Enabling New Use Cases with High-Capacity QLC SSDs

A view on the Ecosystem

Presenter: Javier González | Principal Engineer @ Samsung Electronics

the Future of Memory and Storage

SAMSUNG

2012

Let’s Focus on the Ecosystem

Larger IUs

+

Data Placement

Or, why QLC and high-capacity storage devices rely on the Open-Ecosystem to succeed

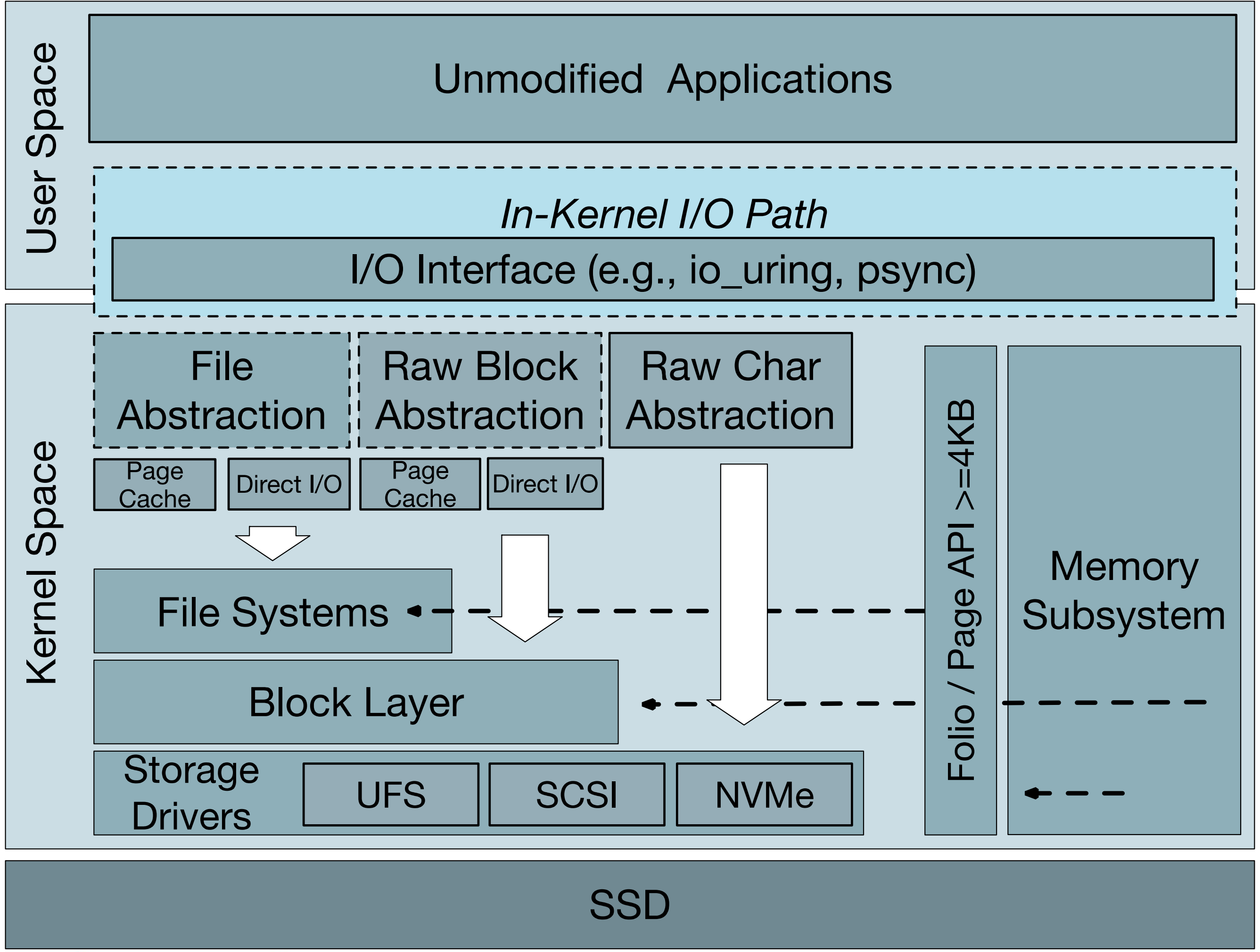
High-Capacity Ecosystem today

- Large Indirection Unit (IU) than LBA format already used in the industry today
  - Writes larger than the IU implies RMW
  - First used to support 4KB, 512B LBA format
    - RMW implied for 512 byte LBA, industry standard today
  - 16KB IUs are norm for 64 TB today, yet max LBA format is 4KB
    - Not the first case where IU size > LBA format
  - OCP 2.0 standardizes IU on NPWG
  - Writes aligned to the IU consume less power
- Demand for large capacities (64,128, 256 TB) begets larger Indirection Units (IU)
  - IO verification -> many workloads do not use 512b or 4k IOs already
    - OCP Smart Health Log page 0xC0 SMART-21 unaligned IU writes
    - OS based runtime traces: Linux trace-cmd / eBPF blkalgn
  - Scalability is an issue (e.g., TCO, blast radius, addressability)
  - Larger IUs fit more naturally >QLC NAND (and simplifies high-capacity HDDs)
  - Larger IUs help reduce WAF
    - GC can work at a larger granularity

IUs >= 16k ecosystem challenges

- LBA format staying at 4k
- Current storage stacks assumptions for 512b / 4KB
- **Operating system free to split I/Os at 4KB**
- Guaranteeing I/Os at IU size is a memory memory management problem
- Vendor independent challenge and affects everyone
- Today’s endurance DWPD is based on JEDEC 218
    - JEDEC 218 implies pure random writes don’t reflect real world workloads
    - DWPD adjustments are in order which consider IU alignment to the IU

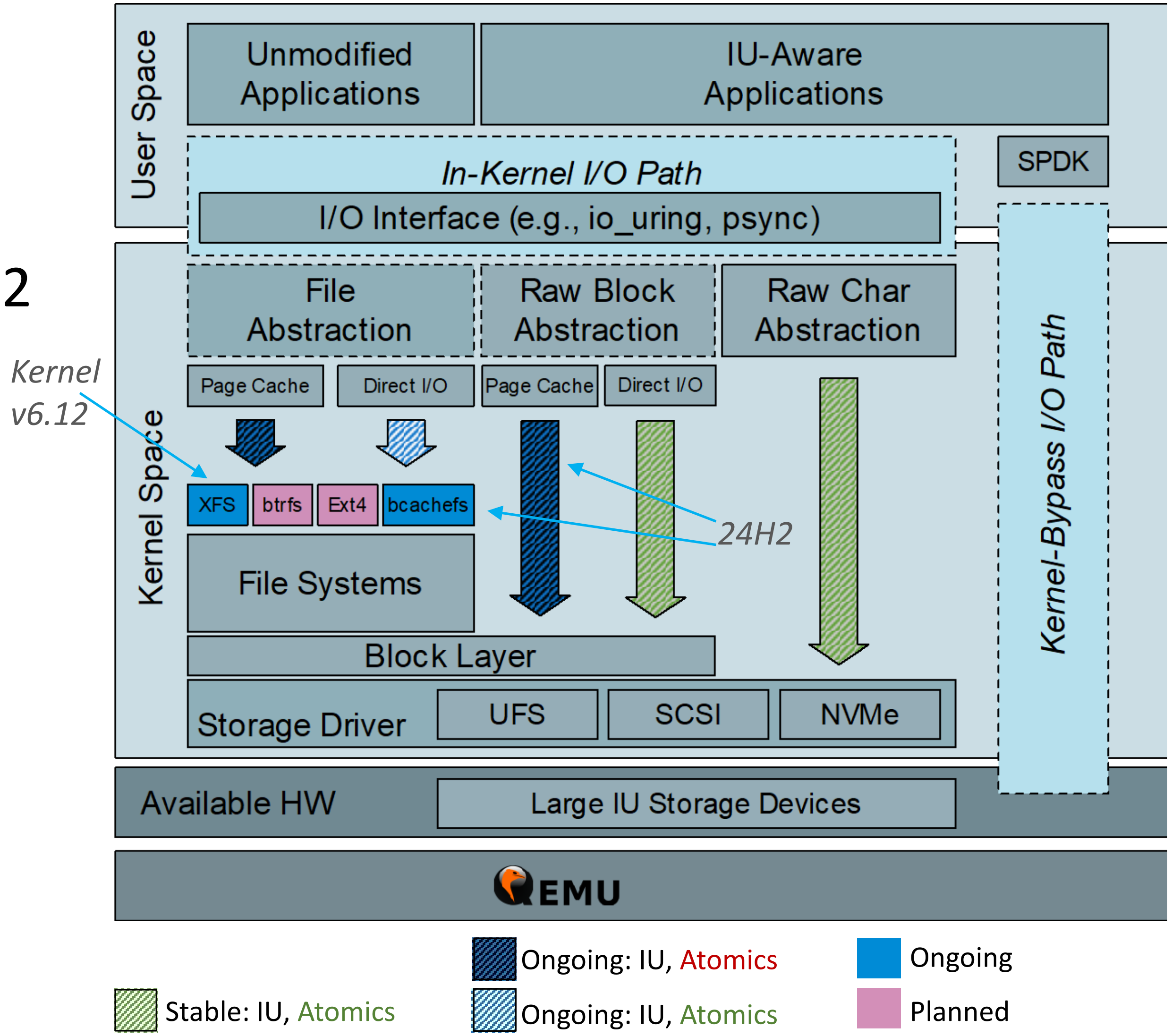
Support in Linux



- It is all about memory
  - Guarantee memory allocations to fit the IU
  - We do this through *folio* abstraction
  - Guaranteed contiguous memory facilitates guaranteeing user I/O size submission
- Nothing changes at NVMe level
  - LBA Format is unchanged
  - Use Namespace Preferred Write Granularity (NPWG) & Atomic Write Unit Power Fail (AWUPF)
  - This work enables larger LBA Formats too in the future!
- Aligns with work on providing atomic guarantees
  - Offload atomicity from SW to HW
  - DBs can provide ACID guarantees without WAL

LBS Ecosystem

- I/O Passthru & Block Direct I/O support already up to 256KB IU
- XFS support planned for kernel 6.12
- Block w/ Page Cache planned for end of 2024
- BcacheFS planned for end of 2024
- Btrfs and Ext4 in pipeline with community
- Extending IU to 2MB planned for 2025
- Open development [1]



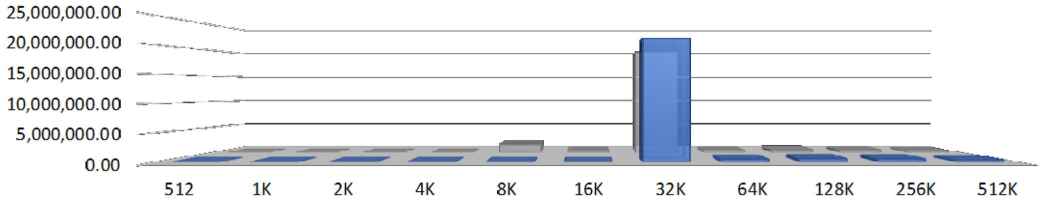
[1] https://kernelnewbies.org/KernelProjects/large-block-size

MySQL with 16KB Block Sizes

12 hour MySQL sysbench

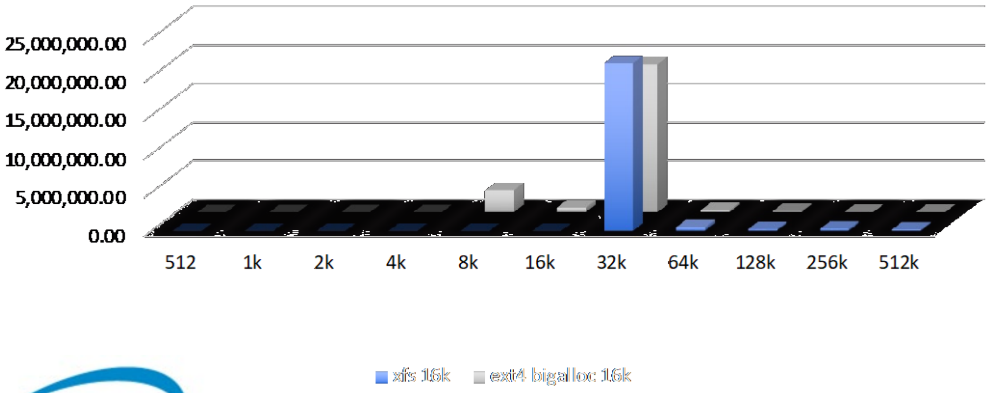
IO count by IO size

IO count by IO size

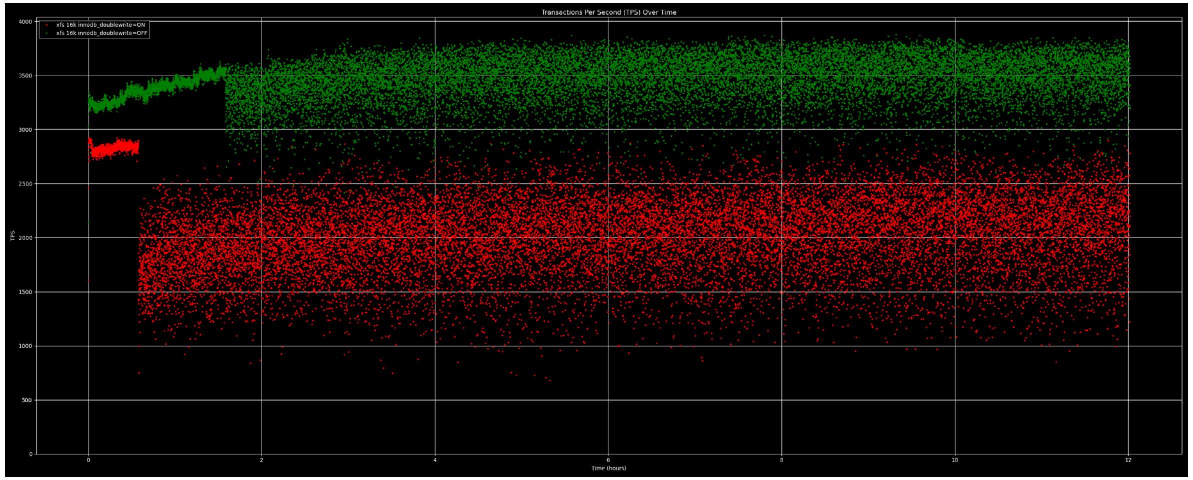


IO alignment □xfs 16k □ext4 bigalloc 16k

IO alignment

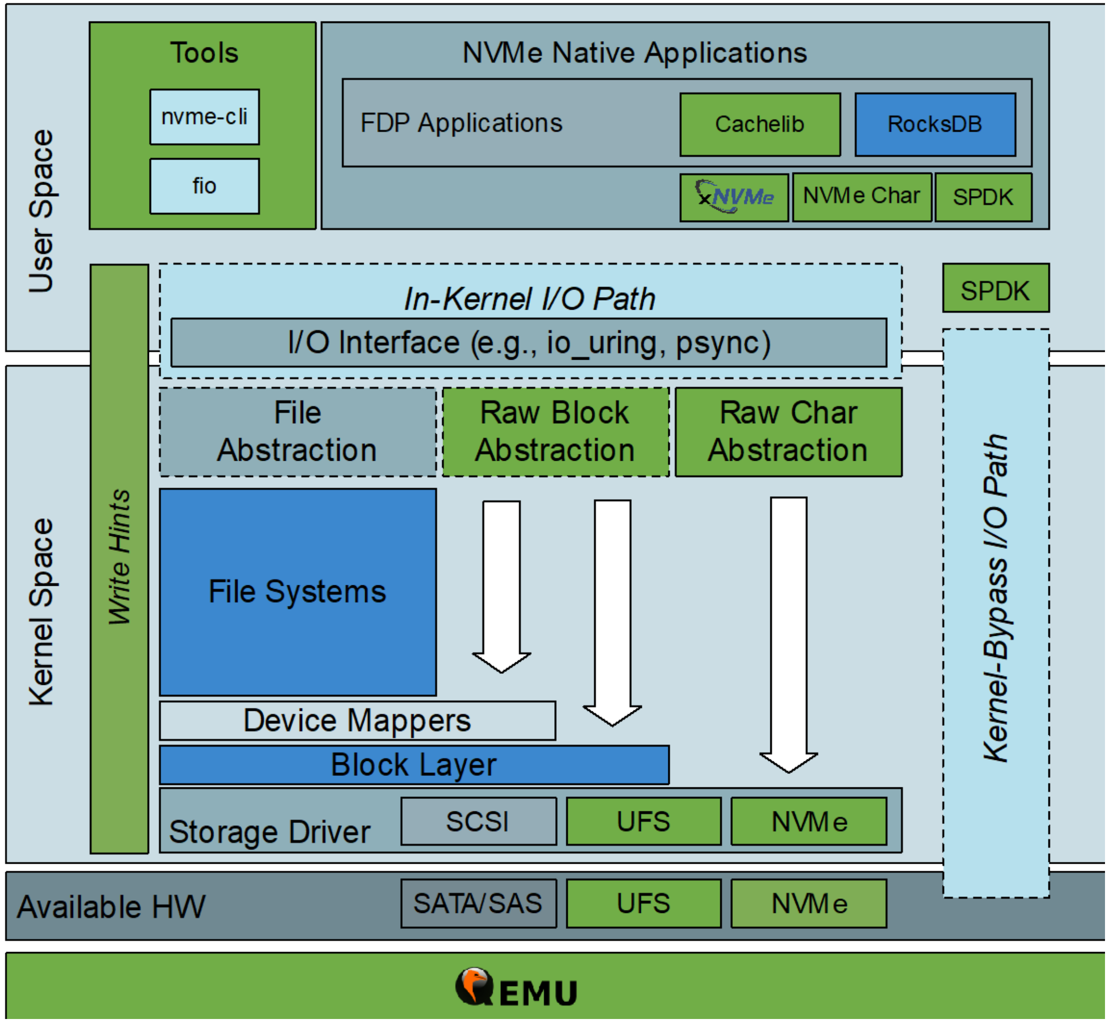


3x – 5x TPS variability gains with 16KB AWUPF turning off innodb_doublewrite



Data Placement in QLC

- QLC can benefit of host-aware data placement technologies
  - FDP is the industry's choice for improving host-device data placement in NAND
  - Use-cases from past technologies (OCSSD, ZNS) converging in the industry
  - All mayor vendors providing FDP solutions to hyperscale and enterprise customers
- FDP Ecosystem is simple and ready for core applications
  - Kernel patches under review for block & file
  - Applications with stream support can use without any changes

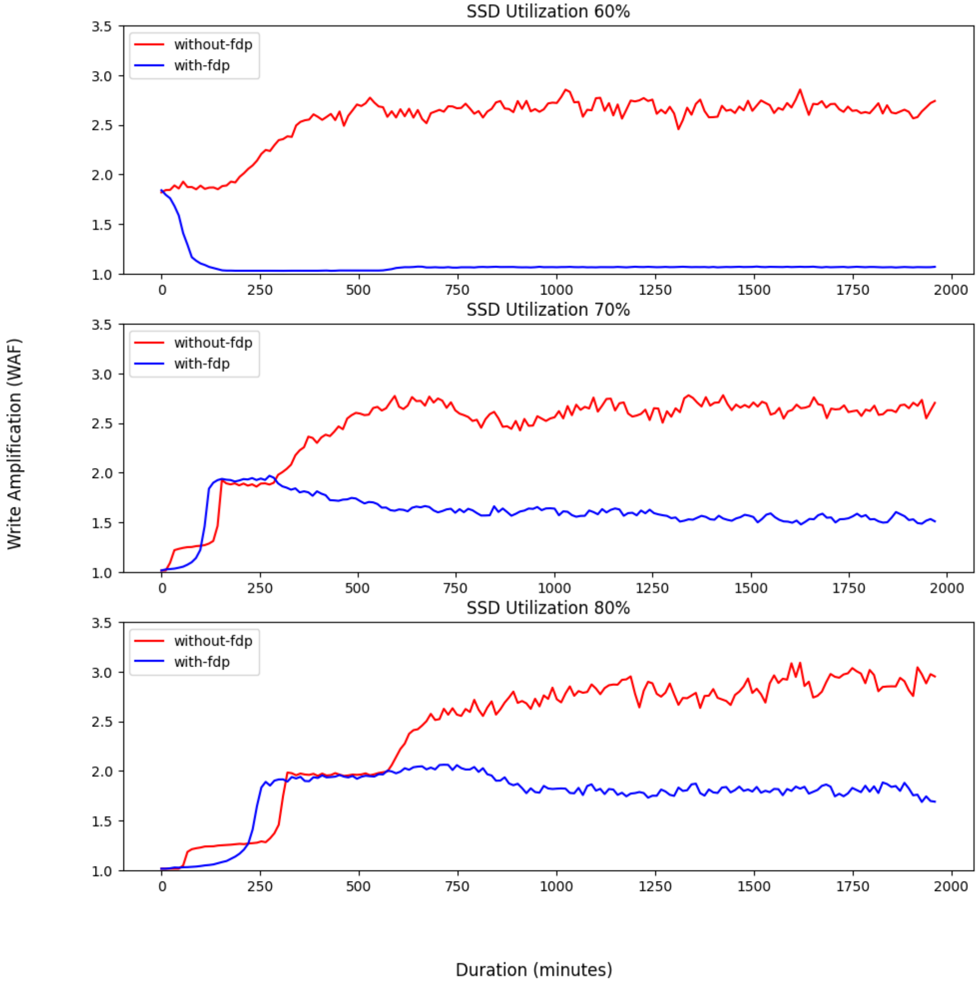


🟩 Stable 🟦 Ongoing 🟪 Planned

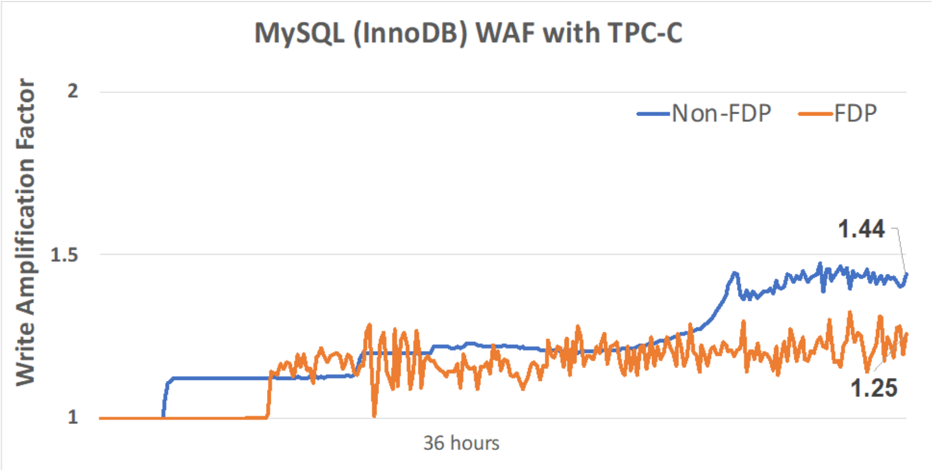
FDP: Real Deployment Data

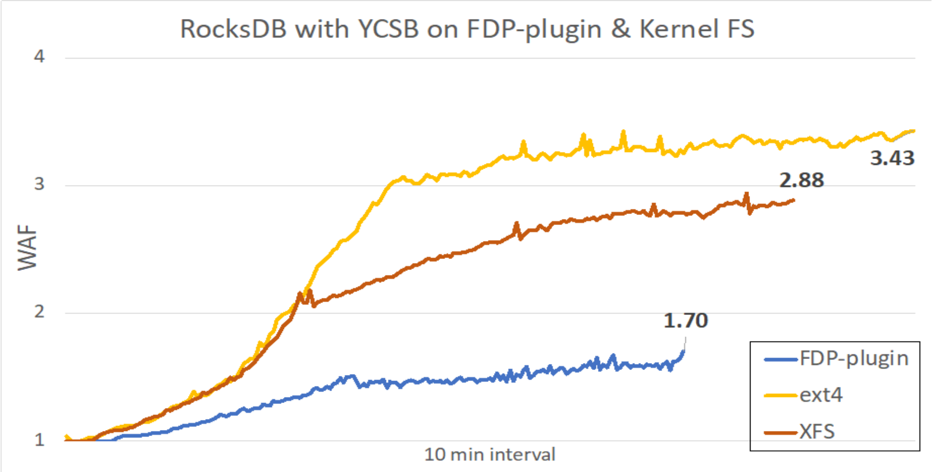
- Significant WAF improvement without major host changes

Cachelib



MySQL & RocksDB





Current Status & Next Steps

- **Large IU**: Working upstream in Linux to provide robust LBS support
  - NVMe and Block Layer as mostly done
  - Current work is focused on file-system enablement. Some patches posted
  - Working with the community on defining IU boundaries imposed by the Linux stack
  - Next step is quantifying benefits on generic workloads without application changes

- **Data Placement**: Finalize FDP ecosystem development. Maintain & Improve
  - Enable generic workloads without application changes. Separate metadata at OS level
    - This is relevant for legacy workloads running on generic SKUs with FDP enabled. Simplify procurement
  - Maintain simple OS APIs and help with a few more high-impact applications
  - Work with industry in expanding FDP to new use-cases in standards and open-ecosystem

Enabling High-Capacity SSDs through QLC

A view on the Ecosystem

Presenter: Javier González | Principal Engineer @ Samsung Electronics

QUESTIONS?

Talk to us in the hallway or at the Samsung booth!