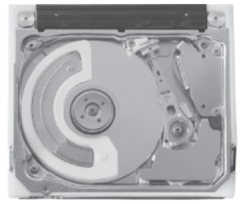
Disk
Data Layer & Other Issues

1955: IBM RAMAC 305



Today: Hitachi MicroDrive

Reference: "Memory Systems: Cache, DRAM, Disk" – Jacob, Ng, & Wang, Ch. 18

"Failure Trends in a Large Disk Drive Population" – FAST07 – Pinheiro, Weber, Barroso, Google

Fixed Size Blocks

- **Ideal case**
  - **contiguous placement**
    - » one seek and rotational latency hit
      - sector reordering in buffer reduces the rotational impact
    - » hard to do for large files
- **Block size choice – fixed vs. variable**
  - **common memory theme**
    - » page allocation to physical memory
- **Fixed size blocks**
  - **+ one unit of allocation – easy map to sectors**
    - » - can't find enough contiguous ones to meet need
      - fix with compaction/GC but this takes time
    - » - internal fragmentation
      - last block will always only be partially full
        - – not a big deal w/ today's densities unless lots of very small files
  - **common: map file to non-contiguous blocks**
    - » trades file access time for reduced GC

Variable Size Blocks

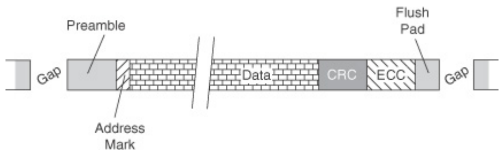
- + No internal fragmentation
- Used in early systems
  - primarily due to small file sizes and crappy density
- Several problems
  - large file writes
    - » not enough contiguous free space
      - GC must happen more often
  - accounting woes
    - » free space entries must be done at a finer grain size
      - increased level of meta-data
    - » count field must be associated with each record
      - or held in metadata

Sectors 1

- Today's choice
  - sectors = fixed sized blocks
  - specified by interface standards
    - » consumer: SATA
    - » and commercial: SCSI, SAS, and FC
- Anatomy of a sector
  - gap
    - » allow time for read and write heads to access the sector
    - » writes – jitter buffer to prevent adjacent sector contamination
    - » also compensates for some drift in clock
  - preamble (a.k.a. sync field)
    - » ~ 10 bytes
    - » establish frequency and amplitude of recorded signals
      - adjust the PLL and AGC circuits

Sectors 2

- **All are vendor specific choices**
  - **data address mark (a.k.a. data sync field)**
    - » a few bytes
    - » special pattern indicates end of preamble and beginning of actual data
  - **data**
    - » 512 bytes RLL encoded into ~544 bytes
  - **CRC**
    - » correct one-time data errors
  - **ECC**
    - » ~40 bytes: burst ECC required
      - both hard and soft error's take out an area



Sector Size

- **Tradition**
  - **512 bytes**
    - » BIOS, drivers, and file system coded for 512B
  - **problems**
    - » as areal density increases
      - burst errors get bigger
      - both SER and HER probabilities increase
    - » hence bigger ECC and CRC fields required
      - efficiency of data to overhead fields goes up
- **Fix w/ bigger sectors in 2005**
  - **4 KB & 512B allowed**
    - » MS Vista written w/ 4KB in mind
- **Internal larger sector**
  - **take pieces of it to make it look 512B at the interface**

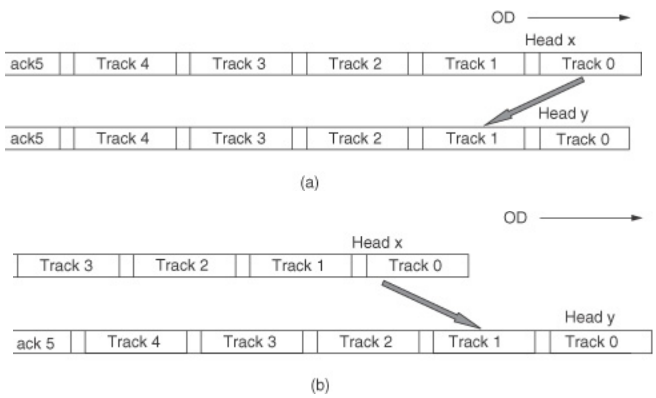
Tracks and Cylinders

- **Track options**
  - concentric rings – used for disks
  - spiral – used for continuous media, CD, DVD, vinyl records
- **Cylinders**
  - vertical set of same tracks on different surfaces
    - » modern precision & variance ➔ not purely aligned
    - » hence switching heads entails minor track center correction
      - max transition ΔV indicates center
- **Numbering**
  - Sectors: 1 to N for each track
  - Tracks: outer track is 0, inner track is m
  - Negative tracks (-1 … -n): even more outer
    - » reserved cylinders to hold non-user data
      - defect maps
      - address maps
      - etc.

Address Mapping

- **External access via sector ID (integer)**
  - essentially a logical block address (LBA)
- **Internally mapped to a physical block address (PBA)**
  - a.k.a. absolute block address (ABA)
  - in reality a CHS index
    - » cylinder, head, sector
- **Next ABA options (n-1 ends track, n starts a new track)**
  - cylinder mode
    - » n is on a different surface
      - minimizes seek
  - track mode
    - » stay w/ existing head
      - minimized head electronics switching delay
    - » turns out today: track mode wins
      - primarily because tracks are not perfectly vertically aligned in cylinder
      - hence seek anyway but which direction

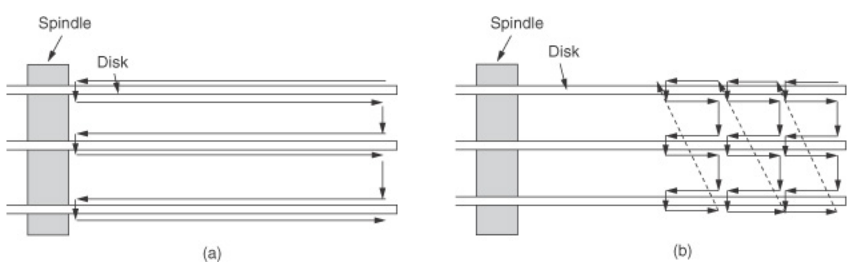
Jaggy Cylinders



High TPI density increases variance

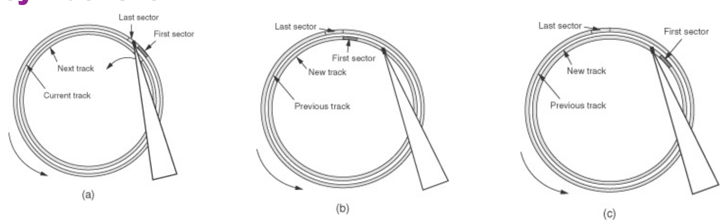
Serpentine Mapping

- **Problem**
  - **stay on same surface as long as possible**
  - **AND stay with out to in numbering → long seek to change surface**
    - » **no need to be stuck with this but devices tend to anyway**



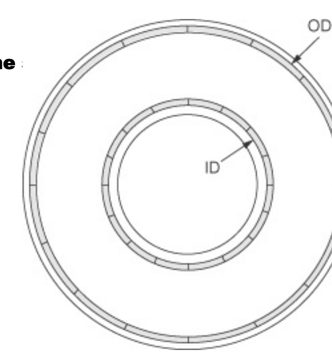
Skewing

- **Track mode – same surface issue**
  - **can't seek in gap time**
    - » **add one whole rev. time to next access**
  - **so stagger the #1 sector (track skew)**
- **Same game can be played in cylinder mode**
  - **cylinder skew**



Variable Density Recording

- **Each bit placed on a radial**
  - **fixed # of sectors/track**
  - **bpi_{MAX} on innermost track**
    - » **cons**
      - **wasted capacity increases as you move to outer tracks**
    - » **pros**
      - **motor rpm stays constant**
      - **bit rate at the heads is always the**
        - – **simplifies timing compliance**
  - **capacity**
    - » **$= tpi \times bpi \times \Pi \times (OD^2-ID_2)/4$**
  - **today's high density disks**
    - » **too much lost capacity**
    - » **higher $/bit**
  - **bad idea**

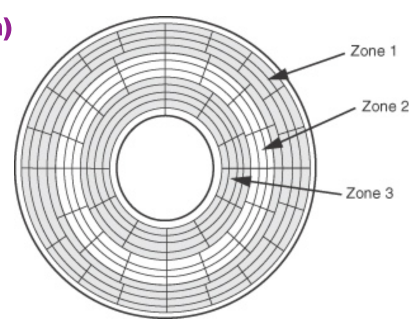


Fixed Density Variable RPM

- **Pro's**
  - **no wasted capacity**
  - **timing per bit constant**
- **BIG NEGATIVE**
  - **change RPM based on track address**
    - » **RPM takes 100's ms to stabilize**
    - » **.5 rotational latency varies by track**
      - **3@ID – 6@OD ms**
      - **> 10x penalty due to the motor**
- **Also a bad idea**

Zoned Bit Recording (ZBR)

- **Divide collections of tracks into zones**
  - **fixed density and RPM**
  - **vary # sectors per zone**
  - **fixed # sectors/zone**
    - » **causes bit rate to vary within a zone**
- **Common**
  - **lots of zones (64-128 common)**
    - » **bit rate variance in zone**
      - **w/in compliance of the clock recovery circuitry**
      - **RLL codes & preamble**
  - **max bpi everywhere**
    - » **no capacity wasted**
    - » **min $/bit**



Servo Comments

- Role – seek to the right track
  - more difficult than it seems
  - inherent drift
    - » continual correction & knowledge of what track you're on
      - since possible to drift into adjacent track
- 2 approaches
  - dedicated
    - » use one surface to store servo information
    - » this head used to track cylinder center
      - non-vertical cylinders
        - – due to thermal variations of platters
      - fix
        - – periodic thermal recalibration
        - – takes 100's ms
        - – access during this time takes way longer than expected
  - least overhead with lots of surfaces
    - » more thermal variation & heat rises → vertical platters

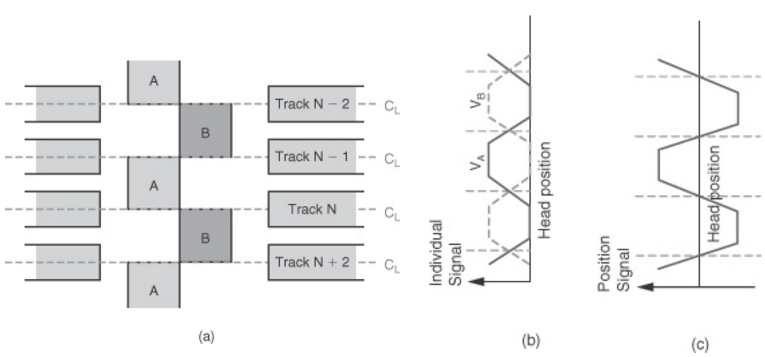
Embedded Servo Data

- Periodic wedges of servo data
  - embedded w/ real data
    - » written at manufacture time
      - 1 disadvantage: care needed to not overwrite servo data
      - not a problem in dedicated approach
- Each head
  - now responsible for it's own servo tracking
    - » hence no thermal issues
      - and no expensive thermal calibration
    - » but seeks get more complicated
      - head needs to confirm it's on the right track via servo data
      - BEFORE it can wait for the right sector to come around
- # of servo sectors per track
  - more: less capacity, drift, and seek time impact
  - typical choice today: 100-200 servo's track

Servo Data

- **2 components**
  - **servo ID**
    - » **polar coordinates of the wedge**
    - » **gray code cylinder encoding**
      - **simplifies seek deceleration**
        - – decelerate based on 1's count of XOR target vs. ID threshold
      - **simplifies tracking**
        - – drift over adjacent tracks read part of servo on one and part on the other
        - – generates more than a 1 bit change → correction needed
  - **server bursts (start simple)**
    - » **duty: keep head over center of the track**
      - **2 special magnetic patterns: A & B**
        - – A_burst is left of center, B_burst is right of center
        - – read channel compares signal strength
        - – if A stronger move right
          - – differential sensing ($V_A - V_B$) positive vs. negative V is feedback
        - – on center when strengths are equal
      - **In practice blocks do double duty**
        - – odd tracks described above, even tracks are opposite

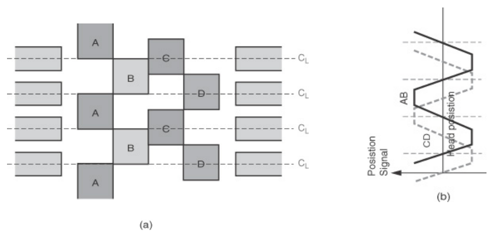
A-B Model Illustrated



Differential PES value:
polarity = direction of error
magnitude = amount of error
analog value sum with VCM drive voltage

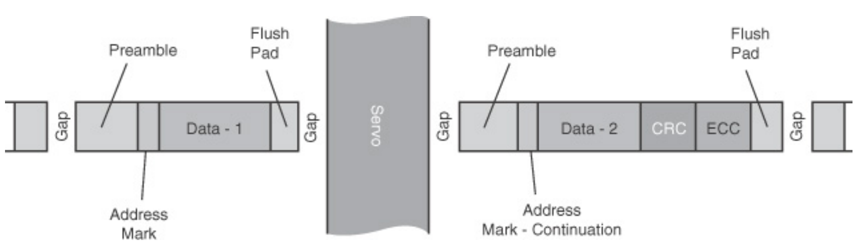
Improving on the AB model

- **AB burst problems**
  - **write wide read narrow & 180 degree phase difference**
    - » **flat spots in the V_A-V_B signal can cause false OK**
- **Current choice: ABCD**
  - **V_C-V_D adds finer grain correction needed for high tpi densities**



Full Servo Wedge

- **2 additional components**
  - **preamble**
    - » **allow PLL synch**
  - **servo address mark (a.k.a. servo sync mark)**
    - » **unique pattern indicating servo data next**



ZBR & Embedded Servo

- Ideally want to place servo wedges between sectors
  - difficult w/ ZBR for all wedges
    - » e.g. 200 wedges & 1001 sectors per track
      - OD: 20% of the sectors will be split by a servo
      - ID: 40% split
  - so let them split
    - » gaps allow time to switch from write to read mode
    - » 2nd sector component needs it's own preamble
  - need to map split sectors
    - » map stored in DRAM and negative tracks
- Split causes overhead
  - recover by moving servo ID to RAM
    - » tracking patterns are unique so count them
  - headerless format

Headerless ZBR

- DRAM data
  - zone number
  - servo number
  - sector number following this servo
  - number of sectors between this servo and the next
  - is the first sector a continuation
  - is the last sector split

Actual Capacity

- In theory
  - capacity = bpi x tpi x recordable_area
- In practice: overheads in terms of area
  - embedded servo wedge overhead: 8-12%
  - overhead due to split sectors: 1%
  - preamble, address mark, CRC, ECC: 12%
  - RLL encoding: 3-8%
  - gaps, ZBR track fragmentation, spare sectors for defect management, negative tracks: 2-4%
- 26-37% capacity lost in order to make it work
  - seem too high
  - welcome to the real world
    - » at least the 74-63% works
- Data rate – similar overheads
  - peak in theory (bps) = bpi x 2 x Π x r x rpm/60

Defect Management

- Some sectors are defective
  - % increases as bpi and tpi are pushed → smaller bit size
    - » lowered signal to noise ration
    - » material defects take out more domains
      - soft error rate w/o ECC: 1 in 10^5
        - 1/20 sectors are bad – unacceptable – more powerful ECC's
      - hard error rate has been held roughly constant as a result
        - 1 in 10^{14} consumer, 1 in 10^{15} commercial servers
        - 1 in 20 or 200 Gsectors respectively
        - note 100 GB disk has about 200 Msectors
      - still it's probability so sector relocation is needed
  - 2 relocation schemes
    - » sector slipping
      - bad sector → shift everything back a sector
        - nice in that it maintains the allocation locality properties
      - sector sparing -> assign LBA to the ABA of a spare
        - only one sector moves
        - locality breaks until disk defragmentation runs
      - hybrid: spare sectors at end of track, slip within a track

Defect Types

- From an operational viewpoint on 2 types matter
  - primary defects
    - » known at manufacturer QA step
    - » ship with P-list (primary defect list) on negative tracks
    - » sector slipping is typical approach here
      - nothing on the disk so no data movement
  - grown defects
    - » happen during the disk lifetime
    - » 2 types
      - permanent – add to G-list
      - transient – quarantine – add to G-list w/ Q tag
        - quarantine too often then G-list permanently
    - » sector sparing use here
      - defragmentation removes the disadvantage
      - takes a long time but there are idle times

Error Recovery

- Single disk
  - reread w/ different head position
    - » covers write jaggies
  - ECC to correct on the fly
  - deep ECC in the processor
    - » involves both ECC and CRC and sometimes works
  - prevent non-recoverable errors
    - » track CRC errors and ECC correction data
      - G-list a sector before you lose the data
- Multiple disk
  - RAIDx
    - » G-list sector on the offending drive
- Lots of other options
  - redundancy somewhere is always the key

Context Switch

- **While we're on reliability**
  - **much of the industry folklore is pure bunk**
    - » **bad: based on small population studies**
    - » **worse: ignore part of the population**
      - no return to the manufacturer then they assume it worked
      - warranty period is short
      - large % of failures likely excluded from this data
    - » **worst: project lifetimes based on selective small sample**
      - MTBF numbers are complete hogwash
- **Interesting paper**
  - **Failure Trends in a Large Disk Drive Population**
    - » **Eduardo Pinheiro, Wolf-Dietrich Weber and Luiz Andr´e Barroso**
      - Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST'07), February 2007
  - **the following is a synopsis of their findings**
    - » **definitely worth reading**
    - » **note their study is based on consumer grade disks**

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**

Google System Health Infrastructure

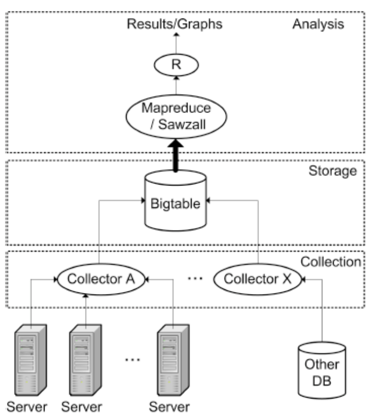


Figure 1: Collection, storage, and analysis architecture.

Daemon on every machine
Collectors of various types
- **machine groups**
  - **environmental parameters**
  - **local SMARTs**
  - **usage data**
- **other DB**
  - **configuration**
  - **repair**
  - **disk swaps**

Bigtable
- **3D**
  - **machines**
  - **parameters**
  - **time**

Analysis via Mapreduce
- **Sawzall language**

Population Details

- > 100K consumer grade ATA drives
  - 5400-7200 rpm
  - 80-400 GB
  - put into production after 2001
  - multiple manufacturers
    - » who remain nameless in this study for obvious reasons
  - operational details
    - » server class rack mounted deployment
    - » professionally managed
    - » initially tested
      - failures here do not count in the data
  - data collected
    - » Dec. 2005 to August 2006

Defining Failure

- Opinions differ
  - manufacturer reports <2% per year
  - Elerath and Shah
    - » 15-60% of failures found to have no defect when returned to the manufacturer
  - Hughes studied 3477 disks
    - » 20-30% of failed drives had no defect
  - Google tests
    - » OK on the bench fails in the field
- Google failure definition
  - drive is considered "failed" if it was replaced
    - » time of failure recorded as replacement time
    - » pretty quick in Google land
  - upgrades don't count
  - spurious or not fully filled out entries not counted
  - odd SMART values were not filtered

Annualized Failure Rate

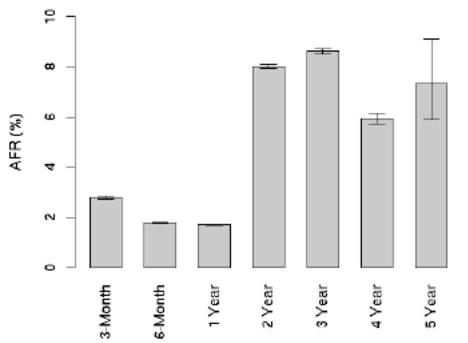


Figure 2: Annualized failure rates broken down by age groups

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

Folklore 1

- **Higher activity is bad**
  - **hard to define duty cycle**
  - **study**
    - » **low = 25%-ile**
    - » **medium = 25-75%-ile**
    - » **high = 75-100%-ile**
- **Results**
  - **true for very old and young**
  - **3 year olds have stamina**
    - » **sounds like the Kentucky Derby**

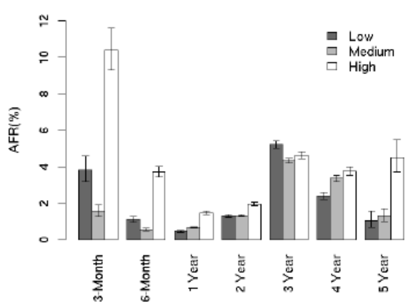


Figure 3: Utilization AFR

Folklore 2

- **Hot is bad**
  - reports indicate 2x failure w/ 15C temp change
- **Results**
  - nope except for older population

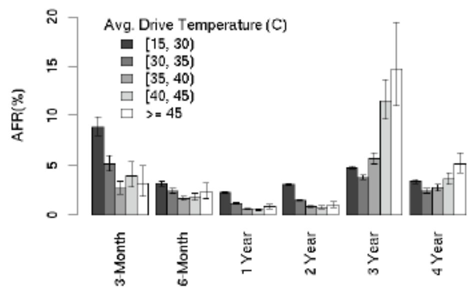


Figure 5: AFR for average drive temperature.

SMART Data Correlation

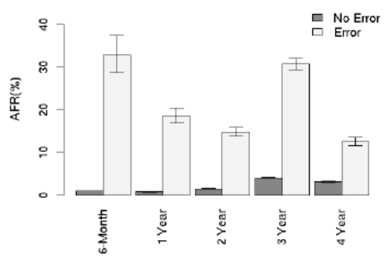


Figure 6: AFR for scan errors.

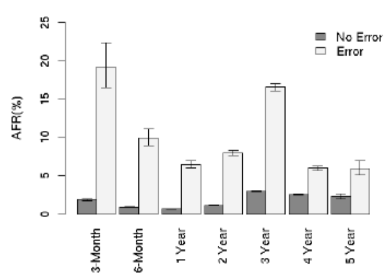


Figure 7: AFR for reallocation counts.

Survival Probability after 1st Scan Error

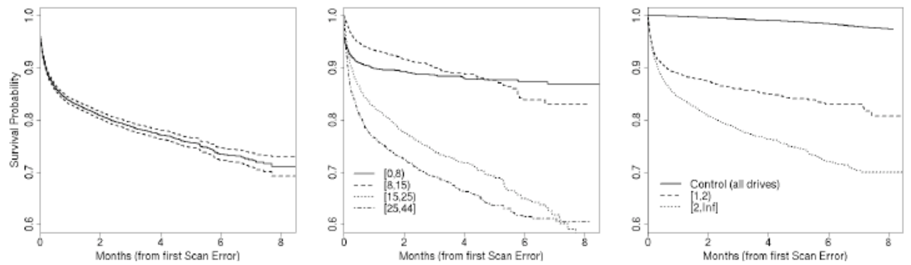


Figure 8: Impact of scan errors on survival probability. Left figure shows aggregate survival probability for all drives after first scan error. Middle figure breaks down survival probability per drive ages in months. Right figure breaks down drives by their number of scan errors.

SMART Looks Good

- **Until**
  - **56% of the failed drives had no smart errors flagged**
  - **if the data is correlated on a per manufacture basis**
    - » **shape of the previous graphs changes a lot**
    - » **one manufacturer had horrible seek errors**
      - **over most models**
      - **wish they'd tell us who not to buy from**
  - **even in extreme 40C temps**
    - » **36% of failed drives had no SMART errors**

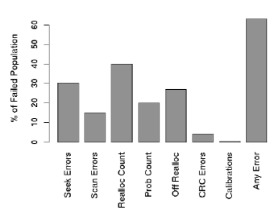


Figure 14: Percentage of failed drives with SMART errors.

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

Disks In General

- Lots of issues that we didn't have time to cover
- Objective here
  - provide the basics
  - enable you to understand the research literature
- Important note
  - disks are disks
  - storage is something very different
    - » it's what the datacenter folks care about
      - only hints of some issues covered here
- Finito

AFR After 1st Count Error

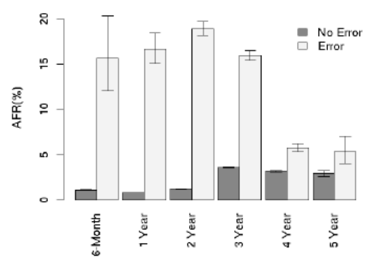


Figure 9: AFR for offline reallocation count.

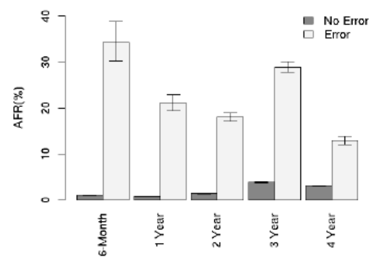


Figure 10: AFR for probational count.

Survival After Count Error

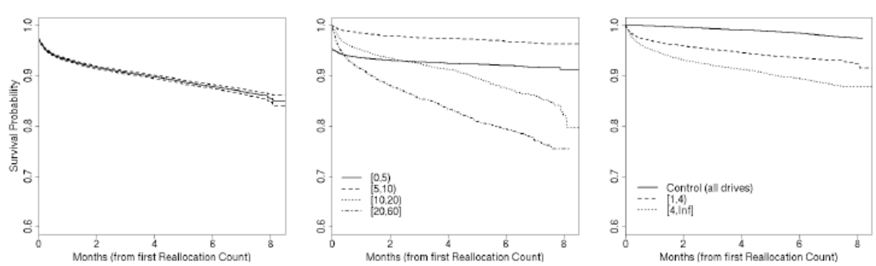


Figure 11: Impact of reallocation count values on survival probability. Left figure shows aggregate survival probability for all drives after first reallocation. Middle figure breaks down survival probability per drive ages in months. Right figure breaks down drives by their number of reallocations.