Revisiting Memory Errors in Large-Scale Production Data Centers
Analysis and Modeling of New Trends from the Field

Justin Meza
Qiang Wu
Sanjeev Kumar
Onur Mutlu

facebook
Carnegie Mellon University
Overview

Study of DRAM reliability:

- on *modern* devices and workloads
- at a *large scale* in the field
Overview

Error/failure occurrence

Page offlining at scale

New reliability trends

Technology scaling

Modeling errors

Architecture & workload
Overview

Error/failure occurrence

Errors follow a **power-law distribution** and a large number of errors occur due to **sockets/channels**.
Overview

We find that **newer** cell fabrication technologies have **higher failure rates**.
Overview

Chips per DIMM, transfer width, and workload type (not necessarily CPU/memory utilization) affect reliability.
Overview

We have made publicly available a statistical model for assessing server memory reliability.
Overview

Page offlining at scale

Error/failure occurrence

First large-scale study of page offlining; real-world limitations of technique

Modeling errors

Architecture & workload
Outline

- background and motivation
- server memory organization
- error collection/analysis methodology
- memory reliability trends
- summary
Background and motivation
DRAM errors are common

- examined extensively in prior work
  - charged particles, wear-out
  - variable retention time (next talk)

- error correcting codes
  - used to detect and correct errors
  - require additional storage overheads
Our goal

**Strengthen understanding of DRAM reliability by studying:**

- new trends in DRAM errors
  - modern devices and workloads
- at a large scale
  - billions of device-days, across 14 months
Our main contributions

- identified new DRAM failure trends
- developed a model for DRAM errors
- evaluated page offlining at scale
Server memory organization
Socket
Memory channels
DIMM slots
Note:
1. Tolerance on all dimensions unless otherwise stated. Units: millimeters.
Note:

1. Tolerance on all dimensions unless otherwise stated.

Units: millimeters
**User data**

**ECC metadata**
additional 12.5% overhead
Reliability events

**Fault**
- the underlying cause of an error
  - DRAM cell unreliably stores charge

**Error**
- the manifestation of a fault
  - *permanent*: every time
  - *transient*: only some of the time
Error collection/analysis methodology
DRAM error measurement

- measured every correctable error
  - across Facebook's fleet
  - for 14 months
  - metadata associated with each error
- parallelized Map-Reduce to process
- used R for further analysis
System characteristics

- 6 different system configurations
  - Web, Hadoop, Ingest, Database, Cache, Media
  - diverse CPU/memory/storage requirements
- modern DRAM devices
  - DDR3 communication protocol
    - (more aggressive clock frequencies)
  - diverse organizations (banks, ranks, ...)
  - previously unexamined characteristics
    - density, # of chips, transfer width, workload
Memory reliability trends
New reliability trends

Error/failure occurrence
Page offlining at scale
Technology scaling
Modeling errors
Architecture & workload
Error/failure occurrence

- New reliability trends
- Technology scaling
- Architecture & workload
- Modeling errors
- Page offlining at scale
Server error rate

- Correctable errors (CE)
- Uncorrectable errors (UCE)

Fraction of servers

Month:
- 7/13
- 8/13
- 9/13
- 10/13
- 11/13
- 12/13
- 1/14
- 2/14
- 3/14
- 4/14
- 5/14
- 6/14
- 7/14
- 8/14

Error rates:
- 3%
- 0.03%
Memory error distribution

![Graph showing memory error distribution with a log-log scale. The graph compares measured data with a power law model. The x-axis represents the normalized device number, and the y-axis represents the number of logged errors. The graph shows a close match between the measured data and the power law model, indicating a power law distribution of memory errors.](image-url)
Memory error distribution

Decreasing hazard rate

[Graph showing the distribution of memory errors with a decreasing hazard rate.]

Number of logged errors

Normalized device number

Measured
Power law
How are errors mapped to memory organization?
### Sockets/channels: many errors

<table>
<thead>
<tr>
<th>Fraction of errors</th>
<th>Socket</th>
<th>Channel</th>
<th>Bank</th>
<th>Row</th>
<th>Column</th>
<th>Cell</th>
<th>Spurious</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

![Bar chart showing fraction of errors for Sockets, Channels, Banks, Rows, Columns, Cells, and Spurious errors.](attachment:chart.png)
Sockets/channels: many errors

Not mentioned in prior chip-level studies
Sockets/channels: many errors

Not accounted for in prior chip-level studies

At what rate do components fail on servers?
Bank/cell/spurious failures are common
Denial-of-service–like behavior
What factors contribute to memory failures at scale?
Analytical methodology

- measure server characteristics
  - not feasible to examine every server
  - 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
New reliability trends

- Error/failure occurrence
- Page offlining at scale
- Modeling errors
- Architecture & workload
- Technology scaling
Prior work found *inconclusive trends* with respect to memory *capacity*.

<table>
<thead>
<tr>
<th>DIMM capacity (GB)</th>
<th>Relative server failure rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.25</td>
</tr>
<tr>
<td>8</td>
<td>0.75</td>
</tr>
<tr>
<td>16</td>
<td>0.20</td>
</tr>
<tr>
<td>24</td>
<td>0.60</td>
</tr>
</tbody>
</table>
Prior work found inconclusive trends with respect to memory capacity.

Examine characteristic more closely related to cell fabrication technology.
Use **DRAM chip density** to examine technology scaling

(closely related to fabrication technology)
Intuition: quadratic increase in capacity
We find that newer cell fabrication technologies have higher failure rates.
New reliability trends

Error/failure occurrence

Page offlining at scale

Technology scaling

Modeling errors

Architecture & workload
DIMM architecture

- chips per DIMM, transfer width
  - 8 to 48 chips
  - x4, x8 = 4 or 8 bits per cycle
- electrical implications
DIMM architecture

- Does DIMM organization affect memory reliability?
- chips per DIMM, transfer width
  - 8 to 48 chips
  - x4, x8 = 4 or 8 bits per cycle
- electrical implications
We make two observations from Figure 9. We examine two aspects of DIMM design that have reliable organization and manufacturing. Prior work \[48, 10\] differences between vendors can arise if vendors use less reliable DIMM vendors.

In Figure 9, we observe two trends depending on whether chips on a DIMM have the same or different transfer widths. First, among devices with the same density, going from a low to high number of chips per DIMM increases server failure rate. For example, for 4 Gb devices, increasing the number of chips from 32 to 48 increases failure rate by 36.1%. Second, once the number of chips per DIMM reaches a certain threshold, increasing the number of chips decreases failure rate further. For 2 Gb devices, going from 8 chips with a transfer width of 8 bits per clock cycle (and are called \(\times 8\) devices) to 16 chips with a transfer width of 8 bits and 32 chips with a transfer width of 4 bits decreases failure rate by 13.2% and 7.1% respectively.

Figure 10 shows the relative failure rate of different vendor DIMMs for 1 Gb, 2 Gb, and 4 Gb devices. The relative cell failure rate of different vendor DIMMs is shown in Figure 11.

We have annotated the graph to show which chip densities correspond to the different chip transfer widths separated by different vendor DIMM vendors put into designing faster and more reliable DRAM cell architectures. Our insight is that increasing DRAM chip capacity will be untenable without decreasing DRAM server failure rates in the future (while still maintaining or increasing DRAM cell reliability as shown by the server failure rate data in Figure 6). Unless more-than–quadratic improvements in DRAM consumption of (e.g., I\(_\text{DD4R}\)\) power noise across the device. Such power noise could induce additional memory errors if, for example, charge were to get trapped in components. Interestingly, we find that, for a given chip density, increasing the number of chips per DIMM increases server failure rate. For example, for 4 Gb devices, increasing the number of chips from 32 to 48 increases failure rate by 36.1%. Second, once the number of chips per DIMM reaches a certain threshold, increasing the number of chips decreases failure rate further. For 2 Gb devices, going from 8 chips with a transfer width of 8 bits per clock cycle (and are called \(\times 8\) devices) to 16 chips with a transfer width of 8 bits and 32 chips with a transfer width of 4 bits decreases failure rate by 13.2% and 7.1% respectively.

Figure 10 shows the failure rate for servers with different chip densities. We find that, in addition to the first-order effect of chip density (e.g., I\(_\text{DD4R}\)\) chip density.

We observe two trends depending on whether chips on a DIMM have the same or different transfer widths. First, among devices with the same density, going from a low to high number of chips per DIMM increases server failure rate. For example, for 4 Gb devices, increasing the number of chips from 32 to 48 increases failure rate by 36.1%. Second, once the number of chips per DIMM reaches a certain threshold, increasing the number of chips decreases failure rate further. For 2 Gb devices, going from 8 chips with a transfer width of 8 bits per clock cycle (and are called \(\times 8\) devices) to 16 chips with a transfer width of 8 bits and 32 chips with a transfer width of 4 bits decreases failure rate by 13.2% and 7.1% respectively.

Figure 10 shows the relative failure rate of different vendor DIMMs for 1 Gb, 2 Gb, and 4 Gb devices. The relative failure rate of different vendor DIMMs is shown in Figure 11.

We have annotated the graph to show which chip densities correspond to the different chip transfer widths separated by different vendor DIMM vendors put into designing faster and more reliable DRAM cell architectures. Our insight is that increasing DRAM chip capacity will be untenable without decreasing DRAM server failure rates in the future (while still maintaining or increasing DRAM cell reliability as shown by the server failure rate data in Figure 6). Unless more-than–quadratic improvements in DRAM consumption of (e.g., I\(_\text{DD4R}\)\) power noise across the device. Such power noise could induce additional memory errors if, for example, charge were to get trapped in components. Interestingly, we find that, for a given chip density, increasing the number of chips per DIMM increases server failure rate. For example, for 4 Gb devices, increasing the number of chips from 32 to 48 increases failure rate by 36.1%. Second, once the number of chips per DIMM reaches a certain threshold, increasing the number of chips decreases failure rate further. For 2 Gb devices, going from 8 chips with a transfer width of 8 bits per clock cycle (and are called \(\times 8\) devices) to 16 chips with a transfer width of 8 bits and 32 chips with a transfer width of 4 bits decreases failure rate by 13.2% and 7.1% respectively.
We make two observations from Figure 9. We examine. The DIMMs that we examine have 8, 16, 32, and different numbers of data chips for each of the densities that transfer width of each chip. We also found a large range in the server failure rate among vendors. Reliable organization and manufacturing. Prior work [48, 10] differences between vendors can arise if vendors use less reliable organizations. We have made the vendors anonymous.

Relative server failure rate varies widely. We observe that failure rate varies by over

![Diagram showing relative server failure rate for different vendors.](image)

- **A:** Chip density (Gb)
- **B:** Relative cell failure rate
- **C:** Relative server failure rate
- **D:** Data chips per DIMM

**Figure 8:** We have annotated the graph to show which chip density counts have transfer widths of 4 bits. DD4R devices (Effect 2). We find that server failure rate trends with respect to getting trapped in components. Interestingly, we find that, for a given chip density, servers with smaller transfer widths of 4 bits per clock cycle (and are called DDR, for example, for 4 Gb devices, increasing the number of chips from 8 to 16 increases failure rate by 40.8% while for 2 Gb devices, going from 8 chips with a width of 8 to 48 increases failure rate by 13.2%.

**Figure 9:** Relative server failure rate for different vendors varies widely. We examine how DIMM architecture affects server reliability and manufacturing. Prior work [48, 10] differences between vendors can arise if vendors use less reliable organizations. We have made the vendors anonymous.

![Diagram showing relative server failure rate for different vendors.](image)

- **A:** Chip density (Gb)
- **B:** Relative cell failure rate
- **C:** Relative server failure rate
- **D:** Data chips per DIMM

**Figure 6:** Unless more-than–quadratic improvements in DRAM architectures. Our insight is that small improvements in DRAM are achieved in future devices, maintaining or increasing DRAM server failure rates in the future (while still decreasing DRAM chip capacity) will be untenable without increasing DRAM cell reliability are achieved in future devices, maintaining or increasing DRAM server failure rates in the future (while still decreasing DRAM chip capacity).
We make two observations from Figure 9.

- The DIMMs that we examine have 8, 16, 32, and different numbers of data chips for each of the densities that have not been studied in published literature before: the number of data chips (not including chips for ECC) per DIMM and the failure rate. We examine two aspects of DIMM design that have been studied in published literature before: the number of chips per DIMM vendor and the failure rate.

- We observe that failure rate varies by over 0.50 relative cell failure rate at different technology nodes (chip densities). Relative failure rate of 1 Gb devices is shown by the black line, 2 Gb devices by the blue line, and 4 Gb devices by the green line. The relative failure rate varies widely between vendors (e.g., Vendor B and Vendor C). The difference in relative failure rate for different vendors varies widely.

Relative cell failure rate are achieved in future devices, maintaining or decreasing DRAM server failure rates in the future (while still being able to take advantage of increasing DRAM chip capacity) will be untenable without stronger hardware and/or software error correction.

Figure 7 shows the failure rate for servers with different DIMM vendor (anonymized). We next examine how DIMM architecture affects server failure rate by 36.1%. Second, once the number of chips per DIMM increases, failure rate by 40.8% while for 2 Gb devices, going from 8 chips with a width of 8 bits per clock cycle to 32 chips with a width of 8 bits per clock cycle decreases failure rate by 13.2%.

Prior work [48, 10] has found that server failure rate trends with respect to the electrical disturbance within a DIMM that may disrupt the electrical noise on the bus. Such power noise could induce additional memory errors if, for example, charge were to get trapped in components. Interestingly, we find that, for a smaller transfer width of 4 bits per clock cycle (and are called DDR3), leading to the trend of net decrease in DRAM chip consumption of Isolef, the number of chips per DIMM increases server failure rate. For 1 Gb devices, going from 16 chips with a transfer width of 8 bits per clock cycle (and are called DDR4) to 48 chips per DIMM has transfer widths of 8 bits per clock cycle (and are called DDR4) to 32 chips with a transfer width of 8 bits per clock cycle (and are called DDR4) decreases failure rate by 7.1%.

We observe two trends depending on whether chips on a particular DIMM vendor (anonymized) induce additional memory errors if, for example, charge were to get trapped in components. Interestingly, we find that, for a smaller transfer width of 4 bits per clock cycle (and are called DDR3), leading to the trend of net decrease in DRAM chip consumption of Isolef, the number of chips per DIMM increases server failure rate. For 1 Gb devices, going from 16 chips with a transfer width of 8 bits per clock cycle (and are called DDR4) to 48 chips per DIMM has transfer widths of 8 bits per clock cycle (and are called DDR4) decreases failure rate by 7.1%.

For 2 Gb devices, going from 8 chips with a transfer width of 8 bits per clock cycle (and are called DDR4) to 32 chips with a transfer width of 4 bits per clock cycle (and are called DDR3) decreases failure rate by 7.1%.
We make two observations from Figure 9. Relative server failure rate for different vendors varies widely. We observe that failure rate varies by over 0.50. Differences between vendors can arise if vendors use less reliable DIMM vendors. We believe that both effects may be partially explained by the large amounts of effort that DRAM manufacturers invest in quality control over the decades since DRAM was invented.

Figure 9 plots the failure rate for servers with DIMMs with different numbers of data chips for each of the densities that transfer width of each chip. The relative failure rate of DIMM vendor (anonymized) is shown by the 2 Gb devices with 32 chips with a smaller transfer width of 4 bits per clock cycle (and are called x4 interface) compared to the other 2 Gb devices with a larger transfer width of 8 bits per clock cycle (and are called x8 interface). This is shown by the 2 Gb devices with 32 chips of the same transfer width, increasing failure rate (Effect 1), there is a consistent increase in cell reliability are achieved in future devices, maintaining or in DRAM reliability as shown by the server failure rate data in Figure 6.

We find that, in addition to the first-order effect of chip density (e.g., I

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor A}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor B}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor C}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor D}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor E}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor F}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor G}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor H}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor I}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor J}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor K}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor L}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor M}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor N}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor O}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor P}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor Q}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor R}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor S}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor T}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor U}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor V}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor W}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor X}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor Y}}{\text{Average failure rate of all vendors}} \]

\[ \text{Relative server failure rate} = \frac{\text{Failure rate of DIMM vendor Z}}{\text{Average failure rate of all vendors}} \]
We make two observations from Figure 9. We examine. The DIMMs that we examine have 8, 16, 32, and different numbers of data chips for each of the densities that transfer width of each chip.

C. DIMM Architecture

DIMM Architecture

DIMM Vendor

DIMM architecture affects server reliability organization and manufacturing. Prior work [48, 10] differences between vendors can arise if vendors use less reliable information on DIMM manufacturer, we do not have information manufacturers and assemble them into DIMMs. While we have information on DIMM vendor, we believe that both effects may be partially explained by deterministic power noise across the device. Such power noise could consumption of (e.g., I

\[ I \times V \]

a larger transfer width increases internal data transfer current to the electrical disturbance within a DIMM that may disrupt the integrity of the signal between components. For example, inducing additional memory errors if, for example, charge were to get trapped in components. Interestingly, we find that, for a given number of chips per DIMM, servers with the same or different transfer widths. First, among servers with the best architecture for device reliability. This illustrates how chip density is a first-order effect when considering memory failure rate (as we showed in Figure 6).

We find that, in addition to the first-order effect of chip density, the failure rates dependent on transfer width alone in Figure 10. For 2 Gb devices, going from 8 chips with a smaller transfer width of 4 bits per clock cycle (and are called x4) compared to the other 2 Gb devices x8, there is a decrease in failure rate by 13.2%. Second, once the number of chips per DIMM increases beyond 8, increasing the number of chips from 32 to 48 increases failure rate by 36.1%. Second, once the number of chips per DIMM increases server failure rate. For 1 Gb devices, going from 16 chips with a transfer width of 4 bits to 32 chips with a transfer width of 8 bits increases beyond 16 and chips start using a smaller transfer width of 4 bits per clock cycle (and are called x4) compared to the other 2 Gb devices x8.

We observe that failure rate varies by over 3.0e-13, leading to the trend of net decrease in DRAM server failure rate in the future (while still decreasing DRAM cell capacity) will be untenable without increasing DRAM chip capacity). As DRAM cell reliability are easily outpaced by the quadratic increase in number of cells per chip, given chip density, increasing failure rate (Effect 1), there is a consistent increase in DRAM chip failure rate by 0.00 to 0.50.

Fig. 7:

Fig. 8:

Fig. 9:

Fig. 10:
Figure 9 plots the failure rate for servers with DIMMs with different numbers of data chips for each of the densities that transfer width of each chip.

The relative per-cell failure rate at different technology nodes (chip cell reliability) is easily outpaced by the quadratic increase in additional power noise across the device. Such power noise could contribute to the electrical disturbance within a DIMM that may disrupt the integrity of the signal between components. For example, for 4 Gb devices, increasing the number of chips from 8 to 16 increases failure rate by 40.8% while for 2 Gb devices, the number of chips per DIMM increases server failure rate. For 1 Gb devices, going from 16 chips with a smaller transfer width of 4 bits per clock cycle (and are called DD4R) to chips per DIMM are dependent on the chip density.

DIMM vendors purchase chips from DRAM chip manufacturers in our systems. While we have information on DIMM manufacturer, we do not have information on the DRAM chip manufacturers in our systems. To confirm the trend related to transfer width, we plotted the relative server failure rate by 36.1%. Second, once the number of chips per DIMM increases, there is a consistent increase in relative server failure rate. For example, for 4 Gb devices, increasing the number of chips from 32 to 48 increases failure rate by 36.1%.

First, for a given number of chips per DIMM, servers with a larger transfer width increase internal data transfer current and result in, first, low transfer width and, second, low chip densities). We find that, in addition to the first-order effect of chip density (e.g., Vendor B and Vendor C). The differences between vendors can arise if vendors use less reliable organizations and manufacturing. Prior work [48, 10] also found a large range in the server failure rate among vendors. Reliable cell organization and cell manufacturing put into designing faster and more reliable DRAM cell architectures. Our insight is that stronger hardware and/or software error correction is needed to increase DRAM server failure rates in the future (while still decreasing DRAM cell failure rate by 7.1% to the electrical disturbance within a DIMM that may disrupt the integrity of the signal between components. For example, for 4 Gb devices, increasing the number of chips from 8 to 16 increases failure rate by 40.8% while for 2 Gb devices, going from 8 chips with a smaller transfer width of 4 bits per clock cycle (and are called DD4R) to chips per DIMM are dependent on the chip density.

DIMM Architecture

We examine two aspects of DIMM design that have different numbers of data chips (not including chips for ECC) per DIMM and the failure rate. We examine two aspects of DIMM design that have

C. DIMM Architecture

We next examine how DIMM architecture affects server reliability and manufacturing. Prior work [48, 10] also found a large range in the server failure rate among vendors. Reliable cell organization and cell manufacturing put into designing faster and more reliable DRAM cell architectures. Our insight is that stronger hardware and/or software error correction is needed to increase DRAM server failure rates in the future (while still decreasing DRAM cell failure rate by 7.1% to the electrical disturbance within a DIMM that may disrupt the integrity of the signal between components. For example, for 4 Gb devices, increasing the number of chips from 8 to 16 increases failure rate by 40.8% while for 2 Gb devices, going from 8 chips with a smaller transfer width of 4 bits per clock cycle (and are called DD4R) to chips per DIMM are dependent on the chip density.

DIMM Vendor

We observe that failure rate varies by over 0.00 0.50 1.00 3.0e−13 3.0e−13. This is shown by the 2 Gb devices with 32 chips per DIMM. The relative failure rate of different 1 Gb devices, going from 16 chips with a smaller transfer width of 4 bits per clock cycle (and are called DD4R) to chips per DIMM are dependent on the chip density.

# Figure 8:

**Relative server failure rate**

- **1 Gb**
- **2 Gb**
- **4 Gb**

**Reduction of failure rate**

- For 2 Gb devices, going from 8 chips with a smaller transfer width of 4 bits per clock cycle (and are called DD4R) to chips per DIMM are dependent on the chip density.

**Effect 1:** In Figure 10, there is a decrease in failure rate. For example, I

**Effect 2:** Increase in failure rate

- For 2 Gb devices, going from 8 chips with a smaller transfer width of 4 bits per clock cycle (and are called DD4R) to chips per DIMM are dependent on the chip density.

**Conclusion:** The relative per-cell failure rate at different technology nodes (chip cell reliability) is easily outpaced by the quadratic increase in additional power noise across the device. Such power noise could contribute to the electrical disturbance within a DIMM that may disrupt the integrity of the signal between components. For example, for 4 Gb devices, increasing the number of chips from 32 to 48 increases failure rate by 36.1%.

**Note:** The diagram shows the relative server failure rate for different numbers of data chips per DIMM and different chip densities. The x-axis represents the number of data chips per DIMM, and the y-axis represents the relative server failure rate.
More chips $\rightarrow$ higher failure rate
We make two observations from Figure 9:

- The relative server failure rate varies widely among vendors (e.g., Vendor B and Vendor C). The differences between vendors can arise if vendors use less reliable chips from a particular foundry or build DIMMs with less reliable organization and manufacturing. Prior work [48, 10] has found a large range in the server failure rate among vendors.

- Increasing DRAM chip capacity will be untenable without small improvements in DRAM cell reliability. Cell failure rates are achieved in future devices, maintaining or increasing DRAM reliability as shown by the server failure rate data in Figure 6. Unless more-than–quadratic improvements in DRAM consumption of (e.g., I_DD) lead to the electrical disturbance within a DIMM that may disrupt the integrity of the signal between components. For example, considering how numbers of chips and transfer width contribute to the electrical noise across the device. Such power noise could induce additional memory errors if, for example, charge were lost at the electrical noise in the system. For example, for 4 Gb devices, increasing the number of chips from 8 to 16 increases failure rate by 40.8% while for 2 Gb devices, increasing the number of chips from 8 to 16 increases failure rate by 36.1%. Second, once the number of chips per DIMM increases beyond 16 and chips start using a transfer width of 8 bits per clock cycle (and are called DD4R), there is a consistent increase in failure rates dependent on transfer width alone in Figure 10. For 2 Gb devices, going from 8 chips with a transfer width of 4 bits to 32 chips with a transfer width of 8 bits increases failure rate by 7.1%.

### Figure 9: Relative Server Failure Rate

The graph shows the relative server failure rate for different vendors. The x-axis represents the data chips per DIMM, and the y-axis represents the relative server failure rate. The legend indicates the DRAM devices: 1 Gb (red), 2 Gb (blue), and 4 Gb (green). The graph illustrates how chip density is a first-order effect when considering memory failure rate (as we showed in Figure 6). Higher chip densities generally have higher average failure rates.
We make two observations from Figure 9. We examine the DIMMs that we examine have 8, 16, 32, and different numbers of data chips for each of the densities that transfer width of each chip. We examine two aspects of DIMM design that have also found a large range in the server failure rate among vendors. Differences between vendors can arise if vendors use less reliable DIMM vendors. We believe that both effects may be partially explained by small improvements in DRAM cell reliability are achieved in future devices, maintaining or decreasing DRAM server failure rates in the future (while still cell reliability are easily outpaced by the quadratic increase in number of cells per chip). We find that, in addition to the first-order effect of chip density, the failure rates dependent on transfer width alone in Figure 10.

We observe two trends depending on whether chips on a DIMM have the same or different transfer widths. First, among DIMMs with the same transfer width, we find that increasing the number of chips per DIMM increases server failure rate. For example, for 4 Gb devices, increasing the number of chips from 32 to 48 increases failure rate by 36.1%. Second, once the number of chips per DIMM increases beyond 16 and chips start using a smaller transfer width of 4 bits per clock cycle (and are called DD4R = 4-dimensional DRAM), the integrity of the signal between components is more likely to get trapped in components. Interestingly, we find that, for a given number of chips per DIMM, servers with chips with a smaller transfer width of 4 bits per clock cycle (and are called DD4R = 4-dimensional DRAM) have the same or different transfer widths. First, among DIMMs with the same number of chips per DIMM are dependent on the chip density. This illustrates how chip density is a first-order effect when higher chip densities generally have higher average failure rates.

We have made the vendors anonymous. We observe that failure rate varies by over 0.001.00 Relative server failure rate 0.0e+00 1.5e−13 3.0e−13 1.00 Relative cell failure rate 1.0. The relative failure rate of the DD4R = 4-dimensional DRAM interface compared to the other 2 Gb devices decreases failure rate by 13.2%.

In order to transfer data at a similar rate, DIMMs counts have transfer widths of 8 and 4 Gb devices) while DIMMs with more chips (32 or 48) can use a transfer width of 8 bits per clock cycle (and are called DD4R = 4-dimensional DRAM). We find that, in addition to the first-order effect of chip density, the failure rates dependent on transfer width alone in Figure 10.

8 16 32 48 Data chips per DIMM

More **bits per cycle** → higher failure rate
Intuition: increased electrical loading
Workload dependence

- prior studies: homogeneous workloads
  - web search and scientific

- warehouse-scale data centers:
  - web, hadoop, ingest, database, cache, media
Workload dependence

- prior studies: homogeneous workloads
- web search and scientific warehouse-scale data centers: web, hadoop, ingest, database, cache, media

What affect to heterogeneous workloads have on reliability?
No consistent trend across CPU/memory utilization.
Relative server failure rate

- Web: 0.00
- Hadoop: 0.50
- Ingest: 1.00
- Database: Varies by up to 6.5x
- Memcache: 0.00
- Media: Varies by up to 6.5x
Chips per DIMM, transfer width, and workload type (not necessarily CPU/memory utilization) affect reliability.
New reliability trends

Error/failure occurrence

Page offlining at scale

Technology scaling

Modeling errors

Architecture & workload
A model for server failure

- use *statistical regression model*
  - compare *control group vs. error group*
  - *linear regression* in R
  - trained using data from analysis

- enable *exploratory analysis*
  - high perf. vs. low power systems
Memory error model

Density
Chips
Age
...

Relative server failure rate
Relative server failure rate

\[ \ln \left( \frac{\hat{F}}{1 - \hat{F}} \right) = \beta_{\text{intercept}} + (\text{Capacity} \cdot \beta_{\text{Capacity}}) + (\text{Density2Gb} \cdot \beta_{\text{Density2Gb}}) + (\text{Density4Gb} \cdot \beta_{\text{Density4Gb}}) + (\text{Chips} \cdot \beta_{\text{Chips}}) \\
+ (\text{CPU} \% \cdot \beta_{\text{CPU} \%}) + (\text{Age} \cdot \beta_{\text{Age}}) + (\text{CPUs} \cdot \beta_{\text{CPUs}}) \]

\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
\textbf{Characteristic} & \textbf{Density2Gb} & \textbf{Density4Gb} & \textbf{Chips} & \textbf{CPU\%} & \textbf{Age} & \textbf{CPUs} \\
\hline
\textbf{Significant?} & Yes & Yes & Yes & Yes & Yes & Yes \\
\hline
\textbf{p-value} & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \\
\textbf{Standard Error} & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \\
\hline
\end{tabular}
Available online

http://www.ece.cmu.edu/~safari/tools/memerr/
We have made publicly available a statistical model for assessing server memory reliability.
New reliability trends

Error/failure occurrence

Page offlining at scale

Technology scaling

Modeling errors

Architecture & workload
Prior page offlining work

- [Tang+, DSN'06] proposed technique
  - "retire" faulty pages using OS
  - do not allow software to allocate them

- [Hwang+, ASPLOS'12] simulated eval.
  - error traces from Google and IBM
  - recommended retirement on first error
    - large number of cell/spurious errors
Prior page off lining work

- [Tang+, DSN'06] proposed technique
  - "retire" faulty pages using OS
- [Hwang+, ASPLOS'12] simulated eval.
  - error traces from Google and IBM
- recommended retirement on first error
  - large number of cell/spurious errors

How effective is page offlining in the wild?
flining – when adapted to function at scale – provides reason-

...at a later time, at the expense of added complexity to system

This, however, could be easily fixed by retrying page-offlining

...example, if its data is being prefetched into the page cache at

...is if its contents cannot be locked for exclusive access. For

...found of why a page may fail to be offlined in the Linux kernel

...the attempts to offline a page initially failed. One example we

...may not always succeed in real systems. We additionally logged

...certain fraction of its pages have been taken offline. Second, it

...memory capacity, which requires repairing a machine after a

...tations that were not addressed in prior work. First, it reduces

...number of reported errors, we find that it has two main limi-

C. Limitations

...not seem to be affected by aggressive page offlining.

...and other techniques that help reduce the error rate that does

...be avoided due to page offlining. Third, we observe a relatively

...is smaller than the 86% to 94% error rate reduction reported

...couple of weeks), the error rate decreased by around 67%. This

...deploying page offlining to 100% of the fleet at day 25,

...offlining to the error rate after day 8.

...the effect of page offlining on error rate. By day 8 its effects were no longer noticeable in

...shows the state of the servers while page offlining was deployed

...different phases of our experiment. Region

...days. We divide the graph into three regions based on the

...Despite these limitations, however, we find that page of-

...memory systems that have not been discussed before in lit-

...We analyzed a variety of factors and how they affect server

...monitor memory for errors to reduce application impact.

...They found that most

...analyzed memory errors on 212 Ask.com servers and evaluated

...increased DRAM error rates [40]. A pair of works by Li et al.

...and showed that increased CPU frequency is correlated with

...controller and memory bus, but did not classify memory errors

...identified occurrences of multi-DIMM errors, and speculated as

...found that most memory errors are permanent and additionally

...purely in simulation, not in a large scale system.

...page offlining, unlike ours (presented in Section VI), was done

...to reduce error rate by 86% to 94%. Note that their study of

...proposed in [49]) on the memory error traces, which they found

...socket and channel failures. The high number of repeat address

...errors from a sample of Google servers and IBM supercomput-

...Section III. Their work formed the basis for what is known of

...by our study in this paper, five years later, as we explained in

...average server error rate is very high – findings clarified

...errors are dominated by device failures (versus alpha particles),

...errors across Google's server population, provided evidence that

...them [44, 16, 47, 48, 10] throughout the paper. We will discuss

...ready presented extensive comparisons to the most prevalent of

...aspects of memory errors in different systems. We have al-

...examining the memory failure rate of systems, and performed

...been identified in prior work (e.g., chip density, transfer width,
Cluster of 12,276 servers

Initial testing → Fully deployed

-67%

Normalized logged errors

Day of study

Normalized logged errors

0.0 0.2 0.4 0.6 0.8 1.0

0 10 20 30 40 50 57

VIII. Conclusions

We performed a comprehensive analysis of the memory failures that occurred on Google's servers and observed a significant decrease in error rates after deploying page offlining. The error rate decreased by around 67% after fully deploying page offlining to the servers.

While page offlining is relatively effective at reducing the number of errors, it is still around 18% of the maximum amount, even after page offlining was deployed to 100% of the fleet at day 25.

We observed that the effects of page offlining were no longer noticeable in their simulation after day 8.

Despite these limitations, we find that page offlining is still a valuable technique for reducing memory errors on Google's servers.
There are three things to note from Figure 18. First, after the moving average and we compare the effectiveness of page offlining to the error rate after day 8.

Prior work: -86% to -94%

Cluster of 12,276 servers

We look forward to future works that analyze the interaction of "able memory error tolerance benefits, as we have demonstrated."
6% of page offlining attempts failed due to OS

Prior work: -86% to -94%
Page offlining at scale

First large-scale study of page offlining; real-world limitations of technique

Error/failure occurrence

Modeling errors

Architecture & workload

Technology scaling

Architecture & workload
Error/failure occurrence

Page offlining at scale

New reliability trends

Modeling errors

Technology scaling

Architecture & workload
More results in paper

- Vendors
- Age
- Processor cores
- Correlation analysis
- Memory model case study
Summary
- Modern systems
- Large scale
Summary

Error/failure occurrence

Page offlining at scale

New reliability trends

Modeling errors

Technology scaling

Architecture & workload
Summary

Error/failure occurrence

Errors follow a *power-law distribution* and a large number of errors occur due to *sockets/channels*
We find that newer cell fabrication technologies have higher failure rates.
Chips per DIMM, transfer width, and workload type (not necessarily CPU/memory utilization) affect reliability.
Summary

We have made publicly available a **statistical model** for assessing server memory reliability.
Summary

Page offlining at scale

First large-scale study of page offlining; real-world limitations of technique

Error/failure occurrence

Modeling errors

Architecture & workload
Revisiting Memory Errors in Large-Scale Production Data Centers
Analysis and Modeling of New Trends from the Field

Justin Meza
Qiang Wu
Sanjeev Kumar
Onur Mutlu

facebook
Carnegie Mellon University
Backup slides
Decreasing hazard rate

\[ \Pr(\text{logged errors} > x) \]

- Red: Measured
- Blue: Pareto ($R^2 = 0.97$)

Number of logged errors:

\[ 10^0, 10^2, 10^4, 10^6, 10^8 \]
<table>
<thead>
<tr>
<th>Errors</th>
<th>54,326</th>
<th>0</th>
<th>2</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>4Gb</td>
<td>1Gb</td>
<td>2Gb</td>
<td>2Gb</td>
</tr>
</tbody>
</table>
### Errors vs. Density

<table>
<thead>
<tr>
<th>Errors</th>
<th>54,326</th>
<th>0</th>
<th>2</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>4Gb</td>
<td>1Gb</td>
<td>2Gb</td>
<td>2Gb</td>
</tr>
</tbody>
</table>

- **Density**: Units: millimeters
  - W X W max 1.27 mm
  - 0.4 max units
- **Errors**: 0.3~0.1 tolerance on all

Dimensions unless otherwise stated.
<table>
<thead>
<tr>
<th>Errors</th>
<th>54,326</th>
<th>0</th>
<th>2</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>4Gb</td>
<td>1Gb</td>
<td>2Gb</td>
<td>2Gb</td>
</tr>
</tbody>
</table>

![Diagram of Buckets](image.png)
Errors 54,326
Density 4Gb

Errors

Density

1Gb
2Gb
2Gb
Case study
Case study

<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>
<tr>
<td>Predicted relative failure rate</td>
<td>0.12</td>
<td>0.78</td>
</tr>
</tbody>
</table>

TABLE III: failure rate comparison for different system configurations.
Case study

Does CPUs or density have a higher impact?

<table>
<thead>
<tr>
<th>Factor</th>
<th>Low-end</th>
<th>High-end (HE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CPUs</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Predicted relative failure rate</td>
<td>0.12</td>
<td>0.78</td>
</tr>
</tbody>
</table>
Exploratory analysis

<table>
<thead>
<tr>
<th>Factor</th>
<th>Low-end</th>
<th>High-end (HE)</th>
<th>HE\downarrow density</th>
<th>HE\downarrow 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>Density2Gb</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Density4Gb</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 next discuss the results of a study performed to examine the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

For instance, increasing failure rates with increasing chip density and number of CPUs, (2) a trend that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.

Using the equation in Table II, we can solve for the predicted failure rate of the high-end server. This agrees with the model-predicted failure rate of the high-end server, which is lower than that of the low-end server. This agrees with the trends that we observed in Section IV, which showed, for example, increasing failure rates with increasing chip density and number of CPUs. Interestingly, the model can be used to provide insight into the relative change in error rate for different configurations.