DISKS & Storage

Today's topics:

- Faults & RAS
- RAID models
- Some underlying disk technology

very brief - more complicated than you might guess
more depth will appear in CS7810

Reliability

- RAS
  - reliability - absence of observable faults (hard, soft, human)
    » redundancy is always the key here
  - availability - system level concept
    » does it still supply the service
    » how much degradation under certain fault models
  - serviceability
    » can system be repaired while it's running
      • lots of engineering issues to enable hot-swap

Faults

- Categories
  - HW
    • did something break
      • several types: wire, component, connector, power supply, cooling, ...
  - design
    • bug in either software or hardware
      • check known errors in any current uP
      • software workarounds are key until next fab run
  - operational
    • most common: screw up by operations/maintenance staff
  - environmental
    • power or network loss, fire, flood, sabotage, ...

Fault Types

- Transient
  - non-recurring
    • causes
      • environmental noise event - lightning
      • alpha particle strike
    • basically impossible to find so you need to compensate by design
      • parity, CRC, ... reboot
  - intermittent
    • recurring but somewhat rare
      • cross-talk
      • transistor malfunction at a certain temp that is rare
    • again compensate by design
  - permanent
    • something just breaks and stays broken
    • finding these are typically easy
    • compensate & service to meet RAS target
Failure Reality

- System is what we care about
  - sum of it's components – weakest link theory applies
  - N components fail N times more often
    - think early multi-engine airplanes
  - today small number of components have increased system reliability
  - somewhat surprising IC property
    - IC failure rate has remained fairly flat
      - even w/ Moore's law growth of transistors
      - we are likely entering a different era
      - how to build reliable systems from flakey components?
        - hot current research topic

• Metrics

\[
\text{ModuleAvailability} = \frac{\text{MTTF}}{\text{MTTF + MTTR}}
\]

FIT Metric

- 1 FIT = 1 failure in 10^9 hours
  - FIT := failure in time (billion hours)
    - billion hours = 114,155 years
    - 3-5 year expected lifetime
    - need ~10^5 FIT reliability

- \text{MTTF} = \text{MTBF}
  - calculating MTBF
    - \( \sum_{i=1}^{n} q_i \times r_i \)
    - \( n \) is the total number of components
    - \( q_i \) is the quantity of the \( i \)th component
    - \( r_i \) is FIT rate of \( i \)th component

Improving Reliability

- Make better parts
  - doable in some cases & huge cost adder
- Use less parts
  - natural consequence of higher levels of integration
- Employ redundancy
  - common choice
    - 2x – OK as long as we agree
    - 3x – vote and 1 can fail
    - Nn – vote and (N/2)-1 can fail
  - duplicate what?
    - bits, components, wires, gates, ...
    - huge choice set
    - bits and components are common choices today
    - wires and gates may be in our future
    - if intra-IC defects become flakey
- Bottom line – Pandora’s box just opened
  - Dan Siewiorek’s book is an excellent reference text

Failure Model

- No design makes sense without a reasonable failure model
  - amazing how many times this mistake is made
  - how reliable does your system have to be & what are the consequences of failure
  - note difference between PC and nuclear power plant monitors
  - characterize your components
    - MTBF equation comes into play
- Examples
  - transistors and wires fail on a chip
    - highly localized
  - noise → burst errors in transmission
  - disk → oxide deterioration affects an area
    - area likely to expand over time
Reliability, Disks, and Modern Systems

• Think selfishly
  • what would be a bigger disaster
    » losing your files
    » losing your PC
  • If they are the same, you really should fix this YESTERDAY

• The point
  • we view disk storage as archival in most cases
  • backups are increasingly on disk
    » commercial archives are often tape based for "old stuff"
    » cheaper but pain in the tuckus to retrieve from the cave
  • checkpoints are always on disk
  • NVRAM option may be cost effective in the future
    » more on this next lecture
• So let’s look at disk reliability
  • and then a brief glance at the underlying technology

RAID

• 1987 – Redundant array of inexpensive disks
  • Patterson, Gibson, Katz @ UCB
    » Gibson now at CMU
    » Katz made it happen while at DARPA
    » now it’s everywhere

• Reliability through redundancy
  • key idea is to stripe data over more than 1 disk
  • avoid disaster on a single point failure
    » e.g. head crash, AWOL controller, ...
    » even better
      • make sure disks are physically separate
        » ERM or earthquakes take out a warehouse
      • striping model determines RAID type
        » also improves access time for large files
        » no additional seeks between tracks
        » also impacts cost

RAID 0

• No redundancy
  • hence a bit of a misnomer
  • cheap but unable to withstand a single failure
    » except for those correctable w/ block CRC’s
      • access advantage is the only benefit


RAID 1

• Mirroring
  • files on both disks
  • CRC check block option says if one disk fails you’ll know
    » you’re betting that both won’t fail concurrently
  • note interesting option
    » read disk that delivers first
      • if taken this destroys arm synchronization which will penalize
        writes
      • as usual – you want to optimize the common case which is read
        access
  • most expensive
    » 2x disks for x capacity
    » w.r.t. RAID 0
      • read energy minimized – same as RAID 0
      • write energy doubles over RAID 0
      • large block access benefit may be less
RAID1 Variants

- **RAID 1+0 (a.k.a. RAID10)**
  - striped mirrors
  - n pairs of disks (4 disk minimum)
  - think of n RAID1 pairs
  - benefit is access time due to striping
  - but more disks = cost
- **RAID 0+1 (a.k.a. RAID01)**
  - mirrored stripes
  - 2 sets of n/2 disks
  - pairs are not fixed
  - and mirroring happens on a separate set
  - more complex than RAID 1+0
  - benefit is multiple drive failure in 1 set’s mirror won’t cause
    - loss

RAID2

- Hamming code parity
  - ECC style memory correction
  - # disks will depend on ECC model
  - If ECC is on the same disk as data then you lose
  - result
    - many configurations possible
    - tend to be rare (non-existent?) in practice
    - better for mental gymnastics than products

RAID3

- Bit-interleaved parity
  - use one additional check disk to hold parity information
    - $Dp = D1 + D2 + D3 + \ldots + Dn$ (+ = XOR here)
  - lose one disk and all is well
  - failure recovery is longer but cost is reduced since there’s
    only 1 extra disk
  - typically a wise choice since failure is rare
  - potential problems
    - writes: all disks must be accessed to determine parity block
    - parity disk is always hammered
    - disks must be rotationally synch’d
  - byte level striping is common
    - e.g. very small block
    - high performance is result

RAID4

- Same idea as RAID3 but with an optimization
  - interleave blocks rather than bits
  - only read the modified disk
  - note the change
  - then fix the parity block
  - allows read parallelism
  - nice balance of small vs. large reads vs. writes
  - remaining problem - parity disk needs to be accessed on every
    write -> bottleneck source
RAID5

- Striped set with distributed parity
  - Interleave the check disk

RAID6

- Striped set with dual distributed parity
  - take RAID5 and add a second check function & disk
  - now resilient under double disk failure
    - parity for each check function must lie on a different disk

Beyond RAID6

- with multiple RAID controllers a hierarchical disk system can be employed
- there are officially RAID7/8/9 as well
  - but hopefully this is sufficient to get the idea across

Non-standard RAID systems are also deployed

- many just slightly varied from a standard RAID approach
- Software RAID – e.g. OS based controller
  - MAC OR X - RAID 0, 1, or 10
  - Linux - RAID 0, 1, 4, 5, 10, ...
  - others exist – slower than HW versions but portable

CGR Better than Moore’s Law

Interfaces & Improvement

- Interfaces
  - Control moves onto the disk
    - replaces motherboard control
    - now - microprocessor and SRAM inside the disk
  - Parallel to high speed serial interfaces
    - limited by short fat cable issues
    - serial Fiber Channel – 1997, SAS, SATA
    - serial enables storage area networks (NAS)
  - Key improvement contributors
    - thinner magnetic platter coating
    - improvements in head design
    - lower flying height
    - accuracy of head positioning servo
      - hard to do cheaply
      - hence BPI CGR leads TPI CGR
Access

- A disk address
  - Indirectly resolved to
    - surface, radius, angle
  - polar coordinates resolve to cylinder & sector

- Performance
  - as always multiple metrics
    - latency vs. response time
      - since seek and rotational latency varies significantly
    - response time usually averaged over large number of accesses
    - bandwidth vs. transfer rate
      - transfer rate = IOPS * average block size
      - dependent on disk RPM and linear density (BPI)
  - multiple requests queued in disk controller
    - hence response time looks exponential w/ increase in
      - throughput, request arrival rate, utilization
      - e.g., increased queuing delay
    - optimization possible be reordering requests

Disk Futures

- Disk demise oft predicted
  - "greatly exaggerated" as Mark Twain said
- Horizontal to vertical transition underway
  - increased areal density should continue
- MAID might threaten tape for offline storage
  - massive array of idle disks
- Reduced form factor
  - may enable RAID
  - and server storage bricks may become available in PC's
    - brick is a bunch of disks, controller, and battery
    - idea even if power goes down disk writes complete
- Common saying
  - Silicon Valley misnomer
    - more money made due to FeO2 than Si

Disk Storage Layers

- Physical Layer
  - physics and engineering to just make disks work
- Data Layer
  - arrangement of data in blocks, sectors, stripes, ...
- Internal Control Layer
  - what the processor in the disk deals with
- Interface Layer
  - specifics of the drive Interfaces
- Cache or External Control Layer
  - use of caches to improve performance
  - issues in management of multiple drives
    - RAS issues such as RAID
    - power issues such as MAID
    - huge issue for the datacenter

Physical Layer

- 3 major components
  - magnetic recording physics
    - ferromagnetic materials
      - magnetized by external field
      - stable after external field is removed
      - common elements: iron, nickel, cobalt
      - rare earth: gadolinium, dysprosium
      - rapidly quenched metal alloys form amorphous FM materials
    - electron spin creates a magnetic field
      - non-FM materials consist of electron pairs w/ opposite spins
      - FM materials
        - non-paired valence shells
        - long range atomic ordering (aligned in parallel) to form a domain
      - beware the Curie temperature
        - above which the FM material loses to thermal entropy
      - electromagnetic and magnetic components
      - integrated electronics in the drive
HDD Anatomy

source: Jacob's book (similarly with subsequent figures)

Electronics

- Small PCB inside
  - Controller
    - receive commands, schedule, and report back when command executes
    - manage the disk cache
    - interface with HDA – e.g. seek and sector targets
    - error recovery and fault management
    - power management
    - start/stop control

Controller Illustrated

DRAM Role

- 3 distinct roles
  - scratch-pad
    - on power up
      - load protected data from platter
      - defect maps
      - ID tables
      - adaptive operational parameters
    - queue of commands
  - speed matching
    - interface and disk bandwidths and timing differ
  - cache
    - read pages
    - write buffer
Write Channel

- Several duties
  - Limit run length of 0’s
    - No transitions for too long ruins clock recovery
  - Several modulation codes possible
    - Obvious 2 bits/logical bit (50% efficient)
    - Need to consider ISI (inter-symbol interference)
      - Mitigated by write precompensation

Read Channel

- GMR yields < 1mv ΔV
  - Differential preamp located in the AEM
  - AGC (auto gain control)
  - Low pass filter to reduce high-freq noise
- Detection, clock recovery, & decode

And Finally

- Motor controls
  - Simple ADC/DAC
  - But with adaptive correction
    - For positioning drift & thermal issues

Disk Reliability

- Beware the manufacturer claims
  - Data extrapolated on accelerated life test data
    - Environmental tests on a small population
  - And from unit returns
    - No idea how the unit was operated or treated
      - Well, hammer marks might be a clue...
    - Warranty expires in 3 years so > 3 year olds are excluded
- Google data
  - Record data on all of their hard drives every few minutes
    - And save forever (how many disks does that take – YOW!)
Key Findings

- Contrary to popular belief
  - Little correlation between failure and elevated temperature or activity levels
- SMART really isn’t that smart
  - Some SMART parameters have a large impact on failure probability
    - Scan errors, reallocation counts, offline reallocation counts, and probational counts
  - However, large fraction of failed drives had no SMART warnings
  - Hence unlikely that SMART data alone can be used to form an accurate predictive model
- Can’t trust the manufacturer or the drive SMART’s
  - What the heck do you do?
  - Take a statistical approach
    - Hmm – obvious Google theme here

Annualized Failure Rate

Figure changes significantly when stats are normalized by model

SMART data didn’t change by model

Conclusions

- Disks are hugely important
  - 90% of the new world knowledge stored there in 2002
  - Likely higher today
- BUT they fail
  - Predicting failure is hard
    - Common temperature, utilization, power-on-off cycles bad
    - Turn out to be not observable in practice by the Google folks
  - Some SMART data gives you an early warning
  - But less than half of the time
- Bottom line
  - If your data is on one drive
  - You’re screwed
  - So fix this problem YESTERDAY

Final Remarks

- What you should remember
  - RAID schemes
  - Disk storage layers
  - General disk anatomy
- There’s a lot we didn’t cover
  - Huge improvement in materials
    - Platter surface, spindle bearings, data encodings, ...
  - Why?
    - Time & focus on system architecture
- Disks are a big deal in the “cloud”?
  - Probably just as important as the processors
  - Battery backed bricks are common
    - Want to finish outstanding writes before system goes down
- Storage is a complicated space
  - We’ve scratched the surface today