, Alexandre Skrzyniarz³, Jalil Boukhobza¹

¹ Lab-STICC, CNRS UMR 6285, ENSTA, Institut Polytechnique de Paris, 29806 Brest, France

² Ecole nationale Supérieure d'Informatique (ESI-Alger)

³ THALES DEFENSE & MISSION SYSTEMS, Brest, France

⁴ Lab-STICC, CNRS UMR 6285, Université de Bretagne Occidentale

* These 2 authors contributed equally to the design of this contribution



Université de Bretagne Occidentale

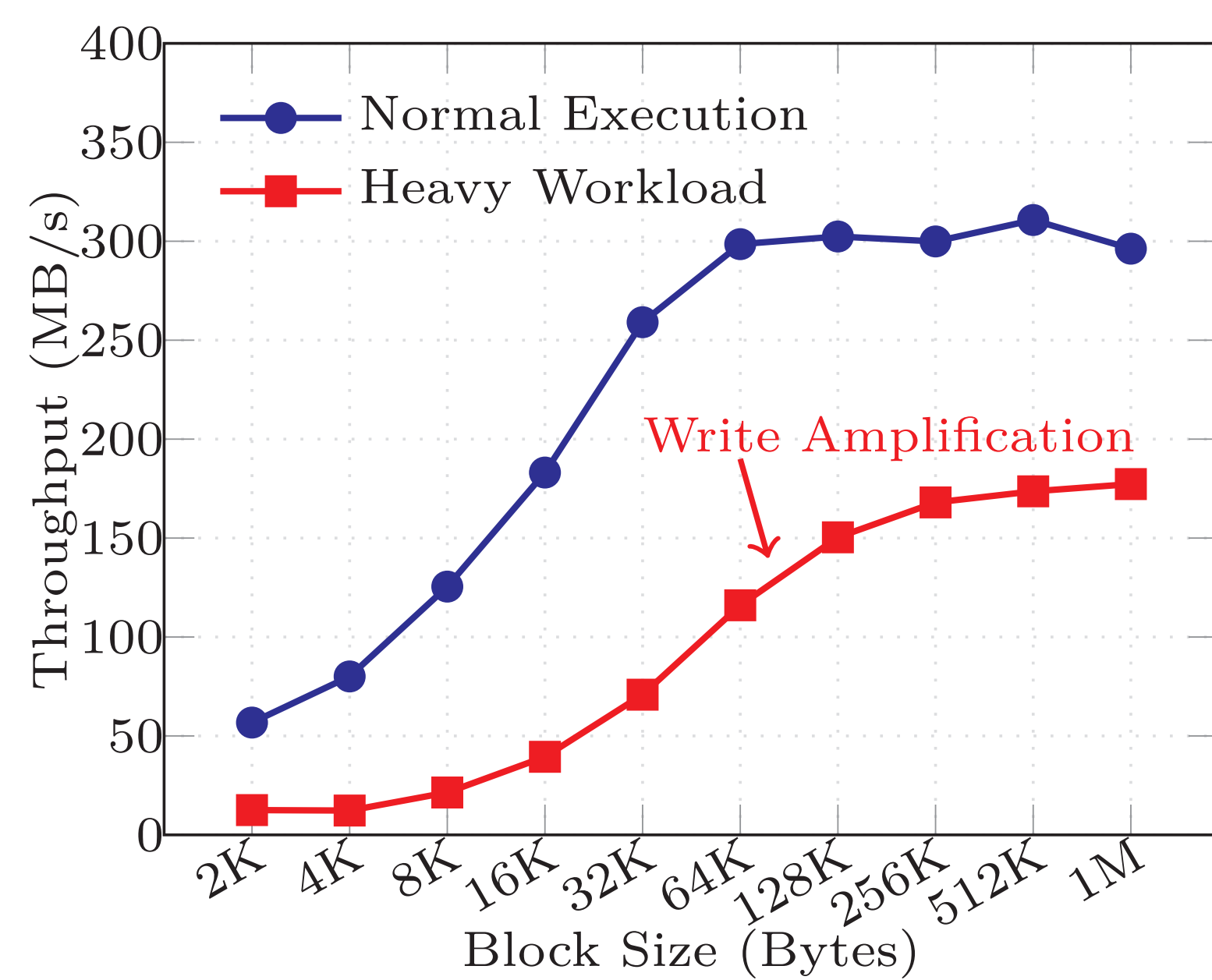
1- Introduction

- SDS brings flexibility and scalability to edge computing [1].
- Under heavy workloads, SDS performance suffers due to complex kernel-level bottlenecks [1].
- Tracing offers fine-grained visibility into I/O operations [2, 3, 4].
- Kernel-level optimizations can mitigate these bottlenecks [5].

Problem statement: How can tracer-based analysis help diagnose and optimize I/O performance in SDS for edge environments?

3- Experimental Motivation

- SSDs are widely adopted in edge SDS environments for their performance and energy efficiency [6].
- However, their internal mechanisms can cause performance degradation [6].
- A major issue is **write amplification**, where the SSD writes more data than initially requested [7].
- To analyze this, we collected I/O traces on a Linux host with various SSD models.
- We tested the SSDs under diverse I/O workloads.
- The results show a significant drop in throughput during heavy workloads.



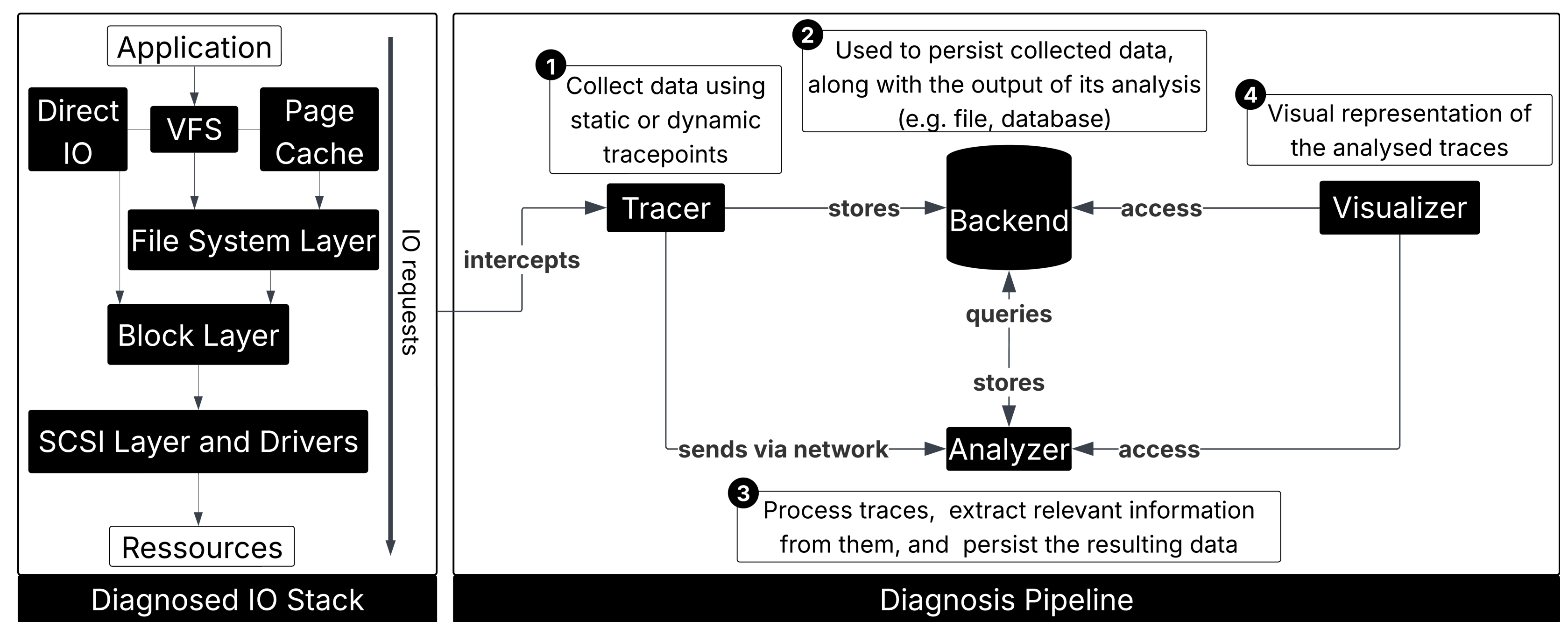
Impact of workload on throughput performance under different block sizes.

References

- [1] Ricardo Macedo, João Paulo, José Pereira, and Alysson Bessani. A survey and classification of software-defined storage systems. *ACM Comput. Surv.*, 53(3), May 2020.
- [2] Tânia de Conceição Araújo Esteves. *Flexible tracing and analysis of applications' I/O behavior*. PhD thesis, Universidade do Minho (Portugal), 2023.
- [3] Houssein Daoud and Michel R Dagenais. Recovering disk storage metrics from low-level trace events. *Software: Practice and experience*, 48(5):1019–1041, 2018.
- [4] Théophile Dubuc. *Détection d'anomalies de latence dans les systèmes distribués avec eBPF*. Theses, Ecole normale supérieure de lyon - ENS LYON, December 2024.
- [5] Ibrahim Umit Akgun, Ali Selman Aydin, Andrew Burford, Michael McNeill, Michael Arkhangelskiy, and Erez Zadok. Improving storage systems using machine learning. *ACM Trans. Storage*, 19(1), January 2023.
- [6] Jean-Emile Dartois, Jalil Boukhobza, Anas Knefati, and Olivier Barais. Investigating machine learning algorithms for modeling ssd i/o performance for container-based virtualization. *IEEE Transactions on Cloud Computing*, 9(3):1103–1116, 2021.
- [7] Xiangqun Zhang, Shuyi Pei, Jongmoo Choi, and Bryan S. Kim. Excessive ssd-internal parallelism considered harmful. In *Proceedings of the 15th ACM Workshop on Hot Topics in Storage and File Systems, HotStorage '23*, page 65–72, New York, NY, USA, 2023. Association for Computing Machinery.
- [8] Iman Kohyarnnejad, Mahsa Shakeri, and Daniel Aloise. System performance anomaly detection using tracing data analysis. In *Proceedings of the 2019 5th International Conference on Computer and Technology Applications, ICCTA '19*, page 169–173, New York, NY, USA, 2019. Association for Computing Machinery.

2- Background

- Tracing-based diagnosis is a technique that collects fine-grained execution traces from the OS or application layers [3].
- Applied to the Linux I/O stack, it enables flexible visibility into kernel-level I/O operations [4].
- The approach typically follows a diagnosis workflow with three main steps [2]:
 - Collection of execution traces
 - Real-time or postmortem analysis to identify performance issues
 - Visualization for interpretation and debugging



Linux I/O stack and tracing-based diagnosis workflow

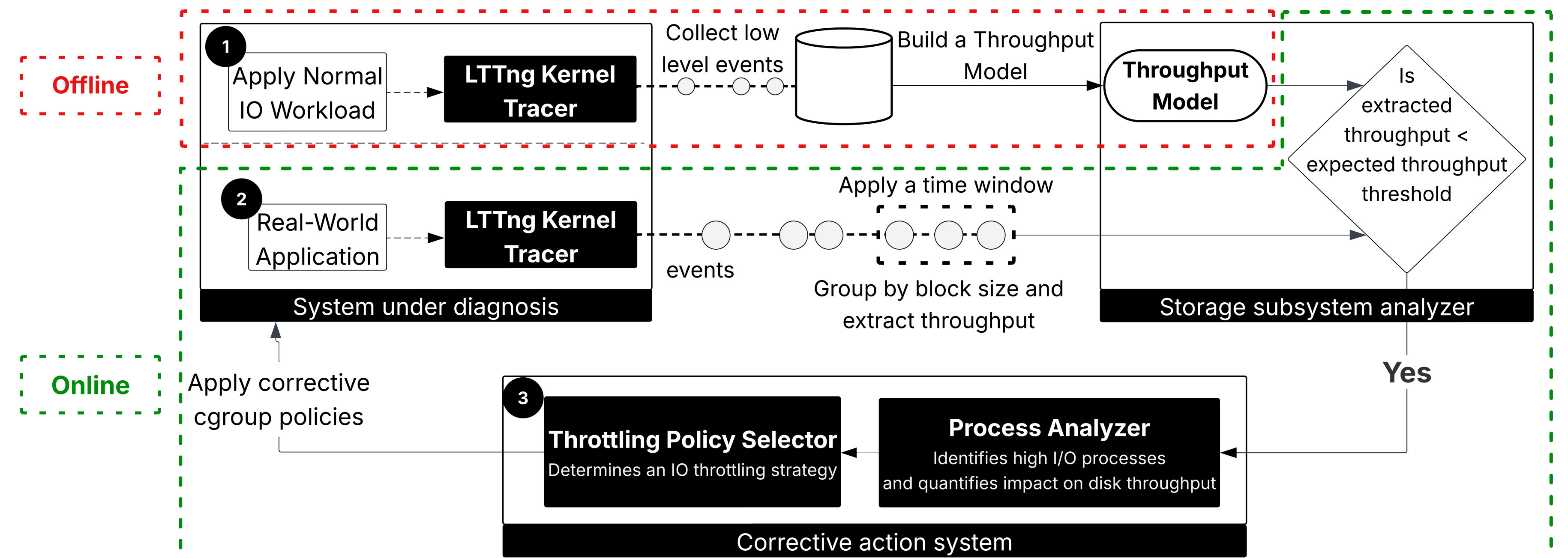
4- Tracing-based diagnosis and correction framework

We propose a tracing framework that automatically detects and mitigates I/O performance degradation using low-level kernel tracing. It operates in three main stages:

1. **Offline modeling:** Builds a baseline throughput model from controlled workloads, using tracepoints focused on SSD.
2. **Online tracing and deviation detection:** During the execution of real workloads, incoming I/O requests are grouped by block size within sliding time windows.

Throughput is measured for each window and compared against the baseline model.

3. **Performance optimization:** When throughput degradation is detected, the system identifies high I/O processes and applies cgroup-based throttling to restore system performance.



Architecture of the tracing-based diagnosis and correction framework

5- Related work

Previous work have proposed various tracing frameworks for diagnosing I/O issues. Some focus on postmortem manual analysis [3], others use ML for automatic detection at higher levels of the I/O stack [8], and some rely on real-time detection of kernel latency spikes [4]. However, these approaches are often limited by their postmortem nature, manual effort, or lack of visibility into low-level SSD behavior. In contrast, our approach extends tracing to the disk layer, enables automatic detection of performance degradation, and introduces an active correction mechanism.

6- Conclusion and Perspectives

We presented a tracing framework for diagnosis and correction of I/O performance issues. Our contributions include:

- Online detection of SSD throughput degradation using low-level kernel tracepoints.
- Adaptive correction via cgroup throttling to mitigate performance issues.

Our future directions:

- Integration with Ceph to evaluate the framework under realistic SDS deployments.
- Extension to block layer tracing, focusing on queue contention and request starvation.