

ASTRA-sim 3.0: Taking Distributed Machine Learning Simulation

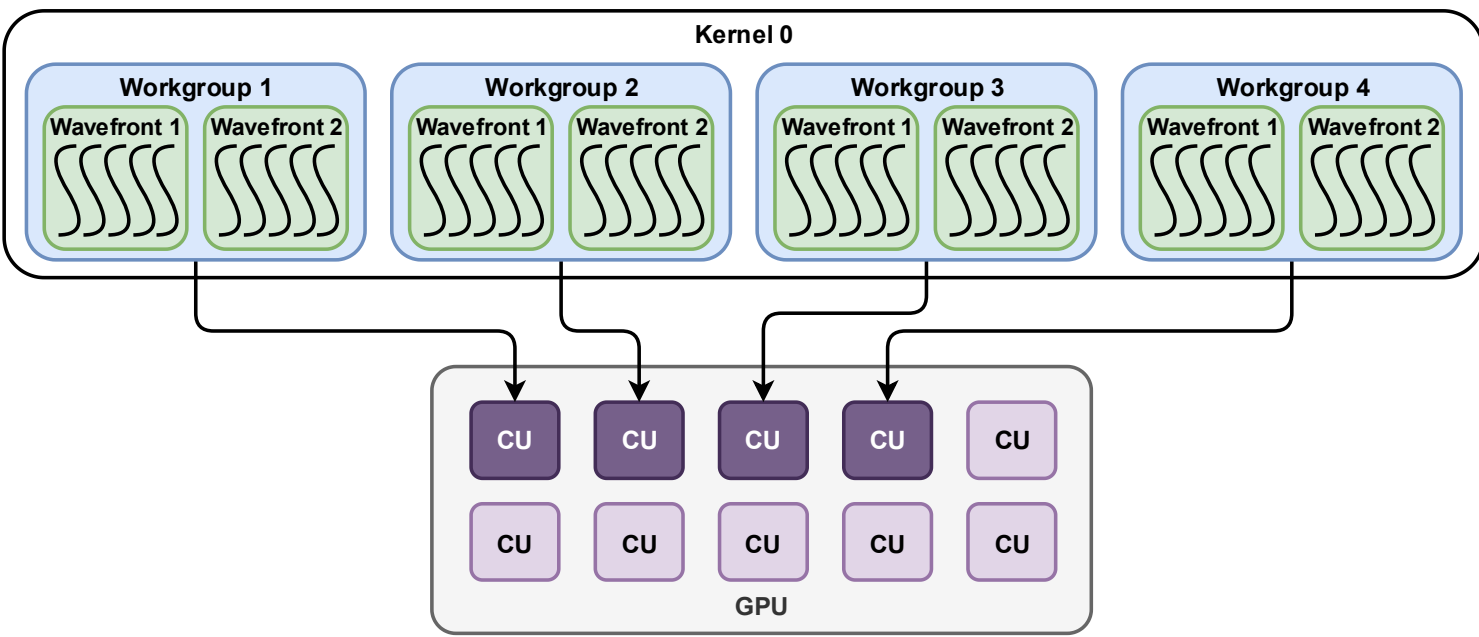
to the Next Level via Fine-Grained GPU Modeling

William Won, Tuan Ta, Moumita Dey, Ruchi Shah, Conor Green, and Bradford M. Beckmann
AMD Research and Advanced Development

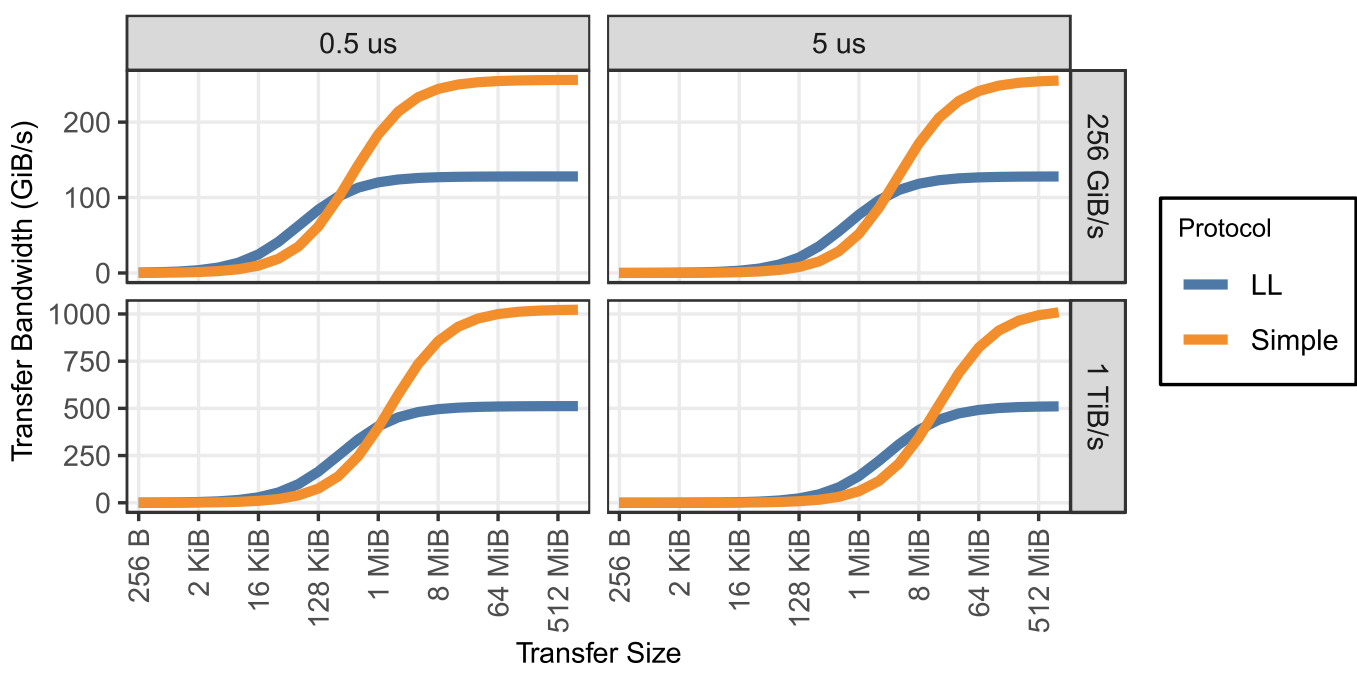
Primary Research Area: Recent Advances in ModSim Implementation

Introduction and Motivation

- Distributed machine learning (ML) modeling needs is in **inference**
 - **Small-sized** collective communications → **latency-sensitive**
- **Faithful latency modeling** is a must
 - **Data Path**: Cache-line-sized load and store transactions
 - **Control Path**: barriers, semaphores, resource constraints, etc.
- Trade-off between **fidelity** and **speed**
 - ASTRA-sim: fast, but coarse-grained: cannot accurately model latency
 - GPU simulators: high-fidelity, but slow and not flexible



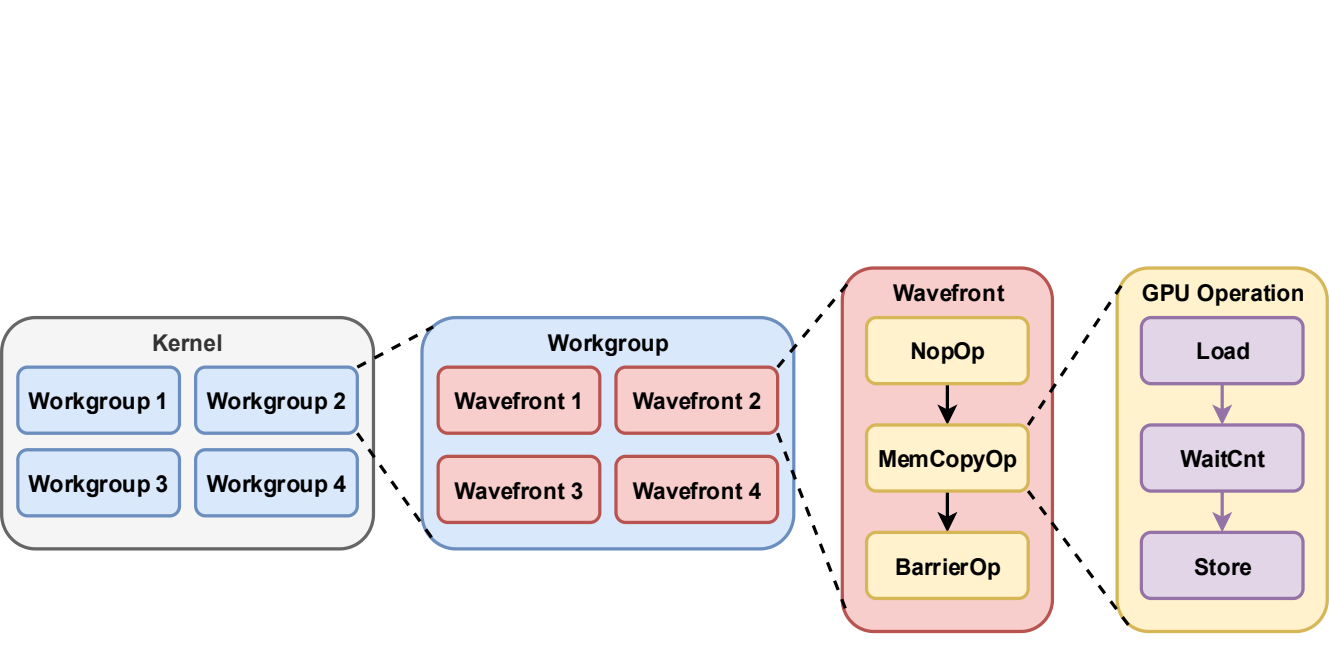
Compute unit (CU) runs **workgroup (WG)**
= multiple **load/store wavefronts (WFs)**



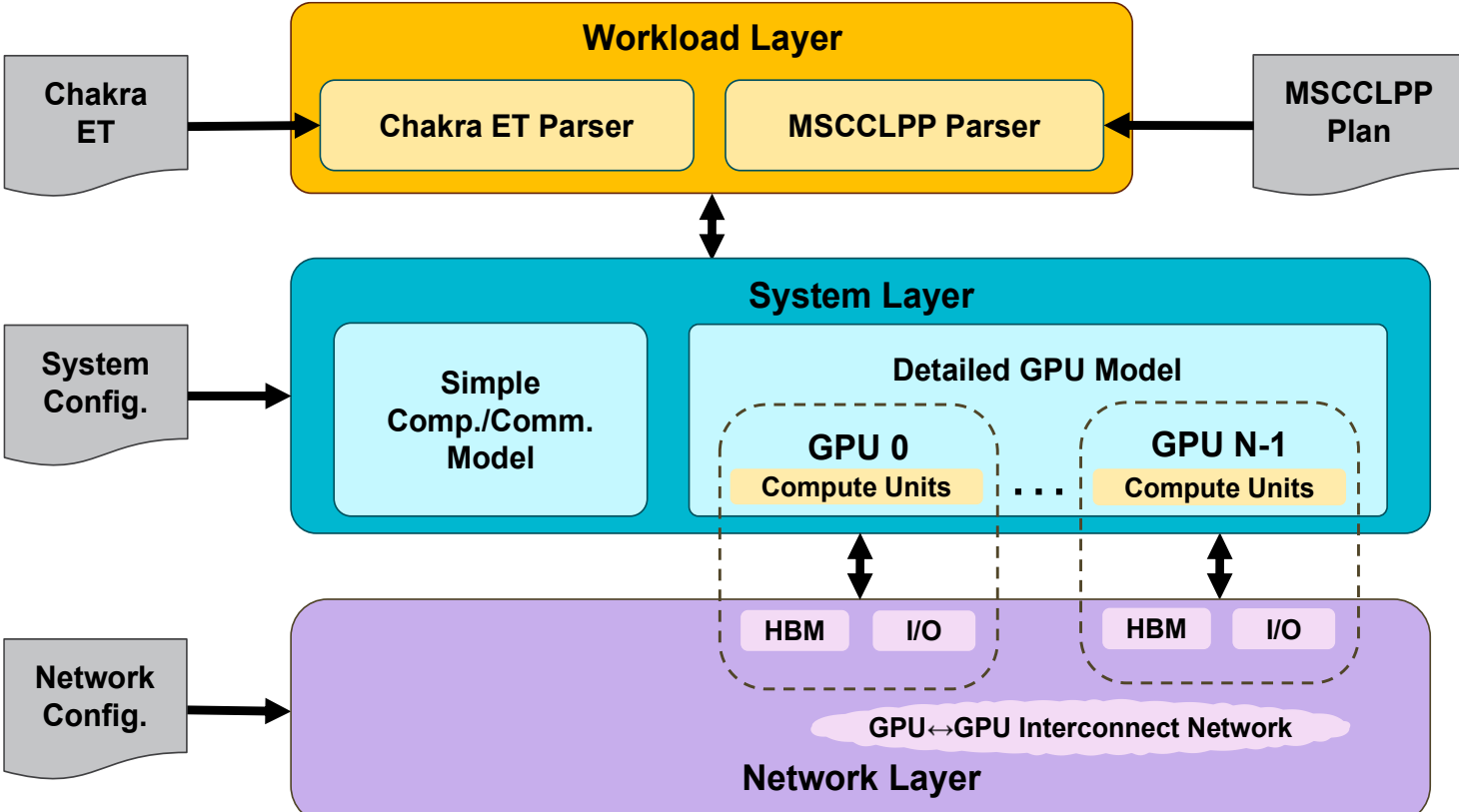
Small latency changes result in
different system implications

ASTRA-sim 3.0

- **Strike the right balance** between fidelity and speed with flexibility
 - Model **cache-line-sized** load/store transactions, but not full GPU assembly
 - New detailed **GPU Model** that simulates execution at CU-level granularity
 - Support **arbitrary collective algorithm** (we parse MSCCL++)



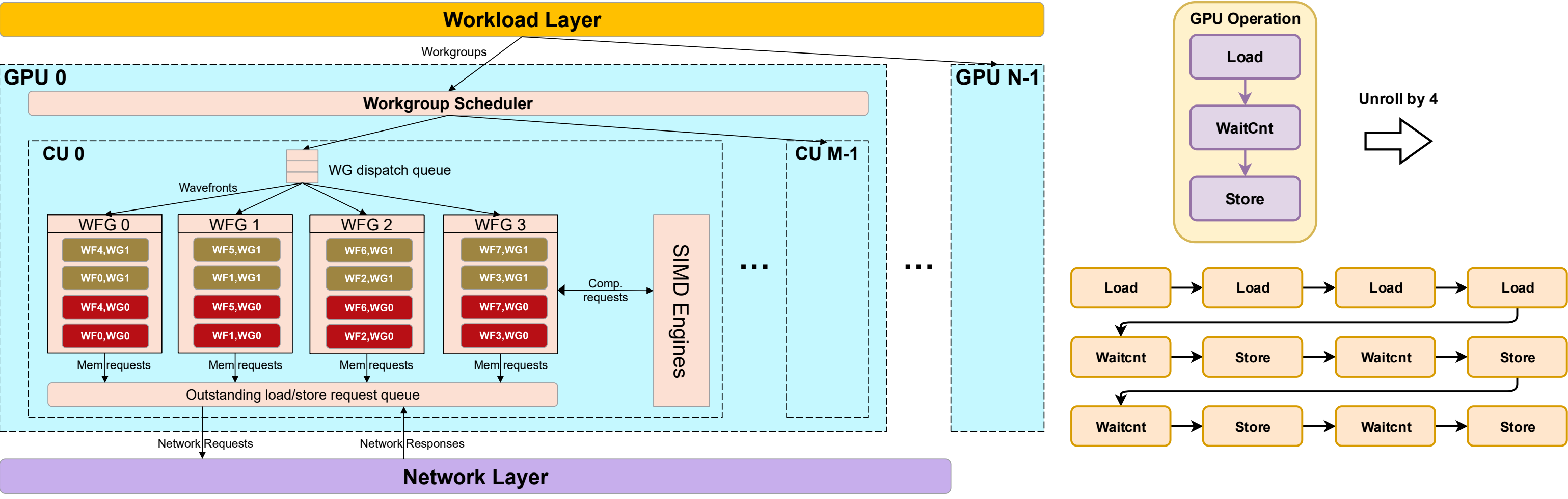
Fine-grained GPU simulation model



ASTRA-sim 3.0 **overview**

Detailed GPU Model

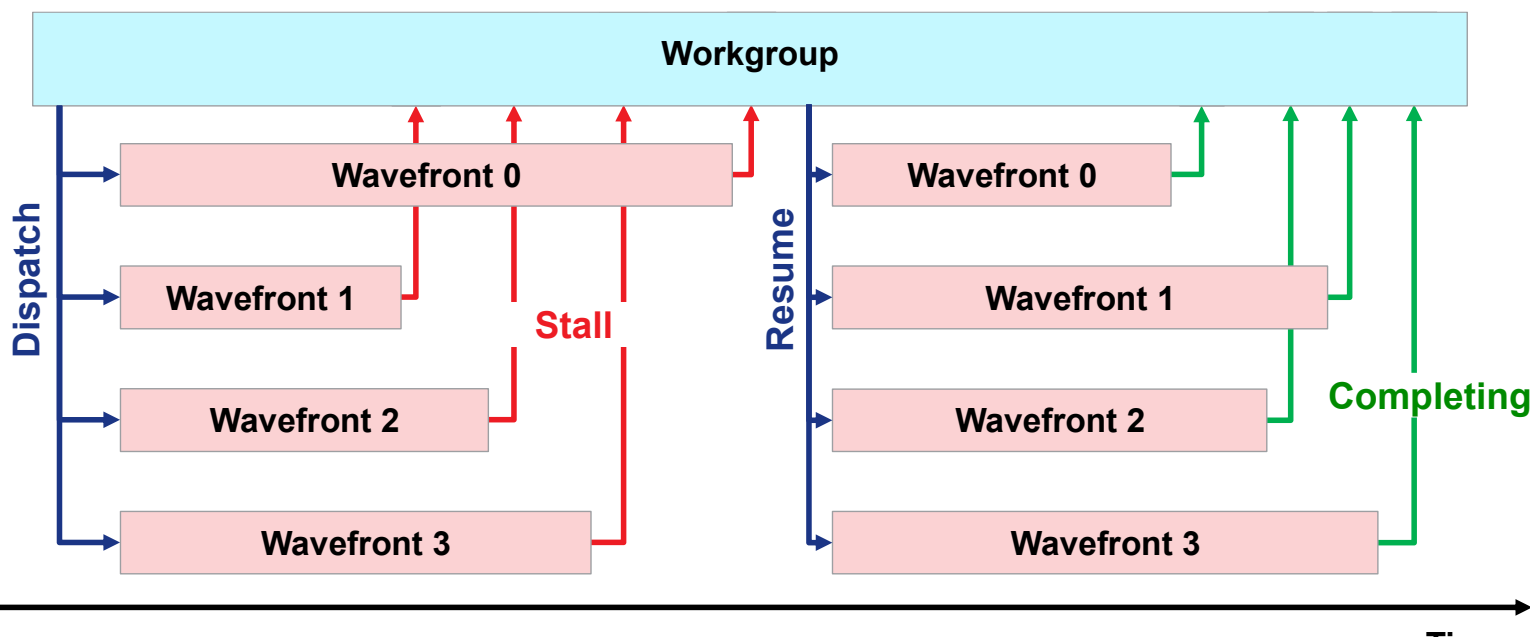
- Simulates **data operations** (load/store), **control** (barrier/semaphore)
- Data path modeling: dispatching cache-line-sized (128B) requests



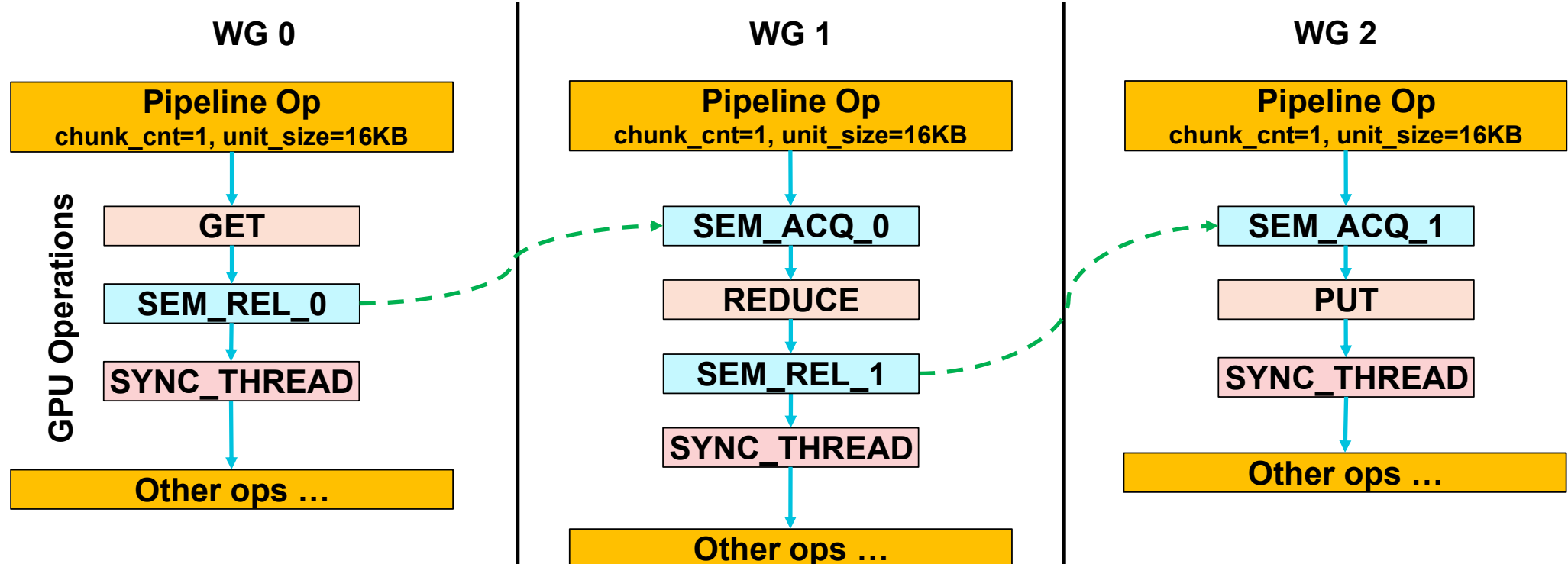
GPUs **execute WGs/WFs**: each WF
generates cache-line (128B) network requests

Loop unrolling enables
instruction-level parallelism

- Control path modeling: barrier (intra-WG) and semaphore (inter-WG)



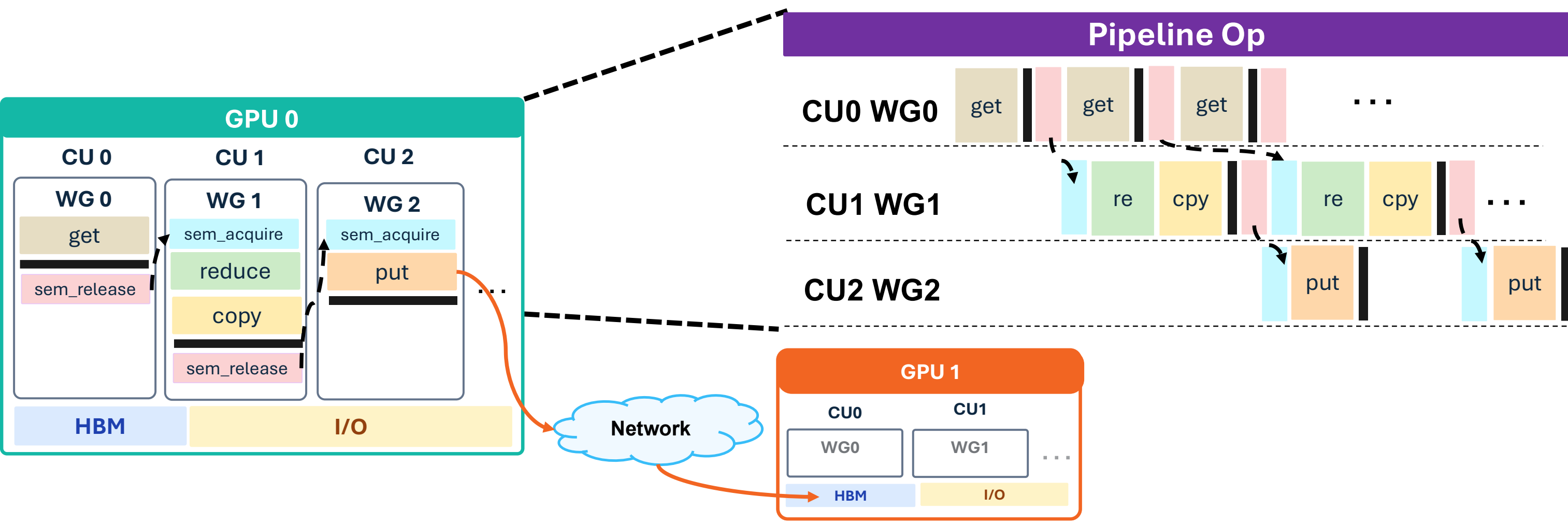
Barrier halts WF execution, synchronizes all WFs within a WG



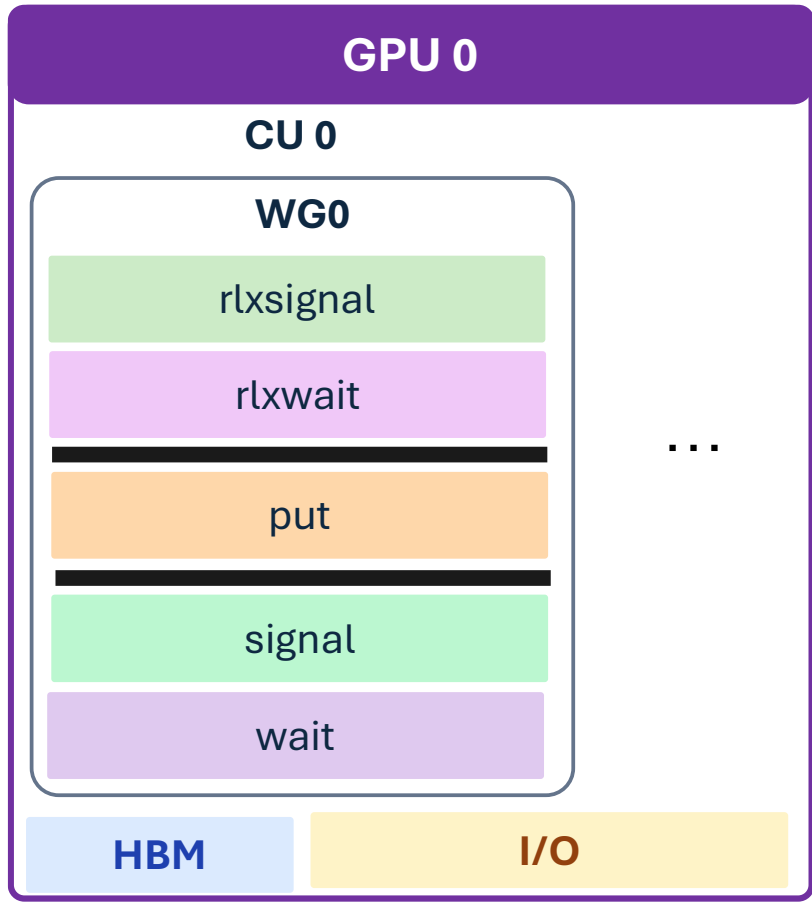
Semaphore halts **WG** execution, controls inter-WG control flow

Custom Collective Algorithm Support via MSCCL++

- Simulator needs a **flexible abstraction** to represent what GPUs do
 - **MSCCL++** is a domain-specific representation for collective communications



- MSCCL++ enables to represent flexible GPU communication behavior

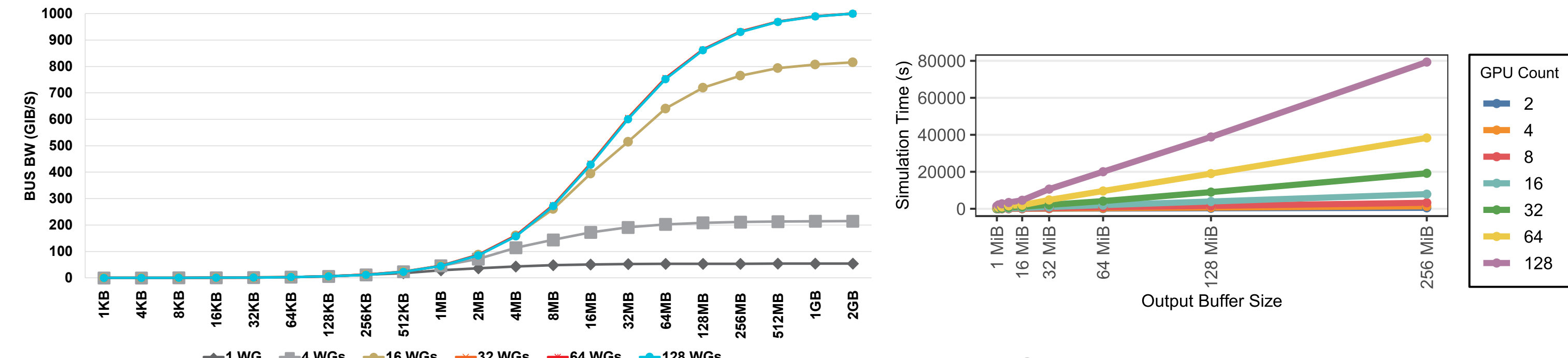


```
"gpus": [{
  "id": 0,
  "input_chunks": 1,
  "threadblocks": [{
    "id": 0,
    "ops": [
      {
        "name": "rxsignal", "channel_type": "memory",
        "src_buff": [{"index": 0, "size": 1}],
        "dst_buff": [{"buffer_id": 0, "size": 1}],
        "channel_type": "memory",
        "name": "nop",
      },
      {
        "name": "put",
        "src_buff": [{"index": 0, "size": 1}],
        "dst_buff": [{"buffer_id": 0, "size": 1}],
        "channel_type": "memory",
        "name": "nop",
      },
      {
        "name": "signal", "channel_type": "memory",
        "src_buff": [{"index": 0, "size": 1}],
        "dst_buff": [{"buffer_id": 0, "size": 1}],
        "channel_type": "memory",
        "name": "wait", "channel_type": "memory",
      }
    ]
  }
}]
```

- MSCCL++ lowers to **JSON**, input to the **ASTRA-sim 3.0**
 - ASTRA-sim 3.0 parses and constructs **GPU instructions** to simulate

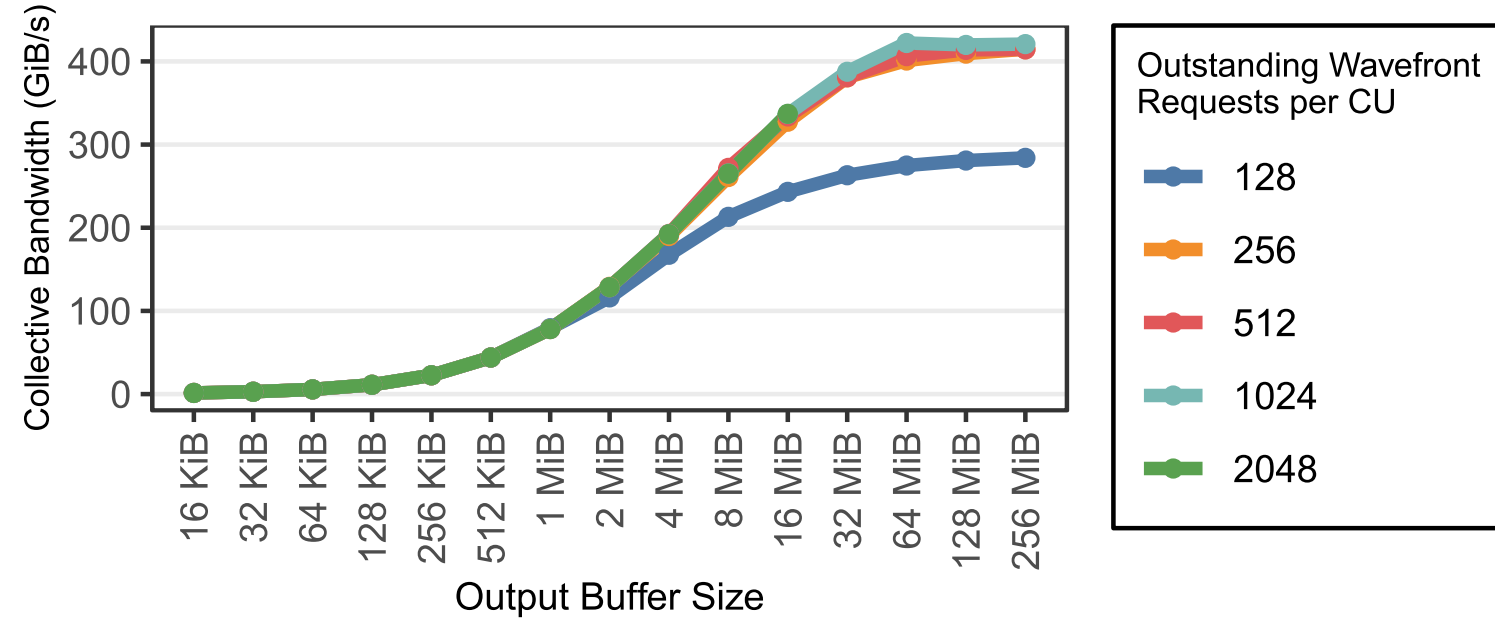
Preliminary Results

- ASTRA-sim 3.0 **enables** interesting design space explorations

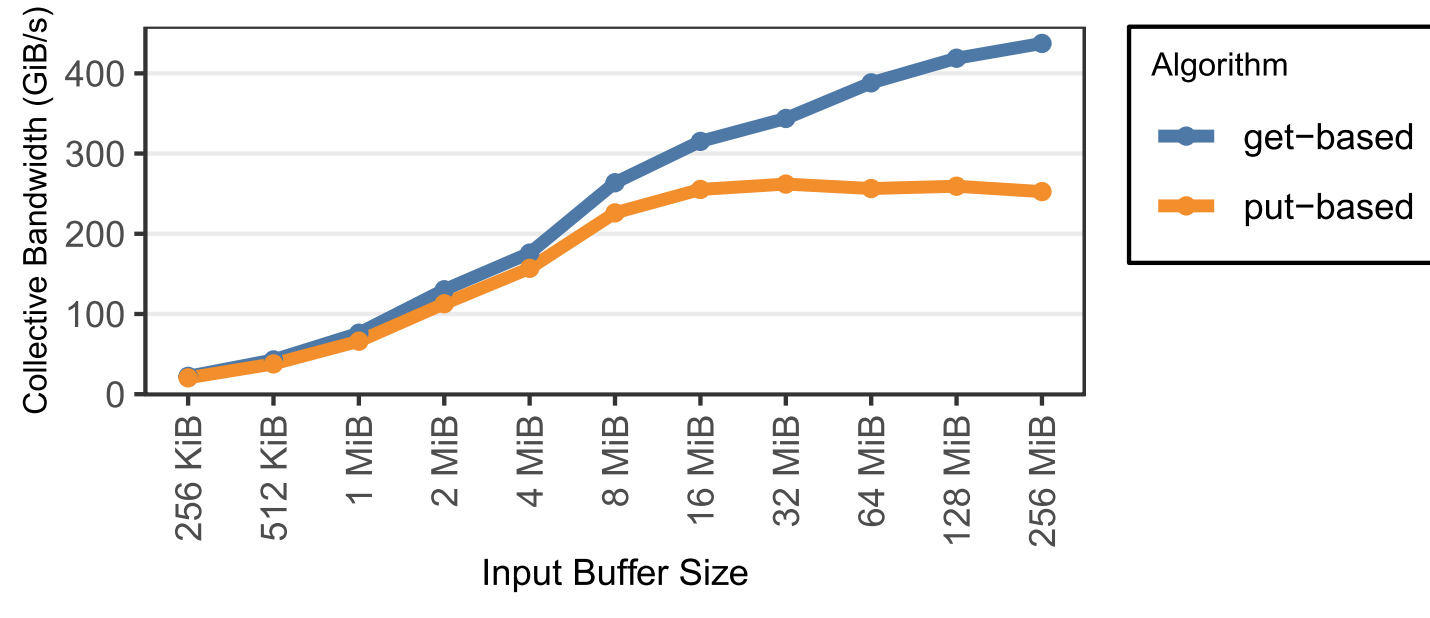


Saturating scale-up bandwidth needs
more workgroups per GPU

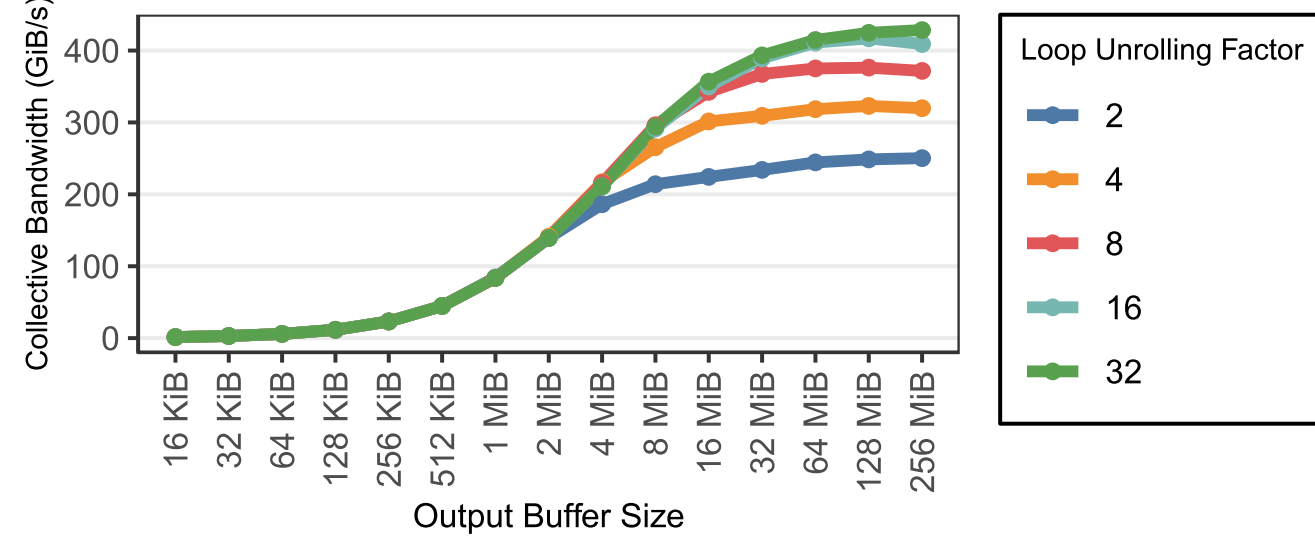
Simulation time scales
linearly to the comm size



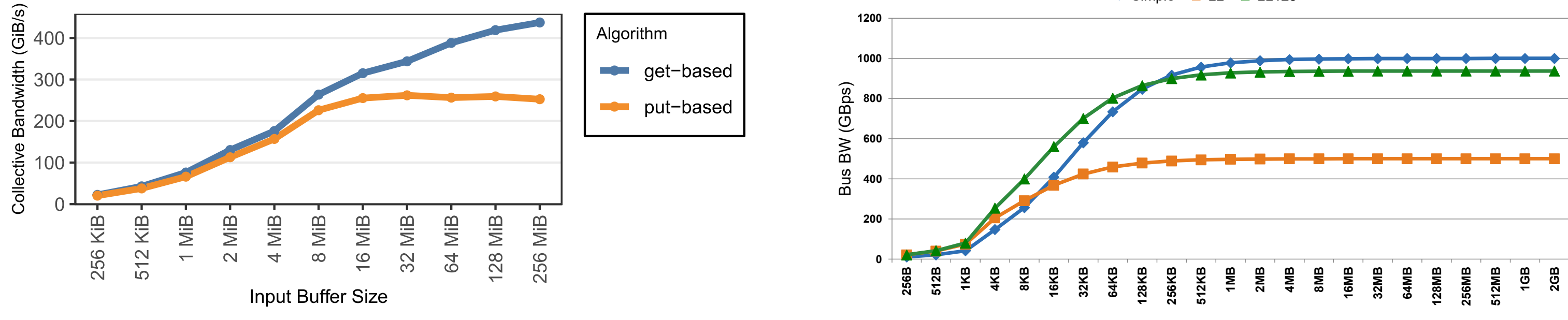
Larger register file yields
better collective performance



get is better than put
for Reduce-Scatter



Instruction-level parallelism yields
better collective performance



Collective size impacts
communication protocol preference

Disclaimer

AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc.

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED "AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING EVENFROM THE USE OF ANY INFORMATION CONTAINED HEREIN, IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.