Implications of Shallower Memory Controller Transaction Queues in Scalable Memory Systems

Mario D. Marino*, Kuan-Ching Li† *Independent Researcher, Italy, mario.dmarino@gmail.com
†Corresponding Author, Providence University, kuancli@gm.pu.edu.tw

Abstract

Scalable memory systems provide scalable bandwidth to the core growth demands in multicores and embedded systems processors. In these systems, as memory controllers (MCs) are scaled, memory traffic per MC is reduced, so transaction queues become shallower. As a consequence, there is an opportunity to explore transaction queue utilization and its impact on energy utilization. In this paper, we propose to evaluate the performance and energy-per-bit impact when reducing transaction queue sizes along with the MCs of these systems. Experimental results show that reducing 50% on the number of entries, bandwidth and energy-per-bit levels are not affected, whilst reducing aggressively of about 90%, bandwidth is similarly reduced while causing significantly higher energy-per-bit utilization.

Keywords: memory, controller, transaction, scalable, RF, optical

I. Introduction

The higher degrees of memory contention achieved when increasing number of cores integrated in a single chip have critical performance factor, therefore scalable memory systems through high-performance and low-energy design strategies aim efficient communication to achieve optimum bandwidth/latency and low power.

Traditional double data rate (DDR) memory design to achieve larger memory bandwidth is based on the application of the pair clock frequency - simply memory frequency - and memory width on the memory ranks - formed by set of memory banks with data output aggregated and sharing addresses.

DDR family generations have been utilizing larger clock frequencies to improve its bandwidth. For example, according to [1] a factor of 10x larger clock frequencies have been applied in DDR memory families. However, since memory power usage is proportional to frequency, the traditional focus on memory design has switched from frequency scaling to memory width scalability.

The straightforward method to increase memory parallelism represented by memory width is via scaling of MCs - MC scalability. For example, in traditional multicores processors and embedded ones, Wide I/O 2 [2] commercial solution presents up to 8 MCs, each one connected to a rank width of 128 bits, performing a maximum total width of 1024 bits, while HyperMemory Cube (HMC) up to 8 MCs/ranks with individual rank width of 55bits (total of 440bits, I/O bit rate of 10Gbits/s).

Comparatively to these mentioned solutions, advanced memory interfaces explore significant larger number of MCs - or MC counts. These advanced interfaces rely on optical- and radio-frequency (RF) technologies, which allow them to use fewer I/O pins, allowing further degrees of MC scalability. For example, Corona [3] utilizes only

two optical pins and DIMM Tree employs 39 RFpins [4], which are significantly less than 240 I/O pins present in traditional DDR families.

As a consequence, Corona [3] is able to scale to 64 optical-MCs while DIMM Tree [4] up to 64 RFMCs (RF-based memory controllers). In previous advanced systems, total memory width is estimated about 4096 bits when interfaced to simple double data rate (DDR) memories.

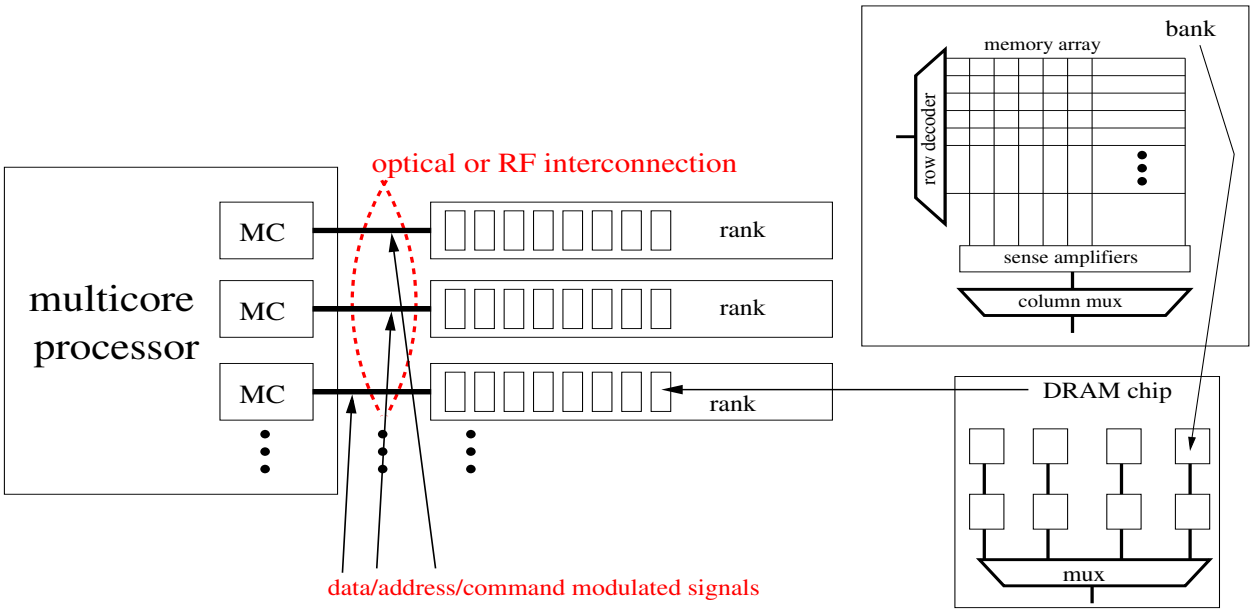

Fig. 1: memory system general overview

In above mentioned memory solutions, as MCs are scaled, the number of transaction queue elements present in each MC is scaled as well. As noticed in [5], as MCs are scaled the amount of memory traffic per channel is reduced, i.e., transaction queue utilization is reduced. We create a model based on these advanced memory interface technologies, and assess it with different transaction queue sizes and different MC-counts using detailed and accurate simulation tools combined with memory bandwidth-bound benchmarks. We envision the opportunity to explore shallower transaction queues aspect in terms of bandwidth and power in order to advance the state of art in scalable memory systems with the following contributions to determine:

- Bandwidth impact when employing shallower transaction queues under several workload conditions,
- Total rank and MC power impact when utilizing the reduced-size queues,
- Conditions to dynamically explore shallower transaction queues.

This paper is organized as follows. Section II presents the background while shallower transaction queues are depicted in Section III. Next, it is discussed in Section IV the experimental results obtained, Section V the related work, and finally, in Section VI the concluding remarks as well as future plans.

II. Background and Motivation

We describe the background of scalable memory systems and the motivations for shallower transaction queues.

A. Background

According to the report [6], a typical MC is composed of (a) front engine (FE), which processes L2 cache requests; (b) transaction engine (TE), which transforms these requests into control and data commands to be sent to the ranks; and (c) physical transmission (PHY), composed by control / data physical channels.

In particular, along its optical or RF memory interfaces, modulation and demodulation of commands, data, clock, and addresses are performed while executing typical read/write memory operations. Along these interfaces, signals are transmitted over the optical/RF interconnection between the optical-MC/RFMC and rank. In addition, while command, clock (CK), and address signals are demodulated at the ranks, these also modulate data to be returned to the MC, when a read operation is performed. Figure 1 illustrates the context where the memory path is utilized.

To illustrate the need for more MCs, we show Figure 2 [1] to illustrate the behavior of rank bandwidth along to different low power memory generations. We observe in this figure that total rank bandwidth is still restricted in terms of magnitude which motivates the need of approaching bandwidth via MC scalability.

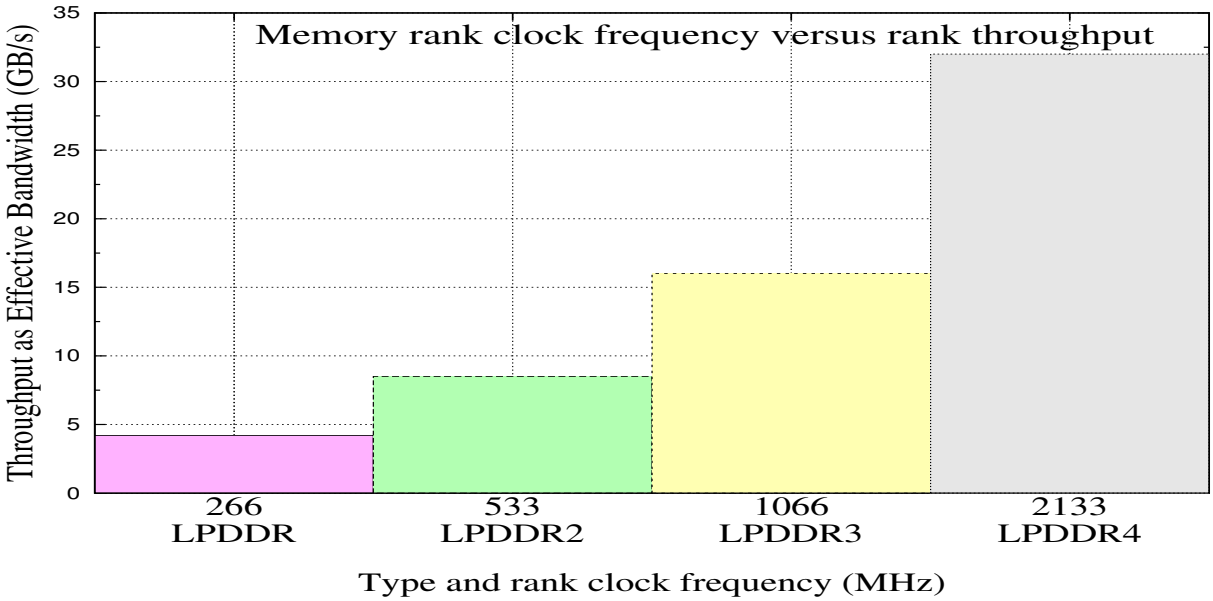

Fig. 2: frequency versus bandwidth, repeated from [1]

Furthermore, recent-developed commercial memory solutions still employ larger number of pins, which can still can restrict MC scalability, i.e., memory width. For instance, Hybrid Memory Cube [7] employs 55 pins and can utilize up to 8 MCs. The maximum aggregated bandwidth in HMC is 320 GB/s while each I/O-link presents individually 10 Gbit/s. Furthermore, Wide I/O 2 [2] employs 128 bits per rank and 8 MCs, thus still MC-count restricted (total width 1024 bits).

Compared to these solutions, advanced solutions that employ shorter amounts of pins present better MC scalability. As previously mentioned, optical Corona [3] presents 2 optical pins only and 64 optical MCs. Moreover, given its similarity to 2.5D integration on silicon interposer, RFiop [8] illustrated in Figure 3a originally designed for 16 RFMCs and 96GB/s (using 6GB/s, 64bit-ranks, total of 1024 bits) at 32nm, if scaled to 22nm, it is likely to achieve 32 RFMCs/ranks and 480GB/s, i.e., significatively larger number of MCs and bandwidth.

Another example of advanced solution is RFiof [5] illustrated in Figure 3b, which presents similar performance benefits of optical-based interfaces. RFiof is designed to scale to 32 RFMCs and 345.6 GB/s, using 10.8GB/s ranks; however given its lower number of pins and the adoption of a conventional RF-interface (FR-board as in DIMM Tree [4]), this technology has the potentiallity to be scaled to use 64 RFMCs and ranks of 17.2 GB/s and likely achieve the bandwidth of 1024GB/s (and total width of 4096 bits), which is an expressive bandwidth level.

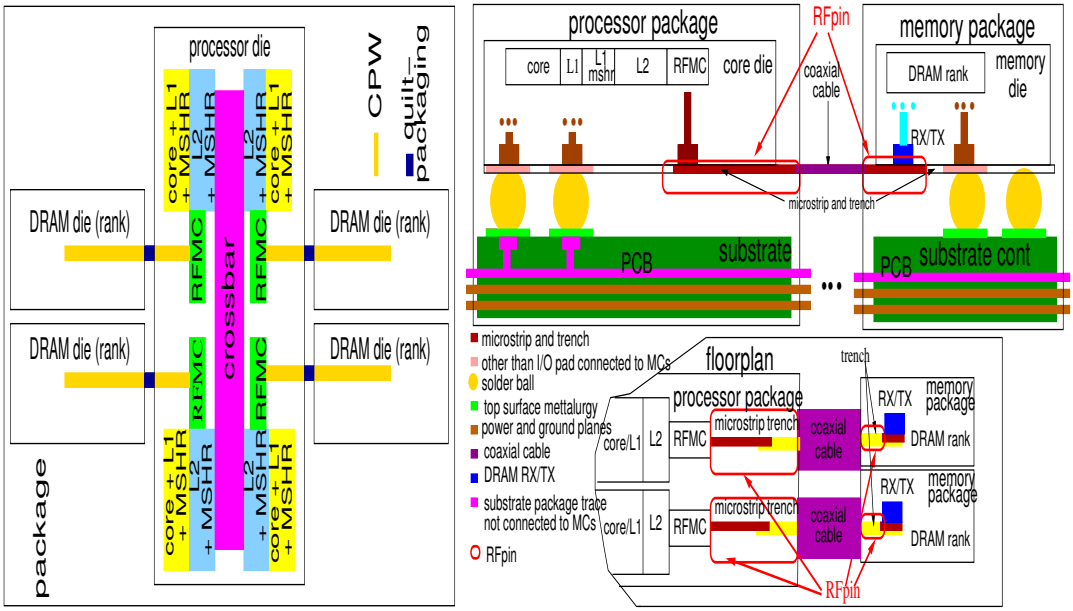

Fig. 3: a and b: left to right, (a) RFiop (replicated from [8], where quilt-packaging [9] is a coplanar waveguide - CPW ; (b) RFiof (replicated from [5])

B. Motivation

Each MC has a transaction queue, which has a certain number of entries utilized to store cache requests in each one. As applications run, these transaction queue entries are filled according to their memory-bound behavior. Applications with a more intense memory-bound behavior are likely to have more entries filled, while applications with a lesser intense memory behavior are likely to utilize less entries.

In scalable memory systems [3][5][8] MCs are scaled to improve bandwidth and decrease power. Simply put, in these systems, the increase of the memory width - memory parallelism - via having multiple MCs and multiple ranks allows the increase the number of simultaneous memory transactions treated in the system.

Since there is more memory parallelism, transaction queue entries - that hold memory requests - are less utilized. Therefore, all memory entries in the transaction queus - limited by transaction queue maximum size - are not fully filled in these conditions.

Furthermore, even for intense memory applications transaction memory queues are less utilized. For example, in RFiop [8] scalable memory system it is observed that, for the intense memory applications STREAM [10] and pChase [11], the number of memory transactions entries occupied is reduced of up to 5x when scaling 2 MCs to 16MCs.

To conclude, in all previously mentioned cases, transaction queue usage entries are not completely occupied with cache requests. Therefore, transaction queue entries are sub-utilized.

III. Transaction Queue Size Reduction

The occupation of the TE transaction queue with memory requests depends on the following factors:

- the ratio between rank clock frequency and processor clock frequency: the higher the ratio is, the higher the time transaction queue is occupied. In addition, the higher the latency is, the lower the bandwidth is. Given that each cache request is transformed to physical memory request at the MC, the higher the ratio MC clock frequency to processor clock frequency is, the higher the time the cache transaction takes to be transformed into memory request, also the higher the latency is.
- memory-bound behavior of the benchmark being executed: the higher the memory-bound behavior of the benchmark executed is, the higher the ratio of memory instructions per cycle (or over a thousand of instructions or misses per kilo-instructions - MPKI) is achieved.

Assuming a typical interleaving of addresses over different last level cache (LLC) unit, MC scaling implies on memory width scaling. Therefore, as the number of simultaneous number of memory transactions on the memory system is increased, it may lead to larger bandwidth.

In this context, the total memory traffic is increased due to larger width, and this behavior may be expressed qualitatively as follows:

$$Bw = Rw \ * \ freq \tag{1}$$

where Bw means total rank bandwidth, Rw the total rank width, and $freq$ the frequency with which ranks are set. Therefore, as MCs are scaled with ranks, total rank bandwidth is similarly scaled.

Given that memory transactions are spread over larger number of MCs, memory traffic per MC is decreased, qualitatively derived to:

$$MtpMC = \sum_{i=1}^{MCs} \frac{TT}{MCs} \quad (2)$$

where $MtpMC$ means memory traffic per MC, TT the total traffic. It is immediate to derive that as MCs are increased, $MtMC$ is increased.

This behavior can be exemplified on a scalable MC system, e.g., RFiop [8] in which each MC TE transaction queue presents 32 entries to store memory requests. From bandwidth-bound benchmark experimental results, RFiop scalable memory system with a 32-element transaction queue size only utilizes 25% of each of its transaction queues. Therefore, given the lower utilization it is possible to reduce transaction queue sizes without having performance ($MtpMC$ and Bw) degradation.

A straightforward way to mitigate the transaction queue sub-utilization is to employ shallower transaction queues. In this case, under the same amount of traffic, the ratio between the total number of utilized entries and the total queue size is increased, i.e., a better utilization is obtained.

It is important to note that while having a better utilization, a transaction queue with a reduced number of elements spends a smaller amount of power. By reducing the number of elements utilized, correspondent circuit area is smaller, thus allowing to save power.

IV. Experimental Results

In this section we perform a series of experiments to demonstrate the power and performance effects due to the reduction of transaction queue sizes in scalable memory systems.

A. Methodology

To have a global picture of the methodology employed in this study, we have listed all the simulators employed and the description of their purpose in Table I. The general methodology employed to obtain bandwidth is adopted from [20]: by using bandwidth-bound benchmarks to stress the memory system, we combine M5 [15] and DRAMsim [13] simulators as follows.

Before describing the experiments, we observe that the baseline for the experiments presents 32 cores and 32 MCs to maintain the ratio core:MC the same (32:32) such as in [5].

In order to evaluate this scalable memory system, we combine detailed accurate simulators using the methodology developed in [20]: we combine the creation of a 32-multicore model in M5 [15], which upon benchmark execution of a multicore model generates memory transactions which are captured by DRAMsim [13] which is properly configured with 32 RFMCs so that core:MC ratio is 32:32. In the sequence, DRAMsim responds to M5 with the result of each memory transaction.

tool	description
Cacti [12]	cache latencies configured with
McPAT [6]	determine power of individual path elements: TE and FE
DRAMsim [13]	Capture memory transactions from M5 configured with 32 RFMCs. Respond to M5 with the result of the memory transaction. Determine power spent [13][14]. Determine the number of memory accesses and transaction queue size.
M5 [15]	Configured as 32-core OOO processor and not L2 shared cluster (avoid sharing). Generates memory transactions which are passed to DRAMsim [13]. Miss-status handling register (MSHR) counts from typical microprocessors [16].
RF-crossbar	Implemented in M5 [15] with RF settings from [17][18].
RF-communication delays	RF-circuitry modeling and scaling [17][19].

TABLE I: methodology: tools and description

The baseline configuration is the one with 32 RFMCs and 1-to-16-element queue elements. We employ a 4.0-GHz (Alpha ISA) and 4-wide out-of-order (OOO) core, while having RFMCs at 2.0GHz (typically at half of microprocessor clock frequency [21]). We use Cacti [12] to obtain cache latencies and adopted MSHR counts of typical microprocessors [16]. We employ 1 MB/core L2 caches, which are interconnected via an 80GB/s-RF-crossbar (magnitude set in order to not restrict total throughput) with 1-cycle latency (adopting same timing settings of [17][18]: 200ps of TX-RX delays, plus the rest of the cycle to transfer 64 Bytes using high speed and modulation).

In order to be general, we have selected a DDR3-rank. Observing the RF-crossbar upper constraint, the selected rank is a medium data-rate DDR3-rank employed in typical PCs and smartphones/pads (64 data bits, based on the DDR3 model Micron MT41K128M8 of 1GB [14], and listed in Table IIa).

Each RFMC is assumed to be connected to one rank in order to extract its maximum bandwidth. In order to not take advantage of locality, we have employed a conservative addressing by interleaving cache lines along the RFMCs, as well as closed page mode since, as reported in [22], this mode benefits performance and energy utilization in multicores. To finalize, all architectural parameters are summarized in Table IIa.

To determine the total energy-per-bit spent, we employ DRAMsim power infrastructure and combine them with

Core	4.0 GHz, OOO, multicore, 32 cores, 4-wide issue, turnament branch predictor
Technology	22 nm
L1 cache	32kB dcache + 32 kB icache; associativity = 2 MSHR = 8, latency = 0.25 ns
L2 cache	1MB/per core ; associativity = 8 MSHR = 16; latency = 2.5 ns
RF-crossbar	latency = 1 cycle, 80GB/s
RFMC trans. queue	32 RFMCs; 1 RFMC/core, 2.0GHz, on-chip entries = 16/MC, close page mode
Memory rank	DDR3 1333MT/s, 1 rank/MC, 1GB, 8 banks, 16384 rows, 1024 columns, 64 bits, Micron MT41K128M8 [14], tras=26.7cycles, tcas=trcd=8cycles
RF interconnection length size delay	 2.5 cm 0.185ns

Benchmark	Input Size	read : write	MPKI
Copy, Add, Scale, Triad (STREAM)	4Mdoubles per core 2 interations	2.54:1	54.3
pChase	64MB/thread, 3 iterations, random	158:1	116.7
Multigrid:MG (NPB)	Class B 2 iterations	76:1	16.9
Scalar Pentadiagonal: SP (NPB)	Class B 2 iterations	1.9:1 1.9:1	11.1 11.1
Hotspot,	6000 x 6000, 3 iter.	2.5:1	12.5
FT: Fourier Transform (NPB)	Class W, 3 iterations	1.3:1	6.8

TABLE II: a and b: methodology tools description; benchmarks description

the memory throughput extracted from the benchmark (ratio of the number of memory transactions and execution time).

To model RF communication, we have considered RF-circuitry modeling and scaling proposed in [17][19] which is also adopted by other reports [17][18][23]. In these models, crosstalk effects, modulation, interference, and noise margin reduction are employed aiming a low bit error rate (BER). In addition, these models are validated with prototypes for different transmission lines [19][24], while following ITRS [25]. We determine RF-interconnection power as in [5]: using McPAT [6] tool at different frequencies to determine FE/TE power components and RF-interconnection power modeling as in [4].

By adopting a methodology similar to the one proposed in [16] to evaluate the memory system, we have selected bandwidth-bound benchmarks with a medium-to-significant number of misses per kilo-instructions (MPKI) taking the following aspects into consideration:

- Generate proper memory traffic and number of outstanding memory transactions in order to utilize a 32 MC-system;
- The selected input sizes are obtained as trade-off between simulation times and memory traffic generated.

In order to evaluate this scalable memory system, we have selected bandwidth-bound benchmarks: (i) STREAM [10] suite, which we decompose in its four sub-benchmarks (Copy, Add, Scale, and Triad); (ii) pChase [11] with pointer chase sequences randomly accessed; (iii) MG, SP, and FFT from NPB benchmarks [26], and Hotspot. All

benchmarks are set to use 32 threads, since we are employing a 32-core processor. No special thread-to-core is applied when executing these benchmarks.

Table IIb lists the benchmarks experimented, input sizes, read-to-write rate, and L2 MPKI obtained in the experiments. In all benchmarks, parallel regions of interest are executed until completion, and input sizes guarantee that all memory space used is evaluated. Average results are calculated based on harmonic average.

B. Results

In this section we present the results regarding the aspects of memory bandwidth, processor performance measured through instruction per cycle (IPC) parameter, and rank energy-per-bit magnitude.

Figure 4a illustrates the results of the bandwidth experiments: as transaction queues are reduced from 16 to 1 entry, we observe a bandwidth reduction of up to 65% for pChase and 91% for STREAM (average of STREAM benchmarks). These largest bandwidth reductions happen for the smallest size transaction queues (1-element).

Bandwidth starts to reduce significantly when transaction queues start to have most of their elements filled: we observe that this specific behavior starts to be observed with 8 elements, but is more present with 4 elements, when bandwidth is significantly reduced. For most of the benchmarks, half of the transaction queue elements can be eliminated without hurting bandwidth magnitudes. The largest bandwidth reduction happens for Hotspot (about 83.5%), while the smallest one happens for FFT (about 87.6%). This can be justified by the bandwidth-bound behavior of the benchmark which is more present in Hotspot (as well as in pChase, STREAM, and MG) and less in FFT.

Figure 4b illustrates the related energy-per-bit results: as transaction queue sizes are reduced from 16 to 1 entry, average energy-per-bit levels reduce since reduced size queues have lesser elements, thus using less amount of energy. The largest energy reduction happens for STREAM while the smallest one happens for FFT. It is interesting to notice that even for 2-element transaction queue, energy-per-bit usage decrease is within the range of (75-90%).

Comparing Figures 4a and b we find that configurations that present interesting trade-offs in terms of performance and energy. For example, configurations with 4 elements is an interesting case, since it presents an interesting bandwidth magnitude - comparable to 16 elements - while presenting some degree of energy-per-bit magnitude savings.

Figure 4c illustrates the effect of shallower transaction queues on processor performance. We observe that, except for FFT, IPC magnitudes generally follow the behavior of bandwidth, i.e., IPC magnitudes reduce as transaction queues are reduced. As sizes are reduced, less amount of memory transactions are processed, thus causing processor performance degradation. Similar to bandwidth, the lowest IPC magnitudes obtained happen for 1-element transaction queues. In particular, since FFT presents a smaller degree of memory-boundness compared to the rest of the benchmarks, this benchmark is less sensitive to IPC variations. Furthermore, in the case of FFT

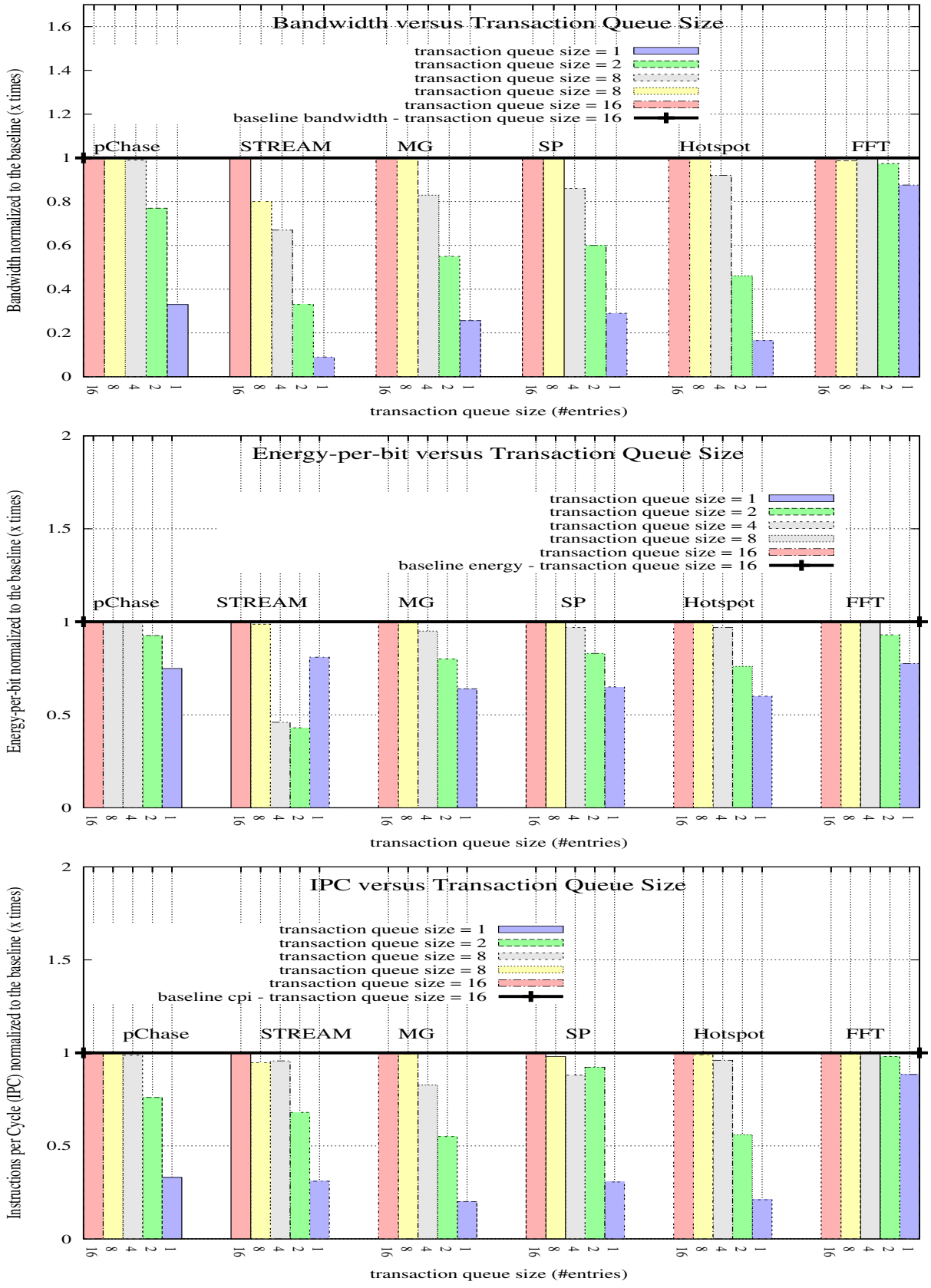

Fig. 4: a, b,c: top to bottom, bandwidth, energy-per-bit, and instruction per cycle (IPC) for STREAM, pChase, MG, SP, Hotspot, and FFT

shallower queues are not only interesting in terms of energy-per-bit but also in terms of bandwidth/IPC.

C. Conditions for a Dynamic Approach

Comparing Figures 4a and b, it is interesting to notice that 4 and 8 entries (up to 50% of the total number of entries 16 entries) have equivalent performance to 16 entries while presenting a potential energy-per-bit saving.

As previously analysed, the point where bandwidth starts to be significantly reduced is for 4 transaction queue sizes. This point is very important for stablishing a dynamical approach: the transaction queue can be configured - or re-configured in the case it relies on a reconfigurable hardware structure - so that transaction queue size could be matched to the bandwidth needs, which is a function of the benchmark, cache L2 MSHRs, and/or number of outstanding memory transactions. We leave this dynamic approach as further efforts.

V. Related Work

Evaluations on transaction queue reduction in terms of power and performance impact was introduced in [27]. In this research, we further advance the investigation of he utilization of shallower queues and evaluated the performance - not only in terms of memory bandwidth but also processor IPC - and power features, verifying and analyzing them for several scientific benchmarks. We also determine conditions for a dynamic approach.

Solutions as Optical- [3] or RF-based [5][8] utilize proper memory interfaces to address the previously mentioned I/O pin restrictions in order to allow MC scalability. Since the adopted approach is focused on transaction queues, it is orthogonal to any of these optical or RF-based solutions. Therefore, it can be applied to any of these systems. The strategy proposed in this investigation can also be applied to commercial systems such as HMC [7] with medium degree of MC scalability. In RFiop [8], it is reported that as MCs are scaled, transaction queue occupation is decreased. This observation is a motivation to further explore the behavior of shallower transaction queues, same as those performed here.

Memscale [28] is a set of software policies and specific hardware power mechanisms which enable the trade-off between memory energy and performance in typical memory systems. It dynamically changes voltage and frequency scaling (DVFS) in terms of memory ranks and memory channels. Memscale is guided by OS performance counters which periodicaly monitor memory bandwidth usage, energy utilization, and the degree of performance degraded in the case of trade-off. The research proposed in this work is orthogonal to Memscale, since we statically explore the impact on the reduction of transaction queues in the context of scalable memory systems, and also determe the conditions for a dynamic approach in terms of transaction queue size reduction.

In [22], Howard et al. propose memory DVFS to address memory power at data centers using bandwidth as a restricting factor. Although our approach is orthogonal to this study - since we focus on evaluating the benefits of

shallower transaction queues - a combined approach DVFS applied to the transaction queue can be a candidate for further investigation.

The study [29] architects servers with mobile memory systems for lower energy-per-bit consumption and efficient idle modes in order to approach energy utilization differences under different bandwidth demands. As part of the architected proposal, this investigation suggests the use of mobile memories with new circuitry to reduce power. That is, this investigation is orthogonal to the mentioned study, since it can be applied to servers with mobile memories that present larger number of MCs where shallower transaction queues can be utilized.

VI. Conclusions and Future Plans

In this paper we have proposed an evaluation on the performance and power impact when performing a static transaction queue size reduction in the context of scalable memory systems. Experimental results obtained show that shallower transaction queues only affect performance in bandwidth-bound applications, as well as energy-per-bit utilization for significant size reductions.

Other types of applications such as those not bandwidth-bound ones are likely to allow an aggressive reduction of transaction queue sizes. As future approach, we plan to investigate a dynamic approach that matches bandwidth and transaction queue size utilization. In addition, methods that combine DVFS [22] to shallower transaction queues under different memory traffic intensities and applications are likely to be considered. Furthermore, heterogeneous systems [30] offer interesting opportunities to evaluate transaction queue size reduction.

From the results achieved in this paper, the evaluation of other scientific benchmarks such as database and bag-of-tasks [31] benchmarks to multicore architectures will also be considered.

In order to improve the energy efficiency of the embedded mobile platforms, the design of a software approach similar those presented in [32] for Software Defined Radio Components are possible to be taken into consideration. Since the approach is targeted to embedded systems, we also plan to develop another approach of the modeling here utilized, referencing the one presented in [33], to facilitate maintenance and model improvement.

VII. Acknowledgements

We would like to thank Maria Amelia Guitti Marino and anonymous reviewers for their important feedbacks, discussions, and suggestions.

References

[1] “LPDDR4 Moves Mobile,” mobile Forum 2013, presented by Daniel Skinner, Accessed date: 06/03/2013; http://www.jedec.org/sites/.../D_Skinner_Mobile_Forum_May_2013_0.pdf.

[2] “JEDEC Publishes Breakthrough Standard for Wide I/O Mobile DRAM,” accessed date: 02/03/2014 ; http://www.jedec.org/.

[3] D. Vantrease et al, “Corona: System Implications of Emerging Nanophotonic Technology,” in *ISCA*. DC, USA: IEEE, 2008, pp. 153–164.

[4] K. e. a. Therdsteerasukdi, "The dimm tree architecture: A high bandwidth and scalable memory system." in *ICCD*. IEEE, 2011, pp. 388–395. [Online]. Available: http://dblp.uni-trier.de/db/conf/iccd/iccd2011.html#TherdsteerasukdiBIRCC11

[5] Marino, M. D., "RFiof: An RF approach to the I/O-pin and Memory Controller Scalability for Off-chip Memories," in *CF, May 14-16 , Ischia, Italy*. ACM, 2013.

[6] Sheng Li et al, "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *MICRO'09*. New York, USA: ACM, 2009, pp. 469–480.

[7] "Hybrid Memory Cube Specification 1.0," accessed date: 03/03/2014 ; http://www.hybridmemorycube.org/.

[8] Marino, M. D., "RFiop: RF-Memory Path To Address On-package I/O Pad And Memory Controller Scalability," in *ICCD, 2012, Montreal, Quebec, Canada*. IEEE, 2012.

[9] Liu, Qing, "QUILT PACKAGING: A NOVEL HIGH SPEED CHIP-TO-CHIP COMMUNICATION PARADIGM FOR SYSTEM-IN-PACKAGE," Ph.D. dissertation, Notre Dame, Indiana, USA, December 2007, Chair-Jacob, Bruce L.

[10] McCalpin, J. D., "Memory Bandwidth and Machine Balance in Current High Performance Computers," *IEEE TCCA Newsletter*, pp. 19–25, Dec. 1995.

[11] "The pChase Memory Benchmark Page," accessed date: 09/12/2012 ; http://pchase.org/.

[12] "CACTI 5.1," accessed Date: 04/16/2013; http://www.hpl.hp.com/techreports/2008/HPL200820.html.

[13] David Wang et al, "DRAMsim: a memory system simulator," *SIGARCH Comput. Archit. News*, vol. 33, no. 4, pp. 100–107, 2005.

[14] "Micron manufactures DRAM components and modules and NAND Flash," accessed date: 12/28/2012 ; http://www.micron.com/.

[15] Nathan L. Binkert et al, "The M5 Simulator: Modeling Networked Systems," *IEEE Micro*, vol. 26, no. 4, pp. 52–60, 2006.

[16] Loh, Gabriel H., "3D-Stacked Memory Architectures for Multi-core Processors," in *ISCA*. DC, USA: IEEE, 2008, pp. 453–464.

[17] M. Frank Chang et al, "CMP Network-on-Chip Overlaid With Multi-Band RF-interconnect," in *HPCA*, 2008, pp. 191–202.

[18] M.C.F. Chang et al., "Power reduction of CMP communication networks via RF-interconnects," in *MICRO*. Washington, USA: IEEE, 2008, pp. 376–387.

[19] M.C.F. Chang et al, "Advanced RF/Baseband Interconnect Schemes for Inter- and Intra-ULSI Communications," *IEEE Transactions of Electron Devices*, vol. 52, pp. 1271–1285, Jul 2005.

[20] Marino, M. D., "On-Package Scalability of RF and Inductive Memory Controllers," in *Euromicro DSD*. IEEE, 2012.

[21] "AMD Reveals Details About Bulldozer Microprocessors," 2011, accessed date: 08/02/2014 - http://www.xbitlabs.com/news/cpu/display/20100824154814 _AMD_Unveils_Details_About_Bulldozer_Microprocessors.html.

[22] David et al., "Memory Power Management via Dynamic Voltage/Frequency Scaling," in *Proceedings of the 8th ACM International Conference on Autonomic Computing*, ser. ICAC '11. New York, NY, USA: ACM, 2011, pp. 31–40.

[23] Sai-Wang Tam et al, "RF-Interconnect for Future Network-on-Chip," *Low Power Network-on-Chip*, pp. 255–280, 2011.

[24] G. Byun et al, "An 8.4Gb/s 2.5pJ/b Mobile Memory I/O Interface Using Bi-directional and Simultaneous Dual (Base+RF)-Band Signaling," in *ISSCC*. IEEE, 2011, pp. 488,490.

[25] "ITRS HOME," accessed date: 09/12/2012 ; http://www.itrs.net/.

[26] "NAS Parallel Benchmarks," accessed date: 03/11/2013; http://www.nas.nasa.gov/Resources/Software/npb.html/.

[27] Marino, M.D; Li K.C., "Reducing Memory Controller Transaction Queue Size in Scalable Memory Systems," in *World Congress on Information Technology Applications and Services*. Jeju, Korea: , 2015.

[28] Deng, Q. et al., "Memscale: active low-power modes for main memory," in *Proceedings of the Sixteenth ASPLOS*. New York, NY, USA: ACM, 2011, pp. 225–238.

[29] Malladi et al, "Towards Energy-proportional Datacenter Memory with Mobile DRAM," in *Proceedings of the 39th Annual International Symposium on Computer Architecture*, ser. ISCA '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 37–48.

[30] Marino, M. D., Li, K.C., "Insights on Memory Controller Scaling in Multi-core Embedded Systems ," *International Journal of Embedded Systems*, vol. 6, no. 4, 2014.

[31] Ami Marowka, "TBBench: A Micro-Benchmark Suite for Intel Threading Building Blocks," *Journal of Information Processing Systems*, vol. 8, no. 2, pp. 331–346, 2012.

[32] Energy Efficient Architecture Using Hardware Acceleration for Software Defined Radio Components, "Chen Liu, Omar Granados, Rolando Duarte and Jean Andrian," *Journal of Information Processing Systems*, vol. 8, no. 1, pp. 133–144, 2012.

[33] Christian Bunse, Yunja Choi and Hans Gerhard Gross, "Evaluation of an Abstract Component Model for Embedded Systems Development," *Journal of Information Processing Systems*, vol. 8, no. 4, pp. 539–554, 2012.