Memory Page Prefetching for Disaggregated Memory Systems

Graduate School of Information Sciences Kobayashi・Sato Lab LI ZECHENG C2IM1503

1 Introduction

As the computational capability of data centers continues to increase, an ever-growing number of computing tasks are allocated to data centers. Simultaneously, there a significant disparity in the granularity of computational workloads, which has resulted in memory resource fragmentation within data centers. In order to better utilize the memory fragments in these nodes, people are attempting to break down the barriers of computer physical architecture by pooling memory fragments and accessing them from other remote nodes to achieve higher resource utilization, this technique is referred as memory disaggregation.

Currently, mainstream approaches to memory disaggregation rely on the virtual memory system, which is commonly referred to as a kernel-based system. In this system, remote memory pages are accessed by triggering page faults through the operating system, which actually creates a critical path since it results in high overhead and a significant drop in application performance.

To solve this problem, recent advancements focused on addressing remote memory access issues through bus-extended systems, such as CXL-based systems[4], which enables access to remote memory at the bus level and are considered a promising solution. The overall advantage of the bus-extended systems over the kernel-based systems is the reduction in page-fault occurrences. However, due to limited information about memory granularity, the bus-extended systems are relatively not ideal in terms of page prefetching compared to the kernel-based systems.

Therefore, this work aims to enhance the efficiency of the bus-extended systems by predicting future memory page needs using cache-line information. By using cache-line access sequence information for a series of machine learning training, this research uncover the relationship between cache-line information and page prefetching, identify a suitable model, ultimately improving the hit rate of remote prefetching pages for disaggregated memory systems.

2 Background

Table 1 includes a comparison of two main related studies of memory disaggregation. it shows a comparison of two main proposals, focusing on aspects such as granularity, programmability, and underlying dependency mechanisms.

Table 1: Taxonomy of disaggregated memory system.

Disaggregate Memory	Granularity	Programability	Mechanism
Object-based	Yes	No	User Apps
Page-based	No	Yes	Virtual Memory

The *object-based* systems provides a fine-grain access to remote memory. However, these systems require to use specific APIs to send information within nodes, which significantly reduces code portability. The *Page-based* disaggregated memory systems provide transparent access to disaggregated memory by using virtual memory to cache remote pages into a local software-managed DRAM cache, sacrificing access granularity for application performance. These systems rely on custom page fault handlers to fetch data from remote hosts in response to page faults. In the context of the page-based systems, efforts have been made to address the latency introduced by page faults through more precise prefetch algorithms. However, a paradox arises: to train efficient prefetching models, a substantial number of page faults is required as training data is required, even though our ultimate goal is to minimize page faults. HoPP [1] have proposed a solution that decouples memory accessing data flow from page transportation, recording page access separately for training. While this method effectively improves prefetching accuracy, it introduces significant software and hardware overhead.

With the ongoing development of Compute Express Link (CXL), there is a reevaluation of the reliance on virtual page-based remote memory access. Kona [2] have introduced an approach to process memory access flows at a finer granularity such as cache line. This approach allows for memory access responses independent of page faults, reducing the burden associated with handling large dirty pages. Consequently, it decreases the occurrence of page faults, successfully addresses the issue of large-grained dirty data, and enhances system performance and reliability.

Despite of the advantages of Kona, it overly focuses on cache-line-level accesses that it overlooks the overall hit rate of memory pages. Kona roughly marks all addresses of the fetched pages as hits, only fetching and replacing cache lines on cache misses. This comes at the cost of triggering more fine-grained remote access operations, which is less ideal for current disaggregated memory systems.

3 Memory Page Prefetching for Disaggregated Memory Systems

To address the issues, this work proposes a novel page prefetching system. Section 3.1 explains the CXL-based system, which forms the foundational of our idea. Section 3.2 introduces an approach that utilizes deep learning to show the validity of hardware prediction algorithms. Section 3.3 discusses the architecture of the system this research proposed.

3.1 CXL-based Systems

Compared to page-based systems, CXL-based systems like Kona have a design principle focused on tracking access at the cache-line granularity level in order to reduce page faults. The implementation of CXL-based systems depends on the widespread adoption of the CXL protocol. Thanks to CXL, these systems can achieve cache coherence between CPUs and connected hardware units. Kona takes advantage of this by offloading memory access sequences acquired through CXL's cache coherence activities to an additional controller. This approach results in more efficient cache-line access within a disaggregated memory system.

One notable difference between CXL-based systems like Kona and kernel-based systems is their approach to memory page handling. Kona assumes that all acquired pages are cache hits by default, meaning it doesn't anticipate any page faults because the pages are already marked as present.

3.2 Deep Learning-based Cache Replacement Policies

While deep learning has achieved remarkable success in various domains, it raises questions about whether deep learning can similarly revolutionize computer architecture, specifically in the context of hardware predictors like data prefetching.

The Glider[3] cache replacement policy uses powerful offline machine learning to develop insights that can improve the design of online hardware predictors. While this solution is impractical for overhead hardware cost, we can gain valuable insights to build a better predictor through model interpretation. More broadly, this approach suggests that deep learning can play a crucial role in systematically exploring features and feature representations that can improve the effectiveness by using much simpler models. The insights and techniques presented in this paper can inspire the design of similar solutions for other microarchitectural prediction problems, such as, branch prediction, value prediction and data prefetching.

3.3 Page Prefetching using Cache-line Information

One major challenge in current CXL-based disaggregated memory systems is efficiently prefetching as this paper described in Section 2. While machine learning has been used to address this, it faces a paradox: training data is often generated from misses. To break free from this paradox, this work can use the information from another granularity for training, which is the accessing sequences of cache-line.

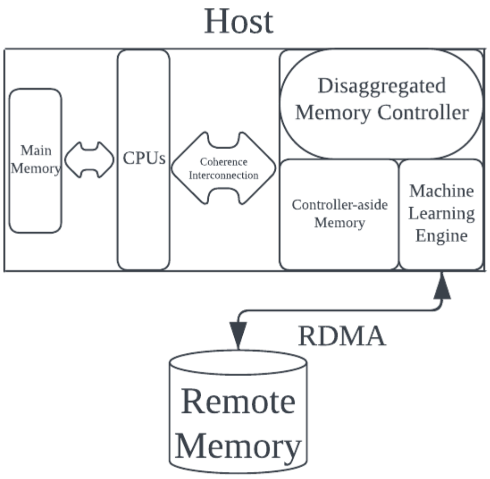


Figure 1: Cache-line based page prefetcher

This research propose a solution for page prefetching in such CXL-based disaggregate memory system. Machine learning is then applied to process cache-line data streams observed by the coherence protocol of CXL, enhancing page prefetching. This work proposes a cache-line based page prefetcher architecture shown in Figure.1. This research supposes the independent controller for the disaggregated memory, and utilize controller-aside memory as an additional cache for pages in remote memory. The cache access patterns are sent to a machine learning engine, once a certain quantity of cache lines is observed by the prediction engine, it triggers page prefetching. As a result, the proposed prefetcher can improve memory access performance of the disaggregated memory systems.

4 Future Plan

The future research will focus on the following aspects: First, this work will attempt to build a CXL-based system using Gem5 simulator, run benchmarks like SPEC2017, analyze the relationship between cache access patterns and page prefetching, and explore the machine learning models mentioned in LSTM for suitability in this scenario. Once achieve the desired results, this work will proceed to deploy this method on a simulator with CXL capabilities to validate the findings.

References

[1] H. Li et al., ”HoPP: Hardware-Software Co-Designed Page Prefetching for Disaggregated Memory,” 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA), Montreal, QC, Canada, 2023, pp. 1168-1181.

[2] Irina Calciu et al., ”Using Local Cache Coherence for Disaggregated Memory Systems.” SIGOPS Oper. Syst. Rev. 57, 1 (June 2023), 21–28.

[3] Zhan Shi, et al., 2019. Applying Deep Learning to the Cache Replacement Problem. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '52). New York, NY, USA, 413–425.

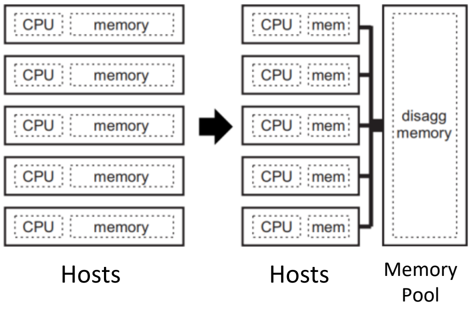
[4] Compute Express Link (CXL).
https://www.computeexpresslink.org/.

Memory Page Prefetching for Disaggregated Memory Systems

Kobayash and Sato Lab, C2IM1503, Zecheng Li

Background & Challenges

Memory Disaggregation



- Memory are **tightly coupled within each server** in traditional datacenters, causeing wide **resource underutilization**.
- Memory disaggregation is to **leverage available memory outside server boundaries like the memory from other server nodes**.
- The memory **latency increases** when the memory system is placed **far** from the hosts, leading to **performance degradation**.

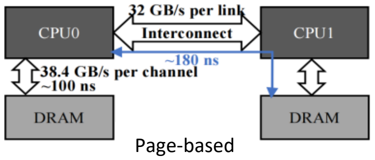
Current Approaches and Challenges

Kernel-based systems

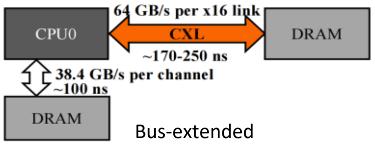
- Relies on the **virtual memory** systems, and do not need extra hardware in current remote accessing systems. Remote memory pages are accessed by triggering **page faults** through the OS.

Bus-extended systems

- Relies on the extra **coherent interconnects hardware** for remote accessing (e.g., **CXL-based** system). Results in more efficient **cache-line** access within a disaggregated memory systems.



Page-based



Bus-extended

- ◆ ***Kernel-based*** systems causes **overhead** in **dirty data tracking** and **eviction** at page level, causes a **significant waste.**
- ◆ ***CXL-based*** systems overly focuses on **cache-line accesses** that it overlooks the overall **hit rate** of **memory pages**.

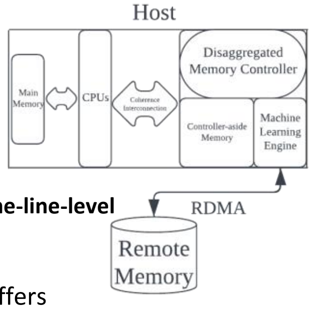
◆ To sum up, excessively favoring either **cache-lines-level** or **page-level** accesses is **not advisable**, a **balance** should be struck between them.

Approach

Multi-granularity ML-based page prefetching for CXL-based memory disaggregation

Prefetcher Architecture

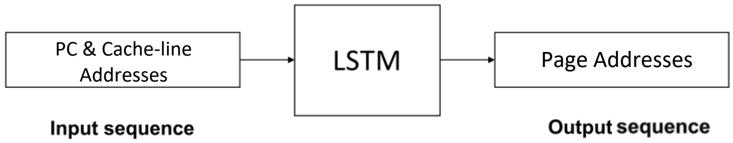
- ● Propose a cache-line based page prefetcher based on CXL-based system.



-Memory access analysis: **cache-line-level**
-Prefetching: **page-level**

- **CXL cache coherence** offers dataflow to train ML engine.
- Utilize controller-aside & host memory as an **additional cache** for remote pages.
- Page prefetching is initiated once **a certain quantity of cache lines** is observed by the prediction engine.

ML-based Prediction Model



- The **LSTM** model has proven effective in **cache behavior prediction** using the sequences of the addresses.[Zhan Shi et al., MICRO 52].
- Analyzing the **attention layer** reveals that caching decisions are primarily **influenced** by a **few key addresses** within the long history. [Zhan Shi et al., MICRO 52].
- This information allows us to create a **smaller model** to handle individual input **values** with **significant influence** with even **better precision** like **SVM**.

Conclusions

- This work aims to improve the **page prefetching** of **CXL-based disaggregated memory** systems using **cache-line** information, through **machine learning** , enhancing the **hit rate** of remote prefetching pages.
- This work will conducting **benchmark** tests like SPEC2017 on **CXL-based disaggregated** system using simulator, analyzing cache **access patterns** and page prefetching relationships, exploring **machine learning models**, with the ultimate goal of **deploying the method** on a CXL-capable simulator to validate the findings.