Do we still need I/O schedulers for low-latency disks?

Presented by Code Busters

Do We Still Need I/O Schedulers?

- HDDs relied on I/O schedulers for performance
- I/O schedulers optimized disk access patterns and reduced seek time
- Today's SSDs and Ull devices offer:
  - Sub-10 µs latency
  - Native parallelism
- Scheduler overhead (5-10 µs) is now significant
- In modern systems, software can be the bottleneck
- I/O schedulers may hurt performance on fast storage

Energy Efficiency Matters in Modern I/O

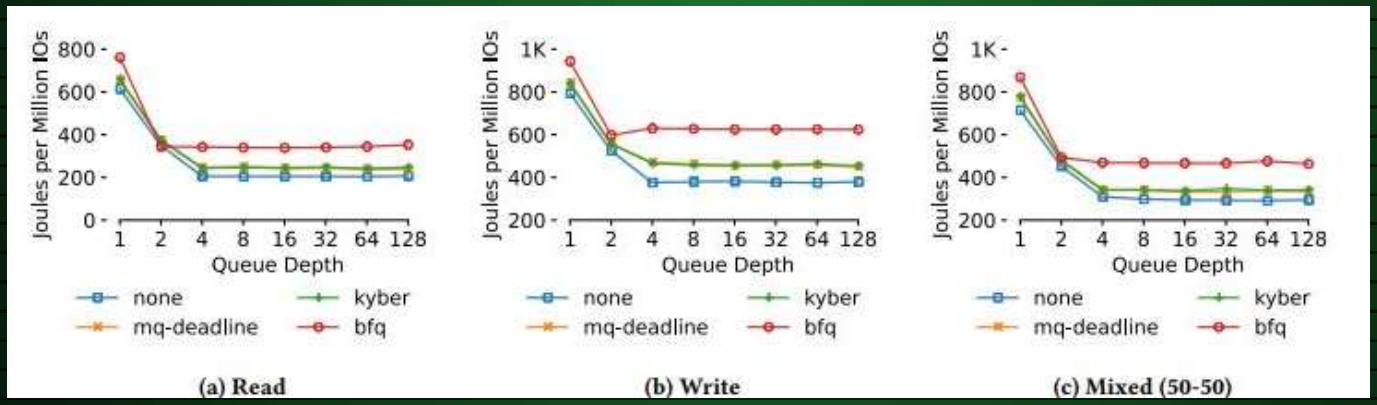
- High-performance apps need both speed and efficiency
- Scheduler delays -> more time -> more power used
- Old assumptions no longer hold:
  - Ordering/merging not always beneficial
  - ULL SSDs access data randomly and in parallel
- Calls for OS-level redesign of the I/O stack
- Goal: Maximize both performance and energy efficiency

Question:

- Are IO schedulers still necessary in the era of Ultra-low latency storage?

Evaluating the Energy Cost of I/O Scheduling

The problem being solved is to determine if schedulers overhead enhances or deteriorates the efficiency for ultra-low latency SSDs. The author benchmarks this by analyzing the joules consumed with none, Kyber, MQ-Deadline and BFQ scheduler enabled.



Here is the different types of schedulers, we will discuss more in the slides to come.

Comparing I/O Scheduler Strategies

Each benchmark experiments with different types of scheduler algorithm to ensure the problem is being solved methodically. **MQ-Deadline** focuses on fairness and deadlines, it's primarily utilized for HDDs & SATA SSDs it aims to prevent resource starvation. **Kyber** focuses on latency control, it's primarily utilized for NVMe SSDs and it is very efficient at handling dynamic latency via token-based system. **BFQ** focuses on providing a sharing resources concisely to ensure fair queuing and prevents a single app from utilizing all the given resources.

Rethinking I/O Scheduling

Conditional Bypass/Disable Strategy

- **General approach**: Disable or bypass I/O schedulers for most scenarios since SSDs don't require traditional scheduling
- **Key insight**: Scheduling overhead now exceeds benefits for typical workloads

Selective Use Cases for Scheduling

The authors recognize that I/O scheduling is still necessary for special cases:

- **Quality of Service (QoS)** - ensuring I/O requests aren't missed
- **Multi-tenancy** - managing multiple processes effectively
- **Write-heavy workloads** - where coordination is still beneficial

Hybrid Scheduler Architecture

The authors propose new hybrid schedulers that perform **minimal work at the OS level**, striking a balance between no scheduling and traditional heavy scheduling.

Two specific Device-Aware Smart I/O Schedulers

Device-Direct Fair Queueing

- **Focus**: Reducing the cost of I/O scheduling
- **Benefit**: Uses the device's own capabilities rather than duplicating work in software

Multi-Queue Fair Queueing

- **Focus**: Eliminating performance bottlenecks from coordination overhead
- **Benefit**: Allows parallel processing without coordination penalties

Testing their ideas

The set-up:

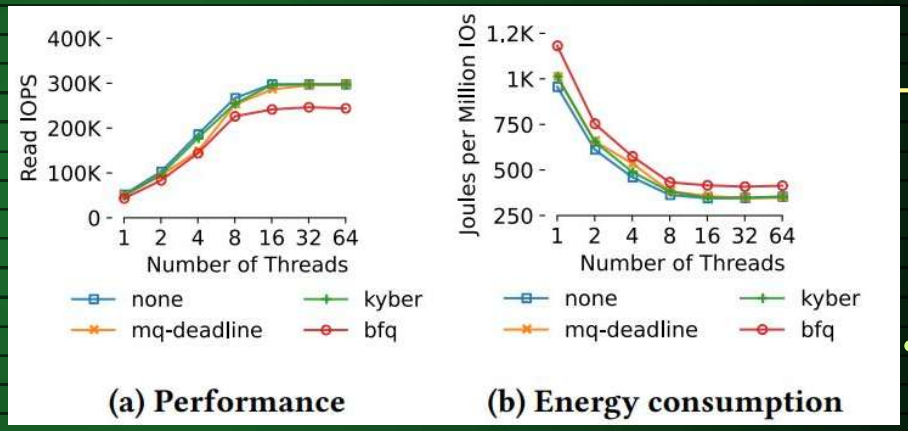
- Dell PowerEdge R230
  - Intel Xeon E3-1230 CPU
  - 64GB RAM
  - Intel Optane P4801X NVMe SSD
- Ubuntu 22.04
  - Linux kernel 5.18
- HOBO UX120-018 Data Logger to measure energy consumption
- Synced power meter and system clock using a time server
- db_bench to collect data on intensity of workload
- Saturated capacity of device

The method:

- Microbenchmarks:
  - Single-tenant experiments
  - Multi-tenant experiments
- Macrobenchmarks:
  - Simulated real-world database application I/O to evaluate practical system performance
- Tested schedulers: none, mq-deadline, kyber, and bfq
- Submitted requests for random reads, random writes, and 50/50 read and writes
- Each workload tested five times for an average

Findings from the tests

- Results from the test show that I/O schedulers hinder performance for modern-day technology
- Looking at the graphs from the results, the greatest performance came from no I/O scheduling
- No I/O scheduler also reduces amount of energy consumed
- I/O scheduler with ultra-low latency SSD results in more latency, impairs performance, and increases energy consumption



Macrobenchmark results of performance and energy consumption for the four different I/O schedulers

Resources

Whitaker, C., Sundar, S., Harris, B., & Altiparmak, N. (2023). Do we still need I/O schedulers for low-latency disks? *Proceedings of the 15th ACM Workshop on Hot Topics in Storage and File Systems (HotStorage '23)*, 44–50. https://dl.acm.org/doi/pdf/10.1145/3599691.3603400