Persistent Data Storage: Disks, (Tapes?), Flash, NVRAMs

Kai Shen

Where we are in the class?

- So far focused on big data programmer's perspective:
  - Big data applications
  - Big data programming paradigms and how to program
- We move on to computer systems issues:
  - Not to make you a computer systems expert, but be knowledgeable on computer systems issues that matter to big data
  - Generally bottom → up, and small → large-scale

Persistent Data Storage

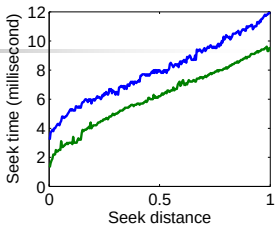
- Persistent storage retains data after sudden system crashes and power losses
  - Disks, tapes, SSDs, Flash drive, non-volatile memory (NVRAMs), …
- As a contrast, memory
  - isn't durable, not surviving software/system/power failures
  - is still comparatively expensive ($/GB)
- Persistent storage is durable, cheap (in general), but slow (though slow in a complex way)

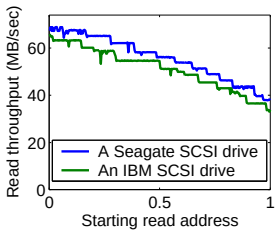
Persistent Storage for Big Data Applications

- When data is too large (or expensive) to fit into memory
  - The crawled web
  - Inverted web indexes for search
  - Online image repository
  - Movie databases
- Collected/output data that needs to be durable
  - Financial transactions, medical records
  - Outputs of data processing

Disk Characteristics

- Characterization of two disks
- Long seeks, but relatively fast sequential access
  - access ~400KB data in the time of a single seek
- Implication in system design:
  - infrequent bulk sequential access
  - put co-accessed data near each other
  - … …





Disks vs. Tapes

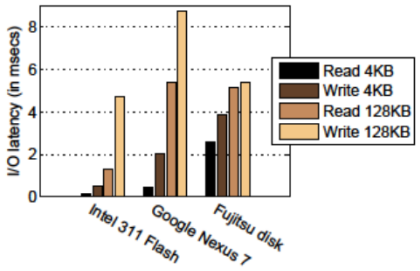
- Disks vs. tapes:
  - Performance characteristics: tapes are horrible for non-sequential access, and even for fine-grained sequential writes
  - Economical sense: tapes are much cheaper. Is it still true?
- Tapes are for archival
  - Backup/archival through disk replication

Solid-State Disks (SSDs)

- Solid state: no (or almost no) mechanical moving parts
- RAM-based disk (with battery and backup disk)
  - Pro: fast! In fact, latency bound by software processing (interrupts), throughput bound by memory bus bandwidth
  - Con: complex backup/restoration during power loss
  - Con: bulky (containing battery)
  - Con: expensive (in hardware and software)

NAND Flash SSDs

- Truly solid state



- Much faster than disks for small I/O, particularly on reads
- Writes are slower than reads due to erasure limitation
- Writes wear out Flash quickly

Storage Performance Implication

- Storage performance
  - Generally slow for disks
  - Slow writes for NAND Flash
  - How much you writes, and how many times you write
- Writes in big data applications
  - Data collection writes a lot
  - Data processing doesn't write as much, but still writes processing results and checkpoints
- Performance implication
  - Sync your data infrequently to improve I/O performance
  - Tradeoff between I/O performance and data durability

Storage Product Varieties and Marketing

- SCSI vs. ATA disks: Beyond interface difference, mostly a business-oriented economical argument [Anderson et al. 2003]
  - Some are willing to pay a lot more with a bit more reliability, controllability, and sometimes performance (typically at a loss of space capacity)
- Google (at least early day Google) didn't buy it
- Different Flash storage
  - Sophistication of the device controller (basically software)
  - Enterprise vs. consumer Flash, desktop/server vs. smartphone/tablet

Storage Reliability

- What does reliability mean?
  - Does not die until after a long lifetime
  - Keep data that it has promised to keep
- Lifetime measurement
  - Need patience and scale
  - Extract from system logs and correlate with workloads, temperature, …
- Does it keep the data?
  - Repeated crash tests – fast, expensive drives aren't always most reliable

Where to keep data? Memory vs. Disk

- Economical rule about whether to keep data in memory or on disk
  - disk system is constrained by throughput
  - memory is constrained by space
- Five-Minute Rule: [Gray&Putzolu 1987]
  - one disk access per second costs about $2000
  - 1KB memory costs $5
  - breakeven economical point – 1KB data accessed once per 400sec
- Seem to still approximately hold after 10 years [Gray&Graefe 1997]
- Absolutely not true for SSDs
  - cause for re-evaluating storage-related system designs (file system, databases, etc.) [Graefe 2007]

Where to keep data? Disk vs. Flash SSD

- Economical rule about whether to keep data on disk or Flash
  - disk system is constrained by throughput
  - Flash is constrained by space
- Breakeven point
  - 60 IOPS per GB [Narayanan et al. 2009]
    - Too expensive to move to Flash SSDs
  - 1.5 IOPS per GB [Albrecht et al. 2013]
    - Figure 4 of the paper
    - Worth doing it selectively for some workloads

NVRAMs or Persistent Memory

- Phase change memory, STT-MRAM, memristor, …
  - Like normal memory, byte-addressable, but persistent
  - Even more expensive than Flash ($/GB)
- People already started talking about a day when Flash is irrelevant
  - At least Prof. Ipek

Storage Considerations

- Tapes → disks → Flash → NVRAMs
- What do I use for my big data systems?
  - Operating costs matters, as shown in previously mentioned studies
  - Latency matters too
  - Software and tools
  - Reliability
- Big trend
  - Manufacturing capability (upfront investment)

Storage Deduplication

- Lots of redundancy in your data
  - Examples?
  - Deduplication conserves space, making it more affordable to phase out an old storage technology
    - Data Domain [Zhu et al. 2008] argued that tapes should be replaced by disks
- How does it work?
  - Keep hashes of data blocks
  - Compare hash of a new write with hashes of existing blocks
  - Be mindful of the cost of hash maintenance and lookup
- Content-addressable storage