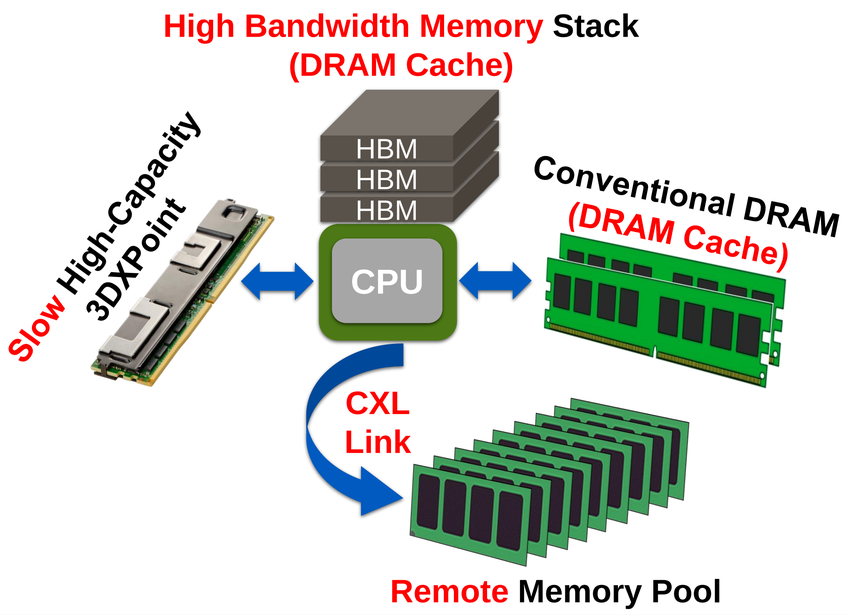
Toward High-Fidelity Heterogeneous Memory System Modeling in gem5

Maryam Babaie, Ayaz Akram, Jason Lowe-Power
DArchR Research Group, CS Department, University of California Davis

UC DAVIS
UNIVERSITY OF CALIFORNIA

Motivation

- HPC systems rely on **heterogeneous** memories.
  - *Intel's Knights Landing, Cascade Lake, Sapphire Rapids*
- In these systems, fast memories can be used as **DRAM cache** to slow memories.
- **Disaggregated** memory resources also will use local DRAM as a cache to a remote memory.
- As these new systems emerge, we need to rethink the memory managements.
- However, there is not an accurate model in the research community.



We extend gem5 with a heterogeneous memory system model for design space exploration of future HPC systems.

We add support for:

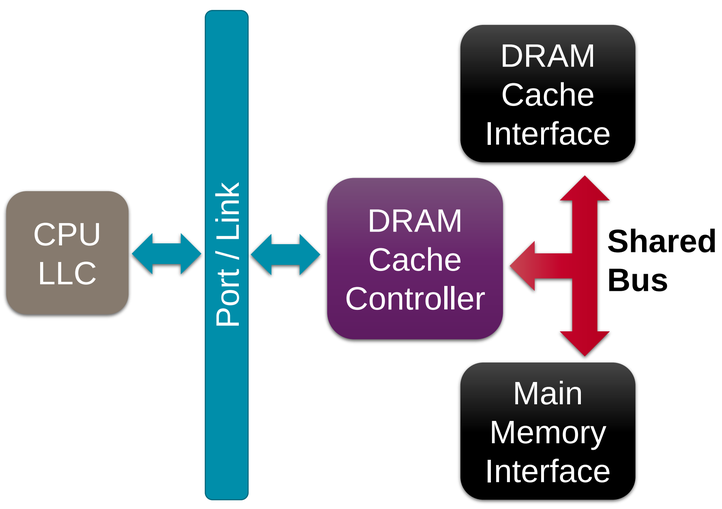
1. **DRAM Cache**
2. **HBM interface and controller**
3. **Modular memory controller design**

gem5's DRAM Cache Support

Goal: Enabling cycle-level analysis of heterogeneous and disaggregated systems.

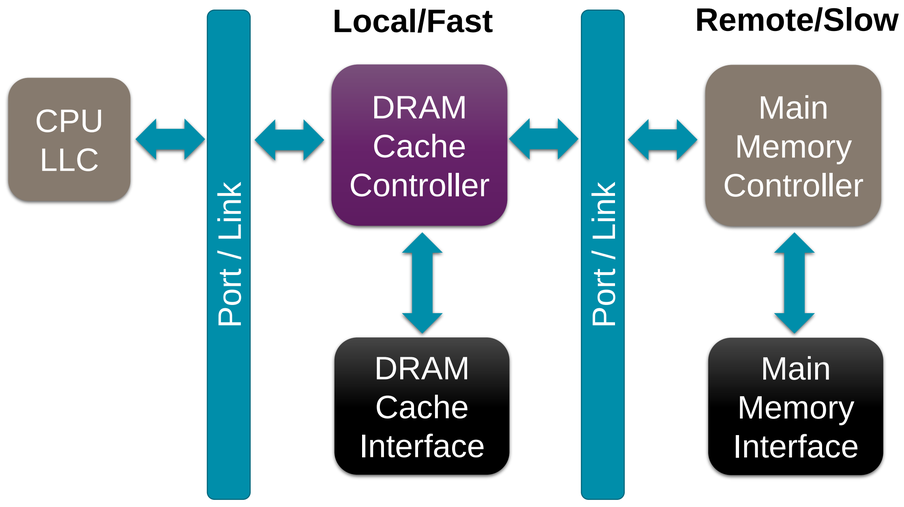
A. Unified Cache/Memory Controller (UDCC)

- Tightly-coupled DRAM cache and main memory
- Connection: shared bus
- Models *Intel's Cascade Lake*



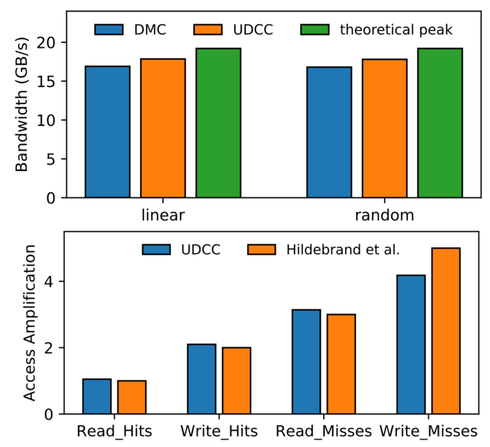
B. Disaggregated DRAM Cache Controller

- Flexible combination of DRAM cache and main memory
- Connection: configurable link



Verification

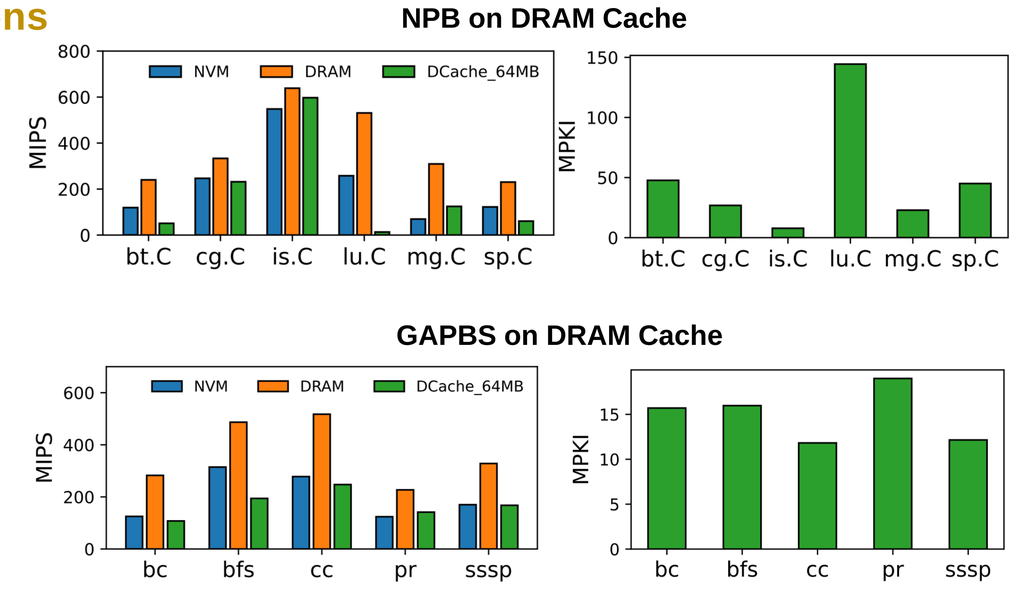
- Top: Peak BW of the DRAM cache VS gem5's default memory controller (DMC).
- Bottom: Access amplification of our model VS the real hardware [1].
  - Write_Misses Case: in real hardware write-fills and data-writes are not merged, where in our model they are.



Performance of HPC Applications

- NPB and GAPBS
  - Million instructions per second (MIPS) and the DRAM cache misses per thousand instructions (MPKI) values are reported.
- UDCC was configured as Cascade Lake.

On DRAM cache, most workloads performed worse than DRAM and NVM main memory, due to high miss rate caused by the rigid cache architecture [1].



Link Latency Case Study

- Using an HBM cache, backed by a (i) DDR4 and (ii) NVM main memory through a link, for a read-only miss-clean traffic.

On lower link latency, far NVM performs better than far DDR4. For higher link latency, NVM performs closely to the DDR4.

Far Mem.	Link Latency (µs)	Tot BW (GB/s)	Avg. Resp. Time (µs)
DDR4	**No Link**	**6.31**	**1.484**
	0.2	5.86	1.599
	1	5.61	1.833
	2.5	5.20	2.985
	5	3.04	5.346
NVM	**No Link**	**6.03**	**2.489**
	0.2	6.03	2.487
	1	6.03	2.491
	2.5	4.86	3.269
	5	2.99	5.460

gem5's HBM Stack Support

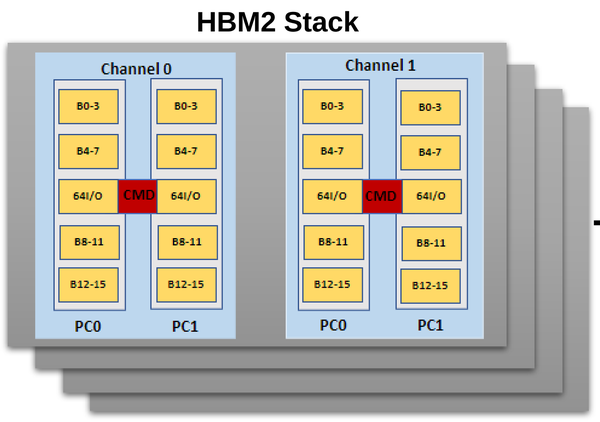
Our contributions to gem5 to enable accurate HBM2 model:

1 HBM Controller

- Controls single physical channel (two pseudo channels, which share a command bus).
- Can perform parallel data accesses if the command BW allows.

2 HBM2 Interface

- Configured a **validated** HBM2 memory interface using JEDEC specifications and other tools.
- Also enables asynchronous read/write latencies as needed in HBM2.



HBM2 Model Evaluation

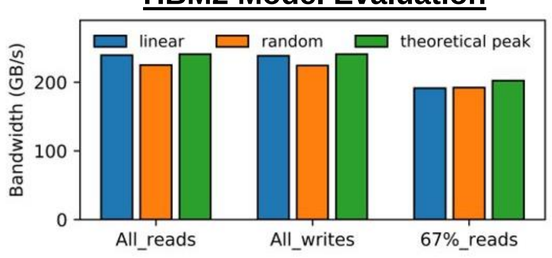


Figure. *We compared the bandwidth of gem5 HBM2 stack (consisting of 8 physical and 16 pseudo channels) with the theoretical peak bandwidth possible for two different traffic patterns (linear, random) for three different read-write combinations (all reads, all writes, and 67% reads) using gem5's traffic generator. The results show that the gem5's observed bandwidth numbers match theoretical peak.*

HBM2 stack in gem5 standard library

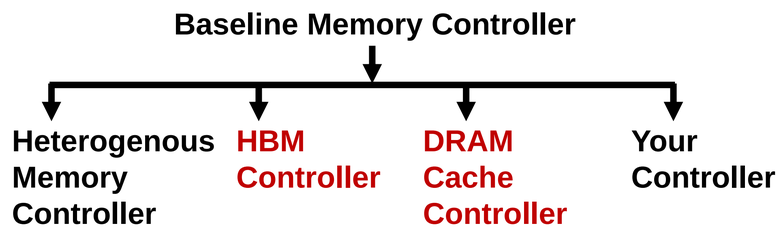
```
def HBM2Stack(
    size: Optional[str] = None,
) -> AbstractMemorySystem:
    if not size:
        size = "4GiB"
    return ChanneledMemory(
        HBM_2000_4H_1x64,
        8,
        128,
        size=size,
        is_hbm=True)
```

Using HBM2 stack component

```
memory = HBM2Stack("4GiB")
```

gem5's New Memory Controllers

We refactored gem5's memory controller to extend modularity, as follows:



Conclusion

- In this work, we introduced heterogenous memory modeling support in gem5.
- The models we described in this work enable research opportunities for next generation of heterogenous and disaggregated HPC systems.

Reference

[1] M. Hildebrand, J. T. Angeles, J. Lowe-Power, and V. Akella, "A case against hardware managed dram caches for nvram based systems," in 2021 ISPASS.