This technical report is a longer version of "Fat Caches for Scale-Out Servers" which appears in IEEE Micro.

An Effective DRAM Cache Architecture for Scale-Out Servers

Stavros Volos
Microsoft Research

Djordje Jevdjic
University of Washington

Babak Falsafi
EcoCloud, EPFL

Boris Grot
University of Edinburgh

ABSTRACT

Scale-out workloads are characterized by in-memory datasets, and consequently massive memory footprints. Due to the abundance of request-level parallelism found in these workloads, recent research advocates for manycore architectures to maximize throughput while maintaining quality of service. On-die stacked DRAM caches have been proposed to provide the required bandwidth for manycore servers through caching of secondary data working sets. However, the disparity between provided capacity and hot dataset working set sizes — resulting from power-law dataset access distributions — precludes their effective deployment in servers, calling for high-capacity cache architectures.

In this work, we find that while emerging high-bandwidth memory technology falls short of providing enough capacity to serve as system memory, it is a great substrate for high-capacity caches. We also find the long cache residency periods enabled by high-capacity caches uncover significant spatial locality across objects. Based on our findings, we introduce Scale-Out Cache (soCache), a distributed cache composed of multiple high-bandwidth memory modules. Each soCache module uses a page-based organization that optimizes for spatial locality while minimizing tag storage requirements. By storing the tags in the logic die (in SRAM) of the high-bandwidth memory modules, soCache avoids the prohibitive complexity of in-DRAM metadata in state-of-the-art DRAM caches. In 14nm technology, soCache reduces system energy by 1.4-4.4x and improves throughput by 28-44% over state-of-the-art memory systems.

1. INTRODUCTION

Scale-out datacenters host a variety of data-intensive services, such as search and social connectivity. To concurrently support billions of users, latency-sensitive online services rely on large amounts of memory to avoid disk accesses [11, 73]. Likewise, analytic engines creating user-specific content, such as advertisements and recommendations, rely on massive memory capacity [18] as they need to plow through enormous amounts of data within 100s of milliseconds. While today's datacenters rely solely on DRAM, the emergence of fast NVRAM will further broaden in-memory computing. The ever-growing popularity of the in-memory computing paradigm leads to datacenter deployments in which memory accounts for a big share of the datacenter's total cost of ownership (TCO) [8, 52, 65].

Optimizing for datacenter's TCO calls for architectures that maximize compute density. In response, processor vendors have turned to customized architectures for datacenters. Following a considerable amount of research, identifying the requirements of scale-out workloads and indicating that these workloads benefit from thread-level parallelism and high core counts [27, 48, 61], industry has started employing specialized manycore processors (e.g., Cavium ThunderX [32] and EZchip Tile-MX [33]) due to the substantial performance and TCO advantages offered by specialization.

Memory systems in scale-out servers are of paramount importance as they have to sustain the vast bandwidth demands of manycore CMPs [45, 61]. Recent advances in *on-die stacked DRAM* technology [10, 59] can eliminate the bandwidth bottleneck that plagues conventional DRAM. As this technology is capacity-limited due to thermal constraints [22], prior research advocates for using it as a cache [44, 45, 46, 60, 81] to provide access to secondary data working sets.

Our analysis shows that on-die stacked DRAM caches are unattractive for scale-out servers. We find that memory accesses follow power-law distributions — similarly to enterprise server workloads [62] — with a hot portion of memory (~10%) accounting for the bulk of accesses (65-95%). Thus, while the vast working sets of scale-out workloads are cacheable, *high-capacity* caches (10s of GB) are required given main memory sizes trending toward 100s of GB. The required cache capacities greatly exceed those of *low-capacity* caches, including on-die stacked DRAM caches.

This work seeks to develop a scalable, high-capacity, and high-bandwidth memory system for scale-out servers by leveraging emerging *high-bandwidth memory modules* as a high-capacity cache. High-bandwidth interconnect technologies allow for connecting the processor to multiple high-bandwidth memory modules via a silicon interposer [50] forming an *on-package* high-capacity cache, or high-speed serial links [75] forming an *off-package* high-capacity cache.

We draw insights on design of high-capacity caches by characterizing their access patterns, finding that they exhibit predominantly coarse access granularity, with 93% of all accesses going to memory regions with high access density.[1] In contrast, only 68% of accesses in low-capacity caches are coarse-grained. The reason why high-capacity caches exhibit higher access density is due to the longer residency times of data in the cache, which increases the likelihood

[1] A *high access density* region is one in which most (over 75%) of cache blocks comprising the region are accessed within the cache lifetime of the first block to be accessed within the region [93].

for unrelated spatially-proximate objects (e.g., adjacent hash bucket headers) to be accessed before an eviction.

Based on the insights of the characterization, we introduce MeSSOS, a *Memory System* architecture for *Scale-Out Servers*. MeSSOS employs a set of off-package high-bandwidth memory modules as a high-capacity hardware-managed scale-out cache (soCache). Each soCache module is backed by a number of conventional DIMMs. Together, an soCache module and its associated DIMMs represent a MeSSOS building block. Growing (cache and memory) capacity and bandwidth to accommodate more cores and larger datasets is simply a matter of adding more building blocks.

The architecture of an soCache module is informed by the memory access behavior of scale-out workloads. Due to the prevalence of coarse-grained accesses and long cache residency times stemming from skewed access distributions, soCache employs a page-based organization. Tags are stored in the logic die of each soCache module, taking advantage of the under-utilized die space and the low storage requirements of page-grain tags (5 MB for tags in a 4 GB soCache module). The page-based design with in-SRAM tags offers fundamental storage and design complexity advantages over state-of-the-art DRAM cache proposals that suffer from high tag and metadata overheads that mandate in-DRAM storage.

Summarizing, our contributions are as follows:

- Scale-out workloads exhibit skewed memory access distributions, but high-capacity caches are needed to capture the hot working sets. Due to disparity between cache capacity and hot working set sizes, on-die stacked DRAM caches exhibit low temporal reuse — e.g., a cache of 1 GB filters 45% of accesses in systems with 100s of GB. The combination of poor cache performance and technological complexity makes on-die stacked DRAM caches unattractive for servers.
- High-capacity caches — enabled by the emergence of high-bandwidth memory modules — are fundamentally different than on-die stacked DRAM caches. High-capacity caches access memory at coarse granularity, making the case for simple page-based organizations. Such organizations avoid the high metadata storage requirements and excessive design complexity of state-of-the-art DRAM caches [44, 45, 46, 60, 81].
- soCache – a practical, scalable, and effective cache architecture for scale-out servers. soCache uses a page-based organization with in-SRAM tags across multiple high-bandwidth memory modules to achieve high capacity in a practical and scalable manner. In doing so, soCache filters the bulk of accesses (84% on average), thus minimizing bandwidth pressure to main memory.

Our system-level evaluation shows that MeSSOS matches the memory requirements of scale-out servers, and improves system energy efficiency by 1.7-4.4x and throughput by 1.3-2.6x compared to traditional and emerging memory systems.

2. BACKGROUND & MOTIVATION

In this section, we examine the memory requirements of emerging scale-out servers and also review the features of emerging DRAM technologies.

Table 1: Requirements of one scale-out server.

Year	Processor		Memory System	
	Cores	Bandwidth	Bandwidth	Capacity
2015	96	115 GB/s	288 GB/s	384 GB
2018	180	216 GB/s	540 GB/s	720 GB
2021	320	384 GB/s	960 GB/s	1280 GB

2.1 Scale-Out Server Requirements

Large-scale online services distribute and replicate their datasets across many servers to ensure in-memory processing and meet tight tail latency requirements. In response, processor and system vendors resort to manycore processors [32, 89, 92, 95] and buffer-on-board chips [17, 40, 92] to boost per-server throughput and memory capacity. Increasing computing density and per-server memory capacity allows datacenter operators to deploy fewer servers for the same throughput and dataset, thus lowering cost [17, 31].

We quantify the memory bandwidth and capacity requirements of a scale-out server for various manufacturing technologies (denoted by their year) in Table 1. The modeled organization is similar to emerging manycore servers, such as Cavium's 48-core processor fabricated in the 28nm technology [32]. Our configuration maximizes throughput by integrating maximum number of cores for a given die area and power budget of 250-280 mm^2 and 95-115 Watt. Our models for future technologies are derived from ITRS [41].

We estimate required memory capacity by examining various datacenter deployments. System vendors anticipate the need for cost-effective high server memory capacity, and sell *extended memory technology*, which relies on multiple buffer-on-board chips. Data analytic engines are provisioned with 2-8 GB per core [18], web search engines deploy 64 GB for 16 cores [80] while web and streaming servers require 1-2 GB per core [20, 27]. Thus, we assume that 4 GB of per-core memory can be deployed cost-effectively.

We measure processor's off-chip bandwidth demands by scaling the per-core bandwidth consumption with the number of cores available on the chip. We measure per-core bandwidth by simulating a 16-core server (Section 5.2 details the configuration). Our study shows that their per-core bandwidth ranges from 0.4 GB/s to 1.2 GB/s corroborating prior work [45]. Peak bandwidth demands are 115 GB/s in 2015, 216 GB/s in 2018, and 384 GB/s in 2021.

High bandwidth utilization levels can adversely impact end-to-end memory latency due to heavy contention on memory resources. As performance of scale-out applications is characterized by tail latencies, memory latency and queuing delays must be minimized. Thus, system designers over-provision memory bandwidth to ensure a low utilization (< 40%) to avoid queuing [27]. As such, memory systems need to supply 288 GB/s in 2015, 540 GB/s in 2018, and 960 GB/s in 2021. Such requirements exceed the capabilities of conventional DRAM systems by 5.5-7.5x.

To summarize, our analysis shows that by 2021 scale-out servers will need a memory system that provides 1 TB/s of memory bandwidth and over 1 TB of capacity while respecting tight blade-level power budgets.

2.2 Emerging DRAM Technologies

Stacked DRAM [10, 59] can provide an order of magnitude higher memory core bandwidth than conventional DRAM due to dense through-silicon vias. It also offers low latency and low DRAM energy per access due to reduced wire spans and smaller page sizes. However, existing deployment options for stacked DRAM fail to satisfy the joint capacity, bandwidth, and power requirements mandated by scale-out servers. Next, we review the deployment options for stacked DRAM and their respective limitations.

On-Die and On-Package Stacked DRAM. Through-silicon vias provide high-bandwidth connectivity between the processor and on-die stacked DRAM. Thermal constraints, however, limit the number of DRAM stacks that can be integrated on top of the processor [10, 22], confining On-Die Stacked DRAM to several 100s of MB — i.e., two-to-three orders of magnitude smaller than the memory capacity demands of servers. Similarly, the high cost of big packages and area-intensive silicon interposers limit the number of stacked DRAM modules in On-Package Stacked DRAM systems. When combined with the thermally-constrained capacity of 1-2 GB per stacked memory module [50], an On-Package DRAM solution fails to provide the requisite memory capacity for scale-out servers.

Off-Package Stacked DRAM. High-speed serial interfaces have the potential to break the bandwidth wall by connecting the processor to multiple Off-Package Stacked DRAM modules. Serial interfaces employ point-to-point differential links with excellent signal integrity as opposed to conventional DDR interfaces, which employ parallel buses and single-ended signaling. The high signal integrity combined with clock recovery of embedded clocking allows for achieving an order of magnitude higher data rates than DDR. For instance, the recently announced HMC utilizes two serial links clocked at 10 Gbps to provide up to 80 GB/s [75] with the same number of pins as a DDR3 channel, which can deliver only 12.8-17 GB/s.

Off-package systems deliver much greater capacity than the systems described above as the number of stacked DRAM modules is no longer limited by any area constraints. However, there are two main factors that prevent such systems from replacing conventional DRAM. First, serial channels impose high idle power as keep-alive packets must be sent at frequent intervals to maintain lane alignment across the channel's lanes [1]. Even at periods of low utilization, high sleep and wake-up times prevent these channels from going to power-efficient sleep states [1, 3, 39, 79]. Second, thermal constraints limit the number of stacked layers per module and necessitate an entire blade-level network of these modules for a big-memory server. Such a network comes at the cost of high idle power consumption due to the use of many serial interfaces (~10x higher than conventional DRAM), and high end-to-end memory latency resulting from a multi-hop chip-to-chip interconnect (Section 6.1).

2.3 State-of-the-art DRAM Caches

Given the disparity between memory capacity requirements and the capacity provided by emerging DRAM technologies, most proposals advocate employing stacked DRAM as a cache to filter accesses to main memory. State-of-the-art cache proposals leveraging mainly On-Die Stacked DRAM have to contend with relatively high miss rates due to its limited capacity. As a result, they are primarily optimized for low cache-memory bandwidth utilization through block-based organizations [60, 81], or sector-based organizations with footprint-aware mechanisms that predict the blocks that will be accessed within a page [45], or address-correlated filter-based caching mechanisms [46].

Unfortunately, block-based and sector-based organizations come with high tag overhead and high design complexity, making such cache designs impractical. For instance, state-of-the-art block-based [81] and sector-based footprint-aware [45] caches require 4 GB and 200 MB of tags, respectively, for a capacity of 32 GB. Due to the prohibitive tag array overhead, recent proposals advocate employing the tag array in DRAM [44, 60, 81]. In-DRAM tag arrays, however, require substantial engineering effort, making state-of-the-art caches less attractive. In addition, footprint-aware caches [44, 45] rely on instruction-based correlation requiring the core-to-cache transfer of the program counter for all memory references, which further increases the design complexity.

2.4 Summary

Memory systems for scale-out servers need to provide high capacity to guarantee fast access to vast datasets, deliver high bandwidth to satisfy the excessive demands of manycore CMPs, and minimize their power footprint. Unfortunately, existing proposals leveraging emerging DRAM technologies are ineffective for big-memory servers.

3. MEMORY ACCESS CHARACTERIZATION OF SCALE-OUT SERVERS

High-bandwidth memory — both on-package and off-package stacked DRAM — modules are an ideal building block for a high-capacity high-bandwidth cache. However, state-of-the-art DRAM caches are hindered by the need to keep metadata in DRAM. In this section, we study the application characteristics that enable designing an effective, practical, and scalable cache architecture.

3.1 Temporal Characterization

We examine the memory access distribution of scale-out applications (i.e., fraction of accesses that go to a memory object) by looking at the characteristics of the dominant types of memory accesses.

Dataset accesses. A high fraction of memory accesses go to memory-resident datasets. We examine the popularity and dataset access distribution of search query terms (AOL [74]), tweets (Twitter), videos (Youtube), and web pages (Wikipedia) based on publicly available data. Figure 1 plots the probability for a dataset object to be referenced as a function of popularity. As shown in the figure, the dataset access distribution of various web services is highly skewed with a small set of dataset objects (10-20%) contributing to most of the dataset accesses (65-80%). Other examples include:

- **Analytics.** Dataset accesses in analytics are highly skewed as recent data are more frequently accessed than archived data while analysts often utilize a small dataset, which results in a refined dataset [19, 76].

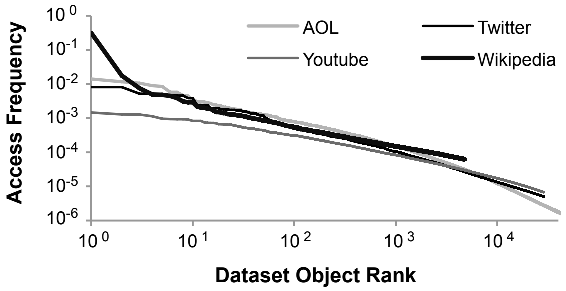


Figure 1: Dataset accesses in web services exhibit power-law distribution. Please note that power-law relationships show linear trends when plotted in log-log scale.

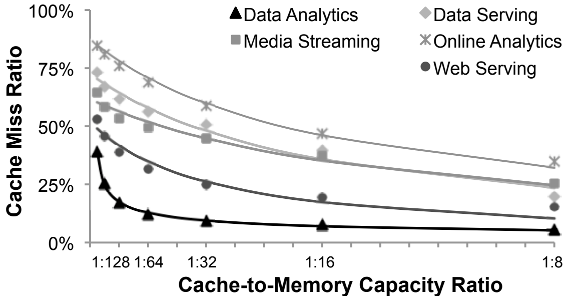


Figure 2: Miss ratio for various Cache-to-Memory Capacity Ratios. Lines denote x-shifted power-law fitting curves.

- **File servers.** Services hosting a large pool of static pages, such as Wikipedia, follow Zipfian distribution as a set of articles are more popular than the rest [96]. Similar distribution is found in streaming services (e.g., Youtube) [15].

- **Search engines.** Popularity of search query terms follows a Zipfian-like distribution in search engines and private search engines, where a set of events (e.g., celebrity scandals) or items are frequently searched by online users [99].

- **Social networks.** Popular users along with their activity absorb the bulk of user requests. For instance, a small fraction of users and their pictures account for most of the user traffic in picture sharing services, such as Flickr [14]. The Zipfian distribution is found in the broader space of social networks, including Facebook and Twitter [4, 58].

Due to the power-law popularity distribution, dataset accesses in analytic engines [19], data stores [24], object caching systems [58], streaming servers [15], web search engines [99], and web servers [23] exhibit power-law distributions.

Accesses to dynamically allocated memory. Server applications frequently access dynamically allocated memory with high temporal reuse. Examples include:

- **Software caches.** Server applications utilize software structures to cache a set of hot objects (e.g., rows and pages in data stores and compiled script code in web servers) or to speed up object lookups by caching their exact position in memory-resident datasets (e.g., key caches in data stores). As these data structures host data/metadata relevant to the dataset, the distribution of memory accesses going to them will follow the power-law distribution of dataset accesses.

- **Client connections.** Server applications and operating systems (OS) employ various data structures per client connection. While each connection allocates only a few 100s of KB, the large number of concurrent connections results in a footprint of a few GB that dwarfs the capacity of on-chip caches. Examples include: (a) buffers in streaming servers keeping media packets, (b) statistics for tracking quality of service of streaming connections, and (c) OS data structures for keeping the state of active TCP connections [9]. The reuse of these structures is high as they are accessed multiple times over the duration of connections.

- **Other.** Partitioning and dynamically built hash tables can improve the temporal locality of dataset accesses upon iterative computations in graph analysis [86] and join operations in relational database systems.

The skew in object popularity and temporal reuse of dynamically allocated memory is expected to be mirrored in the memory access distribution. To confirm this, we examine the memory access distribution of a 16-core scale-out server. To estimate the hot memory footprint of scale-out applications, we employ a state-of-the-art high-bandwidth cache [81] and measure its miss ratio for various capacities.

Figure 2 plots the cache miss ratio for various Cache-to-Memory Capacity Ratios.[2] The markers denote measurements while contiguous lines show x-shifted power-law fitted curves. The figure shows that memory accesses in scale-out servers are skewed so that 6.25-12.5% of the memory accounts for 65-95% of accesses. Our results corroborate prior work showing power-law relationships between cache capacity and miss ratio in multi-MB caches for enterprise server workloads [36, 62].

The figure confirms that existing low-capacity caches (left part of the graph), such as on-board [92], on-package eDRAM [54], and on-die stacked DRAM caches cannot exploit temporal locality in scale-out servers.[3] In extreme cases, such as Data Serving and Online Analytics, on-die stacked DRAM caches are bandwidth-constrained (Section 6) with less than 40% of memory accesses filtered. We thus conclude that the combination of poor cache performance and technological complexity of die stacking [97] limits the usefulness of on-die stacked DRAM caches in servers.

3.2 Spatial Characterization

Scale-out applications frequently operate on large objects (e.g., database rows or memory-mapped files), and hence exhibit a high incidence of coarse-grained accesses [93]. To

[2] The Cache-to-Memory Capacity Ratio is a representative metric of the effective cache capacity (i.e., cache capacity with respect to a given memory capacity) as scaling datasets and memory capacity reduces the effectiveness of caches proportionally [35].

[3] In 2015, on-die stacked cache capacity ranges from 0.5 GB to 1 GB whereas main memory varies from 64 GB to 256 GB, resulting in Cache-to-Memory Capacity Ratio of 1:512 to 1:64.

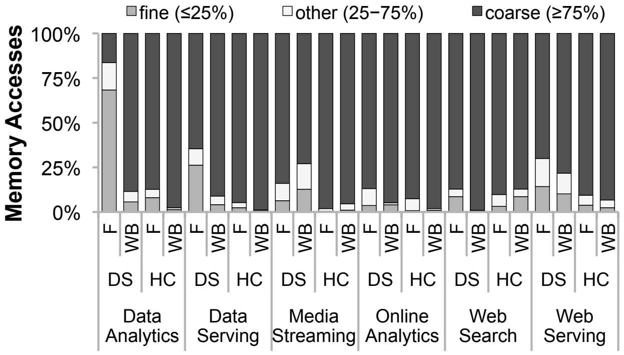


Figure 3: Granularity at which page-sized blocks are fetched (F) from and written back (WB) to DRAM for Die-Stacked (DS) and high-capacity cache (HC) of 1:128 and 1:8 Cache-to-Memory Capacity Ratio, respectively.

allow for retrieving an object in sub-linear time, dataset objects are pinpointed through pointer-intensive indexing data structures, such as hash tables and trees. For instance, NoSQL data stores and object caching systems use a hash table to retrieve data objects. While data objects are accessed at coarse granularity, finding them requires performing a sequence of fine-grained pointer-intensive operations. Thus, a non-negligible fraction of accesses are fine-grained [93].

We examine the granularity at which high-capacity (HC) caches access memory by measuring the access density at which page-sized lines are fetched from and written back to memory in Figure 3. We define *page access density* as the fraction of cache blocks within a page accessed between the page's first access and the page's eviction from the cache [93]. We use a page of 2 KB as it reduces the tag array size significantly with limited tolerance for over-fetch (Section 6.3). The three segments correspond to fine ($\leq$ 25%), coarse ($\geq$ 75%), and other (25–75%) granularity. For comparison, we include a low-capacity cache, labeled as Die-Stacked (DS).

Our analysis shows that Die-Stacked exhibits bimodal memory access behavior [45, 93] — fine-grained and coarse-grained accesses account for 21% and 68% of accesses, respectively. While coarse-grained accesses (which have high spatial locality) are prevalent, the frequent incidence of fine-grained accesses (corresponding to pointer dereferences to non-contiguous memory) must also be accommodated effectively. Due to the limited capacity of on-die stacked DRAM caches, pointer-containing pages show low temporal reuse and are frequently evicted. To avoid massive bandwidth waste in accesses to such pages, state-of-the-art DRAM caches rely on block-based [60, 81] or sector-based footprint-aware [44, 45] organizations that are bandwidth-frugal but carry a high metadata storage cost.

In contrast, high-capacity caches exhibit coarse-grained memory access behavior — 93% of memory accesses on average. This phenomenon is attributed to two reasons. First, the lifetime of pages in the cache is on the order of 10s to 100s of milliseconds. As a result, pages containing a collection of fine-grained objects (e.g., hash bucket headers) can enjoy spatial locality uncovered through long cache residency times, which stem from skewed access distributions. Second, low-access-density pages containing pointer-intensive indexing structures with good temporal reuse (e.g., intermediate tree nodes) are preserved across accesses.

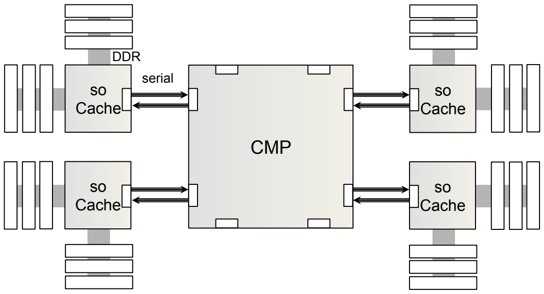


Figure 4: MeSSOS overview.

3.3 Summary

Our study shows that high-capacity caches are needed to capture the skewed memory access distributions of scale-out servers. We also find that the improved spatio-temporal behavior of high-capacity caches offers an opportunity to use a simple page-based cache organization, thus avoiding the storage and complexity overheads associated with state-of-the-art block-based and sector-based designs.

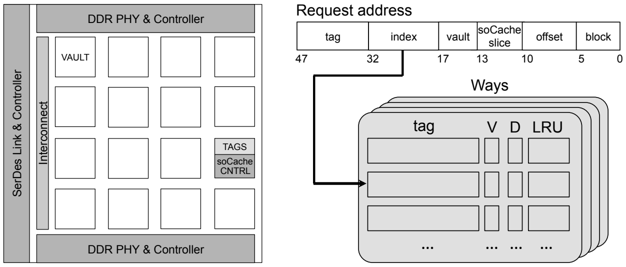
4. MEMORY SYSTEMS FOR SCALE-OUT SERVERS

We present MeSSOS, a *Memory System* architecture for *Scale-Out Servers*, which leverages multiple off-package stacked DRAM modules as a *Scale-Out Cache* (soCache) in front of conventional DIMMs. Figure 4 shows the design overview. In MeSSOS, an on-board building block consists of a soCache slice fronting a set of conventional DIMMs. This design allows capacity and bandwidth to be seamlessly scaled with the number of building blocks. Next, we examine the soCache architecture and its integration with the rest of the system.

4.1 soCache Architecture

MeSSOS utilizes multiple off-package stacked DRAM modules as a high-capacity scale-out cache, which exploits the skewed memory access distributions of scale-out workloads. To avoid communication between soCache slices and to minimize the number of serial links, memory addresses are statically interleaved across the soCache slices. Figure 5a shows the organization of a soCache slice. As shown in the figure, stacked DRAM modules are internally organized as a set of vaults (e.g., 16 vaults of 256 MB each), which are connected to the serial link controller via a fast crossbar [75].

Cache organization. soCache uses a page-based organization leveraging the observation that high-capacity caches uncover spatial locality that is beyond the temporal reach of lower-capacity caches. The page-based design not only naturally captures spatial locality, but also minimizes metadata storage requirements over block-based and footprint-predicting designs. The page-based design also reduces dynamic DRAM energy by minimizing the number of DRAM row activates, which are the most energy-consuming operation in a conventional DRAM architecture [45, 93].



(a) soCache's logic die (b) Tag array architecture

Figure 5: The organization of a soCache slice.

Based on page-size sensitivity analysis in Section 6.3, we find that a page size of 2 KB offers a good trade-off between tag array size and bandwidth overhead stemming from overfetch. We also observe that low associativity (4-way in the preferred design) is sufficient for minimizing the incidence of conflicts while also reducing tag and LRU metadata costs.

Tag array. The page-level organization reduces the tag array overhead to a few MB per slice. For instance, a soCache of 32 GB, consisting of eight 4GB slices, requires 5 MB of tags per slice, or $20mm^2$ in 40nm technology.[4] The small tag array size allows us to embed it in the logic die of the off-package memory modules comprising the soCache. These logic dies are under-utilized, typically housing per-vault memory controllers, a switch fabric, and off-chip I/O interfaces and controllers. Together, these components occupy ~$70mm^2$ in 40nm technology[5] leaving sufficient room for the tags on a typical ~$100-120mm^2$ logic die [79]. Compared to conventional HMCs hosting four SerDes interfaces, our customized HMC hosts only one SerDes interface (of area ~$9mm^2$ [47]), freeing up area resources for the DDR interfaces (of area ~$10mm^2$ each [53]) and the tag array.

To enable low tag lookup latency, we distribute the tag array across the high-bandwidth memory module, placing each tag array slice beneath a vault. Each tag slice requires only 320 KB and 3-4 cycles of access latency. Low associativity and small in-SRAM tags allow for searching the ways in parallel at small latency and energy overhead.

An important advantage provided by in-SRAM tags as compared to in-DRAM tag designs is in the cache miss detection time. With a 20ns DRAM core access latency (~50 cycles), in-SRAM tags offer a 10x reduction in miss detection time over in-DRAM tags.

4.2 soCache-Main Memory Interface

The off-package high-bandwidth memory modules provide not only the functionality of a cache, but also the communication bridge between processor and main memory. Memory requests that miss in the soCache are forwarded directly to local memory modules. To do so, the soCache slice integrates DDR controllers to control the local DDR channels, requiring the implementation of both the DDR protocol and PHY in the soCache's logic die (Figure 5a).

The DDR controllers employ FR-FCFS open-row policy [84] with page-level address interleaving. We map an entire soCache's page-sized cache line to one DRAM row by using the following addressing scheme *Row:ColumnHigh:Rank: Bank:LocalChannel:soCacheSlice:ColumnLow:WordOffset*, where ColumnHigh is 2 bits and ColumnLow is 8 bits. To guarantee that requests missing in a soCache slice are served by local DRAM, the mapping scheme interleaves addresses across local channels using the least significant vault bit.

4.3 Processor-soCache Interface

The processor is connected to the soCache via point-to-point serial links. Both processor and soCache slices implement simple controllers to orchestrate corresponding communication (Figure 4). The controllers consist of a pair of queues to buffer incoming and outgoing packets, and a SerDes interface. Processor-side controllers serialize outgoing requests into packets, before routing them to the soCache slice based on corresponding address bits (Figure 5a), and deserialize incoming data and forwards them to the last-level cache. A soCache-side controller deserializes incoming memory requests and forwards them to the vault's soCache controller based on corresponding address bits (Figure 5a), and serializes outgoing data into packets and forwards them to the processor.

Serial link. As scale-out workloads exhibit variable read-write ratios [93], each serial link consists of 16 request lanes and 16 response lanes. Thus, a serial link requires ~70 pins (control and double-ended signaling for request/response lanes) as opposed to a DDR channel, which requires ~150 pins (control and address, command, and wide data buses). The lower number of per-serial-link pins allows for integrating a high number of processor-side SerDes channels without increasing the number of the processor's pins compared to a processor with DDR channels, thereby keeping the cost associated with the processor's packaging constant.

4.4 System-Level Considerations

Feasibility. MeSSOS implements all necessary functionality in the logic layer of the off-package memory modules without any modifications to the rest of the system. The reduced number of employed SerDes interfaces frees up power resources for the required tag array and low-frequency and low-utilized DDR interfaces. Our estimates show that the power consumption of each soCache slice is lower than 8 Watt and lies within the power range of conventional off-package modules [43, 79].

Scalability. MeSSOS delivers high memory capacity in a scalable manner while relying on cost-effective conventional DIMMs. MeSSOS distributes the required number of DDR channels and their pins across multiple soCache modules as opposed to a single processor chip. This approach resembles that of conventional buffer-on-board systems [17, 40, 92], which employ on-board chips to boost memory capacity in a cost-effective way. In contrast to these systems, MeSSOS does not require additional on-board chips as the functionality of buffer chips is implemented in the logic layer of the soCache modules.

Cost. MeSSOS achieves substantial system cost savings due to lower acquisition and operating costs. By providing

[4] Tag array entries are 20-bit;15 bits for the tag, 2 page-level valid and dirty bits, and 3 bits for maintaining the pseudo-LRU tree.

[5] Estimated by scaling die micrographs [47, 53, 88, 89].

Table 2: Systems configuration.

System	2015 (22nm)	2018 (18nm)	2021 (14nm)
CMP	96 cores, 3-way OoO, 2.5GHz	180 cores, 3-way OoO, 2.5GHz	320 cores, 3-way OoO, 2.5GHz
LLC	24 MB	45 MB	80 MB
Memory	384 GB	720 GB	1280 GB
DDR	4 DDR-1600	5 DDR-2133	6 DDR-2667
	Memory latency: 55ns including off-chip link (15ns) and DRAM core (40ns)		
HBMM	8 32-lane @ 10Gbps	10 32-lane @ 15Gbps	12 32-lane @ 20Gbps
	Memory latency: hop-count*35ns (SerDes & pass-through logic) + 20ns (stacked DRAM access)		
BOB	8 32-lane @ 10Gbps	10 32-lane @ 15Gbps	12 32-lane @ 20Gbps
	16 DDR-1600	20 DDR-2133	24 DDR-2667
	Memory latency: 95ns including SerDes & buffer (40ns), buffer-DDR link (15ns) and DRAM core (40ns)		
Die-Stacked	Cache: 1GB	Cache: 2GB	Cache: 4GB
	Hit latency: ~20ns including predictor lookup and stacked DRAM access (20ns)		
	Miss latency: ~55ns including predictor lookup and off-chip DRAM access (55ns)		
	Off-chip: 4 DDR-1600	Off-chip: 5 DDR-2133	Off-chip: 6 DDR-2667
MeSSOS	CMP-Cache: 8 32-lane @ 10Gbps	CMP-Cache: 10 32-lane @ 15Gbps	CMP-Cache: 12 20-lane @ 20Gbps
	Cache: 8x4GB	Cache: 10x8GB	Cache: 12x8GB
	Tag lookup latency: 35ns including SerDes (30ns) and distributed tag array lookup (5ns)		
	Hit latency: 55ns including tag lookup (35ns) and stacked DRAM access (20ns)		
	Miss latency: 95ns including tag lookup (35ns) and off-chip DRAM access (60ns)		
	Cache-Memory: 16 DDR-1066	Cache-Memory: 20 DDR-1066	Cache-Memory: 24 DDR-1066

the required bandwidth and capacity for a scale-out server, MeSSOS maximizes server throughput, and hence reduces the number of servers required for the same throughput requirements and dataset size. MeSSOS also lowers memory energy consumption by minimizing the static power footprint of its underlying memory interfaces. As MeSSOS employs stacked DRAM modules as a cache, it (a) bridges the processor-bandwidth gap with a minimal number of power-hungry high-speed serial interfaces, (b) efficiently utilizes available serial links and amortizes their high idle power consumption, and c) filters a high degree of memory accesses, and thus infrequent main memory accesses can be served by under-clocked conventional DIMMs.

5. EXPERIMENTAL METHODOLOGY

We evaluate the system performance and energy efficiency of MeSSOS and various memory systems using a combination of cycle-accurate full-system simulations, analytic models, and technology studies.

5.1 Scale-Out Server Organization

We model chips with an area of 250-280 mm^2, and a power budget of 95-115 Watt. We use the scale-out processor methodology to derive the optimal ratio between core count and cache size in each technology [61]. Cores are modeled after Cortex-A15, a 3-way OoO core, resembling those used in specialized manycore CMPs [28, 32, 33, 61].

Table 2 summarizes the details of evaluated designs across technology nodes. For a given technology node, the processor configuration and memory capacity are fixed. We evaluate the following memory subsystems: (i) DDR-only memory; (ii) buffer-on-board (BOB) system [17, 21, 40, 92], which relies on on-board chips to boost bandwidth and capacity through additional DDR channels, but at the cost of an increase in end-to-end memory latency and energy consumption due to serial links (between processor and BOB) and intermediate buffers; (iii) high-bandwidth memory modules (HBMM), which employs off-package memory modules connected with high-speed serial links. HBMM employs a tree network topology to reduce the number of hops in the point-to-point memory network (average and maximum number of network hops is three and four, respectively); (iv) Die-stacked cache with a block-based organization [81] that maximizes effective capacity and minimizes off-chip bandwidth. The cache is backed by DDR-based main memory; and (v) MeSSOS that combines HBMM modules as a scalable soCache with DDR-based memory. Because of the high degree of bandwidth screening provided by soCache, the DDR channels are clocked at half of their peak frequency to reduce static power.

Each serial link in BOB, HBMM, and MeSSOS consists of 16 response and 16 request lanes as scale-out workloads exhibit variable read-write ratios [93].

5.2 Performance and Energy Evaluation

System performance. For performance evaluation, we perform full-system simulation using Flexus [94]. Flexus models the SPARC v9 ISA and runs unmodified operating systems and applications. Flexus extends the Simics functional simulator with timing models of out-of-order cores, caches, on-chip protocol controllers and interconnect, and DRAM. We model stacked DRAM and off-chip DRAM performance, by integrating two separate DRAMSim2 instances [85] into Flexus. We parameterize off-chip DRAM based on commercial device specifications [69]. Latency of stacked DRAM is halved compared to off-chip DRAM [60].

Our analysis is based on a wide range of scale-out workloads. The examined workloads, taken from CloudSuite 2.0 [20, 27], include Data Analytics, Data Serving, Media Streaming, Web Search, and Web Serving. We evaluate online analytics running a mix of TPC-H queries on a modern column-store database engine, MonetDB [49].

We run cycle-accurate simulations using the SMARTS sampling methodology [98]. Our samples are drawn over

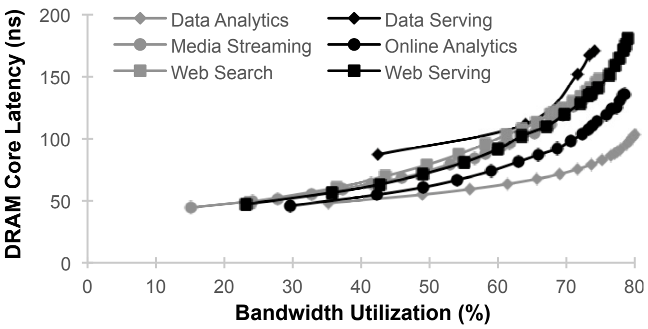


Figure 6: DRAM core access latency sensitivity to bandwidth utilization.

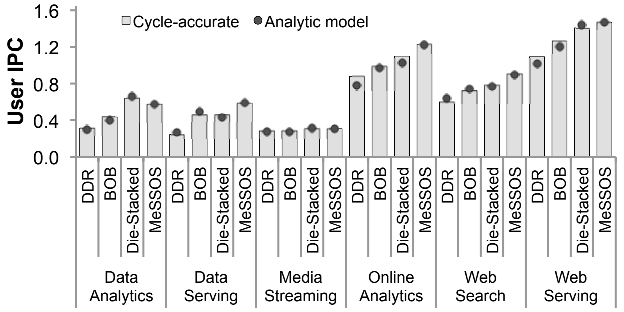


Figure 7: Validation of analytic results.

an interval of 10-30 seconds of simulated time. For each measurement, we launch simulations from checkpoints with warmed caches and branch predictors, and run 800K cycles (2M cycles for Data Serving) to achieve a steady state of detailed cycle-accurate simulation prior to collecting measurements for the subsequent 400K cycles. For performance, we measure User IPC, defined as the ratio of the number of application (user) instructions committed to the total number of cycles (including cycles spent on the operating system); this metric has been shown to reflect system throughput [94].

As simulating processors with 100s of cores and memory capacity of 100s of GB would be prohibitively slow, we augment our simulation-based studies with an analytic model. Our model extends the classical average memory access time analysis [35] to predict per-core performance for a given off-chip memory system; the model is parameterized by simulations results, including core performance, on-chip cache miss rates, and interconnect delay. For off-chip access latency, we include link latency, memory core latency, and queuing delays. To model queuing delays, we run cycle-accurate simulations to measure memory latency for various bandwidth utilization levels for each workload separately (Figure 6).

We validate our analytic models against cycle-accurate full-system simulations of a scaled-down CMP. Because server workloads are request-parallel, performance and memory bandwidth scale with the number of cores. This allows us to validate our model against cycle-accurate simulations of 16-core systems (Table 3), and scale our performance and energy models proportionally. Figure 7 shows that our analytic models (denoted as circles) achieve high prediction accuracy (average error of 5%).

Table 3: Configuration for cycle-accurate simulations.

Parameter	Value
CMP	16 cores, 2.5GHz, 3-way OoO, 60-entry ROB
L1-I/D	64KB, 2-way, 64B blocks 2-cycle load-to-use, 10 MSHRs
LLC	Unified, 4MB, 16-way, 64B blocks 8 banks, 8-cycle hit latency
NOC	CMP: 16x8 crossbar, 5 cycles soCache: 16x3 crossbar, 5 cycles
Memory	soCache: 512 MB-16 GB, DRAM:16-128 GB Off-chip links:15ns/30ns (parallel/serial)

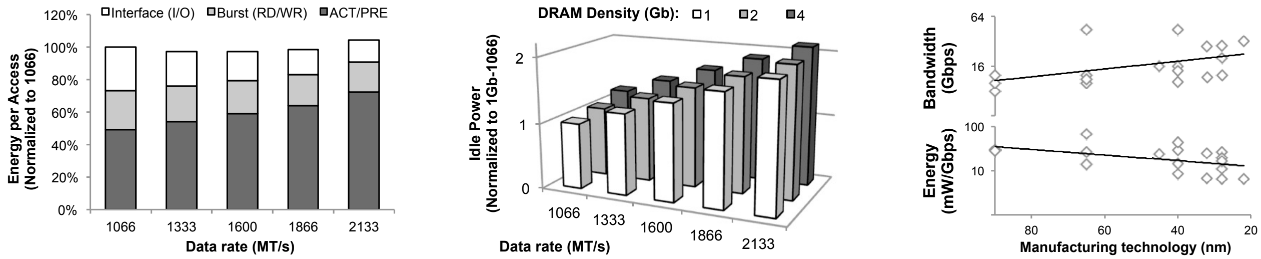
Table 4: System power model in 2015.

Parameter	Value
Core	Peak power: 770mW
LLC	Read/Write energy: .63nJ/.70nJ Leakage power: 750mW per 4MB
Memory Controller	Front-end engine: 0.24mW/Gbps Transaction engine: 1.37 mW/Gbps Physical interface: 1.95 mW/Gbps
DDR-1600 (per 4GB rank 64-byte access)	Idle power: 377mW Activation energy: 19nJ Read/Write energy: 5.2nJ/5.4nJ I/O energy (Read/RRead): 0.6nJ/1.7nJ I/O energy (Write/RWrite): 2.1nJ/2.1nJ
HBMM (4GB)	DRAM array per 64-byte access: 1.9nJ Logic + SerDes: 2.34 + 4.5mW/Gbps
BOB chip & soCache	SerDes, DDR: 4.5, 1.95 mW/Gbps BOB, soCache: 0.24, 1.61 mW/Gbps

Our models are validated for a particular off-chip-bandwidth-to-core ratio. However, the ratio varies across memory systems and technologies. When modeling systems with lower off-chip-bandwidth-to-core ratio than our validated systems and extremely bandwidth-constrained configurations, we estimate core performance under maximum bandwidth utilization (per cycle-accurate simulation results) and scale down system performance by $\frac{ProjectedBandwidthRequirements}{MaximumAvailableBandwidth}$.

For instance, for a 96-core Die-Stacked system with 4 DDR-1600 channels running Data Serving, we estimate system performance as follows. Our cycle-accurate simulation results for a 16-core Die-Stacked system (with one DDR-1333 channel) running Data Serving demonstrate maximum memory bus utilization level of 80%. By projecting off-chip memory bandwidth requirements of the modeled system, we find that 96 cores (at the estimated/modeled core performance) would consume 55 GB/s of off-chip memory bandwidth. However, the modeled off-chip memory system can provide only 41 GB/s (80% of 4 DDR-1600 channels), which is sufficient for only 76 cores. Thus, we scale system performance by a factor of $\frac{41GB/s}{52GB/s} = 76/96 = 0.79$.

Power and energy modeling framework. We use energy consumed per instruction as our energy-efficiency metric.



(a) DDR energy per access as a function of the data rate

(b) DDR idle power as a function of data rate and DRAM density

(c) SerDes bandwidth and energy across manufacturing technologies

Figure 8: Impact of technology scaling on memory technology and memory interfaces.

We develop a custom energy modeling framework to include various system components, such as cores, network-on-chip (NOC), caches, memory controllers, and memory. Our framework, summarized in Table 4, draws on several specialized tools to maximize fidelity through detailed parameter control.

We estimate core power by scaling published measurements of power on a high-IPC workload by the ratio of the actual workload IPC and the reference IPC, assuming that half of the power scales with the workload IPC and the rest is constant (which includes leakage) [2, 56]. We use CACTI to estimate LLC energy and to account for advanced LLC leakage reduction techniques [56, 71]. We use custom models to estimate energy consumed by on-chip network links and switch fabrics derived by public sources [6, 41]. We measure memory controller's power using McPAT [57].

We estimate DDR background power and energy per operation based on Micron models and datasheets [68, 70]. Per JEDEC's specifications and Intel's estimations [5], we anticipate a voltage reduction from 1.5 (DDR3) to 1.2 (DDR4), and various DDR4 optimizations that halve termination energy and reduce refresh power by 20%.

We quantify the energy consumption of a stacked DRAM module using energy measurements reported for the recently announced Hybrid Memory Cube (HMC) [43], including SerDes interfaces, the DRAM arrays, and other logic (vault controllers and crossbar interconnect). Compared to conventional DRAM, HMC reduces energy per access by leveraging through-silicon-via technology [43]. For soCache's logic die, we also consider the memory controller's logic and the DDR physical interfaces.

We model BOB chips to include the power consumption of SerDes (mostly static) and DDR interfaces. We account for buffering incoming requests and outgoing data similar to an integrated memory controller's front-end engine.

5.3 Projection to Future Technologies

To understand the effect of technology scaling on the examined memory systems, we model our systems in 2018 and in 2021. Per ITRS estimates, processor supply voltages will scale from 0.85V (2015) to 0.8V (2018) and 0.75V (2021).

We examine the impact of data rate and memory density on DDR energy. We compute DDR energy per access for different data rates based on Micron's datasheets [68] in Figure 8a. As shown in the figure, energy consumed by RD/WR and I/O operations is higher in slower DDR interfaces due to longer burst periods. In contrast, ACT/PRE energy is higher in faster DDR interfaces due to higher active currents. We also examine the background power of DDR3 devices for different data rates and memory densities [67, 68, 69] in Figure 8b. Static power consumed by DRAM core (leakage) and DLL (active mode) scales linearly with data rate while refresh power scales linearly with density.

Finally, we study the impact of manufacturing technology on power consumption of SerDes interfaces. Figure 8c plots our bandwidth and energy analysis based on published measurements of various SerDes interfaces in different technologies [7, 12, 13, 29, 30, 37, 38, 42, 47, 51, 72, 78, 83, 87, 88, 91]. Our analysis demonstrates that bandwidth and energy scale by 20% and 27% per technology node, respectively.

6. EVALUATION

We compare MeSSOS to conventional and emerging memory systems in terms of system performance and energy efficiency across technology generations.

6.1 Performance and Energy Efficiency Implications

We begin our study with a 96-core CMP in the 22 nm technology. Figure 9 (left) plots the fraction of memory requests that are served by soCache for various Cache-to-Memory Capacity Ratios. The figure demonstrates the ability of MeSSOS to serve the bulk (>95%) of those using its soCache as it exploits temporal locality arising from skewed access distributions (gray bar) as well as spatial locality arising from coarse-grained operations and high cache residency times stemming from skewed access distributions (white bar).

The figure (right) illustrates the DDR bandwidth consumption compared to the baseline system without a cache. As expected, DDR bandwidth savings increase with bigger caches. By capturing the hot working set, soCache is able to absorb 65-95% of memory traffic for a cache size of 12.5% of the memory size (1:8), thereby reducing DDR bandwidth consumption by 3-20x. The light gray bars illustrate the extra traffic generated due to coarse-grain transfers between soCache and the DIMMs. The absolute increase in traffic is small (3% on average) as most accesses are coarse-grain (Section 3.2). For the rest of the evaluation, we use 1:8 Cache-to-Memory Capacity Ratio, unless stated otherwise.

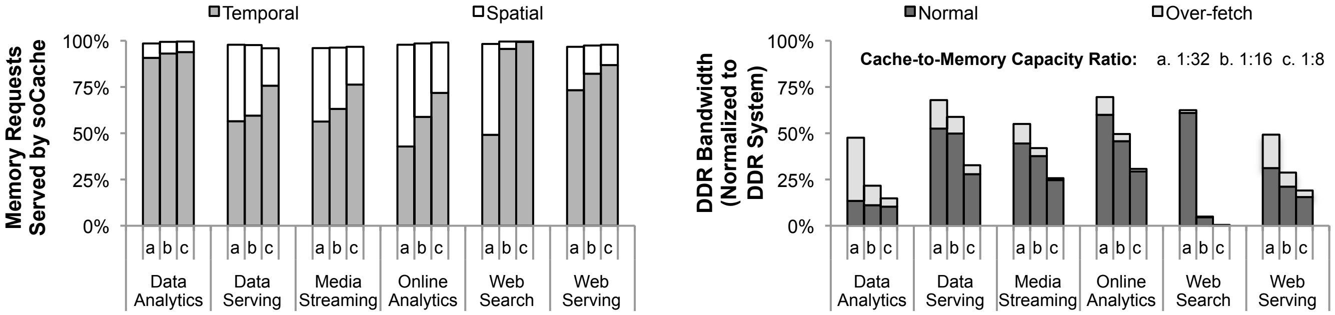


Figure 9: MeSSOS effectiveness for various Cache-to-Memory Capacity Ratio: (a) 1:32, (b) 1:16, and (c) 1:8.

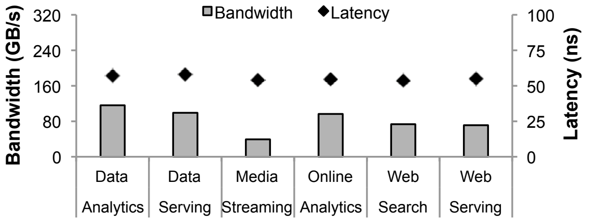


Figure 10: Bandwidth and memory latency in MeSSOS.

Memory performance. In Figure 10, we study the end-to-end memory latency, and the off-chip bandwidth consumption by measuring the bandwidth between the processor and soCache. MeSSOS's off-chip bandwidth consumption ranges from 42 GB/s (Media Streaming) to 114 GB/s (Data Analytics) for a 96-core CMP, which correspond to modest off-chip bandwidth utilization levels (only 14-38%). Due to low link and memory bandwidth utilization levels and the high hit ratio in soCcahe, MeSSOS achieves low queuing times, and fully leverages the low core latency of emerging high-bandwidth memory modules. In doing so, MeSSOS provides both high bandwidth and low latency.

System performance. Figure 11 compares MeSSOS to the baseline system (DDR) as well as high-bandwidth memory modules (HBMM), buffer-on-board (BOB), and Die-Stacked systems, configured as in Table 2.

Because DDR provides insufficient memory bandwidth, its system performance is significantly hurt. BOB and HBMM improve performance over DDR by 49% and 33%, respectively, as they provide sufficient bandwidth to the processor. However, the increase in bandwidth comes at the cost of higher end-to-end memory latency. The BOB system adds an extra 40 ns due to the serial link and the intermediate buffer while the HBMM system requires a point-to-point memory network, which adds a latency of 35 ns per network hop (serial link and pass-through logic). Because HBMM accesses are frequently multi-hop, BOB outperforms HBMM by 12%. Our analysis also shows that on-board SRAM caches found in some BOB systems [92] exhibit low temporal locality (a hit ratio of only 25% on average), and hence provide negligible performance gains.

MeSSOS outperforms all these systems due to its ability to provide high bandwidth at low latency. Compared to the DDR system, MeSSOS improves system performance by 2x on average. MeSSOS outperforms BOB and HBMM by 28% and 43%, respectively, due to lower memory latency.

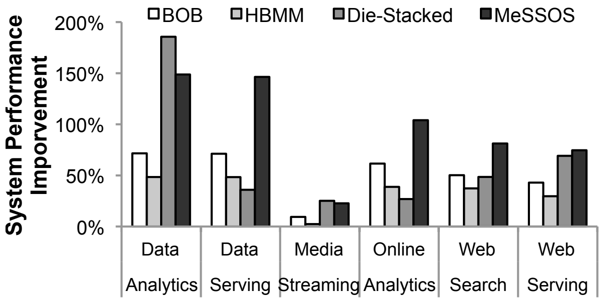


Figure 11: System performance of various memory systems.

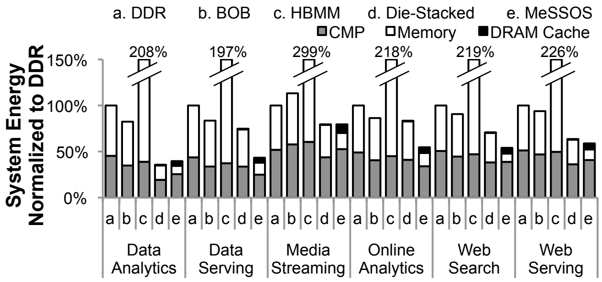


Figure 12: System energy breakdown.

MeSSOS outperforms Die-Stacked by 23% due to lower off-chip bandwidth pressure, resulting from its greater cache capacity which filters a higher fraction of DDR accesses. On average, MeSSOS filters 84% of DDR accesses as compared to 45% in Die-Stacked. For Data Serving and Online Analytics, MeSSOS outperforms Die-Stacked by 81% and 61%, as Die-Stacked is bandwidth-constrained due to is inability to reduce off-chip bandwidth consumption (filters only 38% and 13% of accesses). One exception is Data Analytics where memory accesses are extremely skewed (Figure 2), and hence Die-Stacked achieves high hit ratio (87%) and outperforms MeSSOS due to lower cache access latency.

System energy. Figure 12 plots system energy for the examined systems. As the figure shows, BOB reduces system energy by 12% compared to DDR, primarily due to performance gains. HBMM increases system energy by 2.3x compared to DDR. Although HBMM provides sufficient bandwidth to the processor, leading to higher system throughput, it requires a power-hungry multi-hop chip-to-chip network.

MeSSOS reduces system energy by 1.9x, 1.7x, and 4.3x compared to DDR, BOB, and HBMM. As bulk of the ac-

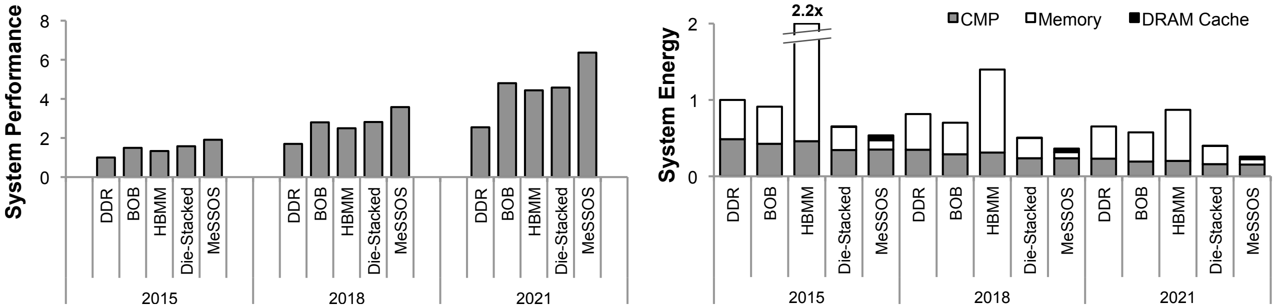


Figure 13: System performance and energy consumption for various technologies normalized to DDR (2015).

cesses are served by soCache, MeSSOS exploits the low-energy access of stacked DRAM modules, thus reducing memory energy significantly. Furthermore, MeSSOS enforces coarse-grain data movement between soCache and DRAM, thus amortizing energy-intensive DRAM row activates. Compared to Die-Stacked, MeSSOS reduces system energy by 23% due to a lower DDR energy footprint resulting from lower off-chip bandwidth consumption.

6.2 Projection to Future Technologies

We study the effect of technology scaling on MeSSOS in 14 nm (2018) and 11 nm (2021) technologies. Figure 13 plots the system performance and system energy consumption per operation averaged across the applications and normalized to DDR in 2015. MeSSOS leverages the abundant bandwidth provided by the SerDes technology, increasing system throughput almost linearly with the number of cores. This near-perfect scalability increases system throughput by 3.7x and 6.6x in 2018 and 2021, respectively, compared to DDR (2015).

Due to poor scalability of DDR interfaces, the bandwidth gap between DDR-based systems and the processor is increasing rapidly. As a result, MeSSOS's performance improvement over DDR and Die-Stacked increases across technologies. In particular, MeSSOS improves system performance by 2.3x (2018) and 2.7x (2021) over DDR, and by 30% (2018) and 43% (2021) over Die-Stacked.

In terms of energy efficiency, the energy footprint of a DDR module increases over technologies due to an increase in static power of their active interfaces. Because MeSSOS employs under-clocked DDR modules, its total energy footprint increases by only a small factor. As a result, MeSSOS reduces system energy by 1.7x in 2015, 2x in 2018, and 2.6x in 2021 compared to DDR and BOB, and by 23% in 2015, 40% in 2018, and 60% in 2021 compared to Die-Stacked. Compared to HBMM, MeSSOS reduces system energy by 4-4.4x due to fewer serial links.

6.3 Comparison to Footprint Cache

We evaluate Footprint Cache [45], the state-of-the-art bandwidth-mitigation technique for DRAM caches in servers. Footprint Cache records the footprint of a page (defined as the set of accessed cache blocks within the page) and transfers only the recorded footprint in future page allocations, thereby lowering cache-memory bandwidth consumption. To do so, Footprint Cache introduces high-storage overhead due to block-level metadata (e.g., 200MB for a cache of 32 GB), requiring a complex in-DRAM metadata implementation [44].

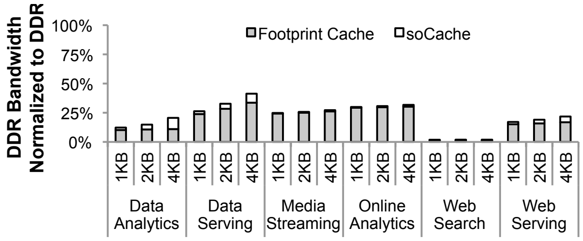


Figure 14: soCache compared to Footprint Cache. The figure also illustrates soCache's sensitivity to page size.

As shown in Figure 14, Footprint Cache achieves lower DDR traffic than soCache (1.1-1.23x on average) but at excessive tag storage overhead – e.g., 5x more than an soCache with a 2 KB page size. Thanks to soCache's high filtering ratio, the overhead amounts to 2% of total off-chip bandwidth. Thus, we conclude that high-capacity page-based caches do not require bandwidth optimizations obviating the need for in-DRAM metadata.

7. DISCUSSION

QoS. MeSSOS captures the hot dataset in soCache. However, the extra level in the memory hierarchy might increase the response latency of infrequent queries that access uncached portions of the dataset. In practice, the effect is limited because accesses to coarse-grain objects dominate traffic (Section 3.2, [93]). By using a page-based organization, MeSSOS amortizes the extra latency of infrequent misses (e.g., pointer chasing to retrieve cold objects) over multiple cache hits upon coarse-grain operations on retrieved objects.

Non-Volatile RAM (NVRAM). NVRAM can enable high memory capacity at low system cost, acting as a great candidate for either replacing or backing conventional DRAM [34, 82]. While the choice of high-capacity memory technology is important for system efficiency and cost, it is largely orthogonal to our work, which focuses on a different level of the memory hierarchy and seeks to provide the memory bandwidth required for manycore CMPs. Nevertheless, MeSSOS's high-capacity and scalable cache architecture exploits skewed memory access distributions, thus hiding NVRAM's high latency and enabling its integration with the rest of the system without impact on performance.

High-Bandwidth Memory Technology. soCache utilizes multiple off-package stacked DRAM modules. Nevertheless, our insights on high-capacity cache design can be exploited for architecting a practical and simple On-Package Stacked DRAM cache. Such a design can lower cache access latency by avoiding chip-to-chip links, but at significant overheads. First, soCache's capacity will be limited by the size of the silicon interposer, hurting soCache's hit rate in systems with high memory capacity. Second, given the pin-count limitations of a single package, it would need to distribute DDR channels across additional buffer-on-board chips so as to afford high memory capacity with conventional DIMMs. These chips would add latency and power overheads on soCache misses.

Cache Coherence. Systems rely on cache coherence for faciliating software development. In doing so, they enforce coherence at the last level of the on-chip memory hierarchy. As soCache is placed at a lower level, support for coherence is not required in single-CMP systems. In multi-CMP systems, however, soCache will need to track coherence in case soCache slices are organized as per-CMP private caches (rather than as a shared cache across all CMPs). Coherence can be tracked at a coarse granularity by augmenting soCache's page-level entries with coherence bits. Maintaining block-level coherence across CMPs is not necessary for scale-out workloads due to negligible data sharing [27].

Sensitivity to LLC size. We employed an LLC of 4 MB per 16 cores as this core-to-cache ratio maximizes throughput for a given die size [61]. Doubling the LLC size reduces bandwidth requirements by 1.17x, but at the cost of silicon area (equal to 4 cores) and lower throughput (25%).

8. RELATED WORK

Prior work has identified DRAM as a major power hog and performance bottleneck, and sought to improve efficiency through interface optimizations and heterogeneity.

Processor-Memory Interface. Leveraging the observation that memory bandwidth is not utilized in today's processors, prior work has either applied frequency scaling to the memory interface and devices [25, 26], or proposed using low-power low-speed memory interfaces [63, 100]. However, emerging manycore servers require large amounts of memory bandwidth, thus mandating high-speed interfaces to maximize per-pin bandwidth. In our work, we use a high-bandwidth cache to filter most of DDR accesses, and utilize under-clocked DIMMs to reduce idle power. Nevertheless, frequency scaling could be leveraged to allow for dynamically adjusting DDR bandwidth.

Prior work sought to reduced idle power of high-speed interfaces by employing a dynamic data rate range or exploiting various power-down states [3]. Unfortunately, there are two main limitations in such approaches. First, interfaces that support a data rate range increase dynamic power compared to fixed-rate interfaces [77], and data rate adjustment requires a time-consuming re-locking process [1]. Second, exploiting power-down states is not practical due to high sleep/wake-up times (a few microseconds) [1, 3].

Wake-up times of DDR interfaces can be reduced to a few tens of nanoseconds by re-engineering their delay-locked loop mechanisms [64]. Such techniques can leverage power-down states of conventional DIMMs at the cost of small performance loss. As soCache serves most of the accesses, DDR latency is not on the critical path, and hence power-down states can be exploited without performance loss.

Heterogeneity. Most systems employ stacked DRAM as a hardware-managed cache [44, 45, 46, 60, 81] with high design complexity and storage overheads (Section 2.3). Lee et al. couple the cache line with OS page and merge address translation with cache management [55]. While this optimization obviates tag arrays, it employs an indirection table with overhead similar to page-level tag arrays while complicating the design of systems with multiple OS page sizes.

Recent work advocates for employing stacked DRAM as part of memory. Software-only epoch-based mechanisms, however, capture only half of the opportunity of hardware-managed caches due to long epochs required to amortize costly software-level page migrations [66]. Many proposals provide hardware-level migration, but at the cost of high-overhead impractical in-DRAM indirection tables [16, 90].

9. CONCLUSION

We presented MeSSOS, a memory system architecture that provides the required memory bandwidth and capacity for scale-out servers. Leveraging insights on skewed access distributions in scale-out workloads, MeSSOS employs multiple high-bandwidth memory modules as a scale-out cache, which is effective in capturing the hot data working sets. Unlike state-of-the-art caches employing impractical in-DRAM block-level metadata, soCache employs a low-overhead in-SRAM page-based organization as coarse-grained access patterns are dominant in high-capacity caches. MeSSOS boosts system throughput and energy efficiency by exploiting the spatial and temporal memory access behavior of scale-out servers.

10. REFERENCES

[1] D. Abts, M. R. Marty, P. M. Wells, P. Klausier, and H. Liu, "Energy proportional datacenter networks," in *International Symposium on Computer Architecture*, Jun. 2010.

[2] J. H. Ahn, J. Leverich, R. Schreiber, and N. P. Jouppi, "Multicore DIMM: An energy efficient memory module with independently controlled DRAMs," *Computer Architecture Letters*, vol. 8, no. 1, pp. 5–8, Jan.-Jun. 2009.

[3] J. Ahn, S. Yoox, and K. C. Choi, "Dynamic power management of off-chip links for hybrid memory cubes," in *Design Automation Conference*, Jun. 2014.

[4] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny, "Workload analysis of a large-scale key-value store," in *International Conference on Measurement and Modeling of Computer Systems*, Jun. 2012.

[5] K. Bains, DDR4 power features for servers. [Online]. Available: http://www.jedec.org/sites/default/files/Kuljit_Bains_SMF_Shenzhen.pdf

[6] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *International Conference on Supercomputing*, Jun. 2006.

[7] V. Ballan, O. Oluwole, G. Kodani, C. Zhong, S. Maheswari, R. Dadi, A. Amin, G. Bhatia, P. Mills, A. Ragab, and E. Lee, "A 130mW 20Gb/s half-duplex serial link in 28nm CMOS," in *International Solid-State Circuits Conference*, Feb. 2014.

[8] L. A. Barroso and U. Holzle, 2009. The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machine. 1st

Edition. Madison: Morgan & Claypool.

[9] A. Belay, G. Prekas, A. Klimovic, S. Grossman, C. Kozyrakis, and E. Bugnion, "IX: A protected dataplane operating system for high throughput and low latency," in *International Symposium on Operating System Design and Implementation*, Oct. 2014.

[10] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCauley, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, and C. Webb, "Die stacking (3D) microarchitecture," in *International Symposium on Microarchitecture*, Dec. 2006.

[11] N. Bronson, Z. Amsden, G. Cabrera, P. Chakka, P. Dimov, H. Ding, J. Ferris, A. Giardullo, S. Kulkarni, H. Li, M. Marchukov, D. Petrov, L. Puzar, Y. J. Song, and V. Venkataramani, "TAO: Facebook's distributed data store for the social graph," in *USENIX Annual Technical Conference*, Jun. 2013.

[12] J. Bulzacchelli, T. Beukema, D. Storaska, P. Hsieh, S. Rylov, D. Furrer, D. Gardellini, A. Prati, C. Menolfi, D. Hanson, J. Hertle, T. Morf, V. Sharma, R. Kelkar, H. Ainspan, W. Kelly, G. Ritter, J. Garlett, R. Callan, T. Toifl, and D. Friedman, "28Gb/s 4-tap FFE/15-tap DFE serial link transceiver in 32nm SOI CMOS technology," in *International Solid-State Circuits Conference*, Feb. 2012.

[13] J. Bulzacchelli, T. Dickson, Z. T. Deniz, H. Ainspan, B. Parker, M. Beakes, S. Rylov, and D. Friedman, "78mW 11.1Gb/s 5-tap DFE receiver with digitally calibrated current-integrating summers in 65nm CMOS," in *International Solid-State Circuits Conference*, Feb. 2009.

[14] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the Flickr social network," in *International World Wide Web Conference*, Apr. 2009.

[15] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of YouTube videos," in *International Workshop on Quality of Service*, Jun. 2008.

[16] C. Chou, A. Jaleel, and M. K. Qureshi, "CAMEO:A two-level memory organization with capacity of main memory and flexibility of hardware-managed cache," in *International Symposium on Microarchitecture*, Dec. 2014.

[17] Cisco. [Online]. Available: http://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-5100-series-blade-server-chassis/at_a_glance_c45-555038.pdf

[18] Cloudera, How-to: Select the Right Hardware for Your New Hadoop Cluster. [Online]. Available: http://blog.cloudera.com/blog/2013/08/how-to-select-the-right-hardware-for-your-new-hadoop-cluster

[19] ——, What Do Real-Life Apache Hadoop Workloads Look Like? [Online]. Available: http://blog.cloudera.com/blog/2012/09/what-do-real-life-hadoop-workloads-look-like/

[20] CloudSuite 2.0. [Online]. Available: http://parsa.epfl.ch/cloudsuite

[21] E. Cooper-Balis, P. Rosenfold, and B. Jacob, "Buffer-on-board memory system," in *International Symposium on Computer Architecture*, Jun. 2012.

[22] A. K. Coskun, "3D stacking as an enabler for low-power high-performance computing," in *Workshop on Design for 3D Silicon Integration*, Jun. 2013. [Online]. Available: hhttp://www.leti-innovationdays.com/presentations/D43DWorkshop/Session5-ThermalAware/D43D13_Session_5_1_AyseCoskun.pdf?PHPSESSID=f7b640026ea14ad29a39a245618e010d

[23] C. R. Cunha, A. Bestavros, and M. E. Crovella, "Characteristics of WWW client-based traces," in *Technical Report BU-CS-95-010, Boston University, Boston, MA*, Mar. 1995.

[24] Datastax, Benchmarking top NoSQL databases. [Online]. Available: http://www.datastax.com/wp-content/uploads/2013/02/WP-Benchmarking-Top-NoSQL-Databases.pdf

[25] H. David, C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu, "Memory power management via dynamic voltage/frequency scaling," in *International Conference on Autonomic Computing*, Jun. 2011.

[26] Q. Deng, D. Meisner, L. E. Ramos, T. F. Wenisch, and R. Bianchini, "MemScale: Active low-power modes for main memory," in *International Conference on Architectural Support for Programming Languages and Operating Systems)*, Mar. 2011.

[27] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, and B. Falsafi, "Clearing the clouds: a study of emerging scale-out workloads on modern hardware," in *International Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2012.

[28] ——, "A case for specialized processors for scale-out workloads," *IEEE Micro*, vol. 34, no. 3, pp. 31–42, May-June 2014.

[29] K. Fukuda, H. Yamashita, F. Yuki, M. Yagyu, R. Nemoto, T. Takemoto, T. Saito, N. Chujo, K. Yamamoto, H. Kanai, and A. Hayashi, "An 8Gb/s transceiver with 3x-oversampling 2-threshold eye-tracking CDR circuit for -36.8dB-loss backplane," in *International Solid-State Circuits Conference*, Feb. 2008.

[30] G. Gangasani, C.-M. Hsu, J. Bulzacchelli, S. Rylov, T. Beukema, D. Freitas, W. Kelly, M. Shannon, J. Qi, H. Xu, J. Natonio, T. Rasmus, J.-R. Guo, M. Wielgos, J. Garlett, M. Sorna, and M. Meghelli, "A 16-Gb/s backplane transceiver with 12-tap current integrating DFE and dynamic adaptation of voltage offset and timing drifts in 45-nm SOI CMOS technology," in *Custom Integrated Circuits Conference*, Sep. 2011.

[31] B. Grot, D. Hardy, P. Lotfi-Lamran, C. Nicopoulos, Y. Sazeides, and B. Falsafi, "Optimizing datacenter TCO with scale-out processors," *IEEE Micro*, vol. 32, no. 5, pp. 52–63, Sep.-Oct. 2012.

[32] L. Gwennap, "ThunderX rattles server market," *Microprocessor Report*, vol. 29, no. 6, pp. 1–4, Jun. 2014.

[33] T. R. Halfhill, "EZchip's Tile-MX grows 100 ARMs," *Microprocessor Report*, vol. 29, no. 3, pp. 1, 6–8, Mar. 2015.

[34] T. J. Ham, B. K. Chelepalli, N. Xue, and B. C. Lee, "Disintegrated control for energy-efficient and heterogeneous memory systems," in *International Symposium on High Performance Computer Architecture*, Feb. 2013.

[35] N. Hardavellas, M. Ferdman, A. Ailamaki, and B. Falsafi, "Power scaling: the ultimate obstacle to 1K-core chips," in *Technical Report NWU-EECS-10-05, Northwestern University, Evanston, IL*, Mar. 2010.

[36] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki, "Toward dark silicon in servers," *IEEE Micro*, vol. 31, no. 4, pp. 6–15, jul-aug 2011.

[37] M. Harwood, N. Warke, R. Simpson, T. Leslie, A. Amerasekera, S. Batty, D. Colman, E. Carr, V. Gopinathan, S. Hubbins, P. Hunt, A. Joy, P. Khandelwal, B. Killips, T. Krause, S. Lytollis, A. Pickering, M. Saxton, D. Sebastio, G. Swanson, A. Szczepanek, T. Ward, J. Williams, R. Williams, and T. Willwerth, "A 12.5Gb/s SerDes in 65nm CMOS using a baud-rate ADC with digital receiver equalization and clock recovery," in *International Solid-State Circuits Conference*, Feb. 2007.

[38] Y. Hidaka, T. Horie, Y. Koyanagi, T. Miyoshi, H. Osone, S. Parikh, S. Reddy, T. Shibuya, Y. Umezawa, and W. W. Walker, "A 4-channel 10.3Gb/s transceiver with adaptive phase equalizer for 4-to-41dB loss PCB channel," in *International Solid-State Circuits Conference*, Feb. 2011.

[39] Hybrid memory cube consortium, Hybrid memory cube specification 1.0. [Online]. Available: http://hybridmemorycube.org/files/SiteDownloads/HMC_Specification%201_0.pdf

[40] Intel scalable memory buffer. [Online]. Available: http://www.intel.com/content/dam/doc/datasheet/7500-7510-7512-scalable-memory-buffer-datasheet.pdf

[41] International technology roadmap for semiconductors. [Online]. Available: http://www.itrs.net/Links/2013ITRS/Home2013.htm

[42] J. Jaussi, G. Balamurugan, S. Hyvonen, T.-C. Hsueh, T. Musah, G. Keskin, S. Shekhar, J. Kennedy, S. Sen, R. Inti, M. Mansuri, M. Leddige, B. Horine, C. Roberts, R. Mooney, and B. Casper, "205mW 32Gb/s 3-Tap FFE/6-tap DFE bidirectional serial link in 22nm CMOS," in *International Solid-State Circuits Conference*, Feb. 2014.

[43] J. Jeddeloh and B. Keeth, "Hybrid memory cube new DRAM architecture increases density and performance," in *International Symposium on VLSI Technology - Digest of Technical Papers*, Jun. 2012.

[44] D. Jevdjic, G. H. Loh, C. Kaynak, and B. Falsafi, “Unison Cache: A scalable and effective die-stacked DRAM cache,” in *International Symposium on Microarchitecture*, Dec. 2014.

[45] D. Jevdjic, S. Volos, and B. Falsafi, “Die-Stacked DRAM caches for servers: Hit-Ratio, latency, or bandwidth? Have it all with Footprint Cache,” in *International Symposium on Computer Architecture*, Jun. 2013.

[46] X. Jiang, N. Madan, L. Zhao, M. Upton, R. Iyer, S. Makineni, D. Newell, Y. Solihin, and R. Balasubramonian, “Chop: Adaptive filter-based dram caching for cmp server platforms,” in *International Symposium on High Performance Computer Architecture*, Jan. 2010.

[47] A. Joy, H. Mair, H.-C. Lee, A. Feldman, C. Portmann, N. Bulman, E. Crespo, P. Hearne, P. Huang, B. Kerr, P. Khandelwal, F. Kuhlmann, S. Lytollis, J. Machado, C. Morrison, S. Morrison, S. Rabii, D. Rajapaksha, V. Ravinuthula, and G. Surace, “Analog-DFE-based 16Gb/s SerDes in 40nm CMOS that operates across 34dB loss channels at Nyquist with a baud rate CDR and 1.2Vpp voltage-mode driver,” in *International Solid-State Circuits Conference*, Feb. 2011.

[48] S. Kanev, J. P. Darago, K. Hazelwood, P. Panganathan, T. Moseley, G.-Y. Wei, and D. Brooks, “Profiling a warehouse-scale computer,” in *International Symposium on Computer Architecture*, Jun. 2015.

[49] M. L. Kersten, S. Idreos, S. Manegold, and E. Liarou, “The researcher’s guide to the data deluge: Querying a scientific database in just a few seconds,” in *International Conference on Very Large Data Bases (VLDB)*, Aug. 2011.

[50] J. Kim and Y. Kim, “HBM: Memory solution for bandwidth-hungry processors,” in *Symposium on High Performance Chips*, Aug. 2014.

[51] H. Kimura, P. Aziz, T. Jing, A. Sinha, R. Narayan, H. Gao, P. Jing, G. Hom, A. Liang, E. Zhang, A. Kadkol, R. Kothari, G. Chan, Y. Sun, B. Ge, J. Zeng, K. Ling, M. Wang, A. Malipatil, S. Kotagiri, L. Li, C. Abel, and F. Zhong, “28Gb/s 560mW multi-standard SerDes with single-stage analog front-end and 14-tap decision-feedback equalizer in 28nm CMOS,” in *International Solid-State Circuits Conference*, Feb. 2014.

[52] J. G. Koomey, Growth in data center electricity use 2005 to 2010. [Online]. Available: http://www.twosides.info:8080/content/rsPDF_218.pdf

[53] R. Kumar and G. Hinton, “Haswell: A family of 45nm IA processors,” in *International Conference on Solid-State Circuits*, Feb. 2009.

[54] N. Kurd, M. Chowdhury, E. Burton, T. Thomas, C. Mozak, B. Boswell, M. Lal, A. Deval, J. Douglas, M. Elassal, A. Nalamalpu, T. Wilson, M. Merten, S. Chennupaty, W. Gomes, and R. Kumar, “Haswell: A family of IA 22nm processors,” in *International Conference on Solid-State Circuits*, Feb. 2014.

[55] Y. Lee, J. Kim, H. Jang, H. Yang, J. Kim, J. Jeong, and J. W. Lee, “A fully associative, tagless DRAM cache,” in *International Symposium on Computer Architecture*, Jun. 2015.

[56] S. Li, K. Chen, J. Ho Ahn, J. B. Brockman, and N. P. Jouppi, “CACTI-P: Architecture-Level modeling for SRAM-based structures with advanced leakage reduction techniques,” in *International Conference on Computer-Aided Design*, Nov. 2011.

[57] S. Li, J. Ho Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, “McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures,” in *International Symposium on Microarchitecture*, Dec. 2009.

[58] K. Lim, D. Meisner, A. G. Saidi, P. Ranganathan, and T. F. Wenish, “Thin servers with smart pipes: Designing SoC accelerators for memcached,” in *International Symposium on Computer Architecture*, Jun. 2013.

[59] G. H. Loh, “D-stacked memory architectures for multi-core processors,” in *International Symposium on Computer Architecture*, Jun. 2008.

[60] G. H. Loh and M. D. Hill, “Efficiently enabling conventional block sizes for very large die-stacked DRAM caches,” in *International Symposium on Microarchitecture*, Dec. 2011.

[61] P. Lotfi-Kamran, B. Grot, M. Ferdman, S. Volos, O. Kocberber, J. Picorel, A. Adileh, D. Jevdjic, S. Idgunji, E. Ozer, and B. Falsafi, “Scale-Out processors,” in *International Symposium on Computer Architecture*, Jun. 2012.

[62] P. Lu and K. Shen, “Virtual machine memory access tracing with hypervisor exclusive cache,” in *Usenix Annual Technical Conference*, Jun. 2007.

[63] K. T. Malladi, F. A. Nothaft, K. Periyathambi, B. C. Lee, C. Kozyrakis, and M. Horowitz, “Towards energy-proportional datacenter memory with mobile DRAM,” in *International Symposium on Computer Architecture*, Jun. 2012.

[64] K. T. Malladi, I. Shaeffer, L. Gopalakrishnan, D. Lo, B. C. Lee, and M. Horowitz, “Rethinking DRAM power modes for energy proportionality,” in *International Symposium on Microarchitecture*, Dec. 2012.

[65] D. Meisner, B. T. Gold, and T. F. Wenisch, “PowerNap: Eliminating server idle power,” in *International Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2009.

[66] M. Meswani, S. Blagodurov, D. Roberts, J. Slice, M. Ignatowski, and G. H. Loh, “Heterogeneous Memory Architectures: A HW/SW approach for mixing die-stacked and off-package memories,” in *International Symposium on High Performance Computer Architecture*, Feb. 2015.

[67] Micron 1Gb: x4, x8, x16 DDR3 SDRAM. [Online]. Available: http://www.micron.com/-/media/documents/products/data%20sheet/dram/ddr3/1gb_ddr3_sdram.pdf

[68] Micron 2Gb: x4, x8, x16 DDR3 SDRAM. [Online]. Available: http://www.micron.com/-/media/documents/products/data%20sheet/dram/ddr3/2gb_ddr3_sdram.pdf

[69] Micron 4Gb: x4, x8, x16 DDR3 SDRAM. [Online]. Available: http://www.micron.com/-/media/documents/products/data%20sheet/dram/ddr3/4gb_ddr3_sdram.pdf

[70] Micron DDR3 SDRAM Power Calculation, http://www.micron.com/products/support/power-calc. [Online]. Available: http://www.micron.com/products/support/power-calc

[71] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, “Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0,” in *International Symposium on Microarchitecture*, Dec. 2007.

[72] N. Nedovic, A. Kristensson, S. Parikh, S. Reddy, S. McLeod, N. Tzartzanis, K. Kanda, T. Yamamoto, S. Matsubara, M. Kibune, Y. Doi, S. Ide, Y. Tsunoda, T. Yamabana, T. Shibasaki, Y. Tomita, T. Hamada, M. Sugawara, T. Ikeuchi, N. Kuwata, H. Tamura, J. Ogawa, and W. Walker, “3 Watt 39.8âĂŞ44.6 Gb/s dual-mode SFI5.2 SerDes chip set in 65 nm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 10, Oct. 2010.

[73] J. Ousterhout, P. Agrawal, D. Erickson, C. Kozyrakis, J. Leverich, D. Mazieres, S. Mitra, A. Narayanan, G. Parulkar, M. Rosenblum, S. M. Rumble, E. Stratmann, and R. Stutsman, “The Case for RAMClouds: Scalable high-performance storage entirely in DRAM,” *SIGOPS Operating Systems Review*, vol. 43, no. 4, pp. 92–105, Dec. 2009.

[74] G. Pass, A. Chowdhury, and C. Torgeson, “A picture of search,” in *International Conference on Scalable Information Systems*, Jun. 2006.

[75] J. T. Pawlowksi, “Hybrid memory cube (HMC),” in *Symposium on High Performance Chips*, Aug. 2011.

[76] M. Poess, R. O. Nambiar, and D. Walrath, “Why you should run TPC-DS: A workload analysis,” in *International Conference on Very Large Data Bases*, Sep. 2007.

[77] J. Poulton, R. Palmer, A. M. Fuller, T. Greer, J. Eyles, W. J. Dally, and M. Horowitz, “A 14-mW] 6.25-Gb/s Transceiver in 90-nm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, Dec. 2007.

[78] M. Pozzoni, S. Erba, P. Viola, M. Pisati, E. Depaoli, D. Sanzogni, R. Brama, D. Baldi, M. Repossi, and F. Svelto, “A multi standard 1.5 to 10Gb/s latch-based 3-tap DFE receiver with a SSC tolerant CDR for serial backplane communication,” in *International Symposium on VLSI circuits*, Jun. 2008.

[79] S. H. Pugsley, J. Jestes, H. Zhang, R. Balasubramonian, V. Srinivasan, A. Buyuktosunoglu, A. Davis, and F. Li, “NDC: Analyzing the impact of 3D-stacked memory+logic devices on MapReduce workloads,” in *International Symposium on Performance Analysis of Systems and Software*, Mar. 2014.

[80] A. Putman, A. Caulfield, E. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G. Gopal, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J.-Y. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. Smith, J. Thong, P. Xiao, and D. Burger, "A reconfigurable fabric for accelerating large-scale datacenter services," in *International Symposium on Computer Architecture*, Jun. 2014.

[81] M. K. Qureshi and G. Loh, "Fundamental latency trade-off in architecting DRAM caches: Outperforming impractical SRAM-tags with a simple and practical design," in *International Symposium on Microarchitecture*, Dec. 2012.

[82] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, "Scalable high-performance main memory system using phase-change memory technology," in *International Symposium on Computer Architecture*, Jun. 2009.

[83] M. Ramezani, M. Abdalla, A. Shoval, M. van Ierssel, A. Rezayee, A. McLaren, C. Holdenried, J. Pham, E. So, D. Cassan, and S. Sadr, "An 8.4mW/Gb/s 4-lane 48Gb/s multi-standard-compliant transceiver in 40nm digital CMOS technology," in *International Solid-State Circuits Conference*, Feb. 2011.

[84] S. Rixner, W. J. Dally, U. J. Kapasi, P. Mattson, and J. D. Owens, "Memory access scheduling," in *Proceedings of the 27th International Symposium on Computer Architecture*, Jun. 2000.

[85] P. Rosenfeld, E. Cooper-Balis, and B. Jacob, "DRAMSim2: A cycle accurate memory system simulator," *Computer Architecture Letters*, vol. 10, no. 1, pp. 16 –19, Jan.-Jun. 2011.

[86] A. Roy, I. Mihailovic, and W. Zwaenepoel, "X-Stream: Edge-centric graph processing using streaming partitions," in *International Symposium on Operating Systems Principles*, Nov. 2013.

[87] J. Savoj, K. Hsieh, P. Upadhyaya, F.-T. An, A. Bekele, S. Chen, X. Jiang, K. W. Lai, C. F. Poon, A. Sewani, D. Turker, K. Venna, D. Wu, B. Xu, E. Alon, and K. Chang, "A wide common-mode fully-adaptive multi-standard 12.5Gb/s backplane transceiver in 28nm CMOS," in *International Symposium on VLSI circuits*, Jun. 2012.

[88] Q. Shaolei, F. Zhong, W. Liu, P. Aziz, T. Jing, J. Dong, C. Desai, H. Gao, M. Garcia, G. Hom, T. Huynh, H. Kimura, R. Kothari, L. Li, C. Liu, S. Lowrie, K. Ling, A. Malipatil, R. Narayan, T. Prokop, C. Palusa, A. Rajashekara, A. Sinha, C. Zhong, and E. Zhang, "1.0625-to-14.025Gb/s multimedia transceiver with full-rate source-series-terminated transmit driver and floating-tap decision-feedback equalizer in 40nm CMOS," in *International Solid-State Circuits Conference*, Feb. 2011.

[89] J. L. Shin, H. Park, H. Li, A. Smith, Y. Choi, H. Sathianathan, S. Dash, S. Turullols, S. Kim, R. Masleid, G. Konstadinidis, R. Golla, M. J. Doherty, G. Grohoski, and C. McAllister, "The next-generation 64b SPARC core in a T4 SoC processor," in *International Conference on Solid-State Circuits*, Feb. 2012.

[90] J. Sim, A. R. Alameldeen, Z. Chishti, C. Wilkerson, and H. Kim, "Transparent hardware management of stacked DRAM as part of memory," in *International Symposium on Microarchitecture*, Dec. 2014.

[91] F. Spanga, L. Chen, M. Deshpande, Y. Fan, D. Gambetta, S. Gowder, S. Iyer, R. Kumar, P. Kwok, R. Krishnamurthy, C.-C. Lin, R. Mohanavelu, R. Nicholson, J. Ou, M. Pasquarella, K. Prasad, H. Rustam, L. Tong, A. Tran, J. Wu, and X. Zhang, "A 78mW 11.8Gb/s serial link transceiver with adaptive RX equalization and baud-rate CDR in 32nm CMOS," in *International Solid-State Circuits Conference*, Feb. 2010.

[92] J. Stuecheli, "Next generation POWER microprocessor," in *Symposium on High Performance Chips*, Aug. 2013.

[93] S. Volos, J. Picorel, B. Falsafi, and B. Grot, "BuMP: Bulk memory access prediction and streaming," in *International Symposium on Microarchitecture*, Dec. 2014.

[94] T. F. Wenisch, R. E. Wunderlich, M. Ferdman, A. Ailamaki, B. Falsafi, and J. C. Hoe, "SimFlex: Statistical sampling of computer system simulation," *IEEE Micro*, vol. 26, no. 4, pp. 18–31, Jul. 2006.

[95] B. Wheeler, "Tilera sees opening in clouds," *Microprocessor Report*, vol. 25, no. 7, pp. 13–16, Jul. 2011.

[96] Wikipedia, Does Wikipedia traffic obey Zipf's law? [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:Does_Wikipedia_traffic_obey_Zipf's_law%3F

[97] R. D. Williams, T. Sze, D. Huang, S. Pannala, and C. Fang, Server memory road map. [Online]. Available: http://www.jedec.org/sites/default/files/Ricki_Dee_Williams.pdf

[98] R. E. Wunderlich, T. F. Wenisch, B. Falsafi, and J. C. Hoe, "SMARTS: Accelerating microarchitecture simulation via rigorous statistical sampling," in *International Symposium on Computer Architecture*, Jun. 2003.

[99] H. Xi, J. Zhan, Z. Jia, X. Hong, L. Wang, L. Zhang, N. S. Sun, and G. Lu, "Characterization of real workloads of web search engines," in *International Symposium on Workload Characterization*, Nov. 2011.

[100] D. H. Yoon, J. Chang, N. Muralimanohar, and P. Ranganathan, "BOOM: Enabling mobile memory based low-power server DIMMs," in *International Symposium on Computer Architecture*, Jun. 2012.