MONet: Heterogeneous Memory over Optical Network for Large-Scale Data Centre Resource Disaggregation

VAIBHAWA MISHRA[1], JOSHUA L BENJAMIN[1*], AND GEORGIOS ZERVAS[1]

[1] *Department of EEE, University College London, United Kingdom*
[*] *Corresponding author: joshua.benjamin.09@ucl.ac.uk*

Compiled March 3, 2021

Memory over Optical Network (MONet) system is a disaggregated data center architecture where serial (HMC) / parallel (DDR4) memory resources can be accessed over optically switched interconnects within and between racks. An FPGA/ASIC-based custom hardware IP (ReMAT) supports heterogeneous memory pools, accommodates optical-to-electrical conversion for remote access, performs the required serial/parallel conversion and hosts the necessary local memory controller. Optically interconnected HMC-based (serial I/O type) memory card is accessed by a memory controller embedded in the compute card, simplifying the hardware near the memory modules. This substantially reduces overheads on latency, cost, power consumption and space. We characterize CPU-memory performance, by experimentally demonstrating the impact of distance, number of switching hops, transceivers, channel bonding and bit-rate per transceiver on bit-error rate, power consumption, additional latency, sustained remote memory bandwidth/throughput (using industry standard benchmark STREAMS) and cloud workload performance (such as operations per second, average added latency and retired instructions per second on memcached with YCSB cloud workloads). MONet pushes the CPU-memory operational limit from a few centimetres to 10s of metres, yet applications can experience as low as 10% performance penalty (at 36m) compared to a direct-attached equivalent. Using the proposed parallel topology, a system can support up to 100,000 disaggregated cards.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

In the era of big data, large-scale data applications, including video streaming and analytics, financial and scientific computation, are migrating to the cloud, which is expected to house 95% of all traffic by 2021 [1]. Such applications tend to use a network of standalone servers that form conventional data centers (CDCs). CDCs follow a server-centric approach, whereby, available resources (computing and physical memory) per server are fixed and limited to the boundaries of the server's tray. In general, a server tray typically consists of multiple heterogeneous processing unit(s) (xPU) attached via one (or multiple) Memory Controller(s) (MC) to a tray-local Random Access Memory (RAM), for rapid instruction read and fast, random read/write byte-level access. The xPUs can also access persistent local storage and I/O devices (e.g. flash storage, accelerators) using a single or a hierarchy of I/O bridges. Similarly, each server tray also hosts one or multiple network interface cards (NICs), leveraging physical connectivity to dedicated network switches and routers (and, also sometimes, the switching capability available on-board on multi-port NICs) to facilitate data exchange between xPUs that reside in distinct server trays and, potentially, in distinct racks. Recent research has also led to the development of heterogeneous memory unit(s) (xMU): DDR4, Hybrid Bandwidth Memory (HBM) and Hybrid Memory Cube (HMC). While parallel bus memory (DDR4 or HBM) use 100-1000 parallel pins each pin operate at low bit rate up to a few Gb/s, serial memory (HMC) uses up to 8-64 pins/links each at high bit-rate up to 15 Gb/s, providing compatibility with serial I/Os and increasing the overall communication bandwidth to $>$ 1Tb/s [2, 3]. On the other hand, HMC memory is limited in memory size (2GB per chip and 16GB if cascaded) whereas HBM (4 GB per chip) and DDR4 (16-32 GB per chip) can scale to large memory size but with more moderate bandwidth. Serial memory technologies like HMC, due to their serialized I/Os, can potentially lead to high bandwidth access, low round trip system latency, and low energy efficiency as they eliminate the need for memory controller attached to memory. However, the ratio of demand for resources, such as memory-to-CPU, varies 3-4 orders of magnitude and, in some a cases, it is higher than what is available in a server's tray [4], [5]. This leads to severe resource under-utilisation despite the high infrastructure cost, power and complexity [6, 7]. Moreover, scaling such under utilized networks increases resource wastage.

To address these drawbacks, disaggregation of server resources (xPUs, xMUs, FPGAs/ASICs) into standalone units across a network was proposed [4]. In addition to creating a pool of accessible resources available across the entire network, disaggregation of resources creates resource modularity and flexibility i.e. resources can be plugged as desired beyond a conventional tray's limits. However, electronically switched network for inter-tray communication rely on standard network TCP/IP protocol [7]. Such protocols incur latency in the order of microseconds, which substantially degrades disaggregated CPU-memory throughput performance. Furthermore, several attempts have been made to maximize bandwidth utilization for network-attached parallel memory such as using RDMA over InfiniBand or converged Ethernet [8–10]. However, as CPU-memory connections exhibit high variance in data exchange sizes, they cannot be efficiently accommodated by data transport architectures, which are optimized for block transfers in data centers. Memory hot-plugging and ballooning memory pool [11] across the entire network can boost resource utilization performance within disaggregated data center networks. However, this requires appropriate hardware support such as virtual-to-physical address translation and identification/mapping of available memory to specific locations within the network. Moreover, memory disaggregation faces two key challenges: network latency and CPU-memory bandwidth [12]. A reconfigurable (FPGA-based), resource utilization boosting, disaggregated data center architecture that uses an optical switch to achieve sub-microsecond latency was proposed in [13, 14]. Achieving high bandwidth can be costly in terms of channel count, performance, energy efficiency and device footprint as multiple channels and memory modules per channel are required. These arrangements will increase latency since they need additional off-chip memory controllers at the memory side [15]. Crucially, none of the above studies accommodate remote memory logic at CPU for two types of memory and have an optical network architecture to support heterogeneous remote memory access. Also, they don't experimentally assess the impact of network bandwidth, distance and communication performance (bit-error-rate) on memory bandwidth.

By creating a disaggregated pool of serial and parallel memories one can compose a computing system with the appropriate type and size of memory for the workloads and applications of interest. However, composable data centers bring a number of fundamental challenges such as (a) higher memory access latency and bandwidth against traditional direct-attached memory, and (b) increased cost and power consumption from additional network resources. Thus, we propose a novel and flexible disaggregated data center architecture called Memory over Optical Network (MONet). MONet can (a) support parallel-type topology for scale-out disaggregated data center system (b) support local and remote access of both serial and parallel memory elements, (c) offer very low hardware latency, (d) minimize the routed path distance between any two CPU and memory and, in turn, reduce network latency, (e) offer increased memory bandwidth against state of the art and (f) scale out to support multiple memory modules locally and remotely.

Further expanding the work in [16], this paper has the following contributions.

i Comparison of MONet with respect to state-of-the-art memory disaggregated architecture research (§2).

ii Simulation of an optical circuit switched network architecture that aims to minimize hops and power consumption due to optical network only (§3,§5.A).

iii In-depth description of MONet logic (§4) and comprehensive study with results showing memory access latency, throughput and power performance of read/write, local/remote, serial/parallel memory for different number of channels (transceivers) and line rates using custom baseline memory benchmarks for a range of optical distances (§5.B)

iv Measured component level power consumption of MONet's disaggregated local/remote serial/parallel and it's implication on network energy consumption (§5.B).

v Physical layer performance (BER) analysis for remote serial memory at different line rates now incorporates the performance study at a link-level (§5.C).

vi Comparison on sustained memory bandwidth for DDR4 and HMC with industry standard STREAM benchmark at different line-rates and channels (transceivers) (§5.D) [17].

vii Demonstration of YCSB cloud workloads on memcached (§5.E) and MONet performance under different distances/line rates and conclusion in §6.

2. RELATED WORK

Several key pieces of research have been carried out and published to demonstrate how latency and memory bandwidth can be optimized for disaggregated data center networks. The relevance of our proposed architecture in this paper (MONet) and its current standing with respect to leading research on optically disaggregated compute to remote memory access is shown in Table 1. In [18], the authors propose the SO-NUMA architecture that attaches remote parallel (DDR3) memory via a remote memory controller (RMC). Though the RMC is tightly coupled to the processor's cache coherence hierarchy, experimental results suggest that remote memory access latency is approximately 1.5 μs while a remote memory bandwidth of 0.225 GB/s is achieved. Although the SO-NUMA simulated platform proposes a best-case round trip latency and remote memory bandwidth of 300 ns and 9.625 GB/s respectively, their hardware demonstration reports that disaggregation severely degrades the CPU-memory performance. Yan et al. [19] proposed a switch and interface card (SIC) as a replacement for standard NICs in order to support Ethernet packet switched and optical time-multiplexed circuit switched services. However, this approach demonstrated memory-to-memory transactions instead of compute-to-memory transactions as the SIC is attached to the CPU (over a PCIe bus). This leads to a memory read/write transaction latency of about 10 μs, even for small data sizes ($\leq$ 1 KB), which is not acceptable for certain applications as it substantially degrades their performance [20].

An optical connected memory has been discussed in [21], where error correction protocols have been demonstrated to improve the physical layer reliability in terms of bit error rate between CPU and memory devices. However, the authors have not discussed the impact of errors on memory bandwidth. In [14], authors have achieved 712 ns round trip remote memory access latency while sustained memory bandwidth of 0.62 GB/s, where remote parallel memory is attached over 10GigE network protocol. Moreover, the complex FPGA hardware architecture employed requires a large amount of resources per remote memory module while achieving very low sustained bandwidth. Using Aurora IP [22] between CPU and remote memory attachments, D. Syrivelis *et al* [23] have demonstrated disaggregation,

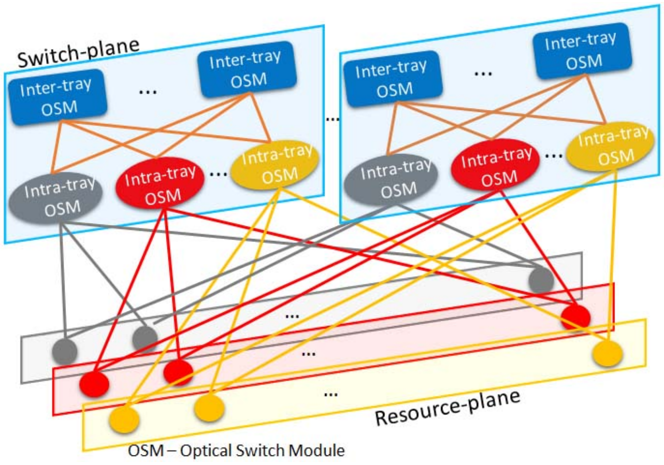

Fig. 1. MONet: Proposed optical network topology

achieving a memory access latency and bandwidth of 857.6 ns and approximately 1 GB/s respectively. A similar work has been reported in [24], where authors proposed a hardware logic block called REMAP, which bridges the CPU and remote memory. REMAP offers remote memory access latency of 750 ns (one hop) with bandwidth of 1.07 GB/s. A prototype of full-stack optically disaggregated system called ThymesisFlow-P was proposed in [25] [26], which achieved 3 GB/s bandwidth with 700 ns round-trip latency using 8 links. Except for ThymesisFlow-P, none of the above systems utilize comparable bandwidth to that offered by MONet as shown in Table 1 when accessing remote DDR4. While ThymesisFlow-P achieves equivalent memory bandwidth for DDR4, MONet offers lower round-trip write latency (526.6 ns, at 8-metre round-trip) compared to ThymesisFlow-P, as shown in Table 1. Although both MONet and ThymesiFlow-P support cloud workloads, they primarily differ in architecture and resources (CPU, memory elements) employed. ThymesisFlow-P employs the IBM POWER9 processor with AC922 architecture [27], a cache coherent OpenCAPI interface and two combined (2GB + 8GB) DDR4 memory to create a large 10 GB memory. On the other hand, MONet uses a quad-core A53 ARM processor with a 512 MB DDR4 and a 2GB (scalable) HMC memory. In our experiment, we demonstrate MONet on an FPGA-based platform equipped with less powerful processors and low-size memory units as discussed above. However, the ReMAT IP (4), the hierarchical and network model proposed in MONet (3) are all transparent to the CPU and the memory element used.

MONet is shown to support HMC memory while the other disaggregated systems only employ DDR4 memory. Table 1 shows that HMC and DDR4 achieve a high sustained bandwidth of 70% and 94% respectively when using the STREAM benchmark. Also, HMC remote memory write latency overhead (R_{HOV}) is found to be only 1.6x slower than local HMC access and 1.3x faster than MONet remote DDR4 access. The remote DDR4 write access latency overhead (R_{DOV}) is around 8-30x slower than local DDR4 memory access.

3. DISAGGREGATED ARCHITECTURE: MONET

We propose MONet, a disaggregated data center architecture, that is based on the proposed parallel-topology and flexible resource-centric system by creating a pool of accessible compute and memory resources. Hence, MONet provides increased levels of scalability and flexibility; resources can be dynamically added/removed as required by applications.

Table 1. Related work comparison: 8-m round-trip optical distance CPU-MEM. (# : Full-Width, * : Half-Width, - : Unstated, STREAM/Baseline Memory Bandwidth (GB/s), SB = Sustained Bandwidth (%) remote-local access comparison, R_{DL} = Remote DDR4 write latency, R_{HL} = Remote HMC write latency, R_{DOV} = Remote DDR4 write latency overhead, R_{HOV} = Remote HMC write latency overhead)

Work		MONet	[24]	[23]	[25]	[18]
Local	DDR4	3.7/15.5	1.9/-	-	-	-
	HMC#	3.8/37.8	-	-	-	-
	HMC*	3.2/22.6	-	-	-	-
Remote	DDR4	2.6/8.1	1.07/-	1.0/-	3.0/-	0.23/-
	HMC*	3/22.5	-	-	-	-
SB (%)	DDR4	70/52	29/-	27/-	81/-	6/-
	HMC*	94/99	-	-	-	-
R_{DL} (ns)		526.6	750	857.6	700	1500
$R_{DOV}(\times)$		8-30	11-43	13-49	10-40	23-87
R_{HL} (ns)		387.2	-	-	-	-
$R_{HOV}(\times)$		1.6	-	-	-	-

MONet follows 2-tier parallel topology to minimize the path distance between any resources. A tray, shown as resource-plane in Fig 1, can host a set of resource cards composed of xPUs and/or xMUs. Compared to the non-parallel fat-tree architectures, it only has two tiers of switches, aiming to minimize the number of hops between any two resources and network latency. To realize the 2-tier parallel structure, the concept of a switch-plane is developed. As shown in Fig 1, the switch-planes are not connected to each other, allowing modular increase per resource-plane bandwidth. Each switch-plane houses a top-tier of inter-tray Optical Switch Modules (OSMs) and a lower-tier of intra-tray OSMs, connected in a spine-leaf topology. Since each switch-plane links all the resources (coloured circles on Fig 1) of all trays, any-to-any card communication can be executed via any individual switch-plane. All cards in one resource-plane are interconnected via a corresponding intra-tray OSM in each switch-plane. For instance, in Fig 1, grey cards are only linked to the grey intra-tray OSM in each switch-plane (first-tier). Communication between the resources in different resource-plane need the participation of the inter-tray OSM (second-tier). The switch-plane number depends on the port number per resources and the channel number per link. For example, if 8 SDM (over 8 separate fibres) channels per transceiver are used between a CPU card and a memory card, then 8 switch-planes will be required. However, if these are two groups/transceivers of four CWDM channels then two planes are necessary. The intra-tray OSM number in each switch-plane is equal to the resources number per resource-plane, which indicates the advantages of scalability and flexibility favouring data-plane communications since the number of resource-planes in the network can be incremented just by increasing the number of inter and intra-tray OSM switches in each plane respectively. This parallel increment of switching elements is enabled as a result of employing multi-transceiver Mid-Board-Optics (MBOs), which facilitates

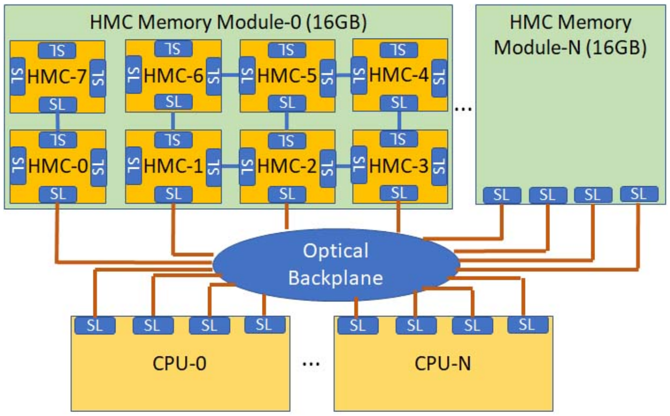

Fig. 2. CPU and remote serial memory module interconnection, with cascaded memory units for scale-out

full connectivity between all resources in this parallel network where each optical I/O channel/fibre of the MBO is routed towards one individual resource-plane. Moreover, in order to increase the port utilization of each optical switch and the modularity of the system, small port-count switches are utilized for both inter and intra-tray OSM. The parallel planes can either be used to support (a) multi-routing between end-points (compute/memory cards) or (b) scale-out to different clusters/racks of trays; fig 1 visualizes the former case. Following the latter case, the system scales based on *cards* × *trays* and can lead to 100,000s of end-points (compute/memory cards) system. This can be achieved with 24 cards per tray and 4096 trays. Each card can be supported by one 8-channel MBO SDM transceiver (each channel connected to one parallel plane, 8 in total) and having moderate radix (48x48) switches for both tiers to provide 1:1 subscription ratio. To scale the disaggregated system from a single rack to a data center, a scale-out network architecture needs to be considered. We use optically circuit switched (20 ms reconfiguration [28]) interconnects in order to pull together such heterogeneous compute and memory resources that operate at virtual-machine (VM) time frames (seconds-hours). MONet proposes the utilization of all 4-links of HMC to scale-up the remotely attached memory capacity. As shown in Fig 2, MONet enables a single CPU to access any available remote HMC memory module within the data center network. One CPU can access 16 GB remote memory (8x 2GB HMC modules directly interconnected within one card) over (a) a single link (up-to 480 Gb/s) or (b) four links (up to 1.9 Tb/s bi-directional link bandwidth). Another option is to access four different remote HMC memory modules (in total 64 GB) over all four serial links of the CPU. In the same manner, one memory module can be also shared by four CPUs at the same time in a specific configuration. Hence, we have defined the available configurations based on the limits imposed by HMC [29] and not the MONet architecture.

4. EXPERIMENTAL SETUP

The experimental demonstration of the proposed MONet architecture, presented in the 3, follows the fully developed and integrated hardware setup shown in Fig 3. The basic building blocks to enable the on-demand disaggregated data center resources are: (a) CPU micro-server (ASIC/FPGA) module, (b) FPGA hosted parallel DDR4 memory card, (c) optically connected (serial) memory card, (d) opto-electronic transceivers, (e) high speed optically switched interconnect and (f) resource manager. Two types of remote memory cards are supported: (1) conventional parallel memory access (with DDR4) and (2) standalone serial memory in the network (with HMC). The first type of memory card is called *ASIC/FPGA Hosted Memory Card* this work will target a re-programmable platform (FPGA), support parallel access and provide a flexible pool of memory modules that can be shared among all CPUs. The second type of memory card, called *Optically connected Serial Memory Card*, will be independent of any host. It only uses HMC memory modules, which embed serial transceivers on the same chip. This directly attaches the memory modules to the network through an electrical or optical interconnect. Unlike traditional electrically interconnected architectures, MONet exploits the large-bandwidth and low-loss supported by optical fibers to achieve high bandwidth longer distance products. Electrical transmission lines, at high bit rates, are limited to a few metres distance, in order to ensure signal integrity. For example, 25 Gb/s electrical transceivers are used for $\leq$ 5 m distances. The low loss of optical fiber ($\leq$ 0.2 dB/Km), on the other hand, allows for a significantly higher reach. Thus, opto-electronic transceivers attached to each card are required to be connected to the optical backplane to provide network access to the resources. Each card uses mid-board optics (MBOs) based SiP boards that house handle electro-optic conversion and modulation on up to 8 bi-directional optical transceivers. As shown in Fig. 3 and elaborated in §4.C, the MBO equips the CPU and memory cards with multi-channel optical transceiver capabilities that enable remote network resource access. A 48 port beam-steering switch [28] (red box in Fig. 3) was used to connect these cards and emulate multi-hop networks in the experiment.

4.A. CPU Micro-Server Card

The compute resources or the CPU micro-server cards feature the Xilinx Zynq UltraScale+ RFSoC ZCU111 Evaluation Board (specific part: EK-U1-ZCU111-G) [30] , which integrates a quad-core A53 ARM Application Processing Unit (APU) and a dual-core ARM Cortex R5 Real-time Processing Unit (RPU). The ARM processing system (PS) is connected to MONet's hardware logic via two AXI memory-mapped master ports. Our proprietary intellectual property facilitates any server CPU to access any type of local or remote memory (elaborated in §4.B). Each CPU micro-server card also has a locally attached DDR4 and HMC memory, which are connected through a hardware interconnect. Local DDR4 memory can be accessed using parallel I/O, whereas the HMC memory can be accessed in two ways: (1) a full-width (FW) configuration (over 16 serial lanes) and (2) a half-width (HW) configuration (over 8 serial lanes). As for the remote memory access, the DDR4 memory is accessed via 1,2,4 or 8 serial Aurora lanes, whereas the HMC memory via 8 serial lanes only.

A custom *Programmable Benchmark Block* (PBB) is used to measure the baseline performance in terms of bandwidth and latency between CPU and memory. The PBB is capable of generating read/write transactions up to 2 GB, for both the DDR4 memory (maximum burst length is 256, burst size is 64 bytes) and the HMC memory (maximum burst length is 1, burst size is 128 bytes). Even though the HMC memory can support a maximum of 240 GB/s link bandwidth in a 4-link configuration, in this experiment, we use only a single link either in full-width (16 serial lanes, 480 Gb/s or 60 GB/s) or in half-width (8 serial lanes, 240 Gb/s or 30 GB/s). The CPU card employs MBO-based transceivers for remote memory access, which can operate up to 25 Gb/s/lane; however, in our experiment, we use line rates of 10, 12.5 or 15 Gb/s due to HMC's transceiver's limitation.

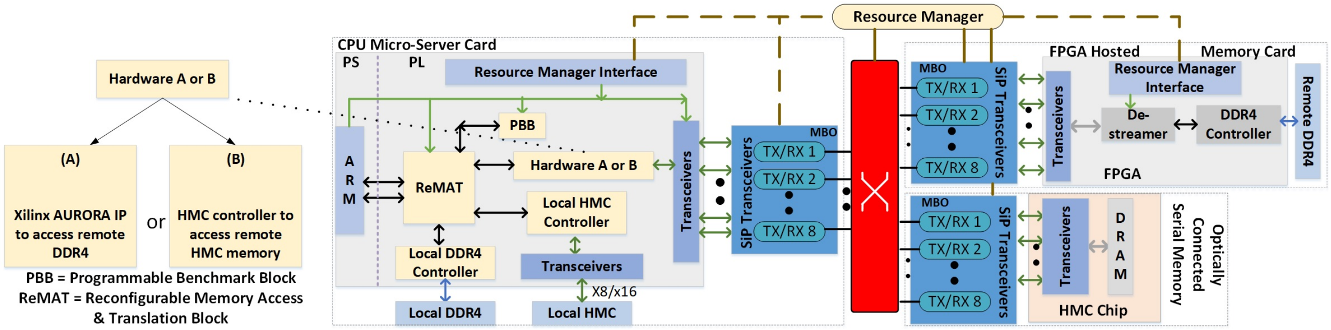

Fig. 3. Experimental Setup: CPU Micro-server and communication network to support local and remote heterogeneous parallel DDR4 memory, and serial HMC memory.

4.B. Reconfigurable Memory Access & Translation: ReMAT

The ReMAT offers low latency, multi-stage pipelined, interconnected hardware logic, which can understand, process and forward memory access requests to desired memory locations. While enabling essential capabilities for disaggregated memory access such as non-blocking and software defined/controlled memory access, ReMAT also supports single and double cache line transactions for faster local or remote memory access. A round-robin arbiter in the ReMAT interconnect combines memory transactions from both master ports of the CPUs to a single transaction to provide time-interleaved memory access (HMC/DDR4, local/remote) over single hardware. The arbiter also contains additional features to combine multiple transactions of same type (write or read) to a single transactions of 64 (for DDR4) and 128 (for HMC) bytes to overcome overhead inferred due to multiple read and write cycles. ReMAT uses a streamer to convert each memory transaction to standard AXI4-stream data, enabling MONet to support standard chip-to-chip data access protocols such as Ethernet.

Successful functionality of MONet depends on mapping tables in ReMAT, which record and translate location, type of device and pattern of memory access. These tables are collectively combined to create *Bridge Function Control Tables* (BFCT) that can be constantly updated by Resource Manager (RM), via the Resource Manager Interface, using ReON [31] without interrupting the local CPU or other system resources. We employ Linux Kernel NUMA extensions to represent remote memory address range as NUMA nodes. Each of these nodes will reflect different subgroups of memory. Whenever a new memory module is attached, the OS's kernel updates table (page) entries and then, the Memory Management Unit (MMU) assigns physical addresses to virtual addresses. ReMAT offers transparency and mapping between physical memory address (as seen by CPU) and remote memory address, irrespective to memory location (local or remote) and technology (serial or parallel). User applications, being unaware of address mapping and translations, use a virtual memory system, which takes care of address translation (virtual-to-physical and physical-to-virtual). Xilinx ZYNQ MPSoC can double its physical memory capacity to 448 GB by using two, rather than one, HP master ports (each can address up to 224 GB), where 1792 × 256 MB memory sections can be accommodated and can be attached to the kernel using memory hot-plug. ReMAT has been developed to support any address range from 256 MB to 448 GB, enabling it to support memory subsystems of diverse sizes (small to large). As a small footprint, 16 memory sections (each of 256 MB, 4 GB in total) can be equally partitioned and assigned to 4 memory modules (1 GB each). These memory modules can be, for example, all combinations of local/remote DDR4/HMC. ReMAT is efficiently pipelined to simultaneously access multiple memory resources to increase the overall memory utilization.

4.C. Optical Data-path

MBOs are seen as an attractive solution to replace copper-based interconnects and exploiting optical printed circuit boards (OPCBs) [32] for interconnecting various on board IT resources. Each channel, on average, has a -3 dBm optical output power and can operate up to 25 Gb/s. To minimize footprint and power consumption and to maximize bandwidth density, each resource card uses MBOs that are integrated with SiP transceivers and manufactured by Luxtera Inc (now Cisco) [33] with a capacity of up to 200 Gb/s (8×25) and a single 1310 nm laser source, on a single chip. The opto-electronic transceiver we used for the purposes of our experiment is the Luxtera LUX62608, which was also showcased in [13, 14]. However, in this setup, each channel operates at 10, 12.5 or 15 Gb/s, as limited by the operational capability of HMC. The HMC memory module supports 64 channels each up-to 15 Gb/s each (1.9 Tb/s bi-directional line rate or 240 GB/s link bandwidth) in a 4-link configuration, where one link can be accessed by a single CPU. Considering the electrical I/Os connected to the MPSoC, each CPU card can potentially support up-to 1.8 Tb/s (28 serial lane each up-to 32.75 Gb/s) bi-directional line rate [34].

A 48-port optical circuit switch (OCS) by Polatis is used for: (1) accessing different multiple resources-planes and switch planes (2) emulating multi-hop networks. The main reason for using a pure circuit switched network is to minimize CPU-to-memory latency up to sub-microsecond level. This emulates the CPU bus structure, which delivers the lowest level of guaranteed latency and bandwidth between processor and memory elements while connecting any number of CPU cards to any number of memory cards. Switching is based on piezo-electric actuation technology [28], which limits reconfiguration time to 25 ms. The optical power loss, incurred when traversing the switch, is only 1 dB on average. Thus, a cascade of such switches can be used to build a large-size network. In the setup, compute card, memory card and fiber loop-backs are all attached together to the switch (acting as a backplane) in order to traverse the switch multiple times and thus emulate a multi-tier network.

4.D. Remote Memory Cards

FPGA hosted memory cards use conventional parallel pins between memory chip and controller (DDR4 controller). In our experiments, we used the MPSoC based FPGA evaluation board developed by HiTech Global (specific board: HTG-Z920) [35] for hosting the memory controller to access remote DDR4. Thus, in this case, an additional FPGA chip is required to host memory controller and additional logic called de-streamer to convert AXI4 stream requests to parallel AXI memory-mapped transactions. Memory access requests can arrive from single or multiple (many bit-synchronous bonded channels) transceivers and from single or multiple CPU; the de-streamer can be pre-configured by resource manager to tackle this scenario. The memory controller hardware is a passive slave to the linked CPU microserver cards with a simplified connection due to the use of a circuit switched network. This arrangement minimizes processing latency for incoming memory access request as the memory cards only store transceiver and port number for managing responses, instead of detailed CPU address/locations. In the case of optically connected serial memory, an additional chip to host memory controller is not required as they have a built-in memory controller and high speed serial I/Os. For our experiment, we employed the 2 GB Hybrid Memory Cube (HMC) FMC+ Module (VITA 57.4) cards manufactured by HiTech Global [36]. However, serial HMC memory require dedicated low-speed distributed clock network (up to 125 MHz) and additional command interface such as I2C to provide boot-time configuration. Each HMC module needs to be booted up before CPU can access them; this is handled by the Resource Manager in the MONet architecture. MONet is a fully flexible architecture that can be extended to support any type of memory as long as a compatible memory controller with AXI4-MM bus interface is employed. We experimentally demonstrated MONet for DDR4 memory; however, the architecture is not limited to support DDR4 parallel memory only. While, for serial memory type, the memory translation and control is hosted in the compute card, parallel memory types require memory translation and control units at both the compute and memory card ends. We expect memory with higher data rate, such as DDR5, to substantially increase MONet's achievable memory throughput.

4.E. Resource Manager: RM

RM keeps an up-to-date information of available resources such as CPU, available memory resources and their allocation, and dynamically control the allocation and memory interconnect configuration. For the experimental demonstration in this paper, the RM Uses dedicated low-speed network such as USB and I2C to configure optical switch network, provide end-to-end communication paths between CPU and memory and to provide boot-time configuration for HMC memory modules. In larger or long-distance networks, RM will be equipped with Ethernet ports to reconfigure resources with the ReON protocol [31]. MONet cannot be adapted in existing server-centric data centers as it requires custom ASIC/FPGA-based processors to host ReMAT, RM Interface, PBB and local controller glue logic that enable address translation and memory control. In the proposed MONet disaggregated DCN, any xPU can access any xMU provided it is co-hosted with our proposed custom hardware that is compatible with AXI4-MM interfaces.

5. RESULT AND DISCUSSION

In this section, we evaluate the baseline performance of MONet for local/remote DDR4/HMC access by measuring the achieved memory bandwidth, round-trip end-to-end latency, network energy consumption/contributors and the physical optical network BER-dependency/limitation. Following this, we not only evaluate the memory performance using the custom and industry standard STREAM benchmark but also analyse cloud workload performance for local/remote DDR4/HMC. The results showcased in §5.A are based on simulations while §5.B-5.E focus on measured experimental results, unless stated otherwise.

5.A. Architecture Characterization

A network simulator has been developed in MATLAB to evaluate the performance of the proposed MONet optical network architecture in Fig. 1 and to compare against conventional non-parallel architectures as well as in terms of network latency (in terms of hops) and power consumption. The simulation procedure is divided into four main steps: 1) generating request, 2) resource allocation, 3) network reconfiguration 4) connection establishment. In step 1, VM requests arrive dynamically following a Poisson distribution with an average inter-arrival time of 10 time units, also containing information like CPU core number, RAM size, CPU-RAM latency, bandwidth required and holding time. The holding time starts from 6300 time units and increases 360 time units for every 100 requests. For the purpose of our simulation, we have assumed that the number of planes is 12-24, resources per plane is 8-24, the CPU-MEM resource card ratio is 1:1, each CPU card has 4 cores, each memory module can be 2-16 GB when serial (Fig. 2) or 4-8 GB when parallel, the total number of VM requests is 1000. A network-aware locality-based algorithm developed [37] for a range of workloads (random, high RAM, high CPU, half-half etc.), is applied in step 2 for resource allocation and a modified K-shortest path algorithm is employed in step 3 for network reconfiguration. MONet resources (CPU, memory and switch ports) are granted and reserved for VM requests only when sufficient resources are available, otherwise, the request is dropped. Simulation parameters are described as follows: link distance used in both non-parallel and MONet data-center architecture is kept identical such as 0.25 m between resource and intra-tray-OSM; 3 m between intra-tray and inter-tray-OSM. An additional 10 m link distance is assumed between intra-rack and inter-rack optical switch in non-parallel architecture. The right side of Fig 4 shows the architecture-level latency cumulative distribution function (CDF) of the round-trip (memory read/write transaction) network latency for non-parallel and the proposed MONet architecture based on the assumptions in [37]. As shown, MONet ensures that access to any resource-plane within 3-hops; these connections include intra-tray-OSM and inter-tray-OSM to reach all resources. Occupying 77% within 1-hop and rest of 23% within 3-hops of end-to-end total traffic, MONet delivers high network utilization and offers

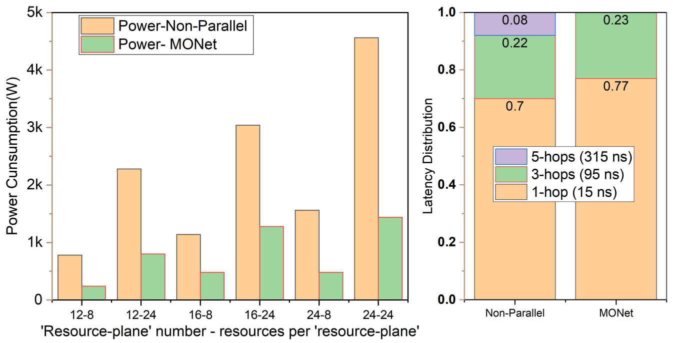

Fig. 4. Switch Plane Characterization: Power and network latency comparison between Non-Parallel (fat tree) and MONet architectures.

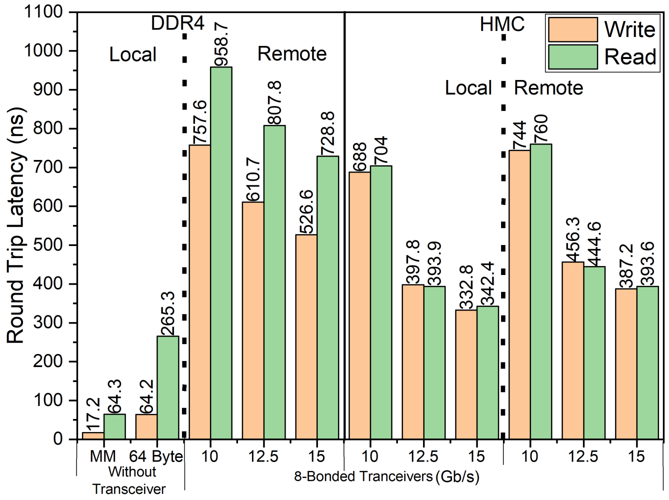

Fig. 5. DDR4/HMC local/remote (8m) memory read/write latency: 8-bonded transceivers each at 10,12.5, 15 Gb/s.

low round-trip latency (95 ns). In Fig. 4, we use experimentally measured per-port power consumption (5 W) [14] of a 48-port SMF Polatis OCS to estimate the overall network switching power consumption for both the Non-Parallel and MONet optical network architecture. As shown, with maximum number (576) of resources, non-parallel architecture consumes 4.5 kW, whereas MONet saves 68 % and consumes only 1.4 kW. In summary, the proposed MONet network topology in Fig. 1 reduces latency, switch hops and overall network power consumption. Resource utilization results of the algorithms used are not reported here, as it is not the focus of the paper; however, it can be found in [37].

5.B. Memory Access Characterization

To evaluate the memory access performance in MONet, baseline characterization of metrics such as latency, throughput and power are important as they indicate hardware penalty introduced by MONet, transceivers, optical data-path and memory technology. To benchmark the baseline performance, the memory throughput for accessing local DDR4 and HMC memory resources are measured.

As shown in Fig 5, remote-attached serial HMC memory at 8 m round-trip distance offers 387.2-760 ns latency compared to 332.8-704 ns of locally attached serial memory; the penalty of the additional 50 ns to 60 ns in latency is purely caused by only the optical data path propagation delay. In contrast, remote-attached parallel DDR4 memory (same distance of 8 m) experiences 30-56x increase in latency when compared to the locally attached case. This is caused by the memory-mapped to Xilinx AXI4 streamer (4x latency increase for either read/write - see MM and 64 Byte bars under DDR4 local section of Fig 5) and Aurora transceiver protocol stack (2.8x-3.6x and 8.2x-11.8x for read and write latency increase respectively - see Fig 5 and in particular, DDR4 64Byte bars vs remote access bars) deployed on both CPU and memory cards. Furthermore, extending the optical network for both types of memory to the intra-rack level, we experimentally observed that the round trip latency is the function of optical distance between CPU and remote memory; it can be mathematically expressed as in Eq. 1:

$$R_{Latency}(d) = L_{Latency} + L_m * d + M_{overhead} \quad \textbf{(1)}$$

where $R_{Latency}$ is remote latency (ns), d is optical round-trip length (m), L_m is optical fiber latency per metre, $L_{Latency}$ is local memory access latency (ns) and $M_{overhead}$ is memory access overhead. According to Eq. 1, remote memory access latency is a linear function of local memory access latency and optical distance. However, a small fraction of memory response time has been added as both parallel and serial memory do not respond to requests uniformly. Experimentally, we have measured that $M_{overhead}$ is 16.2 ns while writing and 37.4 ns when reading. Using Eq. 1, a best-case round-trip write latency of 854.4 ns and read latency of 870 ns for remote serial memory, round-trip write latency of 1.03 μs and read latency of 1.23 μs for remote parallel memory have been theoretically measured for 100 m round-trip optical distance at 15 Gb/s lane rate as depicted in Fig 6.

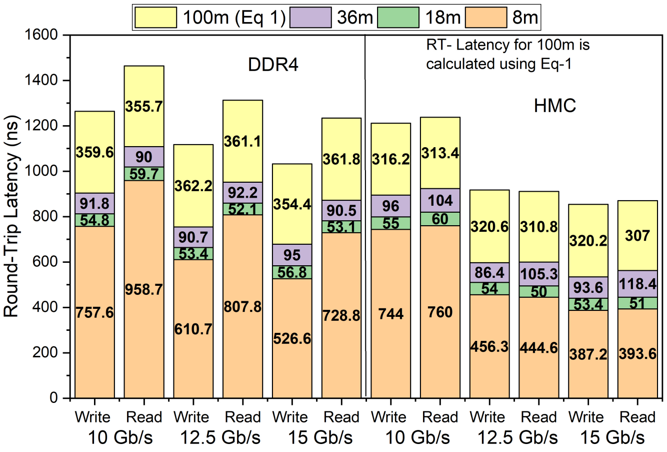

Fig. 6. Remote memory read/write latency: Impact of optical distance b/w CPU and remote memory on round-trip latency. (Values are measured experimentally for 8, 18 and 36 m; only 100 m is based on Eq. 1)

The maximum achievable memory bandwidth by DDR4, used in our experiment, is 17 GB/s [38]. The local DDR4 access performance with memory-mapped (MM) and MONet's streamer is showcased in Fig 7. MONet can support 8(1-lane), 16(2-lane), 32(4-lane), 64(8-lane) bytes streamed data width (DW) without transceivers and optical data path, allowing us to measure MONet's impact on sustainable memory bandwidth. The impact of transceiver lane rates on remote memory performance, link and memory bandwidth utilization at 8 m round-trip distance with streamer and bonded lanes are also shown in Fig 7. While maximum streamed data width of 64 bytes is used, MONet sustains a bandwidth of 53.5% to the local MM mode. This degradation is due to the conversion from five independent memory mapped channels to two stream channels (CPU-to-memory and memory-to-CPU) [39]. A further degradation of 2-18% has been reported while accessing remote memory using 8 transceivers lanes at rates 10, 12.5 and 15 Gb/s, which is mostly due to the Aurora IP [22] and 8-metres round-trip optical distance. Achieved bandwidth in terms of memory and link utilization has been also depicted in Fig 7, showing the impact of number of transceivers and line rate. The worst utilization of memory bandwidth (black line) for remote parallel memory is 10 % using single transceiver at 10 Gb/s rate whereas a best utilization of 47.6 % is achieved when 8-transceivers each at 15 Gb/s rate are used. While best link utilization (red line) of 70.4 % is achieved at 12.5 Gb/s lane rate using single transceiver and worst link utilization is achieved at 15 Gb/s lane rate using 8-transceivers together. Impact of memory and link utilization on remote memory access performance is discussed later on.

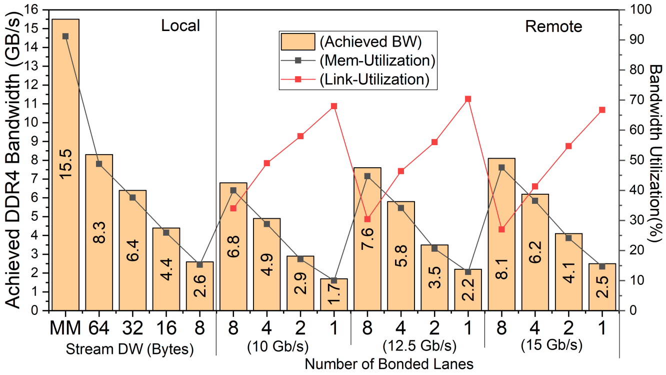

Fig. 7. Achieved bandwidth, link and memory bandwidth utilization for locally and remotely (8m) attached DDR4: transceiver lanes at rates (10, 12.5 and 15 Gb/s)

For HMC, the maximum achievable bandwidth is limited by the vault controllers that can offer a maximum of 160 GB/s memory bandwidth (10 GB/s per vault controller and 16 vault controllers grouped in four quadrants). Since our setup supports a single host accessing memory over a single link (one quadrant), only 40 GB/s memory bandwidth can be theoretically achieved [29]. Though MONet has the capability to support configurable memory access granularity from 8 byte to 128-byte, we choose maximum access granularity (128-byte) to reduce number of write/read cycles. As shown in Fig 8, more than 94.5% throughput is achieved at 15 Gb/s lane rate in full-width configuration for locally attached HMC, while dropping to 88.5% at 12.5 Gb/s and 72.5% at 10 Gb/s. Fig 8 shows the impact of lane rate and memory access configuration on memory and link bandwidth utilization. In half-width configuration at 15 Gb/s, maximum local/remote link utilization is 75.3/75%, while maximum memory bandwidth utilization is 56.5/56.25%. As shown in Fig 7 and Fig 8, a sustained bandwidth above 82% and 99% are achieved when accessing remote (8 m) DDR4 and HMC respectively demonstrating the efficient performance of MONet hardware with 8-transceivers. As observed by the behaviour of the black and red lines in Fig 7, the access of remote DDR4 memory with high memory utilization comes at the price of low link utilization and vice versa. In contrast to this, Fig 8 shows that the link utilization (red line) in remote HMC memory is maintained at above 70% and is independent of memory bandwidth utilization, achieving improved memory-link utilization ratio.

As shown in Fig 7 and Fig 8, both in DDR4 and HMC, the highest communication/link bandwidth can not guarantee 100% memory throughput. As already discussed, DDR4 and HMC can offer up to 17 GB/s and 40 GB/s memory bandwidth respectively. The link capacity for the CPU-to-DDR4 interconnect varies from 2.5 to 30 GB/s while varying from 30 (half-width) to 60 GB/s (full-width) for CPU-to-HMC interconnect. Thus in both cases, memory-to-link utilization ratio varies with transceiver count as shown in Fig 9. Higher memory-to-link utilization ratio show memory bandwidth saturation while lower memory-to-link utilization ratio show link capacity saturation. As already discussed, memory utilization is dependent on link-utilization for DDR4 memory and independent of it for HMC memory. Hence, we see that while memory-to-link ratio (blue line in Fig 9) varies between 0.14-1.76 for DDR4, it is tightly packed between 0.5-1.5 for HMC. This further affects energy efficiency per bit (pJ/bit). Power consumption and breakdown against number of transceivers used between optically connected CPU and memory has also been depicted in the bottom of Fig 9. While using parallel memory, power contribution due to optical data-path, switch and transceivers varies from 14.4% to 54.4% using single and eight bonded-channels link, respectively. In case of serial memory, power contribution due to I/O, switches and optical data-path varies from 36.1% to 56.3% using half and full-width links. This is due to increased number of transceivers (in both CPU and memory) and thus, optical switches ports count and data-path including MBOs. In both cases, almost 32.2% to 60.8% of whole of MONet's power are consumed by fixed resources such as CPU (ARM QUAD core), parallel memory chip (MTA8ATF51264HZ-2G1) and serial memory (MT43A4G40200). Our proprietary hardware ReMAT contributes only up to 7.3% (for serial memory) and 25% (parallel memory). Independent of memory type, the number of transceiver links and lane rate have higher memory-to-link ratio and, in turn, have worse energy efficiency. We have experimen-

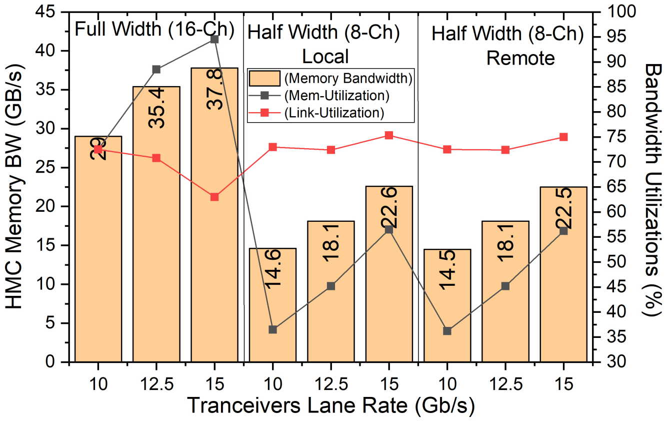

Fig. 8. Achieved bandwidth, link and memory bandwidth utilization for locally and remotely (8 m) attached HMC, compared with achieved maximum memory bandwidth for different lane rates.

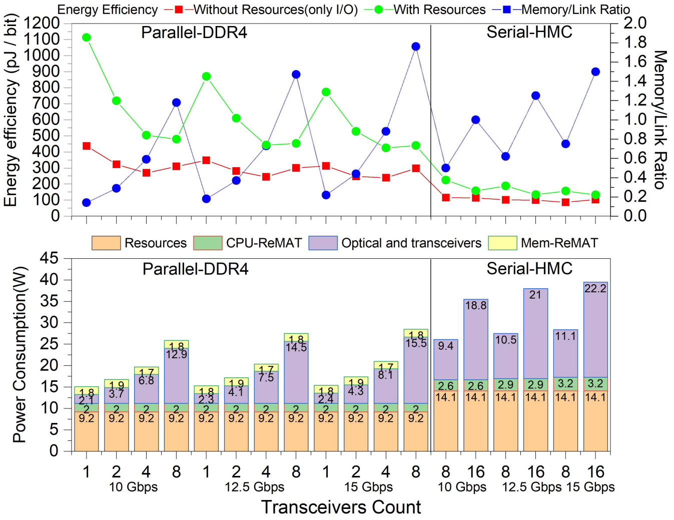

Fig. 9. Bottom: Power consumption distribution between CPU and memory over 8-metre round-trip optical data path. **Top:** Round-trip net energy efficiency (with and without MONet's resources) and memory-to-link ratio over number of transceivers link and lane rate.

Table 2. Normalized power consumption for eight transceivers lane between CPU and memory over 8-metres round-trip optical distance. * = Dynamic power contributor, $ = Experimentally measured, # = Serial transceivers and memory controller are embedded on memory chip, (E) = Electronic. OE = Opto-Electronic.

MONet's Resources	**Average Power Consumption (W) & Contribution (%)**					
	DDR4			**HMC#**		
	10	**12.5**	**15**	**10**	**12.5**	**15**
ARM 4-Core $	2.6 (10)	2.6 (9.5)	2.6 (9.1)	2.65 (10.1)	2.65 (9.7)	2.65 (9.3)
CPU ReMAT$*	1.98 (7.6)	1.98 (7.2)	1.98 (6.9)	2.6 (9.9)	2.85 (10.4)	3.15 (11.1)
MEM ReMAT$*	1.83 (7.0)	1.83 (6.7)	1.83 (6.4)	0	0	0
DDR4 I/O $	0.58 (2.2)	0.58 (2.1)	0.58 (2.0)	0	0	0
Memory [29, 38]	6.6 (25.4)	6.6 (24)	6.6 (23.1)	11.45 (43.8)	11.45 (41.7)	11.45 (40.4)
CPU (E) Trans.$*	2.3 (8.9)	2.7 (9.9)	3.0 (10.6)	1.8 (7)	2.1 (7.6)	2.3 (8.2)
OE Trans.*[33]	6 (23.1)	6.8 (24.7)	7.2 (25.2)	6 (23)	6.8 (24.8)	7.2 (25.4)
Optical Swit.*[28]	1.6 (6.2)	1.6 (5.8)	1.6 (5.6)	1.6 (6.1)	1.6 (5.8)	1.6 (5.6)
MEM (E) Trans. $*	2.5 (9.6)	2.8 (10.2)	3.1 (10.9)	0	0	0
Total Power	**26**	**27.5**	**28.5**	**26.1**	**27.4**	**28.4**
Achieved Rate (GB/s)	**6.8**	**7.6**	**8.1**	**14.5**	**18.1**	**22.6**
Net Energy Effi. (pJ/bit)	**477.3**	**452.4**	**440.5**	**225.3**	**189.4**	**156.9**

tally measured energy efficiency in two cases (a) with MONet's resources (CPU and memory chips) and (b) without MONet's resources (only due to I/O and optical data-path). In second scenario using 8 transceiver's link, the energy efficiency for serial memory varies between 86.6 to 115.7 pJ/bit which is more efficient than parallel memory which is between 239.2 to 436 pJ/bit. For the power consumption values showcased in Fig 9, we have given a detailed breakdown of the components and their individual power values in Table 2, which we used to estimate the overall power. In Table 2, the metrics that are marked with $ indicate the values measured experimentally while cited references indicate the use of external documents or datasheets to estimate the power. A normalized average power utilization for individual MONet's resources has been shown in Table 2 which shows that serial memory consumes more power (40.3-43.8% of MONet's total power) compared to parallel memory (23.1-25.3% of MONet's power) but offers 156.9-225.3 pJ/bit/8-lane (35.6-47.2% more efficient than parallel DDR4). Thus, choice of memory can be made specific to type of application. If application demands high bandwidth but low to moderate memory size, HMC memory is best fit otherwise DDR4 memory is suitable for applications that require high memory size with moderate bandwidth and low power. As shown in Table 2, the total power consumed, by 8 electro-optical transceivers (7.2 W at 15 Gb/s) and the optical switch (1.6 W) in the data path, is measured as 8.8 W. This power consumption can be reduced to ≈ 5.5 W by using an electrical interconnect as reported in the MACOM (MAXP-37161A) documentation [40]. Employing optical communication (transceivers and switching) with the technologies used on this manuscript consumes 60% more power than the use of electronic communication. However, optical switches can support a broad spectrum of optical channels so can better scale to support higher bit rates. In case we use WDM transceivers all channels can be switched by the same port further reducing the power consumption. In addition, electrical interconnects have very short reach (2-3 metres; intra-rack distance) and as we aim to disaggregate physical memory over 10s-100s of metres the use of optical interconnects.

5.C. Physical Layer Characterization

Memory disaggregation in MONet requires an error free optical communication between resources without inline forward error correction (FEC) as the presence of FEC can potentially introduce more than 100 ns of latency, degrading the overall performance. Though, Fig 10 presents the bit error rate (BER) performance for 10, 12.5 and 15 Gb/s bi-directional optical link between CPU and serial memory, an error free optical network between CPU and parallel memory has been already demonstrated in [14]. Provided that each SiP transceivers in the MBO has an average output power of -3 dBm, it is evident that, on average, the SMF and the optical switch can be extended to have an approximate total power budget between 4-11 dB (bit-rate dependent) for BER of 10^{-12}, enabling the emulation of a multi-tier topology considering 1dB insertion loss per switching hop. However, the number of tiers allowed depends strictly on the serial link rate; for example, using 10 Gb/s link connectivity between resources can perform error free requires an optical power of -13.9 dB and supports up to 4-tier (7 hops) while 12.5 and 15 Gb/s links require an optical power of -6.5 dB and can only perform error free up to 2-tier (3 hops). As shown in Fig 4 and experimentally proven error free link within 2-tier topology allow MONet to scale up and support to 100,000 of optically attached serial memory modules. Although re-transmission of memory request/response is enabled in case of packet loss,

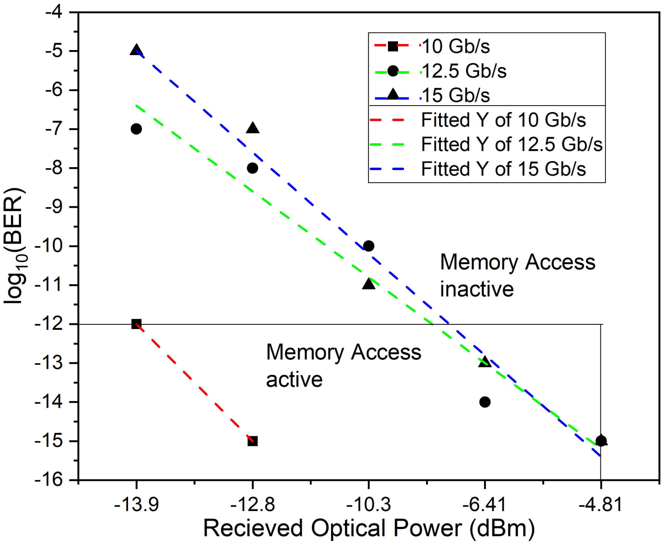

Fig. 10. Physical layer performance of a single bi-directional channel CPU and HMC: Received optical power (dBm) vs $\log_{10}$(BER).

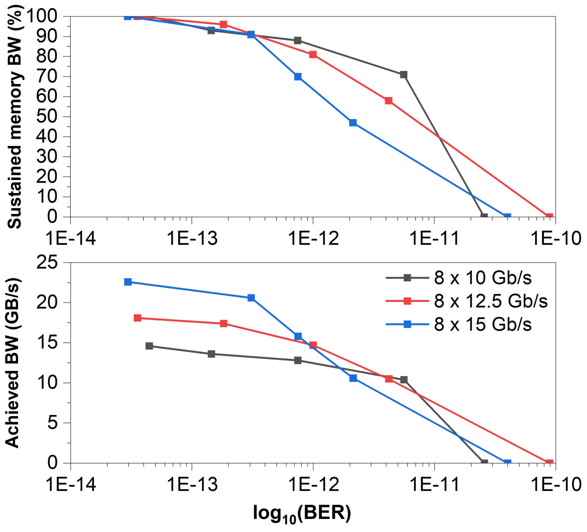

Fig. 11. Impact of Bit Error Rate (BER) on memory bandwidth performance per one HMC half width link (8 transceivers).

interconnection with zero error is essential to achieve high bandwidth. To measure the BER of the end-to-end link, we used Xilinx IBERT IP core [41] with PRBS31 generator on the compute card, which sends the data (CPU-TX) to the HMC (HMC-RX). The loopback feature in the HMC was enabled to resend the received data (HMC-TX) back to the compute card (CPU-RX), where the PRBS31 checker checked for errors. We used optical attenuators to operate at the five optical received power values shown by the x-axis of Fig. 10. At each received optical power, we measured the BER across across all transceivers as well as the resulting throughput performance, as shown in Fig 11. As observed, a BER of 10^{-13} and 10^{-12} is required to operate beyond 95% and 60% sustained memory bandwidth respectively. At a BER of 10^{-12}, a sustained bandwidth of 85.7%, 81.2% and 63.6% is achieved for 8× 10, 12.5 and 15 Gb/s links respectively. The highest penalty occurs in the 8×15 Gb/s link because the the opto-electronic receivers at host and the HMC cards require higher input optical power (has worse receiver sensitivity) at higher channel rates. In Fig 11, bandwidth is shown to drop by more than 50% when BER is more than 10^{-12} due to re-transmission of memory request/response packets. Re-transmissions delay the completion of transactions and also engage the link so that other request/response cannot be served. A steep declining slope has been observed when remote memory is accessed using lane rate 12.5 and 15 Gb/s; however, a consistent performance can be achieved using 10 Gb/s lane rate.

5.D. Sustained Memory Bandwidth Characterization

To evaluate memory throughput at the application layer, a STREAM (10 million array elements, requiring 228.9 MB) test [17] is used. The STREAM test is an industrial standardized subroutine used to evaluate the sustainable memory bandwidth in high performance computing systems that runs on Linux OS on the ARM processor of the CPU card. This is achieved by measuring the perceived throughput from/to the attached memory resource while carrying four logical operations: COPY (1-read, 1-write, 1-FLOP), SCALE (1-read, 1-write, 1-FLOP/interactions), ADD (2-read, 1-write, 1-FLOP/interactions) and TRIAD (2-read, 1-write, 2-FLOPs/interactions) running on 4 CPU cores. Even though MONet is fully pipelined, the maximum theoretical memory throughput that the ARM processor can offer using one port in full duplex mode (write + read) is around 10.66 GB/s, which comes from 333 MHz × 128 bits × 2 directions (read + write) = 10.656 GB/s. Even though the ARM's maximum throughput is impossible to achieve, DDR4 and HMC sustains at 30% and 23.5%, respectively, while attached locally using STREAM benchmark as shown in Fig 12 and Fig 13.

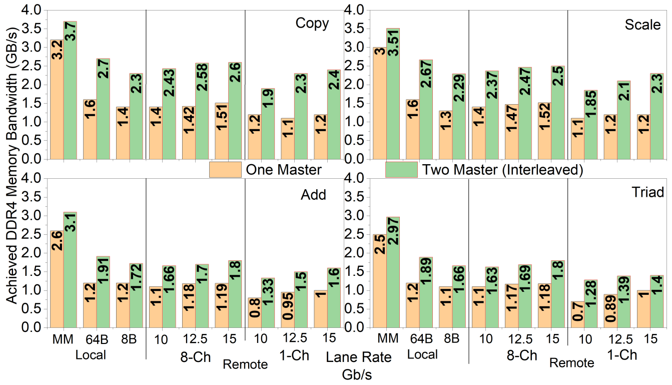

Fig. 12. STREAM benchmark performance for DDR4 at 8-metres round-trip distance using 8 and single channel.

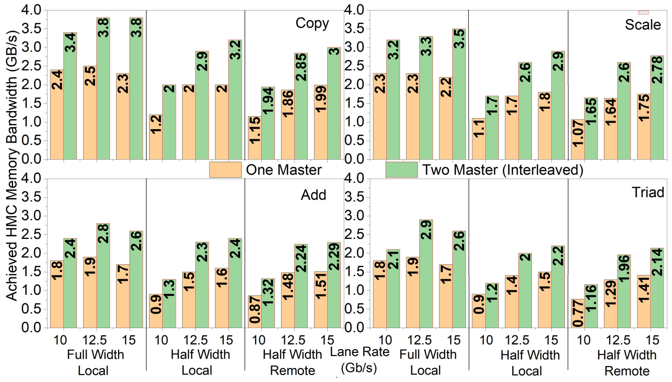

Fig. 13. Application level performance using the STREAM benchmark for accessing serial HMC memory (local and remote at 8 metres round-trip) at 10, 12.5 and 15 Gb/s lane rate.

Furthermore, this also limits STREAM benchmark functions to achieve higher memory throughput for remotely attached DDR4/HMC memory (full and half width, at lane rates of 10, 12.5, 15 Gb/s) using all four cores. Degradation in performance is reasonable as STREAM benchmark includes CPU overhead as well as kernel and application overheads along with transceivers and optical data path latency. The penalty of using optical interconnects and serial channels on bandwidth for both remote DDR4 (single and 8-links) and HMC (8-links) has been depicted in Fig 12 and Fig 13 respectively, where memory bandwidth sustains at 70% (DDR4) and 94% (HMC) at 8-metres round-trip distance using 8-serial links at 15 Gb/s lane rate. While extending the remote memory to a round-trip distance of 36-metres, sustained memory bandwidth for serial HMC sustains at 89, 92.2 and 91.1% at link rates 10, 12.5 and 15 Gb/s, whereas sustained memory bandwidth for DDR4 is 51.6, 61 and 62.2% using STREAM benchmark respectively, as shown in Fig 14. A minimal reduction in baseline memory performance for HMC can be seen from 1.25% at 10 Gb/s lane rate to 4.5% at 15 Gb/s lane rate up to 36 metres round-trip distance, as shown in Fig 14. While extending DDR4 up to round-trip distance of 36-metres can be costly as 41 to 48% reduction in baseline throughput is observed.

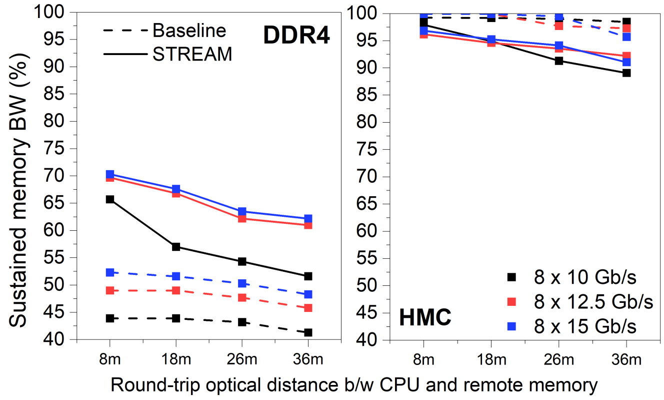

Fig. 14. Sustained STREAM and baseline bandwidth (8-links) over round-trip over round-trip optical distance: 8, 18, 26, 36 metres.

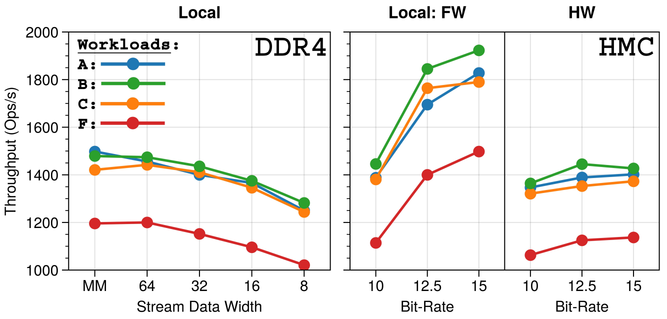

Fig. 15. Achieved Throughput in Workload (A, B, C and F) when both memories are locally attached. For DDR4: parallel accessed (MM), stream data-width size in bytes (8 to 64). For HMC: full-width (FW) (16-lane) and half-width (HW) (8-lane) at 10, 12.5 and 15 Gb/s bit-rates.

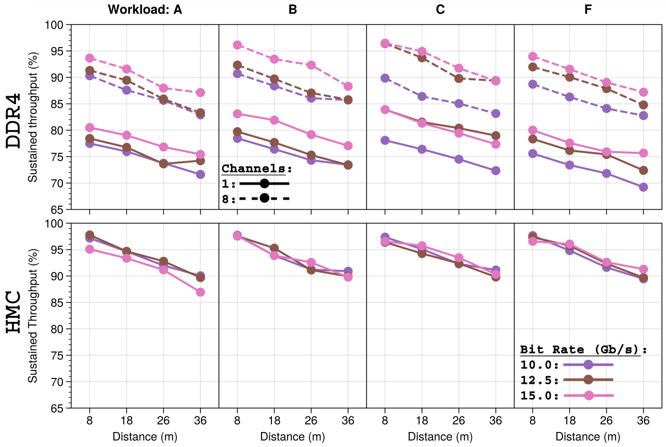

Fig. 16. Sustained Throughput in Workload (A, B, C and F) when both memories are remotely attached at round-trip optical distance 8, 16, 26 and 36-metres. For DDR4: using 8 and one transceivers links each at 15 Gb/s. For HMC: Half-width (8-lane) at 10, 12.5 and 15 Gb/s bit-rates.

This reduction in throughput for DDR4 is due to the fact that local attachment uses only parallel interface (memory-mapped) and offers higher throughput (15.5 GB/s).

5.E. Cloud-Workload Characterization

As MONet is proposed as a memory disaggregation solution in data center networks, it is vital to evaluate MONet's serial/parallel memory local/remote-access performance under cloud workloads. In this section, we have used YCSB (Yahoo! Cloud Serving Benchmark) [42], a popular cloud service workload tool, and an open source, high-performance, distributed in-memory object caching database, memcached [43]. The YCSB tool (1) reads a set of predefined workloads (A-to-F), (2) generates and loads the data sets, (3) runs the operations specified in the workload file, and finally, (4) collects the performance for the load and run phase. In YCSB's configuration, MONet runs 4 threads and 10K operations to evaluate the performance of locally/remotely attached memories across four distinct workloads (A,B,C and F). Each YCSB workload has a unique ratio of operations as shown in [42]: A - Update heavy (50% Read and 50% Update), B - Read heavy (95% Read and 5% Update), C - Read only (100% Read) and F - Read-Modify-Write (50% Read, 25% Update and 25% Read-Modify-Write). The YCSB is also configured to follow the 'zipfian' distribution while accessing the records during the run phase.The comparison among local/remote, serial/parallel memories are made based on throughput (operations/second) and average latency observed in each workload.

As shown in Fig 15, the locally attached HMC, in its best configuration (FW at 15 Gb/s), achieves a minimum throughput of 1498 operations/second (ops/s) in workload-F (Read-Modify-Write) and a maximum throughput of 1923 ops/s in workload-C (Read Only), which is 1.2-1.3 × more compared to locally attached memory mapped (MM) DDR4. When dropping down from FW to HW configuration, the HMC throughput decreases to 1137-1427 ops/s, which is 94-97% of what memory mapped DDR4 achieves. The sustained throughput achieved when extending both memories to a round-trip distance of upto 36 metres is shown in Fig 16. In all four workloads, at 36 metres, serial memory averages around 90% while parallel memory averages around 88% and 76% of their local counterpart when using 8-transceivers and single-transceiver respectively at 15 Gb/s/transceiver. The sustained throughput gradient across all workloads for DDR4 8-bonded channels and single channel at 15 Gb/s is -0.25 and -0.22%/m, whereas for HMC it is -0.24%/m. The throughput drop in remote-access DDR4 parallel memory is higher and attributed to the 'MM to stream conversion (MM2S)' hardware logic and Aurora transceivers which increases added hardware latency (40× write and 14× read latency) while latency added in serial memory is purely due to propagation delay and optical distance (1.60× write and 1.64× read latency) as shown in Fig 5 and Fig 6.

The added hardware latency also affects the overall application level average latency in DDR4 remote access for all YCSB workloads A, B, C and F, as depicted in Fig 17. Latency is as high as 3.53, 2.71 and 3.46 ms in case of memory mapped DDR4, locally attached FW and HW HMC memory respectively in workload-F at 15 Gb/s. However, for other workloads, the latency averages around 2.22, 1.78 and 2.32 ms. In workload F, the latency is higher due to a two-phase operation: (a) the Read phase + (b) the Read-Modify-Write/Update phase. Extending DDR4 memory up to 36-metres optical distance, increases average latency by 20% (8-transceivers) and 45% (single-transceiver) compared to memory mapped configuration as shown in Fig 18. In contrast, the added latency for extending HMC by up to 36 metres is only 10% in HW configuration at 15 Gb/s. The increase in added latency is found to be 0.29 and 0.32%/m across all workloads for DDR4 8-bonded channel and single channel

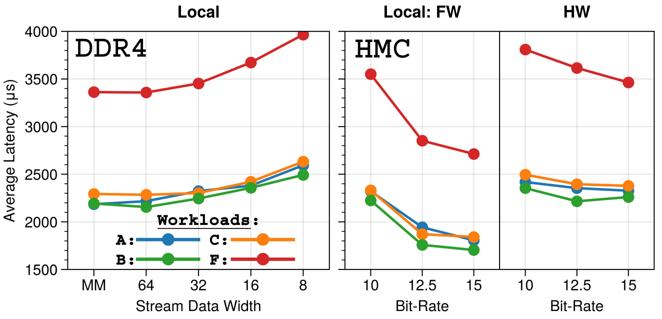

Fig. 17. Achieved Average Latency in Workload (A, B, C and F) when both memories are locally attached. For DDR4: parallel accessed (MM), stream data-width size in bytes (8 to 64). For HMC: full-width (FW) (16-lane) and half-width (HW) (8-lane) at 10, 12.5 and 15 Gb/s bit-rates.

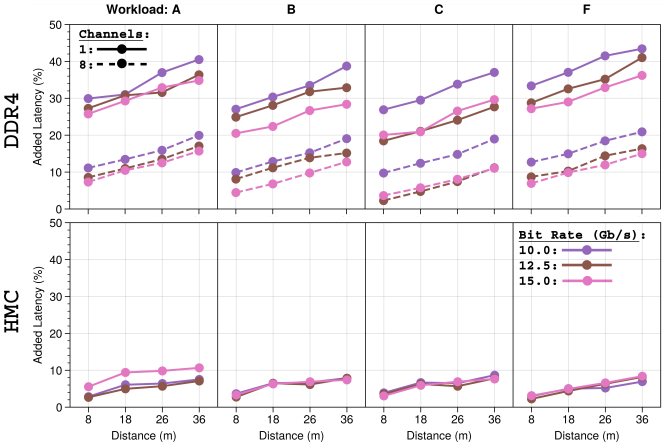

Fig. 18. Sustained Average Latency in Workload (A, B, C and F) when both memories are remotely attached at round-trip optical distance 8, 16, 26 and 36-metres. For DDR4: using 8 and one transceivers links each at 15 Gb/s. For HMC: Half-width (8-lane) at 10, 12.5 and 15 Gb/s bit-rates.

respectively at 15 Gb/s, while for HMC it averages around 0.17%/m.

To better understand the impact of type of memory and round-trip optical distance on CPU performance, we have also profiled YCSB workloads in each scenario using Linux *perf* tools [44]. We have measured the average retired instruction per cycle (IPC) for workloads (A, B, C and F) when both memories are locally attached. To measure the average IPC for ARM Cortex-A53, we have used instructions and cycles *perf* events. Average IPC for all workloads varies between 0.35-0.36 for locally attached memory mapped DDR4 and half-width HMC at 15 Gb/s, while increasing to 0.42-0.43 for full-width HMC. The increase in retired IPC correlates with previously shown low-latency and high-throughput behaviour exhibited by FW HMC. On the other hand, the impact of increasing optical distance across various workloads, channels and line-rates on average IPC is shown in Fig 19. As latency increases over distance, the sustained IPC reduces at a rate of 0.53%/m to 70-83% at 36 metres round-trip distance for both type of memories.

6. CONCLUSIONS

In this paper, we proposed and experimentally demonstrated a novel FPGA-based optically disaggregated network architecture

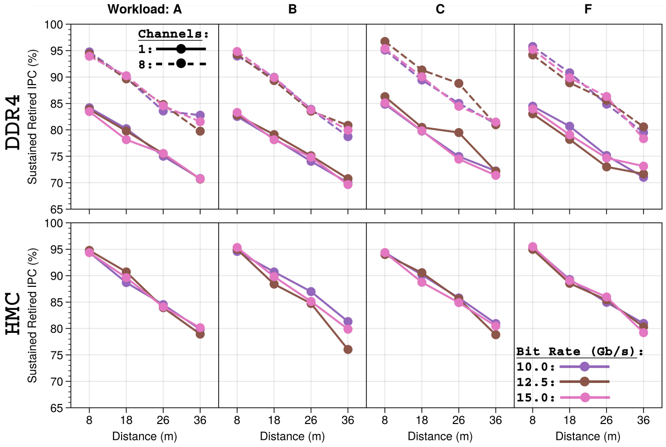

Fig. 19. Impact of optical distance on IPC in workload (A,B, C and F) for whole CPU. For DDR4: using 8 and one transceivers links each at 15 Gb/s. For HMC: Half-width (8-lane) at 10, 12.5 and 15 Gb/s bit-rates.

(MONet), showcasing locally/remotely attached serial/parallel memory access. This architecture is the first of its kind to demonstrate the support of both parallel DDR4 and serial HMC memory for disaggregated data center networks. We demonstrated the CPU-memory performance of both memory types, evaluated pros/cons and thoroughly reviewed and characterized the physical layer requirements, the baseline performance, industry-standard STREAM benchmark performance and performance under YCSB-based cloud workloads. Overall, the light nature of the MONet's hardware structure outperforms all disaggregated data center network competitors in terms of bandwidth, latency, power consumption and architecture. We showcased for the a number of resources and resource plane ratios, MONet supports lower hops (3-hops), lower latency and consumes lower power (< 33%) compared to other Non-Parallel architectures. In MONet, remote memory access latency were shown to be about 678.4 ns for DDR4 memory and 534.2 ns, with minimal local-remote penalty, for HMC memory. MONet achieves a maximum memory bandwidth of 8.1 GB/s for remote DDR4 and 22.5 GB/s for remote HMC. MONet's energy efficiency was shown to have a strong correlation with memory/link utilization, consuming a maximum of 239.2 pJ/bit and 86.6 pJ/bit for remote DDR4 and HMC memory respectively. MONet's physical layer was characterized and we demonstrated FEC free operation for a 2-tier and 4-tier topology at 12.5/15 and 10 Gb/s operable lane rates respectively ensuring 100% of data center resources can be accessed within 3-hops. At the end, we have characterised disaggregated local/remote memory with memcached benchmark using four distinct workloads (A, B, C and F) running on four concurrent clients initiated by YCSB and we analysed the throughput (ops/s), average latency and retired IPC. Based on experimental results, MONet shows that locally attached DDR4 and HMC (in half-width) have similar performance: throughput ≈ 1400 ops/s, average latency ≈ 2.5 ms and IPC ≈ 0.36. However, in full-width configuration mode (locally attached), HMC outperforms both MM DDR4 and HW HMC. Our experiments demonstrate that remotely attached serial memory has less impact on performance up to 36-metres optical round-trip distance. The proposed architecture advocates use of optically attached serial memory as the ideal candidate due to (1) higher sustained bandwidth (95.7% using baseline and 91.1% using STREAM benchmark with ability to deliver average low write-read latencies (only 196-230.4 ns extra round-trip latency over 36 metres round-trip distance) at 15 Gb/s lane rate, (2) higher sustained throughput across all YCSB workloads (more than 90%) with very low additional contribution to average latency (only up to 10%) over 36 metres round-trip distance at 15 Gb/s, (3) elimination of an additional chip (e.g. ASIC, FPGA, MPSoC) to host the memory controller, due to serial communication and inherent compatibility with high-speed interconnects and network operation, that reduces cost, power consumption and footprint.

ACKNOWLEDGEMENTS

The work is supported by EPSRC TRANSNET (EP/R035342/1) programme grant and OptoCloud (EP/T026081/1) programme.

REFERENCES

1. Cisco, “95% of data centre traffic will come from cloud by 2021,” (2018). [Online]: https://www.cloudpro.co.uk/leadership/7304/cisco-95-of-data-centre-traffic-will-come-from-cloud-by-2021, accessed: April 2020.
2. J. Schmidt, H. Fröning, and U. Brüning, “Exploring Time and Energy for Complex Accesses to a Hybrid Memory Cube,” in *Proceedings of the Second International Symposium on Memory Systems,* (ACM, New York, NY, USA, 2016), MEMSYS ’16, pp. 142–150.
3. J. Schmidt and U. Bruning, “openHMC - a configurable open-source hybrid memory cube controller,” in *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig),* (2015), pp. 1–6.
4. S. Han, N. Egi, A. Panda, S. Ratnasamy, G. Shi, and S. Shenker, “Network support for resource disaggregation in next-generation datacenters,” in *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks,* (2013).
5. K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and T. F. Wenisch, “Disaggregated memory for expansion and sharing in blade servers,” in *ISCA ’09: Proceedings of the 36th annual international symposium on Computer architecture,* (2009).
6. L. A. Barroso and U. Hölzle, “The Case for Energy-Proportional Computing,” Computer **40**, 33–37 (2007).
7. S. M. Rumble, D. Ongaro, R. Stutsman, M. Rosenblum, and J. K. Ousterhout, “It’s time for low latency,” in *Proceeding HotOS’13 Proceedings of the 13th USENIX conference on Hot topics in operating systems,* (2011).
8. S. Liang, R. Noronha, and D. K. Panda, “Swapping to Remote Memory over InfiniBand: An Approach using a High Performance Network Block Device,” in *2005 IEEE International Conference on Cluster Computing,* (2005), pp. 1–10.
9. J. Gu, Y. Lee, Y. Zhang, M. Chowdhury, and K. G. Shin, “Efficient memory disaggregation with infiniswap,” in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI),* (2017), pp. 649–667.
10. J. Vienne, J. Chen, M. Wasi-Ur-Rahman, N. S. Islam, H. Subramoni, and D. K. Panda, “Performance Analysis and Evaluation of InfiniBand FDR and 40GigE RoCE on HPC and Cloud Computing Systems,” in *2012 IEEE 20th Annual Symposium on High-Performance Interconnects,* (2012), pp. 48–55.
11. Martine J. Silbermann, “Resizing Memory With Balloons and Hotplug,” in *Linux Symposium,* (2006), pp. 305–314.
12. P. S. Rao and G. Porter, “Is memory disaggregation feasible? A case study with Spark SQL,” in *2016 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS),* (2016), pp. 75–80.
13. G. Zervas, F. Jiang, Q. Chen, V. Mishra, H. Yuan, K. Katrinis, D. Syrivelis, A. Reale, D. Pnevmatikatos, M. Enrico, and N. Parsons, “Disaggregated compute, memory and network systems: A new era for optical data centre architectures,” in *2017 Optical Fiber Communications Conference and Exhibition (OFC),* (2017), pp. 1–3.
14. A. Saljoghei, V. Mishra, M. Bielski, I. Syrigos, K. Katrinis, D. Syrivelis, A. Reale, D. N. Pnevmatikatos, D. Theodoropoulos, M. Enrico, N. Parsons, and G. Zervas, “dRedBox: Demonstrating Disaggregated Memory in an Optical Data Centre,” in *2018 Optical Fiber Communications Conference and Exposition (OFC),* (2018), pp. 1–3.
15. F. Lin and B. Keeth, “Memory Interface Design for Hybrid Memory Cube (HMC),” in *2016 IEEE Workshop on Microelectronics and Electron Devices (WMED),* (2016), pp. 1–5.
16. V. Mishra, J. L. Benjamin, and G. Zervas, “Demonstrating optically interconnected remote serial and parallel memory in disaggregated data centers,” in *2020 Optical Fiber Communications Conference and Exhibition (OFC),* (2020), pp. 1–3.
17. J. D. McCalpin, “STREAM: Sustainable Memory Bandwidth in High Performance Computers,” [Online]: https://www.cs.virginia.edu/stream/, accessed: March 2020.
18. S. Novakovic, A. Daglis, E. Bugnion, B. Falsafi, and B. Grot, “Scale-out numa,” in *Proceedings of the 19th international conference on Architectural support for programming languages and operating systems,* (2014).
19. Y. Yan, G. M. Saridis, Y. Shu, B. R. Rofoee, S. Yan, M. Arslan, T. Bradley, N. V. Wheeler, N. H. Wong, F. Poletti, M. N. Petrovich, D. J. Richardson, S. Poole, G. Zervas, and D. Simeonidou, “All-Optical Programmable Disaggregated Data Centre Network Realized by FPGA-Based Switch and Interface Card,” J. Light. Technol. **34**, 1925–1932 (2016).
20. H. Meyer, J. C. Sancho, J. V. Quiroga, F. Zyulkyarov, D. Roca, and M. Nemirovsky, “Disaggregated computing an evaluation of current trends for datacenters,” in *Procedia Computer Science,* vol. 108 (2017), pp. 685–694.
21. D. Brunina, C. P. Lai, D. Liu, A. S. Garg, and K. Bergman, “Optically-connected memory with error correction for increased reliability in large-scale computing systems,” in *OFC/NFOEC,* (2012), pp. 1–3.
22. Xilinx, “Aurora 64B/66B,” (2018). [Online]: https://www.xilinx.com/products/intellectual-property/aurora64b66b.html, accessed: February 2020.
23. D. Syrivelis, A. Reale, K. Katrinis, I. Syrigos, M. Bielski, D. Theodoropoulos, D. N. Pnevmatikatos, and G. Zervas, “A software-defined architecture and prototype for disaggregated memory rack scale systems,” in *2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS),* (2017), pp. 300–307.
24. D. Theodoropoulos, A. Reale, D. Syrivelis, M. Bielski, N. Alachiotis, and D. Pnevmatikatos, “REMAP: Remote mEmory Manager for disAggregated Platforms,” in *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP),* (2018), pp. 1–8.
25. D. Syrivelis, A. Reale, and M. Gazzetti, “Dynamic synthesis of disaggregated hardware platforms via cache coherent interconnect optical bridge,” in *2019 European Conference on Optical Communication (ECOC),* (2019).
26. C. Pinto, D. Syrivelis, M. Gazzetti, P. Koutsovasilis, A. Reale, K. Katrinis, and H. P. Hofstee, “ThymesisFlow: A Software-Defined, HW/SW co-Designed Interconnect Stack for Rack-Scale Memory Disaggregation,” in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO),* (2020), pp. 868–880.
27. S. K. Sadasivam, B. W. Thompto, R. Kalla, and W. J. Starke, “IBM Power9 Processor Architecture,” IEEE Micro **37**, 40–51 (2017).
28. HUBER+SUHNER Polatis, “All Optical Switches,” (2020). [Online]: https://www.polatis.com/, accessed March 2020.
29. Micron, “Hybrid Memory Cube – HMC Gen2,” (2018). [Online]: https://www.micron.com/-/media/client/global/documents/products/data-sheet/hmc/gen2/hmc_gen2.pdf, accessed: January 2020.
30. Xilinx, “Zynq UltraScale+ RFSoC ZCU111 Evaluation Kit,” (2019). [Online]: https://www.xilinx.com/products/boards-and-kits/zcu111.html, accessed: August 2020.
31. V. Mishra, Q. Chen, and G. Zervas, “REoN: A protocol for reliable software-defined FPGA partial reconfiguration over network,” in *2016 International Conference on ReConFigurable Computing and FPGAs (ReConFig),* (2016), pp. 1–7.
32. M. Neitz, M. Wöhrmann, R. Zhang, M. Fikry, S. Marx, and H. Schröder, “Design and Demonstration of a Photonic Integrated Glass Interposer

for Mid-Board-Optical Engines," in *2017 IEEE 67th Electronic Components and Technology Conference (ECTC),* (2017), pp. 538–544.

33. Luxtera (now Cisco), "Mid-Board Optic Transceivers," (2018). [Online]: http://www.luxtera.com/embedded-optics/, accessed: December 2018.
34. Xilinx, "High Speed Serial," (2019). [Online]: https://www.xilinx.com/products/technology/high-speed-serial.html, accessed: March 2020.
35. HiTech Global, "Xilinx Zynq MPSoC UltraScale+ Board," (2019). [Online]: http://www.hitechglobal.com/Boards/MPSOC_UltraScale+.html, accessed: November 2020.
36. HiTech Global, "2GB or 4GB Hybrid Memory Cube (HMC) FMC+ Module (VITA57.4)," (2020). [Online]: http://www.hitechglobal.com/FMCModules/FMC_HMC.html, accessed: November 2020.
37. G. Zervas, H. Yuan, A. Saljoghei, Q. Chen, and V. Mishra, "Optically Disaggregated Data Centers With Minimal Remote Memory Latency: Technologies, Architectures, and Resource Allocation," J. Opt. Commun. Netw. **10**, A270–A285 (2018).
38. Micron, "DDR4 SDRAM SODIMM," (2018). [Online]: https://www.micron.com/products/dram-modules/sodimm/part-catalog/mta8atf51264hz-2g1, accessed: January 2020.
39. Xilinx, "AXI Memory Mapped to Stream Mapper v1.1," (2017). [Online]: https://www.xilinx.com/support/documentation/ip_documentation/axi_mm2s_mapper/v1_1/pg102-axi-mm2s-mapper.pdf, accessed: October 2019.
40. MACOM, "Macom-Product Detail-MAXP-37161," [Online]: https://www.macom.com/products/product-detail/MAXP-37161/ accessed: October 2020.
41. Xilinx, "IBERT for UltraScale GTH Transceivers v1.4," (2018). [Online]: https://www.xilinx.com/support/documentation/ip_documentation/ibert_ultrascale_gth/v1_4/pg173-ibert-ultrascale-gth.pdf accessed: December 2018.
42. B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with YCSB," in *SoCC '10: Proceedings of the 1st ACM symposium on Cloud computing,* (June 2010), pp. 143–154.
43. Dormando, "Memcached - a distributed memory object caching system," (2018). [Online]: http://www.memcached.org/ accessed: November 2020.
44. Arnaldo Carvalho de Melo, "The New Linux 'perf' tools," (2010). [Online]: http://www.linux-kongress.org/2010/slides/lk2010-perf-acme.pdf, accessed: December 2020.