Gaurav Singh | UFID: 36398850

COP 6726: Database Systems Implementation
Spring 2018
Weekly Assignment 2

01-30-2018

- SSD is not just one technology, it has many different types of implementations all of which fall under the umbrella of Solid State Drives.
    - SLC – Single Level Cells - was the good old tech, think of it as the start of this technology, now imagine you could write 2 levels in a particular unit of length.
    - MLC – Multi Level Cells - you can write 4 levels here and thus store more data in the same space.
    - TLC – Triple Level Cells – You can write 8 levels here, thus again doubling the amount of data stored in the same space.
- P/E cycles: These tell how much can you use a cell before it degrades and can't store data reliably. For HDD, it's in millions we don't even talk about it much, but for SSDs its much more limited.
- 100K for SLC, 10K for MLC, 5K TLC, but people didn't care if the bits died faster, they just wanted more storage, so we have primarily shifted to MLC and TLC.
- Seek Latency: in SSDs its nonexistent but its 9000 for HDD because of the need of header movement.
- Erase latency is SSD is really high. They are easy to turn zeros to one, but cost of going from 1 to 0 takes longer. Also, you have to remove large amounts of data when performing that operation.
- Information is SSD is also written in blocks, but SSD saves collection of blocks together as Page.
- SSD performance is significantly high and designing a DB that uses it would be great.
- Reads are truly exceptional in SSDs
- As long as you can push data out of OS layer to Physical Layer fast enough, you should be fine.
- Gaming PC have 96 PCI bus.
- None of the IO directly talks to the processor.
- NUMA – Non-Uniform Memory Access
- If you could control layout of data structure in memory you can run the code faster.
- L1 cache works in 1 clock cycle, L2 – 10-14 clock cycles, L3 – 20-30 clock cycles.

02-02-2018

- C++ templates can be problematic, be careful if using those.
- Use git blame plugin and you can see who exactly wrote which line of code.
- Mostly DBs are not in Petabytes, in fact they are in the order of TBs.

- It is kind of quircky that many consumer drives can hold that much data.
    - L1: 16KB instructions + 16KB data
    - L2: 512KB per processor
    - L3: 20-40 MB common data
- L1 has only 64 entries.
- GPGPU are idiot proof as you cannot write code if it didn't match the architecture.
- If you have hashes which can fit in L2 cache, you would see much performance improvement.
- With Pentium, came new things all together.
    - Pipepline Execution
    - Out of order execution
    - Multiple Execution Units
- Multiple Execution Units:
    - They added 2-3 ALU(Arithmetic and Logical Units) and
    - 2-3 FPU(Floating Point Units)
    - These enabled many instructions to be processed at any time for the CPU
- Out of Order Execution
    - Intel found that even the best most good compiler also would leave the execution units not running, so they just made very smart systems that could see incoming code and run instructions out of order to keep the CPU engaged as much as possible.
- Pipeline Execution
    - They made 20 – 30 stages that meant that at least it felt as the code was moving along faster.
    - Pipeline flush is bad, try to make the code as simple as possible, that ways processor is not working extra hard to see what you did.