

清華大學 交叉信息研究院
Institute for Interdisciplinary Information Sciences, Tsinghua University


上海期智研究院
SHANGHAI QI ZHI INSTITUTE


上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

NDPBridge: Enabling Cross-Bank Coordination in Near-DRAM-Bank Processing Architectures

Boyu Tian, Yiwei Li, Li Jiang, Shuangyu Cai, Mingyu Gao

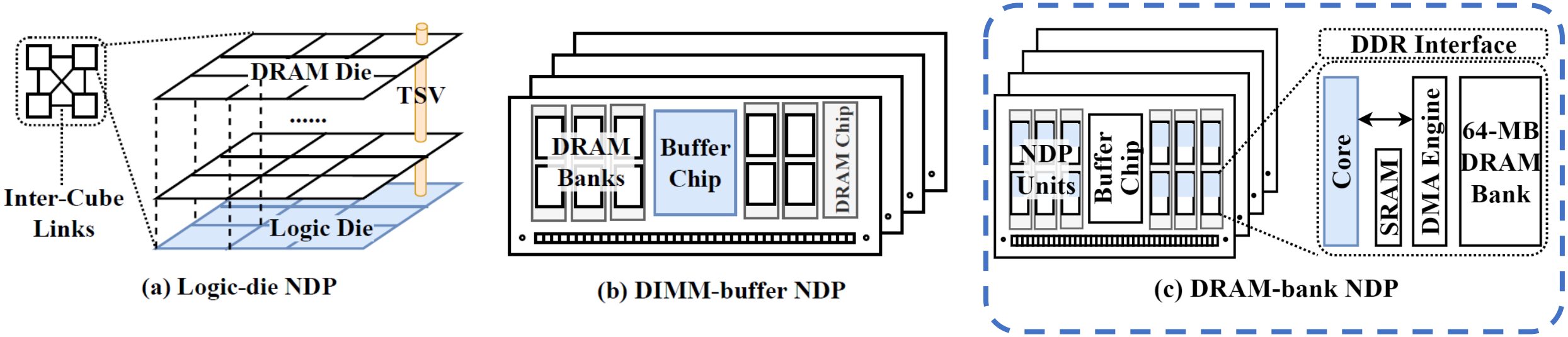
Tsinghua University
Shanghai Qi Zhi Institute
Shanghai Jiao Tong University
Huawei Technologies Co., Ltd.


清華大學
Tsinghua University

ISCA 2024

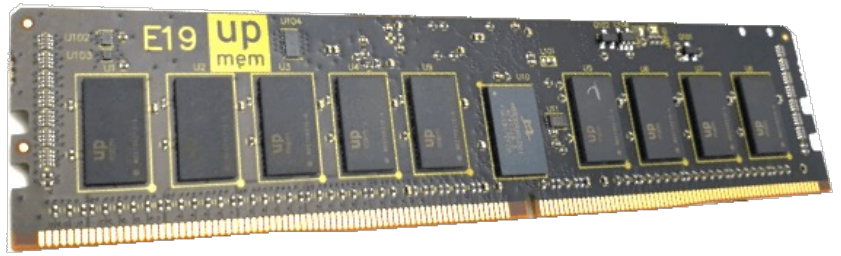
Near-Data Processing (NDP)

- Near-data processing (NDP): place compute logic near data memory
  - Shorter distance → lower latency and energy
  - Higher bandwidth
- Various memory technologies to realize NDP:

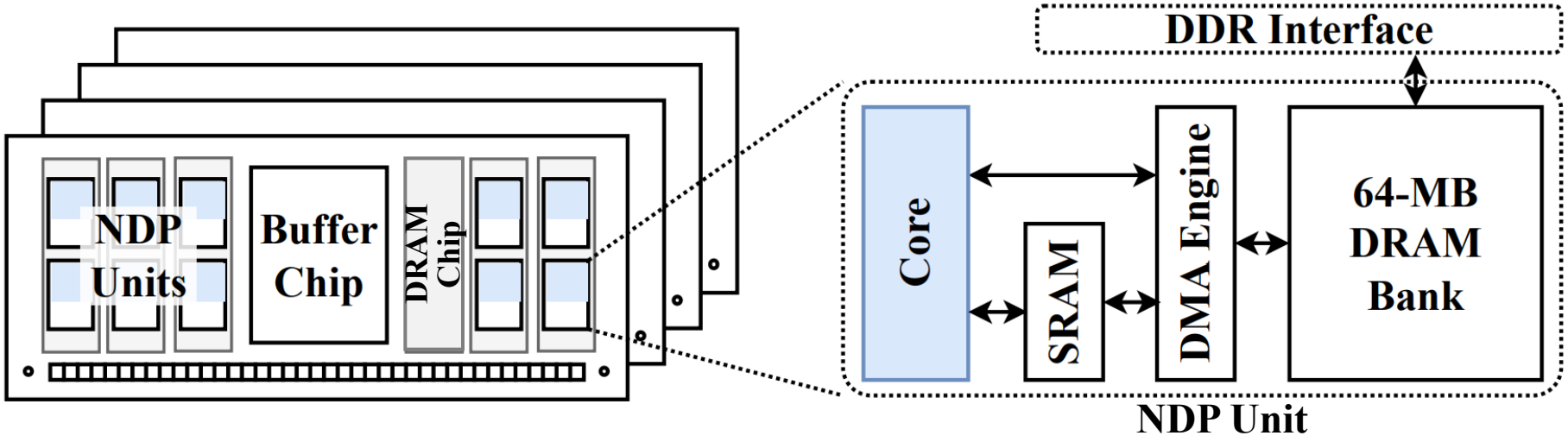


DRAM-Bank NDP Systems

- Add computing logics inside/near DDR banks
- Fine granularity, high bandwidth, high parallelism
  - Thousands of units
- Typical commercial products:
  - UPMEM, Samsung's HBM-PIM, SK Hynix's AiM



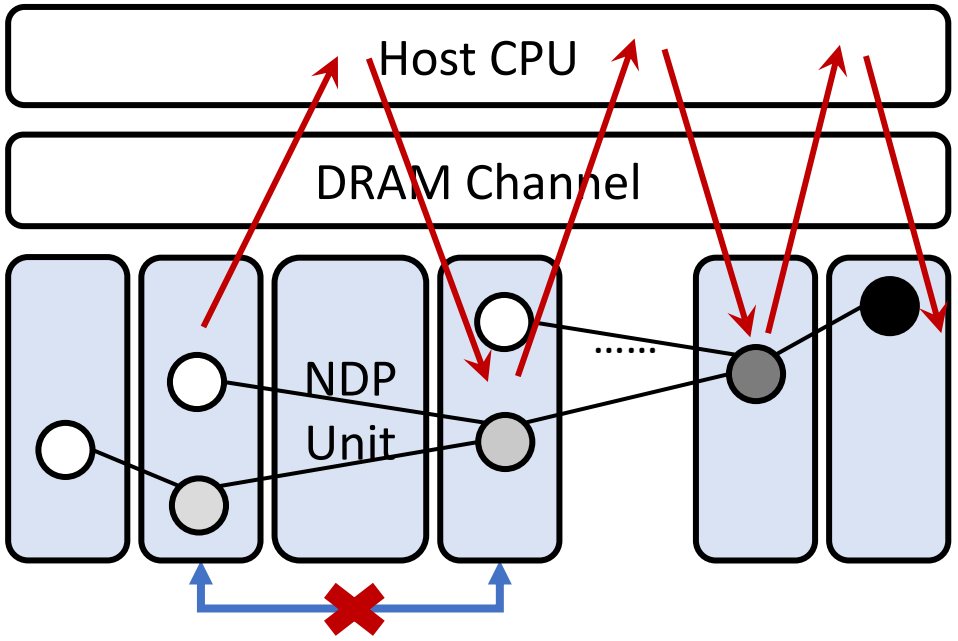
UPMEM Chip



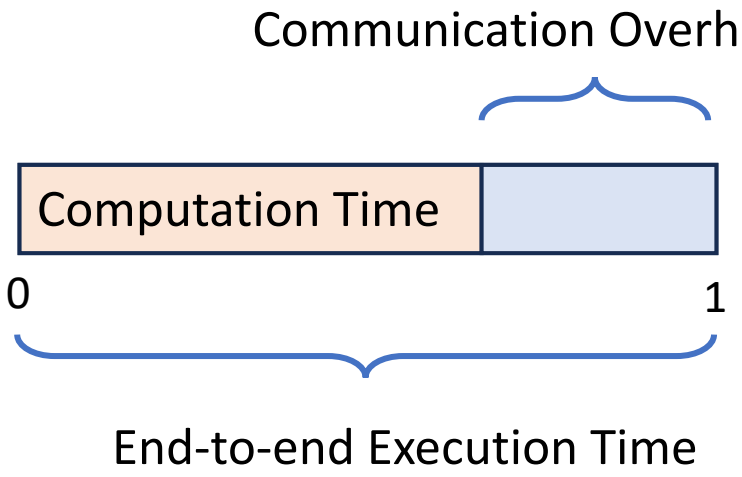
(c) DRAM-bank NDP

Limitation 1: Lack of Communication Support

- Different DRAM banks cannot communicate directly.
- Applications of DRAM-Bank NDP follows **data-local execution paradigm**.
- Communication is done through expensive host CPU forwarding.
- Adding physical links between banks is prohibitively expensive.

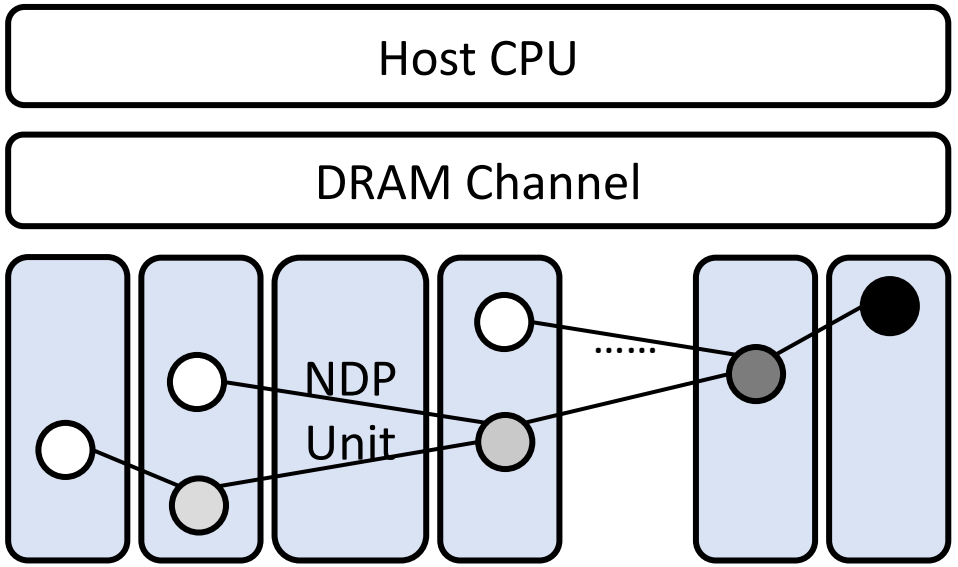


Example: Tree-Traversal

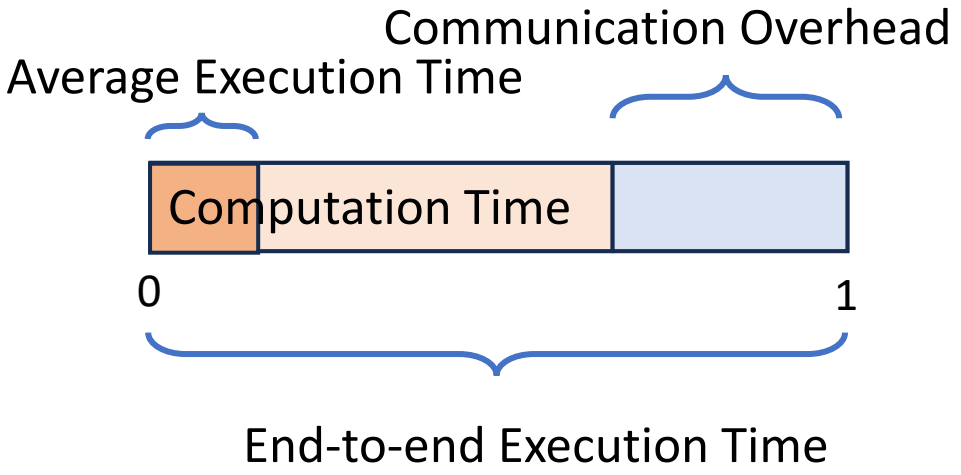


Limitation 2: Load Imbalance

- Thousands of NDP cores in DRAM-Bank NDP.
- Static assignment cannot suit applications generating tasks dynamically.
- Dynamic load balancing is not enabled, due to lack of communication.
- The data-local execution makes the scheduling more complex.



Example: Tree-Traversal



Our Contributions

- Enabling cross-bank communication without altering DRAM form factors.
- Enabling cross-bank load balancing compatible with communication.

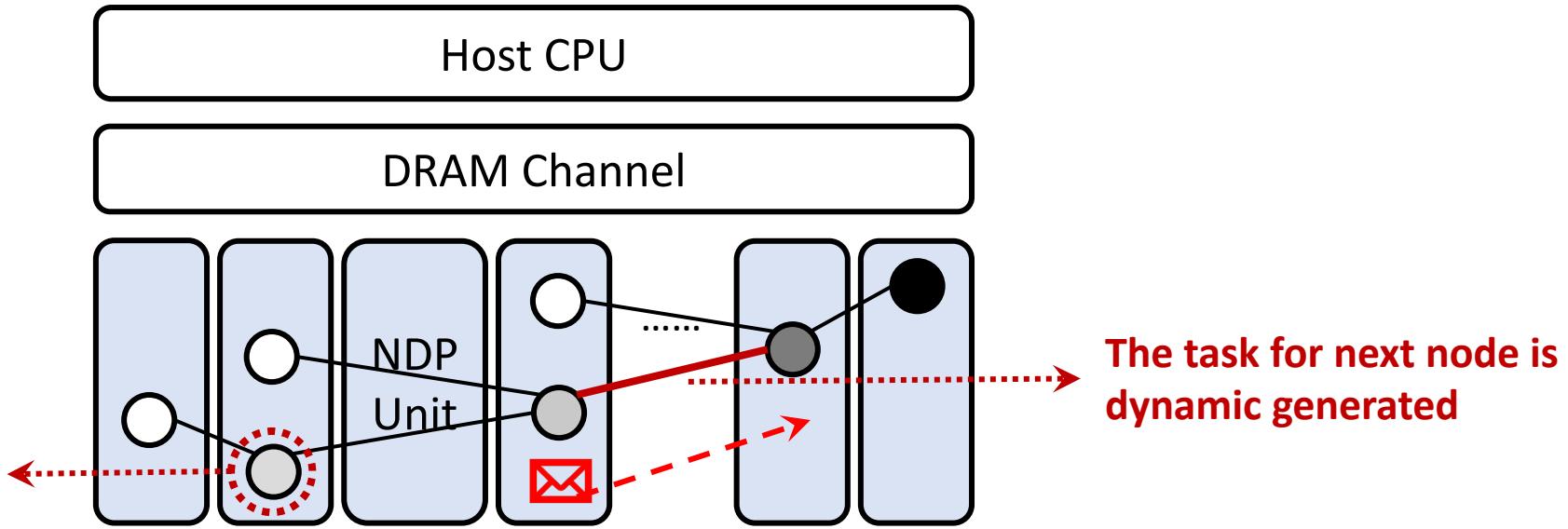
1. Task-based message-passing programming model

2. Cross-bank communication scheme using “**bridges**”.

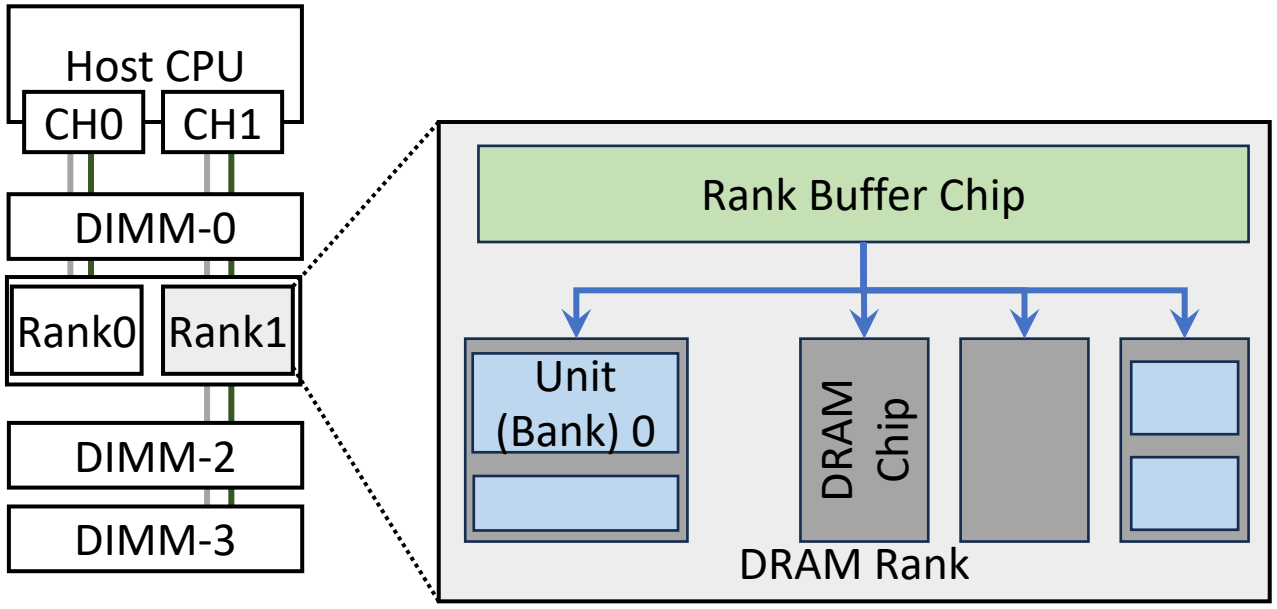
3. Data-transfer-aware scheduling policy.

Task-Based Programming Model

- A task is the basic unit for execution and scheduling.
- Tasks spawn child tasks dynamically.
- Each task is associated with one data element.
- Communication is done through pushing tasks by message passing instead of pulling data.

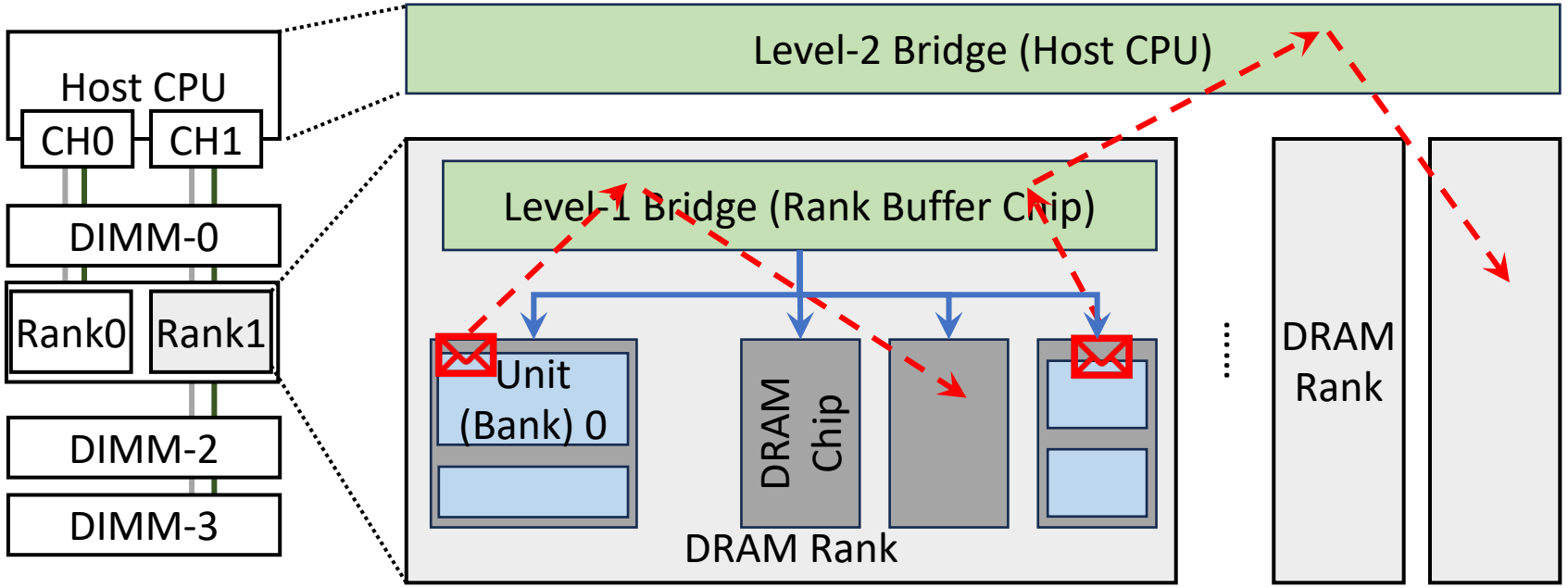


NDPBridge Overview

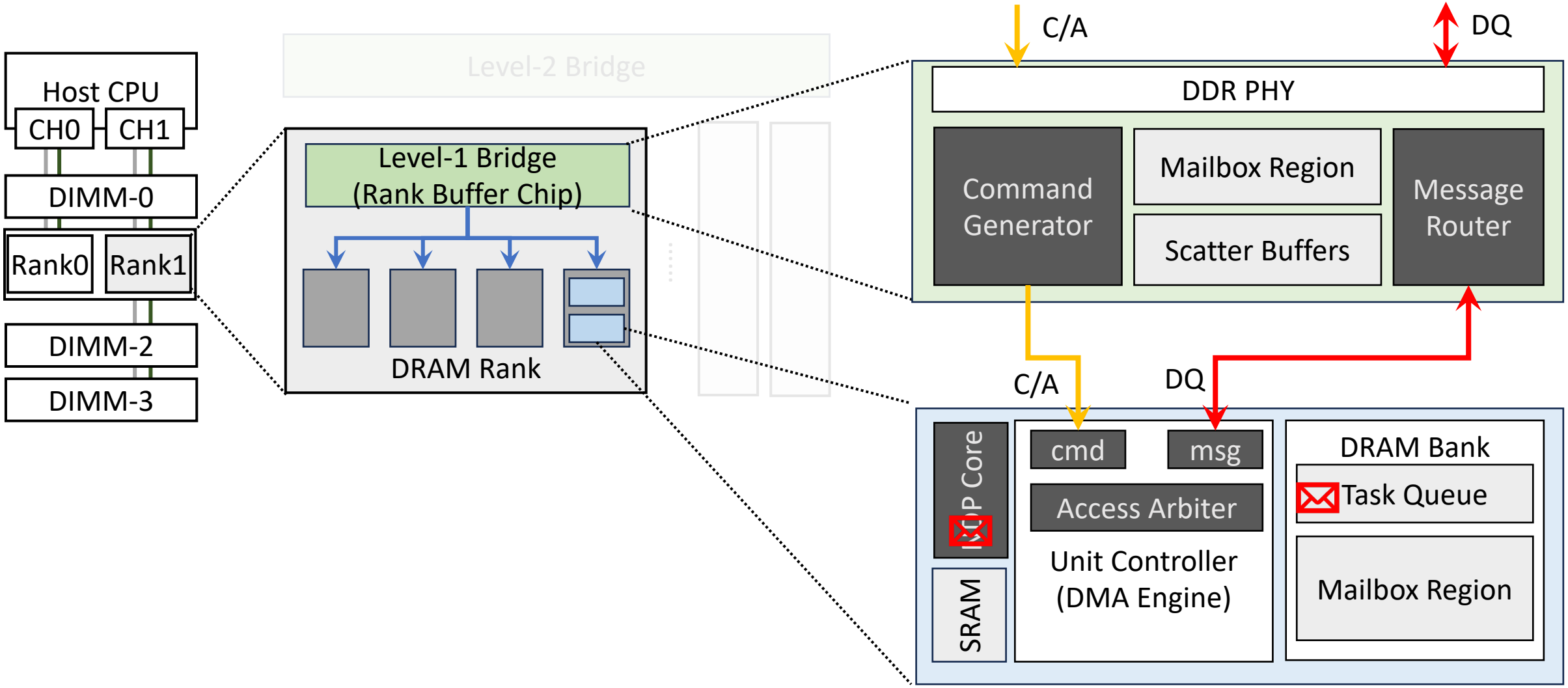


NDPBridge Overview

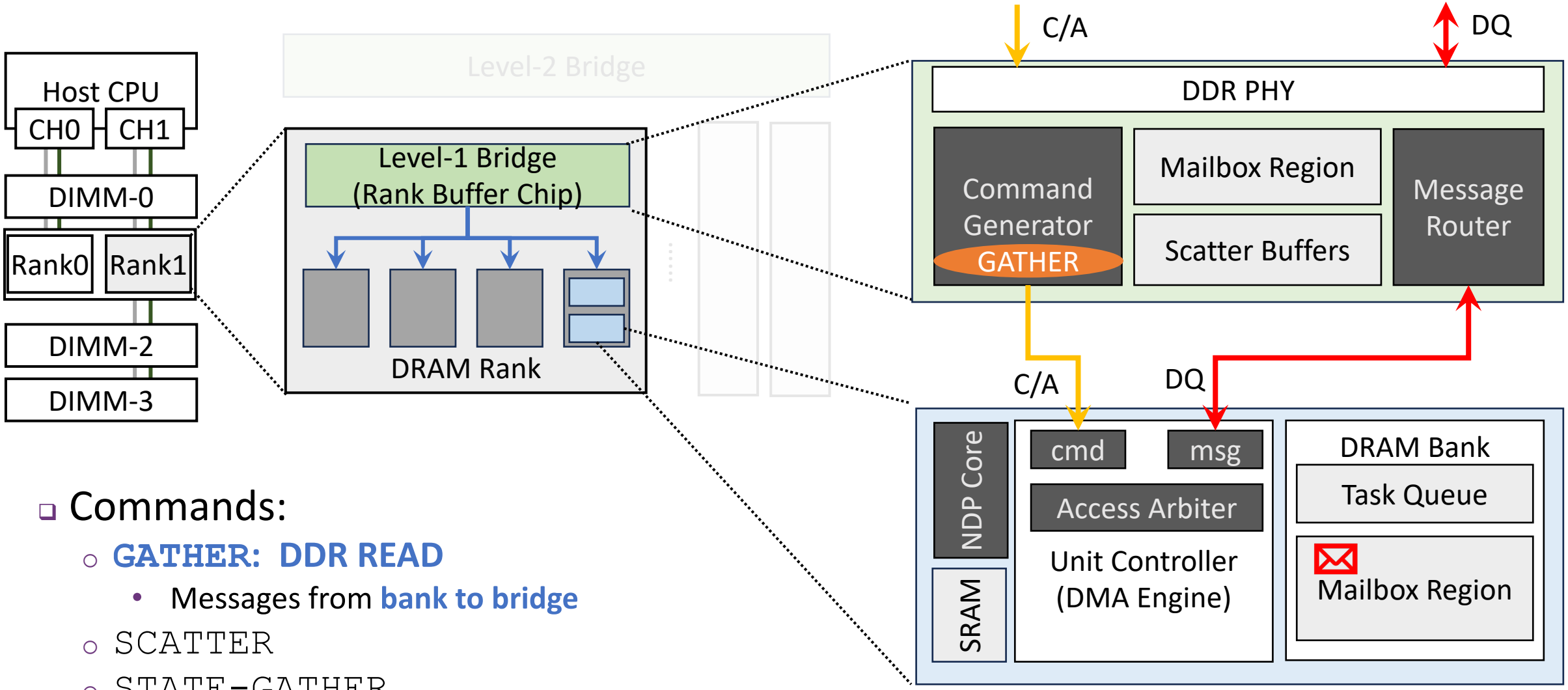
- Idea: add **bridges** into each level of the memory hierarchy
  - Bridges gather/scatter messages from child node mailboxes
  - Existing physical links and DDR commands
  - All modifications are within standalone modules



Bridge-Based Communication

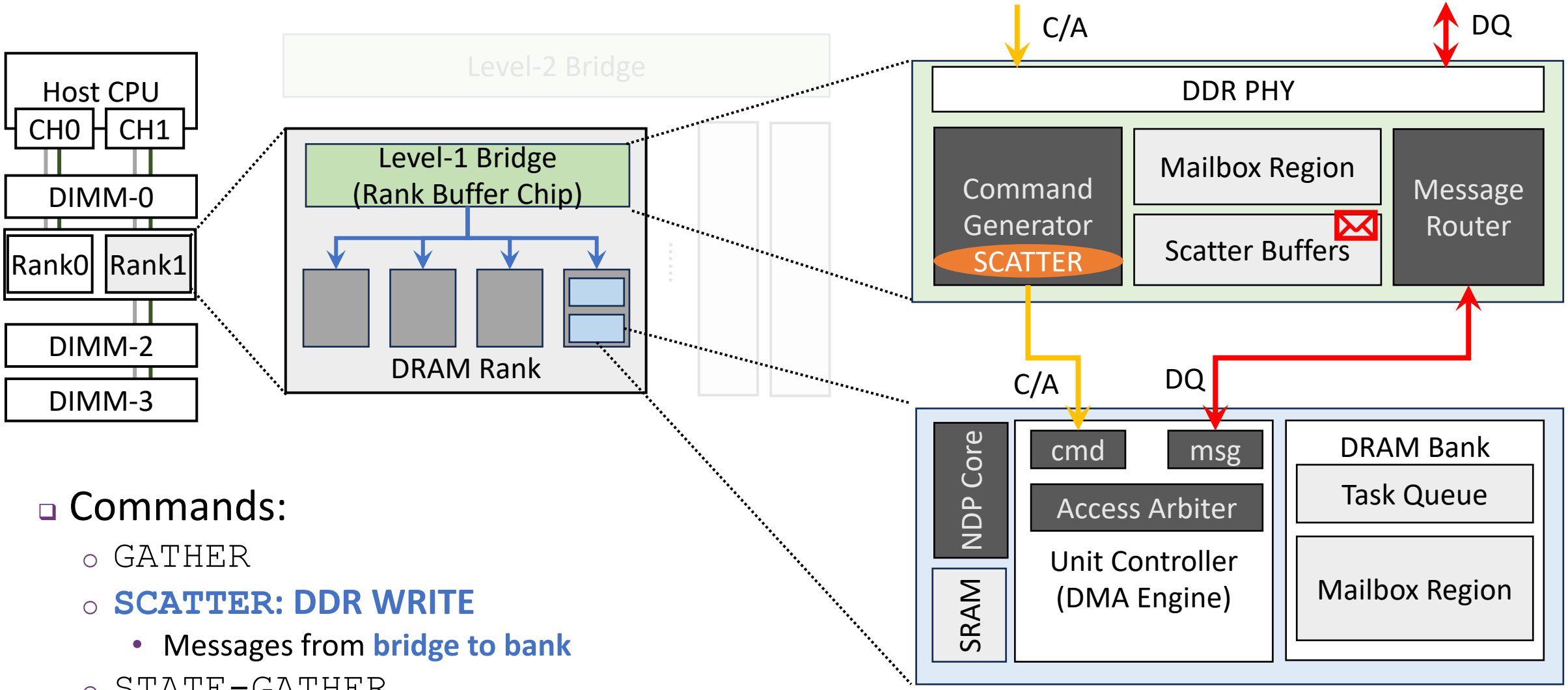


Bridge-Based Communication



- Commands:
  - **GATHER: DDR READ**
    - Messages from **bank to bridge**
  - SCATTER
  - STATE-GATHER
  - SCHEDULE

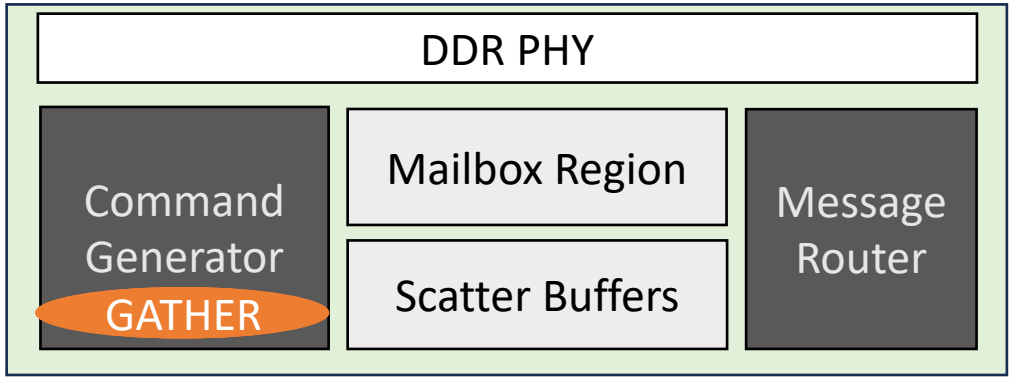
Bridge-Based Communication



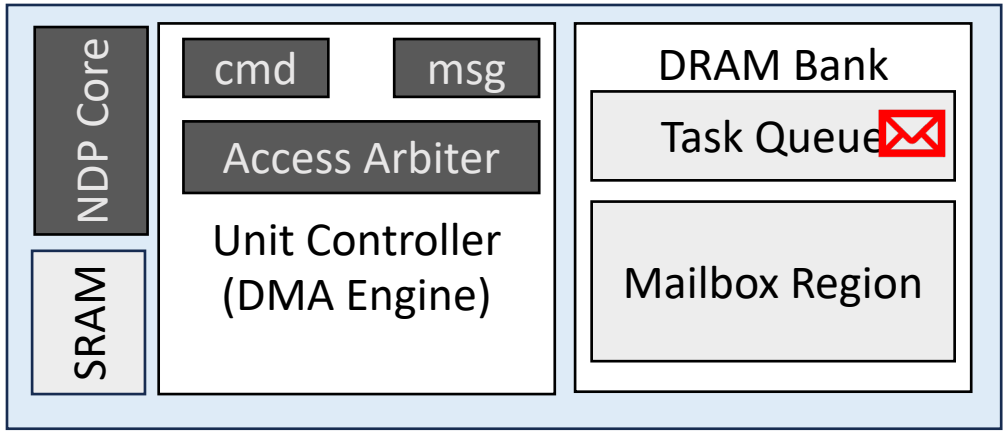
- Commands:
  - GATHER
  - **SCATTER: DDR WRITE**
    - Messages from **bridge to bank**
  - STATE-GATHER
  - SCHEDULE

Bridge-Based Load Balancing

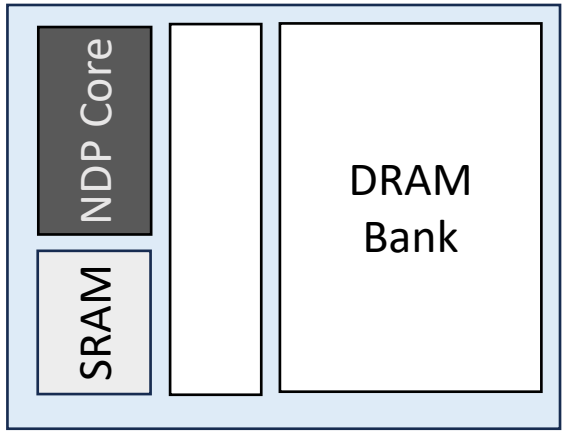
- Bridge commands scheduling
- Unit prepare tasks
- Bridge gathers tasks
- Bridge assigns & dispatches tasks



- Commands:
  - GATHER
  - SCATTER
  - STATE-GATHER
  - **SCHEDULE: DDR ACTIVATE**
    - Messages: **GATHER (bank->bridge) + SCATTER (bridge->bank)**



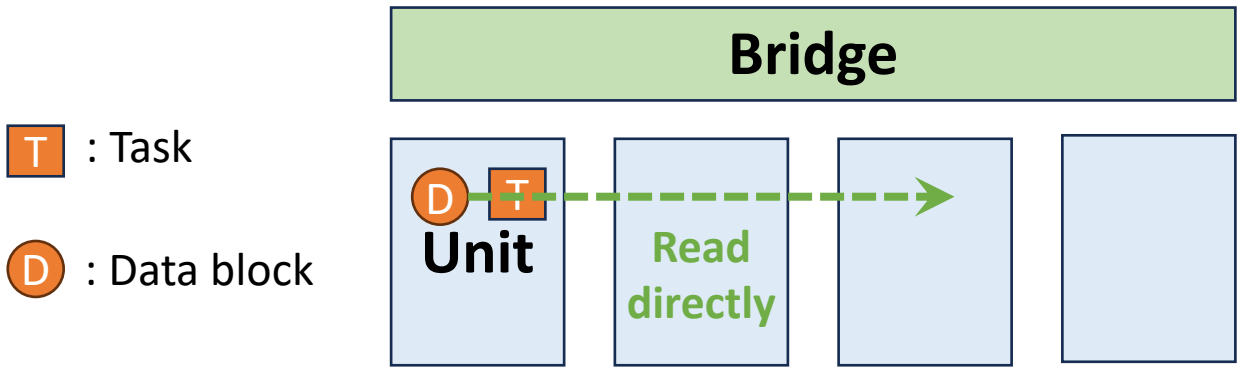
Giver



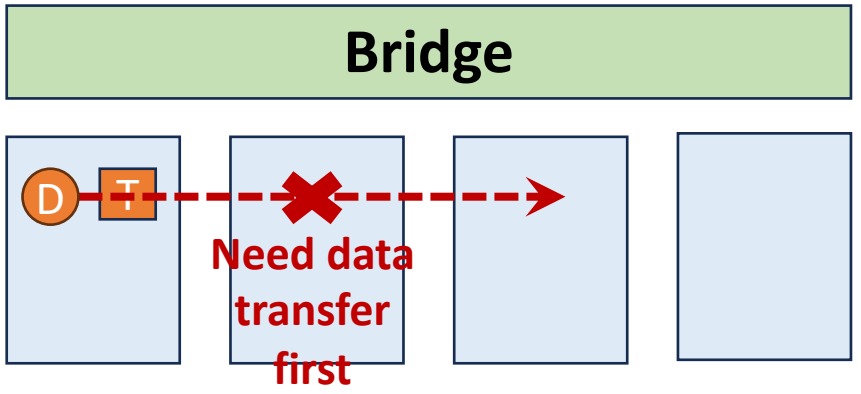
Receiver

Load Balance: Data-First Scheduling Problem

- **Data-first scheduling** problem
  - Must move data to tasks



Shared-Memory Architecture

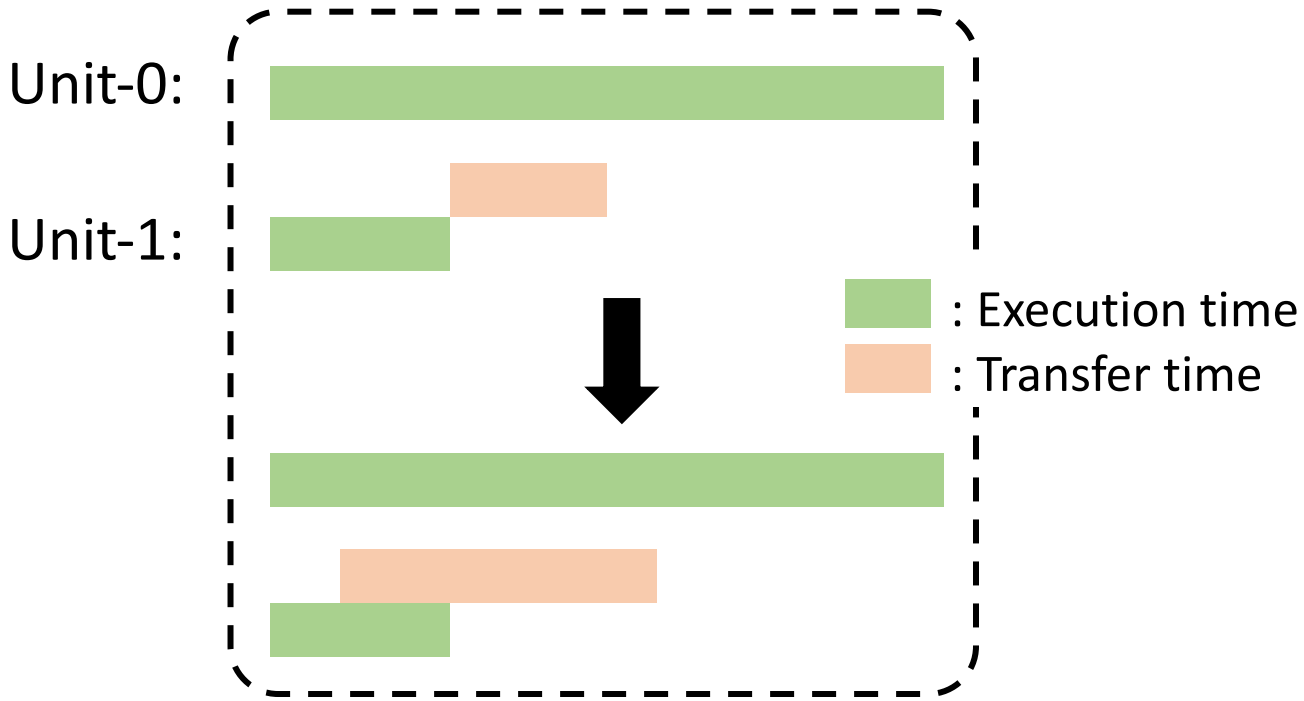


NDP Architecture with Data-Local Execution Paradigm

- Data transfer takes time
- We need data-transfer-aware load balancing

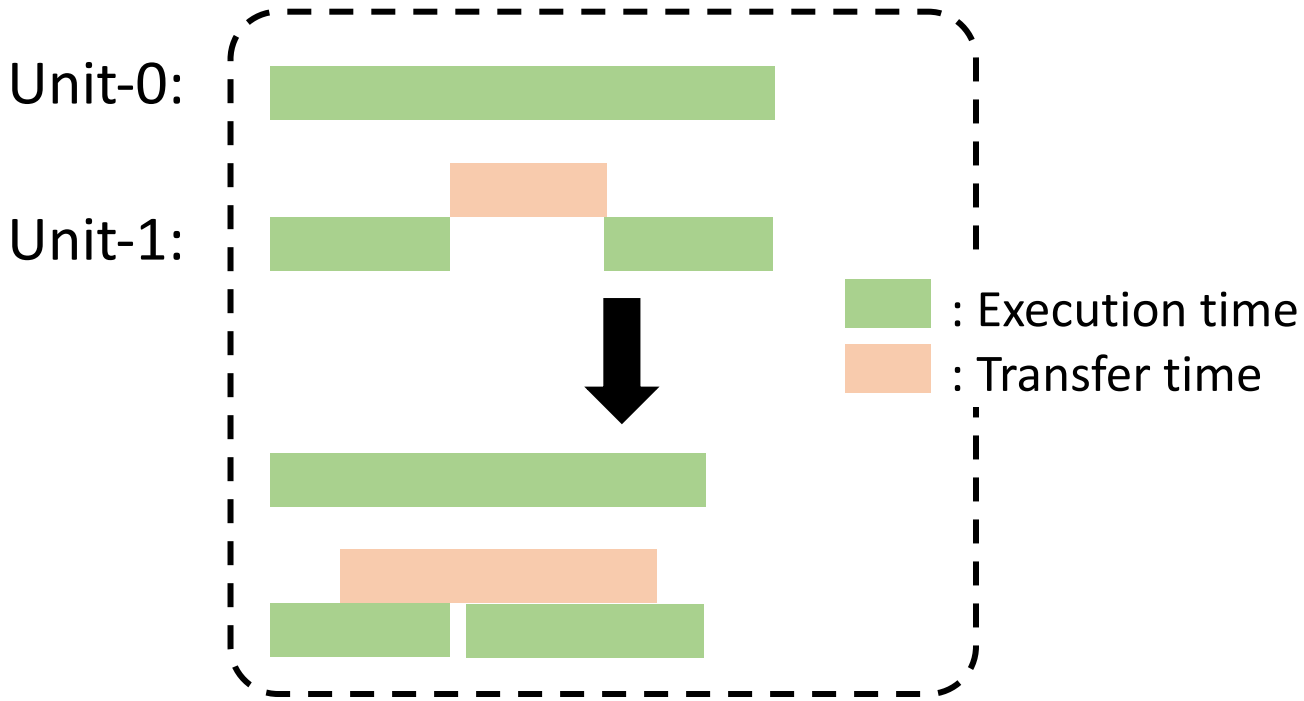
Load Balance: Data-Transfer-Aware Scheduling

- **Hide transfer latency:**
  - Traditional scheduling steal tasks when local queue is empty
  - Schedule tasks in advance
  - Overlap the transfer latency

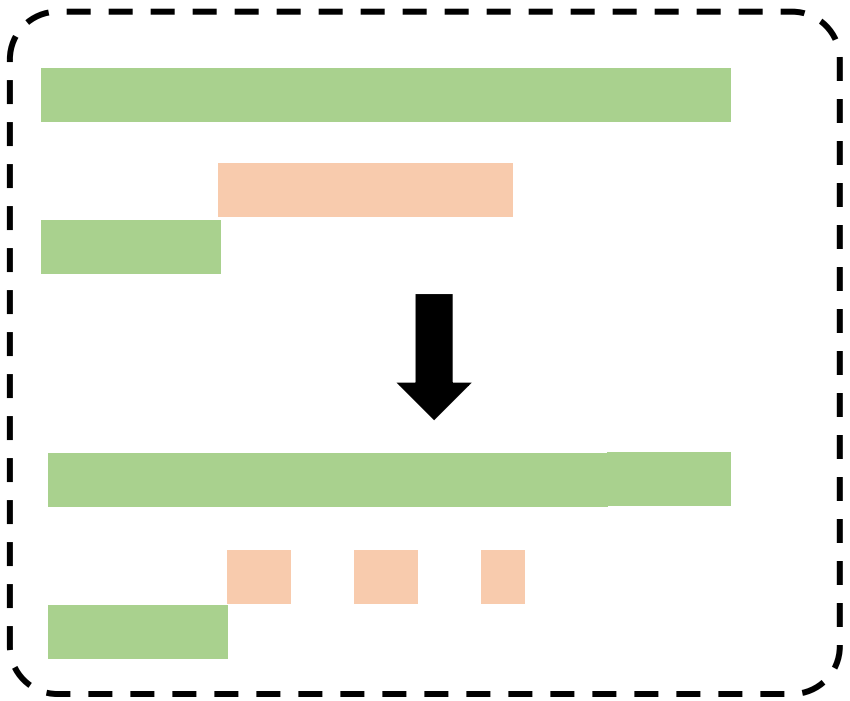


Load Balance: Data-Transfer-Aware Scheduling

- **Hide transfer latency:**
  - Traditional scheduling steal tasks when local queue is empty
  - Schedule tasks in advance
  - Overlap the transfer latency

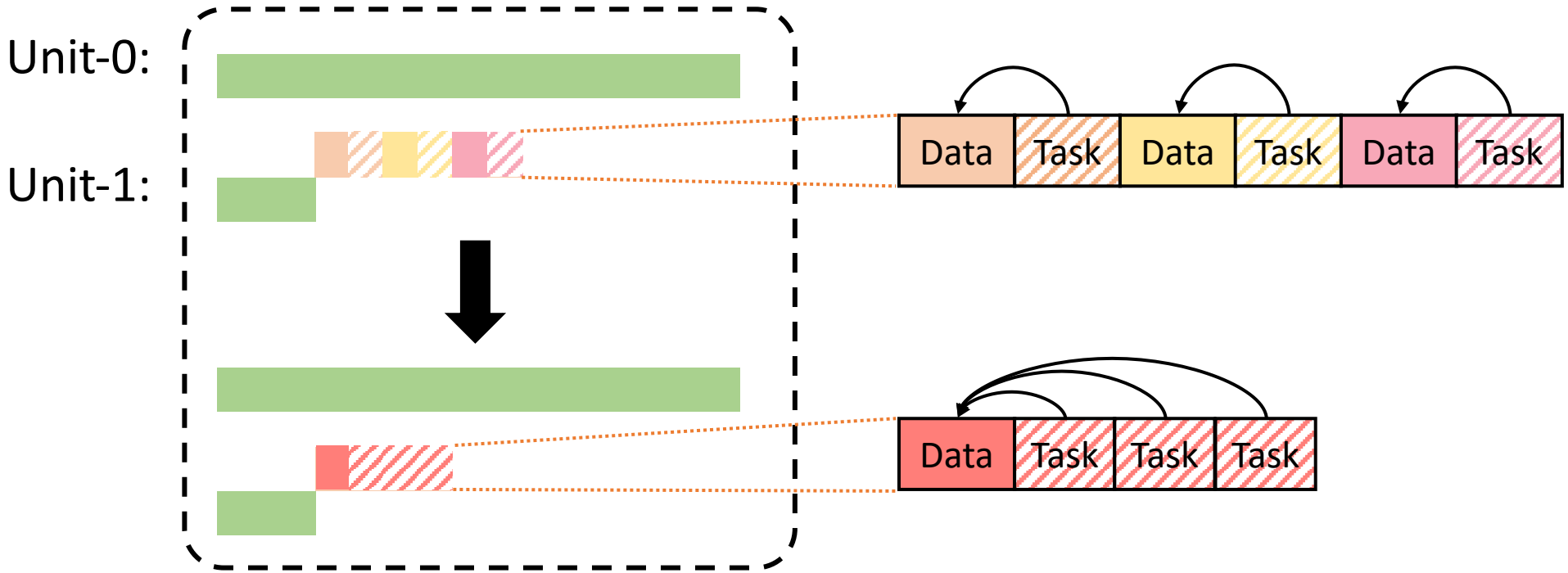


- **Avoid transfer congestion:**
  - Traditional work-stealing: steal half of the victim queue
  - Fine-grained scheduling



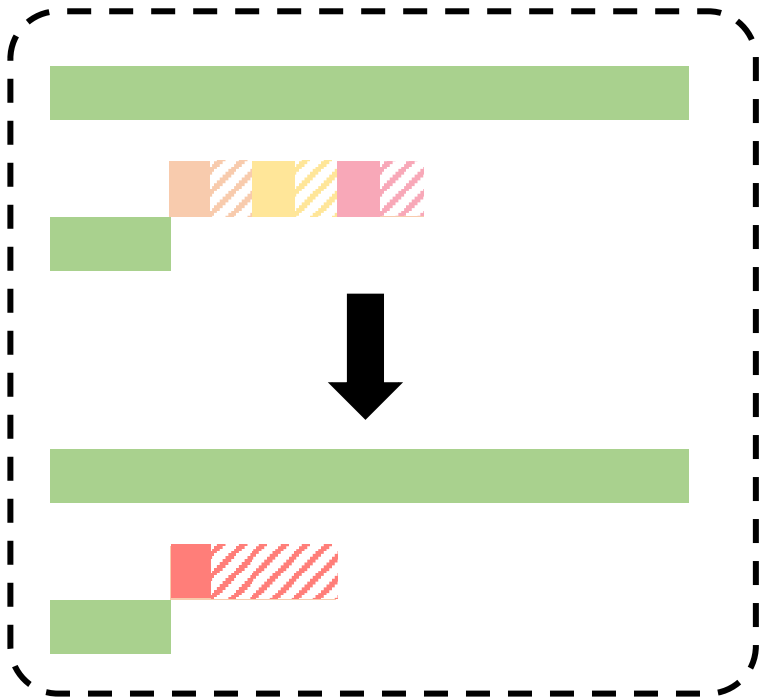
Load Balance: Data-Transfer-Aware Scheduling

- **Reduce transfer traffic:**
  - Traditional work stealing steals tasks from task queue tail
  - Scheduling hot data can reduce data traffic.

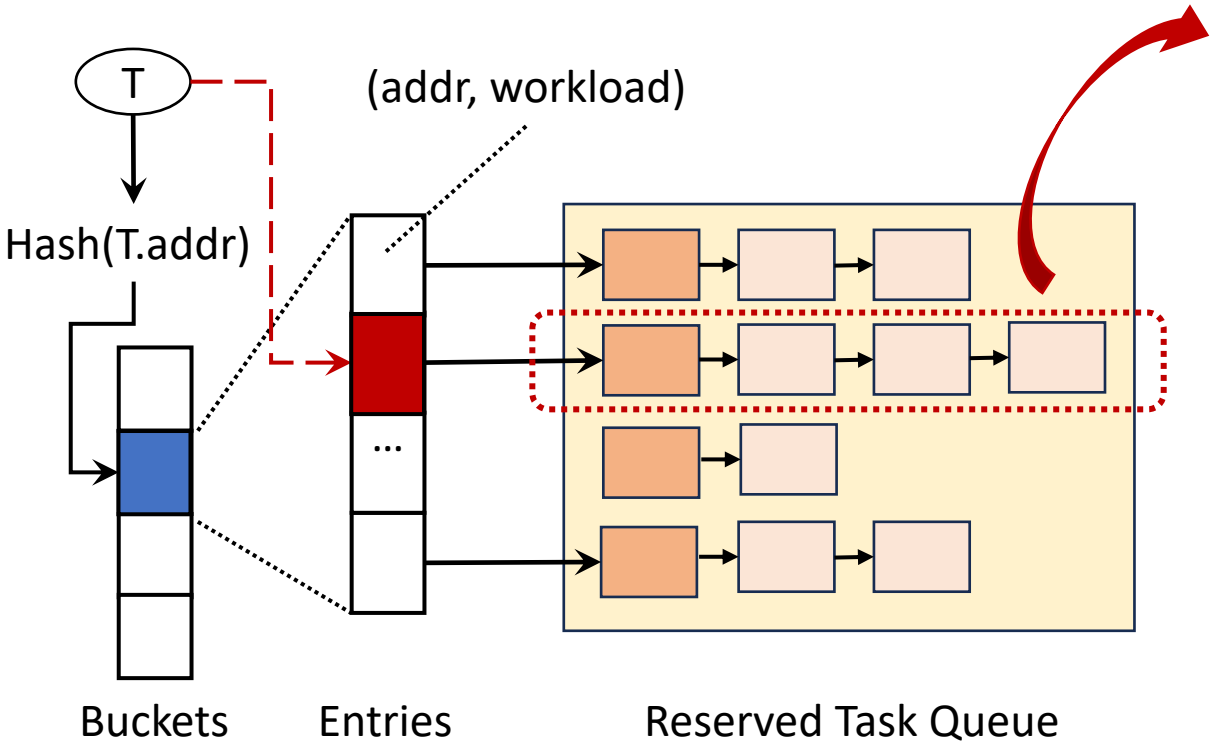


Load Balance: Data-Transfer-Aware Scheduling

- **Reduce transfer traffic:**
  - Traditional work stealing steals tasks from task queue tail
  - Scheduling hot data can reduce data traffic.



  - We use sketch to filter hot data
    - Similar to HeavyGuardian [KDD' 18]
    - Tasks of hot data are stored separately
    - Storage overhead: 2.2 KB in SRAM



Methodology

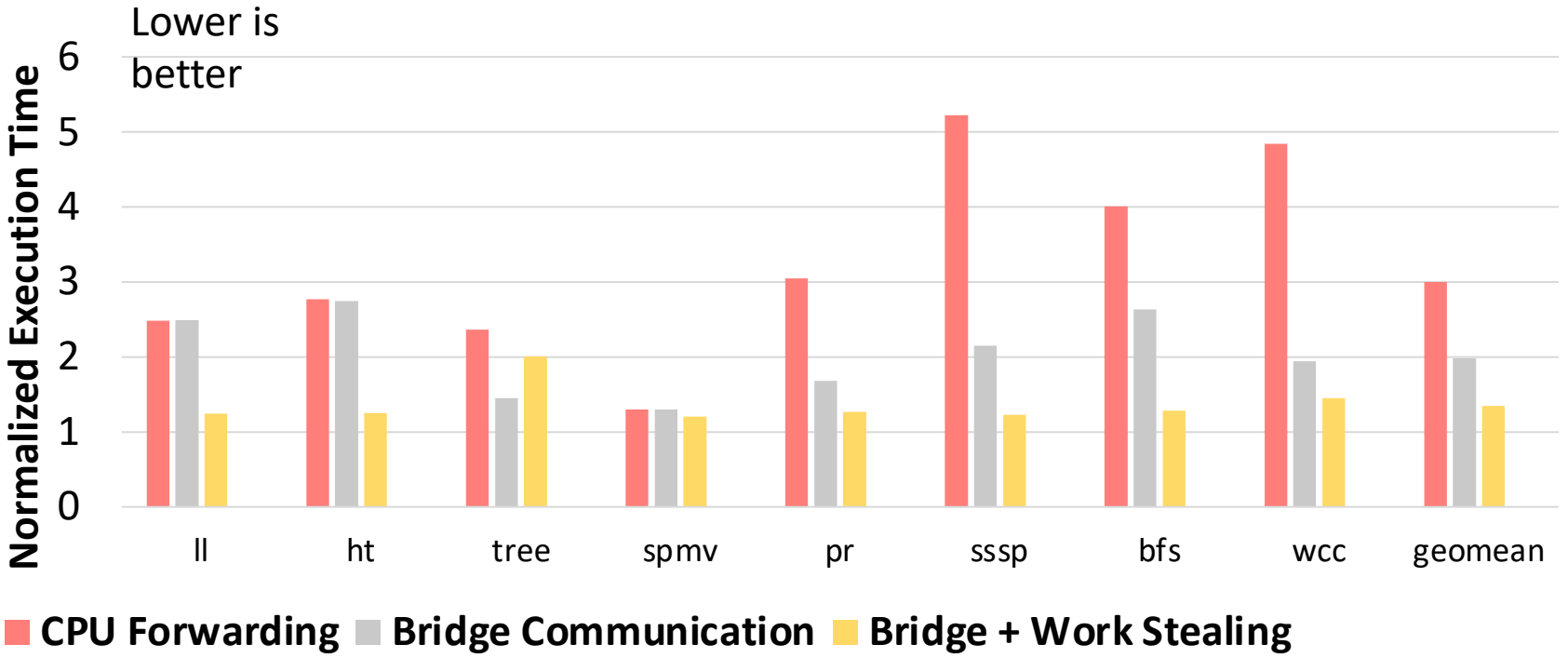
- Simulated platform
  - 2 channels × 4 ranks/ch × 8 chips × 8 banks, 512 units in total
  - Simulated using zsim
- Workloads
  - Linked list (**ll**).
  - Hash table (**ht**)
  - Tree traversal (**tree**)
  - SpMV (**spmv**)
  - Page rank (**pr**).
  - Breadth-first search (**bfs**)
  - Single-source shortest path (**sssp**)
  - Weakly-connected component (**wcc)**

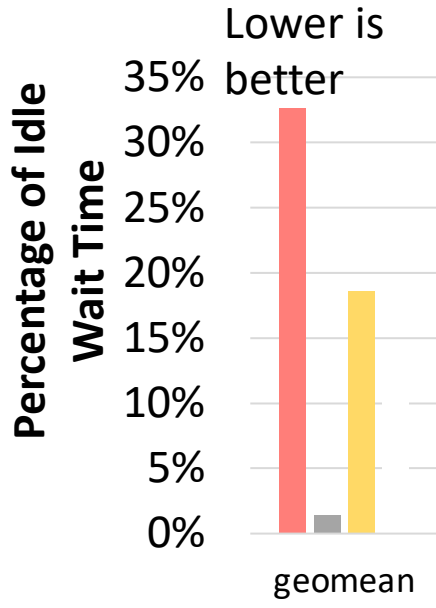
- Baselines

Communication	**Load Balancing**
CPU Forwarding	-
Bridge-based Communication	-
Bridge-based Communication	Work Stealing
Bridge-based Communication	Data-Transfer-Aware Scheduling

Experiment Results

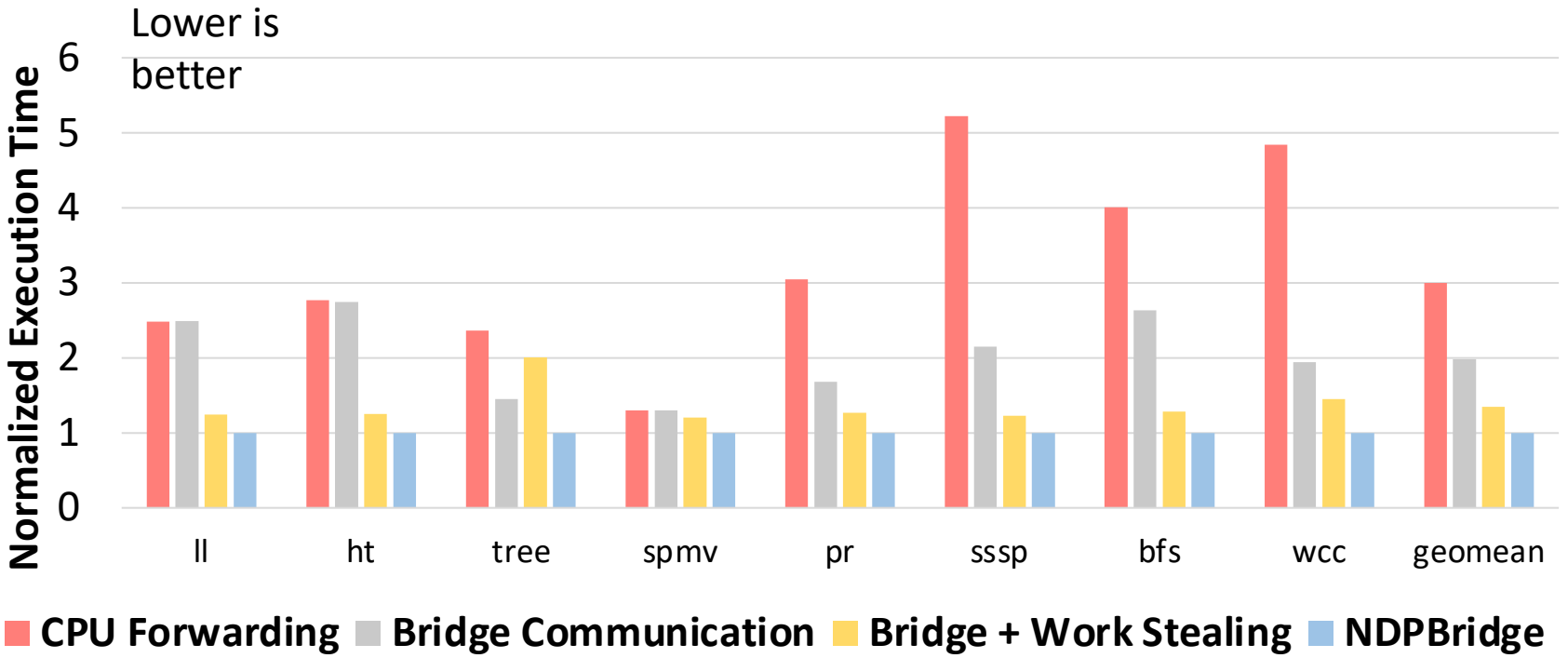
- Bridge-based communication: 1.51× speedup than CPU forwarding
  - Due to reduced communication overhead (**32.7% -> 1.4% idle wait time**)
  - Still suffers **load imbalance**
- Bridge + Work Stealing: 1.45× speedup than no scheduling
  - More communication overhead(**1.4% -> 18.6% idle wait time**)

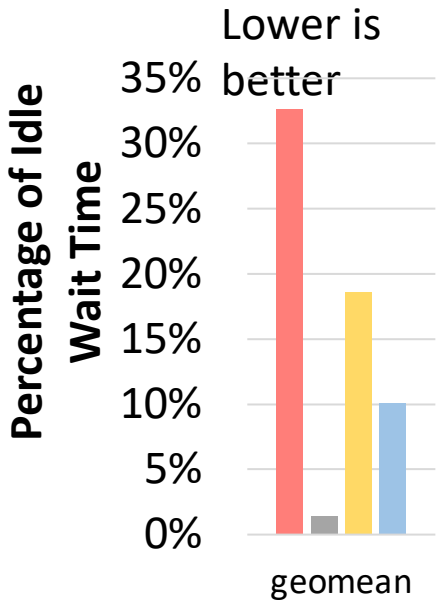




Experiment Results

- NDPBridge: best performance, **2.98× speedup** than CPU forwarding
  - 1.35× against Bridge+Work Stealing, **18.6% -> 10.0% idle wait time**





Summary

- The lack of cross-bank communication and load balancing support hinders the adoption of DRAM-bank NDP architectures.

- Our contributions:
  - **Bridge-based communication**: supports cross-bank communication with acceptable hardware cost
  - **Data-transfer-aware scheduling**: supports cross-bank load balancing built upon the communication scheme and with reduced data transfer overhead
  - **NDPBridge**: promotes wider and easier adoption of DRAM-bank NDP architectures

Thank you!