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.)

Carnegie Mellon University


MODERN
DATA CENTERS

100's

SOFTWARE SYSTEMS

[Hahn LISA'18]

1,000,000's

CONTAINERS

[Hahn LISA'18]

1,000,000,000's

REQUESTS PER SECOND

[Hahn LISA'18]

WANT
HIGH RELIABILITY

PROBLEM

Device failures disrupt data center workloads


1. INTERDEPENDENCE
2. DISTRIBUTION
3. COMMODITY HW

PROBLEM 1: INTERDEPENDENCE

WEB SERVER

PROGRAM

CACHE

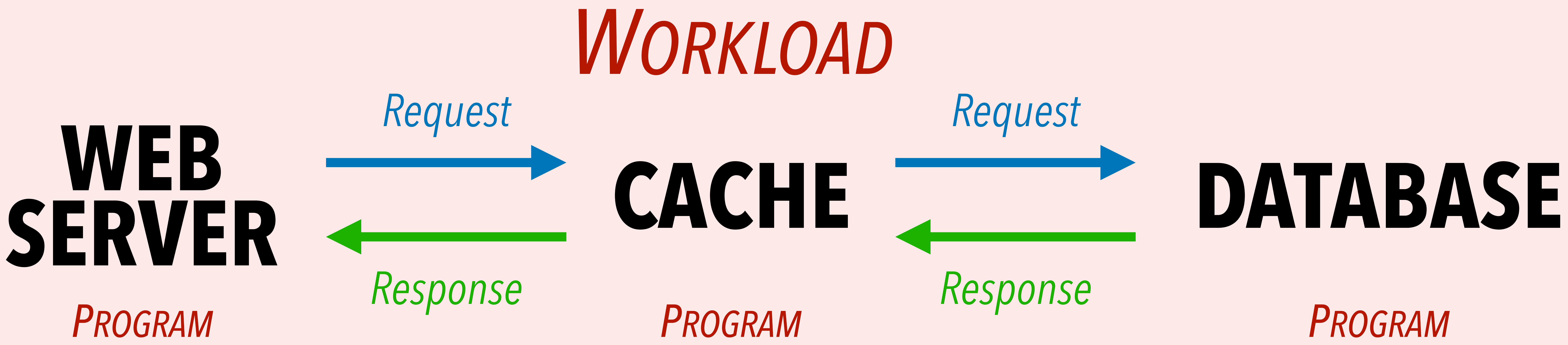
PROGRAM

DATABASE

PROGRAM

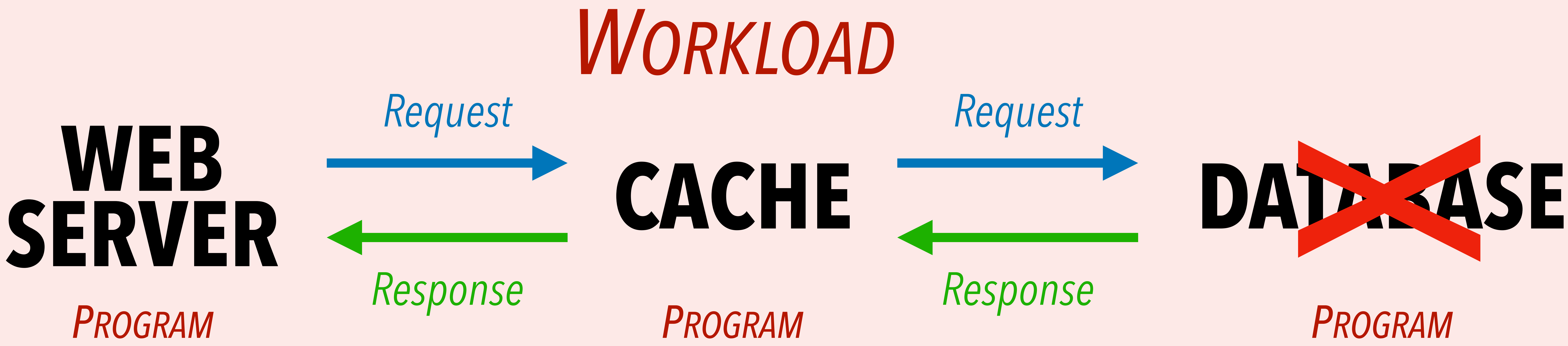
The programs running in modern data centers make up larger workloads.

PROBLEM 1: INTERDEPENDENCE



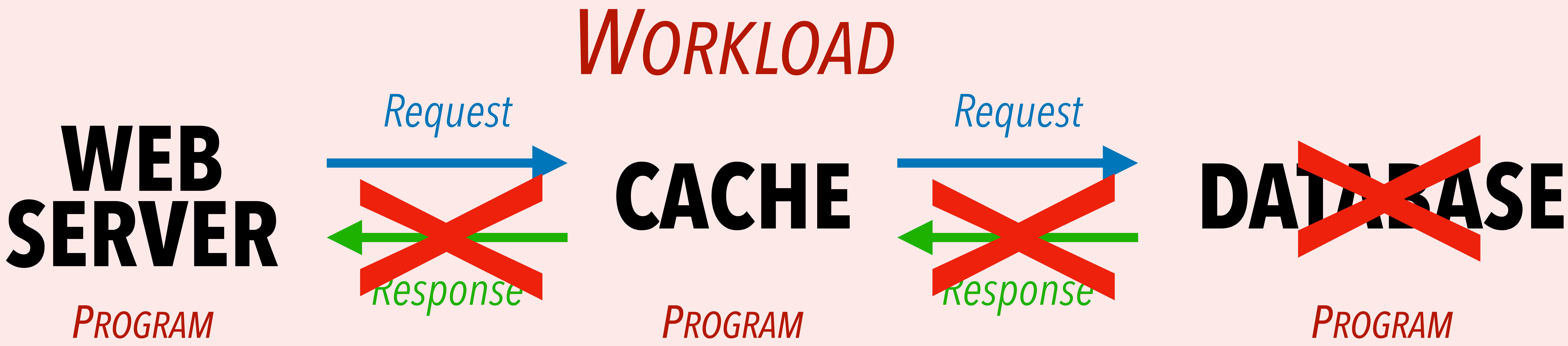
The programs running in modern data centers make up larger workloads.

PROBLEM 1: INTERDEPENDENCE



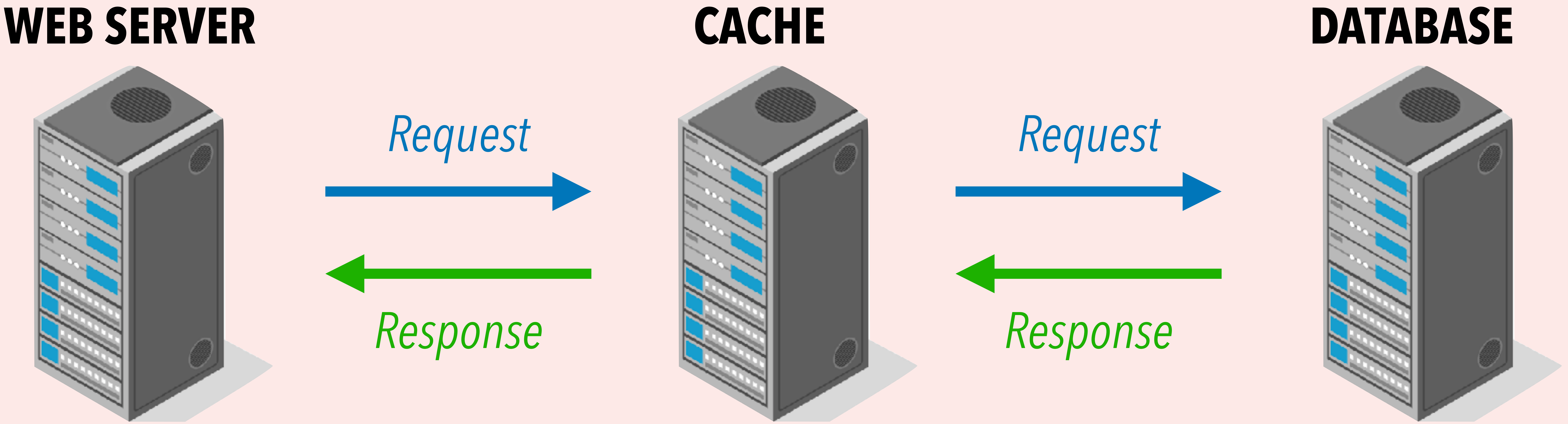
The programs running in modern data centers make up larger workloads.

PROBLEM 1: INTERDEPENDENCE



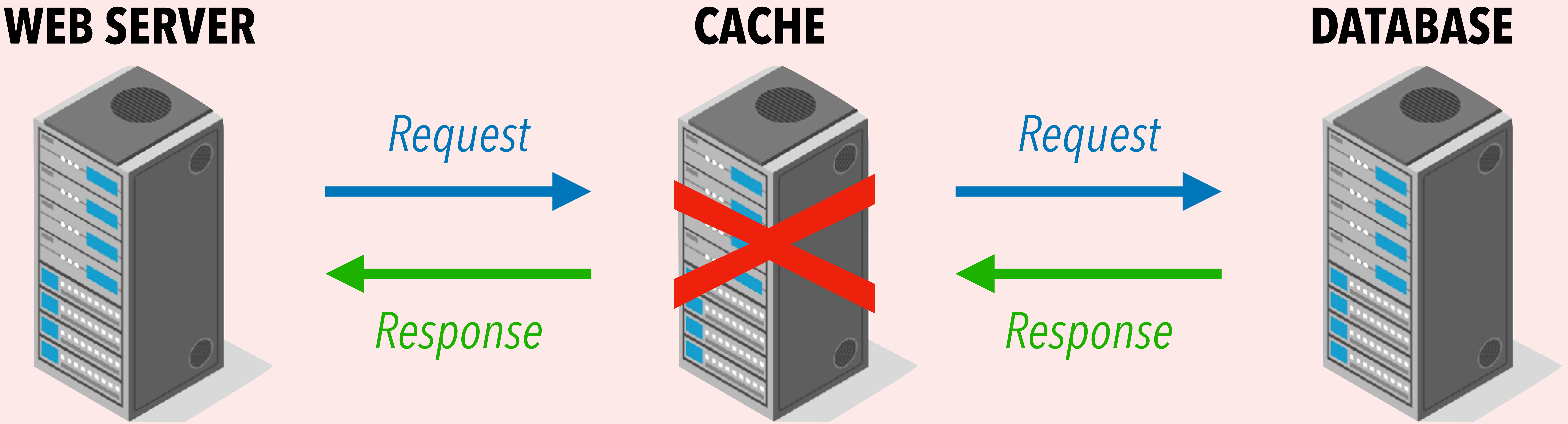
The programs running in modern data centers make up larger workloads.

PROBLEM 2: DISTRIBUTION



Workloads in modern data centers are distributed across many servers.

PROBLEM 2: DISTRIBUTION

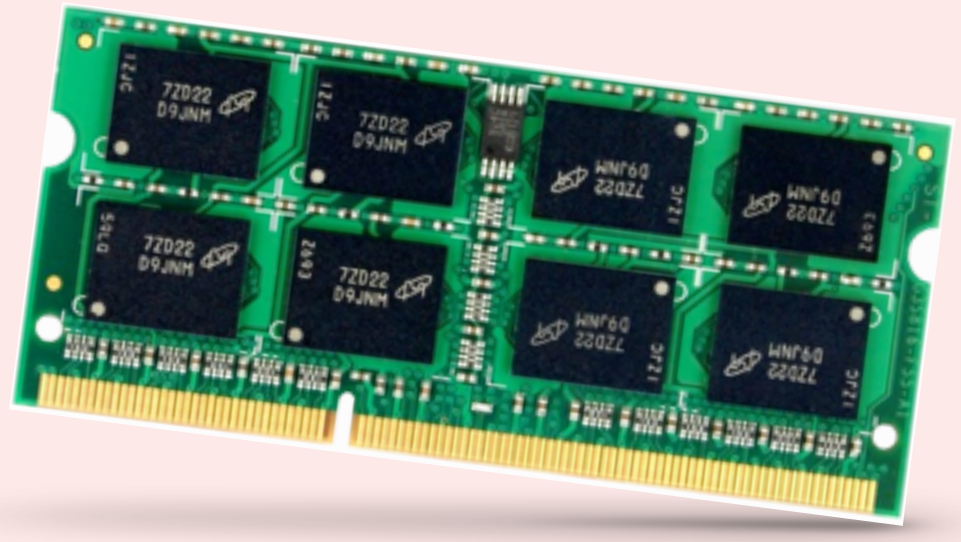


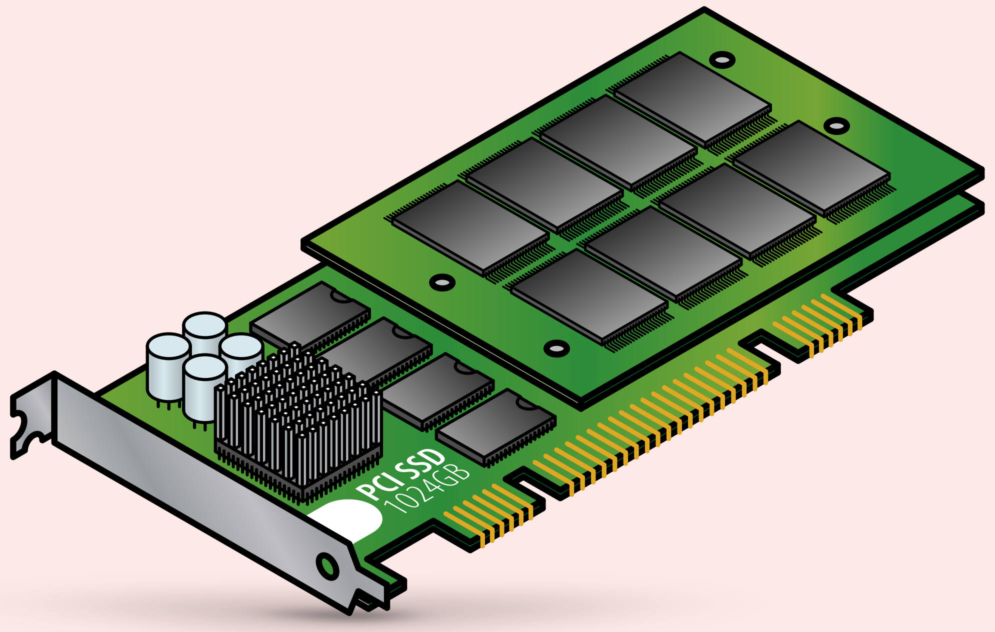
Workloads in modern data centers are distributed across many servers.

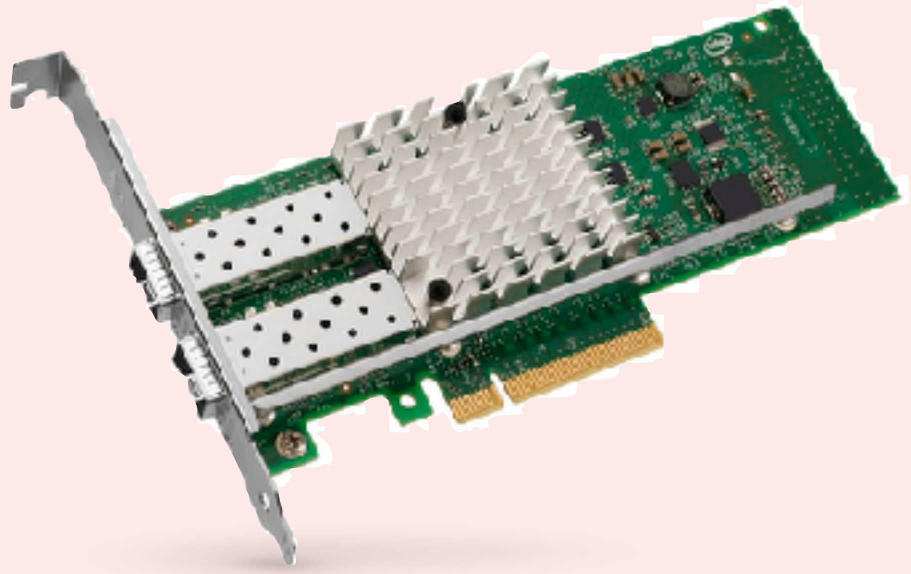
PROBLEM 2: DISTRIBUTION

Workloads in modern data centers are distributed across many servers.

PROBLEM 3: COMMODITY HW

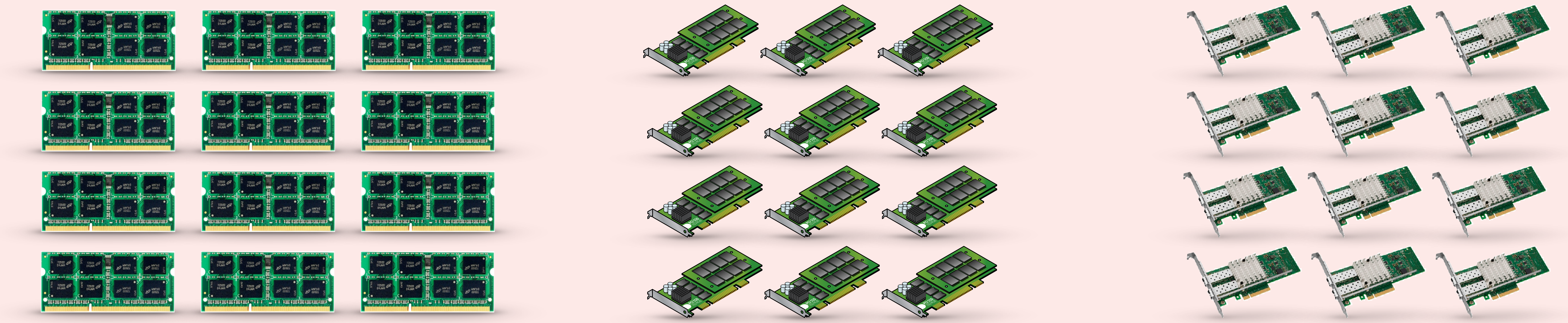






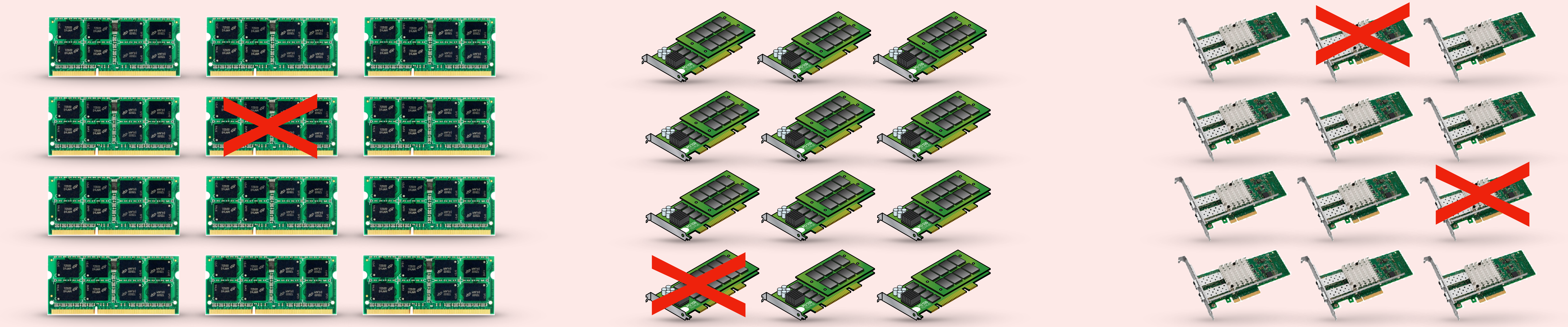
Modern data centers trade off reliability for using simpler, commodity hardware.

PROBLEM 3: COMMODITY HW



Modern data centers trade off reliability for using simpler, commodity hardware.

PROBLEM 3: COMMODITY HW



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, how Microsoft got it back online

ARIF BACCHUS EMAIL @ABACJOURN SEP 17TH, 2018 IN LATES

HOME > NEWS > MIDDLE EAST

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

TECHNOLOGY NEWS DECEMBER 12, 2010 / 9:14 PM / 8 YEARS AGO

Amazon websites outage was due to hardware failure

GitHub suffers major outage caused by faulty storage appliance

Where is your Octocat now?

Fail-Slow at Scale: Evidence of Hardware Performance Faults in Large Production Systems

Haryadi S. Gunawi[1], Riza O. Suminto[1], Russell Sears[2], Casey Golliher[2], Swaminathan Sundararaman[3], Xing Lin[4], Tim Emami[4], Weiguang Sheng[5], Nematollah Bidokhti[5], Caitie McCaffrey[6], Gary Grider[7], Parks M. Fields[7], Kevin Harms[8], Robert B. Ross[8], Andree Jacobson[9], Robert Ricci[10], Kirk Webb[10], Peter Alvaro[11], H. Birali Runesha[12], Mingzhe Hao[1], and Huaicheng Li[1]

[1]University of Chicago, [2]Pure Storage, [3]Parallel Machines, [4]NetApp, [5]Huawei, [6]Twitter, [7]Los Alamos National Laboratory, [8]Argonne National Laboratory, [9]New Mexico Consortium, [10]University of Utah, [11]University of California, Santa Cruz, and [12]UChicago Research Computing Center

[FAST'18]

"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

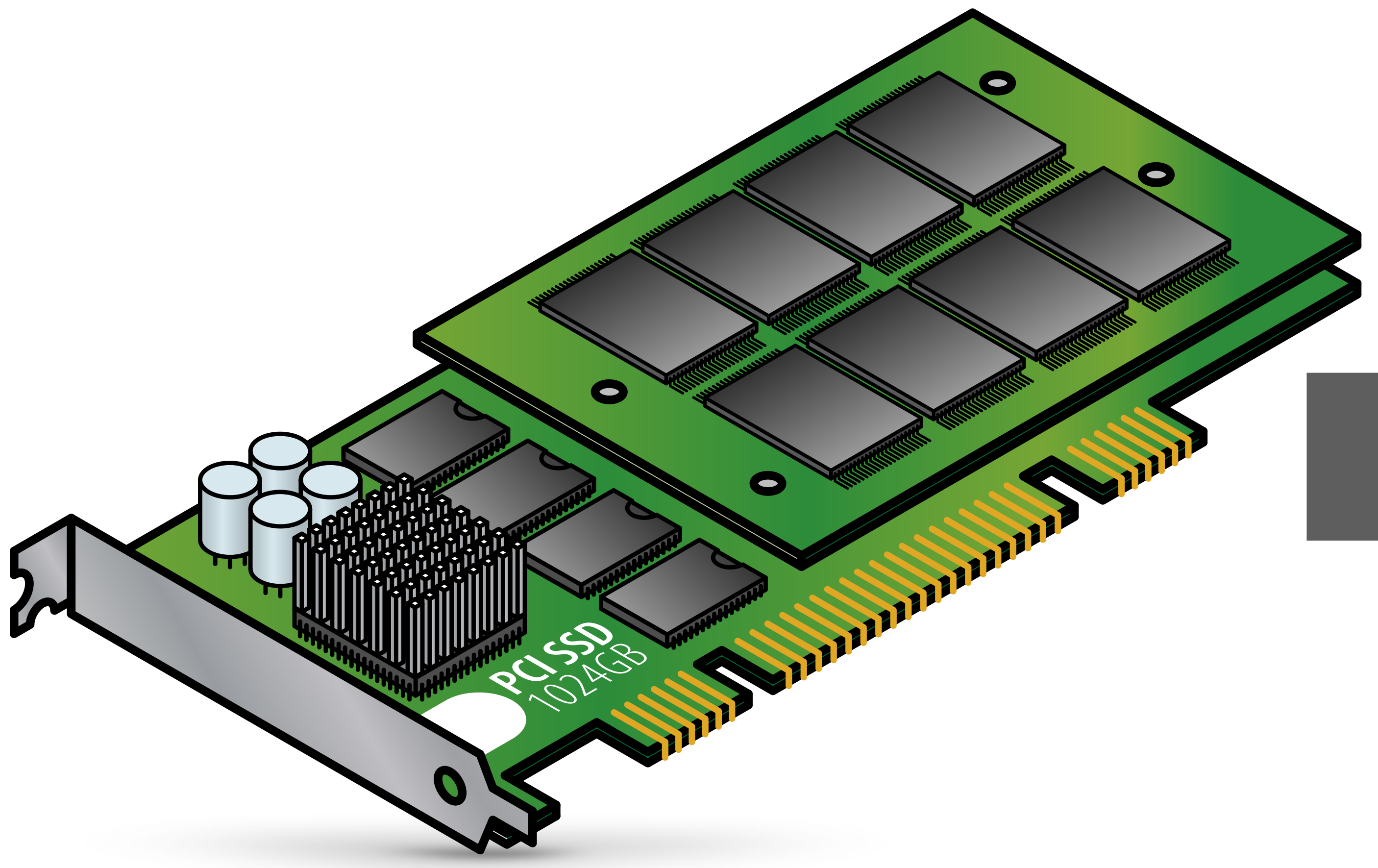
CHALLENGES

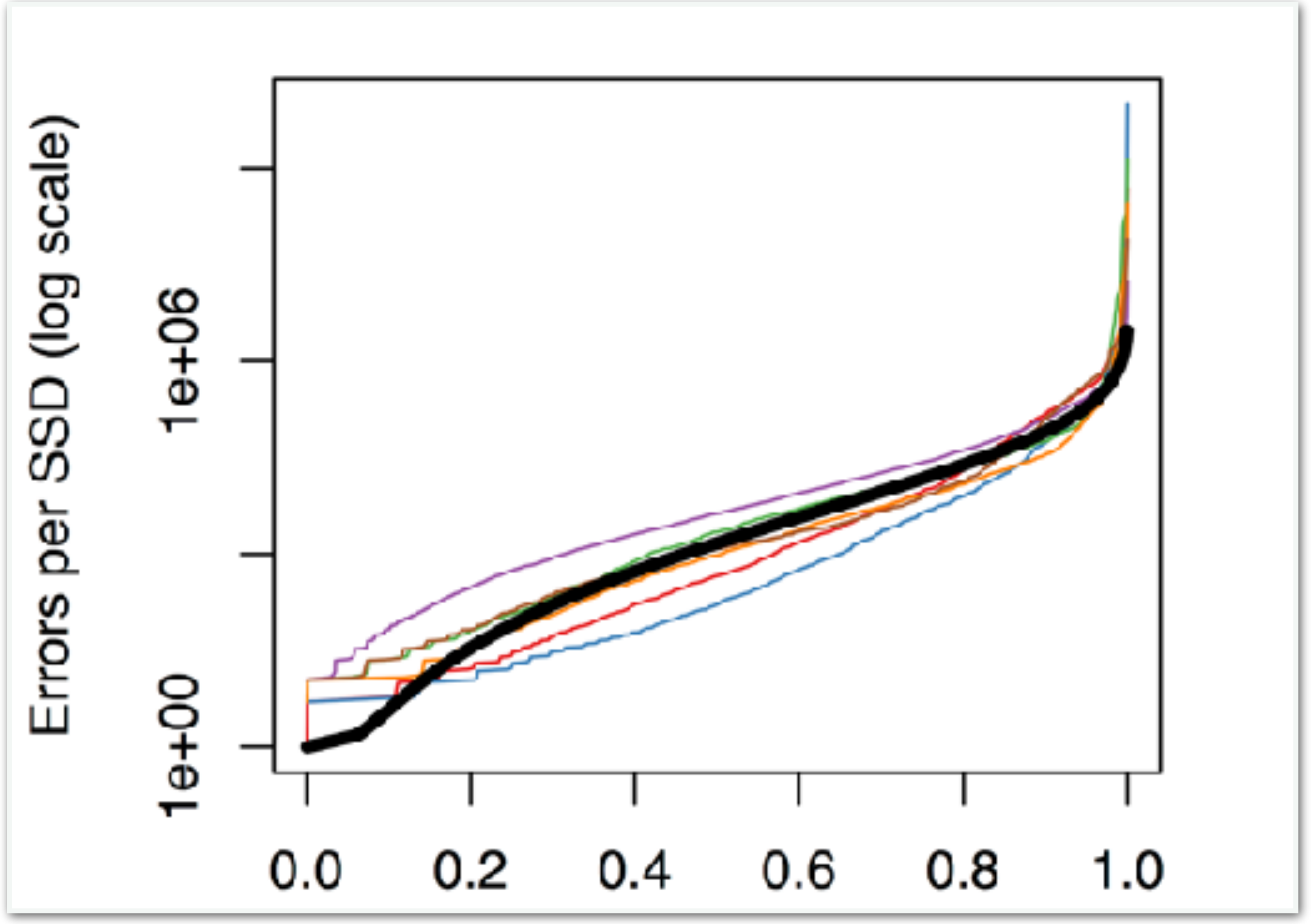
1. *Most device reliability studies are small scale*
2. *Prior large scale studies hard to generalize*
3. *Limited evaluation of techniques in the wild*

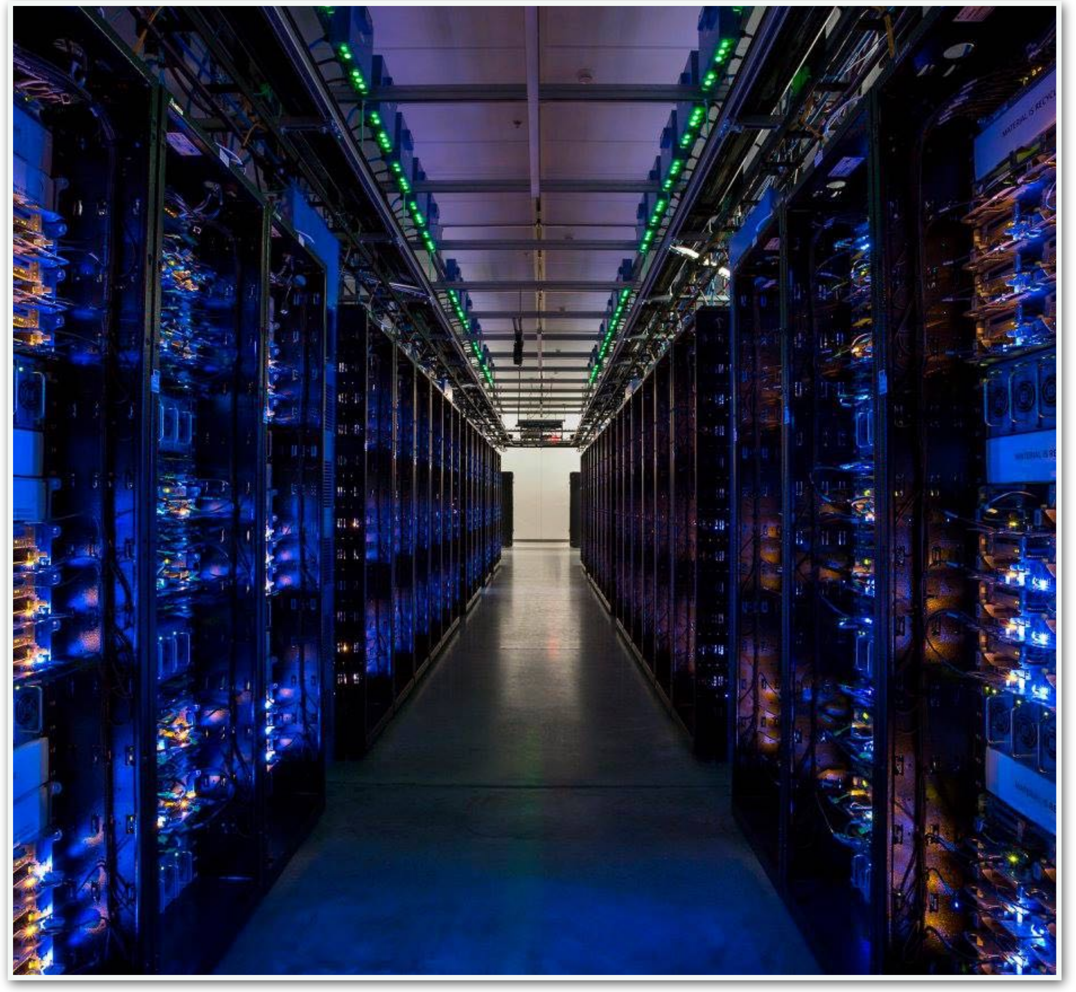
THESIS STATEMENT

*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

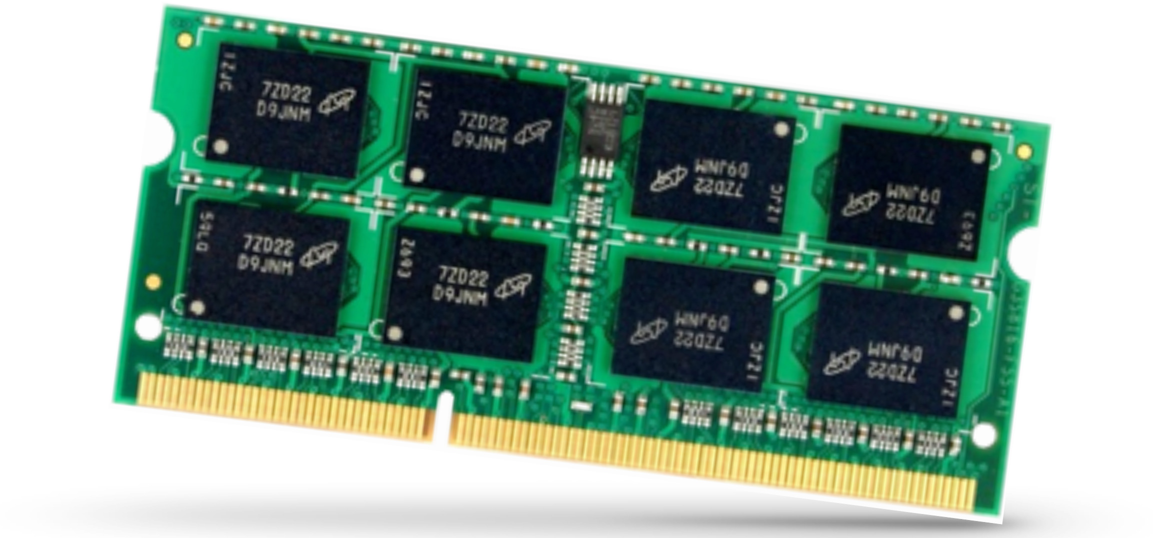






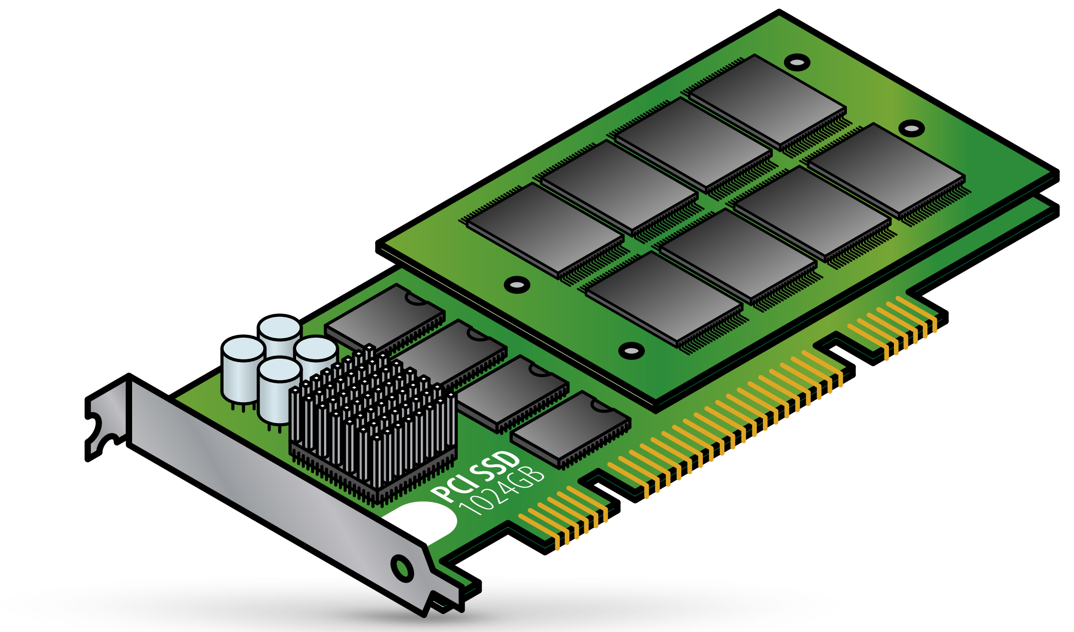
CONTRIBUTIONS

1. Large scale failure studies



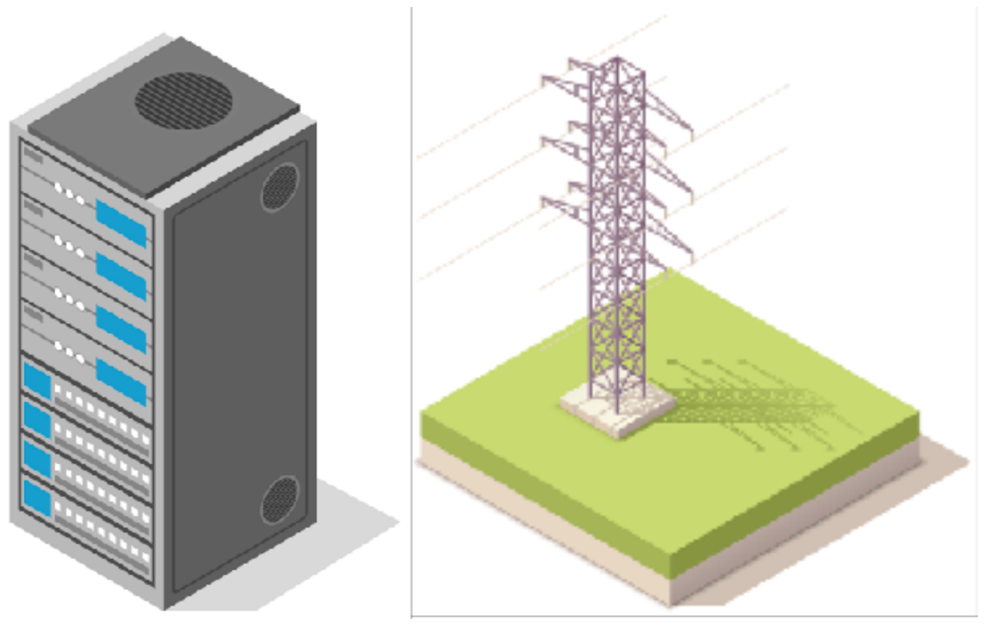
DRAM

[DSN '15]



SSDs

[SIGMETRICS '15]



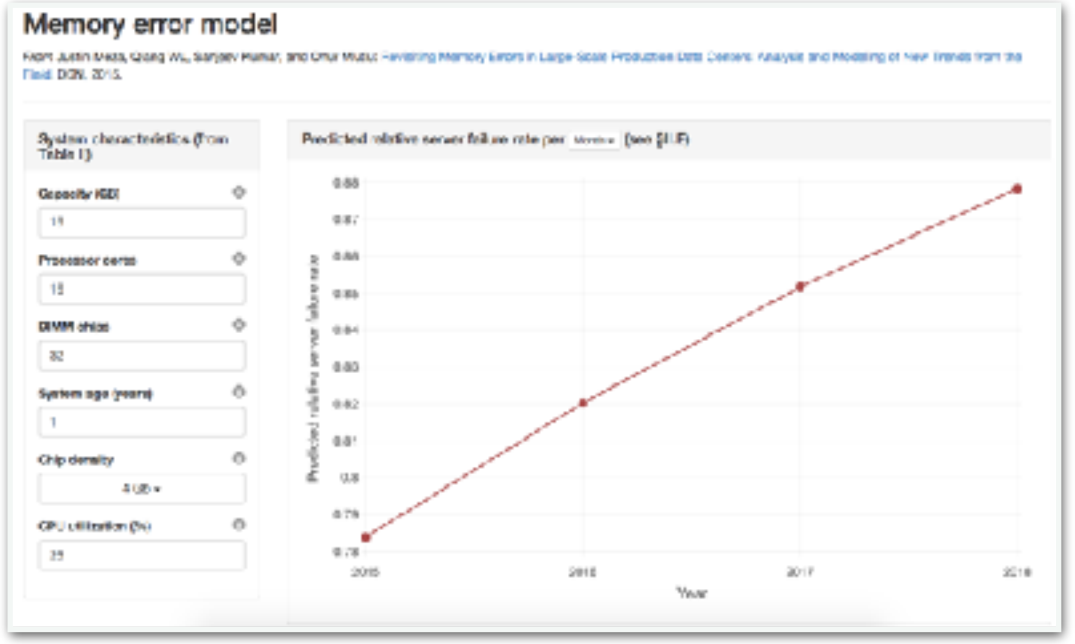
Networks

[IMC '18]

We shed new light on device trends from the field

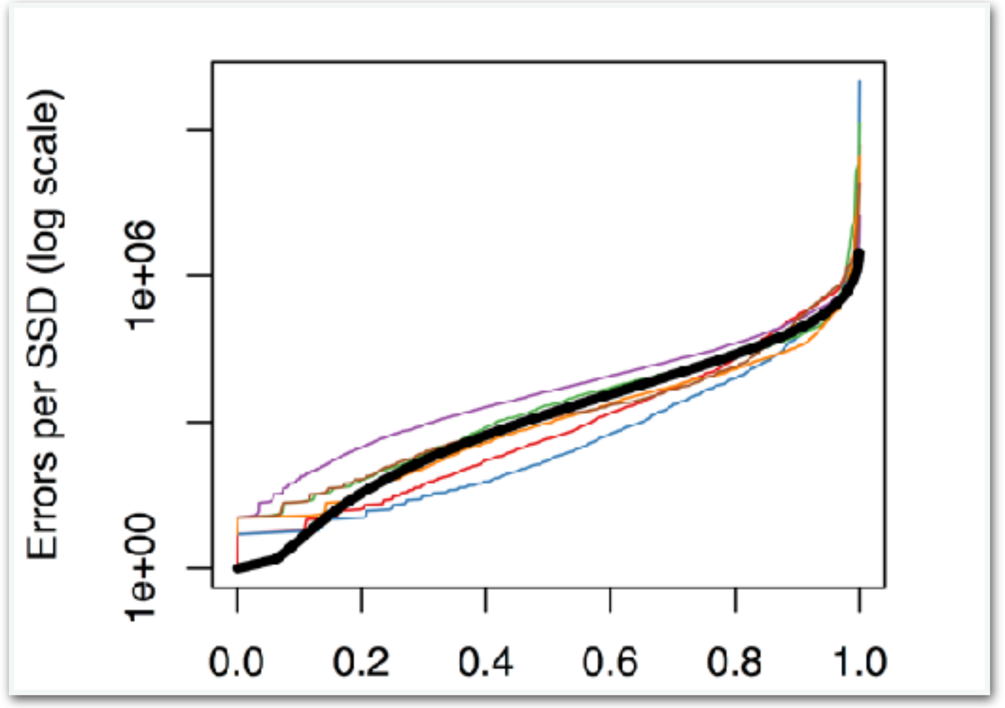
CONTRIBUTIONS

2. Statistical failure models



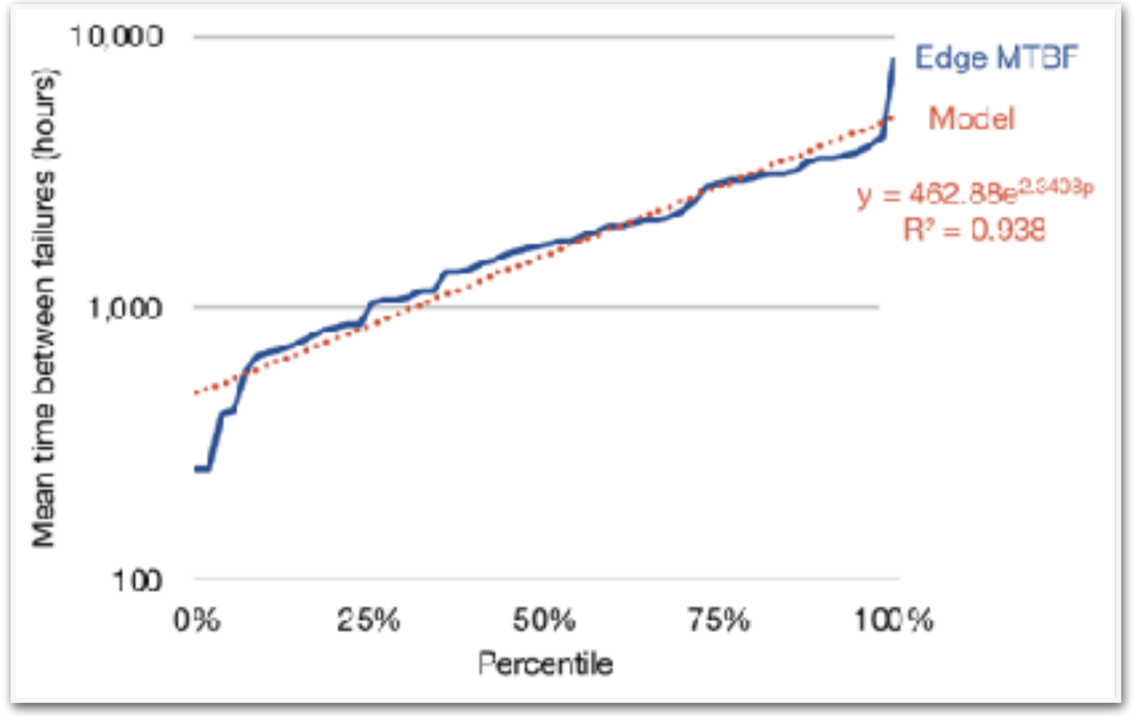
DRAM

[DSN '15]



SSDs

[SIGMETRICS '15]



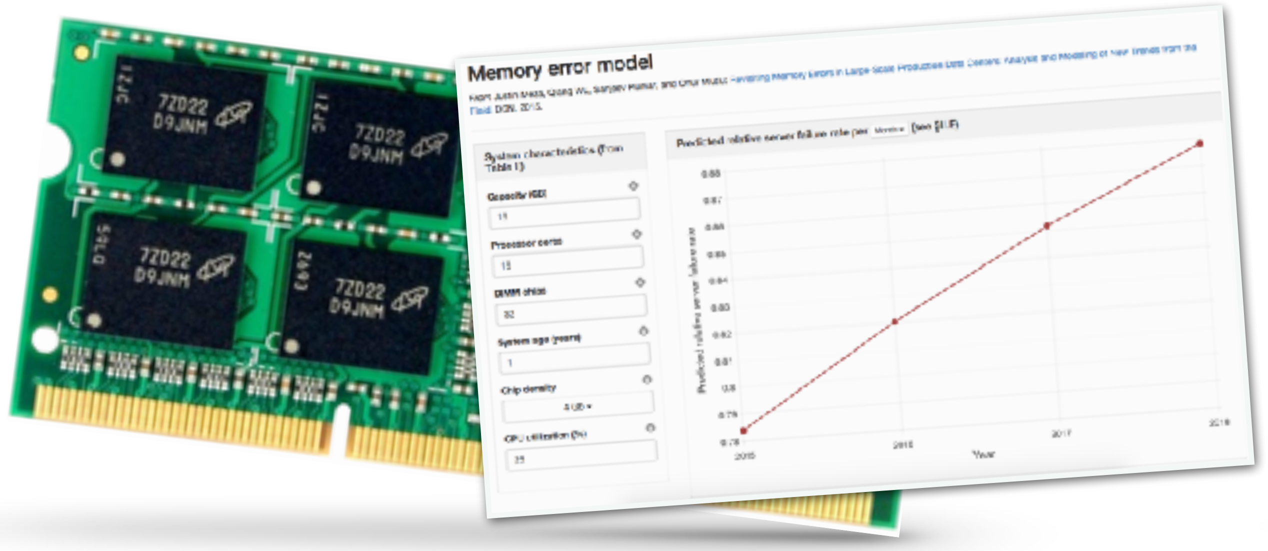
Networks

[IMC '18]

We enable the community to apply what we learn

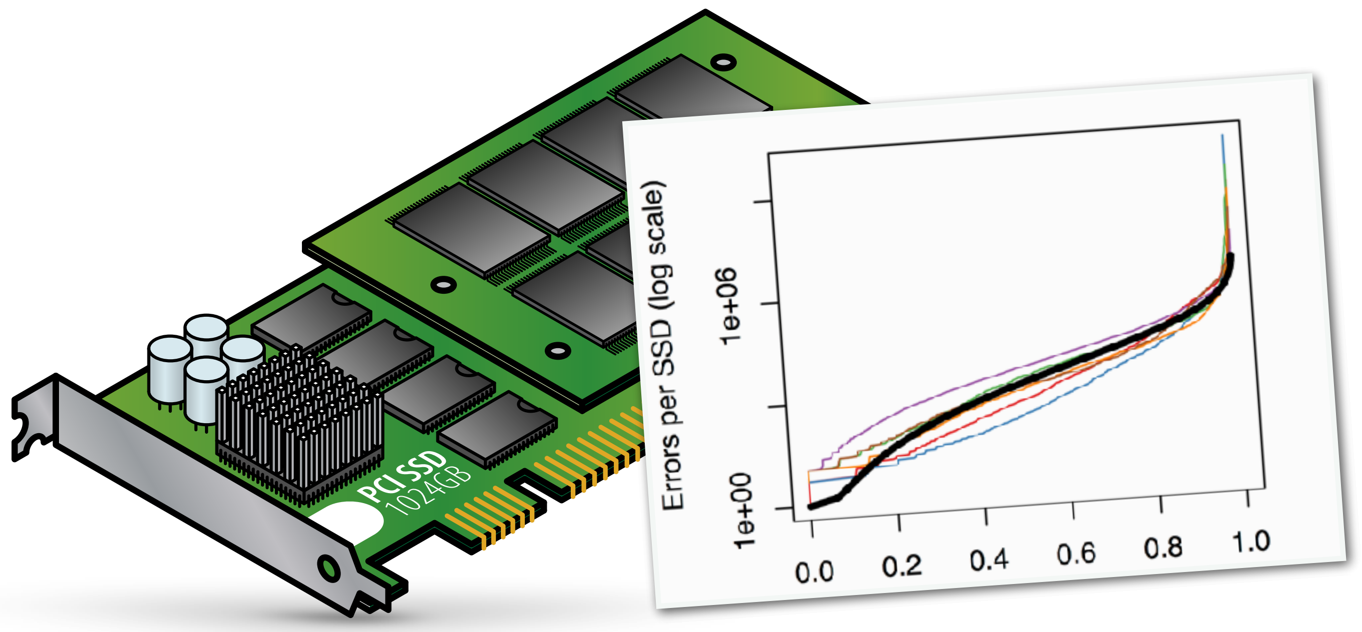
CONTRIBUTIONS

3. Evaluate best practices in the field



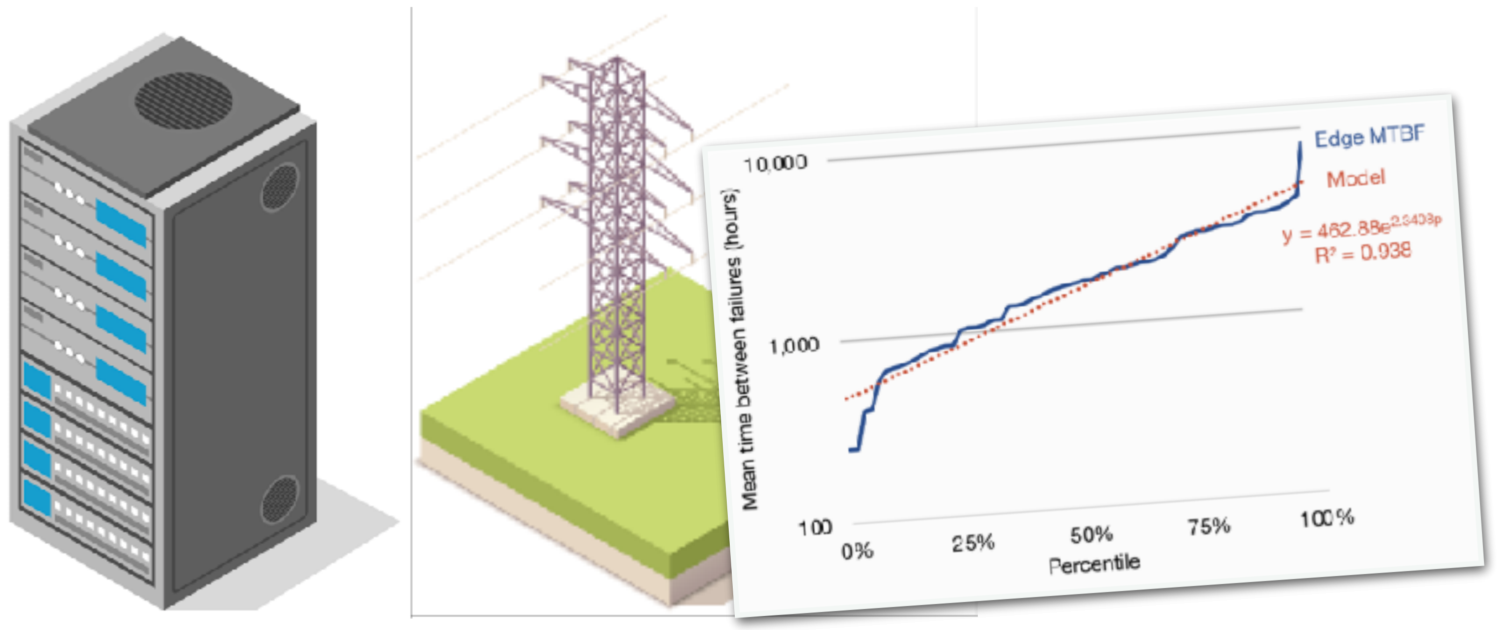
DRAM

Page offlining



SSDs

OS write buffering



Networks

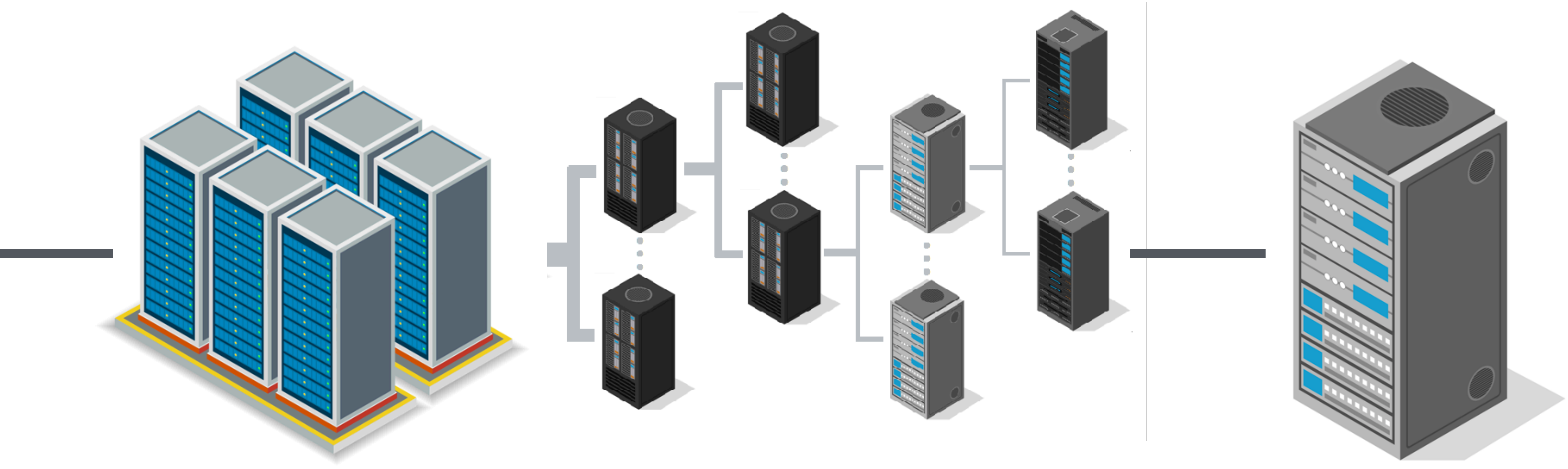
Software-based networks

We provide insight into how to tolerate failures

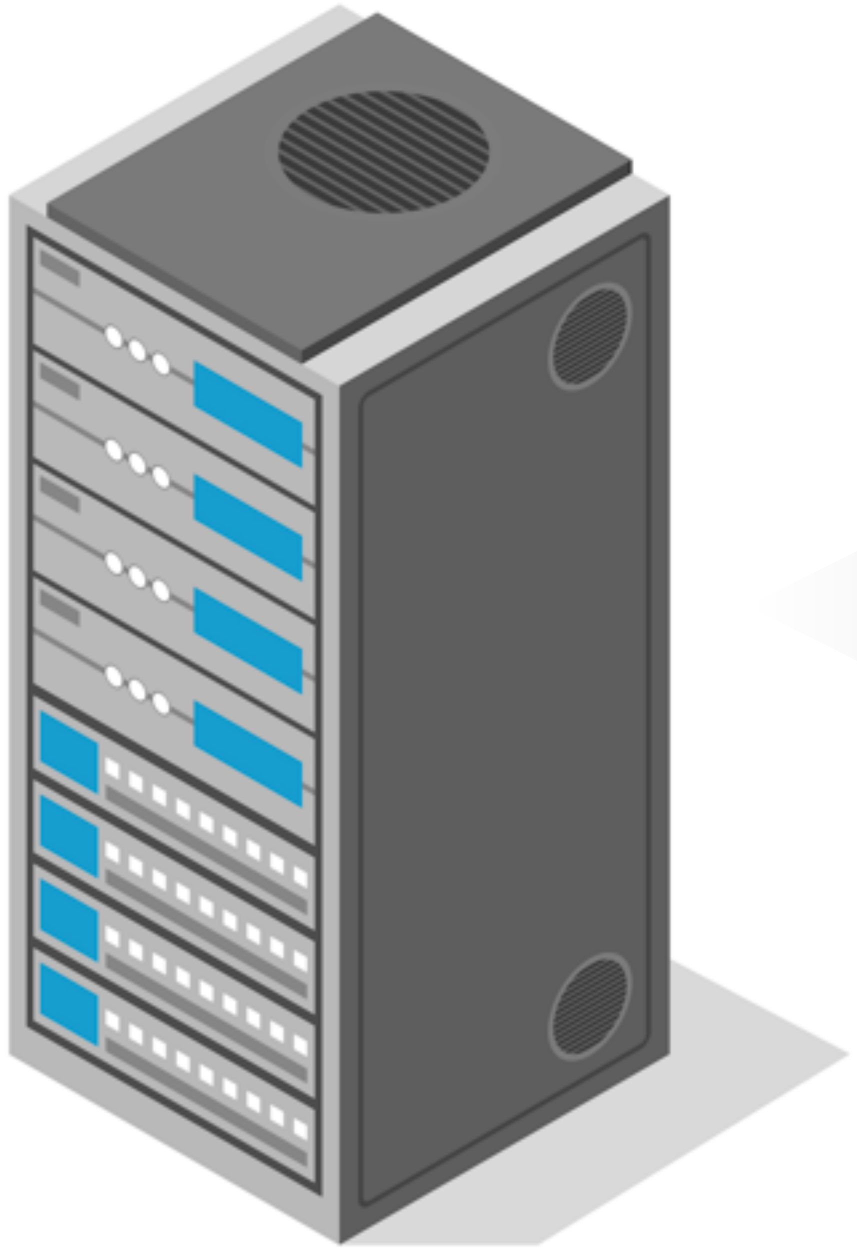
OUTLINE

1. Modern data center background
2. Large scale device failure studies
   - Memory: DRAM
   - Storage: SSDs
   - Network: Switches and WAN
3. Conclusion


WAN
Internet
ISP
Edge Node

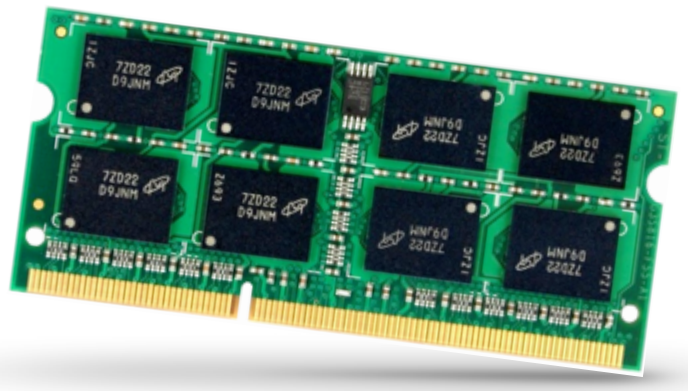


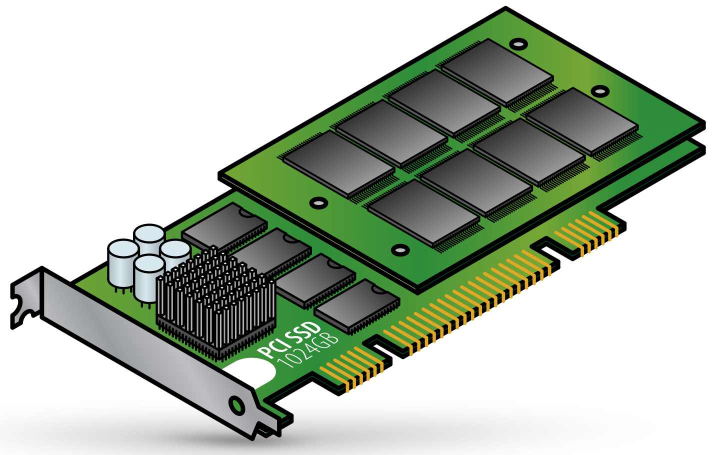
Core Switches Data Center Fabric Top of Rack Switch



Server Rack

Server Sleds

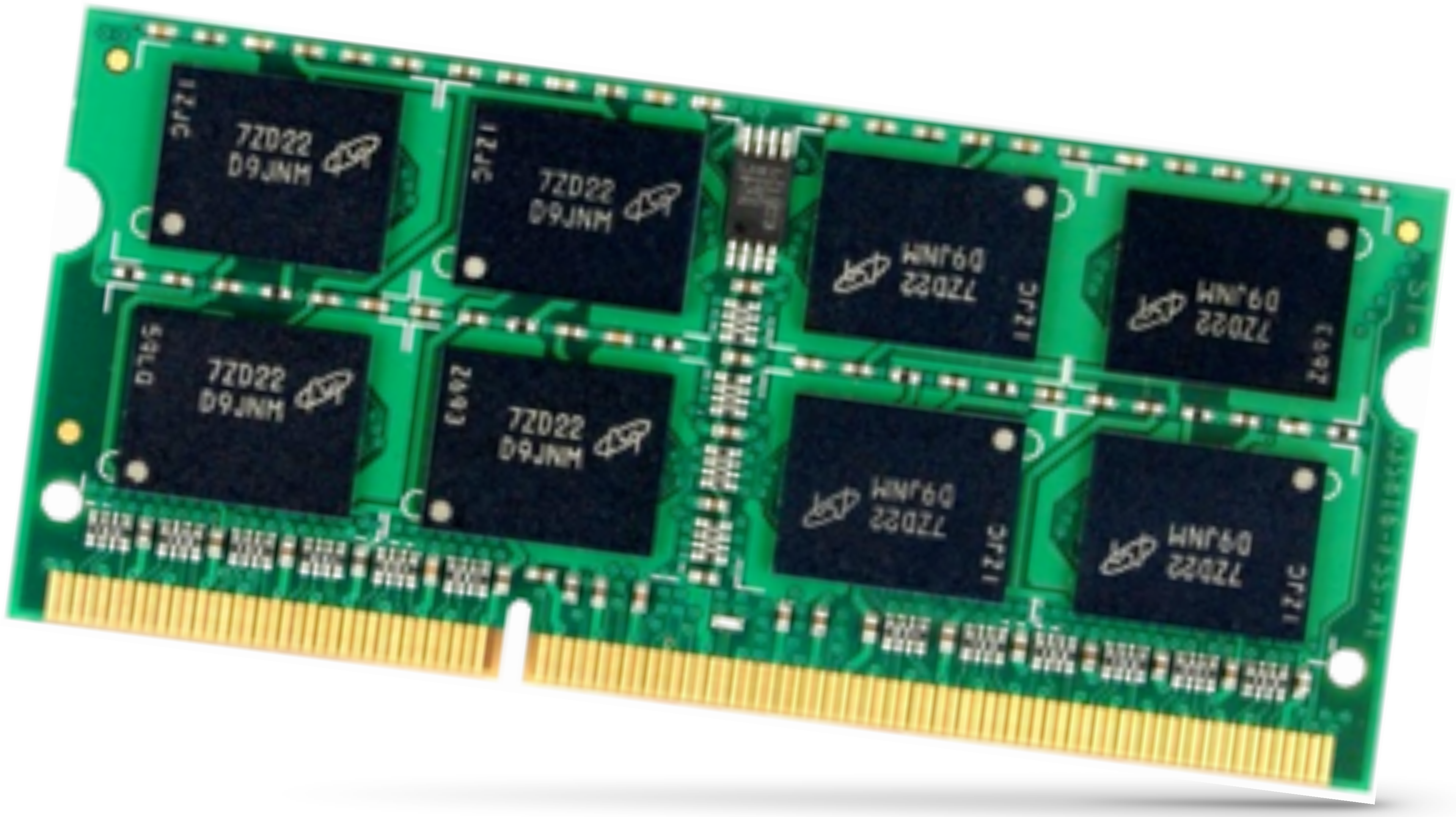




Devices

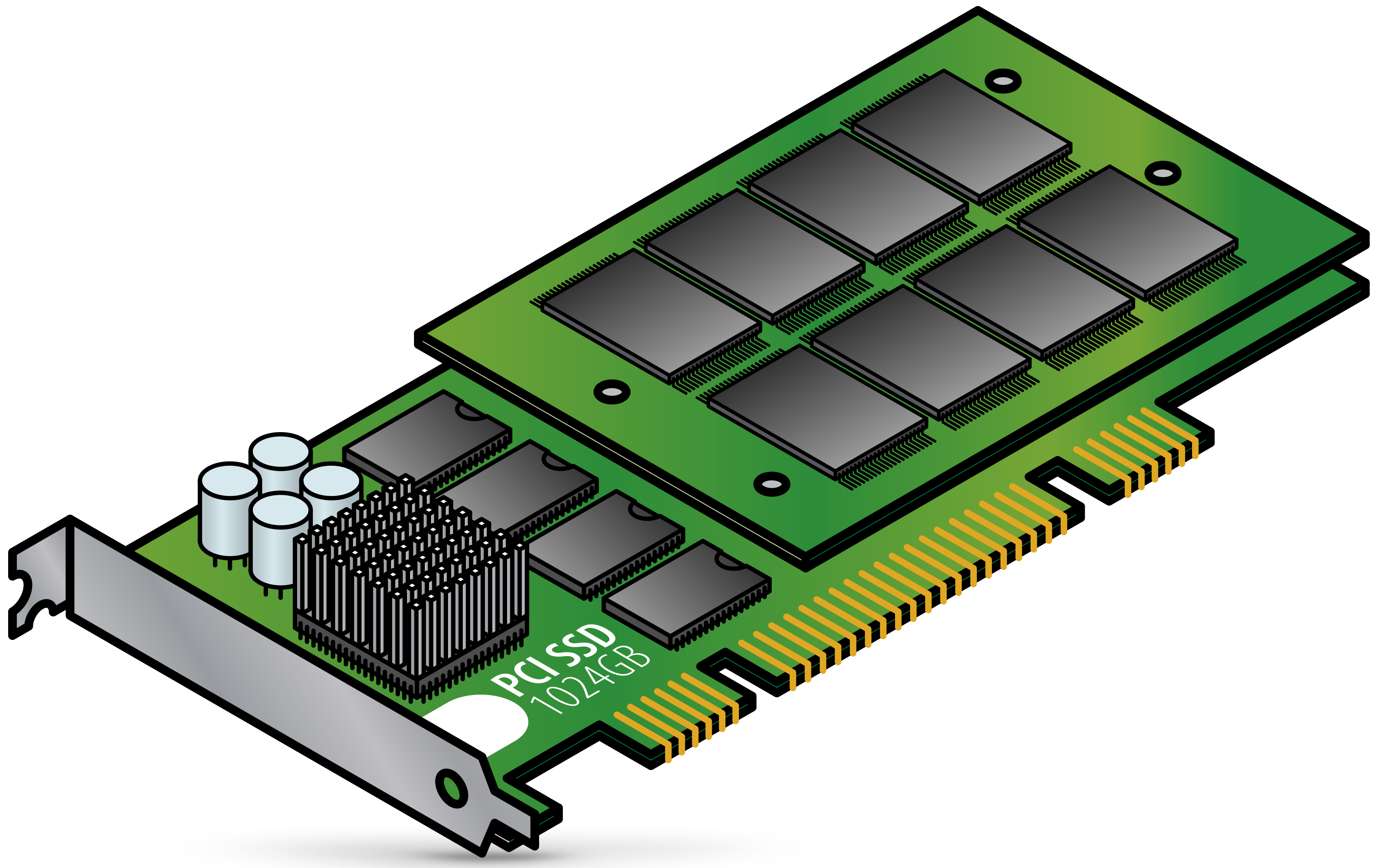
MEMORY

Dynamic Random Access Memory (DRAM)



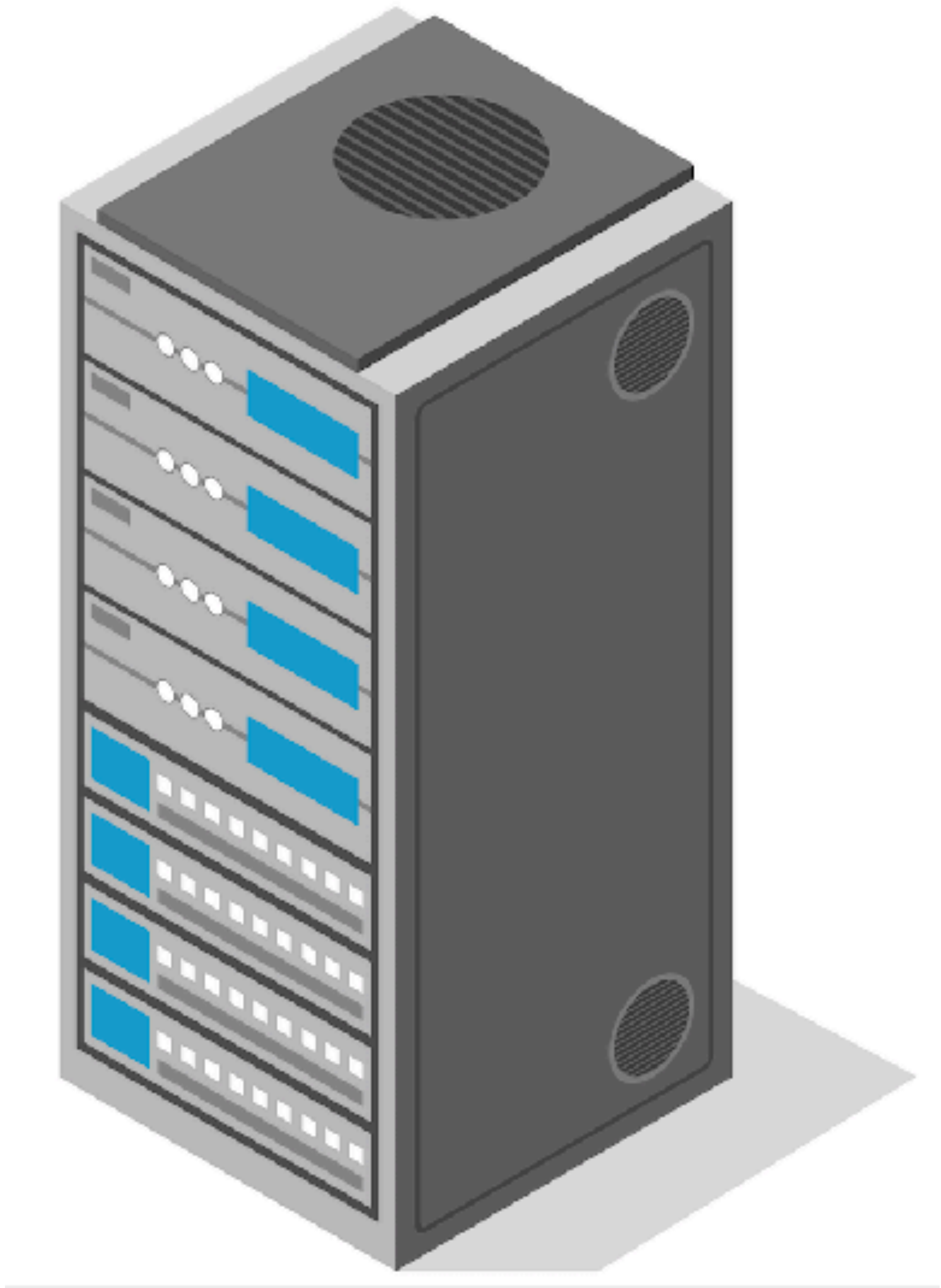
STORAGE

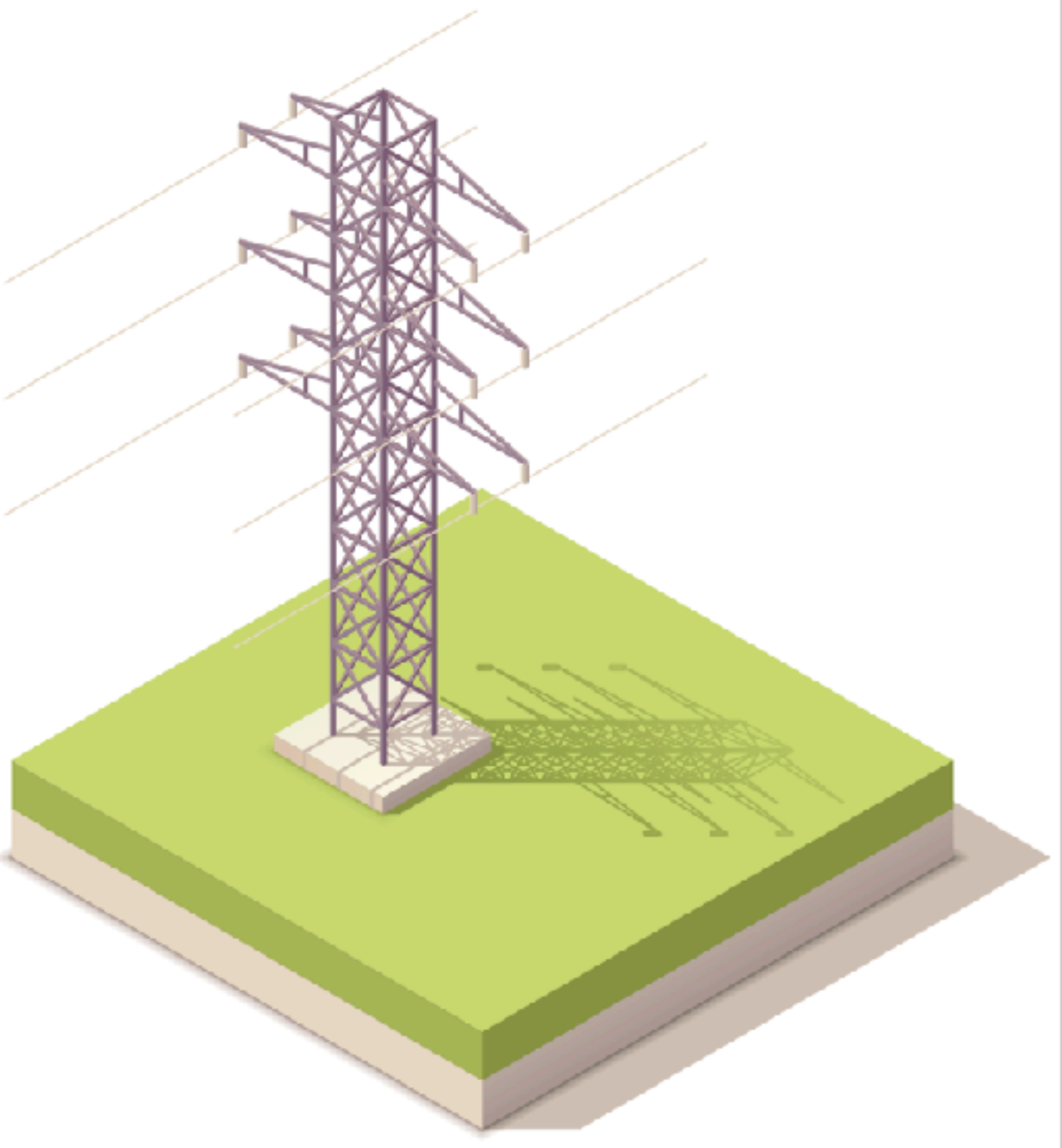
Solid State Drives (SSDs)



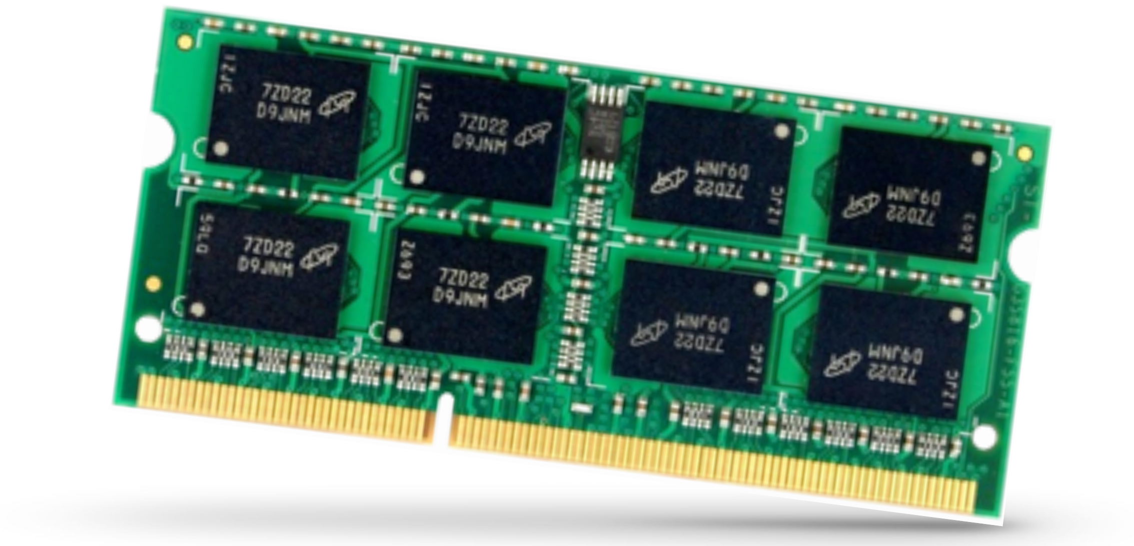
NETWORK

Switches and Wide Area Network (WAN) Backbone



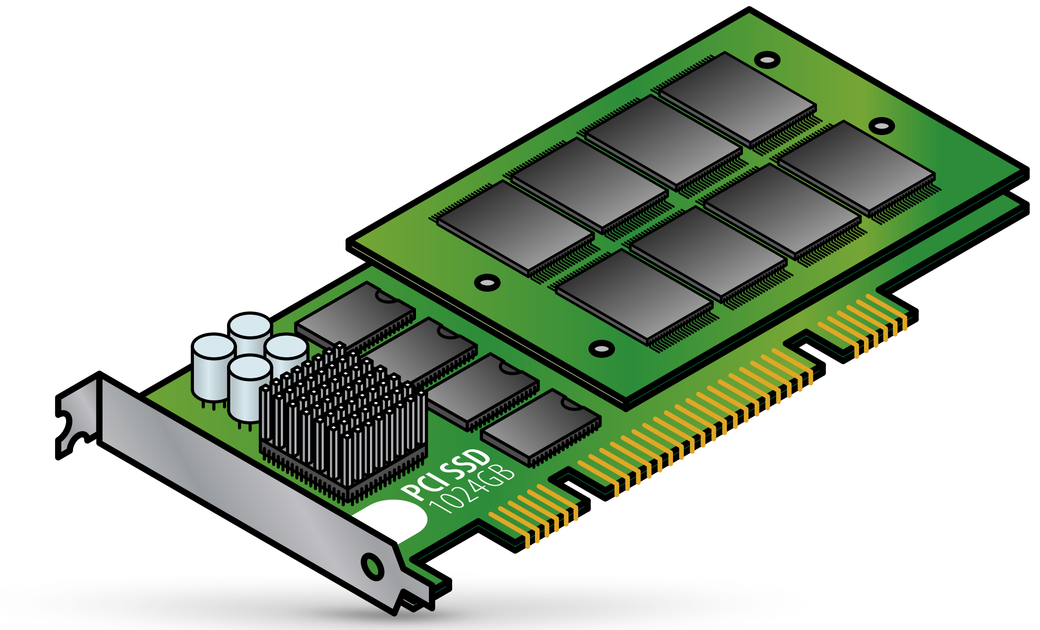


WHY DO DEVICES FAIL?



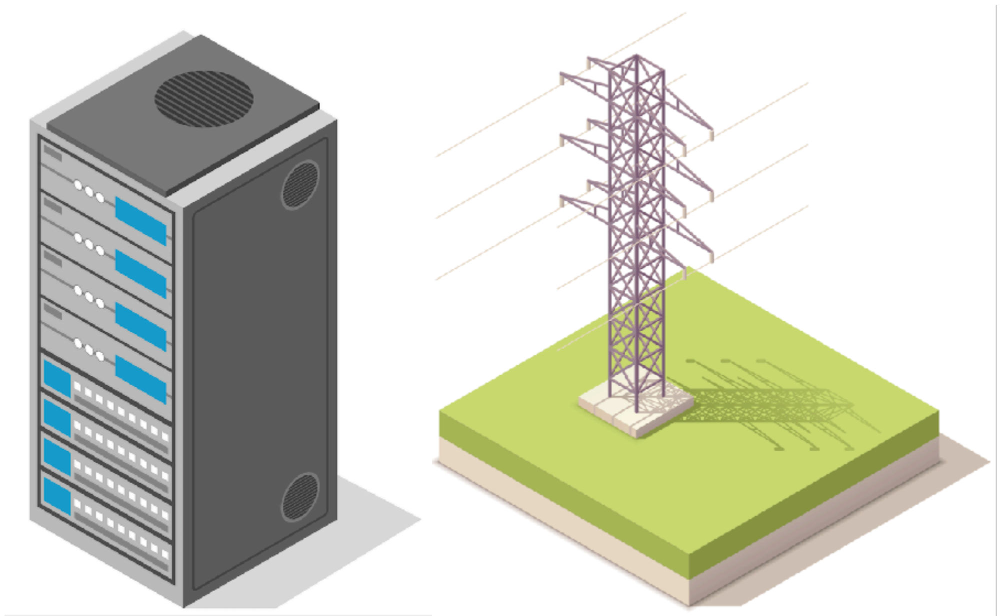
DRAM

- 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*

RELIABILITY EVENTS

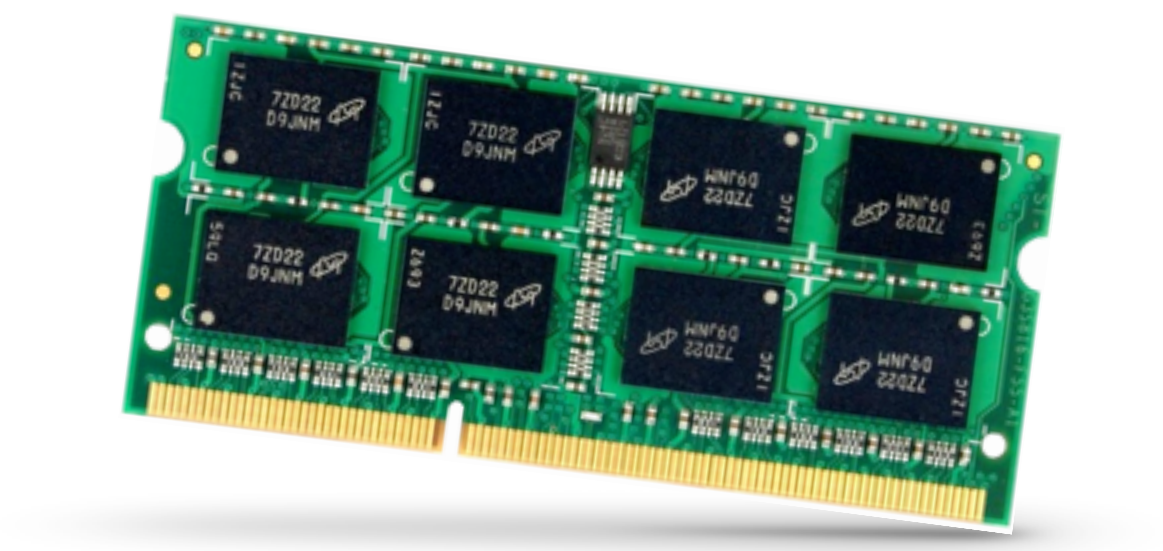
ERROR

- How failures manifest in software using a device

FAULT

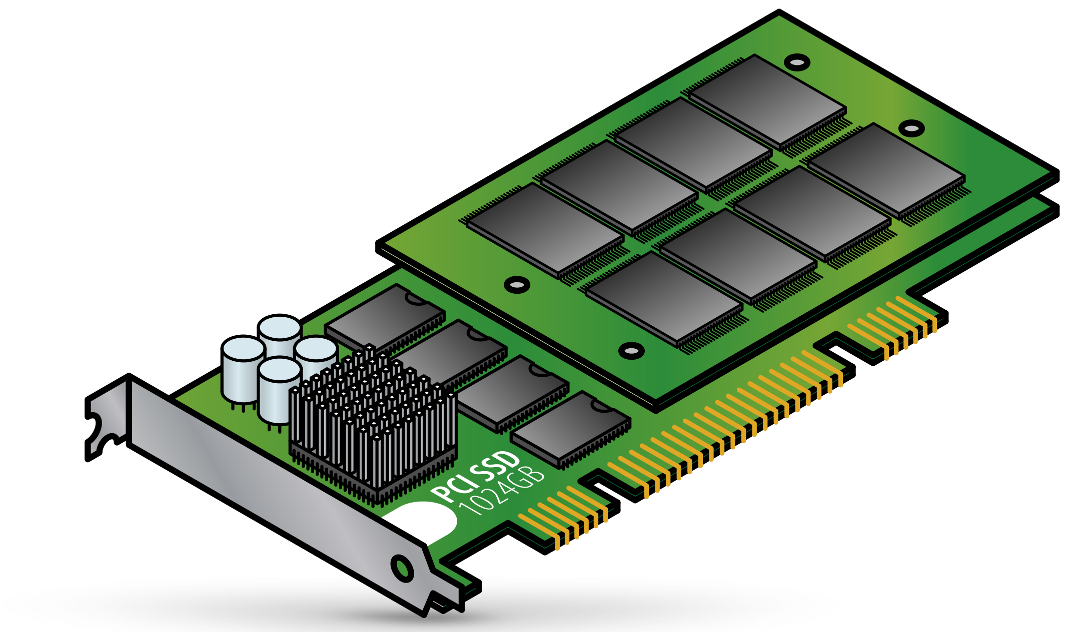
- The underlying reason why a device fails
- ***Permanent:*** the fault appears every time
- ***Transient:*** the appears only sometimes

LARGE SCALE STUDIES



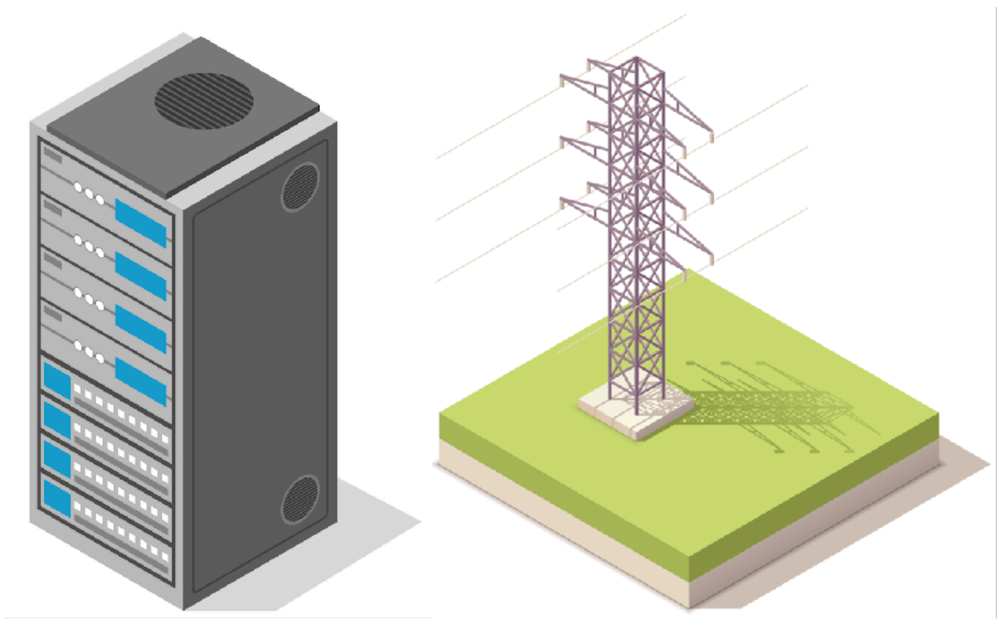
DRAM

[DSN '15]



SSDs

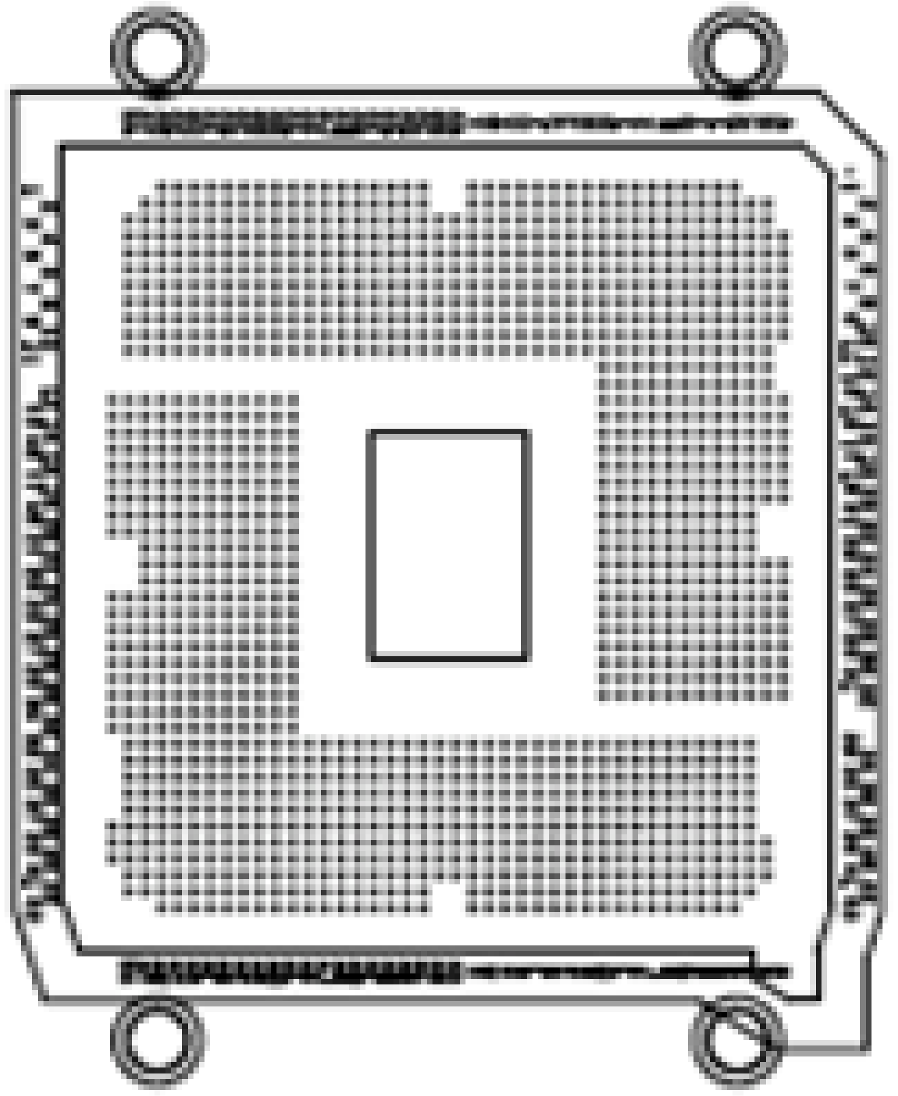
[SIGMETRICS '15]



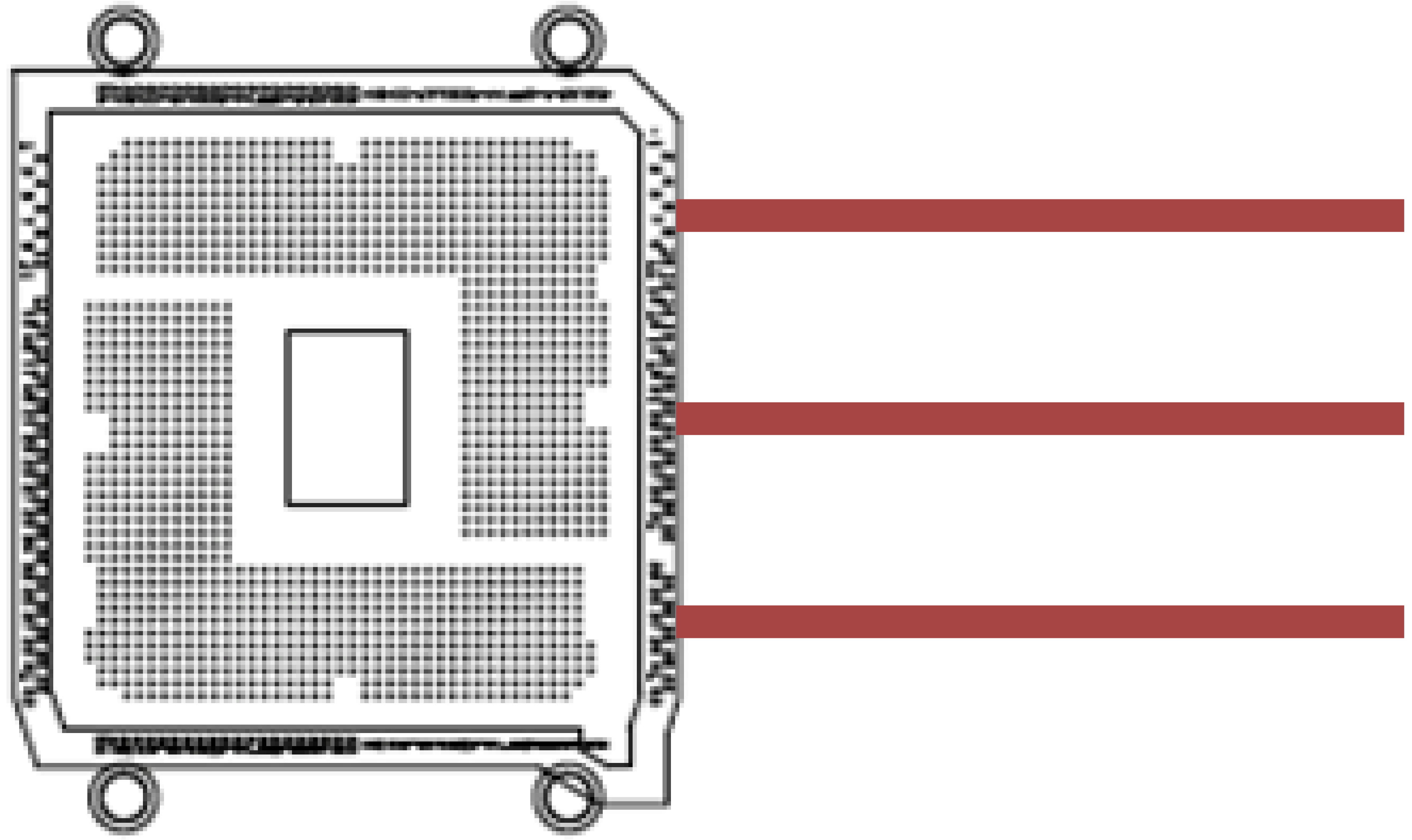
Networks

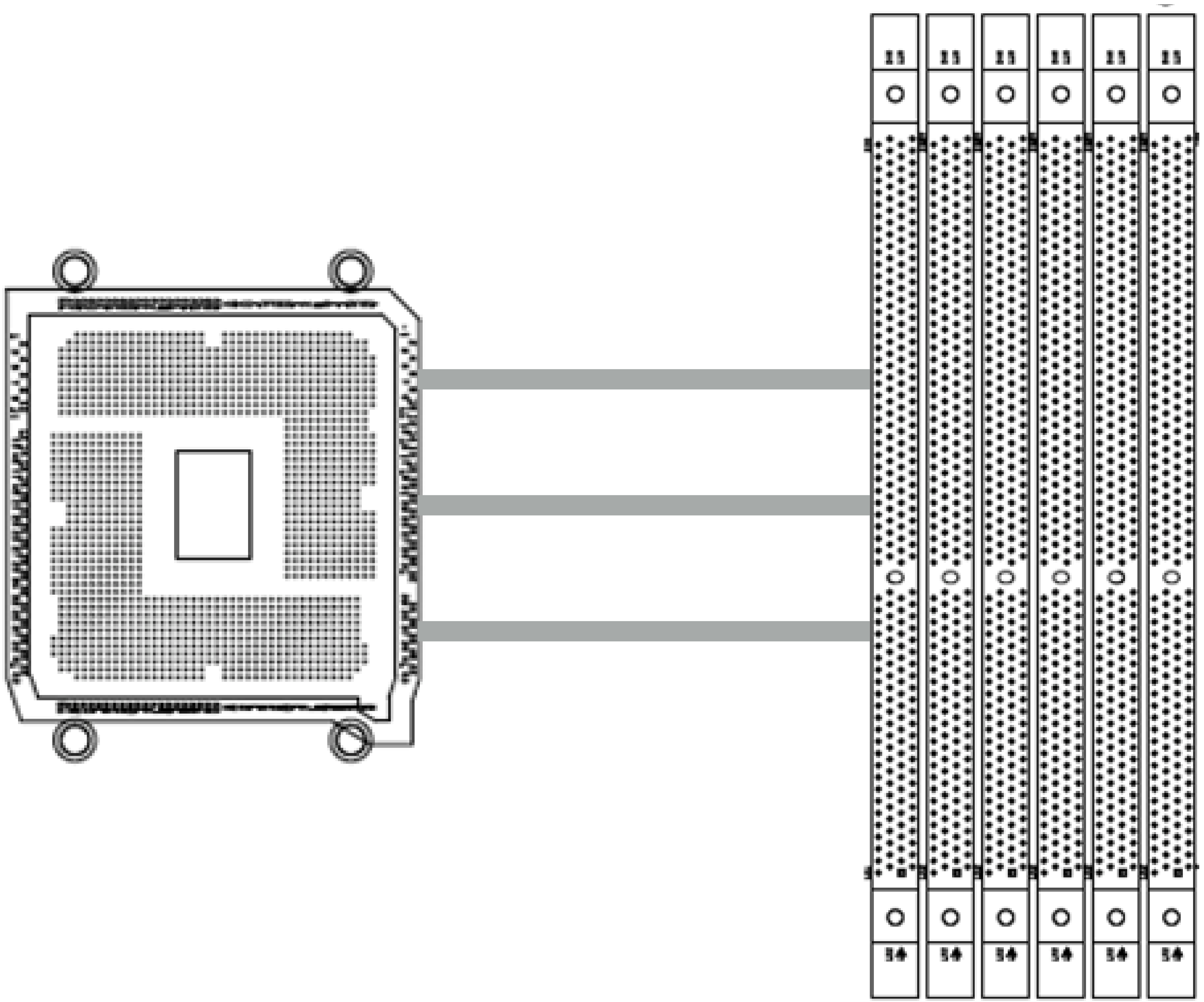
[IMC '18]

Socket



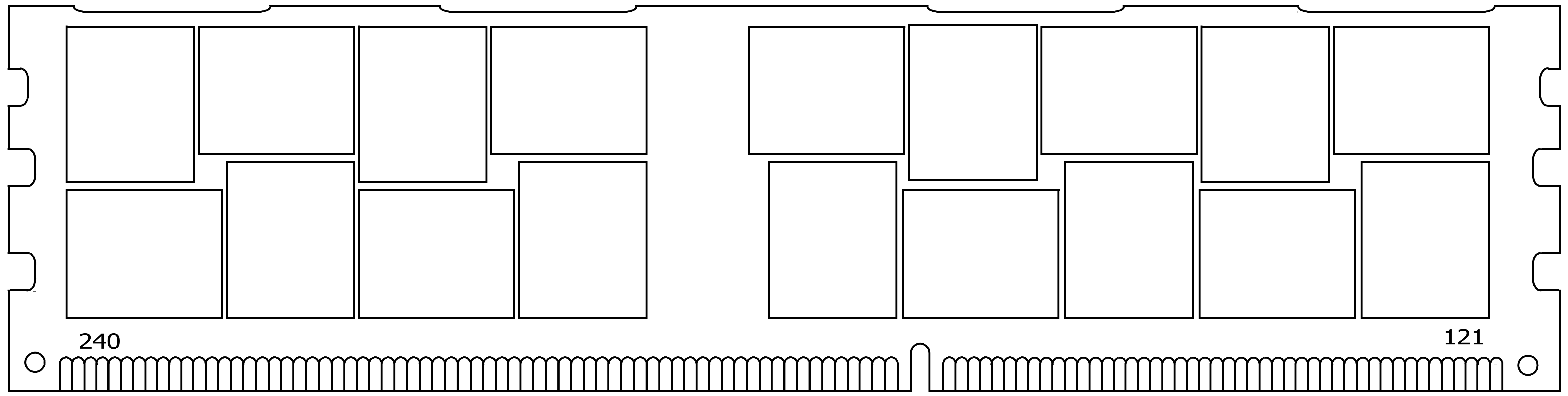
Memory channels



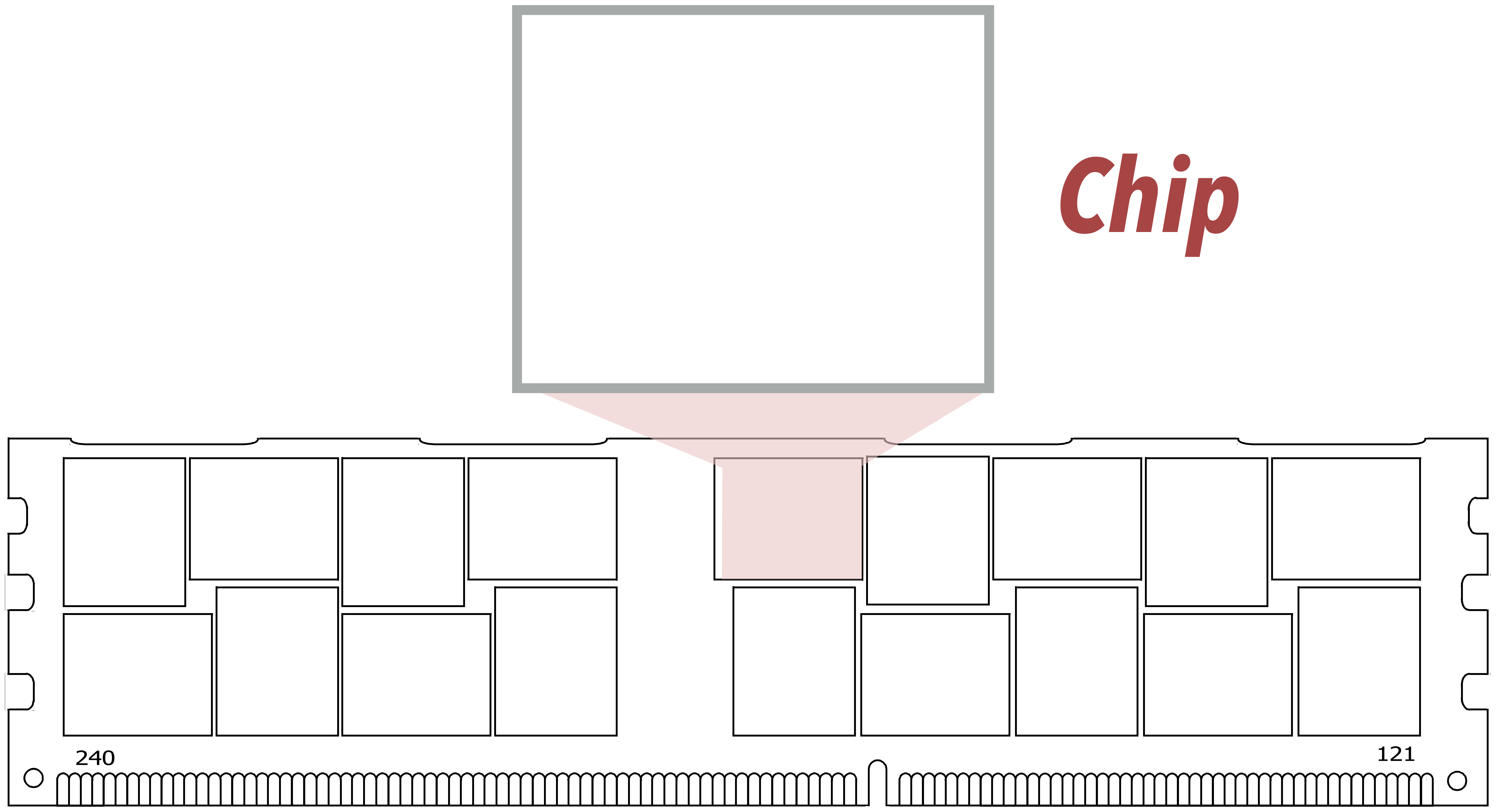


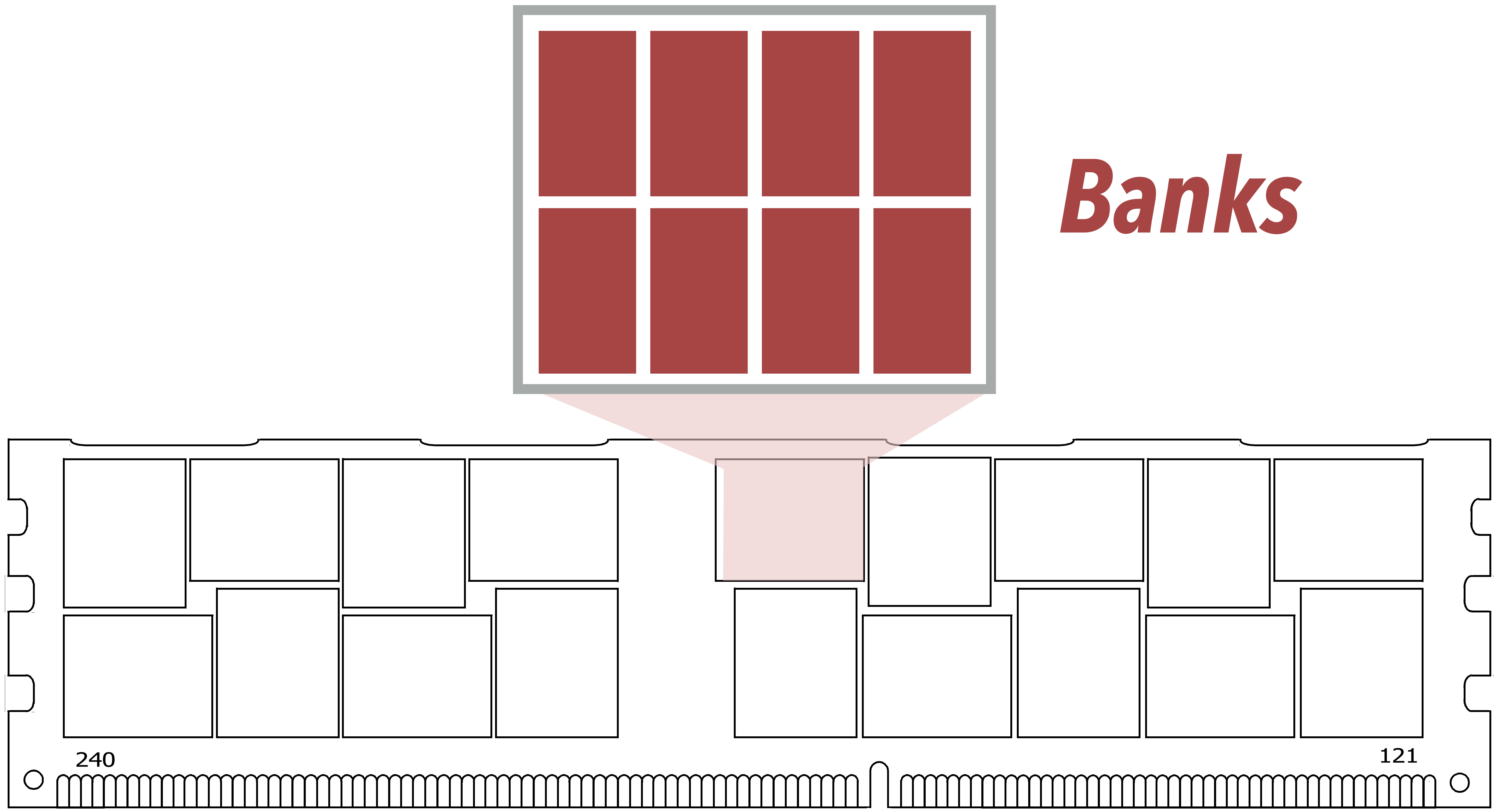
Dual In-line Memory Module (DIMM) slots

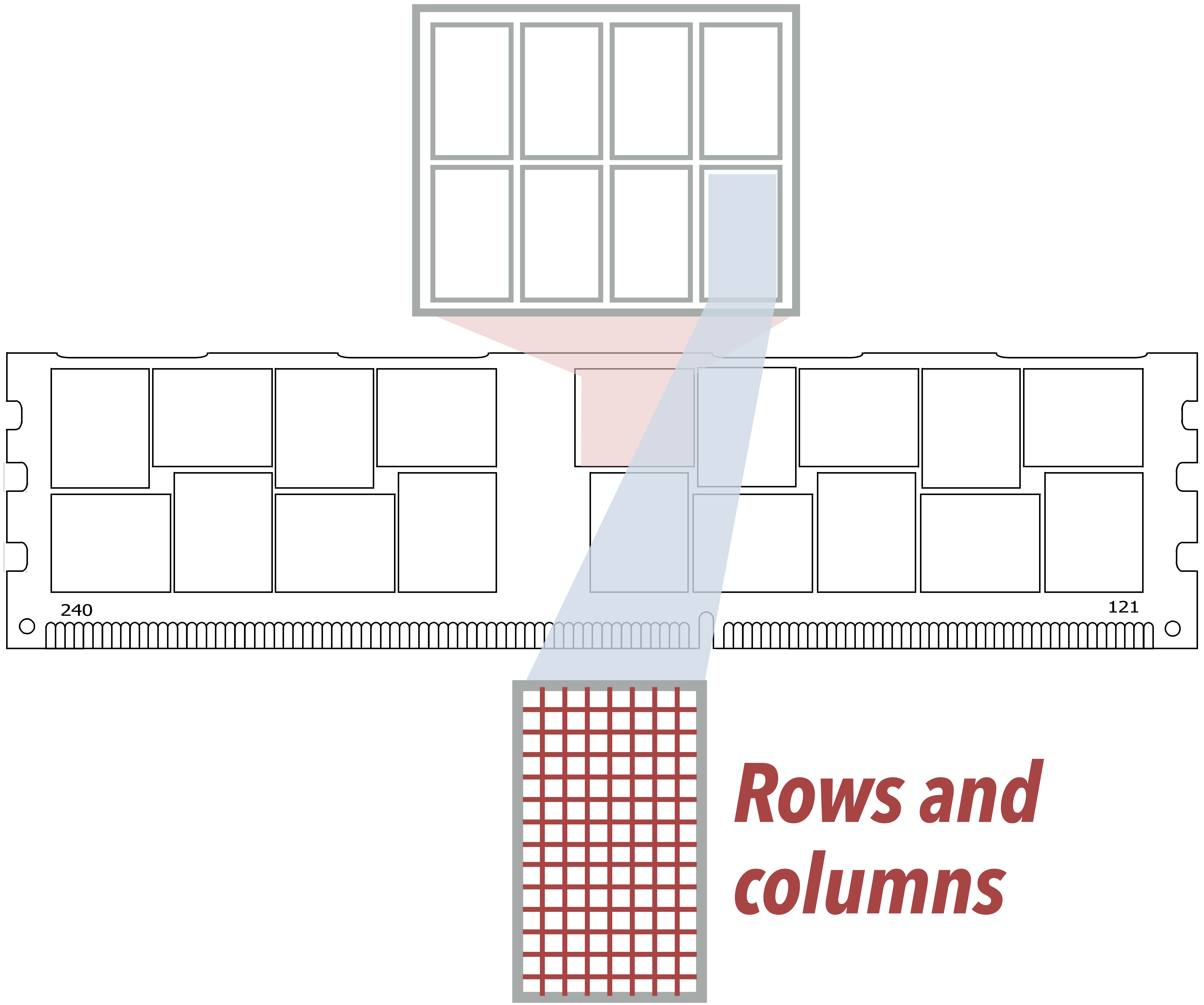
DIMM



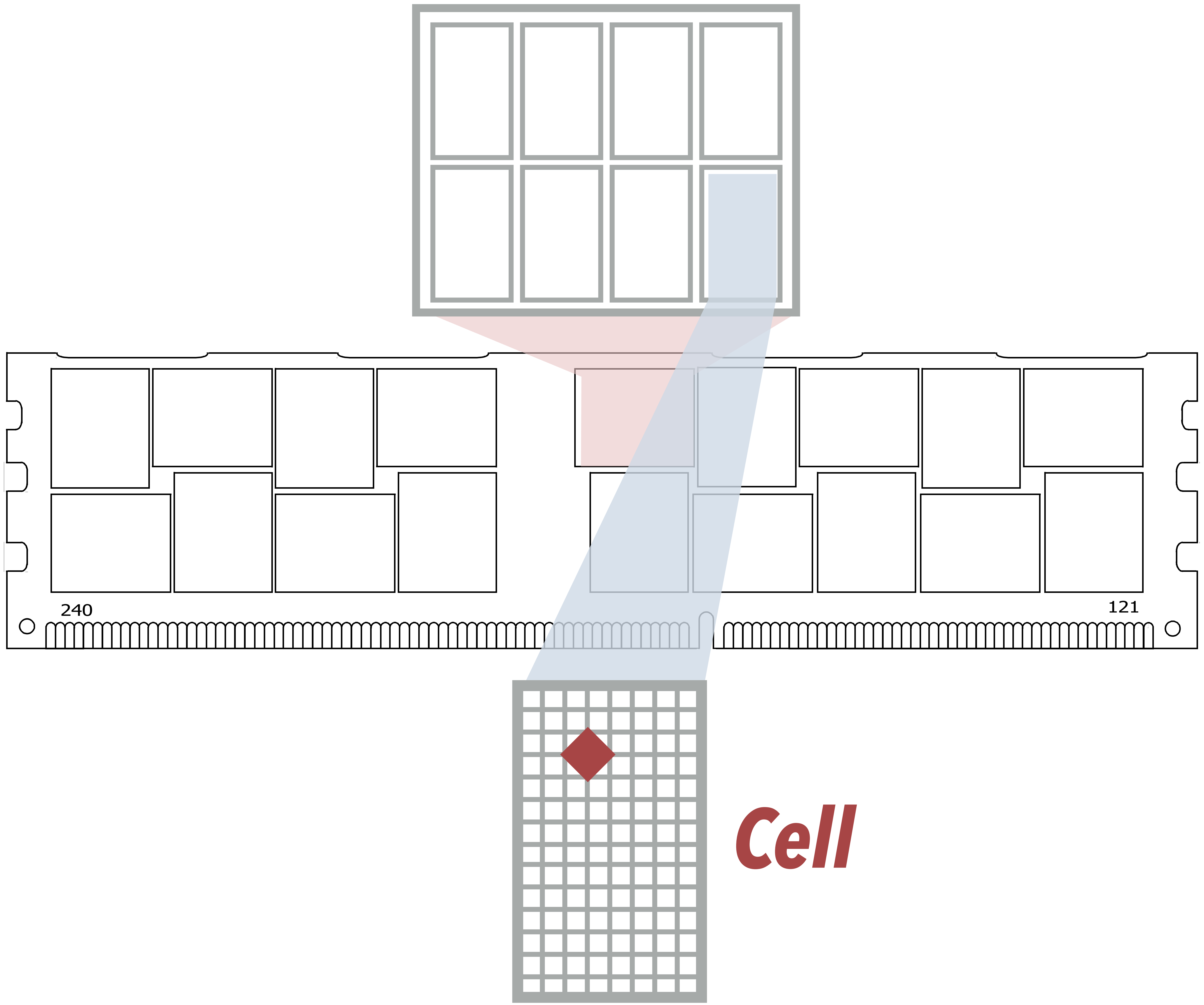
Chip



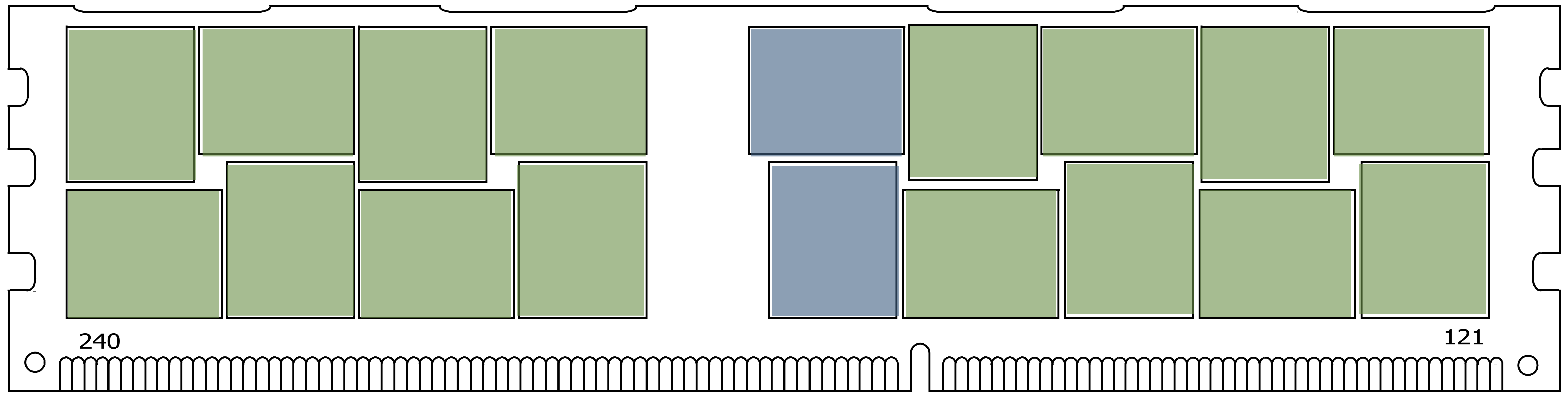

Banks
240
121



Rows and columns


240
121
Cell

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***

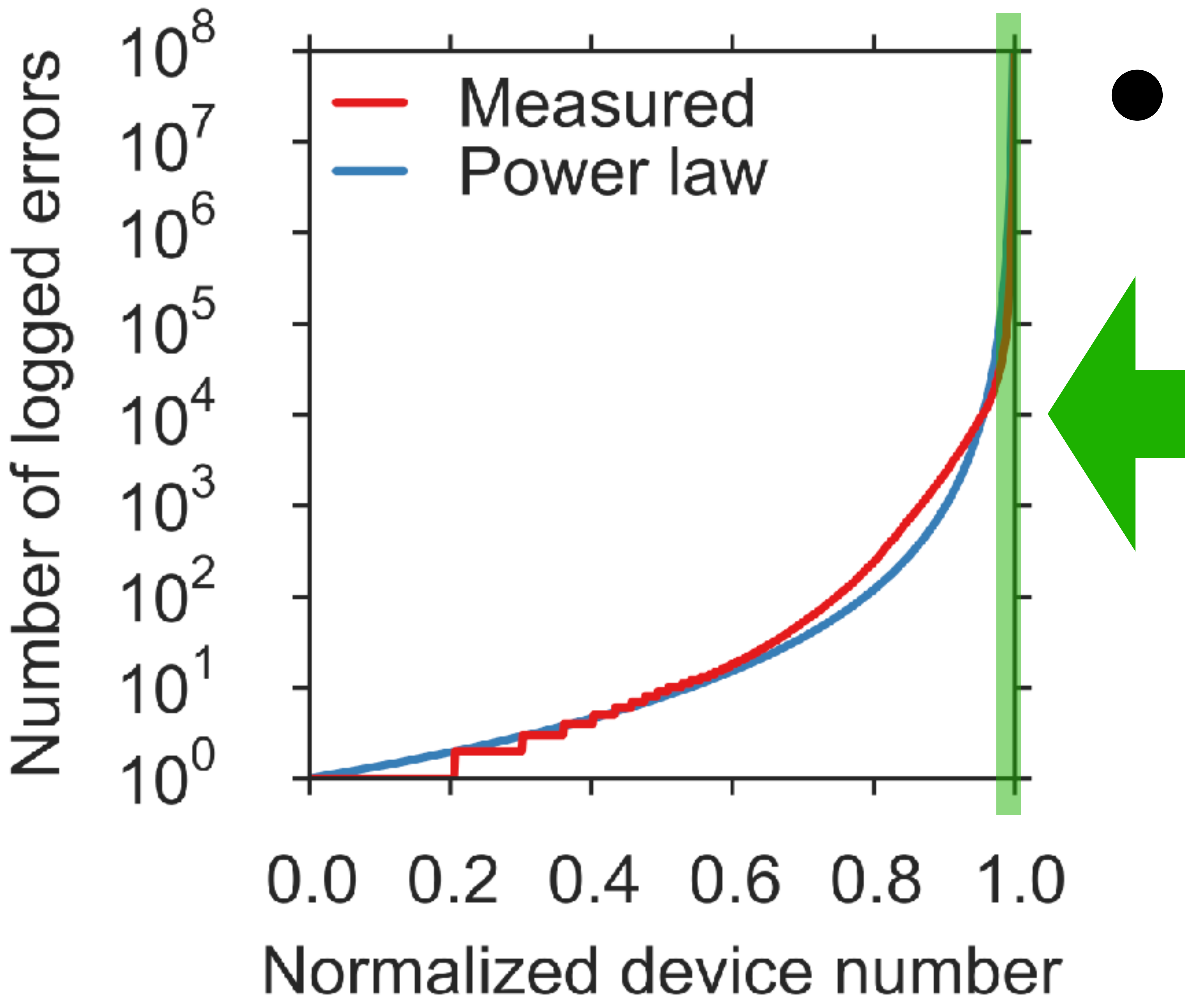
ANALYTICAL METHODOLOGY

- ***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

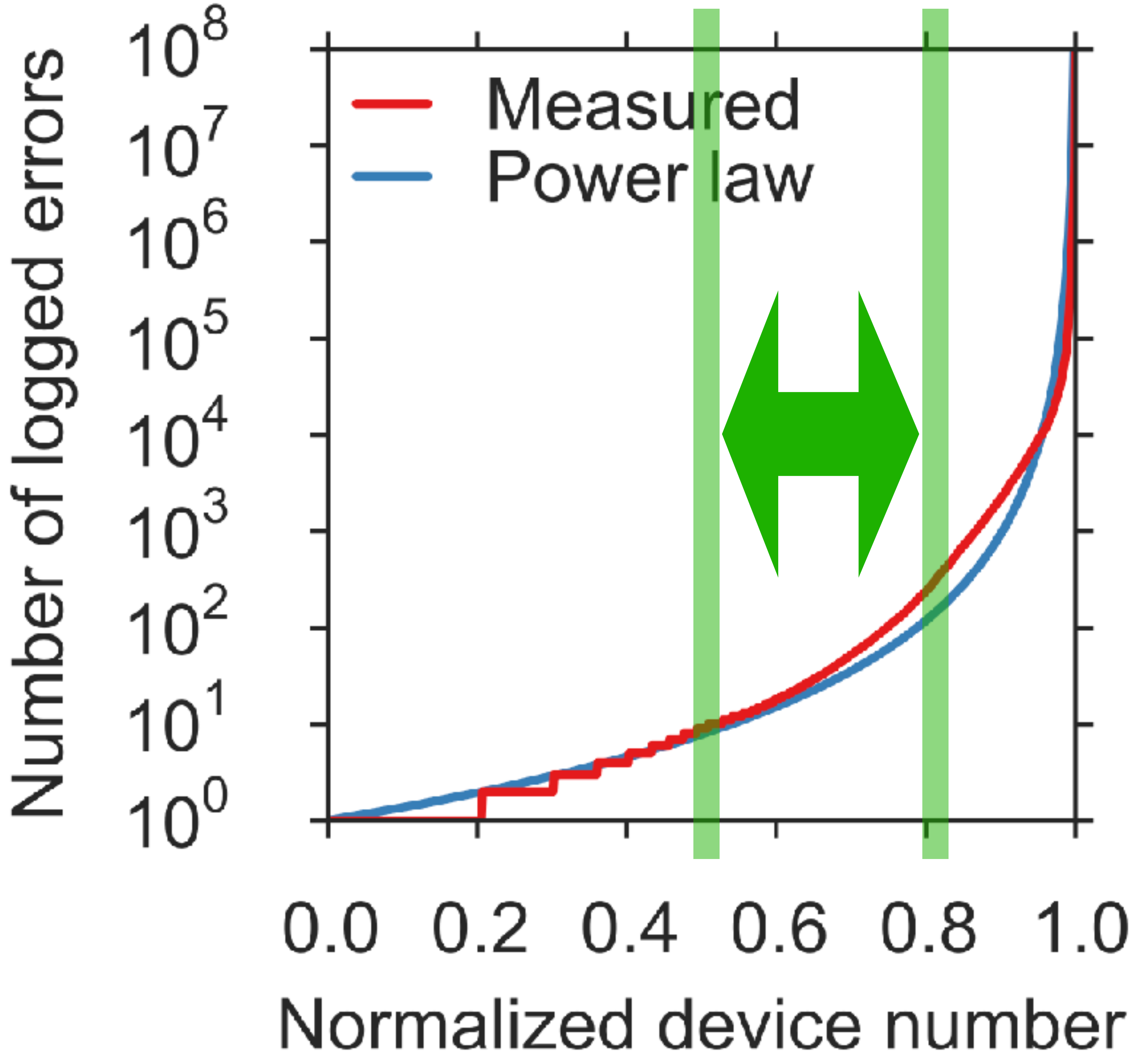
- Errors follow a power-law distribution
- Denial of service due to socket/channel
- Higher density = more failures
- DIMM architectural effects on reliability
- Workload influence on failures
- Model, page-offlining, page randomization

POWER-LAW DISTRIBUTION



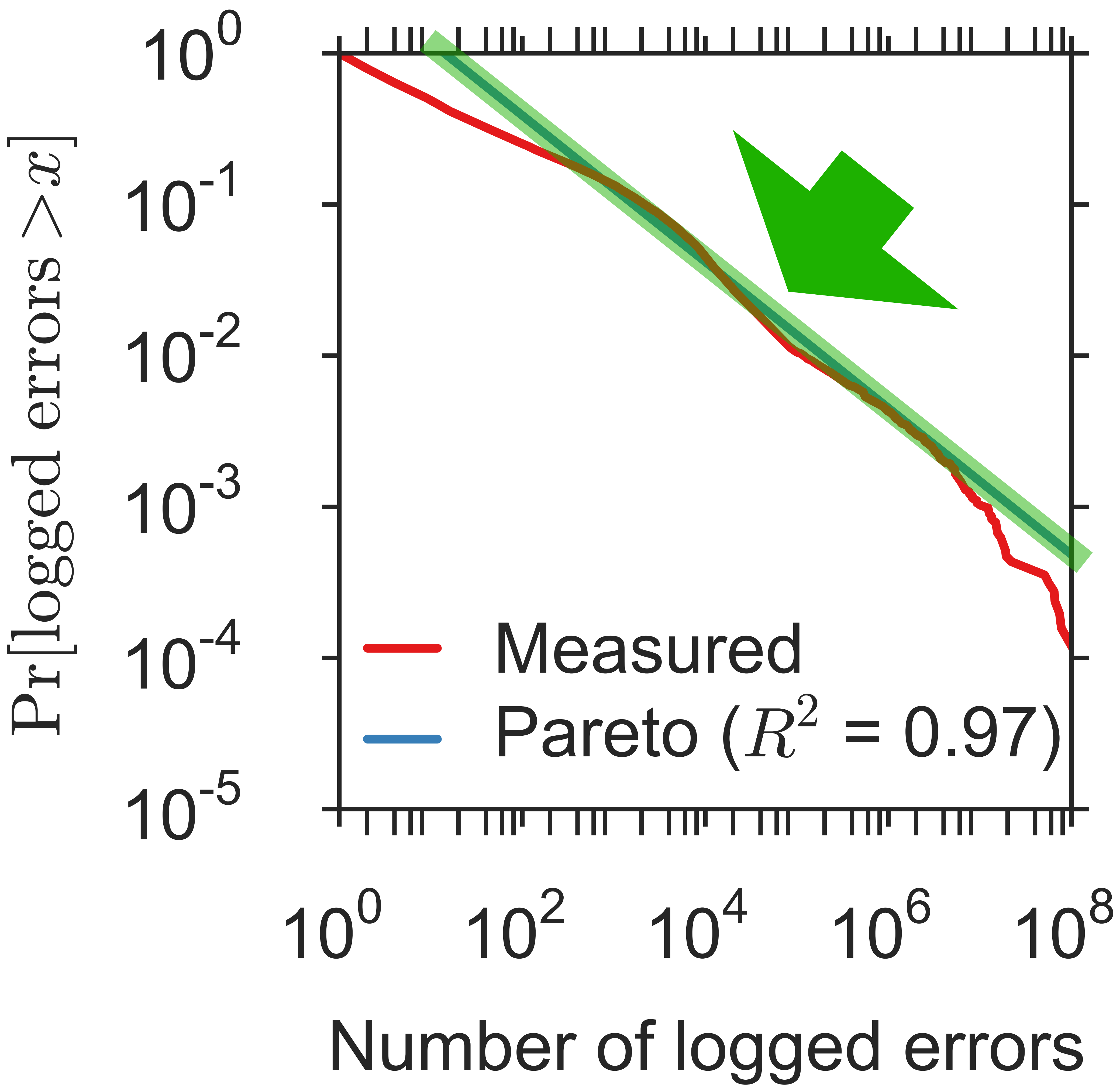
- 1% of servers = 97.8% errors

POWER-LAW DISTRIBUTION



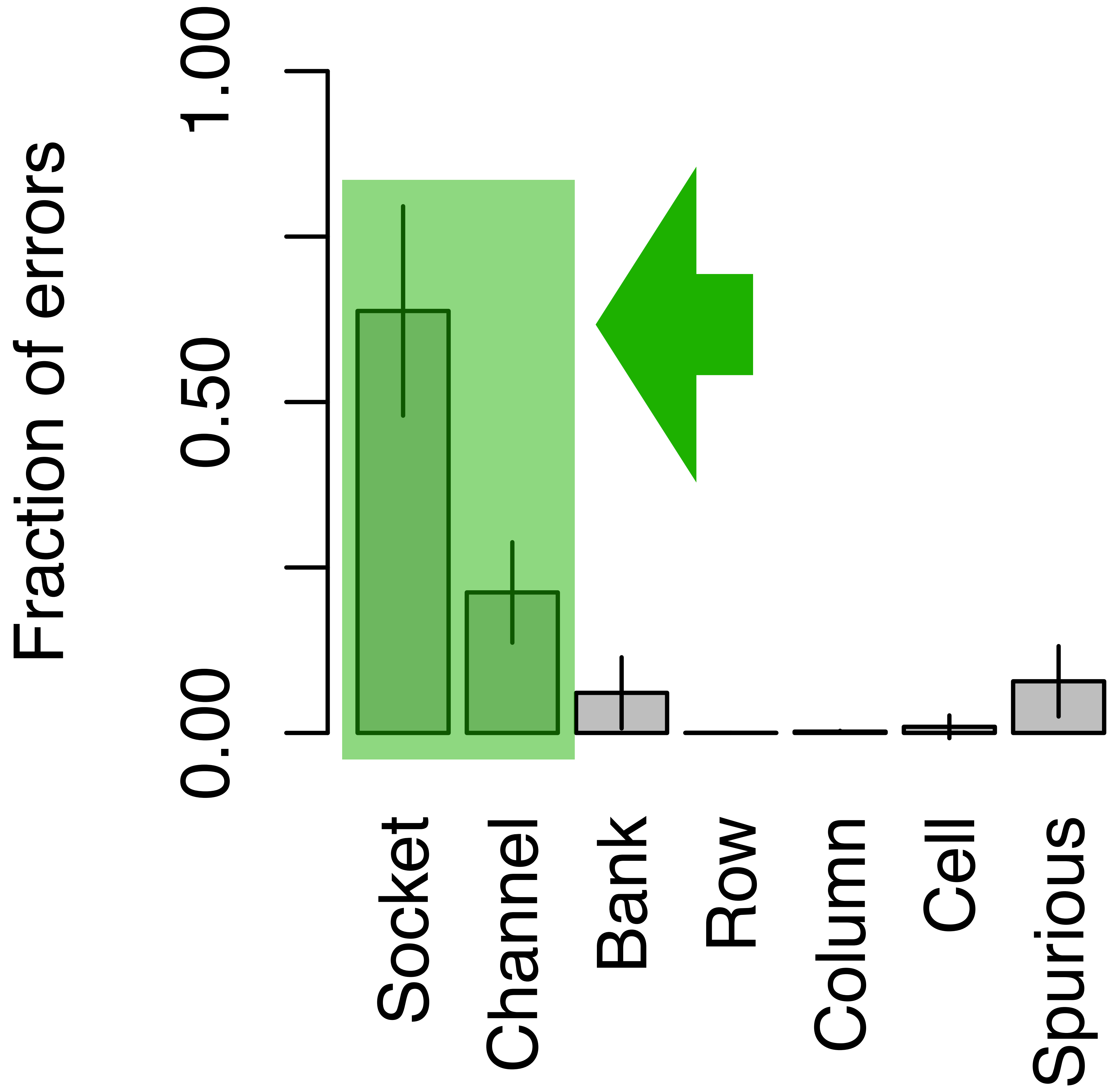
- 1% of servers = 97.8% errors
- Average is 55X median

POWER-LAW DISTRIBUTION



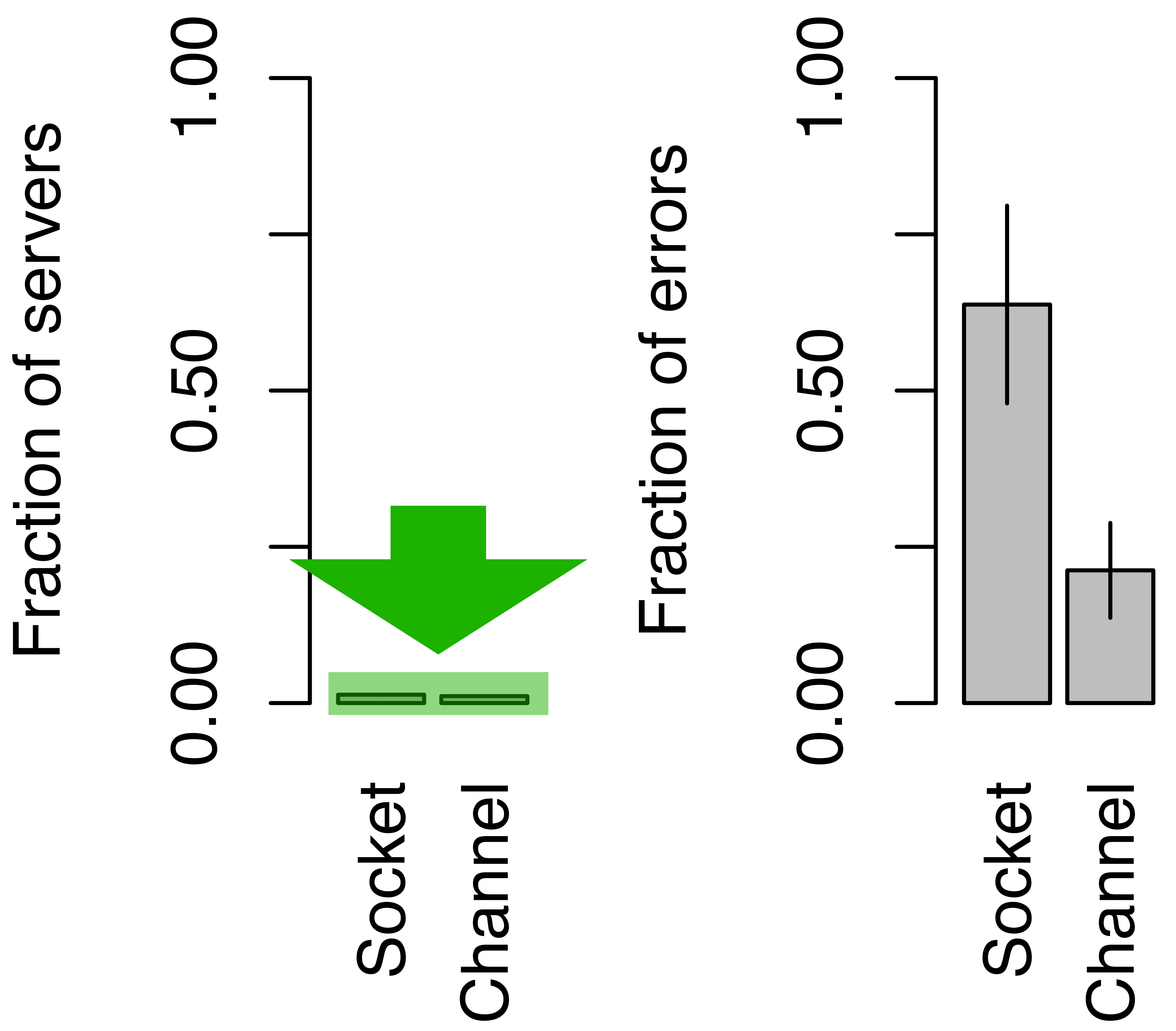
- 1% of servers = 97.8% errors
- Average is 55X median
- Pareto distribution fits
  - Devices without errors tend to stay without errors

SOCKET/CHANNEL ERRORS



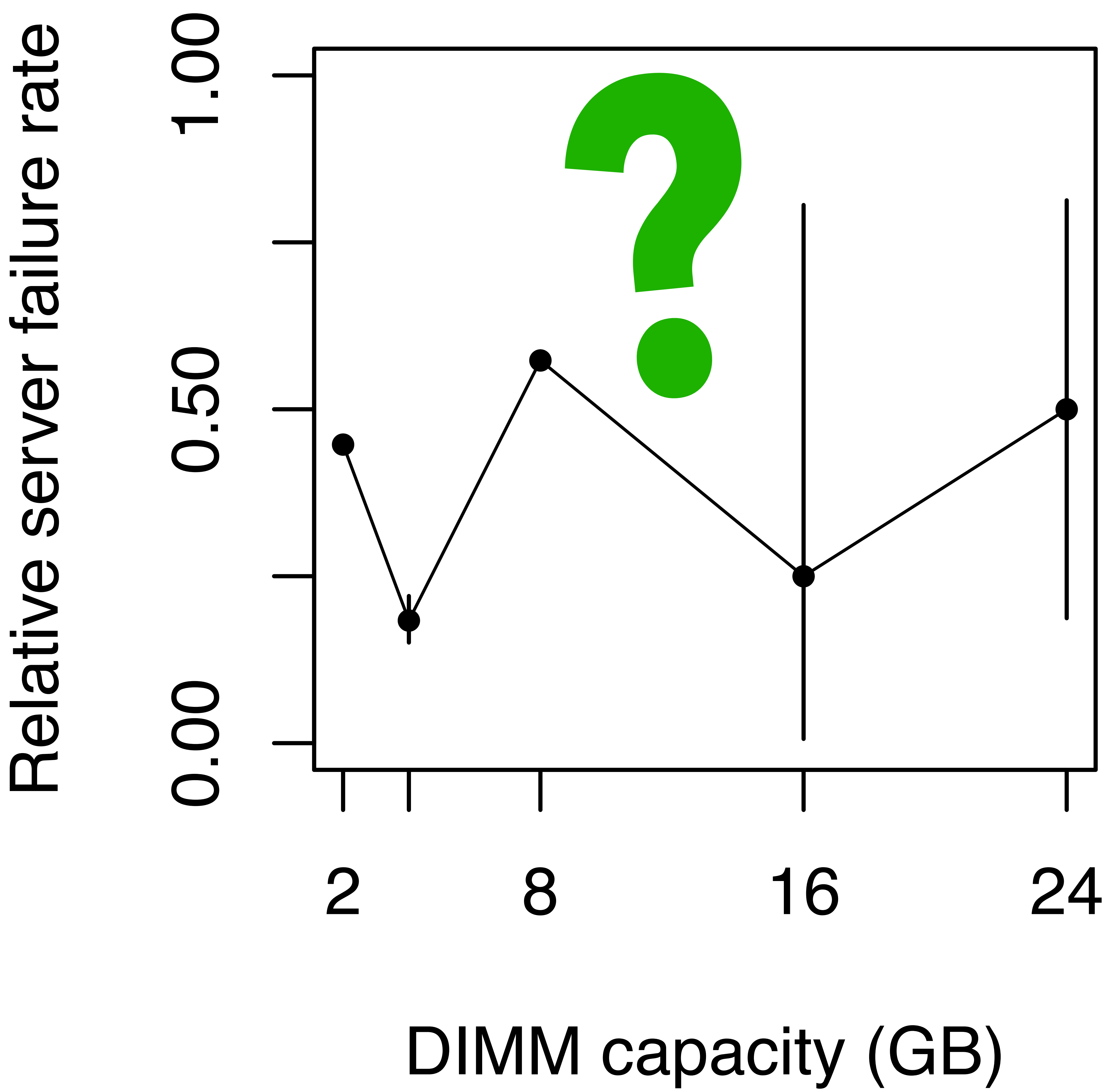
- Contribute majority of errors

SOCKET/CHANNEL ERRORS



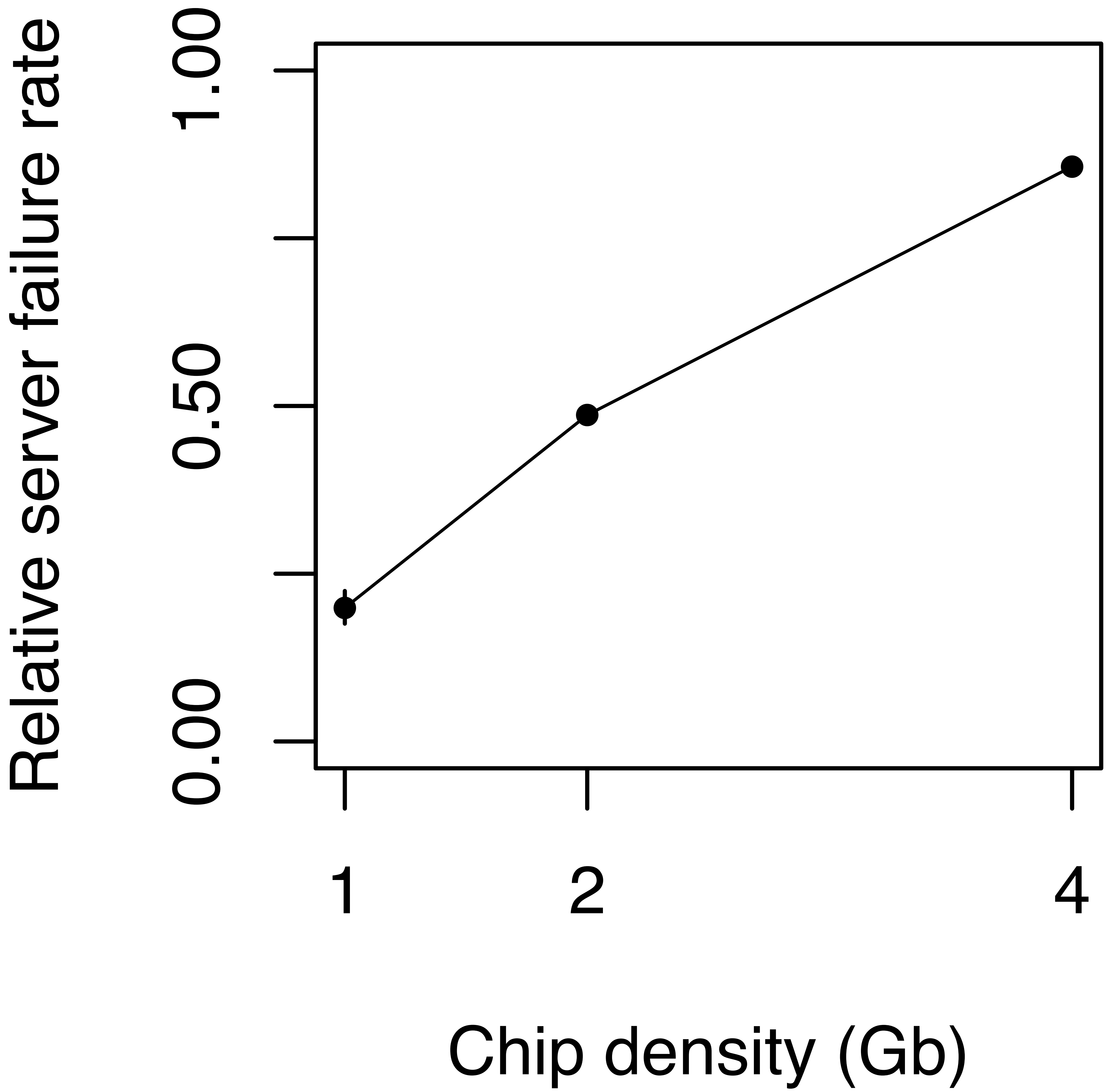
- Contribute majority of errors
- Concentrated on a few hosts
- Symptoms ≈ server DoS

HIGHER DENSITY TRENDS



- Capacity, NO! Density, YES!

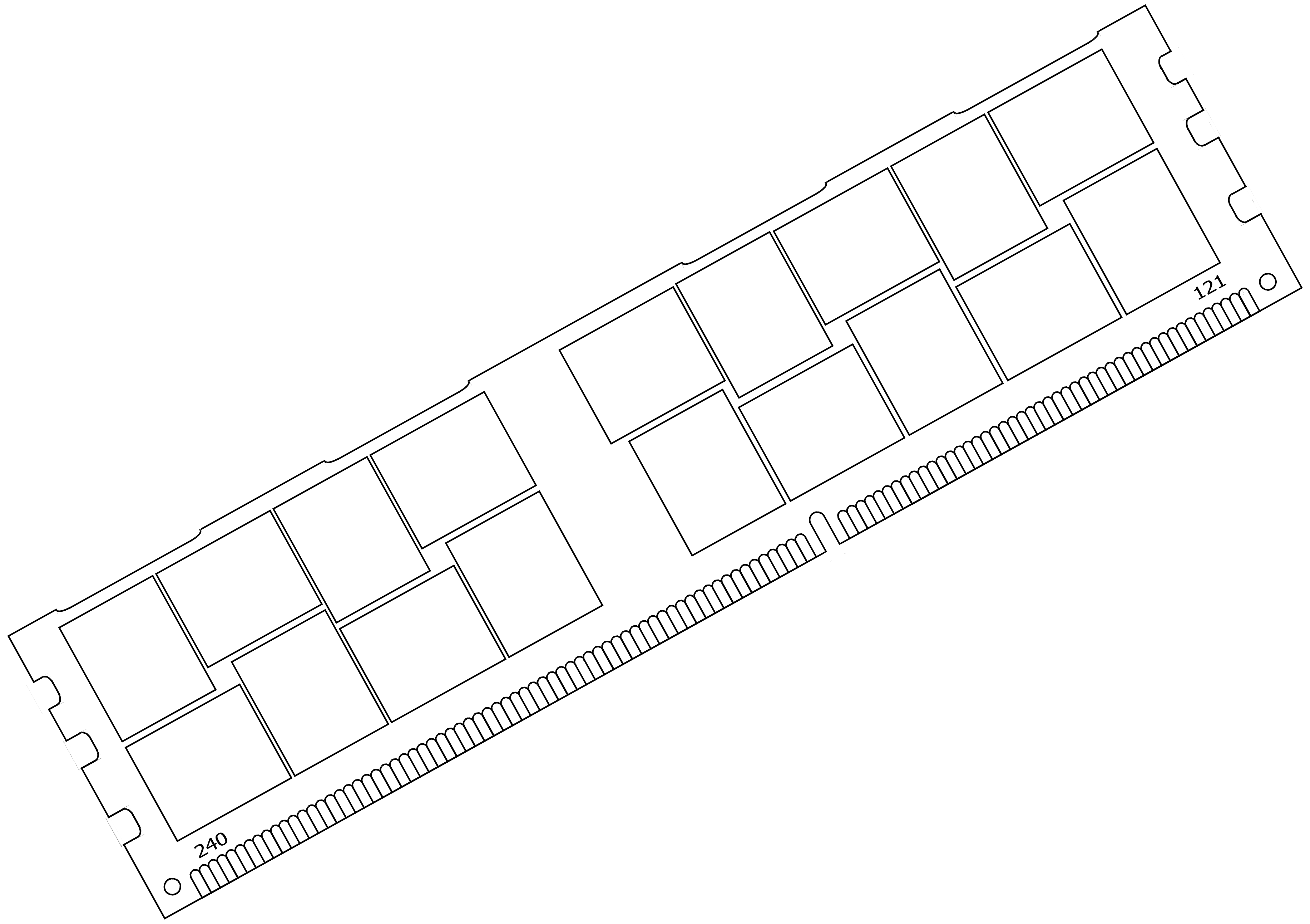
HIGHER DENSITY TRENDS



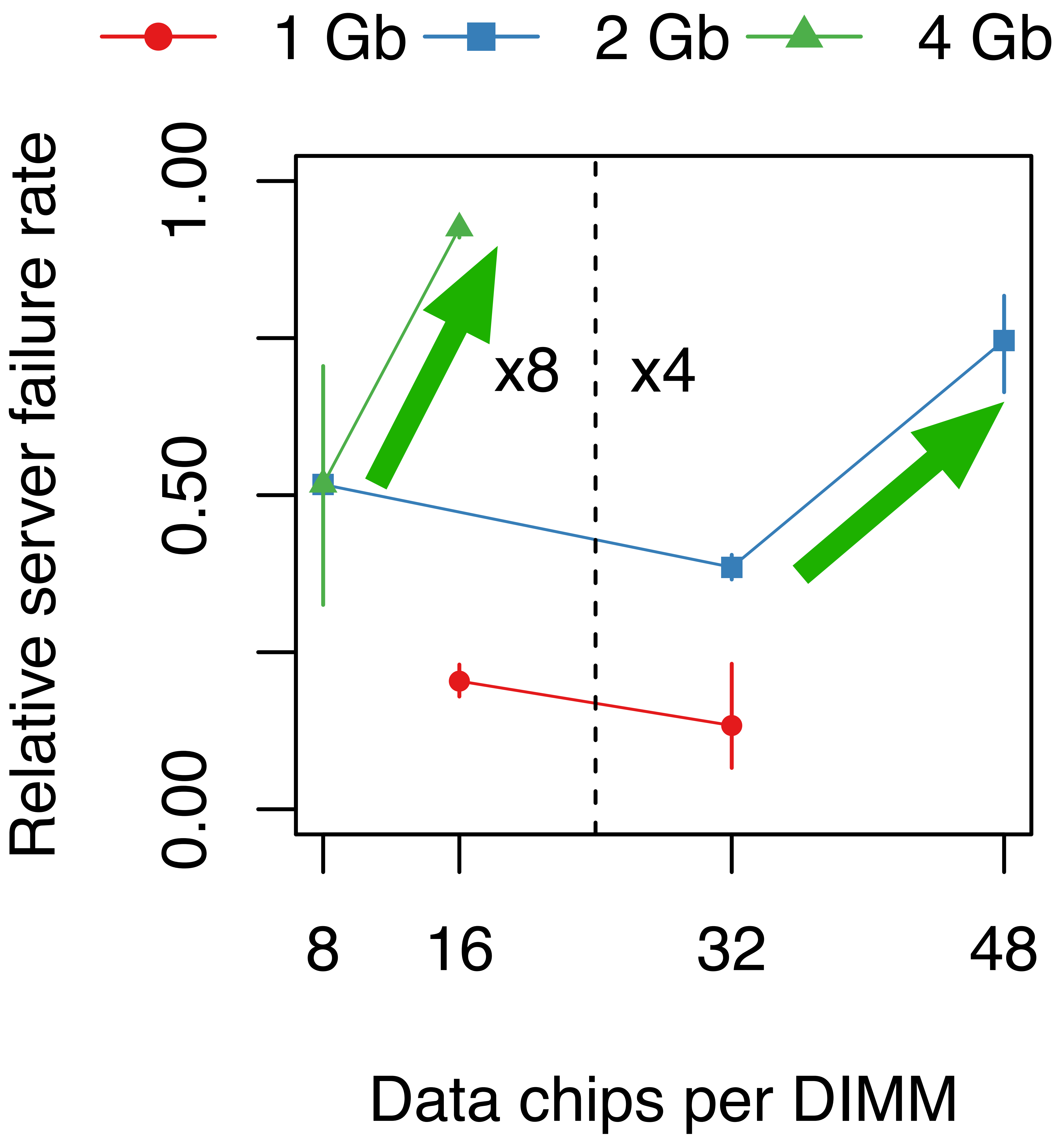
- Capacity, NO! Density, YES!
- Higher density, more failure
  - Due to smaller feature sizes

DIMM architecture

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

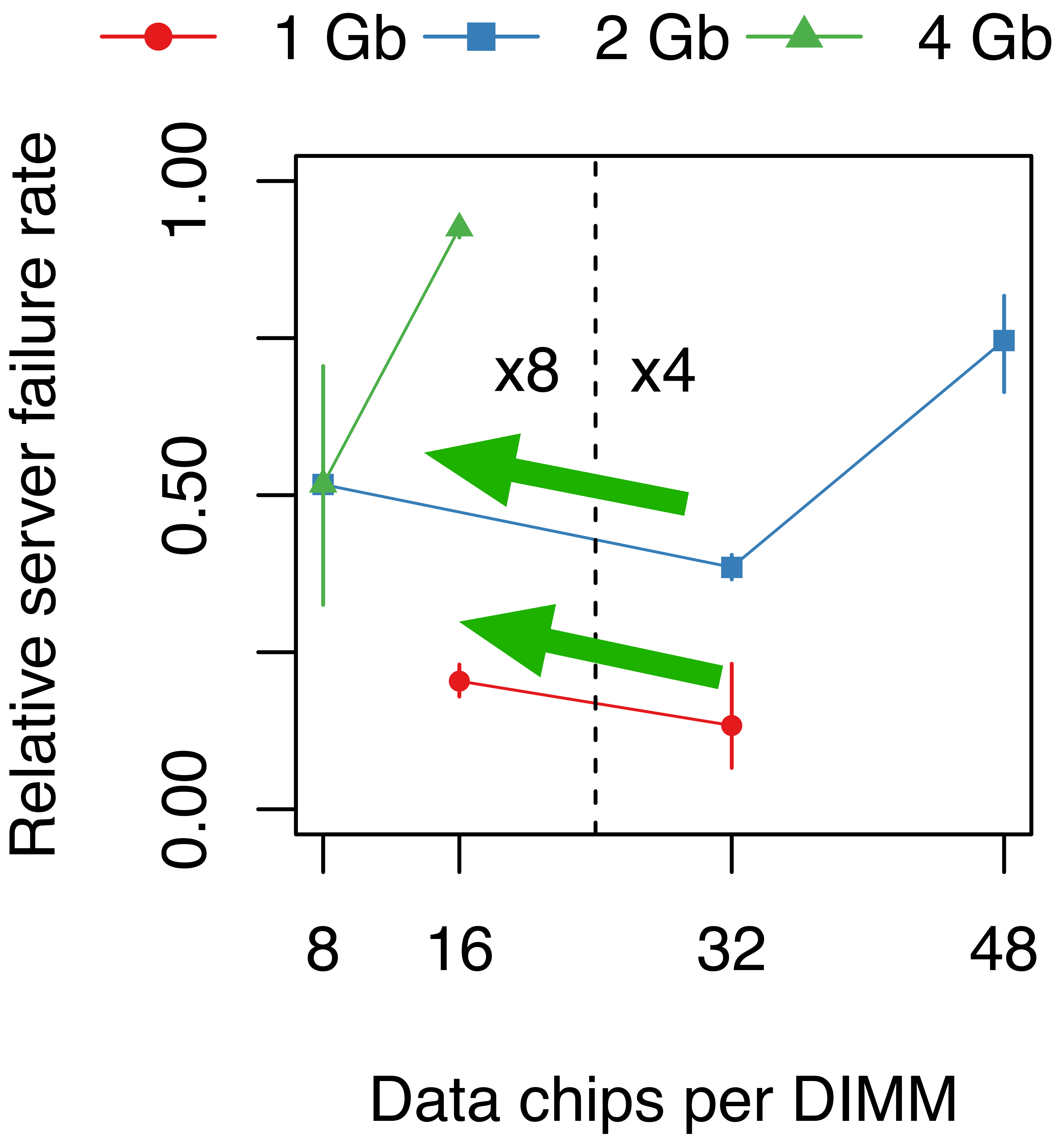


ARCHITECTURAL EFFECTS



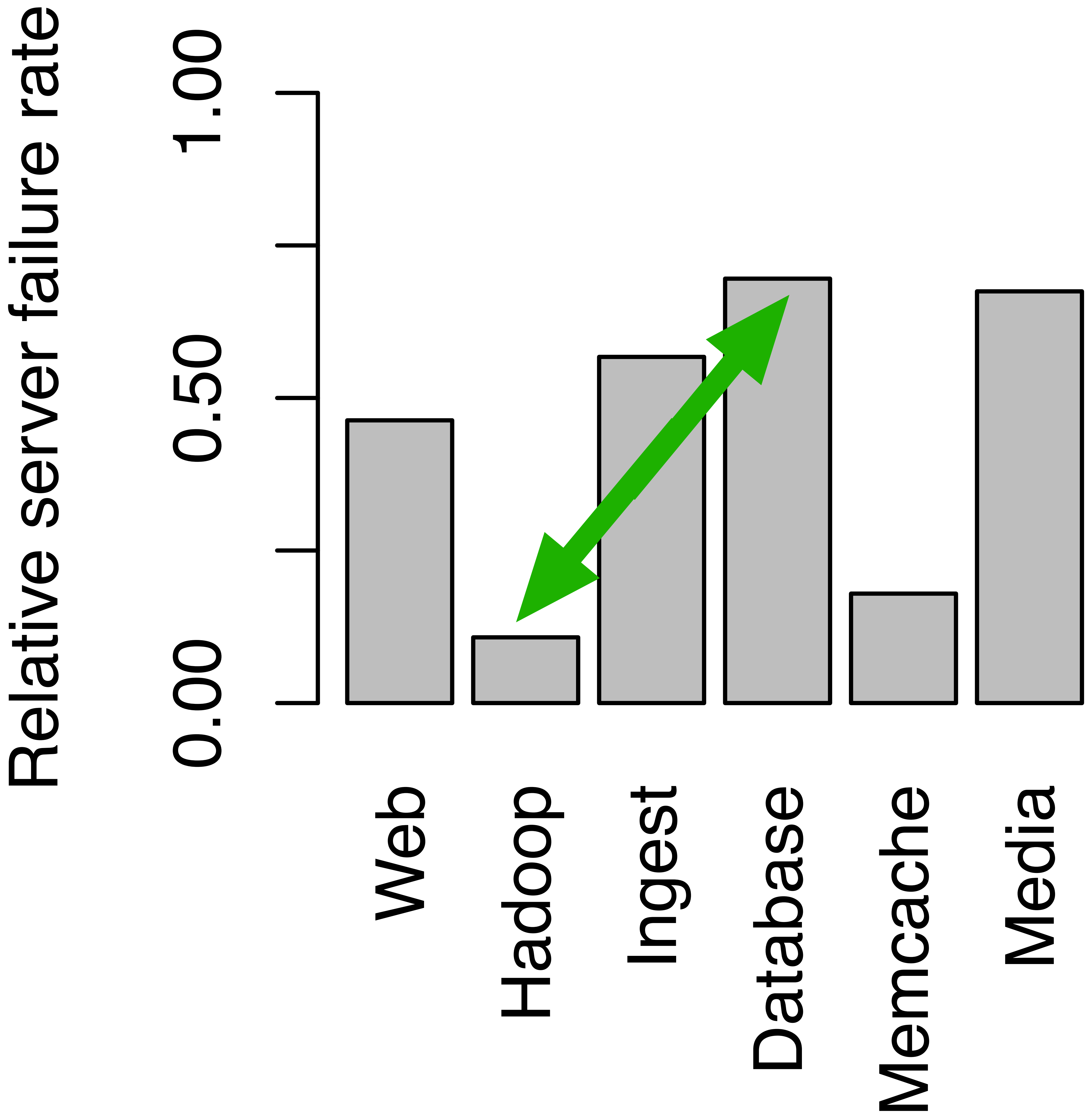
- For the <u>same transfer width</u>:
- More chips = more failures

ARCHITECTURAL EFFECTS



- For the <u>same transfer width</u>:
- More chips = more failures
- For <u>different transfer widths</u>:
- More bits = more failures
  - Likely related to electrical noise

WORKLOAD INFLUENCE



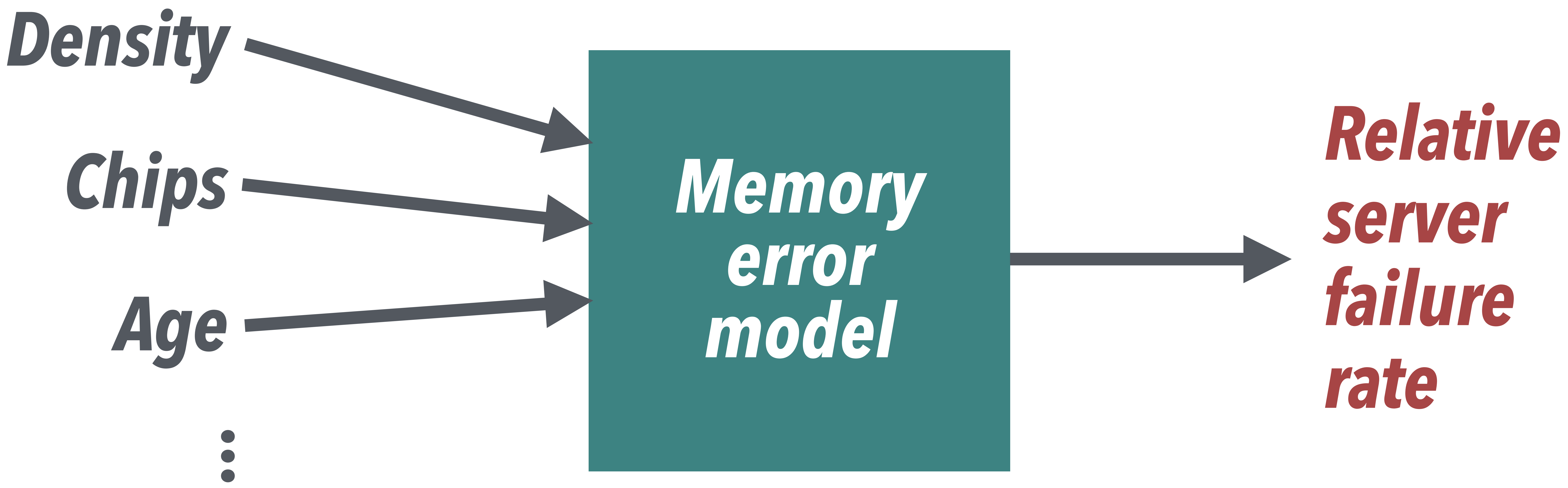
- No consistent trends across CPU and memory utilization
- But workload varies by ~6X
  - May be due to distribution for read/write behavior

MODELING MEMORY FAILURES

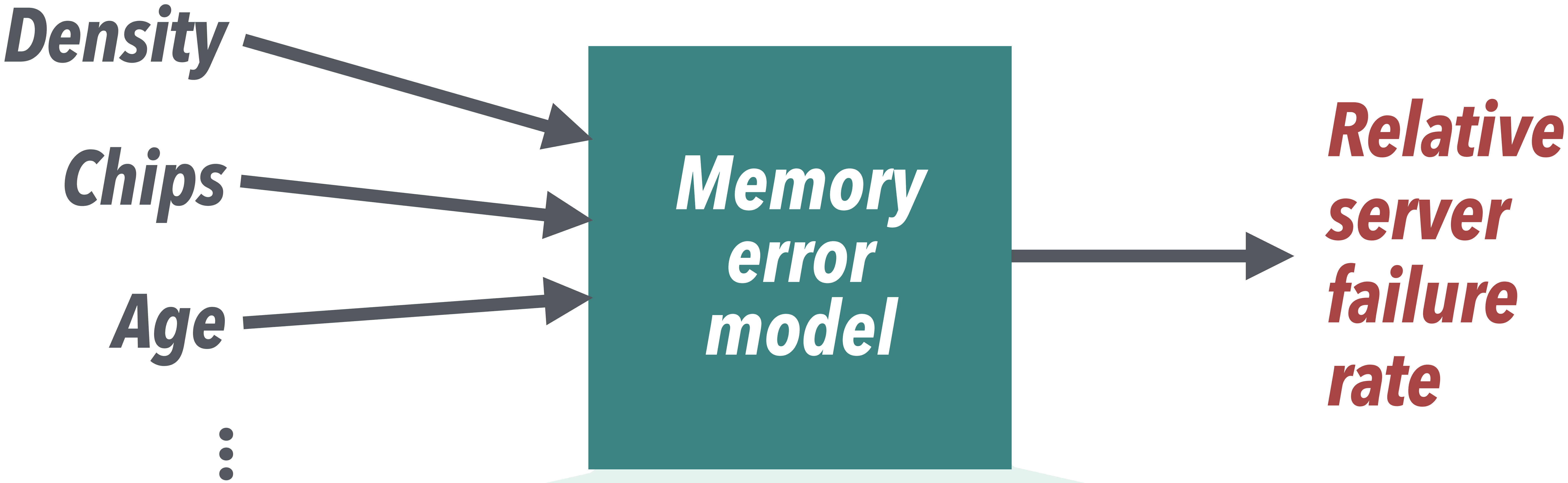
- ***Use statistical regression model***
  - Compare *control group* versus *error group*
  - *Logistic (linear) regression* in R
  - Trained using data from analysis

- Enable *exploratory analysis*

MODELING MEMORY FAILURES



MODELING MEMORY FAILURES



$$\ln\left[\mathcal{F}/(1-\mathcal{F})\right] = \beta_{Intercept} + (Capacity \cdot \beta_{Capacity}) + (Density2Gb \cdot \beta_{Density2Gb}) + (Density4Gb \cdot \beta_{Density4Gb}) + (Chips \cdot \beta_{Chips}) \\ + (CPU\% \cdot \beta_{CPU\%}) + (Age \cdot \beta_{Age}) + (CPUs \cdot \beta_{CPUs})$$

EXPLORATORY ANALYSIS

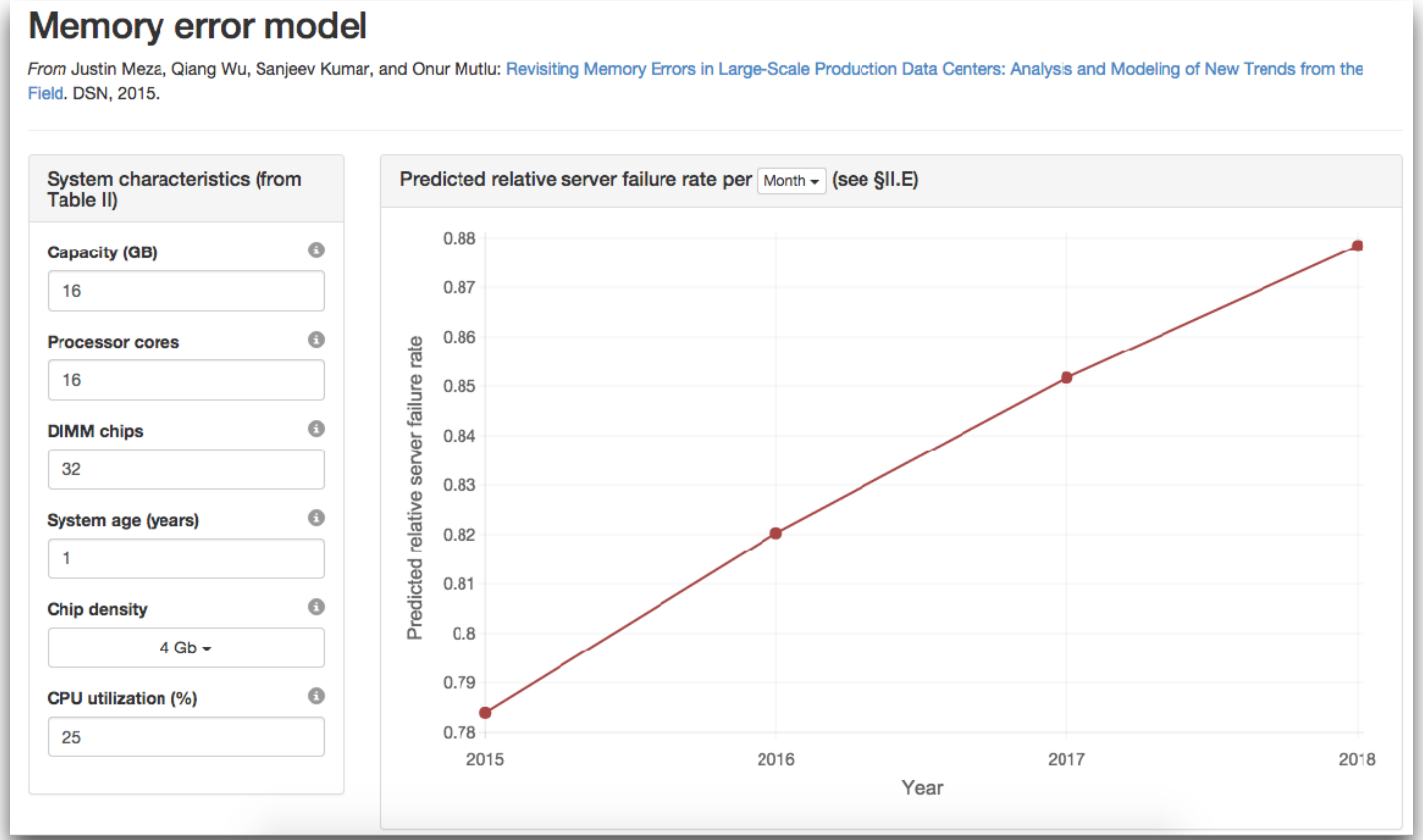
Factor	Low-end	High-end (HE)
Capacity	4 GB	16 GB
Density2Gb	1	0
Density4Gb	0	1
Chips	16	32
CPU%	50%	25%
Age	1	1
CPUs	8	16
Predicted relative failure rate	**0.12**	**0.78**

Inputs

Output

6.5X difference in yearly failures

TOOL AVAILABLE ONLINE

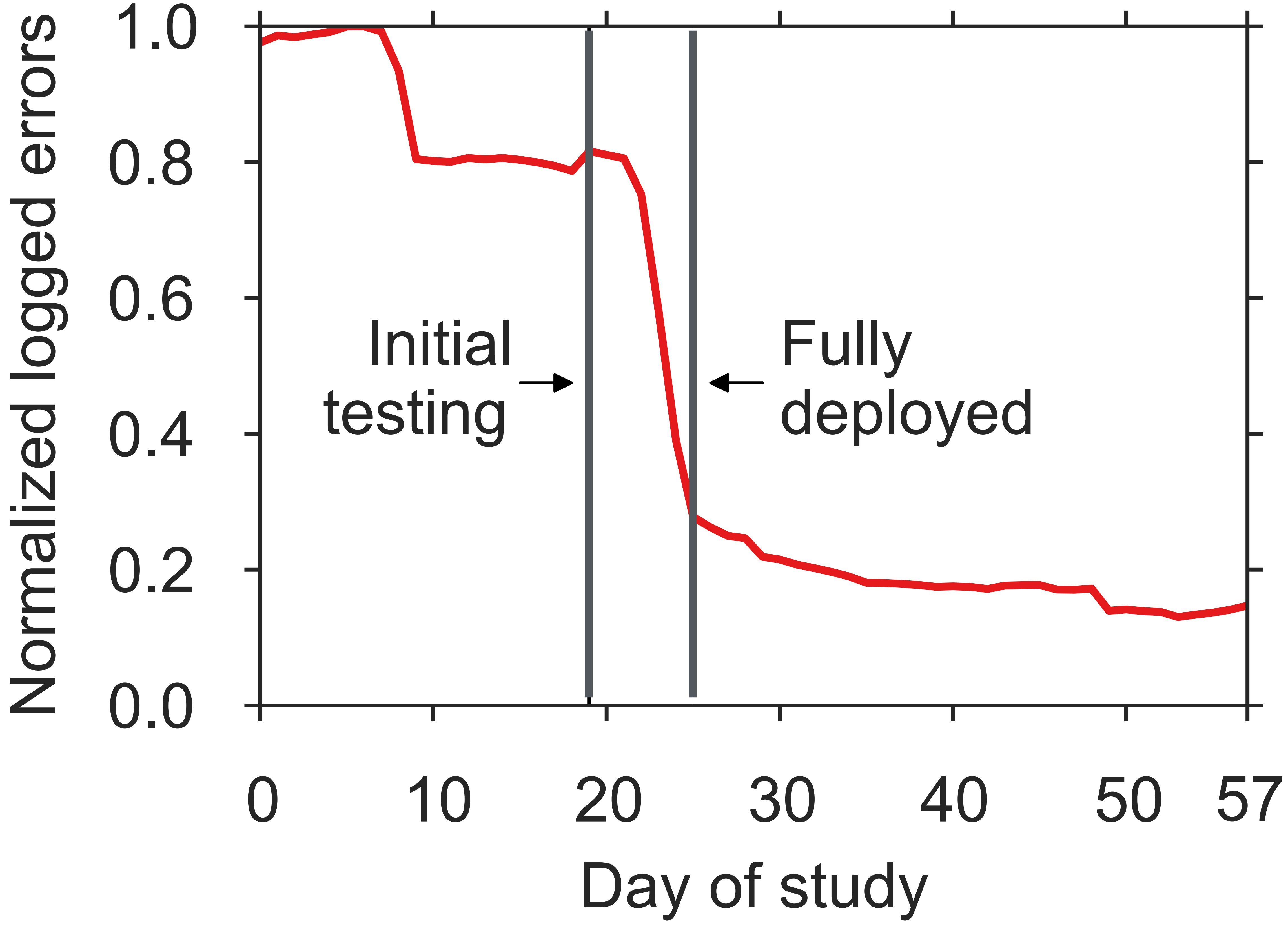


http://www.ece.cmu.edu/~safari/tools/memerr/

Page offlining

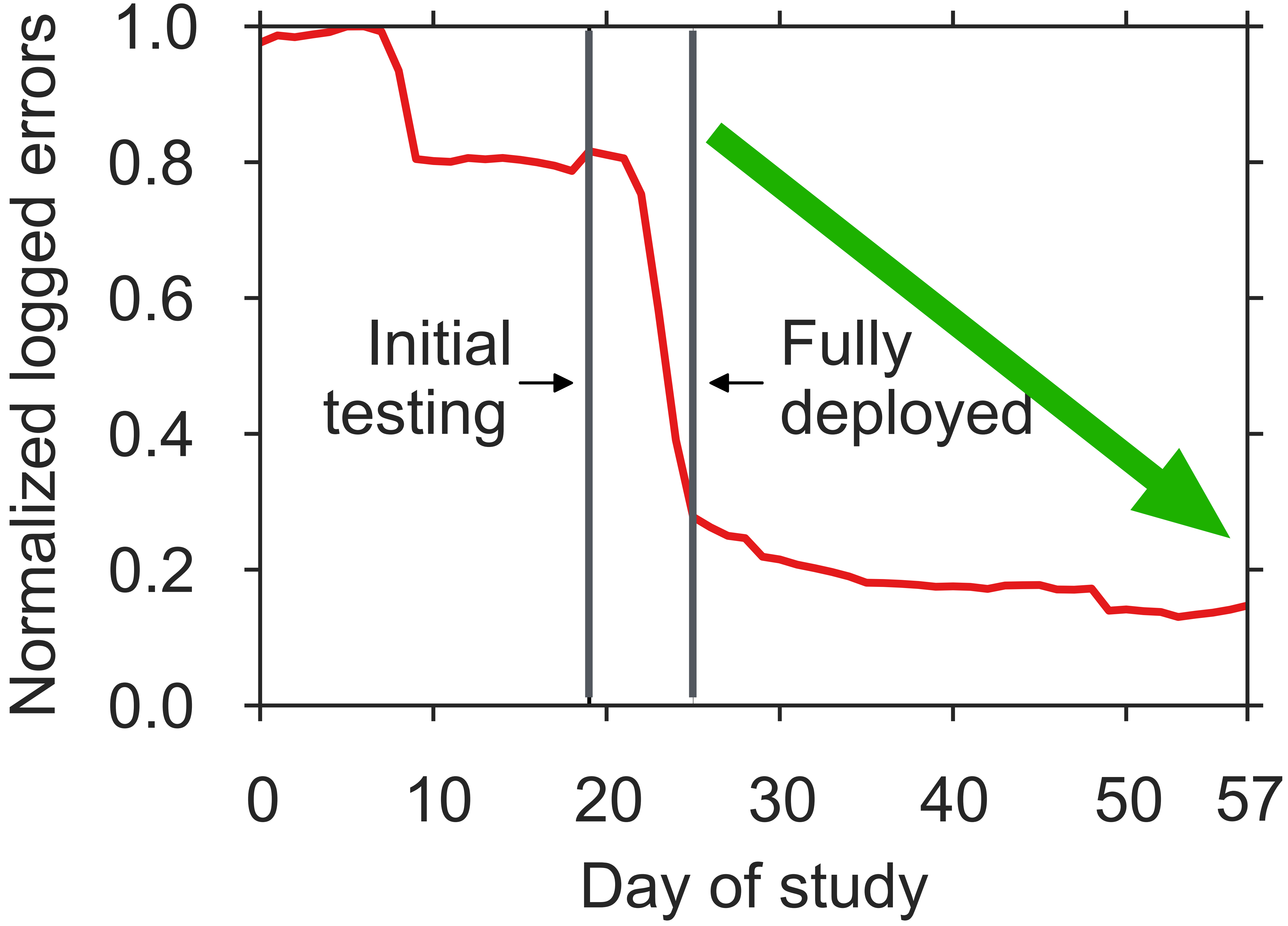
- ***System-level technique to reduce errors***
- ***When a page has an error, take the page offline***
  - *Copy* its contents to a new location
  - *Poison* the page to prevent allocation

PAGE OFFLINING AT SCALE



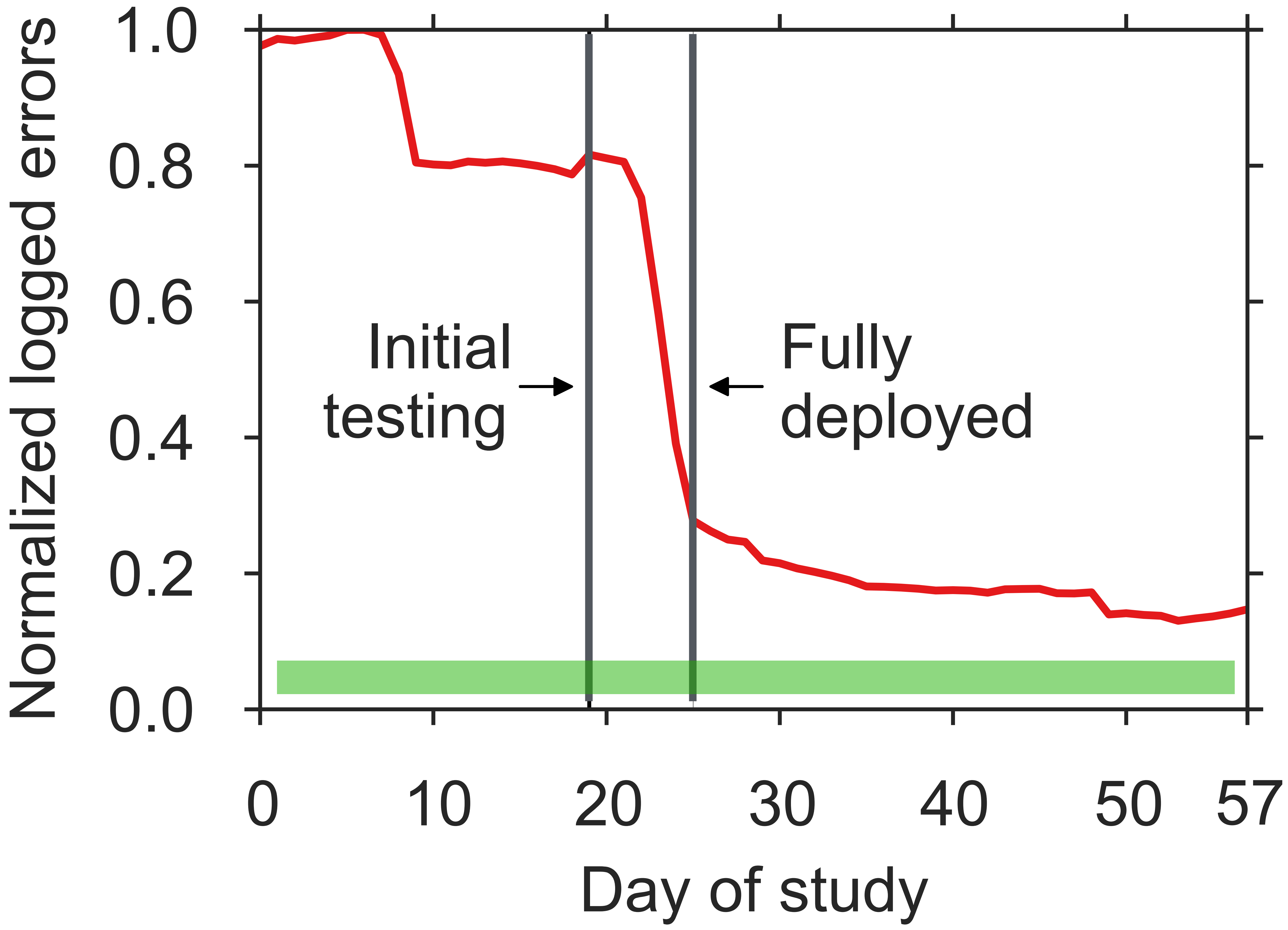
- First study at large scale
  - Cluster of 12,276 servers

PAGE OFFLINING AT SCALE



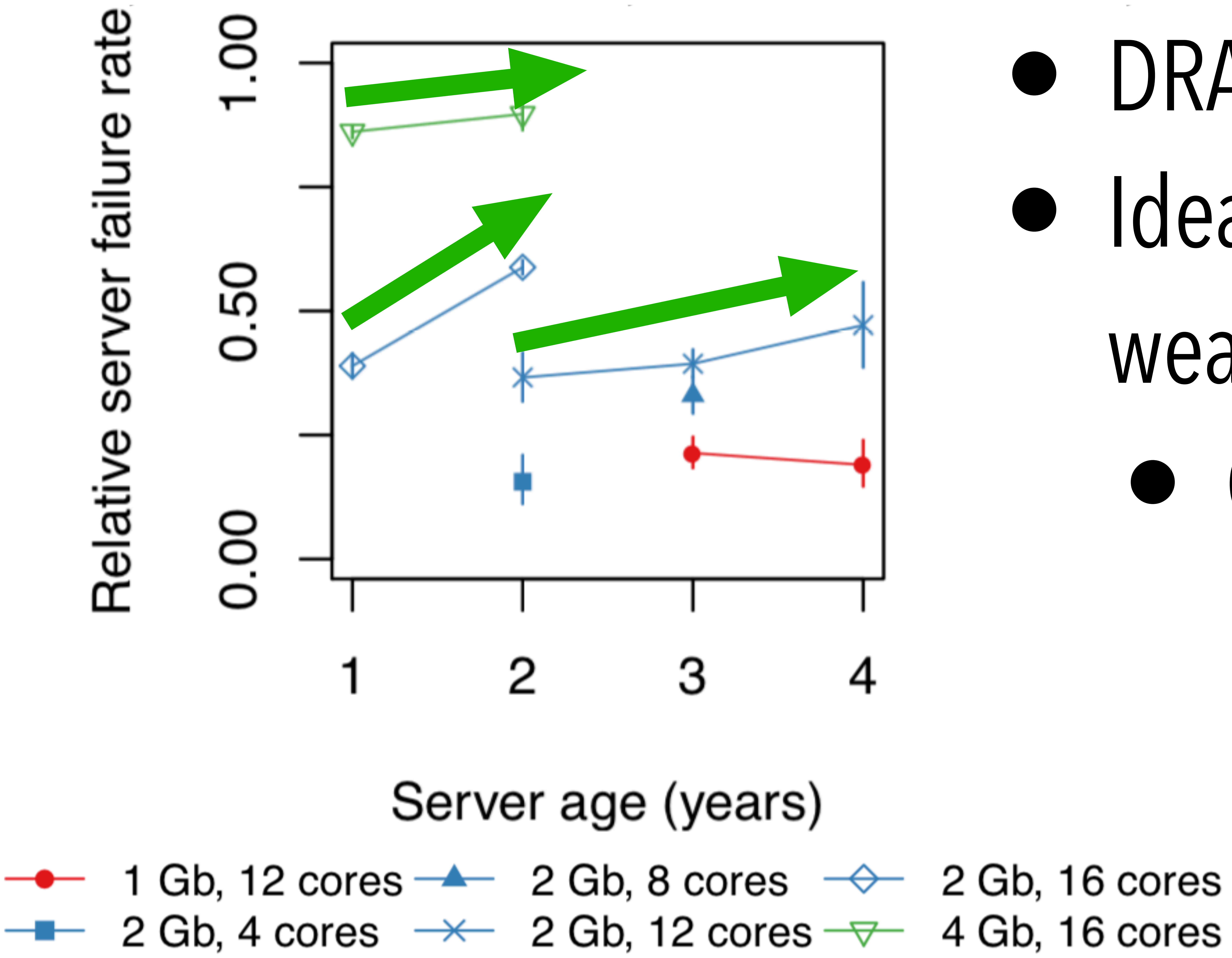
- First study at large scale
  - Cluster of 12,276 servers
- Reduced error rate by 67%

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

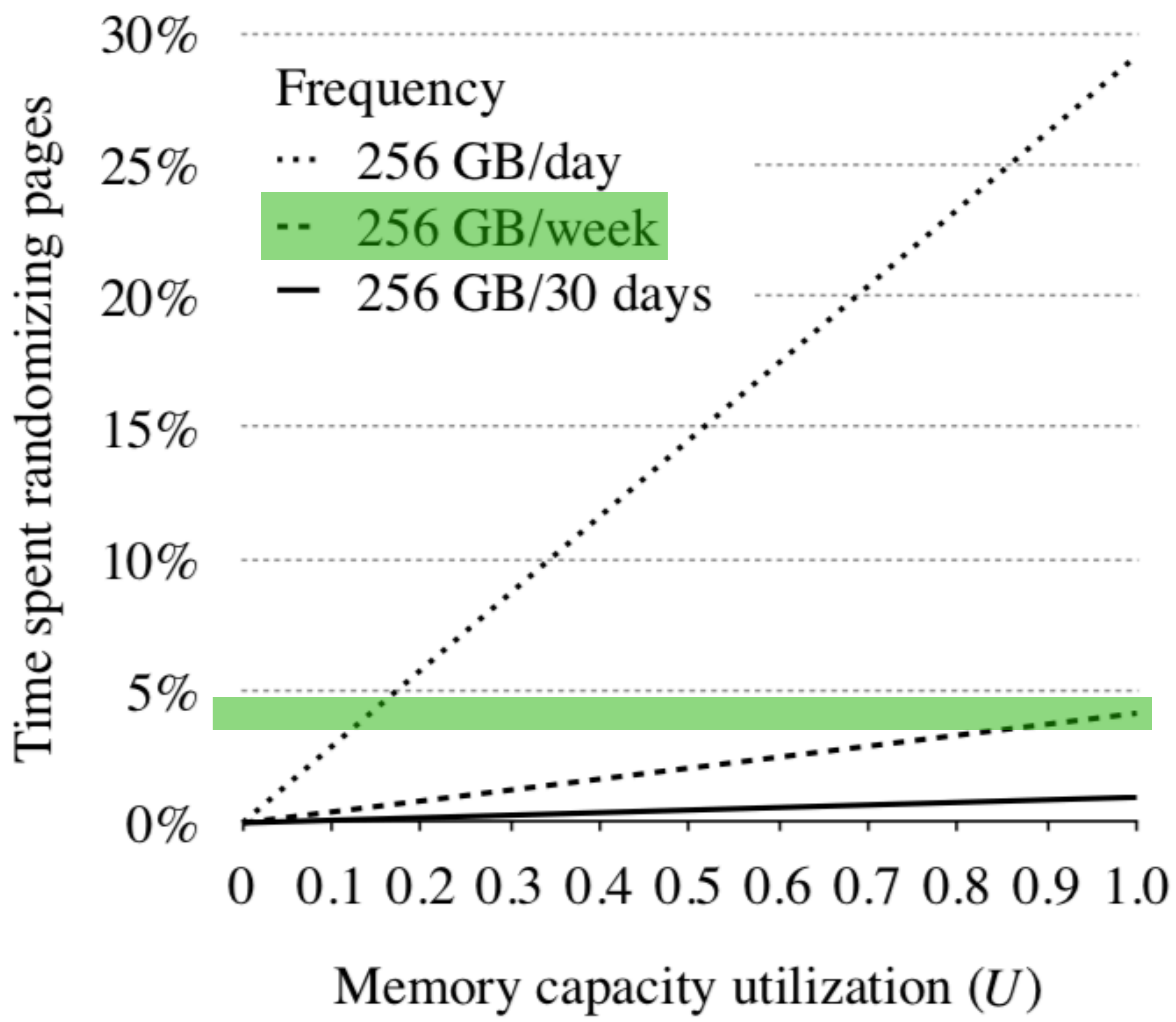
PAGE RANDOMIZATION

Input: The address of a physical page to randomize.

```
Lock the page.
    Flush any pending updates to the page.
    Randomly select a new free page to allocate.
    Migrate the contents of the old page to the new page.
    Update the page table mappings and remove any stale TLB entires.
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

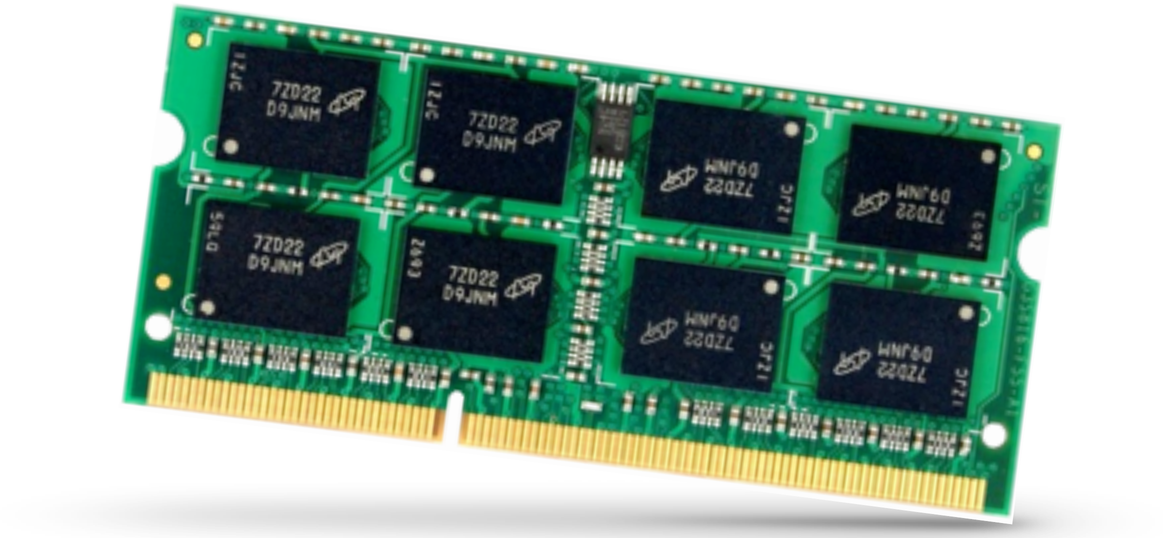
KEY DRAM CONTRIBUTIONS

- 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

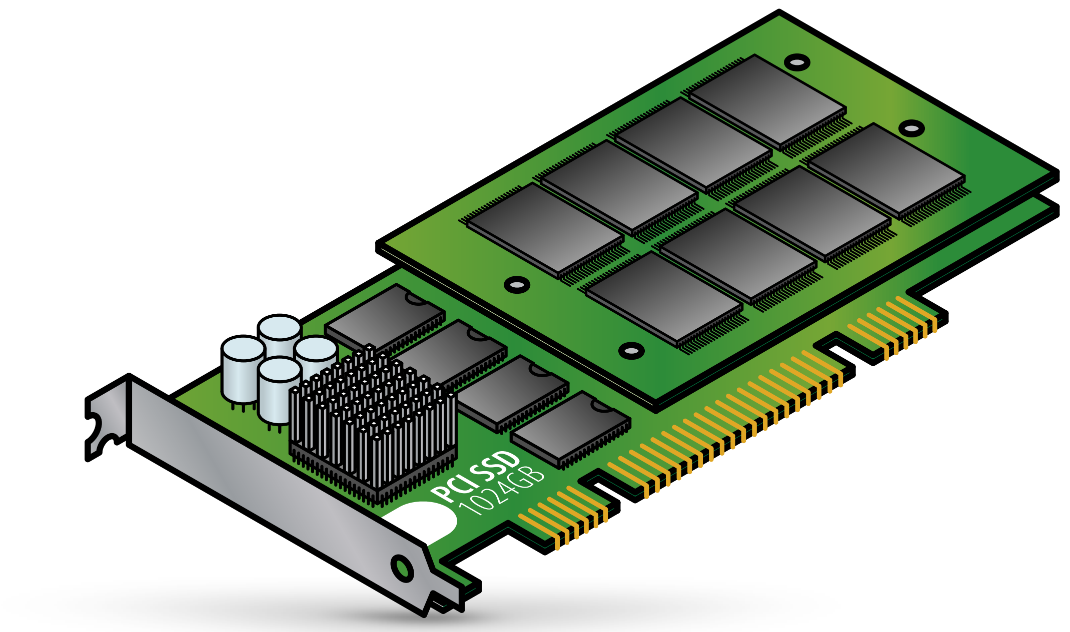
- ***DRAM errors at Google***
  [Schroeder+ SIGMETRICS'09]
- ***Component failures + simulated page offlining***
  [Hwang+ ASPLOS'12]
- ***Error correction, location, multi-DIMM errors***
  [Sridharan+ SC'12, SC'13; DeBardeleben+ SELSE'14]

LARGE SCALE STUDIES



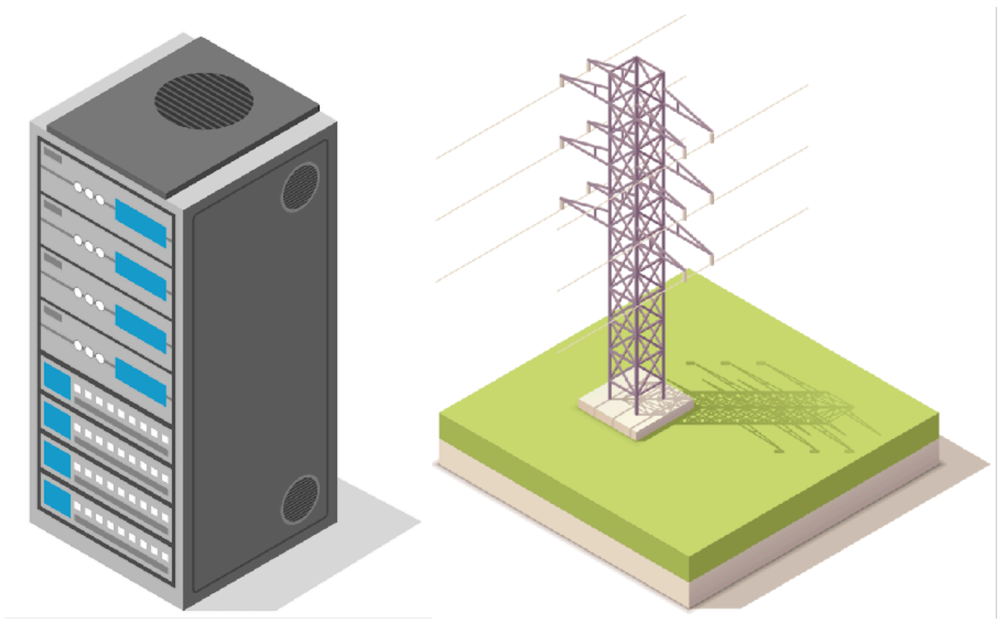
DRAM

[DSN '15]



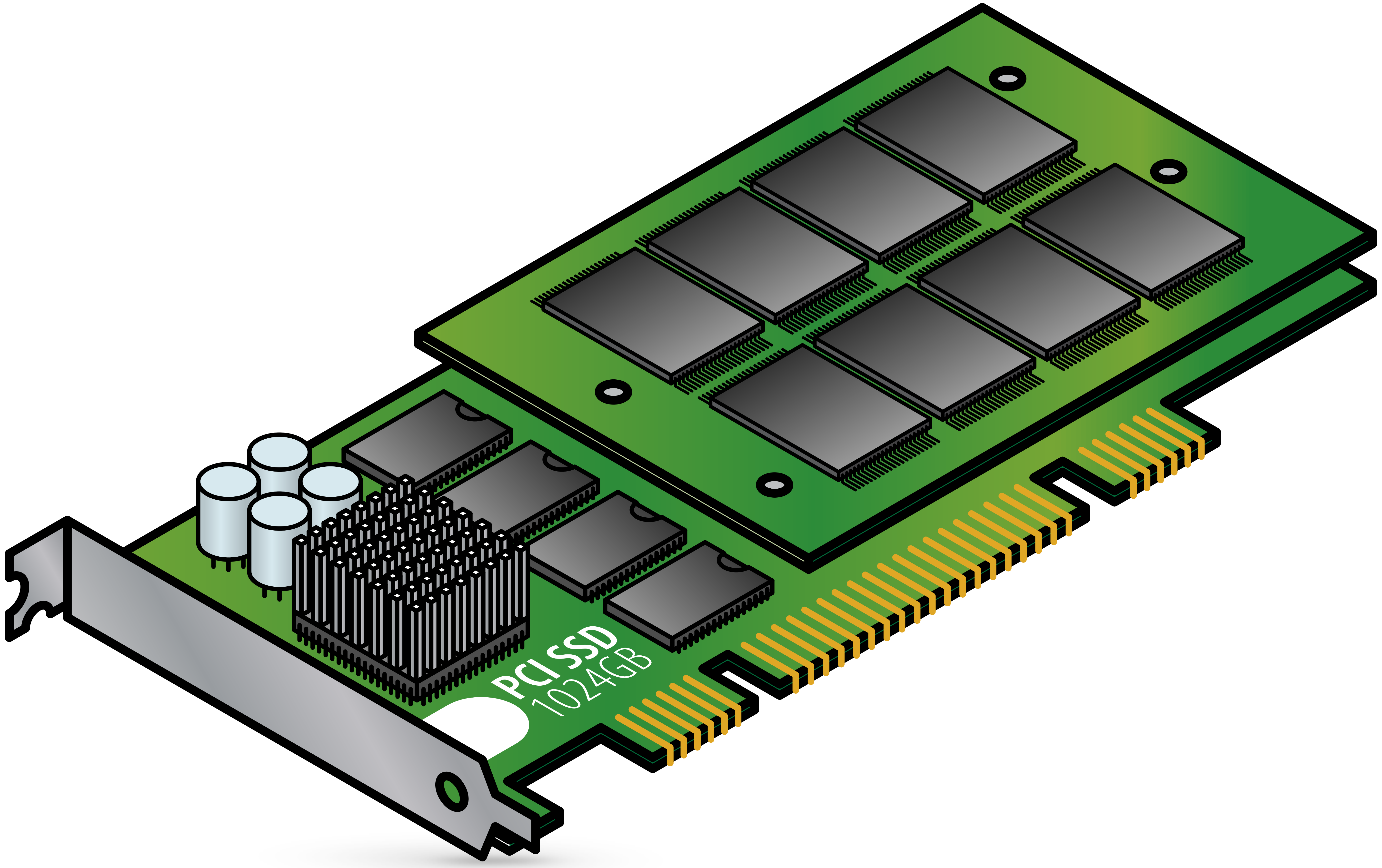
SSDs

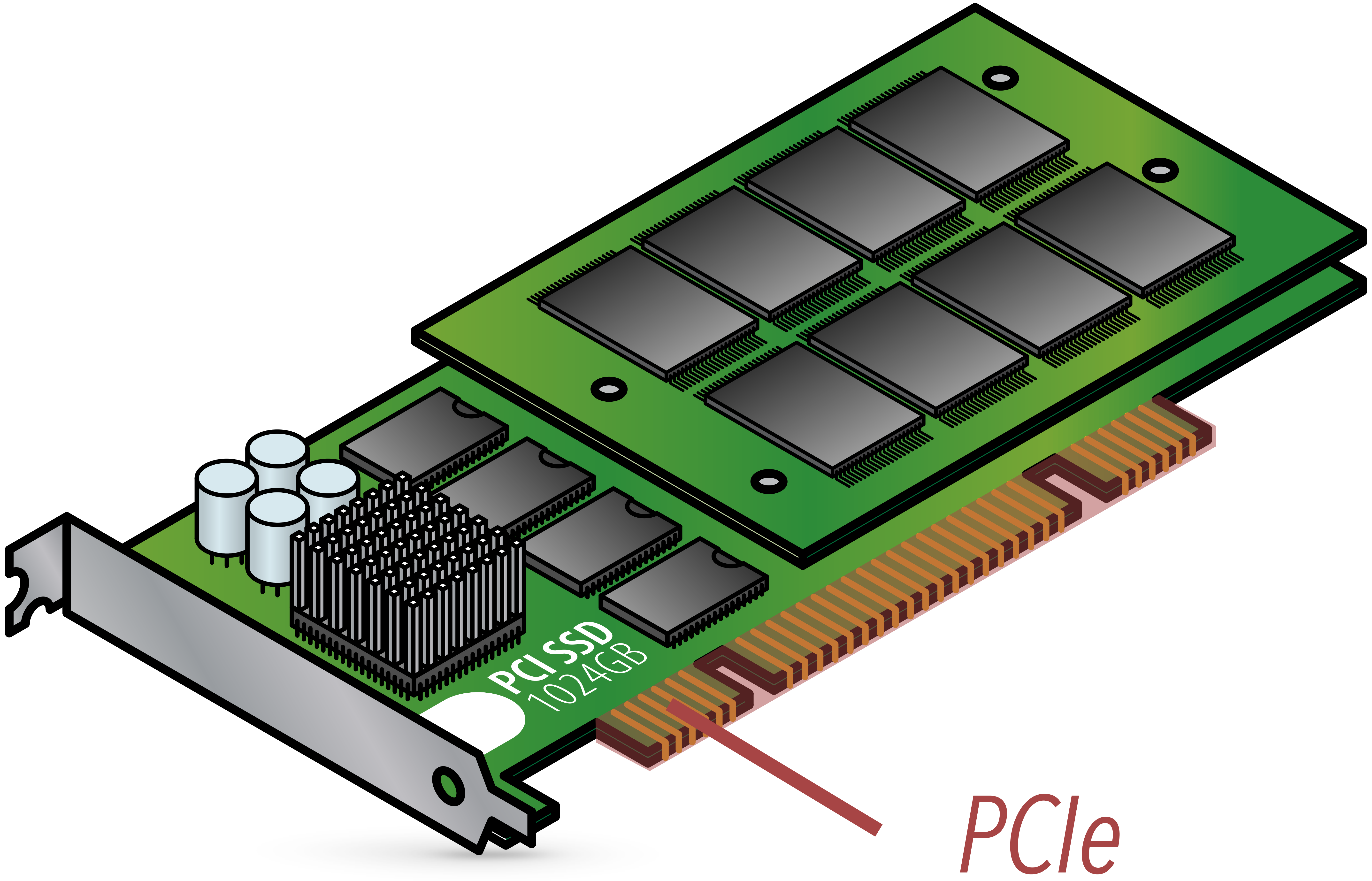
[SIGMETRICS '15]

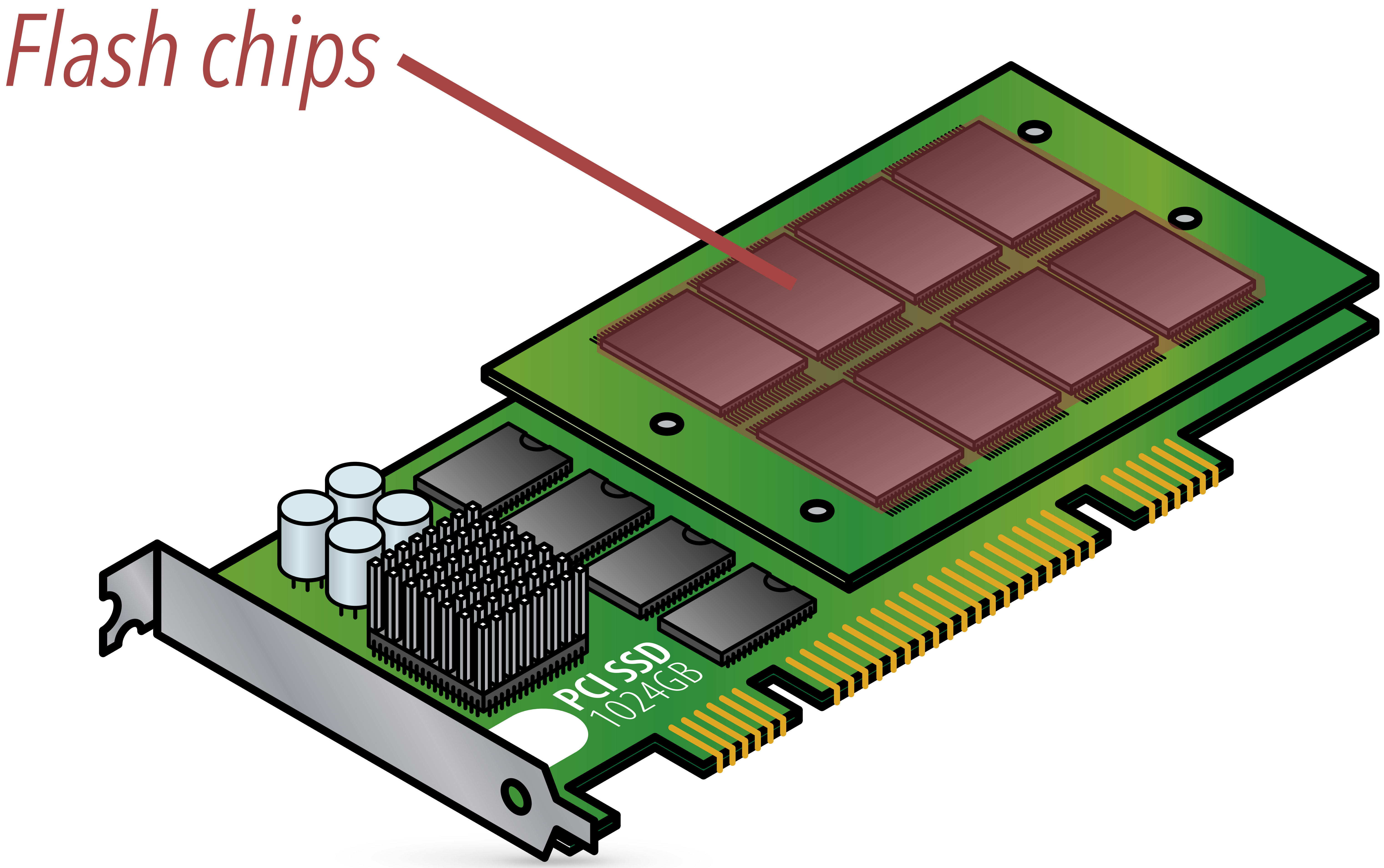


Networks

[IMC '18]


PCI SSD
1024GB

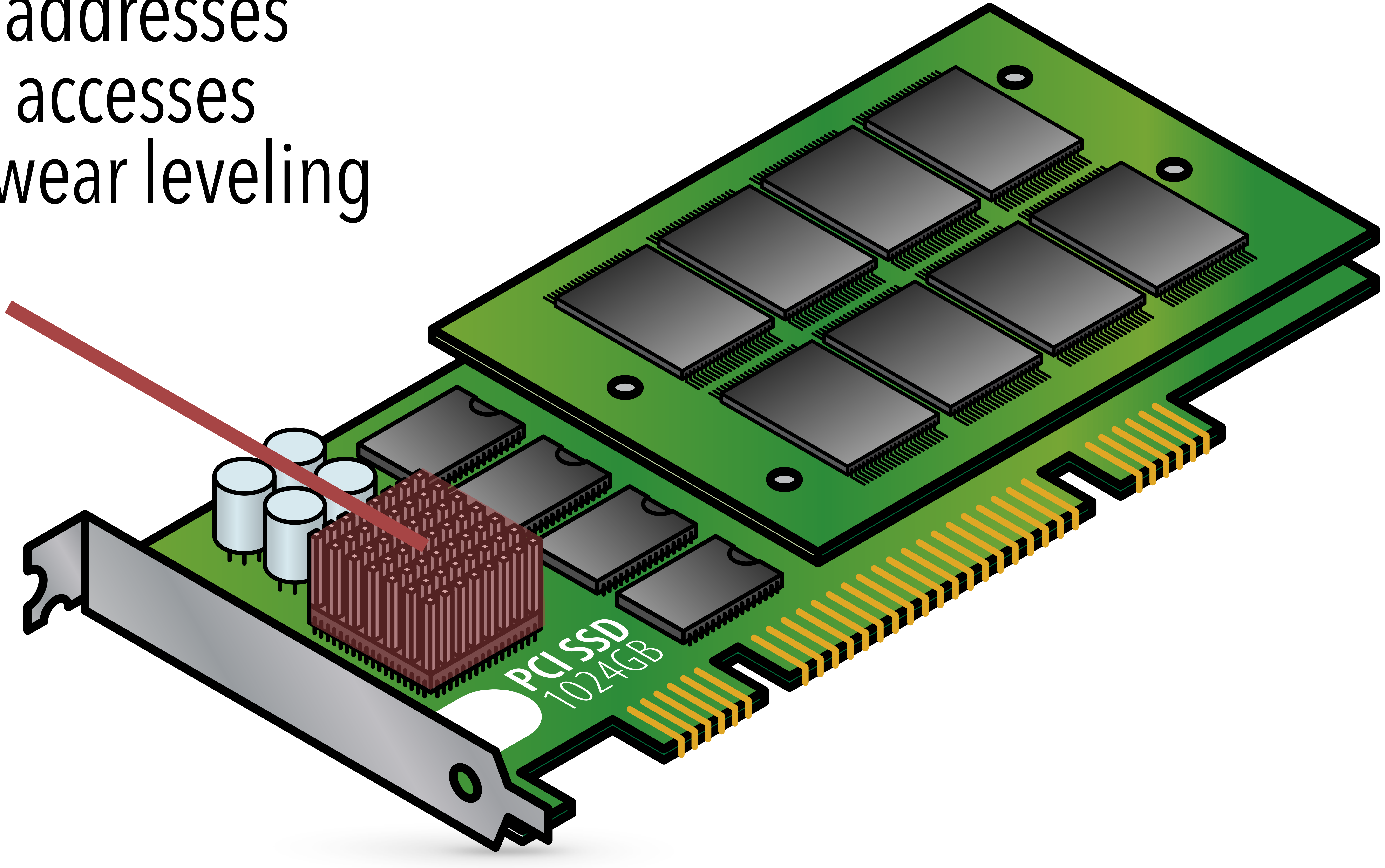

PCI SSD
1024GB
PCIe

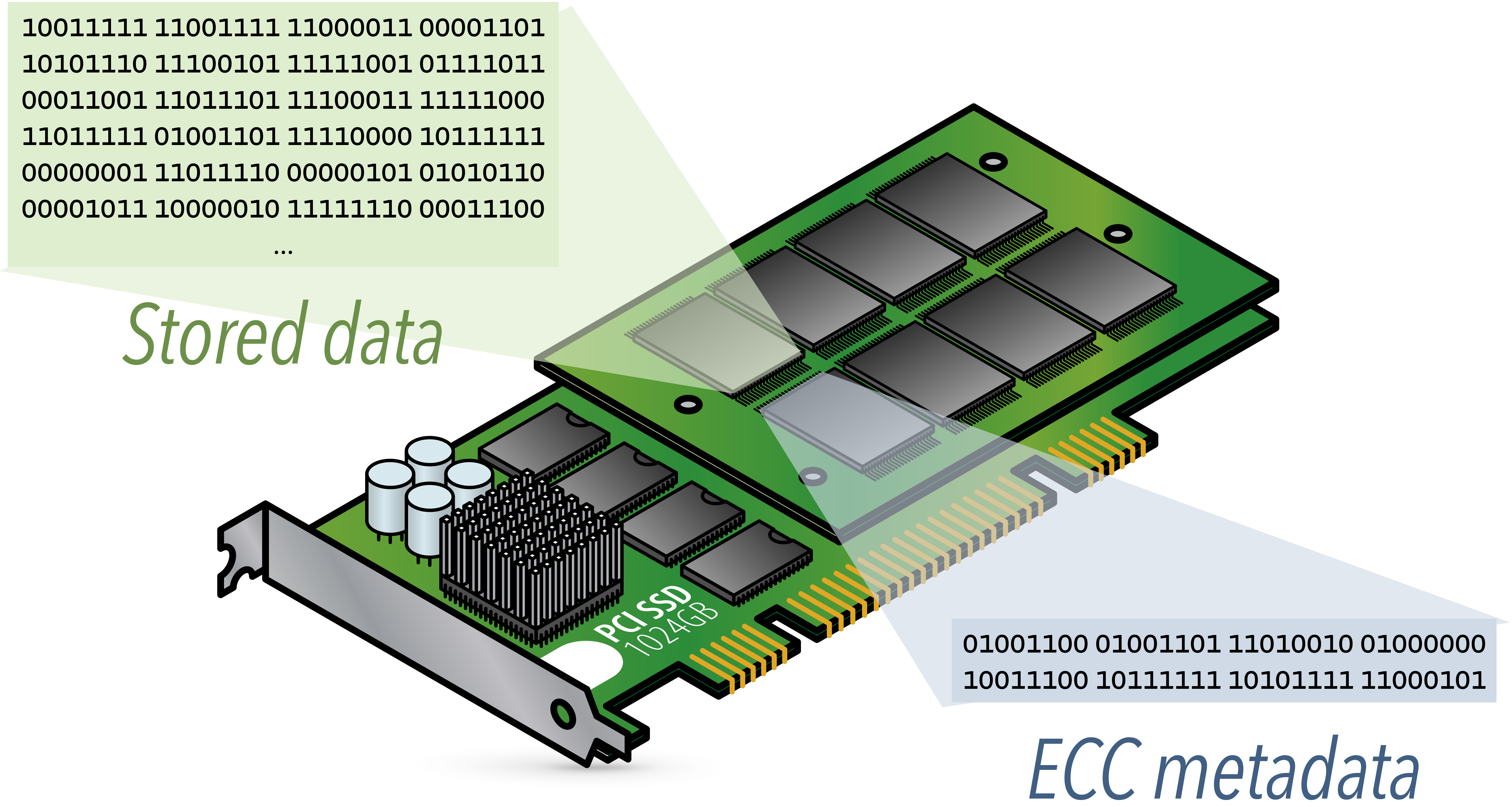


Flash chips
PCI SSD
1024GB

SSD controller

- translates addresses
- schedules accesses
- performs wear leveling




10011111 11001111 11000011 00001101
10101110 11100101 11111001 01111011
00011001 11011101 11100011 11111000
11011111 01001101 11110000 10111111
00000001 11011110 00000101 01010110
00001011 10000010 11111110 00011100
...
Stored data
PCI SSD
1024GB
01001100 01001101 11010010 01000000
10011100 10111111 10101111 11000101
ECC metadata

TYPES OF SSD FAILURES

Ones that cause SMALL ERRORS

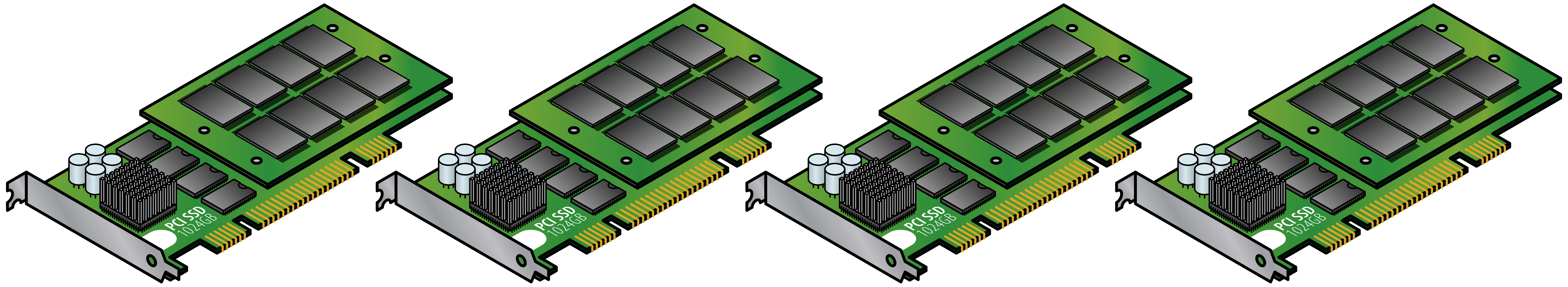
- 10's of flipped bits per KB
- Silently corrected by SSD controller

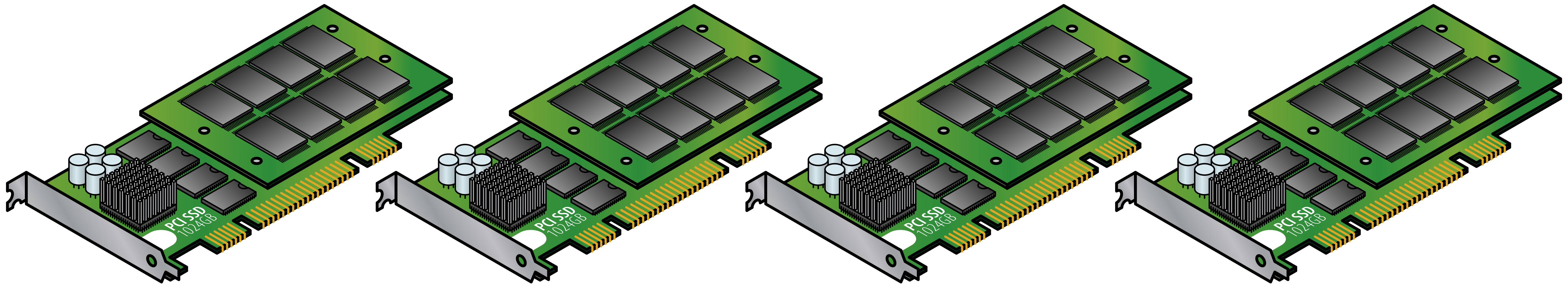
Ones that cause LARGE ERRORS

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

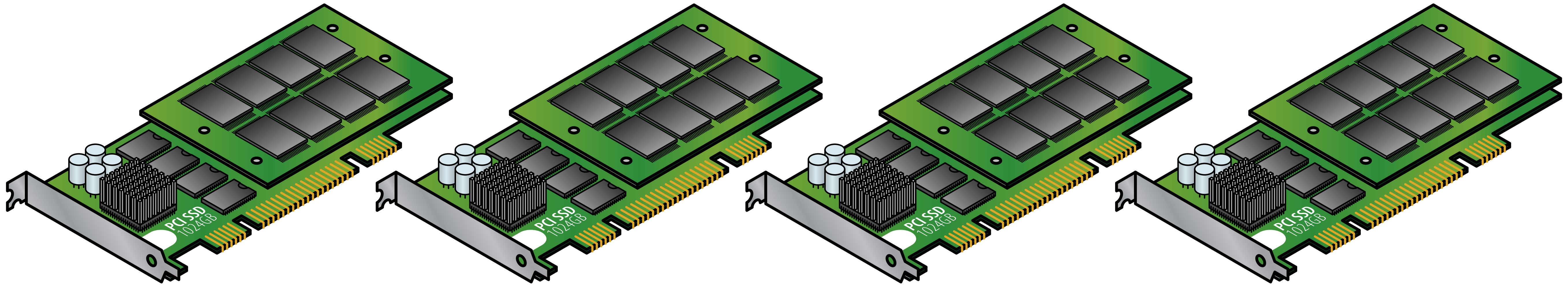
MEASURING SSD FAILURES

- ***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***



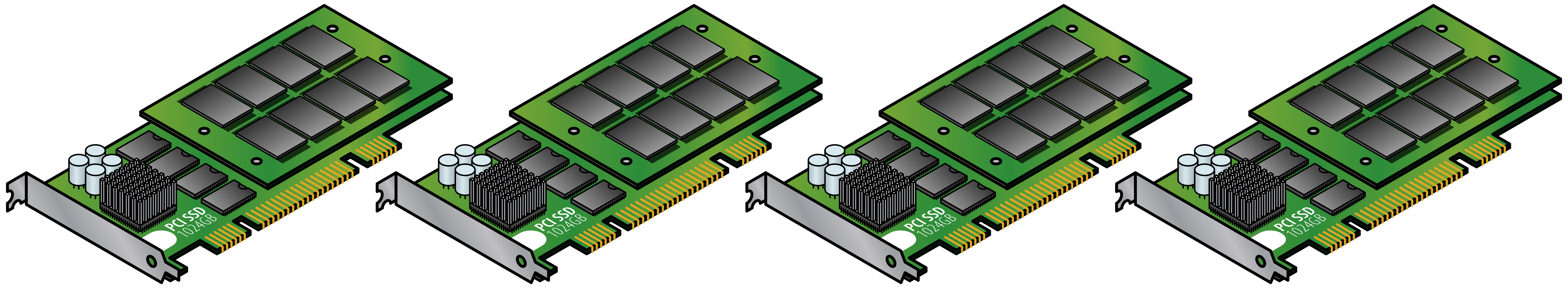


Errors	54,326	0	2	10
Data written	10TB	2TB	5TB	6TB



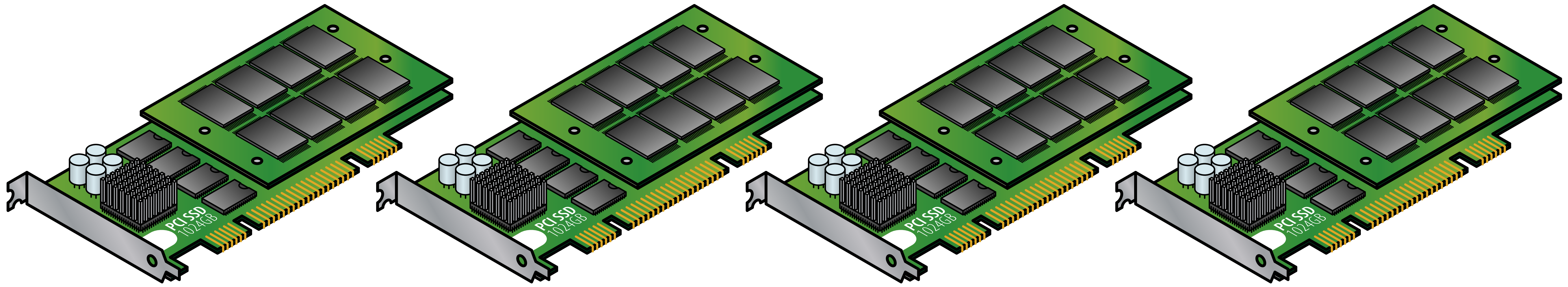
Errors	54,326	0	2	10
Data written	10TB	2TB	5TB	6TB

2018-12-3



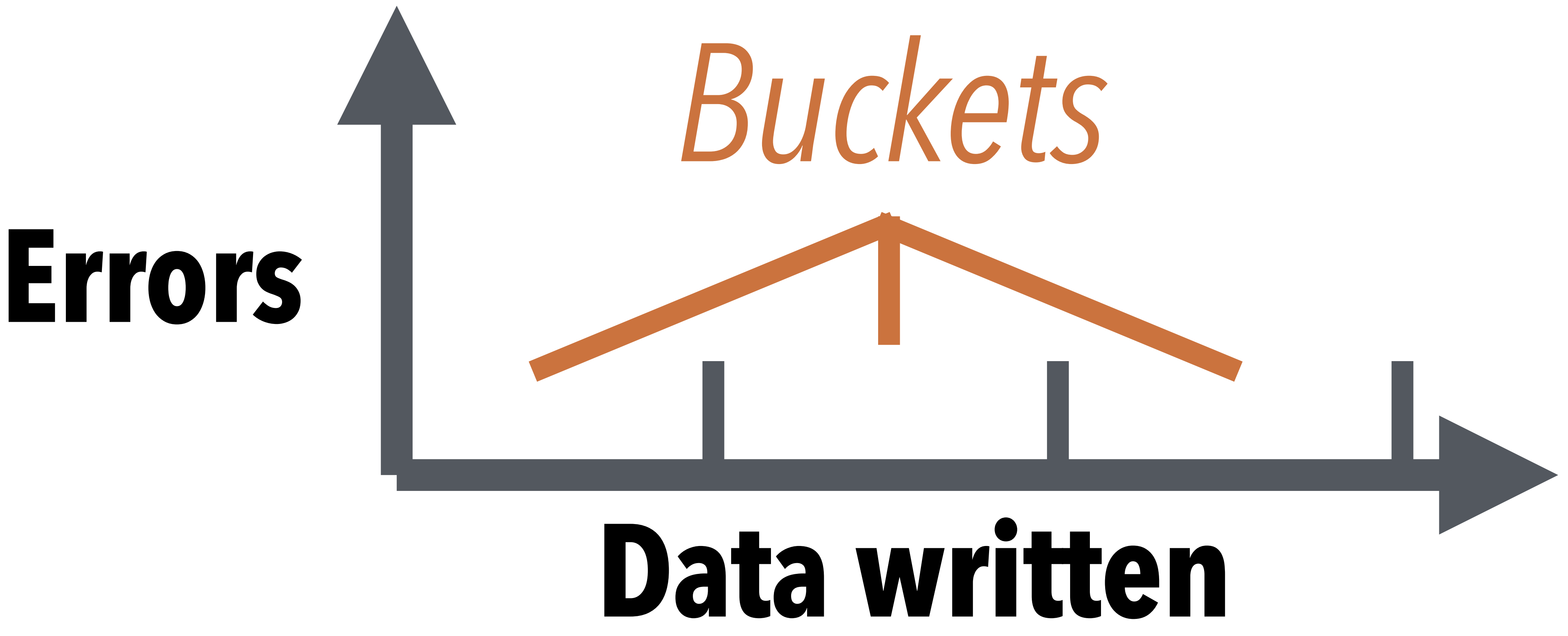
Errors	54,326	0	2	10
Data written	10TB	2TB	5TB	6TB

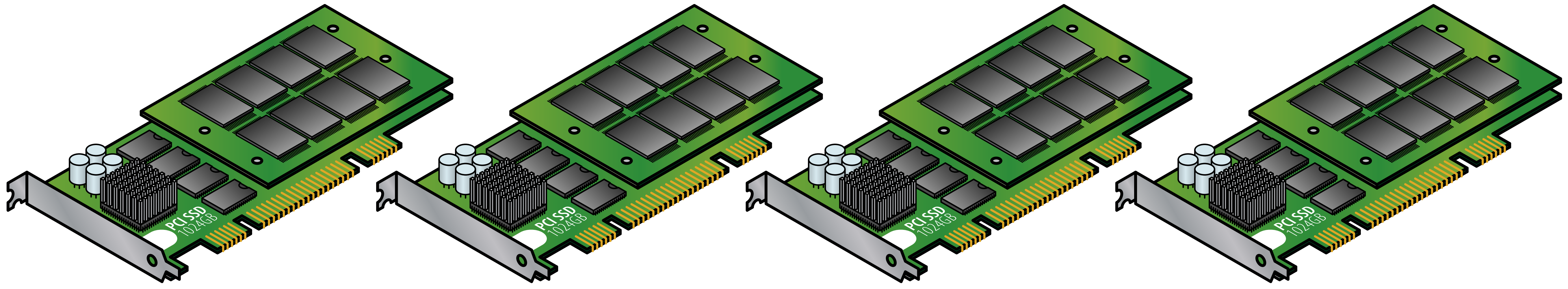
2018-12-3



Errors	54,326	0	2	10
Data written	10TB	2TB	5TB	6TB

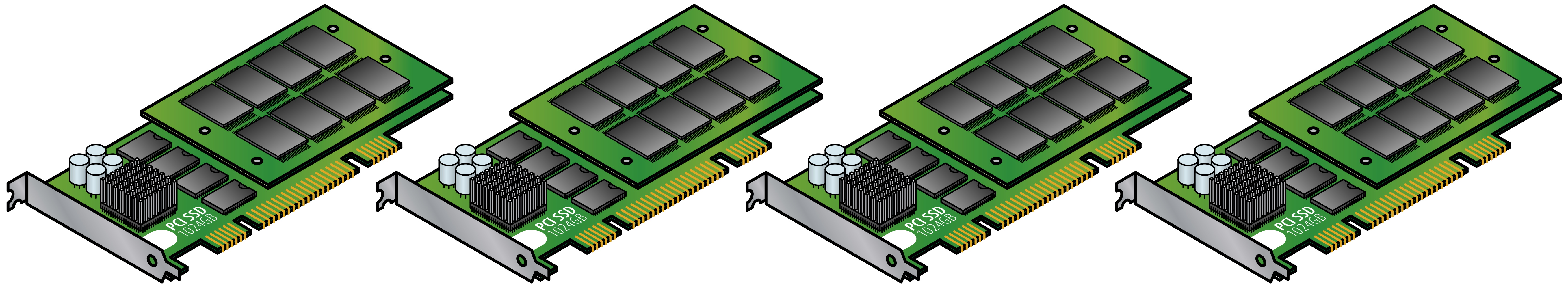
2018-12-3





Errors	54,326	0	2	10
Data written	10TB	2TB	5TB	6TB

2018-12-3



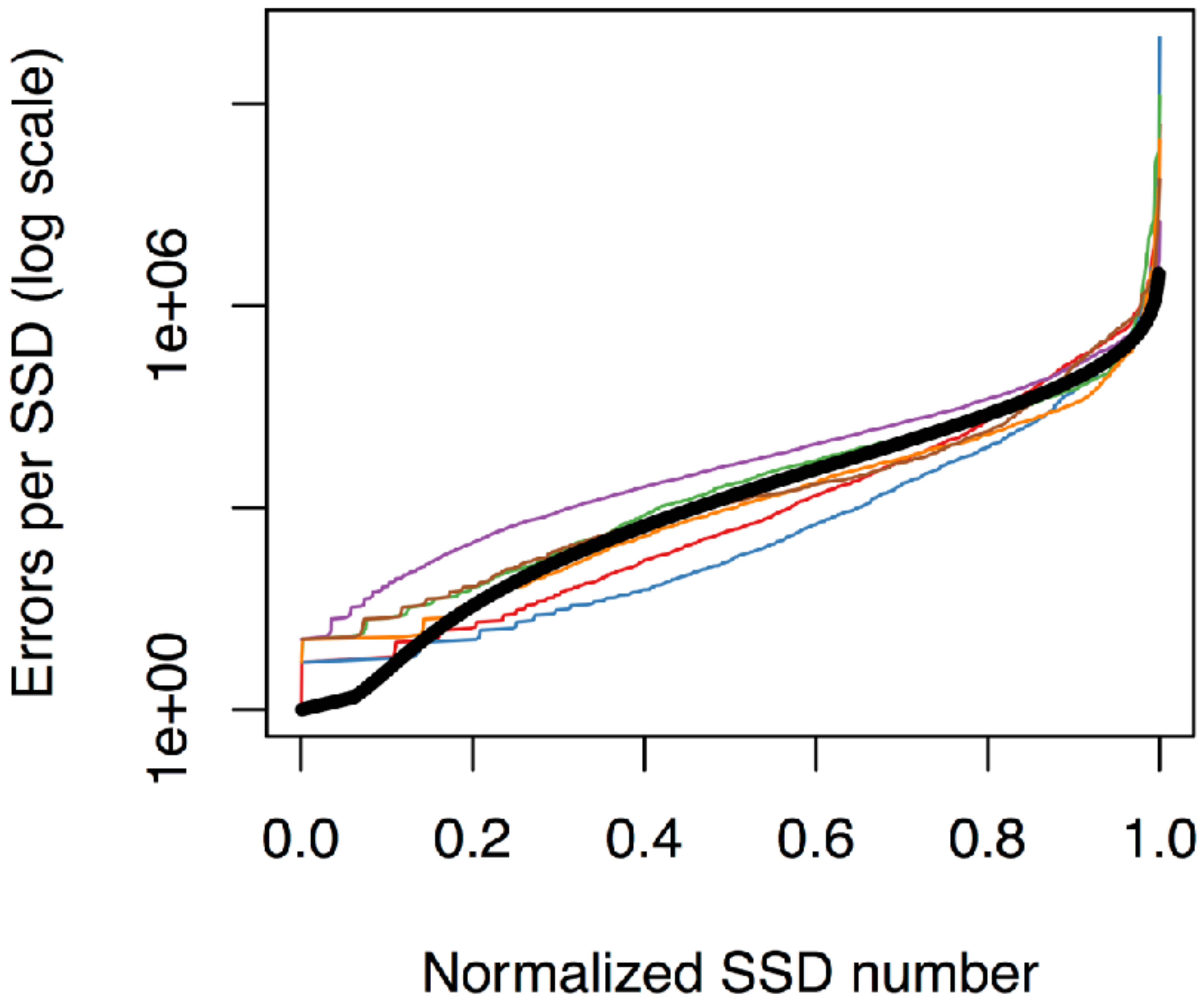
Errors	54,326	0	2	10
Data written	10TB	2TB	5TB	6TB

2018-12-3

KEY SSD CONTRIBUTIONS

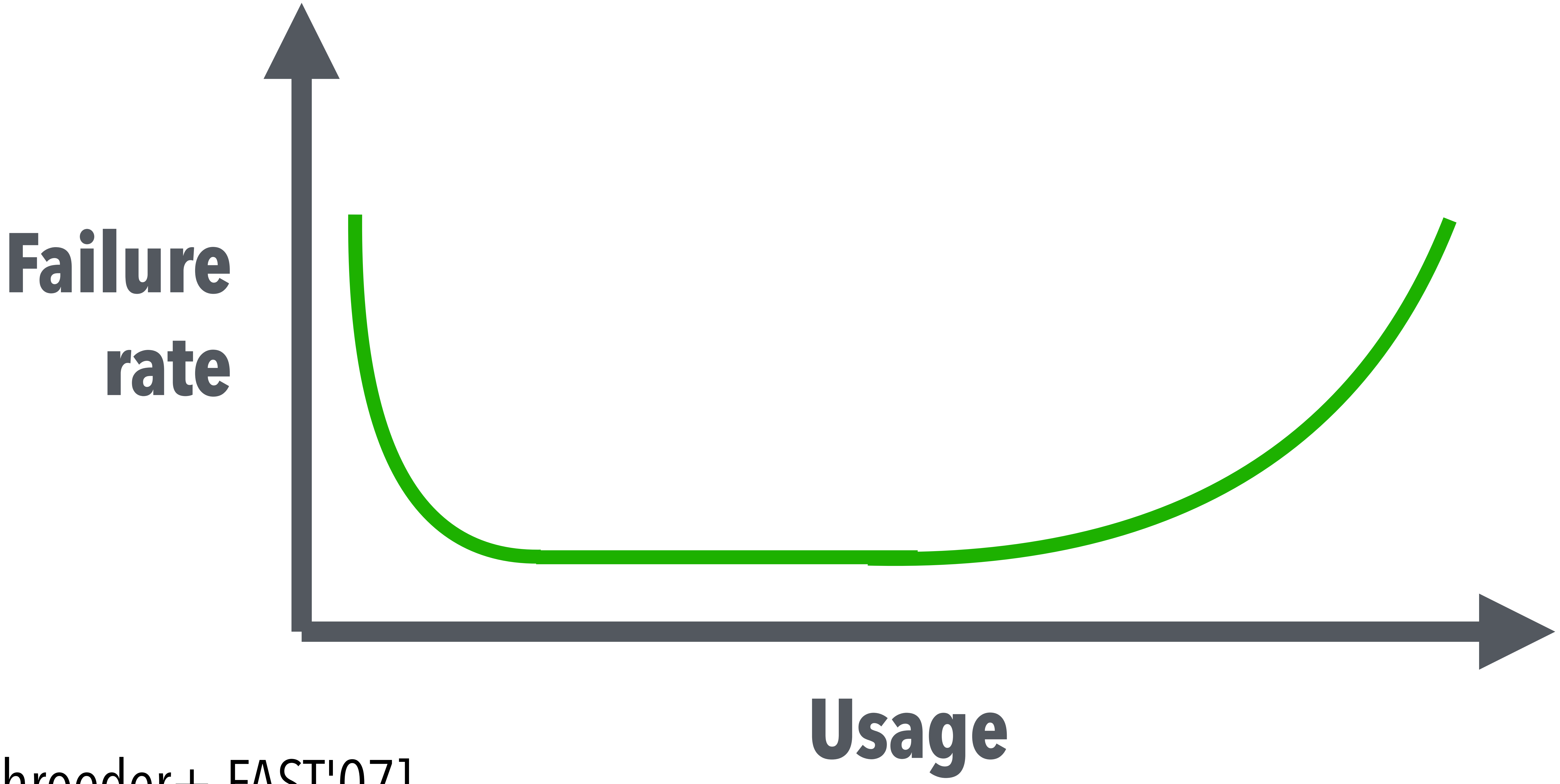
- Distinct lifecycle periods
- Read disturbance not prevalent in the field
- Higher temperatures cause more failures
- Amount of data written by OS is misleading
- Write amplification trends from the field

FAILURE MODELING



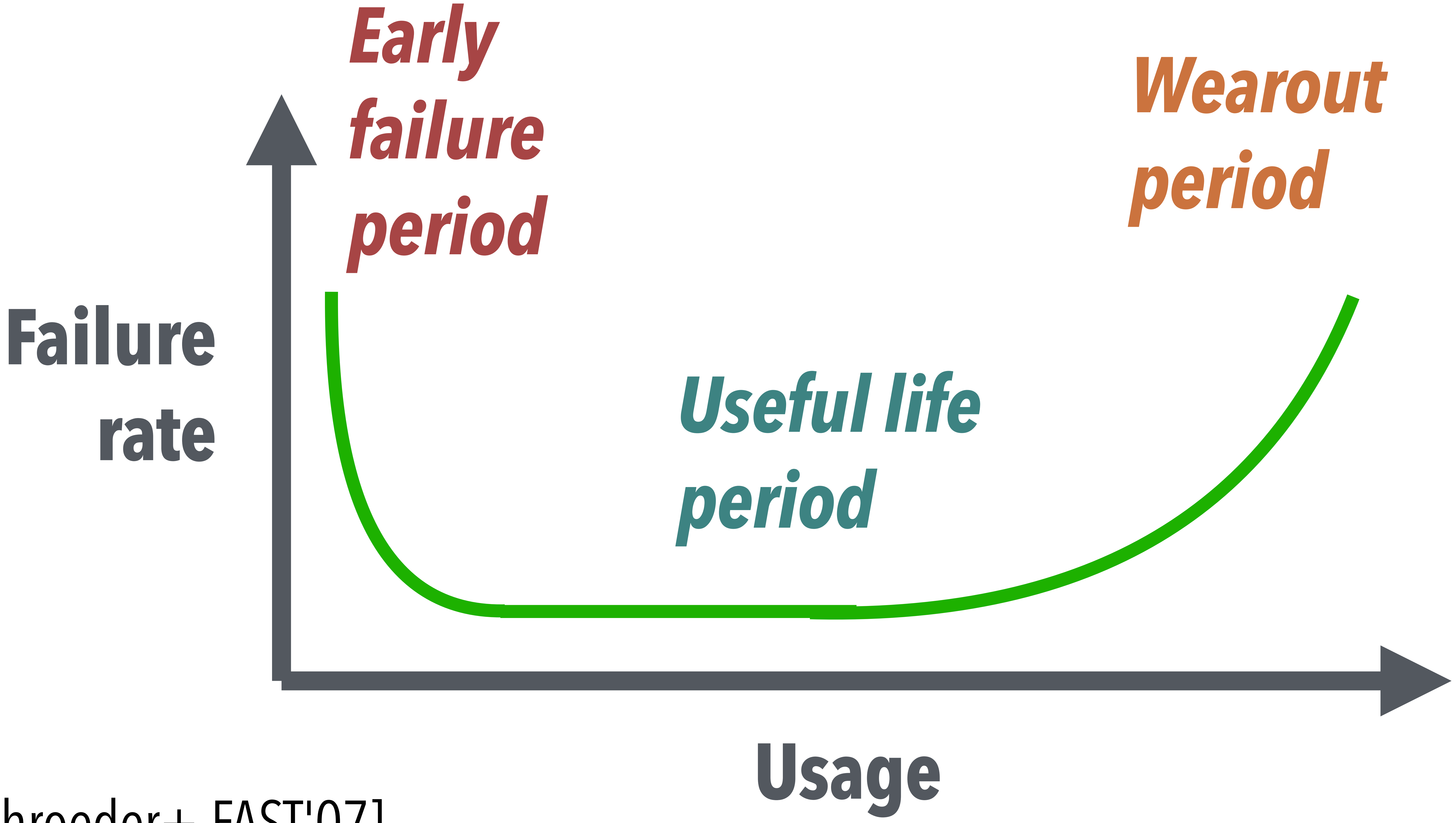
- Built a model across 6 SSD server configurations
- **Weibull** *(0.3, 5e3)*
- Most errors are from a small set of SSDs

Storage lifecycle background:
the *bathtub curve* for disk drives



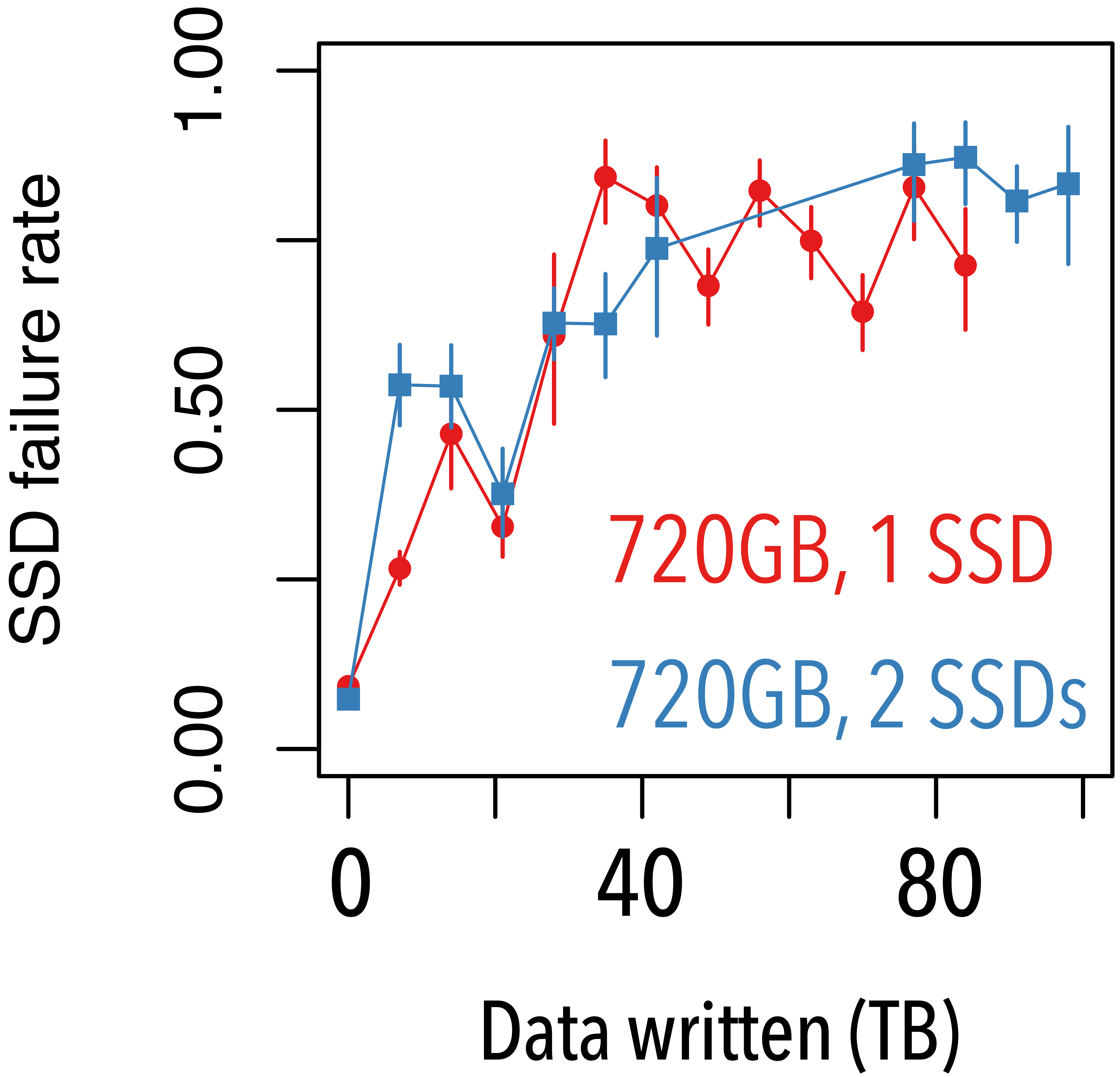
[Schroeder+,FAST'07]

Storage lifecycle background: *the bathtub curve* for disk drives

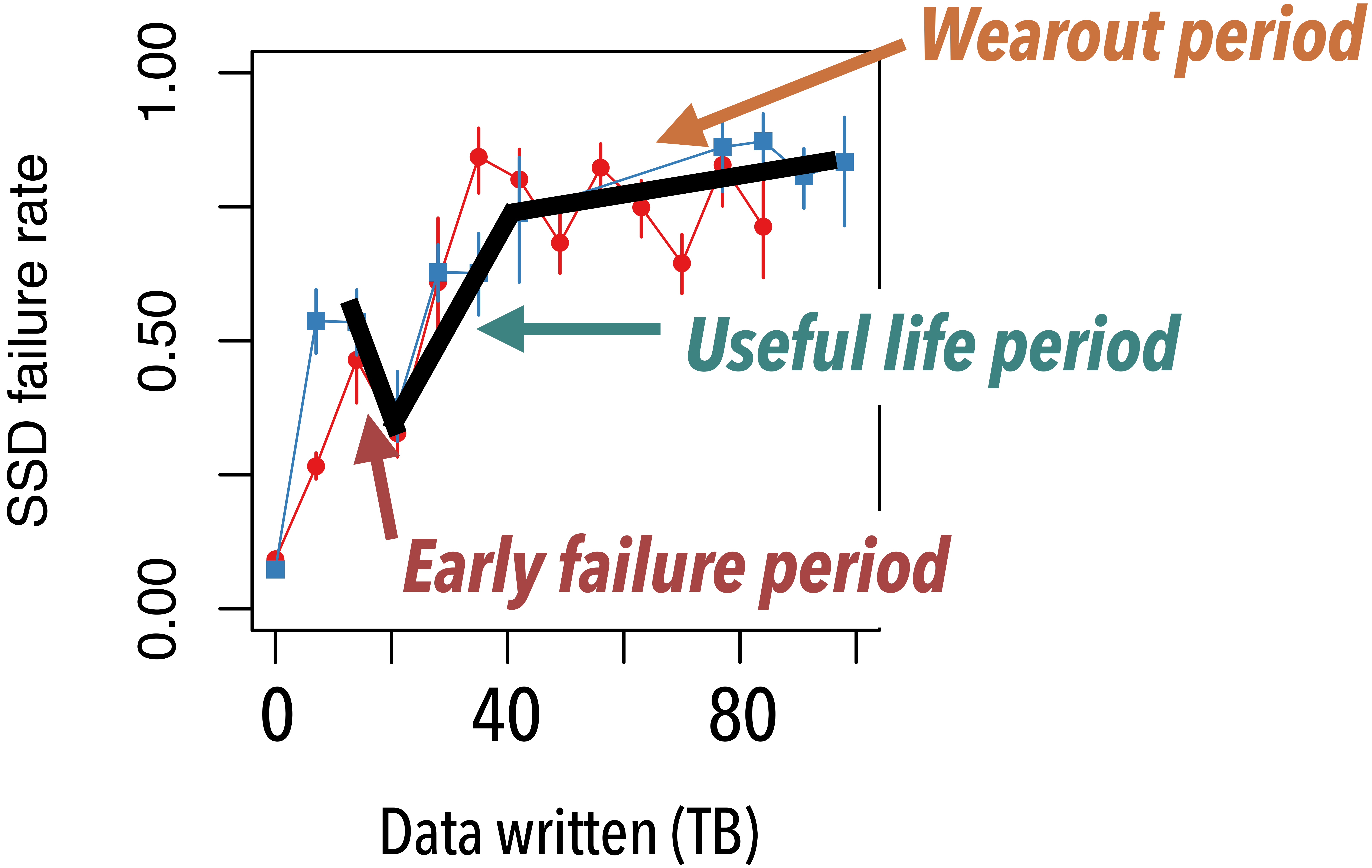


[Schroeder+,FAST'07]

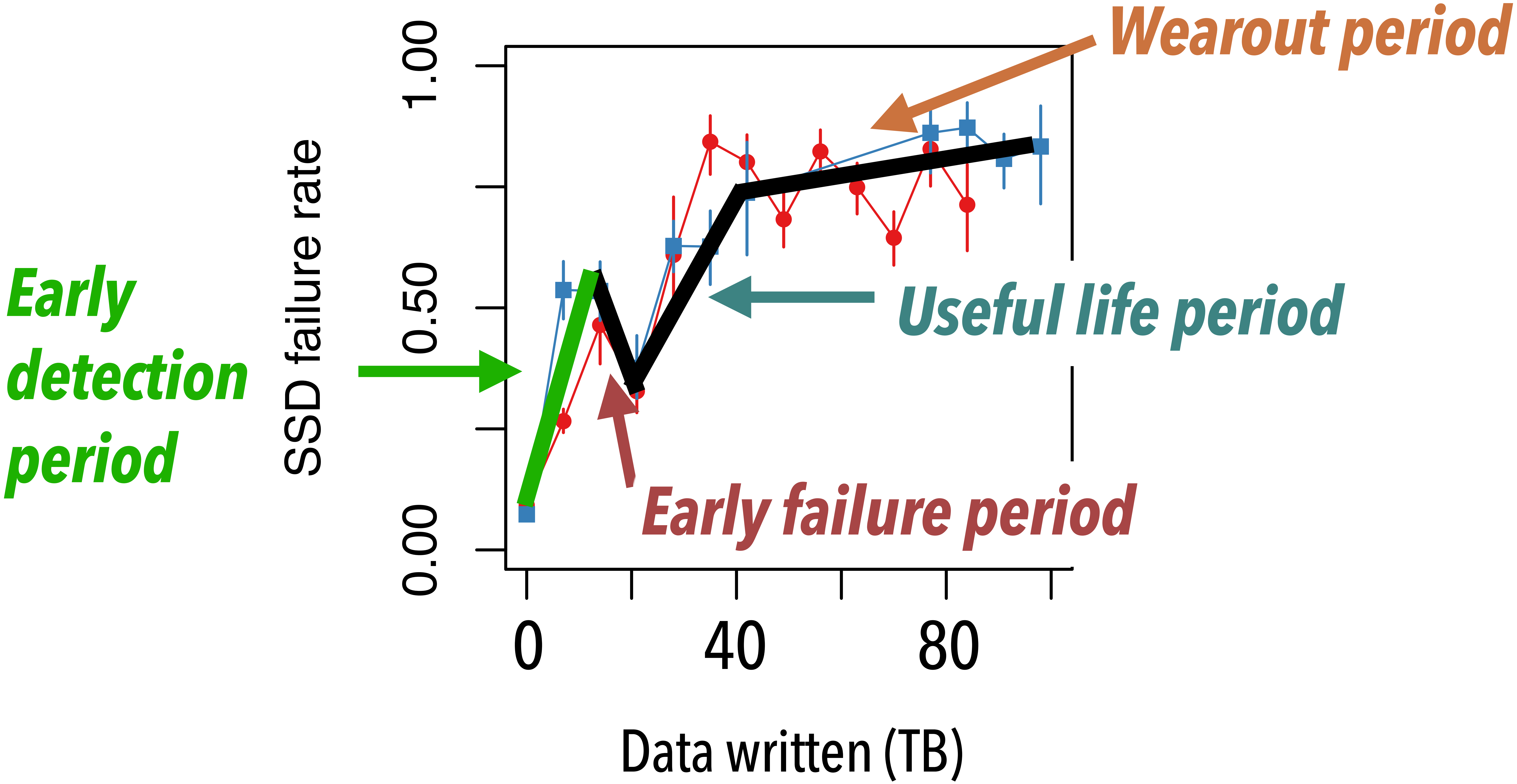
SSD LIFECYCLE PERIODS



SSD LIFECYCLE PERIODS



SSD LIFECYCLE PERIODS



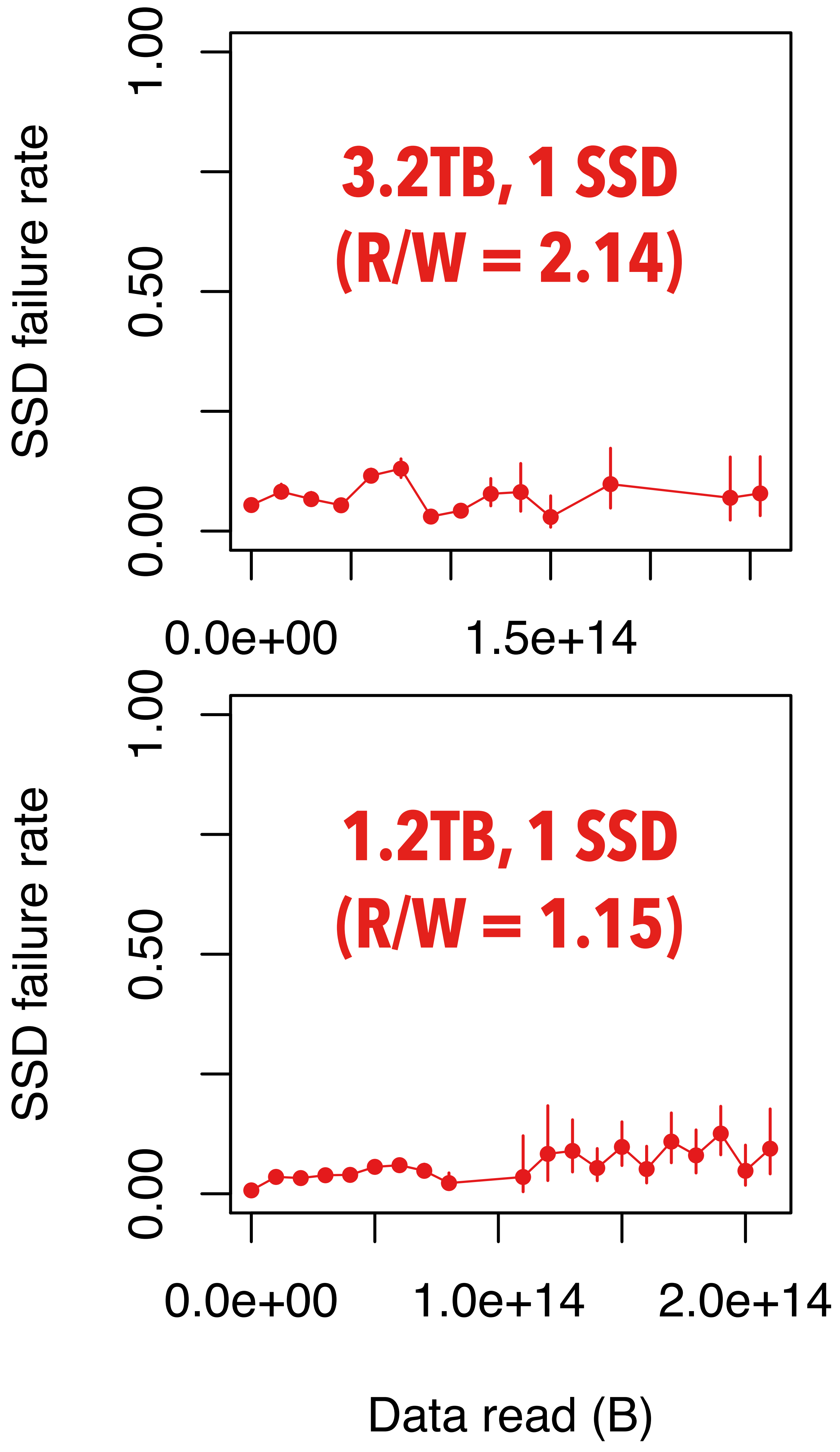
SSD LIFECYCLE PERIODS

- 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

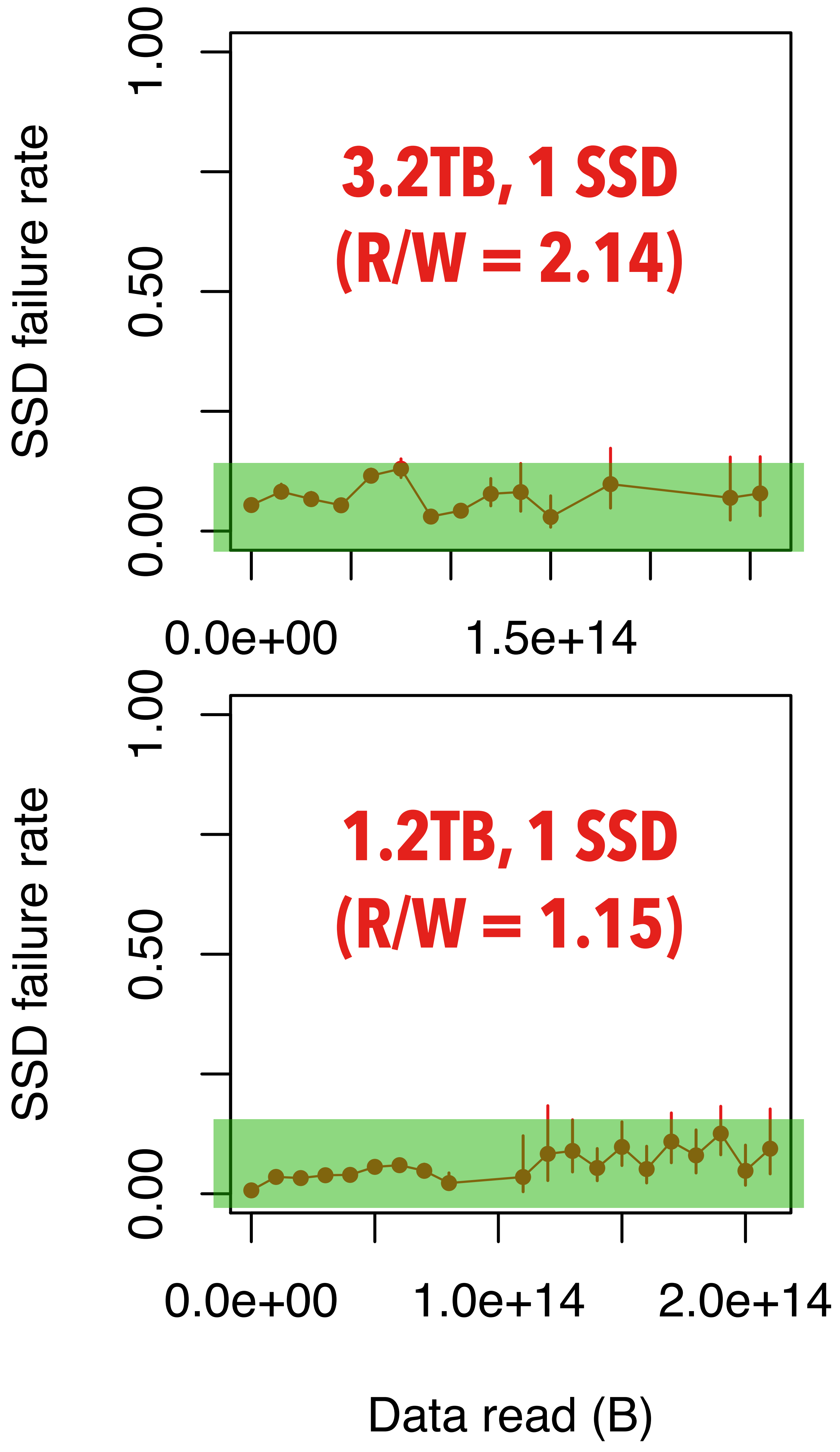
- Charge drift from reads to neighboring cells
- Documented in prior controlled studies on chips

READ DISTURBANCE ERRORS



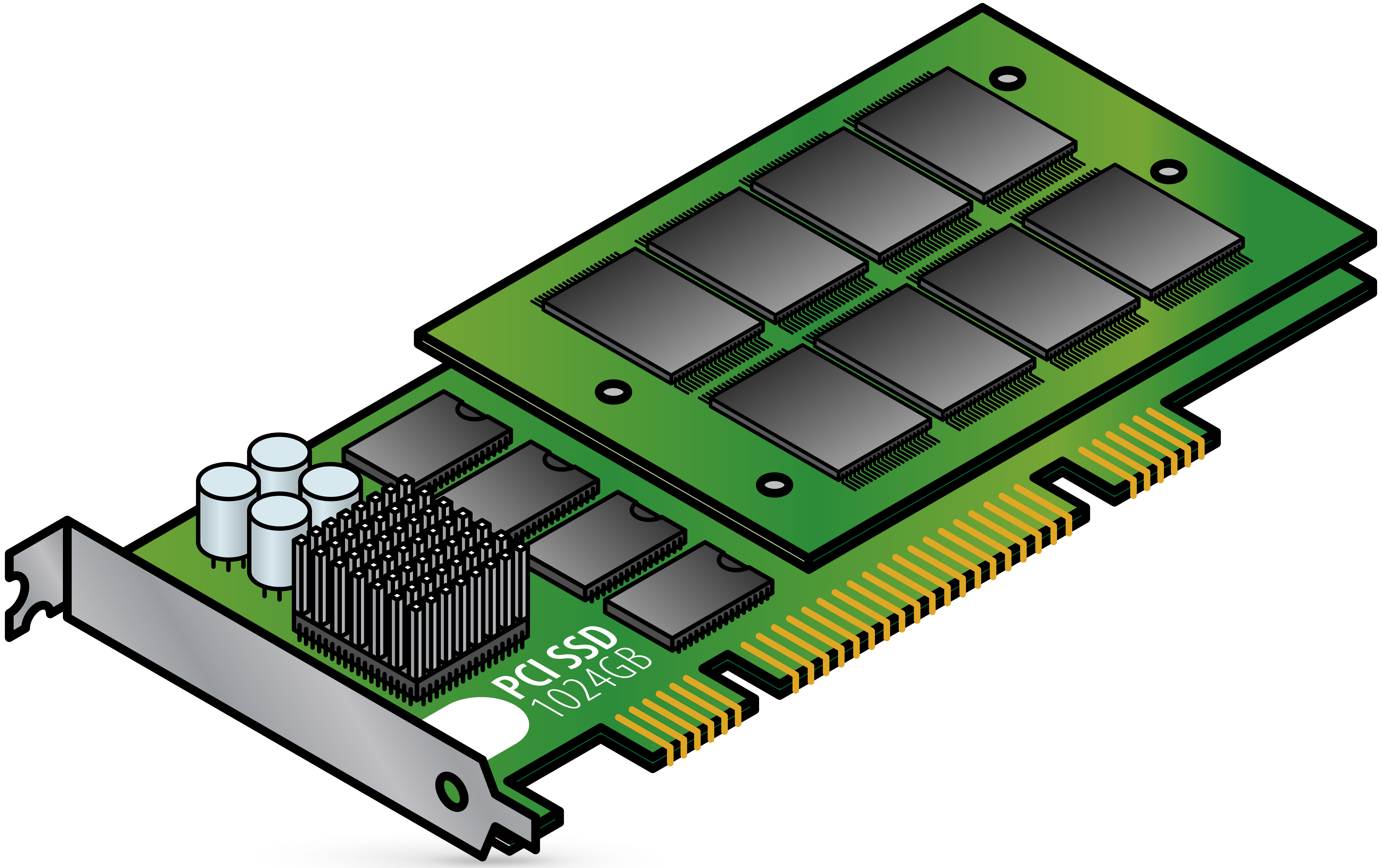
- SSDs with the most reads

READ DISTURBANCE ERRORS



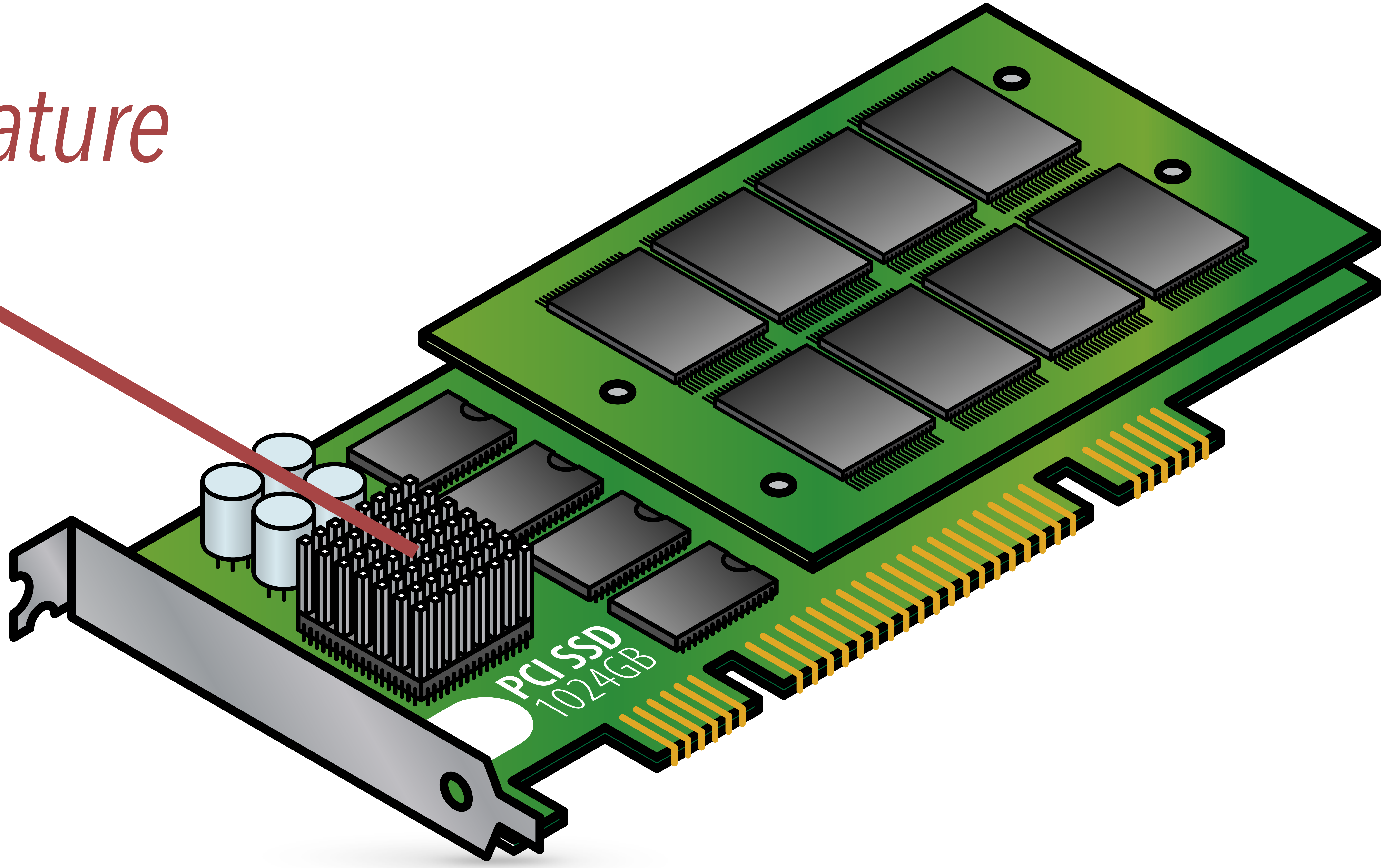
- SSDs with the most reads
- No statistically significant difference at low data read versus high data read

TEMPERATURE DEPENDENCE

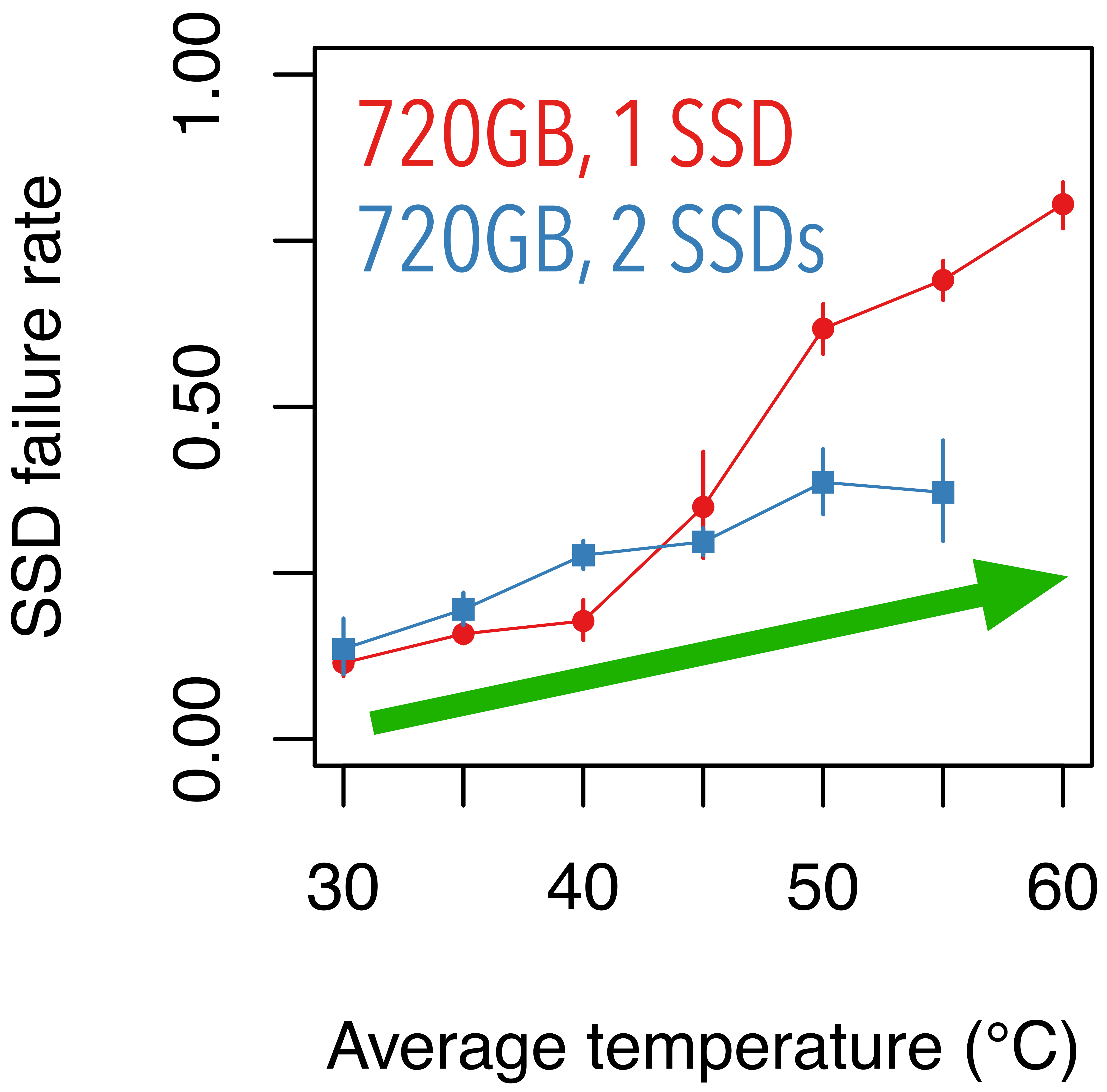


TEMPERATURE DEPENDENCE

Temperature sensor

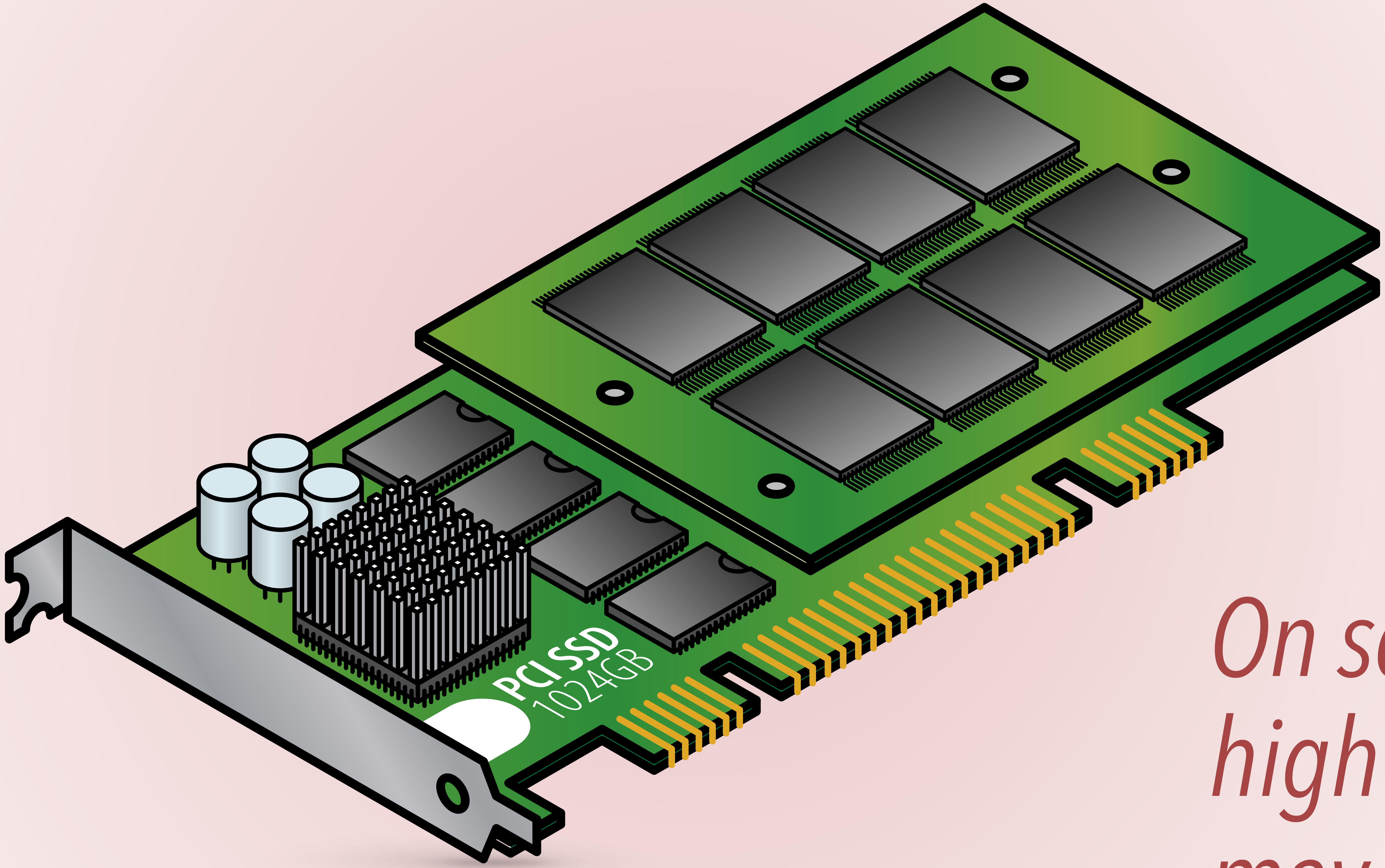


TEMPERATURE DEPENDENCE



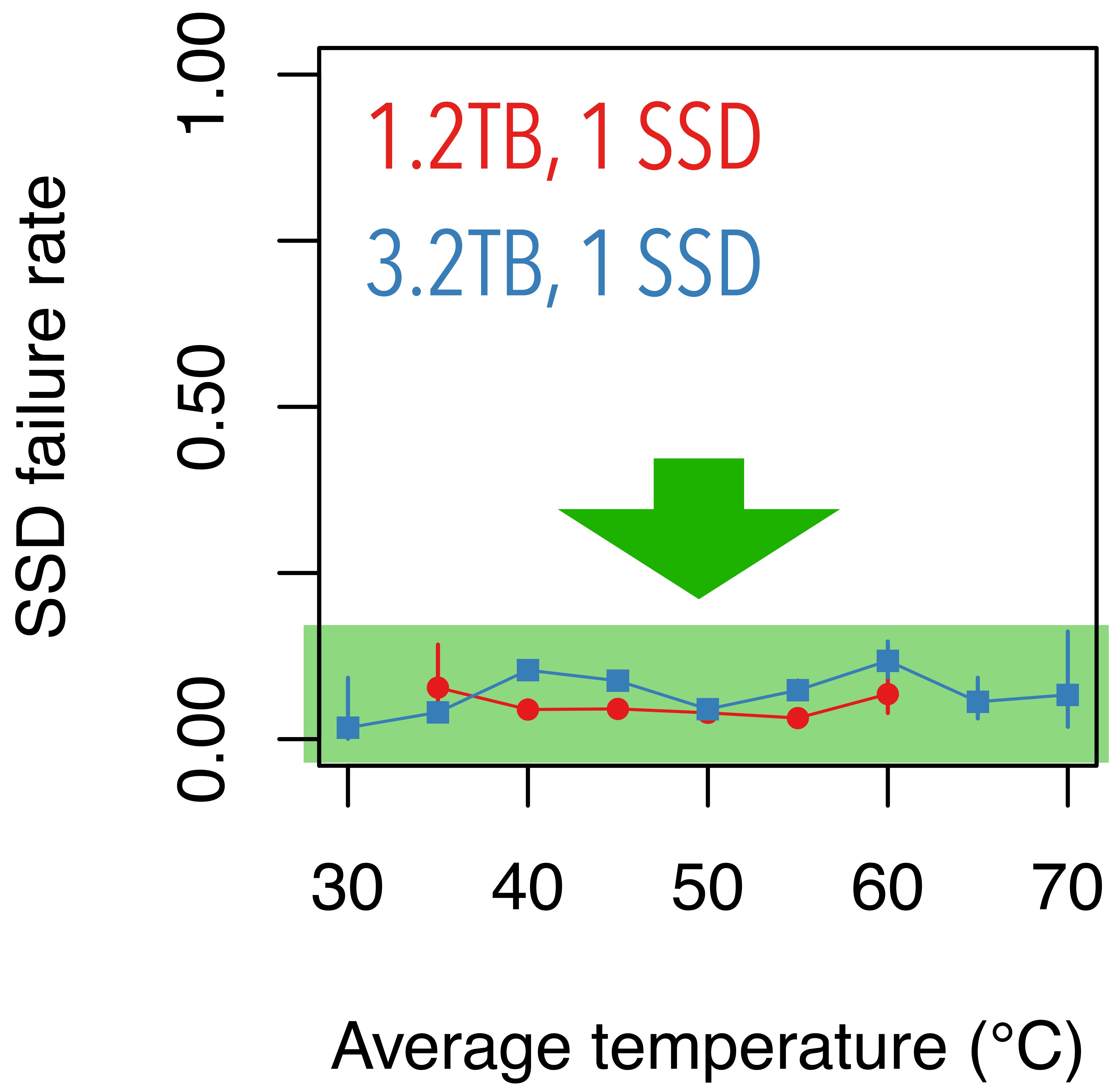
- Higher temperature = more failures

TEMPERATURE DEPENDENCE



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

Access patterns and SSD writes

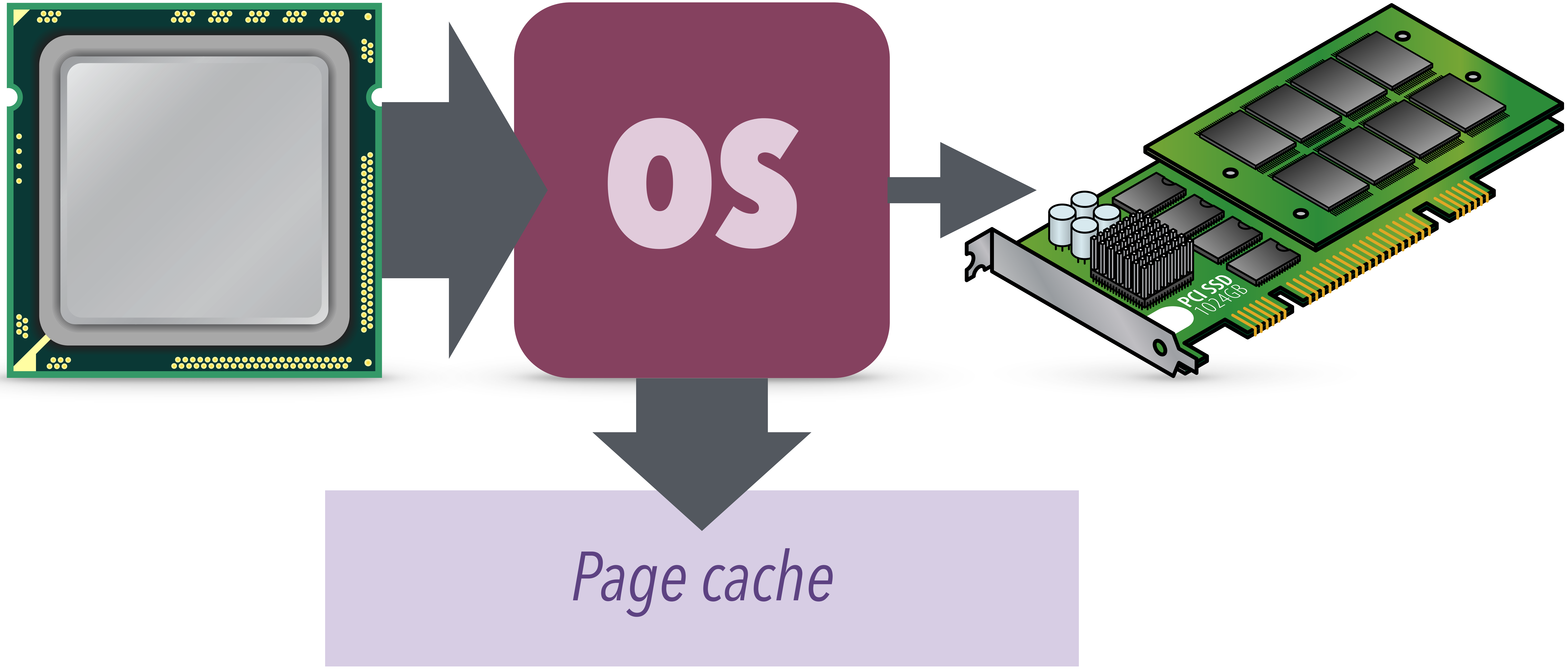
System buffering

- Data served from OS caches
- Decreases SSD usage

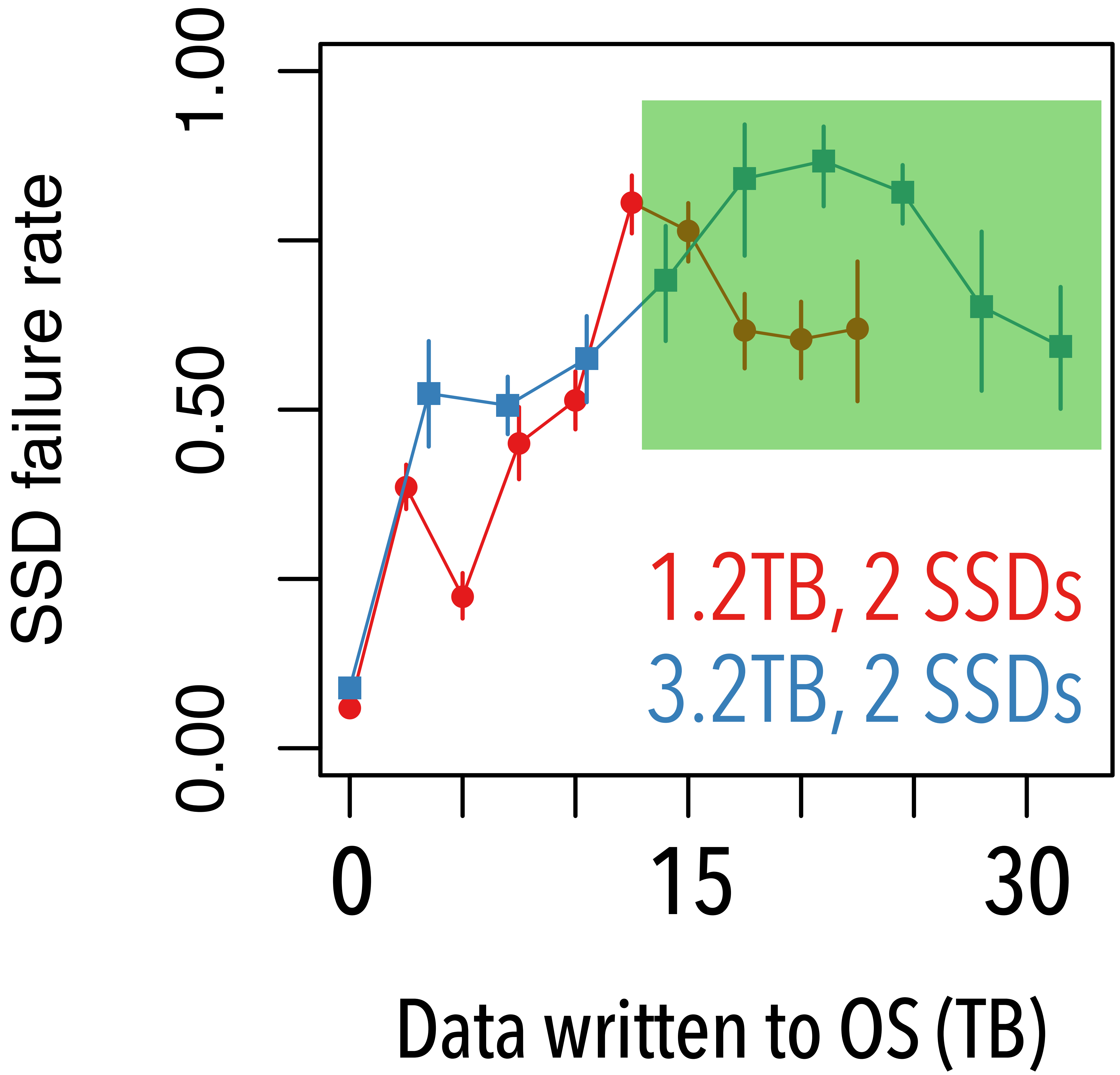
Write amplification

- Updates to small amounts of data
- Increases erasing and copying

System caching reduces the impact of SSD writes

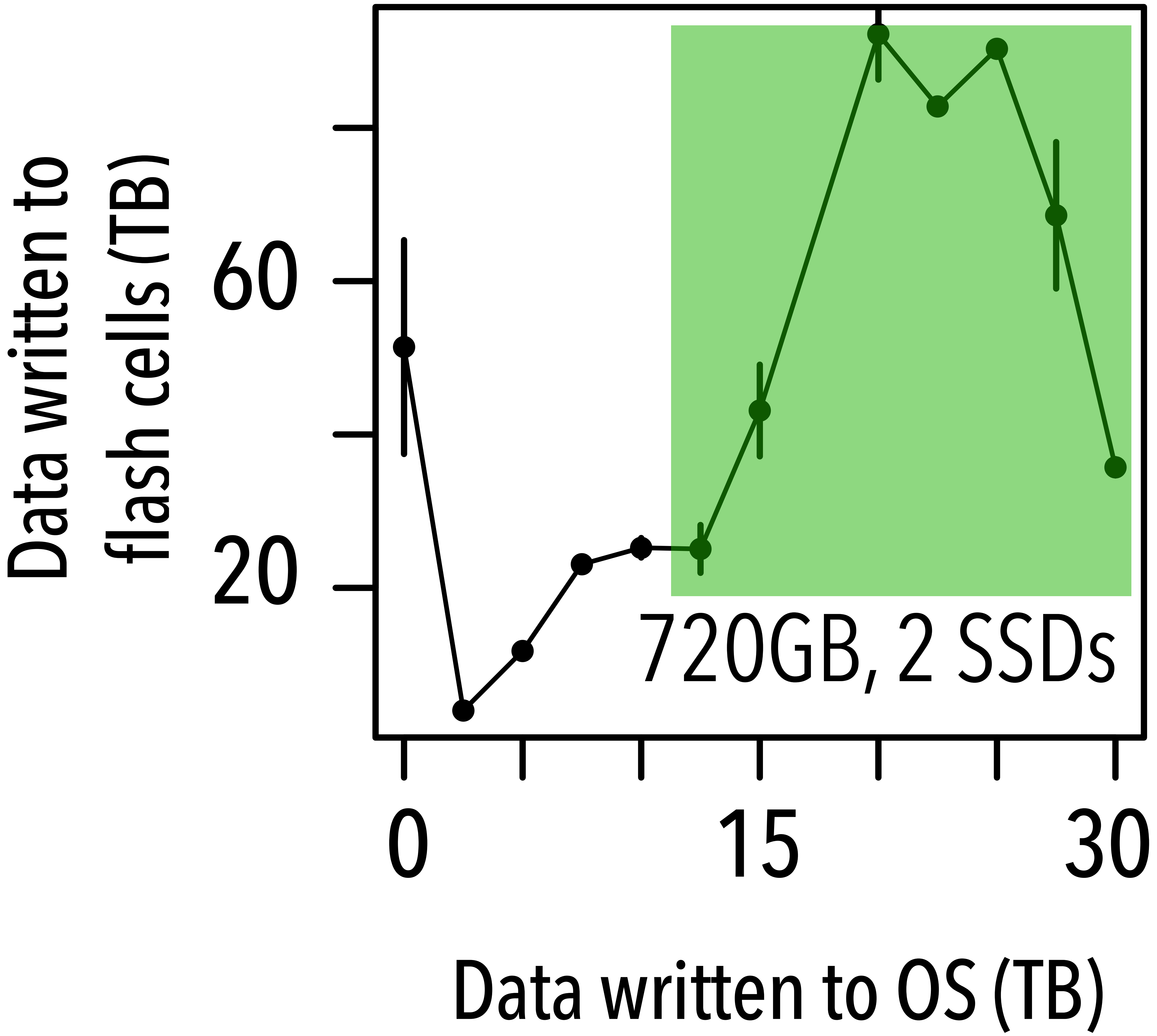


OS WRITES MISLEADING



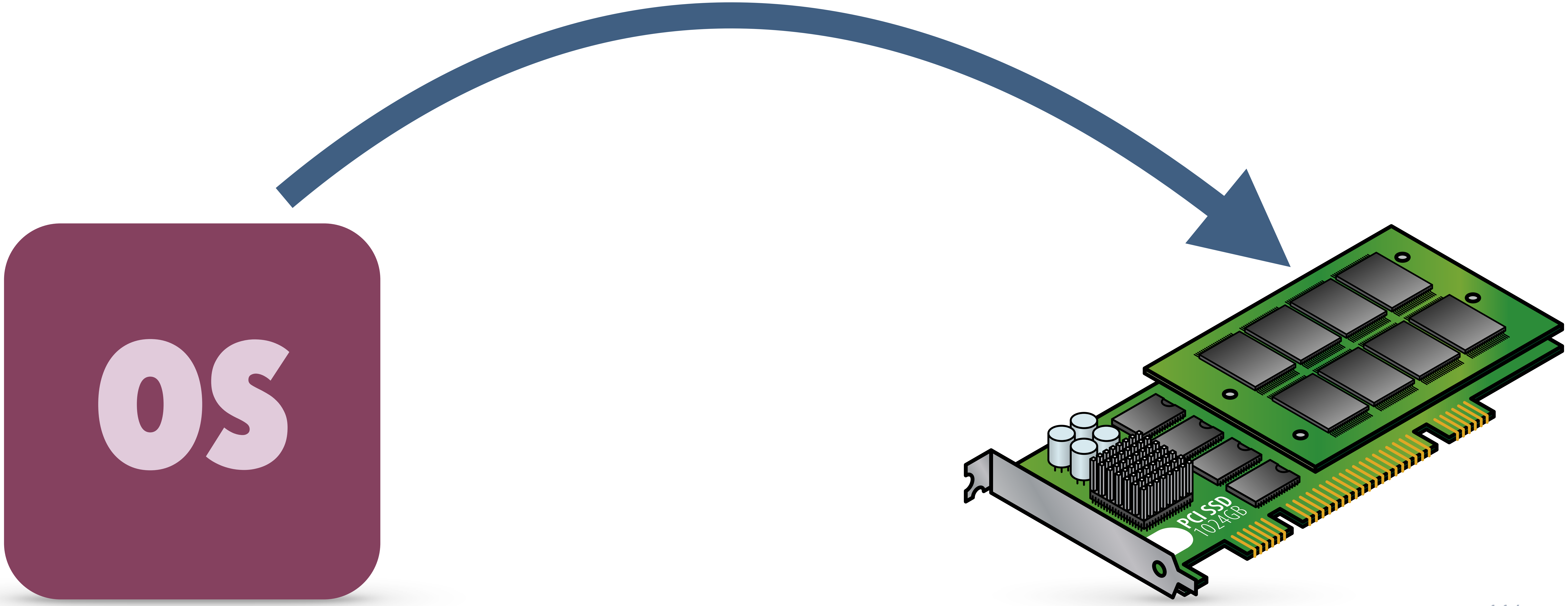
- No statistically significant correlation with failures at high write volume

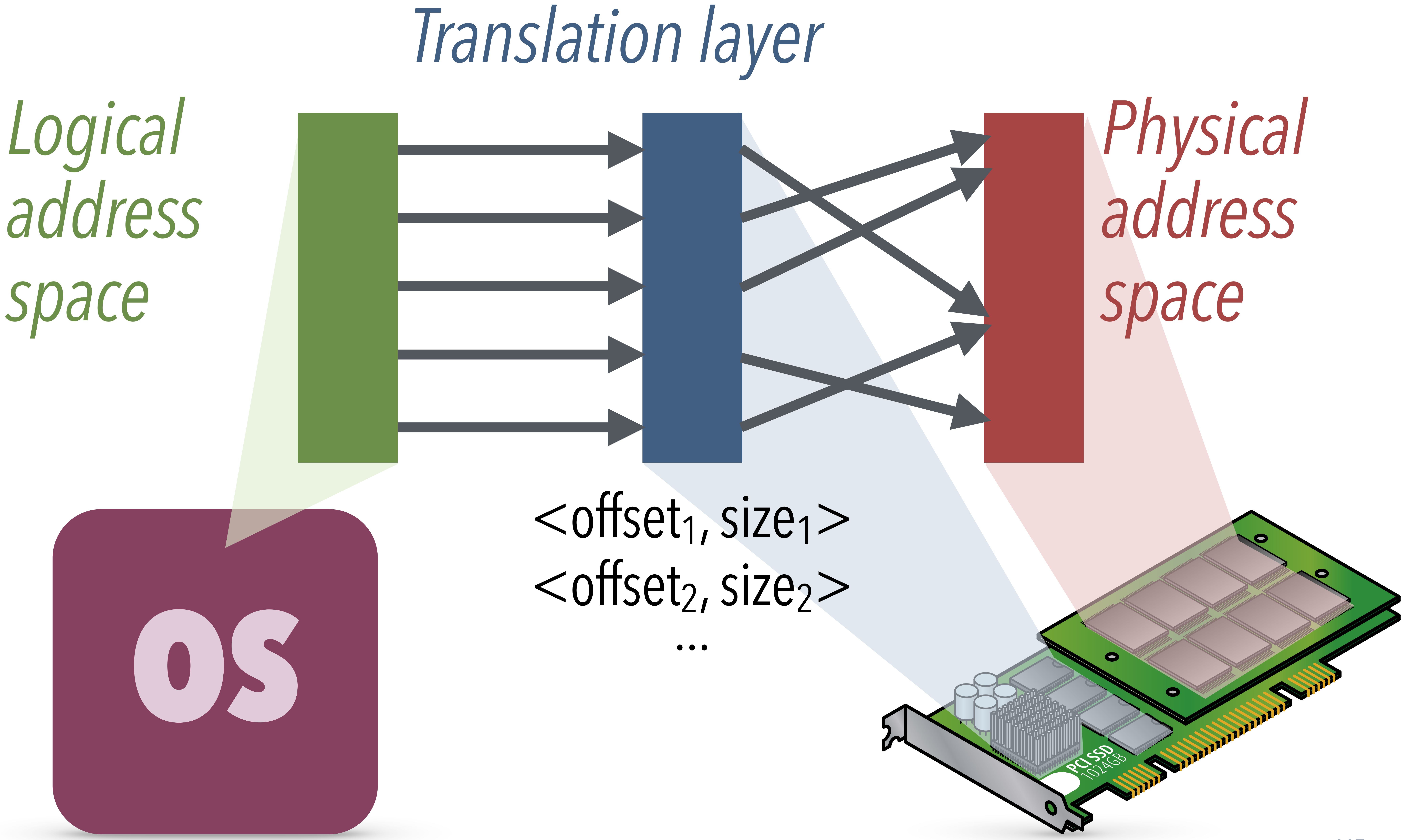
OS WRITES MISLEADING



- 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*

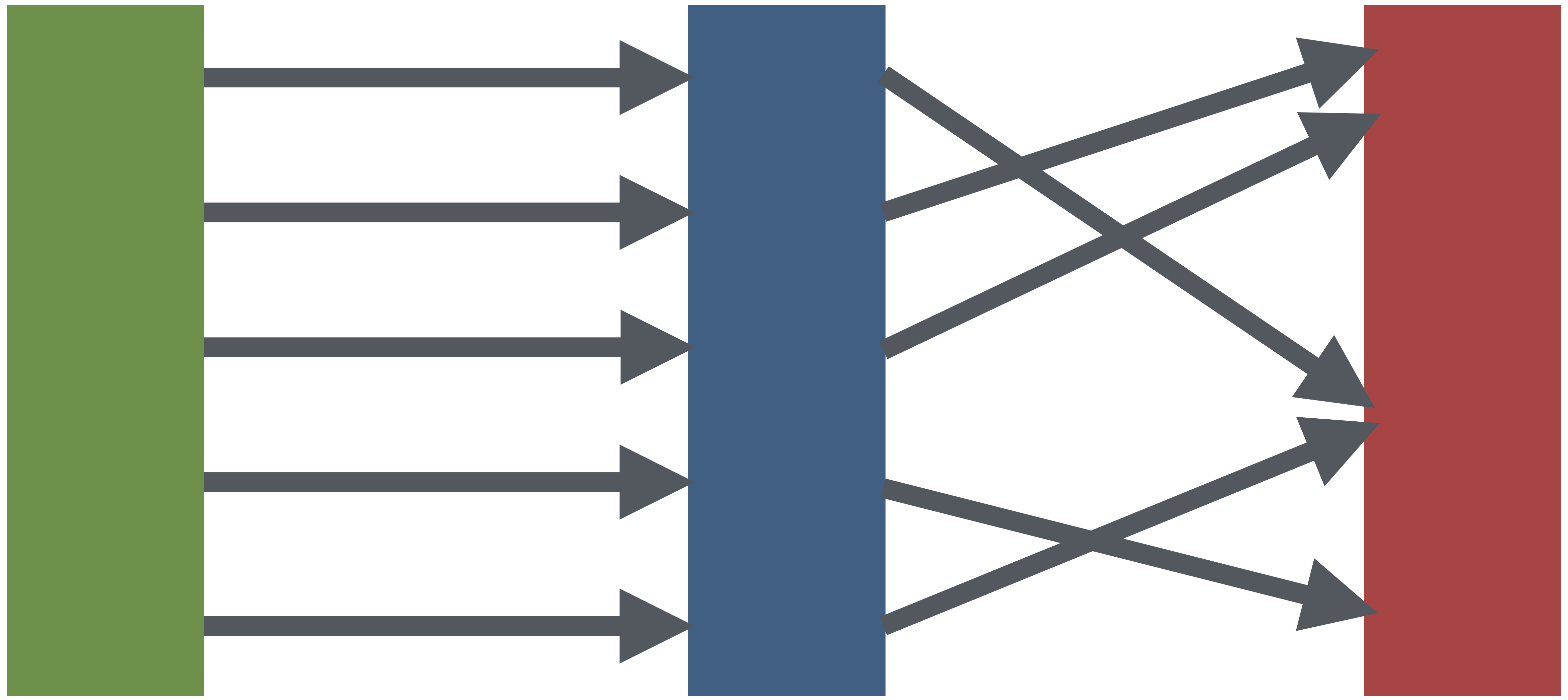



Translation layer
Logical address space
Physical address space
<offset$_1$, size$_1$>
<offset$_2$, size$_2$>
...
OS
PCI SSD
1024GB

Sparse data layout

more *translation metadata*

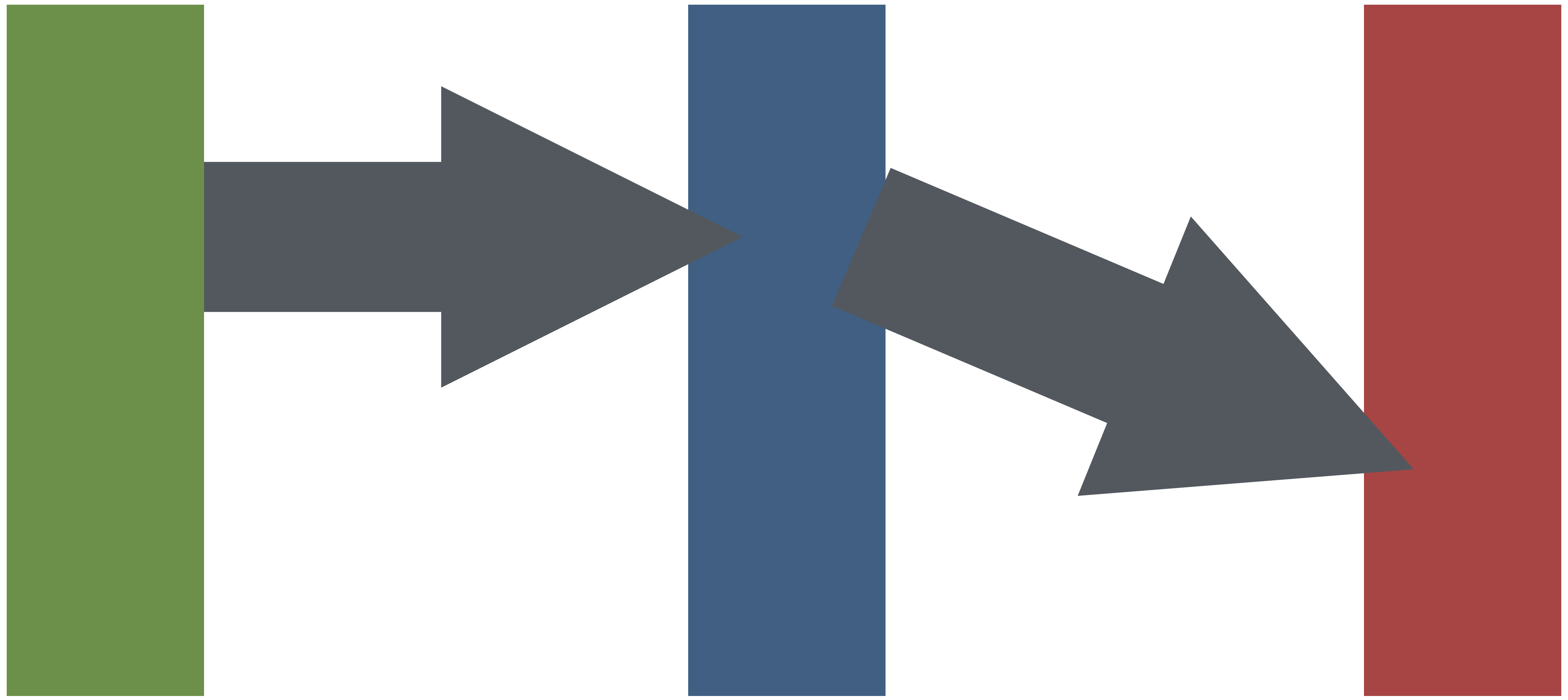
potential for ***higher*** **write amplification**



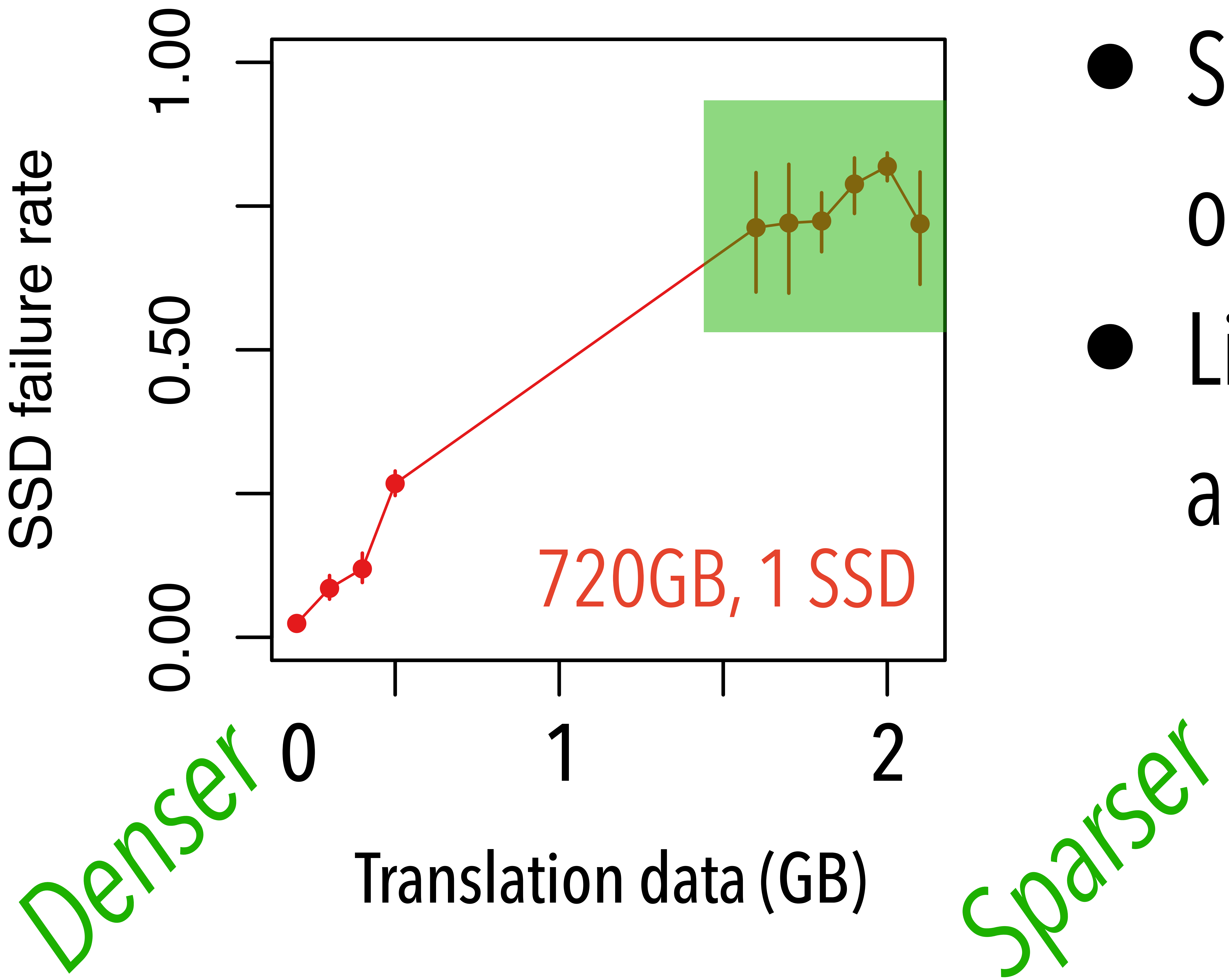
Dense **data layout**

less *translation metadata*

potential for ***lower*** **write amplification**



WRITE AMPLIFICATION



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

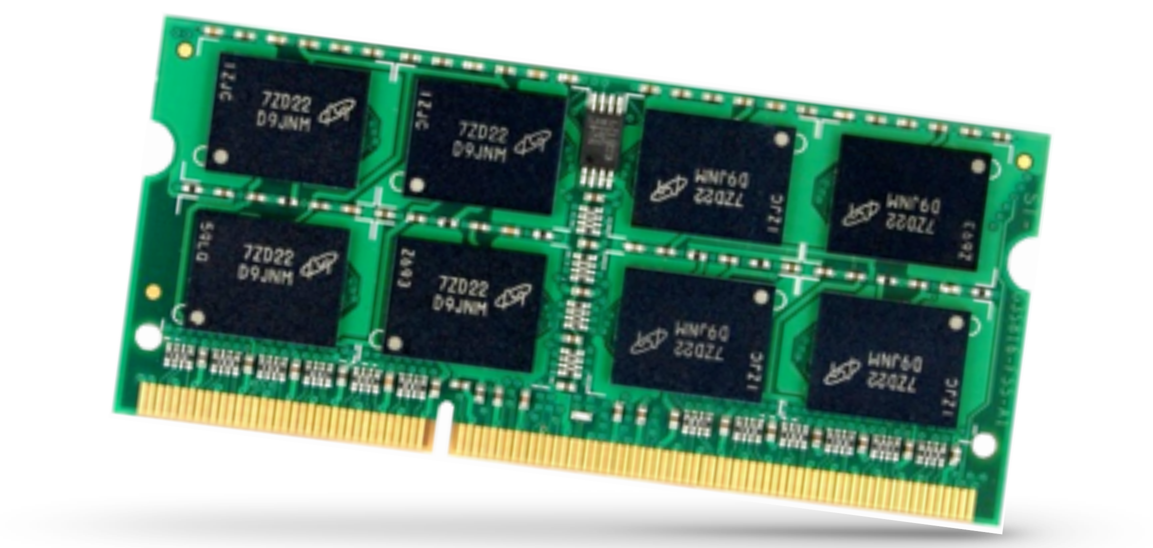
KEY SSD CONTRIBUTIONS

- 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

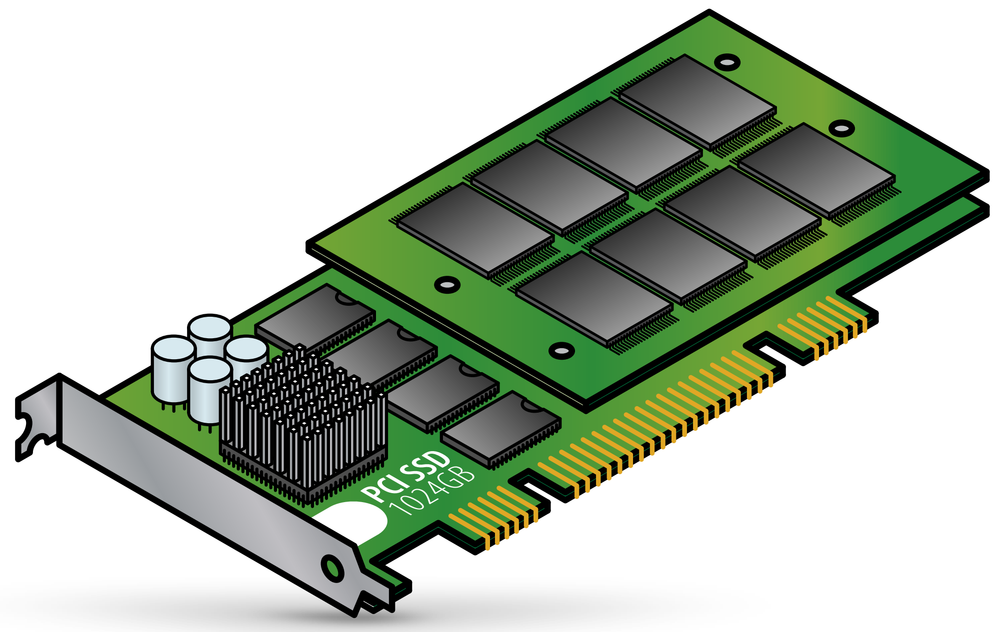
- ***Examined chip-level failures***
  E.g., [Cai+ DATE'12, ICCD'12, DATE'13, ICCD'13, DSN'15, HPCA'17]
- ***Examined a simulated SSD controller with 45 flash chips***
  [Grupp+ FAST'12]
- ***Reliability of SSD controllers (NOT chips)***
  [Ouyang+ ASPLOS'14]
- ***Microsoft and Google SSDs over multiple years***
  [Narayanan+ SYSTOR'16, Schroeder+ FAST'16]

LARGE SCALE STUDIES



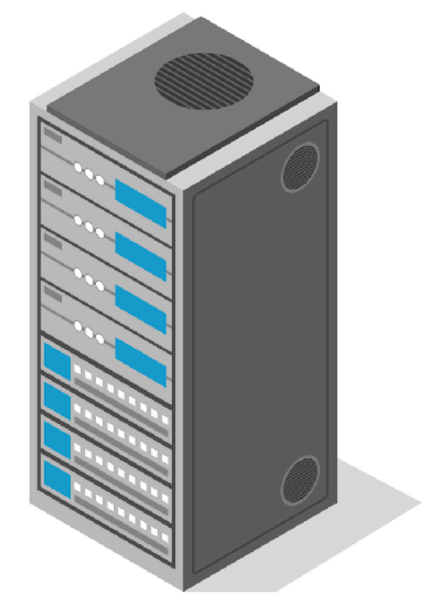
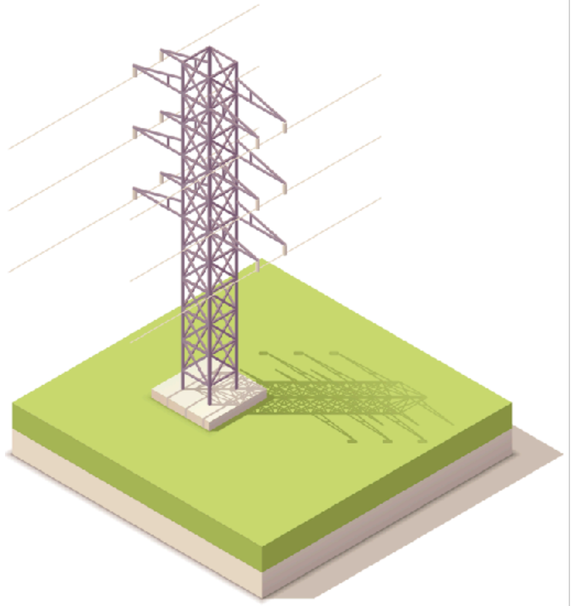
DRAM

[DSN '15]



SSDs

[SIGMETRICS '15]

Networks

[IMC '18]

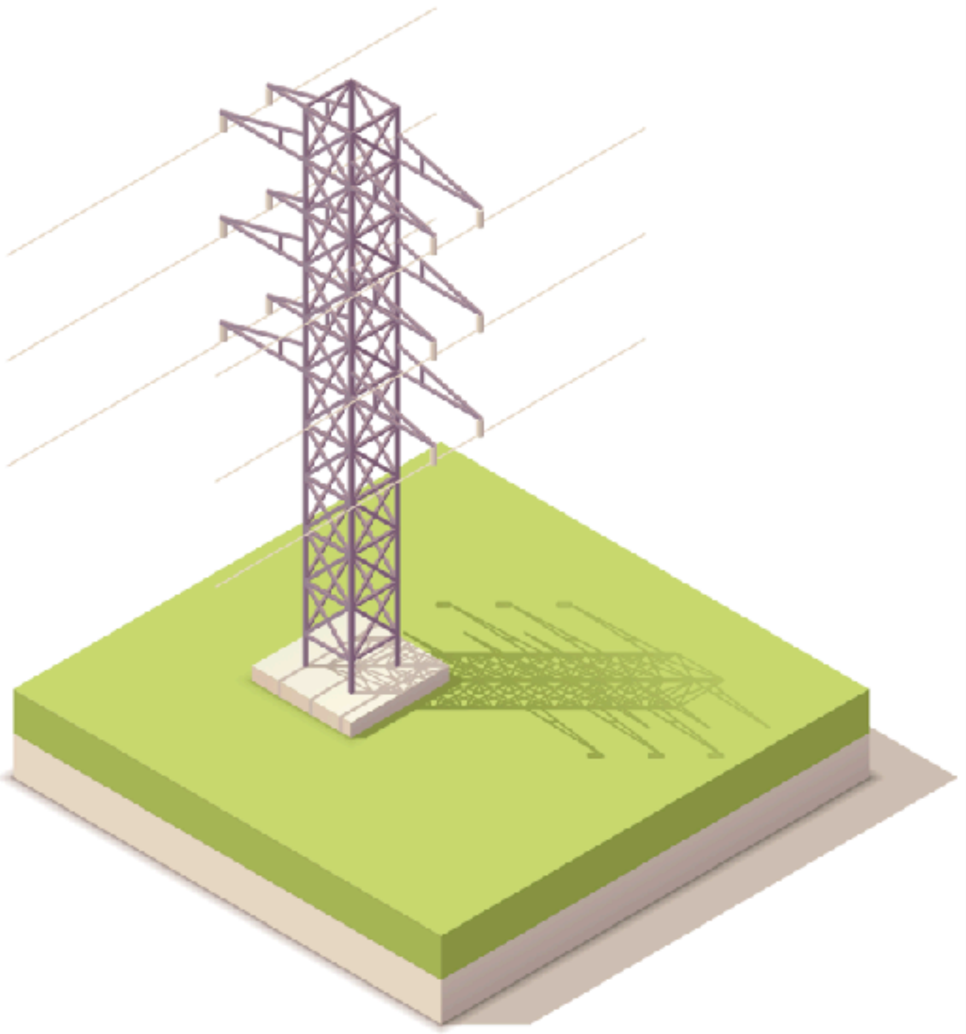

WAN
Internet
ISP
Edge Node
Core Switches
Data Center Fabric
Rack Switch

SOFTWARE-AIDED NETWORKS

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

MEASURING NETWORK FAILURE

DATA CENTER
NETWORK



WIDE AREA
NETWORK

- ***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

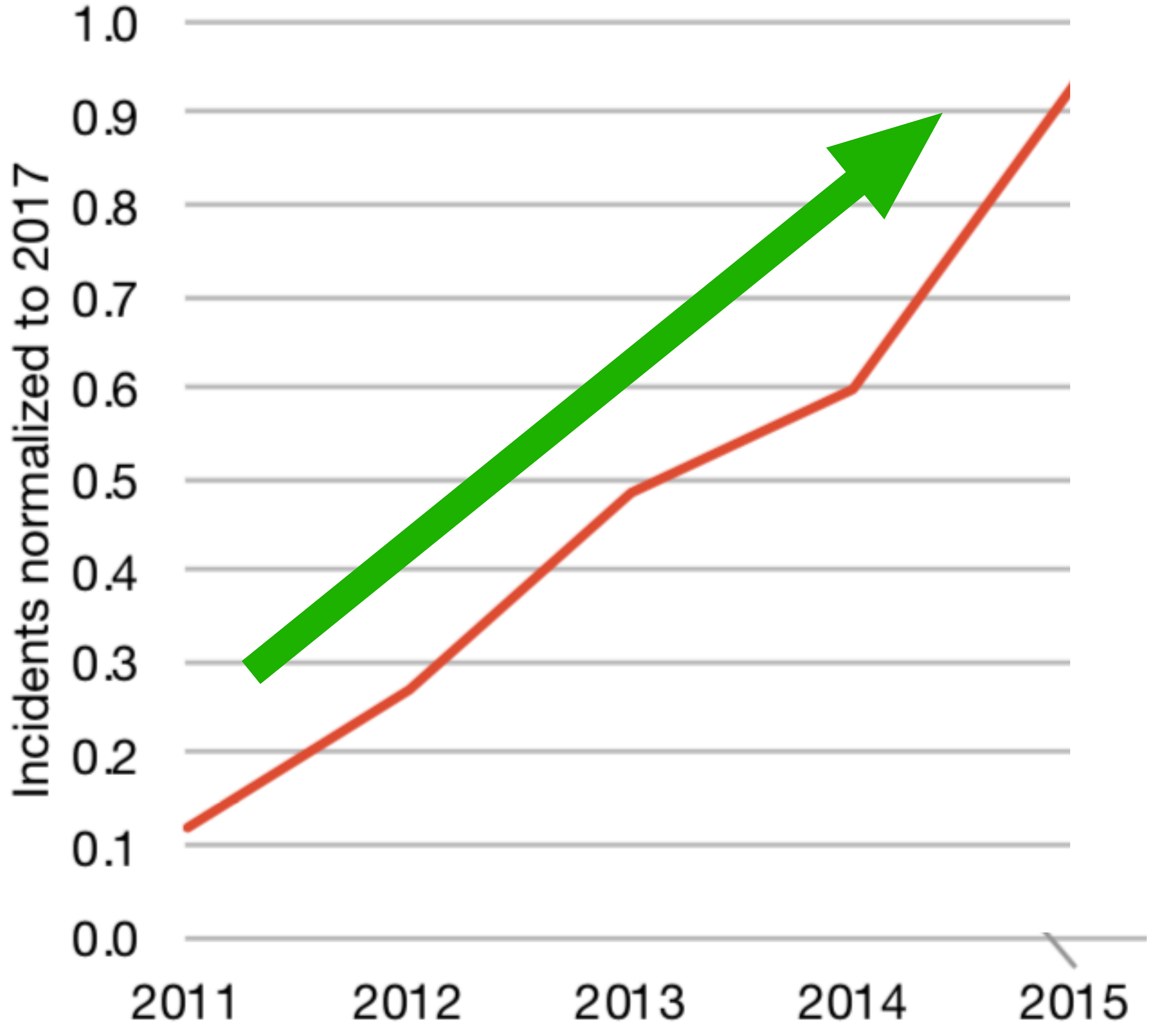
that result in

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

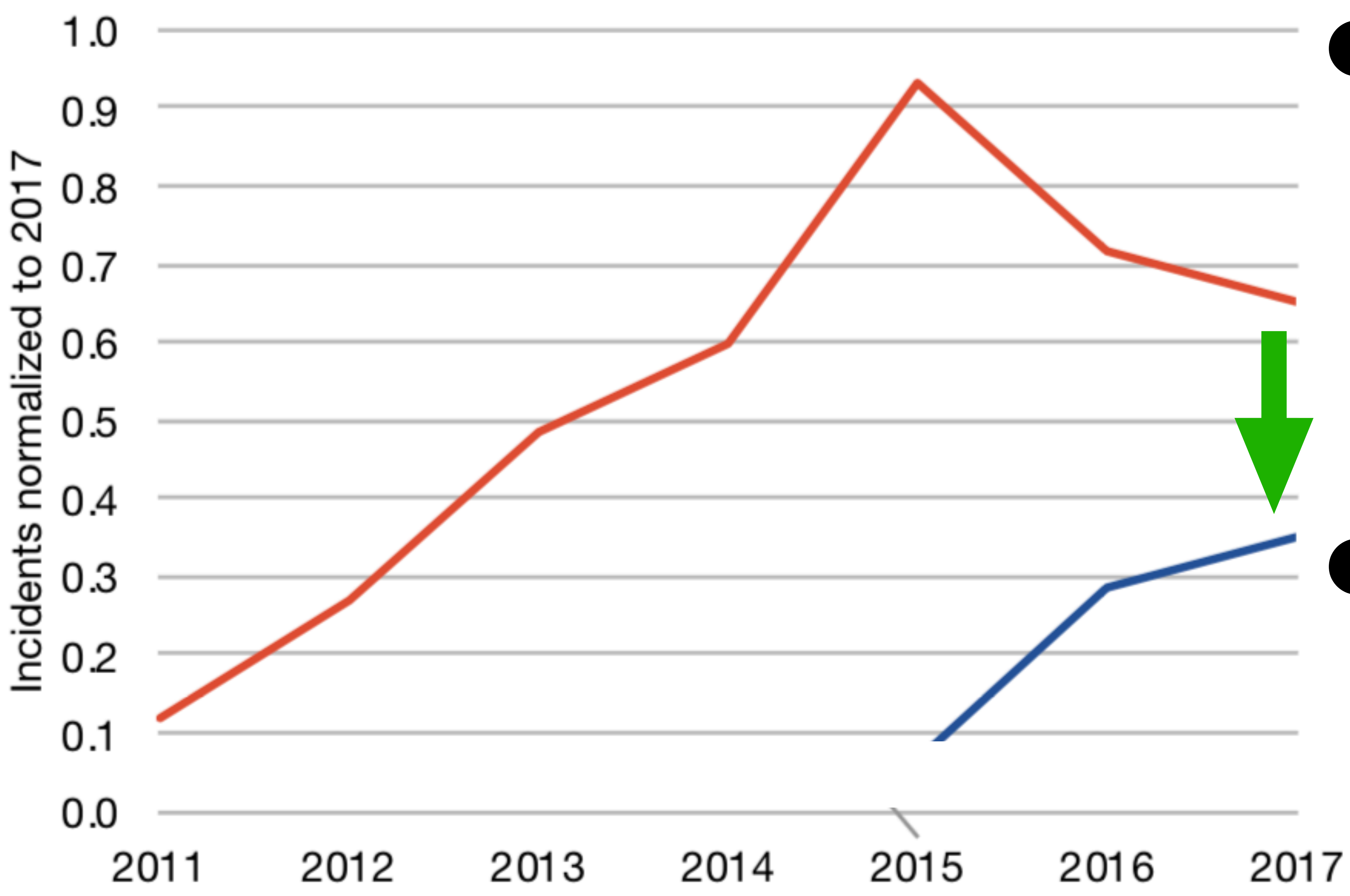
NETWORK DESIGN TRENDS



Hard-wired network

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

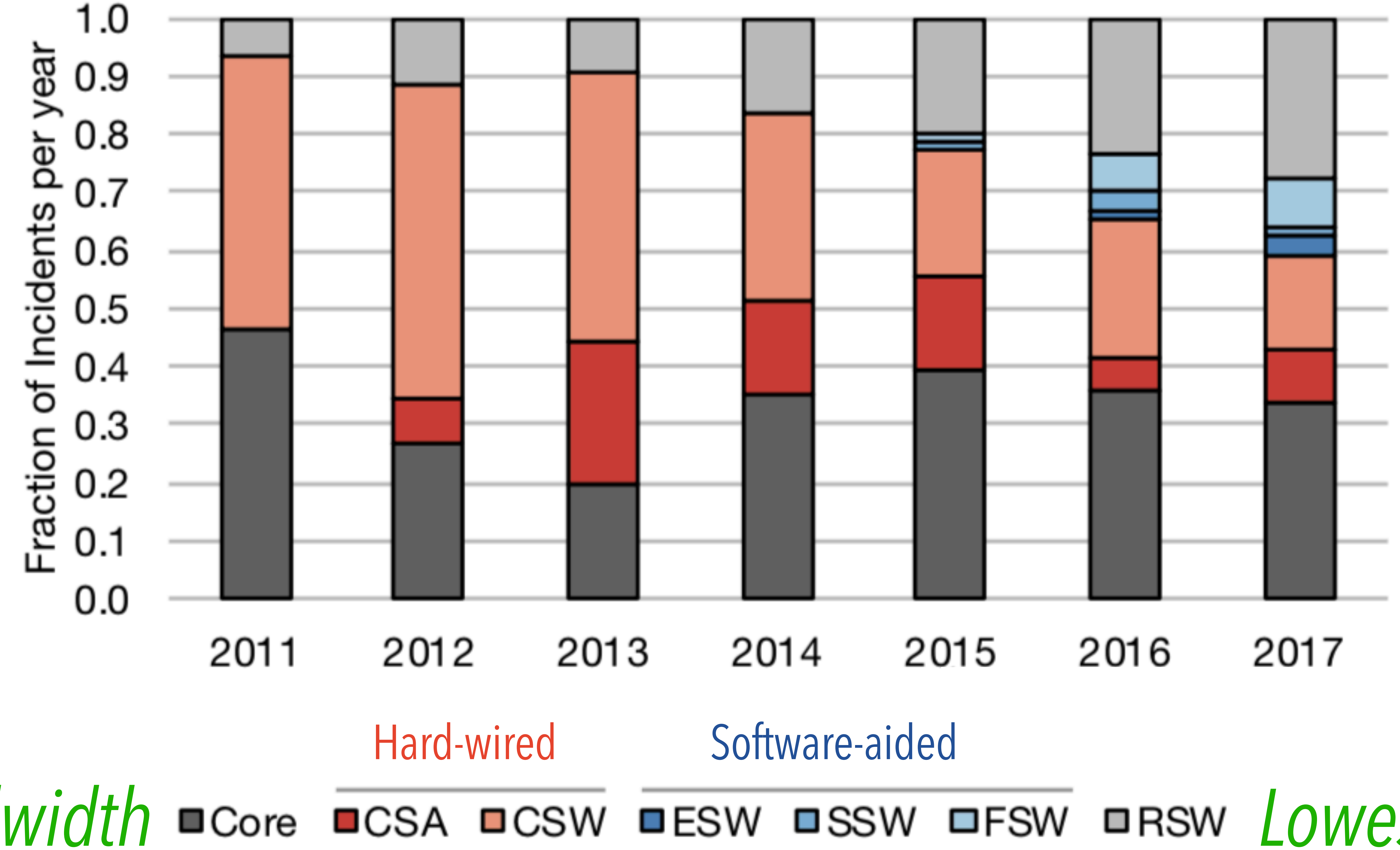
NETWORK DESIGN TRENDS



- 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



Hard-wired

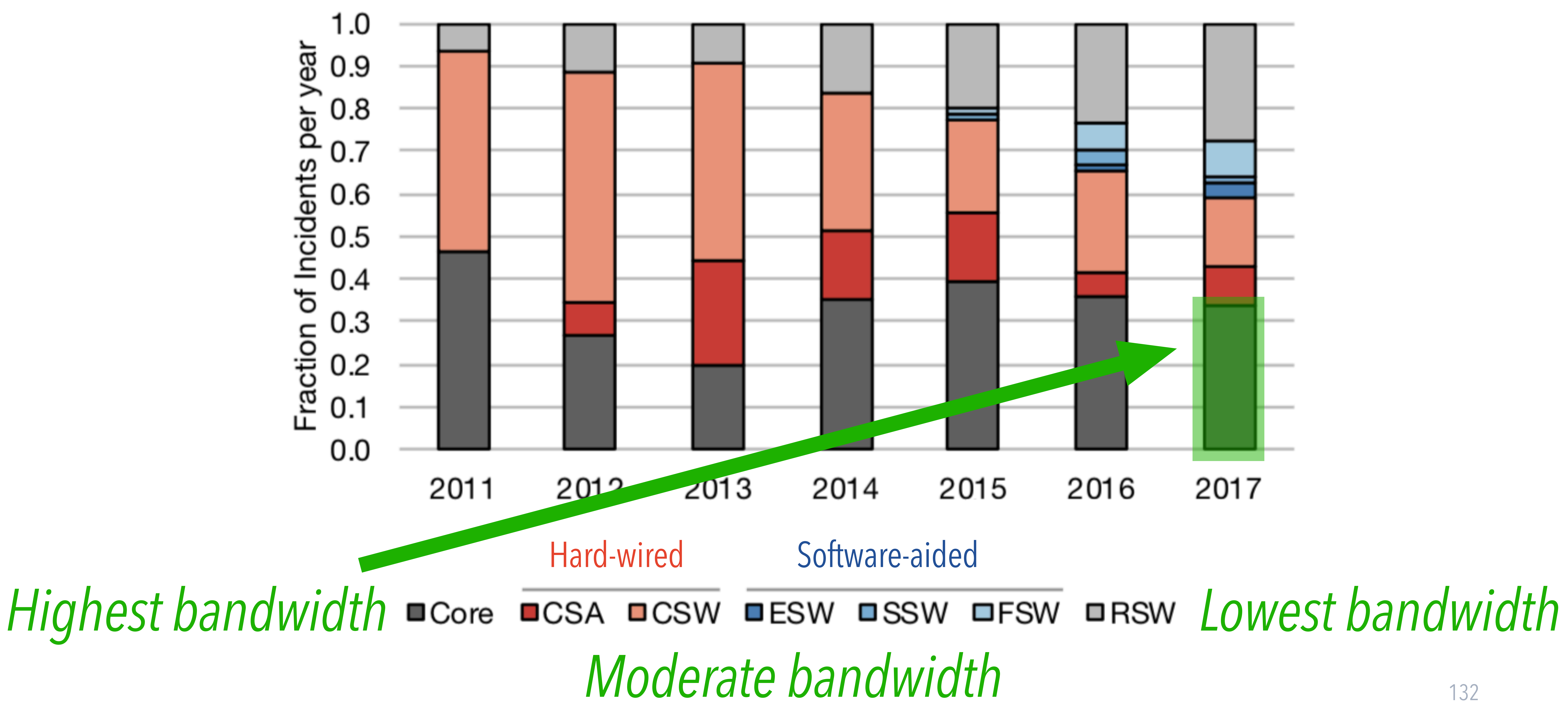
Software-aided

Highest bandwidth

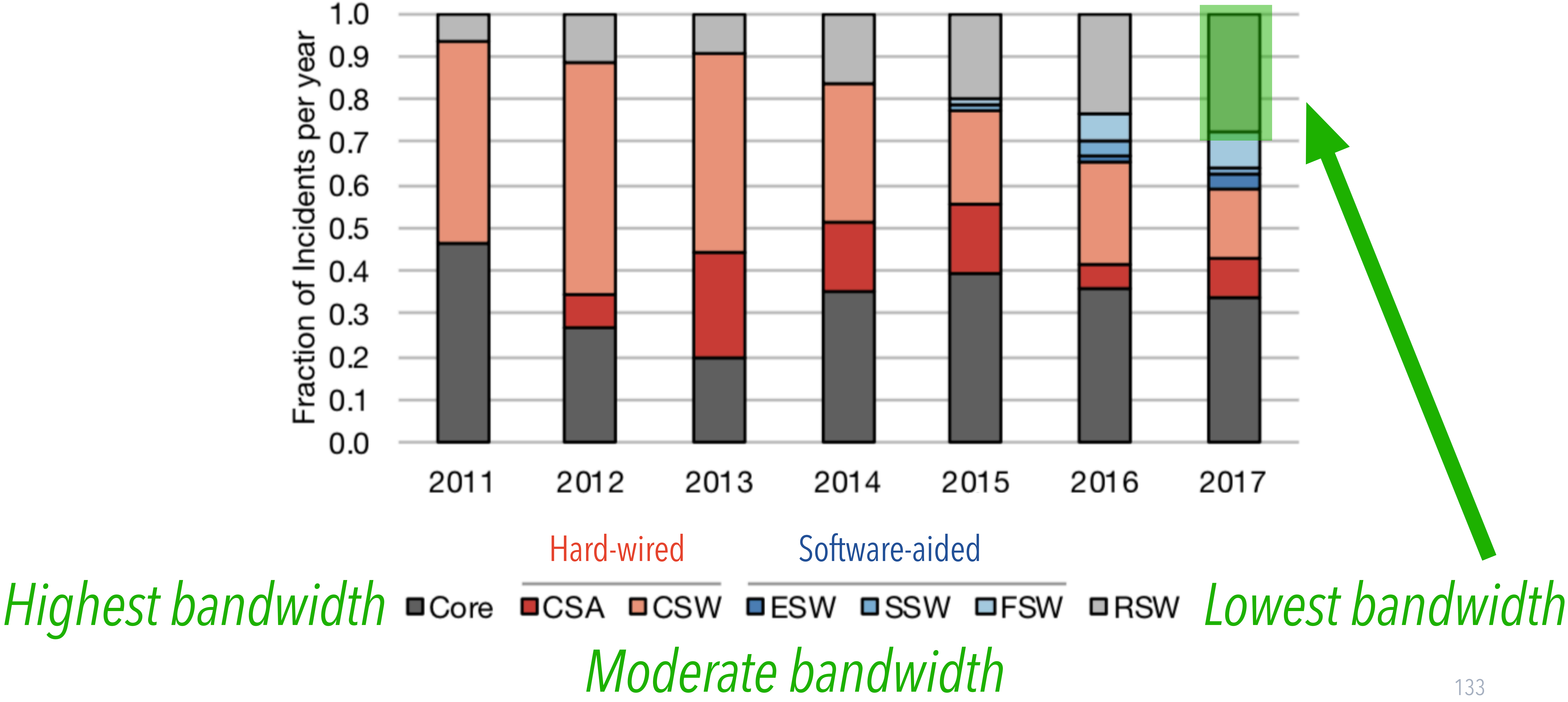
Lowest bandwidth

Moderate bandwidth

SWITCH TYPE TRENDS



SWITCH TYPE TRENDS



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

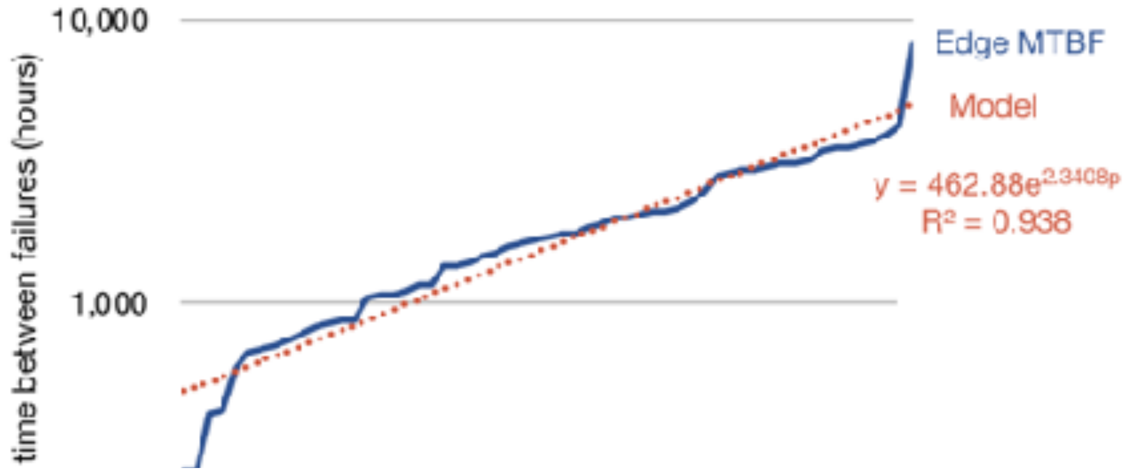
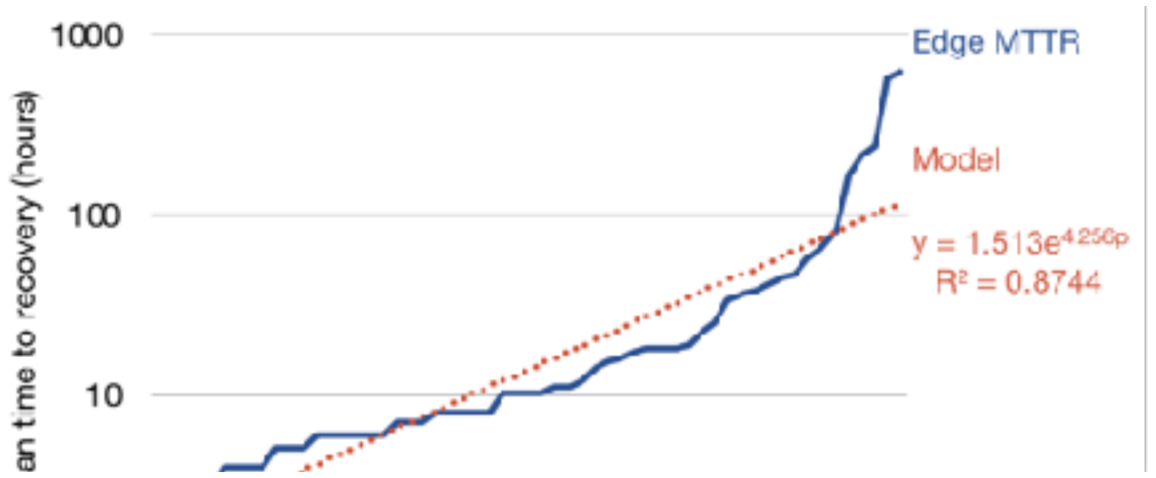
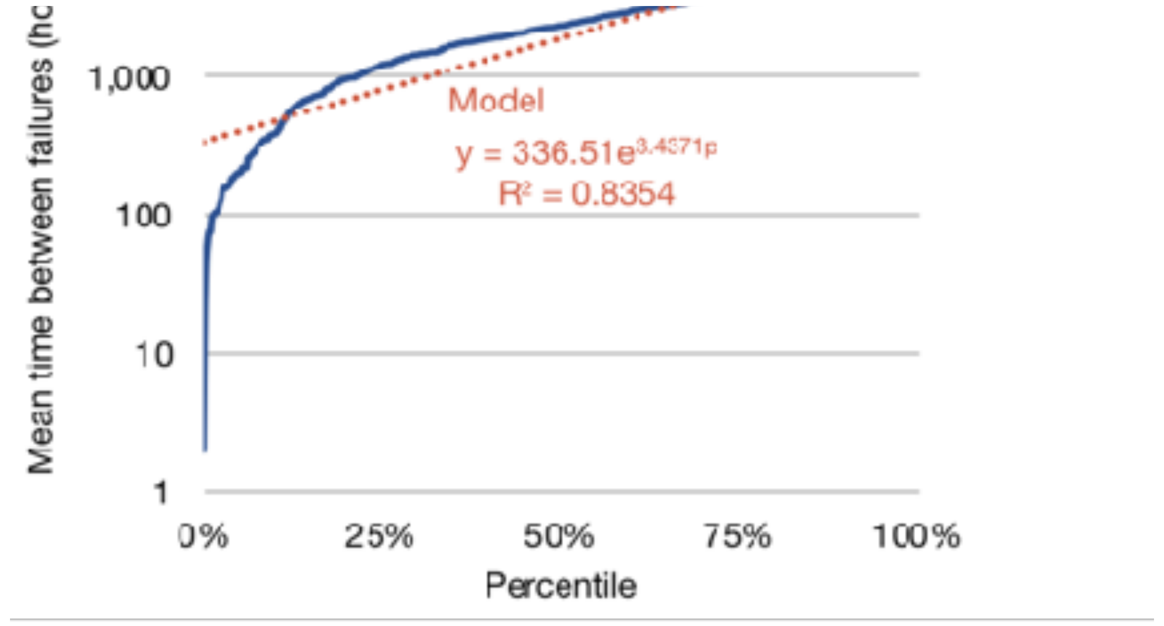
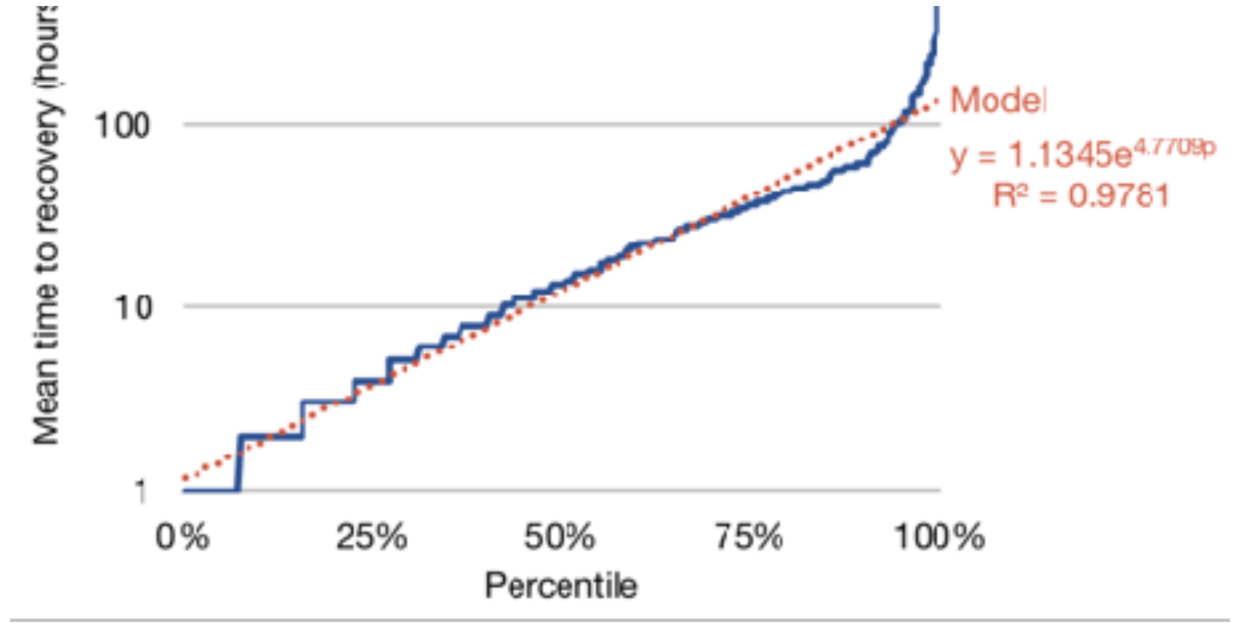
MODELING WAN RELIABILITY

	Failure rate	Repair rate
Edge		
Link		

MODELING WAN RELIABILITY

	Failure rate	Repair rate
Edge	*O(months)*	*O(hours)*
Link	*O(months)*	*O(days)*

MODELING WAN RELIABILITY

	Failure rate	Repair rate
Edge		
Link		

— *We provide open models* —

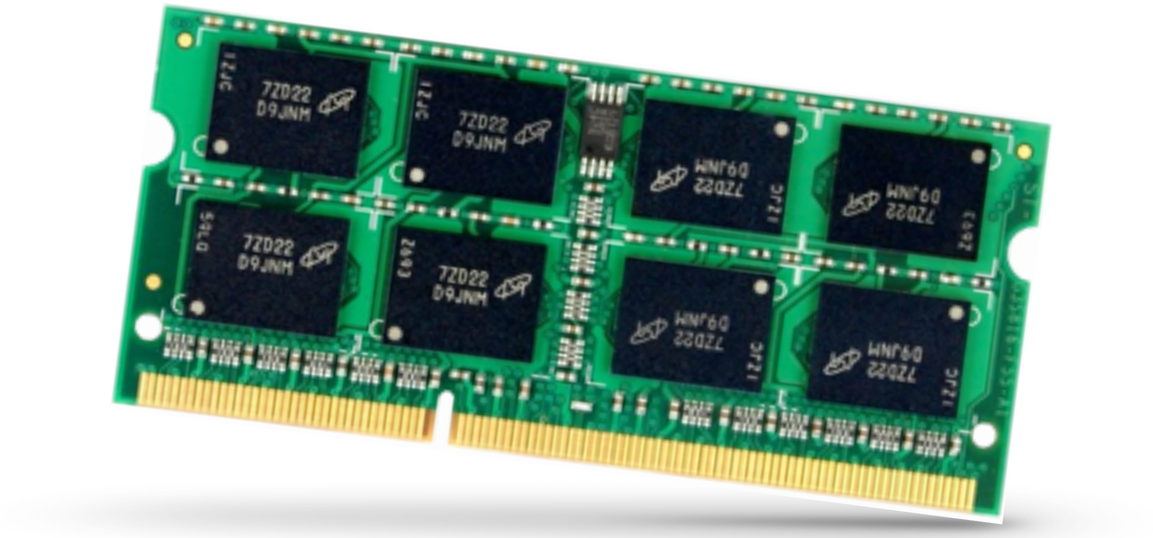
KEY NETWORK CONTRIBUTIONS

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

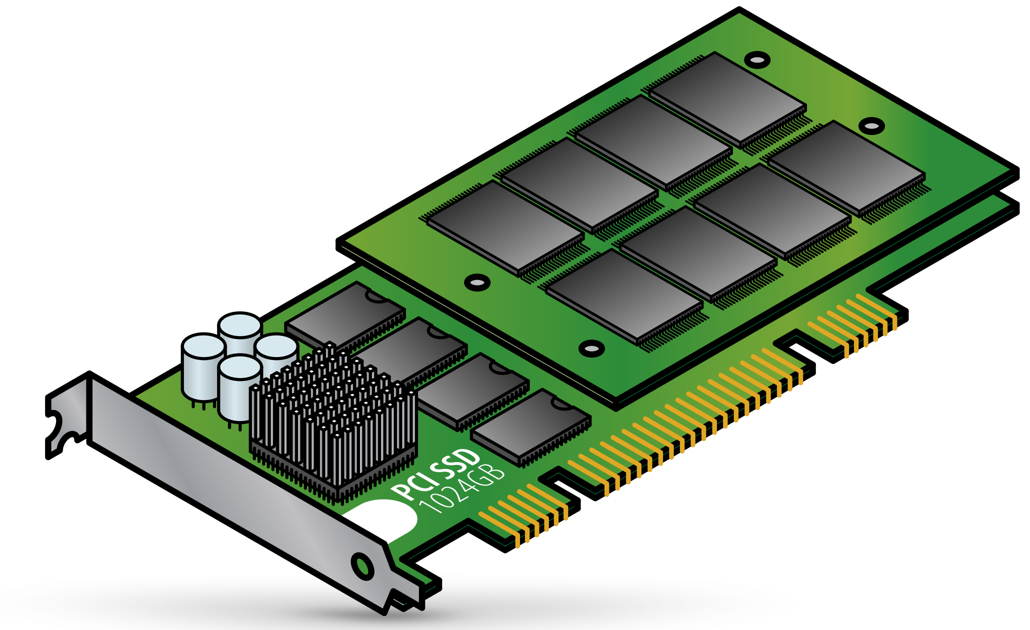
RELATED WORK

- ***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]

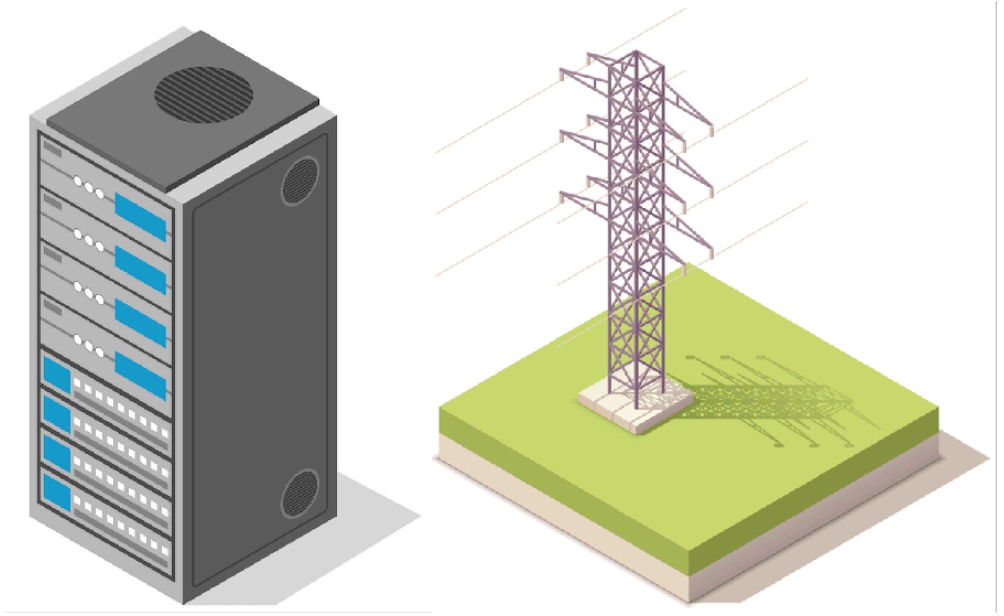
LARGE SCALE STUDIES



DRAM
[DSN '15]



SSDs
[SIGMETRICS '15]



Networks
[IMC '18]

THESIS STATEMENT

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

THESIS ORAL

JUSTIN MEZA

Committee
Prof. Onur Mutlu (Chair)
Prof. Greg Ganger
Prof. James Hoe
Dr. Kaushik Veeraraghavan (Facebook, Inc.)

Carnegie Mellon University

BACKUP SLIDES

More Techniques?

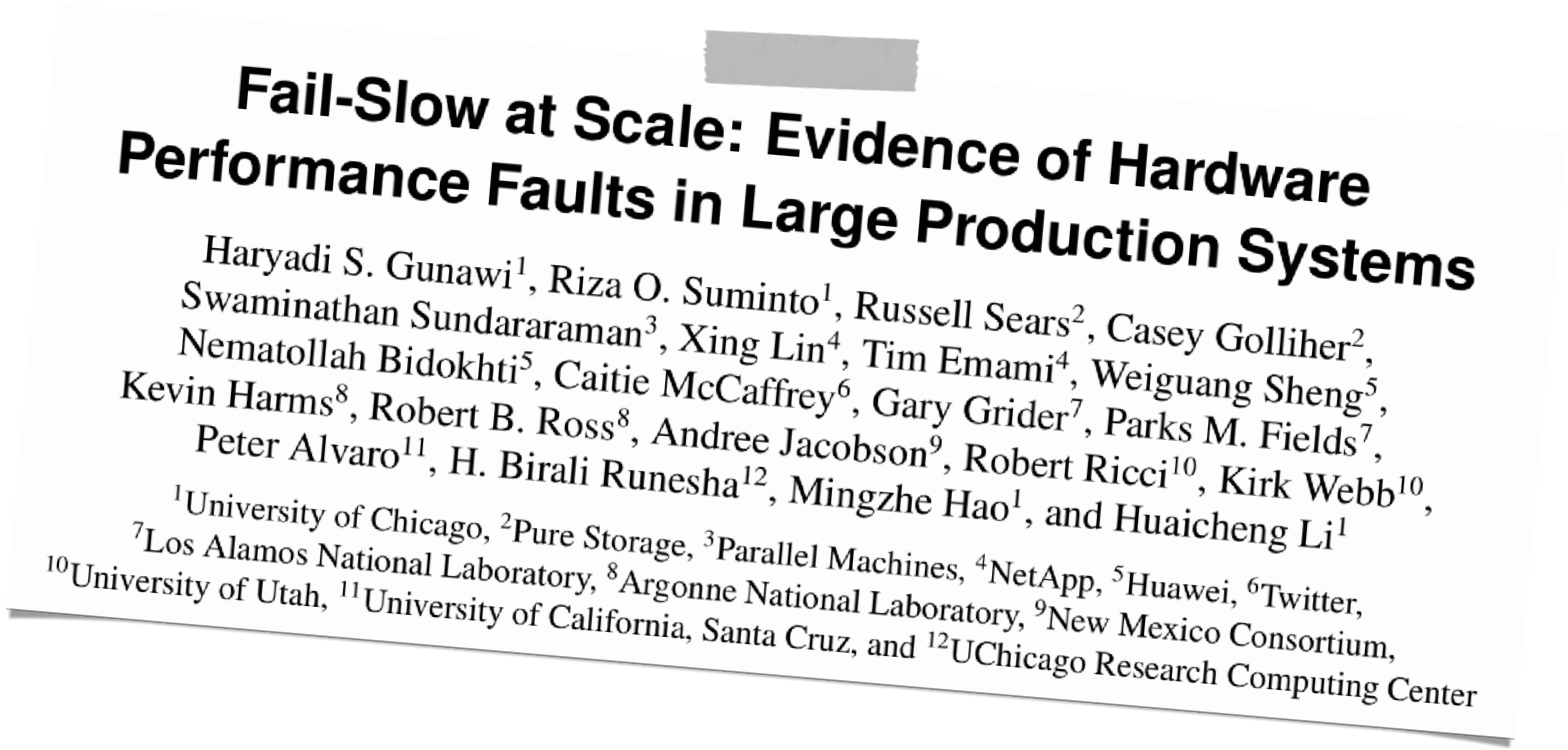
- We believe our DRAM work provides a promising direction
  - Analyze failures, build models, design techniques
- At the same time, we wanted to focus on:
  - Instrumentation + analysis of new devices (SSDs)
  - Going more in depth in software-level effects (networks)
- We sketch how to extend our methodology 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



Fail-Slow at Scale: Evidence of Hardware Performance Faults in Large Production Systems

Haryadi S. Gunawi[1], Riza O. Suminto[1], Russell Sears[2], Casey Golliher[2], Swaminathan Sundararaman[3], Xing Lin[4], Tim Emami[4], Weiguang Sheng[5], Nematollah Bidokhti[5], Caitie McCaffrey[6], Gary Grider[7], Parks M. Fields[7], Kevin Harms[8], Robert B. Ross[8], Andree Jacobson[9], Robert Ricci[10], Kirk Webb[10], Peter Alvaro[11], H. Birali Runesha[12], Mingzhe Hao[1], and Huaicheng Li[1]

[1]University of Chicago, [2]Pure Storage, [3]Parallel Machines, [4]NetApp, [5]Huawei, [6]Twitter, [7]Los Alamos National Laboratory, [8]Argonne National Laboratory, [9]New Mexico Consortium, [10]University of Utah, [11]University of California, Santa Cruz, and [12]UChicago Research Computing Center

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 phemonena

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

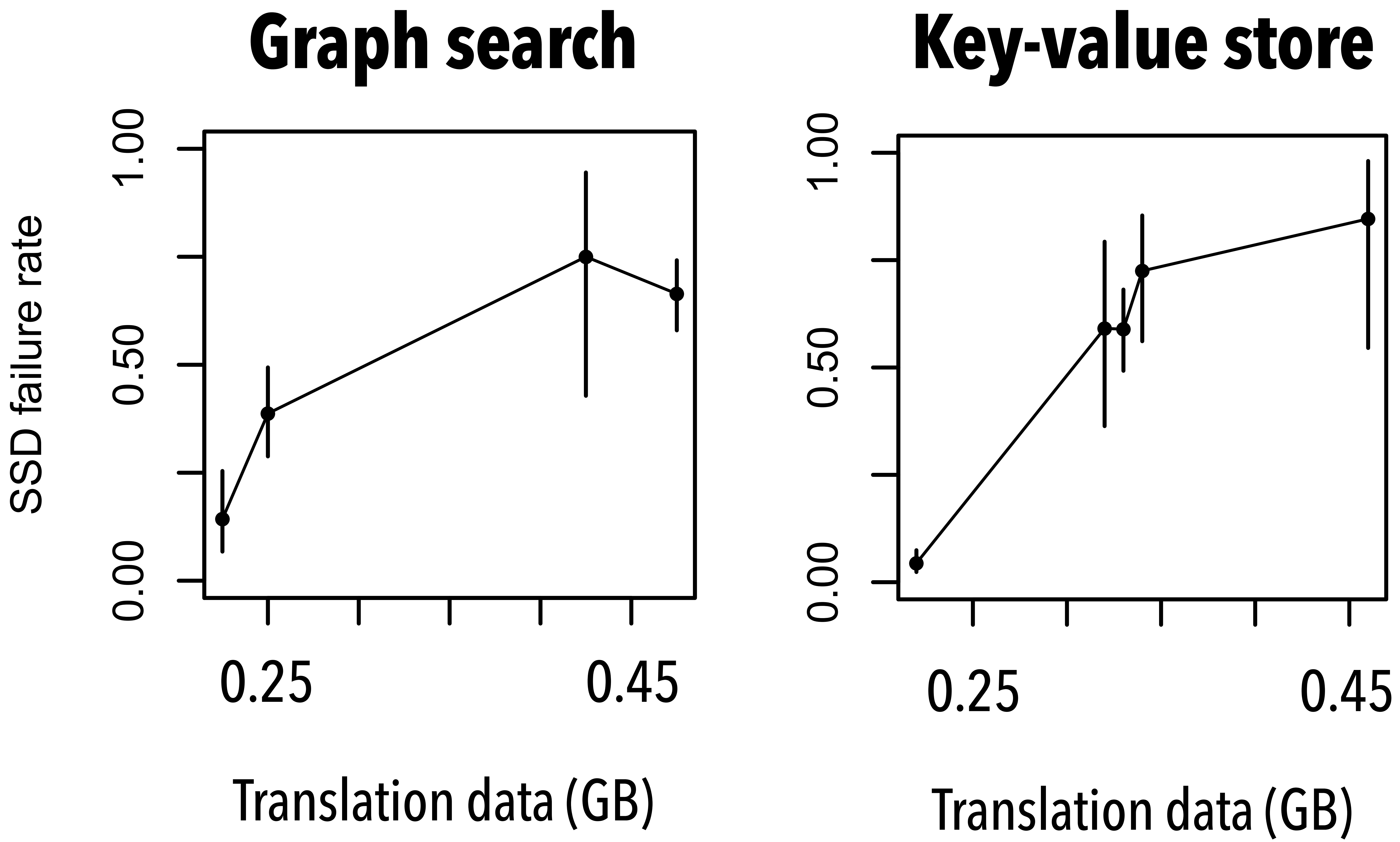
Network Failure Details

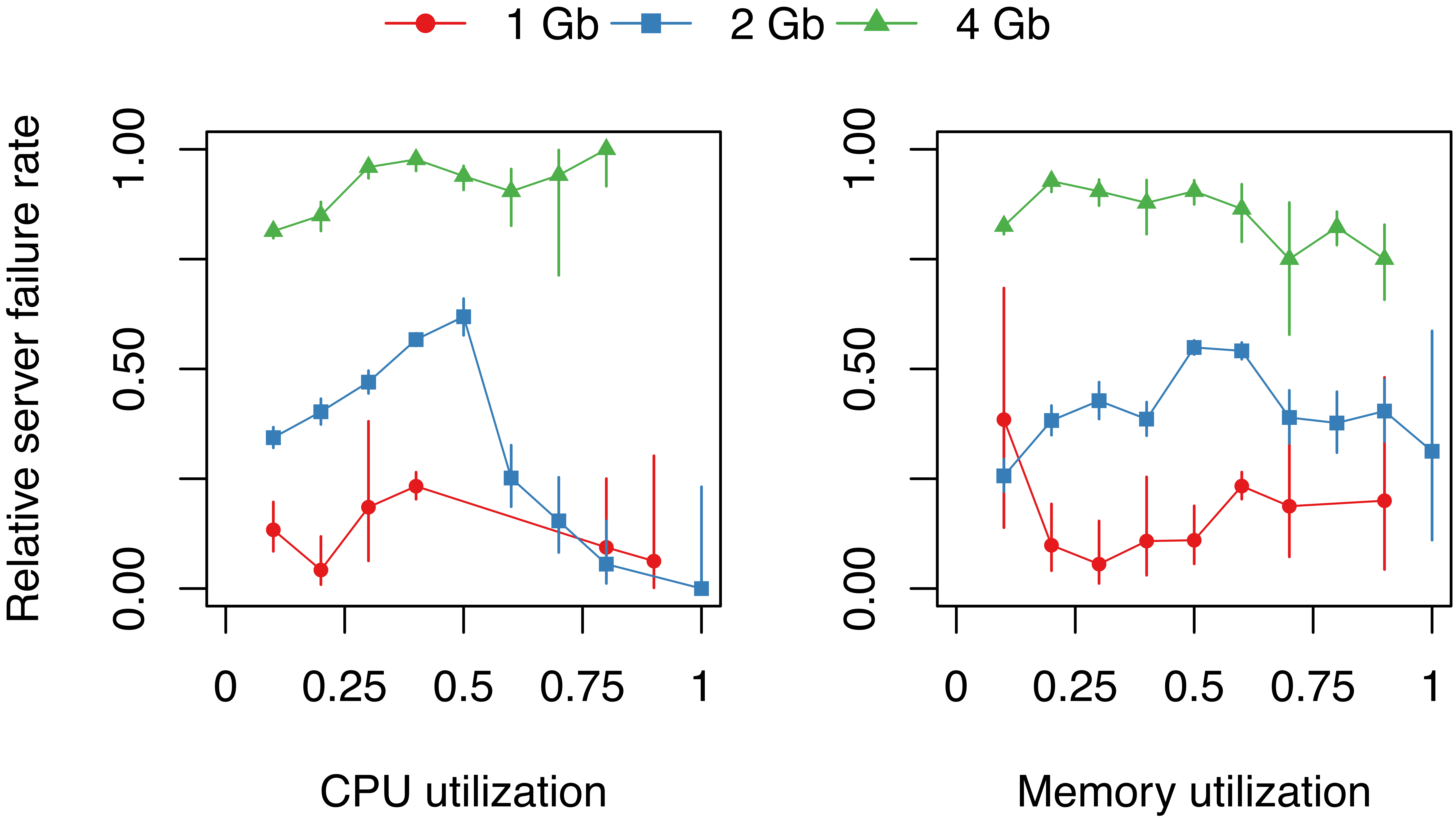
- Hardware (see DRAM and SSD failure details)
- Unplanned fiber cuts
  - Everything from anchors dragging to backhoes
- Bugs
  - Switches run a variety of software, can be buggy
- Operational mistakes
  - Attempting to repair a switch without turning it off

Exploratory analysis

Factor	Low-end	High-end (HE)	HE/↓density	HE/↓CPUs
Capacity	4 GB	16 GB	4 GB	16 GB
Density2Gb	1	0	1	0
Density4Gb	0	1	0	1
Chips	16	32	16	32
CPU%	50%	25%	25%	50%
Age	1	1	1	1
CPUs	8	16	16	8
Predicted relative failure rate	**0.12**	**0.78**	**0.33**	**0.51**

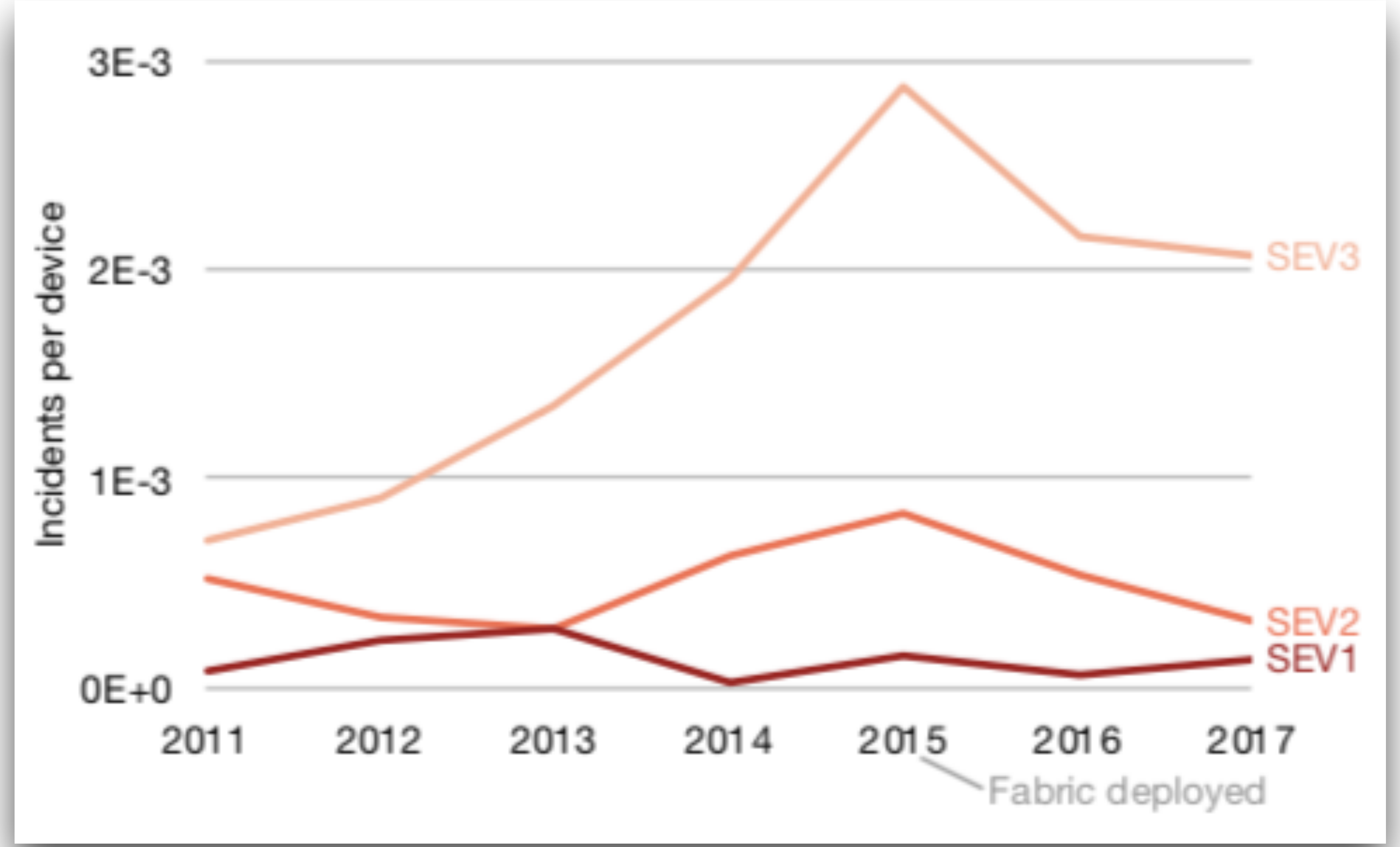
WRITE AMPLIFICATION



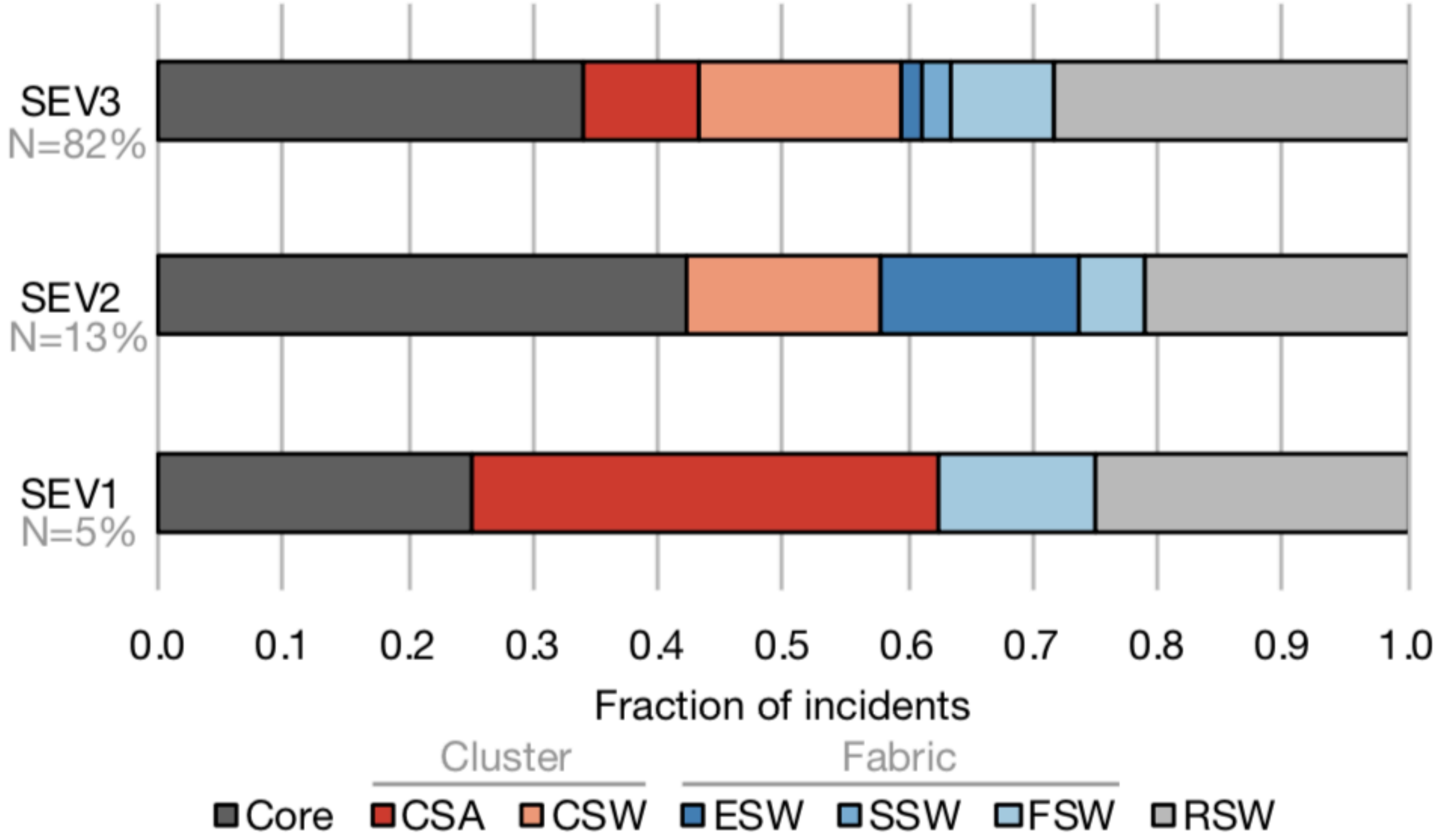


1 Gb
2 Gb
4 Gb
Relative server failure rate
0.00
0.50
1.00
0
0.25
0.5
0.75
1
CPU utilization
Memory utilization

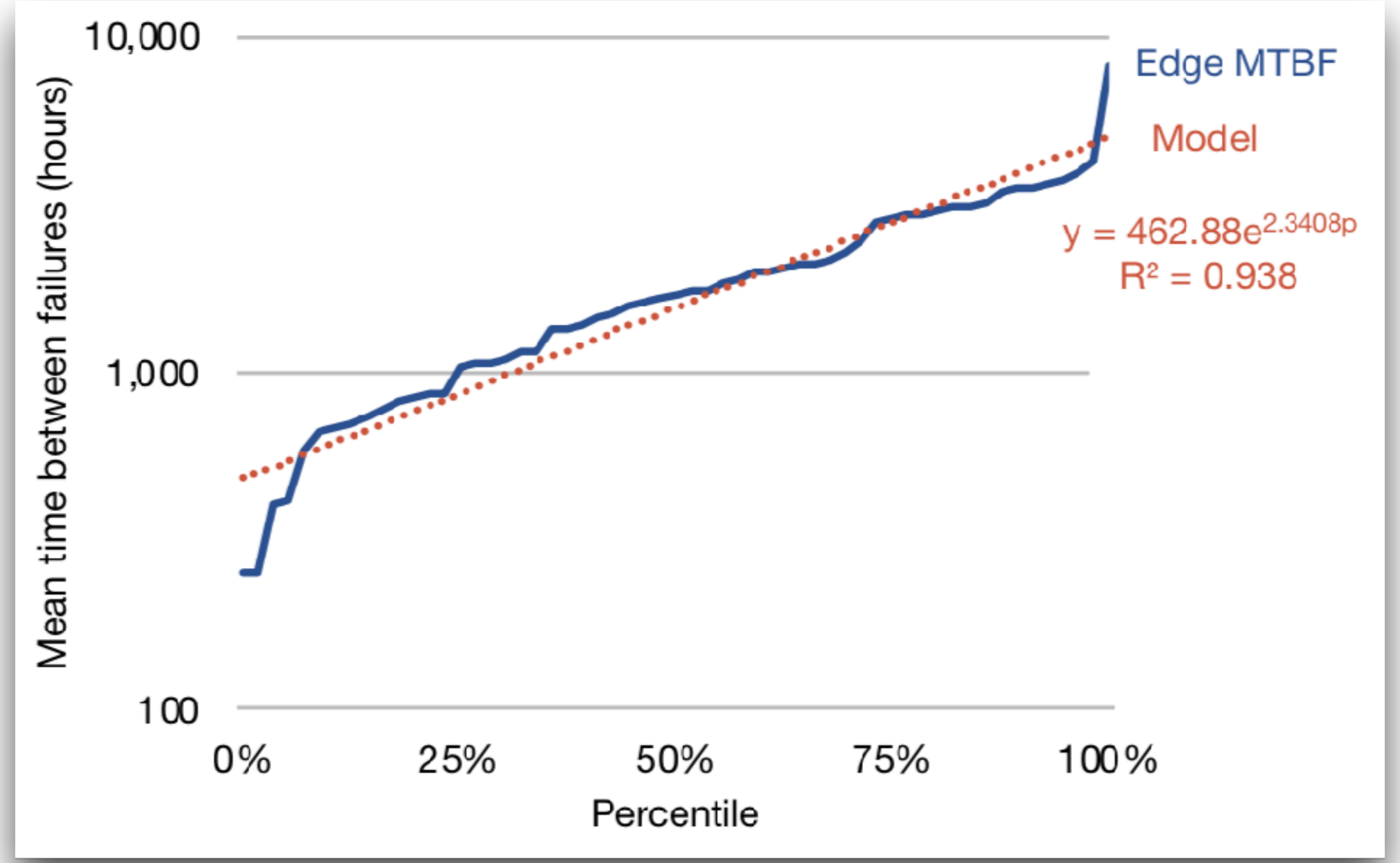
DC fabric has fewer incidents



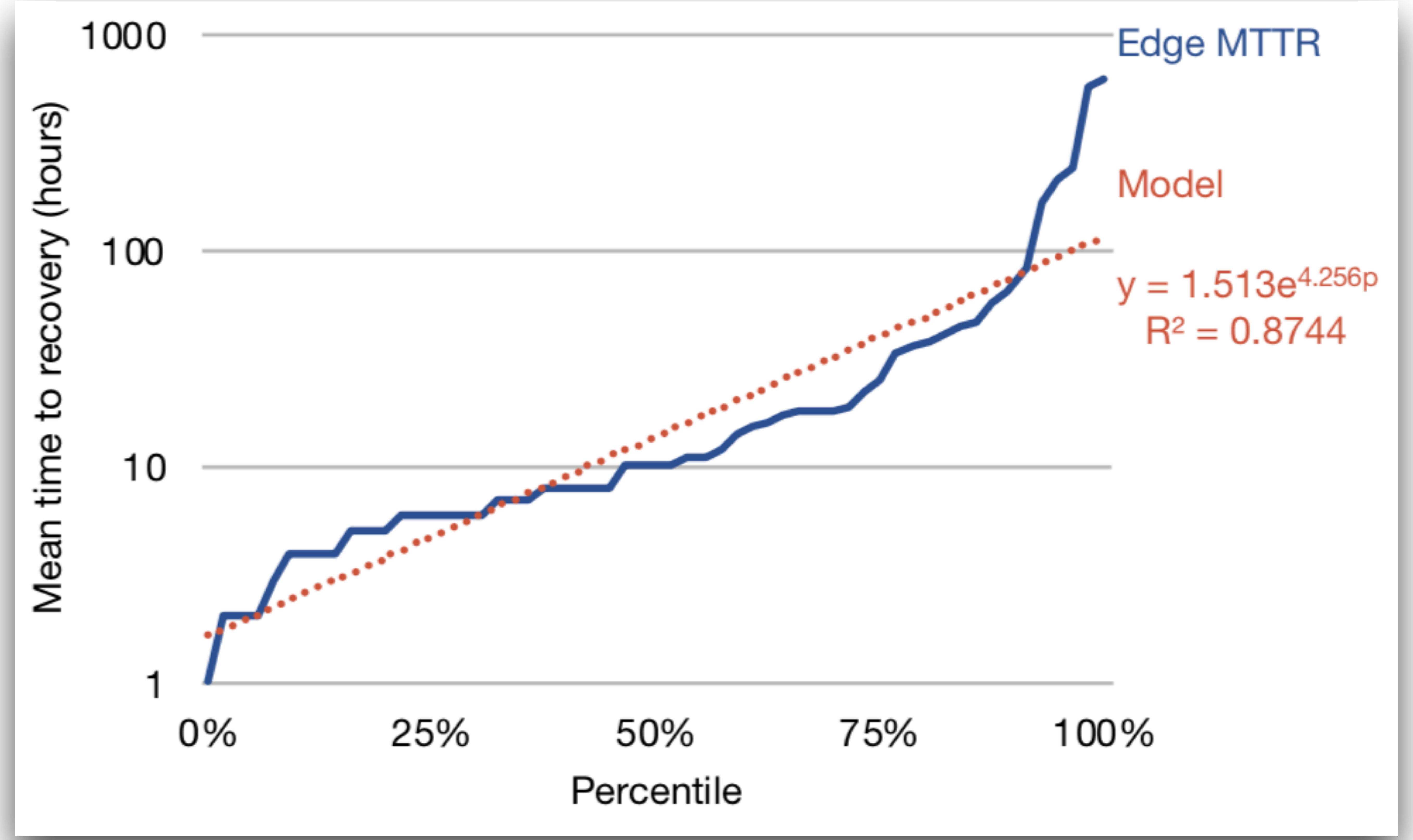
Main cause across all severities



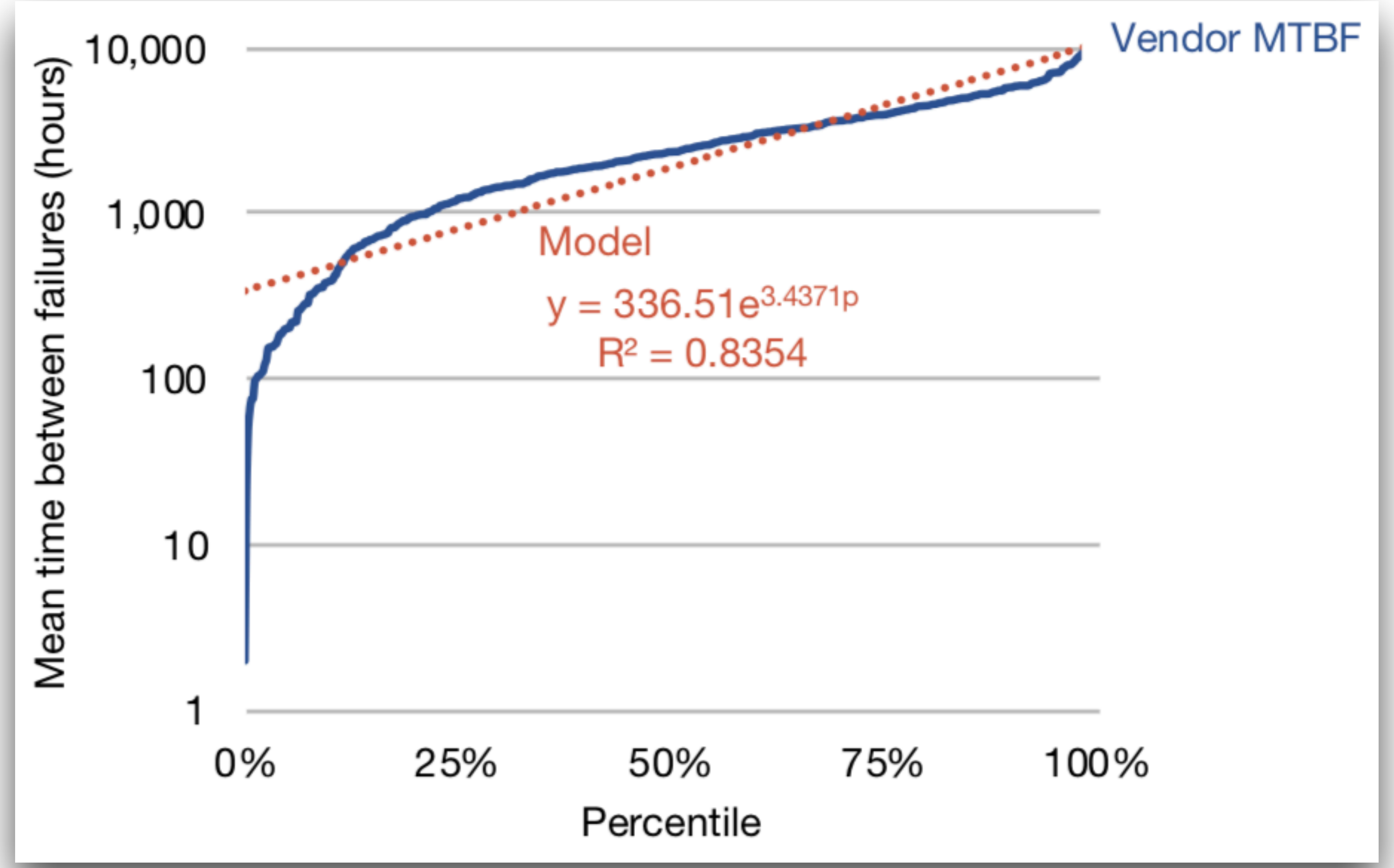
Edge node MTBF distribution



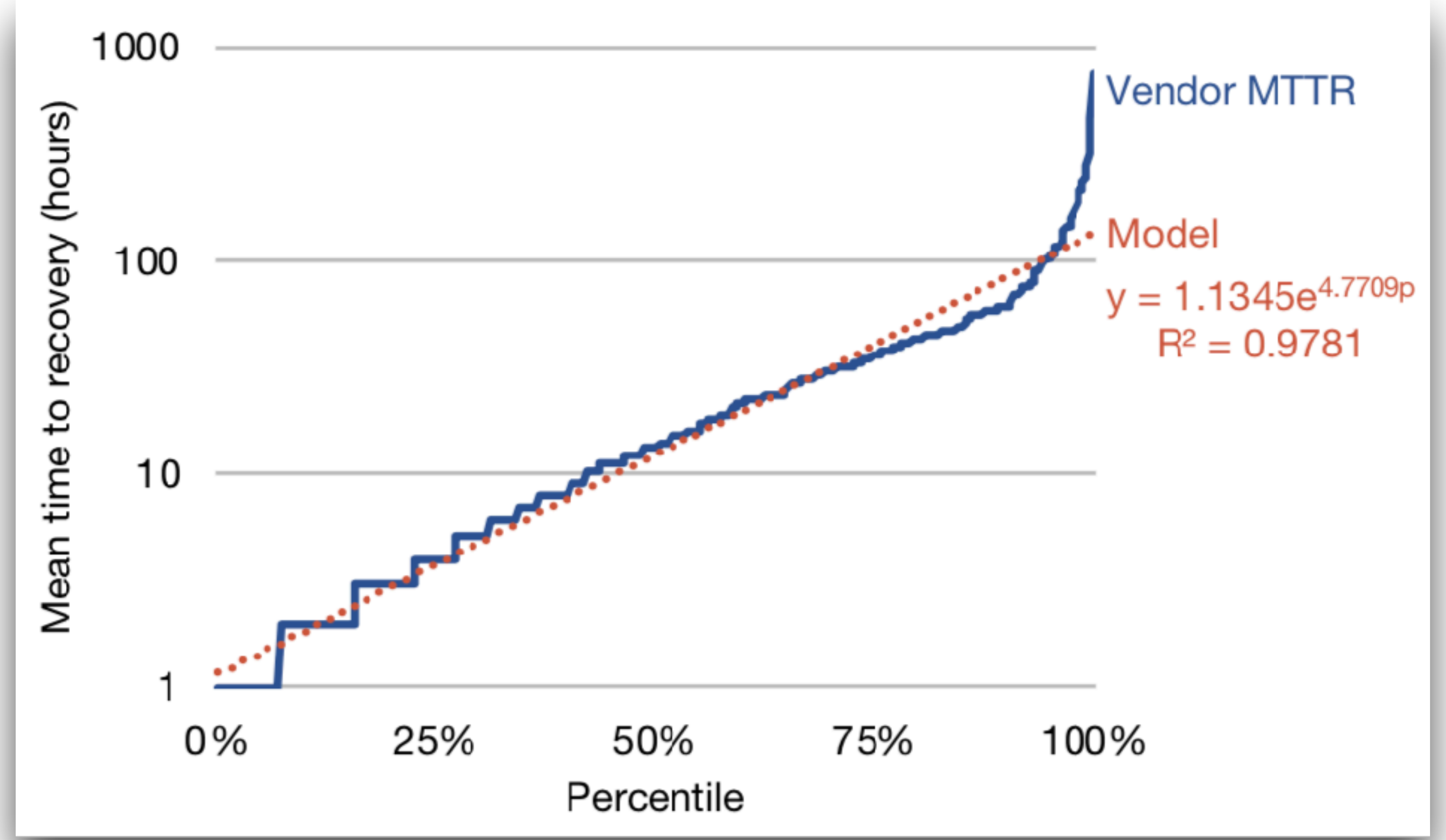
Edge node MTTR distribution



Fiber vendor MTBF distribution



Fiber vendor MTTR distribution



Minimizing backbone outages

$y = 462.88e^{2.3408p}$

$y = 1.513e^{4.256p}$

$y = 336.51e^{3.4371p}$

$\vdots$

Simulation

objective = six 9's

Capacity plan

Node1: Links A, B

Node 2: Links X, Y