Identifying latency outliers in workload testing

Sayali Shirode
Systems Performance Engineer

Micron®

Problem statement

When running and collecting workload traces of RocksDB, we sometimes see large latency spikes. To debug this issue, we used FIO, which is relatively "simple tool" and we still see the latency spikes.

Is this the SSD misbehaving, or is it the system?

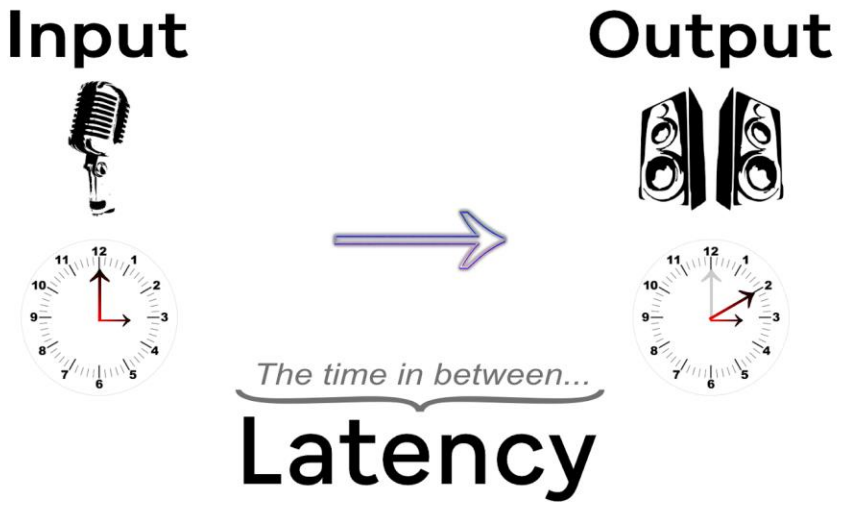
Real world application – RocksDB YCSB

- RocksDB is a storage focused NoSQL database maintained by Meta.
- After executing multiple runs of RocksDB YCSB read-heavy workload, started observing read latency of 113ms, expected was <5ms

Are these high latencies from the SSD or the system?

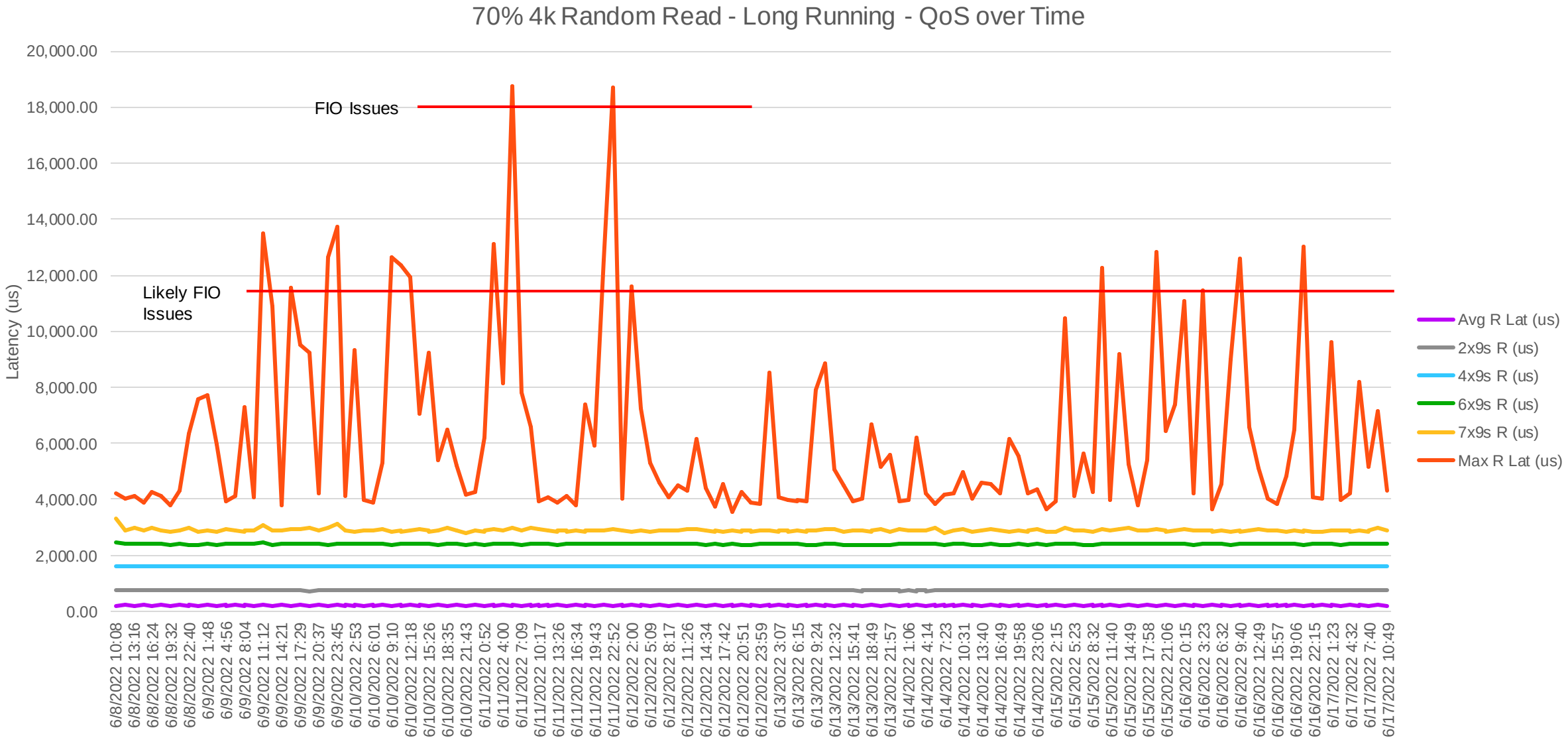
Why latency is critical for SSDs?

- Lower latency means faster access to data
- Impact the performance of applications and user experience.
- The factors affecting the latency include various hardware components, network stack, workload characteristics, storage architecture, software stack.



Synthetic workload - FIO

- Max latency of 18ms
- FIO job is stuck, because the core was busy with some other operation and not able to reach the completion command
- Takeaway – Latency excursion is not a drive issue, but likely the behavior of layer above it
  - – Even “simple” tools like FIO can experience system effects that result in the reporting of high latencies
  - – Noise on the system can affect max latency reporting in FIO.
    - QoS latencies out to 7x9’s look consistent


70% 4k Random Read - Long Running - QoS over Time
Latency (us)
20,000.00
18,000.00
16,000.00
14,000.00
12,000.00
10,000.00
8,000.00
6,000.00
4,000.00
2,000.00
0.00
FIO Issues
Likely FIO Issues
Avg R Lat (us)
2x9s R (us)
4x9s R (us)
6x9s R (us)
7x9s R (us)
Max R Lat (us)
6/8/2022 10:08
6/8/2022 13:16
6/8/2022 16:24
6/8/2022 19:32
6/8/2022 22:40
6/9/2022 1:48
6/9/2022 4:56
6/9/2022 8:04
6/9/2022 11:12
6/9/2022 14:21
6/9/2022 17:29
6/9/2022 20:37
6/9/2022 23:45
6/10/2022 2:53
6/10/2022 6:01
6/10/2022 9:10
6/10/2022 12:18
6/10/2022 15:26
6/10/2022 18:35
6/10/2022 21:43
6/11/2022 0:52
6/11/2022 4:00
6/11/2022 7:09
6/11/2022 10:17
6/11/2022 13:26
6/11/2022 16:34
6/11/2022 19:43
6/11/2022 22:52
6/12/2022 2:00
6/12/2022 5:09
6/12/2022 8:17
6/12/2022 11:26
6/12/2022 14:34
6/12/2022 17:42
6/12/2022 20:51
6/12/2022 23:59
6/13/2022 3:07
6/13/2022 6:15
6/13/2022 9:24
6/13/2022 12:32
6/13/2022 15:41
6/13/2022 18:49
6/13/2022 21:57
6/14/2022 1:06
6/14/2022 4:14
6/14/2022 7:23
6/14/2022 10:31
6/14/2022 13:40
6/14/2022 16:49
6/14/2022 19:58
6/14/2022 23:06
6/15/2022 2:15
6/15/2022 5:23
6/15/2022 8:32
6/15/2022 11:40
6/15/2022 14:49
6/15/2022 17:58
6/15/2022 21:06
6/16/2022 0:15
6/16/2022 3:23
6/16/2022 6:32
6/16/2022 9:40
6/16/2022 12:49
6/16/2022 15:57
6/16/2022 19:06
6/16/2022 22:15
6/17/2022 1:23
6/17/2022 4:32
6/17/2022 7:40
6/17/2022 10:49

How was the latency measured?

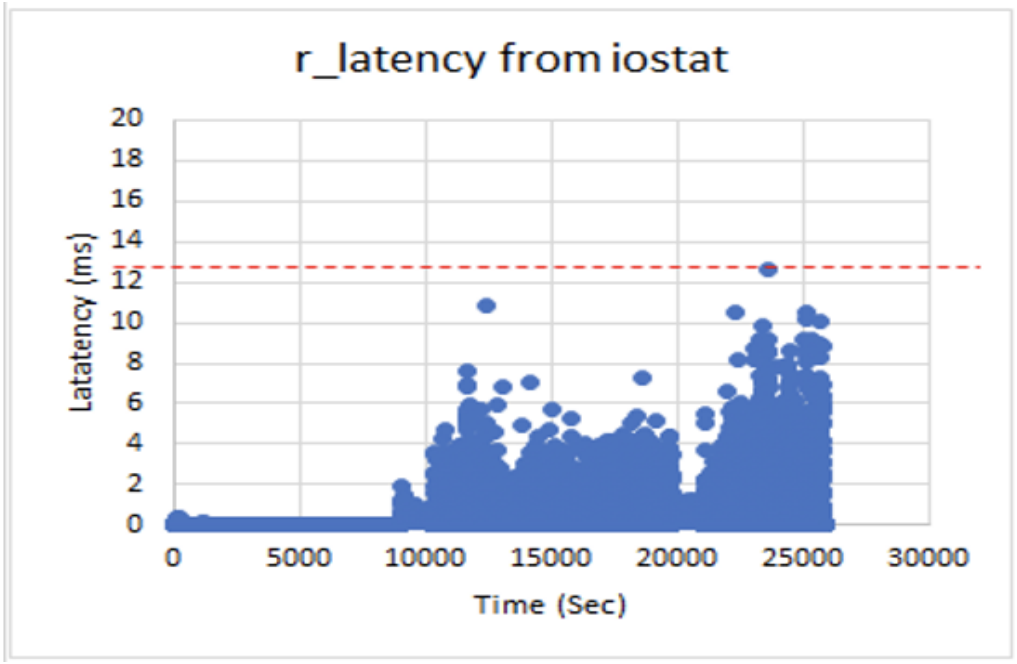
- BPF trace scripts
- Bpftrace – tracing framework for Linux that allows to trace and profile various aspects of the system at runtime
- Biosnoop – details for each disk I/O with latency and look for time-ordered patterns
- OCP Latency monitor

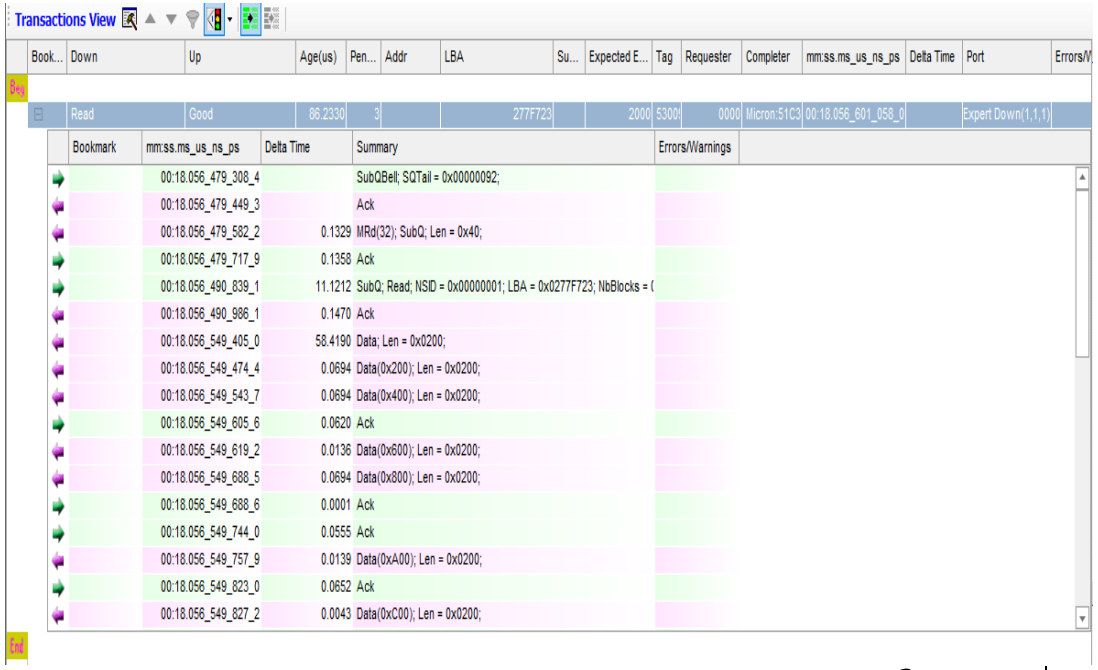
```
#!/usr/bin/env bpftrace
/*
 * biosnoop.bt   Block I/O tracing tool, showing per I/O latency.
 *               For Linux, uses bpftrace, eBPF.
 *
 * TODO: switch to block tracepoints. Add offset and size columns.
 *
 * This is a bpftrace version of the bcc tool of the same name.
 *
 * 15-Nov-2017  Brendan Gregg   Created this.
 */

#ifndef BPFTRACE_HAVE_BTF
#include <linux/blkdev.h>
#include <linux/blk-mq.h>
#endif

BEGIN
{
        printf("%-12s %-7s %-16s %-6s %7s\n", "TIME(ms)", "DISK", "COMM", "PID", "LAT(ms)");
}

kprobe:blk_account_io_start,
kprobe:__blk_account_io_start
{
        @start[arg0] = nsecs;
        @iopid[arg0] = pid;
        @iocomm[arg0] = comm;
        @disk[arg0] = ((struct request *)arg0)->q->disk->disk_name;
}

kprobe:blk_account_io_done,
kprobe:__blk_account_io_done
/@start[arg0] != 0 && @iopid[arg0] != 0 && @iocomm[arg0] != ""/
{
        $now = nsecs;
        printf("%-12u %-7s %-16s %-6d %7d\n",
            elapsed / 1e6, @disk[arg0], @iocomm[arg0], @iopid[arg0],
            ($now - @start[arg0]) / 1e6);

        delete(@start[arg0]);
        delete(@iopid[arg0]);
        delete(@iocomm[arg0]);
        delete(@disk[arg0]);
}

END
{
        clear(@start);
        clear(@iopid);
        clear(@iocomm);
        clear(@disk);
}
```

```
# biosnoop
TIME(s)        COMM           PID    DISK    T  SECTOR    BYTES   LAT(ms)
0.000004001    supervise      1950   xvda1   W  13092560  4096       0.74
0.000178002    supervise      1950   xvda1   W  13092432  4096       0.61
0.001469001    supervise      1956   xvda1   W  13092440  4096       1.24
0.001588002    supervise      1956   xvda1   W  13115128  4096       1.09
1.022346001    supervise      1950   xvda1   W  13115272  4096       0.98
1.022568002    supervise      1950   xvda1   W  13188496  4096       0.93
1.023534000    supervise      1956   xvda1   W  13188520  4096       0.79
1.023585003    supervise      1956   xvda1   W  13189512  4096       0.60
2.003920000    xfsaild/md0    456    xvdc    W  62901512  8192       0.23
2.003931001    xfsaild/md0    456    xvdb    W  62901513  512        0.25
2.004034001    xfsaild/md0    456    xvdb    W  62901520  8192       0.35
2.004042000    xfsaild/md0    456    xvdb    W  63542016  4096       0.36
2.004204001    kworker/0:3    26040  xvdb    W  41950344  65536      0.34
2.044352002    supervise      1950   xvda1   W  13192672  4096       0.65
[...]
```

Experiment

- Iostat - system input/output statistics for devices and partitions
- Bus analyzer – Captures and analyzes the data transmitted on PCIe bus
  - Individual transaction information – like transaction type, address, data payload
  - Shows the signal timings





Conclusion

- Latency between submission and completion on PCIe link is not seen to be 100ms
- Latency monitor does not show high latencies either
- We can conclude that these high latencies are on the system stack

References

- BPF Performance Tools (Book) (brendangregg.com)
- GitHub - brendangregg/bpf-perf-tools-book: Official repository for the BPF Performance Tools book
- https://www.opencompute.org/