

Analysis and Modeling of Memory Errors from Large-scale Field Data Collection

Taniya Siddiqua*, Athanasios E. Papathanasiou[‡], Arijit Biswas*, Sudhanva Gurumurthi[§]

*Intel Corp.

[‡]Teradata Aster

[§]Dept. of Computer Science, University of Virginia

Abstract— Main memory reliability plays a crucial role in overall system reliability. Unfortunately, our collective understanding of the rate, pattern, and impact of memory errors is inadequate and can hinder our ability to innovate new fault-tolerant designs. This paper presents an in-depth study of observed corrected error data from the main memory system of a large server population deployed in data centers. Our analysis includes multiple structures on the memory path, such as the memory controllers, busses, channels, and memory modules. Based on our observations, we present a taxonomy of potential faults in the memory path. We provide a detailed characterization of the faults and present novel insights into the nature of these faults and the errors that they induce.

Keywords- Reliability, DRAM, Errors, Faults, Data Analysis

I. INTRODUCTION

Reliability is an ever-increasing concern in the design of increasingly complex computing systems. Server systems require reliability to be a first-class constraint. Data centers and server farms deploy server systems into high-density racks where they are generally highly utilized and have a high need for dependability. Reliability issues on even a few server systems can have a detrimental impact across the entire deployment.

With the advent of numerous services, ranging from internet search to cloud computing, there has been a significant expansion in data center capacity worldwide. These data centers contain hundreds to hundreds of thousands of servers, and users often use the applications hosted on them on a 24/7 basis worldwide. This requires various guarantees of dependability, criticality, and service quality. Given the scale of the computing infrastructure used to host such applications, performance, and dependability requirements, and total cost of ownership (TCO) constraints, it is imperative to design individual servers so their fault rate is very low. The fault rate of a server system is ultimately the sum of the fault rates of its individual platform components, including the processor, memory, storage, and network interfaces. A single fault in any of these components can generate an arbitrary and large number of errors. There has been a large body of research into developing techniques to enhance the reliability of each of these components and on assessing their reliability under controlled laboratory or accelerated testing conditions. However, it is equally important to understand the nature of the faults when the servers are under realistic usage conditions and over long periods of time as they are actually deployed in the field. Information and insights obtained from such field data serve as a valuable feedback to server architects and designers and also to data center administrators who deploy and manage the servers that host the applications.

This paper will present an in-depth study focusing specifically on the main memory component of system reliability for a large population of servers deployed in data centers. Modern servers use large and ever-increasing amounts of main memory to house application working sets and therefore failures in the memory system can have a profound impact on performance, availability, and TCO. Despite the importance of memory errors, our collective understanding about the rate, pattern and impact of memory errors is not well developed. Existing research on memory errors often utilizes synthetic error models [2, 3, 13] because of the lack of realistic error data. However, there have been a few recent field-data studies on memory errors [1, 4, 14, 15]. These field data analyses indicated that the main memory systems of servers experience both soft errors and hard errors, hypothesized that all these errors occur in the memory modules (DIMMs), and suggested that these errors are correlated with temperature and utilization. While these prior works serve as motivation to further examine the nature of errors in the memory system, they restrict their analysis to errors in the memory modules. They also do not attempt to classify the nature of the errors observed in these components.

Due to the requirement for main memory to be highly dependable, server systems generally protect this component to a high degree using techniques such as ECC and/or CRC. Such techniques greatly increase the system memory’s fault tolerance and the ability to recover from otherwise debilitating errors. In this paper we focus primarily on corrected errors observed in main memory. Corrected errors occur far more often than uncorrected errors and provide excellent insights into the types of errors and underlying phenomena affecting main memory. By understanding corrected errors better we can predict the trends that will affect uncorrectable errors before they become a real problem. In this paper, we expand the scope of studying correctable memory system errors and make the following key contributions:

- We extend the field-data collection (collected from over 30,000 systems during a period of 3 years) and analysis of the main memory system to include multiple structures on the memory path, including the memory controllers, buses, channels, and memory modules. Based on our observations, we present a novel taxonomy for errors in the memory path.
- We provide a detailed characterization of the errors and present novel insights into their nature.

The next section discusses the related work. Sections III provides a brief overview of memory systems and memory errors respectively. Sections IV and V describe the field-data monitoring system and the methodology for error analysis. Section VI shows the error analysis results. Finally, Section VII concludes this paper.

II. RELATED WORK

Previous work investigated the impact of particle-induced errors on DRAM under controlled laboratory conditions [8, 10, 11, 12]. More recent studies [1, 4, 14, 15] explored DRAM errors in large server populations. They indicated that main memories of servers experience both soft and hard errors and suggested they are correlated with server temperature and utilization. The analysis is made under the assumption that all errors occur in memory modules (DIMMs). While our study supports some findings of these earlier studies, we also consider errors in several other components along the memory path, provide additional novel insights and provide evidence that contradicts certain observations in the previous work. In general, we believe that this work complements the existing body of research on memory system errors and motivates further field-data analysis studies.

III. MEMORY ERRORS/FAULTS

Modern processors often integrate on-die memory controllers, which are connected to multiple Dual Inline Memory Modules (DIMMs) via channels [5]. Channels are complex signaling systems, which transmit information or signals between the DIMMs and memory controller in the memory system. Collectively, we refer to all the channels as the memory bus. Each DIMM can contain one or more independent ranks, where each rank is a set of DRAM chips. Internally each of the DRAM chips are implemented with one or more independent banks and each bank is composed of memory arrays.

To understand memory reliability, there are three main terms that must be clearly understood: fault, error and failure. A *fault* is a state corruption in a memory system where one or more bits become corrupted due to either hardware defects or transient factors. The presence of a fault in the memory may or may not lead to an error or a failure. When a faulty bit is accessed, the outcome of that access is called an *error*. A *failure* is an instance in time when a system displays behavior that is contrary to its specification and is recorded at the system boundary. It is basically an error that has propagated to the system boundary and has become observable. In this work, we primarily focus on memory faults and errors. Since all the memory errors we observed were corrected, none of them resulted in a failure.

There are primarily three types of memory faults: hard faults, particle-induced soft faults, and transient faults. Depending on the origin of the faults in the main memory system, we classify the memory faults in four more categories: Module Fault (MF), Memory Controller Fault (MC), Channel Fault (CF) and Bus Fault (BF). Depending on their appearance or their pattern of occurrence over time, each of the faults is classified as hard, soft or transient fault.

IV. MEMORY ERROR MONITORING SYSTEM

Memory error data are collected using a software tool called System Environment Monitoring Agent (SEMA). SEMA is a distributed software infrastructure that collects processor and operating system data from a large pool of systems. The SEMA infrastructure consists of two main

components: a lightweight client that collects and transmits data and a back-end system that maintains the data repository and provides data analysis support (Figure 1). SEMA currently collects a wide range of system identification, processor and operating system data, such as error data, system utilization, memory usage, power states and temperature. Custom-made analysis tools periodically analyze the collected data and produce analysis reports. The SEMA client (Figure 2) is supported on both Windows and Linux operating systems. It consists of a user-level application that periodically collects, pre-processes, and transmits data of interest and a kernel-level driver. The kernel driver provides access to a processor’s Model Specific Registers (MSRs) [6].

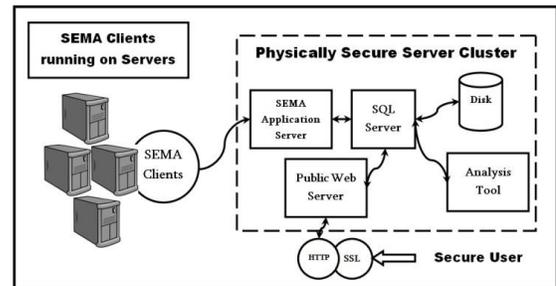


Figure 1: The SEMA Infrastructure

Hardware errors are collected using the machine-check architecture which defines several error-reporting banks [6]. Each error-reporting bank is associated with a specific hardware unit (or group of hardware units) in the processor. SEMA monitors these error-reporting registers by periodic polling and keeps track of all memory error events with detailed information, such as timestamp, memory address, DIMM and channel numbers and event type that caused.

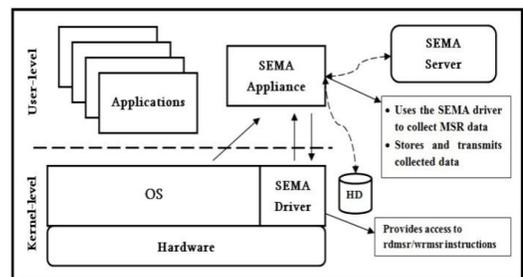


Figure 2: SEMA client architecture

Memory error data are collected from over 30,000 systems during a period of 3 years. The memory capacities of these systems range from 0.25GB to 256GB. These systems are not employed with any special hard error handling techniques (i.e. mapping out pages with hard errors, decommissioning the failed blocks etc.). However, every system is protected by techniques such as ECC and/or CRC. Along with those, two different types of hardware scrubbing are also employed: patrol scrub (periodic) and demand scrub (whenever a memory access detects an error). Both types of scrubbing rewrite the data, which repairs transient faults. These techniques greatly increase system memory’s fault

tolerance and its ability to recover from otherwise debilitating errors. Corrected memory errors (CME) occur far more often than uncorrectable memory errors (UME) because of these reliability techniques and provide excellent insights into the types of faults and underlying phenomena affecting main memory. By understanding CME better we can predict the trends that will impact UME before they become a real problem and improve the protection on our systems proactively. Our monitoring system isolates the errors generated from the scrubbers and less than 0.5% of the errors are observed due to scrubbing. In this work, we therefore focus primarily on CME (reported by ECC) observed in main memory.

V. METHODOLOGY FOR ERROR/FAULT ANALYSIS

During analysis of the CME data, we first identify the primary type of faults (hard/soft/transient). For this purpose we use the timestamp of the errors and the corresponding memory addresses. We collapse multiple error reports with a common address pattern (at least 8 address bits in common) into a single fault. We classify faults based on the following observations:

i) *Corrected Hard Faults*: Repeated occurrence of corrected errors at a regular fashion in a system is most likely due to hard faults. These repeated errors could be either generated from the same addresses or from different addresses depending on the error sources. All the corrected errors that show persistent behavior over time are classified to be generated from corrected hard faults.

ii) *Corrected Transient Faults*: Corrected errors that are repeated but occur rarely and are not regularly spaced in time are highly unlikely to be outcomes of particle induced soft faults. Very often such errors tend to show up in bursts as an outcome of unwanted noise in the systems and get generated from different addresses. We classify these types of errors to be generated from corrected transient faults.

iii) *Corrected Soft Faults*: Random occurrences of corrected errors in random addresses are classified to be results of soft faults. If more than one corrected error gets logged from the same address at different points of time, they are not classified as corrected soft faults.

TABLE I. TRUTH TABLE TO DECIDE FAULT TYPE BASED ON DIMM, CHANNEL AND ADDRESS PATTERN (X = DON'T CARE CONDITION)

DIMM	Channel	Address	Fault Type
Same	Same	X	MF
Not Same	Same	Pattern	MF (multiple faulty DIMMs)/MC
Not Same	Same	No Pattern	CF
X	Not Same	Pattern	MC
X	Not Same	No Pattern	BF

Once we identify the primary fault type associated with the error, we next categorize the hard faults into four different groups- MF, MC, CF and BF, as described in Section III. We do not attempt to classify transient faults further into additional groups, since we do not have enough data for this purpose. Occurrences of transient errors are

sporadic in nature and there exist several possible sources of transient errors (e.g., skew, jitter, weak bits) with similar error patterns.

Depending on the origin of the CME for each system, we classify the hard faults into MF, MC, CF or BF. Origin identification is based on the DIMM number (single or multiple DIMMs), the channel number (single or multiple channels) and the memory address pattern. Memory addresses are considered to follow a common pattern if they have at least eight address bits in common. Table I shows the possible combinations of the DIMM number, channel number and address pattern that is used to decide the type of memory faults.

Errors generated from the same DIMM and channel, irrespective of the actual addresses, indicate a corruption in a particular DIMM and, hence, are classified as MF. In the case of errors generated from different DIMMs, but the same channel, we check the address patterns to decide whether the faults are MF or MC or CF. If the errors exhibit common patterns in the addresses across all the DIMMs, we speculate that the memory controller decoder contributes to the errors and the system experiences MC. On the other hand, if the errors exhibit different address patterns and the errors are confined within each DIMM, we assume the system has multiple corrupted DIMMs and experiences MF. In the event of no pattern in the addresses, since all the errors are coming from the same channel, we assume the system experiences CF. In the case of errors generated from different channels irrespective of the DIMMs, we again look at the addresses for patterns to decide whether the faults are BF or MC. If the addresses exhibit patterns, the system experiences MC, otherwise the faults are BF.

VI. ERROR/FAULT ANALYSIS

A. High-level analysis

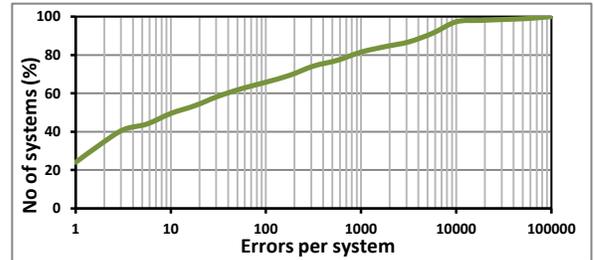


Figure 3: CDF of the number of errors per system

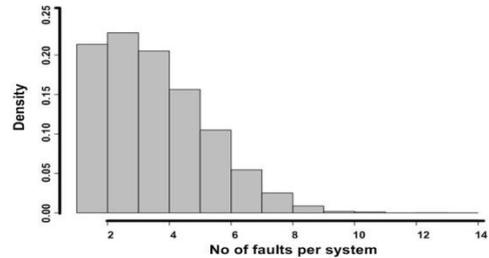


Figure 4: Distribution of number of faults per system

Having described our methodology for classifying the CME in the main memory system, we now present an

analysis of our field data and discuss the trends that we observe. While we are unable to disclose the exact number of systems that reported errors due to confidentiality reasons, it was in the order of a few thousands. The number of CME reported by the systems ranges from 1 to 100000, which gives a total of billions of CME over a 3 year period. Figure 3 shows the cumulative distribution function (CDF) of the number of CME per system. Only systems that have reported at least one error are considered in this analysis. About 20% of the systems experience a single CME and 46% of the systems experience more than 100 CME. 20% of the systems report more than 1000 CME and contribute 95% of the total error reports. We find that systems that experience less than 50 CME are the potential candidates of soft faults and transient faults.

After analyzing all the error reports, we categorize the faults into soft, transient or hard faults. We look at the total number of hard faults in each system across the population. We collapse multiple error reports with a common address pattern (at least 8 address bits in common) into a single fault. Figure 4 shows the distribution of the number of faults across the population. Overall, most of the systems have few numbers of faults (between 2 to 6) and a tiny percentage of the systems have more than 10 faults. It is important to note that a small number of faults generate a large number of errors (98.5%). Another interesting observation is that mostly the faults are co-located in the memory system. These insights might assist in developing intelligent hard error management techniques.

TABLE II. SYSTEM AND ERROR REPORT PERCENTAGE PER FAULT TYPE

Fault Type	System (%)	Error Reports (%)
Soft Fault	17%	0.02%
Transient Fault	23%	1.48%
Hard Fault	60%	98.5%

Table II shows the percentage of systems and the percentage of error reports per fault type. We can observe that hard fault is the most prevalent type of fault. 60% of the systems experience hard faults whereas rest of the 40% of the systems contributes to soft and transient faults. Since error is the outcome of a fault, it is important to understand the characteristics of the faults in the memory systems. Now we begin our detailed analysis by looking at the characteristics of hard faults.

B. Hard Faults

i. Breakdown of Errors

Table III presents a breakdown of systems and error reports by hard fault type. Only systems that have reported due to hard faults are considered in this table. The table provides significant insight about the dominance (fraction of systems affected) and intensity (fraction of error reports) of each fault type.

Corrected module faults (MF) are the most dominant fault type, affecting 83% of the systems with corrected errors. This result is in-line with our expectations, since the

DIMMs constitute the largest part of the main memory system. A non-trivial number of systems experience corrected bus (BF), memory controller (MC), and channel (CF) faults. Specifically, 9% of systems experience BF, 4% MC and 4% CF. These results highlight the importance of these components for main memory reliability analysis. Recent field data analysis studies indicate that main memory systems experience more hard faults than expected and attribute all errors to DIMM faults [1, 4, 14, 15]. Our data analysis suggests that corrected errors in memory components beyond the DIMMs are not uncommon (17% of systems).

TABLE III. SYSTEM AND ERROR REPORT PERCENTAGE PER HARD FAULT TYPE

Hard Fault Type	System (%)	Error Report (%)
MF	83%	66%
MC	4%	22%
CF	4%	4%
BF	9%	8%

Corrected memory controller faults (MC) correspond to 22% of the corrected errors and affect only 4% of the systems. Hence, systems experiencing MC generate a larger number of error reports and MC dominates over other error types. We explain this phenomenon as follows. Memory controllers consist of many decoders. Decoders have one-to-many mapping between input bits (address bits) and output bits (connected to other decoders or DIMMs) enabling access to multiple memory locations. Consequently, a corrupted address bit can propagate to multiple memory components resulting in multiple erroneous accesses.

TABLE IV. BREAKDOWN OF MF BY MEMORY SIZE, BF BY DIMM NUMBER, CF BY CHANNEL NUMBER

Memory Size	Module Fault (%)	DIMM	Bus Fault (%)	Channel	Channel Fault (%)
256 GB	42%	DIMM ₀	40%	Channel ₀	35%
128 GB	33%	DIMM ₁	60%	Channel ₁	27%
64 GB	25%	DIMM ₂	X	Channel ₂	38%

ii. Hard Fault Analyses

Impact of memory size on MF is shown in column 1 and 2 of Table IV. The memory sizes of the monitored systems range from 0.25 GB to 256 GB. Our data analysis shows that systems with large memories are more susceptible to corrected module faults (MF). Intuitively, the corrected memory error rate should increase as memory size increases, since larger memories consists of more DIMMs and, hence, transistors and other hardware. Our analysis supports this intuition. Note the rate of module fault increase is not necessarily proportionate to the rate of memory size increase. This is also expected as phenomenon such as AVF has been shown to attenuate the actual error rate increase [16]. *Column 3 and 4 of Table IV show the DIMM configuration sensitivity by presenting a breakdown of BF by DIMM number.* DIMM₁ is located further away from the memory controller than DIMM₀. Since longer

signal trace length can lead to dielectric loss, signal attenuation, or higher likelihood of crosstalk, DIMMs that are located further away from the memory controller may experience a higher rate of BF. Our data analysis corroborates this statement. *Column 5 and 6 of Table IV show the channel sensitivity by presenting the fraction of channels contributing to CF.* CF are distributed across all channels. A random distribution is expected because of two reasons. First, channels across the entire population are utilized in a uniform fashion. Second, the probability of a channel having a manufacturing defect should be equal for all the channels. The channel fault percentages seen are within 5% of the uniform distribution to each other, therefore, meeting our expectation.

iii. Impact of aging on memory devices

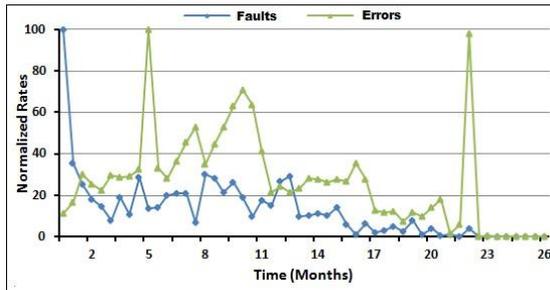


Figure 5: Fault Rate vs. Error Rate

Age is an important factor to consider when analyzing memory system reliability. To analyze the effect of age on the memory system, we look at the fault and error behavior over time across the entire population. Figure 5 shows the normalized fault rates and corrected error rates over time (in months). We calculate the normalized rate by deducting the minimum observed value from the actual value and dividing it by the range of the observed values (maximum value - minimum value). The corrected error rate varies significantly over time and do not demonstrate any relationship with time. Therefore, error rate change over time is not appropriate to examine the impact of aging on memory systems. Fault rate changes over time and follows the first two of the three distinct periods of the typical bathtub curve. Until 10-12 months, the fault rate decreases with time indicating behavior consistent with infant mortality. The fault rate approaches level state after this period, which is consistent with the normal life period. Our monitoring time was not long enough to capture any period with increasing fault rate. Unlike the previous study by Schroeder *et al* [1], our data shows a decreasing fault rate (the first 10-12 months) followed by a steady fault rate period. Schroeder *et al.*[1] uses error rate as metric to demonstrate the impact of aging and shows that their systems do not experience any decreasing error rate, however it increases somewhere in between 10-18 months. However, our data do not show any persistent increase in either of the error rate or the fault rate. One interesting insight provided by this data is that, most of the faults get exposed much earlier in the lifetime which can also be utilized to develop intelligent hard error management techniques.

C. Transient and Soft Faults

Corrected errors due to transient faults show up in bursts. According to our field data analysis, systems with transient faults experience either a single burst of errors or multiple bursts repeated in an irregular fashion. Single bursts appear in 30% of the systems with transient faults. The burst lasts between 1 and 18 hours. During the error burst, systems experience two to three corrected errors per hour. The majority of systems (70%) experiences multiple bursts characterized by different error patterns. Systems with multiple bursts tend to have burst lengths of 30 minutes and experience up to 30 corrected errors per burst.

Random occurrences of individual errors are classified to be generated from soft faults. In addition, a special case of corrected error bursts can be attributed to soft faults. This type of errors is known as block errors [17]. A single soft fault in the memory address decoder can lead to a burst of errors in the memory system. A read/write memory request starts with a Row Address Strobe (RAS) command. The RAS command carries the subset of address bits that identify the bank number and the row address within the specified bank. Each array within the specified bank reads an entire row of data. The number of data bits read out of the arrays involved in a bank access is usually equal to the page size (4KB-4MB). Of these, a piece of data is communicated on the memory, which is identified by the Column Address Strobe (CAS) command and associated subset of address bits. Consequently, a soft fault that occurs in the row decoder and corrupts the row address will affect all subsequent accesses to the arrays causing a burst of errors. In this case all errors will share a common address pattern. We find that 13% of the soft faults produce block errors (we consider systems with only one error burst).

D. Error vs. Utilization

This section investigates the impact of processor utilization and memory usage on memory errors. Prior work [1] suggests that there is correlation between error rate and utilization. We explore whether our field data corroborate this statement. The average number of memory accesses over time is a good metric for memory utilization. However, our data collection infrastructure does not collect data about memory accesses. Therefore, we use processor utilization (defined as fraction of time the processor is not idle) and memory utilizations (defined as fraction of system memory used) as proxies for memory accesses.

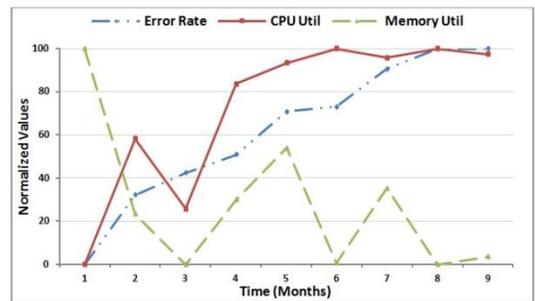


Figure 6: The effect of utilization on error rate

First, we examine the monthly error rate and utilization. We average the error rate, CPU utilization, and memory utilization across the whole population for each month during the entire monitoring period. Figure 6 shows a subset of the normalized monthly error rates, CPU utilization and memory utilization over time in months sorted by error rate (first month represents the month with the lowest error rate). We observe that CPU utilization follows closely the error rate: error rate increases with CPU utilization. This concurs with the findings in previous work [1]. In contrast to CPU utilization, memory utilization does not show clear correlation to error rate, which contradicts the findings by Schroeder *et al* [1]. A possible reason behind this discrepancy could be the differences in memory technologies used in and workloads running on the systems of the two studies. In order to ascertain the reason behind the error rate and CPU/memory utilization trends shown in Figure 6, we look at trends in the data at a finer granularity, over a time scale of hours. Figure 7 shows two representative examples of the observed trends in the error rates, CPU and memory utilizations (normalized values) over a period of hours. In both examples, the error rate is insensitive to memory utilization, a conclusion that is in-line with the analysis conducted at a monthly time scale. Figure 7(a) shows that CPU utilization has an inverse relation to error rate. This behavior could be explained by an inverse relationship between CPU utilization and memory traffic. Several classes of applications exhibit I/O intensive and compute intensive phases in an interleaved fashion. During an I/O phase CPU utilization is lower and memory traffic is higher (due to increased I/O) which can cause a higher error rate. Figure 7(b) reveals a different behavior. In the first part of Figure 7(b) (until the 70th hour), CPU utilization varies with time, while in the second part (beyond the 70th hour) it stays constant over time. In the first part, the error rate increases by a small amount when the CPU utilization decreases, while in the second case the error rate is insensitive to CPU utilization.

In summary, CPU utilization appears to correlate with error rate in a macroscopic scale (monthly analysis), possibly because higher CPU utilization may be associated with higher memory traffic over long monitoring periods. However, in a microscopic scale there is no strong correlation between CPU utilization and error rate. Hence, we conclude that CPU and memory utilization are not good predictors for error rates.

VII. CONCLUSION

Server systems require reliability to be a first-class constraint. It is important to understand the nature of faults when the servers are under realistic usage conditions and over long periods of time. Information and insights obtained from field data serve as a valuable feedback to both server architects and designers and also for data center administrators who deploy and manage the servers. This paper presents an in-depth study focusing on the main memory reliability for a large population of servers

deployed in data centers. We attempt to classify the errors observed in the components on the memory path and present a novel taxonomy.

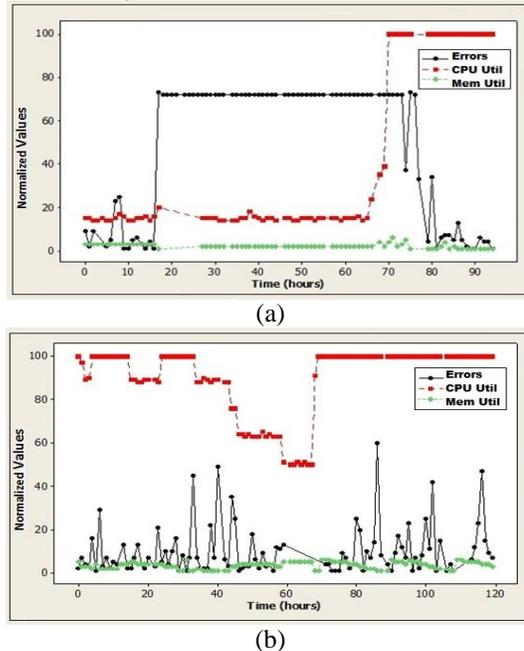


Figure 7: Error vs. Utilization at a finer granularity

REFERENCES

- [1] B. Schroeder et. al., "DRAM Errors in the Wild: A Large-Scale Field Study," In *Proceedings of the 11th international joint conference on Measurement and modeling of computer systems*, 2009.
- [2] M. Li et. al., "Understanding the Propagation of Hard Errors to Software and Implications of Resilient System Design," *ASPLOS*, 2008.
- [3] K. S. Yim et. al., "Measurement-based analysis of Fault and Error Sensitivities of Dynamic Memory," *DSN*, 2010.
- [4] X. Li et. al., "A Realistic Evaluation of Memory Hardware Errors and Software System Susceptibility," *USENIX ATC*, 2010.
- [5] B. Jacob et. al., "Memory Systems Cache, DRAM, Disk,".
- [6] "Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3 (3A & 3B): System Programming Guide".
- [7] S. S. Mukherjee et. al., "Cache scrubbing in microprocessors: Myth or necessity?," *IEEE PRISDC*, 2004.
- [8] A. H. Johnston, "Scaling and technology issues for soft error rates," *ACR*, 2000.
- [9] B. Schroeder and G. A. Gibson, "A large scale study of failures in high-performance-computing systems," *DSN*, 2006.
- [10] R. Baumann, "Soft errors in advanced computer systems," *IEEE DTC*, 2005.
- [11] T. C. May and M. H. Woods, "Alpha-particle-induced soft errors in dynamic memories," *TED*, 1979.
- [12] E. Normand, "Single event upset at ground level," *IEEE TNS*, 1996.
- [13] S. Nomura et. al., "Sampling+DMR: Practical and Low-overhead Permanent Fault Detection", *ISCA*, 2011.
- [14] A. A. Hwang et. al., "Cosmic Rays Don't Strike Twice: Understanding the Nature of DRAM Errors and the Implications for System Design", *ASPLOS*, 2012.
- [15] V. Sridharan and D. Liberty, "A Study of DRAM Failures in the Field", *SC*, 2012.
- [16] A. Biswas et. al., "Explaining Cache SER Anomaly Using DUE AVF Measurement", *HPCA*, 2012.
- [17] C. Poivey et. al., "Heavy Ion Induced Gigantic Multiple Errors in State of the Art Memories", *ESCC*, 2000.