Optimized for Data Centers: Deployment-ready CXL® Memory Expansion with 5th Gen AMD EPYC™

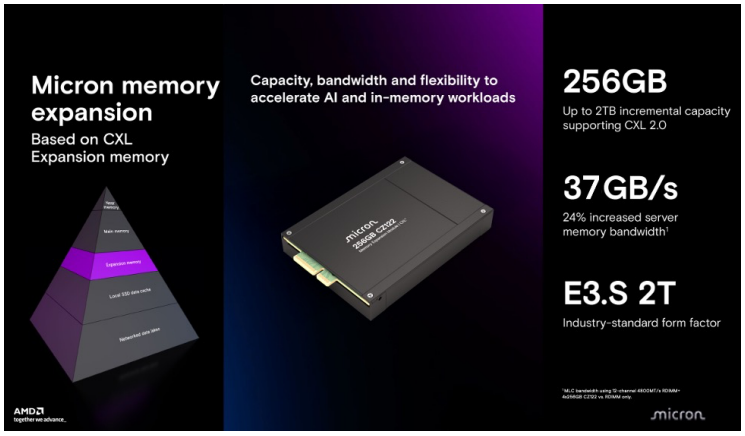
Venkata Ravi Shankar Jonnalagadda
Eishan Mirakhur
Srinivasulu Thanneeru
Vinicius Petrucci
Vishal Tanna

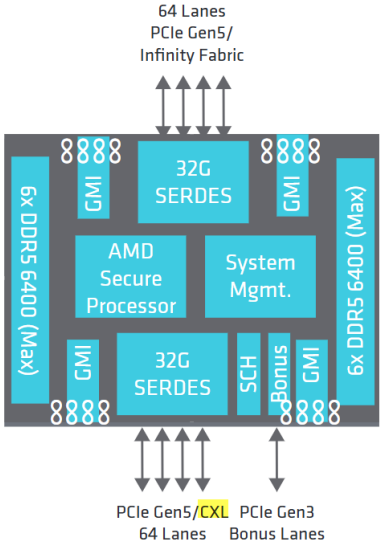
Hasan Al Maruf
Rita Gupta

Boosting Server Efficiency by Leveraging CXL® Memory for High-Core Systems

Critical data-intensive cloud workloads often face limitations in fully utilizing the computing resources available due to restricted memory capacity per core in high-core systems. This can result in reduced system efficiency, where customers invest in high-core systems but cannot effectively maximize their usage. For example, data analytics and deep learning models often require about 16 GB and 32 GB of memory per core[1], respectively. However, high-capacity RDIMMs (128 GB) on high-end servers like AMD Turin (128 cores & 12 memory channels) can only provide 12 GB per core.

By adding CXL® memory, customers can scale their virtual machines (VMs) more effectively using high core count CPUs, enhancing system infrastructure without the complexity of scaling out. Additionally, CXL® can offer memory bandwidth improvements, benefiting workloads that need extra bandwidth performance. The figures below illustrate the key characteristics of memory expansion using Micron CXL® memory modules and AMD's common I/O die of 'Zen 5/Turin' CPU enabling PCIe Gen5 64 lanes connectivity for CXL memory expansion.





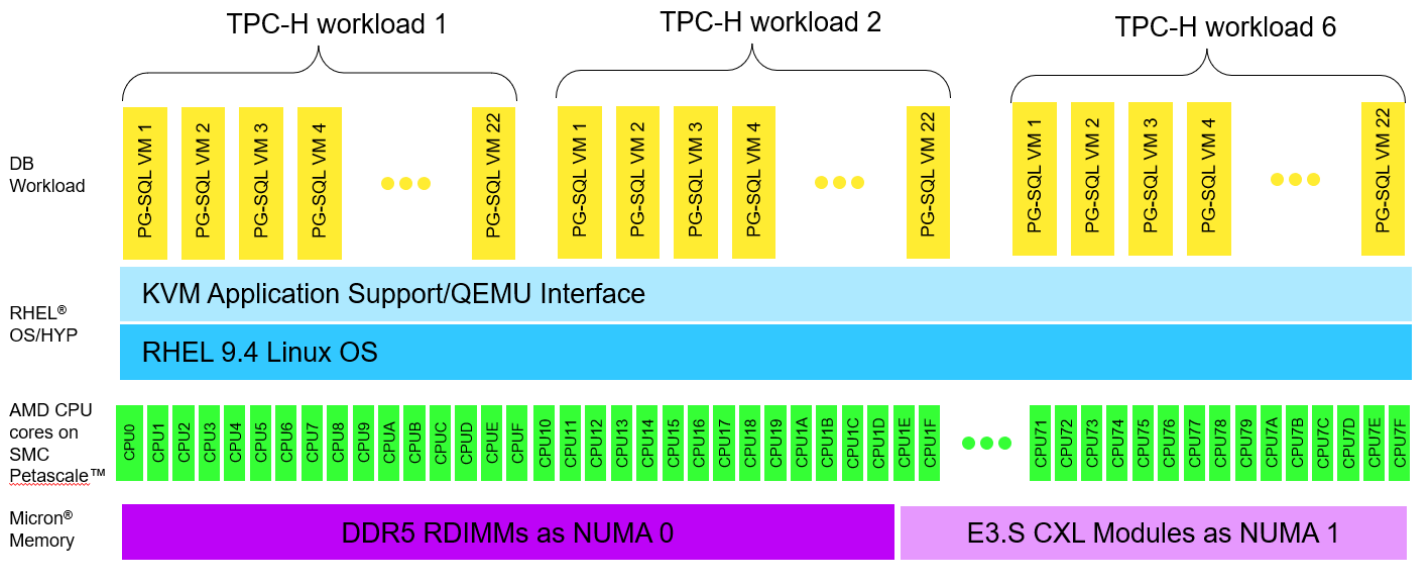
In general, when scaling workloads, adding more machines to the system can increase complexity and overhead, potentially leading to lower CPU utilization if the required GB per core is not met. CXL® allows for scaling up first, enhancing existing system infrastructure with fewer machines, and contributing to a lower total cost of ownership (TCO).

Leveraging CXL® for Scaling Data Analytics (TPC-H) in Virtualized Systems

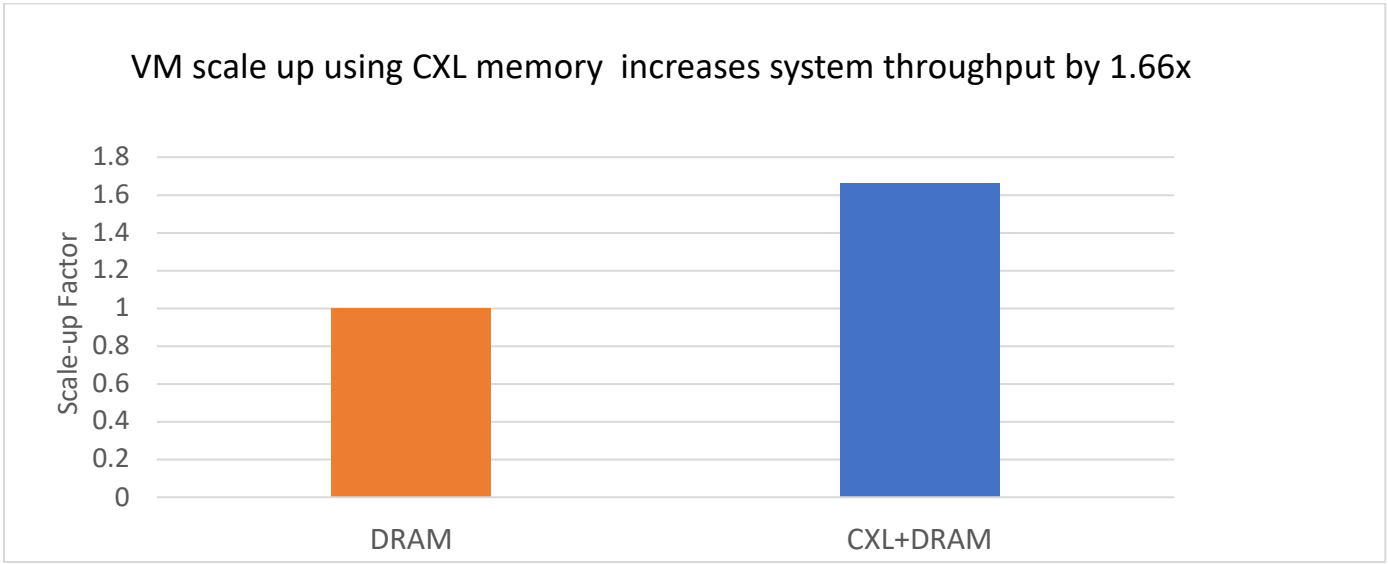
By expanding memory capacity (and bandwidth) with CXL®, customers can maximize the performance of high-core systems, ensuring efficient workload management without extensive scaling out. This approach simplifies infrastructure management and enhances the overall efficiency and economy of server utilization.

Multi-tenant workloads were deployed on **AMD 5th Gen EPYC™ ("Turin")** systems equipped with Micron's DDR5 and **CXL® memory modules**. The primary objective was to enhance CPU system utilization by providing additional CXL® memory capacity. Additional CXL® memory modules were introduced to enhance the server's memory capacity, thereby enabling it to handle more virtual machines (VMs) and demanding workloads effectively. This approach addresses the limitation of memory per core in high-core systems by providing supplementary memory capacity, which is crucial for data-intensive applications like TPC-H. The mapping of the TPC-H workload virtual machines to the system is detailed below.

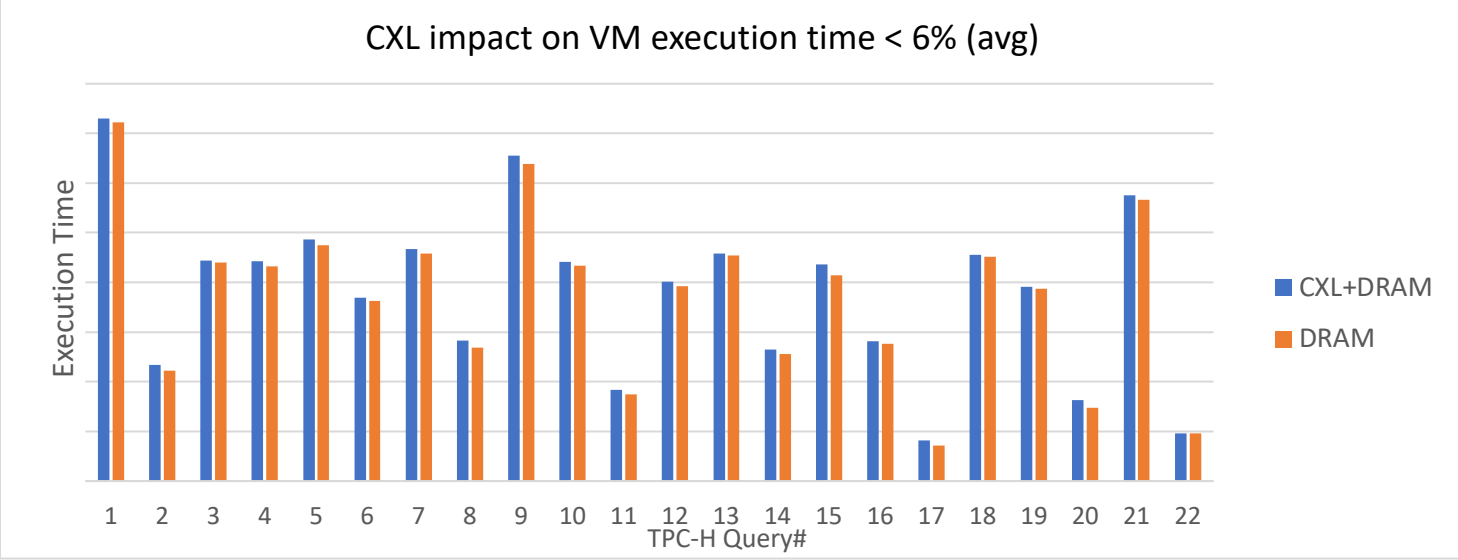
[1] Managing Memory Tiers with CXL in Virtualized Environments, OSDI '24



As shown in the plot below, expanding the memory with CXL® significantly boosts overall server performance, increasing throughput by 66%. This improvement is achieved by augmenting the server's capacity with an additional 1TB of memory using four 256GB CZ122 modules, complementing the existing 1.5TB main memory provided by 12x 128GB DDR5 RDIMMs @ 6400 MT/s. This substantial memory enhancement ensures that workloads requiring high memory per core are efficiently managed, optimizing CPU core utilization and enhancing overall server performance.



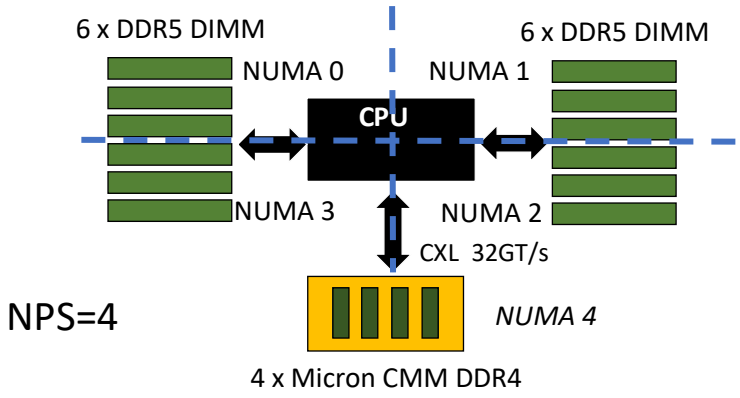
It is worth noting that individual virtual machines (VMs) running with CXL memory experienced less than a 6% average performance delta, demonstrating the minimal impact on individual VM performance while enabling higher overall memory capacity. This improved memory infrastructure allows each VM to access the memory resources it needs without compromising on user-facing performance, thereby maximizing the benefits of high-core systems. This means CXL® memory can be a viable and cost-effective strategy for maximizing the compute potential of high-core systems, enhancing system efficiency and economy, and ultimately benefiting customers through improved performance and lower costs.



CXL® Memory Expansion for AI Systems: LLM Use Case Analysis

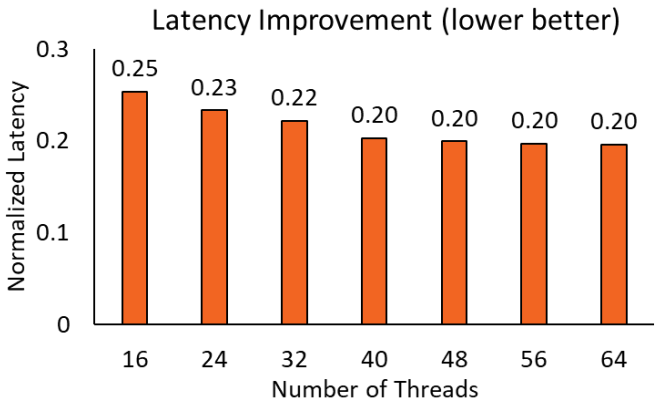
In high-core server systems, CXL® memory can improve Large Language Model (LLM) performance by increasing *both* memory capacity and bandwidth beyond DDR memory configurations. The AMD 5th Gen EPYC™ ("Turin") platform with 12x 64GB DDR5 and 4x 256GB DDR5 Micron CXL® memory was tested using an LLM inference workload (llama.cpp) to assess the workload performance. The model used was the "llama-2-70b.Q4_K_M" in GGUF format, running it across all physical cores on a single socket. The token input/output count was 512 and the batch size was 2048. To extract balanced memory bandwidth across DDR and CXL® memory tiers, NUMA-based interleaving was used. The system configuration is depicted in the figure below:

System Configuration

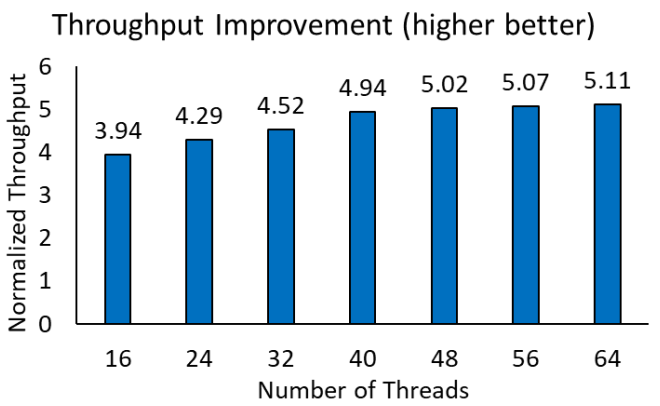


NUMA interleaving: 80% DRAM, 20% CXL

The system was configured with AMD's NPS set to 4, resulting in a 4:1 interleaving (80% of the working set on local memory and 20% interleaved between local and CXL memory). As shown in the experimental results from the plots below, the latency of LLM reduces by 80% while the throughput improves by 5x over all running on the local CPU-attached memory configuration.



80% speedup in latency over baseline



5x speedup in throughput over baseline

RAS Features and Tooling for CXL® Memory

Micron's CXL® memory modules support advanced RAS (Reliability, Availability, and Serviceability) features to ensure system reliability, maintain high availability, and facilitate serviceability. The memory errors have been reported using Linux kernel 6.14.0-rc1 after applying a patch series from AMD-CXL joint collaborative work[2]. The verification of CXL protocol error reporting, along with error injection to components or CXL-attached memory, has been conducted. Key snippets of major error injections and reporting on CXL.io, CXL.mem, and CXL-attached memory endpoints are provided in the experiments. The errors were observed in Linux trace events and the rasdaemon tool as shown below table.

S.no	Injected Error Type	Test Report
1	CXL.io EINJ-Based Correctable Tx Error	**Pass**
2	CXL.io EINJ-Based Correctable Rx Error	**Pass**
3	CXL.io EINJ-Based Uncorrectable Fatal Tx Error	*Not Supported/TBD*
4	CXL.io EINJ-Based Uncorrectable Fatal Rx Error	**Pass**
5	CXL.io EINJ-Based Uncorrectable Nonfatal Tx Error	**Pass**
6	CXL.io EINJ-Based Uncorrectable Nonfatal Rx Error	**Pass**

AMD RAS Tool with EINJ-Based Error Injection and Reporting.

The table describes the major error injections and reporting on CXL.io and CXL.mem, conducted using the AMD RAS tool with the Linux-based EINJ (Error Injection Interface)[3], and error injection to CXL-attached memory endpoints using Micron's MXCLI through Vendor Specific commands.

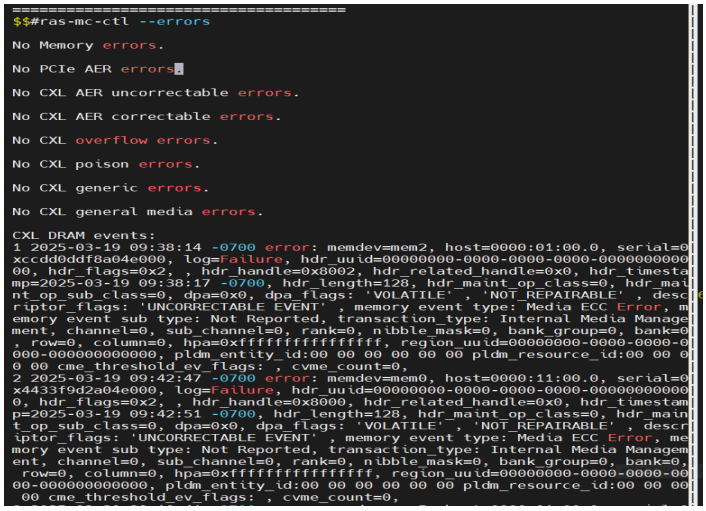
[2] https://patchwork.kernel.org/project/cxl/list/?series=947614
[3] https://www.kernel.org/doc/Documentation/acpi/apei/einj.txt

Uncorrectable Error Injection and Reporting with CXL® attached memory endpoint.

Errors were injected via VS commands into the endpoint using Micron's MXCLI tool[4]. The MXCLI tool is a Micron developed management and support tool for sending various mailbox commands to a CXL device. It can be used to retrieve identity, monitor health information, read logs or issue resets. Standard and Extended Capability registers can be retrieved and parsed in a user-friendly manner. It supports both command line arguments and an interactive mode with auto complete commands/fields. This tool was used extensively to demonstrate RAS capabilities of CXL device and platform through vendor specific commands.

Interface for Error Injection using MXCLI tool

Error Reporting in Linux Kernel traces and Rasdaemon



It is worth mentioned that Micron and AMD have developed, debugged, and introduced new RAS and error injection features for seamless performance using the AMD Turin platform and Micron CXL® CZ122 memory modules.

Conclusion

This white paper demonstrated that integrating CXL with the current memory infrastructure allows the system to meet higher memory requirements per core, thus optimizing the compute potential of high-core systems. The experimental results showed enhancements in workload performance metrics, particularly in data analytics and LLM systems, which typically experience limitations due to memory constraints in high-core environments.

The key findings of the integration of CXL® memory into high-core systems reveal significant performance enhancements across various workloads. Specifically, both TPC-H and LLM inference workloads demonstrate substantial improvements in throughput, latency, and overall system efficiency.

The expanded memory capacity and bandwidth provided by CXL® memory enable high-core systems to manage demanding data-intensive applications more effectively, optimizing CPU core utilization and minimizing performance impact on individual virtual machines. This can facilitate a more streamlined and economical management of server infrastructure, ultimately lowering the total cost of ownership for customers and benefiting them through improved performance and lower operational costs.

[4] https://github.com/cxl-micron-reskit/mxcli