LARGE SCALE STUDIES OF MEMORY, STORAGE, AND NETWORK FAILURES IN A MODERN DATA CENTER

THESIS ORAL

JUSTIN MEZA

Committee
Prof. Onur Mutlu (Chair)
Prof. Greg Ganger
Prof. James Hoe
Dr. Kaushik Veeraraghavan (Facebook, Inc.)
100's SOFTWARE SYSTEMS

[Hahn LISA'18]
1,000,000's CONTAINERS

[Hahn LISA'18]
WANT HIGH RELIABILITY
PROBLEM

Device failures disrupt data center workloads
1. INTERDEPENDENCE
2. DISTRIBUTION
3. COMMODITY HW
The programs running in modern data centers make up larger workloads.
The programs running in modern data centers make up larger workloads.
The programs running in modern data centers make up larger workloads.
The programs running in modern data centers make up larger workloads.
Workloads in modern data centers are distributed across many servers.
**PROBLEM 2: DISTRIBUTION**

Workloads in modern data centers are distributed across many servers.
Workloads in modern data centers are distributed across many servers.
PROBLEM 3: COMMODITY HW

Modern data centers trade off reliability for using simpler, commodity hardware.
PROBLEM 3: COMMODITY HW

Modern data centers trade off reliability for using simpler, commodity hardware.
Modern data centers trade off reliability for using simpler, commodity hardware.
Even a single device failure can have a widespread effect on the workloads running in modern data centers.
Here's why Azure's South Central US data center went down earlier this month. Low Microsoft got it online back online.

Visa details cause of widespread outage, blames data center switch.

GitHub suffers major outage was due to storage appliance failure.

Amazon websites outage was due to hardware failure.

Where is your Octocat now?
“A fail-slow hardware can collapse the entire cluster performance; for example, a degraded NIC made many jobs lock task slots/containers in healthy machines, hence new jobs cannot find enough free slots.”
GOAL

Measure, model, and learn from device failures to improve data center reliability
CHALLENGES

1. Most device reliability studies are small scale
2. Prior large scale studies hard to generalize
3. Limited evaluation of techniques in the wild
If we measure the device failures in modern data centers, then we can learn the reasons why devices fail, develop models to predict device failures, and learn from failure trends to make recommendations to enable workloads to tolerate device failures.
MEASURE  MODEL  EVALUATE
CONTRIBUTIONS

1. Large scale failure studies

DRAM
[DSN '15]

SSDs
[SIGMETRICS '15]

Networks
[IMC '18]

We shed new light on device trends from the field
CONTRIBUTIONS

2. Statistical failure models

DRAM
[DSN '15]

SSDs
[SIGMETRICS '15]

Networks
[IMC '18]

We enable the community to apply what we learn
CONTRIBUTIONS

3. Evaluate best practices in the field

**DRAM**
Page offlining

**SSDs**
OS write buffering

**Networks**
Software-based networks

We provide insight into how to tolerate failures
1. Modern data center background
2. Large scale device failure studies
   • Memory: DRAM
   • Storage: SSDs
   • Network: Switches and WAN
3. Conclusion
Core Switches  

Data Center Fabric  

Top of Rack Switch
MEMORY
Dynamic Random Access Memory (DRAM)
STORAGE
Solid State Drives (SSDs)
NETWORK

Switches and Wide Area Network (WAN) Backbone
WHY DO DEVICES FAIL?

DRAM
- Retention
- Disturbance
- Endurance

SSDs
- Endurance
- Disturbance
- Temperature

Networks
- Bugs
- Faulty hardware
- Human error
DATA CENTER DIVERSITY

• **Different system configurations**
  - Diverse workloads (Web, Database, Cache, Media)
  - Diverse CPU/memory/storage requirements

• **Different device organizations**
  - Capacity, frequency, vendors, ...
  - Across various stages of lifecycle
KEY OBSERVATIONS

1. Large scale data centers have diverse device populations
2. Large sample sizes mean we can build accurate models
3. We can observe infrequent failure types at large scale
ERROR

- How failures manifest in software using a device

FAULT

- The underlying reason why a device fails
- *Permanent*: the fault appears every time
- *Transient*: the fault appears only sometimes
LARGE SCALE STUDIES

DRAM [DSN '15]

SSDs [SIGMETRICS '15]

Networks [IMC '18]
Socket
Memory channels
Dual In-line Memory Module (DIMM) slots
<table>
<thead>
<tr>
<th>Side</th>
<th>Width (mm)</th>
<th>Height (mm)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front</td>
<td>240</td>
<td>120</td>
<td>1.27 · W X W</td>
</tr>
<tr>
<td>Back</td>
<td>133.35</td>
<td>128.95</td>
<td></td>
</tr>
</tbody>
</table>

**Detail of Contacts A**
- Width: 3 mm, Height: 0.1 mm
- 1.50 · 0.10

**Detail of Contacts B**
- Width: 0.3 mm, Height: 0.15 mm
- 0.3~0.1

**Detail of Contacts C**
- Width: 2.10 mm, Height: 0.15 mm

**Detail of Contacts D**
- Width: 0.3 mm, Height: 0.05 mm

**Registering Clock Driver**
- SPD
- VTS

---

**Note:**
1. Tolerance on all dimensions unless otherwise stated. Units: millimeters.
Banks
Memory data

Error Correcting Code (ECC) metadata
MEASURING DRAM ERRORS

• Measured every logged error
  • Across Facebook's fleet
  • For 14 months
  • Metadata associated with each error

• Parallelized Map-Reduce to process

• Used R for further analysis
ANALYTICAL METHODOLOGY

• Measure server characteristics
  • Examined all servers with errors (error group)
  • Sampled servers without errors (control group)
• Bucket devices based on characteristics
• Measure relative failure rate
  • Of error group vs. control group
  • Within each bucket
KEY DRAM CONTRIBUTIONS

- Errors follow a power-law distribution
- Denial of service due to socket/channel
- Higher density = more failures
- DIMM architectural effects on reliability
- Workload influence on failures
- Model, page-offlining, page randomization
POWER-LAW DISTRIBUTION

• 1% of servers = 97.8% errors
POWER-LAW DISTRIBUTION

- 1% of servers = 97.8% errors
- Average is 55X median
• 1% of servers = 97.8% errors
• Average is 55X median
• Pareto distribution fits
  • Devices without errors tend to stay without errors
SOCKET/CHANNEL ERRORS

• Contribute majority of errors
SOCKET/CHANNEL ERRORS

• Contribute majority of errors
• Concentrated on a few hosts
• Symptoms $\approx$ server DoS
HIGHER DENSITY TRENDS

- Capacity, NO! Density, YES!

![Graph showing relative server failure rate vs. DIMM capacity (GB)]
HIGHER DENSITY TRENDS

- Capacity, NO! Density, YES!
- Higher density, more failure
  - Due to smaller feature sizes
DIMM architecture

- **Chips per DIMM, transfer width**
  - 8 to 48 chips
  - x4, x8 = 4 or 8 bits per cycle
  - Electrical implications
ARCHITECTURAL EFFECTS

- For the same transfer width:
  - More chips = more failures

![Graph showing the relationship between data chips per DIMM and server failure rate for different DRAM densities.](image)

- The relative failure rate of 2 Gb devices, going from 8 chips with a 4-bit transfer width to 48 chips with an 8-bit transfer width decreases failure rate by 7.1%.
- For 1 Gb devices, going from 16 chips with an 8-bit transfer width to 48 chips with a 16-bit transfer width decreases failure rate by 36.1%.
- Second, once the number of chips per DIMM increases beyond 16 and chips start using a larger transfer width of 8 bits per clock cycle (and are called DDR3), leading to the trend of net decrease in DRAM server failure rates in the future (while still improving DRAM cell reliability are achieved in future devices, maintaining or decreasing DRAM server failure rates in the future (while still improving DRAM cell reliability are easily outpaced by the quadratic increase in architectural power noise across the device). Such power noise could induce additional memory errors if, for example, charge were not evenly distributed to the electrical disturbance within a DIMM that may disrupt the electrical charge distribution across a DRAM chip.

- Considering how number of chips and transfer width contribute in failure rate going from 8 chips of the same transfer width, we find that increasing the number of chips from 32 to 48 increases failure rate by 36.1%. Second, once the number of chips per DIMM have the same or different transfer widths. First, among DDR4 vendors put into designing faster and more reliable DRAM cell architectures. Our insight is that differences between vendors can arise if vendors use less reliable DRAM chip manufacturers in our systems.

- The relative per-cell failure rate at different technology nodes (chip densities).

- Since DRAM server failure rate trends with respect to chip transfer widths separated by interface compared to the other 2 Gb devices, leading to additional memory errors if, for example, charge were not evenly distributed to the electrical disturbance within a DIMM that may disrupt the electrical charge distribution across a DRAM chip.
ARCHITECTURAL EFFECTS

- For the same transfer width:
  - More chips = more failures
- For different transfer widths:
  - More bits = more failures
    - Likely related to electrical noise
WORKLOAD INFLUENCE

- No consistent trends across CPU and memory utilization
- But workload varies by ~6X
  - May be due to distribution for read/write behavior
• Use statistical regression model
  • Compare control group versus error group
  • Logistic (linear) regression in R
  • Trained using data from analysis

• Enable exploratory analysis
MODELING MEMORY FAILURES

Memory error model

Density
Chips
Age

Relative server failure rate

68
MODELING MEMORY FAILURES

\[ \ln \left( \frac{F}{1 - F} \right) = \hat{\beta}_{\text{Intercept}} + (\text{Capacity} \cdot \hat{\beta}_{\text{Capacity}}) + (\text{Density2Gb} \cdot \hat{\beta}_{\text{Density2Gb}}) + (\text{Density4Gb} \cdot \hat{\beta}_{\text{Density4Gb}}) + (\text{Chips} \cdot \hat{\beta}_{\text{Chips}}) \\
+ (\text{CPU\%} \cdot \hat{\beta}_{\text{CPU\%}}) + (\text{Age} \cdot \hat{\beta}_{\text{Age}}) + (\text{CPUs} \cdot \hat{\beta}_{\text{CPUs}}) \]
# Exploratory Analysis

## Inputs

<table>
<thead>
<tr>
<th>Factor</th>
<th>Low-end</th>
<th>High-end (HE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>4 GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>Density2Gb</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Density4Gb</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Chips</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>CPU%</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>Age</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CPUs</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>

## Output

| Predicted relative failure rate | 0.12 | 0.78 |

6.5X difference in yearly failures
TOOL AVAILABLE ONLINE

http://www.ece.cmu.edu/~safari/tools/memerr/
Page offlineing

- *System-level technique to reduce errors*
- *When a page has an error, take the page offline*
  - *Copy* its contents to a new location
  - *Poison* the page to prevent allocation
PAGE OFFLINING AT SCALE

- First study at large scale
- Cluster of 12,276 servers
PAGE OFFLINING AT SCALE

- First study at large scale
- Cluster of 12,276 servers
- Reduced error rate by 67%
PAGE OFFLINING AT SCALE

- First study at large scale
- Cluster of 12,276 servers
- Reduced error rate by 67%
- Prior simulations: 86 to 94%
- Did not account for OS failures to lock page
DRAM WEAROUT IN THE FIELD

- DRAM shows signs of wear
- Idea: What if we performed wear leveling in DRAM?
- Can be done in OS without modifying hardware
**PAGE RANDOMIZATION**

**Input:** The address of a physical page to randomize.

1. Lock the page.
2. Flush any pending updates to the page.
3. Randomly select a new free page to allocate.
4. Migrate the contents of the old page to the new page.
5. Update the page table mappings and remove any stale TLB entries.
6. Unlock the page.

*Prototype implemented in Debian 6.0.7 kernel*
PAGE RANDOMIZATION

- Can perform with low overhead (< 5%)
- Can fine-tune desired rate of randomization
Errors follow a power-law distribution
• Denial of service due to socket/channel
• Higher density = more failures
• Architectural effects on reliability
• Workload influence on failures
• Model, page-offlining, page randomization
RELATED WORK

- **DRAM errors at Google**
  [Schroeder+ SIGMETRICS'09]

- **Component failures + simulated page offlining**
  [Hwang+ ASPLOS'12]

- **Error correction, location, multi-DIMM errors**
  [Sridharan+ SC'12, SC'13; DeBardeleben+ SELSE'14]
LARGE SCALE STUDIES

DRAM
[DSN '15]

SSDs
[SIGMETRICS '15]

Networks
[IMC '18]
Flash chips
SSD controller

- translates addresses
- schedules accesses
- performs wear leveling
Stored data

ECC metadata
TYPES OF SSD FAILURES

Ones that cause SMALL ERRORS

- 10's of flipped bits per KB
- Silently corrected by SSD controller

Ones that cause LARGE ERRORS

- 100's of flipped bits per KB
- Corrected by host using driver
- Referred to as SSD failure
Examined lifetime hardware counters

- Across Facebook's fleet
- Devices deployed between 6 months and 4 years
- 15 TB to 50 TB read and written
- Planar, Multi-Level Cell (MLC)

Snapshot-based analysis
<table>
<thead>
<tr>
<th>Errors</th>
<th>54,326</th>
<th>0</th>
<th>2</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data written</td>
<td>10TB</td>
<td>2TB</td>
<td>5TB</td>
<td>6TB</td>
</tr>
</tbody>
</table>
Errors 54,326 0 2 10

Data written 10TB 2TB 5TB 6TB

2018-12-3
<table>
<thead>
<tr>
<th>Errors</th>
<th>Data written</th>
</tr>
</thead>
<tbody>
<tr>
<td>54,326</td>
<td>10TB</td>
</tr>
<tr>
<td>0</td>
<td>2TB</td>
</tr>
<tr>
<td>2</td>
<td>5TB</td>
</tr>
<tr>
<td>10</td>
<td>6TB</td>
</tr>
<tr>
<td>Errors</td>
<td>54,326</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>Data written</td>
<td>10TB</td>
</tr>
</tbody>
</table>

2018-12-3

**Buckets**
Errors | Data written |
---|---|
54,326 | 10TB |
0 | 2TB |
2 | 5TB |
10 | 6TB |

2018-12-3
<table>
<thead>
<tr>
<th>Errors</th>
<th>54,326</th>
<th>0</th>
<th>2</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data written</td>
<td>10TB</td>
<td>2TB</td>
<td>5TB</td>
<td>6TB</td>
</tr>
</tbody>
</table>
KEY SSD CONTRIBUTIONS

- Distinct lifecycle periods
- Read disturbance not prevalent in the field
- Higher temperatures cause more failures
- Amount of data written by OS is misleading
- Write amplification trends from the field
FAILURE MODELING

- Built a model across 6 SSD server configurations
- **Weibull** $(0.3, 5e3)$
- Most errors are from a small set of SSDs
Storage lifecycle background: the *bathtub curve* for disk drives

[Schroeder+, FAST'07]
Storage lifecycle background: the *bathtub curve* for disk drives

- **Early failure period**
- **Useful life period**
- **Wearout period**

Failure rate vs. Usage

[Schroeder+, FAST'07]
SSD LIFECYCLE PERIODS

SSD failure rate

Data written (TB)

720GB, 1 SSD
720GB, 2 SSDs
SSD LIFECYCLE PERIODS

- **Early failure period**
- **Useful life period**
- **Wearout period**

![Graph showing SSD failure rate against data written (TB)]
SSD LIFECYCLE PERIODS

- **Early detection period**
- **Useful life period**
- **Wearout period**

Data written (TB)

SSD failure rate

- Platform A
- Platform B
SSD LIFECYCLE PERIODS

• We believe there are two distinct pools of flash cells
  • The "weak" pool fails first, during early detection
  • The "strong" pool follows the bathtub curve
• Burn-in testing is important to help the SSD identify the weak pool of cells
Read disturbance errors

• Charge drift from reads to neighboring cells
• Documented in prior controlled studies on chips
READ DISTURBANCE ERRORS

- SSDs with the most reads

![Graph showing read disturbance errors for 3.2TB, 1 SSD (R/W = 2.14) and 1.2TB, 1 SSD (R/W = 1.15).]
READ DISTURBANCE ERRORS

- SSDs with the most reads
- No statistically significant difference at low data read versus high data read
TEMPERATURE DEPENDENCE
TEMPERATURE DEPENDENCE

- Higher temperature = more failures

<table>
<thead>
<tr>
<th>Average temperature (°C)</th>
<th>SSD failure rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.00</td>
</tr>
<tr>
<td>40</td>
<td>0.50</td>
</tr>
<tr>
<td>50</td>
<td>1.00</td>
</tr>
</tbody>
</table>

- 720GB, 1 SSD
- 720GB, 2 SSDs
On some devices, high temperature may throttle or shut down SSD.
TEMPERATURE DEPENDENCE

- Throttling is an effective technique to reduce failures
- Potentially decreases device performance, however

![Graph showing SSD failure rate vs. average temperature]

- 1.2TB, 1 SSD
- 3.2TB, 1 SSD

Average temperature (°C)
Access patterns and SSD writes

System buffering
- Data served from OS caches
- Decreases SSD usage

Write amplification
- Updates to small amounts of data
- Increases erasing and copying
System caching reduces the impact of SSD writes.
OS WRITES MISLEADING

- No statistically significant correlation with failures at high write volume

![Graph showing SSD failure rate vs. data written to OS (TB)]

- Platform A: 1.2TB, 2 SSDs
- Platform B: 3.2TB, 2 SSDs

Data written to OS (TB)
OS WRITES MISLEADING

- No statistically significant correlation with failures at high write volume
- Data written to OS versus SSD is not correlated for high write volume
Flash devices use a translation layer to locate data.
**Translation layer**

**Logical address space**

<offset₁, size₁>

<offset₂, size₂>

...

**Physical address space**
Sparse data layout

more translation metadata

potential for higher write amplification
Dense data layout

less translation metadata

potential for lower write amplification
WRITE AMPLIFICATION

- Sparse data shows signs of higher failure rates
- Likely due to write amplification

Diagram showing SSD failure rate against translation data (GB) for 720GB, 1 SSD.
KEY SSD CONTRIBUTIONS

• Distinct lifecycle periods
• Read disturbance not prevalent in the field
• Higher temperatures cause more failures
• Amount of data written by OS is misleading
• Write amplification trends from the field
RELATED WORK

• *Examined chip-level failures*
  E.g., [Cai+ DATE'12, ICCD'12, DATE'13, ICCD'13, DSN'15, HPCA'17]

• *Examined a simulated SSD controller with 45 flash chips*
  [Grupp+ FAST'12]

• *Reliability of SSD controllers (NOT chips)*
  [Ouyang+ ASPLOS'14]

• *Microsoft and Google SSDs over multiple years*
  [Narayanan+ SYSTOR'16, Schroeder+ FAST'16]
LARGE SCALE STUDIES

DRAM [DSN '15]

SSDs [SIGMETRICS '15]

Networks [IMC '18]
SOFTWARE-AIDED NETWORKS

- Simple, custom switches
- Software-based fabric networks
- Automated repair of common failures
MEASURING NETWORK FAILURE

- **Incident reports**
  - Across Facebook's fleet
  - Over 7 years
  - Details on faulty device, severity, ...

- **Vendor repair tickets**
  - Across Facebook's fleet
  - Over 14 months
  - Details on location, timing, ...
INCIDENT REPORTS

*Switch Failures* cause *Software Failures* that result in *Incidents (with reports)*
KEY NETWORK CONTRIBUTIONS

- Software-aided networks greatly reduce errors
- High bandwidth switches cause more incidents
- Rack switches are a bottleneck for reliability
- Data center WAN reliability models
NETWORK DESIGN TRENDS

- Older hard-wired networks
- 9X incident increase over 4 years

Hard-wired network
NETWORK DESIGN TRENDS

- Older hard-wired networks
- 9X incident increase over 4 years
- Newer software-aided designs
- 2X fewer incidents
- 2.8X on a per-device basis

Hard-wired network  Software-aided network
SWITCH TYPE TRENDS

Highest bandwidth
Moderate bandwidth
Lowest bandwidth
SWITCH TYPE TRENDS

Highest bandwidth
Moderate bandwidth
Lowest bandwidth

Graphic showing trends in switch types from 2011 to 2017.
SWITCH TYPE TRENDS

Highest bandwidth

Moderate bandwidth

Lowest bandwidth

Hard-wired

Software-aided
Rack switches make up 82% of network devices.
WAN architecture

**Edge nodes**
- Route requests across different network paths
- Connected by multiple links

**Links**
- Optical fiber cables that connect edges
MODELING WAN RELIABILITY

Edge

Failure rate  Repair rate

Link
### Modeling WAN Reliability

<table>
<thead>
<tr>
<th></th>
<th>Failure rate</th>
<th>Repair rate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edge</strong></td>
<td>$O(\text{months})$</td>
<td>$O(\text{hours})$</td>
</tr>
<tr>
<td><strong>Link</strong></td>
<td>$O(\text{months})$</td>
<td>$O(\text{days})$</td>
</tr>
</tbody>
</table>
MODELING WAN RELIABILITY

Edge Link

Failure rate Repair rate

We provide open models
Software-aided networks greatly reduce errors
High bandwidth switches cause more incidents
Rack switches are a bottleneck for reliability
Data center WAN reliability models
RELATED WORK

• *Identify network incidents as leading cause*
  [Barroso+ DCaaC, Gunawi+ SoCC'6, Oppenheimer+ USITS'03, Brewer Google Tech. Rep. '17, Wang+ DSN'17]

• *Hard-wired network studies*
  [Zhuo+ SIGCOMM'17, Gill+ SIGCOMM'11, Potharaju+ IMC'13]

• *Complementary large scale works focused on device trends*
  [Potharaju+ SoCC'13, Turner+ SIGCOMM'10, Govindan+ SIGCOMM'16]
LARGE SCALE STUDIES

DRAM [DSN '15]

SSDs [SIGMETRICS '15]

Networks [IMC '18]
If we measure the device failures in modern data centers, then we can learn the reasons why devices fail, develop models to predict device failures, and learn from failure trends to make recommendations to enable workloads to tolerate device failures.
CONCLUSION

The problem of understanding why data center devices fail can be solved by using the scale of modern data centers to observe failures and by building robust statistical models to understand the implications of the failure trends.
CONTRIBUTIONS

1. Large scale failure studies
   We shed new light on device trends from the field

2. Statistical failure models
   We enable the community to apply what we learn

3. Evaluate best practices in the field
   We provide insight into how to tolerate failures
LIMITATIONS

Only examined one company's data centers

Do not consider combination of device effects

Do not consider silent data corruption
FUTURE RESEARCH

Further field study based analysis

Other devices, statistical techniques, environments

HW/SW cooperative techniques

Use learnings to inform design decisions

Introspective fault monitoring and reduction

Systems that can identify and adapt their behavior
THESIS PUBLICATIONS

Large scale reliability studies

- **DRAM** [Meza+ DSN'15]
- **SSDs** [Meza+ SIGMETRICS'15]
- **Network** [Meza+ IMC'18]
OTHER PhD PUBLICATIONS

Non-volatile memory

- **DRAM + NVM** [Meza+ CAL'12]
- **Persistent Memory** [Meza+ WEED'13]
- **Multi-Level Cell** [Yoon+ TACO'14]
- **Row Buffers Locality** [Yoon+ ICCD'15]
- **Row Buffer Sizes** [Meza+ ICCD'12]

Main memory architecture

- **Bit Flips** [Luo+ DSN'14]
- **Overview** [Mutlu+ KIISE'15]

Datacenter Energy

- **Sustainable DC Design** [Chang+ ASPLOS'12]
EARLIER PUBLICATIONS

Energy efficiency studies

- JouleSort [Rivoire + Computer'07]
- DB Energy [Harizopoulos + CIDR'09]
- OLTP Energy [Meza + ISLPED'09]
- Sustainable DC Design [Meza + IMCE'10]
- Sustainable Server Design [Chang + HotPower'10]
FACEBOOK PUBLICATIONS

Systems architecture + reliability

- **Power Management** [Wu+ ISCA'16]
- **Time Series DBs** [Pelkonen+ VLDB'15]
- **Load Testing** [Veeraraghavan+ OSDI'16]
- **Disaster Recovery** [Veeraraghavan+ OSDI'18]
ACKNOWLEDGEMENTS

• My advisor, Onur, who had confidence in me even when I didn't
• My committee – Greg, James, Kaushik – who were always there to listen and guide me
• The SAFARI group at CMU for lifelong friendships
• Family, friends, and colleagues (too many to list!) who kept me going (Partha, Kim, Yee Jiun ...)
LARGE SCALE STUDIES OF MEMORY, STORAGE, AND NETWORK FAILURES IN A MODERN DATA CENTER

T H E S I S O R A L

J U S T I N M E Z A

Committee
Prof. Onur Mutlu (Chair)
Prof. Greg Ganger
Prof. James Hoe
Dr. Kaushik Veeraraghavan (Facebook, Inc.)
BACKUP SLIDES
More Techniques?

• We believe our DRAM work provides a promising direction
  • Analyze failures, build models, design techniques

• At the same time, we wanted to focus on:
  • Instrumentation + analysis of new devices (SSDs)
  • Going more in depth in software-level effects (networks)

• We sketch how to extend our methodology in the thesis
Other Data Centers

• We tie our results to fundamental device properties
• We build models that control for data center specifics
  • E.g., DRAM: Workload has an effect, but our models can factor that in to other features (e.g., CPU util)
• We do see evidence of similarities to other data centers
  • E.g., Networks: Data center networks ≈ B4, WAN ≈ B2 in [Jain+SIGCOMM'13, Govindan+SIGCOMM'16]
How Widespread is the Impact?

- For DRAM and SSDs we observe fail-slow behavior
- Slow devices can cause cascading failures [FAST'18]
- For Network devices, failure domain is large leading to widespread effects
DRAM Failure Details

- Retention
  - Cells must be refreshed
  - Variable retention time complicates matters
- Disturbance
  - Bit flips due to charged particles
  - Data pattern disturbance & RowHammer effect
- Endurance
  - Wear out due to physical phenomonena
SSD Failure Details

- **Endurance**
  - Cells wear out after many program-erase cycles
  - Floating gate loses ability to adequately store charge

- **Temperature**
  - Shrinks and expands boards and components
  - Arrhenius effect ages cells at accelerated rate

- **Disturbance**
  - Pass through voltage causes neighboring cell disturbance

- **Program failures, retention failures**
Network Failure Details

- Hardware (see DRAM and SSD failure details)
- Unplanned fiber cuts
  - Everything from anchors dragging to backhoes
- Bugs
  - Switches run a variety of software, can be buggy
- Operational mistakes
  - Attempting to repair a switch without turning it off
Exploratory analysis

Table III: Predicted relative failure rates among different server types.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Low-end</th>
<th>High-end (HE)</th>
<th>HE/↓density</th>
<th>HE/↓CPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>4 GB</td>
<td>16 GB</td>
<td>4 GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>Density 2Gb</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Density 4Gb</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Chips</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>CPU%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>Age</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CPUs</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Predicted relative failure rate</td>
<td>0.12</td>
<td>0.78</td>
<td>0.33</td>
<td>0.51</td>
</tr>
</tbody>
</table>
We find that SSDs fall into one of three categories with respect to temperature increases. Similar failure rates or slight increases in failure rates are observed as the temperature range of 30 to 40°C. Temperature sensors embedded on the SSD cards, which correspond to flash-based SSDs in order to make adequate data available quickly due to the temperature-activated Arrhenius effect. Temperatures have been shown to cause cells to age more quickly and increase the work required to cool the SSDs. As a result, SSDs attempt to change their behavior (e.g., reduce the frequency of SSD access or, in the case of platforms C and higher, perform as aggressive temperature reduction techniques as other components in the machine). In our experiments, we observed an event that can be correlated with temperature thresholds. The SSD controllers take in response to temperature events, we examined an event that can be correlated with temperature.

Figure 10 plots the failure rate for SSDs that have various combinations of temperature sensors at the server or rack level. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime.

Figure 8: SSD failure rate vs. DRAM buffer usage (B) across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B.

One hypothesis is that temperature-sensitive SSDs with integrated temperature sensors are especially important to understand for flash-based SSDs in order to make adequate data available quickly due to the temperature-activated Arrhenius effect. Temperatures have been shown to cause cells to age more quickly and increase the work required to cool the SSDs. As a result, SSDs attempt to change their behavior (e.g., reduce the frequency of SSD access or, in the case of platforms C and higher, perform as aggressive temperature reduction techniques as other components in the machine). In our experiments, we observed an event that can be correlated with temperature thresholds. The SSD controllers take in response to temperature events, we examined an event that can be correlated with temperature.

Figure 10 plots the failure rate for SSDs that have various combinations of temperature sensors at the server or rack level. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime. We examine the effects of temperature, PCIe bus power, and system-level writes on the errors observed over an SSD's lifetime.

Table 5. The role of external factors affects the operation of flash-based SSDs. In flash-based SSDs, the errors observed over an SSD's lifetime are especially important to understand for flash-based SSDs in order to make adequate data available quickly due to the temperature-activated Arrhenius effect. Temperatures have been shown to cause cells to age more quickly and increase the work required to cool the SSDs. As a result, SSDs attempt to change their behavior (e.g., reduce the frequency of SSD access or, in the case of platforms C and higher, perform as aggressive temperature reduction techniques as other components in the machine). In our experiments, we observed an event that can be correlated with temperature thresholds. The SSD controllers take in response to temperature events, we examined an event that can be correlated with temperature.

Figure 8: SSD failure rate vs. DRAM buffer usage (B) across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B. We observe similar DRAM buffer usage across six applications that run on Platform B.
DC fabric has fewer incidents
Main cause across all severities
Edge node MTBF distribution

Typical edge node failure rate is on the order of months.

\[ y = 462.88e^{2.3408p} \]

\[ R^2 = 0.938 \]
Edge node MTTR distribution

$$y = 1.513e^{4.256p}$$
$$R^2 = 0.8744$$

Edge node mean time to repair is on the order of hours
Fiber vendor MTBF distribution

Model
\[ y = 336.51e^{3.4371p} \]
\[ R^2 = 0.8354 \]
Fiber vendor MTTR distribution

Model

$$y = 1.1345e^{4.7709p}$$

$$R^2 = 0.9781$$
Minimizing backbone outages

Simulation
objective = six 9's

Capacity plan
Node 1: Links A, B
Node 2: Links X, Y