

## Case studies from the real world: The importance of measurement and analysis in system building and design

Bianca Schroeder  
University of Toronto



## Some background

- Main interest: system reliability
- Why and how do systems fail in the wild?



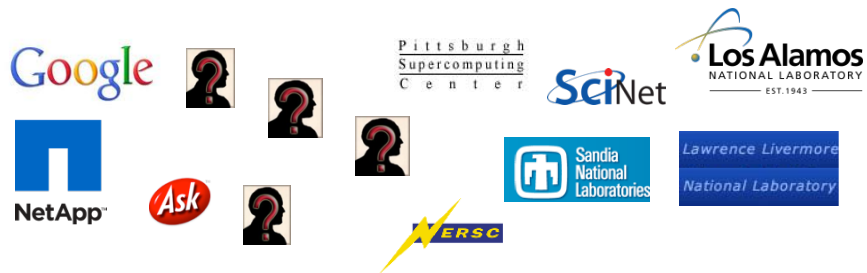
# Publicly available information

The collage features three main components:



- Seagate Advertisement:** Promotes USB 2.0 Portable External Hard Drives in various capacities (40GB to 160GB) with a headline: "Colorado man ticks shooting his computer was worth it".
- The Telegraph Article:** Titled "Computer rages", it reports that "More than half of Britons have experienced a computer rage" and "led as 'computer rage' in their computers, a new study has found".
- Toshiba Product Page:** Details the "MK1011GAH 100GB 1.8-inch HDD" with specifications and pricing.

# Field data

Data from a large number of large-scale production systems at different organizations:



## Field data

- Different hardware failure events
    - Hardware replacements
    - Correctable and uncorrectable errors in DRAM
    - Server outages
    - Hard disk drive failures
    - Sector errors in hard disk drives
    - Data corruption in storage systems
    - Failures in solid state drives 
  - Job logs
    - Google, OpenCloud (Hadoop cluster at CMU), Yahoo! Hadoop trace 
- Observations often **different from expectations**
  - Surprising to operators as well as manufacturers

## Field data

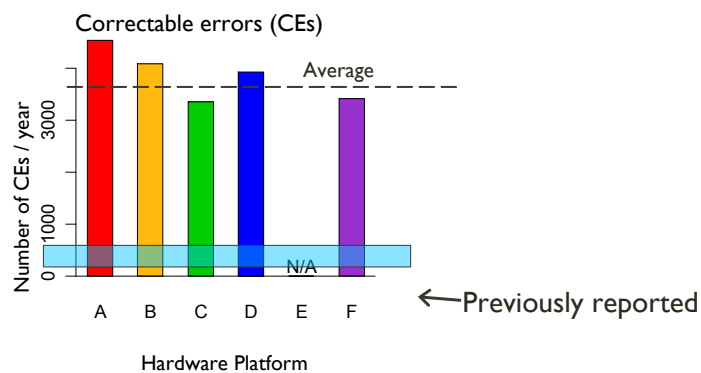
- Different hardware failure events
    - Hardware replacements
    - Correctable and uncorrectable errors in DRAM
    - Server outages
    - Hard disk drive failures
    - Sector errors in hard disk drives
    - Data corruption in storage systems
    - Failures in solid state drives
  - Job logs
    - Google, OpenCloud (Hadoop cluster at CMU), Yahoo! Hadoop trace
- Observations often **different from expectations**
  - Surprising to operators as well as manufacturers

## Errors in DRAM

- Why DRAM errors?
  - DRAM is one of the most frequently replaced H/W components
- What are DRAM errors?
  - Cell has different value from what was written to it
  - Can be correctable (using ECC) or uncorrectable
- How do they happen?
  - Soft errors:
    - Cosmic rays, alpha particles, leakage, random noise
    - Transient, not repeatable
  - Hard errors:
    - Permanent hardware problem, repeatable

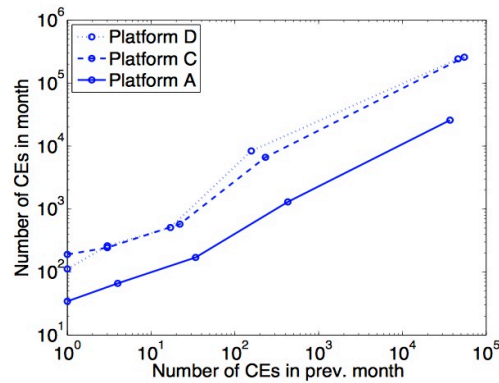
7

## How common are DRAM errors?



- 34% of machines had correctable errors, 1.3% uncorrectable errors
- Much higher frequency than previously reported
  - Why?

## After an error, more are likely to follow...



- Not consistent with soft errors ....

## Error patterns in DRAM dimms

Error Mode	BG/L Banks	BG/P Banks	Google Banks
Repeat address	80.9%	59.4%	58.7%
Repeat row	4.7%	31.8%	7.4%
Repeat column	8.8%	22.7%	14.5%
Whole chip	0.52%	2.20%	2.02%
Single Event	17.6%	29.2%	34.9%

- The patterns on the majority of banks can be linked to **hard errors**.
- Different error mode than commonly assumed!

## Many errors are hard errors – so what?

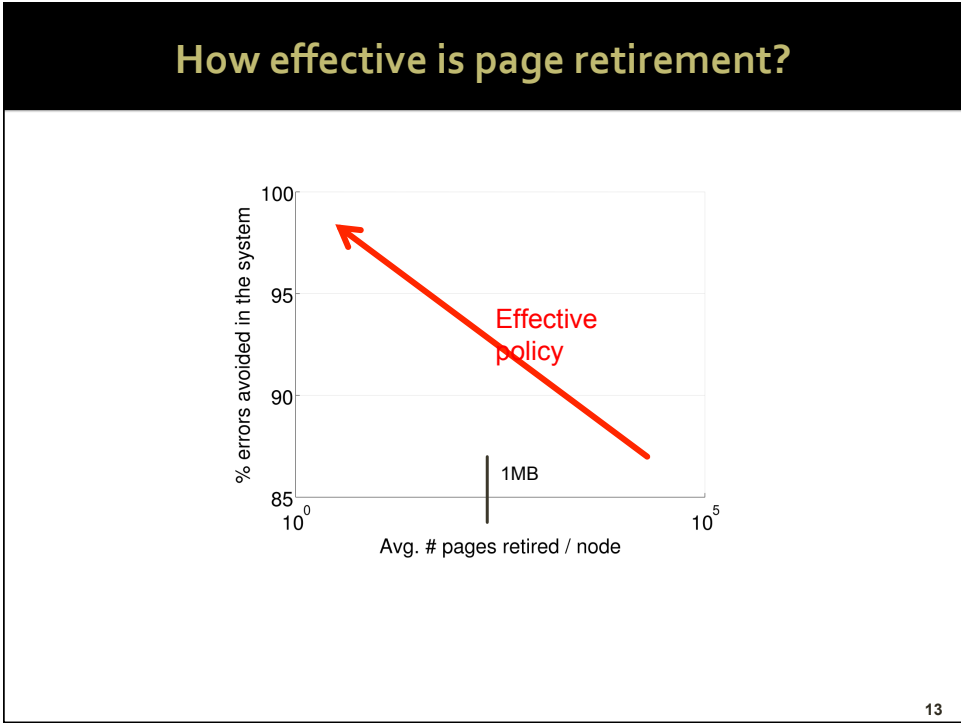
- How to protect against errors?
  - Most commonly: only ECC
- For hard errors: Page retirement
  - Move page's content to different page and discontinue use
- Some page retirement mechanisms exist
  - Solaris
  - BadRAM patch for Linux
  - But:
    - Rarely used in practice
    - No existing evaluation of policies on real traces

11

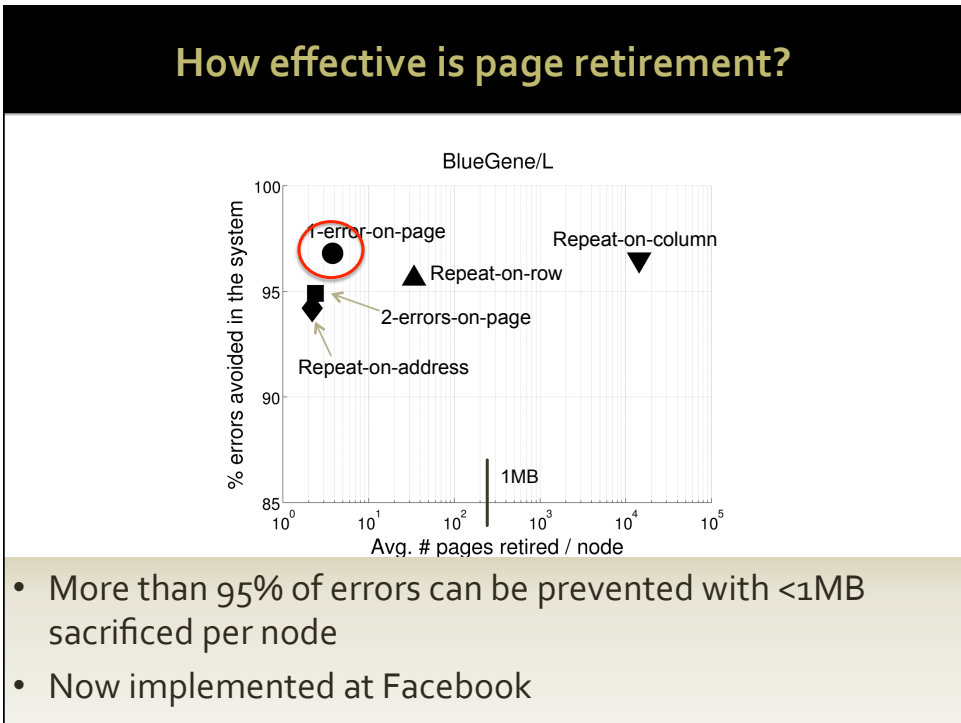
## Policies for retiring pages

- First error on page => retire page
- Second error on page => retire page
- Repeat error on address => retire page
- Repeat error on row => retire row
- Repeat error on column => retire column

12



13




## DRAM reliability – key points

- DRAM errors occur at significant rate
- Often different from common assumptions
  - Hard errors rather than soft errors
    - => Can effectively protect with page retirement
  - Some parts of address space (kernel space) more error prone
    - => Special protection for kernel space
  - Little sensitivity to temperature
    - => adapt cooling policies

15

## Field data

- Different hardware failure events
  - Hardware replacements
  - Correctable and uncorrectable errors in DRAM
  - Server outages
  - Hard disk drive failures
  - Sector errors in hard disk drives
  - Data corruption in storage systems
  - Failures in solid state drives 
- Job logs
  - Google, OpenCloud (Hadoop cluster at CMU), Yahoo! Hadoop trace

16

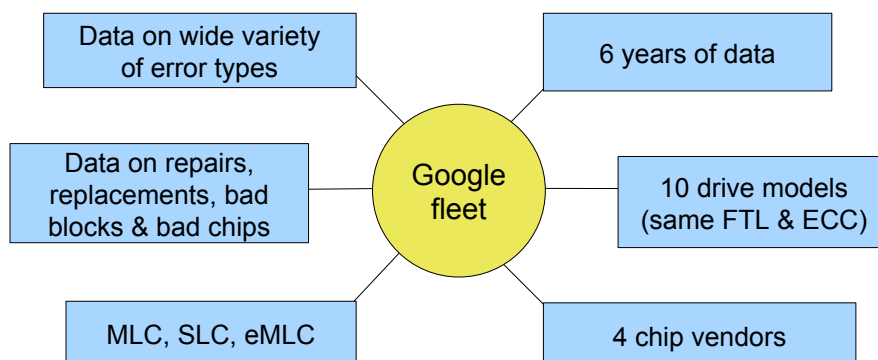


## Flash reliability

- Why flash?
  - More and more data is living on flash
    - => data reliability depends on flash reliability
  - Worry about flash wear-out
- Little prior work on *production systems*
  - Lab studies using accelerated testing
  - Only one field study (Sigmetrics'15)

17

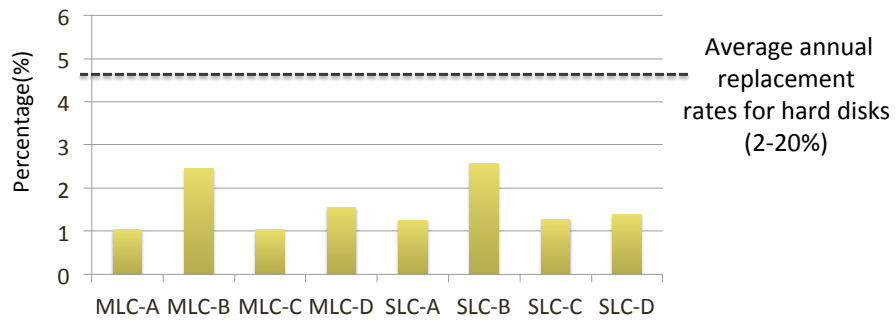
## The data



18

## Drive replacements

- Percentage of drives replaced annually due to suspected hardware problems over the first 4 years in the field:

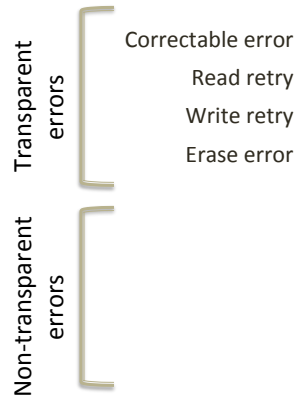


- ~1-2% of drives replaced annually, much lower than hard disks!
- 0.5-1.5% of drives developed bad chips per year
  - Would have been replaced without methods for tolerating chip failure

## Errors experienced during a drive's lifecycle

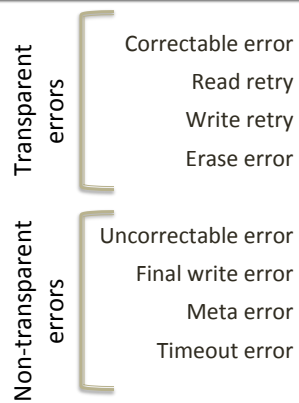


## Errors experienced during a drive's lifecycle



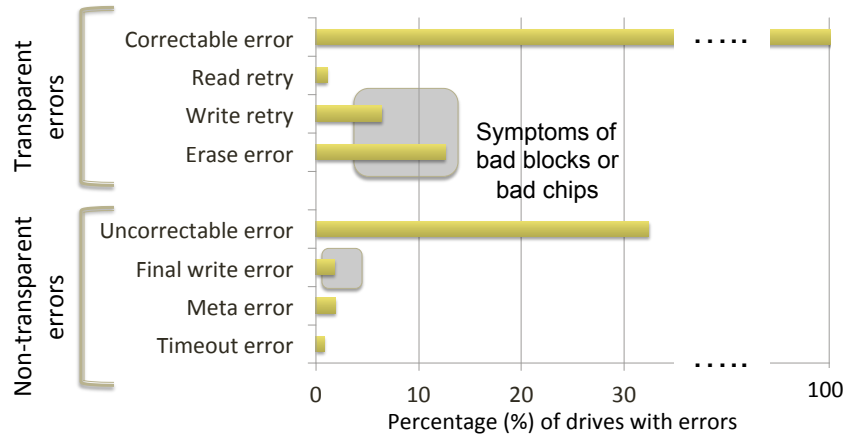
21

## Errors experienced during a drive's lifecycle



22

## Errors experienced during a drive's lifecycle



### Non-transparent errors common:

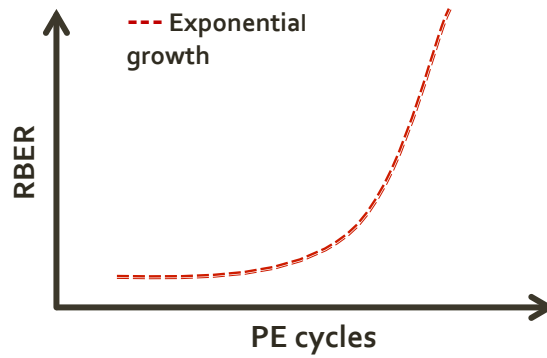
- 26-60% of drives with uncorrectable errors
- 2-6 out of 1,000 drive days experience uncorrectable errors
- Much worse than for hard disk drives (3.5% experiencing sector errors)!

## What factors impact flash reliability?

- Wear-out (limited program erase cycles)
- Technology (MLC, SLC)
- Lithography
- Age
- Workload
  
- What reliability metric to use?
  - Raw bit error rate (RBER)
  - Probability of **uncorrectable** errors
    - Why not UBER? We shall see ...

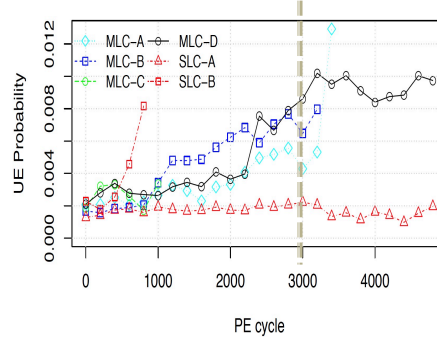
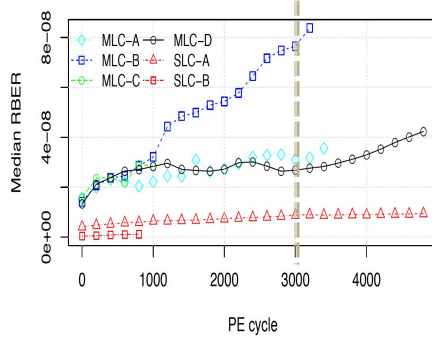
## Effect of wear-out (program erase cycles)

Common expectation:  
Exponential increase of RBER with PE cycles



25

## Effect of wear-out (program erase cycles)



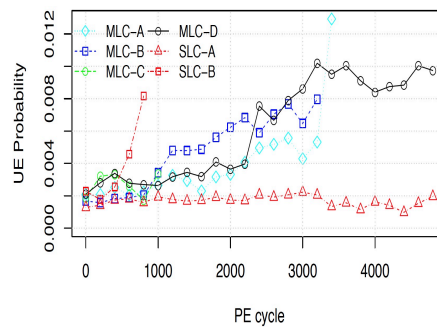
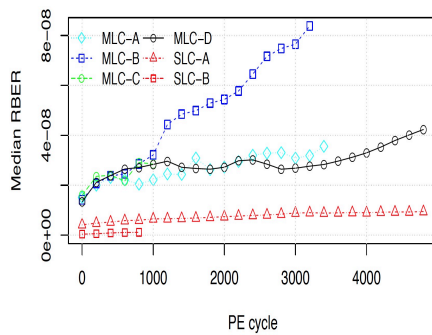
- Big differences across models (despite same ECC)
- Linear rather than exponential increase
- No sudden increase after PE cycle limit

## Effect of type of flash (SLC versus MLC)

Common expectation:  
Lower error rates under SLC (\$\$\$) than MLC

27

## Effect of type of flash (SLC versus MLC)



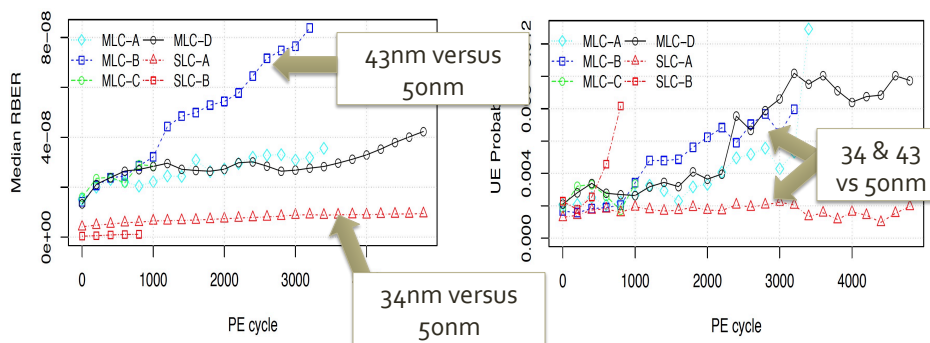
- RBER is lower for SLC drives than MLC drives
- Uncorrectable errors are not consistently lower for SLC drives
- SLC drives don't have lower rate of repairs or replacement

## Effect of lithography

Common expectation:  
Higher error rates for smaller feature size

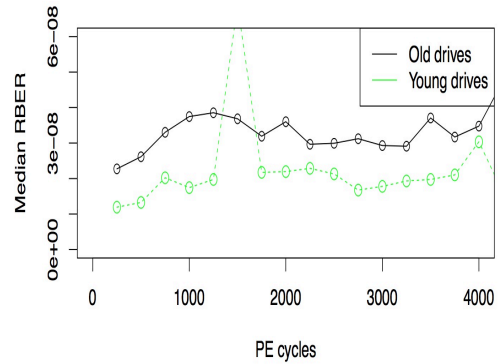
29

## Effect of lithography



- Smaller lithography => higher RBER
- Lithography has no clear impact on uncorrectable errors

## Effect of age (time in production)?



- Age has an effect beyond PE-cycle induced wear-out

## Effect of workload?

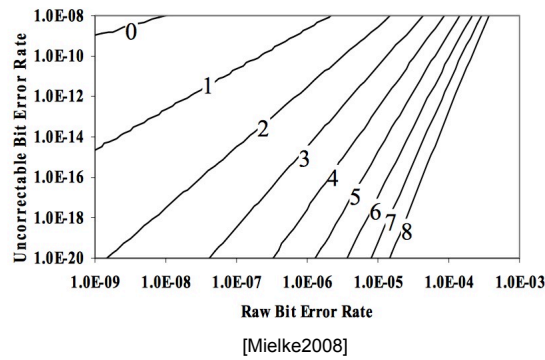
- Lab studies demonstrate workload induced error modes
  - Read disturb errors
  - Program disturb errors
  - Incomplete erase operations

- Evidence of read disturb affecting RBER for some models
  - No effect of erases and writes
- Workload does not affect uncorrectable errors
  - UBER (uncorrectable bit error rate) is not a meaningful metric



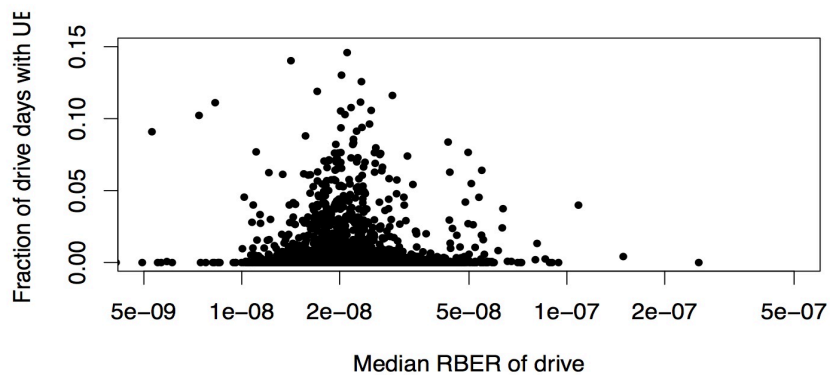
## RBER and overall reliability

- The main purpose of RBER is as a metric for overall drive reliability
- Allows for projections on uncorrectable errors



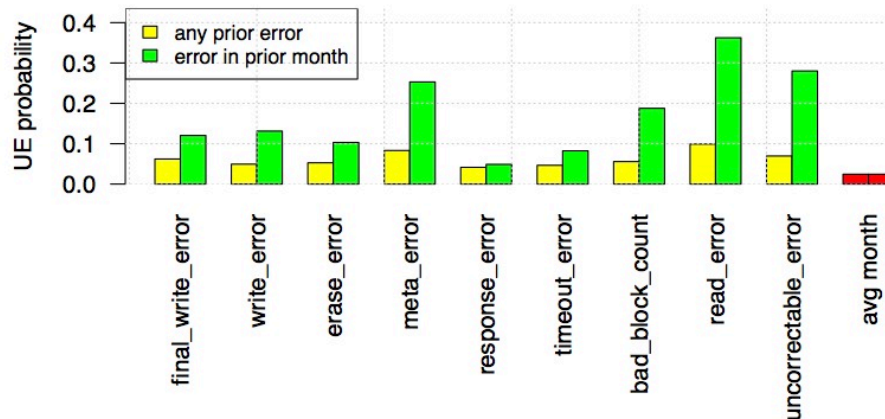
33

## RBER and uncorrectable errors



- Drives (or drive days) with higher RBER don't have higher frequency of uncorrectable errors
- **RBER is not a good predictor of field reliability**
- Uncorrectable errors caused by other mechanisms than corr. errors?

## What is predictive of uncorrectable errors?



- Prior errors highly predictive of later uncorrectable errors
- Potential for prediction?
  - Initial results say yes!

## Flash reliability – key points

- Significant rate of non-transparent errors
  - Higher than hard disk drives
  - Need to protect against those!
  - To some degree predictable
    - Work in progress on how to use predictions
- Many aspects different from expectations
  - Linear rather than exponential increase with PE cycles
  - RBER not predictive of non-transparent errors
  - SLC not generally more reliable than MLC
- Many other results not covered in talk ...
  - Bad chips, bad blocks, factory bad blocks, rate of repair and replacement, comparison of projections with field RBER, ...

36

## FIELD DATA

- Different hardware failure events

- Hardware replacements
- Correctable and uncorrectable errors in DRAM
- Server outages
- Hard disk drive failures
- Sector errors in hard disk drives
- Data corruption in storage systems
- Failures in solid state drives

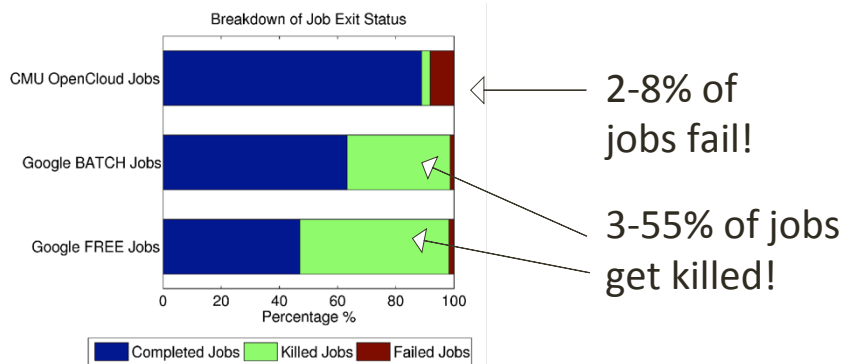
- Job logs

- Google, OpenCloud (Hadoop cluster at CMU), Yahoo! Hadoop trace

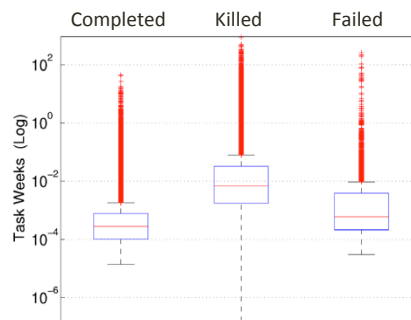


37

## Exit status of jobs?



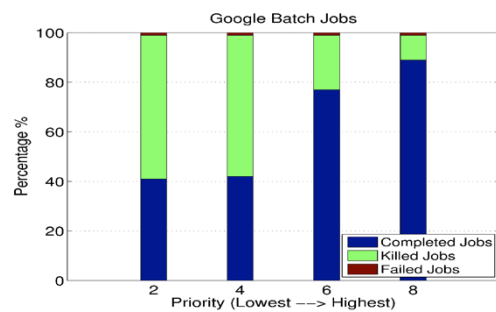
## Impact of job length



Long jobs more likely to fail or get killed.  
More parallelism => more likely to fail or get killed

## What brings jobs down?

- Node failure?
  - Small fraction of failed/killed jobs suffered evictions
- Resource usage (memory) exceeds requested resources?
  - Happens very rarely
- Preemption by higher priority jobs?

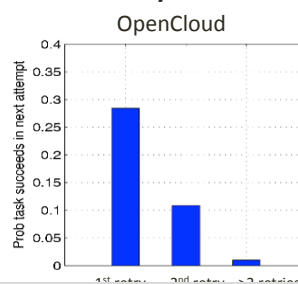
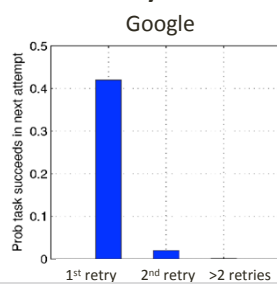


## What brings jobs down?

- Node failure?
  - Small fraction of failed/killed jobs suffered evictions
- Resource usage (memory) exceeds requested resources?
  - Happens very rarely
- Preemption by higher priority jobs?
  - Production jobs and prio-8 jobs still see 15% killed
- Task failure?
  - < 4% of Google jobs with a failed task complete
  - < 60% of CMU jobs with a failed task complete

## Can individual tasks recover from failure?

- Recovery mechanisms: task retry



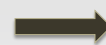
Retrying more than once or twice is futile!

Users are (too) optimistic!

.70-90% retry more than once

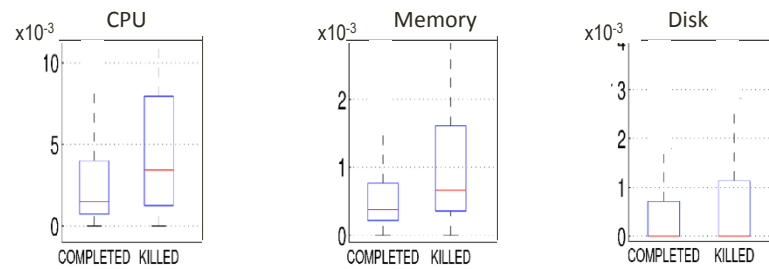
.15-30% retry more than twice

.Some retry > 100 times

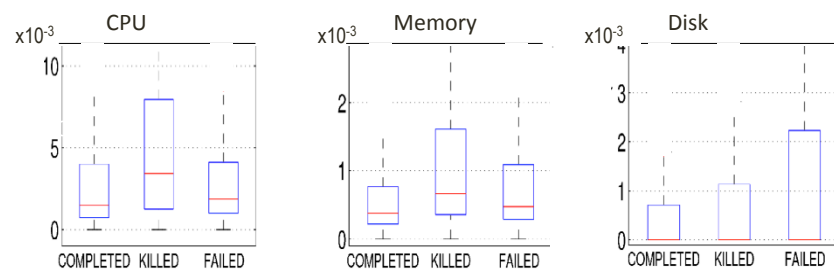


Waste of resources!

## Is resource usage different for failed/killed tasks?



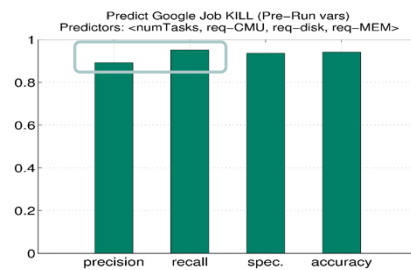
## Is resource usage different for failed/killed tasks?



Failed and killed jobs use more resources.  
They also requested more resources.

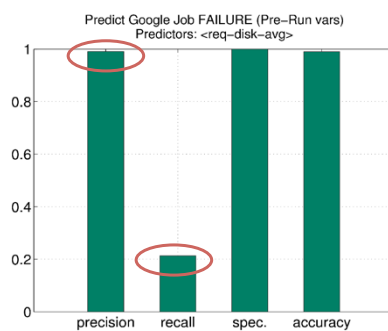
## Can we predict whether a job will get killed?

- Using only information available at start time



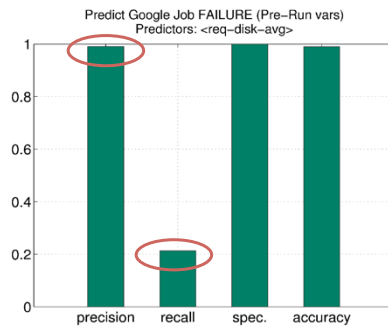
Can predict whether a job will get killed with high precision and recall (before it even runs).

## Can we predict whether a job will fail?



Can predict job failure with high precision, but lower recall...

## Can we predict whether a job will fail?

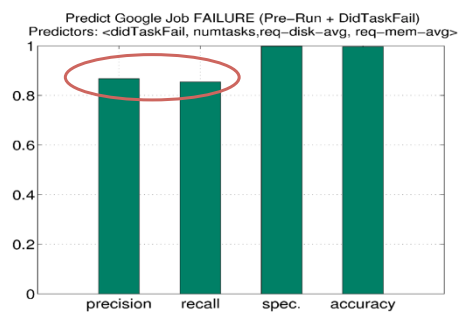


Can predict job failure with high precision,  
but lower recall...

Adding resource usage improves recall a bit (~30%)

## Can we predict whether a job will fail?

- Pre-run information + one task failure



Using information on task failure brings  
recall up to 85%.



## Can we predict whether a task will fail?

- Pre-run information + online monitoring of resource usage



83% precision and 98% recall using online resource monitoring.

## Key points: job log analysis

- Surprisingly large fraction of jobs fails or gets killed
- Patterns: e.g. resource hungry jobs more likely to die
- Failed tasks have low chances of recovering
- Strong potential to predict job failures / killings
- Work in progress on how to use predictions!

## Talk conclusion

- Failures in the real world often very different from common assumptions or observations in the lab
  - Both for hardware and software failures
- Results from field data help in building more resilient systems
  - E.g. often potential for prediction
  - Importance of measuring & analyzing systems