Flash-Based SSDs

Annalise Tarhan

March 18, 2021

1 Run the simulation with the following flags. Can you figure out which operations took place?

Seed 1
write (12, u) → success
write(32, M) → success
read(32) → M
write(38, 0) → success
write(36, e) → success
trim(36) → success
read(32) → M
trim(32) → success
read(12) → u
read(12) → u

Seed 2
write(36, F) → success
write(29, 9) → success
write(19, I) → success
trim(19) → success
write(22, g) → success
read(29) → 9
read(22) → g
write(28, e) → success
read(36) → F
write(49, F) → success

2 Now just show the commands and see if you can figure out the intermediate states of the Flash.

Seed 3

```
FTL
EEEEEEEEEE

FTL ?:0
vEEEEEEEEE
0+

FTL ?:0 12:1
vvEEEEEEEE
0+e+

FTL 12:1
ivEEEEEEEE
-e+

FTL 12:1 23:2
ivvEEEEEEE
-e+D+

FTL 12:1 23:2 37:3
ivvvEEEEEE
-e+D+F+

FTL 12:1 23:2 35:4 37:3
ivvvvEEEEE
-e+D+F+S+

FTL 12:1 23:2 35:4 37:3 46:5
ivvvvvEEEE
-e+D+F+S+o+
```

3 Add the -r 20 flag. What differences does this cause in the commands?

This causes 20% of read commands to be issued for invalid addresses, which then fail.

4 Run the same workload again as above, but without showing any intermediate states. Default erase time is 1000 microseconds, program time is 40, and read time is 10. Can you estimate how long this workload will take to complete?

Seed 4
In the first block, there are six valid pages and no invalid ones, implying that out of the 10 commands, six were writes, four were reads, and none were trims. The first write caused the entire block to be erased, costing 1000 microseconds. The six writes cost 40 each, or 240 microseconds total. The four reads cost a total of 40 microseconds, for a grand total of 1280 microseconds for the entire workload.

5 Now, compare the performance of the log-structured approach and the direct approach. First, estimate how you think the direct approach will perform, then check your answer. In general, how much better will the log-structured approach perform than the direct one?

At first, the difference won't be enormous, since the first write costs an entire block erase with any approach. Quickly, though, the log-structured approach amortizes the cost with subsequent cheap writes, but the direct approach will cost a block erase and block re-write every time it either overwrites an old value or writes to a new block. The difference in cost is basically a function of the number of writes, so for a seed (like seed 5) with six writes and four reads, most of the writes will cost 1040 microseconds each, plus another 40 for reads. In addition, any valid pages in an erased block will need to be read and rewritten. The log-structured approach costs 1000 microseconds the first time, but only 40 each for the six writes and another 40 for reads. The total is then 1280 microseconds for the log-structured approach versus around 6280 for the direct approach.

6 Run the following workload, which includes 1000 commands, without garbage collection. What do you think will happen?

Once the log is full, even if some pages have been invalidated, it won't be able to accept any new writes, so any write commands will fail.

7 Turn on the garbage collector by setting -G and -g. What watermark values do you think will make for a working system?

Based on the workload from the previous problem, the high watermark must be set to 7 or less, and the value of the low watermark doesn't matter.

8 Run the following workload with the -J flag, which shows the garbage collector's activity. What do you notice about the GC? How many extra reads and writes occur due to garbage collection? Compare this to the ideal SSD; how much extra reading, writing, and erasing is there due to the nature of Flash? Compare it also to the direct approach.

Once the SSD starts to fill up, about half of the reads and writes occur due to garbage collection. Compared to the ideal SSD, the log-structured approach takes about five times as long, mostly due to all the erasing. The direct approach is still far worse, taking a full 30 times longer than the ideal.

9 Add a workload skew, which causes a higher fraction of the writes to go to a smaller part of the logical block space. Pick some different skews and perform many randomly-chosen operations. What do you expect will happen? What might the impact on the garbage collector be of skewing the workload such that the first 100 writes are not skewed, but the rest are?

The 80/20 skew had a negative impact with the direct approach, but a positive impact with the log-structured approach. The tighter the skew, the bigger the benefit. Making the first 100 writes random instead of skewed had little impact with the direct approach, but almost completely eliminated the improvement the skew created with the log-structured approach.