

Streamlining CXL Adoption for Hyperscale Efficiency

Angelos Arelakis*

Nilesh Shah*

Yiannis Nikolakopoulos*

Dimitrios Palyvos-Giannas*

<first_name>@zptcorp.com

ZeroPoint Technologies AB, Gothenburg
Sweden

ABSTRACT

In our exploration of Composable Memory systems utilizing CXL, we focus on overcoming adoption barriers at Hyperscale, underscored by economic models demonstrating Total Cost of Ownership (TCO). While CXL addresses the pressing memory capacity needs of emerging Hyperscale applications, the escalating demands from evolving use cases such as AI outpace the capabilities of current CXL solutions. Hyperscalers resort to software-based memory (de)compression technology, alleviating memory capacity, storage, and network constraints but incurring a notable "Tax" on Compute CPU cycles. As a pivotal guide to the CXL community, Hyperscalers have formulated the groundbreaking Open Compute Project (OCP) Hyperscale CXL Tiered Memory Expander specification. If implemented, this specification lowers TCO adoption barriers, enabling diverse CXL deployments at both Hyperscaler and Enterprise levels. We present a CXL integrated solution, aligning with the aforementioned specification, introducing an energy-efficient, scalable, hardware-accelerated, Lossless Compressed Memory CXL Tier. This solution, slated for mid-2024 production and open for integration with Memory Expander controller manufacturers, offers 2-3X CXL memory compression in nanoseconds, delivering a 20-25% reduction in TCO for end customers without requiring additional physical slots. In our discussion, we pinpoint areas for collaborative innovation within the CXL Community to expedite software/hardware advancements for CXL Tiered Memory Expansion. Furthermore, we delve into unresolved challenges in Pooled deployment and explore potential solutions, collectively aiming to make CXL adoption a "No Brainer" at Hyperscale.

KEYWORDS

Composable Memory Systems, OCP Hyperscale Tiered Memory Expander Spec, CXL, Hardware Accelerated Lossless Memory Compression, Compressed Memory Tiers

1 INTRODUCTION

1.1 Background

DRAM is a key driver of performance and cost in the Data Center. CXL offers a path towards improved DRAM utilization (cost efficiency) in both Tiered and Pooled scenarios, but the Total Cost of system Ownership (TCO) increases non trivially due to added infrastructure components [Berger et al. 2023]. Moving/adding DRAM off the DDR interface to CXL requires significant return on investment and increased efficiency. The ground breaking OCP specification for

the Hyperscale CXL Tiered Memory Expander offers a crucial solution to achieve efficiency by specifying the addition of a compressed DRAM tier. Compression increases the effective capacity of a CXL memory device, reducing the power necessary to manufacture and operate DRAM components.

1.2 OCP Specification/ requirements

The OCP Specification [Chauhan et al. 2023] calls for a sustainable, *transparent* and cost-efficient method to compress memory on CXL Type 3 devices on a variety of compute platforms with a diversity of memory technologies. The OCP Spec calls for Access to a cache line in a compressed block within 250ns, with tail latency for accessing a cache line in an compressed block of <1us including worst case lookup latency, decompression, power-state transitions. Furthermore, decompression speeds of 46GB/s must be matched to 4 Channels at 1867MT/s compressed data, with 4kB/1kB blocks. State of the Art solutions fail to meet these requirements.

1.3 State of the Art

General-purpose, software-based lossless data "(de)compression" techniques are used widely in hyperscale systems to alleviate the memory, storage, and network cost with significant associated compute overheads "datacenter taxes" in warehouse-scale data-center services [Jeong et al. 2023; Karandikar et al. 2023]. These (de)compression services consume 2.9% to 4.6% of fleet CPU cycles and 10-50% cycles in key services. The existing (de)compression algorithm implementations (at block size granularity) do not meet the latency, power efficiency and preservation of software investment tenets spelled out in the OCP Specification. Waiting for an entire block or page prior to decompress data incurs latency measured in microseconds, which is suitable for storage but not acceptable for memory performance. Software and Intel QAT (De)compression solutions require the use of area and power intensive Xeon cores [Will 2023], falling order of magnitude short of the latency and bandwidth requirements specified in the OCP Spec. State of the Art hardware accelerators reported in open source (Zipline) or in other studies (CDPU) reported consuming 1.3-5.7mm sq area while achieving 5-11GB/s but with unspecified latency and performance scalability [Karandikar et al. 2023]. These designs are not geared /portable for integration into CXL Tiered Memory Expander devices as a complete solution to the best of our knowledge.

*All authors contributed equally to this research.

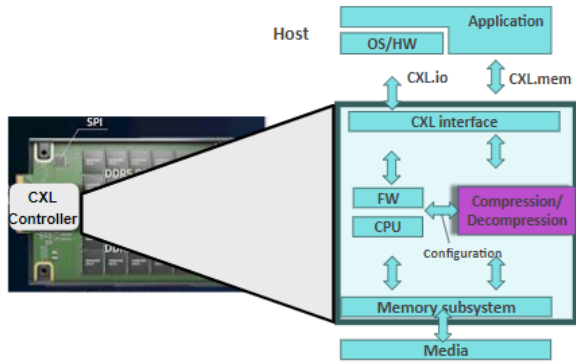


Figure 1: (De)Compression Solution integrated into CXL Expander Type 3 Device

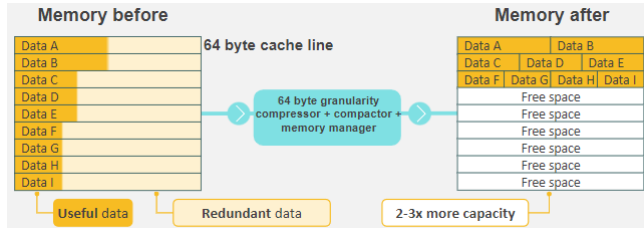


Figure 2: Cache line (64 byte) compression algorithm

2 SOLUTION

2.1 Invention

We present our Hardware accelerated, Lossless memory (De) Compression integrated IP solution [Figure 1] that increases the effective CXL Type 3 Device memory capacity by 2x through transparent, in-line memory compression/decompression at 64 byte cache line granularity. It is designed as an area and power-efficient solution composed of multiple integrated IP blocks, portable across the latest process nodes, supporting (LP)DDR4 and (LP)DDR5 memory technologies. This solution meets the latency and bandwidth requirements of the OCP Hyperscale CXL Tiered Memory expander spec by implementing a proprietary (De)compression algorithm at cache line granularity, with dual hardware accelerator implementation of the open source LZ4 algorithm to operate at page or block granularity for legacy compatibility.

2.2 Innovation

Unlike state-of-the-art solutions that conduct compression at block/page granularity, our approach operates at a more refined 64-byte granularity (as illustrated in Figure 2). This innovation seamlessly integrates into the CXL Type 3 device System-on-Chip (SoC), supporting both AXI4 and CHI specifications. The communication with the device occurs over CXL (2.0, 3.0, 3.1) command sets translated to AXI4, operating at a frequency of 1.2GHz (@4nm Samsung). Our solution exposes a compressed memory region as an additional NUMA tier in the memory hierarchy.

Achieving a 2-3x effective capacity increase with minimal impact on device bandwidth and nanosecond-level latency, our solution

IP Solution Performance Characteristics	Value
Compression Ratio	2-3X
Block cache (SRAM) hit latency	<30ns
Cache line in uncompressed region latency	<90ns
Cache line in an uncompressed block latency	<150ns
Cache line in a compressed block latency	<250ns
Tail latency [cache line in a compressed block]	<1us
Decompress bandwidth[4x 1867MT/s]	>46G/s

Figure 3: IP Solution Performance Characteristics. Exceed OCP Specification requirements

dynamically manages the compressed memory tier within the CXL Type 3 device. It implements real-time compression/decompression with compaction, operating at main memory speed and throughput. Additionally, an optional adaptive feature tunes performance based on diverse workloads using low-level telemetry data.

When the host processor demotes pages from the directly-connected DRAM to the CXL device, the CXL (device) controller converts CXL.mem data to AXI/CHI commands and calls our IP block in real time. The IP block intercepts data at cache line granularity (with a proprietary (de)compression algorithm) or at page/block granularity, as instructed over CXL.io. The compression algorithm is then applied, and memory allocation for the compressed CXL tier is managed. Compressed data is stored in the DRAM memory media over the DDR interface, with the IP block situated between the CXL controller interface and the DDR controller interface, communicating via AXI4 or CHI.

Our solution not only performs compression but also conducts compaction, ensuring effective space utilization. The IP block implements capacity reporting telemetry to the host. When a compressed page is requested, the IP accesses the compressed cache lines or pages/blocks, decompresses the data, performs necessary address translation on-the-fly, and transmits the data over the AXI interconnect from DRAM media to the CXL interface. This entire process is orchestrated by lightweight firmware running on an embedded processor in the CXL Type 3 expander SoC, enabling real-time compression/decompression at CXL line speed.

2.3 Performance Summary

The IP Solution described above delivers **2-3X Compression Ratios** while compressing memory at **Cache Line (64 Byte) granularity** across a breadth of representative Data Center workloads including **SPEC2017INT/FP, Renaissance, MLPerf/ Training, MonetDB+TPC-H** [Figure4]. The solution has been verified to operate at 1.2GHz and fits in an area of approximately **0.9mm² (at 4nm Samsung) where 75% of the IP solution area is occupied by SRAM**. This solution currently **supports (LP)DDR4, (LP)DDR5** memory technologies and **decompresses data in single digit clock cycle latency**. The overall CXL IP solution performance characteristics [Figure 3] satisfy the requirements in the OCP Spec [Chauhan et al. 2023]. The key IP Solution performance characteristics are in Figure 3. The IP solution is slated to go into production mid 2024 into CXL controller products and is expected to provide savings of 20-25% TCO [Figure 6] using previously published TCO model assumptions [Apostol 2022], factoring in 2X

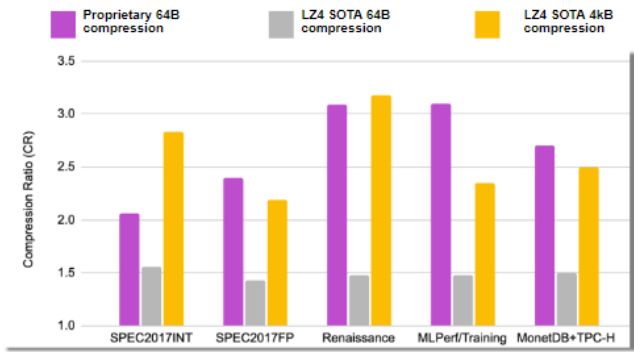


Figure 4: Geomean Compression Ratio across applications of each dataset

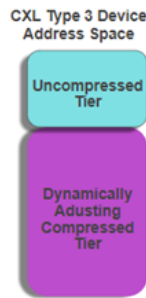


Figure 5: Compressed Memory Tier introduced and managed by the IP Solution within the CXL Memory Expander Type 3 device controller

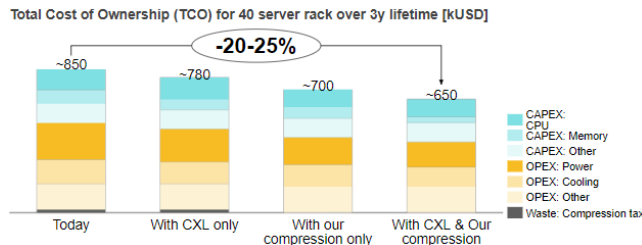


Figure 6: Reduced TCO. Addition of Compressed Memory Tier increases effective capacity (GB) leading to reduced \$/GB

capacity expansion through an additional Compressed Memory Tier.

3 IMPLEMENTATION / EVALUATION

3.1 Proof of Concept Demonstration

To validate the practical implementation and effectiveness of the proposed CXL solution, a Proof of Concept (PoC) demonstration has been developed [Figure 7]. The demonstration comprises two integral components: the frontend, based on QEMU, emulating the host system and managing the migration of identified cold pages to the CXL subsystem, and the backend, featuring an inline

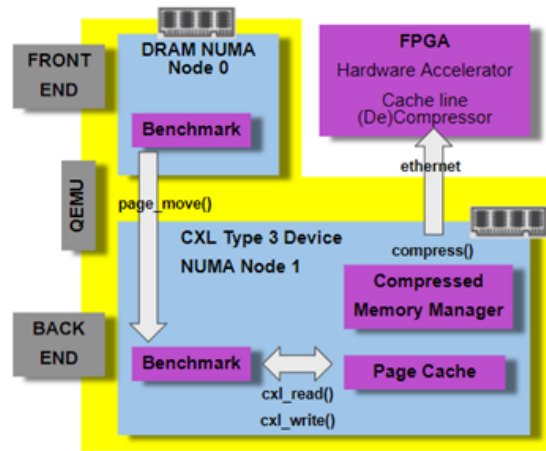


Figure 7: Proof Of Concept Implementation Demonstration

memory compression/decompression and management accelerator implemented on an FPGA, along with its accompanying firmware running on the FPGA-based platform.

3.2 Frontend: QEMU-based Emulation

The QEMU frontend emulates a Linux host capable of running realistic workloads. Currently, arbitrary pages of the workload are migrated into the emulated CXL memory. This can be extended with several mechanisms for detecting cold pages (e.g. LRU or MGLRU algorithms, Senpai, DAMON).

3.3 Backend: FPGA-based Accelerator and Firmware

The backend is centered around the FPGA, hosting an inline memory compression/decompression and management accelerator. This FPGA-based solution, coupled with its firmware, is responsible for compressing, storing, and managing the data migrated from the host system. The FPGA acts as the compressed memory subsystem of the CXL device, encapsulating both hardware and firmware components.

3.4 Demonstration Workflow

- (1) **Migration of Pages:** The QEMU frontend migrates pages to the CXL subsystem. This process involves invoking the IP solution on the FPGA, which takes on the responsibilities of compression, storage, and bookkeeping of the data.
- (2) **Requesting Compressed Pages:** The QEMU frontend, representing the host system, requests the migration of compressed pages by sending corresponding read requests to the IP solution on the FPGA. This step simulates the real-world scenario where the host system interacts with the CXL device to access compressed data.
- (3) **Real Application Utilization:** The demonstration includes a real application running on the system, showcasing the practical advantages of an expanded memory on the FPGA. This expanded memory acts as the compressed memory



Figure 8: FPGA Proof of Concept. Dynamically adjusting CXL Compressed Memory Tier

subsystem of the CXL device, incorporating both hardware and firmware functionalities.

- (4) **Live Telemetry Display:** Throughout the demonstration, live telemetry data is presented, including compression ratios, expansion factors, and other metrics [Figure 8] aligned with the OCP requirements for Hyperscale CXL Tiered Memory Expander. This real-time feedback provides insight into the system’s performance and adherence to industry standards.

The PoC demonstration serves as a tangible representation of the proposed CXL solution’s capabilities. By combining emulation with FPGA-based hardware acceleration, the demonstration provides a holistic view of the solution’s functionality, validating its potential for real-world applications in memory-intensive scenarios.

3.5 Summary

CXL addresses the pressing memory capacity needs of emerging Hyperscale applications. Hyperscalers [Chauhan et al. 2023] have formulated the groundbreaking (OCP) Hyperscale CXL Tiered Memory Expander specification to lower TCO and enable diverse CXL deployments. This paper presents a CXL integrated solution, aligning with the aforementioned specification, introducing an energy-efficient, scalable, hardware-accelerated, Lossless Compressed Memory CXL Tier offering 2-3X CXL memory compression in nanoseconds and delivering a 20-25% reduction in TCO for end customers without requiring additional physical slots.

4 OPEN QUESTIONS / CALL FOR COMMUNITY COLLABORATION

- (1) **Upstream Linux Driver Development:** Collaboration within the CXL community extends to upstream Linux driver development. A lightweight Linux driver is under development and is set to be upstreamed to the Linux kernel. This driver provides APIs that allow the host to utilize the oversubscribed compressed memory region of the device, and is intended to be fully compliant with the CXL upstream Linux driver. Collaboration in this area will ensure seamless integration with the Linux ecosystem, providing a standardized interface for CXL memory expansion.
- (2) **Integration and Testing/Benchmarking:** To ensure the robustness and performance of the proposed CXL solution, collaborative efforts could be directed towards integration and testing/benchmarking with data center applications, operating systems, and hypervisors. A comprehensive testing framework will be established to validate the solution’s

compatibility and efficiency across diverse environments, addressing the unique requirements of different workloads.

- (3) **Collaboration with Hyperscalers and Device Manufacturers:** Critical to the success of CXL adoption is collaboration with hyperscalers and device manufacturers. Aligning on requirements, gathering feedback, and understanding the practical challenges faced by these stakeholders are paramount to ensure smooth production deployment. Collaborative efforts involve iterative discussions, prototype testing, and refining solutions to ensure they meet the specific needs of hyperscalers and device manufacturers.
- (4) **Addressing adoption challenges raised by Hyperscalers:** Hyperscaler End customers have recently argued against the value of implementing CXL memory pools [Levis et al. 2023], their main arguments being: The cost of a CXL pool will outweigh any savings from reducing RAM. CXL has substantially higher latency than main memory, enough so that using it will require substantial rewriting of network applications in complex ways. To reduce Total Cost of Ownership (TCO) and enhance CXL adoption, the community can convert this challenge into an opportunity to path find better together solutions with compression, optical interconnects, native CXL controllers, and integrating flash media into the CXL memory tier to create holistic solutions that address diverse aspects of CXL infrastructure to maximize efficiency and performance at hyper scale. The open-source community in particular has the opportunity to take leadership, rally around to develop solutions that require no fundamental programming changes, ensuring a smooth transition for existing software stacks.

REFERENCES

George Apostol. 2022. *Using Pools of Shared Resources to Lower Latency and Improve System Performance*. <https://drive.google.com/file/d/1cZGC64WFY491-Jrf7jAHY64xR-YyDIOD/view>

Daniel S. Berger, Daniel Ernst, Huaicheng Li, Pantea Zardoshti, Monish Shah, Samir Rajadnya, Scott Lee, Lisa Hsu, Ishwar Agarwal, Mark D. Hill, and Ricardo Bianchini. 2023. Design Tradeoffs in CXL-Based Memory Pools for Public Cloud Platforms. *IEEE Micro* 43, 2 (mar 2023), 30–38. <https://doi.org/10.1109/MM.2023.3241586>

Meta: Prakash Chauhan, Chris Petersen, Google: Brian Morris, and Jerome Glisse. 2023. *OCP Hyperscale CXL Tiered Memory Expander Specification, Revision 1 Version 1.0 Base Specification, Template v1.2, Effective October 27, 2023*. <https://www.opencompute.org/documents/hyperscale-cxl-tiered-memory-expander-for-ocp-base-specification-1-pdf>

Geonhwa Jeong, Bikash Sharma, Nick Terrell, Abhishek Dhanotia, Zhiwei Zhao, Niket Agarwal, Arun Kejarawal, and Tushar Krishna. 2023. Characterization of Data Compression in Datacenters. In *2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. 1–12. <https://doi.org/10.1109/ISPASS57527.2023.00010>

Sagar Karandikar, Aniruddha N. Udipi, Junsun Choi, Joonho Whangbo, Jerry Zhao, Svilen Kanev, Edwin Lim, Jyrki Alakuijala, Vrishab Madduri, Yakun Sophia Shao, Borivoje Nikolic, Krste Asanovic, and Parthasarathy Ranganathan. 2023. CDPU: Co-designing Compression and Decompression Processing Units for Hyperscale Systems. In *Proceedings of the 50th Annual International Symposium on Computer Architecture (Orlando, FL, USA) (ISCA '23)*. Association for Computing Machinery, New York, NY, USA, Article 39, 17 pages. <https://doi.org/10.1145/3579371.3589074>

Philip Levis, Kun Lin, and Amy Tai. 2023. A Case Against CXL Memory Pooling. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks (HotNets '23)*. Association for Computing Machinery, New York, NY, USA, 18–24. <https://doi.org/10.1145/3626111.3628195>

Brian Will. 2023. *Intel® QuickAssist Technology Zstandard Plugin, an External Sequence Producer for Zstandard*. <https://community.intel.com/t5/Blogs/Tech-Innovation/Artificial-Intelligence-AI/Intel-QuickAssist-Technology-Zstandard-Plugin-an-External/post/1509818>