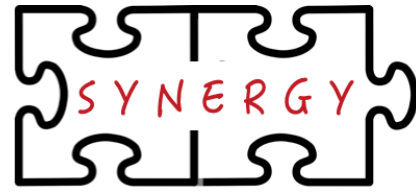




Georgia Tech School of Electrical and  
Computer Engineering  
College of Engineering



<http://synergy.ece.gatech.edu>



# ASTRA-SIM Description

## *System Layer*



**Saeed Rashidi**

Ph.D. Student, School of Electrical & Computer Engineering  
Georgia Institute of Technology

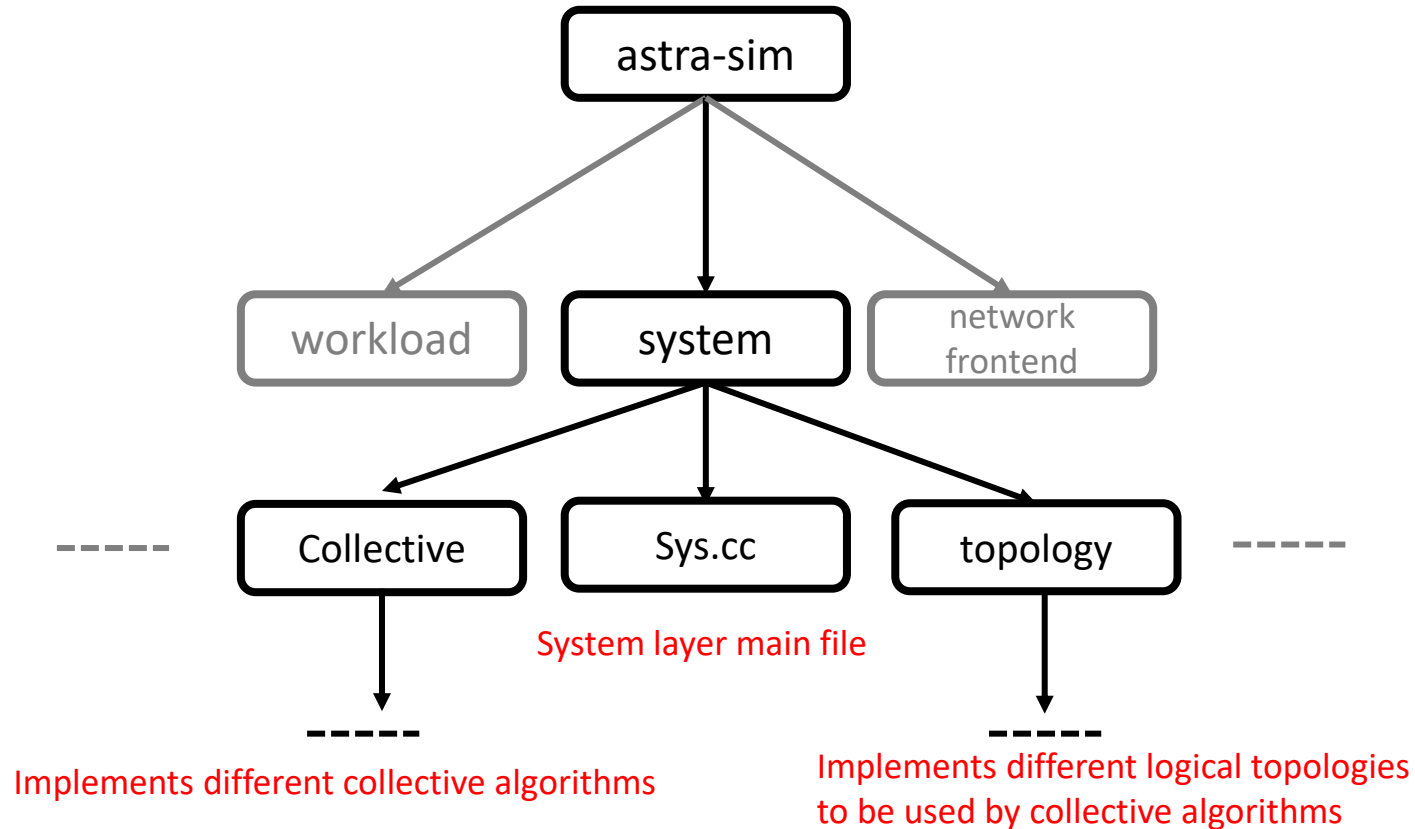
saeed.rashidi@gatech.edu

**Acknowledgments:** Srinivas Sridharan (Meta), Sudarshan Srinivasan (Intel)

# System Layer

# System Layer Collective Implementation

- Code Structure

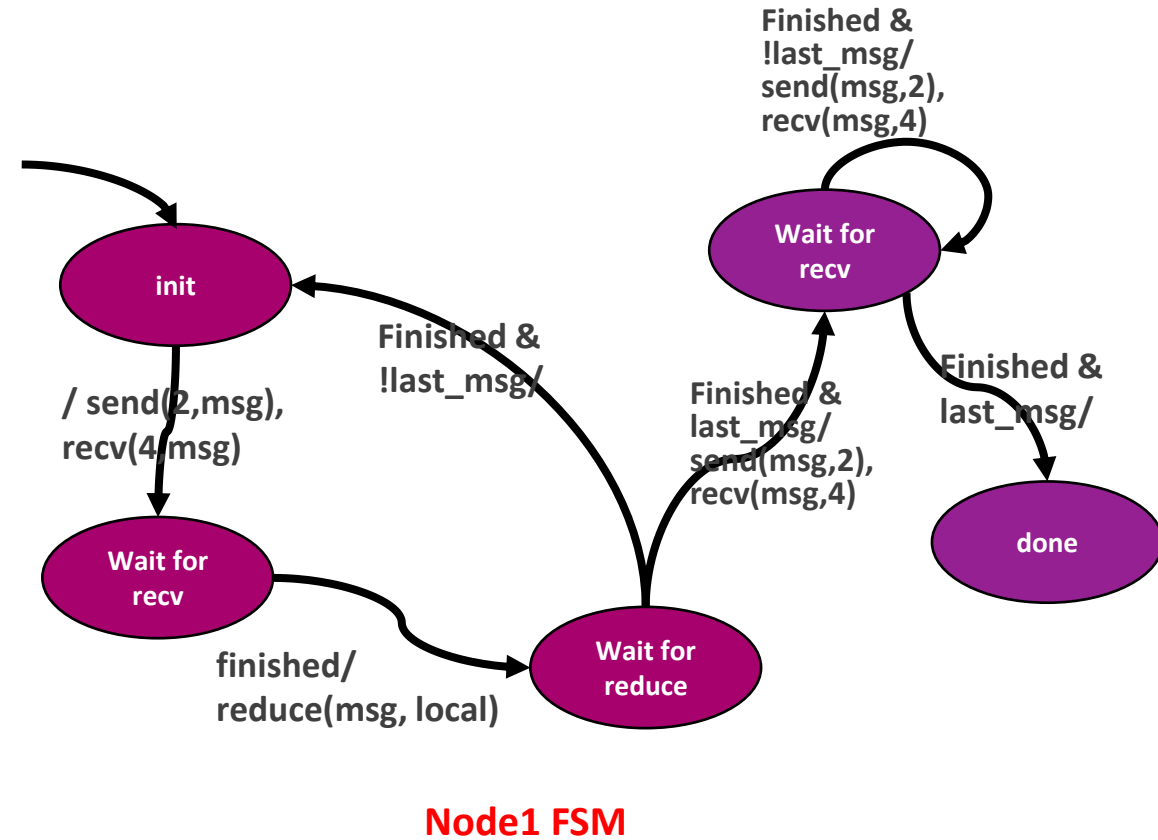
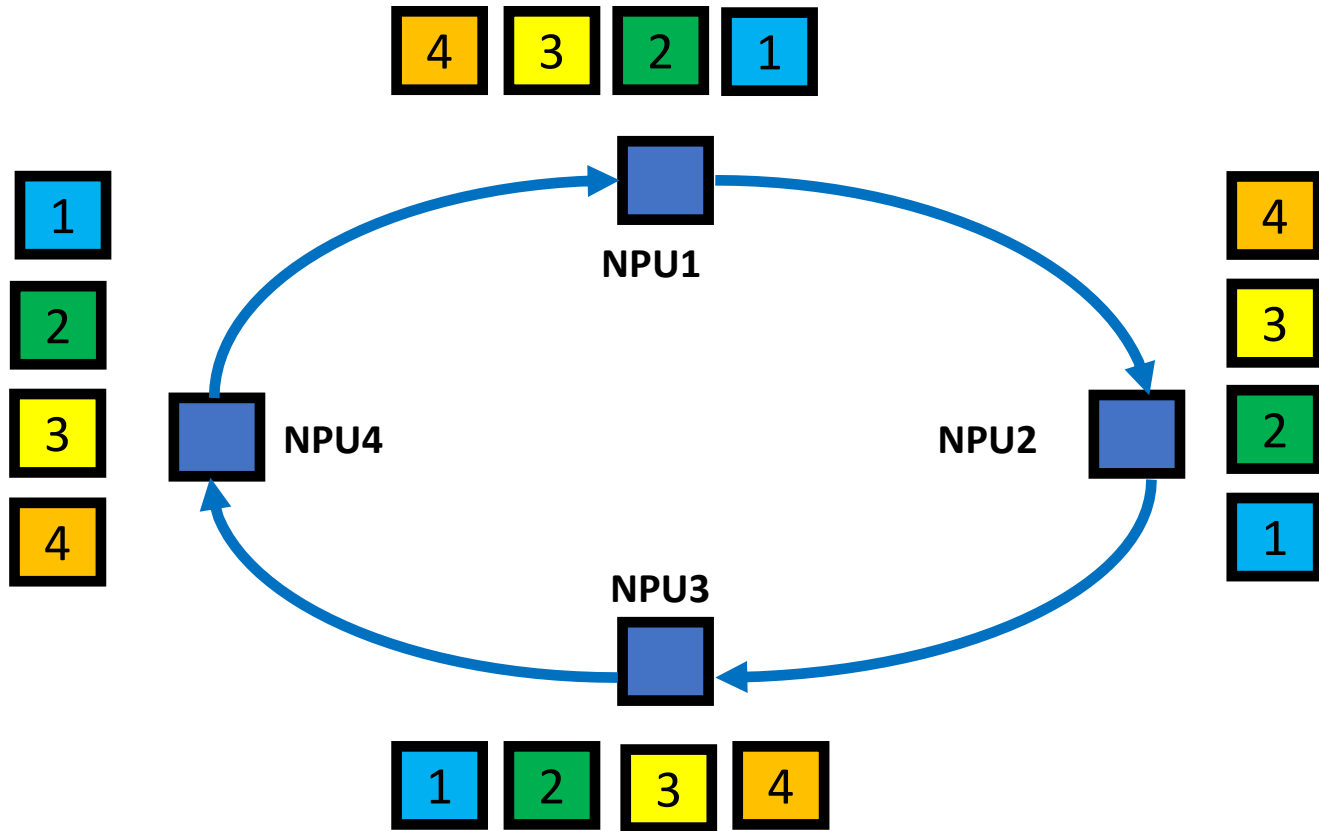


# System Layer Collective Implementation

- Each collective algorithm works based on a logical topology.
- Logical topologies are implemented in “system/topology/\*” and instantiated in sys.cc.
- Collective algorithms are implemented in “system/topology/\*” and instantiated in sys.cc.
- Collective algorithms can be implemented using **state machines**.

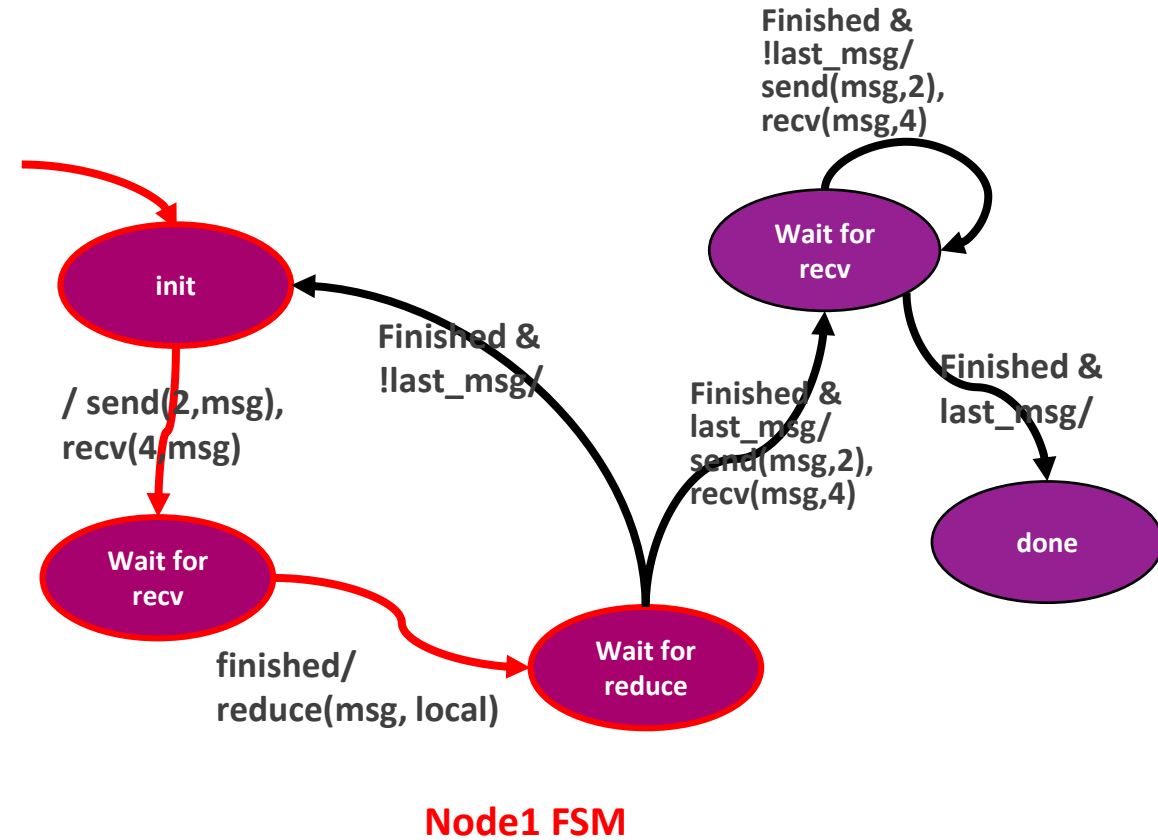
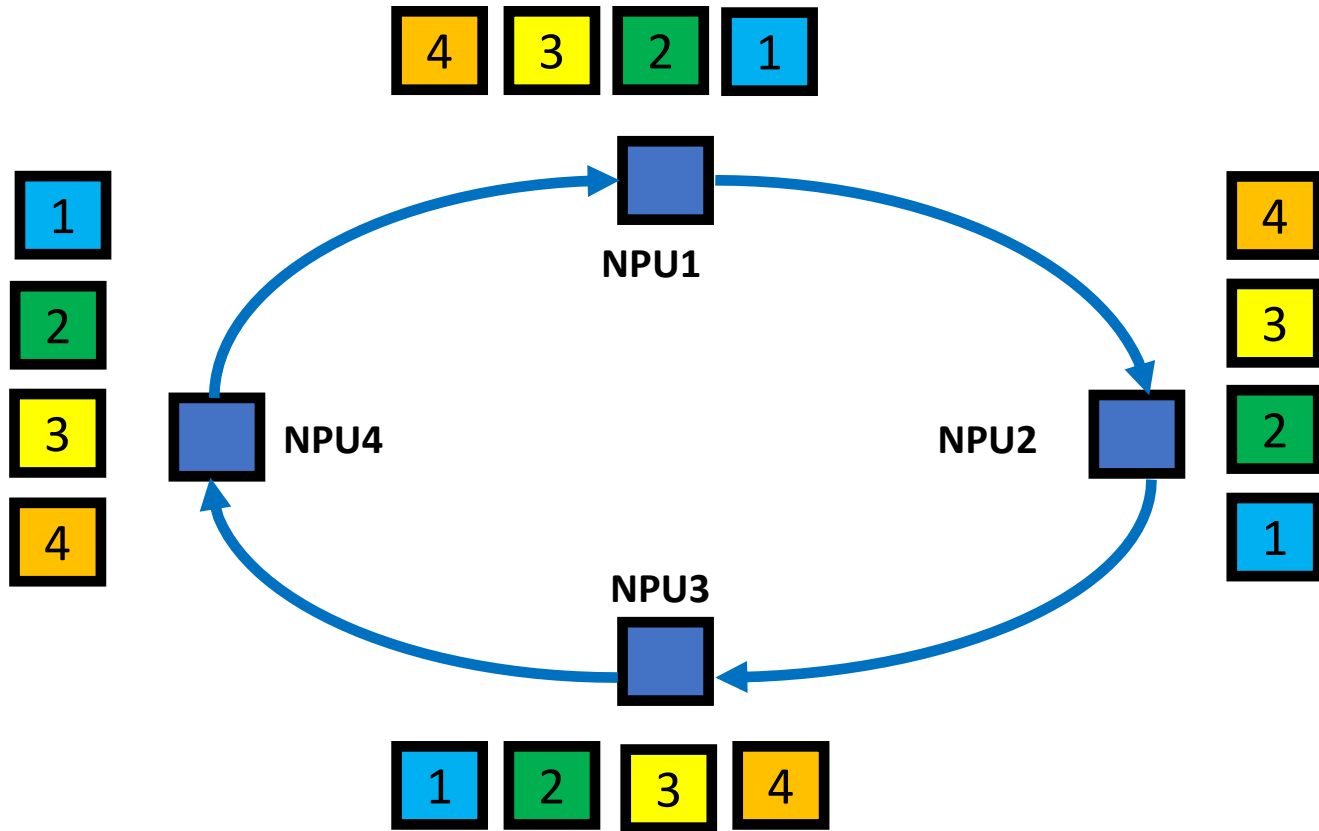
# System Layer Collective Implementation

- Collective algorithms can be implemented using **state machines**.



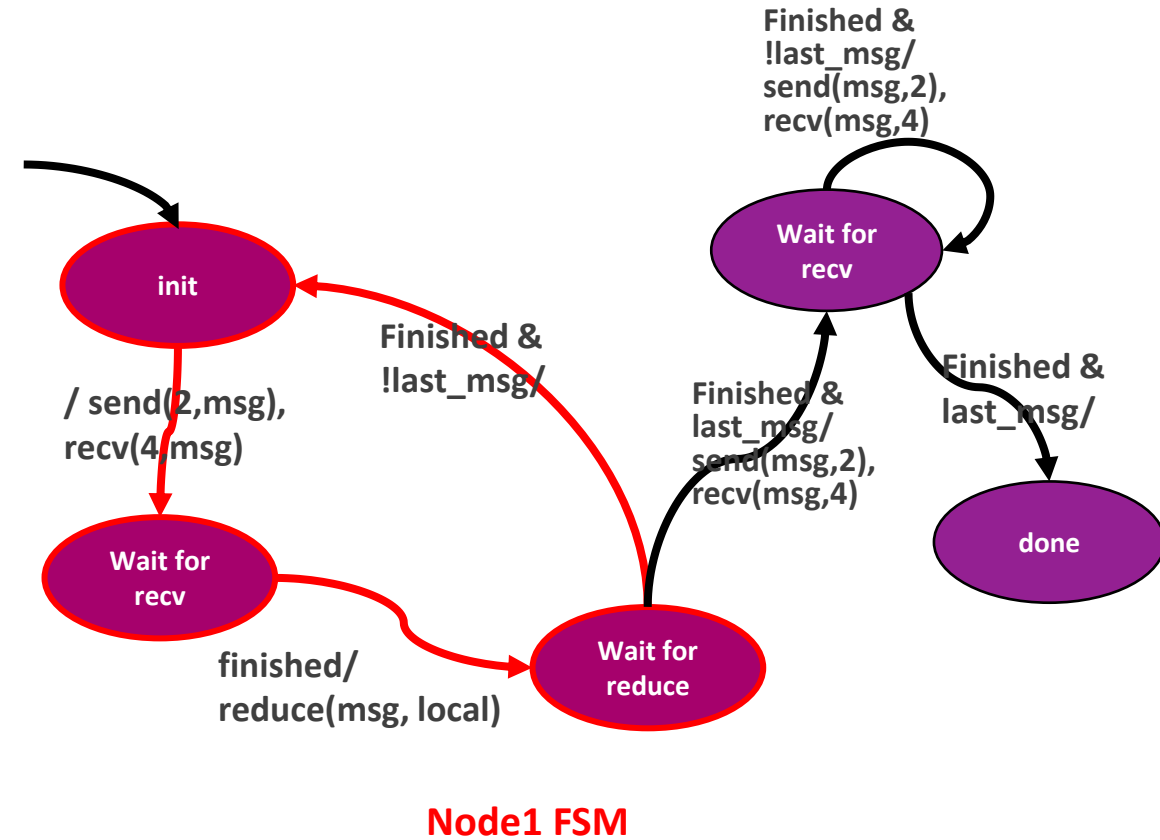
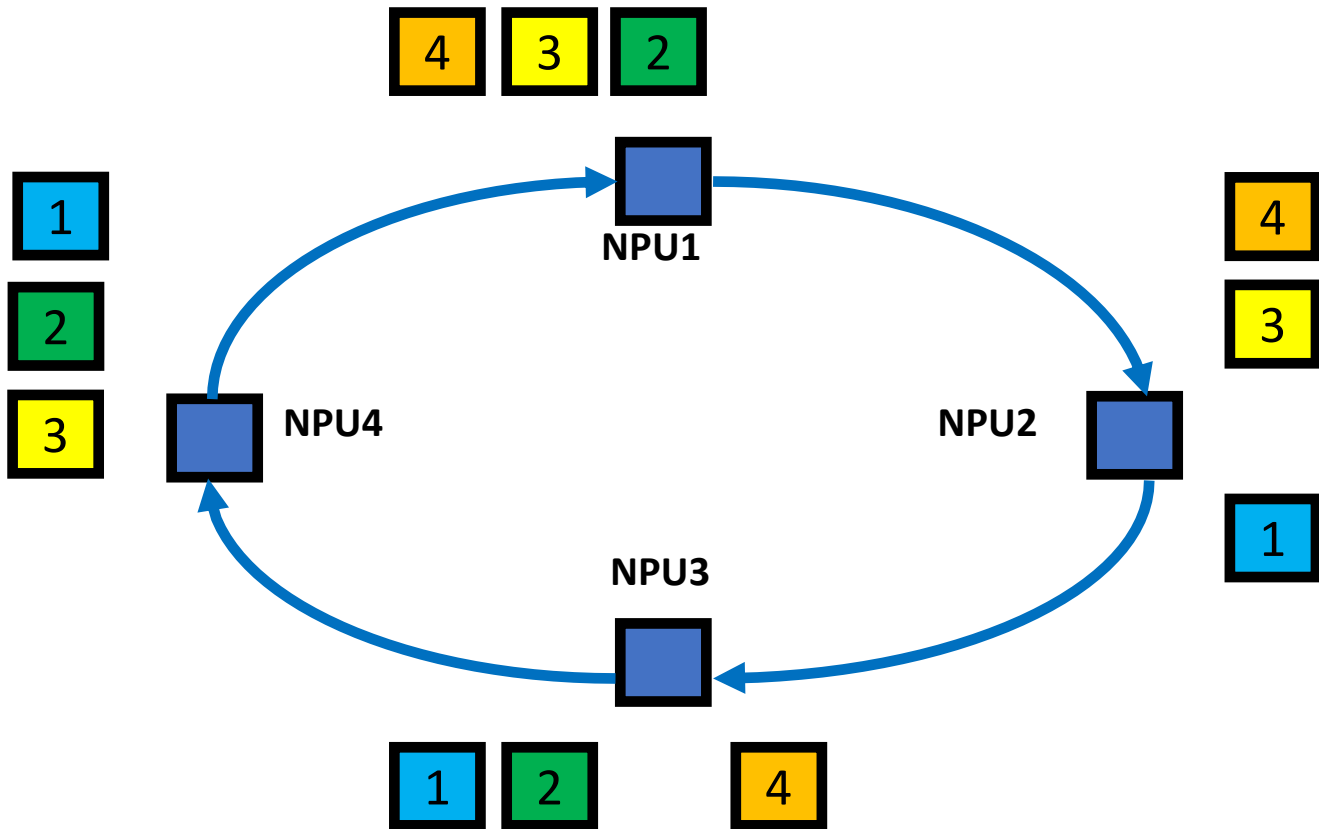
# System Layer Collective Implementation

- Collective algorithms can be implemented using **state machines**.



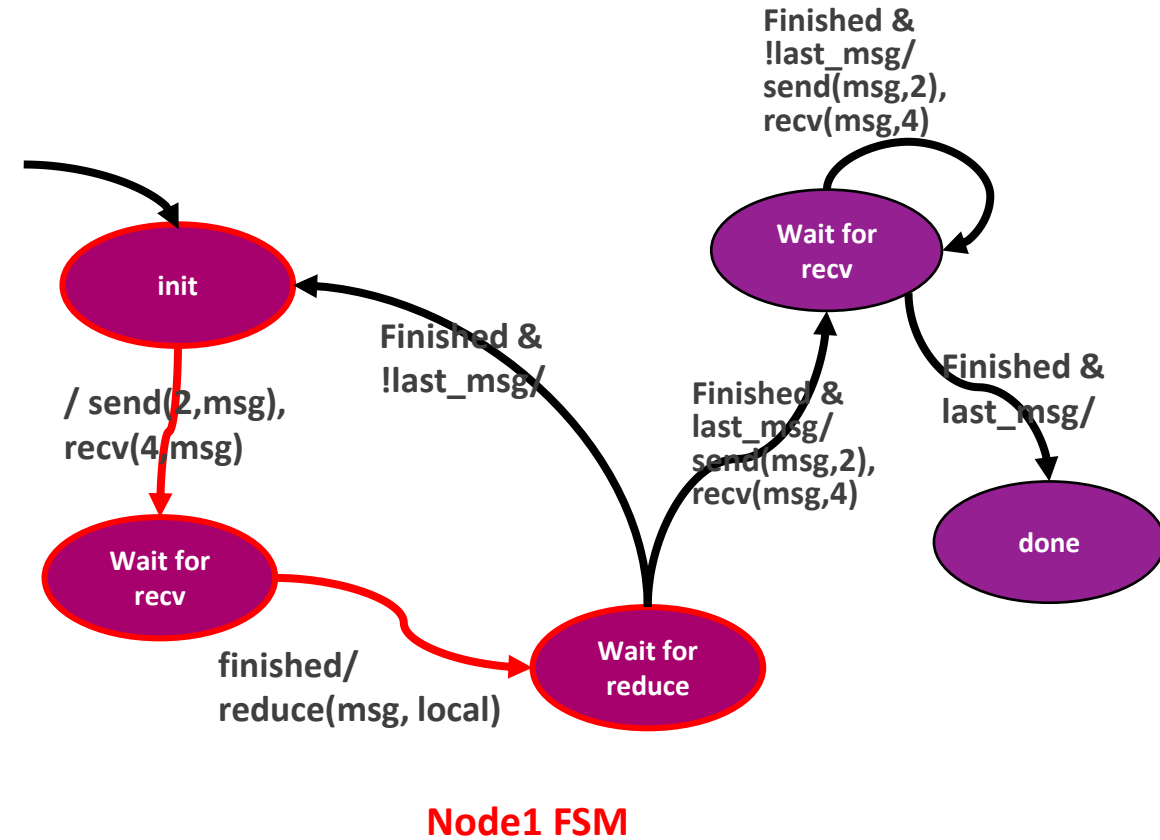
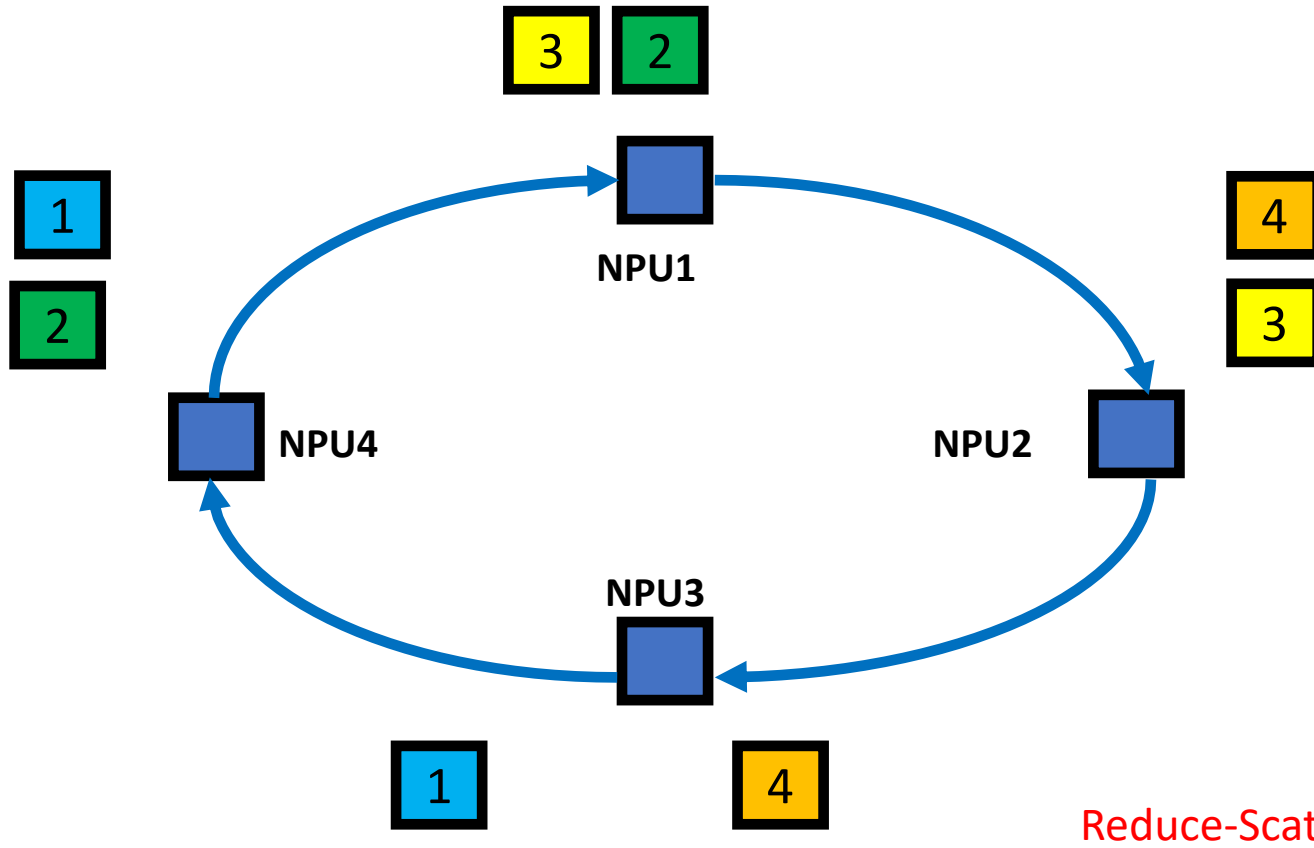
# System Layer Collective Implementation

- Collective algorithms can be implemented using **state machines**.



# System Layer Collective Implementation

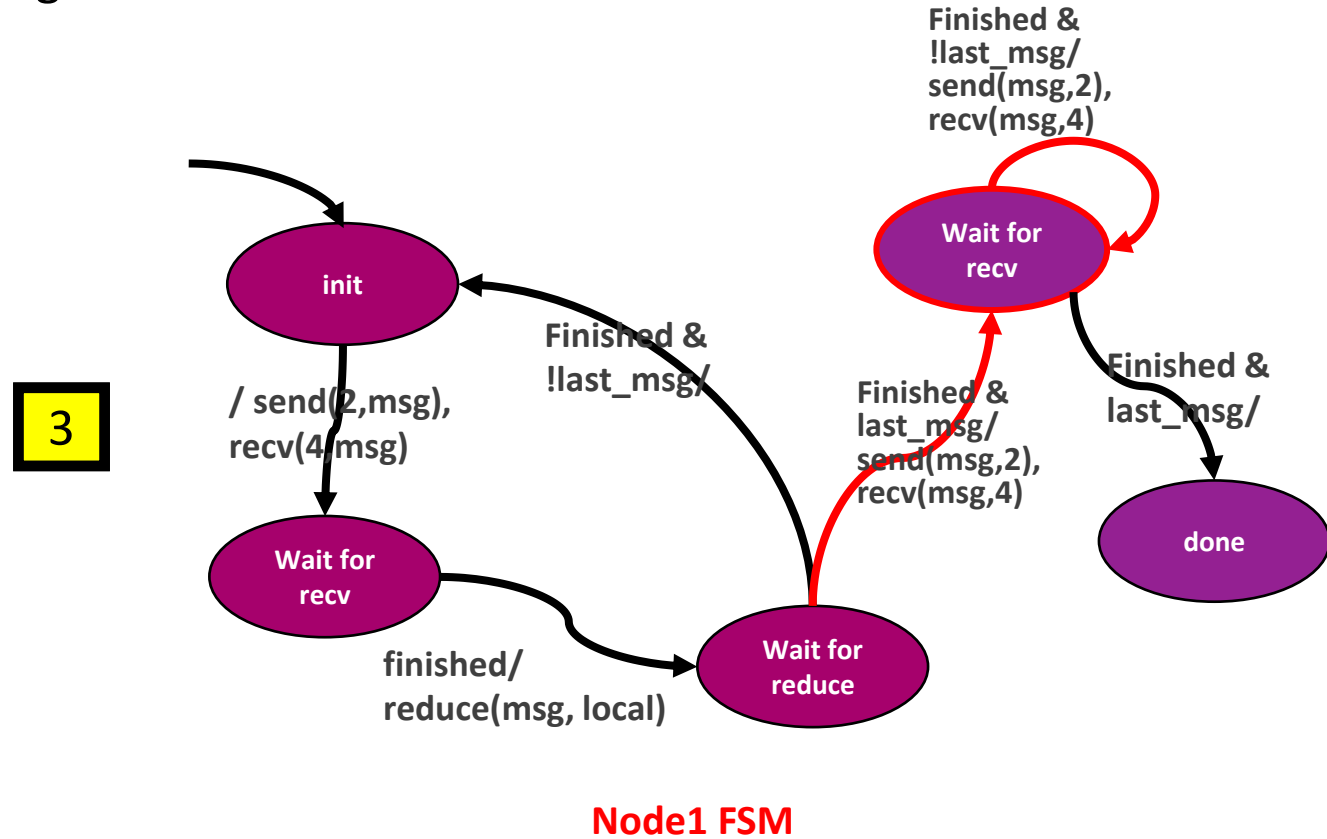
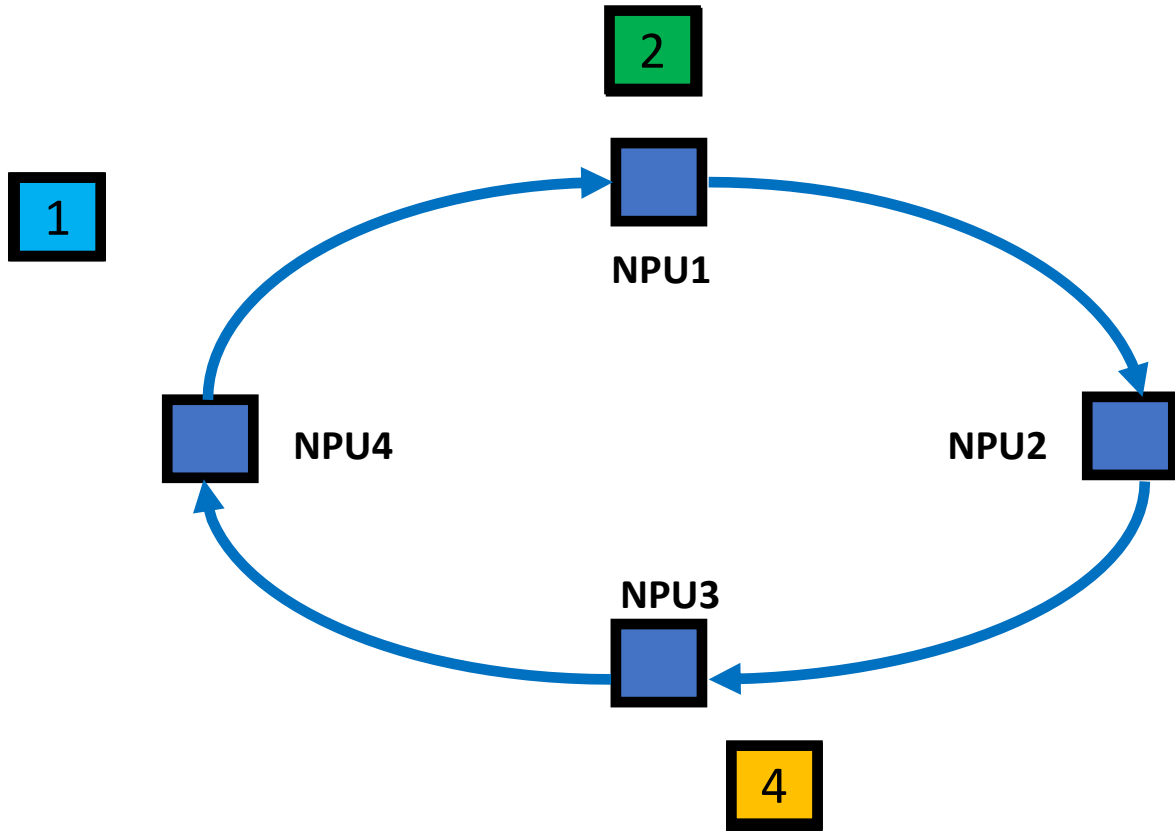
- Collective algorithms can be implemented using **state machines**.





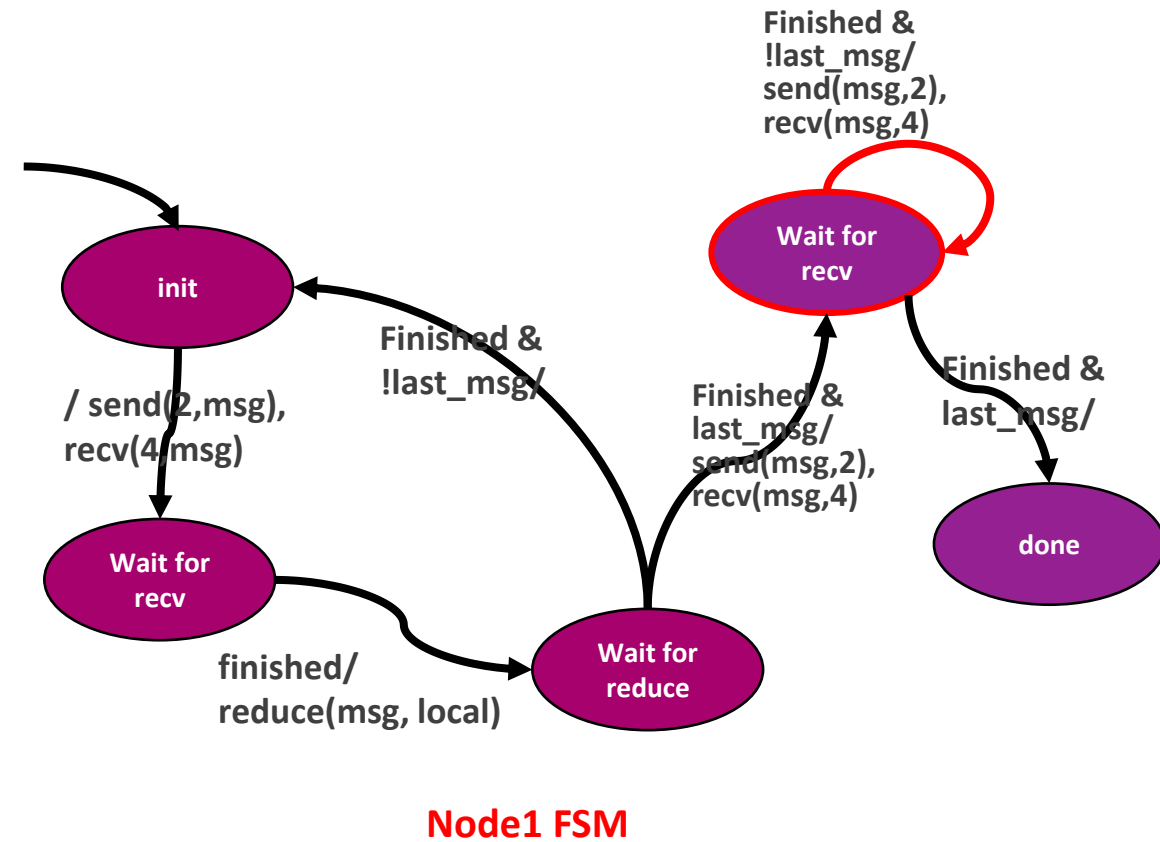
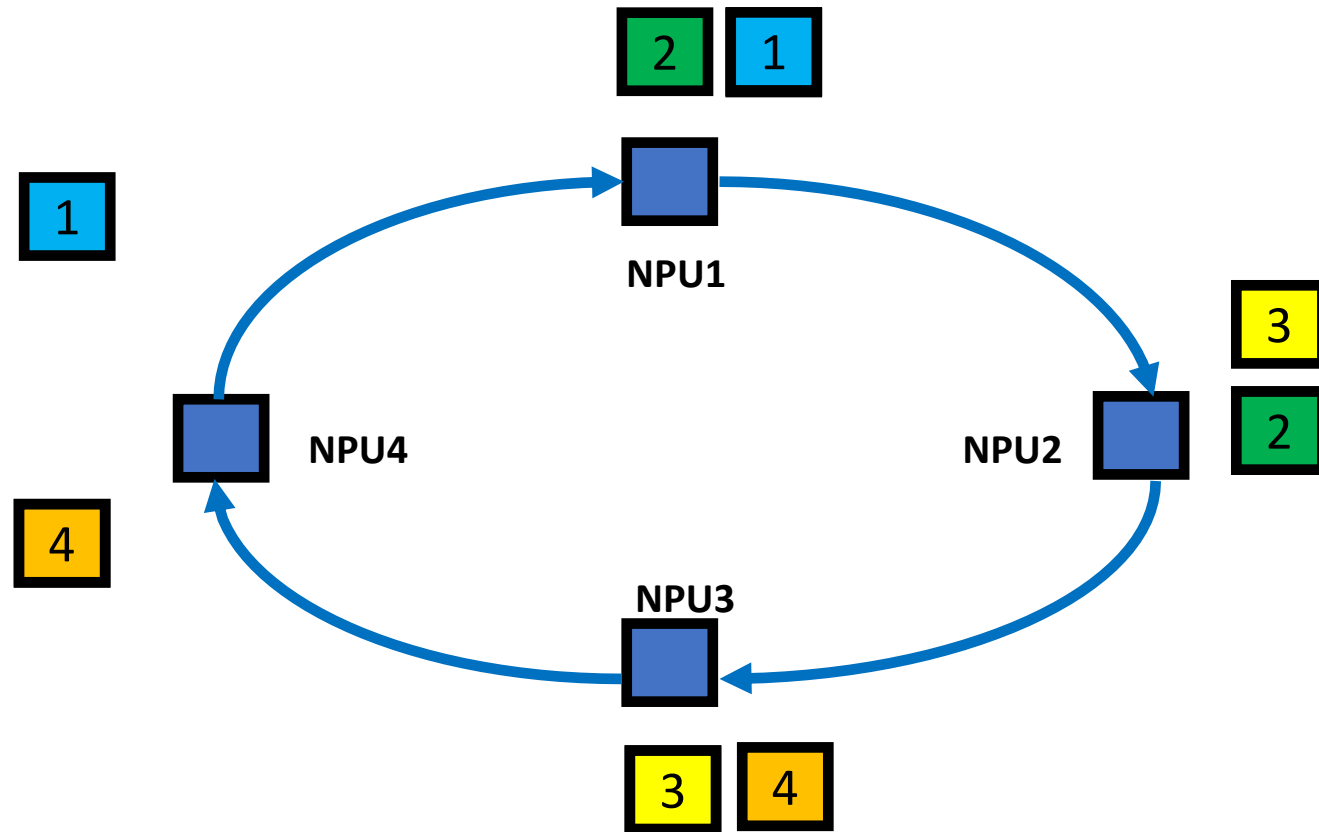
# System Layer Collective Implementation

- Collective algorithms can be implemented using **state machines**.



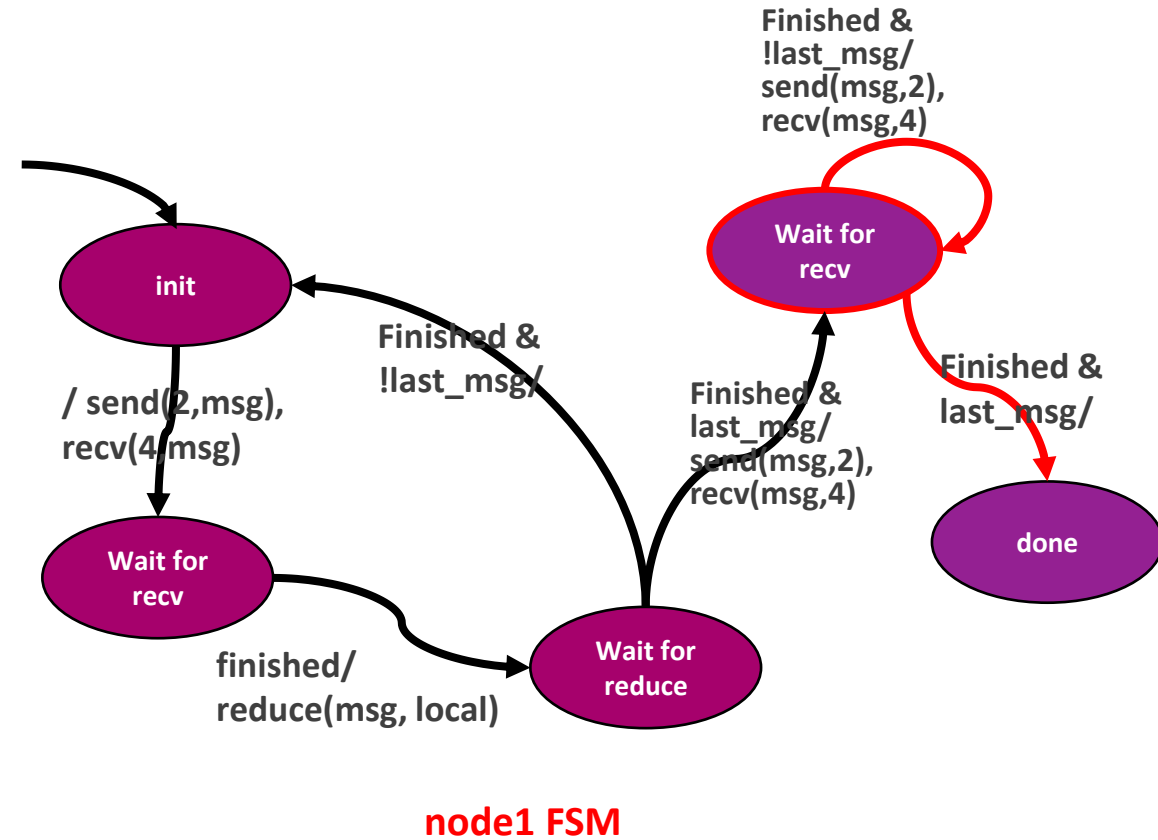
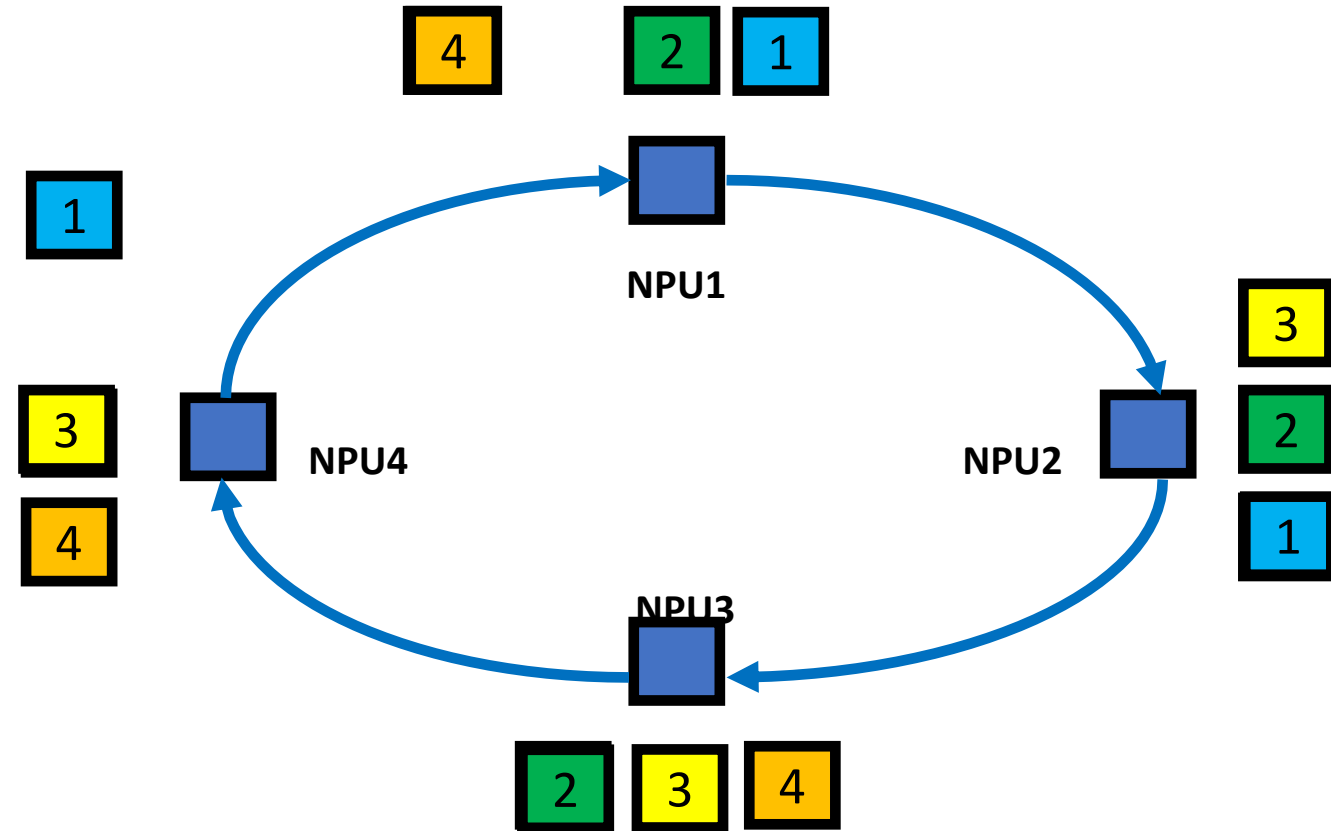
# System Layer Collective Implementation

- Collective algorithms can be implemented using **state machines**.



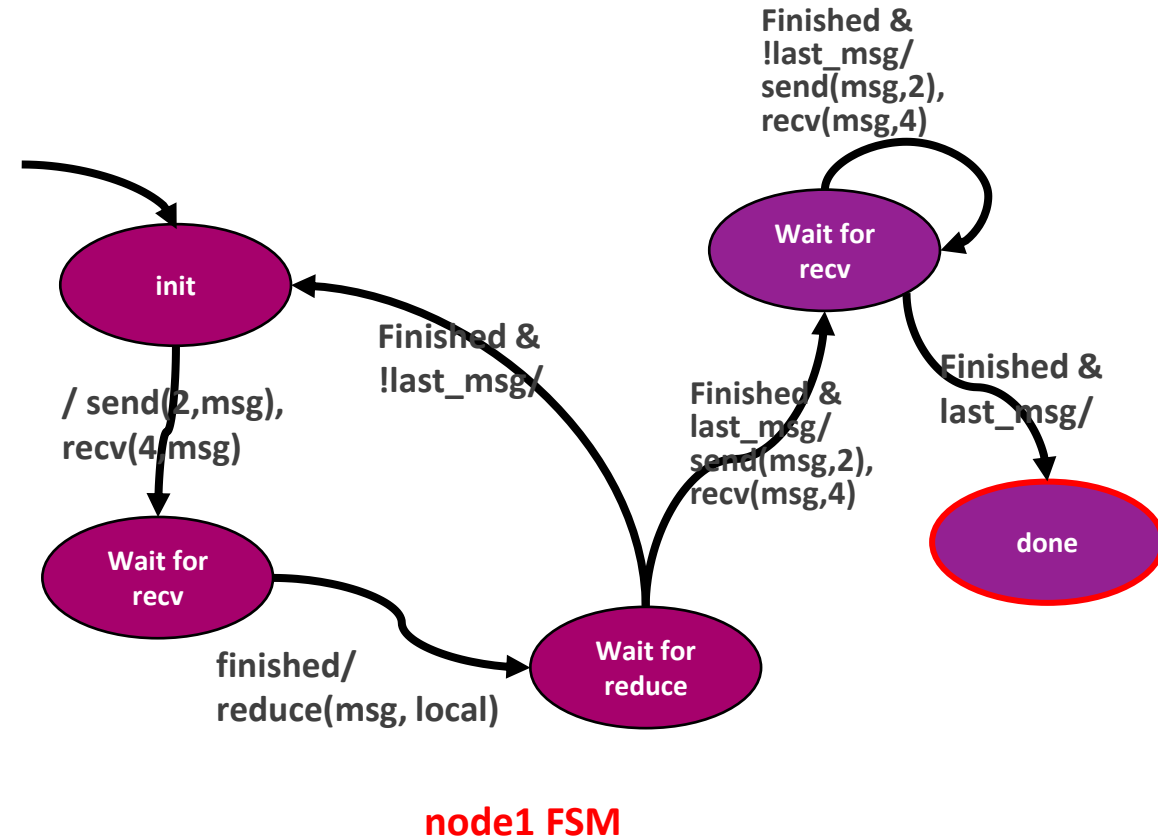
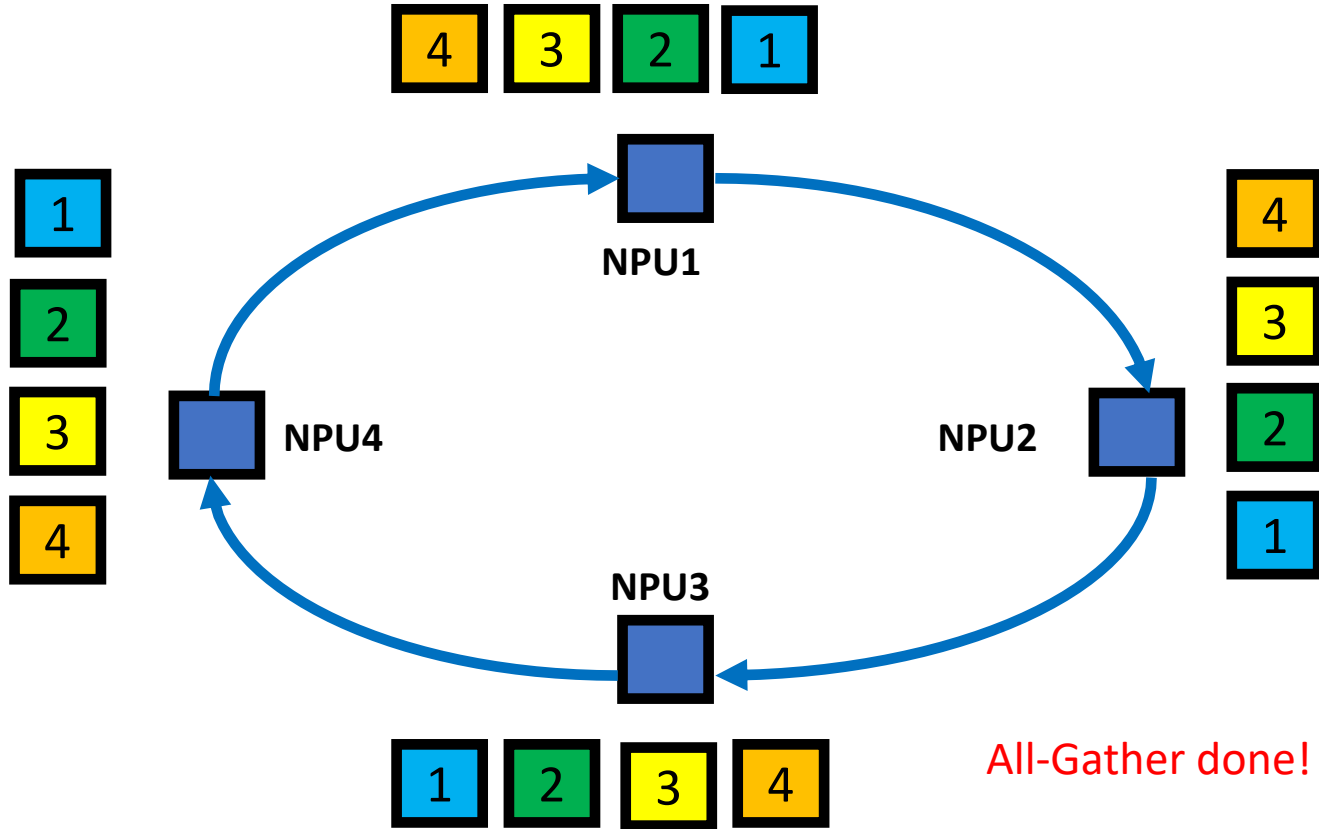
# System Layer Collective Implementation

- Collective algorithms can be implemented using **state machines**.



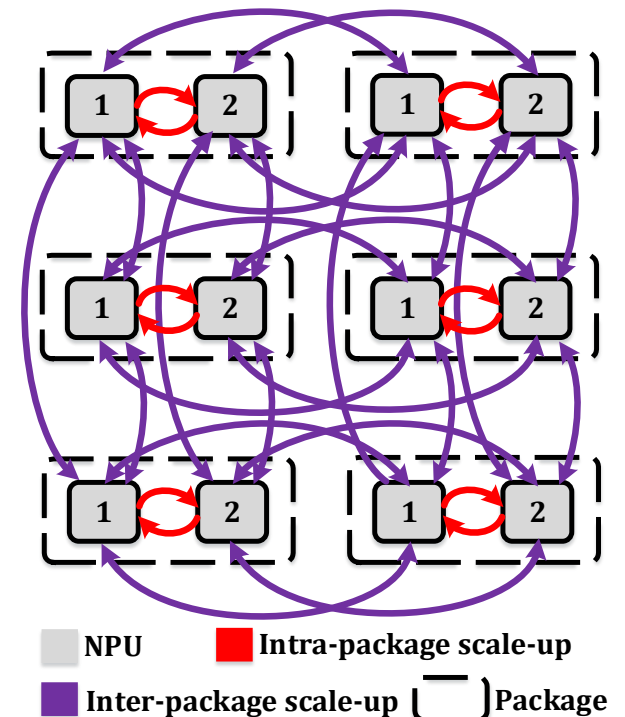
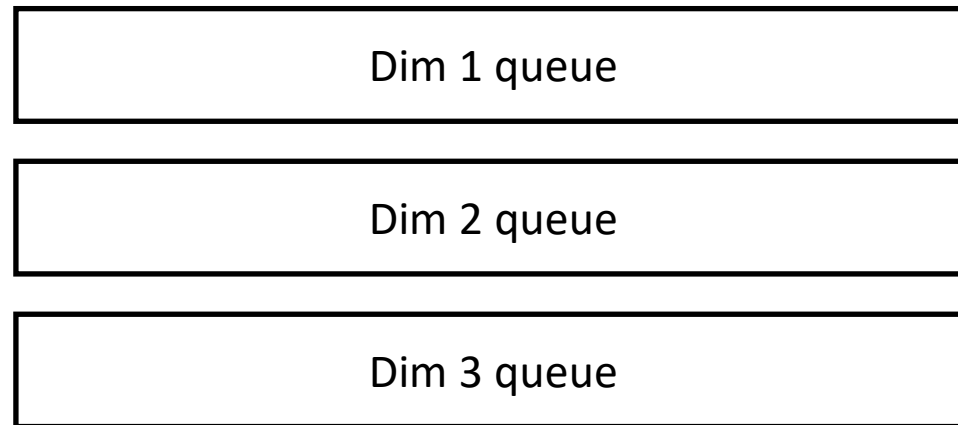
# System Layer Collective Implementation

- Collective algorithms can be implemented using **state machines**.



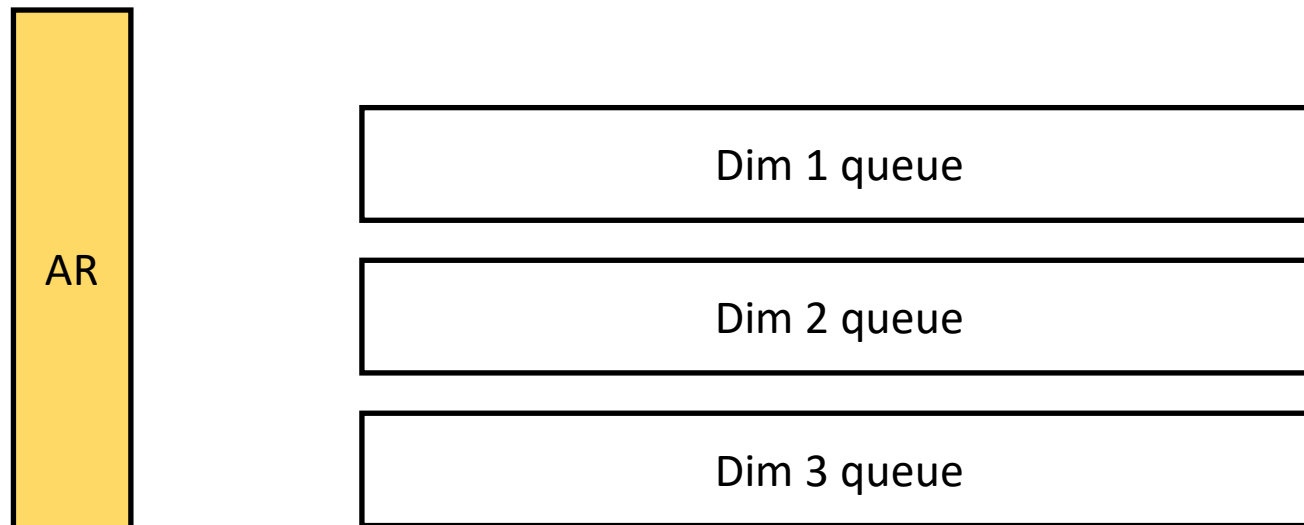
# Collective Scheduler

- There are one/multiple queue(s) per each physical network dimension.
- A collective is broken into multiple chunks and inserted into the first queue.
- Queues process chunks in-order.

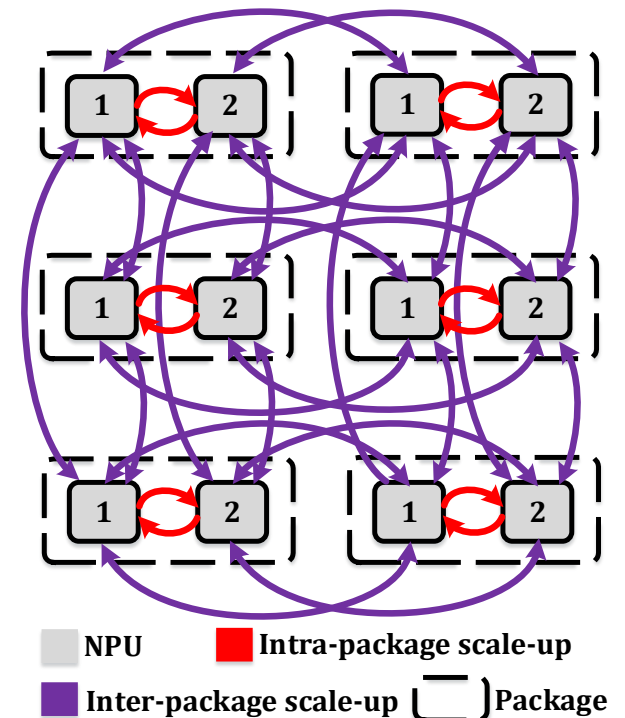


# Collective Scheduler

- There are one/multiple queue(s) per each physical network dimension.
- A collective is broken into multiple chunks and inserted into the first queue.
- Queues process chunks in-order.

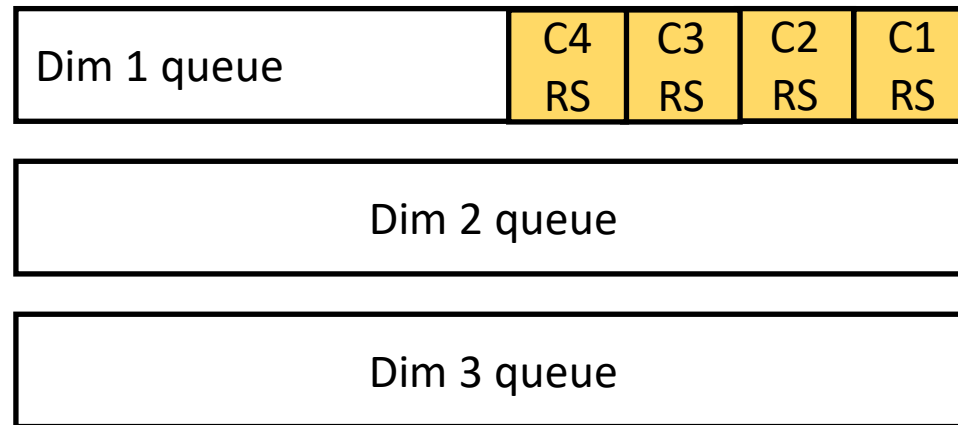


AR: All-Reduce  
RS: Reduce-Scatter  
AG: All-Gather

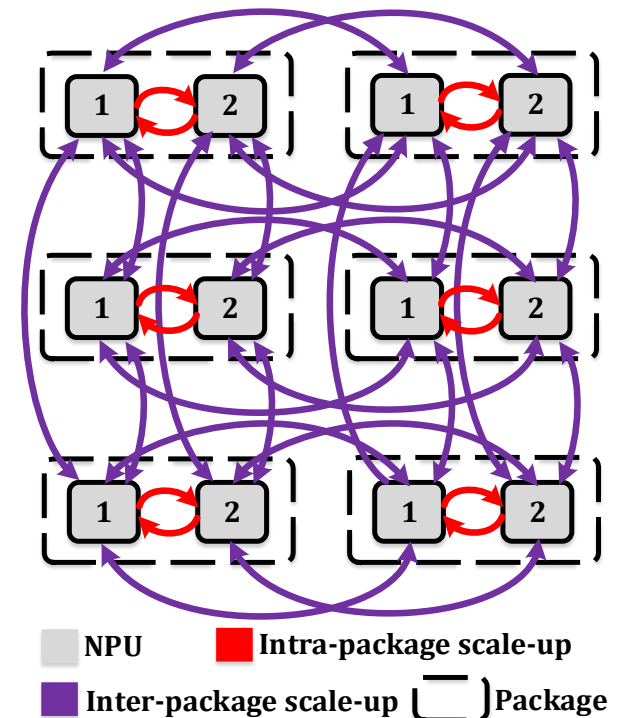


# Collective Scheduler

- There are one/multiple queue(s) per each physical network dimension.
- A collective is broken into multiple chunks and inserted into the first queue.
- Queues process chunks in-order.

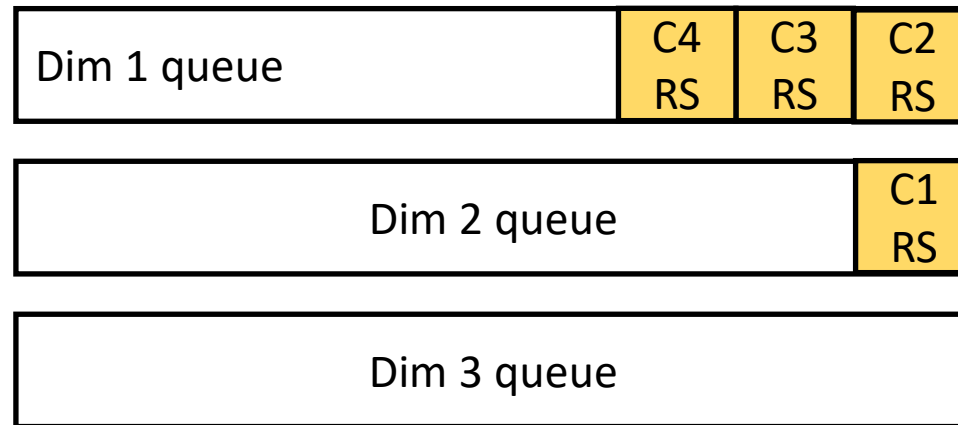


AR: All-Reduce  
RS: Reduce-Scatter  
AG: All-Gather

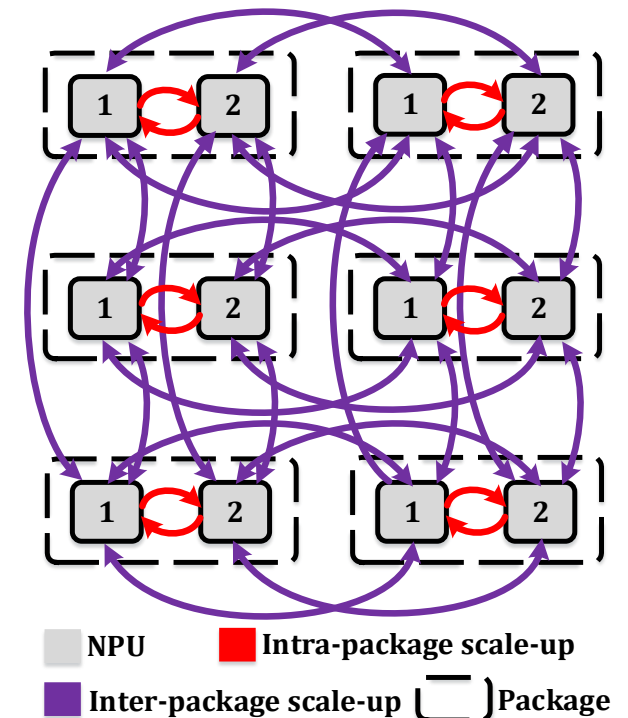


# Collective Scheduler

- There are one/multiple queue(s) per each physical network dimension.
- A collective is broken into multiple chunks and inserted into the first queue.
- Queues process chunks in-order.



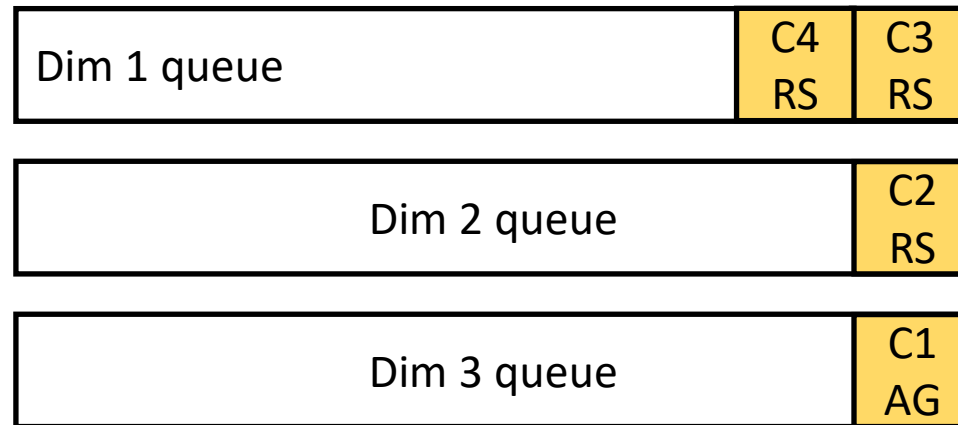
AR: All-Reduce  
RS: Reduce-Scatter  
AG: All-Gather



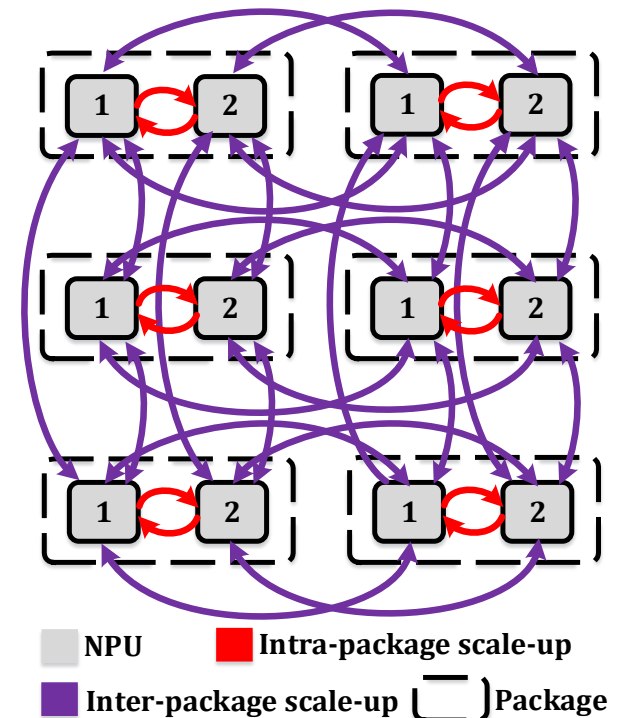


# Collective Scheduler

- There are one/multiple queue(s) per each physical network dimension.
- A collective is broken into multiple chunks and inserted into the first queue.
- Queues process chunks in-order.

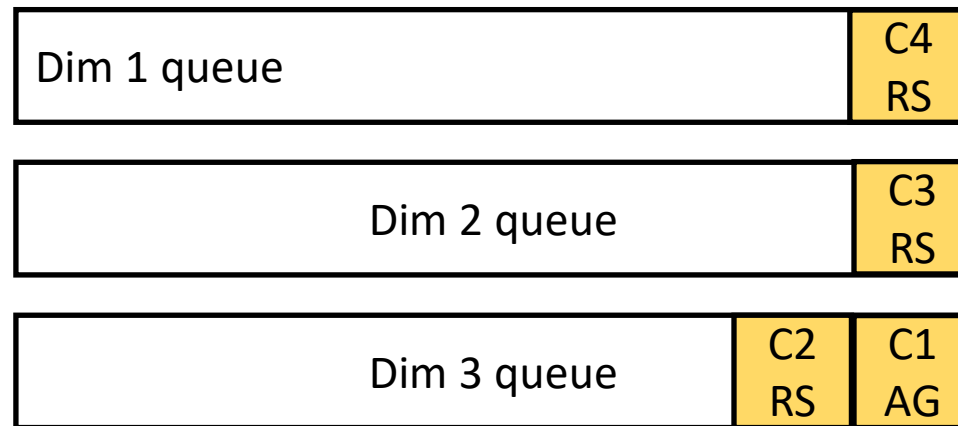


AR: All-Reduce  
RS: Reduce-Scatter  
AG: All-Gather

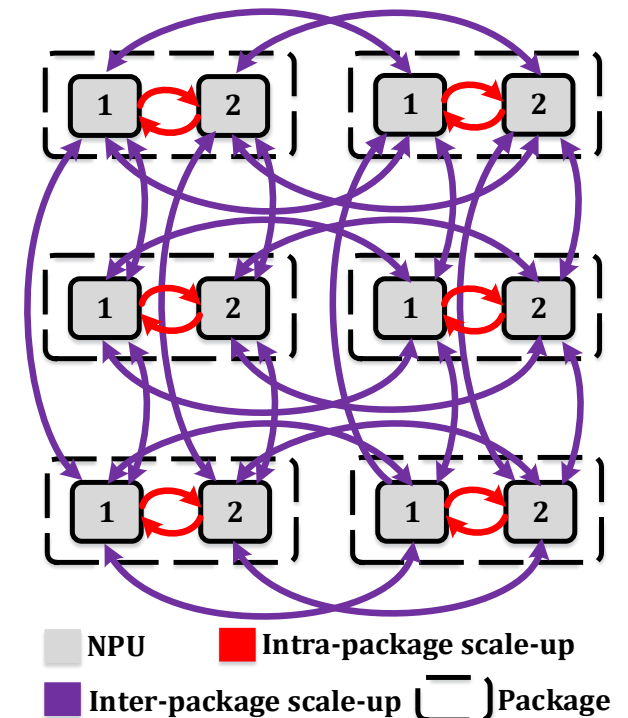


# Collective Scheduler

- There are one/multiple queue(s) per each physical network dimension.
- A collective is broken into multiple chunks and inserted into the first queue.
- Queues process chunks in-order.

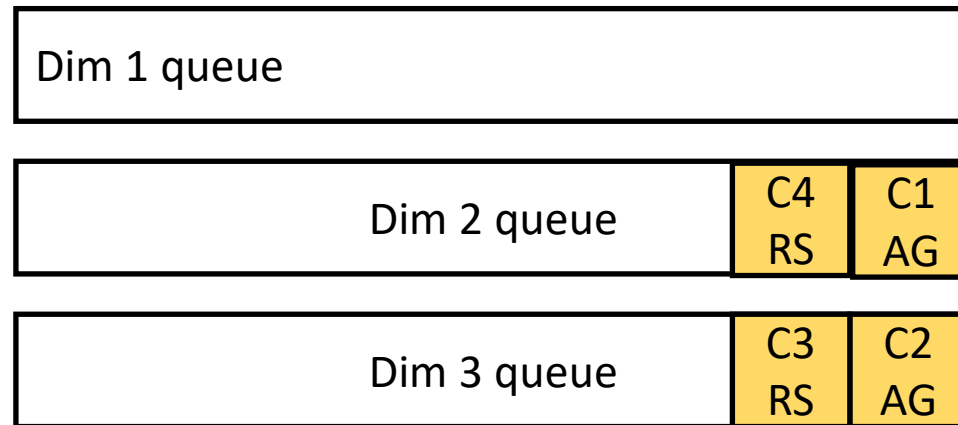


AR: All-Reduce  
RS: Reduce-Scatter  
AG: All-Gather

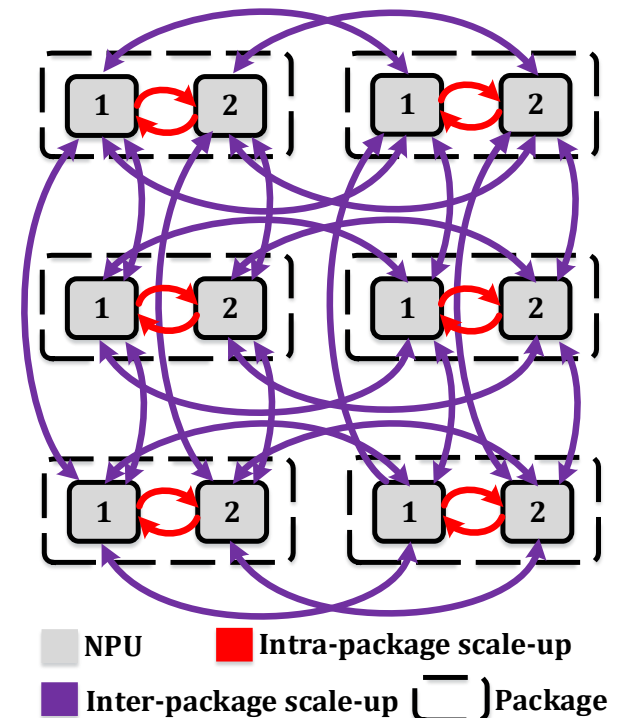


# Collective Scheduler

- There are one/multiple queue(s) per each physical network dimension.
- A collective is broken into multiple chunks and inserted into the first queue.
- Queues process chunks in-order.

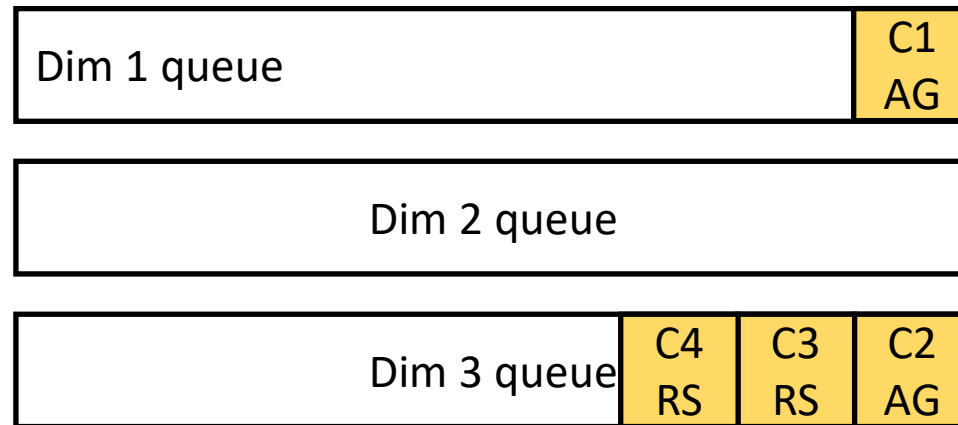


AR: All-Reduce  
RS: Reduce-Scatter  
AG: All-Gather

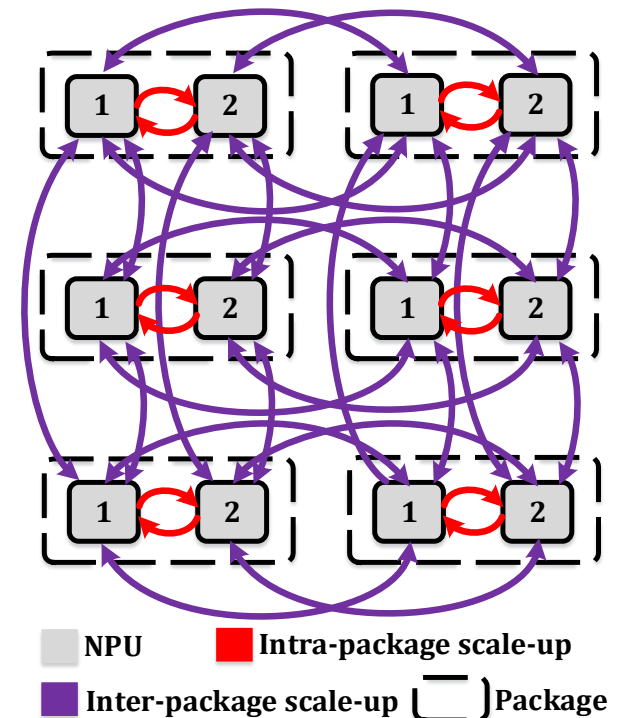


# Collective Scheduler

- There are one/multiple queue(s) per each physical network dimension.
- A collective is broken into multiple chunks and inserted into the first queue.
- Queues process chunks in-order.

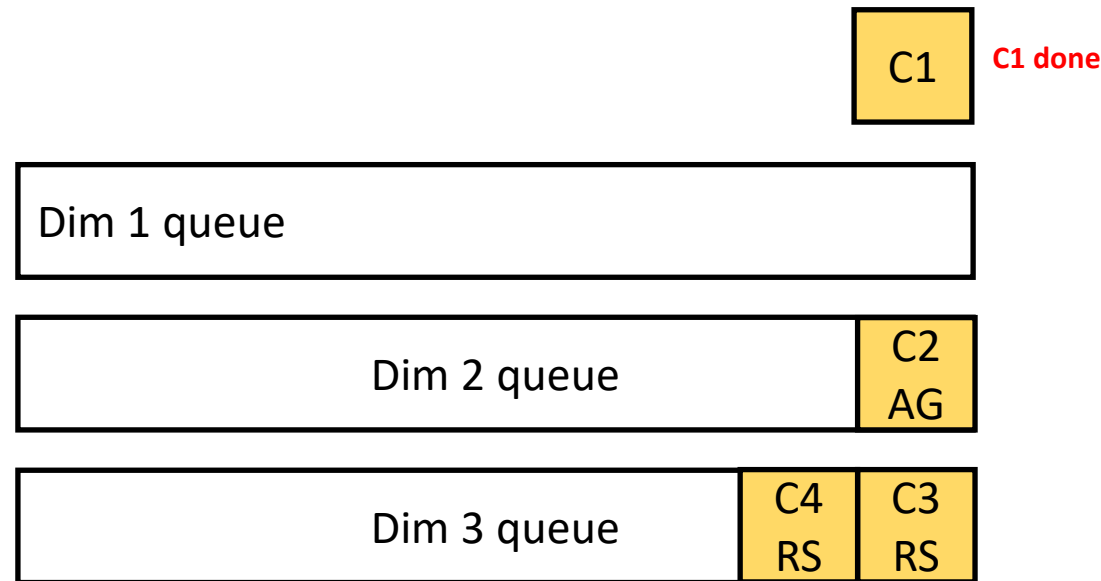


AR: All-Reduce  
RS: Reduce-Scatter  
AG: All-Gather

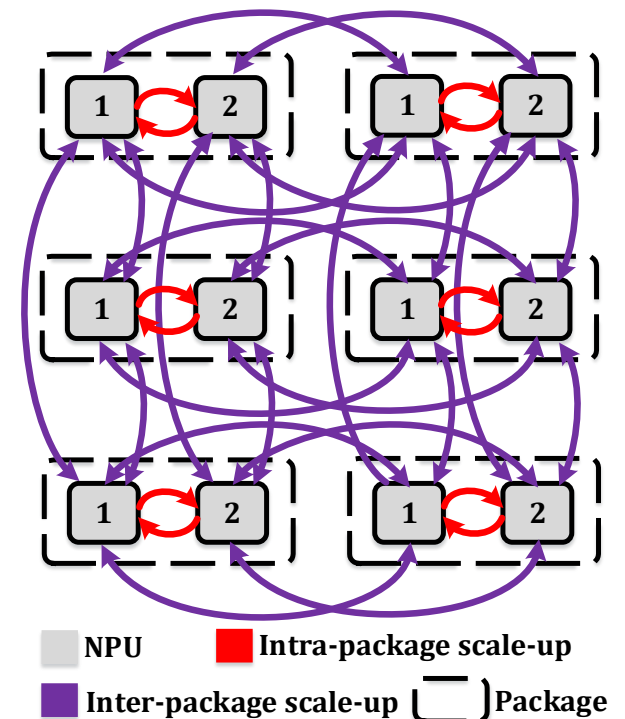


# Collective Scheduler

- There are one/multiple queue(s) per each physical network dimension.
- A collective is broken into multiple chunks and inserted into the first queue.
- Queues process chunks in-order.



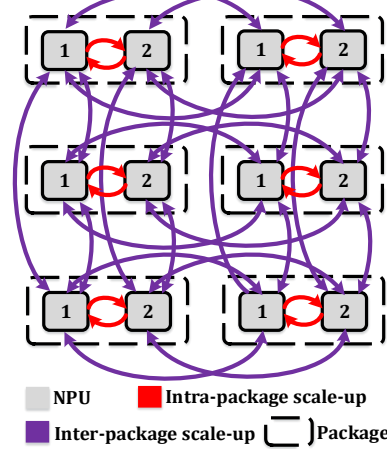
AR: All-Reduce  
RS: Reduce-Scatter  
AG: All-Gather



# System Input

```
sample_torus_sys.txt x
1 scheduling-policy: LIFO
2 endpoint-delay: 1
3 active-chunks-per-dimension: 1
4 preferred-dataset-splits: 4
5 boost-mode: 0
6 all-reduce-implementation: ring_ring_ring
7 all-gather-implementation: ring_ring_ring
8 reduce-scatter-implementation: ring_ring_ring
9 all-to-all-implementation: ring_ring_ring
10 collective-optimization: localBWAware
11
```

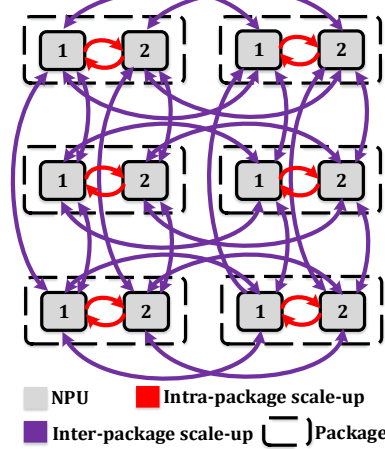
Collective Policy for



# System Input

Constant delay before NPU sending a message

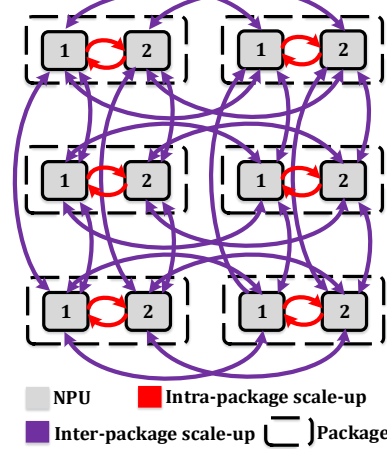
```
sample_torus_sys.txt x
1 scheduling-policy: LIFO
2 endpoint-delay: 1
3 active-chunks-per-dimension: 1
4 preferred-dataset-splits: 4
5 boost-mode: 0
6 all-reduce-implementation: ring_ring_ring
7 all-gather-implementation: ring_ring_ring
8 reduce-scatter-implementation: ring_ring_ring
9 all-to-all-implementation: ring_ring_ring
10 collective-optimization: localBWAware
11
```



# System Input

Max running chunks per each physical network dimension

```
sample_torus_sys.txt x
1 | scheduling-policy: LIFO
2 | endpoint-delay: 1
3 | active-chunks-per-dimension: 1
4 | preferred-dataset-splits: 4
5 | boost-mode: 0
6 | all-reduce-implementation: ring_ring_ring
7 | all-gather-implementation: ring_ring_ring
8 | reduce-scatter-implementation: ring_ring_ring
9 | all-to-all-implementation: ring_ring_ring
10 | collective-optimization: localBWAware
11
```

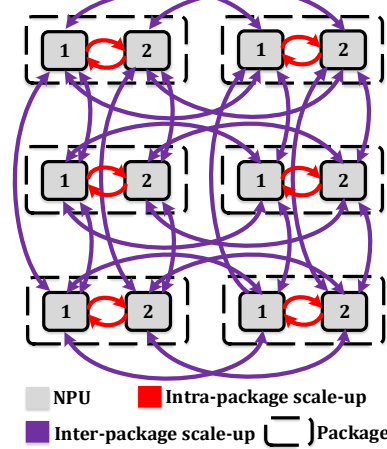




# System Input

# of chunks to split each collective into

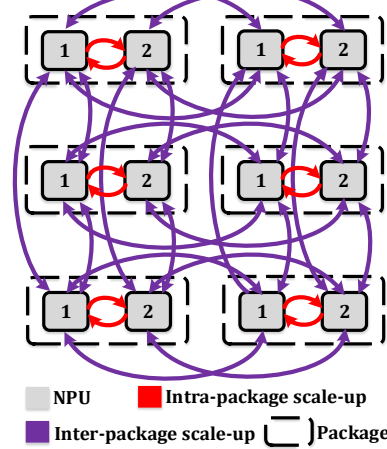
```
sample_torus_sys.txt x
1 |scheduling-policy: LIFO
2 |endpoint-delay: 1
3 |active-chunks-per-dimension: 1
4 |preferred-dataset-splits: 4|
5 |boost-mode: 0
6 |all-reduce-implementation: ring_ring_ring
7 |all-gather-implementation: ring_ring_ring
8 |reduce-scatter-implementation: ring_ring_ring
9 |all-to-all-implementation: ring_ring_ring
10|collective-optimization: localBWAware
11|
```



# System Input

```
sample_torus_sys.txt x
1 scheduling-policy: LIFO
2 endpoint-delay: 1
3 active-chunks-per-dimension: 1
4 preferred-dataset-splits: 4
5 boost-mode: 0
6 all-reduce-implementation: ring_ring_ring
7 all-gather-implementation: ring_ring_ring
8 reduce-scatter-implementation: ring_ring_ring
9 all-to-all-implementation: ring_ring_ring
10 collective-optimization: localBWAware
11
```

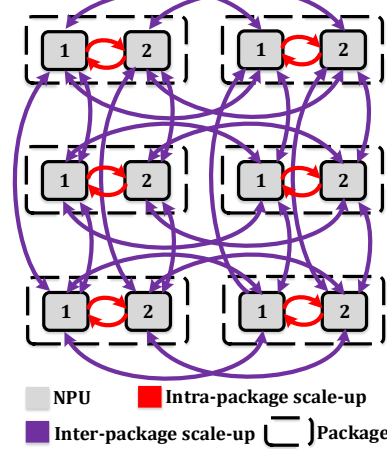
Speed-up the simulation



# System Input

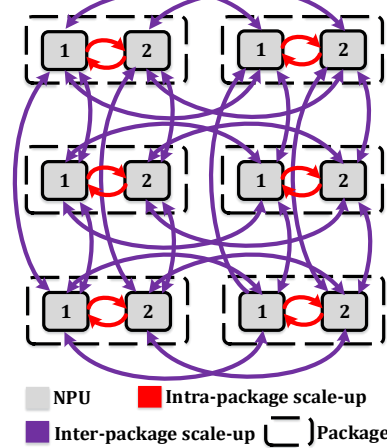
Hierarchical collective algorithm implementation

```
sample_torus_sys.txt x
1 | scheduling-policy: LIFO
2 | endpoint-delay: 1
3 | active-chunks-per-dimension: 1
4 | preferred-dataset-splits: 4
5 | boost-mode: 0
6 | all-reduce-implementation: ring_ring_ring
7 | all-gather-implementation: ring_ring_ring
8 | reduce-scatter-implementation: ring_ring_ring
9 | all-to-all-implementation: ring_ring_ring
10 | collective-optimization: localBWAware
11
```



# Additional System Inputs

```
sample_torus_sys.txt x
1 scheduling-policy: LIFO
2 endpoint-delay: 1
3 active-chunks-per-dimension: 1
4 preferred-dataset-splits: 4
5 boost-mode: 0
6 all-reduce-implementation: ring_ring_ring
7 all-gather-implementation: ring_ring_ring
8 reduce-scatter-implementation: ring_ring_ring
9 all-to-all-implementation: ring_ring_ring
10 collective-optimization: localBWAware
11
12 inter-dimension-scheduling: themis
13 intra-dimension-scheduling: SCF
14
```



**Opportunities for intelligent communication scheduling at the system layer**

S. Rashidi et al., "Themis: A Network Bandwidth-Aware Collective Scheduling Policy for Distributed Training of DL Models". **ISCA 2022.**

# Thank you!