SDM: Sharing-enabled Disaggregated Memory System with Cache Coherent Compute Express Link

Hyokeun Lee[+] **Kwanseok Choi**[*] Hyuk-Jae Lee[*] Jaewoong Sim[*]

[+]North Carolina State University [*]Seoul National University

PACT-32 | October 2023

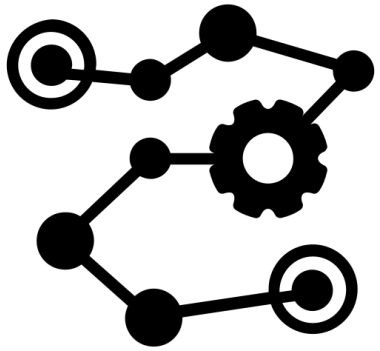
Outline

- **Introduction**
- **Motivation**
- **SDM: Sharing-enabled Disaggregated Memory System**
  - CXL-compatible Designs
- **Evaluation**
- **Conclusion**

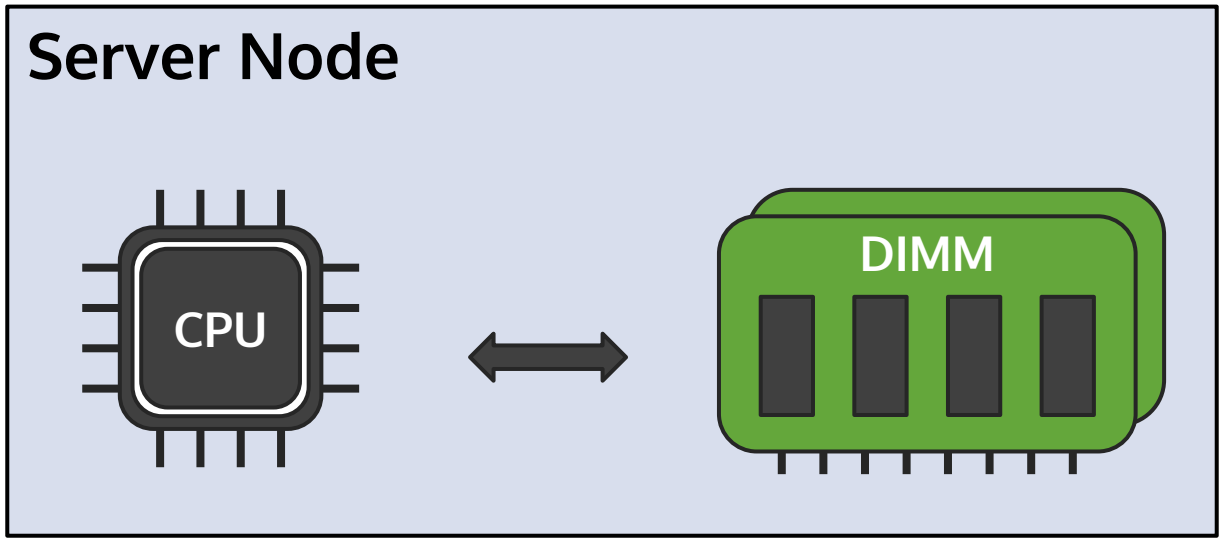
Outline

- Introduction
- Motivation
- SDM: Sharing-enabled Disaggregated Memory System
  - CXL-compatible Designs
- Evaluation
- Conclusion

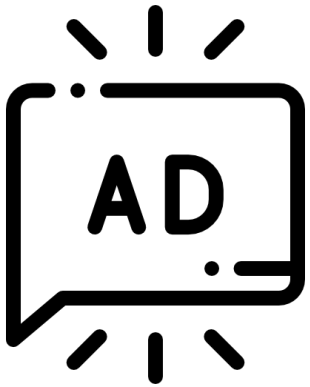
Demand for Large Memory Capacity



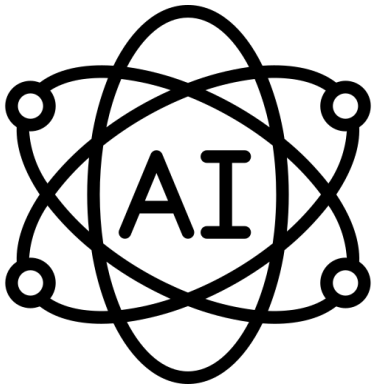
Social Networks



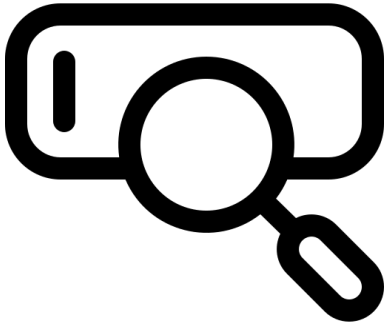
Data Analytics



Advertisement

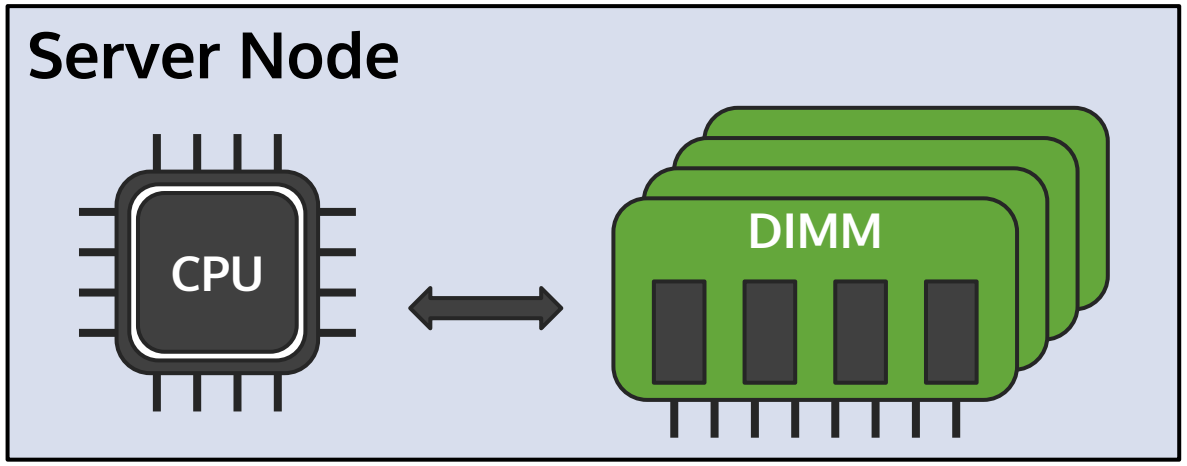


AI



Web Search

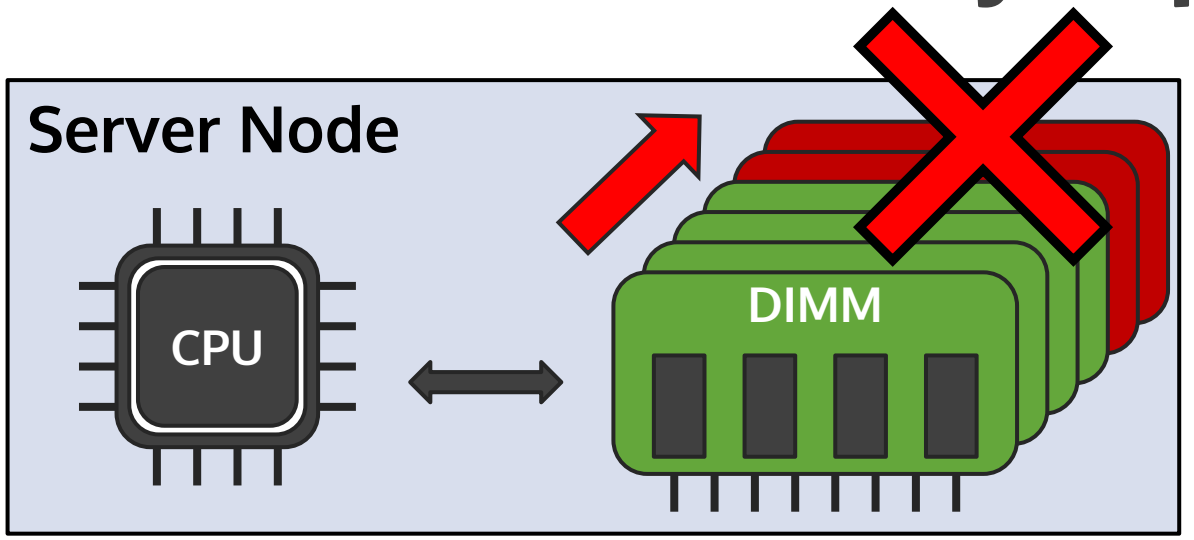
How to Scale Memory Capacity



More DIMMs within the node?

- Limited # Pins

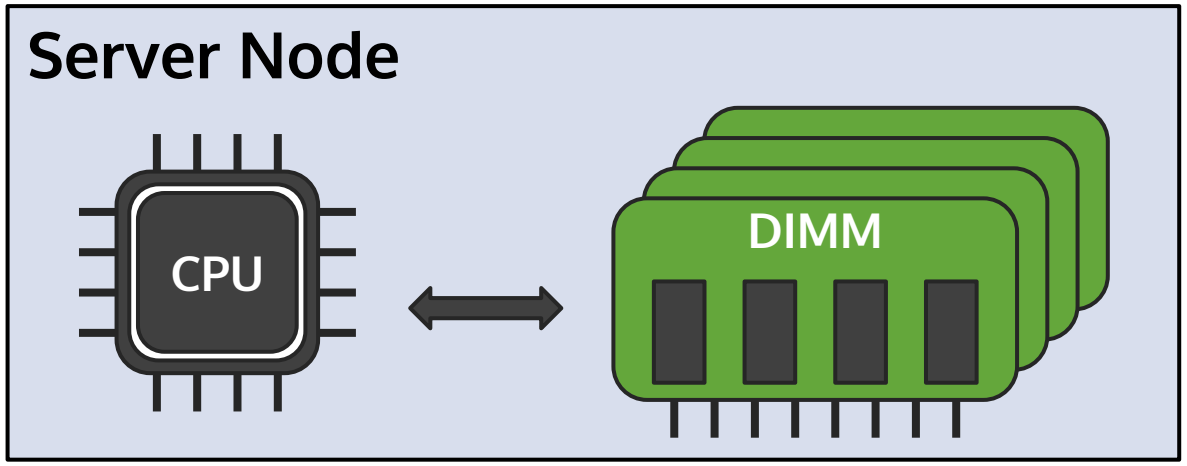
How to Scale Memory Capacity



More DIMMs within the node?

- Limited # Pins

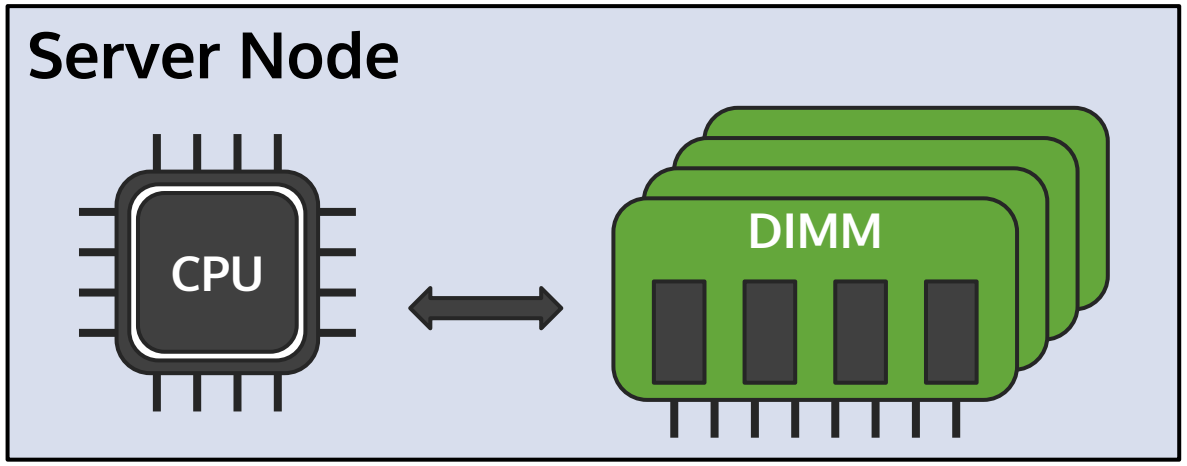
How to Scale Memory Capacity



~~More DIMMs within the node?~~

- ~~Limited # Pins~~

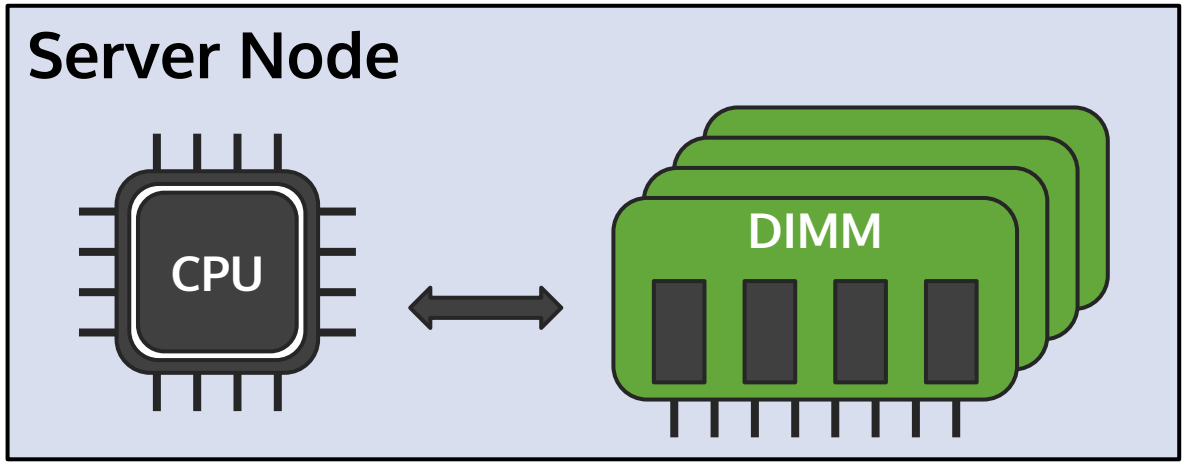
How to Scale Memory Capacity



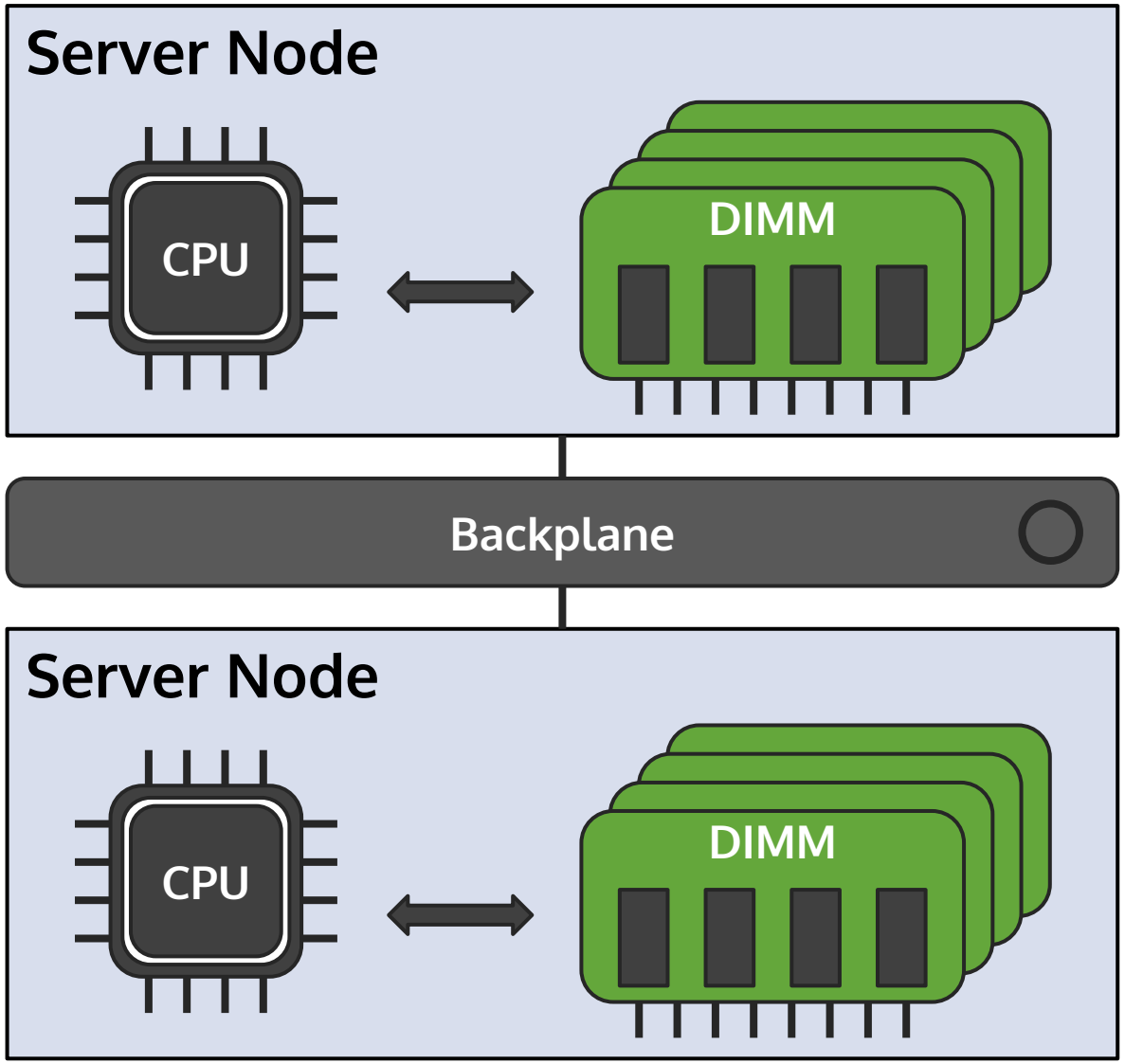
~~More DIMMs within the node?~~

- ~~Limited # Pins~~

Integrate more server nodes?



How to Scale Memory Capacity

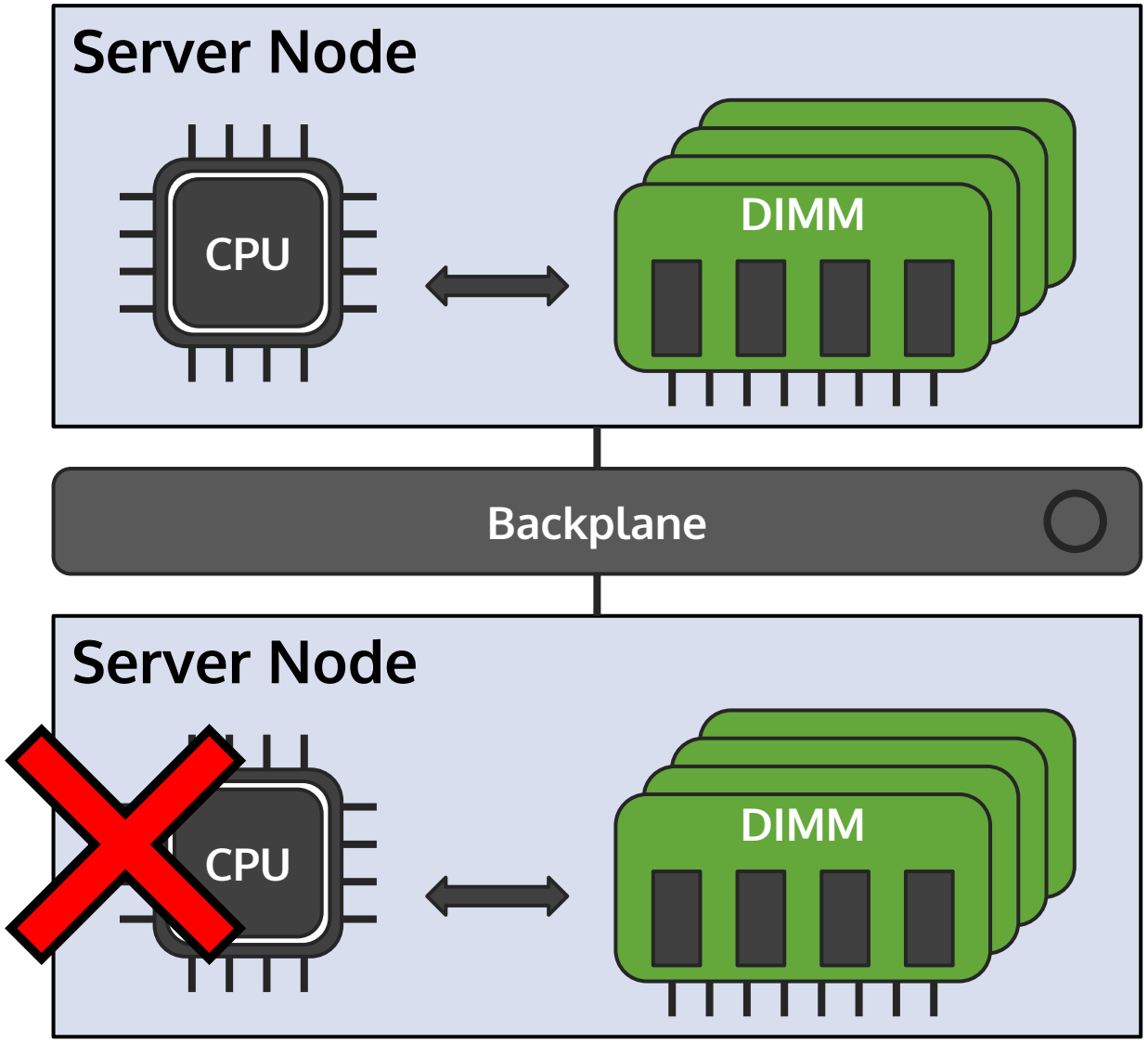


~~More DIMMs within the node?~~

- ~~Limited # Pins~~

Integrate more server nodes?

How to Scale Memory Capacity



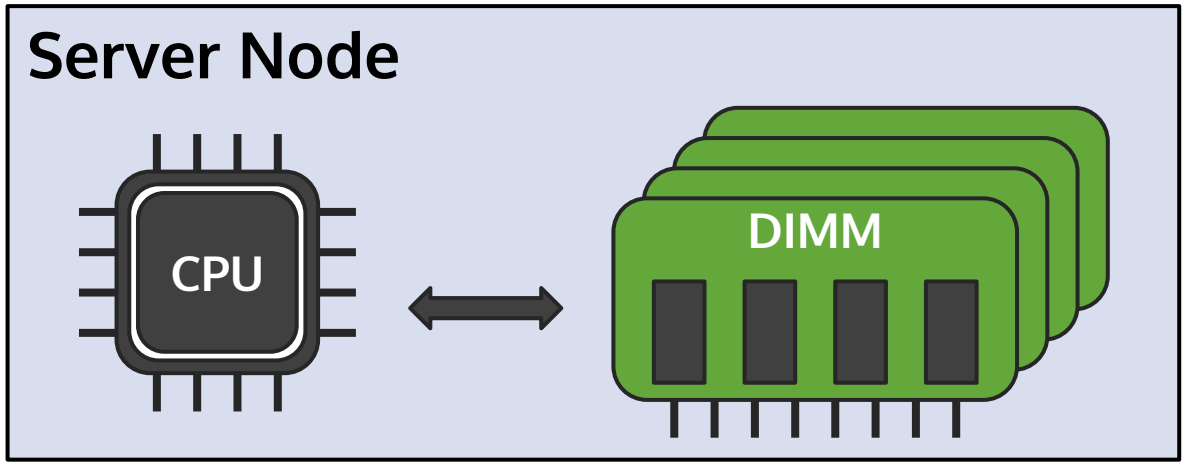
~~More DIMMs within the node?~~

- ~~Limited # Pins~~

Integrate more server nodes?

- Underutilized Cores

How to Scale Memory Capacity



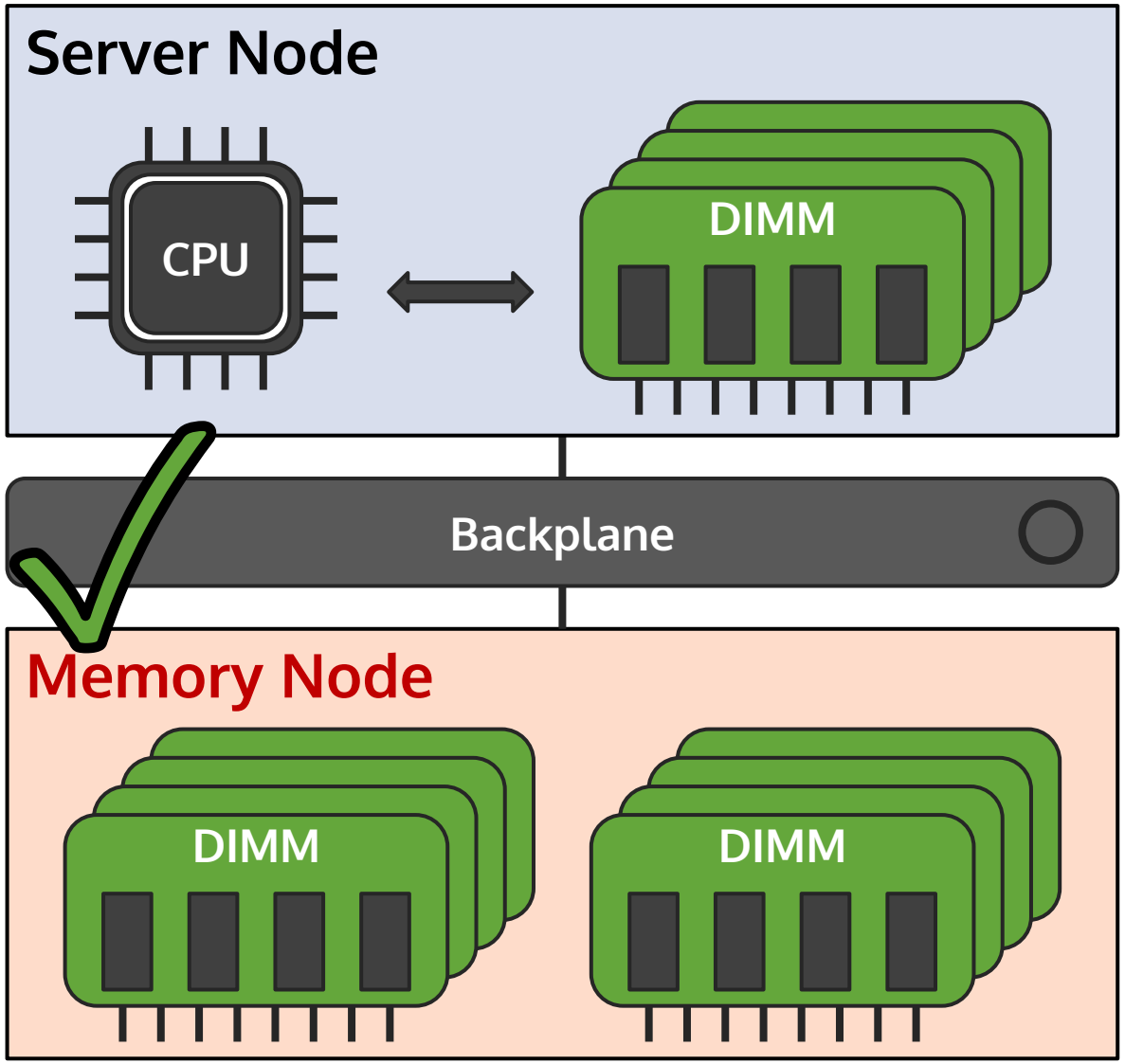
~~More DIMMs within the node?~~

- ~~Limited # Pins~~

~~Integrate more server nodes?~~

- ~~Underutilized Cores~~

How to Scale Memory Capacity



~~More DIMMs within the node?~~

- ~~Limited # Pins~~

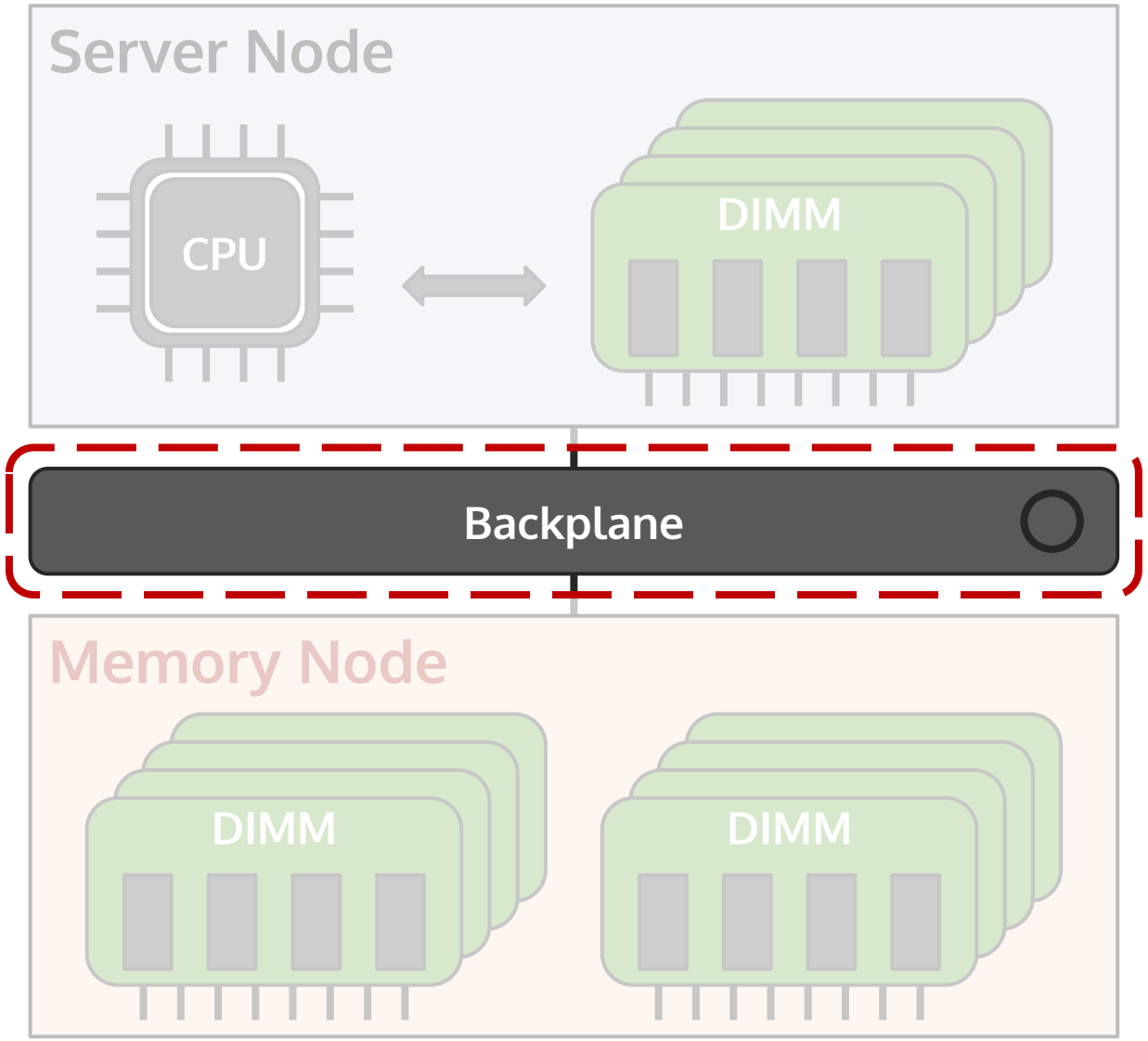
~~Integrate more server nodes?~~

- ~~Underutilized Cores~~

Solution:

Disaggregated Memory!

How to Scale Memory Capacity



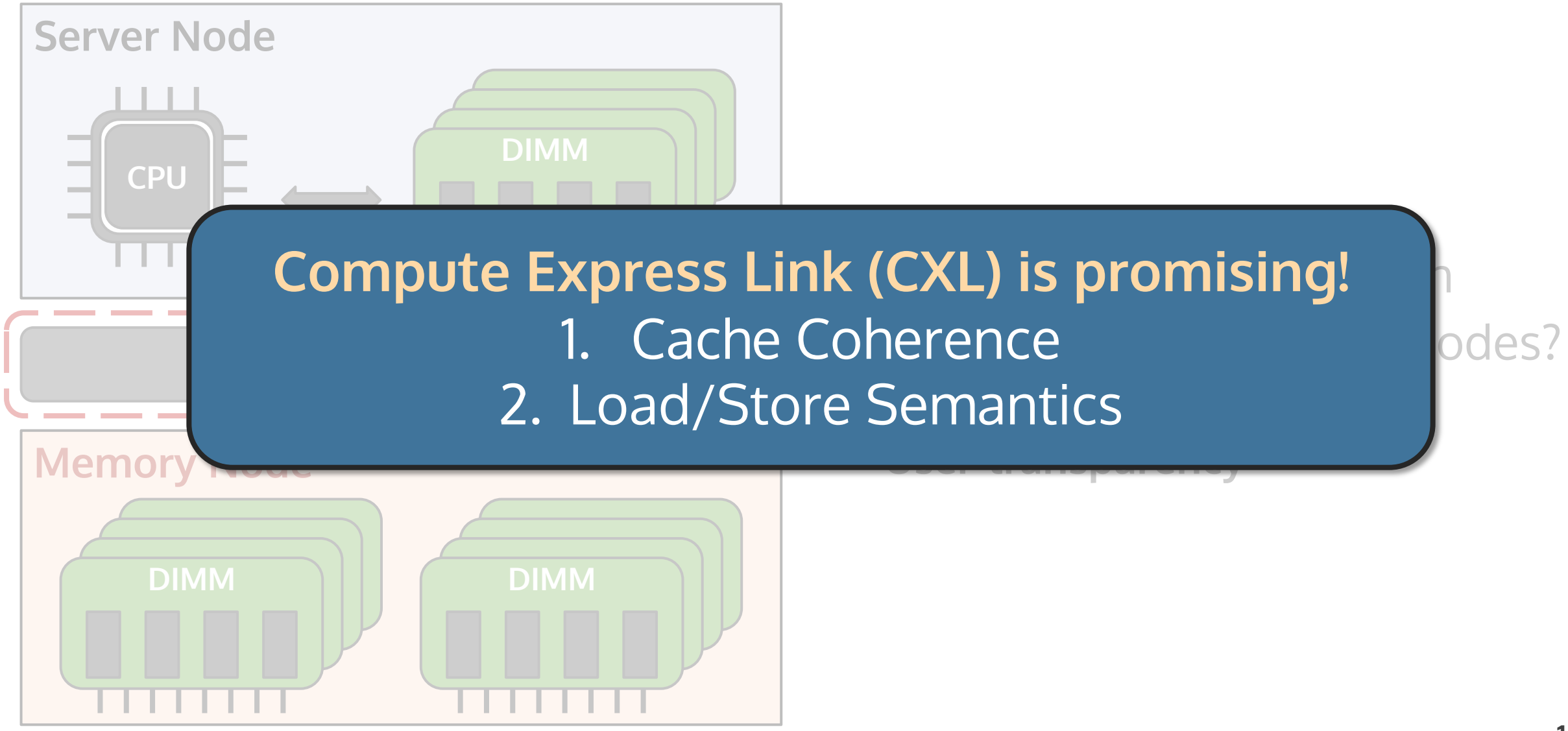
Q: **Communication** between server nodes and memory nodes?

- **Low latency**
- **User-transparency**

How to Scale Memory Capacity

	RDMA	CXL
Low-latency	✗ Software-stack Overhead	✓ Cache Coherence
User-transparency	✗ RDMA API	✓ Load/Store Semantics

How to Scale Memory Capacity



Compute Express Link (CXL) is promising!

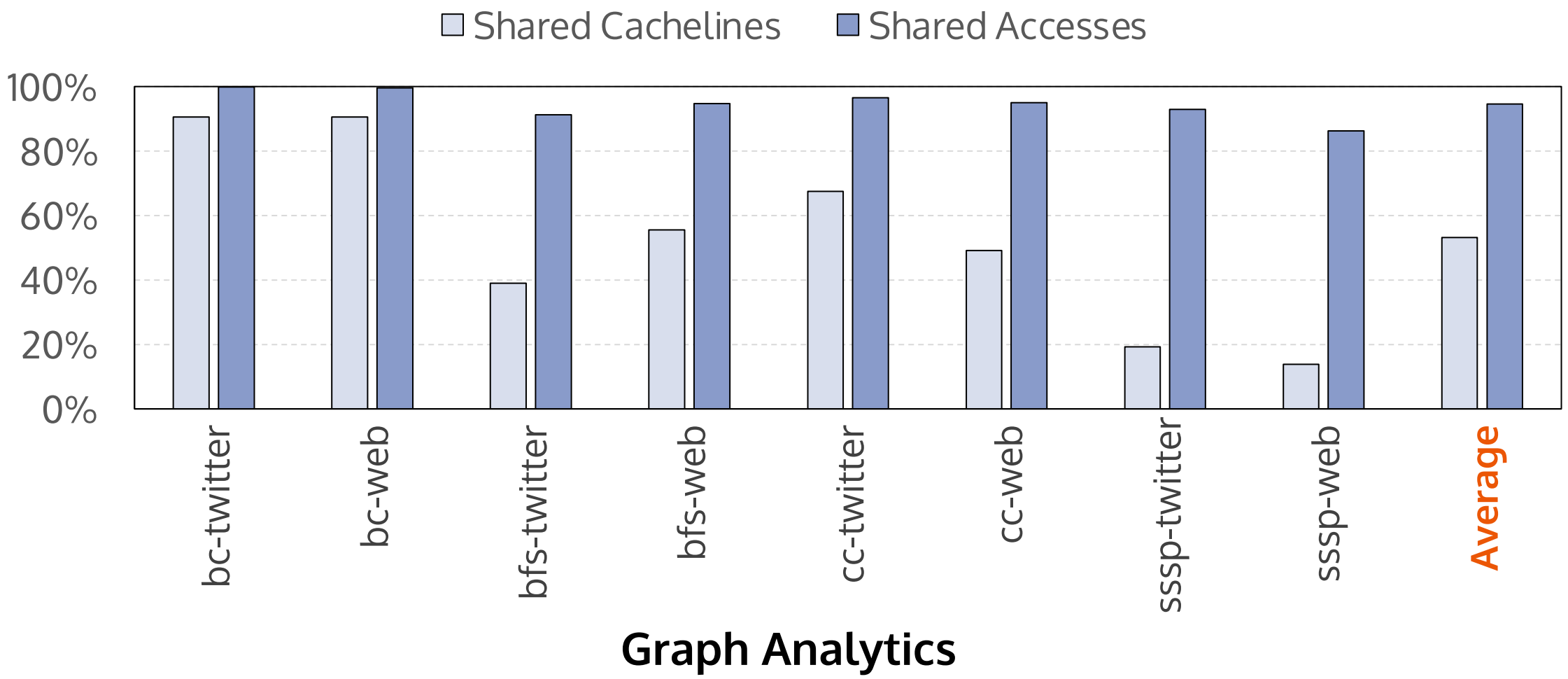
1. Cache Coherence
2. Load/Store Semantics

Outline

- Introduction
- Motivation
- SDM: Sharing-enabled Disaggregated Memory System
  - CXL-compatible Designs
- Evaluation
- Conclusion

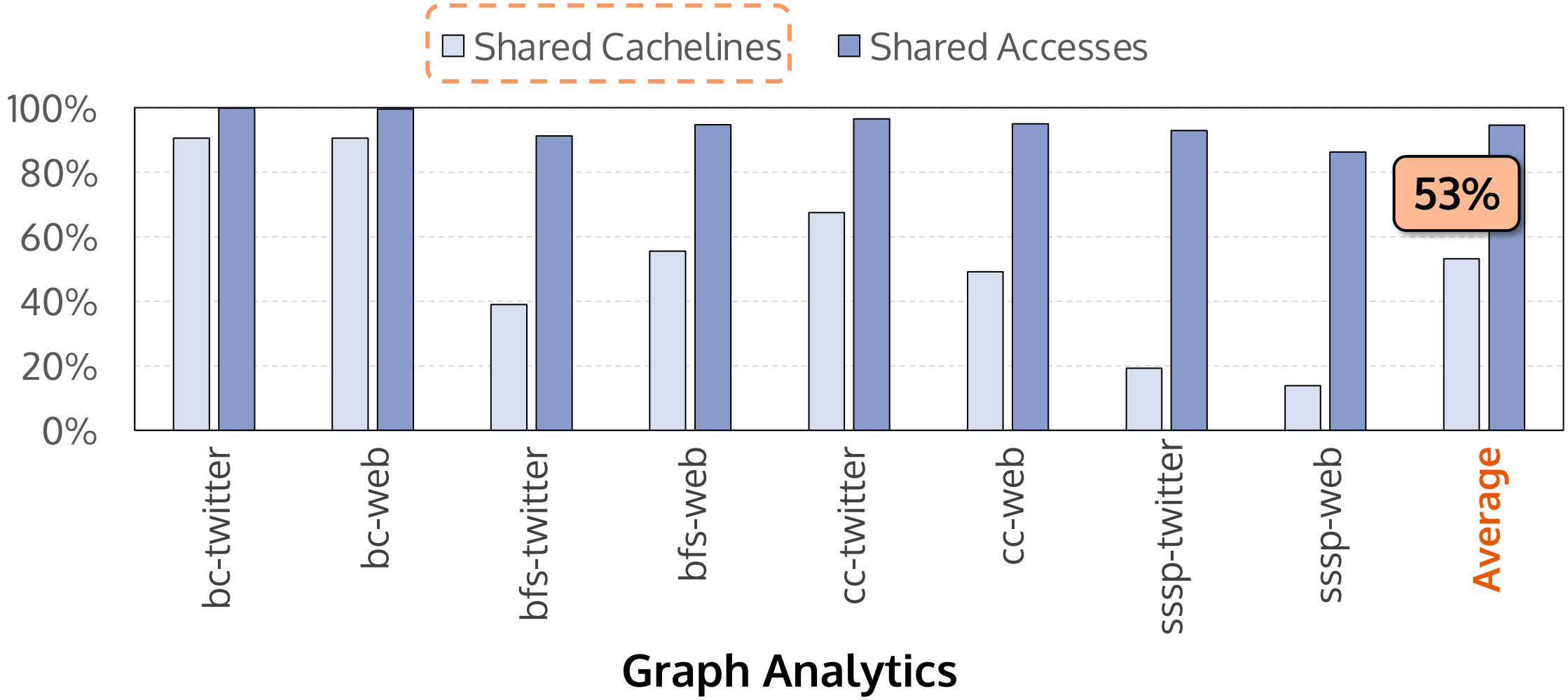
Multi-Host Data Sharing Opportunity

Q: How many cachelines are shared across hosts?



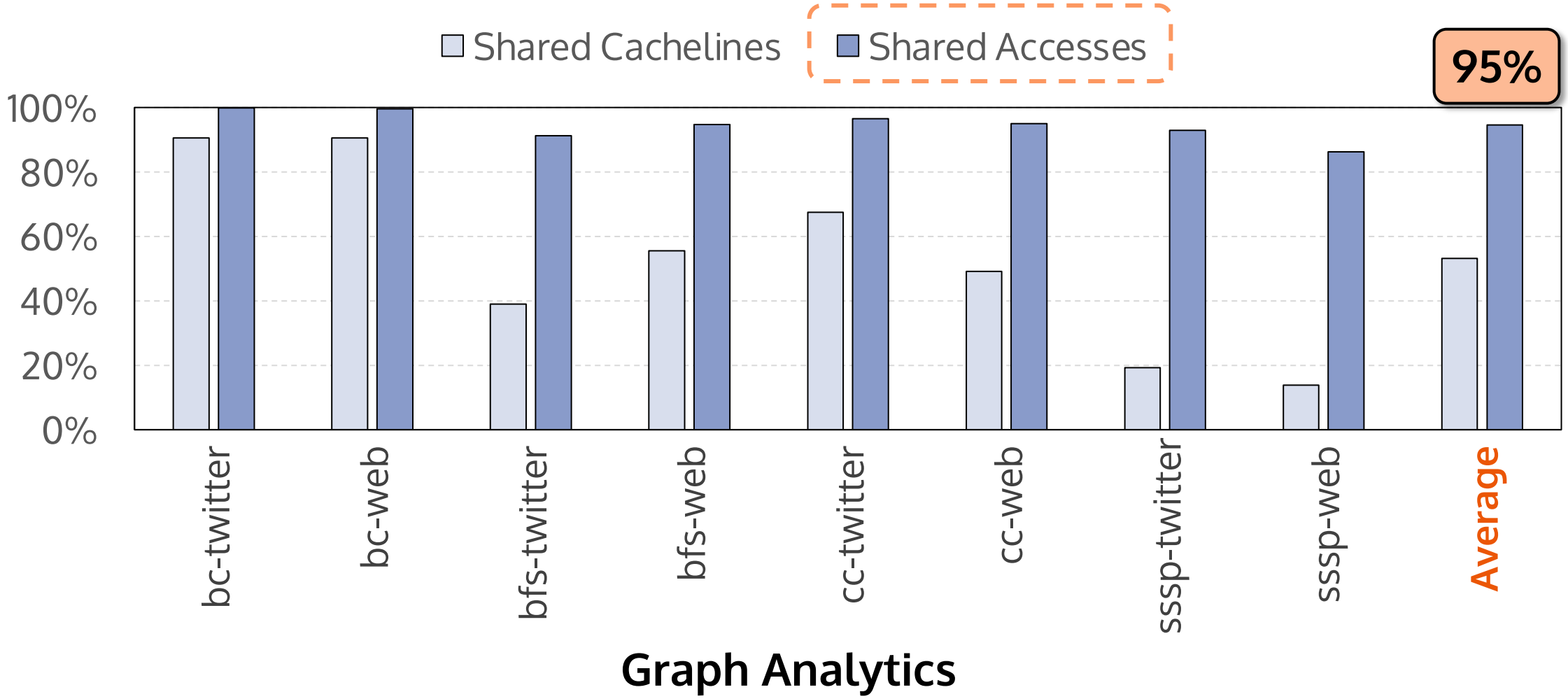
Multi-Host Data Sharing Opportunity

Q: How many cachelines are shared across hosts?



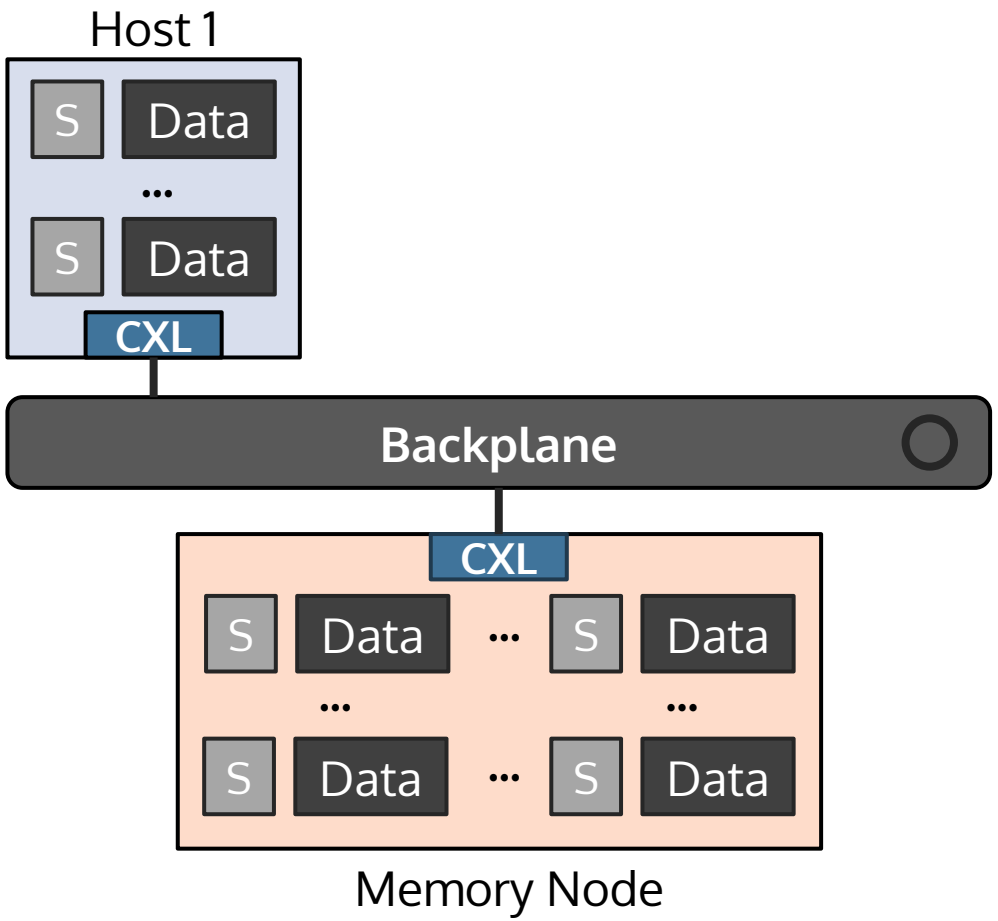
Multi-Host Data Sharing Opportunity

Q: How many times are the shared cachelines accessed?



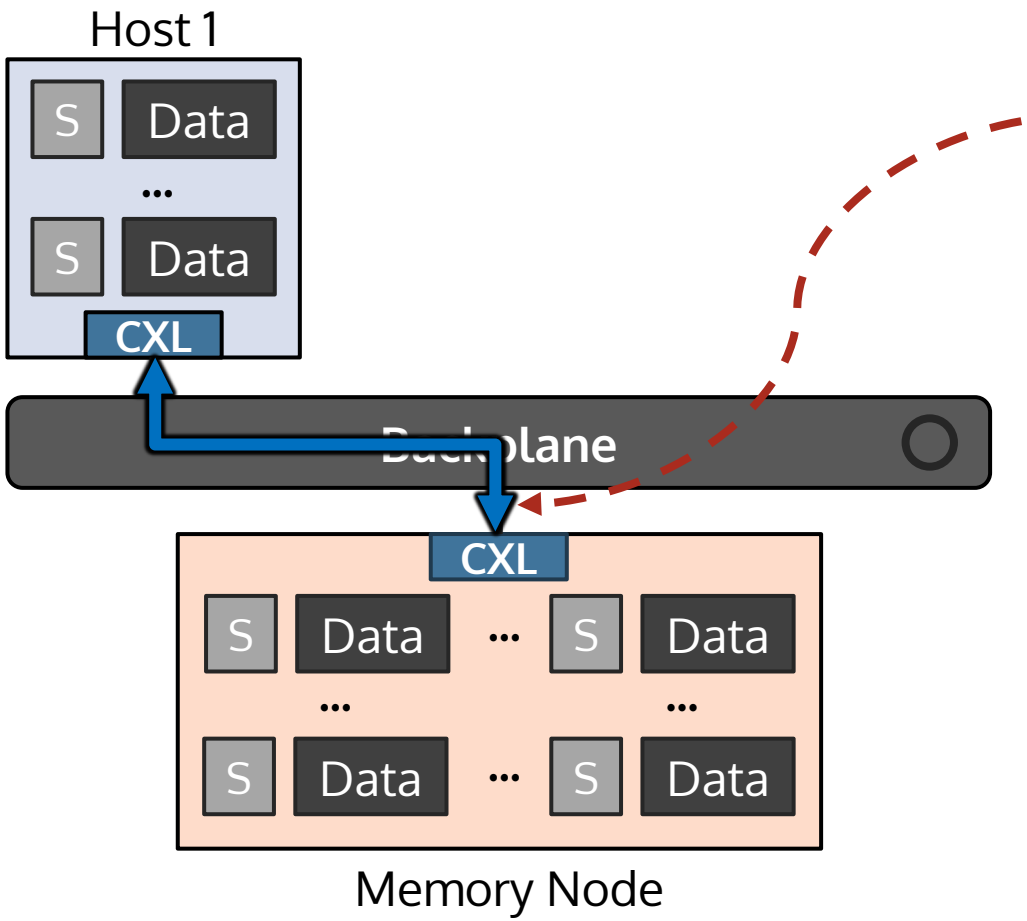
Enabling Multi-host Data Sharing

Goal 1: How can hosts share their coherence states?



Enabling Multi-host Data Sharing

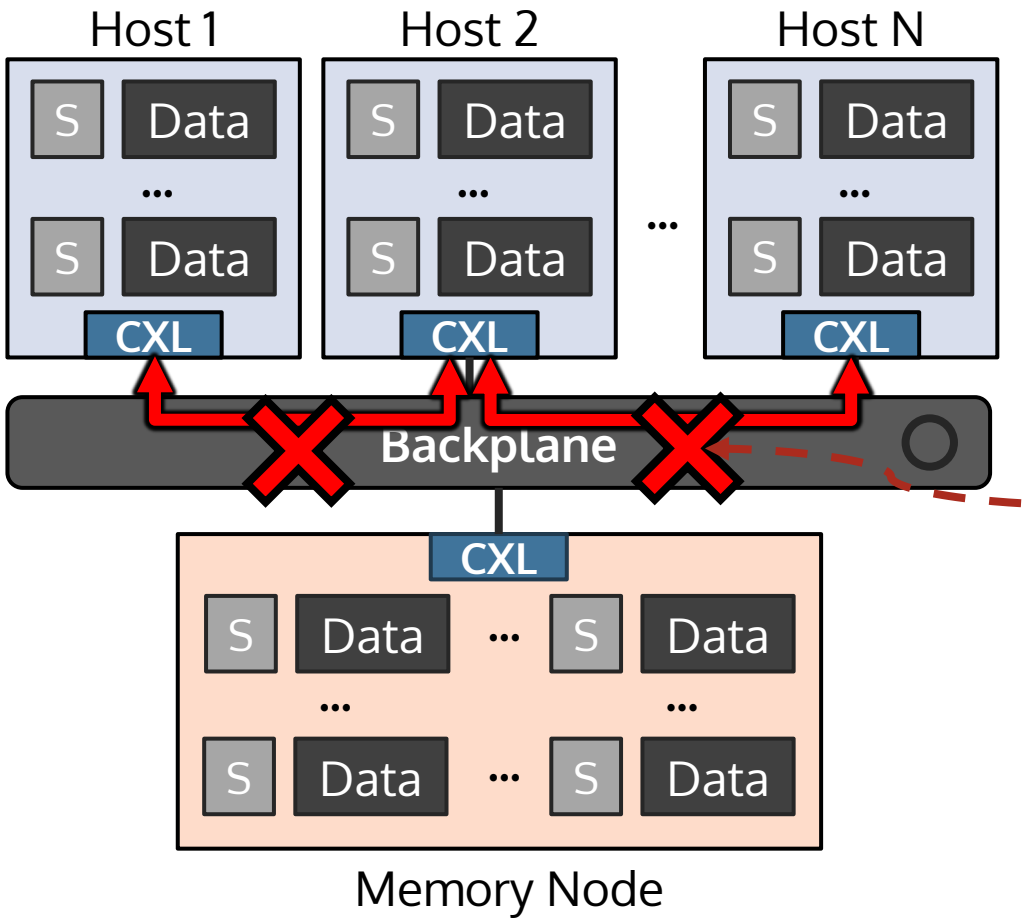
Goal 1: How can hosts share their coherence states?



CXL protocol defines coherence *only* between host and device

Enabling Multi-host Data Sharing

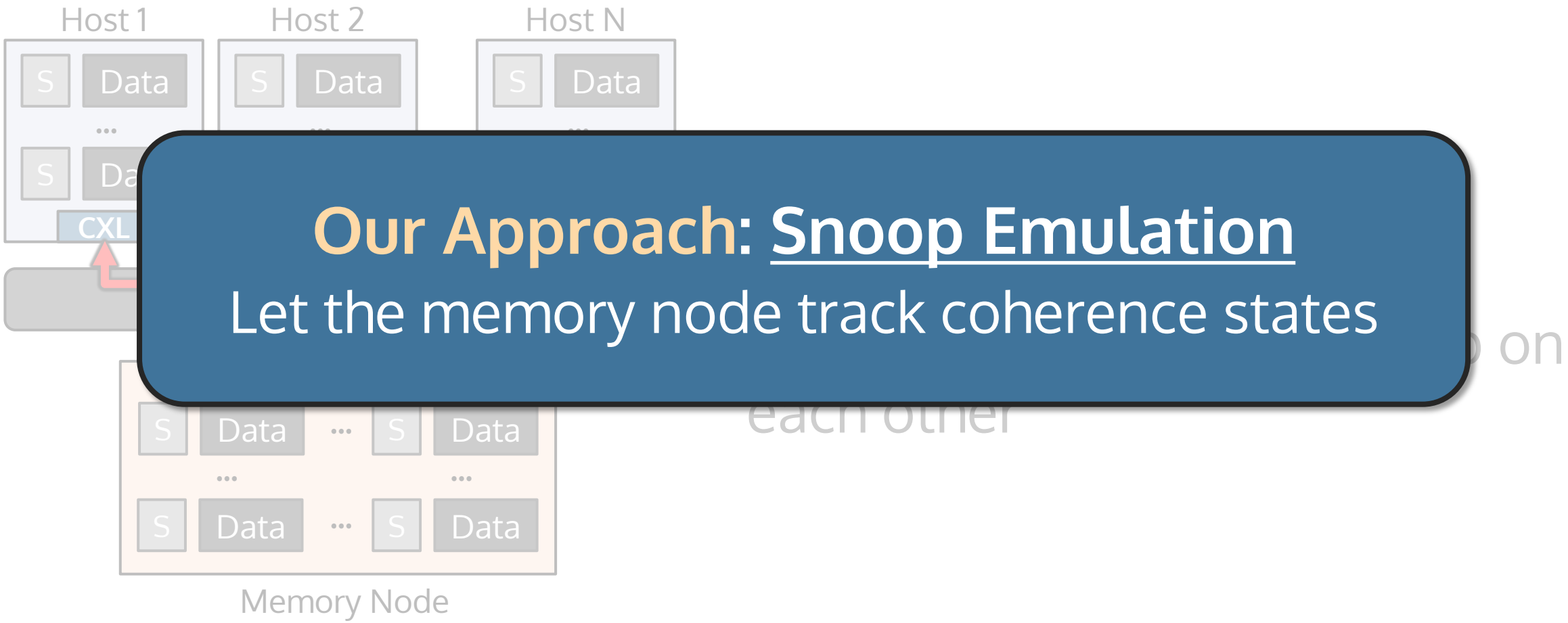
Goal 1: How can hosts share their coherence states?



Hosts *cannot* directly snoop on each other

Enabling Multi-host Data Sharing

Goal 1: How can hosts share their coherence states?



Our Approach: Snoop Emulation

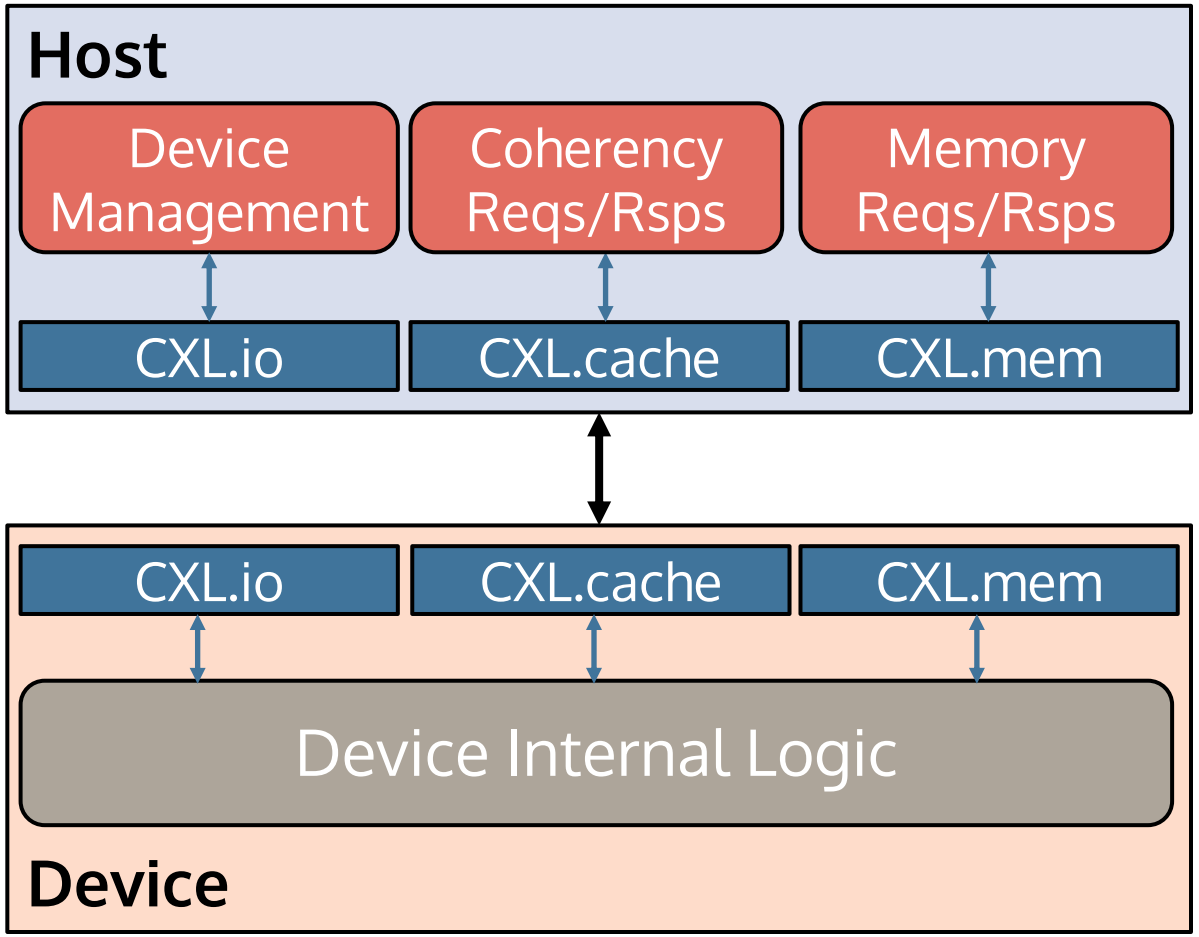
Let the memory node track coherence states

Enabling Multi-host Data Sharing

Goal 2: How to design a multi-host coherence control flow?

CXL Protocols

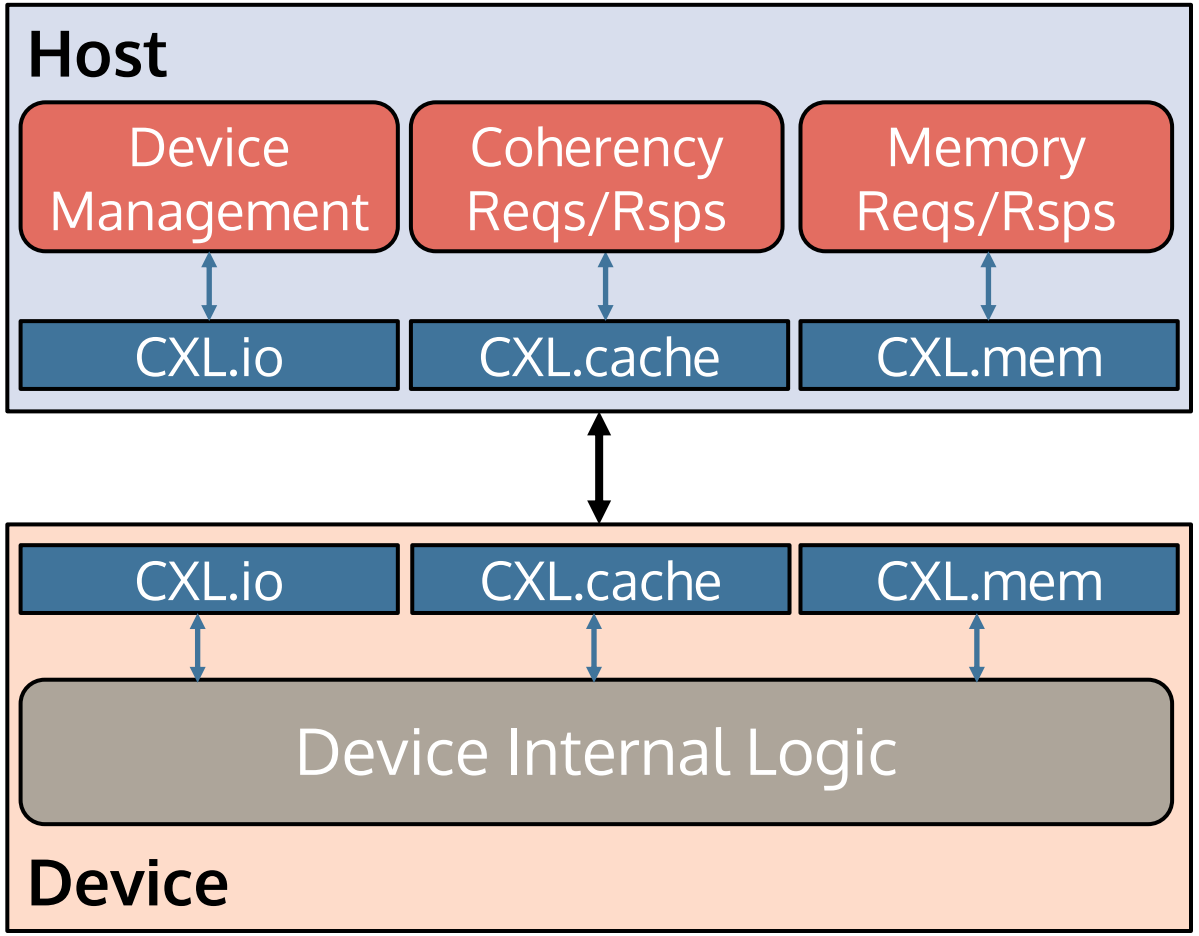
1. CXL.io
   - Device Management
2. CXL.cache
   - **Coherency Management**
3. CXL.mem
   - Memory Read/Write



Enabling Multi-host Data Sharing

Goal 2: How to design a multi-host coherence control flow?

Protocol	Message	Type
CXL.cache	RdOwn	Device Request
	RdAny	
	CLFlush	
	GO-*	Host Response
CXL.mem	MemRd	Host Request
	MemWr	
	MemInv	
	MemRdFwd	
	Cmp-*	Device Response



Enabling Multi-host Data Sharing

Goal 2: How to design a multi-host coherence control flow?

Protocol	Message	Type
CXL.cache	RdOwn	Device Request
	RdAny	
	CLFlush	
	GO-*	Host Response
CXL.mem	MemRd	Host Request
	MemWr	
	MemInv	
	MemRdFwd	
	Cmp-*	Device Response

CXL Protocols

- A set of valid request/response pairs between Host and Device
- Design a sharing-enabled control flow strictly using the valid pairs

Enabling Multi-host Data Sharing

Goal 2: How to design a multi-host coherence control flow?

Protocol	Message	Type
	RdOwn	
CXL.mem	MemInv	Host Request
	MemRdFwd	
	Cmp-*	Device Response

CXL Protocols

- A set of valid request/response

Our Approach: Sharing-enabled Control Flow

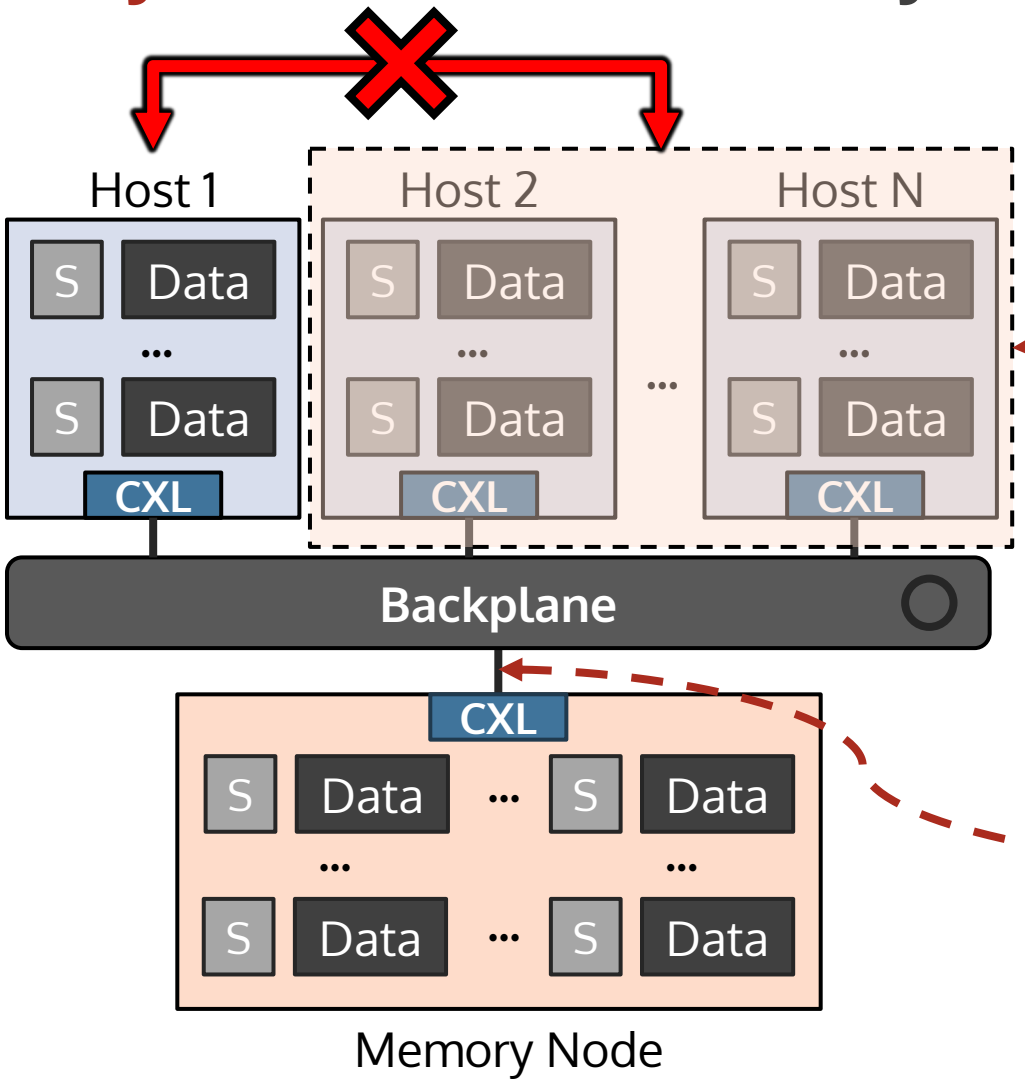
Let's exploit CXL.cache messages

Outline

- Introduction
- Motivation
- **SDM: Sharing-enabled Disaggregated Memory System**
  - CXL-compatible Designs
    - Snoop Emulation
    - CXL-compatible Control Flow
- Evaluation
- Conclusion

Snoop Emulation

Key Idea: Let the memory node track coherence states



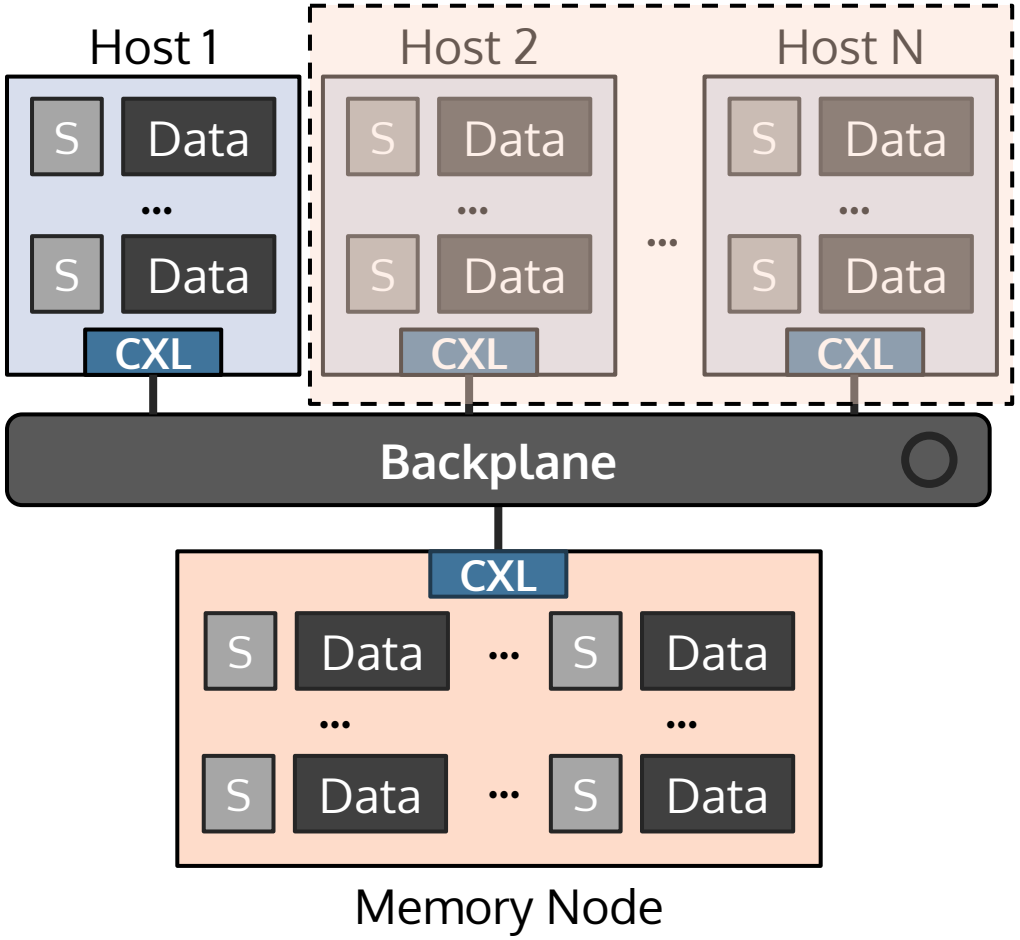
Abstract all nodes as devices except for the requester compute node

Requester and other compute nodes do not directly interact!

Memory emulates snooping on hosts by leveraging CXL.cache

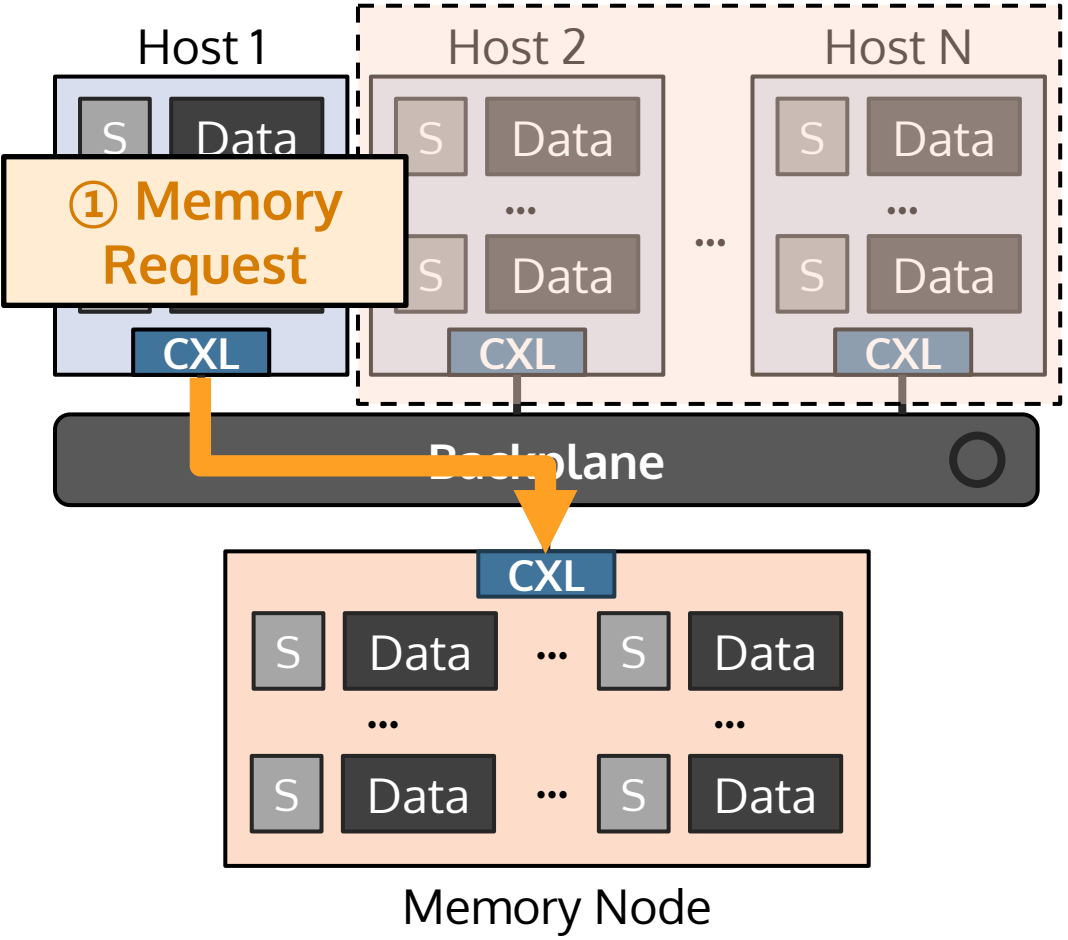
Snoop Emulation

Key Idea: Let the memory node track coherence states



Snoop Emulation

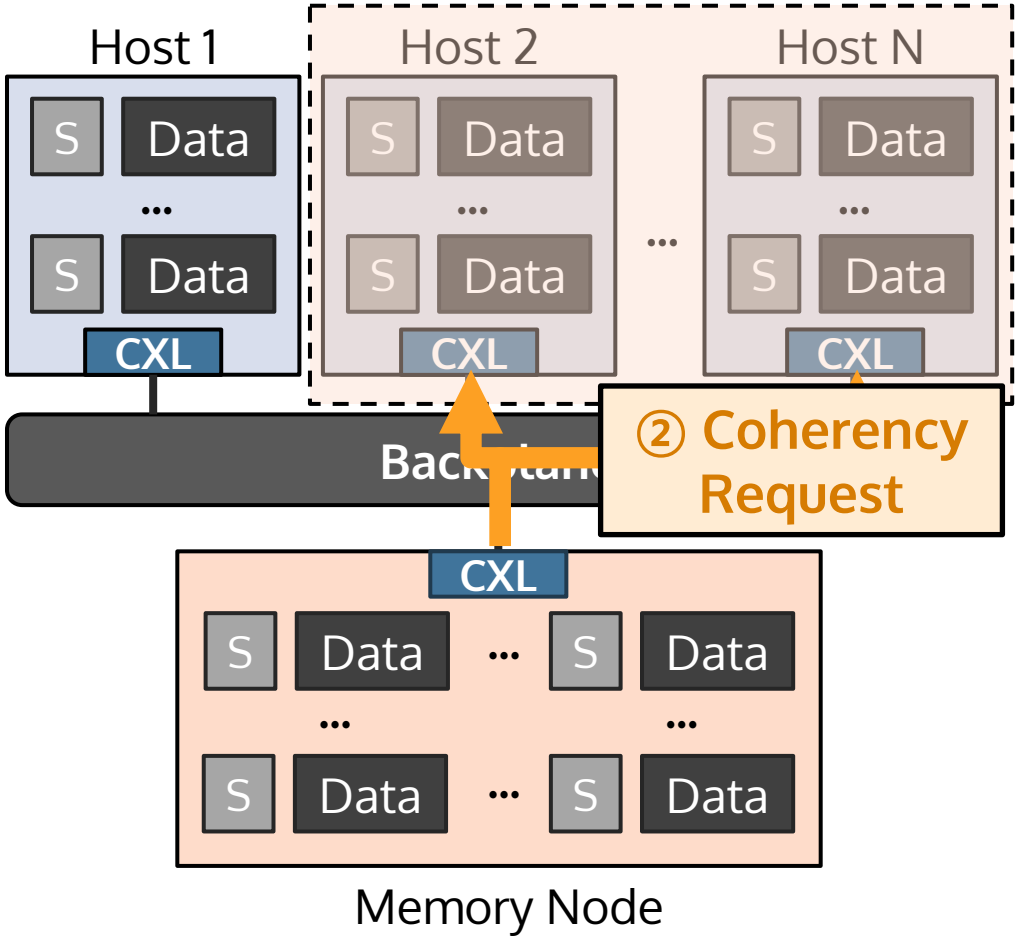
Key Idea: Let the memory node track coherence states



1. **Memory Request**
   from **Host** to **Device**

Snoop Emulation

Key Idea: Let the memory node track coherence states

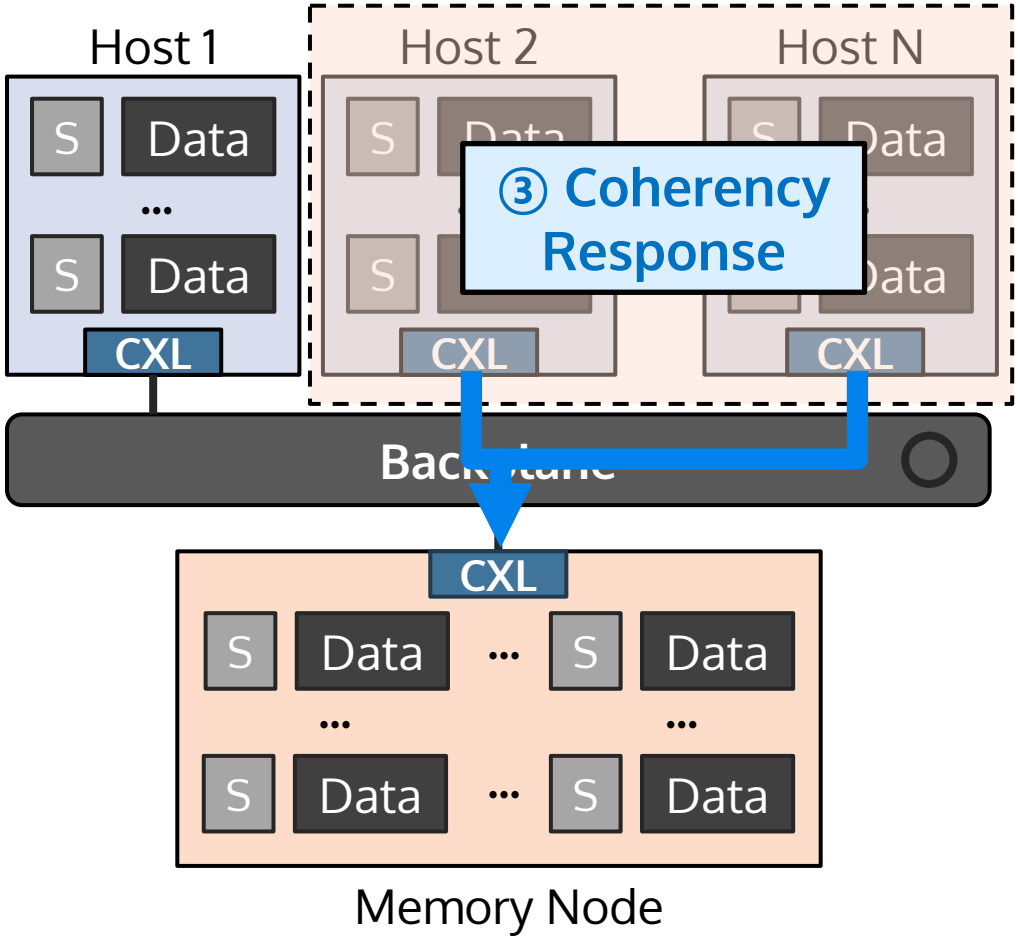


1. **Memory Request**
   from **Host** to **Device**

2. **Coherency Request**
   from **Device** to other **Hosts**

Snoop Emulation

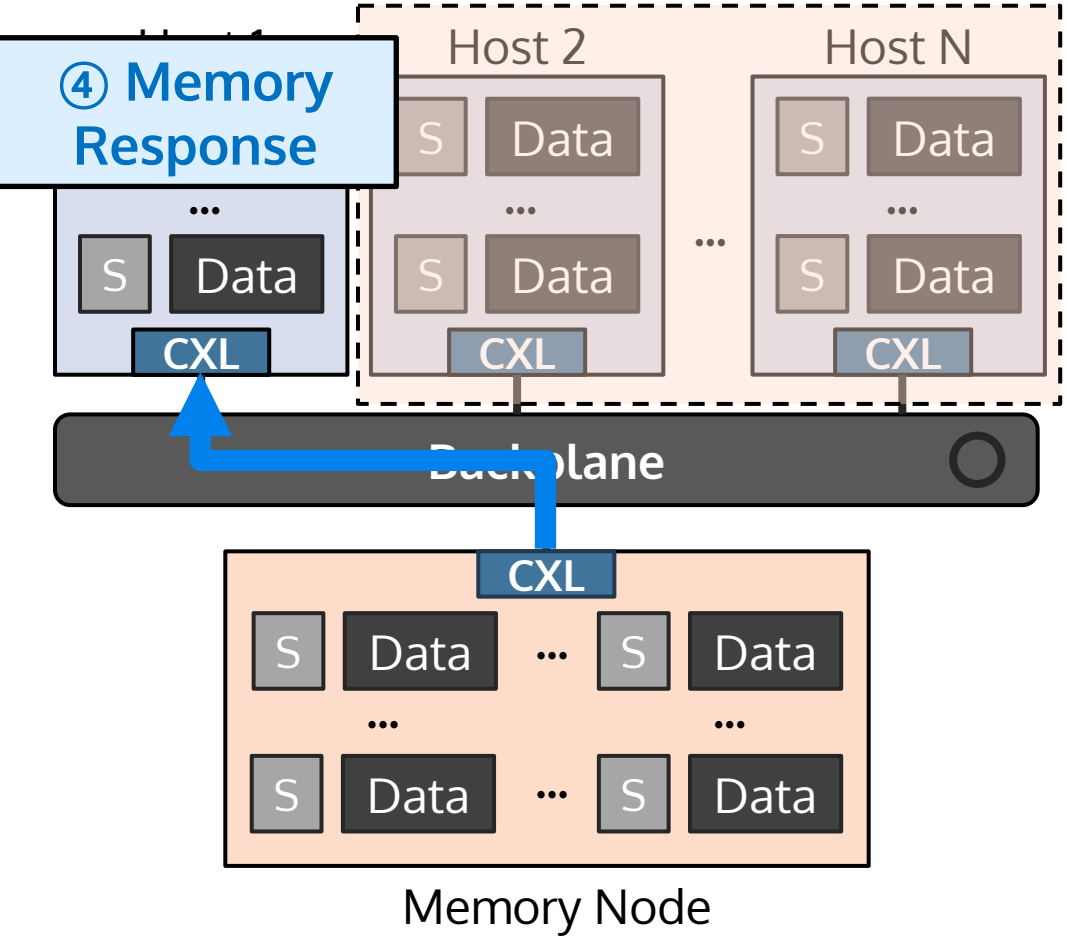
Key Idea: Let the memory node track coherence states



1. **Memory Request**
   from **Host** to **Device**
2. **Coherency Request**
   from **Device** to other **Hosts**
3. **Coherency Response**
   from other **Hosts** to **Device**

Snoop Emulation

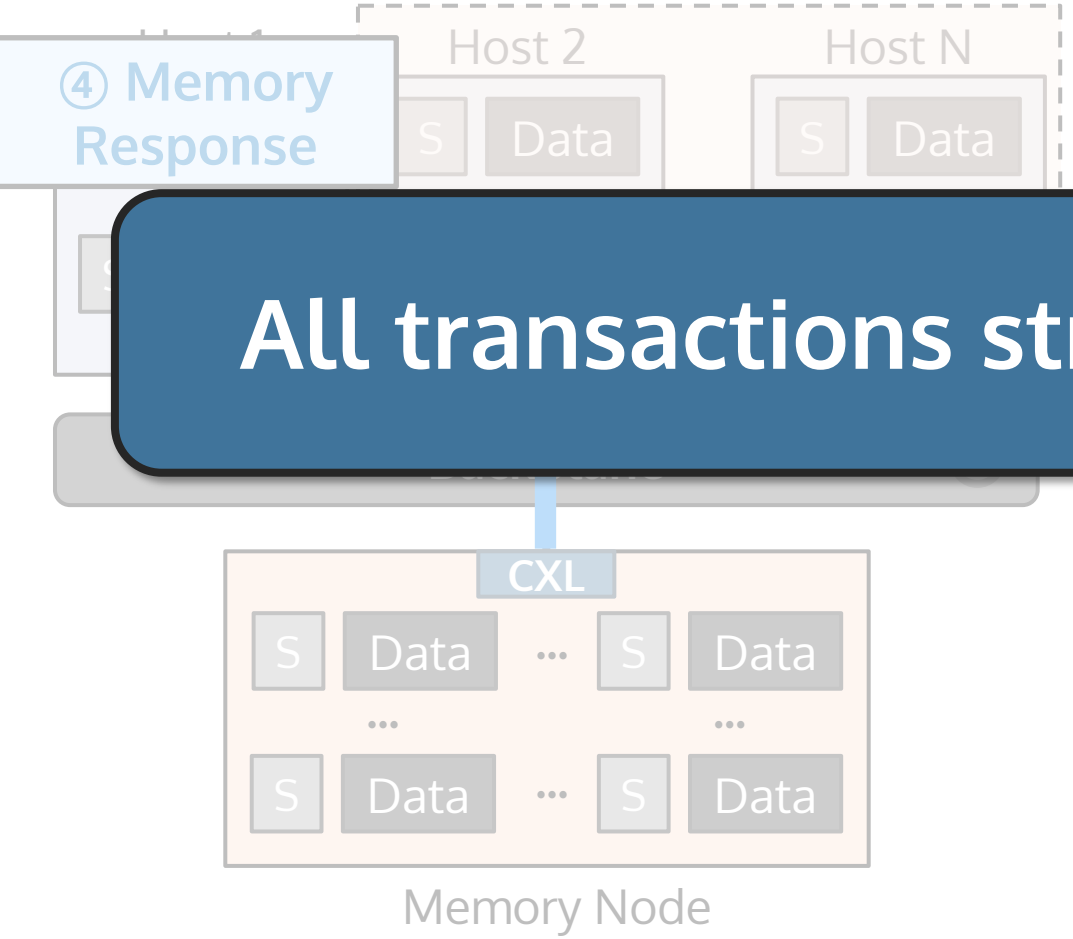
Key Idea: Let the memory node track coherence states



1. **Memory Request**
   from **Host** to **Device**
2. **Coherency Request**
   from **Device** to other **Hosts**
3. **Coherency Response**
   from other **Hosts** to **Device**
4. **Memory Response**
   from **Device** to **Host**

Snoop Emulation

Key Idea: Let the memory node track coherence states



1. Memory Request
   from **Host** to **Device**

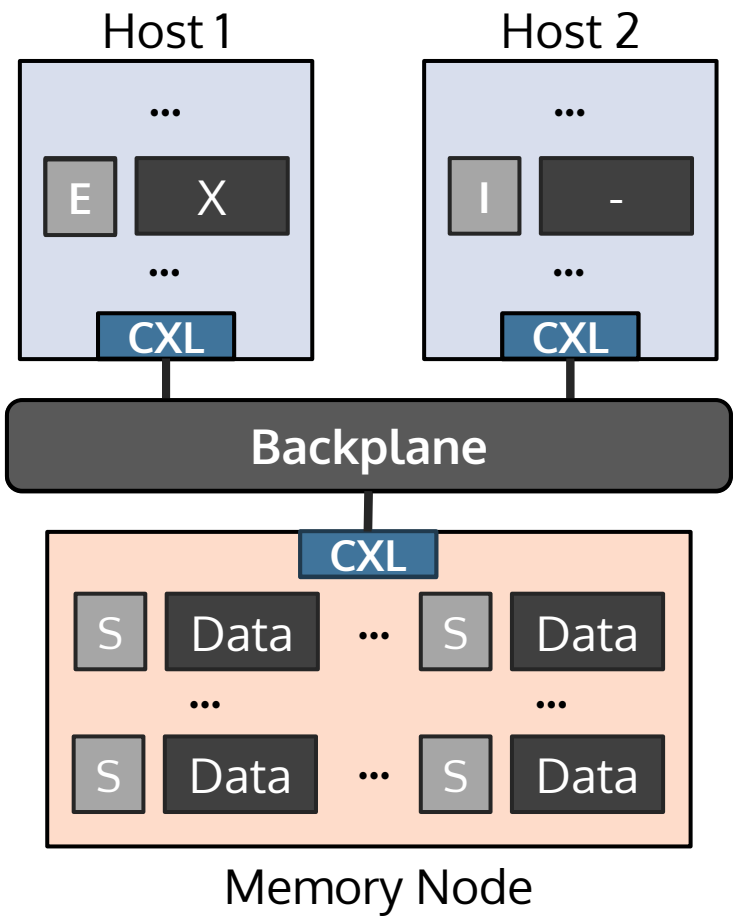
3. Coherency Response
   from other **Hosts** to **Device**

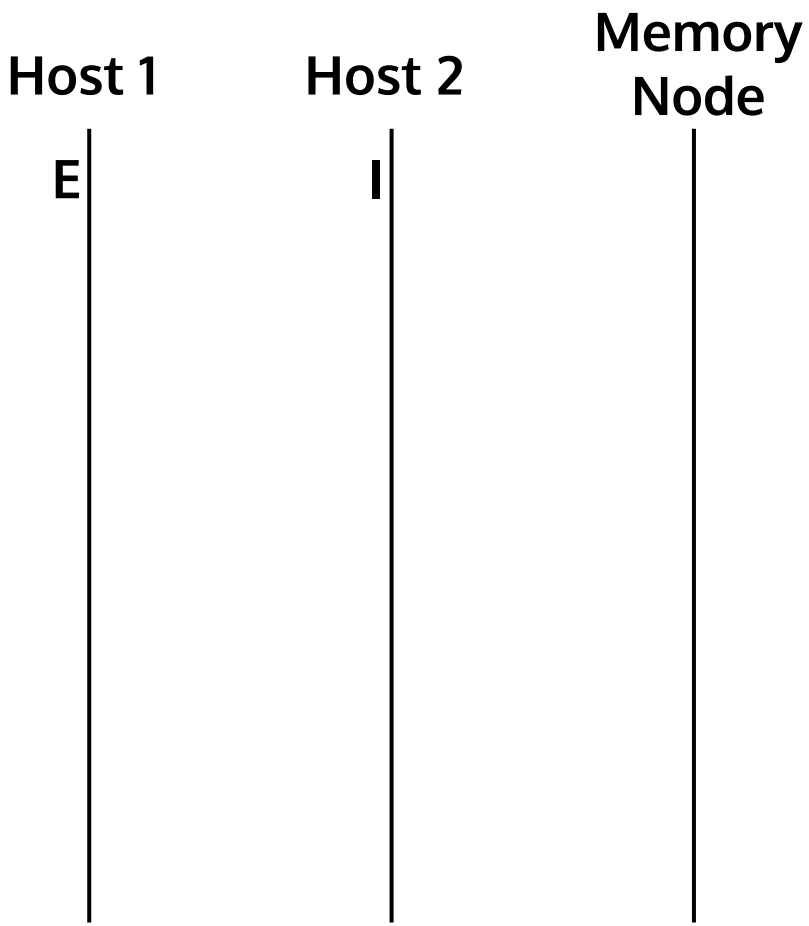
4. Memory Response
   from **Device** to **Host**

All transactions strictly follow CXL protocol!

CXL-compatible Control Flow

Straightforward: Invalidation-based Control Flow (INV-CF)

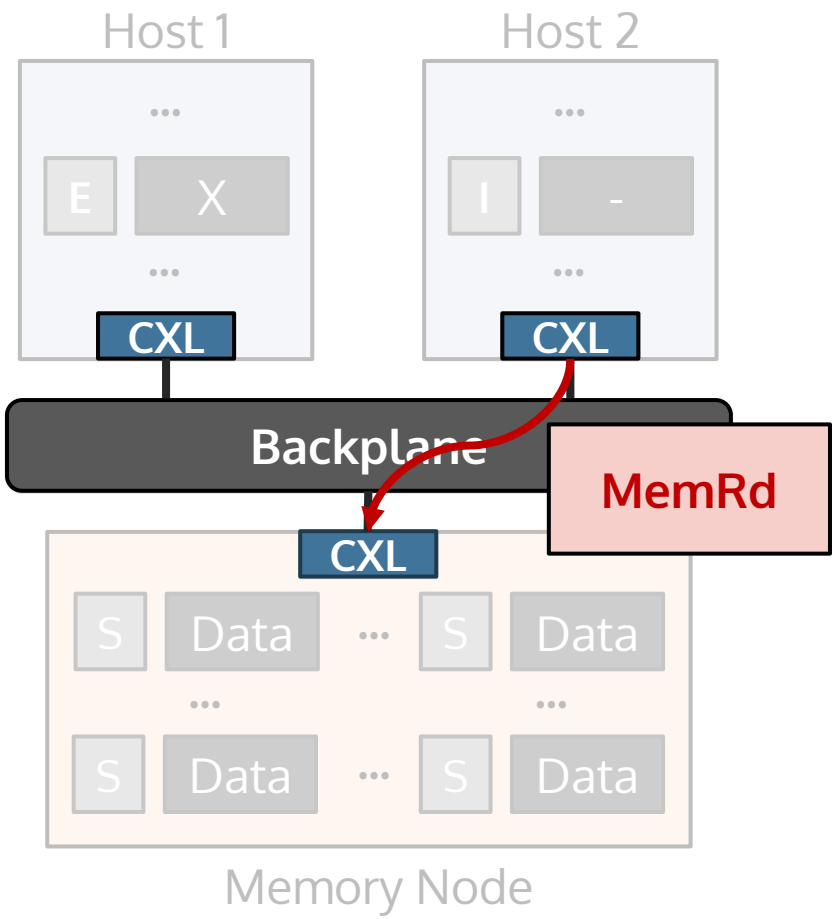


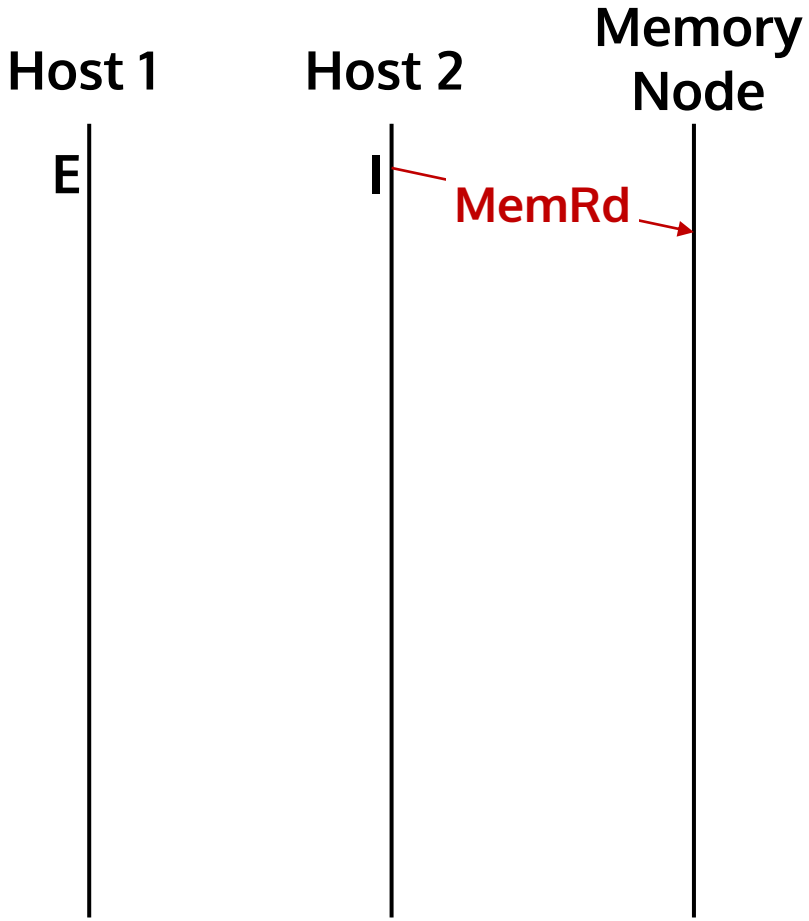


INV-CF (Read)

CXL-compatible Control Flow

Straightforward: Invalidation-based Control Flow (INV-CF)

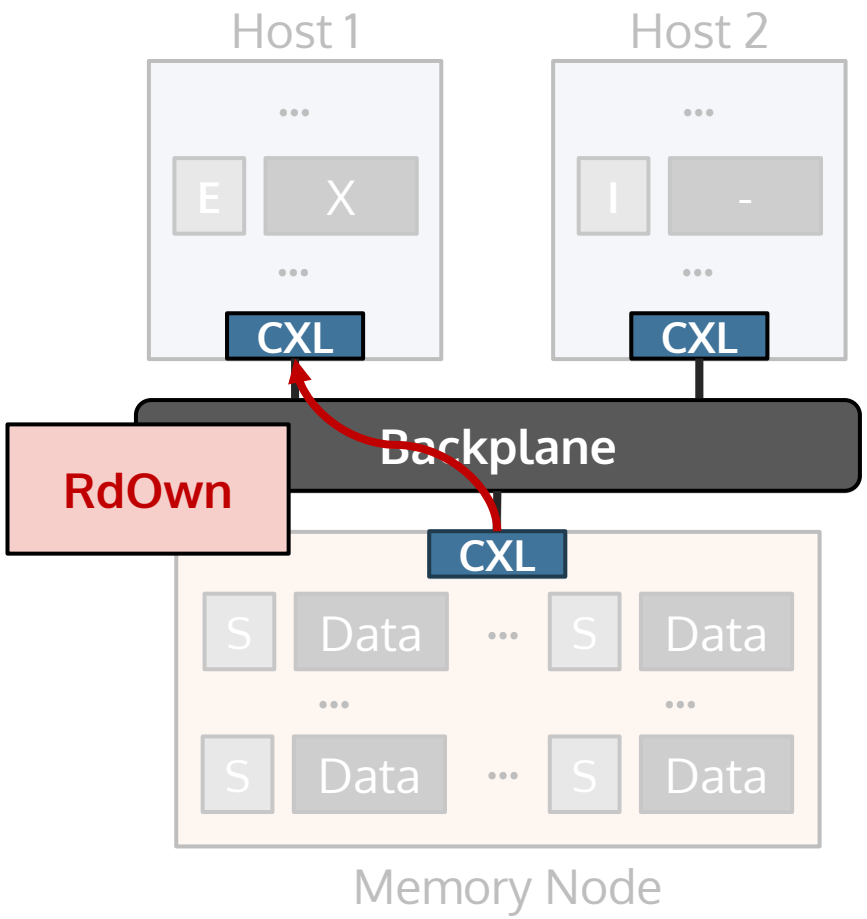


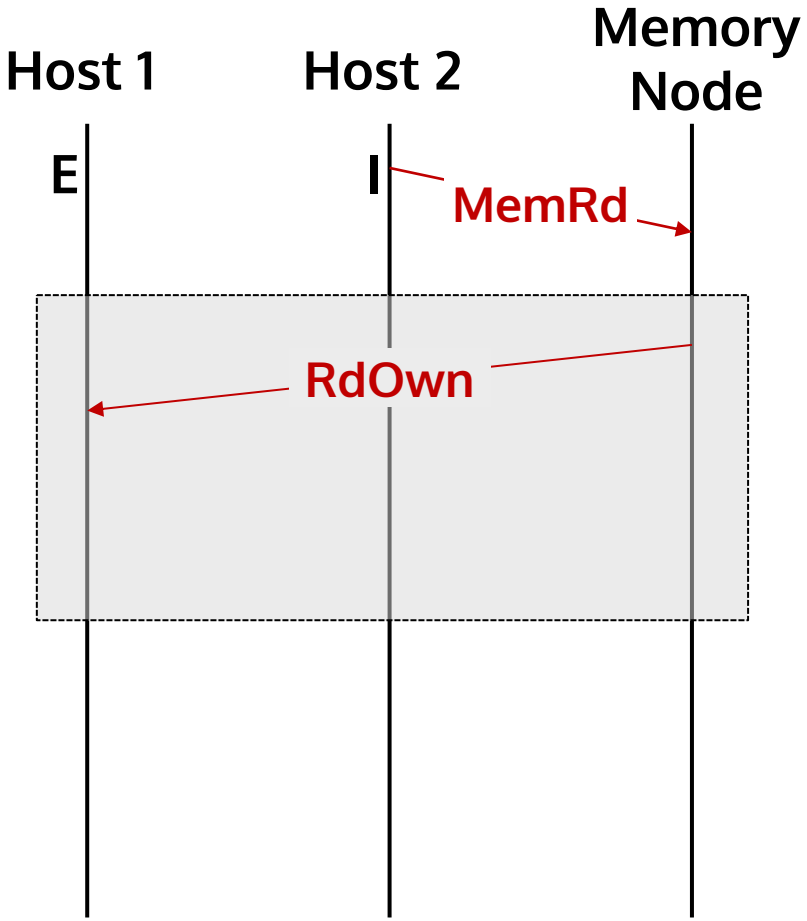


INV-CF (Read)

CXL-compatible Control Flow

Straightforward: Invalidation-based Control Flow (INV-CF)

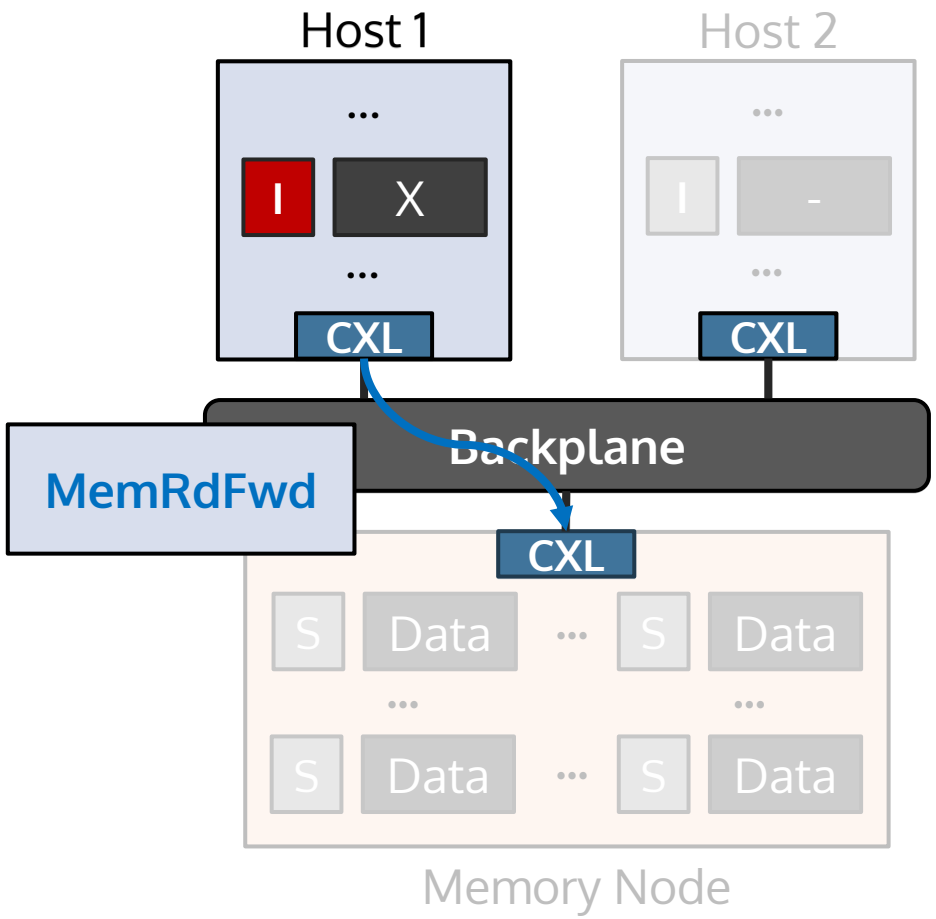


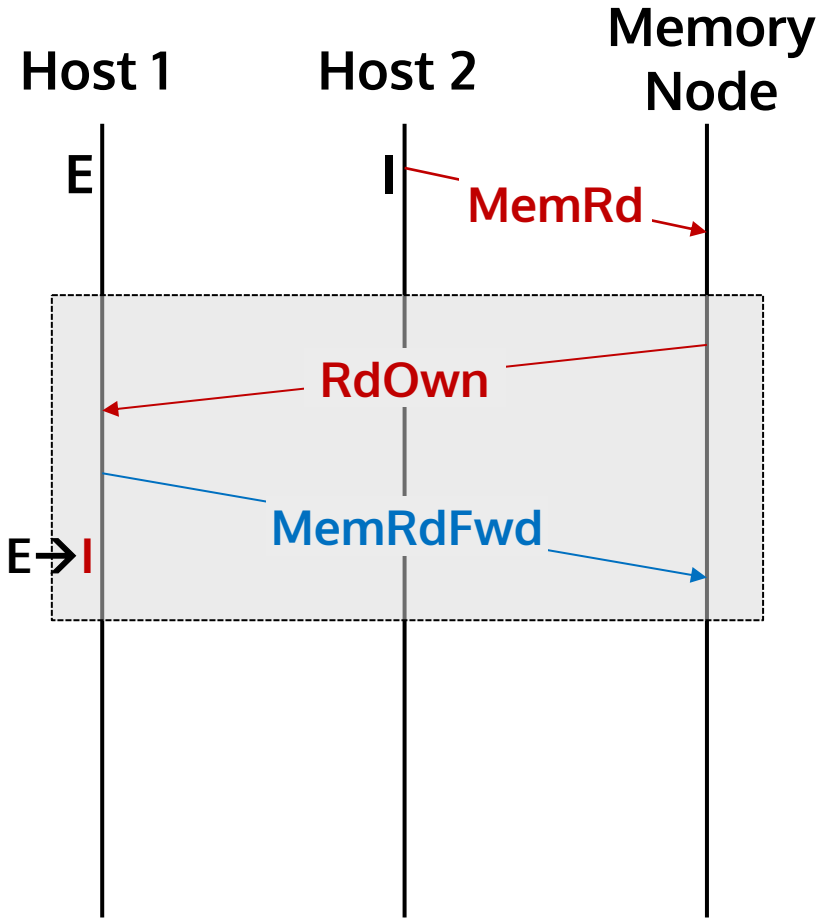


INV-CF (Read)

CXL-compatible Control Flow

Straightforward: Invalidation-based Control Flow (INV-CF)

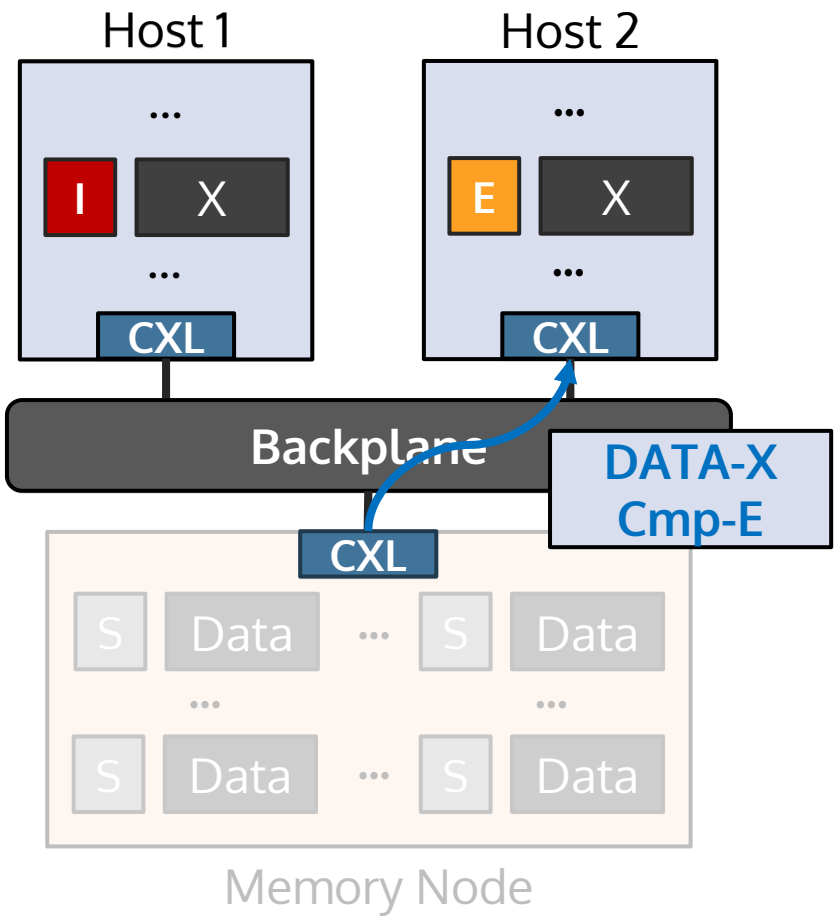


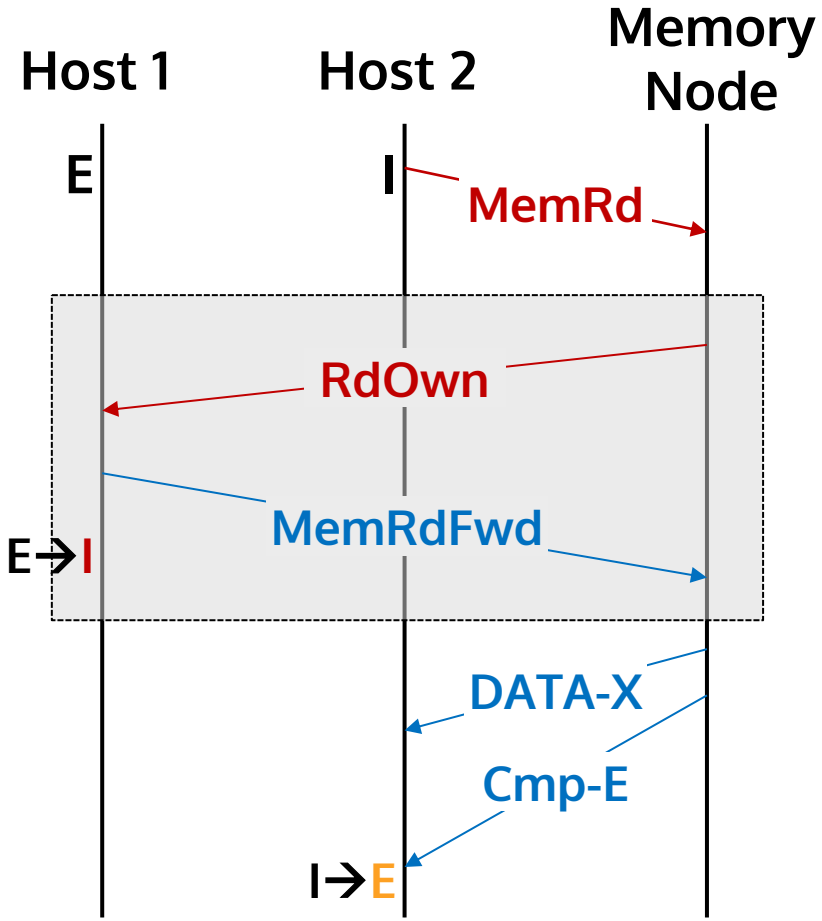


INV-CF (Read)

CXL-compatible Control Flow

Straightforward: Invalidation-based Control Flow (INV-CF)

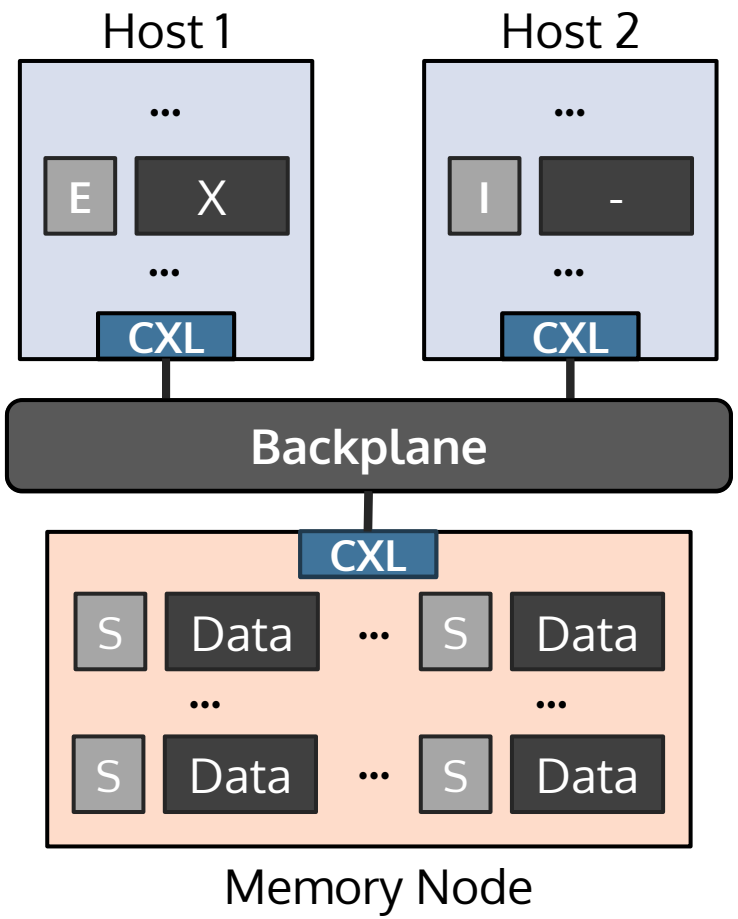


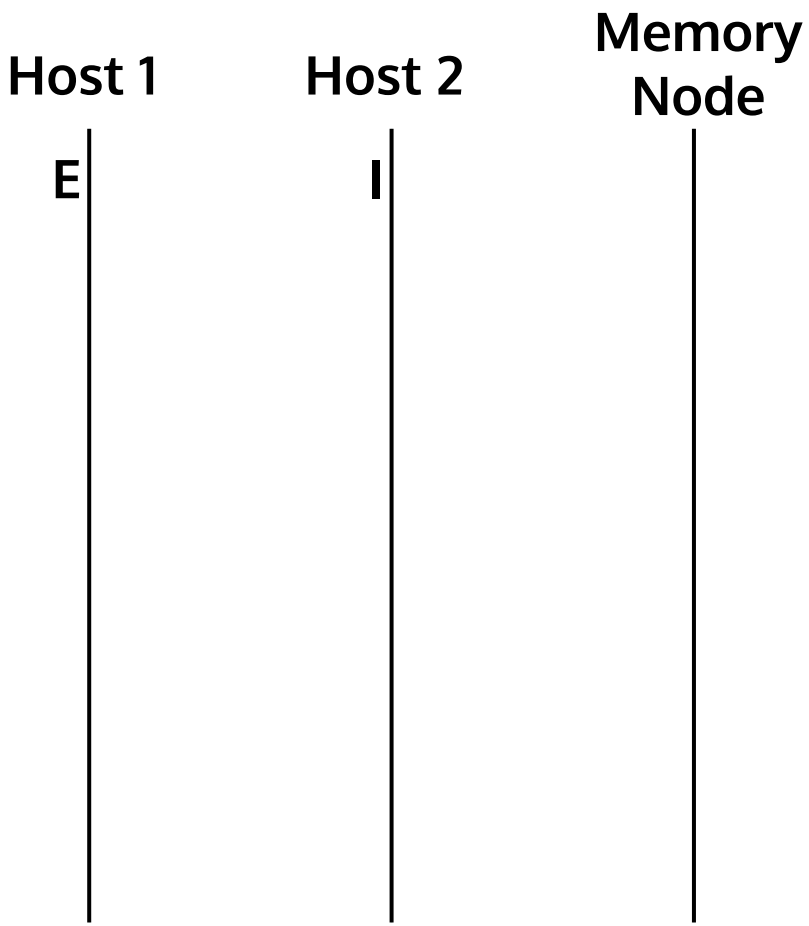


INV-CF (Read)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

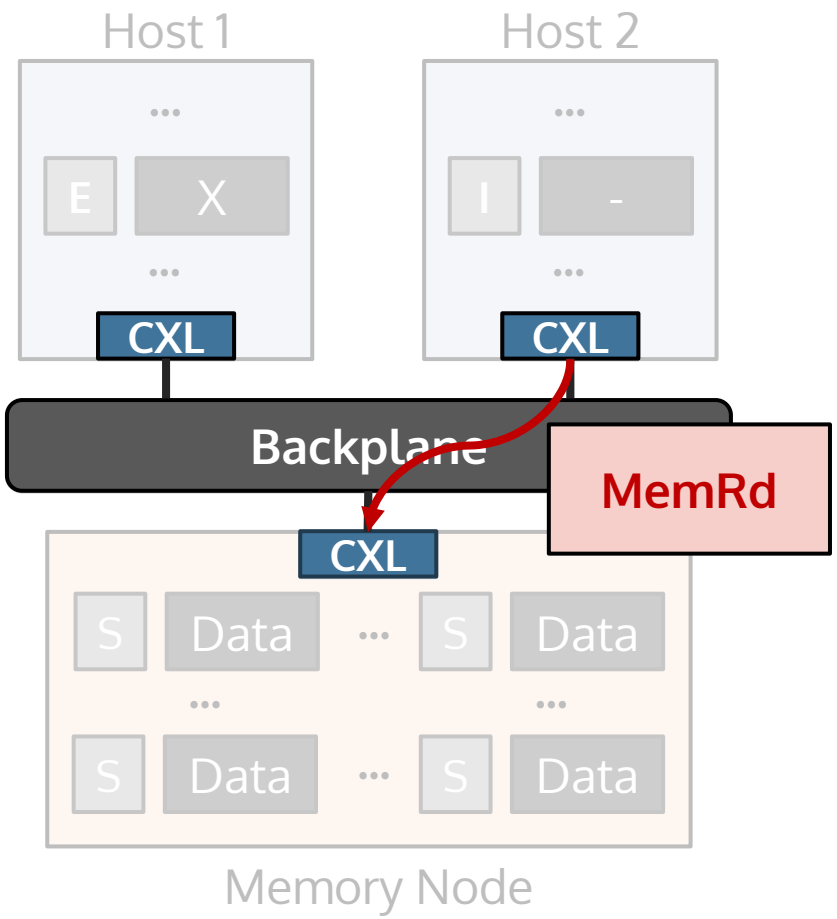


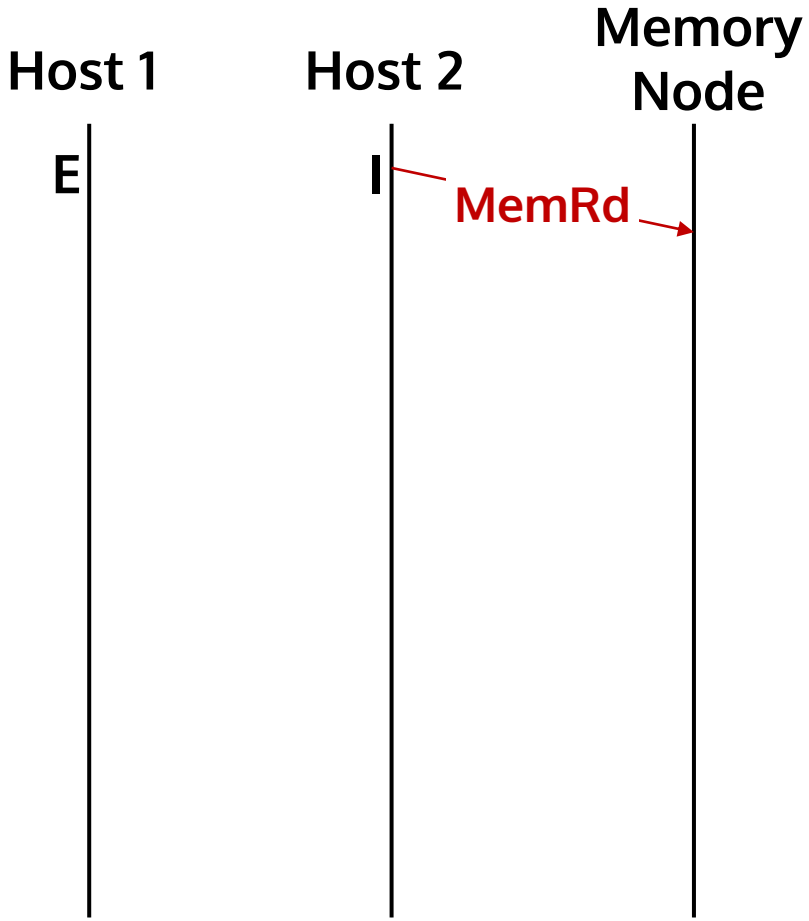


SHA-CF (Read)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

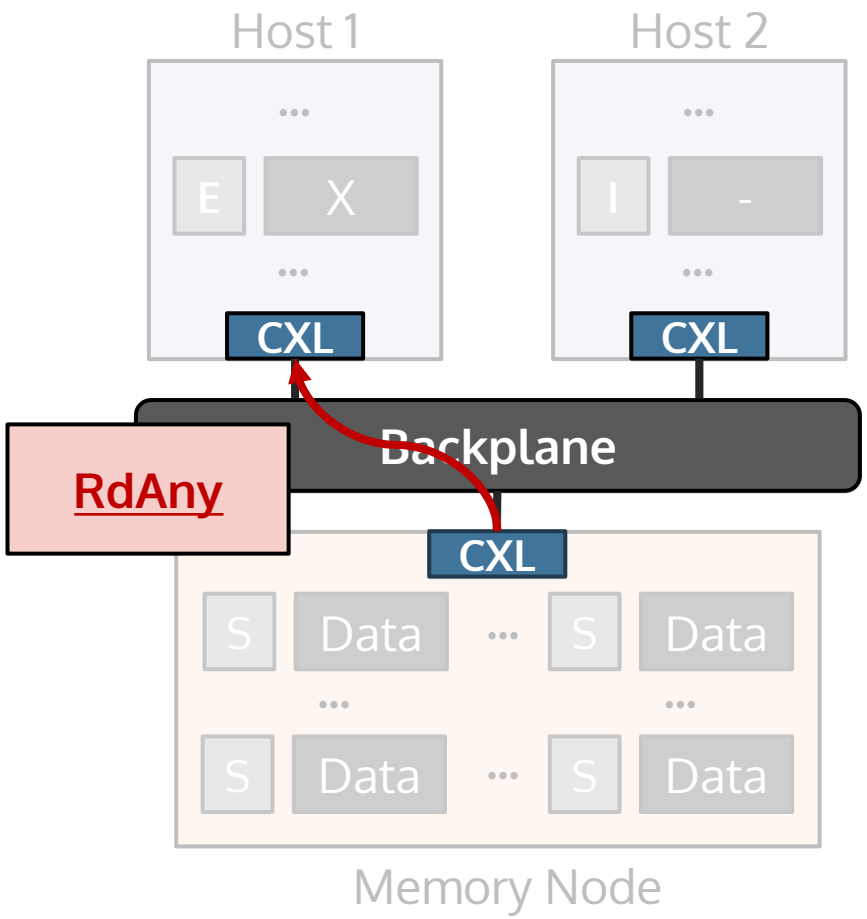


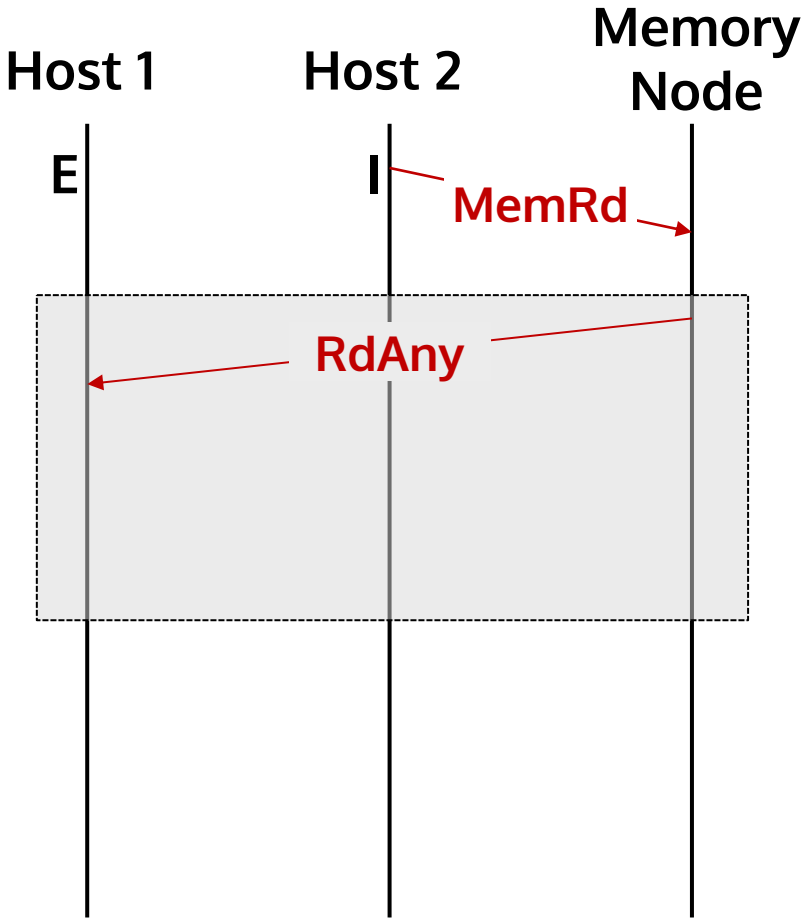


SHA-CF (Read)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

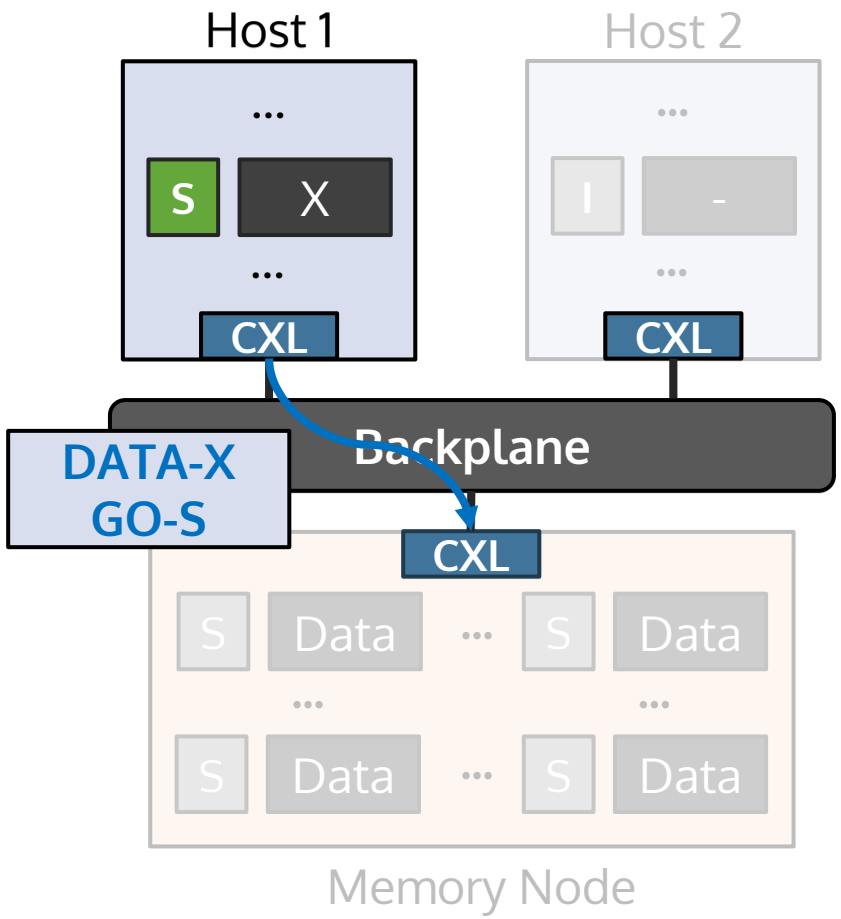


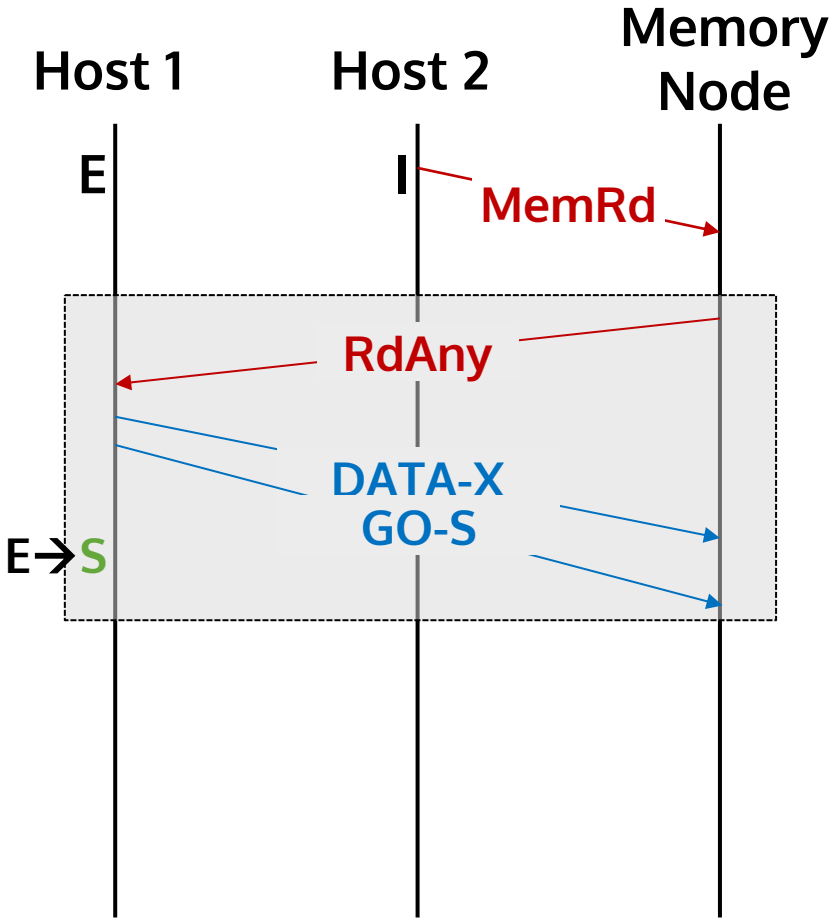


SHA-CF (Read)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

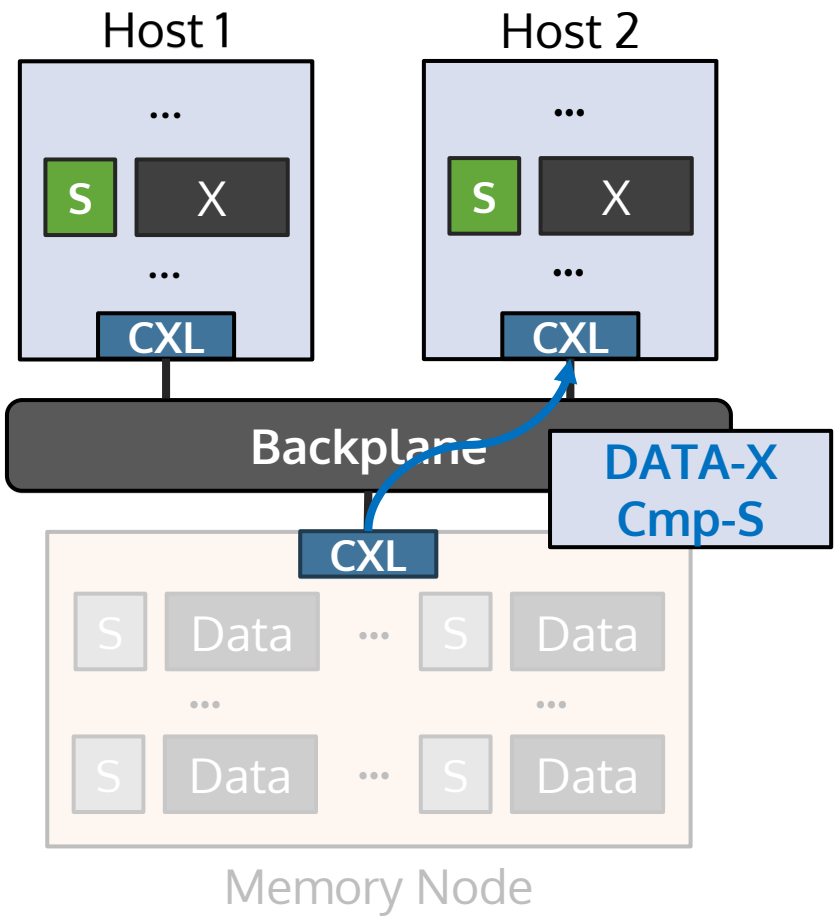


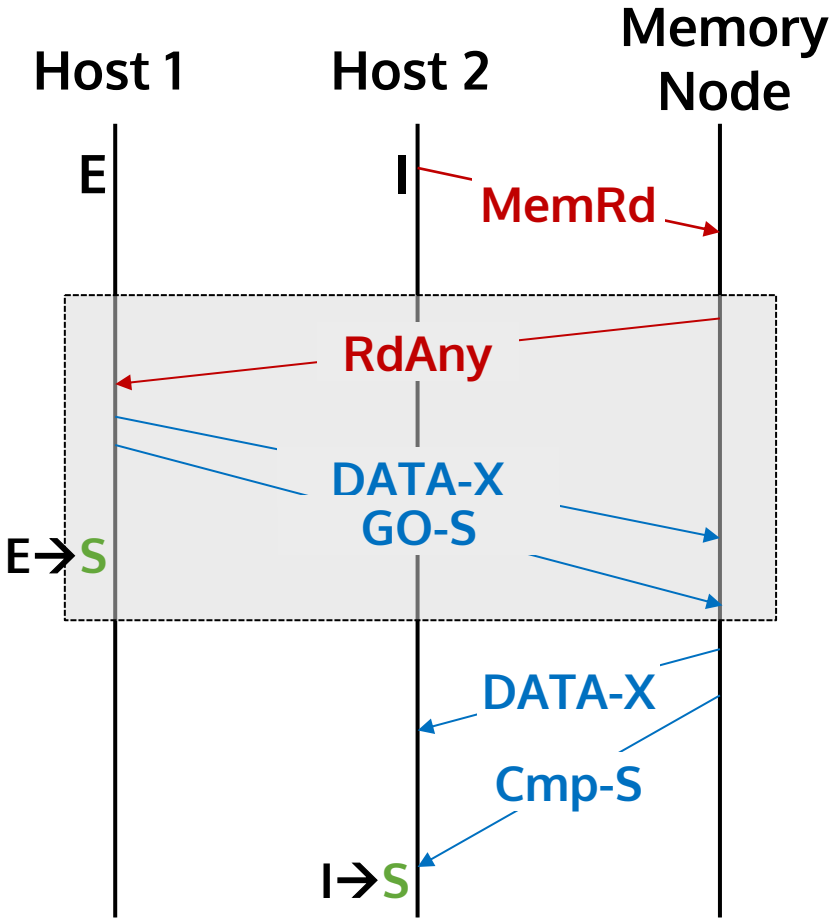


SHA-CF (Read)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

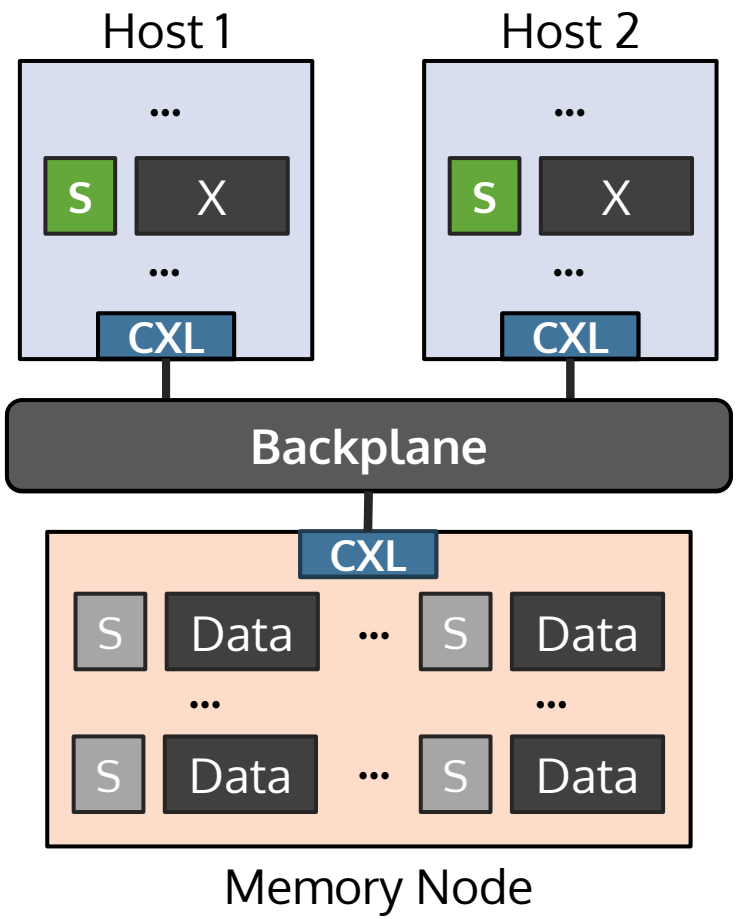


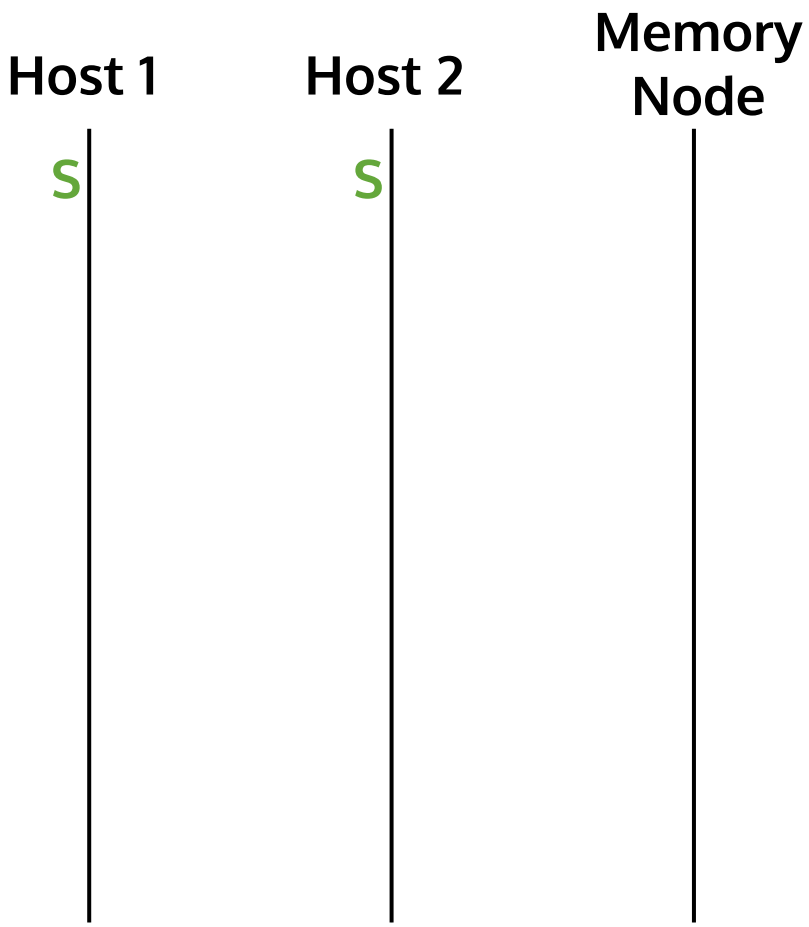


SHA-CF (Read)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

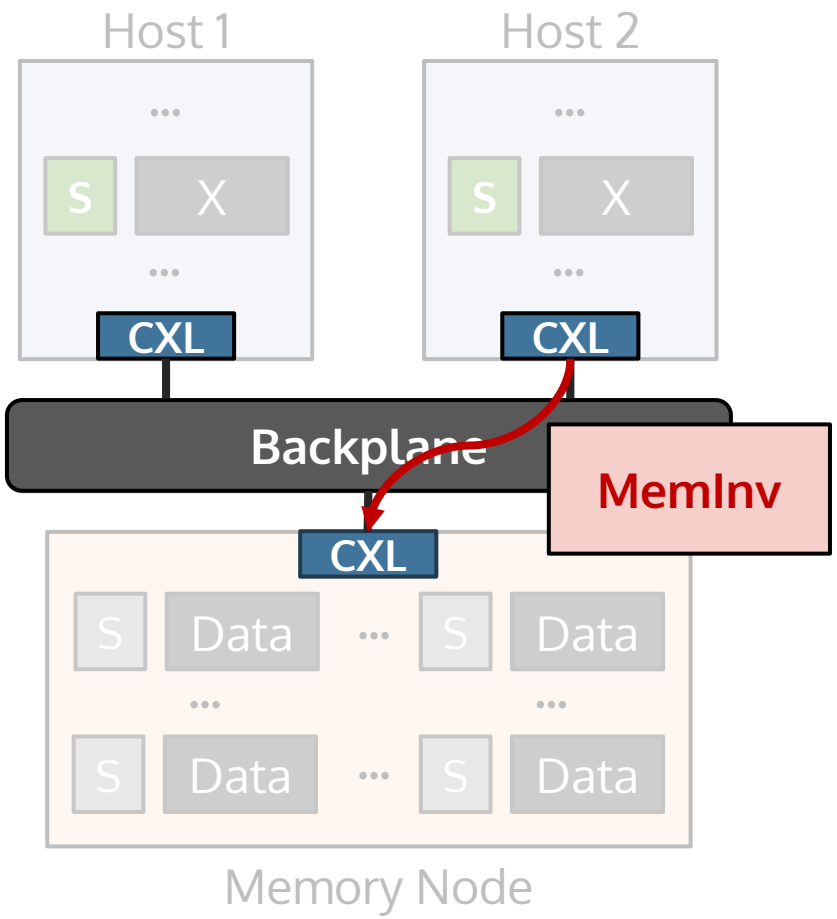


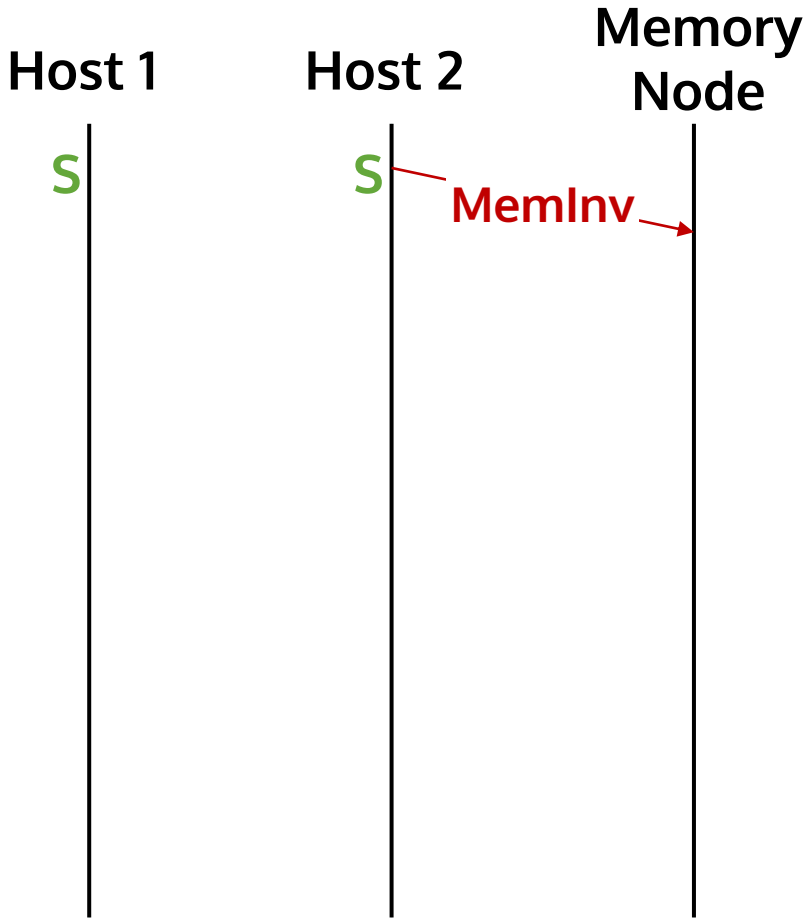


SHA-CF (Write)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

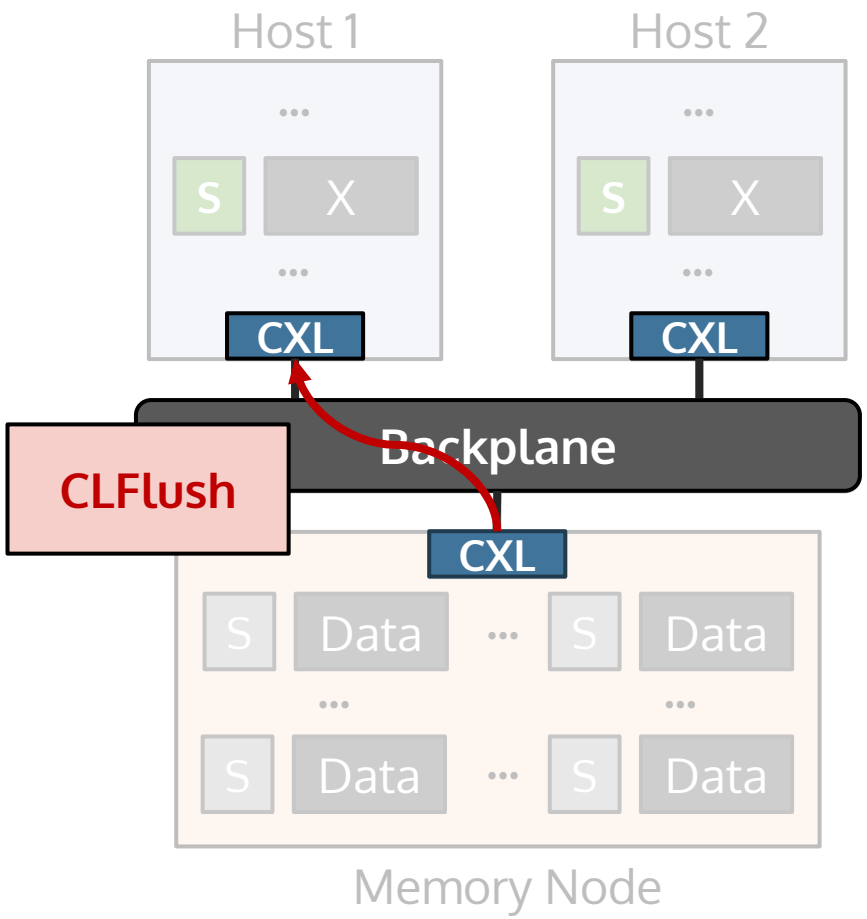


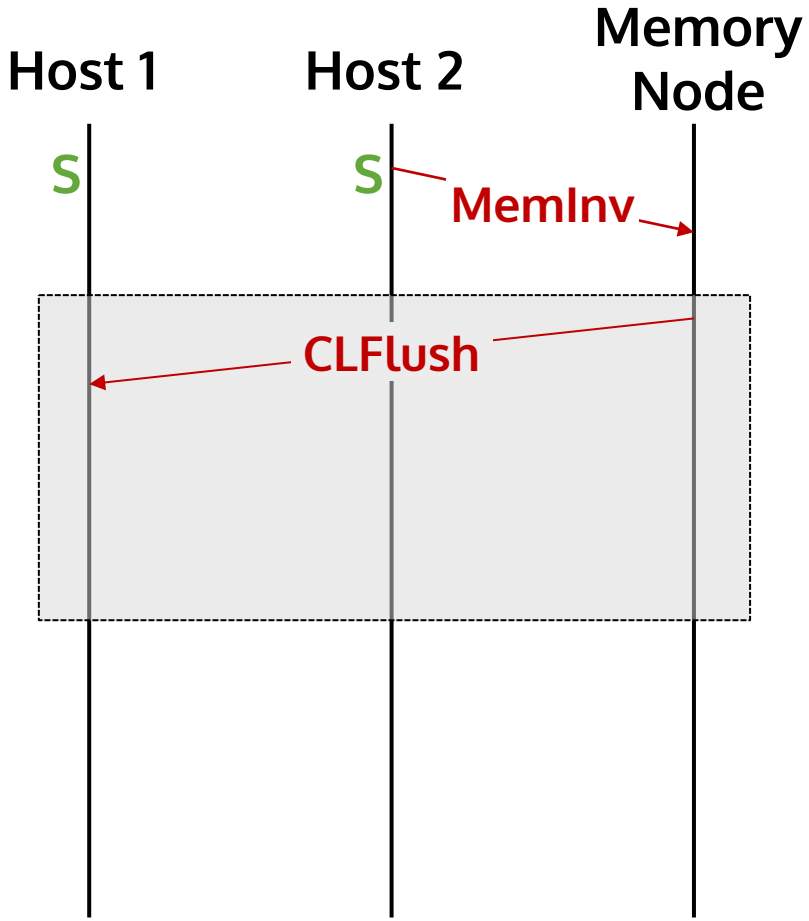


SHA-CF (Write)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

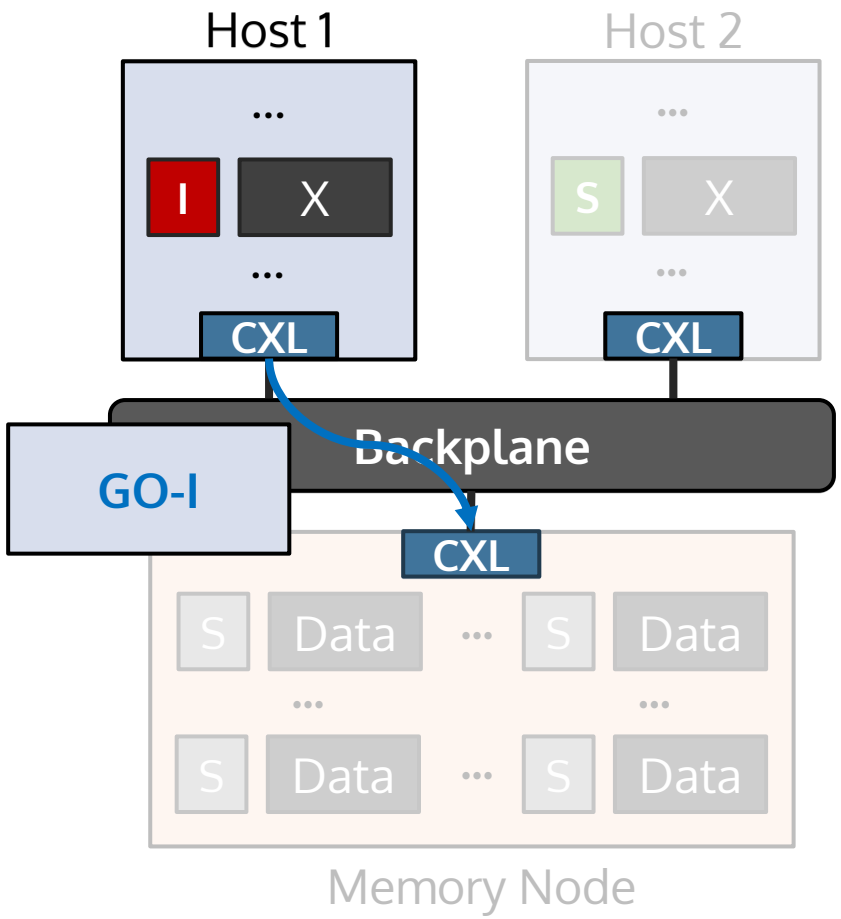


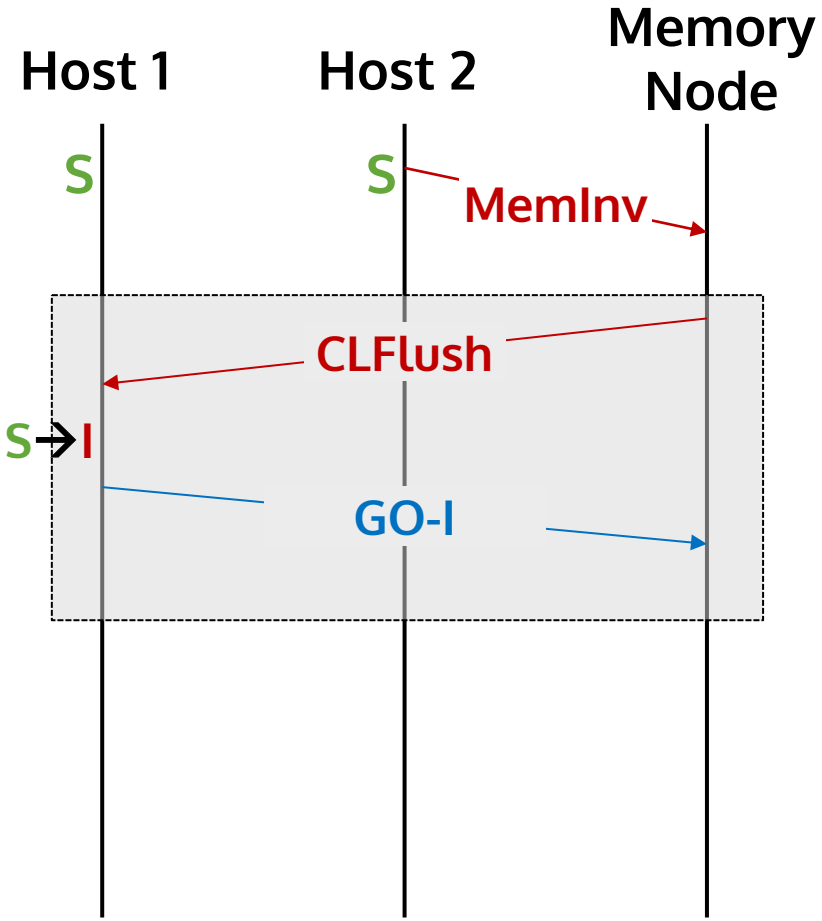


SHA-CF (Write)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

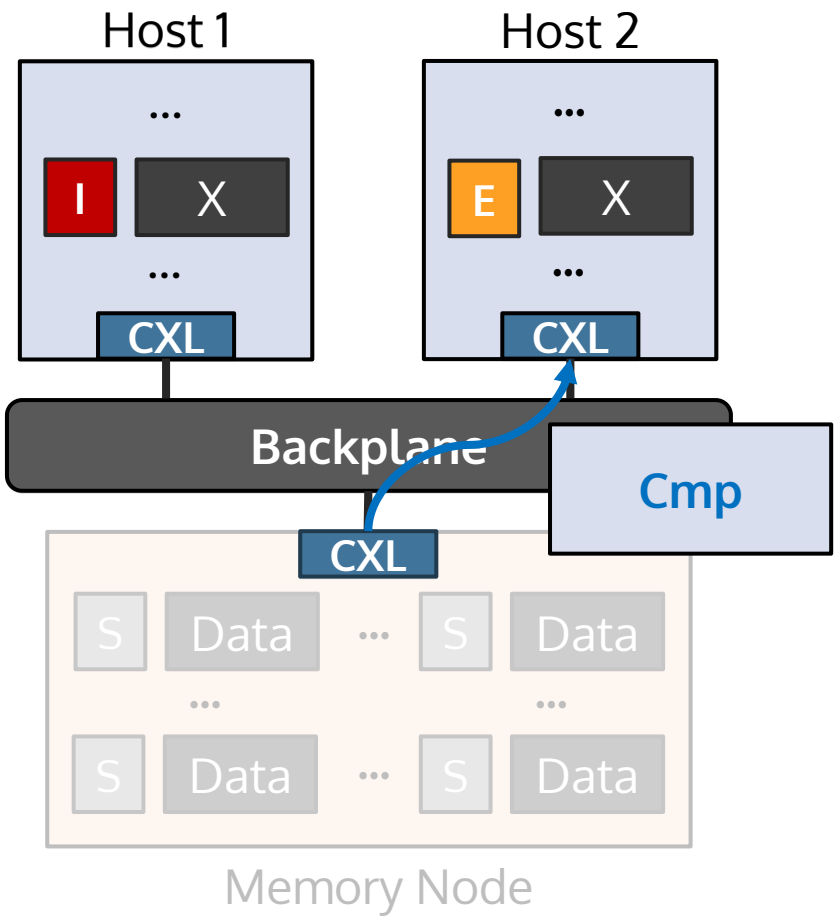


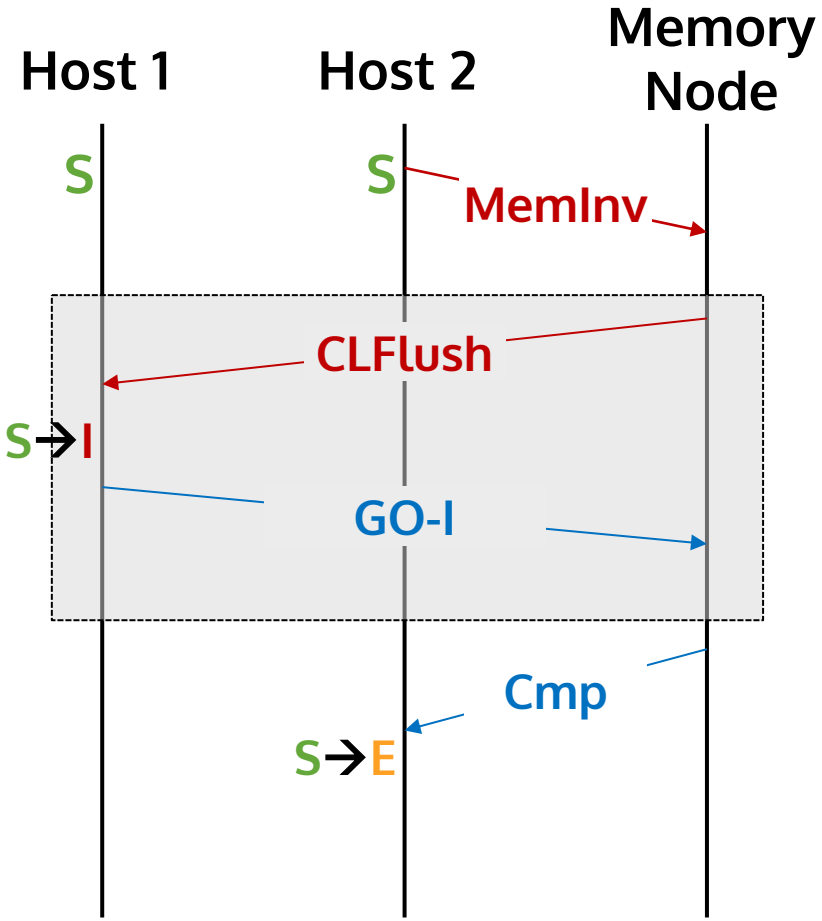


SHA-CF (Write)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)

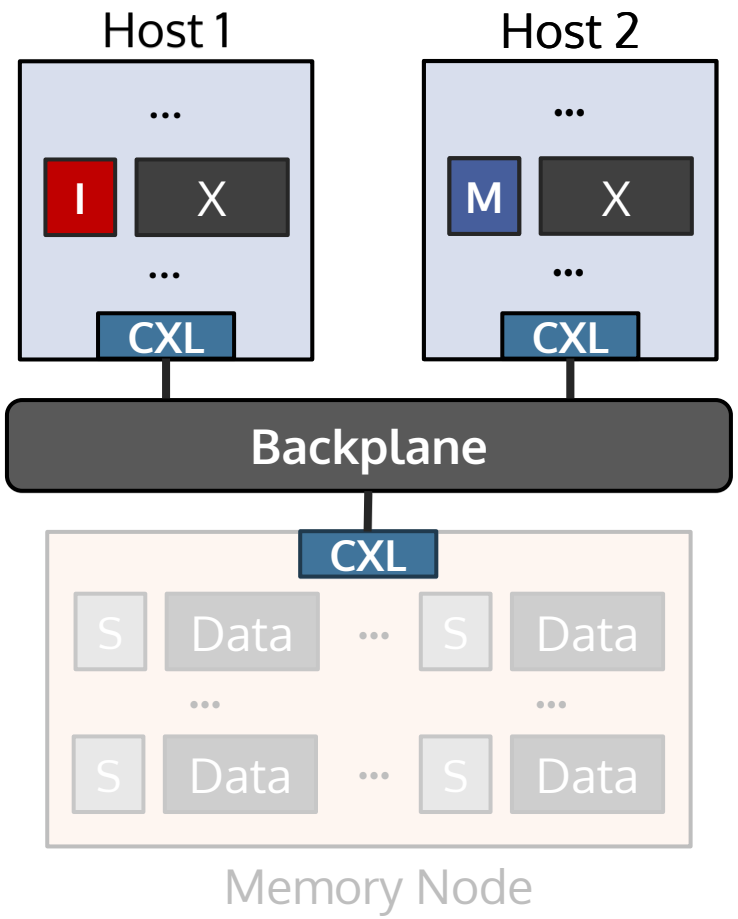


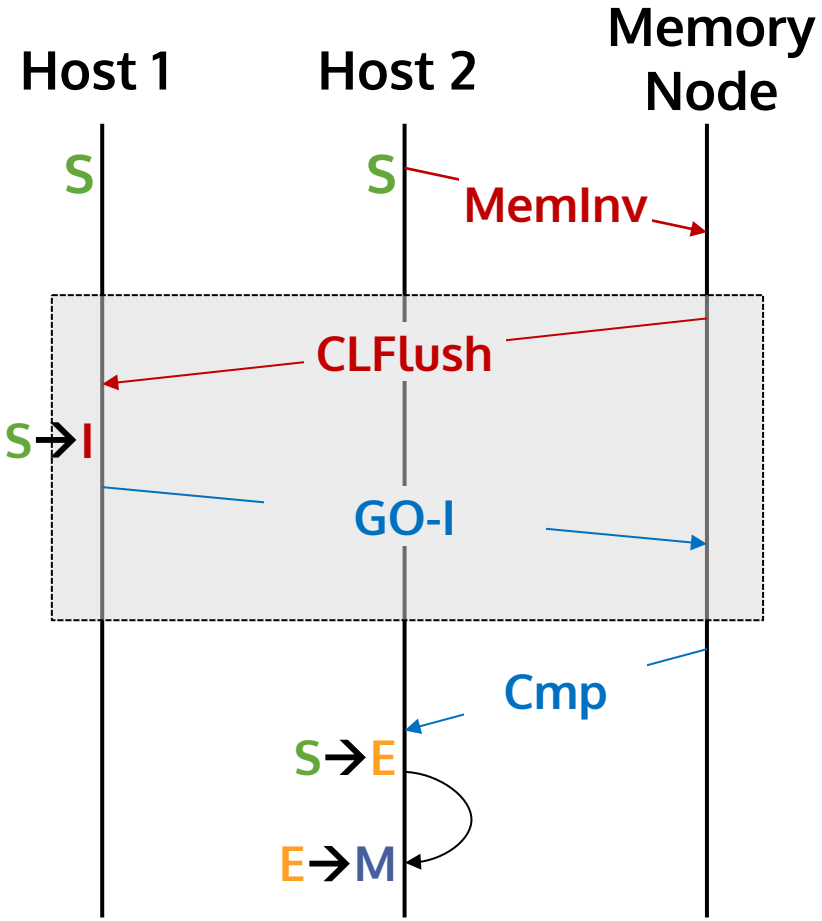


SHA-CF (Write)

CXL-compatible Control Flow

Sharing-enabled Control Flow (SHA-CF)





SHA-CF (Write)

CXL-compatible Memory Management

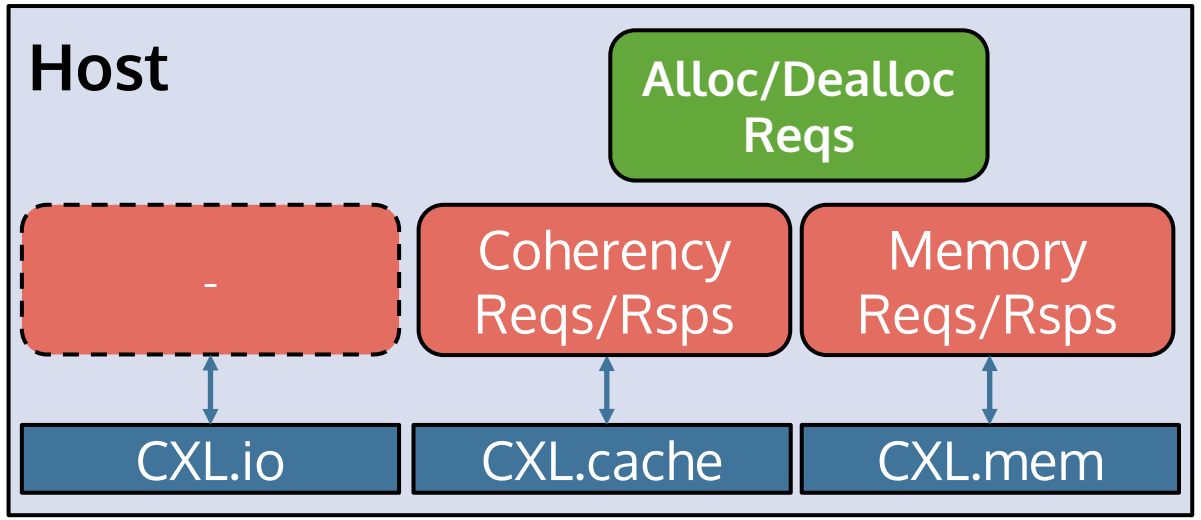
Q: How to allocate/deallocate pages from disaggregated memory?

Requirement

- Alloc/Dealloc messages **should not interfere with** RD/WR requests

Our Approach

- Define a new message using byte-15 of CXL.io vendor-defined message fields



CXL-compatible Memory Management

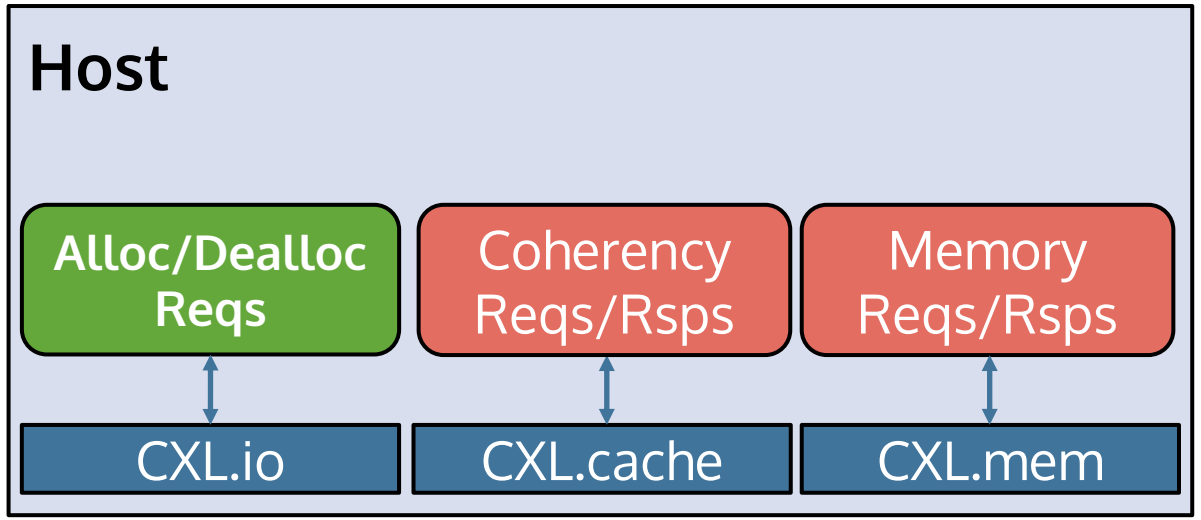
Q: How to allocate/deallocate pages from disaggregated memory?

Requirement

- Alloc/Dealloc messages **should not interfere with** RD/WR requests

Our Approach

- Define a new message using byte-15 of CXL.io vendor-defined message fields



CXL-compatible Memory Management

Q: How to allocate/deallocate pages from disaggregated memory?

Requirement

- Alloc/Dealloc messages **should not interfere with** RD/WR requests

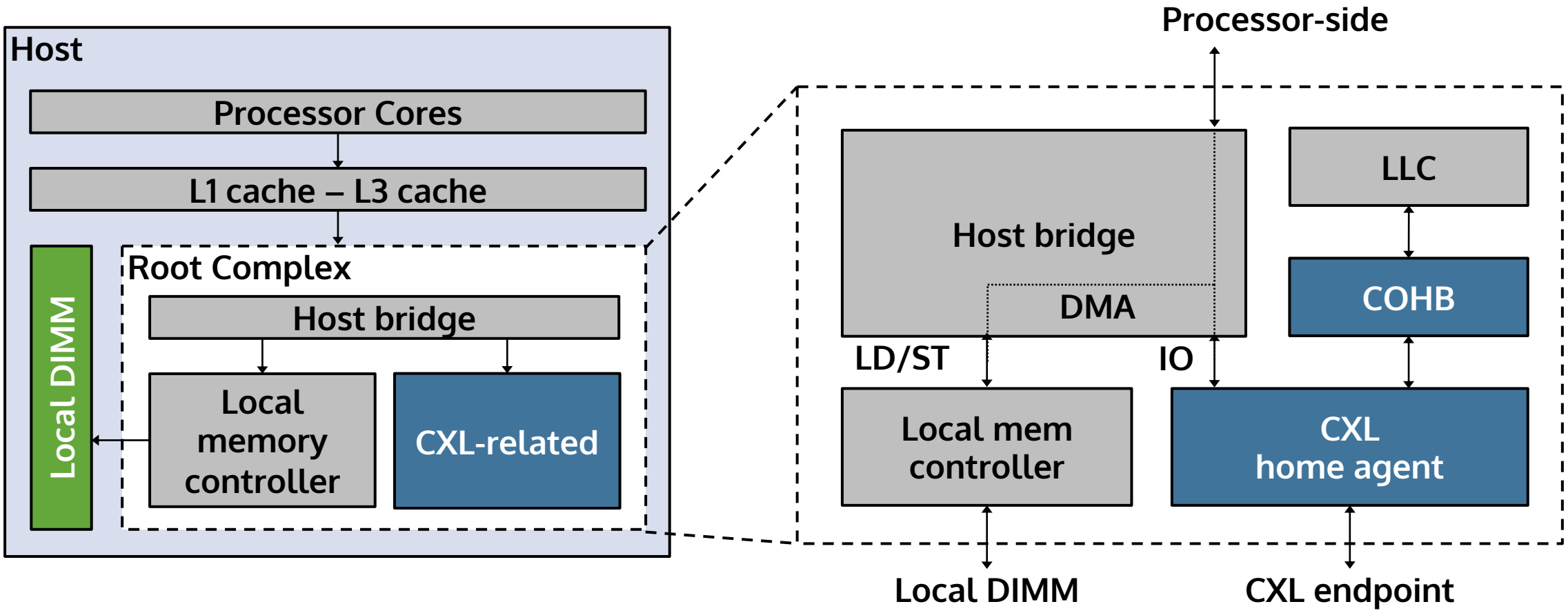
Our Approach

- Define a new message using byte-15 of CXL.io vendor-defined message fields

	+0		+1						+2					+3
Byte 0>	Fmt	Type	R	TC	R	Attr	LN	TH	TP	EP	Attr	AT	Length	
Byte 4>	Requestor ID								Tag					Message Code
Byte 8>	Reserved								**Vendor ID = CXL**					
Byte 12>	Reserved													**CXL VDM Code**

SDM Architecture

Host-side CXL-compatible hardware



SDM Architecture

Host-side CXL-compatible hardware

Coherence Bridge (COHB)

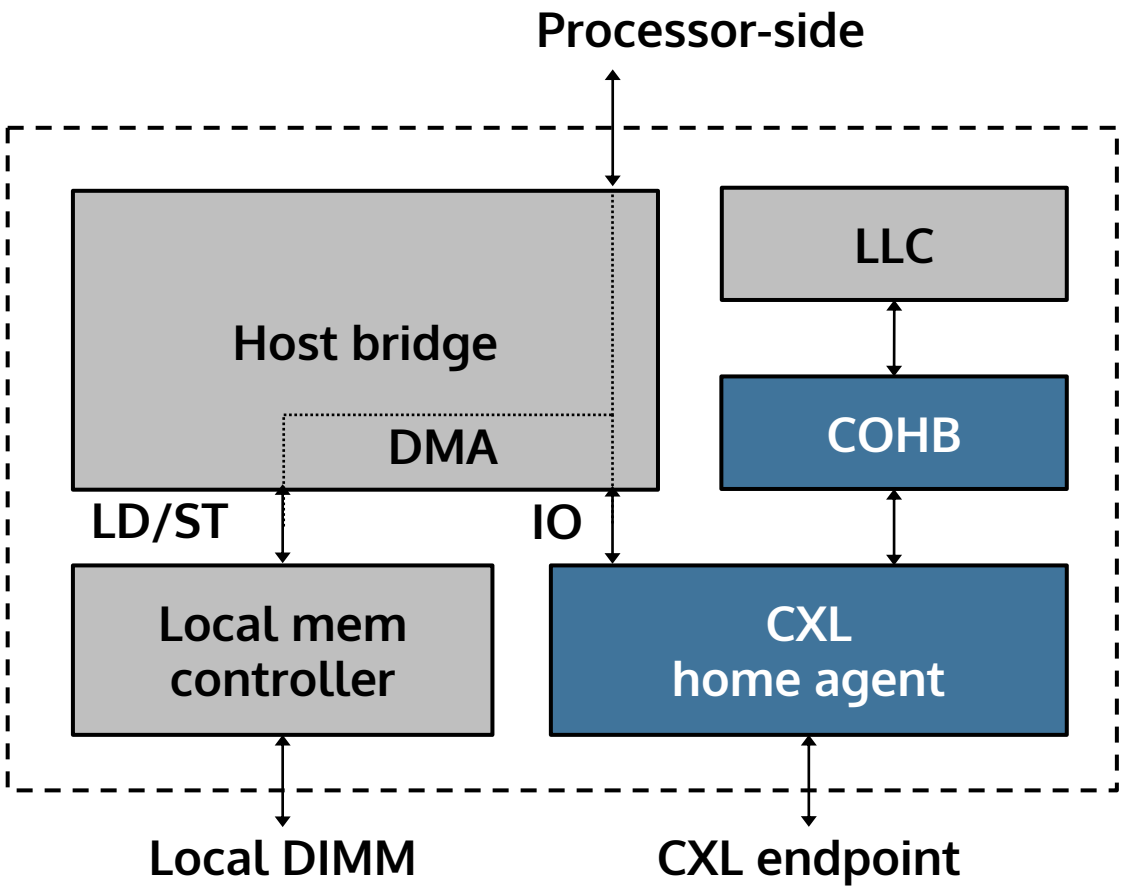
- Manage **system-level directory**

CXL Home Agent

- Generate **coherence messages**

Our Extension

- Generate allocate/deallocate messages



SDM Architecture

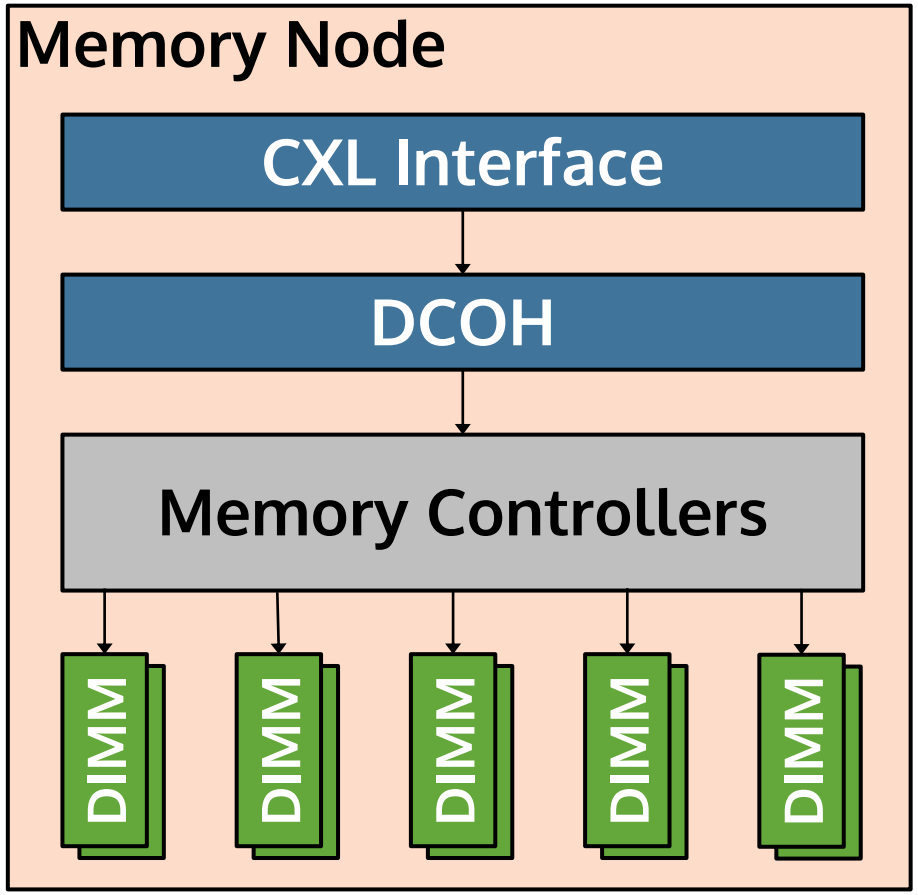
Device-side CXL-compatible hardware

Device Coherency Agent (DCOH)

- Generates **CXL.cache** messages
- Can have a **snoop filter**

Our Extension

- Send snoop messages <u>to abstracted hosts</u>



More Discussions in the Paper

- Snoop Emulation
- Sharing-enabled Control Flow
- Memory Management Mechanism
- SDM Architecture

- Address Translation Mechanism
  - How to implement it with CXL.io messages

- Speculative Access
  - How to overcome the overhead of access control check

Outline

- Introduction
- Motivation
- SDM: Sharing-enabled Disaggregated Memory System
  - CXL-compatible Designs
- Evaluation
- Conclusion

Methodology

Performance Evaluation

- In-house simulator using Intel PIN tool

Evaluated Workloads

- PARSEC (Compute-intensive)
- Intel GAP (Memory-intensive)

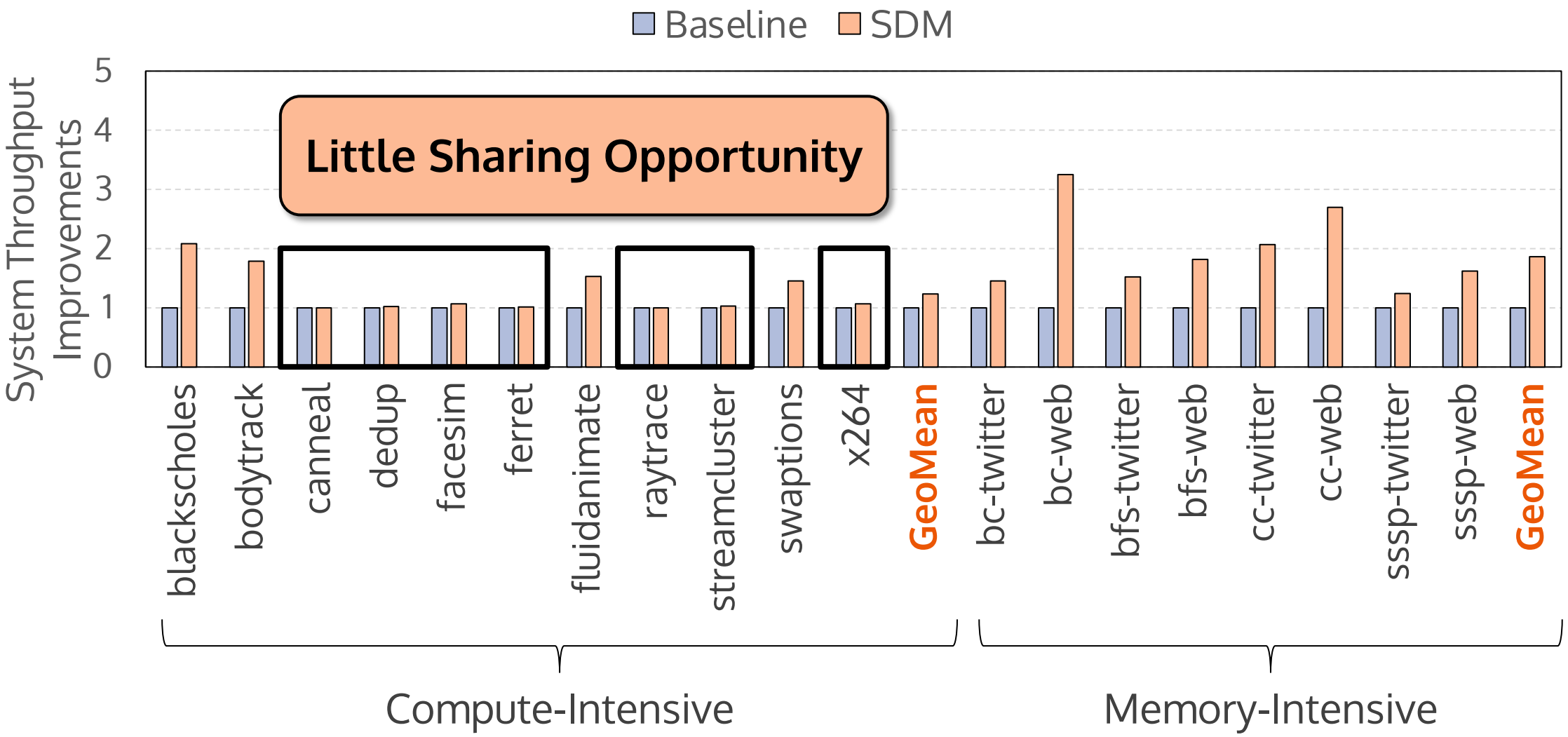
Baseline

- INV-CF: CXL 3.0-like invalidation-based control flow

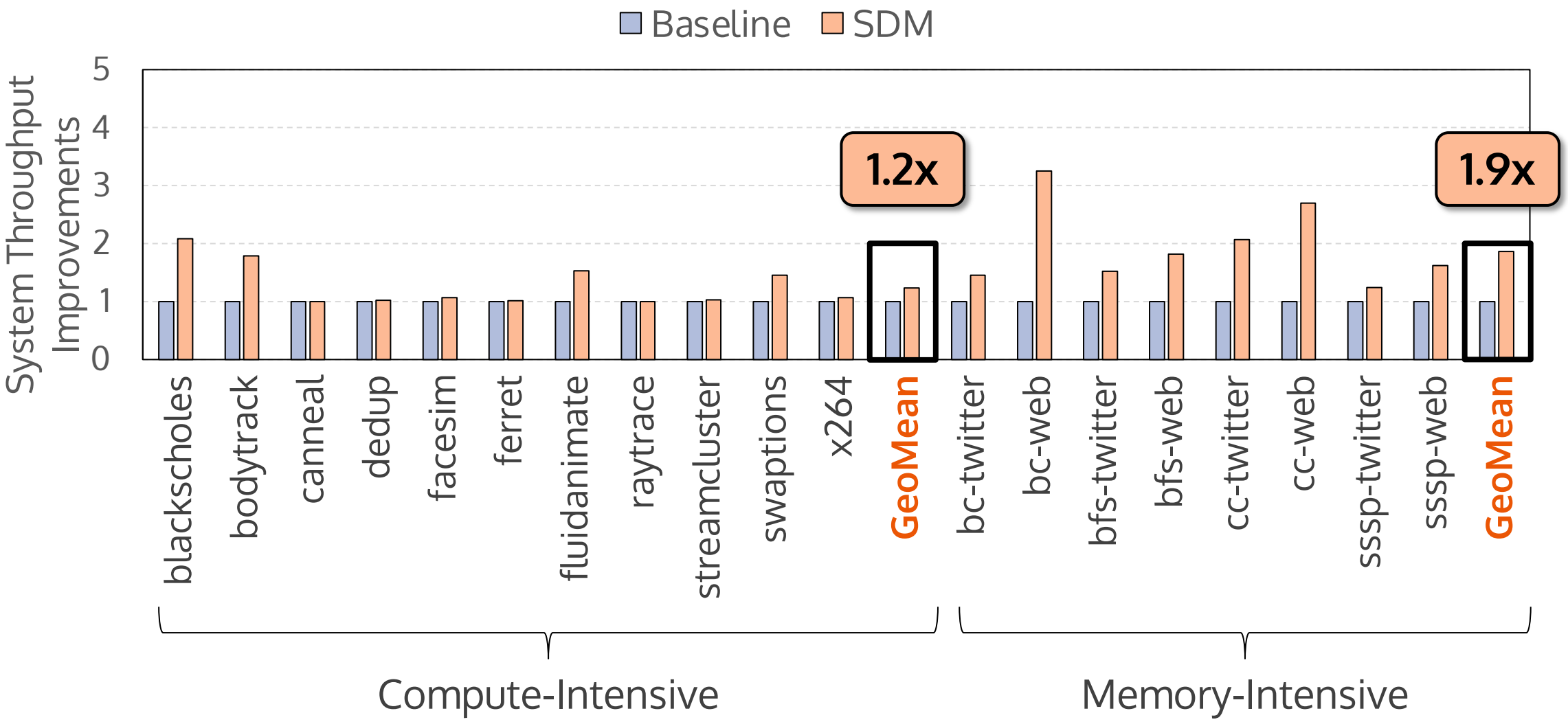
System Parameters

System	
Configuration	4 Compute Nodes 1 Memory Node
Compute Node	
Core	8 cores
L1 Cache	8-way, 32KB, 1ns
L2 Cache	4-way, 256KB, 4ns
L3 Cache	16-way, 2MB, 40ns
Memory Node	
Latency	80ns
Interconnect	
Latency	500ns

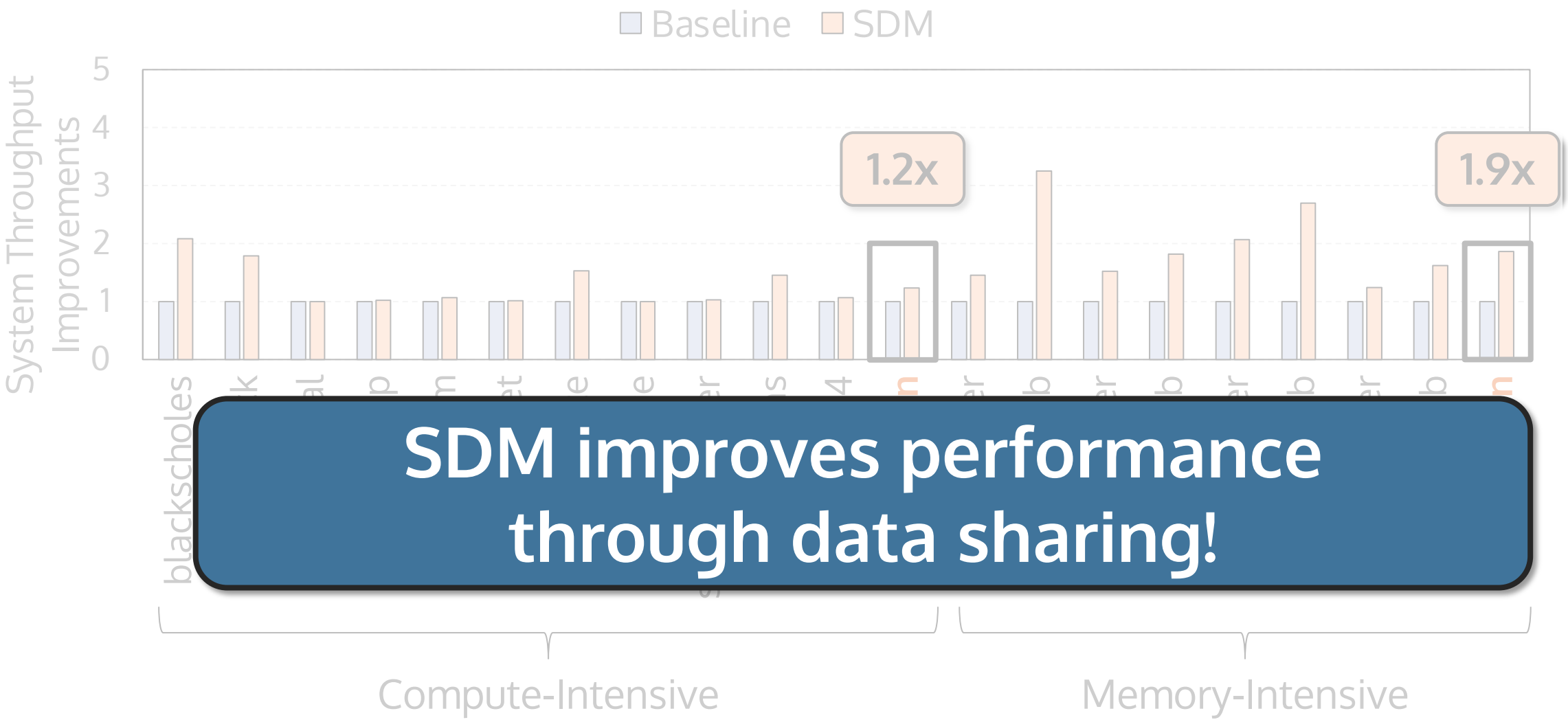
Performance



Performance

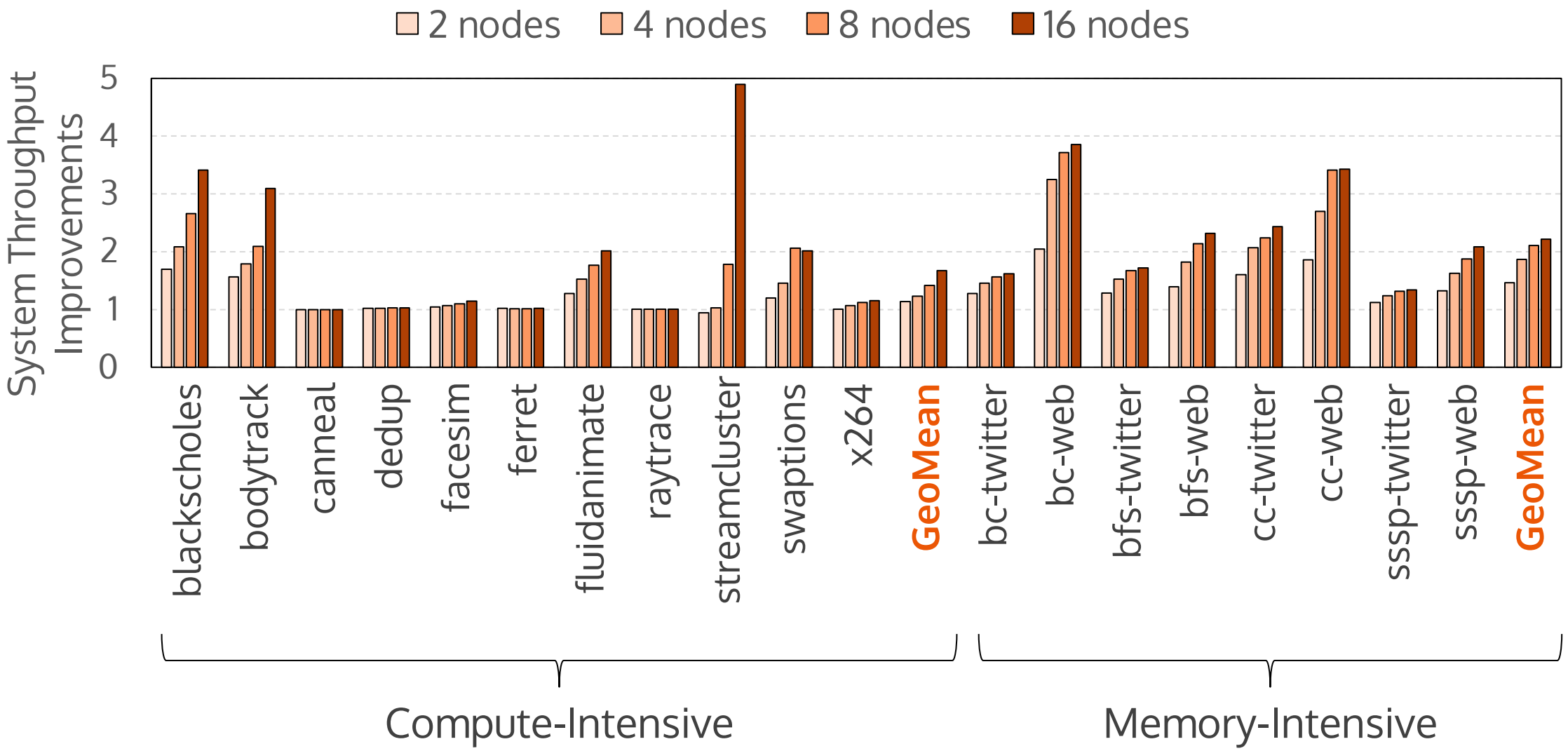


Performance

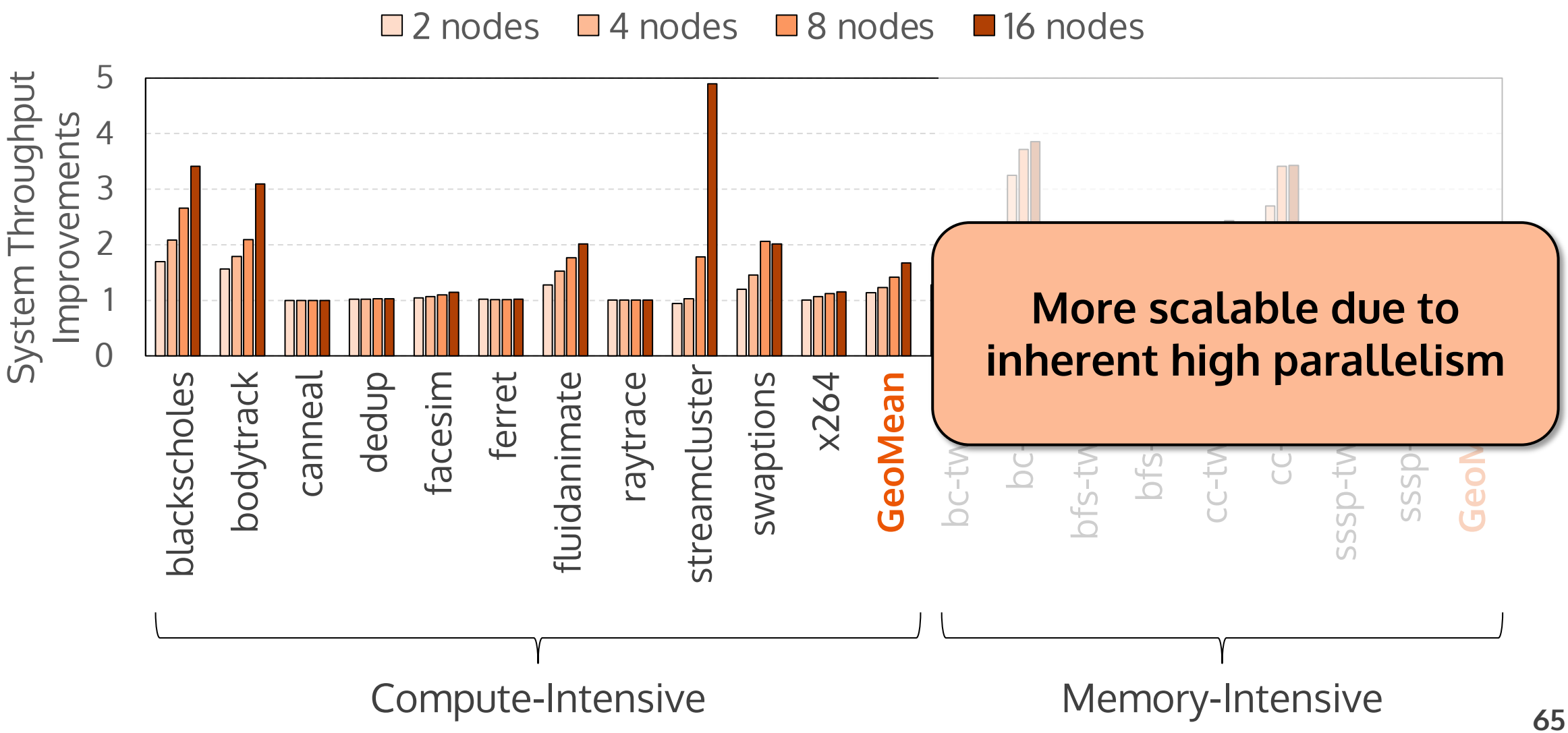


SDM improves performance through data sharing!

Scalability Analysis

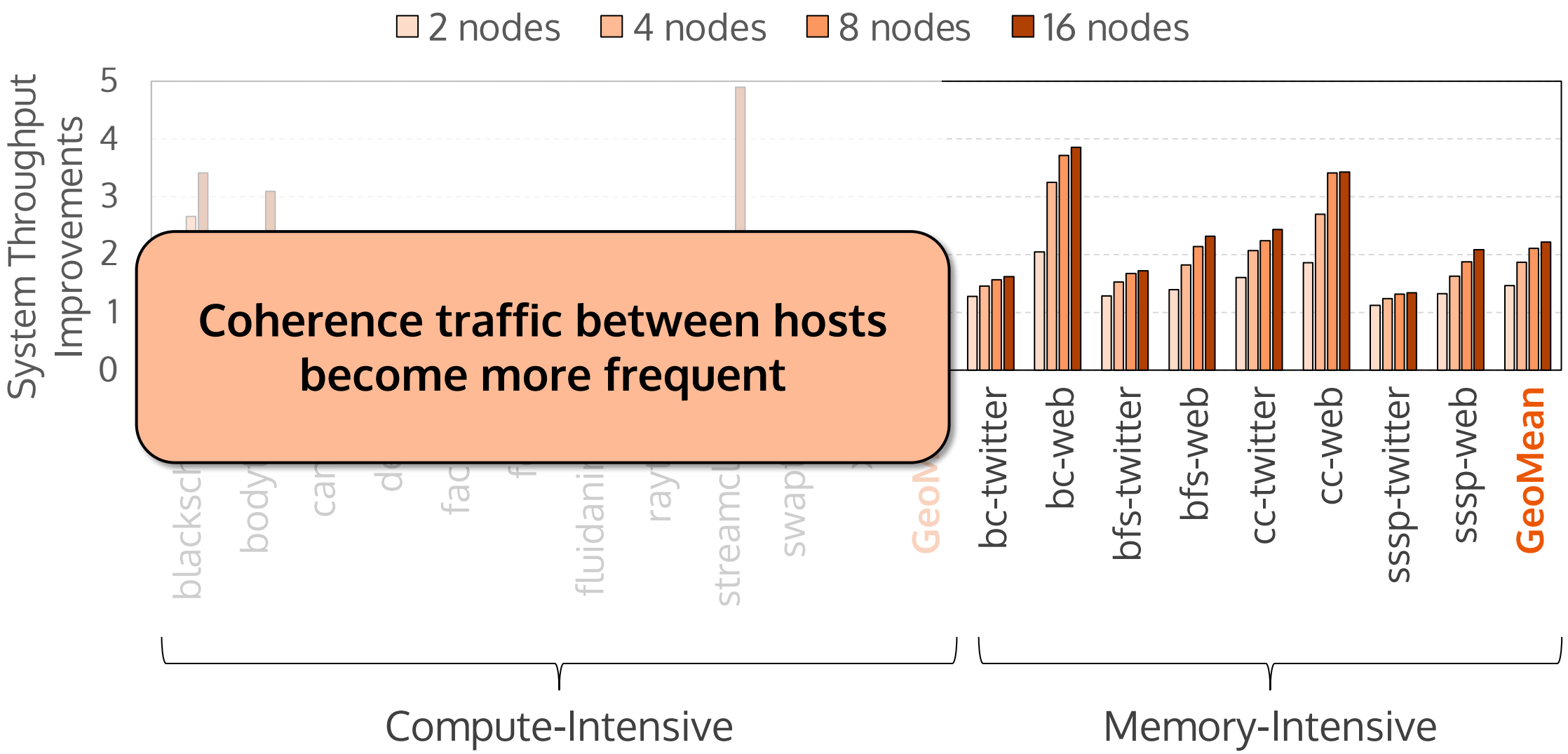


Scalability Analysis



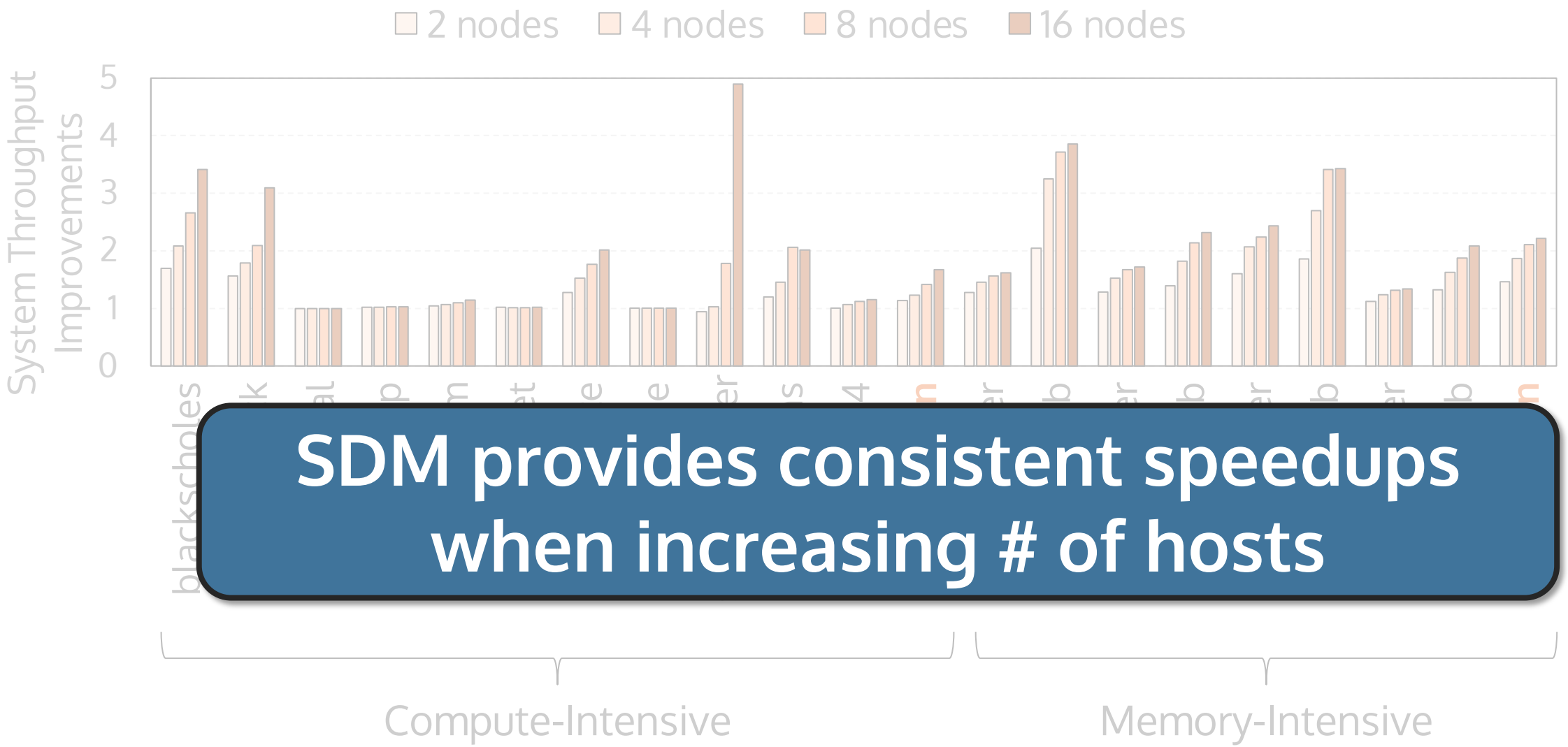
More scalable due to inherent high parallelism

Scalability Analysis



Coherence traffic between hosts become more frequent

Scalability Analysis



SDM provides consistent speedups when increasing # of hosts

Outline

- Introduction
- Motivation
- SDM: Sharing-enabled Disaggregated Memory System
  - CXL-compatible Designs
- Evaluation
- Conclusion

Conclusion

Goal

- Design a **CXL-compatible**, **Sharing-enabled** Disaggregated Memory System

Solution

- **Snoop Emulation** enables multi-host coherence management
- **SHA-CF** enables data sharing between multiple hosts

Result

- SDM achieves an average of **1.5x speedup** over naïve CXL-based disaggregated memory systems

Thank You!