System Implications of LLC MSHRs in Scalable Memory Systems

Mario Donato Marino[1], Kuan-Ching Li[2]

Abstract

By exploring the scalability of memory controllers (MCs) and ranks in scalable memory systems, larger degrees of memory bandwidth are offered when scaling cores in traditional multicores and embedded systems, and the ratio computation versus memory width - expressed as ratio between the number of cores and MCs - favors the former in detriment to the latter. In scalable memory systems, this ratio tends to balance the number of cores and MCs. Furthermore, since each core has their last level cache (LLC) strongly subject to the number of miss status holding registers (MSHRs) present, which retain information on all outstanding misses of a specific cache line, it is fundamental to evaluate the impact of these elements in scalable memory systems. Experimental results show that, as reducing the number of MSHRs, bandwidth levels are reduced about 64% and energy-per-bit levels are increased of about 36% for stream-based patterns and remain unaltered for random ones.

Keywords:
memory; controller; scalable; miss; handling; status; register; mshr.

Keywords: memory, controller, scalability, frequency, scaling.

1. Introduction

The high number of cores in current embedded and traditional multicores has put a high pressure on the memory system. As an effort to approach the I/O pin scalability - determinant factor of memory controller (MC) scalability - scalable memory systems utilize memory interfaces that allow I/O pin reduction and large-magnitude data rates, thus allowing MC scalability.

In order to improve memory bandwidth, traditional double data rate (DDR) memory design have focused on memory frequency, that is, applying larger clock frequencies to memory formed by set of memory banks with data output aggregated and sharing addresses. Scalable memory systems also present the advantage of power, by shifting the the traditional focus on FS to memory width, represented by MC and rank scalability, assuming one rank for each MC or memory channel [1]. For instance, Corona [2] is able to scale to 64 optical-MCs while DIMM Tree [3] up to 64 RFMCs (RF-based memory controllers).

As reported in [2][5][1][3], as MCs matched to ranks are scaled, memory bandwidth is scaled as well. Thus, higher traffic going through the caches and respective network-on-chip (NoC) is increased. Under this higher traffic scenario, L2 caches are representative of LLC in this study, and the number of MSHRs can significantly impact the memory system. According to the report [6], by aggressively banking its structure or by using highly-associative MSHRs in a unified structure, bandwidth is limited.

Due to these limitations, which potentially limit with the growing of embedded/traditional multicore bandwidth demands, MSHR implications need to be further evaluated. In this scenario, the evaluation of MSHR impact on bandwidth in the space of scalable memory systems has not been yet widely explored. To address these challenges, we propose to evaluate the impact of MSHR elements in these memory systems. We create a model based on scalable interface technologies, and assess it with different LLC MSHR counts (number of MSHRs) using detailed and accurate simulation tools com-

Email address: m.d.marino@leedsbeckett.ac.uk, kuancli@pu.edu.tw (Kuan-Ching Li)
[1]Leeds Beckett University (UK)
[2]Providence University (Taiwan)

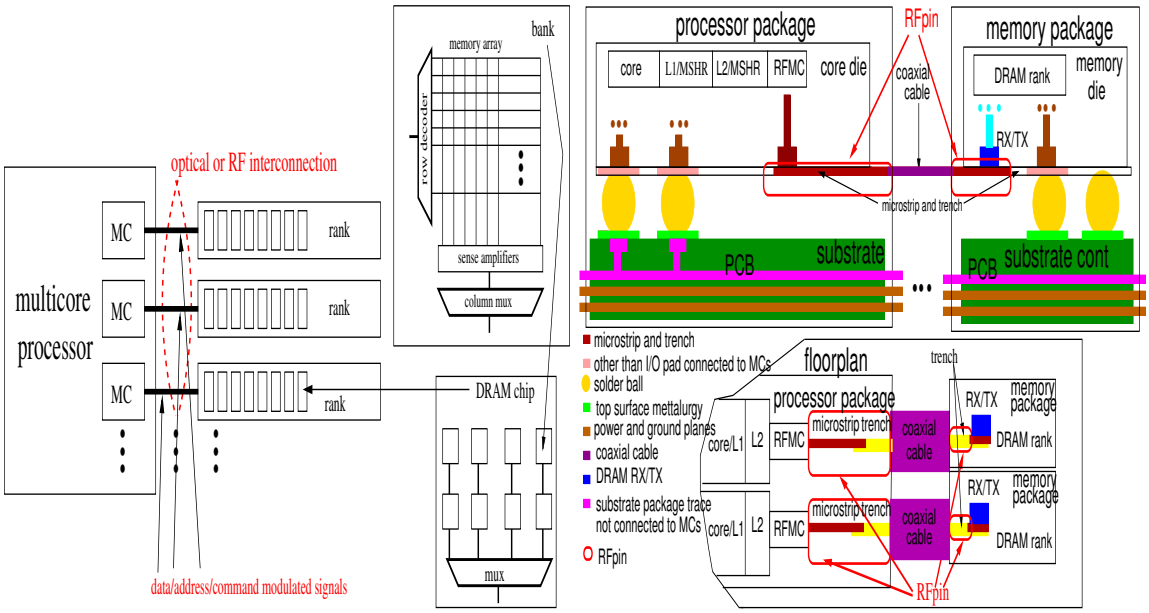

Figure 1: left to right: (a) memory system general overview replicated from [4]; (b) RFiof replicated from [1]

bined with memory bandwidth-bound benchmarks. We envision to advance the state of art of scalable memory systems through the following contributions:

- Perform a design space exploration of MSHRs in scalable memory systems.
- Determination of bandwidth impact when experimenting different counts (number) of LLC MSHR structures, under different workload conditions.
- Determination of the processor performance impact when varying the number of MSHRs under different workload conditions.
- Evaluating rank energy-per-bit impact impact when utilizing different LLC MSHR counts.

Similar to [7], we focus on the implications of MSHR structures in terms of bandwidth in scalable systems rather than on MSHR structures itself. We assume that MSHR structures in each L2 cache as in scalable memory systems [2] do not employ significant area.

This paper is organized as follows. Section 2 presents the background while implications on scaling MSHRs are depicted in Section 3. Next, Section 4 presents the experimental results obtained, Section 5 the related work, and finally, in Section 8 the concluding remarks as well as future plans.

2. Background

In this section we present a background about scalable memory systems and compare them to typical commercial solutions.

Recently developed commercial memory solutions still employ large number of pins, which can restrict MC scalability. For instance, Hybrid Memory Cube [8] employs 55 pins and can utilize up to 8 MCs, presenting the maximum aggregated bandwidth of 320 GB/s while each I/O-link presents individually 10 Gbit/s. Furthermore, wide I/O 2 [9] employs 128 bits per rank and 8 MCs, and still MC-count restricted, since total width is 1024 bits.

In particular, along scalable memory 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. Moreover, while command, clock (CK), and address signals are demodulated in these ranks, they also modulate data to be returned to the MC, when a read operation is performed. Figure 1a illustrates the context where the memory path is utilized.

As example of scalable solution, we illustrate RFiof [1] in Figure 1b. RFiof is designed to scale up 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 [3]), this technology has the potentiality to be scaled to use 64 RFMCs and ranks of 17.2 GB/s and likely to achieve the bandwidth of 1024GB/s (and total width of 4096 bits) - similar to bandwidth magnitude achieved in optical technologies [2][10].

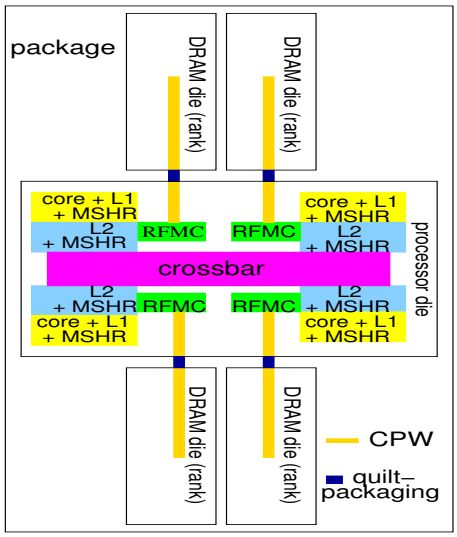

Figure 2: MSHRs in the context of a scalable memory system (replicated from [5], where quilt-packaging [11] is a coplanar waveguide - CPW)

As we have described and compared scalable memory systems to traditional memory systms when it comes to achieving larger bandwidths, we address next the effect of MSHR elements in scalable memory systems.

3. MSHRs and Scalable Memory Systems

In this section, we describe the main function of the MSHRs and how these elements are important when having a scalable memory system.

3.1. MSHR and MSHR scalability

Current multicores and embedded systems present LLC system typically implemented as a shared unit, where each of its slices is distributed and connected by a NoC [12][13], as illustrated in Figure 2. In this unit, each of its L2 caches present MSHRs.

According to [6], typically the MSHR structure or miss handling architecture (MSA) is the logic circuitry required to be able to support outstanding misses events. The report by Kroft [14] is the first to propose an MSHR structure that supports outstanding misses, via the utilization of one MSHR file, which is designed to assist miss detections and combined to the store of missed buffers, forwarding to subsequent misses.

Regardless of the cache level, according to [6], a cache miss on a line is defined as primary one where there is currently no outstanding miss on the line (an entry on the MSHR file) and which requires the allocation of a new MSHR.

A secondary miss, according to Tuck et al.'s report [6], indicates that there is a pending miss already happening on the line. Still according to Tuck's report, in the case of a secondary miss, the already created MSHR is enlarged to store the new miss, and no request is issued to the memory system; similarly, if other misses on that line happen, the MSHR keeps the information for all these outstanding misses (MSHR sub-entries that point to the destination register that is going to be the operation destination) on that line.

In addition, according to the same report [6], as programs are executed, MSHRs entries and or sub-entries are utilized until these are finished - locks-up the cache and from that point, that cache does not respond further requests from the processor, which may eventually lead to the processor stall and performance bottleneck.

As observed in Figure 2, each MSHR bank directly communicates with its paired MC. Furthermore, the study by Loh [15] shows that by providing less MSHR counts can prevent the full utilization of the memory and parallelism provided by multiple MCs. Moreover, by increasing the number of MSHRs we control the number of L2 outstanding misses.

To summarize, given their fundamental importance, MSHRs are responsible for controlling the bandwidth requested from/to the MCs/ranks.

3.2. Scalable Memory Systems and System Cache Interconnection

Scalable memory solutions improve the bandwidth on the physical memory side [3][5][1]. However, to have these higher levels of bandwidth delivered to the processor, not only the memory system has to be improved but also the interconnection and the cache system [2], otherwise the higher levels of bandwidth are not able

tool	description
Cacti [16]	cache latencies configured with
Gem5 [17]	Capture memory transactions from Gem5 processor simulator to Gem5 memory simulator, which responds with the result of the memory transaction. Determine energy-per-bit spent and the number of memory accesses.
Gem5 processor simulator module [17]	Configured as 32-core OOO processor and not L2 shared cluster (avoid sharing). Generates memory transactions which are passed to Gem5 memory simulator module [17]. Miss-status handling register (MSHR) counts 1 to 20.
RF-crossbar	Implemented in Gem5 [17] with RF settings from [18][19].
RF-communication delays	RF-circuitry modeling and scaling [18][20].

Table 1: methodology: tools and description

to achieve the processor side, i.e., they are restricted at the interconnection.

Therefore, in order to have the higher bandwidths achieving the processor, the cache system and interconnection has to be improved as well. An overall picture of the memory system can be observed in 3, where the MSHRs are placed at the LLC levels. When data are brought from memory or saved to memory, data should be kept coherent among the caches if that is required. Therefore, the interconnection has a fundamental role and should perform accordingly without penalizing memory bandwidth or latency.

For example, on optical Corona system [2], an optical crossbar is employed to enable significant higher degrees of bandwidth to be transferred from the optical scalable memory system to the processor and vice-versa. With the benefits of an optical low-latency and high-bandwidth cache interconnection, Corona memory system is able to achieve higher levels of bandwidth and improve processor performance [2].

Another example of cache interconnection system that enables higher memory bandwidth is implemented on the reports by Chang et al. [18][19]. Through a configurable RF-based cache interconnection, low-latency and high-bandwidth cache interconnection, higher levels of memory bandwidth produced by 64 MCs/ranks in DIMM Tree memory system are able to achieve the processor and vice-versa.

Similarly, a high-throughput 80GB/s RF-crossbar interconnecting the private LLC in RFiof [1], enables the 32 MC/rank scalable memory system to achieve the processor with lower-degree latencies and high-degree of memory bandwidth.

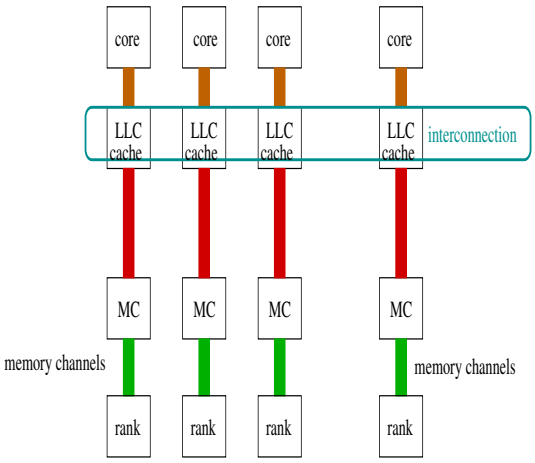

Figure 3: overall picture, MSHRs in the LLC cache)

3.3. MSHRs, scalable memory systems and Investigation Trade-offs

Given that the LLC and the physical memory system are part of the memory hierarchy, the reduction on the MSHR elements and LLC bandwidth, the physical memory system bandwidth is likely to be certainly affected.

Given that memory requests - reads and writes - are generated by programs, bandwidth generated as a result of program requests is generated at the ranks. If the number of MSHR is reduced, the number of memory requests - simultaneous - on a multicore are likely to be reduced. Moreover, on a scalable memory system, if the number of MSHRs are restriced, the number of oustanding parallem miss memory requests is restricted and, even under a higher availability of MCs - such as on scalable memory systems, bandwidth generated is likely to be restricted by the lesser availability of MSHRs. We demonstrate these

aspects in Section 4 where we perform a design space exploration of the number of MSHRs and how these impact the bandwidth of a scalable memory system.

Furthermore, if memory bandwidth is affected, bandwidth-bound programs - the typical target of scalable memory systems, processor performance is likely to be affected as well. The design space exploration previously mentioned (Section 4) also includes the investigation of the effect of these elements on processor performance - typically measured as instructions per cycle or IPC, where different benchmarks with different memory access patterns are likely utilized as further described. Such investigation is likely to point whether the absence of MSHRs affects most memory access patterns and/or if some particular programs are not affected. If not affected, a likely shutdown could pottentially be applied thus benefiting LLC cache power reduction.

Given that MSHRs are fundamental elements of the memory hierarchy, and given the augment on the number of cores in recent multicore generations, the pressure on the memory system has triggered the need for scalable memory solutions [1][5].

In this investigation it is also important to determine the behavior of the bandwidth - for example, if it is proportional to the number of MSHRs a processor designer can stablish the bandwidth trade-offs when designing this elements. Therefore, the designer can potentially stablish the acceptable bandwidth limits when reducing the amount of these elements. Similarly, since bandwidth reflects into processor performance, the investigation is likely to elucidate similar trade-offs between number of MSHRs and IPCs on scalable memory systems.

The reduction on LLC MSHR elements can save LLC cache power reduction. However, energy-per-bit consumption of the rank elements on a scalable memory system are likely to be affected by bandwidth reduction, which justifies the design space exploration.

On another note, energy-per-bit behavior is a function of the read-to-write ratio [21], which depends on the memory access pattern of each program. To investigate how different programs are affected by a reduction of the number of MSHRs, in the design space exploration previously mentioned (Section 4), we determine the impact of these elements on the rank energy-per-bit utilized on scalable memory systems, using different benchmarks - as previously described.

4. Experimental Results

In this section we perform a series of experiments to demonstrate the impact of MSHR counts in scalable memory systems.

4.1. Methodology

To have a global picture of the methodology employed in this study, we have listed all simulators employed and corresponding description are found in Table 1. The general methodology employed to obtain bandwidth is adopted from [22]: by using bandwidth-bound benchmarks to stress the memory system, we combine Gem5 memory module simulator module [17] to Gem5 processor simulator module [17] as follows.

Before describing the experiments, we observe that the baseline for experiments presents 32 cores and 32 RFMCs to maintain the ratio core:MC same (32:32) such as in [1]. The higher number of MCs/RFMCs reflects scalable memory systems, while the larger number of cores is adopted as a representative of future systems. We adopted a balanced core:MC rate such as in [2][3] to be able to provide a proper core-to-MC ratio needed to approach bandwidth-bound applications.

In order to evaluate this scalable memory system, we combine detailed accurate simulators using the methodology developed in [22]: we combine the creation of a 32-multicore model in Gem5 processor simulator module [17], which upon benchmark execution of a multicore model generates memory transactions then captured by Gem5 memory simulator module [23], that is configured with 32 MCs/RFMCs so that core:MC ratio is 32:32. In the sequence, Gem5 memory simulator responds to Gem5 processor simulator with the result of each memory transaction.

We employ a 2.0-GHz (Alpha ISA) and 4-wide out-of-order (OOO) core, while having RFMCs at 1.0GHz (typically at half of microprocessor clock frequency [24]). We use Cacti [16] to obtain cache latencies and adopt MSHR counts of typical microprocessors [15]. 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 [18][19]: 200ps of TX-RX delays, plus the rest of the cycle to transfer 64 Bytes using high speed and modulation).

Observing the RF-crossbar upper constraint, we have selected a medium data-rate DDR3-rank

Core	2.0 GHz, OOO, multicore, 32 cores, 4-wide issue, turnament branch predictor
Technology	22 nm
L1 cache	32kB dcache + 32 kB icache; associativity = 2 MSHR = 4, latency = 0.5 ns
L2 cache	1MB/per core ; associativity = 8 MSHR = 1 to 20; latency = 1 ns
RF-crossbar	latency = 1 cycle, 80GB/s
RFMC trans. queue	32 RFMCs; 1 RFMC/core, 1.0GHz, on-chip entries = 16/RFMC, close page mode
Memory rank	DDR3 1600MT/s, 1 rank/RFMC, 1GB, 8 banks, 16384 rows, 1024 columns, 64 bits, Micron MT41J512M8[21] reduced to 1 Gbit, tburst=5ns, trcd=tcl=13.75ns tras=35ns,
RF interconnection length size delay	 2.5 cm 0.185ns

Benchmark	Input Size	read : write	MPKI
Copy, Add, Scale, Triad (STREAM)	4Mdoubles per per core 2 inter	2.54:1	54.3
pChase	64MB/thread, 3 iter, random	158:1	116.7
Scalar Pentadiagonal: SP (NPB)	Class B 2 iter	1.9:1 1.9:1	11.1 11.1
Multigrid:MG (NPB)	Class B 2 iterations	76:1	16.9
Srad	2048 elements 2 iter.	2.5:1	14.9
Hotspot,	2000 x 2000 3 iter.	2.5:1	12.5
Backprop, (Rodinia)	20000 elements 2 iter.	-	

Table 2: a and b: methodology tools description; benchmarks description

employed in typical PCs and smartphones/pads (64 data bits, based on the DDR3 model Micron MT41K128M8 of 1GB [21], and listed in Table 2a).

Each RFMC is assumed to be connected to one rank 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 [25], this mode benefits performance and energy utilization in multicores. Last, all architectural parameters are summarized in Table 2a.

We obtain the total rank energy-per-bit magnitudes spent from Gem5 [17] memory module simulator. Gem5 energy-per-bit memory modeling is based on memory maker design circuits (Micron [26]).

To model RF communication, we have considered RF-circuitry modeling and scaling proposed in [18][20], also adopted by other reports [18][19][27]. 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 [20][28], following ITRS [29]. We determine RF-interconnection power as in [1]: using McPAT [30] tool at different frequencies to determine FE/TE power components and RF-interconnection power modeling as in [3].

By adopting a methodology similar to the one proposed in [15] 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 baseline configuration presents 32 RFMCs and 20 MHSRs.
- The selected input sizes are a trade-off between simulation times and memory traffic generated.

In order to evaluate this scalable memory system under significant different benchmark memory access patterns, we have selected 10 bandwidth-bound benchmarks from different suites: (i) STREAM ADD, COPY, SCALE and TRIAD from the STREAM suite [31] ; (ii) pChase [32] with pointer chase sequences randomly accessed; (iii) Scalar Pentadiagonal (SP)

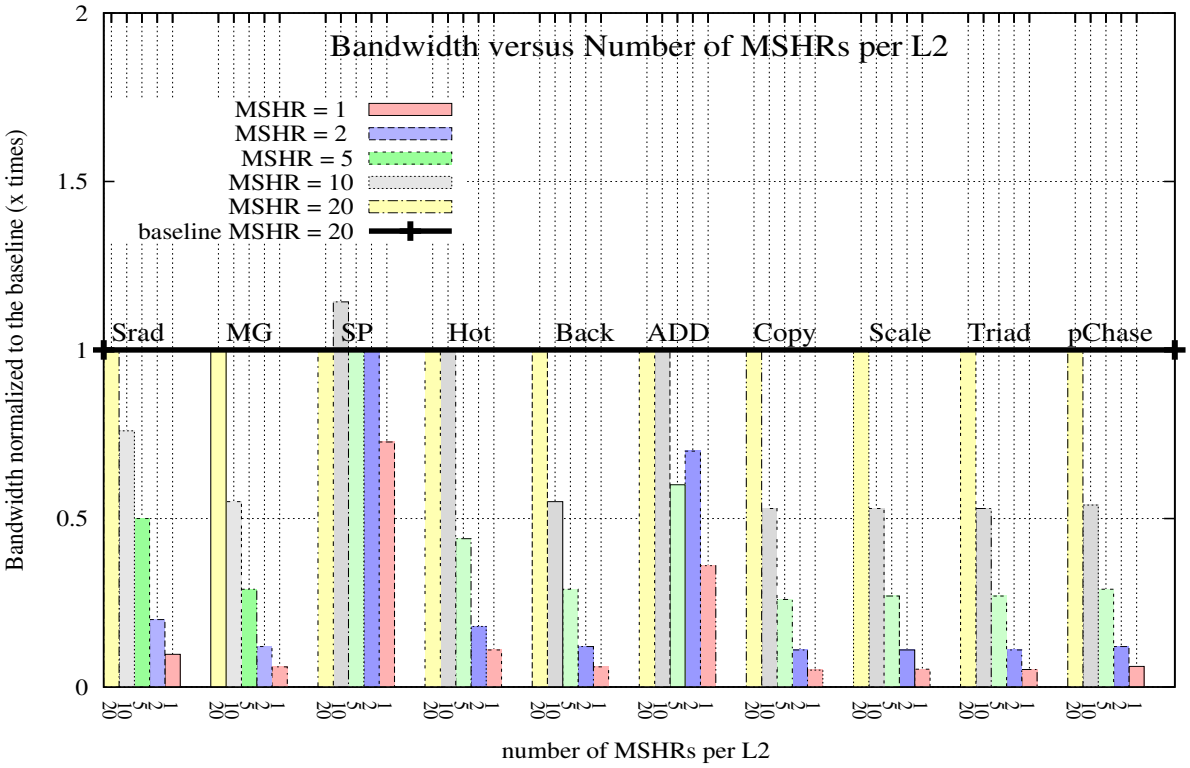

Figure 4: bandwidth versus number of MSHRs

and Multigrid (MG) from NPB [33]; and (iv) Srad, Hotspot and Backpropagation from Rodinia suite [34]. All benchmarks are set to use 32 threads, since we are using a 32-core processor. It is important to highlight that we are experimenting a stream-based patern (STREAM) - that can be predicted. However, we are also experimenting pChase set with random behavior, which significantly difficults prediction or prefetching. Besides both previous patterns, the other benchmarks are general applications which do not follow any specific patterns.

No special thread-to-core mapping is applied when executing these benchmarks. Table 2b lists the benchmarks chosen, 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.

4.2. Results

In this section we present the results regarding the aspects of memory bandwidth, processor performance (measure in terms of instructions per cycle - IPC) and rank energy-per-bit magnitude.

Figure 4 illustrates the results of the bandwidth experiments. As a general observation, as we reduce the number of MSHRs, and given the bandwidth-bound behavior of all these programs, bandwidth reduces proportionally for most of the benchmarks. In this case, upon MSHR reduction, the number of available register positions (and related information) at MSHRs minorly reduce the number of simultaneous misses and concurrency, thus causing performance reduction.

The exceptions are SP and Hotspot, where the availability of number of MSHRs after or equal $MSHR = 5$ strongly suggests that more than 10 MSHRs do not bring bandwidth benefits. Therefore, for programs such as Hotspot and SP bandwidth magnitudes remain constant regardless the number of MSHRs. This behavior is justified since the miss holding registers present in MSHR are likely not to predict the memory accessed patterns of these programs. The largest bandwidth reductions happen in Backpropagation, Copy, Scale and Triad. Bandwidth degradation happens for general programs such as SP, MG and Hotspot as well as for stream-based ap-

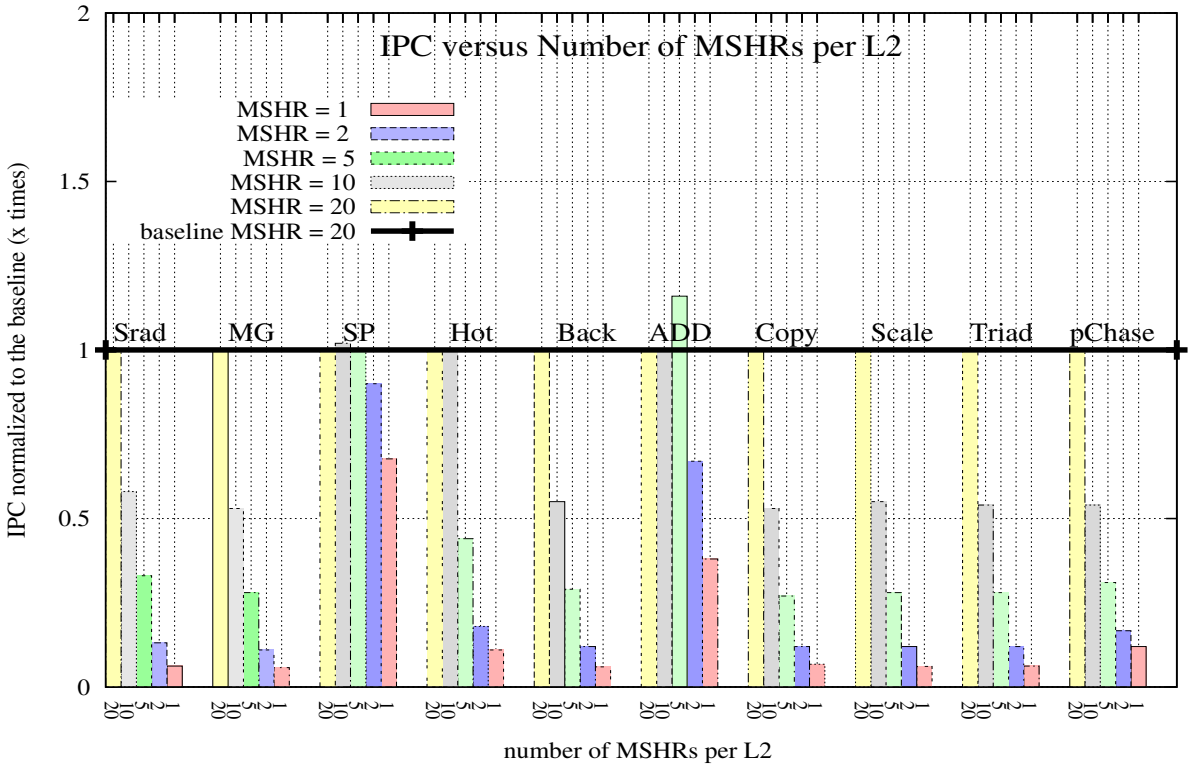

Figure 5: IPC versus number of MSHRs

plications (STREAM suite) and random-pattern ones (pChase).

Similarly, obtained IPCs follow bandwidth behavior as observed in Figure 5. Given that these benchmarks are bandwidth-bound ones, processor performance is directly a function of the memory system, reason why these results are observed. The largest IPC reductions happen for MG, Backpropagation, Copy, Scale and Triad.

Moreover, by analyzing the statistics of our simulation infrastructure, we observed that some L2 slices presented significantly different L2 miss rates as MSHRs are varied. This context is similar to the churn phenomenon - when a larger number of requests to the memory system and responses to these requests are generated by scaling MSHRs as described in [15], will not necessarily decrease L2 miss rates.

Total energy-per-bit represents the total energy spent on each rank, as illustrated in Figure 6. In general, except for MG and SP, we observe that MSRHs can be reduced with energy-per-bit benefits, however this reduction does significantly affect performance.

Comparing Figures 4 and 6 we can observe that resulting magnitudes present interesting trade-offs in terms of bandwidth and energy-per-bit as MSHRs are varied. For stream-based patterns and some scientific applications, we could observe that, after less than 10 MSHRs, as MSHRs are increased, bandwidth is reduced and energy-per-bit is increased.

Interestingly, memory energy-per-bit reduction is not proportional to MSHR reduction. That is an interesting outcome for the processor designer, and several trade-offs can be derived. For example, observing Figure 6, if energy-per-bit is the restriction factor and the designer has a budget of about 5x the energy-per-bit baseline ($numberofMSHRs = 20$), for most of the benchmarks only 2 MSHRs are required, which allows a significant LLC MSHR reduction (from 20 to 2 MSHRs), thus likely saving cache energy. If the budget is set to 3x energy-per-bit baseline, up to 5 MSHRs are needed, thus similarly allowing a significant LLC MSHR reduction (from 20 to 5 MSHRs), thus likely saving cache energy.

Some other trade-offs are likely to be observed:

- bandwidth and IPC reduction is not proportional to the number of MSHRs for all benchmarks: Srad, SP, Hotspot and pChase present larger bandwidth/IPC or are not

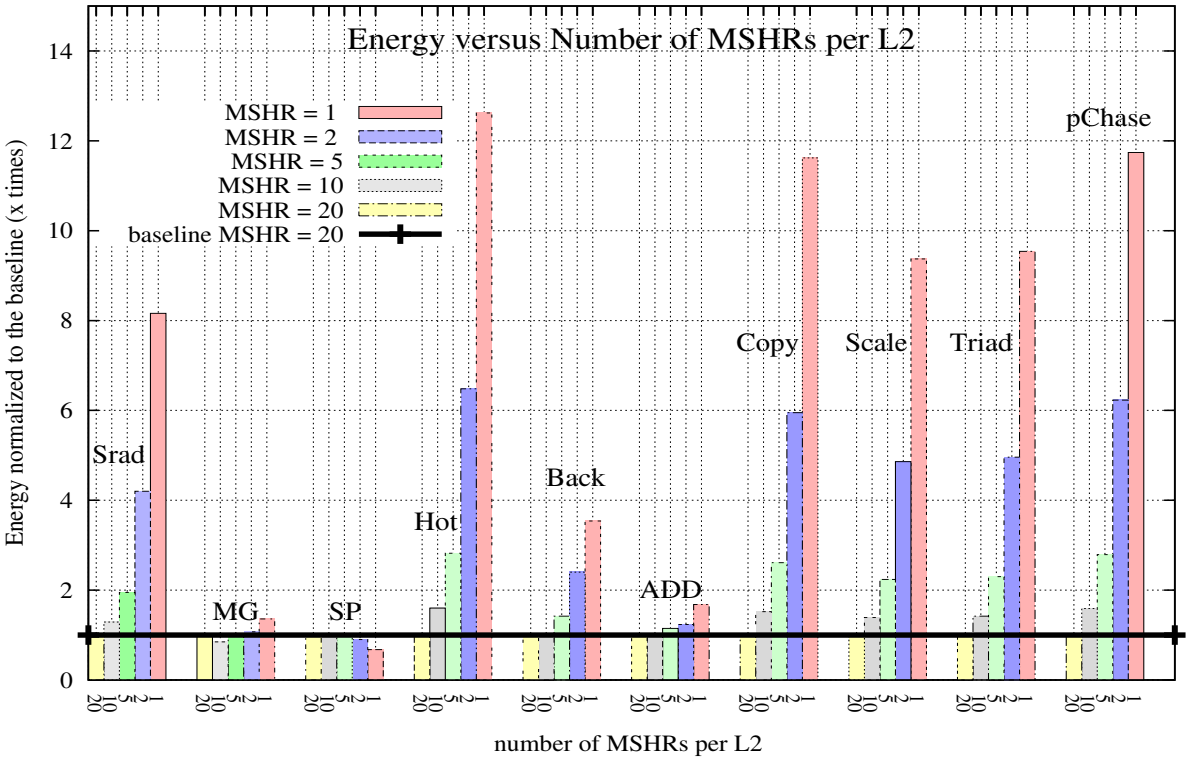

Figure 6: rank energy-per-bit versus number of MSHRs

affected. This is an important observation since for these benchmarks a likely L2 MSHR shutdown aiming energy savings are likely to be considered.

- Still in the previous case, as less MSHRs are required, it is an interesting observation for the design architect - the cache system can be optimized and not only area but also power can be saved.
- if bandwidth/IPC reduction - upon MSHR reduction - is not important and rank energy-per-bit levels are important, MSHRs can be reduced significantly such as in MG, SP and pChase. bandwidth and IPC reduction is not proportional to the number of MSHRs for all benchmarks:

5. Related Work

Sohi and Franklin evaluated the bandwidth advantages in non-blocking caches, each one with its own MSHR file [7]. We adopted similar strategy on focusing the implications of MSHR utilization, in terms of bandwidth rather than MSHR structure.

In [6], it was proposed a scalable miss handling architecture (MSA) that employs a smaller MSHR per cache bank shared among caches, and a bloom filter to reduce MSHR searches in shared MSHR. In this study, we did not consider the application of such scalable MSA, though we follow traditional MSHR utilization of structures on each LLC structure. Studies of 3DStacking memory systems and implications of scaling MHA via scaling MSHRs are discussed in [15]. The focus in this paper is related to off-chip scalable memory systems that present more MCs also many ranks, and therefore, present different challenges on the number of MSHRs.

The study [35] 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 study suggests the use of mobile memories with new circuitry to reduce power. We instead are focused on approaching high-bandwidth demands by having a larger number of MCs, and are evaluating memory FS implications not only in terms of bandwidth but also energy behavior. Similar to the

previous study, we perform a design space exploration of different FS configurations in order to understand the impact on memory system. We can take advantage of the former proposed techniques to reduce power and energy.

While DIMM Tree [3], RFiop [5], and RFiof [1] are RF-memory organizations which aim to improve power and performance, where the two latter are respectively focused on pad/pin reduction and RFMC scalability. The report by Marino et al. [36], focus on the energy and performance implication of the RFMC transaction queue sizes. Memory FS here investigated is orthogonal to RFMC scalability, and is applicable to any of these previously mentioned studies. Our approach is focused on the modeling and design space exploration of an advanced memory system model and different to the proposed by Marino [4], where frequency scaling is based on the amount of time higher frequencies are applied, bandwidth and temperature.

6. Sensitivity Analysis

We perform a sensitivity analysis to assess the impact of the key aspects: number of memory MSHRs and MCs/ranks; number of cores; and high-speed transmission delays.

6.1. Number of MSHRs and MCs/ranks and Number of Cores

To select a proper balance between computation and memory utilization, a proper number of cores and number of MCs have to be selected. According to the reports by Marino [4] [1][5], the use of equivalent number of MCs to cores favors memory width, which is the case of scalable memory systems.

We have selected the ratio cores:MCs set as 32:32 (32 cores, 32 MCs/32 ranks) in this experiments to find out the performance effects under different MSHR magnitudes. Other core:MC ratios - by evaluating other MC count magnitudes such as extensively performed in [1][5] - could be evaluated however, given that smaller MC magnitudes would favor lower LLC demand, we have not selected to experiment these settings.

Moreover, we have selected 32 cores to explore future multicore generations given current high performance microprocessors are currently using 16 cores [24].

6.2. High-speed transmission delays

It is intuitive to notice that if a high-speed transmission interconnection (optical or RF) is utilized along the memory path, delays regarding these interconnection part are not relevant on the total memory path [5][1]. It is important to notice that this is not the case of typical DDR memory systems, or HMC [8] systems, where interconnection delays are significant. In addition, if the target selected is a traditional DDR system instead, delays on the digital interconnection are likely to cause significantly larger latency magnitudes and bandwidth degradation. Therefore, if current memory systems are evaluated the impact of delays along the memory system should be considered when performing a MSHR magnitude investigation.

6.3. Performance: high number of MCs/ranks and Crossbar

In this investigation we perform a MSHR investigation on how memory bandwidth is affected when having a scalable memory system (32 MCs/32 ranks), which contains significant higher amount of MCs than in typical microprocessors [1].

Furthermore, given that RF-memory systems aim to achieve larger memory paralellism by having a scalable number of MCs, we adopt a MC:rank proportion as 1:1 to explore the maximum bandwidth provided by each rank. If ranks share the same MC, channel contention is likely to happen, there bandwidth available is likely to be lower [23].

The crossbar utilized here has its larger bandwidth limit set to 80 GB/s (table **??**). Given we adopted similar RF-interconnection settings to [1], its upper bandwidth is designed so that when MCs and ranks are scaled, it does not restrict their bandwidth scaling that could pottentially disturb the MSHR experiments.

7. Summing Up Achieved Contributions

We have demonstrated that we advance the state of art of scalable memory by understanding through complex, extensive and detailed design exploration simulations, the effects of of the number of MSHRs on scalable memory systems in terms of:

- memory system bandwidth: as a general behavior (general programs and STREAM-based ones), if the number of MSHRs increases, bandwidth proportionally decreases

given the lower number of simultaneous cache misses decreases as lesser number of MSHRs are available. Random-memory access programs (such as pChase) do not have bandwidth affected by the reduction of MSHRs. Therefore, for this type of program this structures could be shut down in order to lower LLC power consumption.

- processor performance: IPCs follow bandwidth, i.e., the reduction on the number of MSHRs decrease processor performance. Some programs such as SP and pChase are not significantly affected by MSHR reduction, indicating a likely case for MSHR shut down and LLC power consumption reduction.

- rank memory energy: in general, as the number of MSHRs are reduced rank energy-per-bit levels increase as a consequence of larger memory utilization. Surprisingly, for some programs (MG, SP and pChase), rank memory energy-per-bit usage has remained roughly invariable (except for MSHR = 1). For pChase that is quite expected behavior due to its ramdom access pattern behavior, however MG and SP - general scientific programs from NPB - open the opportunity for saving LLC power consumption.

8. Conclusions and Future Plans

In this paper, we have evaluated the impact of MSHRs in scalable memory systems, by performing a design space exploration for MSHR counts in such systems. Experimental results show that, for stream-based patterns and for a memory-bound scientific program, as MSHRs are scaled bandwidth is scaled, while total energy-per-bit usage is reduced. For random patterns, either memory bandwidth and energy-per-bit usage kept unaltered.

In all memory behaviors observed, given that there are significant number of MCs, we have found conditions where the reduction of MSHRs does not alter bandwidth and energy-per-bit behavior.

As a future approach, we plan to propose a combined strategy which involves MSHR and MC scalability to approach memory bandwidth and cache power utilization. This approach is interesting since it involves MSHRs - which belong to the network-on-chip (NoC) - and MCs - which belong to the memory system. Furthermore, we intend to evaluate the cache energy impact of the different MSHR utilizations as well as other memory traffic patterns.

[1] 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, pp. 100–110.

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

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

[4] Marino, M.D., ABaT-FS: Towards adjustable bandwidth and temperature via frequency scaling in scalable memory systems, Microprocessors and Microsystems.

[5] 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, pp. 183–188.

[6] J. Tuck et al., Scalable Cache Miss Handling for High Memory-Level Parallelism, in: MICRO, IEEE, DC, USA, 2006, pp. 409–422.

[7] G. S. Sohi, M. Franklin, High-bandwidth data memory systems for superscalar processors, SIGOPS Oper. Syst. Rev. 25 (Special Issue) (1991) 53–62.

[8] Hybrid Memory Cube Specification 1.0, accessed date: 11/23/2015 ; http://www.hybridmemorycube.org/.

[9] JEDEC Publishes Breakthrough Standard for Wide I/O Mobile DRAM, accessed date: 12/16/2015 ; http://www.jedec.org/.

[10] Aniruddha N. Udip, Designing Efficient Memory for Future Computing Systems , in: PhD Thesis, University of Utah, School of Computing, Utah, USA, 2012, pp. 1–126.

[11] Liu, Qing, QUILT PACKAGING: A NOVEL HIGH SPEED CHIP-TO-CHIP COMMUNICATION PARADIGM FOR

SYSTEM-IN-PACKAGE, Ph.D. thesis, University of Notre Dame, Notre Dame, Indiana, USA, Chair-Jacob, Bruce L. (December 2007).

[12] 32-core CMP with Multi-sliced L2: 2 and 4 Cores Sharing a L2 Slice, SBAC-PAD '06, IEEE Computer Society, Washington, DC, USA, 2006.

[13] L2-Cache Hierarchical Organizations for Multi-core Architectures, Vol. 4331 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006.

[14] D. Kroft, Lockup-free instruction fetch/prefetch cache organization, in: Proceedings of the 8th Annual Symposium on Computer Architecture, ISCA '81, IEEE Computer Society Press, Los Alamitos, CA, USA, 1981, pp. 81–87.

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

[16] CACTI 5.1, accessed Date: 01/20/2016; http://www.hpl.hp.com/techreports/2008/HPL-200820.html.

[17] Binkert, Nathan et al, The Gem5 Simulator, SIGARCH Comput. Archit. News 39 (2) (2011) 1–7.

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

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

[20] M.C.F. Chang et al, Advanced RF/Baseband Interconnect Schemes for Inter- and Intra-ULSI Communications, IEEE Transactions of Electron Devices 52 (2005) 1271–1285.

[21] Micron manufactures DRAM components and modules and NAND Flash, accessed date: 11/28/2015 ; http://www.micron.com/.

[22] Marino, M. D., On-Package Scalability of RF and Inductive Memory Controllers, in: Euromicro DSD, IEEE, 2012, pp. 923–930.

[23] David Wang et al, DRAMsim: a memory system simulator, SIGARCH Comput. Archit. News 33 (4) (2005) 100–107.

[24] AMD Reveals Details About Bulldozer Microprocessors, accessed date: 11/10/2015 - http://www.xbitlabs.com/news/cpu/display/20100824154814_AMD_Unveils_Details_About_Bulldozer_Microprocessors.html (2011).

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

[26] Calculating Memory System Power for DDR3 Introduction, accessed date: 10/28/2012 ; http://www.micron.com/.

[27] Sai-Wang Tam et al, RF-Interconnect for Future Network-on-Chip, Low Power Network-on-Chip (2011) 255–280.

[28] 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.

[29] ITRS HOME, accessed date: 11/23/2015 ; http://www.itrs.net/.

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

[31] McCalpin, J. D., Memory Bandwidth and Machine Balance in Current High Performance Computers, IEEE TCCA Newsletter (1995) 19–25.

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

[33] NAS Parallel Benchmarks, accessed date: 01/03/2016; http://www.nas.nasa.gov/Resources /Software/npb.html/.

[34] Shuai Che et al , Rodinia: A benchmark suite for heterogeneous computing., in: IISWC, IEEE, 2009, pp. 44–54.

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

[36] Marino, M.D; Li K.C., Implications of Shallower Memory Controller Transaction Queues in Scalable Memory Systems, Journal of Supercomputing.