



CXL Performance Monitoring Status and Outlook

Linux Plumbers Conference 2026

5-7 October, Prague

Device and Specific Purpose Memory MC

Robert Richter <rrichter@amd.com>

CXL Performance Monitoring - Status and Outlook

- CXL memory architecture and challenges
- CPMU kernel support and discovery
- CXL event counting and event groups
- Event coverage: host-side to CPMU equivalents
- Discussion

CXL Architecture

- CXL is a high-performance I/O bus architecture to interconnect peripheral devices:
 - traditional non-coherent I/O devices
 - memory devices
 - accelerators with additional capabilities
- CXL protocols:
 - CXL.io (discovery, config, enumeration, error handling)
 - CXL.cache (device coherent cache)
 - CXL.mem (device memory access)
- CXL device types for:
 - coherent cache (Type 1)
 - accelerator with device memory (Type 2)
 - device memory, managed by the host (Type 3)

CXL Memory Devices and System Memory

- CXL memory devices (Type 3) are exposed as system memory through the CXL subsystem
- CXL device memory can be mapped into system coherent address space
- Once online, CXL-attached DRAM is managed by the kernel as regular memory nodes
- Used for general-purpose allocation, NUMA-aware placement, or tiered memory configurations

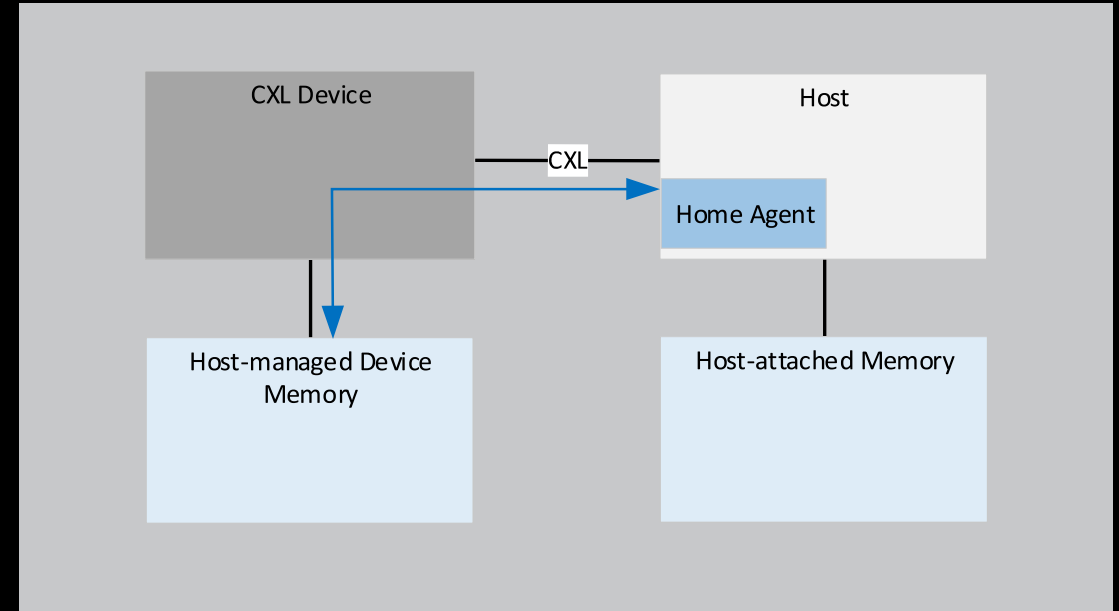


Figure 1: CXL Type 3 device (CXL Specification Rev 4.0)

Source: CXL Consortium, "Compute Express Link (CXL) Specification Revision 4.0, Version 1.0," August 2025, Figure 2-6

Challenges When Using CXL Memory as System Memory

- The host's memory controller is not used to access CXL memory
- Error handling differs from local DRAM:
 - CXL devices report errors through CXL protocol events and DVSEC error registers, not through standard EDAC/MCA paths
 - CXL error handling support being continuously added (esp. v6.7, v7.0)
 - currently: [PATCH v20 0/9] Enable CXL PCIe Port Protocol Error handling and logging (Terry)
- Performance monitoring differs:
 - no host-side PMU counters for CXL memory
 - host CPU only sees cache fills from "extension memory" but not DDR-level detail behind the CXL link

CXL Performance Monitoring

- Component Performance Monitoring Unit (CPMU) introduced to allow device-side performance monitoring
- Performance monitoring is part of the CXL Spec:
 - CPMU exposed through the PCIe DVSEC Register Locator capability
 - provides standardized event counting for CXL devices
- CXL Spec 4.0 references:
 - 8.2.7 CPMU Register Interface
 - 13.2 Performance Monitoring
- CPMU Discovery Scope
 - Spec does not limit CPMU to CXL endpoint devices
 - Any PCI device may implement Register Locator DVSEC and provide CPMU registers, including:
 - CXL root ports
 - Switch upstream/downstream ports
 - CXL memory devices (DDR interface events)
 - Host bridges do not implement CPMU registers

CPMU Kernel Support

- cxl_pmu: kernel driver, CONFIG_CXL_PMU, drivers/perf/cxl_pmu.c
 - Merged in v6.5 (Jonathan Cameron)
 - CXL 4.0 event list added in v7.3 (Harshal Thakkar)
- CXL devices may provide one or more CPMU Register Blocks
- Driver discovers CPMU blocks using DVSEC Register Locator capability (ID 0008h, Register Block Identifier 04h)
- Driver binds to CXL endpoint devices via cxl_pci
- Creates pmu_mem* devices for each detected CPMU Register Block
- Only CXL endpoint devices are currently discovered, root ports and switch ports are not supported
- Raw event programming supported via perf config format:
 - config: mask (bits 0-31), gid (bits 32-47), vid (bits 48-63)
 - Named: perf stat -e cxl_pmu_mem0.0/clock_ticks/ -a sleep 1
 - Raw: perf stat -e cxl_pmu_mem0.0/config=0x1e98000000000001/ -a sleep 1
 - Events are validated against the device's Event Capability registers

CXL Event Counting

- Driver exposes CXL PMU events through the perf subsystem
- Event collection is poll-based using perf stat
 - Example: `perf stat -e cxl_pmu_mem0.0/DDR_act/ -a sleep 1`
- No sampling support:
 - System-wide only, no per-task context
 - IRQ handler for counter overflow exists, but only handles wraparound, not perf sample generation
 - `perf top -e cxl_pmu_mem0.0/DDR_act/` not available
 - Driver-only limitation, can be added

CPMU Event Groups (CXL 4.0, Table 13-5)

Available CPMU event groups and perf key events:

GID Range	Groups	Description	Key Events
0000h	Clock Ticks	Base clock counter	clock_ticks
0010h-0014h	CXL.cache (5)	D2H/H2D req/rsp, cache data	d2h_req, h2d_req
0020h-0025h	CXL.mem (6)	M2S/S2M protocol messages	m2s_req_meminv, s2m_bisnp
0030h-0031h	Device (2)	Device load, M2S residency	device_load, m2s_residency
8000h	DDR Interface	DDR commands, refresh, power down	ddr_act, ddr_casrd, ddr_caswr
8001h	Queue Occupancy	Queue depth	rd_queue_occ, wr_queue_occ
8002h	Queue Residency	Queue latency	mc_rd_resid_cnt, mc_wr_resid_cnt
8003h	Retry Events	Link retry counters	retry_event_trig_by_rd_crc
8004h	Throttle	Throttle events	thermal_throttle_event

PMUs in IP Blocks of Today's Processors

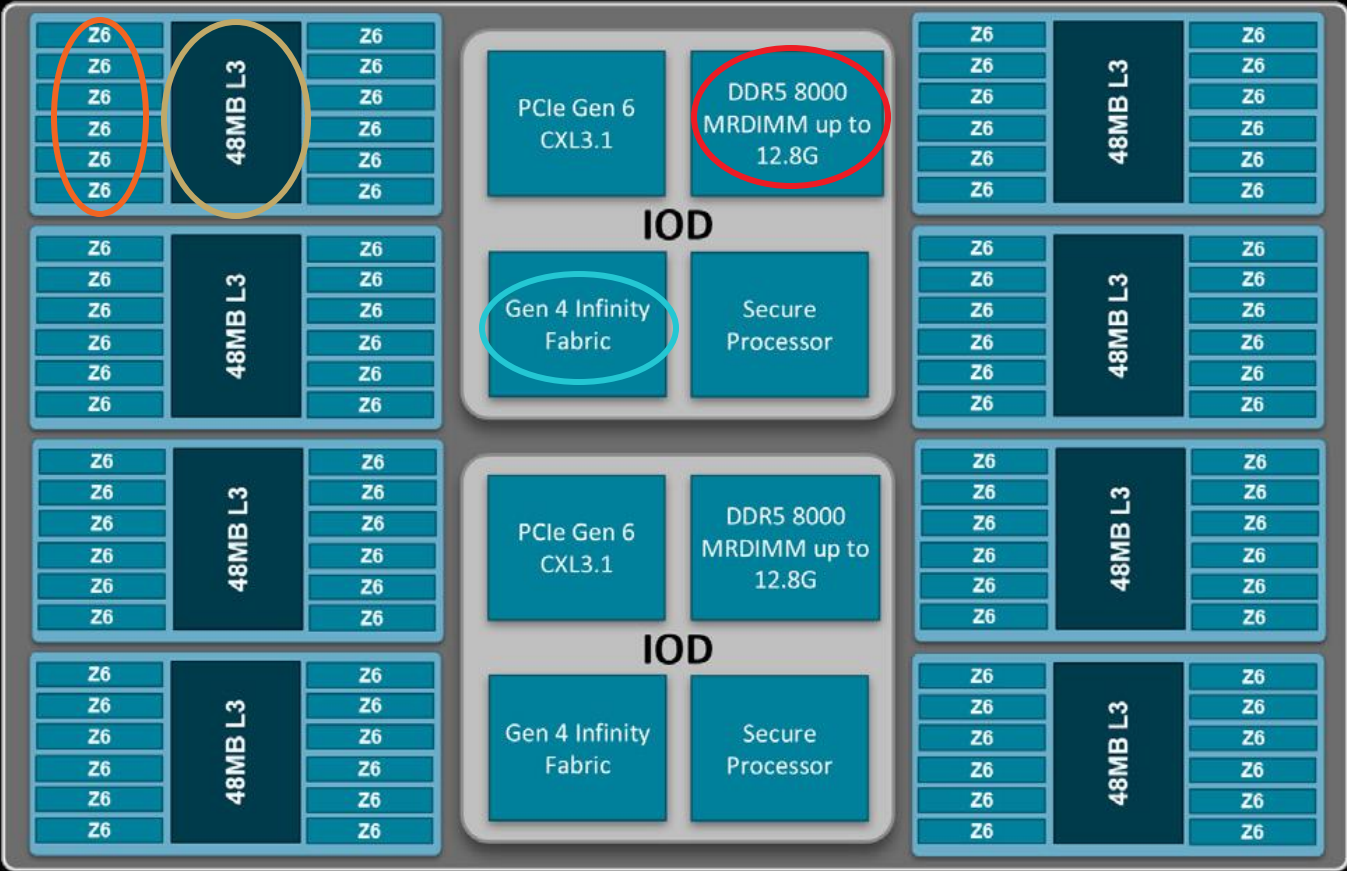


Figure 2: AMD EPYC™ 9006 processor, “Zen 6” chiplet (CCD + IOD)
 Source: AMD, "AMD EPYC™ Server CPU Architecture and Performance Overview," 2026, Figure 5, Page 7

IP Block	perf PMU	Description
Core	cpu	Instructions, branches, TLB, L1/L2 caches
L3 Cache	amd_l3	L3 hits/misses, sampled latency
Data Fabric	amd_df	Interconnect traffic, DRAM beats
UMC	amd_umc	DDR commands, bandwidth

- CXL memory is not connected to the UMC but to the I/O links, not visible to host-side PMUs
- DDR Interface (8000h) events are comparable to host-side UMC memory controller events (amd_umc)

Event Coverage - Named Perf Key Event Equivalents

Named amd_umc perf tool key events (.../amdzen6/memory-controller.json):

Host-Side (amd_umc)	CPMU Event (driver)	CPMU (spec)	CPMU Match
umc_mem_clk	clock_ticks	0000h/00h	approx
umc_act_cmd.all	ddr_act	8000h/00h	exact
umc_act_cmd.rd/.wr	-	-	none 1)
umc_pchg_cmd.all	ddr_pre	8000h/01h	exact
umc_pchg_cmd.rd/.wr	-	-	none 1)
umc_cas_cmd.rd	ddr_casrd	8000h/02h	exact
umc_cas_cmd.wr	ddr_caswr	8000h/03h	exact
umc_cas_cmd.all	ddr_casrd + ddr_caswr	8000h/02+03	exact
umc_data_slot_clks.all	-	-	none 2)
umc_data_slot_clks.rd/.wr	-	-	none 2)

none: for footnotes see Event Coverage - Conclusion

Note: Remaining CPMU events are not covered by publicly documented perfctr events.

Event Coverage - Perf Recommended UMC Metric Equivalents

Recommended perf tool event metrics (.../amdzen6/recommended.json):

Metric/Event	Expression	CPMU Match
umc_mem_clk	umc_mem_clk	approx
umc_mem_bandwidth	umc_cas_cmd.all * 64 / 1e6 / time	exact
umc_mem_read_bandwidth	umc_cas_cmd.rd * 64 / 1e6 / time	exact
umc_mem_write_bandwidth	umc_cas_cmd.wr * 64 / 1e6 / time	exact
umc_data_bus_utilization	umc_data_slot_clks.all / 2 / umc_mem_clk	none
umc_cas_cmd_rate	umc_cas_cmd.all * 1000 / umc_mem_clk	exact
umc_cas_cmd_read_ratio	umc_cas_cmd.rd / umc_cas_cmd.all	exact
umc_cas_cmd_write_ratio	umc_cas_cmd.wr / umc_cas_cmd.all	exact
umc_activate_cmd_rate	umc_act_cmd.all * 1000 / umc_mem_clk	exact
umc_precharge_cmd_rate	umc_pchg_cmd.all * 1000 / umc_mem_clk	exact

exact: 8 of 9 metrics are implementable with CPMU DDR events

approx: umc_mem_clk counts DDR MEMCLK cycles, CPMU clock_ticks counts the device base clock

none: umc_data_bus_utilization is DDR bus and UMC specific, no direct CPMU equivalent exists. CXL link occupancy from M2S/S2M protocol events should be used as a substitute

Event Coverage - Conclusion

- Good coverage of all standard memory controller and CPMU events
- Fewer CPMU events: More UMC::PMC events plus event masks vs. 17 CPMU events
- Non-matching events:
 1. Less granularity due to missing event mask, no perfctr or CPMU equivalent
 2. DDR bus utilization, UMC specific, use CXL link occupancy as substitute
 3. Missing event: No UMC equivalent for power events

Discussion Points

- Extend CPMU discovery to root ports and switch ports? Endpoint devices are the most important event source to cover DDR interface events
- Spec-defined events provide good coverage, any missing events?
- Sampling support: add perf record/perf top for CPMU?
- Unified memory performance view: CXL and DDR memory can be used interchangeably. Multiple PMUs make memory event tracing harder across different system memory blocks and even across multiple CXL devices. How to solve that? Should perf metric selection enable events on all corresponding PMUs? Multi-PMU metric groups? Auto-discovery of memory controller PMUs?
- More questions?

References

Resources:

- CXL Consortium, "Compute Express Link (CXL) Specification," Rev 4.0, Version 1.0, August 2025.
<https://www.computeexpresslink.org/spec-landing>
- AMD, "AMD EPYC™ Server CPU Architecture and Performance Overview," Whitepaper, September 2026.
<https://www.amd.com/.../white-papers/amd-AMD-EPYC-9006-server-cpus-architectural-leadership.pdf>
- AMD, "AMD EPYC™ 8005 Series Processor Architecture Overview," Pub 72128, Rev 1.0, May 2026.
- AMD, "Processor Programming Reference (PPR) for AMD AMD EPYC™ 9006 Processors (Family 1Ah Models 50h-57h, Revision B0)," Pub 58264.
- AMD, "Performance Monitor Counters for AMD Family 1Ah Models 50h-57h Processors," Pub 69163, Rev 1.00, December 2025.
- Linux kernel, tools/perf/pmu-events/arch/x86/amdzen6/
- Linux kernel, drivers/perf/cxl_pmu.c

LPC Talks:

- LPC 2026 (Prague): CXL Performance Monitoring - Status and Outlook
<https://lpc.events/event/20/contributions/2521/>
- LPC 2024 (Vienna): State of CXL Error Handling (with Terry Bowman)
<https://lpc.events/event/18/contributions/1769/>
- LPC 2022 (Dublin): Linux plumbing of CXL error reporting (with Yazen Ghannam)
<https://lpc.events/event/16/contributions/1340/>

Copyright and disclaimer

©2026 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD Arrow logo, AMD EPYC, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS.' AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION

Thank You

