Project #3: SSD Performance Profiling

Modern SSDs (especially NVMe) can deliver massive IOPS and GB/s when requests are issued concurrently. Like memory, storage exhibits a classic throughput–latency trade-off governed by queuing theory: increasing queue depth improves utilization and throughput up to a saturation **knee**, after which **latency** rises sharply with little additional throughput.

⚠️ **Data-loss warning**
Never benchmark against a partition that contains valuable data. Create a dedicated, empty partition or use a raw device reserved for tests. If you must test via a file, ensure direct I/O (to bypass the page cache) and use a large file on an otherwise empty filesystem. You are responsible for preventing data loss.

Learning Goals (What your experiments must reveal)

- **Zero-queue (QD=1) latency** for reads and writes under both **random 4 KiB** and **sequential 128 KiB** patterns.
- **Maximum small-IOPS** (4 KiB) and **maximum large-block throughput** (≥128 KiB, reported in MB/s or GB/s).
- The **throughput–latency trade-off** as **queue depth / parallelism** increases, and identification of the **knee**.
- **Block-size and access-pattern effects** (sequential vs. random) on latency, IOPS, and bandwidth.
- **Read/Write mix effects** (read-only, write-only, 70/30, 50/50) at fixed conditions.
- **Working-set / LBA-range effects** and the difference between **burst** and **steady-state** behavior (e.g., SLC cache exhaustion, thermal throttling).
- **Tail latency** characterization (p95/p99) and its relationship to queueing and device state.

Tools You'll Use

- **FIO (Flexible I/O Tester)** for generating controlled storage workloads (reads/writes, block sizes, queue depth, parallel jobs, random/sequential access, read/write mixes, time-based runs, percentiles).
- (Optional) **Basic observability**: OS disk stats and SSD SMART/health tools to note temperature, throttling indicators, and total writes.

Experimental Knobs (orthogonal axes)

1. **Block size**: 4 KiB, 16 KiB, 32 KiB, 64 KiB, 128 KiB, 256 KiB (and optionally 512 KiB/1 MiB for sequential).
2. **Access pattern**: sequential vs. random; 4 KiB alignment for random; contiguous LBA for sequential.
3. **Read/Write ratio**: 100%R, 100%W, 70/30, 50/50 (hold other knobs fixed when sweeping mix).
4. **Queue depth & parallelism**: vary **iodepth** and/or **numjobs/threads** (e.g., QD 1→2→4→8→16→32→64→128→256 when device supports it).
5. **Data pattern**: incompressible vs. compressible payload (impacts some consumer SSDs); keep consistent and document.

Required Experiments & Plots

1. **Zero-queue baselines**
   Measure QD=1 **latency** for: (a) 4 KiB random reads & writes, (b) 128 KiB sequential reads & writes. Report average and percentiles (p95/p99). Provide a clearly labeled table.
2. **Block-size sweep (pattern fixed)**
   Hold pattern fixed (do both **random** and **sequential** in separate runs). Sweep 4 KiB→256 KiB (and optionally 512 KiB/1 MiB). From the **same runs**, produce **IOPS/MB/s** and **average latency** (two panels or dual axis). Mark where reporting naturally shifts from IOPS (≤64 KiB) to MB/s (≥128 KiB).
3. **Read/Write mix sweep (knobs fixed)**
   With block size and pattern fixed (e.g., 4 KiB random), run **100%R, 100%W, 70/30, 50/50**. Plot throughput and latency from the same runs and discuss differences.

4. **Queue-depth/parallelism sweep (trade-off curve)**
   Using 4 KiB random (and optionally 128 KiB sequential), increase queue depth/numjobs across ≥5 points. Produce a single **throughput vs. latency** trade-off curve. **Identify the knee** and relate it to Little's Law (Throughput ≈ Concurrency / Latency).
5. **Tail-latency characterization**
   For at least one workload (e.g., 4 KiB random read at mid-QD and near-knee QD), report **p50/p95/p99/p99.9** latency and discuss queueing impact and SLA implications.

Reporting & Deliverables (commit everything to GitHub)

- **Scripts/configs**, raw results, and plotting code (re-runnable).
- **Setup/methodology:** SSD model, interface (PCIe gen/lanes or SATA), capacity used, system CPU/OS, filesystem vs raw device, direct I/O usage.
- **Clearly labeled plots/tables** with units and error bars (≥3 runs when feasible); indicate which knobs are fixed vs varied.
- **Analysis** grounded in queuing theory and device behavior.
- **Limitations/anomalies** with hypotheses (e.g., background GC, thermal events, host-side cache interference).

Grading Rubric (Total 170 pts)

1. **Zero-queue baselines (30)**
   - (10) Correct isolation of QD=1 latency (page cache bypassed; alignment documented).
   - (10) Accurate average & percentile latencies for 4 KiB random and 128 KiB sequential (R/W).
   - (10) Clear tabular presentation with units.
2. **Block-size & pattern sweep (40)**
   - (15) Complete coverage of required sizes for both patterns using one coherent matrix.
   - (10) Plots show IOPS/MB/s **and** latency from the same runs (proper axes/legends).
   - (15) Insightful discussion of prefetching/queue coalescing, controller limits, and cross-over from IOPS- to bandwidth-dominated regimes.
3. **Read/Write mix sweep (30)**
   - (10) Correct implementation of four mixes under fixed conditions.
   - (10) Coherent explanation of differences (write buffering, WA, flushes).
   - (10) Proper labeling/units and comparison on matched axes.
4. **Queue-depth/parallelism sweep (40)**
   - (15) ≥5 QD points; single **throughput–latency** curve with error bars.
   - (10) Clear identification and justification of the **knee** via Little's Law.
   - (10) % of interface or vendor-spec peak; discussion of diminishing returns.
   - (5) Tail-latency note at or near the knee (p95/p99) and implications.
5. **Synthesis & reporting quality (30)**
   - (15) Full reproducibility (configs, versions, environment) and data hygiene.
   - (15) Thoughtful anomalies/limitations section with plausible hypotheses.

Tips for Successful Execution

- **Use direct I/O and 4 KiB alignment** to avoid the page cache and partial-block penalties. Document filesystem vs raw device.
- **Precondition for steady-state** when testing random writes (e.g., fill target range with random data first). Note any TRIM/discard.
- **Control device temperature** (consistent airflow); record temperature and watch for throttling.
- **Randomize trial order** and **repeat** runs to capture variance; report mean ± stdev and latency percentiles.

- **Keep data patterns consistent** (compressible vs incompressible) to avoid misleading results on consumer SSDs.
- **Isolate the host** (CPU governor fixed, background tasks minimized) to reduce host-side noise.
- **Note interface ceilings** (e.g., SATA ~550 MB/s; PCIe 3.0×4 ≈ 3.5 GB/s; PCIe 4.0×4 ≈ 7.5–7.8 GB/s) when interpreting limits.

Vendor reference (for discussion)

The Intel Data Center NVMe SSD **D7-P5600 (1.6 TB)** lists **≈130K 4 KiB random write IOPS**. Compare your results to this enterprise spec and explain discrepancies (e.g., SLC caching, data pattern compressibility, controller/firmware policy, interface limits, host CPU effects). Provide reasoned analysis in your report.