An Analysis of Data Corruption in the Storage Stack

Garth Goodson
NetApp, Inc

Lakshmi Bairavasundaram
Andrea C. Arpaci-Dusseau
Remzi H. Arpaci-Dusseau
University of Wisconsin-Madison

Bianca Schroeder
University of Toronto

Corruption Anecdote

- There is much anecdotal evidence of data corruption
  - E.g., this is a photo stored on an author's laptop

- System designers know of similar occurrences
  - Data protection often based on anecdotes
- Anecdotes: interesting, but not enough for system design
  - A more rigorous understanding is needed

Our Analysis

- First large scale study of data corruption
  - 1.53 million disks in 1000s of NetApp systems
- Time period
  - 41 months (Jan 2004 – Jun 2007)
- Corruption detection
  - Using various data protection techniques
  - Data from NetApp Autosupport Database
    - Also used in latent sector error [Bairavasundaram07], disk and storage failure [Jiang08] studies

Questions we had about corruption

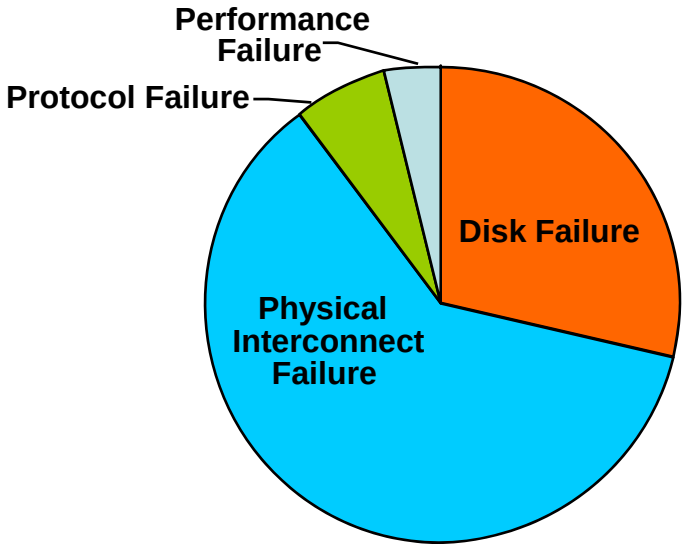
- What kinds of corruption occur and how often ?
- Does disk class matter ?
  - Expensive enterprise (FC) disks versus cheaper nearline (SATA) disks
- Does disk drive family/product matter ?
- Are corruption instances independent ?
- Do corruption instances have spatial locality?

Talk Outline

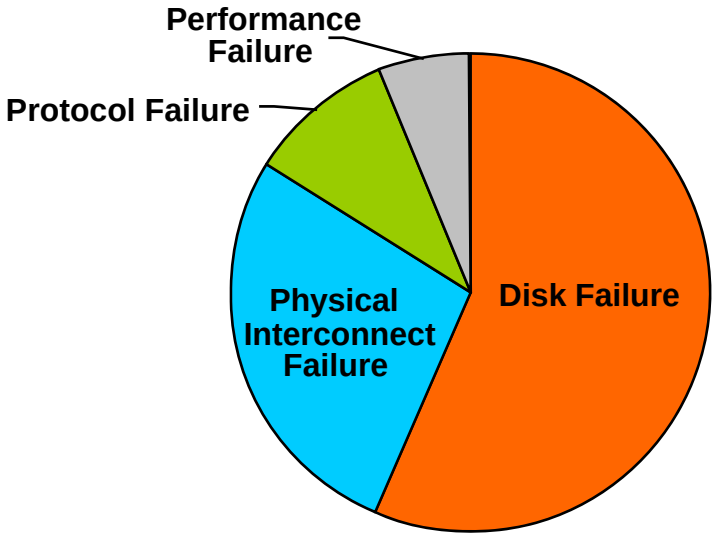
- Introduction
- Background
  - Data corruption
  - Protection techniques
- Results
- Lessons
- Conclusion

Should we care about disk errors?

- Joint UIUC/NetApp system failure analysis
  - 44 months; 39,000 systems; 1.8 million disks



High-End Systems



Nearline Systems

*W. Jiang, et. al, "Are disks the dominant contributor of storage failures?", USENIX FAST, 2008

Disk system failure rates

- From failure rate pie charts:
  - High-end: 29% of system errors are disk errors
  - Nearline: 57% of system errors are disk errors
- What's going on?
  - Software is generally the same
  - Hardware platforms are somewhat different
  - But real difference is in the type of disk in use
    - i.e., Fibre-channel vs SATA

Types of disk errors

- Operational/component failures
  - Fundamental problem with the drive hardware
    - Bad servo, head, electronics, etc.
  - Firmware bugs
    - Failure to flush cache on power-down, etc.
- Partial failures
  - Only affects small subset of disk sectors
  - Errors during writing
    - Bad media, high-fly write, vibration, etc.
  - Errors during reading (write was successful)
    - Scratches, corrosion, thermal asperities, etc.

Unreported Disk Errors

- Operational failures are easy to detect
  - Usually fail-stop; something stops working
- Latent sector errors are reported via SCSI errors
  - Occurs when a disk sector is read
- What about errors that go undetected?
  - Observed errors not corrected by disk's ECC
  - Can not correct them unless detected first
  - Result is usually some form of corruption

Data Corruption

- Data stored on a disk block is incorrect
- Many sources
  - Software bugs
    - File system, software RAID, device drivers, etc.
  - Firmware bugs
    - Disk drives, shelf controllers, adapters, etc.
- Corruption is silent
  - Not reported by the disk drive
  - Could have greater impact than other errors

Forms of Data Corruption

- *Bit corruption*
  - Contents of existing disk block are modified
  - Data being written to a disk block is corrupted
- *Lost writes*
  - Data not written but completion is reported
- *Misdirected writes*
  - Data is written to the wrong disk block
- *Torn writes*
  - Data partially written but completion is reported

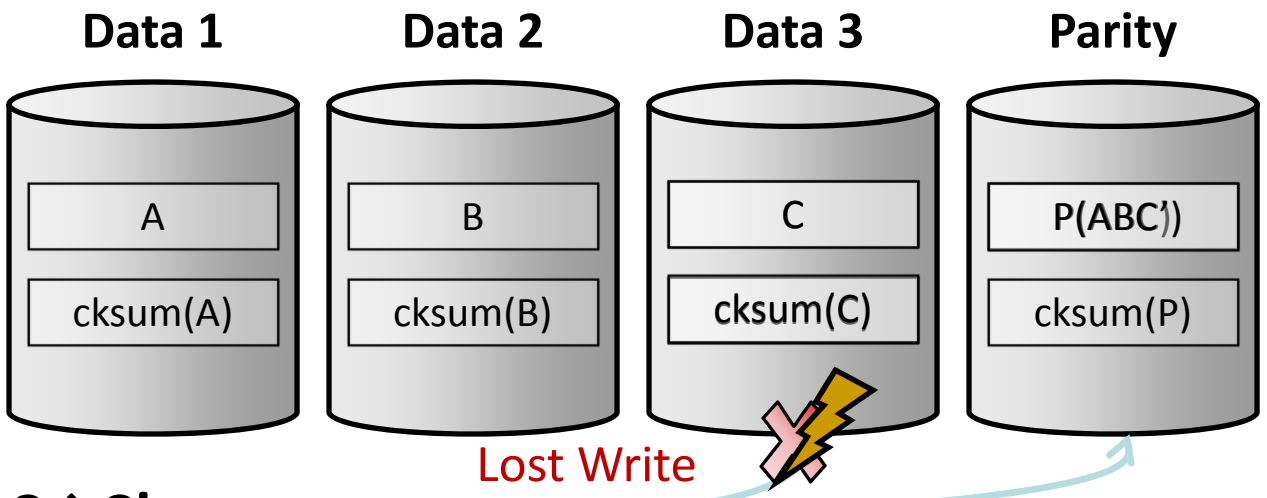
In all cases, data passes disk's internal ECC

Detecting data corruption

- Basic idea:
  1. Generate checksum of data (64 Bytes/4KB)
  2. Store checksum along with data (4KB FS block)
  3. Verify checksum whenever reading data
- Simple checksum has limited protection
  - Detects bit corruption and torn (partial) writes
  - No protection against lost or misdirected writes
    - Since data was not overwritten

Checksum problems: lost writes

- Block checksums



Overwrite C→C'

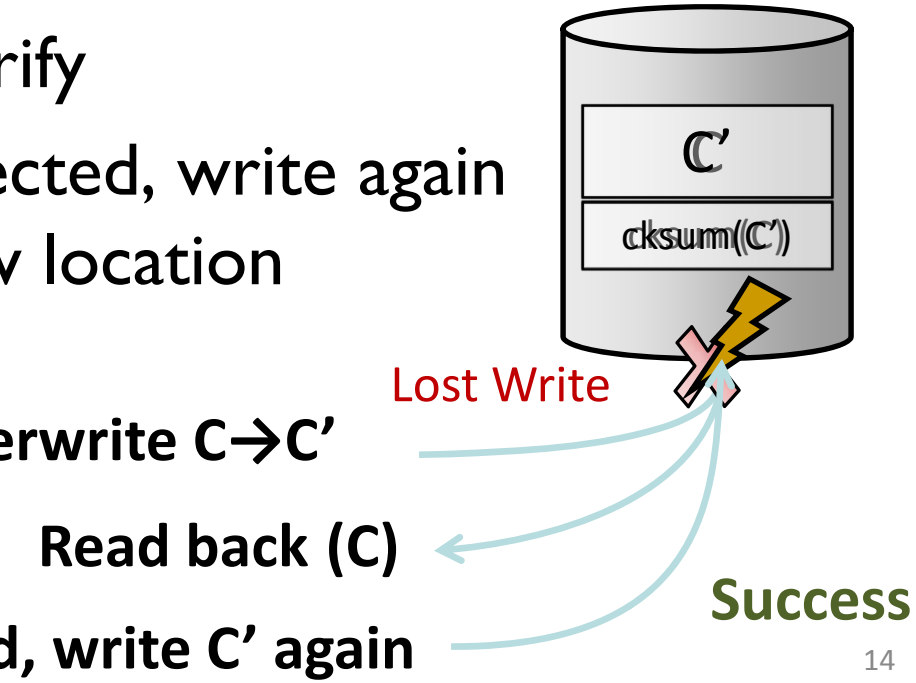
Read file ABC'

CKSUM = **Return data (ABC)**

Return Corrupt Data (C instead of C')

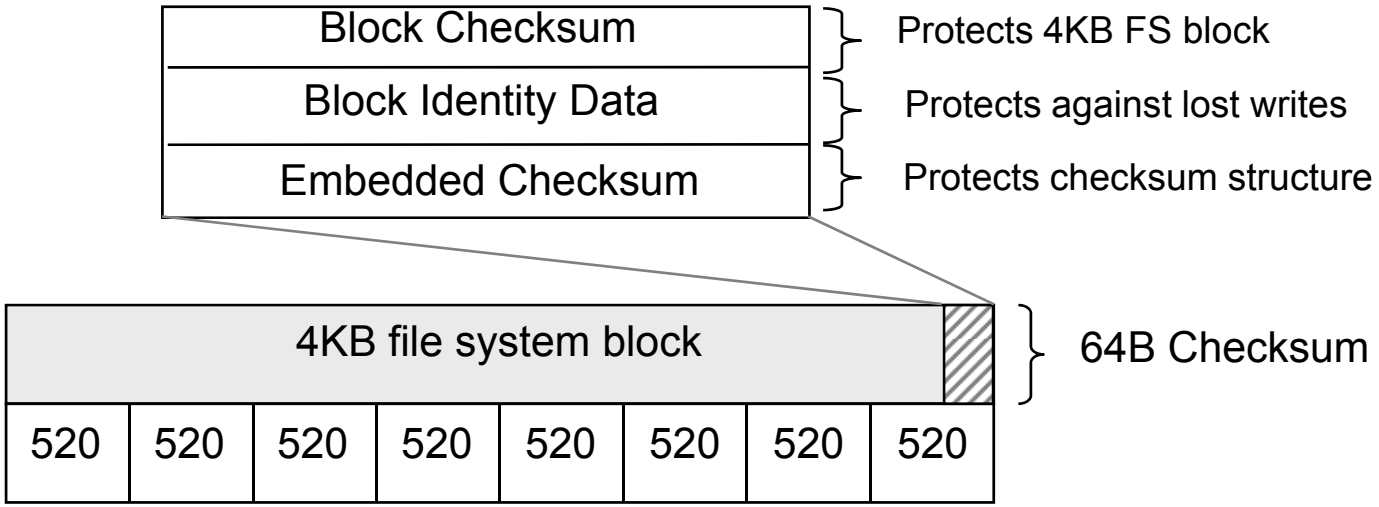
Write verify: a partial solution

- Attempt to solve lost write problem
- Costly solution, expect good protection
- Procedure:
  1. Write data to disk
  2. Read back to verify
  3. If lost write detected, write again or remap to new location



Lost write protection: a better way

- Need logical information pertaining to block identity
  - Something external to data being stored
- Store inode, FS block number within checksum
  - Verified by file system at read time
- We also add a checksum of checksum structure



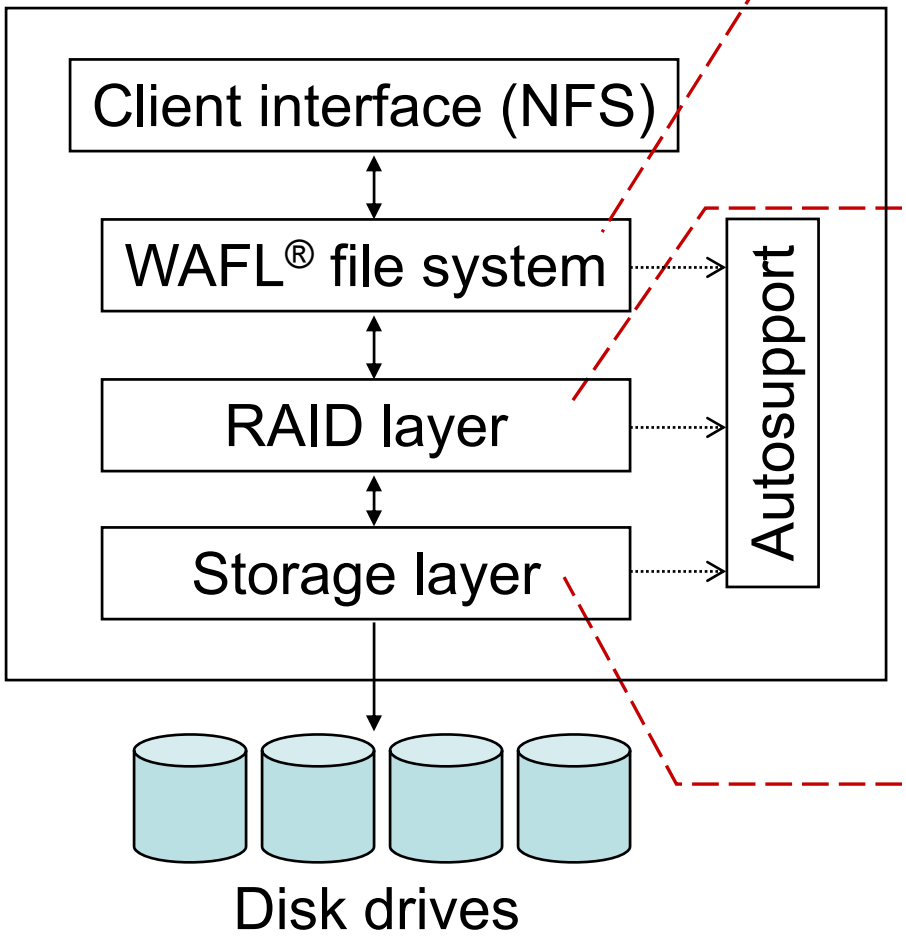
Summary: Data Corruption Classes

- Checksum mismatch
  - Causes: bit corruption, torn/misdirected write
  - Detection: block checksum mismatch
- Identity mismatch
  - Causes: lost or misdirected write
  - Detection: block identity mismatch
- Parity mismatch
  - Causes: lost write, bad parity
  - Detection: RAID parity computation mismatch

Talk Outline

- Introduction
- Background
- Results
  - System architecture
  - Overall results
  - Checksum mismatch results
- Lessons and Conclusion

NetApp® System



- Store, verify block identity (Inode X, offset Y)
- Detect identity discrepancy
- Lost or misdirected writes

3

- Parity generation
- Reconstruction on failure
- Data scrubbing
  - read blocks, verify parity
  - Detect parity inconsistency
  - Lost or misdirected writes, parity miscalculations

2

- Store, verify checksum
- Detect checksum mismatch
- Bit corruptions, torn writes

1

Overall Numbers

What percentage of disks are affected by the different kinds of corruption?

Overall Numbers

(% disks affected in 17 months of use)

	Corruption type	Nearline (SATA)	Enterprise (FC)
1	Checksum mismatches	0.661%	0.059%
2	Parity inconsistencies	0.147%	0.017%
3	Identity discrepancies	0.042%	0.006%

- ~10 times fewer disks than latent sector errors
- Higher % of Nearline disks affected
  - Order of magnitude more than enterprise disks
- Bit corruptions or torn writes affect more disks than lost or misdirected writes

Checksum Mismatch (CM) Analysis

1. Factors
2. Characteristics
3. Correlations with other errors
4. Request type

- Disk class (Nearline / Enterprise)
- Disk model
- Disk age
- Disk size (capacity)
- Workload

- CMs per corrupt disk
- Independence
- Spatial locality
- Temporal locality

- Not ready conditions
- Latent sector errors
- System reset

- Scrubs vs. FS reads etc.

Checksum Mismatch (CM) Analysis

1. Factors
2. Characteristics
3. Correlations with other errors
4. Request type

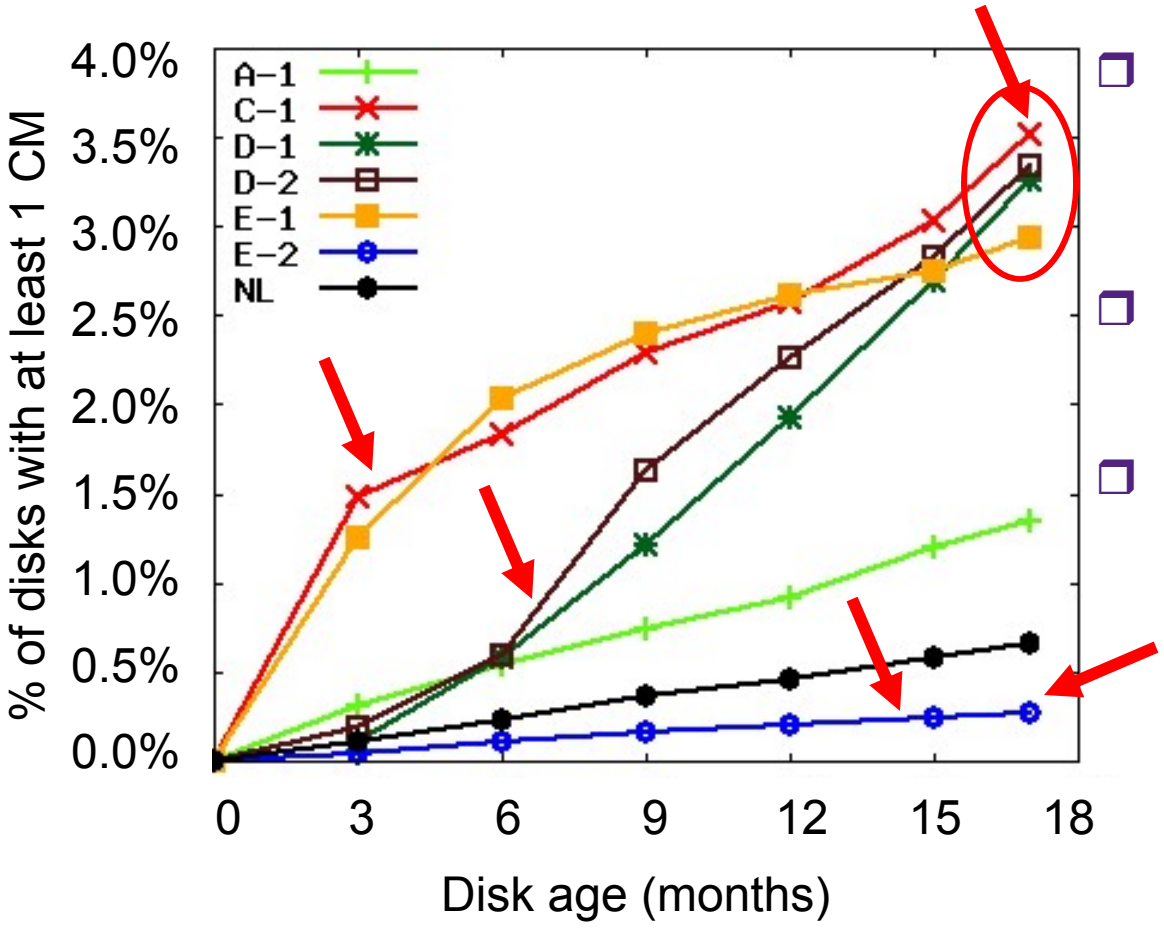
- Disk class
- Disk model
- Disk age
- Disk size
- Workload

Factors

- Do disk class, model, or age affect development of checksum mismatches?
  - Disk class: Nearline (SATA) or Enterprise (FC)
  - Disk model: Specific disk drive product
    (say Vendor V's disk product P of capacity 80 GB)
  - Disk age: Time in the field since ship date

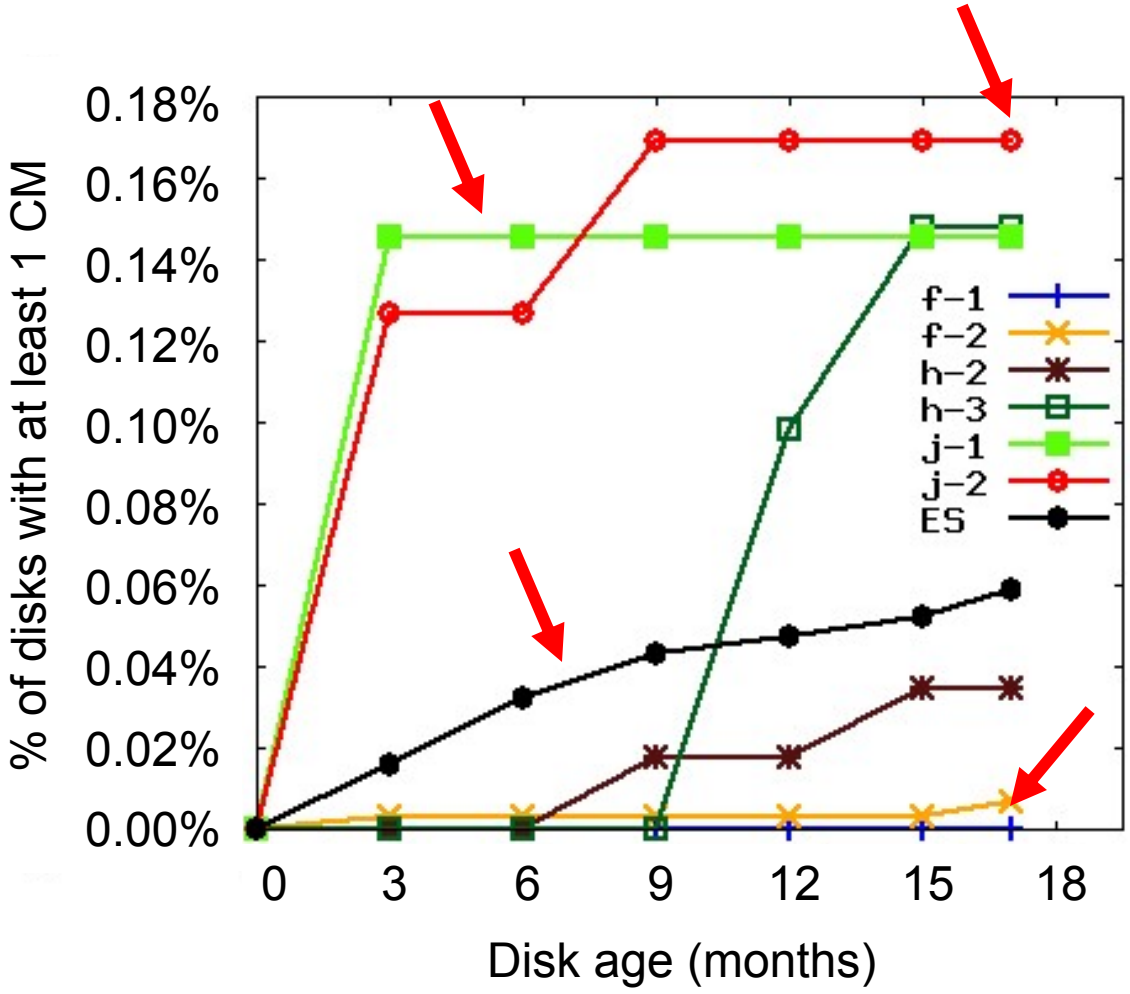
- Can we use these factors to determine corruption handling policies or mechanisms?
  - Ex: Aggressive scrubbing for some disks

Class, Model, Age – Nearline



- Fraction of disks affected varies across models
  - From 0.27% to 3.51%
- More than 3%
  - 4 out of 6 models
- Response to age also varies

Class, Model, Age – Enterprise



- Fraction of disks affected varies across models
  - From 0% to 0.17%
  - All less than lowest Nearline (0.27%)
- Response to age also varies

Factors – Summary

- Class, Model matter
  - Nearline disks require greater attention
- Effect of age is unclear
  - Cannot use age-specific corruption handling

Checksum Mismatch (CM) Analysis

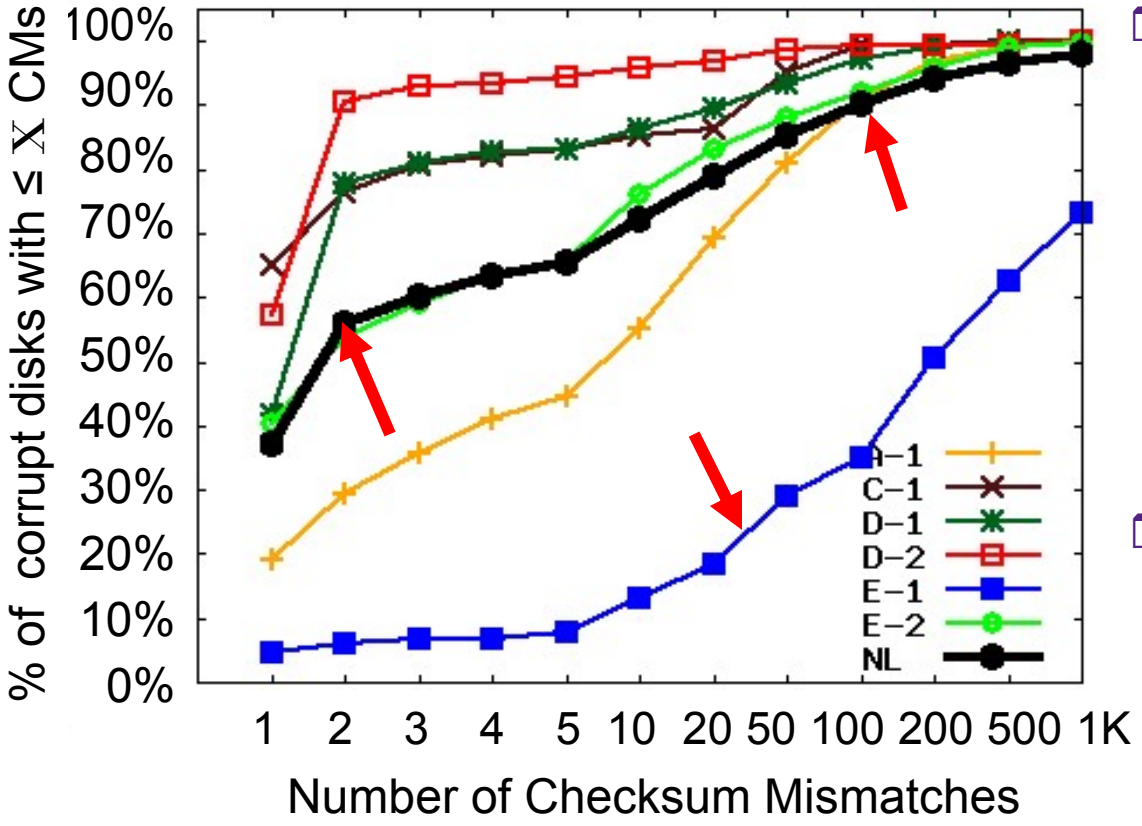
1. Factors
2. Characteristics
3. Correlations with other errors
4. Request type

- CMs per corrupt disk
- Independence
- Spatial locality
- Temporal locality

Checksum Mismatches per Corrupt Disk

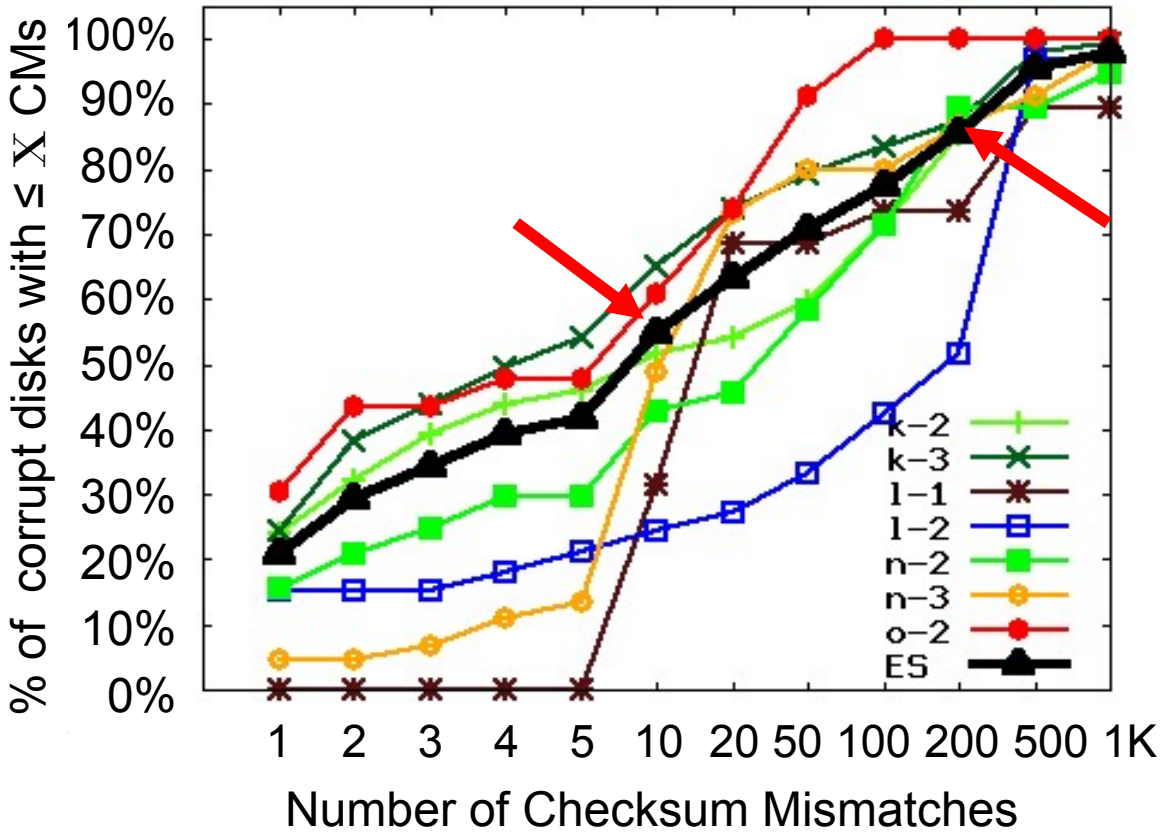
- Corrupt disk: A disk with at least 1 checksum mismatch (CM)
- How many CMs does a corrupt disk have?
- Should we "fail-out" disks when one corruption is detected?

CMs per Corrupt Disk – Nearline



- CMs per corrupt disk is low
  - 50% of corrupt disks have ≤ 2 CMs
  - 90% of corrupt disks have ≤ 100 CMs
- Anomaly: E-1
  - Develops many CMs

CMs per Corrupt Disk – Enterprise



CMs per corrupt disk higher

- 50% of corrupt disks have ≤ 10 CMs (2 for Nearline)
- 90% of corrupt disks have ≤ 200 CMs (100 for Nearline)

CMs per Corrupt Disk – Summary

- Class and model matter
- Fewer enterprise disks have CMs, but corrupt disks have more CMs
  - Fail-out enterprise disks on first CM
- Corrupt nearline disks develop fewer CMs
  - There can be anomalies (Disk model E-1)

Other Characteristics

- Very high spatial locality
  - When multiple checksum mismatches occur, they are often for *consecutive* disk blocks
- High temporal locality
- Not independent
  - Over *different* disks in *same* system
  - Defect may be in common hardware components (Example: shelf controller)

Checksum Mismatch (CM) Analysis

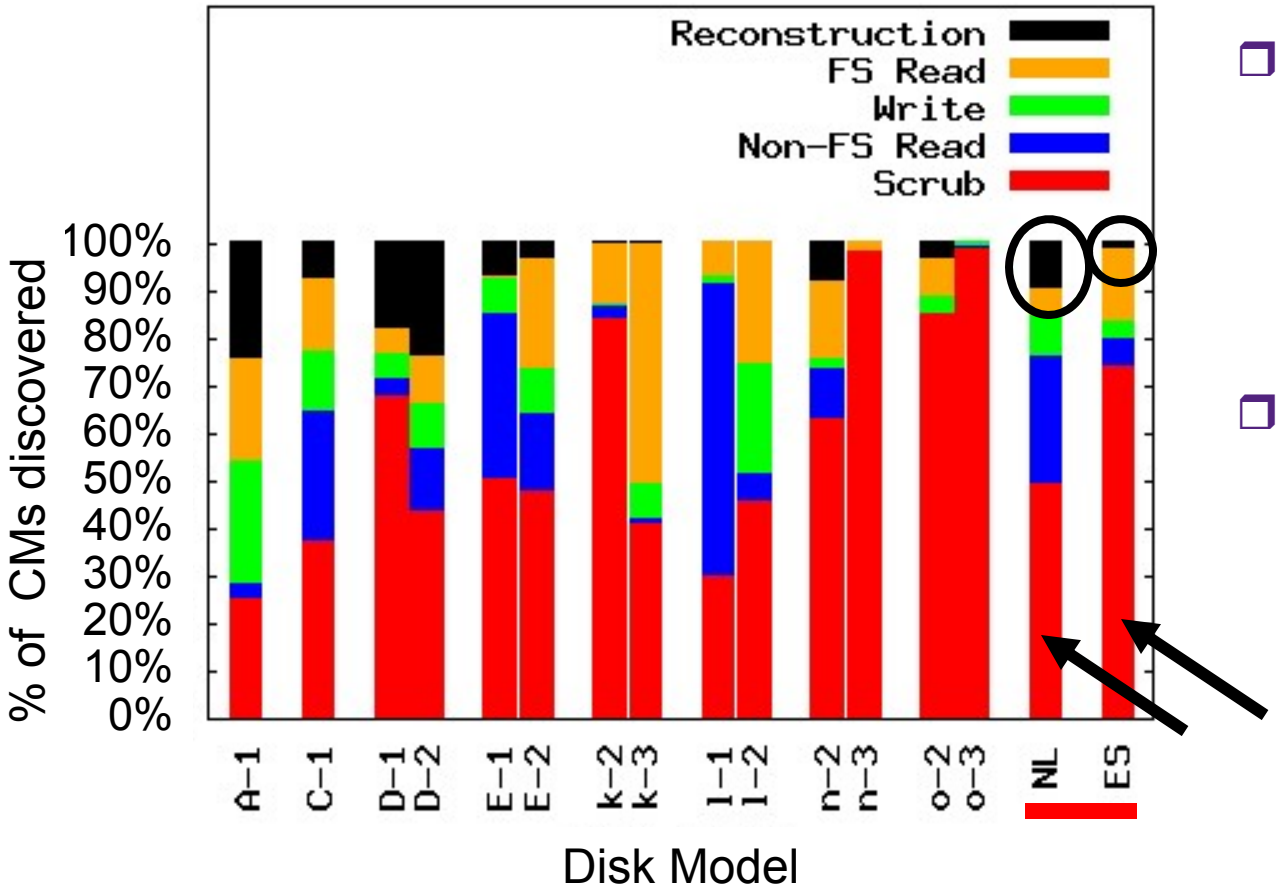
1. Factors
2. Characteristics
3. Correlations with other errors
4. Request type

- Scrubs vs. FS reads etc.

Request Type

- What types of disk requests detect checksum mismatches?
- Is data scrubbing useful?

Request Type



- Data scrubbing finds most CMs
  - Nearline: 49%
  - Enterprise: 73%
- Reconstruction finds CMs
  - Nearline: 9%
  - Enterprise: 4%

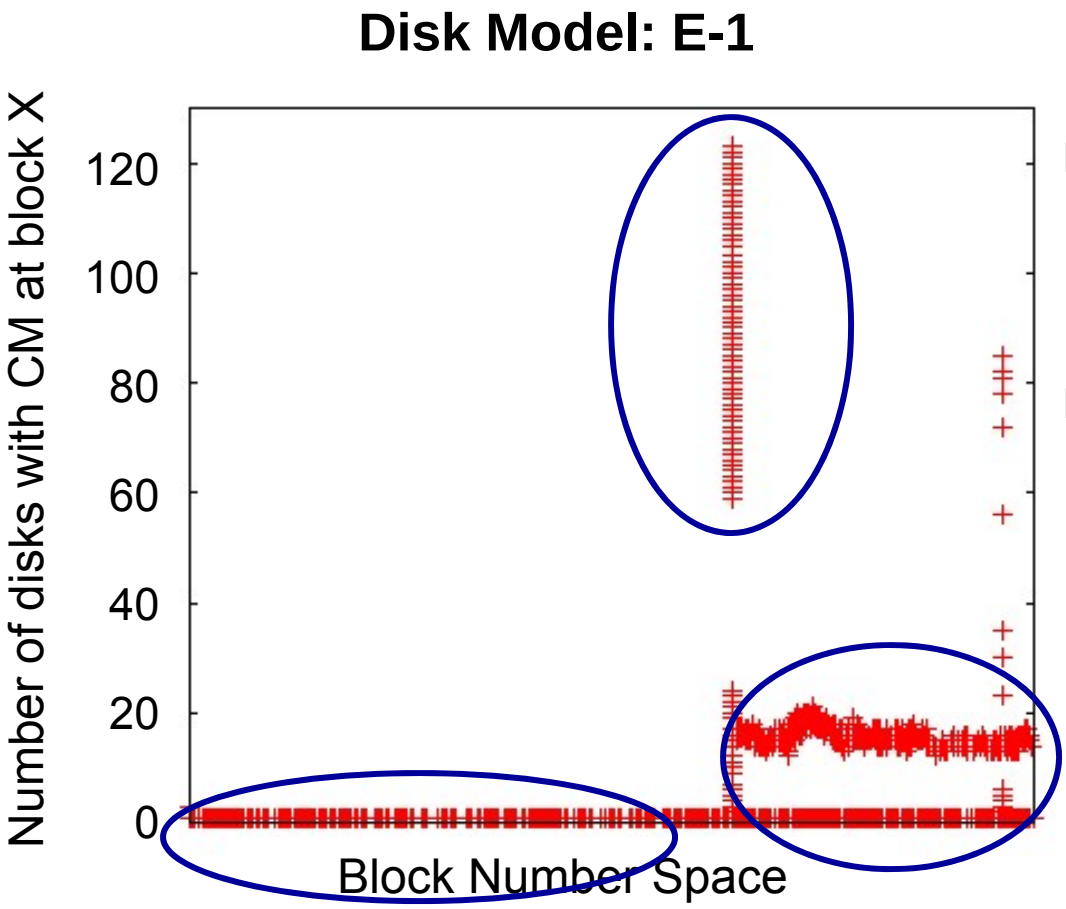
Request Type – Summary

- Data scrubbing appears to be very useful
  - Study of scrub rates, workload needed
- Mismatches found during reconstruction
  - Data loss without double disk failure protection [Alvarez97, Blaum94, Corbett04, Park95, Hafner05]
  - More aggressive scrubbing may be needed

Interesting Behavior

Do system designers need to factor in any abnormal behavior?

Block numbers are not created equal!



- Typically, each block number has 1 disk where it is corrupt
- A series of block numbers are corrupt in many disks
  - A block-number specific bug?

Talk Outline

- Introduction
- Background
- Results
- Lessons
- Conclusion

Lessons

- Data corruption does occur
  - Even rare errors like lost writes do occur
  - Corruption handling mechanisms are essential
- Very few enterprise disks develop corruption
  - "Fail-out" these disks on first corruption detection
- High spatial locality
  - Spread out redundant data within the same disk

Lessons (contd.)

- Temporal locality, consecutive blocks affected
  - May be corruption occurs during the same write op
  - Write redundant data with separate disk requests, spaced out over time

Conclusion

- Our analysis
  - First large scale study of data corruption
  - Corruptions detected by NetApp production systems
- Data corruptions do occur
  - Affect ~10 times fewer disks than latent sector errors
  - Nearline (SATA) disks are most affected
  - Corruption handling mechanisms are essential
- Data corruption characteristics
  - Depend on disk class and disk model
  - Not independent (both within disk and within system)
  - High spatial and temporal locality
  - May occur at specific block numbers

Thank You!

Advanced Technology Group (ATG)
NetApp, Inc
http://www.netapp.com/company/research/

Advanced Systems Lab (ADSL)
University of Wisconsin-Madison
http://www.cs.wisc.edu/adsl

Department of Computer Science
University of Toronto
http://www.cs.toronto.edu/~bianca