SAMSUNG

SPDK State of the Project

Jim Harris

Principal Engineer - Open Source, R&D - DTC
Samsung Semiconductor

Overview

- **SPDK codebase continues to be active**
  - >1500 patches since September 2023 release
- **SPDK contributors continue to be varied**
  - Almost 100 contributors from 25+ companies
- **Key areas of development (recent and ongoing)**
  - Accelerators and memory domains (Ben!)
  - Power savings
  - NVMe over Fabrics
  - Logical volumes and RAID
  - Tracing
  - NUMA

SAMSUNG

Power Savings

- **Two avenues towards power savings**
  - Schedulers and Governors
  - Interrupt Mode

- **Schedulers and Governors**
  - More sophisticated scheduling algorithms
  - Scheduling period preemption
  - Better amortization of TCP syscall overhead across multiple spdk_threads

- **Interrupt Mode**
  - Plumb SPDK socket layer for interrupts
  - Add interrupt support to NVMe target (TCP, RDMA)
  - Add PCIe device interrupt support to SPDK NVMe driver
  - Add interrupt support to bdev/nvme module

- **Work in progress**
  - Parts will start landing in v24.09 release

NVMe over Fabrics

- **Authentication support (v24.05)**
  - Target and host driver support
  - Pluggable keyring library
- **Namespace masking (v24.05)**
  - Limit namespaces in controller based on hostnqn
- **Discovery referrals (v24.01)**
- **Custom reservation handlers (v24.01)**
- **Better NVMe feature passthrough**
  - Enable NVMe-oF hosts to observe NVMe-specific parameters
  - FDP (v24.05)
  - optperf, atomic (target v24.09)

Logical Volumes and RAID

- **Logical Volumes**
  - Extend lvolstore at runtime
  - Better unmap support
  - Shallow copies
- **RAID**
  - Progressing towards a REAL RAID stack
  - RAID-1
  - RAID-5F
  - On-disk metadata
  - Rebuild

Tracing

- **Tracepoint owners**
  - Map event to specific bdev, TCP connection, NVMe queue, etc.
- **Enable tracing for user-created pthreads**
- **New tracepoints and related features**
  - Current queue depth for existing nvme, bdev, nvmf IO tracepoints
  - Sock (TCP) layer tracepoints
  - Map events to spdk_thread name

NUMA

- SPDK has ignored NUMA to date
- Increased focus with chiplet designs
- NUMA optimizations in progress
  - Allocate PCIe CQs from socket-local memory
  - Map NVMe host controller (PCIe, TCP, RDMA) to socket ID
  - Map bdevs to socket ID
  - Map NVMe target controller (TCP, RDMA) to socket ID
  - Allocate benchmarking (fio, SPDK tools) buffers based on socket ID
  - Plumb iobuf to support per-socket buffer pools
  - Allocate target application buffers based on heuristics (I/O type, nvmf socket ID, bdev socket ID)

Thank You