CMS: Hotness Tracking Requirements

Version 1.0

Date: Oct 17, 2023

Author (s) :
Samir Rajadnya, Microsoft
Durgesh Srivastava, Nvidia

Reviewer (s):
Anjaneya Reddy Chagam, Intel Corporation
Manoj Wadekar, Meta

Executive Summary

This document defines the **Open Compute Project (OCP) Composable Memory Systems (CMS)** Hotness Tracking Requirements proposal. CMS is an emerging paradigm that facilitates dynamic and unified memory management across diverse memory technologies, interconnects, and hierarchies. Hotness tracking is a technique used to monitor hot memory regions on a far CXL memory to address the additional latency due to CXL memory. This document proposes a hardware technique to be implemented inside the CXL memory controller and to perform monitoring, providing information over a software interface to the guest application or operating system.

Table of Contents

1. Compliance with OCP Tenets

1.1 Openness

The OCP CMS Hotness Tracking requirements document was developed in dialogue with the entire OCP CMS community membership and the resulting document represents a set of features needed for implementing hotness tracking features by the device vendors. OCP CMS plans to work with CXL consortium to develop CXL Hotness Tracking ECN to enable standards based hotness tracking interfaces and capabilities..

1.2 Efficiency

The OCP CMS Hotness Tracking requirements document lays the foundation for developing hotness tracking capability by the device vendors using open interfaces. This helps in enabling interoperable, open hardware and software co-designed composable memory solutions.

1.3 Impact

Having a standard and consistent device interface for hotness tracking helps easier integration, enables interoperability and faster time to market.

1.4 Scale

CMS Hotness Tracking fosters consistent vendor neutral composable solutions and enables high degree of reusability by conforming to standard device interfaces..

1.5 Sustainability

Hotness Tracking capability is designed to optimize performance and resource utilization by intelligent data movement among physical memory tiers thereby achieving data center energy efficiency.

2. Introduction

The rise in the number of CPU cores in modern computing systems has led to an increase in the demand for system memory. To achieve the required capacity points, the Compute Express Link (CXL) technology allows for the mix and match of different memory technologies. Many use cases will benefit from this increased capacity and bandwidth options. Emerging technologies like memory pooling and memory sharing are set to increase the importance of CXL attached memory even further. With the ability to pool and share memory resources, CXL memory provides a flexible and efficient solution for modern computing needs. As these technologies continue to develop, we can expect to see even more benefits from CXL attached memory in the future.

CXL memory can be used in two ways. The first method involves using two memory regions, which require software modification. While this route may not be preferred by many software applications, some may be willing to rewrite for two types of memory regions. The second method involves using a single memory region as seen by software. When using the two-memory approach, certain techniques must be employed to address the performance drawbacks associated with higher latency memory, such as CXL.

A primary downside of CXL-based memory is the added latency. This additional latency affects application performance and introduces run-to-run variations depending on memory location in DRAM versus CXL memory. Hotness tracking is a mechanism that provides information to software for moving hot pages to local DRAM.

3. Hotness Tracking

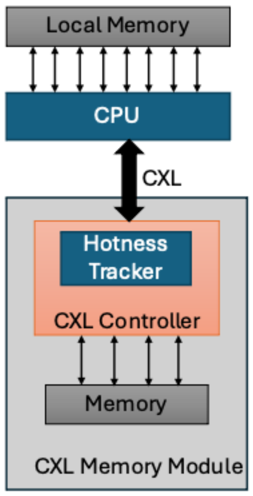
Hotness tracking is a technique used to monitor hot memory regions on a far CXL memory. This hardware technique is implemented inside the CXL memory controller and only performs monitoring, providing information over a software interface to the guest application or operating system. The OS or guest application is then responsible for page migration, where hot memory regions from far CXL memory are moved into near DDR memory. With this approach, hotness tracking enables efficient use of CXL memory resources, ensuring that the most frequently accessed data is stored in the fastest and most accessible memory regions.

4. Current Implementation Limitations

Let's examine the limitations of existing hotness tracking methods that use different counters inside the CPU to track memory usage. There are several ways to perform tracking, including using an access bit in the page table entry, processor event-based sampling (PEBs), and instruction-based sampling (IBS). However, these methods have certain limitations. For example, PEBS does not track certain types of main memory requests, such as writes, read-for-ownership, and prefetch regular read requests. This means that PEBS can only track 49% of main memory traffic, making it less effective for certain use cases.

5. Hotness Tracking in CXL Controllers – A New Proposal

Adding the hotness tracking feature inside the CXL memory controller provides flexibility across different CPU vendors and families. CXL memory controller vendors can choose their own implementation method, allowing for flexibility around CXL 2.0 and 3.0 implementations. Cold tracking, on the other hand, is best done on DDR and is more easily tracked in CPUs. While it is easier to track bandwidth in CPUs, it is more difficult to track hotness using instruction-based sampling (IBS) or address tracking. Additionally, CPU cycles are consumed when running the tracking algorithm, further highlighting the importance of implementing hotness tracking on the memory controller.



Hotness tracking is a multi-step process that enables efficient use of CXL memory resources. In the first step, the system monitors segments at a larger configurable granularity, counting accesses into each segment to identify the hottest segment. Once the hottest segment is identified, the system moves to step two, which involves identifying the hottest pages within that segment at a smaller variable granularity. The top N pages are then selected using a method (For example: least frequently used (LFU)). This step results in candidates for movement from pooled memory or expansion memory to near memory. In step three, software such as application software or the operating system moves the selected pages. Finally, in step four, once a page is swapped to near memory from far memory, the corresponding entry or entries are removed from the hotness buffer. With its ability to identify the most frequently accessed data and move it to the fastest and most accessible memory regions, hotness tracking is an important technique for optimizing memory usage and improving overall system performance.

We would like to propose following high-level requirements for a hotness tracking widget:

- This widget is for hotness tracking only. For cold tracking other telemetry methods will be used.
- Widget counts CXL memory access from the host CPU only. It does not count for any other access such as CXL.IO or P2P accesses.
- Widget counts all CXL memory accesses CXL.mem only.
- The widget will operate at CXL port level.
    - In the case of multi-ported devices there will be a widget per CXL port.

- In case of interleaved CXL ports from a single host there will be a single widget for this host.
- In the case of an MLD device there will be a hotness tracker per LD.

- The method of hotness tracking is implementation specific. Each CXL controller vendor is free to choose any implementation method.
- Widget should collect telemetry data at customizable granularities that are independent of CPU page size mapping.
- Once configured, the widget should operate independently to collect telemetry data that can be queried by software periodically.
- Widget expects that the CPU can query periodically within the range of 1ms to 10sec.
- Telemetry data should include a list of ranked hot pages.
- The following parameters should be configurable.
    - Block size
    - Read, write or both.
    - Counting interval
    - Hot threshold (May be thresholds)
    - Sampling rate
    - Number of hot pages for a given threshold
    - Hotness definition (based on recency or frequency)
    - SW should be able to tell widget to zero-in on hotness tracking for some set of address regions.
    - If configured, the widget should be able to sort hot pages in priority (Hottest on top of list)
    - If configured, the widget should be able to classify these hot pages in three buckets such as high, medium, and low.
- Widget should work within confidential computing environment (Maintain trust domain)
- Widget should report hot page information based on Device Physical Address (DPA) (address available to CXL memory controller on CXL port)
- This widget is limited to CXL endpoints only (Not for CXL switches or RP)
- Vendor to vendor interop is important and should be considered as part of interop testing.

To effectively communicate hotness tracking information, a standardized software interface is necessary. The Host is responsible for determining if this capability exists, while the Hypervisor/OS pulls data opportunistically and/or periodically based on the size of data and frequency of access. The Device implements a Door-Bell mechanism using CCI, with a defined structure, size, format, and software interface. To enable vendor-specific extended capability (DVSEC), a custom DVSEC register is created in the CXL device to indicate the presence of counters and provide information about their location and access methods. The Device driver communicates with the SoC device and accesses the hotness tracker, parsing the DVSEC, identifying the counter's location and access method, and exposing an interface to the operating system for accessing the counters. Memory mapping is implemented to allow the SoC to access the CXL device's memory for hotness tracking. The device driver exposes the hotness tracker through sysfs entries or a custom IOCTL interface and handles read requests from the SoC, accessing the appropriate memory location, retrieving the counter data, and returning it.

6. Conclusion

We believe that hotness tracking is a crucial enabling technology for the widespread adoption of CXL memory. To promote the development and adoption of this interface technology, we are calling on the OCP community to help define and adopt it. By separating interface definition from implementation, we can ensure that everyone in the ecosystem benefits from this technology.

7. License

8. About Open Compute Foundation

At the core of the Open Compute Project (OCP) is its Community of hyperscale data center operators, joined by telecom and colocation providers and enterprise IT users, working with vendors to develop open innovations that, when embedded in products, are deployed from the cloud to the edge. The OCP Foundation is responsible for fostering and serving the OCP Community to meet the market and shape the future, taking hyperscale led innovations to everyone. Meeting the market is accomplished through open designs and best practices, and with data center facility and IT equipment embedding OCP Community-developed innovations for efficiency, at-scale operations and sustainability. Shaping the future includes investing in strategic initiatives that prepare the IT ecosystem for major changes, such as AI & ML, optics, advanced cooling techniques, and composable silicon. Learn more at www.opencompute.org.