Offloaded MPI message matching: an optimistic approach

Jerónimo S. García, Salvatore Di Girolamo, Sokol Kosta,
J.J. Vegas Olmos, Rami Nudelman, Torsten Hoefler, Gil Bloch

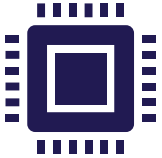
Aalborg Universitet

NVIDIA

ETH Zürich

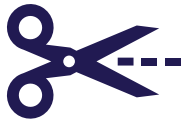
AALBORG
UNIVERSITY

Optimistic tag matching: Goals



Complete offload of tag matching (incl. unexpected msgs)

Save CPU cycles

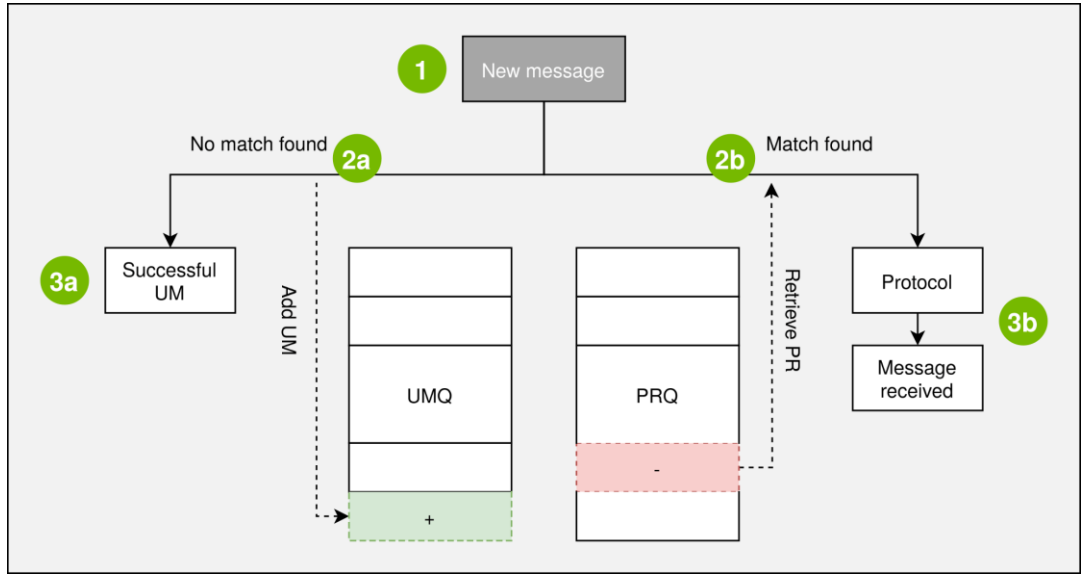


Remove the CPU from the network-GPU data-path

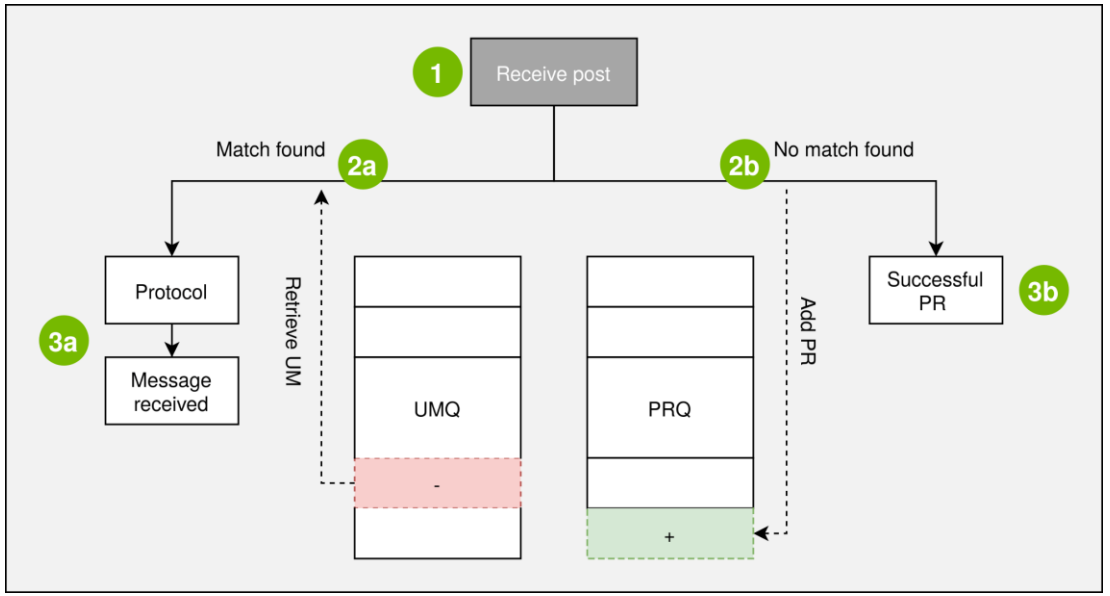
Traditional tag matching

- Tag matching is the critical operation that enables MPI point-to-point communications
- Usage of 2 queues:
  - UMQ – Unexpected messages queue
  - PRQ – Posted receives queue

Receiving a message



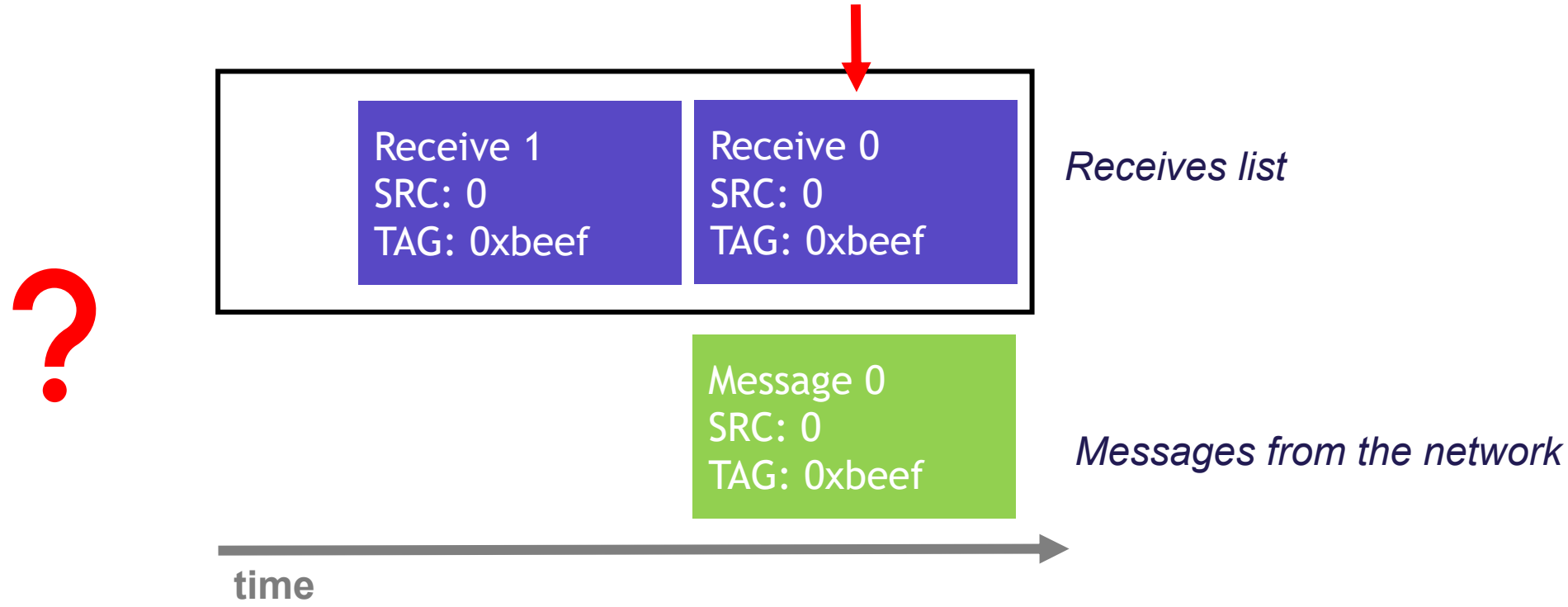
Requesting a message



MPI point-to-point constrains

C1: Order of posted receives

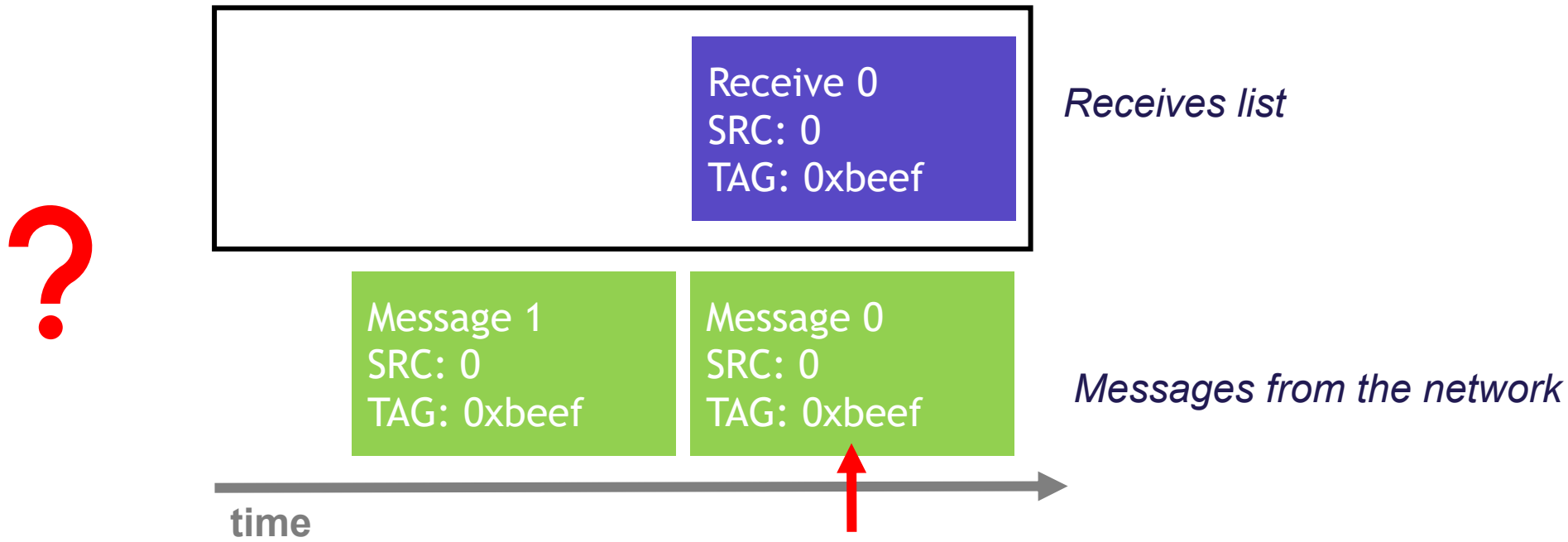
Receives matching the same message must be served in order



MPI point-to-point constrains

C2: No overtaking of messages

Messages from the same sender cannot overtake each other.



Other tag matching approaches

Software strategies

- Redesigning tag matching data structures and algorithms
- Dynamically adjusting the tag matching approach

Hardware strategies

- Complete offload traditional tag matching to accelerators
- Dedicated silicon space on NICs (hardware TM)

Optimistic Tag Matching

Challenge

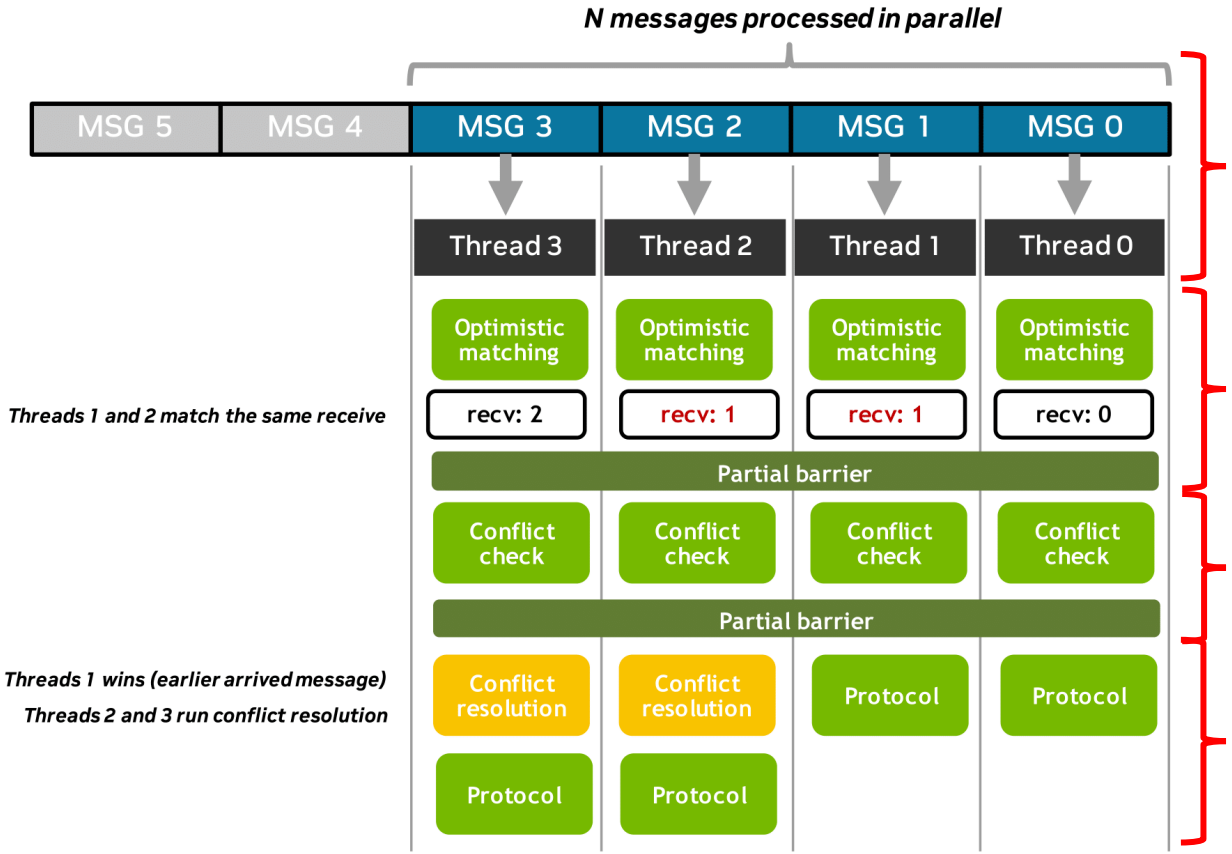
- MPI constrains serialize tag matching
- Need to extract parallelism from tag matching

- Redesign of MPI tag matching to fit to the highly parallel on-path NIC architectures

Tag matching

Optimistic Tag Matching: An overview

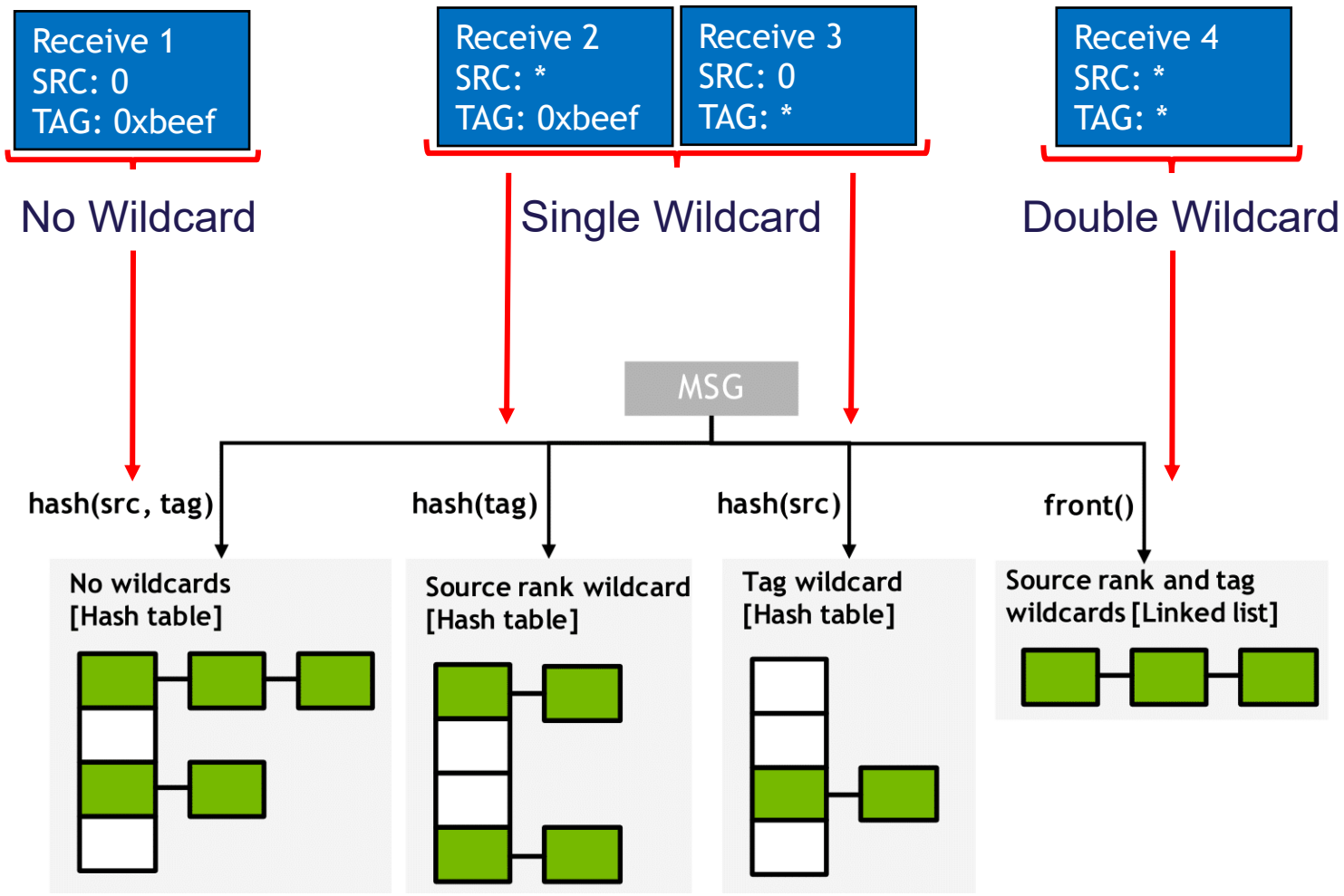
Optimistic Tag Matching



- N messages are processed concurrently
- Matching as if no violation of constrains
- Conflict check to detect violations of constrains
- Resolution of conflicts
- Handling of protocol

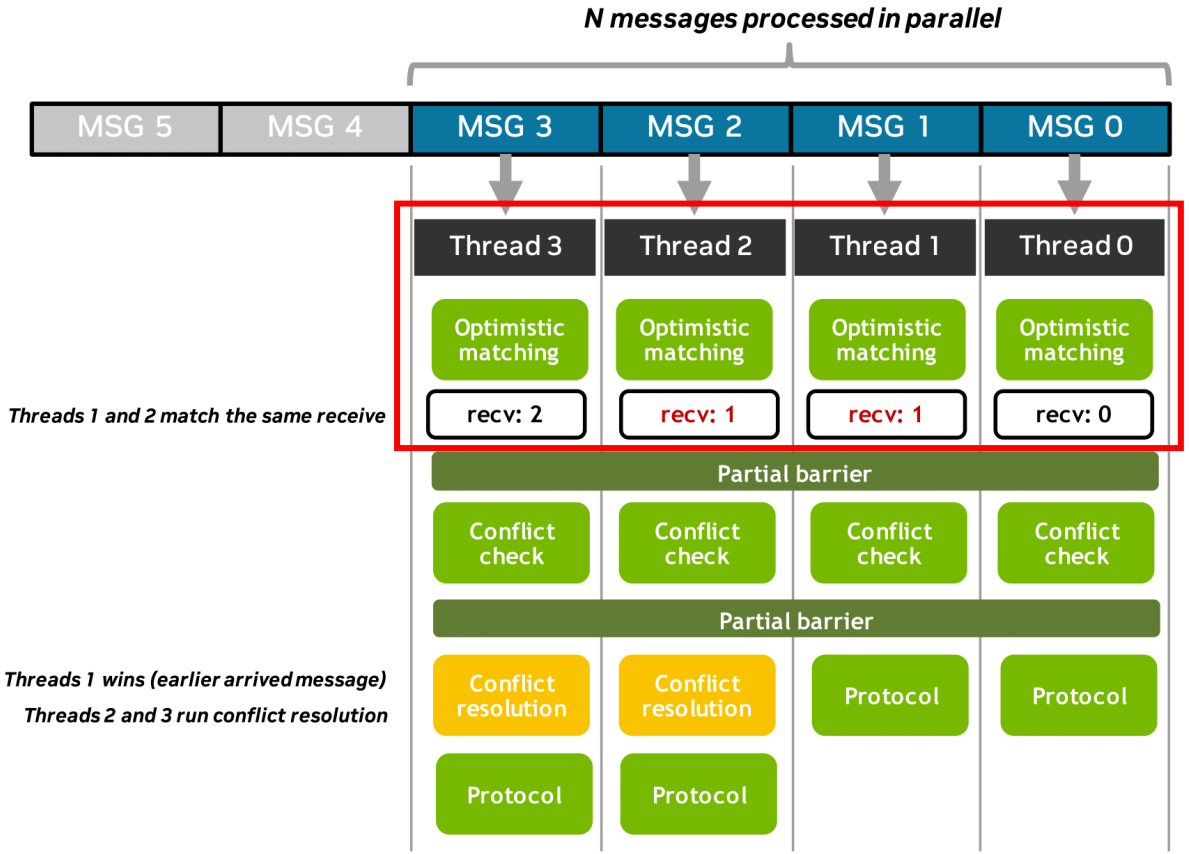
Indexing receives

Optimistic Tag Matching



Phase 1: Optimistic Matching

Optimistic Tag Matching

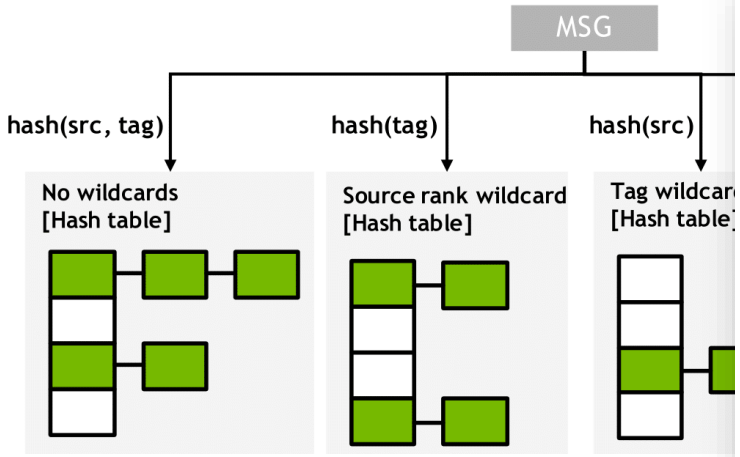


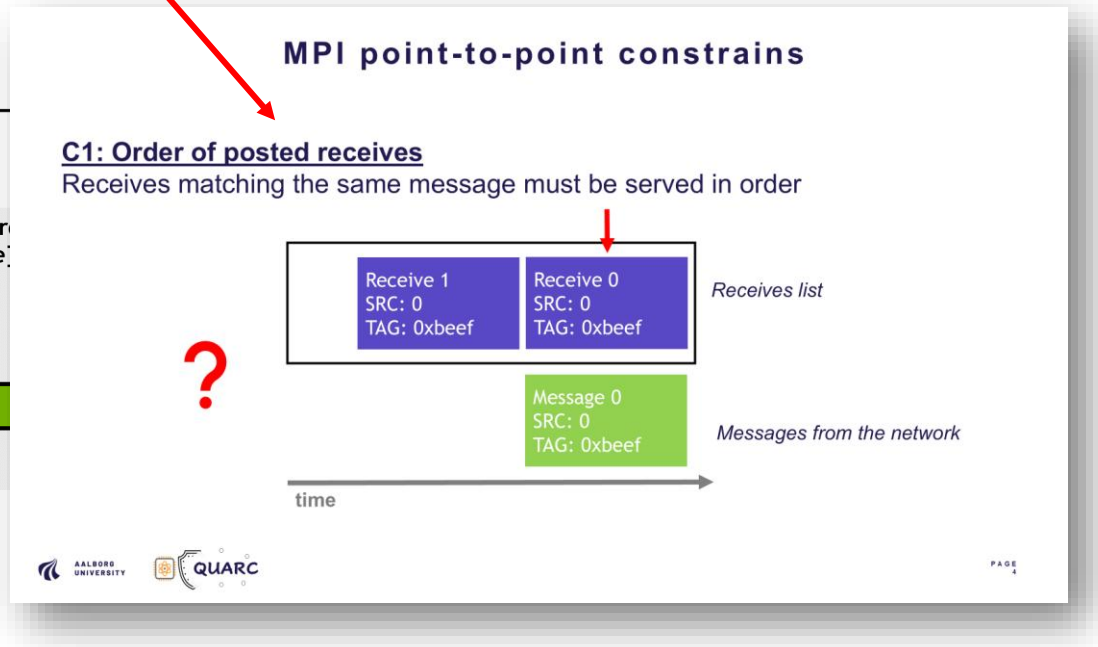
- N messages are processed concurrently
- Matching as if no violation of constrains
- Conflict check to detect violations of constrains
- Resolution of conflicts
- Handling of protocol

Phase 1: Optimistic Matching

Optimistic Tag Matching

- Optimistically search for a match while respecting constrain **C1**

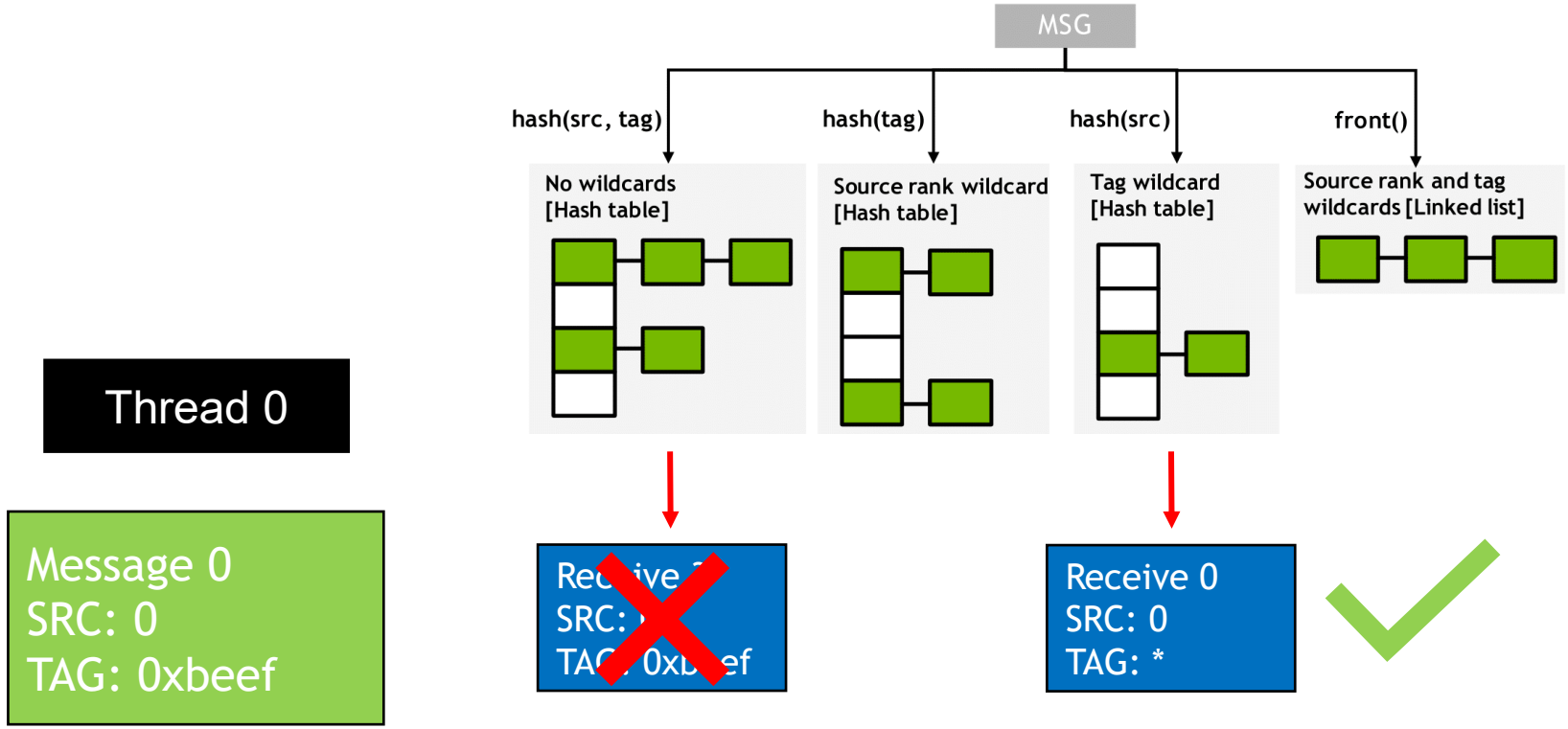




Phase 1: Optimistic Matching

Optimistic Tag Matching

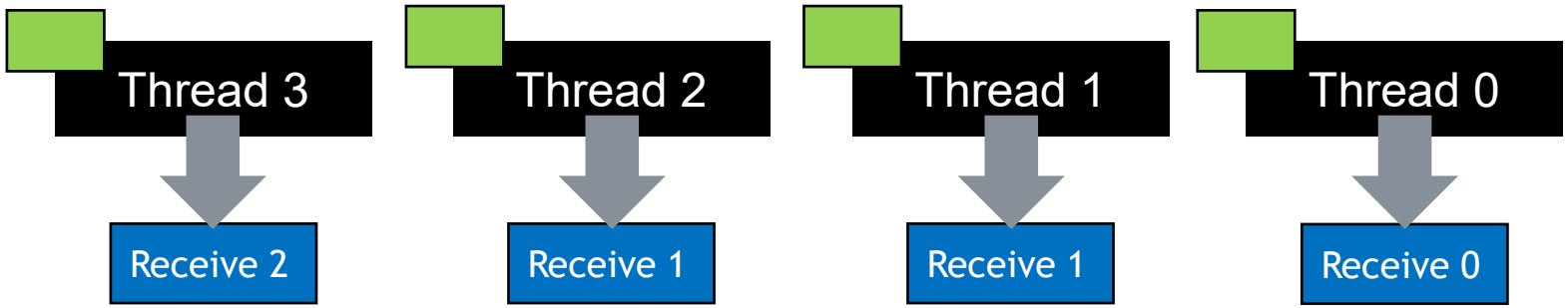
- Optimistically search for a match while respecting constrain **C1**



Phase 1: Optimistic Matching

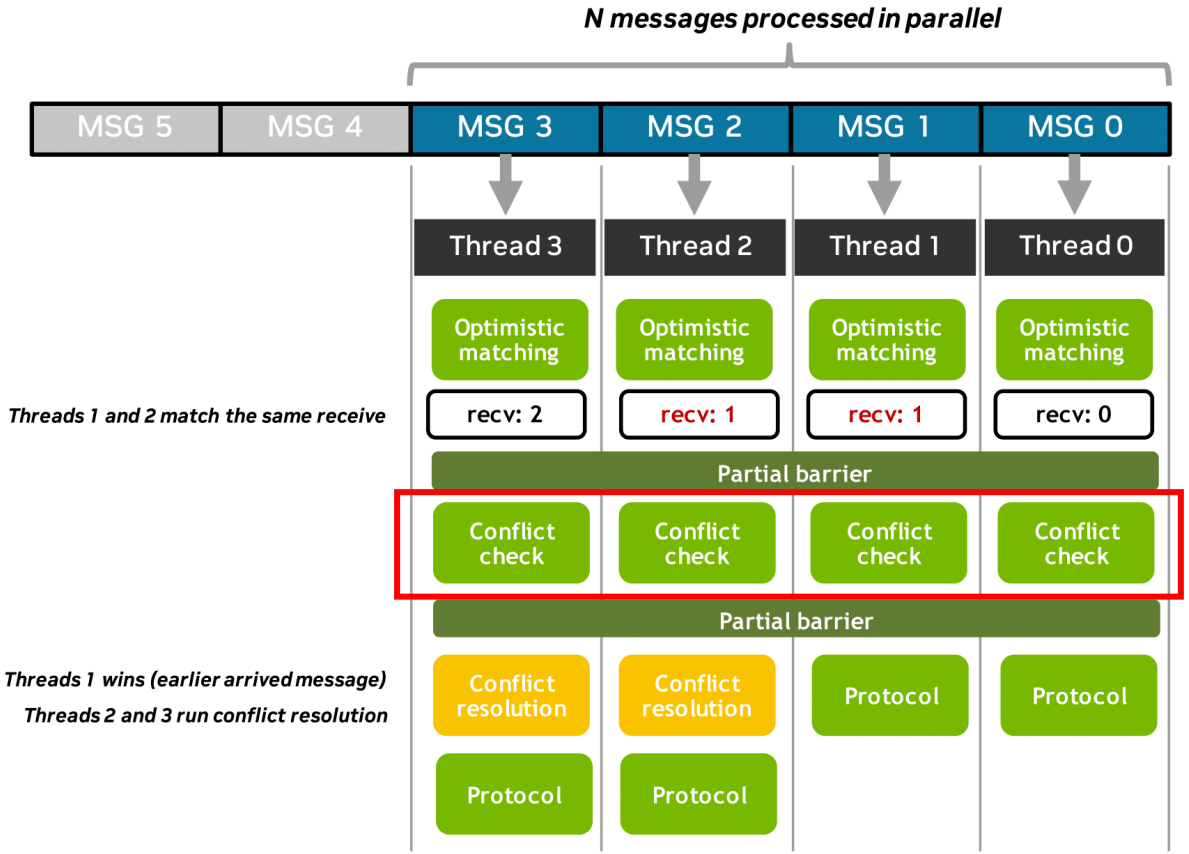
Optimistic Tag Matching

- Optimistically search for a match while respecting constrain **C1**



Phase 2: Conflict detection

Optimistic Tag Matching

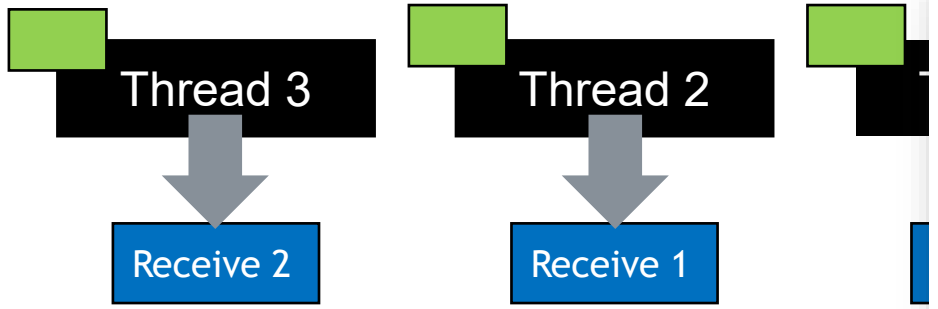


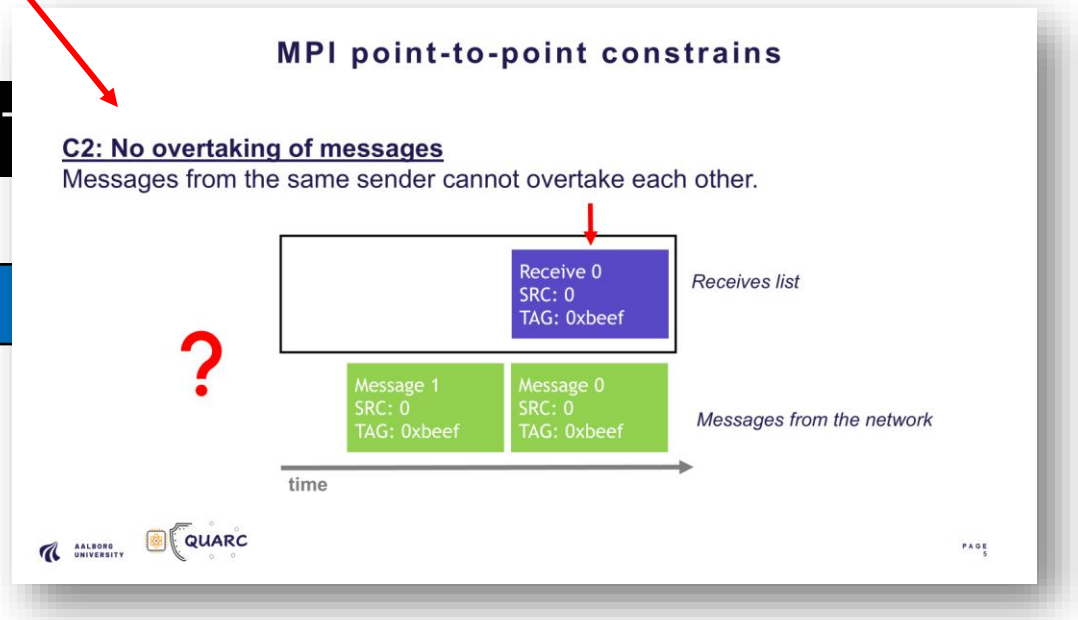
- N messages are processed concurrently
- Matching as if no violation of constrains
- Conflict check to detect violations of constrains
- Resolution of conflicts
- Handling of protocol

Phase 2: Conflict detection

Optimistic Tag Matching

- Verify that there are no conflicts while respecting **C2**

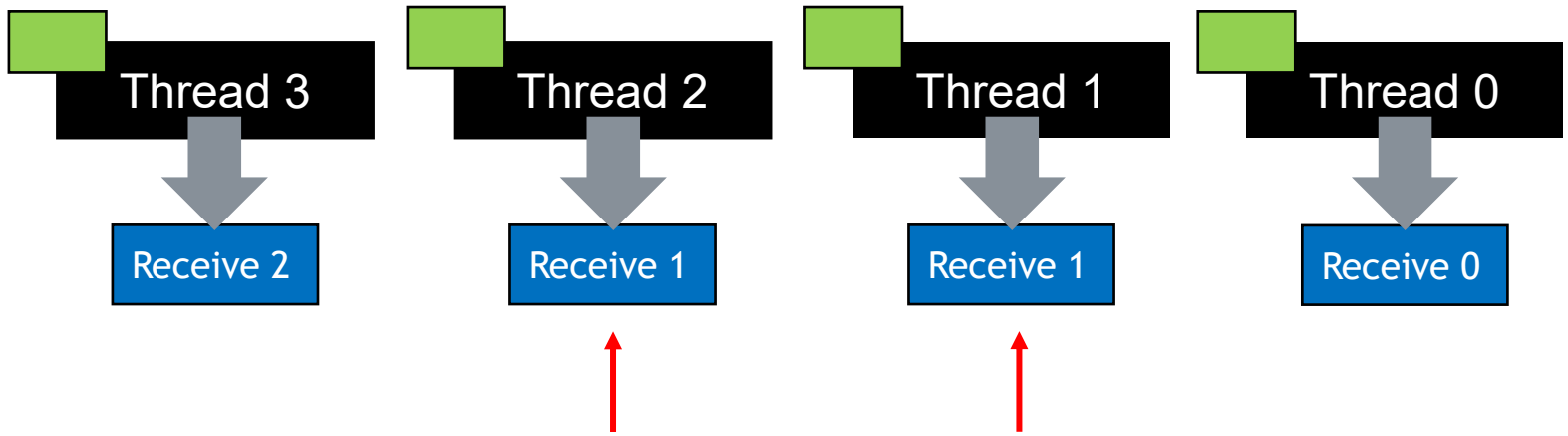




Phase 2: Conflict detection

Optimistic Tag Matching

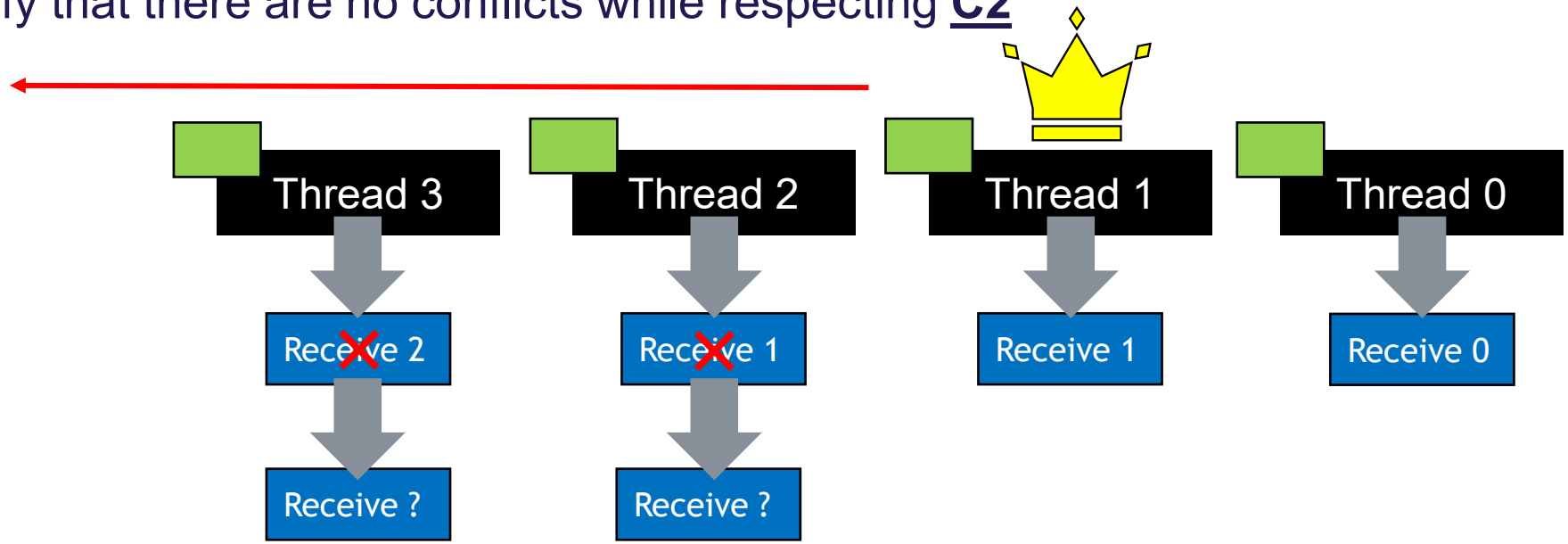
- Verify that there are no conflicts while respecting **C2**



Phase 2: Conflict detection

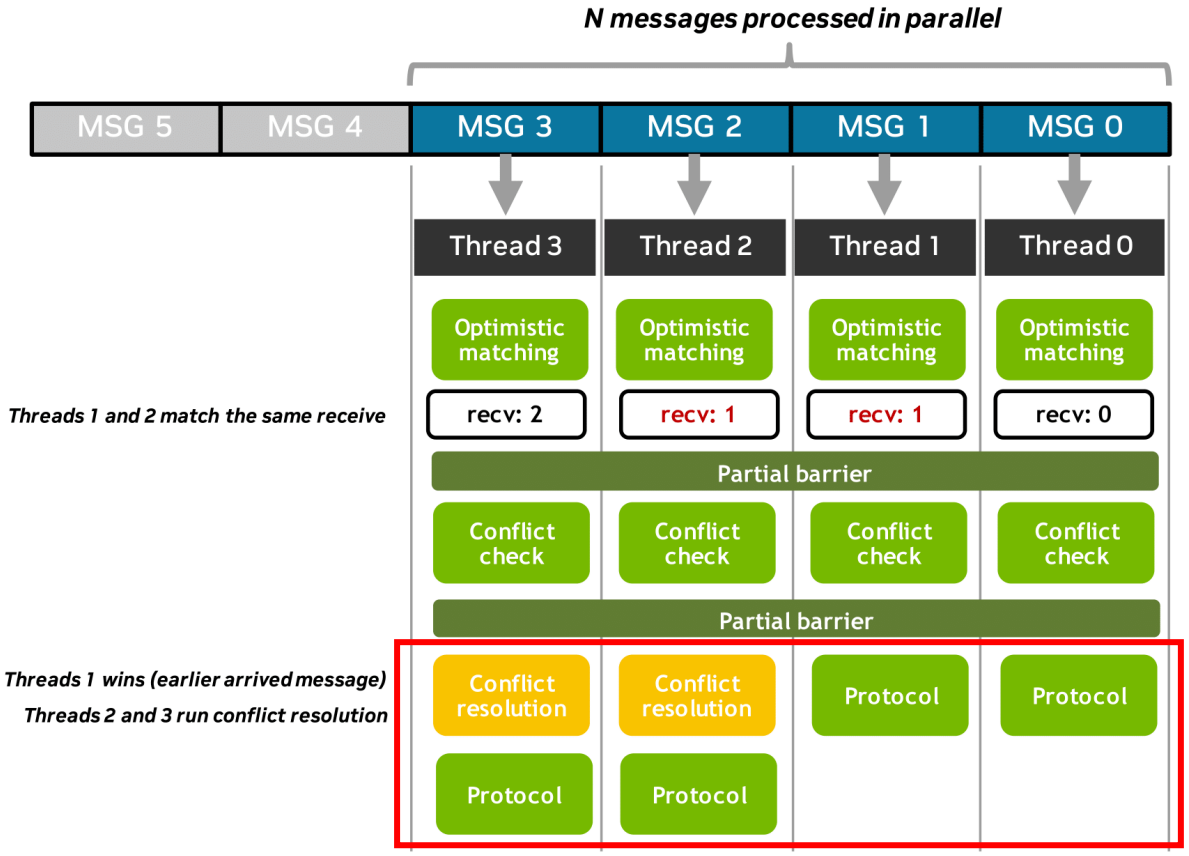
Optimistic Tag Matching

- Verify that there are no conflicts while respecting **C2**



Phase 3: Conflict resolution

Optimistic Tag Matching

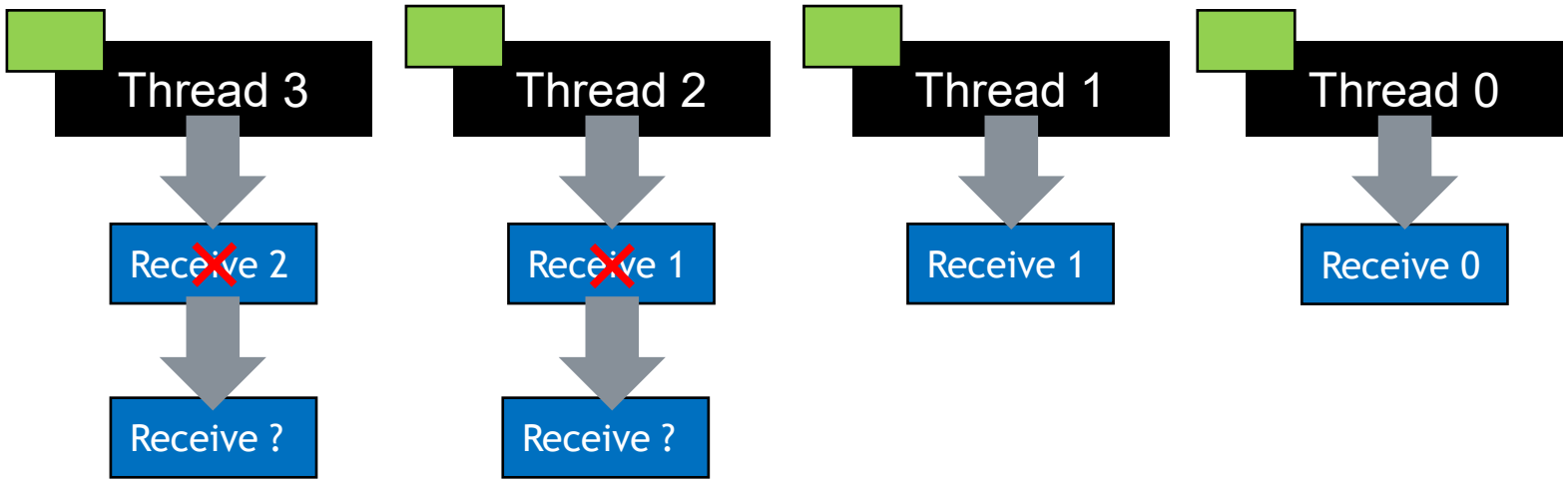


- N messages are processed concurrently
- Matching as if no violation of constrains
- Conflict check to detect violations of constrains
- Resolution of conflicts
- Handling of protocol

Phase 3: Conflict resolution

Optimistic Tag Matching

- Solve conflicting matches



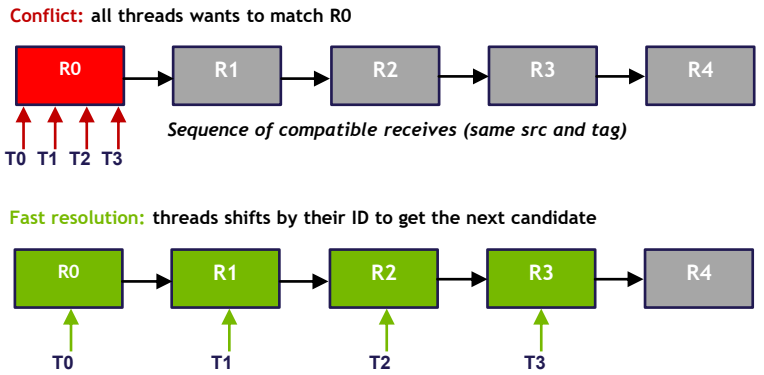
Phase 3: Conflict resolution

Optimistic Tag Matching

- Solve conflicting matches

Slow path

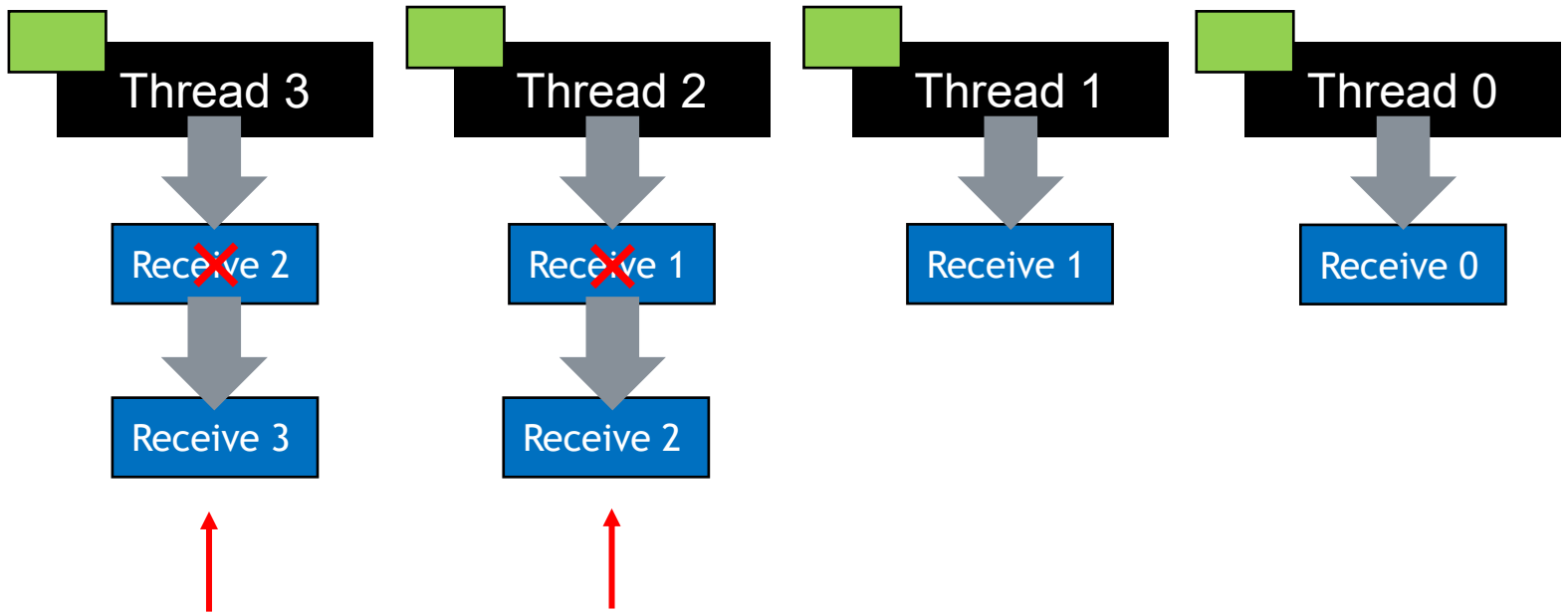
Fast path



Phase 3: Conflict resolution

Optimistic Tag Matching

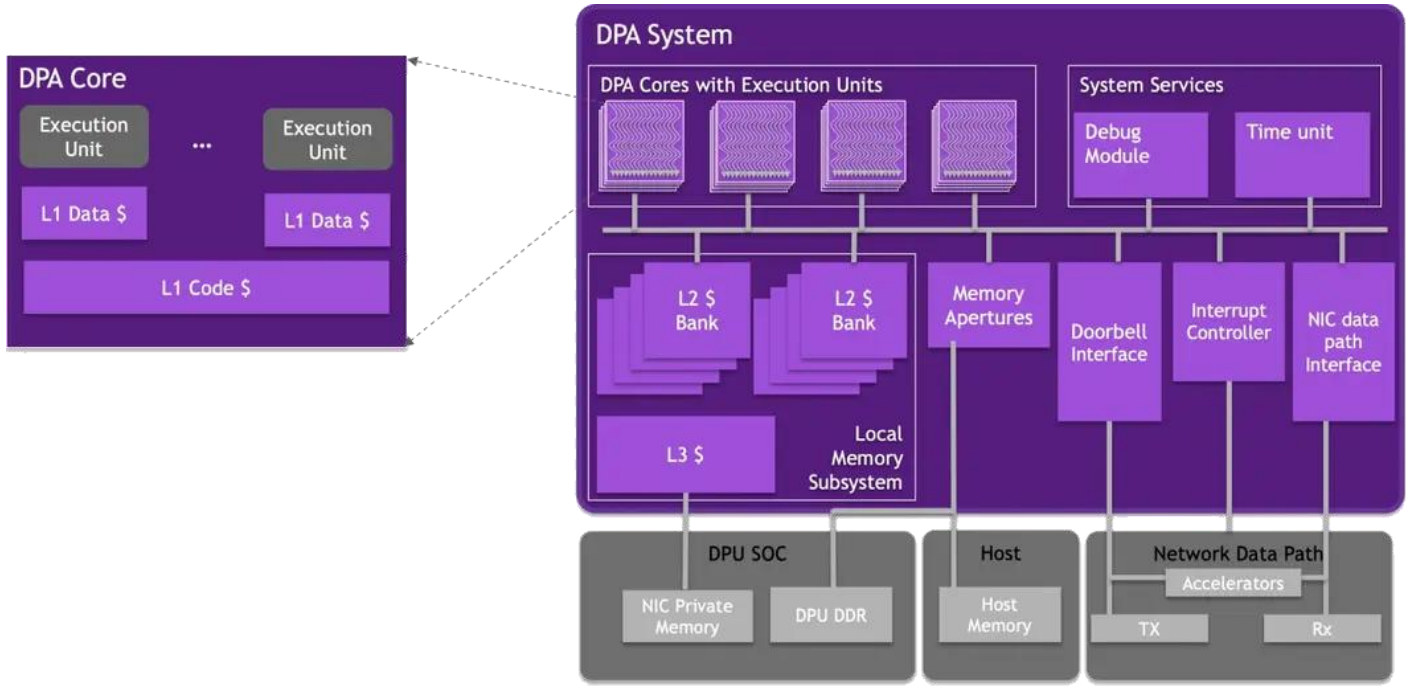
- Retrieve the message data



Data Path Accelerator: An introduction

Optimistic Matching on DPA

- 16 cores with 16 threads each
- 1.5 MiB of L2 and 3 MiB of LLC
- Direct access to NIC interrupts and doorbells
- Full access to NIC accelerators



DPA subsystem – DOCA documentation v2.8.0
NVIDIA Docs, 2024

Data Path Accelerator: An introduction

Optimistic Matching on DPA

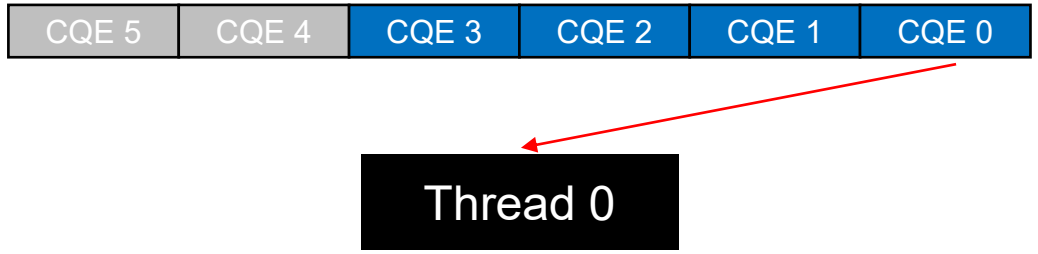
- Event handler executes its function each time an **event** occurs
- **Event:** RDMA completion received when the CQ is in armed state
- The event triggers an internal DPA interrupt that activates the event handler

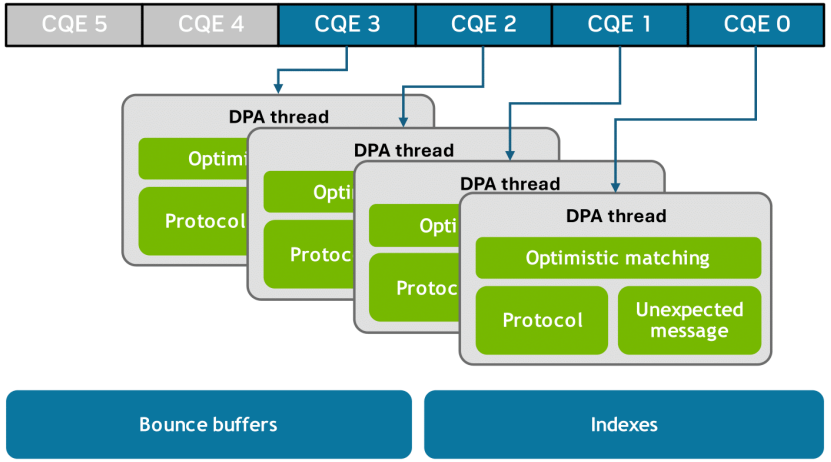
```
// Device code
__dpa_global__ myFunc(void *myArg)  {
    struct my_db *db = (struct my_db *)myArg;
    get_completion(db->myCq)
    work();
    arm_cq(myCq);
    return;
}

// Host code
main() {
    /* Load the application code into the DPA */
    flexio_process_create(device, application, &myProcess);
    /* create event handler to run my_func with my_arg */
    flexio_event_handler_create(myProcess, myFunc, myArg, &myEventHandler);
    /* Associate the event hanlder with a specific CQ */
    create_cq(&myCQ, ... , myEventHandler)
    /* start the event handler */
    flexio_event_handler_run(myEventHandler)
    ...
}
```

Mapping Optimistic to the DPA

Optimistic Matching on DPA

- MPI message = CQE (Completion Queue Entry)
- Striding over CQEs

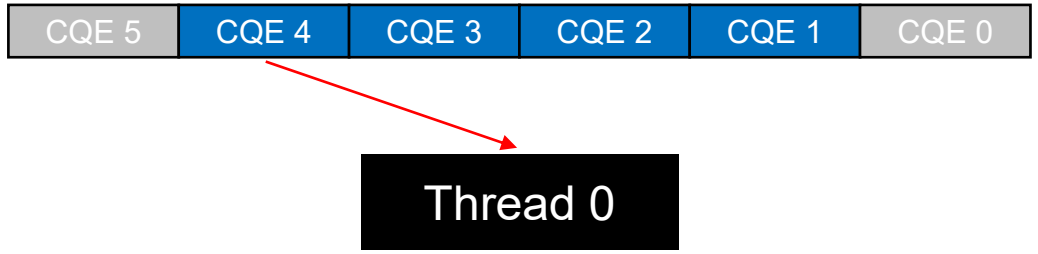


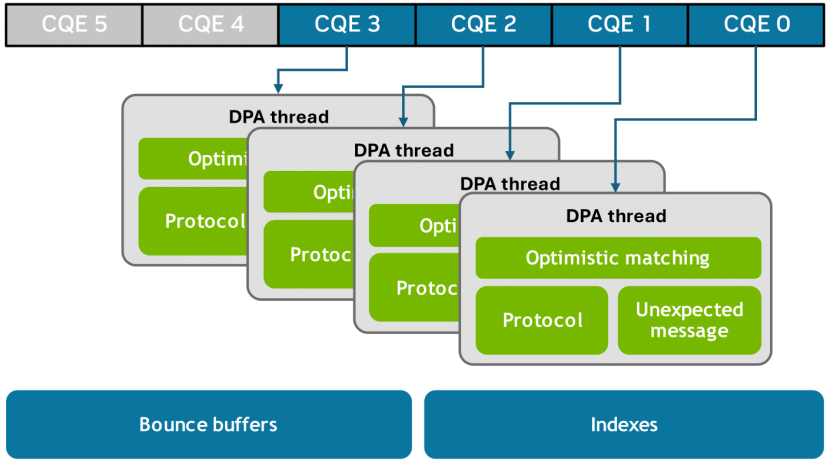


Mapping Optimistic to the DPA

Optimistic Matching on DPA

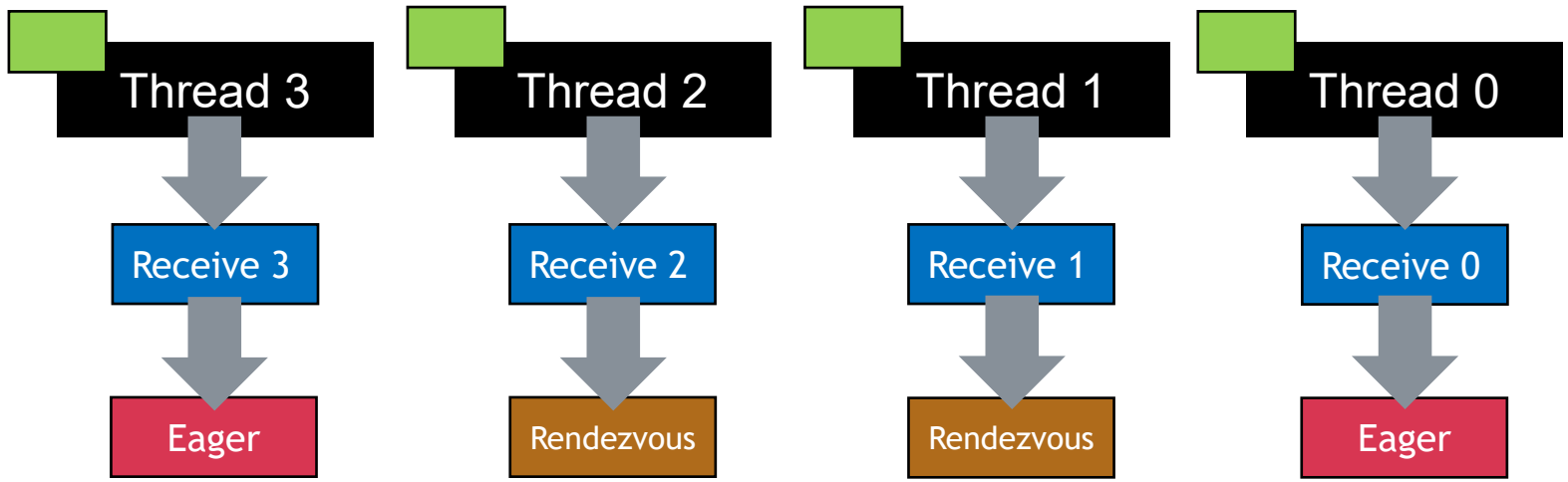
- MPI message = CQE (Completion Queue Entry)
- Striding over CQEs





Handling protocol

Optimistic Matching on DPA



Can be handled by the DPA

Adopted optimizations

Calculate hash in sender

Early booking check

Lazy removal

Collisions analysis: MPI proxy apps

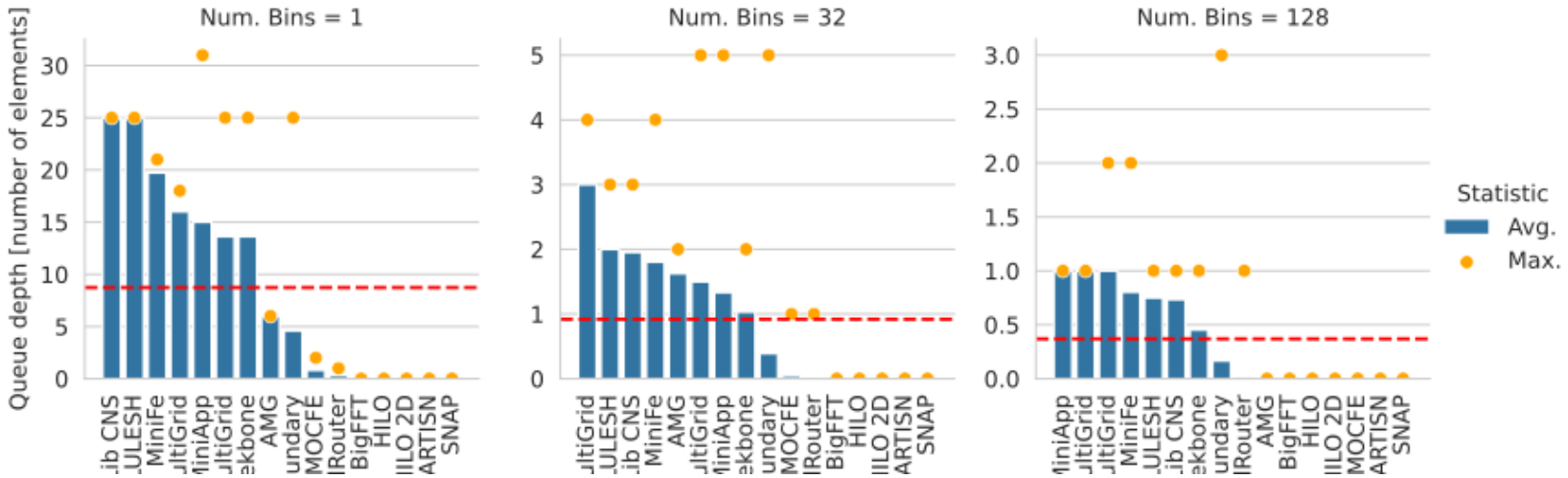
Results

- **16** MPI **proxy apps** analyzed
- For the majority, +90% of their operations **are p2p**

Application	Description	Number of processes
AMG	Algebraic MultiGrid. Linear equation solver	8
AMR MiniApp	Single step AMR for hydrodynamics	64
BigFFT	Distributed Fast Fourier Transform	1024
BoxLib CNS	Compressible Navier Stokes equations integrator	64
BoxLib MultiGrid	Single step BoxLib linear solver	64
CrystalRouter	Proxy application for the Nek5000 scalable communication pattern	100
FillBoundary	Proxy application for ghost cell exchange using MultiFabs	1000
HILO	Modeling of Neutron Transport Evaluation and Test Suite	256
HILO 2D	Modeling of Neutron Transport Evaluation and Test Suite in 2D multinode	256
LULESH	Proxy application for hydrodynamic codes	64
MiniFe	Proxy application for finite elements codes	1152
MOCFE	Proxy application for Method of Characteristics (MOC) reactor simulator	64
MultiGrid	MultiGrid solver based on BoxLib	1000
Nekbone	Proxy application for the Nek5000 poison equation solver	64
PARTISN	Discrete-ordinates neutral-particle transport equation solver	168
SNAP	Proxy application for the PARTISN communication pattern	168

Collisions analysis: Results

Results

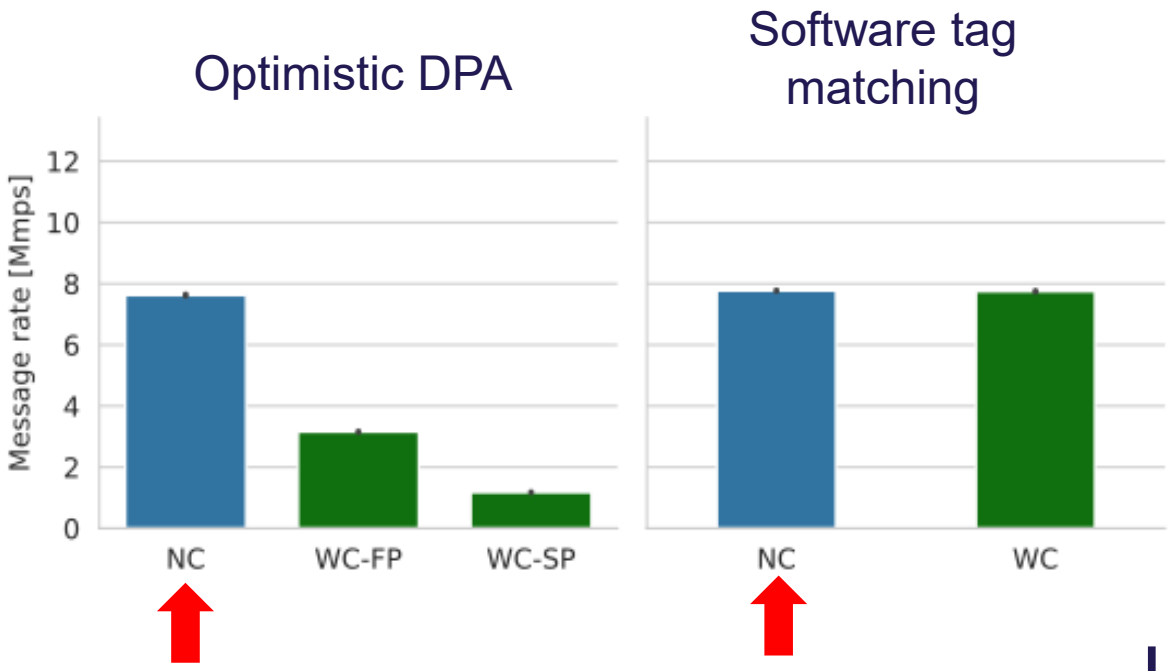


- Generated with MPI trace analyzer for tag matching

- **Trend:** More bins == reduction of queue depths
- **Takeaway: Over 32 bins, collisions are not likely to happen**

Message rate on the DPA

Results



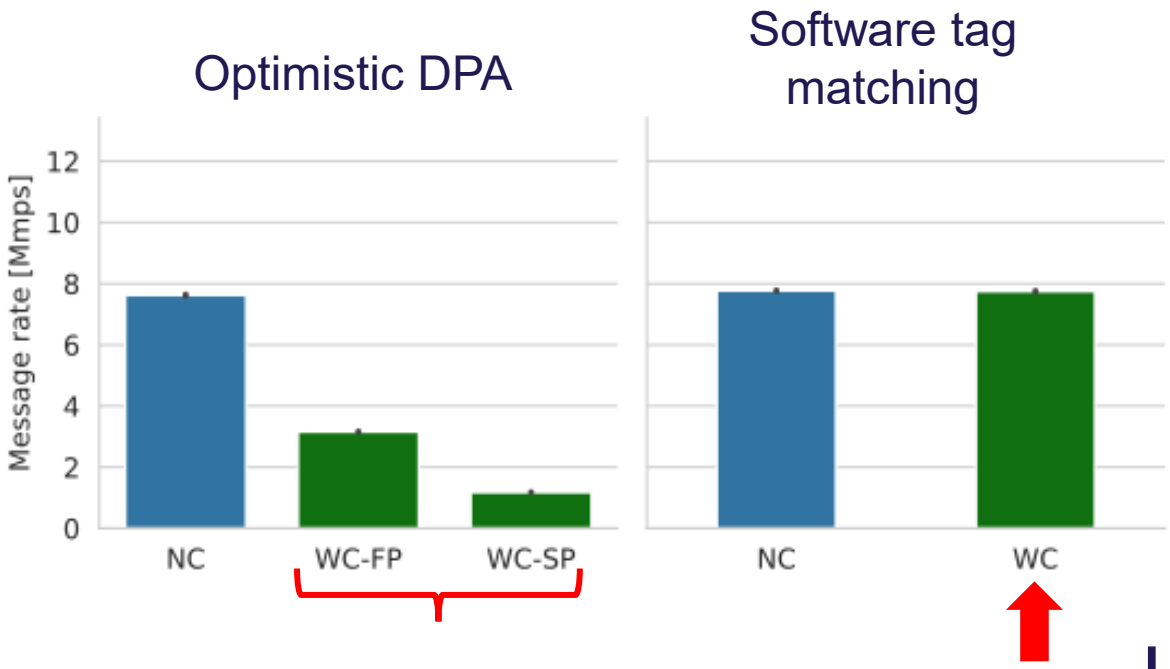
- **Performance is comparable when no collisions**
- **But not when there are collisions**

Legend

- **NC**: No collisions
- **WC**: With collisions
  - **FP**: Fast path
  - **SP**: Slow path

Message rate on the DPA

Results



- **Performance is comparable when no collisions**
- **But not when there are collisions**

Legend

- **NC**: No collisions
- **WC**: With collisions
  - **FP**: Fast path
  - **SP**: Slow path

A trip down memory lane

Recap

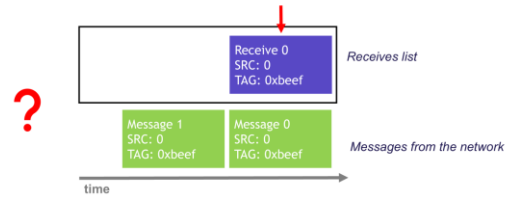
Traditional tag matching

- Tag matching is the critical operation that enables MPI point-to-point communications
- Usage of 2 queues:
  - UMQ – Unexpected messages queue
  - PRQ – Posted receives queue

MPI point-to-point constrains

C1: Order of posted receives

Receives matching the same message must be served in order



Optimistic Tag Matching: An overview

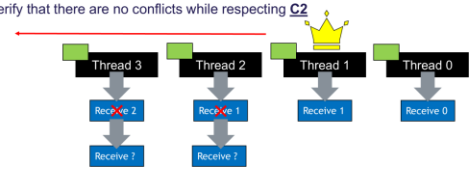
Optimistic Tag Matching

- N messages are processed concurrently

Phase 2: Conflict detection

Optimistic Tag Matching

- Verify that there are no conflicts while respecting **C2**

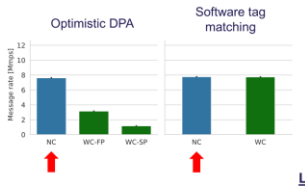


Collisions analysis: Results

Results

Message rate on the DPA

Results



- **Performance is comparable when no collisions**
- **But not when there are collisions**

Legend

- **NC**: No collisions
- **WC**: With collisions
  - **FP**: Fast path
  - **SP**: Slow path


Thanks for your attention!
AALBORG
UNIVERSITY