

The Implications of Different DRAM Protection Techniques on Datacenter TCO

Panagiota Nikolaou
University of Cyprus

Yiannakis Sazeides
University of Cyprus

Marios Kleanthous
Mesoyios College
University of Cyprus

Lorena Ndreu
University of Cyprus

Abstract—¹ This paper proposes a framework for modeling the implications of DRAM failures and DRAM error protection techniques on the Total Cost of Ownership (TCO) of a data center. The framework captures the effects and interactions of several key parameters including among other: the choice of DRAM protection technique (e.g. single vs dual channel chipkill), DRAM capacity, device width (x4 or x8), power, FITs for various failures modes, the performance overhead of a protection technique for a given service and mix of co-located services. The usefulness of the proposed framework is demonstrated through several studies that identify the best, in terms of TCO, DRAM protection technique in each case. Interestingly, our analysis reveals that among the three DRAM protection techniques considered there is no one that is always superior to all others. Moreover, each technique is better than the others for some cases. This underlines the importance and the need for the proposed framework to avoid making suboptimal memory protection data center design decisions.

Keywords- *memory; DRAM; reliability; transient errors; permanent errors; total cost of ownership; datacenter; online services; offline services; availability; co-running services; peak throughput; performance.*

I. INTRODUCTION

During the last few years, datacenters (DC) have increased in number, size and utilization. Large DC that aggregate thousands to tens of thousands of servers are used to deliver services, such as e-mail, web search, social networking, maps etc., to billions of users. Key implications of this scaling is an increase in the cost and energy consumption of DC and, consequently, an urgent need for efficient methodologies and techniques for optimizing a data center's cost. The Total-Cost-of-Ownership or TCO, as it is typically referred to the cost of a DC, accounts for various DC costs such as hosting-facility, power provisioning, cooling equipment, server acquisition, software licenses, energy costs, repairs, management and personnel.

This work proposes a framework for analyzing the implications of DRAM failures and DRAM protection techniques on the TCO of a DC. DRAM failures and memory protection have received a lot of attention recently with several studies showing that DRAM is one of the main culprits for machine crashes and component replacements in today's datacenters and large supercomputers [1], [2].

In this work we argue that it is not straightforward to decide which DRAM protection scheme is best for a given DC setup. This challenge stems from the cost-benefit trade-off of each protection scheme with each offering distinct combination of

fault coverage, power, performance, and server overprovisioning. Server overprovisioning is needed to: (i) ensure peak throughput in the presence of errors since some servers may need to be offline until they are repaired or replaced, and (ii) compensate for possible performance degradation due to the protection scheme used. What is more, the specific cost-benefits may vary depending on the application characteristics run on the server, e.g. its memory sensitivity, the DC utilization and service co-location. These and other parameters, to be identified later, are used as inputs to the framework we propose in this work to determine what is the best in terms of TCO memory protection scheme for a given DC.

A recent work [3], very relevant to ours, performs software fault injection campaigns in DRAM to characterize the SDC rates of web services. They observe different sensitivity to faults across memory regions. They propose and analyze the cost of a heterogeneous memory protection scheme that employs in the same server DIMMs with and without ECC and maps pages to DIMMs depending on their SDC vulnerability. The methodology used in [3] to analyze the cost resembles the one we propose, but, with some notable differences. In particular, we a) account for the performance and power implications of ECC, b) consider the ramifications of co-located services, c) measure DC TCO, not only server cost, d) explore replacement and other maintenance policies and e) provide a detailed description to make easy the framework use. The organization of the rest of the paper is as follows: Section II covers background related to ECC and general DC organization. Section III presents an overview of the proposed framework. Section IV describes the evaluation methodology and Section V discusses the results. The paper concludes in Section VI.

II. BACKGROUND

Memory errors can be categorized into transient and permanent errors. Transient errors do not damage hardware but they can cause the reading of incorrect values from a memory location until overwriting the location. Permanent errors can cause physical damage and the faulty memory location can consistently return incorrect values [4]. Therefore, to detect and correct errors, memories typically include reliability features such as error correction codes (ECC). Depending on the ECC strength and the type of error, an error can be correctable (CE), detectable but uncorrectable (DUE) and non detectable (NDE) [5].

DRAM is protected from errors using extra devices per DIMM to store ECC codes. Typically codes today use 8/16 bits of ECC for every 64/128 data bits. A DDR3 memory channel is 72 data signals wide, 64 for data bits and 8 for ECC bits. Each memory channel can support one or more DIMMS. A

¹The research leading to this paper is supported by the European Commission FP7 project "Harnessing Performance Variability (Project No:612069 "Harpa")"

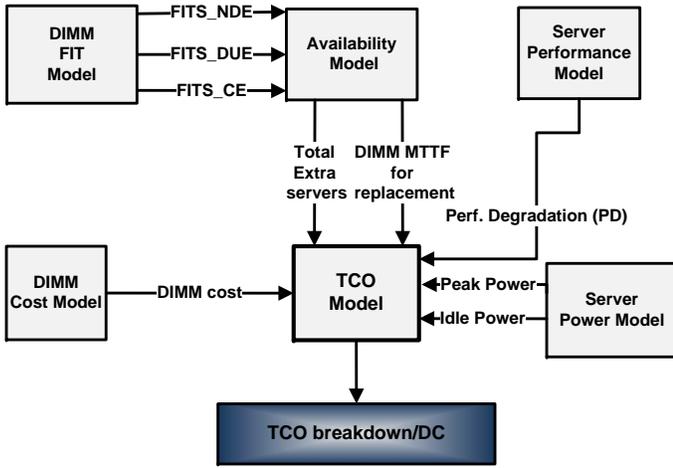


Fig. 1. Proposed framework Components and Information Flow

single DIMM consists of multiple DRAM devices that all or a subset, operate together to provide 72 bits. Each device can provide 4, 8 or 16 bits (referred to as x4, x8 or x16, devices respectively).

A processor may support various ECC options, with distinct code strength and overheads, of which one is selected at boot time. Below we describe three commonly used memory protection ECC codes that are also analyzed in this work.

Single Error Correction-Double Error Detection (SECCDED) [6] allows correction of all single-bit errors and detection of all two-bit errors in a 64 bit word using 8 bit ECC. Many triple bit errors are detected as DUE but some are miscorrected and lead to NDE. Also some of the quadruple bit errors are detected as DUE but some of them cannot be detected (NDE). SECCDED can be supported for both x4 and x8 devices. DIMMs with x8 devices consume less power than x4 DIMMs because they provide the same capacity with fewer devices [7].

Single-Chip error correction and Double-Chip error detection or Chipkill [8], is commonly used for DRAM protection in high availability servers and large scale systems because it can correct all the errors that appear in a DRAM device and detect errors in two DRAM devices. Chipkill relies on symbol-based coding to perform error detection and correction. In a symbol-based code, each codeword is composed of multiple symbols, with each representing a group of bits.

Modern processors usually support an implementation of Chipkill that employs 16 ECC bits for 128 data bits that are interleaved across two DIMMs found in two channels [9]. This implementation, using standard DDR3 with burst length of 8, reads two 64 byte blocks per access one of which is wasted for systems with 64B cache block size. Consequently, Chipkill can be wasting bandwidth, hurt performance and burn more energy [10]. To read only one 64 byte cache line per memory access, burst chop is used to reduce the burst length from the usual eight down to four [11]. Although, burst chop can be used to save the energy of four bursts, the timing of some accesses still requires 8 bursts and the bandwidth is still wasted [10]. We refer to this dual channel chipkill technique as **ChipkillDC**.

Another Chipkill implementation is similar to ChipkillDC with 16 ECC bits for 128 data bits, but the data and ECC bits come from one DIMM in a single channel [9]. This Chip-

kill implementation produces a codeword every two bursts. It is able to correct all errors in single device and detect 99.9999963% of the errors in two devices [9]. We refer to this single channel chipkill technique as **ChipkillSC**. A possible drawback of ChipkillSC is that it may not take advantage of the Critical Word First (CWF) optimization [12]. Consequently, this implementation may hurt performance because it needs to wait two bursts before forwarding data.

III. FRAMEWORK OVERVIEW AND METHODOLOGY

In this section we introduce a framework for assessing the implications of DRAM errors and protection techniques on the TCO of a DC. The proposed framework components and information flow are shown in Figure 1. As far as we know this is the first framework that attempts to combine all these variables together and eventually produce the TCO of a DC. The framework consists of six different models, Power, Cost, FIT, Availability, Performance and TCO. We next elaborate each model in more detail.

Server Power model: The Power model is used to estimate peak and idle power for a server. Regarding DRAM, the model estimates the power for 4GB DIMMs, using MICRON's DDR3 power calculator spreadsheet [13]. Based on the power calculator we determined the peak and idle power for x4 and x8 DRAM devices. The power numbers for each DIMM are presented in Table IV. These numbers are comparable with results reported in [7]. The power inputs for the other components (processor, disks, board) are also shown in Table IV. These numbers are derived from publicly available data [14], [15].

Server Performance Model: The purpose of the server performance model is to determine, using representative benchmarks, the performance of the various online and offline services using different ECC techniques at the datacenter scale. The proposed model, also, facilitates the comparison between DCs that differ in the ECC technique. For this purpose the performance model takes as input the performance difference of two servers used in different DCs. This is denoted as input (PD) to the TCO model and is used to calculate the extra servers required by a DC to match the performance of one without the performance difference.

It has been claimed in previous work that ChipkillDC, the technique with the strongest code, can incur up to 38% performance overhead compared to SECCDED for memory intensive workloads [16] due to its wasteful use of bandwidth (Section II). Similar claims have been made in several other studies [16], [17]. In this work we determine the ECC performance implications on real hardware where services are deployed.

FIT Model and DRAM Grades: The FIT model is used to produce the FIT rates for CE, DUE and NDE errors per DIMM for a specific ECC protection technique given a specific DIMM configuration. The failure rates can either be produced analytically using projected rates and failures distributions or rely on findings in field studies of DRAM errors.

For our experimentation we use the failure rates reported in [18]. To compute CE, DUE and NDE FIT rates for transient and permanent errors per DIMM for a given ECC technique we use analytical failure models, based on probabilities for

TABLE I
FAULT RATES OF TRANSIENT AND PERMANENT, CE, DUE AND NDE
ERRORS FOR EACH PROTECTION TECHNIQUE IN FIT/DEVICE

	Correctable (FITS_CE)		Uncorrectable (FITS_DUE)		NDE (FITS_NDE)	
	Tr.	Pr.	Tr.	Pr.	Tr.	Pr.
ChipkillDC	19.925	20.405	1.61E-4	5.53E-4	1.52E-16	1.81E-15
ChipkillSC	19.924	20.404	1.66E-4	5.65E-4	6.13E-13	2.09E-12
x4SECEDED	17.13	16.99	2.72	3.32	0.069	0.091
x8SECEDED	34.26	33.98	5.44	6.65	0.138	0.182

spatial errors. For SECEDED the probabilities are obtained for a given number of faulty bits whereas for chipkill for a number of faulty symbols. Each analytical model computes the probability for all device combinations that can produce a given number of faults. The probability for ChipkillDC DUE errors (P_{DUE}) is given by the following equation ²:

$$P_{DUE} = P_{fail2dev} \sum_{y=1}^7 \sum_{x=1}^7 (DF_x \binom{n}{1} P_x (1 - P_x)^{n-1}) \binom{n-1}{1} P_y (1 - P_y)^{n-2} \quad (1)$$

where the double sum is used to account for all the errors that experienced in two devices. The $P_{fail2dev}$ is the probability of an n -device DIMM to experience two device errors. The P_y is the probability of a device failure due to one of the 7 different types of errors (i.e. single-bit, single-word, single-column, single-row, single-bank, multiple-bank and multiple-rank) that occurs along with any failure in another device P_x . These two probabilities are derived using the raw FITS from [18]. We derate each combination with an appropriate factor, DF_x , to account for the likelihood of a fault combination happening in the same codeword. Note, that this equation estimates the total number of DUE FITS for both transient and permanent errors. Each combination contributes to different repair action depending whether it includes only transient faults or it has at least one permanent error. Finally, the DIMM $FITS_{DUE}$ for ChipkillDC is given as the product of $P_{DUE} \times 10^9$.

The results of the representative analytical equations are shown in Table I. Table I also presents fault rates for SECEDED with x8 devices assuming they are twice bigger than SECEDED with x4 devices. Since [18] does not provide raw fault rates for x8 devices we double the FIT rates of x4 devices. This effectively assumes no fault overlapping. This is a reasonable assumption for the fault rates in this work since the probability of overlapped faults is extremely low, 1E-15. We optimistically assume that there are no multibit errors with more than four bits in a x8 device.

One other FIT model parameter is the DRAM grade. The DRAM grade attempts to capture the variation in DRAM quality [19] with better grade DRAM experiencing less failures. It is expected that lower grade DRAMs cost less [19], [20] and thus present an opportunity for trading reliability for TCO optimization. The DRAM grade is given in the FIT model as a numeric factor that is used to multiply the fault rates in Table I, i.e. the larger the DRAM grade factor the higher the failure rates. The range of factors considered is hypothetical

²The other equations have similar structure and are not presented due to space limitations.

TABLE II
MTTR FOR VARIOUS REPAIR ACTIONS DUE TO
DIFFERENT TYPES OF FAILURES

	Details	Time
$MTTR_{DIMM_rpl}$	Replace DIMM	1440 min
$MTTR_{pg_r}$	Page retirement	100 min
$MTTR_{r_bt}$	Server reboot	100 min
$MTTR_{ecc}$	ECC	0 min

TABLE III
SERVER AND MAIN MEMORY CONFIGURATION

Number of CPUs	2
CPU	Intel Xeon E5620 2.4GHz
Number of cores per CPU	4
DRAM technology	DDR3
Channels per CPU	2
DIMMs/channel	1
Ranks/DIMM	2
DRAM device	x4
DIMM capacity	4GB
Turbo mode	disabled

and aims to explore if and how big of an opportunity DRAM grades present for TCO optimization.

Availability Model: The Availability model takes as inputs the FIT rates for CE, DUE, and NDE for a given ECC scheme, produced by the FIT model and using different hardware and software repair techniques estimates the extra servers that are needed to ensure the peak throughput.

We considered different repair techniques such as: ECC protection, page retirement, server reboot and DIMM replacement. ECC can be used to repair transient correctable errors. Page retirement can be used to repair correctable permanent errors. Server reboot can be used to repair transient uncorrectable errors. In this work the replacement policy that we used when a component fails is to replace only the faulty component and not the entire server. Therefore, DIMM replacement can be used to repair uncorrectable permanent errors.

Table II lists the mean time to repair assumed for each repair technique. We assume that ECC correction has a negligible MTTR. We have checked a range of values for reboot and page retirement and do not observe much sensitivity due to the rarity of these events. It should be noted that the model is not specific to the techniques and repair times that are shown in Table II and other repair schemes and MTTRs can be added to the model.

To compensate the performance loss due to the time required to repair faulty DIMMS a DC needs to be overprovisioned with extra servers. The following are used to calculate the extra servers to cover performance loss due to: page retirement repairs (N_{pg_r}) and server reboot repairs (N_{r_bt}). The ECC repairs (N_{ecc}) is zero because $MTTR_{ecc}$ is negligible.

$$N_{pg_r} = \left(1 - \frac{\frac{MTTF_{pr_CE}}{\#n_DIMMS}}{\frac{MTTF_{pr_CE}}{\#n_DIMMS} + MTTR_{pg_r}}\right) * (N_{srvmodulesreq}) \quad (2)$$

$$N_{r_bt} = \left(1 - \frac{\frac{MTTF_{tr_DUE}}{\#n_DIMMS}}{\frac{MTTF_{tr_DUE}}{\#n_DIMMS} + MTTR_{r_bt}}\right) * (N_{srvmodulesreq}) \quad (3)$$

where the $N_{srvmodulesreq}$ is the initial number of server modules required for the peak workload assuming no failures and $\#n_DIMMS$ is the number of DIMM slots per server module. The $MTTF_{pr_CE}$ for page retirement is given by $\frac{10^9}{FITS_{pr_CE} * \#devices}$, the $MTTF_{tr_DUE}$ for server reboot is

TABLE IV
SERVER CONFIGURATION AND PARAMETERS

Components	Cost(\$)	Power(W)	Power idle(W)
1 Processor	193 [21]	89 [14]	20 [14]
2 Disks	60 [22]	6.8 [22]	0.8 [22]
Other(Case, power supply & motherboard)	308 [15]	13.4 [15]	6 [15]
DRAM Protection			
x4	75.84 [23]	5.95	1.29
x8	57.68 [24]	3.50	0.81
Server Parameters	Value		
server utilization	0.3		
# active cores	2		

TABLE V
DATACENTER CONFIGURATION

Cost of building	3000\$/m ²	Maintenance salary per rack	200\$ (monthly)
Cost of cooling	12.5\$/W	DC depreciation	15 years
Cost of electricity	0.07\$	DC utilization	30%
Cooling area over.	1.2	Server depreciation	3 years
Network per rack	10K\$	Server modules	50,000

obtained using $\frac{10^9}{FIT_{S_{tr_DUE}} * \#devices}$. Finally the total number of extra servers needed to makeup for performance lost due to memory modules repairs is determined by summing the above as:

$$Total_{extra_servers} = [N_{pg_r} + N_{rbt}] \quad (4)$$

Finally, another output of this model is the MTTF for a DIMM replacement due to a DIMM uncorrectable permanent errors, $MTTF_{pr_DUE}$ is calculated using the following:

$$MTTF_{pr_DUE} = \frac{10^9}{FIT_{S_{pr_DUE}} * \#devices} \quad (5)$$

This is used by the TCO model to determine the number and cost of the replaced DIMMS.

DIMM Cost Model: The parameters of the DIMM cost model for 4GB DIMMs for x4 or x8 devices, are shown in Table IV. These values are obtained from public data [23], [24].

TCO model: The last component is the TCO Model that is used to estimate the DC cost. The model is based on the COST-ET tool [25] extended to take as inputs the various outputs produced by the various models presented so far. We implemented the TCO model as a wrapper around this earlier tool. To evaluate our model we assumed High-Performance processors from Intel. For each experiment an initial population of 50,000 server modules with an average server utilization of 30% [26] is assumed.

IV. EXPERIMENTAL METHODOLOGY

To evaluate ChipkillDC's and ChipkillSC's performance overhead we use dual socket Intel Xeon E5620 systems with the configuration shown in Table III. To measure the performance degradation of ChipkillDC the server memory system is first set in "Lockstep Mode". This mode combines two DIMMs in different channels together to form a 144 bit word [27]. Then the servers are set in "Advance ECC Mode". This mode uses a single channel and corresponds to ChipkillSC. We set these settings by accessing the BIOS through a BIOS Serial Command Console interface (CLI) [27]. The evaluation used an online (*Web Search*) benchmark from CloudSuite [28] and an offline (*MCF*) SPEC CPU2006 program [29]. MCF is

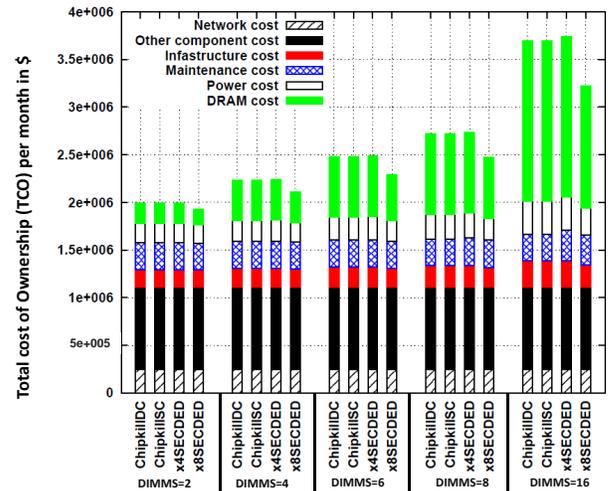


Fig. 2. TCO results for different DIMMS slots

selected because is a memory intensive application, and can help underline the significant impact on performance caused by ChipkillDC. To evaluate the performance overhead of Web Search benchmark, four servers are used: one client server with multithreaded client processes where each process runs on a single core, one frontend server, one index server and one document server.

For the Web Search experiments a traffic of 100K queries is used with a working-set of 6GB. To increase the number of concurrent requests we increased the number of clients.

We run each experiment 5 times and each time we collected the total execution time. The results present the average execution time after by removing the min and max values. We also run the two applications co-located. Co-location can improve the machine's utilization by increasing the number of active cores in a server, but this should come at minimal cost for the QoS of the online services [30]. The memory intensive offline application (MCF) is run concurrently in the server that performs the indexing of the Web Search(index server). The performance degradation is measured for the Online service by running it first in isolation and then in combination, for the different ECC schemes and number of threads. Unless noted otherwise TCO results are presented without co-location assuming servers running 2 index Web Search threads on two cores. The analysis with co-location runs on two cores Web Search and on the other two cores MCF. The cost and power inputs for the server that are shown in Table IV are derived from publicly available data [14], [15]. Finally, the parameters for the datacenter configuration are shown in table V. These parameters are obtained from the literature [26], [31], [32], [25] and real datacenters.

V. RESULTS

This Section investigates how various parameters affect the DC TCO and the choice of the DRAM protection technique. We present different case studies to assess the impact of (a) the number of DIMM slots, (b) DRAM grades (increasing fault rates), and (c) ChipkillDC Performance and TCO overhead for online and co-located services.

A. Implications of DRAM Capacity on the DC cost

In this case study, we investigate how the TCO for each protection technique is affected by the number of DIMM

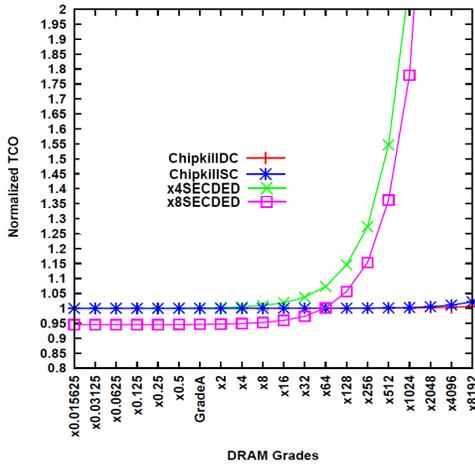


Fig. 3. TCO results for different DRAM grades

slots (4GB per DIMM) per server. Figure 2, shows for each protection technique how the number of DIMM slots per server node affect the DC TCO. The x-axis presents different protection techniques according to the number of DIMMs per server and the y-axis shows the TCO breakdown (infrastructure, network, maintenance, power, DRAM and other server components cost such as disk, cpu, board etc) per month in \$. As shown in Figure 2, while the number of DIMM slots in a server increases, the DC cost for all the protection techniques also increases. Also we observe that as the number of DIMM slots increases the x8SECEDED offers better TCO as compared to the other protection techniques presented. More specifically, in the case with 16 DIMMS per server, the TCO of x8SECEDED is 12.7% better compared to the TCO of ChipkillISC and ChipkillIDC, and 13.7% better compared to the TCO of x4SECEDED. As shown in the breakdown, this improvement comes from a reduction in DRAM and power cost. The x8SECEDED TCO increases at a slower rate with increasing DIMM slots, because of the lower cost and power of the x8 devices.

B. DRAM grades and TCO

The server configuration for the following use cases has four DIMMs, two in each channel, for a total of 16GB per server node.

As mentioned earlier, it is interesting to investigate the trade-offs of several grades of DRAM quality with higher cost for better quality and more reliable parts. Our analysis assumes a large range of DRAM grades to better explore the possible opportunities from having DRAM products with varying quality (fault rates). Figure 3 shows the normalized TCO results for different DRAM grades. We present 20 grades including GradeA. The FITs of GradeA correspond to the ones in Table I. At this point we consider DIMM cost to be the same for all grades, we will examine the cost issue later. All grades are correlated to GradeA by some factor (e.g. x2 is derived multiplying the fault rates of GradeA by two). The various curves correspond to different protection techniques and are normalized to the TCO with respect to a DC with ChipkillIDC using GradeA.

As shown in Figure 3, when DRAM grades are x64 or less, the x8SECEDED is the best TCO choice. For the grade range

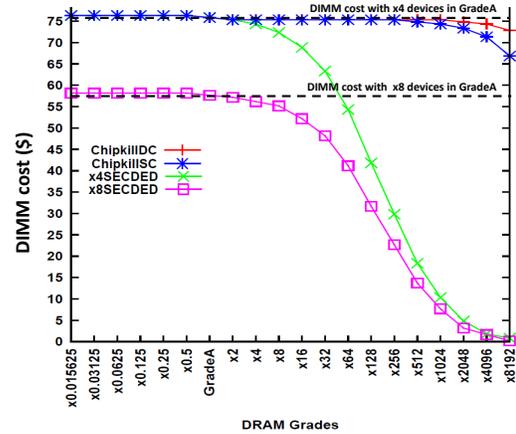


Fig. 4. DRAM cost in \$ for different fault rates (DRAM Grades)

between x64-x512, the TCO of x8SECEDED increases, the TCO of ChipkillISC and ChipkillIDC are equal and these two protection techniques become the best choices. For DRAM grades larger than x512, we observe that ChipkillIDC is better with a small difference from ChikillISC. The results show that the TCO of ChipkillIDC and ChipkillISC is significantly less sensitive to increasing failure rates as compared to x4SECEDED and x8SECEDED.

Next we explore the trade-off between DRAM cost and reliability. The results presented in Figure 4 show what should be the DRAM cost for each protection technique to maintain the TCO constant. There is an opportunity to reduce the TCO in the case where DRAM cost is lower than the reduction needed to keep the TCO constant (below each line). For both SECEDED schemes the DIMM cost needs to become significantly less to break even on TCO with lower grade DRAMs (for x4SECEDED \$50 less per DIMM for a x256 fault rate increase). On the other hand, the results reveal that ChipkillIDC(ChipkillISC) with a x2048(x256) grade memory can achieve the TCO cost of GradeA, with only a \$1 reduction in the per DIMM cost.

C. ChipkillIDC Performance Overhead and TCO for Online and Offline jobs

To determine the performance of ChipkillIDC and ChipkillISC, we run Online and Offline services in isolation and co-located using both memory protection schemes. The online application used is Web Search benchmark and as an offline application we use the SPEC 2006 (MCF) benchmark.

Figure 5 presents how the performance implications of Web Search running in isolation and co-running with MCF affects the TCO by estimating the extra overhead needed of a datacenter to match its performance with another using a different memory protection scheme.

Figure 5 presents the TCO for ChipkillIDC and ChipkillISC while running Web Search on two cores per server (recall that we assume only two cores are used to meet a QoS constraint) and the TCO while co-running two Web Search with one MCF and two Web Search with two MCF instances. The TCO values are normalized to the TCO of ChipkillIDC while running only two Web search instances.

As shown in Figure 5 for two Web Search application running alone the TCO with ChipkillIDC is slightly better

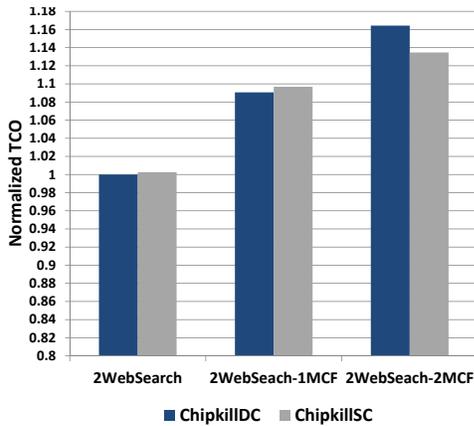


Fig. 5. TCO results for co-located services

than with ChipkillSC. On the other hand, if two Web Search instances are co-running with MCF instances, there is a moderate increase on TCO for both ChipkillDC and ChipkillSC. The extra TCO is due to the performance degradation from the co-location of Web Search and MCF and also due to the increase in the number of active cores. Increasing the number of active cores leads to an increase in the peak power resulting to an increase in the power cost and at the total TCO. Another key observation is that the TCO of ChipkillDC, while co-running two Web Search with one MCF applications is lower than the TCO of ChipkillSC. This may indicate that ChipkillDC benefits from the "critical word first" while ChipkillSC cannot. On the other hand, co-running two Web Search with two MCF results in the TCO of ChipkillSC to be lower than the TCO of ChipkillDC. This indicates, and confirms the earlier observation, that the relative performance implications of MCF when co-located with Web Search are different depending on the protection scheme. Also, it underlines the importance of understanding the usage and characteristics of all the services to be run in a DC before making memory protection design choices.

VI. CONCLUSIONS AND FUTURE WORK

This work shows that choosing the DRAM protection scheme that optimizes TCO is not a straightforward problem. The paper investigates the implications of DRAM failures and protection on the TCO of a DC. This work considers many salient parameters related to main memory protection, including performance, power, reliability, cost, etc. Several case studies are performed that reveal that no single protection scheme among those considered is the best in all cases. Furthermore, each technique is better than the others in some cases. Consequently, this demonstrates the usefulness and importance of the proposed framework to make successful DC design decisions. The findings of this analysis calls for manufacturers, vendor designers and researchers to consider using such a framework while exploring for a memory protection scheme that maximizes TCO. This framework can also be used for processor designers to quantify the benefits of new ECC options towards the DC TCO. Finally, the findings of this paper point to several directions of future research including more comprehensive evaluation of TCO for co-located services, analyzing the cost-benefits of 3D-stacked systems and explore the cost-benefits of new ECC schemes.

ACKNOWLEDGMENT

The authors would like to thank Zacharias Hadjilambrou from University of Cyprus for providing the Web Search infrastructure and the anonymous reviewers for their constructive critique and feedback that helped improve the paper quality.

REFERENCES

- [1] B. Schroeder and G. Gibson, "A large-scale study of failures in high-performance computing systems," *Dependable and Secure Computing, IEEE Transactions on*, 2010.
- [2] V. Sridharan and D. Liberty, "A study of dram failures in the field," in *High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for*, 2012, pp. 1–11.
- [3] Y. Luo, S. Govindan, B. Sharma, M. Santaniello, J. Meza, A. Kansal, J. Liu, B. Khessib, K. Vaid, and O. Mutlu, "Characterizing application memory error vulnerability to optimize datacenter cost via heterogeneous-reliability memory," *DSN*, 2014.
- [4] C. Constantinescu, "Impact of deep submicron technology on dependability of vlsi circuits," in *International Conference on Dependable Systems and Networks, DSN 2002.*, 2002, pp. 205–209.
- [5] C. Weaver, J. Emer, S. S. Mukherjee, and S. K. Reinhardt, "Techniques to reduce the soft error rate of a high-performance microprocessor," in *ACM SIGARCH Computer Architecture News*. IEEE Computer Society, 2004.
- [6] M. Y. Hsiao, "A class of optimal minimum odd-weight-column sec-ded codes," *IBM J. Res. Dev.*, vol. 14, no. 4, pp. 395–401, Jul. 1970.
- [7] S. Ankireddi and T. Chen, "Configuring and using DDR3 memory with HP ProLiant Gen8 Servers, Best Practice Guidelines for ProLiant servers with Intel Xeon processors," February 2014.
- [8] T. J. Dell, "A white paper on the benefits of chipkill-correct ecc for pc server main memory," *IBM Microelectronics Division*, pp. 1–23, 1997.
- [9] "Bios and kernel developers guide (bkdg) for amd family 15h," February 2014.
- [10] J. H. Ahn, N. P. Jouppi, C. Kozyrakis, J. Leverich, and R. S. Schreiber, "Future scaling of processor-memory interfaces," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, ser. SC '09. New York, NY, USA: ACM, 2009.
- [11] "MICRON.2Gb: x4, x8, x16 DDR3 SDRAM."
- [12] "Intel E8501 Chipset eXternal Memory Bridge (XMB)," 2006.
- [13] "Micron, <http://www.micron.com/products/support/power-calc.>"
- [14] "Intel Xeon Processor E3-power, [http://www.servethehome.com/intel-xeon-e3-1220-v3-benchmark-review-haswell-xeon/.](http://www.servethehome.com/intel-xeon-e3-1220-v3-benchmark-review-haswell-xeon/)"
- [15] "Intel cpu configuration, [http://www.rect.coreto-europe.com/track-server/lu-intel-server/.](http://www.rect.coreto-europe.com/track-server/lu-intel-server/)"
- [16] S. Li, K. Chen, M.-Y. Hsieh, N. Muralimanohar, C. Kersey, J. Brockman, A. Rodrigues, and N. Jouppi, "System implications of memory reliability in exascale computing," in *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for*, Nov 2011, pp. 1–12.
- [17] X. Jian, H. Duwe, J. Sartori, V. Sridharan, and R. Kumar, "Low-power, low-storage-overhead chipkill correct via multi-line error correction," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '13. New York, NY, USA: ACM, 2013, pp. 24:1–24:12.
- [18] V. Sridharan, J. Stearley, N. DeBardleben, S. Blanchard, and S. Gurumurthi, "Feng shui of supercomputer memory: Positional effects in dram and sram faults," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '13, 2013.
- [19] I. Cecil Ho, CST, "Innovative testing puts fallout dram back into systems," January 2003.
- [20] "Memory Test Background, 2000, [http://tinyurl.com/.](http://tinyurl.com/)"
- [21] "Intel Xeon Processor E3-cost, [http://ark.intel.com/.](http://ark.intel.com/)"
- [22] "Desktop Drive 500GB-cost and power, [http://www.ebuyer.com/.](http://www.ebuyer.com/)"
- [23] "Kingston Technology ValueRAM 4GB-x4 1600MHz DDR3-cost, [http://www.amazon.com/Kingston-Technology-PC3-12800-Workstation-KVR16R11S4/.](http://www.amazon.com/Kingston-Technology-PC3-12800-Workstation-KVR16R11S4/)"
- [24] "Kingston ValueRam 4GB-x8 1600 MHz DDR3-cost, [http://www.amazon.com/4GB-Module-1600MHz-Server-Premier/.](http://www.amazon.com/4GB-Module-1600MHz-Server-Premier/)"
- [25] D. Hardy, M. Kleanthous, I. Sideris, A. Saidi, E. Ozer, and Y. Sazeides, "An analytical framework for estimating tco and exploring data center design space," in *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*, 2013, pp. 54–63.
- [26] U. Hoelzle and L. A. Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 1st ed. Morgan and Claypool Publishers, 2009.
- [27] "Hp rom-based setup utility user guide," February 2014.
- [28] "Cloudsuite web search site, [http://parsa.epfl.ch/cloudsuite/search.html.](http://parsa.epfl.ch/cloudsuite/search.html)"
- [29] "Standard performance evaluation corporation, spec cpu 2006, [http://www.spec.org/cpu2006/.](http://www.spec.org/cpu2006/)"
- [30] J. Mars, L. Tang, R. Hundt, K. Skadron, and M. L. Soffa, "Bubble-up: Increasing utilization in modern warehouse scale computers via sensible co-locations," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-44. New York, NY, USA: ACM, 2011, pp. 248–259. [Online]. Available: <http://doi.acm.org/10.1145/2155620.2155650>
- [31] L. A. Barroso, J. Clidaras, and U. Hlzl, "The datacenter as a computer: An introduction to the design of warehouse-scale machines, second edition," *Synthesis Lectures on Computer Architecture*, vol. 8, no. 3, pp. 1–154, 2013.
- [32] J. Hamilton, "Overall data center costs, [http://perspectives.mvdirona.com/2010/09/18/overalldatacentercosts.aspx.](http://perspectives.mvdirona.com/2010/09/18/overalldatacentercosts.aspx)"