Future Scaling of Processor-Memory Interfaces

Jung Ho Ahn, Norman P. Jouppi, Robert S. Schreiber

HP Laboratories
HPL-2009-180

Keyword(s):
DRAM, memory system, rank subsetting, Multicore DIMM, mini-rank, ECC, chipkill

Abstract:

Continuous evolution in process technology brings energy efficiency and reliability challenges, which are harder for memory system designs since chip multiprocessors demand high bandwidth and capacity, global wires improve slowly, and more cells are susceptible to hard and soft errors. Recently, there are proposals aiming at better main-memory energy efficiency by dividing a memory rank into subsets. We holistically assess the effectiveness of rank subsetting in the context of system-wide performance, energy-efficiency, and reliability perspectives. We identify the impact of rank subsetting on memory power and processor performance analytically, then verify the analyses by simulating a chip-multiprocessor system using multithreaded and consolidated workloads. We extend the design of Multicore DIMM, one proposal embodying rank subsetting, for high-reliability systems and show that compared with conventional chipkill approaches, it can lead to much higher system-level energy efficiency and performance at the cost of additional DRAM devices.

External Posting Date: August 6, 2009 [Fulltext] Approved for External Publication
Internal Posting Date: August 6, 2009 [Fulltext]

To be presented at SC09, the International Conference for High Performance Computing, Networking, Storage and Analysis, November 14-20, 2009 at Portland, Oregon

Future Scaling of Processor-Memory Interfaces

Jung Ho Ahn
HP Labs
jung-ho.ahn@hp.com

Norman P. Jouppi
HP Labs
norm.jouppi@hp.com

Christos Kozyrakis
Stanford University
kozyraki@stanford.edu

Jacob Leverich
Stanford University
leverich@stanford.edu

Robert S. Schreiber
HP Labs
rob.schreiber@hp.com

ABSTRACT

Continuous evolution in process technology brings energy-efficiency and reliability challenges, which are harder for memory system designs since chip multiprocessors demand high bandwidth and capacity, global wires improve slowly, and more cells are susceptible to hard and soft errors. Recently, there are proposals aiming at better main-memory energy efficiency by dividing a memory rank into subsets.

We holistically assess the effectiveness of rank subsetting in the context of system-wide performance, energy-efficiency, and reliability perspectives. We identify the impact of rank subsetting on memory power and processor performance analytically, then verify the analyses by simulating a chip-multiprocessor system using multithreaded and consolidated workloads. We extend the design of Multicore DIMM, one proposal embodying rank subsetting, for high-reliability systems and show that compared with conventional chipkill approaches, it can lead to much higher system-level energy efficiency and performance at the cost of additional DRAM devices.

1. INTRODUCTION

Performance, energy-efficiency, and reliability are all critical aspects of modern computer systems, and all of them must be considered carefully when a new architectural idea is suggested. Process technology scaling makes transistors smaller and faster, enabling high-performance chip multiprocessors (CMPs) and main-memory DRAM chips with billions of transistors to be commodities. However, It is challenging to make integrated systems with these small and dense transistors achieve high energy-efficiency and reliability. Leakage and short-circuit power have become comparable to switching power on high-performance circuits, hurting energy efficiency. Small transistors and narrow wires increase the frequency of hard and soft errors.

Chip multiprocessors demand high memory throughput and capacity. Their throughput demands are high since many cores are simultaneously requesting memory and on-chip cache capacity is limited. Their memory capacity demands also increase, since the consolidation of workloads on a multi-core processor entails the amalgamation of their working sets. Since global wires, which are used to connect computation cores and storage cells, scale worse than local wires and transistors [11], meeting these dual demands of high throughput and high capacity is even more challenging when either energy efficiency or reliability is taken into account, and especially in high-availability systems where both are required. Moreover, power consumption has emerged as a major limiting factor in the design of contemporary datacenters since the cost to power a server can outweigh the cost to purchase the server over its life-cycle [4]. This motivates new tradeoffs in terms of capital and operating expenditures related to computing systems.

Recent proposals [2, 35, 37] share the main goal of saving dynamic main-memory access energy by dividing a memory rank into subsets, and making a subset rather than a whole rank serve a memory request. This saves dynamic power by reducing memory overfetch [2]. We refer to this category of proposals as *rank subsetting*. While promising, these studies on rank subsetting are limited. Module threading [35] focused on micro-architectural details and bus utilization; mini-rank [37] treated memory power and processor performance in isolation; Multicore DIMM (MCDIMM) [2] did not address memory capacity issues or evaluate DRAM low-power modes. Since a subset of a rank effectively has a narrower datapath than a full rank, data transfer takes longer, which can negate the benefit of dynamic power saving. It is therefore hard to judge if rank subsetting provides enough benefits to computer systems when both processor and main memory are considered and other architectural techniques are taken into account. Moreover, none of these previous studies analyzed or devised solutions for high-reliability systems, which are critical for enterprise computing.

This is the first paper which holistically assesses the effectiveness of rank subsetting on the performance, energy-efficiency, and reliability of a whole system, not just of individual components. We first model memory system power analytically and relate the degree of rank subsetting with memory capacity, system performance, and reliability. We validate these analyses by simulating a chip-multiprocessor system using multithreaded and consolidated workloads. We also develop a novel solution which extends the Multicore DIMM design for high-reliability systems, and show that compared with conventional chipkill [6] approaches it can lead to much higher system-level energy efficiency and per-

SC09 November 14-20, 2009, Portland, Oregon, USA

formance at the expense of DRAM capacity. Throughout our evaluation, we consistently use a *system energy-delay product* metric to judge the efficacy of rank subsetting in the context of a whole system.

Our key findings and contributions regarding rank subsetting in modern and future processor-memory interfaces are as follows:

- From model analyses and simulations, we show that 2 to 4 rank subsets is a sweet spot in terms of main-memory power and system energy-delay product. The best configuration varies by application characteristics and memory system capacity.
- We also identify that the effectiveness of subsetting memory ranks and exploiting DRAM low-power modes are largely complementary, since the former technique is applied to saving dynamic energy on memory accesses while the latter one is effective when DRAM chips mostly stay idle. They can be synergistic as well, which is especially apparent on high-reliability systems since both access and static power of main memory take a large portion of total system power.
- Finally, we demonstrate that rank subsetting affords a new category of trade-off in the design of high-reliability memory systems. Traditional chipkill solutions achieve energy-*inefficient* chip-level error recovery at no capacity or component cost relative to conventional ECC, whereas rank subsetting enables energy-*efficient* reliability in exchange for reduced capacity-efficiency.

2. ENERGY EFFICIENT AND RELIABLE MEMORY MODULES

In this section we first review modern processor-memory interfaces and the concept of rank subsetting, which has been recently proposed to improve the energy efficiency of main memory accesses. System-level impacts of rank subsetting are analyzed on performance, energy-efficiency, and reliability, which are all tightly coupled. Then we extend the Multicore DIMM design for high-reliability systems.

2.1 Background

The bandwidth and capacity demands from a modern microprocessor keep increasing since the processor has more computation cores, the cache size per core does not increase much, and emerging applications, such as in-memory databases, need even higher bandwidth, more capacity, or both from main memory. A single DRAM chip therefore cannot satisfy the latency, bandwidth, and capacity demands of a microprocessor as a main memory. As a result, several (typically 8 or 16) DRAM chips compose an access unit called a *rank* [15]. All DRAM chips in a rank operate in unison, i.e. receiving the same control (address and command) signals and transferring data in parallel. One or more ranks are packaged on a printed circuit board and called a memory module. Memory modules are connected to a memory controller, which feeds control signals to ranks and exchanges data through a shared bus to form a memory channel.

Memory controllers have historically been placed outside of microprocessors (in a chip called the northbridge), but are more recently integrated. Figure 1 shows a conventional memory channel which contains two DIMMs (dual in-line memory modules) attached to a memory controller through a bus. As data transfer rates increase, the maximum number of ranks attached to a bus decreases and control signals are registered per rank, due in both cases to signal integrity issues. To enhance energy efficiency, memory controllers may utilize the low-power states of DRAMs when the requests from the processors are infrequent. Error correcting codes (ECCs [29]) are often employed to cope with hard and soft errors on data storage and communication paths. Single bit error correction and double bit error detection (SECDED) is the most common scheme, but some computer systems need higher levels of error correction. The most well known technique to correct multi-bit errors is chipkill [6], which protects against single memory-chip failure.

Recently, there are new processor-memory interface proposals: module threading [35], Multicore DIMM [2], and mini-rank [37]. These proposals showed that more than half of DRAM power can be due to activating and precharging rows in DRAM banks. Since the size of a row in a modern DRAM chip (typically 8 or 16 Kbits) is larger than a cache line, it is observed that much of the DRAM power spent on row activation and precharging is effectively wasted. This inefficiency is called *overfetch*. These proposals alleviate the overfetch problem by dividing the DRAM chips within a rank into multiple subsets and making a subset (not a whole rank) serve a memory access. We call this technique *rank subsetting*. Rank subsetting requires minimal changes to the existing processor-memory interface, since conventional DRAM chips can be used without modification. We compare the differences between these proposals in Section 5. These proposals primarily save DRAM access energy, but have additional costs and benefits. Rank subsetting increases DRAM access latency and changes the effective bandwidth of memory channels. Since it changes the number of DRAM chips involved in a memory access, traditional reliability solutions such as chipkill must be revisited as well. So the effectiveness of rank subsetting must be assessed in the context of the performance, energy efficiency, and reliability of a whole system, not just of individual components.

2.2 Implications of Rank Subsetting

Rank subsetting addresses the overfetch problem by dividing each rank into smaller subsets of chips and sending memory commands only to a subset. Figure 2 shows an exemplary Multicore DIMM memory channel with two memory ranks, one per DIMM. Each rank is divided into two rank subsets called virtual memory devices (VMDs). Physically the same data bus as in a conventional memory channel is used, but it is divided into two logical data buses, each occupying half of the physical bus. A demux register is placed on each rank, which routes (demultiplexes) control signals to the proper rank subset to provide independent operations.

The primary goal of rank subsetting is to improve the energy efficiency of memory systems by saving DRAM access energy, which is important since main memory DRAM power can reach or surpass the processor power in high memory capacity or reliable systems as shown in Section 4. In order to understand how much energy can be saved by rank subsetting, we first identify the sources of DRAM power consumption. DRAM power can be categorized into two parts—static power and dynamic power. Static power is independent of activity, and mainly comprised of power consumed from peripheral circuits (like DLL and I/O buffers),

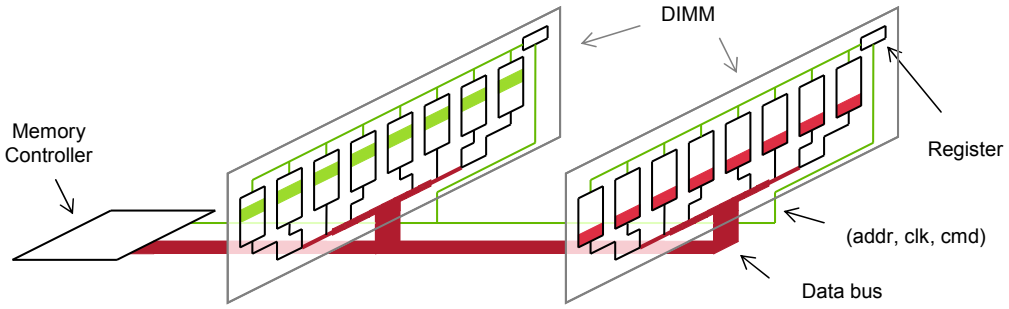


Figure 1: A conventional memory channel where a memory controller and two memory modules are connected through a shared bus. Each module is composed of one or more ranks, each rank with 8 or 16 DRAM chips.

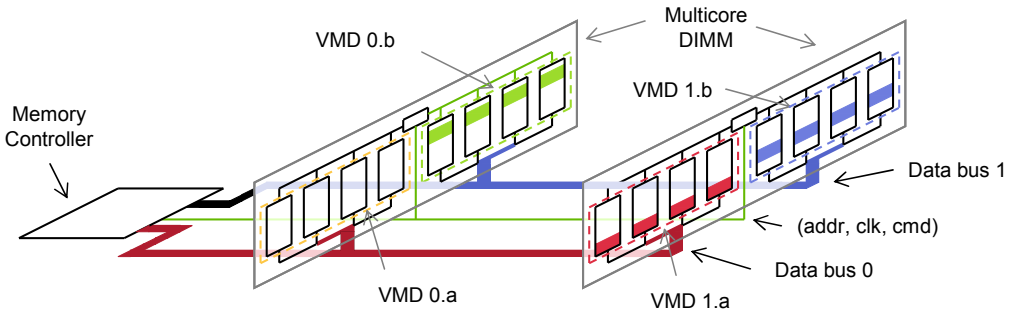


Figure 2: A memory channel with two Multicore DIMMs (MCDIMMs) with each divided into two subsets called virtual memory devices (VMDs).

leaky transistors, and refresh operations. Dynamic power can further be categorized into two parts since DRAM access is a two step process. First, bitlines in a bank of a DRAM chip are precharged, and data in a row of the bank is delivered to the bitlines and latched (activated) to sense amplifiers by row-level commands. This consumes activate-precharge power. Then, a part of the row is read or updated by column-level commands. This consumes read-write power. Dynamic power consumption is proportional to the rate of each operation. However since a row can be read or written multiple times once it is activated, the rates of activate-precharge and read-write operations can be different.

We can model the total power consumed in a memory channel as follows. When D is the number of DRAM chips per subset, S is the number of subsets per rank, and R is the number of ranks per channel,

$$\begin{aligned}&\text{Total main memory power}\\&= D \cdot S \cdot R \cdot SP + E_{RW} \cdot BW_{RW} + D \cdot E_{AP} \cdot f_{AP}, \quad (1)\end{aligned}$$

where SP is the static power of a DRAM chip, E_{RW} is the energy needed to read or write a bit[1], BW_{RW} is the read-write bandwidth per memory channel (measured, not peak), E_{AP} is the energy to activate and precharge a row in a DRAM chip, and f_{AP} is the frequency of the activate-precharge operation pairs in the memory channel. The first term of Equation (1) is the static power portion of the total memory power, the second is the read-write power, and the third is the activate-precharge power which, due to overfetch, grows linearly with D. Assuming that misses from last-level caches are the dominant portion of memory access requests, $BW_{RW} = f_{CM} \cdot CL$ where f_{CM} is the frequency of cache misses and CL is the line size of the last-level caches. If we analyze f_{AP} further,

$$f_{AP} = \frac{f_{AP}}{f_{CM}} \cdot f_{CM} = \frac{f_{AP}}{f_{CM}} \cdot \frac{BW_{RW}}{CL} = \beta \cdot \frac{BW_{RW}}{CL}, \quad (2)$$

showing that the dynamic power of main memory is proportional to the read-write bandwidth per memory channel and β, the ratio of the number of rows being activated to the number of memory requests to the memory channel. β indicates the frequency of bank conflicts. (A bank conflict occurs when successive requests to the same bank hit different rows forcing an activate-precharge.)

Rank subsetting lowers D while keeping $D \cdot S$ constant. So Equation (1) and (2) indicate that it mainly decreases activate-precharge power, which can be more than half of DRAM power [1]. (We show below that SP and BW_{RW} are affected by rank subsetting as well.) Saving activate-precharge power is more significant if β is close to 1, which is the case in modern chip multiprocessors running many threads or processes simultaneously. Figure 3 shows that for multithreaded and consolidated workloads the bank conflict ratio can approach unity as the number of software threads increases. Activate-precharge power can be lowered by increasing the cache line size (CL). However CL is typically around 64bytes while the size of a row in a rank is typically 8 or 16Kbytes, which is orders of magnitude larger.

[1]The major portion of read or write power consumption is from wires transferring control and data signals through chip-to-chip I/O and DRAM chip global interconnects, which is similar for both read and write operations. We therefore assume that both operations consume the same power as a first order approximation.

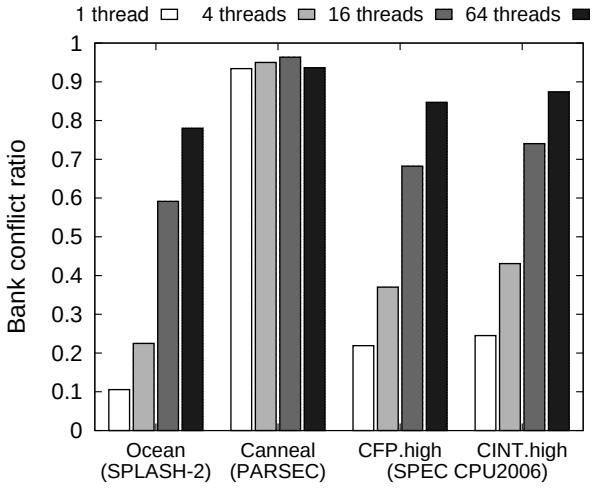


Figure 3: The bank conflict ratio (β) goes up as the number of software threads on a chip multiprocessor increases. A ratio of 1.0 means that a new row must be activated on every memory request, and hence is not able to exploit the spatial locality offered by a long row. These simulated results assume 4 memory channels having 1 rank per channel, 8 banks per rank, and memory access scheduling (See Section 3 for details on experimental setup).

Also larger cache line sizes can increase miss rate and harm application performance [8, 36].

Both dynamic power terms in (1) are proportional to the bandwidth of memory channels BW_{RW}, which is in turn proportional to instructions per cycle (IPC). Rank subsetting increases the access latency of memory requests. However, modern throughput-oriented CMPs such as the Sun Niagara [16] and the Intel Core i7 [14] can amortize or tolerate this latency with features like simultaneous multithreading or speculative out-of-order execution. They allow memory controllers to reorder and pipeline memory requests, to insulate bandwidth from memory latency.

Effective bandwidth depends on multiple factors, such as load-balancing across memory channels and DRAM banks, memory access patterns, and inherent timing constraints of DRAM chips. Recent DRAM chips are limited in the rate at which they can activate different banks in a device (`tRR` and `tFAW`). This limitation can lower the effective throughput since the bank conflict ratio is high on modern CMPs. This throughput degradation can be alleviated when R or S increases. As R increases, there are more banks a memory controller can utilize and there is no timing constraint on activating rows in different ranks. Rank subsetting also effectively increases the number of independent DRAM banks on a module, since each subset can be accessing different banks. As S increases, cache line transfer time per rank subset increases reaching or surpassing the minimal interactivate time, which effectively renders it irrelevant. Increasing S reduces the effect of other timing constraints as well, such as switches between read and write operations and bus ownership changes. So the impact of rank subsetting (varying S) on system performance is determined by the interplay of these different factors, and also depends on the characteristics of the applications running on the system.

Furthermore, the number of ranks in a memory channel can significantly affect the memory system power. When R is large or the memory channel is not utilized frequently (BW_{RW} is small), static power dominates. The memory controller can then utilize the low-power modes of DRAM to decrease the static power (SP). This typically lowers BW_{RW} as well since it takes time for DRAM chips to enter or exit low-power modes, and commands to utilize low-power modes compete with other commands to DRAM.

It can be seen that rank subsetting changes both the energy efficiency and performance of computer systems. So we need a metric to measure the effectiveness of rank subsetting at the system level, combining both performance and energy efficiency instead of presenting memory system power and processor performance separately. We pick system energy-delay product (EDP).

2.3 Adding Reliability to Multicore DIMMs

Soft and hard errors occur frequently on modern DRAM chips. ECCs are widely used in computer systems requiring high reliability. These codes are typically a compromise between capacity overhead and resilience. We define capacity overhead to be the ratio of parity bits to codeword size. The most popular codes enable single-bit error correction and double-bit error detection (SECDED). A codeword is a set of data and parity bits. Typically, one or two DRAM chips are added to a rank to provide the parity bits.

There are two ways to support SECDED-level reliability on Multicore DIMMs. In a first alternative, one chip can be added per VMD. The energy efficiency of these ECC-enabled Multicore DIMMs is better than that of conventional DIMMs, since fewer chips are used per row access. However, this incurs higher capacity overhead than standard ECC DIMM solutions, except in the case of 2 VMDs having 9 $\times$4 DRAM chips each, or when each DRAM device can have a wider data path, such as $\times$9, instead of $\times$8. ($\times$4 and $\times$8 indicate that the data-path size of a DRAM chip is 4 and 8.) Though not popular, some DRAM vendors provide such configurations (RLDRAM [25] and RDRAM [30]).

SECDED schemes protect against single-bit errors, such as DRAM bit cell faults and wire stuck-at faults. High-availability systems often demand the stronger reliability of single-chip error correct, double-chip error detect (SCCDCD) schemes, sometimes called *chipkill*. These schemes can correct the failure of an entire DRAM chip, and detect the failure of two DRAM chips.

There are two common practices for implementing SCCDCD reliability in memory systems: interleaving SECDED-quality codewords across multiple ranks [6], as implemented by IBM's xSeries, or employing stronger error correcting codes [3], as found in the AMD Opteron. The first scheme observes that codewords can be interleaved across ranks such that no more than 1 bit comes from a single DRAM chip. For example, 4 ranks are sufficient to interleave bits from $\times$4 chips. In this case, four SECDED codes of 72-bits interleaved across 288-bits will suffice to recover from whole chip failures. While conceptually simple, this solution suffers from the fact that requiring 4 ranks of $\times$4 chips to be accessed in each memory transaction means that each access activates 72 individual DRAM chips, leading to poor dynamic energy efficiency. This sort of SCCDCD solution is also impractical with $\times$8 and $\times$16 chips, since they would re-

quire 8 and 16 ranks per transaction, respectively. It should be noted that DRAMs with long minimum burst lengths (like DDR3, which has a minimum burst length of 8) render SCCDCD schemes that interleave transactions across several ranks unreasonable, since they grow the memory transaction length beyond that of a typical cache line. For example, were the IBM chipkill solution to be implemented with ×4 DRAM chips with burst length 8, each memory transaction would be 2048 bits (256 bytes) long.

Alternative approaches observe that DRAM chip failures manifest as a burst of bit errors in codewords with a burst length the same as the width of the chip's data path (e.g. failure of an ×4 chip is a burst error of length 4). Codes smaller than naïve SECDED codes can be developed for SCCDCD utilizing this property. For example, the chipkill protection in the AMD Opteron only uses 2 ranks of ×4 chips to construct a 144-bit code [3]. Reiger [31] shows that in order to correct b bits of burst error (up to b consecutive bits have wrong data), at least $2b$ parity bits are necessary. To detect l additional bits of burst error, l additional parity bits are necessary. This result establishes that SCCDCD reliability requires at least 3 parity chips in addition to the nominal data chips for a memory transaction. The Opteron solution, with 4 parity chips, comes close to this theoretic lower bound. However, this result also means that the Opteron solution is only practical with ×4 chips, since 2 ranks of ×8 chips would only provide 2 parity chips. Despite the coding efficiency of the Opteron solution, it still activates 36 DRAM chips per transaction, leading to poor energy efficiency. Both the IBM and Opteron solutions achieve good capacity efficiency, sacrificing 1 in 9 chips (11%) of their total capacity to error coding.

Multicore DIMMs can be augmented with SCCDCD reliability by adding 3 chips per VMD. In the case of 2 VMDs ($S = 2$) with ×4 chips, this equates to 11 chips per VMD (8 for data and 3 for parity). Compared to the conventional chipkill solutions described earlier, this is a 22% increase in capacity (11 vs. 9). However, only 11 DRAM chips are activated per transaction instead of 36 (Opteron) or 72 (IBM), leading to substantial energy savings. The number of chips per transaction for several configurations of MCDIMMs is compared with that of conventional chipkill solutions in Table 1. While the Opteron and IBM chipkill solutions are only practical for ×4 chips, it is feasible to implement SCCDCD MCDIMMs across several potential configurations ranging from 2 to 8 VMDs and using either ×4, ×8, or ×16 chips. Each configuration represents a compromise between capacity overhead and energy efficiency. For example, when comparing SCCDCD with 2 VMDs of ×4 and ×8 chips, the ×4 configuration activates 11 chips per transaction while the ×8 configuration only activates 7, leading to improved energy efficiency. On the other hand, the ×4 configuration only incurs a 27% overhead, while the ×8 configuration incurs 42%.

We can apply Equation (1) and (2) to model the power consumption of a Multicore DIMM memory channel augmented with SCCDCD reliability. Here D is the number of DRAM chips activated to provide both data and parity bits, and R is the number of ranks that operate independently within a channel. For example, $R = 1$ on a memory channel with 2 ranks employing the chipkill protection scheme in the AMD Opteron since all 36 chips in both ranks operate in unison. When the total memory capacity (excluding parity bits) of a system stays constant, implementing SCCDCD-level reliability increases D, decreases R, and increases E_{RW} due to parity overhead. Dynamic power plays a bigger role in the memory system power due to these changes.

Practical implementations of the high-reliability techniques discussed in this section would need to take into account issues of form factor, module compatibility, and memory channel wire count into consideration. For instance, while it is easy to imagine a single physical module specification that could be shared between modules with 2, 4, or 8 rank subsets, SCCDCD protection presumes a distinct datapath width and part count between different numbers of rank subsets. First, we observe that module slots already have pins dedicated to ECC even though many systems use DIMMs that do not have ECC chips. Similarly, we expect that module standards for DIMMs which incorporate rank-subsetting would include some number of pins for ECC chips. Second, like the Opteron SCCDCD solution, the implementation of SCCDCD MCDIMMs need not match the theoretic lower bound for coding efficiency or capacity efficiency. For example, on an SCCDCD MCDIMM with 2 rank subsets and using ×4 or ×8 chips, SCCDCD reliability could be achieved by spreading codewords across both rank subsets. While counterproductive towards the goal of minimizing overfetch, this solution is still far more energy efficient than the conventional SCCDCD alternatives.

3. EXPERIMENTAL SETUP

To evaluate the impact of rank subsetting in the context of system-wide performance, energy efficiency, and reliability, we assume a system architecture consisting of multiple cores and memory channels, similar to a Niagara processor [16] (Figure 4). A CPU has 16 in-order cores and 4 threads per core. The cores run at 2GHz and process up to 1 instruction and 1 memory access per cycle. Each core has its own L1 instruction and data cache, while there is an L2 cache shared by each cluster of 4 cores. L2 caches are not shared between clusters. All caches have 64B cache lines. There is a crossbar between the 4 L2 caches and 4 memory controllers. A hierarchical MESI protocol is used for cache coherency and a reverse directory, similar to that implemented in the Niagara processor, is associated with each memory controller. A memory controller is connected to from 1 to 4 ranks and there are either 1, 2, 4, or 8 rank subsets per memory rank (S). When $S = 1$, cache line transfer time in a data bus is 4ns or 8 cycles. As S doubles, cache line transfer time doubles as well. The controller has 32-entry scheduling buffers and employs a closed memory access scheduling policy [32], where a DRAM page is closed (precharged) with an read/write auto-precharge command unless there are more pending requests to the same page of the same bank in the buffers. Note that this policy is more sophisticated than a naïve closed-page policy which always uses auto-precharge commands. An XOR-based [7] interleaving scheme is used to map memory addresses across memory controllers, ranks, and rank subsets pseudo-randomly in rank-subset page granularity. (A rank-subset page is the product of the number of DRAM devices per subset and the number of columns in a DRAM bank.)

A 32nm process technology based on ITRS projections is assumed for both processor and memory chips. We use CACTI 5 [34] to compute access time, cycle time, dynamic

	No coding	IBM	Opteron	$S = 2$			$S = 4$			$S = 8$	
Chip width	×4	×4	×4	×4	×8	×16	×4	×8	×16	×4	×8
Min TX (bytes)	64	256	128	32			16			8	
Chips/TX	16	72	36	11	7	5	7	5	4	5	4
Overhead	0%	11%	11%	27%	42%	60%	42%	60%	75%	60%	75%

Table 1: This table compares several configurations of SCCDCD MCDIMMs (reliability-enhanced MCDIMM) against conventional chipkill solutions [3, 6]. Min TX is the minimum memory transaction size assuming DDR3's burst length of 8. Chips/TX is the number of DRAM chips activated per memory transaction. The IBM chipkill solution can correct certain types of multi-chip errors while the Opteron and MCDIMM solutions provide protection against equivalent categories of single-chip errors.

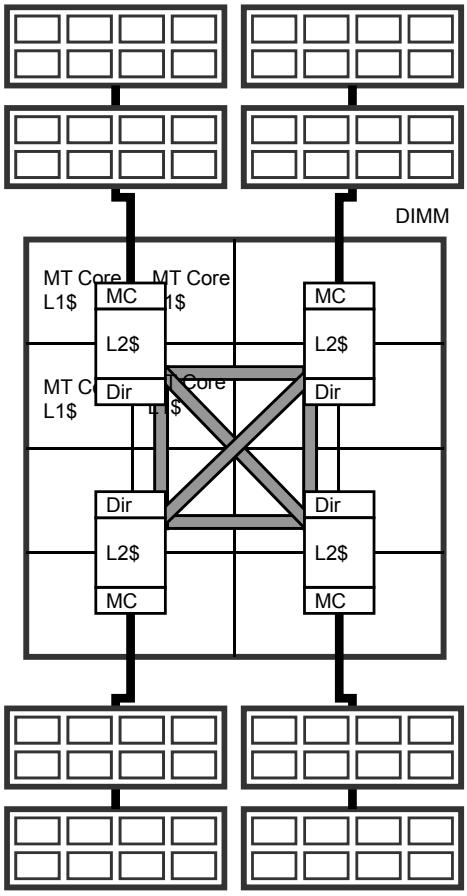


Figure 4: System architecture assumed in this paper. A processor consists of 16 cores, 4 threads per core, 16 L1I and L1D caches, 4 L2 caches, and 4 memory channels. Either one, two, or four ranks (i.e., two dual-rank DIMMs) are connected per memory channel.

energy, and static power of caches, directories, and DRAM chips, as summarized in Table 2. We compute the power of the processor chip by scaling the power of the Niagara processor [18] to a 32nm process. We estimate a peak power of 31W for 16 cores, where half is static power (including leakage), and the other half is dynamic power proportional to the instructions per cycle of the processor. DDR3 DRAM chips are assumed for the main memory, with a prefetch size of 8, and 2Gbps data pin bandwidth. The product of prefetch size and data-path size is the smallest possible number of bits accessed when a row is read or written. The energy dissipated by a data bus of a memory channel is calculated as the DC power of the output driver and termination resistance specified in the DDR3 standard [1] after scaling the operating voltage from 1.5V to 1.0V in order to account for a 32nm process. The energy consumption of the address and command bus is calculated by computing the capacitance of driver, wire, and load [9]. A memory controller puts a DRAM in a low-power state (similar to the precharge power-down mode in DDR3) when all banks in it are at the precharge state. We assume that a DRAM chip in a low-power state consumes 1/5th of normal static power and needs two cycles to enter and exit the state [23, 24].

We developed a multi-core simulation infrastructure in which a timing simulator and a functional simulator are decoupled in a way similar to GEMS [20]. A user-level thread library [28], which was developed as a Pin [19] binary instrumentation tool, is augmented to support additional pthread library APIs, such as `pthread_barriers`, and used as a functional simulator to run multi-threaded applications. An event-driven timing simulator, which models in-order cores, caches, directories, and memory channels, controls the flow of program execution in the functional simulator and effectively operates as a thread scheduler.

We perform experiments with the SPLASH-2 [36], PARSEC [5], and SPEC CPU2006 [22] benchmark suites. For multi-threaded workloads, 64 threads are spawned per workload and each thread is mapped to a hardware thread statically. All 11 SPLASH-2 applications are used while only 6 PARSEC applications (canneal, streamcluster, blackscholes, facesim, fluidanimate, and swaptions) are used (due to a Pin limitation). PARSEC applications are executed with the simlarge dataset while our SPLASH-2 inputs are summarized in Table 3. To model multiprogrammed workloads, we consolidate applications from SPEC CPU2006. The SPEC CPU2006 benchmark suite has single-threaded applications consisting of integer (CINT) and floating-point (CFP) benchmarks. We made 3 groups each of integer and floating-point benchmarks, 4 applications per group, based on their L2 cache miss ratio [10], which are listed in Table 3. It also shows the number of L2 misses per instruction, which is measured by using the baseline configuration in Section 4.1. Simpoint 3.0 [33] is used to find several simulation phases (100 million instructions per phase) and weights. For each CINT and CFP set, 16 simulation phases per application are consolidated so that each hardware thread executes a simulation phase. The number of instances per phase of each SPEC 2006 benchmark is proportional to its weight. We simulate each workload until the end or up to 2 billion instructions after skipping initialization phases.

Memory type	Capacity	Associativity	Access time	Cycle time	Dynamic read energy	Static power
L1 I cache	16KB ×16	4	1 cycle	1 cycle	0.091nJ	4.8mW
L1 D cache	32KB ×16	4	2 cycles	1 cycle	0.095nJ	8.9mW
L2 cache	1MB ×4	8	4 cycles	2 cycles	0.183nJ	185mW
Directory	×4	32	4 cycles	2 cycles	0.021nJ	86.5mW
×4 DRAM chip	4Gb	N/A	93 cycles	55 cycles	1.32nJ	75.6mW
×8 DRAM chip	8Gb	N/A	95 cycles	59 cycles	1.52nJ	104.8mW

Table 2: Power and performance parameters of the memory hierarchy used in this study. On DRAM chips, dynamic read energy includes precharge and activation energy assuming random access sequences. All caches use 64-byte blocks.

SPLASH-2

Application	Dataset	L2 miss per instr
Barnes	16K particles	0.0003
Cholesky	tk17.O	0.0030
FFT	1024K points	0.0051
FMM	16K particles	0.0006
LU	512×512 matrix	0.0004
Ocean	258×258 grids	0.0073
Radiosity	room	0.0003
Radix	8M integers	0.0186
Raytrace	car	0.0017
Volrend	head	0.0007
Water-Sp	512 molecules	0.0001

SPEC CPU2006

Set	Applications	L2 miss per instr
CFP		
high	433.milc, 450.soplex, 459.GemsFDTD, 470.lbm	0.0219
med	410.bwaves, 434.zeusmp, 437.leslie3d, 481.wrf	0.0099
low	436.cactusADM, 447.dealII, 454.calculix, 482.sphinx3	0.0073
CINT		
high	429.mcf, 462.libquantum, 471.omnetpp, 473.astar	0.0189
med	403.gcc, 445.gobmk, 464.h264ref, 483.xalancbmk	0.0046
low	400.perlbench, 401.bzip2, 456.hmmer, 458.sjeng	0.0037

Table 3: SPLASH-2 datasets and SPEC 2006 application mixes.

4. EVALUATION

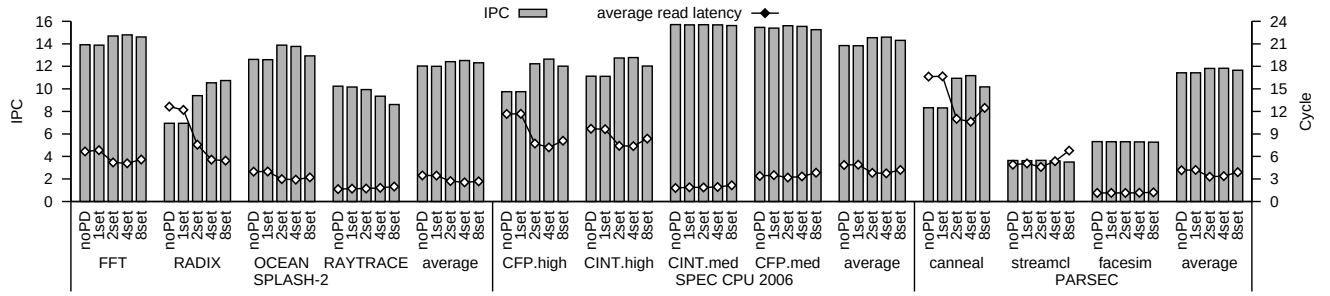
We evaluate the impact of rank subsetting on memory power, processor performance, and system energy-delay product using the configurations described in the previous section. In particular, we focus on understanding the interplay between subsetting DRAM ranks and utilizing DRAM power-down modes as the capacity and the reliability level of the memory systems are varied, which was not studied before.

4.1 Single-Rank Performance and Power Efficiency

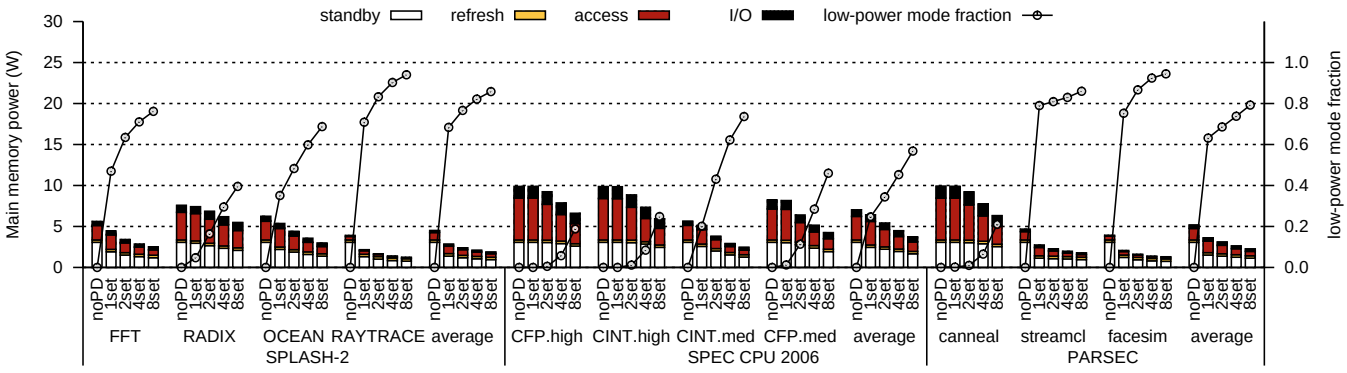
Figure 5 shows the performance and power result of 3 benchmark suites on a system with Multicore DIMMs where each memory controller has 1 memory rank ($R = 1$). There are five configurations on each application. The left-most configuration has one subset per memory rank and does not exploit the low-power mode of DRAM chips, which is the baseline configuration. Four remaining configurations have their memory controllers exploit the low-power mode and the number of rank subsets are varied from 1 to 8. For each suite, applications whose performance is not sensitive to the number of rank subsets are not shown due to space limitations, but they are included when average values are computed. Off-chip memory demands depend on the instructions per cycle (IPC), the number of memory requests per instructions, and the last-level cache miss rates. Figure 5(a) shows the IPC and the average read latency while Figure 5(b) shows the memory power breakdown of applications. Static power is divided into refresh and standby power, while dynamic power is divided into chip I/O power and access power within DRAM chips performing read, write, activate, and precharge operations. RADIX, CFP.high, CINT.high, and canneal are applications having high main memory bandwidth demand, which consume more DRAM dynamic power than other applications. The performance of these applications, which is closely correlated with their average read latency due to their high cache miss rate, strongly depends on the number of rank subsets per rank. Except for RADIX, the IPC increases until there are 2 or 4 subsets per rank and then decreases. As explained in Section 2, it is primarily due to the interaction of two factors: access latency and effective bandwidth of memory channels. For example, RADIX, an integer sorting application, takes advantage of higher memory bandwidth so that its performance keeps improving as S increases even when cache line transfers take 64 cycles in a data bus. By contrast, RAYTRACE doesn't stress the memory channel bandwidth but is very sensitive to access latency, so that its IPC drops as a memory rank is split into more subsets.[2] The energy-delay product improves substantially on applications with high main memory bandwidth demand, on average 39.7% among four applications when $S = 4$.

FFT, OCEAN, CINT.med, and CFP.med are applications with medium main memory bandwidth demand. The relationship between the IPC and the number of rank subsets is similar to that of the applications with high bandwidth demand but with smaller variation. However, compared to other applications, the fraction of time that DRAM chips stay in a low-power mode (which is shown in Figure 5(b)) substantially increases as the number of rank subsets in-

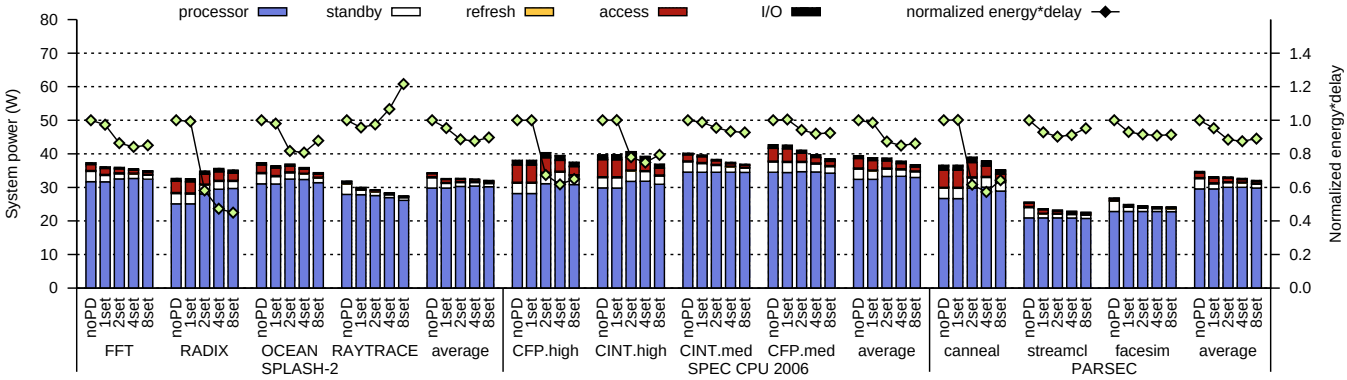
[2]The number of threads per core affects the application behavior over the number of subsets as well. When a smaller number of threads are used, more applications behave similar to RAYTRACE since it becomes harder to hide the increase in memory access latency due to rank subsetting.



(a) IPC and average read latency. The `noPD` configuration does not utilize DRAM power-down modes, and has 1 subset per rank. `nset` configurations utilize DRAM power-down modes, and have `n` subsets per rank.



(b) Memory power breakdown and mean fraction DRAM chips in a low-power mode.

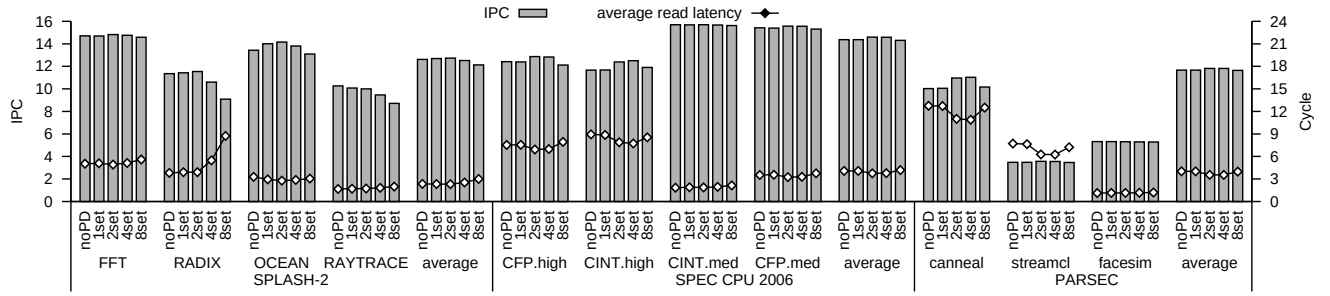


(c) System power breakdown and energy×delay (lower is better).

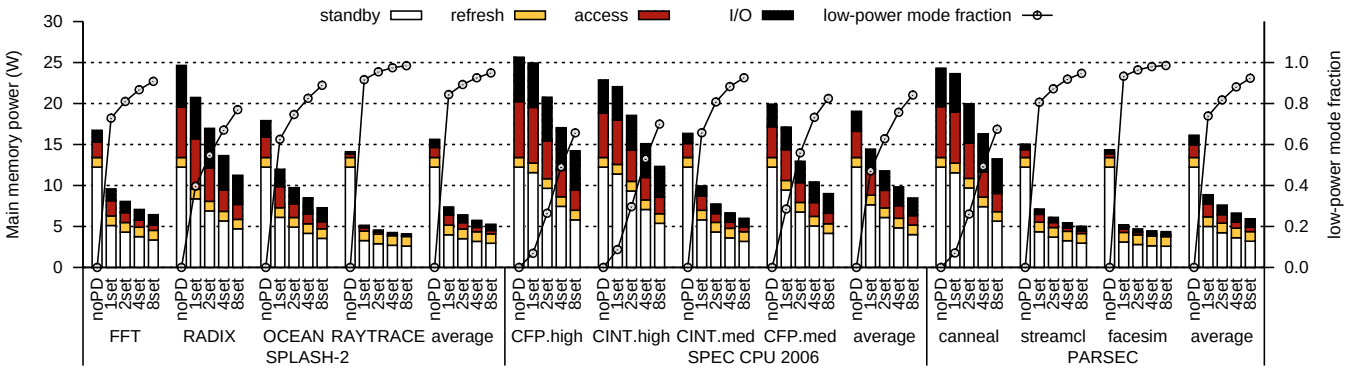
Figure 5: Memory and system level power and performance on a system with 1 rank per memory channel on 3 benchmark suites. For each suite, applications whose performance is not sensitive to the number of rank subsets are not shown due to space limitations, but they are included when average values are computed. Each rank consists of 8Gb ×8 DRAM chips.

creases. The chances that the DRAMs of a subset are in a low-power state increase with larger S, but on applications with low bandwidth demand, 1 subset per rank is already enough for the memory controller to put DRAMs in a low-power state most of the time. Conversely, applications with high bandwidth demand rarely leave rank subsets idle, regardless of the number of subsets, so the power-down mode is not often used.

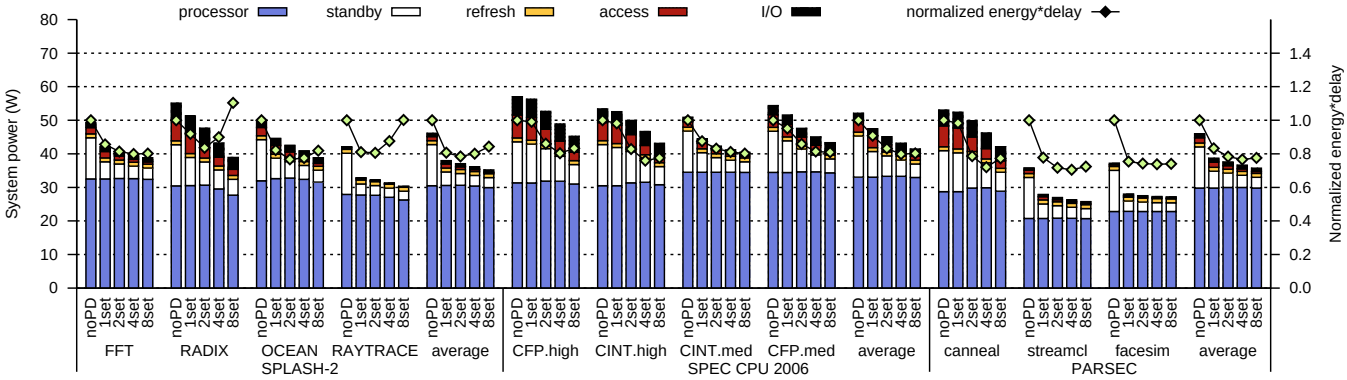
Regardless of memory demands of the application, there are substantial savings in dynamic energy from rank subsetting. However, since dynamic power is also proportional to the performance (IPC) of an application, this reduction in dynamic power is less apparent when its performance improves. Figure 5(c) shows the system power breakdown and the system energy-delay product of workloads. Across all the workloads, the energy-delay product is improved by 4.6%, 1.5%, and 4.7% on SPLASH-2, SPEC CPU 2006, and the PARSEC benchmarks by utilizing a DRAM power-down mode. Rank subsetting brings an additional 8.2%, 13.8%, and 8.2% improvement when $S = 4$. This shows that the effectiveness of the two techniques is complementary. The main memory power is always improved as the number of rank subsets increases. However, the system energy-delay product degrades on most applications when S increases from 4 to 8 since the performance is degraded more than the power saving. Finally, when memory channels are highly



(a) IPC and average read latency. The `noPD` configuration does not utilize a DRAM power-down mode, and has 1 subset per rank. `nset` configurations utilize a DRAM power-down mode, and have `n` subsets per rank.



(b) Memory power breakdown and mean fraction DRAM chips in a low-power mode.



(c) System power breakdown and energy×delay (lower is better).

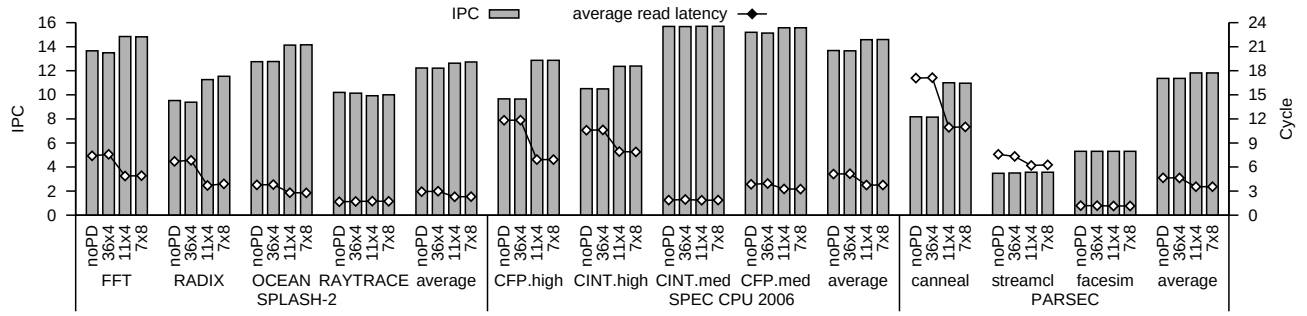
Figure 6: Memory and system level power and performance on a system with 4 ranks per memory channel on 3 benchmark suites. For each suite, applications whose performance is not sensitive to the number of rank subsets are not shown due to space limitations, but they are included when average values are computed. Each rank consists of 8Gb ×8 DRAM chips.

utilized, dynamic power is much larger than static power, and rank subsetting provides more improvement on system energy-delay product than DRAM power-down modes.

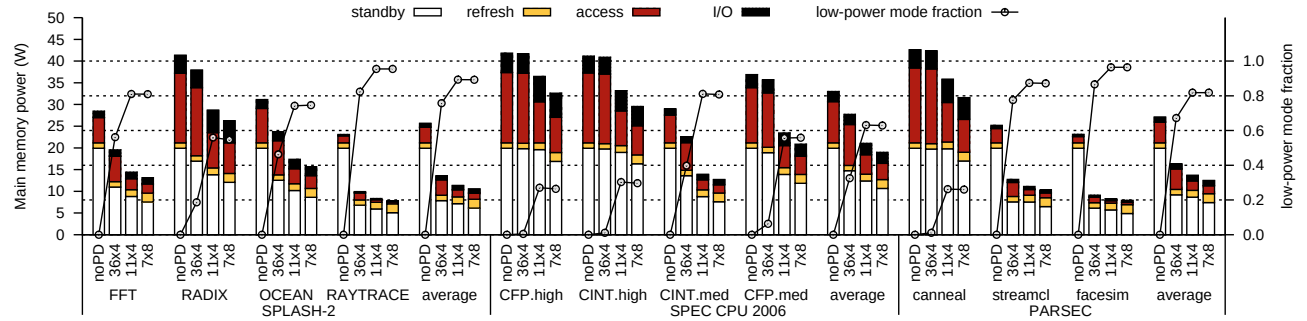
4.2 Four-Rank Performance and Power Efficiency

The relationship between application performance, memory power, and system energy-delay product changes when more ranks are attached per memory channel (i.e., for higher main memory capacity). Figure 6 shows the power and performance of the 3 benchmarks on a system with 2 dual-rank DIMMs per channel ($R = 4$). The increase in the IPC from 1 to 2 rank subsets on applications with high memory bandwidth demand is not as much as in the previous system configuration. As analyzed in Section 2, with 4 times more independent DRAM banks per channel, the activate-to-activate time constraint becomes a smaller problem as a memory controller can issue commands to other ranks, leading to high performance even without rank subsetting. Still, 2 rank subsets perform better than 1, since the timing constraints on each switch of bus ownership limit performance, and this is alleviated with multiple subsets as each DRAM transaction takes longer.

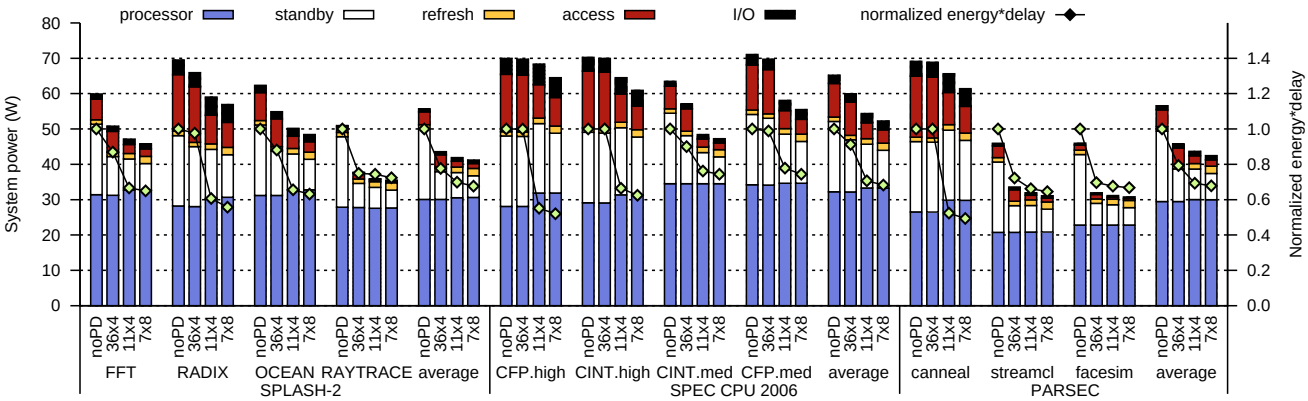
With 4 ranks per channel, static memory power (such as standby and refresh power) and I/O power increase substantially, becoming a significant part of the total memory



(a) IPC and average read latency. The `noPD` configuration does not utilize DRAM power-down modes, and has 1 subset per rank. `nset` configurations utilize DRAM power-down modes, and have `n` subsets per rank.



(b) Memory power breakdown and mean fraction DRAM chips in a low-power mode.



(c) System power breakdown and energy×delay (lower is better).

Figure 7: Power and performance of applications on systems with chipkill-level reliability. There are three configurations per application: `36x4` for the conventional system with 36 ×4 4Gb DRAM chips per rank, `11x4` with 11 ×4 4Gb DRAM chips per MCDIMM rank, and `7x8` with 7 ×8 8Gb DRAM chips per MCDIMM rank. For each suite, applications whose performance is not sensitive to the number of rank subsets are not shown due to space limitations, but they are included when average values are computed.

power as shown in Figure 6(b). Since the peak bandwidth per channel is the same as with 1 rank per channel, banks are idle more often, hence it is more likely that the memory controller can exploit low-power modes. I/O power increases since there are more termination resistors per data bus, and sometimes even surpasses the access power within DRAM chips, highlighting the need for more energy-efficient technologies such as differential signaling or point-to-point connections. The total memory power becomes comparable to the processor power on applications with high memory demand (Figure 6(c)). However, since the performance of these applications varies less than before as the number of rank subsets changes, the energy-delay product improves less as multiple subsets are used: 0.8% on SPLASH-2 with 2, 12.1% on SPEC CPU 2006 with 4, and 7.9% on PARSEC with 4 rank subsets all compared to the configuration utilizing a low-power mode but no rank subsetting. Rather, there are bigger savings by utilizing DRAM low-power modes even without rank subsetting: 19.3% on SPLASH-2 and 16.7% on PARSEC. The SPEC CPU 2006 benchmarks access main memory more often than others, so the 9.2% improvement in energy-delay product from putting DRAMs in a low-power mode is less than the additional improvement due to the rank subsets.

4.3 Power and Performance of Chipkill-level Reliability

Figure 7 shows the performance and energy efficiency of 4 different memory systems supporting chipkill level reliability. On each application, the first two columns (`noPD` and `36x4`) have the values for a conventional chipkill solution, in which each rank consists of 36 ×4 4Gb DRAM chips (32 chips for data and 4 for parity and other information). The first column does not utilize the DRAM low-power mode while the second column does. Since each DRAM chip has a prefetch length of 8, a minimum transfer size $36 \times 8 \times 4 =$ 1152bits of data should be read or written. The cache line size of the system is 72B = 576bits including ECC in internal caches, so half of the data are not used. Although burst chopping [24] can be used to save I/O power by not transferring unused data, substantial DRAM dynamic energy is still wasted. In the second memory system, denoted `11x4`, each Multicore DIMM rank consists of 2 subsets, each with 11 ×4 4Gb DRAM chips. As a result, 2/9 = 22.2% more DRAM chips are used over a system that only provides parity or ECC, but only 11 chips, less than 1/3 of DRAM chips compared to the conventional chipkill DIMMs, are used per memory access. The third system, denoted `7x8`, has 2 Multicore DIMM VMDs (subsets) per rank, each with 7 ×8 8Gb DRAM chips. Three DRAM chips are used for error correction per subset, but it needs fewer than 1/5th as many DRAM chips per access compared to `36x4`. All configurations have the same data capacity. The `36x4` configurations have 2 ranks per channel, and the last two have 4 ranks (two dual-rank DIMMs) per channel and 2 subsets per rank.

In traditional chipkill systems, memory power can surpass the processor power, emphasizing the importance of improving energy efficiency of processor-memory interfaces. The `36x4` configuration clearly performs worse than others since its effective per-rank bandwidth is lower while the `11x4` and `7x8` configurations obtain benefits from having multiple rank subsets—more banks and less frequent timing constraint conflicts, as shown in Section 4.2. There are major savings in DRAM dynamic power on configurations with high-reliability MCDIMMs (Figure 7(b)) since far fewer DRAM chips are used per access. This also helps memory controllers to idle subsets so that the static power of MCDIMMs can be even lower than conventional chipkill DIMMs unless the whole memory system is largely idle like on RAYTRACE, streamcl, and facesim. This is true even though systems with high-reliability MCDIMMs have more DRAM chips than `36x4`. Therefore both subsetting DRAM ranks and exploiting DRAM low-power modes are equally important to enhancing energy-delay product. By utilizing DRAM low-power modes, system energy delay product on `36x4` is improved by 18.4%. Rank subsetting leads to additional improvement on system energy delay product: 14.3% on `11x4` and 16.9% on `7x8` compared to the `36x4` with a low-power mode utilized.

5. RELATED WORK

A large body of computer architecture research aims to improve the performance, energy efficiency, and reliability of main memory systems. One key idea is to group together memory accesses with the same access type and similar addresses by reordering, in order to minimize the performance degradation due to various timing constraints on DRAM accesses. There have been proposals to exploit these characteristics to achieve higher performance on vector [21], stream [32], and single-core and multicore processors [26,27]. Higher performance typically leads to higher energy efficiency by reducing execution time and saving static power. We use these techniques in the evaluations throughout this paper.

Multiple power states were introduced in Rambus DRAM (RDRAM [30]). Some studies try to exploit these low power states in RDRAMs by allocating and migrating OS pages in order to put DRAM chips into a low power state for longer periods [12,17]. Modern DDRx DRAM chips also have multiple low power states. Hur et al. suggest ways to exploit them in a memory scheduler [13]. Ghosh and Lee suggest a memory controller design with smart refresh [9] to save refresh power. This is complementary to our ideas as well.

Among the proposals advocating rank subsetting, module threading [35] relies on high speed signaling. The memory controller outputs separate chip select signals for selecting a subset of devices. Multicore DIMM [2] replaces a register per memory rank with a demux register, which routes or demultiplexes address and command signals to the selected subset. Zheng et al. called a subset a mini-rank [37] and proposed a design in which all mini-ranks in a memory rank send and receive data to/from a memory controller through a mini-rank Buffer. They did not consider CPU power and reliability in their evaluation of their architecture. The key difference between Multicore DIMM and mini-rank is the placement of the data mux and address/command demux. Mini-rank has a demux per memory rank while Multicore DIMM has one per memory channel. As a result, mini-rank is more costly in energy and component count. Multicore DIMM has one address/command demux per memory rank, while mini-rank does not have any. Since address and command signals must be registered per rank anyway due to signal integrity issues, the incremental cost of the Multicore DIMM demux register is minimal. Both proposals need chip select signals per rank subset.

6. CONCLUSION

Memory power is becoming an increasingly significant percentage of system power as the number of cores per processor increases. This paper is the first to holistically assess the effectiveness of rank subsetting on the performance, energy efficiency, and reliability at the system level rather than only looking at the impact on individual components. This paper is also the first to quantify the interactions between DRAM power-down modes and rank subsetting. For a single-rank memory system, across the SPLASH-2, SPEC CPU 2006, and PARSEC benchmarks, we found that power-down modes without rank subsetting could save an average of 3.6% in system energy-delay product. When rank subsetting is added, an additional average savings of 10.1% is obtained. The cost of rank subsetting is very low: for example in the Multicore DIMM approach a latch on each DIMM is converted to a demux latch. Thus the 10.1% *system* energy-delay product savings is a remarkably high return given the insignificant investment required.

Rank subsetting increases the amount of data read out of a single chip, so it also can increase the probability of a large number of bit errors when an entire chip fails. In this paper we also extended the MCDIMM design for high-reliability

systems. Enhancing reliability involves a tradeoff between energy and storage efficiency. Traditional chipkill solutions optimize storage efficiency at the cost of reduced energy efficiency. With the cost of powering datacenters now exceeding their capital cost over their lifetimes, solutions which strike a balance between capacity and energy efficiency make more sense.

In summary, we expect rank subsetting, especially reliability-enhanced Multicore DIMM, to be compelling alternatives to existing processor-memory interfaces for future DDR systems due to their superior energy efficiency, tolerance for DRAM timing constraints, similar or better system performance, and ability to provide high reliability.

7. REFERENCES

[1] "Calculating Memory System Power for DDR3," Micron, Tech. Rep. TN-41-01, 2007.

[2] J. Ahn, J. Leverich, R. S. Schreiber, and N. P. Jouppi, "Multicore DIMM: an Energy Efficient Memory Module with Independently Controlled DRAMs," *Computer Architecture Letters*, vol. 7, no. 1, 2008.

[3] AMD, *BIOS and Kernel Developer's Guide for AMD NPT Family 0Fh Processors*, Jul 2007, http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/32559.pdf.

[4] L. A. Barroso, "The price of performance," *Queue*, vol. 3, no. 7, pp. 48–53, 2005.

[5] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC Benchmark Suite: Characterization and Architectural Implications," in *PACT*, Oct 2008.

[6] T. J. Dell, "A White Paper on the Benefits of Chipkill-Correct ECC for PC Server Main Memory," *IBM Microelectronics Division, Nov. 1997.*

[7] J. M. Frailong, W. Jalby, and J. Lenfant, "XOR-Schemes: A Flexible Data Organization in Parallel Memories," in *ICPP*, Aug 1985.

[8] J. Gee, M. D. Hill, D. N. Pnevmatikatos, and A. J. Smith, "Cache Performance of the SPEC92 Benchmark Suite," *IEEE Micro*, vol. 13, 1993.

[9] M. Ghosh and H.-H. S. Lee, "Smart Refresh: An Enhanced Memory Controller Design for Reducing Energy in Conventional and 3D Die-Stacked DRAMs," in *MICRO*, Dec 2007.

[10] J. L. Henning, "Performance counters and development of SPEC CPU2006," *Computer Architecture News*, vol. 35, no. 1, 2007.

[11] R. Ho, K. Mai, and M. A. Horowitz, "The Future of Wires," *Proceedings of the IEEE*, vol. 89, no. 4, 2001.

[12] H. Huang, P. Pillai, and K. G. Shin, "Design and Implementation of Power-Aware Virtual Memory," in *USENIX*, Jun 2003.

[13] I. Hur and C. Lin, "A Comprehensive Approach to DRAM Power Management," in *HPCA*, Feb 2008.

[14] Intel, "Intel® Core™ i7 Processor, http://www.intel.com/products/processor/corei7/."

[15] B. Jacob, S. W. Ng, and D. T. Wang, *Memory Systems: Cache, DRAM, Disk*. Morgan Kaufmann, 2007.

[16] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-Way Multithreaded Sparc Processor," *IEEE Micro*, vol. 25, no. 2, 2005.

[17] A. R. Lebeck, X. Fan, H. Zeng, and C. Ellis, "Power Aware Page Allocation," in *ASPLOS*, Nov 2000.

[18] A. Leon, B. Langley, and J. L. Shin, "The UltraSPARC T1 Processor: CMT Reliability," *CICC*, Sep 2006.

[19] C.-K. Luk, *et al.*, "Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation," in *PLDI*, Jun 2005.

[20] M. M. K. Martin, *et al.*, "Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset," *SIGARCH Computer Architecture News*, vol. 33, no. 4, 2005.

[21] B. K. Mathew, S. A. Mckee, J. B. Carter, and A. Davis, "Design of a Parallel Vector Access Unit for SDRAM Memory Systems," in *HPCA*, Jan 2000.

[22] H. McGhan, "SPEC CPU2006 Benchmark Suite," in *Microprocessor Report*, Oct 2006.

[23] Micron Technology Inc., *DDR2 SDRAM Datasheet*, 2008, http://www.micron.com/products/dram/ddr2/.

[24] ——, *DDR3 SDRAM Datasheet*, 2008, http://www.micron.com/products/dram/ddr3/.

[25] ——, *RLDRAM Datasheet*, 2008, http://www.micron.com/products/dram/rldram/.

[26] O. Mutlu and T. Moscibroda, "Stall-Time Fair Memory Access Scheduling for Chip Multiprocessors," in *MICRO*, Dec 2007.

[27] K. J. Nesbit, N. Aggarwal, J. Laudon, and J. E. Smith, "Fair Queuing Memory Systems," in *MICRO*, Dec 2006.

[28] H. Pan, K. Asanović, R. Cohn, and C.-K. Luk, "Controlling Program Execution through Binary Instrumentation," *SIGARCH Computer Architecture News*, vol. 33, no. 5, 2005.

[29] W. W. Peterson and J. E. J. Weldon, *Error-Correctin Codes*, 2nd ed. MIT Press, 1972.

[30] Rambus, "RDRAM, http://www.rambus.com," 1999.

[31] S. H. Reiger, "Codes for the Correction of "Clustered" Errors," in *IRE Transactions on Information Theory*, 1960.

[32] S. Rixner, W. J. Dally, U. J. Kapasi, P. R. Mattson, and J. D. Owens, "Memory Access Scheduling," in *ISCA*, Jun 2000.

[33] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder, "Automatically Characterizing Large Scale Program Behavior," in *ASPLOS*, Oct 2002.

[34] S. Thoziyoor, J. Ahn, M. Monchiero, J. B. Brockman, and N. P. Jouppi, "A Comprehensive Memory Modeling Tool and its Application to the Design and Analysis of Future Memory Hierarchies," in *ISCA*, Jun 2008.

[35] F. A. Ware and C. Hampel, "Improving Power and Data Efficiency with Threaded Memory Modules," in *ICCD*, Oct 2006.

[36] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," in *ISCA*, Jun 1995.

[37] H. Zheng, J. Lin, Z. Zhang, E. Gorbatov, H. David, and Z. Zhu, "Mini-Rank: Adaptive DRAM Architecture for Improving Memory Power Efficiency," in *Micro*, Nov 2008.