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 work arounds 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**

$$\text{ModuleAvailability} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}$$

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
- **MTTF = MTBF**
  - calculating MTBF
    - » r_i = FIT rate of i^{th} component
    - » q_i is the quantity of the i^{th} component
    - » n is the total number of components

$$MTBF = \frac{1 \times 10^9}{\sum_{i=1}^{n} q_i \times r_i}$$

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
    - » Nx – 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 devices 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

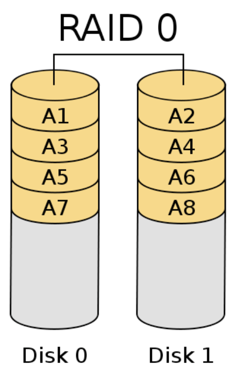
- Think selfishly
  - what would be a bigger disaster
    - » losing your files
    - » losing your PC
    - » if they are the same, you really should fix this YESTERDAY
- The point
  - we view disk storage as archival in most cases
  - backups are increasingly on disk
    - » commercial archives are often tape based for "old stuff"
      - cheaper but a 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 he was 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
        - – EMP or earthquake takes 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

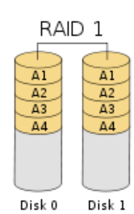
- **No redundancy**
  - **hence a bit of a misnomer**
  - **cheap but unable to withstand a single failure**
    - » **except for those corrrectable w/ block CRC's**
      - **access advantage is the only benefit**



source: Wikipedia

RAID1

- **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. RAID0**
      - **read energy minimized – same as RAID 0**
      - **write energy doubles over RAID 0**
      - **large block access benefit may be less**



RAID1 Variants

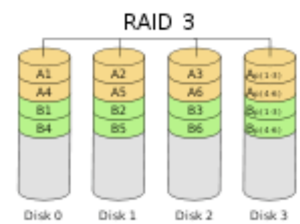
- **RAID 1+0 (a.k.a. RAID10)**
  - **striped mirrors**
    - » n pairs of disks (4 disk minimum)
    - » think of n RAID1 pairs
      - benefit is access time due to striping
      - but more disks → cost
- **RAID 0+1 (a.k.a. RAID01)**
  - **mirrored stripes**
    - » 2 sets of n/2 disks
    - » pairs 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

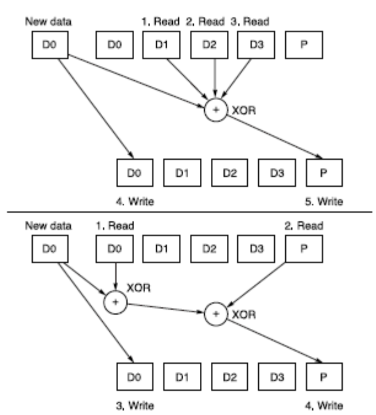
RAID3

- **Bit-interleaved parity**
  - **use one additional check disk to hold parity information**
    - » **Dp = D1 + D2 + D3 + … + 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

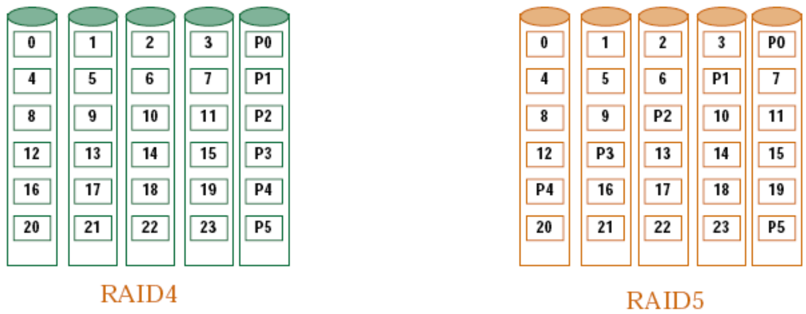
allows read parallelism

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 w/ distributed parity**
  - **Interleave the check disk**



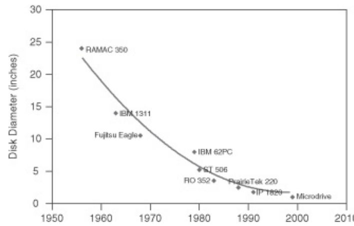
write & read parallelism now supported

RAID6

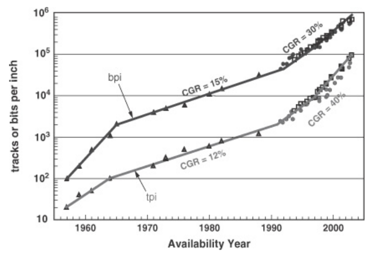
- **Striped set with dual distributed parity**
  - **take RAID5 and add a second check function & disk**
  - **now resilient under double disk failure**
    - » **parity for each check function must lie on a different disk**
- **Beyond RAID6**
  - **with multiple RAID controllers a hierarchical disk 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 OS 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

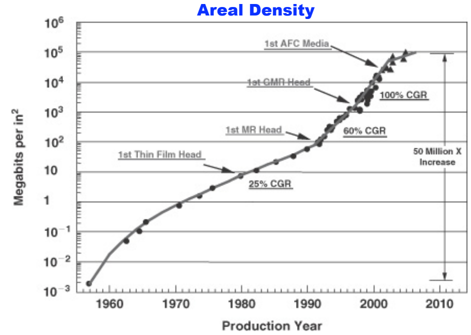
Form Factor



Lineal Density



Areal Density



Interfaces & Improvement

- **Interfaces**
  - **Control moves onto the disk**
    - » replaces motherboard control
    - » now – microprocessor and SRAM inside the disk
  - **Parallel to high speed serial interfaces**
    - » parallel SCSI – 1983, IDE/ATA – 1986
      - 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 ::= response time
      - since seek and rotational latency varies significantly
      - response time usually averaged over large number of accesses
    - » bandwidth ::= transfer rate
      - transfer rate = IOPS*average block size
        - – dependent on disk RPM and lineal density (BPI)
  - multiple requests queued in disk controller
    - » hence response time looks exponential w/ increase in
      - throughput, request arrival rate, utilization
      - e.g. increased queueing 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

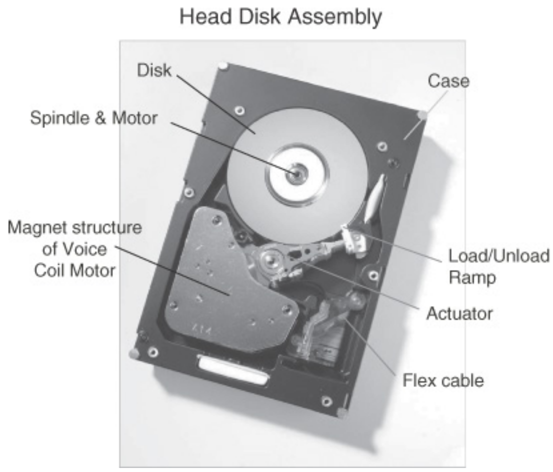
Disk Storage Layers

- **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
  - electromechanical and magnetic components
  - integrated electronics in the drive

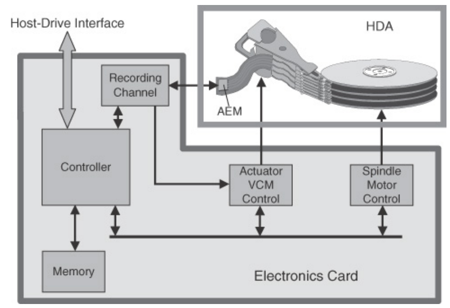
HDD Anatomy



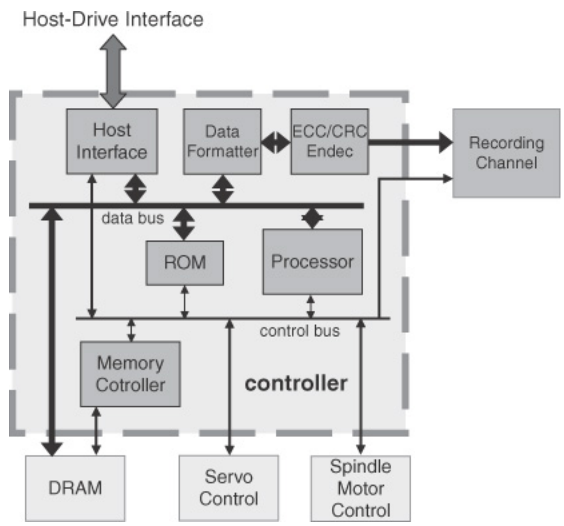
source: Jacob's book (similarly with subsequent figures)

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

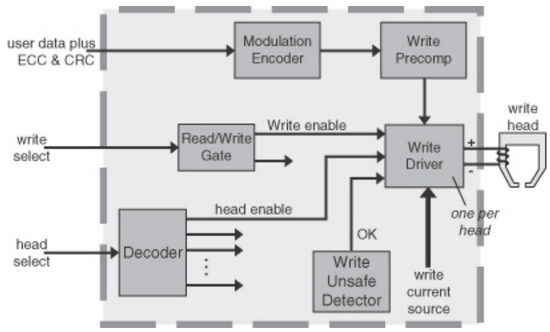


DRAM Role

- 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

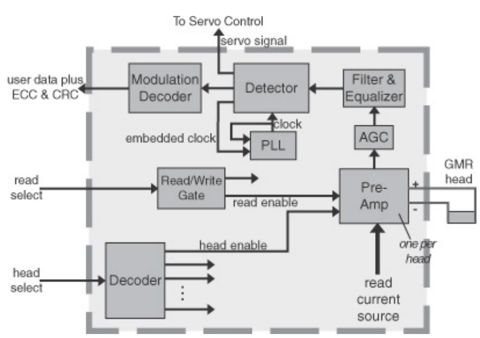
Write Channel

- **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**



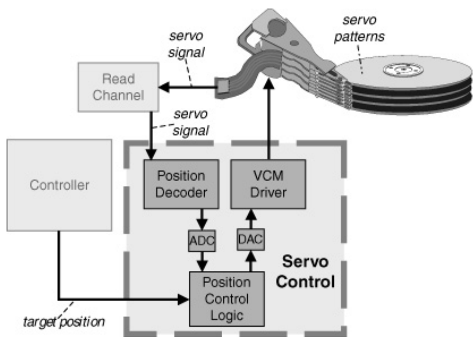
Read Channel

- **GMR yields < 1mv ΔV**
  - **differential preamp located in the AEM**
  - **then 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!)**
    - » **includes SMART parameters**
      - **Self-Monitoring Analysis and Reporting Technology**
      - **believed to be good indicator of drive health**

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

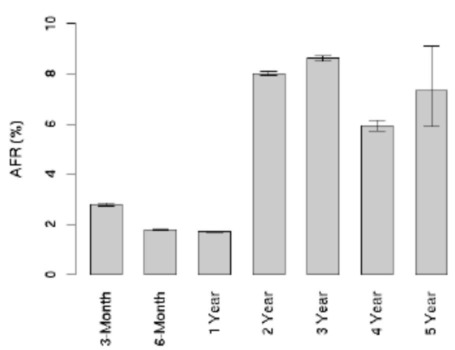


Figure 2: Annualized failure rates broken down by age groups

Note: 3&4 year old failure more correlated to model than age

significant infant mortality rate seen in 3, 6, and 12 month age population

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 you're 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