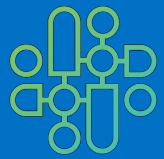
CXL Memory as Persistent Memory for Disaggregated HPC

A Practical Approach



SC23
Denver, CO | i am hpc.

IXPUG Annual Conference 2023

Contributors and Sponsors

Yehonatan Fridman
Ben-Gurion University, NRCN, Israel

Suprasad Mutalik Desai
Intel, India

Navneet Singh
Intel, India

Thomas Willhalm
Intel, Germany

Gal Oren
Technion, NRCN, Israel
galoren@cs.technion.ac.il

Centers of Excellence

Full Paper: https://arxiv.org/pdf/fdp.2308.10714
MTSA'23: https://sc23.supercomputing.org/presentation/?id=wksp114&sess=sess113

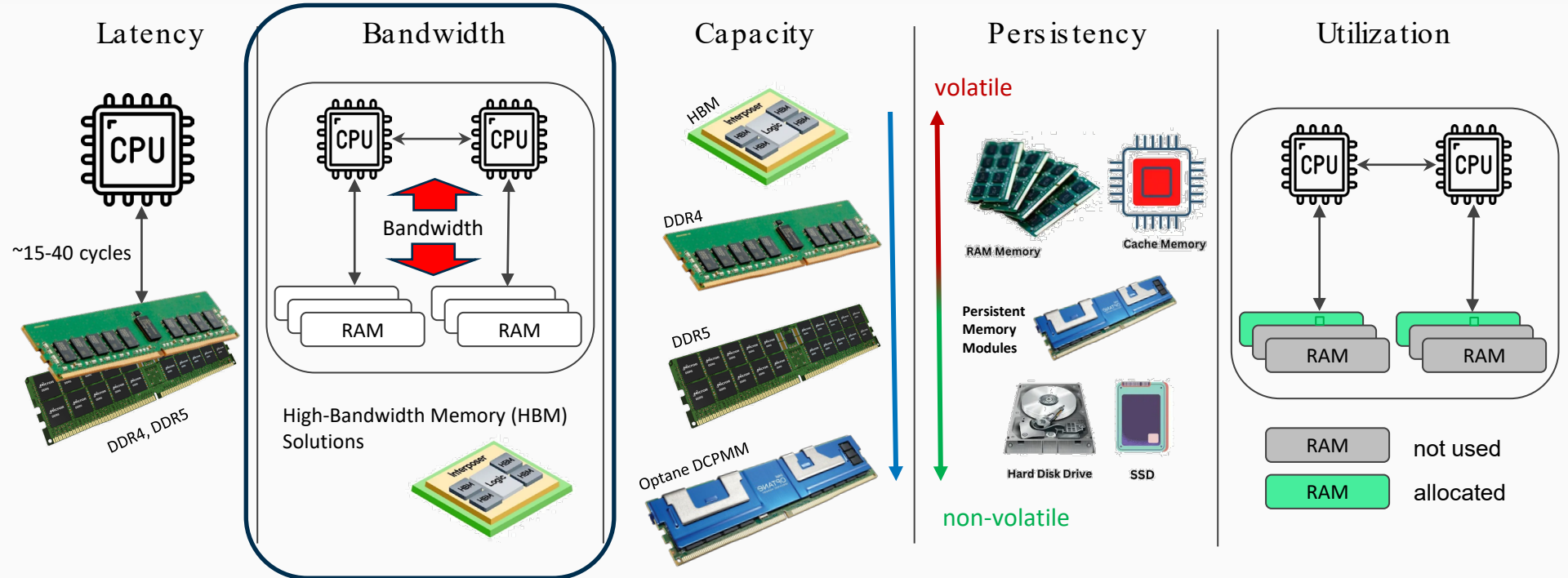
Talk Outline

- What are the Challenges with Memory?
- Persistent Memory in HPC
- CXL Disaggregated Memory for HPC
- CXL as Persistent Memory
- Physical Experiment Setup
- Performance Evaluation
- Conclusions & Future Work

Talk Outline

- **What are the Challenges with Memory?**
- Persistent Memory in HPC
- CXL Disaggregated Memory for HPC
- CXL as Persistent Memory
- Physical Experiment Setup
- Performance Evaluation
- Conclusions & Future Work

What are the Challenges with Memory?



Talk Outline

- What are the Challenges with Memory?
- **Persistent Memory in HPC**
- CXL Disaggregated Memory for HPC
- CXL as Persistent Memory
- Physical Experiment Setup
- Performance Evaluation
- Conclusions & Future Work

Persistent Memory for HPC

Persistent Memory Characteristics:

Direct access

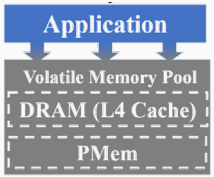
Persistency

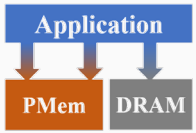
Large capacity

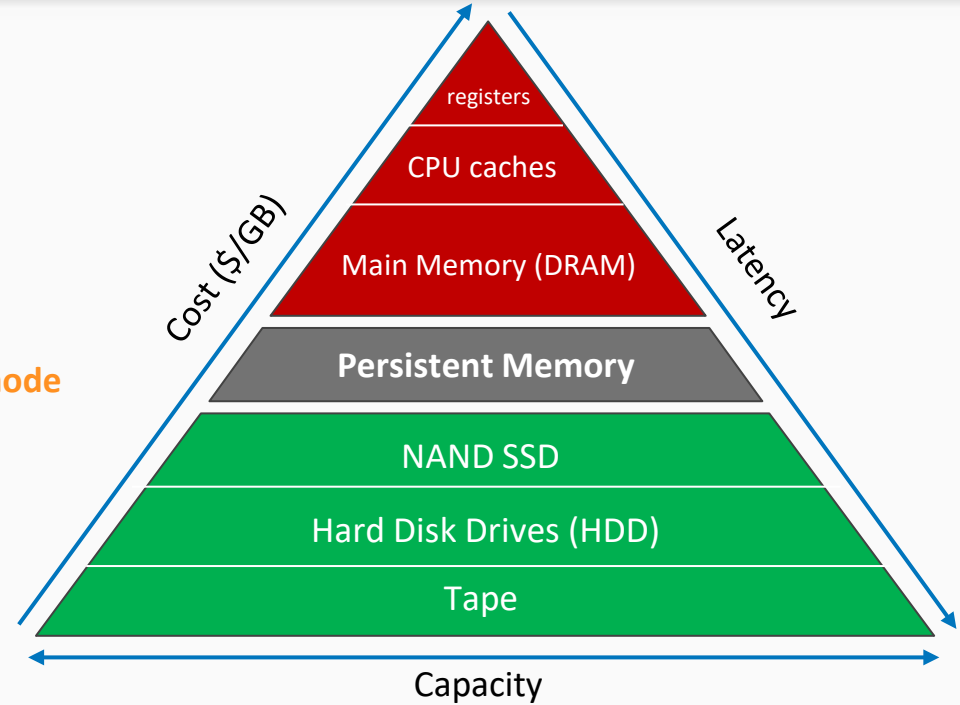
Performance

Persistent Memory modules

Memory mode or **App-Direct mode**







Persistent Memory for HPC

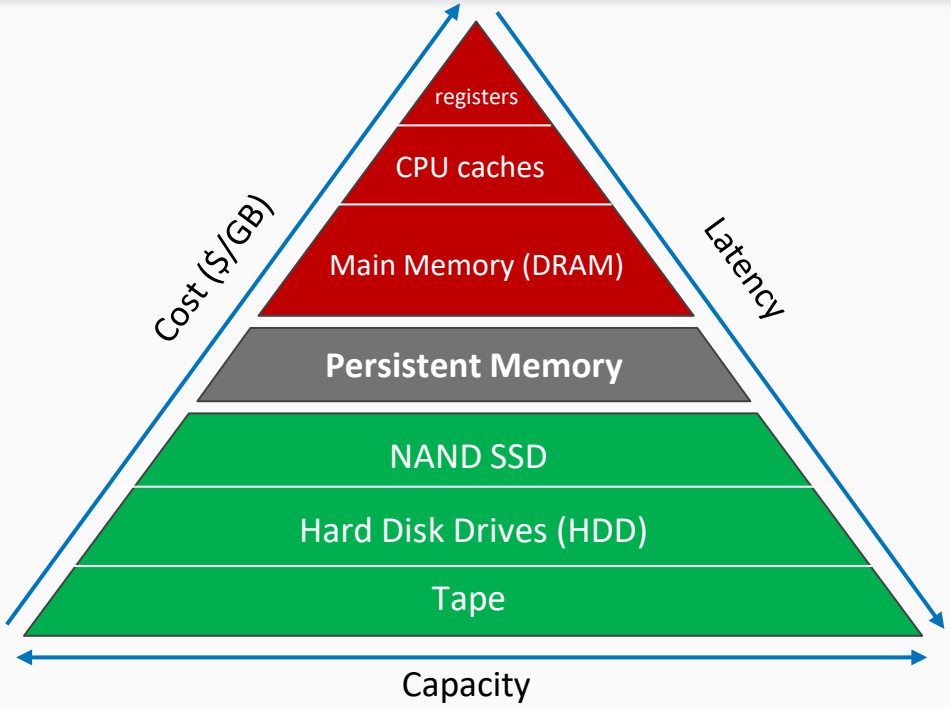
Advantages for HPC:

Main memory expansion

- Execution of larger scientific problems.

Fast storage for diagnostics and fault tolerance

- Using direct access file systems (or)
- Accessing directly within applications using the PMDK programming model.



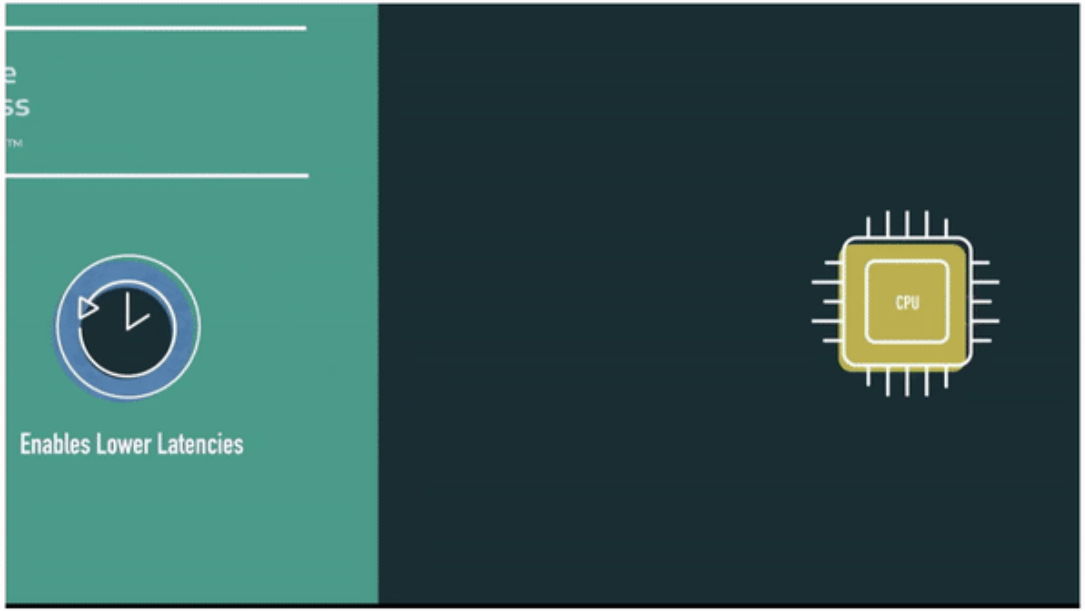
Talk Outline

- What are the Challenges with Memory?
- Persistent Memory in HPC
- **CXL Disaggregated Memory for HPC**
- CXL as Persistent Memory
- Physical Experiment Setup
- Performance Evaluation
- Conclusions & Future Work

CXL Disaggregated Memory for HPC

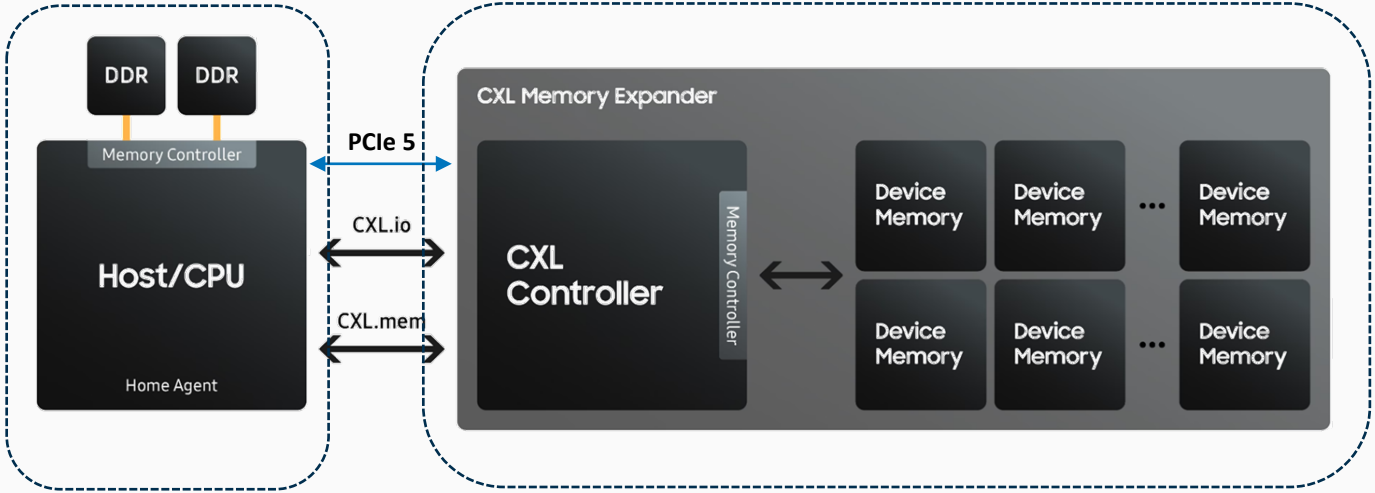
Compute Express Link (CXL):

- New breakthrough **high-speed** CPU-to-Device interconnect.
- Builds upon **PCI Express®** infrastructure.
- Allows **common memory space** between host and devices.
- Maintains **memory coherency**.
- Delivered as an **open industry** standard.



https://www.intel.com/content/www/us/en/products/details/fpga/intellectual-property/interface-protocols/cxl-ip.html

CXL Disaggregated Memory for HPC



Representative CXL Type 3 Device

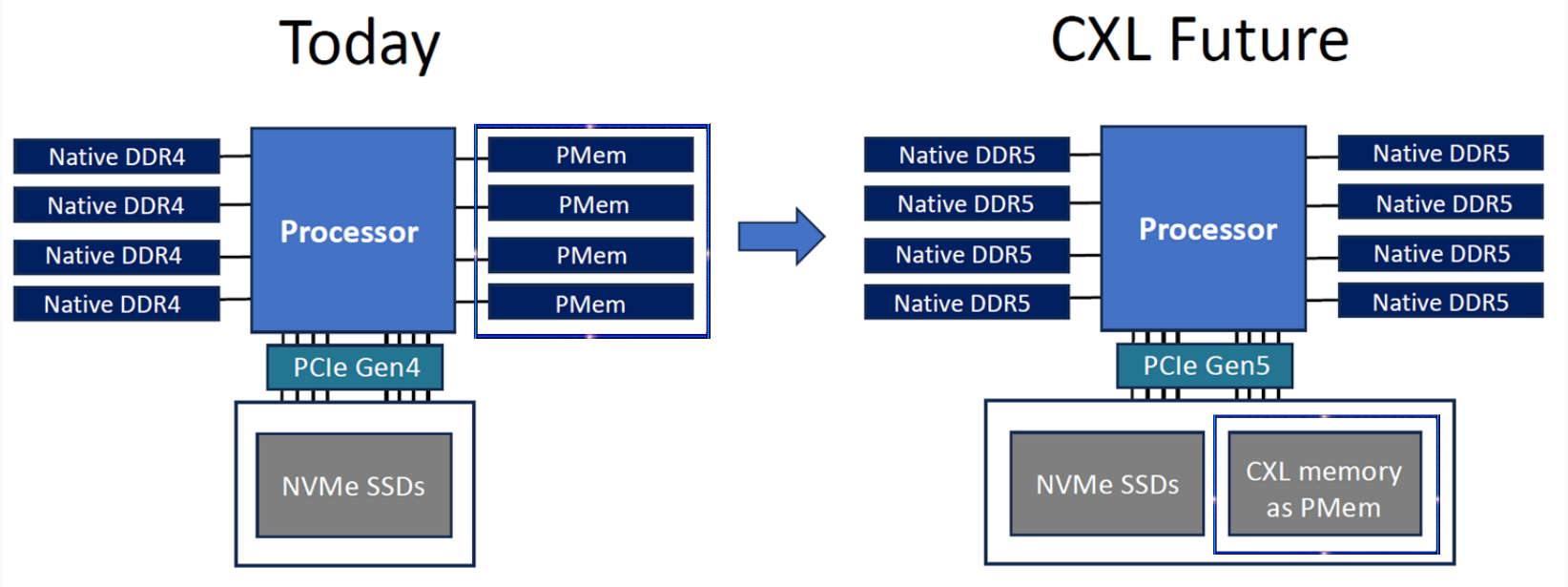
https://semiconductor.samsung.com/emea/news-events/tech-blog/expanding-the-limits-of-memory-bandwidth-and-density-samsungs-cxl-dram-memory-expander/

Our Question:

Can CXL memory serve as Persistent Memory?

- Considering both **Memory Mode** & **App-Direct Mode** functionalities.
- What about performance?

CXL as Persistent Memory



Talk Outline

- What are the Challenges with Memory?
- Persistent Memory in HPC
- CXL Disaggregated Memory for HPC
- **CXL as Persistent Memory**
- Physical Experiment Setup
- Performance Evaluation
- Conclusions & Future Work

CXL as Persistent Memory

CXL as Persistent Memory Characteristics:

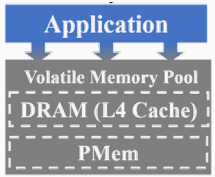
Direct access

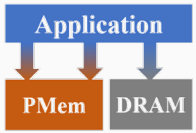
Persistency

Large capacity

Performance

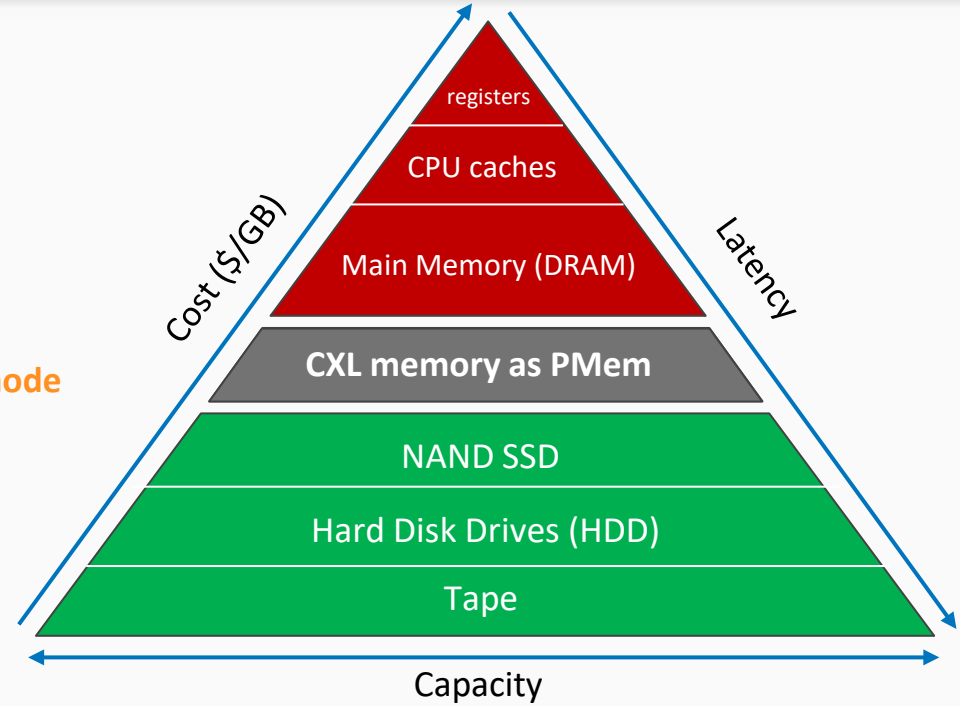
Memory mode or **App-Direct mode**





Persistent Memory modules

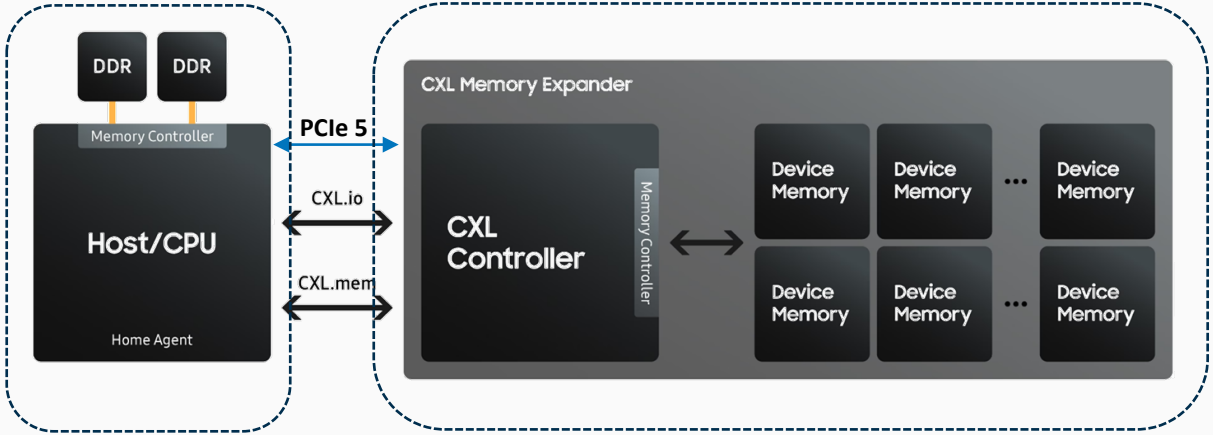
CXL-attached Memory



CXL as Persistent Memory

In **Memory Mode**:

- CXL implements **memory and cache coherency** without software interventions.



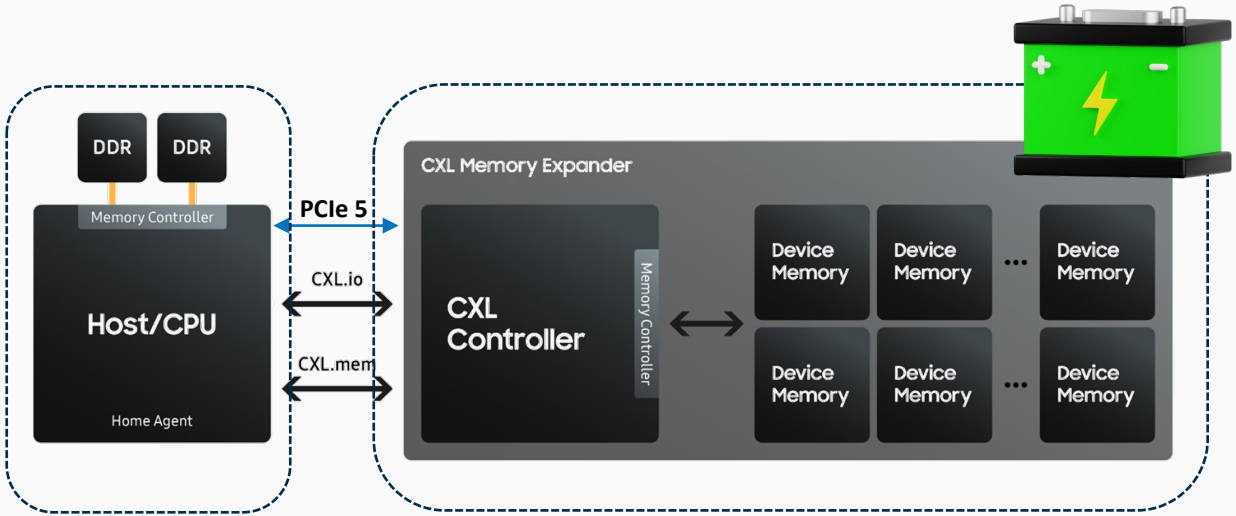
Representative CXL Type 3 Device

https://semiconductor.samsung.com/emea/news-events/tech-blog/expanding-the-limits-of-memory-bandwidth-and-density-samsungs-cxl-dram-memory-expander/

CXL as Persistent Memory

In **App-Direct Mode**:

- CXL 2.0 supports the **persistent memory** programming model.
- CXL-attached memory can function as persistent memory with the support of **backup batteries** or on-market **NVRAM** products.



Representative CXL Type 3 Device

https://semiconductor.samsung.com/emea/news-events/tech-blog/expanding-the-limits-of-memory-bandwidth-and-density-samsungs-cxl-dram-memory-expander/

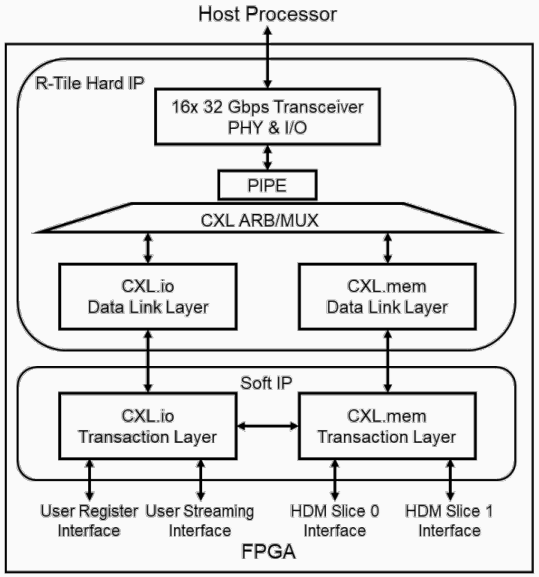
Talk Outline

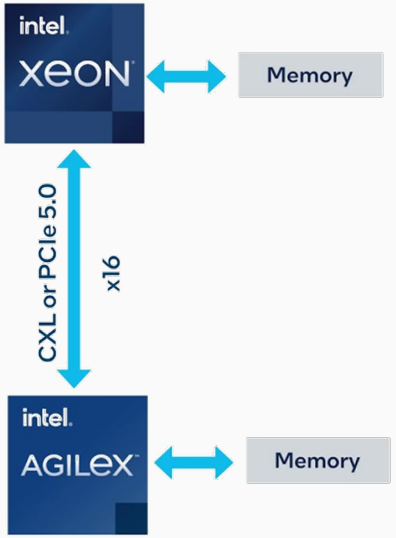
- What are the Challenges with Memory?
- Persistent Memory in HPC
- CXL Disaggregated Memory for HPC
- CXL as Persistent Memory
- **Physical Experiment Setup**
- Performance Evaluation
- Conclusions & Future Work

Physical Experimental Setup

- **CXL Prototype**
- Setup

Intel® FPGA Compute Express Link (CXL) IP

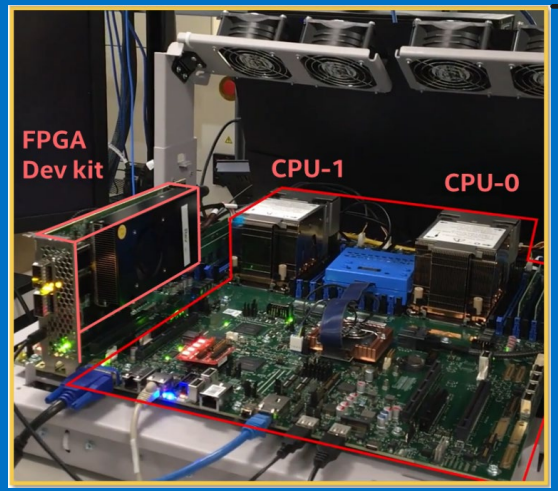




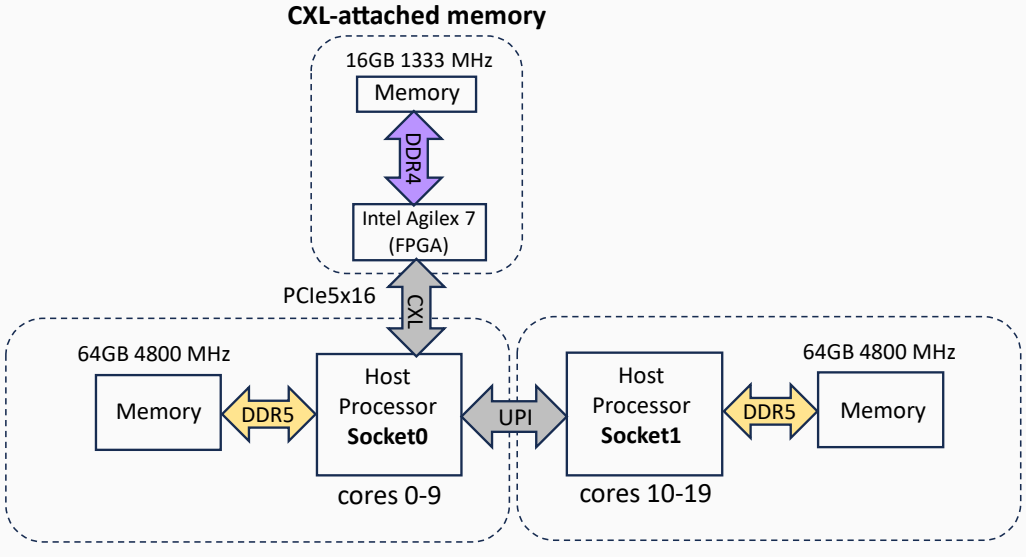
https://www.intel.com/content/www/us/en/products/details/fpga/intellectual-property/interface-protocols/cxl-ip.html

Physical Experimental Setup

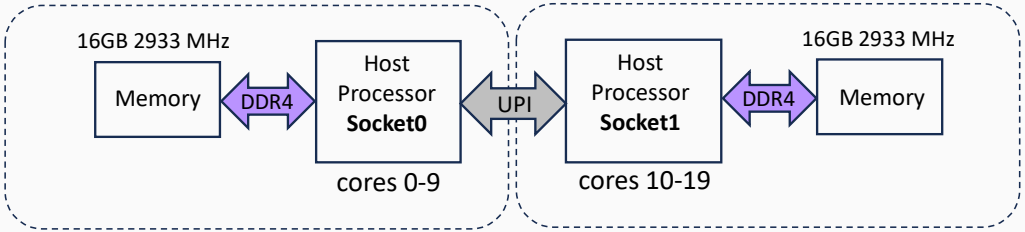
- CXL Prototype
- **Setup**



Setup #1 (with CXL) Intel $4th$ generation Xeon (Sapphire Rapids)



Setup #2 (without CXL) Intel Xeon Gold 5215



Talk Outline

- What are the Challenges with Memory?
- Persistent Memory in HPC
- CXL Disaggregated Memory for HPC
- CXL as Persistent Memory
- Physical Experiment Setup
- **Performance Evaluation**
- Conclusions & Future Work

Performance Evaluation

- **STREAM**
- STREAM-PMem
- Test configurations
- Results

STREAM Benchmark

Measure “Sustainable Memory Bandwidth” for four operations:

- COPY $x(i) = y(i)$
- SCALE $x(i) = a * y(i)$
- ADD $x(i) = y(i) + z(i)$
- TRIAD $x(i) = y(i) + a * z(i)$

- STREAM operations are parallelized with OpenMP threads.
- We set array size to 100MB. Memory consumption = 2.2GB.

https://github.com/jeffhammond/STREAM

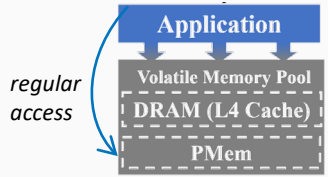
Performance Evaluation

- STREAM
- **STREAM-PMem**
- Test configurations
- Results

STREAM-PMem Benchmark

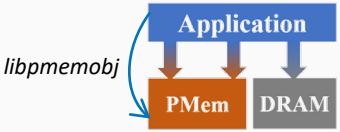
- Leverages PMDK *libpmemobj* to allocate arrays and operate on PMem (direct access).

```
1  #ifndef STREAM_TYPE
2  #define STREAM_TYPE double
3  #endif
4  static STREAM_TYPE a[STREAM_ARRAY_SIZE+OFFSET],
5                     b[STREAM_ARRAY_SIZE+OFFSET],
6                     c[STREAM_ARRAY_SIZE+OFFSET];
```



(Original STREAM benchmark code, initialization at lines 175-181)

```
1  PMEMobjpool *pop;
2  POBJ_LAYOUT_BEGIN(array);
3  POBJ_LAYOUT_TOID(array, double);
4  POBJ_LAYOUT_END(array); //Declearing the arrays
5  TOID(double) a, b, c;
6  void initiate() { //Initiating the arrays.
7     POBJ_ALLOC(pop, &a, double,
           (STREAM_ARRAY_SIZE+OFFSET)*sizeof(STREAM_TYPE),
           NULL, NULL); //Same for b and c.
```



(STREAM-PMem replaced code for initialization)

https://github.com/SveinGunnar/Master_Thesis_2020/tree/master

Performance Evaluation

- STREAM
- STREAM-PMem
- **Test configurations**
- Results

5 Test Groups:

Class 1: App-Direct [running STREAM-PMem]

- **1.a.** Local memory App-Direct access.
- **1.b.** Remote memory App-Direct access.
- **1.c.** Remote memory App-Direct access (thread affinity).

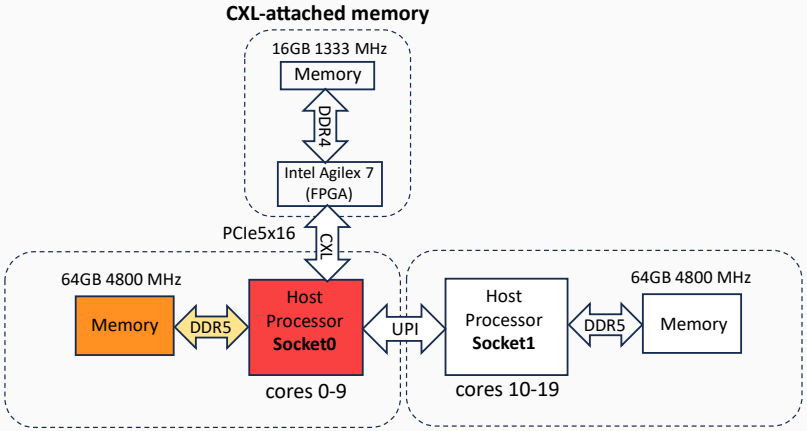
Class 2: Memory mode [running original STREAM]

- **2.a.** Remote CC-NUMA in Memory mode.
- **2.b.** Remote CC-NUMA in Memory mode (all cores).

Performance Evaluation

- STREAM
- STREAM-PMem
- **Test configurations**
- Results

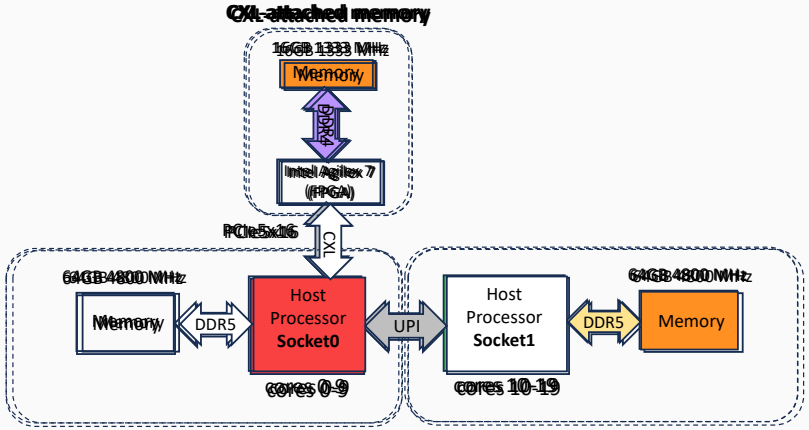
1.a. Local memory App-Direct access:



Performance Evaluation

- STREAM
- STREAM-PMem
- **Test configurations**
- Results

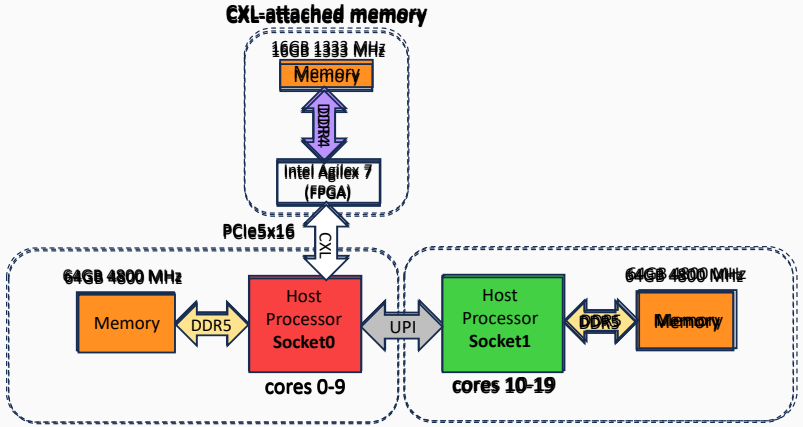
1.b. Remote memory App-Direct access:



Performance Evaluation

- STREAM
- STREAM-PMem
- **Test configurations**
- Results

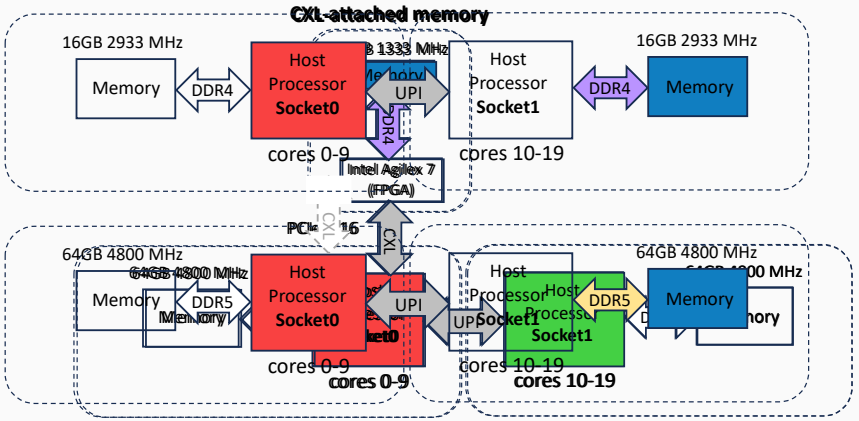
1.c. Remote memory App-Direct access (thread affinity):



Performance Evaluation

- STREAM
- STREAM-PMem
- **Test configurations**
- Results

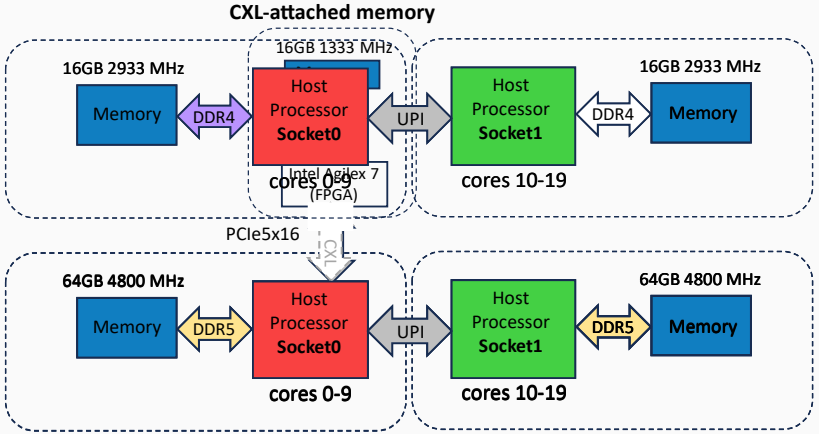
2.a. Remote CC-NUMA in Memory mode:



Performance Evaluation

- STREAM
- STREAM-PMem
- **Test configurations**
- Results

2.b. Remote CC-NUMA in Memory mode (all cores):



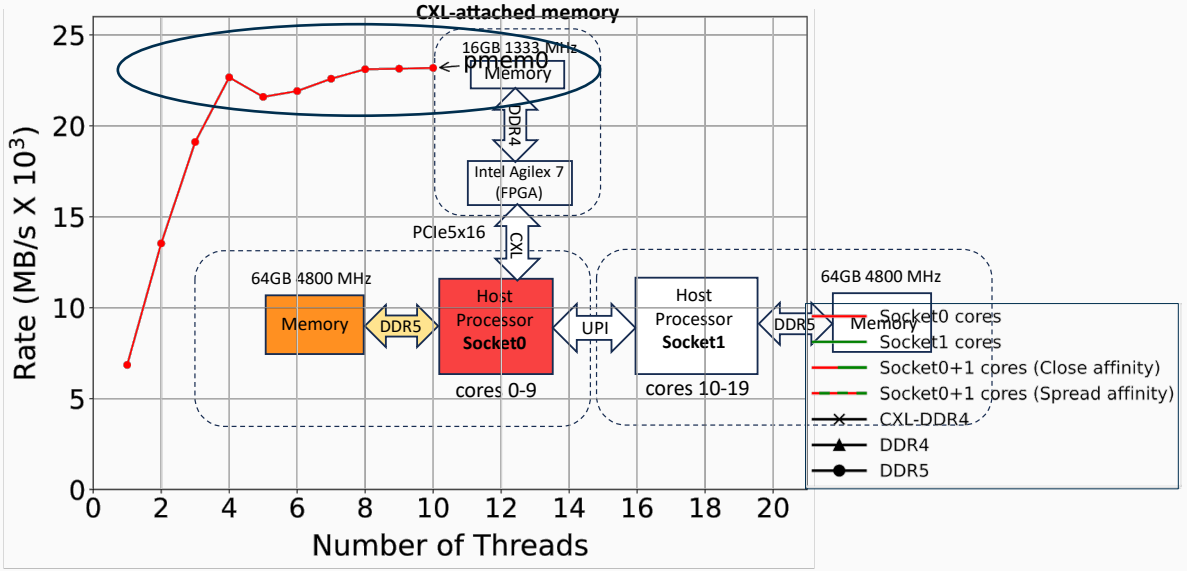
Performance Evaluation

- STREAM
- STREAM-PMem
- Test configurations
- **Results**

Observations:

1) **App-Direct** access using PMDK to the **local DDR5** memory is saturated around 23 GB/s.

1.a. Local memory App-Direct access: **TRIAD**



Performance Evaluation

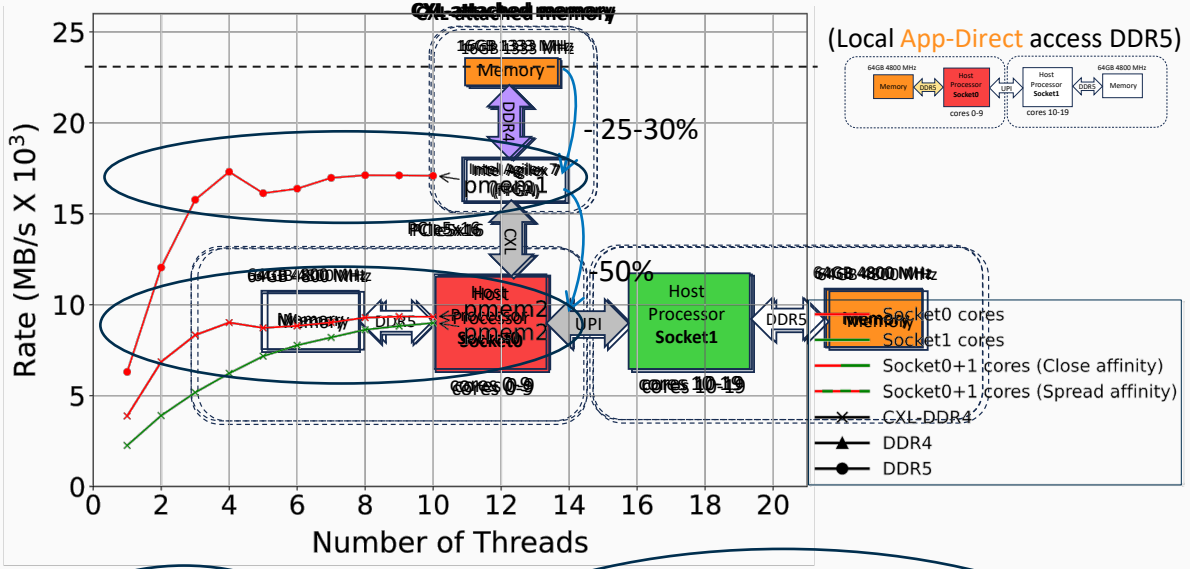
- STREAM
- STREAM-PMem
- Test configurations
- **Results**

Observations:

1) **App-Direct** access to the emulated remote PMem (**DDR5** on the alternate socket) results in a **decrease of 25-30%** (~6 GB/s) of performance.

2) App-Direct access to remote **CXL memory (DDR4)** experiences **50% decrease** in performance in comparison to the emulated PMem on alternate socket DDR5.

1.b. Remote memory App-Direct access: **TRIAD**



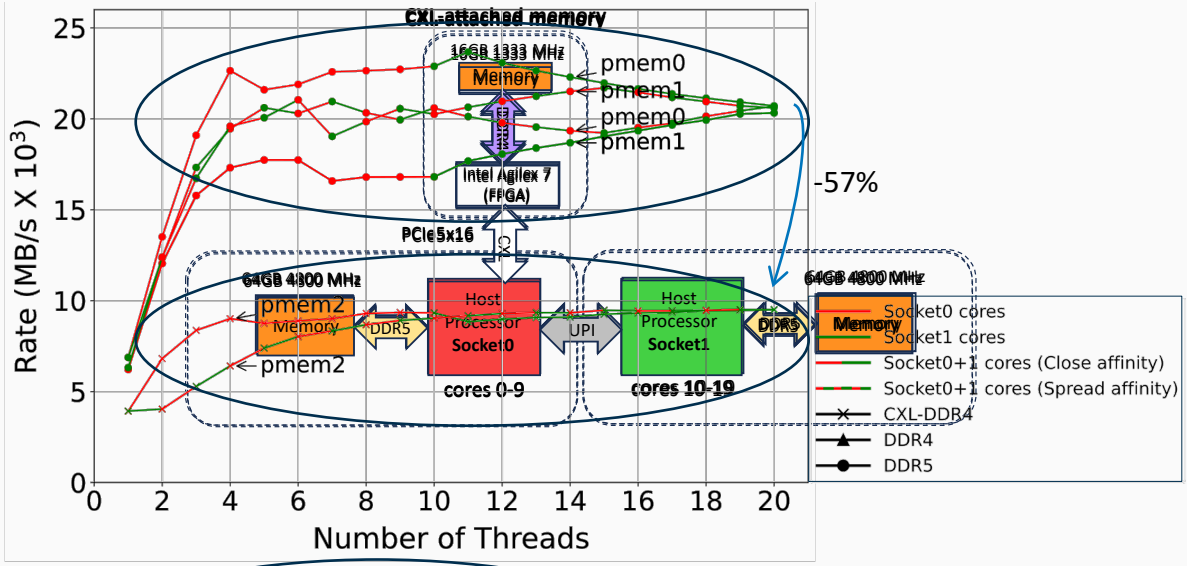
Performance Evaluation

- STREAM
- STREAM-PMem
- Test configurations
- **Results**

Observations:

1) when both sockets are operating with the entire core count, the results converge for on-node DDR5 and remote CXL memory, separately. Notably, accessing **remote CXL memory (DDR4)** leads to 57% observed degradation compared to **on-node DDR5**.

1.c. Remote memory App-Direct access (thread aff): **TRIAD**



Performance Evaluation

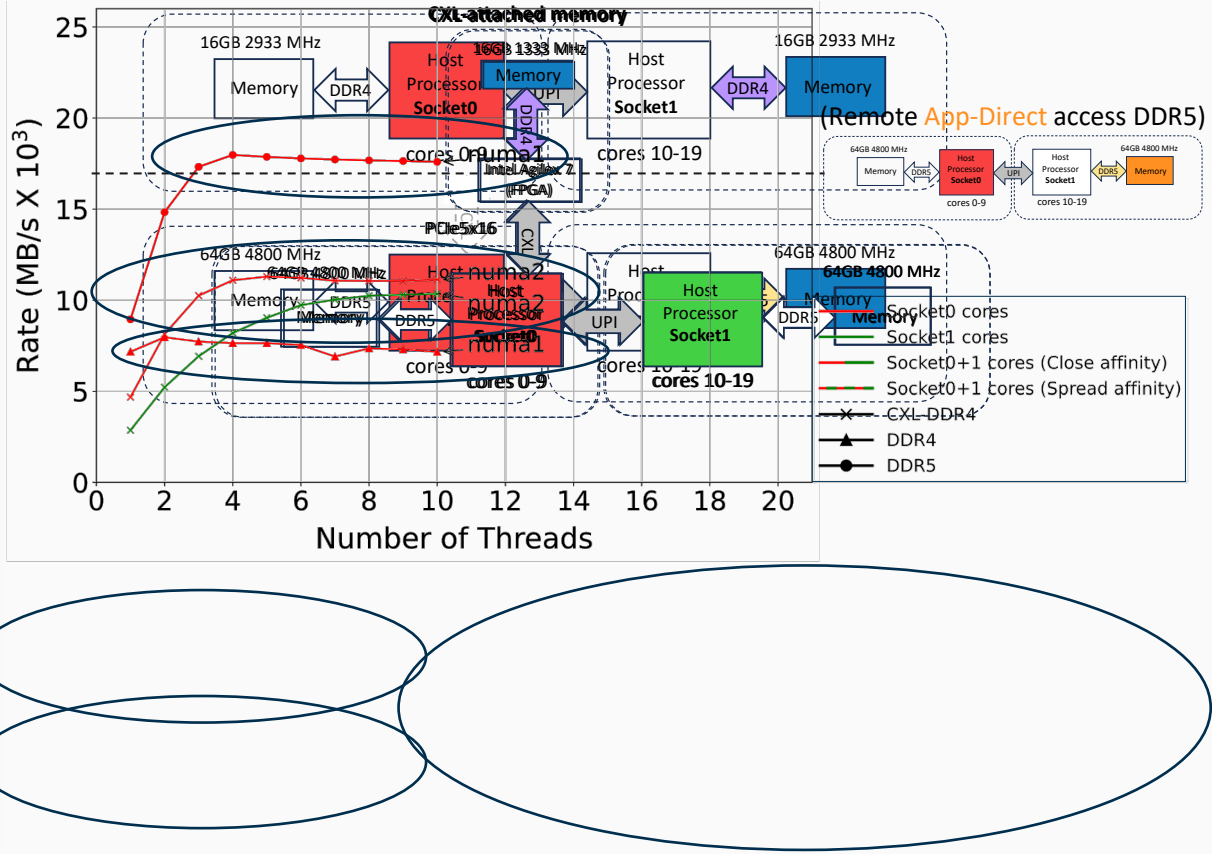
- STREAM
- STREAM-PMem
- Test configurations
- **Results**

Observations:

1) **PMDK** App-Direct **overhead** is 10-15%.

2) Accessing **CXL-attached DDR4** on Sapphire Rapids setup (11 GB/s) is faster than accessing **DDR4 on alternate socket** on Xeon Gold setup (7 GB/s).

2.a. Remote CC-NUMA in Memory mode: **TRIAD**



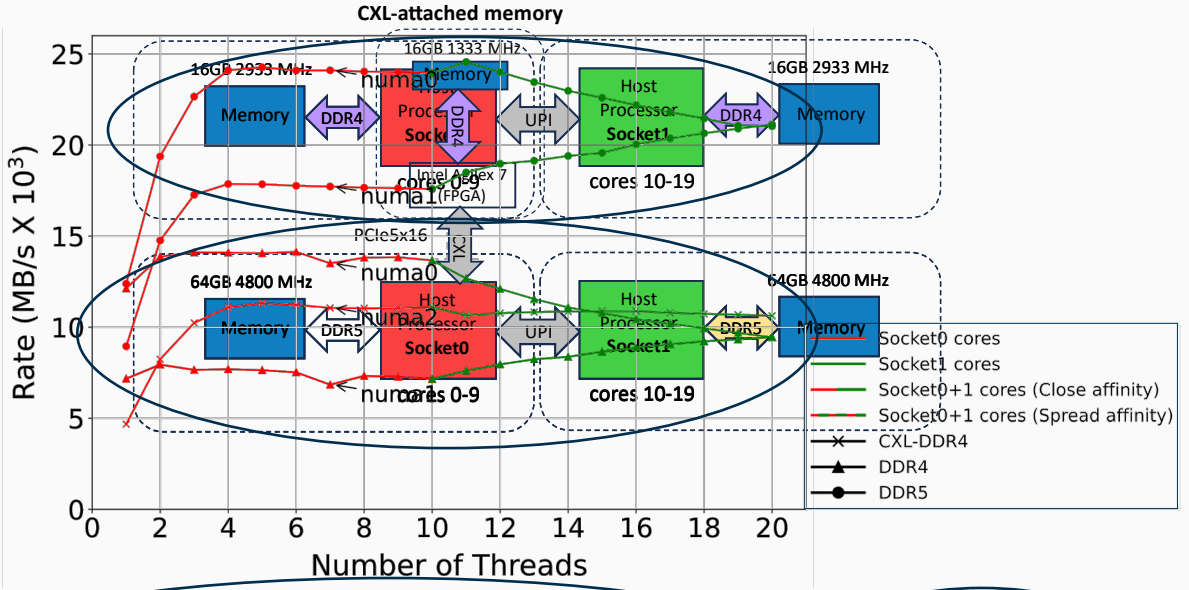
Performance Evaluation

- STREAM
- STREAM-PMem
- Test configurations
- **Results**

Observations:

1) Accessing **on-node DDR4** using all core count converges to the same result as accessing **DDR4 CXL memory** (slightly advantage to the latter).

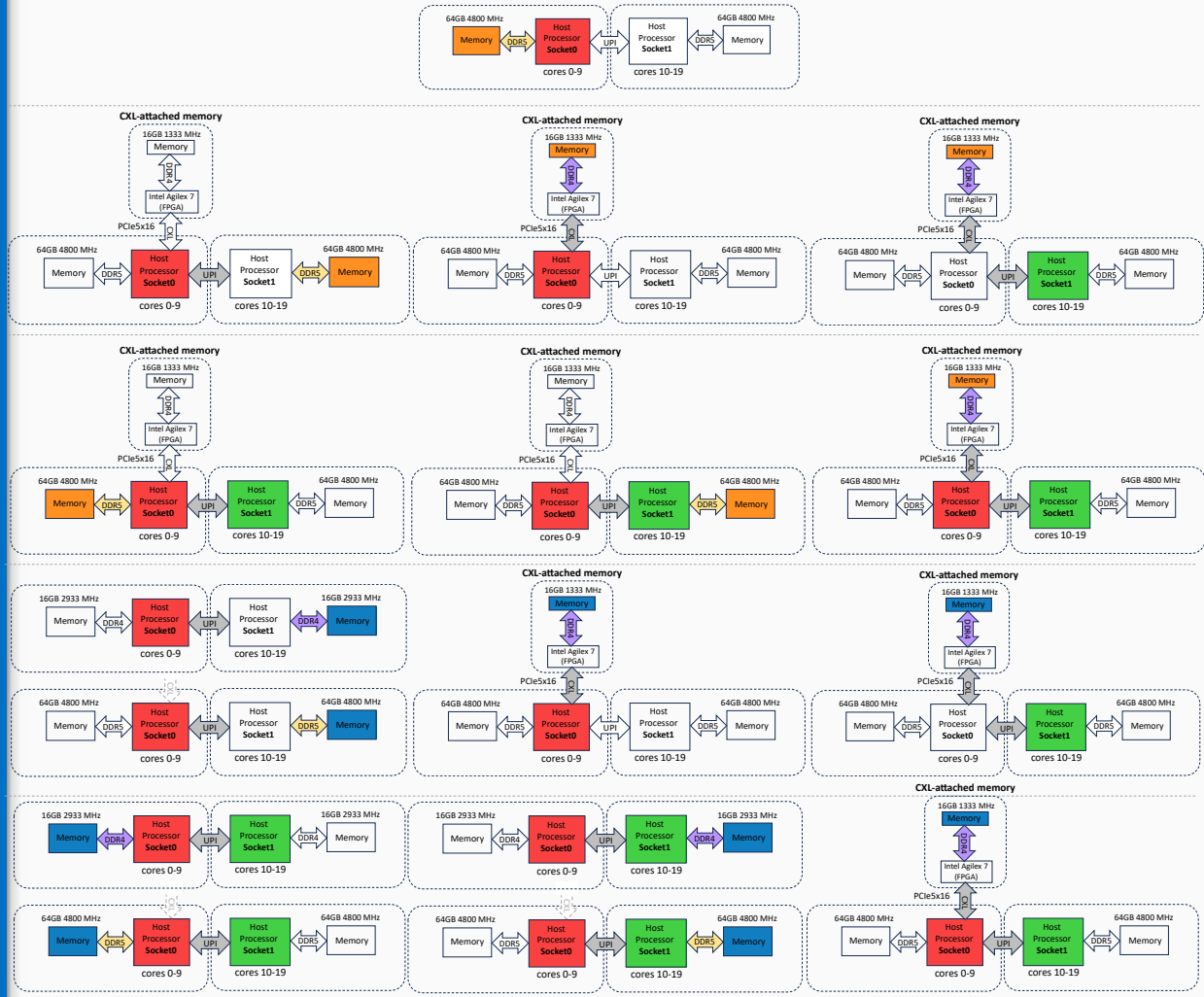
2.b. Remote CC-NUMA in Memory mode (all cores): **TRIAD**



Performance Evaluation

- STREAM
- STREAM-PMem
- Test configurations
- Results

Evaluating various access patterns and configurations for Memory and App-Direct modes



Talk Outline

- What are the Challenges with Memory?
- Persistent Memory in HPC
- CXL Disaggregated Memory for HPC
- CXL as Persistent Memory
- Physical Experiment Setup
- Performance Evaluation
- **Conclusions & Future Work**

Conclusions & Future Work

Conclusions

- A **CXL prototype** on an FPGA card was implemented, showcasing compliance with CXL 1.1/2.0 standards.
- **CXL memory** can effectively function as **persistent memory** in disaggregated HPC systems.
- CXL memory surpasses benchmarks for Optane DCPMM in terms of bandwidth performance.
- CXL demonstrates **modest decrease in performance** compared to local memories.
- The transition from PMem to CXL was **seamless** (both in **Memory** and **App-Direct** modes).

Conclusions & Future Work

Future work

- Scalability and Performance Optimization.
- Hybrid Architectures.
- Real-World Applications.
- Fault Tolerance and Reliability.

Thanks!

Contact us:

galoren@cs.technion.ac.il