

Bandwidth Expansion via CXL: A Pathway to Accelerating In-Memory Analytical Processing

Wentao Huang*, Mo Sha⁺, Mian Lu[‡], Yuqiang Chen[‡], Bingsheng He*, Kian-Lee Tan*

*National University of Singapore [†]Alibaba Cloud

[‡]4Paradigm Inc.

© Copyright National University of Singapore. All Rights Reserved

Data driven application

Data driven application
 o high performance computing

Data driven application

 high performance computing
 video processing

- Data driven application

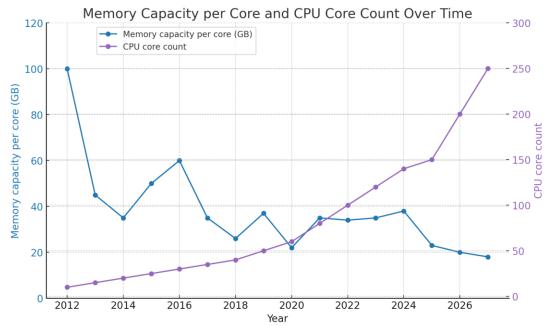
 high performance computing
 video processing
 large language model
 - 0

- Data driven application

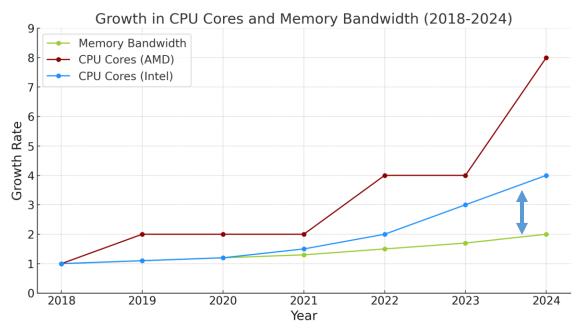
 high performance computing
 video processing
 large language model

- More data \rightarrow more memory
 - larger capacity
 - o faster memory access
 - \circ lower TCO

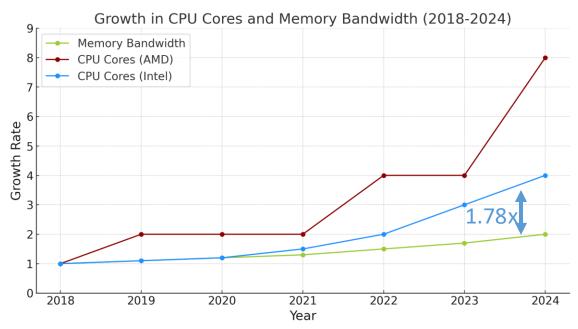
Memory capacity



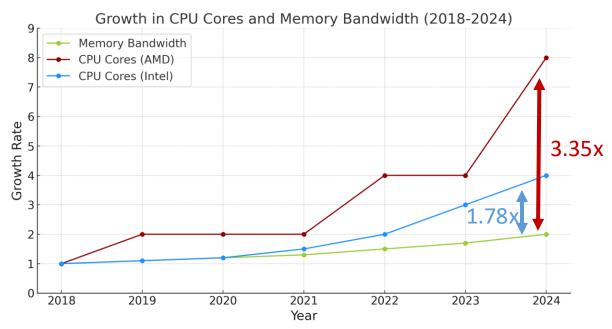
Memory bandwidth



Memory bandwidth



Memory bandwidth



• More channels per socket

More channels per socket

 increasing pin count and routing complexity

More channels per socket

 increasing pin count and routing complexity

More memory DIMMs per channel

- More channels per socket

 increasing pin count and routing complexity
- More memory DIMMs per channel

 bandwidth reduction due to signal integrity issues

- More channels per socket

 increasing pin count and routing complexity
- More memory DIMMs per channel
 bandwidth reduction due to signal integrity issues
- Memory packing technologies, e.g., 3D stacking

- More channels per socket

 increasing pin count and routing complexity
- More memory DIMMs per channel
 bandwidth reduction due to signal integrity issues
- Memory packing technologies, e.g., 3D stacking
 o escalating TCO

- More channels per socket

 increasing pin count and routing complexity
- More memory DIMMs per channel
 bandwidth reduction due to signal integrity issues
- Memory packing technologies, e.g., 3D stacking
 o escalating TCO



CXL: Compute Express Link



CXL: Compute Express Link

New protocols based on PCIe

 CXL.cache (cache coherence)
 CXL.mem (memory expansion)
 CXL.io (peripheral configuration)



CXL: Compute Express Link

New protocols based on PCIe

 CXL.cache (cache coherence)
 CXL.mem (memory expansion)
 CXL.io (peripheral configuration)

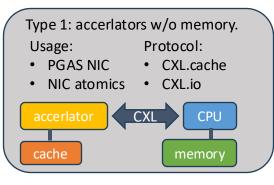


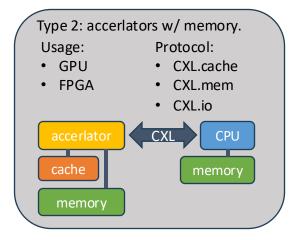
- CXL specifications
 - CXL 1.1 single machine
 - CXL 2.0 2-16 machines (single switch)
 - CXL 3.0 100+ machines (multiple switches)

CXL End-Point Devices

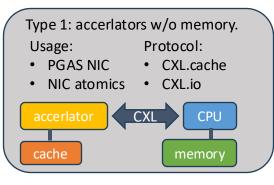
© Copyright National University of Singapore. All Rights Reserved.

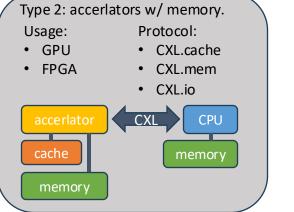
CXL End-Point Devices

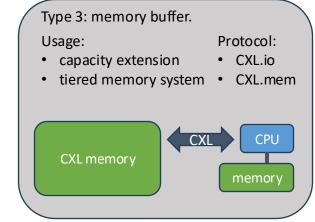




CXL End-Point Devices



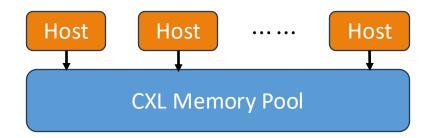




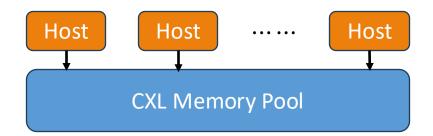
© Copyright National University of Singapore. All Rights Reserved.

- Scale-out
 - o CXL 2.0/3.0
 - \circ resource pooling
 - \circ economical cost

- Scale-out
 - o CXL 2.0/3.0
 - \circ resource pooling
 - \circ economical cost

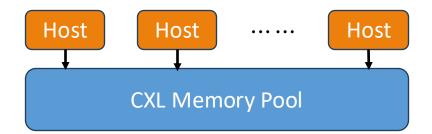


- Scale-out
 - o CXL 2.0/3.0
 - \circ resource pooling
 - o economical cost

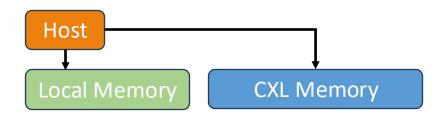


- Scale-up
 - o CXL 1.1
 - o capacity expansion
 - o bandwidth expansion
 - \circ lower TCO

- Scale-out
 - o CXL 2.0/3.0
 - \circ resource pooling
 - o economical cost



- Scale-up
 - o CXL 1.1
 - o capacity expansion
 - o bandwidth expansion
 - $\circ~$ lower TCO



• Expansion via tiering

Expansion via tiering

A Three-Tier Buffer Manager Integrating CXL Device Memory for Database Systems

Niklas Riekenbrauck Marcel Weisgut Daniel Lindner Tilmann Rabl Hasso Plattner Institute Hasso Plattner Institute Hasso Plattner Institute Hasso Plattner Institute Potsdam, Germany Potsdam, Germany Potsdam, Germany Potsdam, Germany niklas.riekenbrauck@student.hpi.de marcel.weisgut@hpi.de daniel.lindner@hpi.de tilmann.rabl@hpi.de

Abstract-Compute Express Link (CXL) is a new interconnect HyMem and Spitfire use pointer swizzling to address Autority Computer supersystems from Control of a stacking by breadfressable memory on PCI-connected devices to a CPU. The interconnect allows a database system to place buffered pages, which is an invasive technique that requires a buffer-managed data structure to be adapted accordingly [5]. data on local memory, CXL device memory, and persistent disk storage. While three-tier buffer managers integrating persistent memory (PMem) exist, CXL device memory has not been data on local memory, CXL device memory, and persistent disk integrated into a multi-tier buffer manager architecture. Exist-ing three-tier buffer managers integrating PMem use pointer to pointer swizzling [5]. While the approach shows high wizzling to address buffered pages, which is an invasive and hard-to-implement technique. This work presents a hard-to-lead low implementation complexity, it lacks buffer manager KLK device memory. The design hadri manger tata migratei CA. sevce neukof, fie Seugo In this work, we present a three size hubbles with a seven to a seven the seven as the seven the seven to a seven the se a simple integration of CXL device memory into a DBMS. We demonstrates that expanding a server's memory with CXL device evaluate the buffer manager design and its components in an

I. INTRODUCTION

on the physical layer of PCIe. CXL can connect a peripheral buffer manager into the in-memory DBMS Hyrise [7] device with a CPU, allowing cache-coherent access to the In summary, this work makes the following contributions device memory [1]. Accessing memory over CXL exhibits 1) We demonstrate the integration of CXL device memory different memory characteristics, such as higher latency than into a database system's buffer manager using virtual local memory connected via Double Data Rate (DDR) [1], [2]. Traditional disk-based database management systems (DBMSs) use secondary disk storage as primary data location. For query processing, a buffer manager loads data into local 2) We experimentally evaluate the buffer manager with promemory. With additional CXL device memory, data can be located on three tiers: on byte-addressable local and device memory and on persistent disk storage. While three-tier buffer managers exist for DRAM, persistent memory (PMem), and solid-state drive (SSD) [3], [4], CXL device memory has not 3) We discuss how page migration across multiple memory been integrated into a multi-tier buffer manager architecture. HyMem is a single-threaded buffer manager using PMem

and DRAM as selective caches on top of the SSD level [3]. Zhou et al. [4] extended the work on HyMem with Spitfire. policy to determine on which tier pages are located and has superior performance over HyMem's page migration.

In this work, we present a three-tier buffer manager that sory can be used to keep more data in memory and to reduce isolated manner with different configurations and workloads spilling data to slow disk storage. Index Termi-CXL, buffer manager, page management, database system, virtual memory, probabilistic page migration memory can be used for a buffer manager to keep more data in memory and to reduce spilling data to slow disk storage. We present primitives to integrate the proposed design into Compute Express Link (CXL) is a new interconnect based an in-memory DBMS and demonstrate the integration of the

> memory-based PID translation and probabilistic page migration (Section III). We provide integration details and an open-source implementation1 (Section III-G). totypical CXL device memory and show its benefit of supporting larger-than-local-memory workloads with higher

> throughput than a traditional two-tier design locating data only on local memory and SSD (Section IV). tiers can further be optimized (Section V).

II. BACKGROUND

This section introduces the CXL interconnect and buffer a concurrent buffer manager. It uses a probabilistic migration pool management concepts that we build our work upon.

Source code: https://github.com/hyrise/hyrise/tree/paper/buffermanager

Expansion via tiering

CXL-ANNS: Software-Hardware Collaborative Memory Disaggregation and Computation for Billion-Scale Approximate Nearest Neighbor Search

Junhyeok Jang*, Hanjin Choi*¹, Hanyeoreum Bae*, Seungjun Lee*, Miryeong Kwon*¹, Myoungsoo Jung*¹ *Computer Architecture and Memory Systems Laboratory, KAIST "Pannnesia. Inc.

Abstract

We propose CXL-ANNS, a software-hardware collaborative approach to enable highly scalable approximate nearest neigh bor search (ANNS) services. To this end, we first disageregate DRAM from the host via compute express link (CXL) and place all essential datasets into its memory pool. While this CXL memory pool can make ANNS feasible to handle billionpoint graphs without an accuracy loss, we observe that the search performance significantly degrades because of CXL's far-memory-like characteristics. To address this, CXL-ANNS considers the node-level relationship and caches the neighbors in local memory, which are expected to visit most frequently. For the uncached nodes, CXL-ANNS prefetches a set of nodes most likely to visit soon by understanding the graph traversing behaviors of ANNS. CXL-ANNS is also aware of the architectural structures of the CXL interconnect network and lets different hardware components therein collaboratively search for nearest neighbors in parallel. To improve the performance further, it relaxes the execution dependency of neighbor search tasks and maximizes the degree of search narallelism by fully utilizing all hardware in the CXL network.

Our empirical evaluation results show that CXL-ANNS exhibits 111.1× higher QPS with 93.3% lower query latency than state-of-the-art ANNS platforms that we tested. CXL-ANNS also outperforms an oracle ANNS system that has DRAM-only (with unlimited storage capacity) by 68.0% and 3.8×, in terms of latency and throughput, respectively.

1 Introduction

A Th

Niklas

Hasso F Potsda

niklas.riekenb

Abstract—Comp for attaching byte-a to a CPU. The into

data on local mem storage. While thr memory (PMem) integrated into a r ing three-tier buff

swizzling to addre hard-to-implement

buffer manager th

combines hardware translation and a p

on which tier pay

ory into a dat:

approaches co

with different on

benchmark on a C

demonstrates that memory can be u

spilling data to slo

database system

Compute Expr

on the physical lay

device with a CI

device memory [different memory

local memory con

(DBMSs) use seco

For query process

memory With ad

located on three

memory and on p

managers exist fo solid-state drive (

been integrated i

HyMem is a s

and DRAM as so

Zhou et al. [4] e

a concurrent buffe

policy to determin

superior performa

Traditional dis

Index Terms-

Dense retrieval (also known an secrets neighbor search) has taken on an important too land provides frammenda upporf for various search engines, data mining, databases, and machine learning applications such as recommendation systems [1–6]. It accumate to the classic patternibriting based or volgets using the classics patternibriting based or volgets using the classics patternibriting based or volgets using the classics and articross agiven number of objects, similar to the query object, referred to as a kenergine galabor (2007) [1–1]. To this end, dense retrieval embedds input information into a fere thousand dimensional spaces of each object called a feature vectors. These thousare works appeared a volde spectrum of datametrization (e.g., langues of query) semantic, remaining in new constructs-avaer and



(a) Previous studies. (b) CXL-based approaches. Figure 1: Various billion-scale ANNS characterizations.

accurate results than traditional search [6, 12, 13]. Even though kNN is one of the most frequently used search paradigms in various applications, it is a costly operation taking linear time to scan data [14, 15]. This computation complexity unfortunately makes dense retrieval with a billion point dataset infeasible. To make the kNN search more practical, approximate nearest neighbor search (ANNS) restricts a query vector to search only a subset of neighbors with a high chance of being the nearest ones [15-17]. ANNS exhibits good vector searching speed and accuracy, but it significantly increases memory requirement and pressure. For example, many production-level recommendation systems already adopt billion-point datasets, which require tens of TB of working memory space for ANNS; Microsoft search engines (used in Bing/Outlook) require 100B+ vectors, each being explained by 100 dimensions, which consume mon than 40TB memory space [18]. Similarly, several of Alibaba's e-commerce platforms need TB-scale memory spaces to accommodate their 2B+ vectors (128 dimensions) [19].

To address these memory pressure issues, modern ANNS techniques leverage how compression methods or employ persistent storage, such as solid state dials (SDA) and persistent memory (PMAM). for their memory capasion. For example, [76-23] split large diatest and group them isso product only has predicted symmetry and the second product only has predicted symmetry and the second person of the how the symmetry and the second the hierarchical approach [24–33] accommodates the dianese to SDAPMEM, but not how classification of the second technical symmetry and the second symmetry and the technical symmetry of the second symmetry and the second symmetry and the second symmetry of the second technical symmetry of the second symmetry of the second technical symmetry of the second symmetry of the second technical symmetry of the second symmetry of the second technical symmetry of the second symmetry of the second technical symmetry of the second symmetry of the second technical symmetry of the second symmetry of the second technical symmetry of the second symmetry of the second technical symmetry of the second symmetry of the second technical symmetry of the second symmetry of the seco

USENIX Association

2023 USENIX Annual Technical Conference 585

Expansion via tiering

Niklas Hasso F Potsda niklas.riekenbe Junhyeok Jang Abstract—Comp for attaching byte-a to a CPU. The into to a CPU. The intr data on local memory storage. While three memory (PMem) integrated into a m ing three-tier buffree hard-to-implement buffer manager tha combines hardware translation and a pr on which ther mea We propose CXL approach to enabl bor search (ANN) DRAM from the place all essentia CXL memory po on which tier pag approaches co point graphs with memory into a data with different confi search performa far-memory-like benchmark on a C considers the nod demonstrates that memory can be use spilling data to slo Index Terms-4 in local memory For the uncached most likely to vis database system behaviors of AN tectural structur different hardwa for nearest neigh Compute Expr further, it relaxes on the physical lay tasks and maxin device with a CI utilizing all hard device memory | different memory exhibits 111.1× local memory con than state-of-the Traditional dis ANNS also out (DBMSs) use sec DRAM-only (w For query process 3.8×, in terms o memory. With ad located on three 1 Introduct memory and on a Dense retrieval managers exist fo solid-state drive (taken on an imp been integrated in port for various machine learni HyMem is a si tems [1-8]. In and DRAM as so search, dense ret Zhou et al. [4] e ent objects usin; a concurrent buff of objects, simila policy to determin superior performa neighbor (kNN) input information of each object. can encode a w documents, sou

A Th

00 . .

CXL-AN

Our empirica

put query's sem LISENIX Assoc

Con

Pond: CXL-Based Me	mory Pooli	ng System	s for Cloud Platforms			
Huaicheng Li Virginia Tech Carnegie Mellon University USA	Daniel S Microso University of US	ft Azure Washington	Lisa Hsu Unaffiliated USA			
Daniel Ernst Microsoft Azure USA	Pantea Z Microso US	ft Azure	Stanko Novakovic Google USA			
Monish Shah Microsoft Azure USA	Samir Rajadnya Microsoft Azure USA		Scott Lee Microsoft USA			
Ishwar Agarwal Intel USA	Mark Microso University of Wi US	ft Azure sconsin-Madison	Marcus Fontoura Stone Co USA			
	Ricardo I Microso US	ft Azure				
ABSTRACT KEYWORDS Public doud providers seek to meet stringent performance require- Compute Express Link; CXL; memory disaggregation; men						
ments and low hardware cost. A key driver cost in main memory. Memory pooling promise utilization and thereby reduce costs. However, ing under cloud performance requirements. ³ Pord, the first memory pooling system that be formance goals and significantly reduces DRA on the Compute Express Link (CKU) standard to pool memory and two key inguidhs. First, et production traces shows that pooling across 4- to achieve most of the benefits. This enables	of performance and is to improve DRAM pooling is challeng- This paper proposes oth meets cloud per- M. cost. Pond builds for load/store access sur analysis of cloud 16 sockets is enough a small-pool design	Compare Justices Land, e-Constance Justices and Justices Landsmither and Constantiation of the Constantiationo				
with low access latency. Second, it is possibl learning models that can accurately predict I pool memory to allocate to a virtual machin	tow much local and	1 INTRODUCTION				
poot memory to allocate to a virtual machin same-NUMA-node memory performance. Our workloads shows that Pond reduces DRAM c formance within 1-5% of same-NUMA-node V CCS CONCEPTS	evaluation with 158 osts by 7% with per-	Motivation. Many public cloud custemers deploy their workloads in the form of virtual machines (VMs), for which they get virtual- ized compute with performance approaching that of a dedicated cloud, but without having to manage their own on-premises data- center. This creates a major challenge for public cloud providers: achieving excellent performance for opague VMA (c.g. provides to				
 Computer systems organization → Cl Hardware → Emerging architectures. 	oud computing: •	activitying excleant perioritaniance for Opaque visa (Ce., Hoviners to not know and should not inspect what is running inside the VMs) at a competitive hardware cost. A key driver of both performance and cost is main memory. The gold standard for memory performance is for accesses to be served by the same VUMA node as the cores that issue them, leading				
This work is licensed under a Creative Commons J timal License. ASROG 72, March 22–28, 2023, Vancouver, RC, Canada C 2023 Copyligh hold by the emonication(s). ACM 1000 V973-4669-9714-62303. https://doc.org/10.1145/375603.378805	attribution 4.0 Interna-	to latencies in tens preallocate all VM # cores. Preallocating the use of virtualiz default, for example, DRAM has become a	v noce as the cores that issue them, leading of nanoseconds. A common approach is to nemory on the same NUMA node as the VM's and statically pinning memory also facilitate ation accelerators [4], which are enabled by on AWS and Azure [12, 14]. At the same time, major portion of hardware conduct to its poor ith only nascent alternatives [72]. For example,			
	57	74				

0111

approaches co

with different on

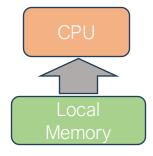
database system

mory can be a

Expansion via tiering

00 A Th Niklas Hasso I Potsd CXL-AN niklas.riekenb Com Pond: C Abstract—Comp for attaching byte-a to a CPU. The int Junhyeok Jang Hı data on local men data on local mem storage. While thr memory (PMem) integrated into a r ing three-tier buff Carnegie **TPP: Transparent Page Placement for CXL-Enabled** We propose CXI Tiered-Memory D swizzling to addre hard-to-implement approach to enab Mie bor search (ANN) Hao Wang Abhishek Dhanotia Hasan Al Maruf buffer manager th DRAM from th University of Michigan NVIDIA Meta Inc. combines hardwar translation and a p M place all essentia USA **ÚSA** USA CXL memory po on which tier pa Mir Johannes Weiner Niket Agarwal Pallab Bhattacharya point graphs wi ory into a dat: search perform Meta Inc NVIDIA NVIDIA Ishv far-memory-like USA LISA LISA benchmark on a considers the no demonstrates that Chris Petersen Mosharaf Chowdhury Shobhit Kanaujia in local memory Meta Inc University of Michigan Meta Inc spilling data to sl For the uncaches USA USA USA Index Termsmost likely to vi behaviors of Al Prakash Chauhan tectural structu Meta Inc. different hardwa TISA for nearest neigh ABSTRACT Compute Exp ABSTRACT an ideal baseline (<1% gap) that has all the memory in the local tier. further, it relaxes Public cloud prov on the physical la It is 18% better than today's Linux, and 5-17% better than existin The increasing demand for memory in hyperscale applications has tasks and maxin ments and low ha device with a CI solutions including NUMA Balancing and AutoTiering. Most of the led to memory becoming a large portion of the overall datacenutilizing all hard cost is main memo device memory | TPP patches have been merged in the Linux v5.18 release while the Our empiric ter spend. The emergence of coherent interfaces like CXL enables utilization and th different memor main memory expansion and offers an efficient solution to this probremaining ones are just pending for more discussion. exhibits 111.1× ing under cloud pe local memory con lem. In such systems, the main memory can constitute different Pond, the first mer than state-of-th Traditional di CCS CONCEPTS formance goals an memory technologies with varied characteristics. In this paper, we ANNS also out (DBMSs) use sec characterize memory usage patterns of a wide range of datacenter Software and its engineering → Operating systems; Memory on the Compute Er to pool memory ar DRAM-only (w For query proces applications across the server fleet of Meta. We, therefore, demon $agement; * Hardware \rightarrow Emerging architectures; Mem-$ 3.8×, in terms of memory With ad oduction traces s strate the opportunities to offload colder pages to slower memory ory and dense storage. iers for these applications. Without efficient memory managemen located on three to achieve most of 1 Introduc with low access la however, such systems can significantly degrade performance KEYWORDS memory and on p We propose a novel OS-level application-transparent page place Dense retrieval learning models th managers exist fo Datacenters, Operating Systems, Memory Management, Tieredment mechanism (TPP) for CXL-enabled memory. TPP employs a solid-state drive (taken on an im pool memory to a Memory, CXL-Memory, Heterogeneous System same-NUMA-node lightweight mechanism to identify and place hot/cold pages to ap port for various been integrated i propriate memory tiers. It enables a proactive page demotion from ACM Reference Format workloads shows t machine learni HyMem is a s Hasan Al Maruf, Hao Wang, Abhishek Dhanotia, Johannes Weiner, Niket ocal memory to CXL-Memory. This technique ensures a memory formance within 1 tems [1-8]. In and DRAM as s Ararwal, Pallah Bhattacharya, Chris Petersen, Mosharaf Chowdhury, Shoh headroom for new page allocations that are often related to request eight was, Faano Saanakaarya, Chin Pertenen, Annana Chowary, Sito hit Kanaujia, and Prakash Chavahan. 2023. TPP: Transparent Page Placement for CXL-Enabled Tiered-Memory. In Proceedings of the 28th ACM Internasearch, dense re Zhou et al. [4] e processing and tend to be short-lived and hot. At the same time, TPP CCS CONCEP ent objects usin; a concurrent buff can promptly promote performance-critical hot pages trapped in Bional Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3 (ASPLOS '23), March 25–29, 2023, Vancouver, BC, Canada, ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/ Computer syst Hardware → Em of objects, simil the slow CXL-Memory to the fast local memory, while minimizing policy to determi neighbor (kNN) both sampling overhead and unnecessary migrations. TPP works superior perform transparently without any application-specific knowledge and can input information be deployed globally as a kernel release. of each object. We evaluate TPP with diverse memory-sensitive workloads in can encode a w 1 INTRODUCTION (c) (t) the production server fleet with early samples of new x86 CPUs with documents, sou The surge in memory needs for datacenter applications [12, 61], CXL 1.1 support. TPP makes a tiered memory system performant as put query's sem This work is licenses tional License. combined with the increasing DRAM cost and technology scaling challences [49, 54] has led to memory becoming a significant infras ASPLOS '23, March 25 sission to make digital or hard copies of all or part of this work for persor tructure expense in hyperscale datacenters. Non-DRAM memory © 2023 Copyright held ACM ISBN 978-1-4505technologies provide an opportunity to alleviate this problem by LISENIX Assoc building tiered memory subsystems and adding higher memory capacity at a cheaper \$/GB point [5, 19, 38, 39, 46]. These technolopublish, to post on servers or to redistribute to lists, requires prior specific permiss gies, however, have much higher latency vs. main memory and ASPLOS '21 Merch 25-29 2021 Vanceuver BC Canada can significantly degrade performance when data is inefficiently 0 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 973-1-4503-9918-0-23103., \$15.00 placed in different levels of the memory hierarchy. Additionally prior knowledge of application behavior and careful application

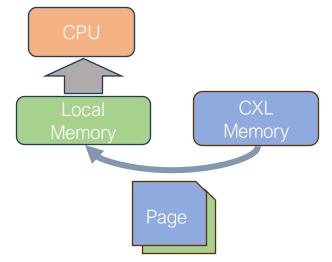
• Expansion via tiering





	T D C						
A Th							
т							
4							
	1						
Niklas							
Hasso P Potsda							
niklas.riekenbi	CXL-AN						
IIIKIIIS A JEKEUO	Com						
	Com						
	Junhyeok Jang	Pond: C					
Abstract—Compo or attaching byte-a	Junifycok Jung						
to a CPU. The inte		Hu					
data on local memo storage. While thre		Carnegie 🖲					
nemory (PMem)		Carrieger	TPP: Transpar	ent Page Pla	acement fo	or CXL-Enabled	
ntegrated into a m ng three-tier buffe	We propose CXL			Tiered-/			
wizzling to addres	approach to enabl	Di Mir		nereu-/	vieniory		
ard-to-implement suffer manager that	bor search (ANN)	MB	Hasan Al Maruf	Hao	Wang	Abhishek Dhanotia	
combines hardware	DRAM from the		University of Michigan	NVI	DIA	Meta Inc.	
translation and a pr	place all essentia	M	USA	U	SA	USA	
on which tier pag approaches combin	CXL memory po point graphs with	Mis	Johannes Weiner	Niket A	garwal	Pallab Bhattacharya	
memory into a data	search performan		Meta Inc.	NVI		NVIDIA	
with different confi benchmark on a C	far-memory-like	Ishv	USA	U	5A	USA	
demonstrates that e	considers the nod		Chris Petersen	Mocharaf	Chowdhury	Shobhit Kanaujia	
memory can be use	in local memory,		Meta Inc.	University		Meta Inc.	
spilling data to slov Index Terms—C	For the uncached most likely to visi		USA		5A	USA	
database system, v	behaviors of AN			Prakash	Chaultan		
	tectural structure				chaunan a Inc.		
	different hardwar			U			
Compute Expres	for nearest neight	ABSTRACT	ABSTRACT		on ideal baseline (1% gap) that has all the memory in the local tier.	
on the physical lay	further, it relaxes tasks and maximi	Public cloud provic ments and low ha	The increasing demand for memory in hyper	marks applications has		n today's Linux, and 5-17% better than existing	
device with a CP	utilizing all hards	cost is main memo	led to memory becoming a large portion o		solutions including	NUMA Balancing and AutoTiering. Most of the	
fevice memory [] fifferent memory	Our empirical	utilization and the	ter spend. The emergence of coherent interf	aces like CXL enables		een merged in the Linux v5.18 release while the	
local memory conr	exhibits 111.1×1	ing under cloud p	main memory expansion and offers an efficier lem. In such systems, the main memory ca		remaining ones an	e just pending for more discussion.	
Traditional dis	than state-of-the	Pond, the first mer formance goals an	memory technologies with varied characteri		CCS CONCEP	TS	
(DBMSs) use seco	ANNS also outp	on the Compute E:	characterize memory usage patterns of a wid	characterize memory usage patterns of a wide range of datacenter .Software and its engineering → Operating systems;		engineering → Operating systems; Memory	
For query process	DRAM-only (wit 3.8×, in terms of	to pool memory as		applications across the server fleet of Meta. We, therefore, demon- management; + Hardware → Emerging architectures; Me		$Hardware \rightarrow Emerging architectures; Mem-$	
memory. With add located on three t		production traces s to achieve most of	strate the opportunities to offload colder pay tiers for these applications. Without efficient		ory and dense st	orage.	
located on three t memory and on pe	1 Introduct	to achieve most of with low access la	however, such systems can significantly deg	rade performance.	KEYWORDS		
managers exist for	Dense retrieval (a	learning models th	We propose a novel OS-level application-t	We propose a novel OS-level application-transparent page place-			
solid-state drive (S	taken on an imp	pool memory to a				Memory, CXL-Memory, Heterogeneous System	
been integrated int	port for various s	same-NUMA-node workloads shows t	propriate memory tiers. It enables a proactive page demotion from ACS			ACM Reference Format:	
HyMem is a sit	machine learning tems [1-8]. In c	formance within 1	local memory to CXL-Memory. This technique ensures a memory Ha		Hasan Al Maruf, Ha	Hasan Al Maruf, Hao Wang, Abhishek Dhanotia, Johannes Weiner, Niket Agarwal, Pallab Bhattacharya, Chris Petersen, Mosharaf Chowdbury, Shob-	
and DRAM as sel Zhou et al. [4] ex	search, dense ret		headroom for new page allocations that are o	ften related to request	hit Kanaujia, and Pral	kash Chauhan. 2023. TPP: Transparent Page Placement	
a concurrent buffe	ent objects using	CCS CONCEP	processing and tend to be short-lived and hot. At the same time, TPP can promptly promote performance-critical hot pages trapped in		for CXL-Enabled Tiered-Memory. In Proceedings of the 28th ACM Interna-		
policy to determin	of objects, simila	 Computer syst 	the slow CXL-Memory to the fast local memory, while minimizing		tional Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3 (ASPLOS '23), March 25–29, 2023, Vancouver,		
superior performan	neighbor (kNN)	$Hardware \rightarrow Em$	both sampling overhead and unnecessary m	igrations. TPP works	BC, Canada. ACM, N	lew York, NY, USA, 14 pages. https://doi.org/10.1143/	
	input informatio of each object, c		transparently without any application-specif be deployed globally as a kernel release.	ic knowledge and can	3582016.3582063		
	can encode a wi		We evaluate TPP with diverse memory-s	ensitive workloads in	1 INTRODU	CTION	
	documents, soun	· ·	the production server fleet with early samples	of new x86 CPUs with		ory needs for datacenter applications [12, 61],	
	put query's sema	This work is licensed	CXL 1.1 support. TPP makes a tiered memory	system performant as	combined with the	ory needs for datacenter applications [12, 61], increasing DRAM cost and technology scaling	
		tional License.			challenges [49, 54]	has led to memory becoming a significant infras-	
		ASPLOS '23, March 25-1 © 2023 Copyright held I	Permission to make digital or hard copies of all or part or classroom use is granted without fee provided that copies for profit or commercial advantage and that copies bear th	of this work for personal or are not made or distributed		n hyperscale datacenters. Non-DRAM memory	
	USENIX Associa	ACM ISBN 978-1-4505-1 https://doi.org/10.1145/	fer profit or commercial advantage and that copies bear th on the first page. Copyrights for components of this wor	is notice and the full citation k owned by others than the		ide an opportunity to alleviate this problem by mory subsystems and adding higher memory	
	00001111 A00000	understanged in 1600.	author(s) must be honored. Abstracting with credit is perr republish, to not on servers or to redistribute to lists, requi		capacity at a chear	er \$/GB point [5, 19, 38, 39, 46]. These technolo-	
			and for a fee. Request permissions from permissions/Daca	nes pror specific permission morg.	gies, however, hav	e much higher latency vs. main memory and	
			ASPLOS '23, March 25–29, 2023, Vancouver, BC, Canada 9 2023 Copyright held by the owner/asthor(s), Publicatio	on rights licensed to ACM.		legrade performance when data is inefficiently	
			ACM ISBN 978-1-4505-9918-0:23/03\$15.00 https://doi.org/10.1145/3582016.3582063	-		levels of the memory hierarchy. Additionally, of application behavior and careful application	

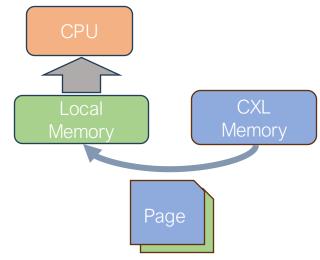
Expansion via tiering



A Th								
1								
Niklas	1							
Hasso P								
Potsda								
niklas.riekenbi	CXL-AN							
	Com							
Abstract-Comp	Junhyeok Jang	Pond: C						
Abstract—Compt r attaching byte-a	Sumycok Sung							
a CPU. The int		Hı						
ta on local memo orage. While thre		V D						
emory (PMem)		Carnegie 🦲	TPP. Transnar	ant Page Pla	acement fo	r CXI -Enabled		
tegrated into a m		- grann	TPP: Transparent Page Placement for CXL-Enabled					
g three-tier buffe	We propose CXL	Di		Tiered-/	Memory			
izzling to addres rd-to-implement	approach to enabl	Mis						
ffer manager that	bor search (ANN) DRAM from the		Hasan Al Maruf	Hao		Abhishek Dhanotia		
mbines hardware	place all essentia	M	University of Michigan	NVI		Meta Inc.		
anslation and a pr which tier pag	CXL memory po	Mi	USA	US	5A	USA		
proaches combin	point graphs with	MB	Johannes Weiner	Niket A	garwal	Pallab Bhattacharya		
emory into a data	search performan		Meta Inc.	NVI	DIA	NVIDIA		
th different confi	far-memory-like	Ishv	USA	US	5A	USA		
nchmark on a C monstrates that e	considers the nod		Chris Petersen	Mosharaf C	The second have seen	Shobhit Kanaujia		
emory can be use	in local memory,		Meta Inc.	University of		Meta Inc.		
illing data to slov Index Terms-C	For the uncached		USA	US	a Muchigan	USA		
tabase system, v	most likely to visi		00A			CON		
	behaviors of AN tectural structure			Prakash				
	different hardwar			Meta				
	for nearest neight	ABSTRACT		US	5A			
Compute Expres	further, it relaxes	Public cloud provid	ABSTRACT		an ideal baseline (<)	(% gap) that has all the memory in the local tier.		
the physical lay	tasks and maximi	ments and low ha	The increasing demand for memory in hypers	scale applications has	It is 18% better than	today's Linux, and 5-17% better than existing		
vice with a CP	utilizing all hards	cost is main memor	led to memory becoming a large portion of			NUMA Balancing and AutoTiering. Most of the		
vice memory [] fferent memory	Our empirical	utilization and the	ter spend. The emergence of coherent interfa	ces like CXL enables	TPP patches have be	ren merged in the Linux v5.18 release while the		
cal memory conr	exhibits 111.1×1	ing under cloud p	main memory expansion and offers an efficient		remaining ones are	just pending for more discussion.		
Traditional dis	than state-of-the	Pond, the first mer	lem. In such systems, the main memory can memory technologies with varied characterist		CCS CONCEPT	15		
BMSs) use seco	ANNS also outp	formance goals an on the Compute E1	characterize memory usage patterns of a wid			engineering → Operating systems: Memory		
r query processi	DRAM-only (wit	to pool memory as	applications across the server fleet of Meta. V		management; + Hardware → Emerging architectures; Mem-			
emory. With add	3.8×, in terms of	production traces s	strate the opportunities to offload colder page	es to slower memory	ory and dense sto	rage.		
cated on three t	1 Introduct	to achieve most of	tiers for these applications. Without efficient n		VEVWORCE			
emory and on pe	Dense retrieval (with low access la	however, such systems can significantly degr We propose a novel OS-level application.tr		KEYWORDS			
anagers exist for	Dense retrieval (taken on an imp	learning models th pool memory to a	We propose a novel OS-level application-transparent page place- ment mechanism (TPP) for CXL-enabled memory. TPP employs a		Datacenters, Opera	ting Systems, Memory Management, Tiered-		
lid-state drive (S	port for various s	same-NUMA-node	lightweight mechanism to identify and place hot/cold pages to ap-			aory, Heterogeneous System		
en integrated int HyMem is a sit	machine learning	workloads shows t	propriate memory tiers. It enables a proactive page demotion from		ACM Reference For	mat: Wang, Abhishek Dhanotia, Johannes Weiner, Niket		
d DRAM as sel	tems [1-8]. In c	formance within 1				Wang, Abhishek Dhanotia, Johannes Weiner, Niket acharya, Chris Petersen, Mosharaf Chowdhury, Shob-		
iou et al. [4] ex	search, dense ret		neuroscine and two dt be chort-lised and hot At the neuro time TDP hit Kanaujia, and Prakash Chauhan. 2023. TPP: Transparent Page Placemen			ash Chauhan. 2023. TPP: Transparent Page Placement		
concurrent buffe	ent objects using	CCS CONCEP	can promptly promote performance-critical	hot pages trapped in		ed-Memory. In Proceedings of the 28th ACM Interna-		
licy to determin	of objects, simila	 Computer syst 	the slow CXL-Memory to the fast local memory, while minimizing		tional Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3 (ASPLOS '23), March 25–29, 2023, Vancouver,			
perior performan	neighbor (kNN)	Hardware → Em	both sampling overhead and unnecessary mi	igrations. TPP works	BC, Canada. ACM, No	w York, NY, USA, 14 pages. https://doi.org/10.1145/		
	input informatio		transparently without any application-specifi	c knowledge and can	3582016.3582063			
	of each object, c		be deployed globally as a kernel release. We evaluate TPP with diverse memory-se	mitime workloads in				
	can encode a wi		the production server fleet with early samples of	of new x86 CPUs with	1 INTRODUC			
	documents, soun put query's sema		CXL 1.1 support. TPP makes a tiered memory :			ory needs for datacenter applications [12, 61],		
	put query's send	This work is licensed tional License.			combined with the	increasing DRAM cost and technology scaling as led to memory becoming a significant infras-		
		ASPLOS '23, Merch 25-1	Permission to make digital or hard copies of all or part of	f this work for personal or		as led to memory becoming a significant infras- hyperscale datacenters. Non-DRAM memory		
		© 2023 Copyright held I ACM ISBN 978-1-4500-1	classroom use is granted without fee provided that copies a fer profit or commercial advantage and that copies bear this	are not made or distributed a notice and the full citation		le an opportunity to alleviate this problem by		
	USENIX Associa	https://doi.org/10.1145/	on the first page. Copyrights for components of this work author(s) must be honored. Abstracting with credit is permi	owned by others than the	building tiered mer	mory subsystems and adding higher memory		
			republish, to post on servers or to redistribute to lists, requir	es prior specific permission		rr \$/GB point [5, 19, 38, 39, 46]. These technolo-		
			and/or a fee. Request permissions from permissions@acm ASTON '21 March 25-22 2021 Timemar BC Consider	Lorg.		e much higher latency vs. main memory and		
						grade performance when data is inefficiently		
			© 2023 Copyright held by the owner/author(s). Publication ACM ISBN 978-1-4505-9918-0-23/03\$15.00	n rights licensed to ACM.		levels of the memory hierarchy. Additionally,		

 \checkmark

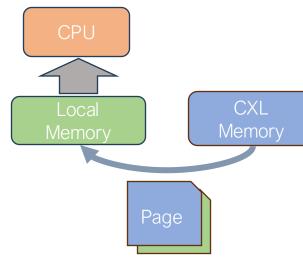
Expansion via tiering
 capacity expansion



A Th		., .							
т									
1									
	1								
Niklas									
Hasso P									
Potsda	CXL-AN								
das.riekenbe	Com								
	Com								
	Junhyeok Jang	Pond: C)							
ract—Compu- ching byte-a	Junitycok Jang								
PU. The inte		Hu							
a local memo		v 🔿							
y (PMem)		Carnegie 🦲	TPP: Transnan	ent Page Pla	cement fo	r CXI -Enabled			
ted into a n			iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	TPP: Transparent Page Placement for CXL-Enabled					
ree-tier buffe	We propose CXL approach to enabl	Di		Tiered-Memory					
-implement	bor search (ANN)	Mis	Hasan Al Maruf	Hao V	lang	Abhishek Dhanotia			
manager that ies hardware	DRAM from the		University of Michigan	NVII		Meta Inc.			
es hardware tion and a pr	place all essentia	M	USA	US		USA			
ich tier pag	CXL memory po	Mir							
ches combin y into a data	point graphs with		Johannes Weiner Meta Inc.	Niket Aj		Pallab Bhattacharya NVIDIA			
fferent confi	search performan far-memory-like	Ishy	USA	US		USA			
nark on a C	considers the nod								
strates that e y can be use	in local memory,		Chris Petersen	Mosharaf C		Shobhit Kanaujia			
data to slow	For the uncached		Meta Inc.	University of		Meta Inc.			
x Terms—C se system, v	most likely to visi		USA	US.	`	USA			
se system, t	behaviors of AN			Prakash C	hauhan				
	tectural structure different hardwar			Meta					
	for nearest neight	ABSTRACT		US.	`				
apute Expres	further, it relaxes	Public cloud provis	ABSTRACT		an ideal baseline (<	1% gap) that has all the memory in the local tier.			
physical lay with a CP	tasks and maximi	ments and low ha	The increasing demand for memory in hyper-			today's Linux, and 5–17% better than existing			
memory []	utilizing all hards	cost is main memor	led to memory becoming a large portion of			NUMA Balancing and AutoTiering. Most of the een merged in the Linux v5.18 release while the			
nt memory	Our empirical	utilization and the ing under cloud p	ter spend. The emergence of coherent interfa main memory expansion and offers an efficien	aces like CXL enables t solution to this proh-		just pending for more discussion.			
nemory conr	exhibits 111.1×1 than state-of-the-	Pond, the first mer	lem. In such systems, the main memory car						
litional dis	ANNS also outp	formance goals an	memory technologies with varied characteris	tics. In this paper, we	CCS CONCEP				
Ss) use seco ery processi	DRAM-only (wit	on the Compute Er	characterize memory usage patterns of a wid applications across the server fleet of Meta. V		 Software and its 	$engineering \rightarrow Operating systems; Memory$			
y. With add	3.8×, in terms of	to pool memory as production traces s	applications across the server neet of Meta. strate the opportunities to offload colder pag		ory and dense sto	ardware → Emerging architectures; Mem-			
I on three t		to achieve most of	tiers for these applications. Without efficient r	nemory management,	ory and denie sto	ange.			
y and on pe	1 Introduct	with low access la	however, such systems can significantly degr	rade performance.	KEYWORDS				
ers exist for	Dense retrieval (learning models th	We propose a novel OS-level application-tr ment mechanism (TPP) for CXL-enabled me	ansparent page place-	Datacenters, Operating Systems, Memory Management, Tiered-				
tate drive (S	taken on an imp port for various s	pool memory to a same-NUMA-node	lightweight mechanism to identify and place	lightweight mechanism to identify and place hot/cold pages to ap-		Memory, CXL-Memory, Heterogeneous System			
ntegrated int fem is a sit	machine learning	workloads shows t	propriate memory tiers. It enables a proactive	propriate memory tiers. It enables a proactive page demotion from		ACM Reference Format: Hasan Al Maruf, Hao Wang, Abhishek Dhanotia, Johannes Weiner, Niket			
RAM as sel	tems [1-8]. In c	formance within 1			Hasan Al Maruf, Hao Azarwal, Pallah Bhatt	Wang, Abhishek Dhanotia, Johannes Weiner, Niket acharya, Chris Petersen, Mosharaf Chowdhury, Shob-			
et al. [4] ex	search, dense ret		hit Kanaujia, and Prakash Chauhan. 2023. TPP: Transparent Page Placem			ash Chauhan. 2023. TPP: Transparent Page Placement			
urrent buffe	ent objects using	CCS CONCEP	can promptly promote performance-critical		for CXL-Enabled Tier	red-Memory. In Proceedings of the 28th ACM Interna- Irchitectural Support for Programming Languages and			
to determin	of objects, simila	Computer syst	the slow CXL-Memory to the fast local memory	ory, while minimizing	Operating Systems, Vi	slume 3 (ASPLOS '23), March 25–29, 2023, Vancouver,			
or performan	neighbor (kNN)	$Hardware \rightarrow Em$	both sampling overhead and unnecessary m transparently without any application-specifi	igrations. TPP works	BC, Canada, ACM, N 3582016 3582063	rw York, NY, USA, 14 pages. https://doi.org/10.1143/			
	input informatio of each object, c		be deployed globally as a kernel release.	e ano wreuge and can	3362016.3582063				
	can encode a wi		We evaluate TPP with diverse memory-se		1 INTRODU	CTION			
	documents, soun	<u> </u>	the production server fleet with early samples			ory needs for datacenter applications [12, 61],			
	put query's sema	This work is licensed tional License.	CXL 1.1 support. TPP makes a tiered memory	system performant as	combined with the	increasing DRAM cost and technology scaling			
		tional License. ASPLOS '23, March 25-1	Permission to make digital or hard copies of all or part o	fills and for more l	challenges [49, 54] I	has led to memory becoming a significant infras-			
		© 2023 Convright held I				hyperscale datacenters. Non-DRAM memory			
	USENIX Associa	ACM ISBN 978-1-4503-4 https://doi.org/10.1145/	for profit or commercial advantage and that copies bear thi on the first page. Copyrights for components of this work	is notice and the full citation k owned by others than the		de an opportunity to alleviate this problem by mory subsystems and adding higher memory			
			author(s) must be honored. Abstracting with credit is perm resublish to need on servers or to redistribute to lists, remain	itted. To copy otherwise, or respector specific permission	capacity at a cheap	er \$/GB point [5, 19, 38, 39, 46]. These technolo-			
			and/or a fee. Request permissions from permissions(Pacm	norg.	gies, however, hav	e much higher latency vs. main memory and			
-			ASPLOS '23, March 25–29, 2023, Vanceuver, BC, Canada © 2023 Copyright held by the owner/author(s). Publicatio	n rights licensed to ACM.	can significantly d	egrade performance when data is inefficiently			
				-		levels of the memory hierarchy. Additionally,			
			https://doi.org/10.1145/3582016.3582063			f application behavior and careful application			

Expansion via tiering

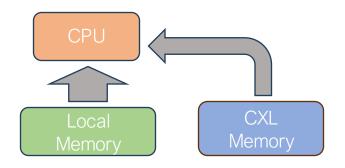
 capacity expansion
 bandwidth expansion
 X



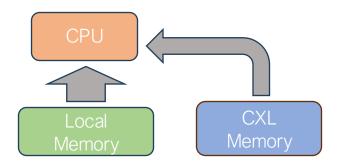


742

• Expansion within a unified tier

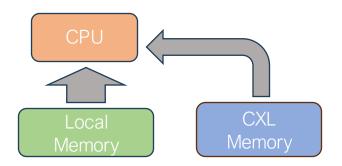


Expansion within a unified tier
 ○ capacity expansion ✓



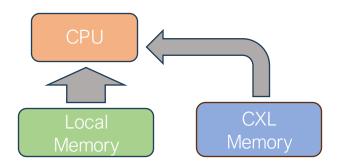
 \checkmark

- Expansion within a unified tier
 - capacity expansion
 - \circ bandwidth expansion \checkmark
 - DDR bandwidth
 - PCIe bandwidth

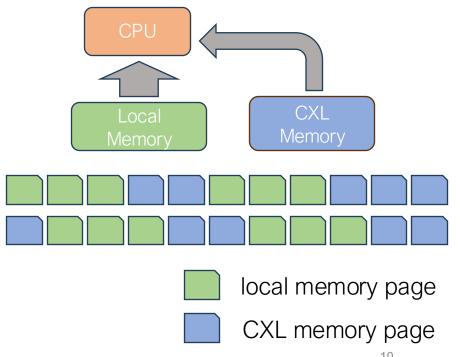


 \checkmark

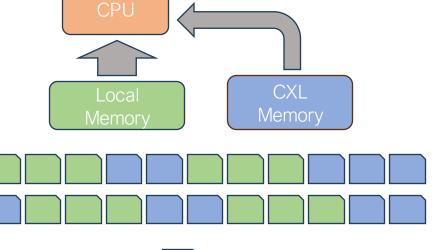
- Expansion within a unified tier
 - capacity expansion
 - \circ bandwidth expansion \checkmark
 - DDR bandwidth
 - PCIe bandwidth
- Memory interleaving



- Expansion within a unified tier
 - capacity expansion
 - \circ bandwidth expansion \checkmark
 - **DDR** bandwidth •
 - PCle bandwidth •
- Memory interleaving



- Expansion within a unified tier
 - o capacity expansion
 - \circ bandwidth expansion \checkmark
 - DDR bandwidth •
 - PCle bandwidth •



 Memory interleaving bandwidth-aware load balancing

local memory page



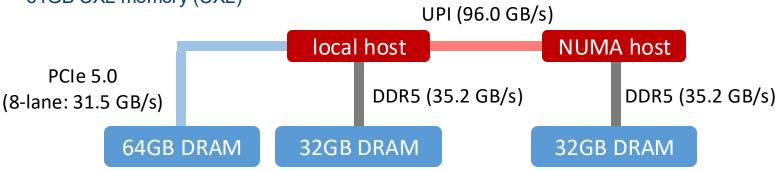
- Testbed
 - Montage CXL type3 controller

- Montage CXL type3 controller
- Intel Sapphire Rapids CPU (32 cores)

- Montage CXL type3 controller
- Intel Sapphire Rapids CPU (32 cores)
- Sub-NUMA clustering (SNC) mode

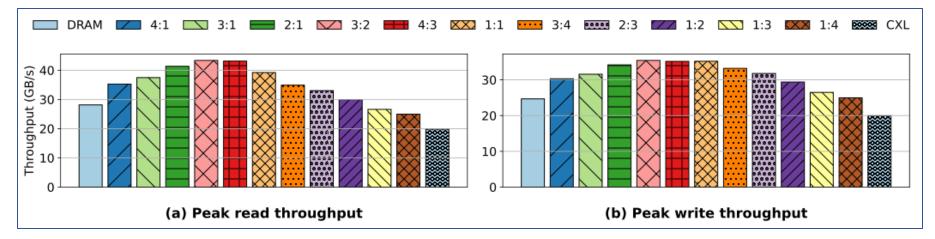
- Montage CXL type3 controller
- Intel Sapphire Rapids CPU (32 cores)
- Sub-NUMA clustering (SNC) mode
 - 32GB local memory (DRAM)
 - 32GB NUMA memory (NUMA)
 - 64GB CXL memory (CXL)

- Montage CXL type3 controller
- Intel Sapphire Rapids CPU (32 cores)
- Sub-NUMA clustering (SNC) mode
 - 32GB local memory (DRAM)
 - 32GB NUMA memory (NUMA)
 - 64GB CXL memory (CXL)



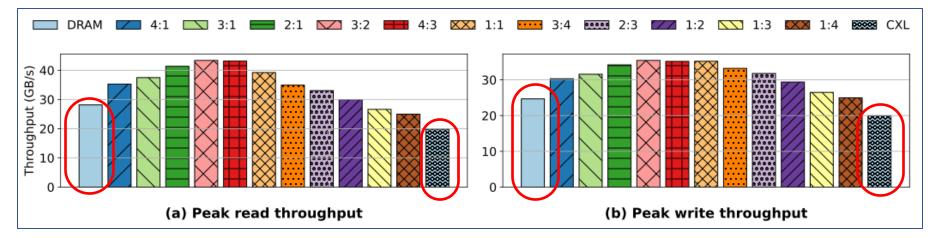
Experiment workload

sequential access over 2 billion records (8-byte) with 32 cores.



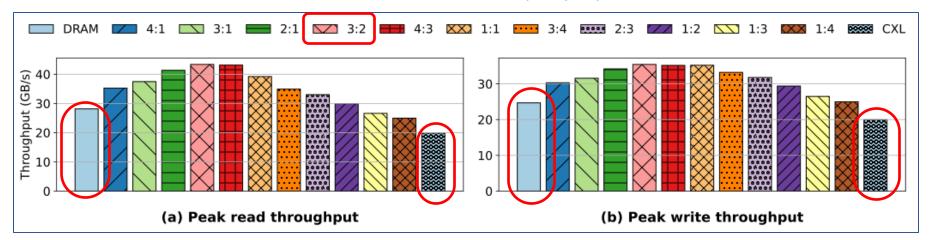
Experiment workload

sequential access over 2 billion records (8-byte) with 32 cores.



Experiment workload

o sequential access over 2 billion records (8-byte) with 32 cores.



Experiment workload

sequential access over 2 billion records (8-byte) with 32 cores.

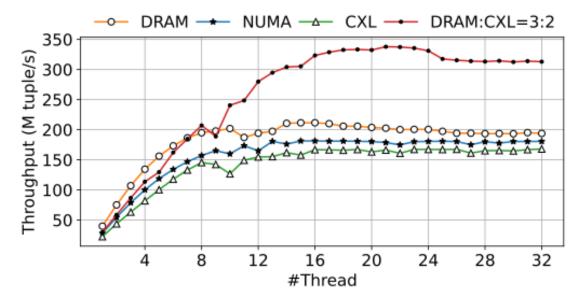




Hash Join Evaluation

Experiment workload

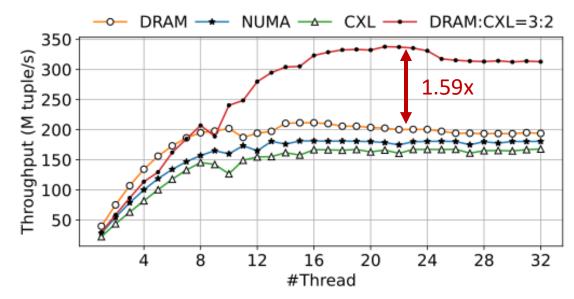
o uniform distribution, 256M ⋈ 1024M (8-byte)



Hash Join Evaluation

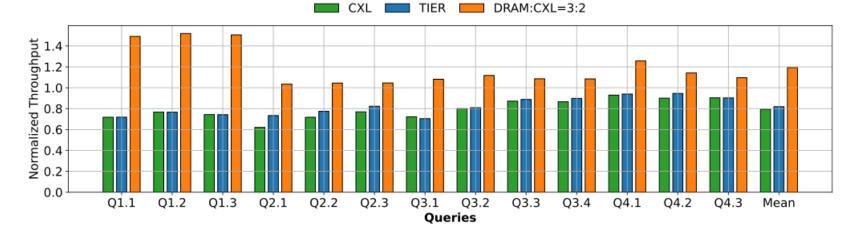
Experiment workload

o uniform distribution, 256M ⋈ 1024M (8-byte)



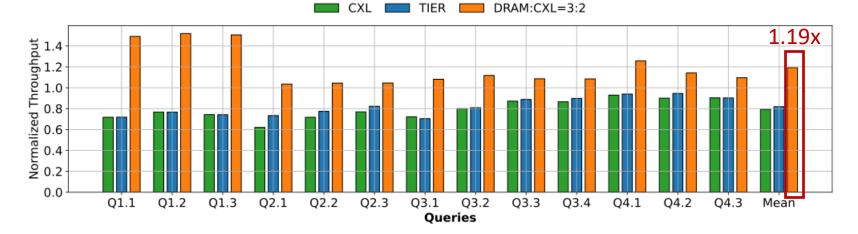
Star Schema Benchmark

- SSB workload
 - o column store
 - scaling factor: 30



Star Schema Benchmark

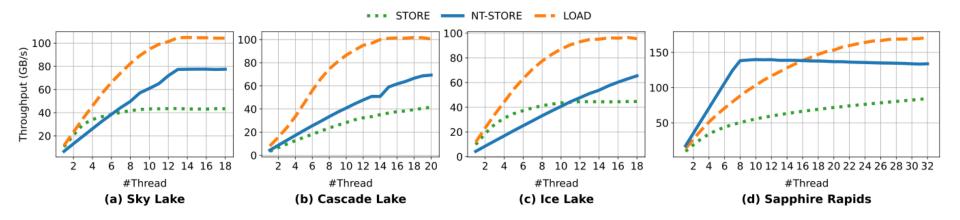
- SSB workload
 - o column store
 - scaling factor: 30



Limited Bandwidth in Socket

Experiment workload

sequential access over 2 billion records (8-byte)









CXL scale-up solution

 capacity expansion
 bandwidth expansion





- CXL scale-up solution

 capacity expansion
 bandwidth expansion
- Perspectives
 - o memory interleaving tuning
 - o fine-grained data placement
 - o bandwidth-aware load balancing





- CXL scale-up solution

 capacity expansion
 bandwidth expansion
- Perspectives
 - o memory interleaving tuning
 - o fine-grained data placement
 - o bandwidth-aware load balancing





GitHub Link

THANK YOU