Asynchronous Broadcast on the Intel SCC using Interrupts

Darko Petrović, **Omid Shahmirzadi**,
Thomas Ropars, André Schiper

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Synchronous VS Asynchronous Communication

Synchronous	Asynchronous
Poll based reception	Interrupt based reception

Context and Motivation

SCC Communication Libraries	Synchronous	Asynchronous
Point to point comm.	RCCE, IRCCE	SCC port of Barrelfish, S-NET
Group comm.	RCKMPI, RCCE_COMM, OC_BCAST [SPAA2012]	?

- Inter-Processor-Interrupts (IPI) were shown to be costly in asynchronous p2p communication.

→ Can IPIs be used efficiently to implement asynchronous group communication, e.g. broadcast?

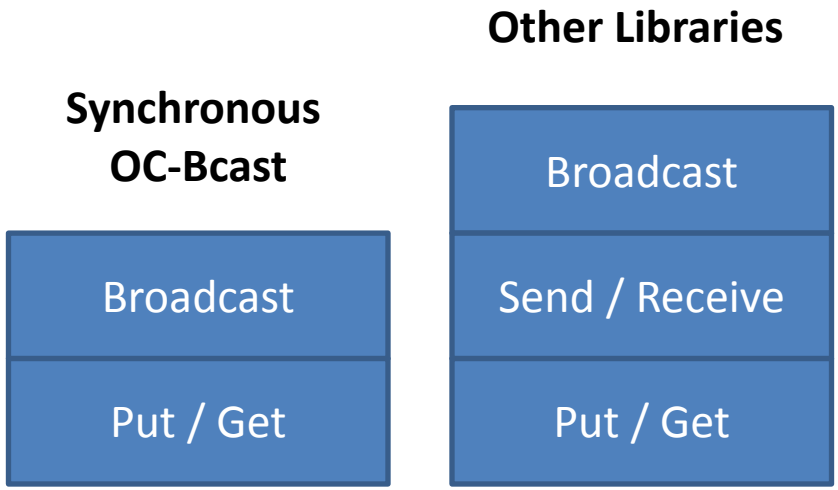
Contributions

- An efficient asynchronous broadcast based on the synchronous OC-Bcast
- A user-space library to work with IPIs

Outline

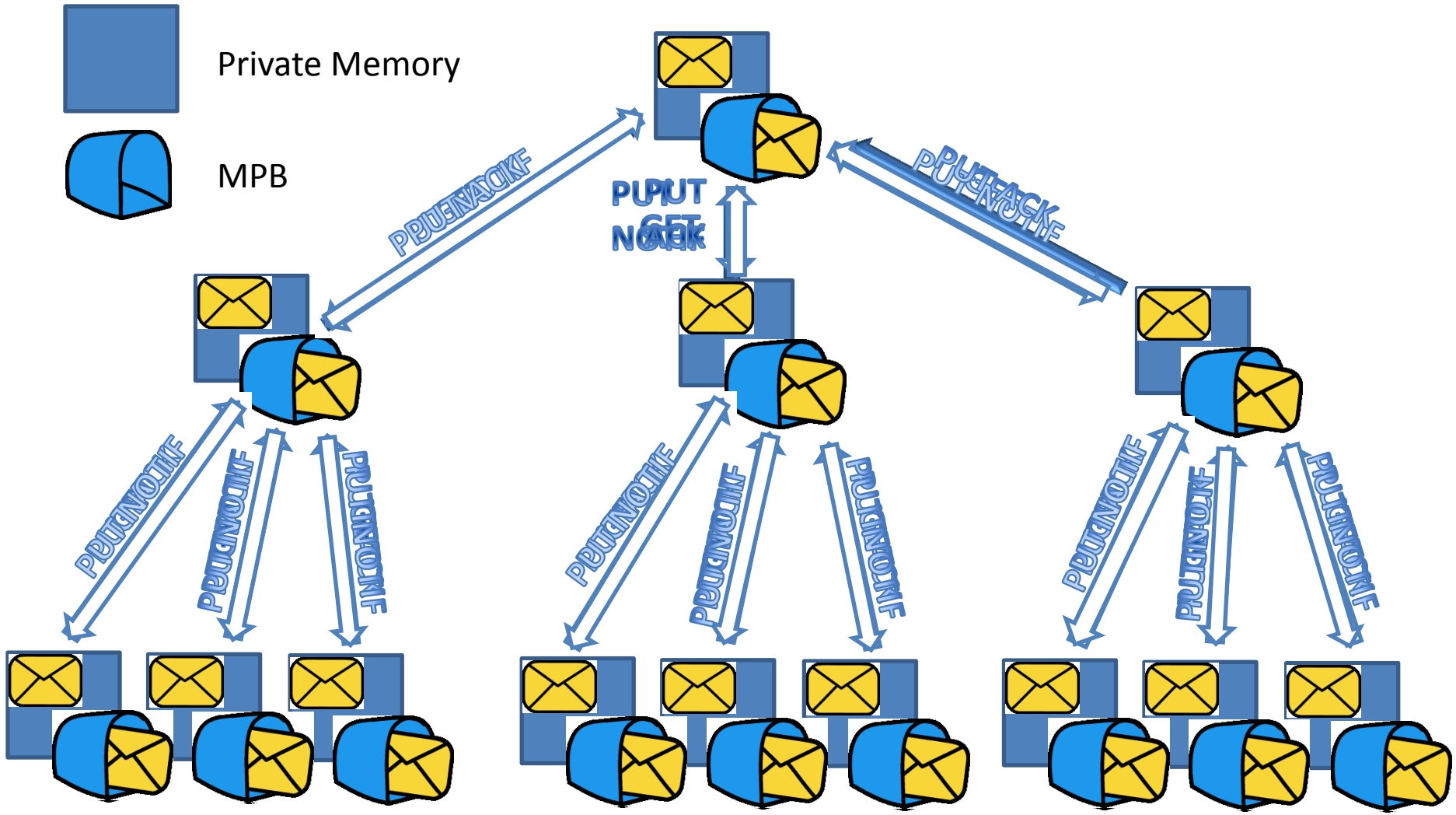
- Synchronous OC-Bcast overview
- Interrupt handling hardware
- Asynchronous broadcast based on OC-Bcast
- Evaluation

Synchronous OC-Bcast: Basic Idea



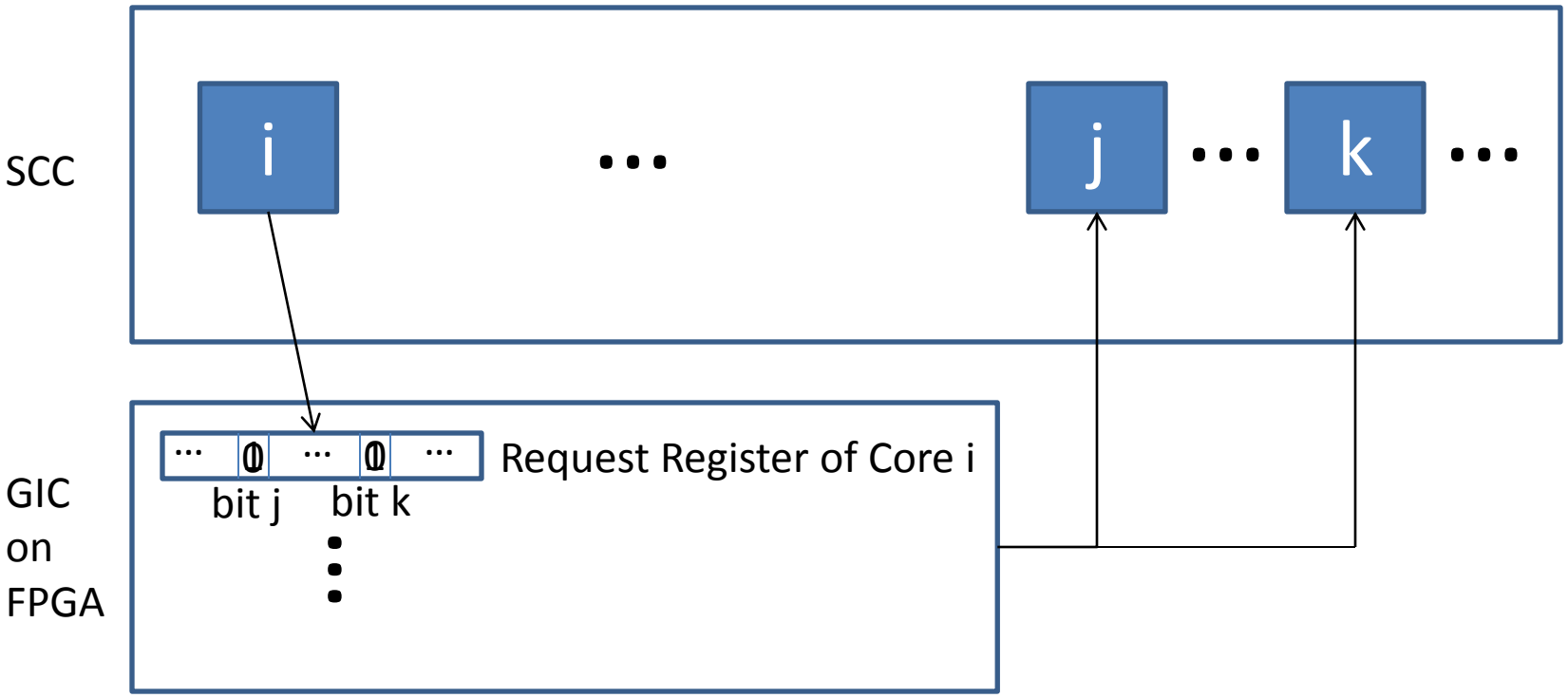
High Performance: 3 times better throughput and at least 27% better latency than the best available broadcast solutions for the SCC.

Synchronous OC-Bcast: Mechanism



Interrupt Handling Hardware (1)

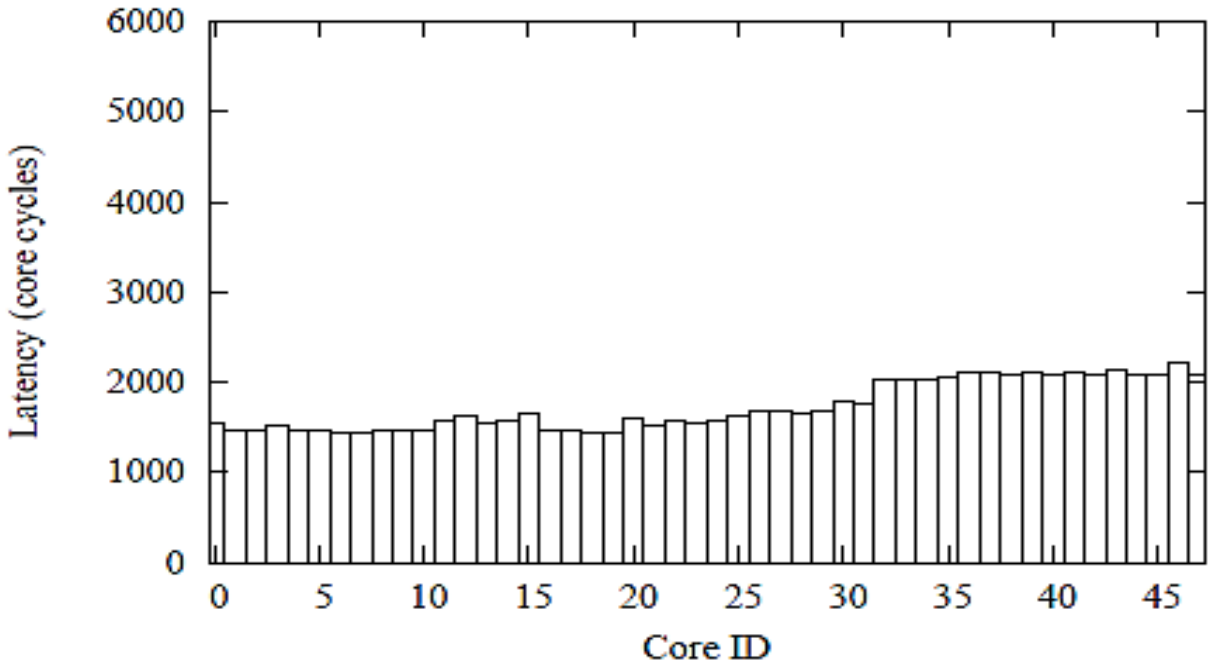
- IPI mechanism: using Global Interrupt Controller (GIC) on the FPGA



- We provide a user space library to work with IPIs.

Interrupt Handling Hardware (2)

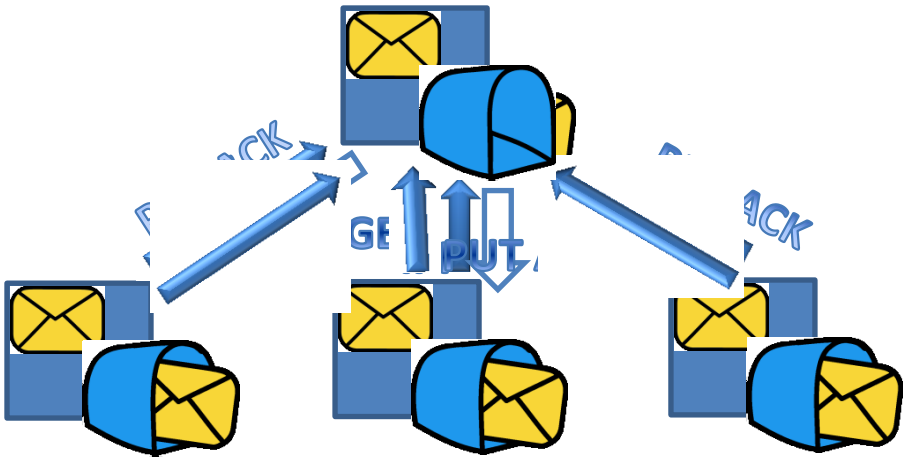
- Latency of point to point IPI: ~ 2000 core cycles
- Experiment: each core sends an interrupt to all other cores, including itself, and measures the time when it receives its own interrupt in its kernel space.



→ IPIs could be sent in parallel to many cores with the cost of a p2p IPI.

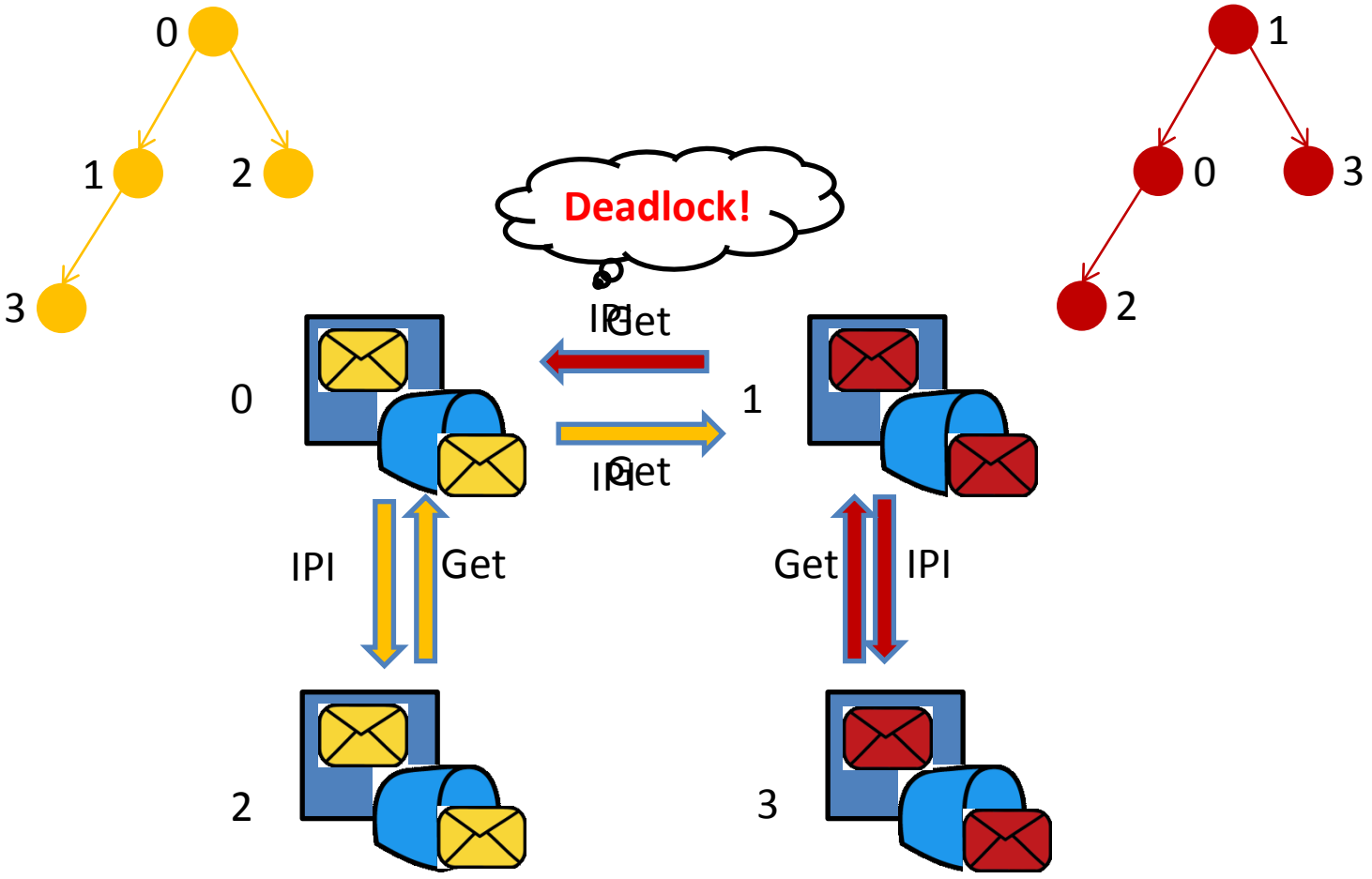
Asynchronous Broadcast

Idea: using parallel IPIs to notify the children



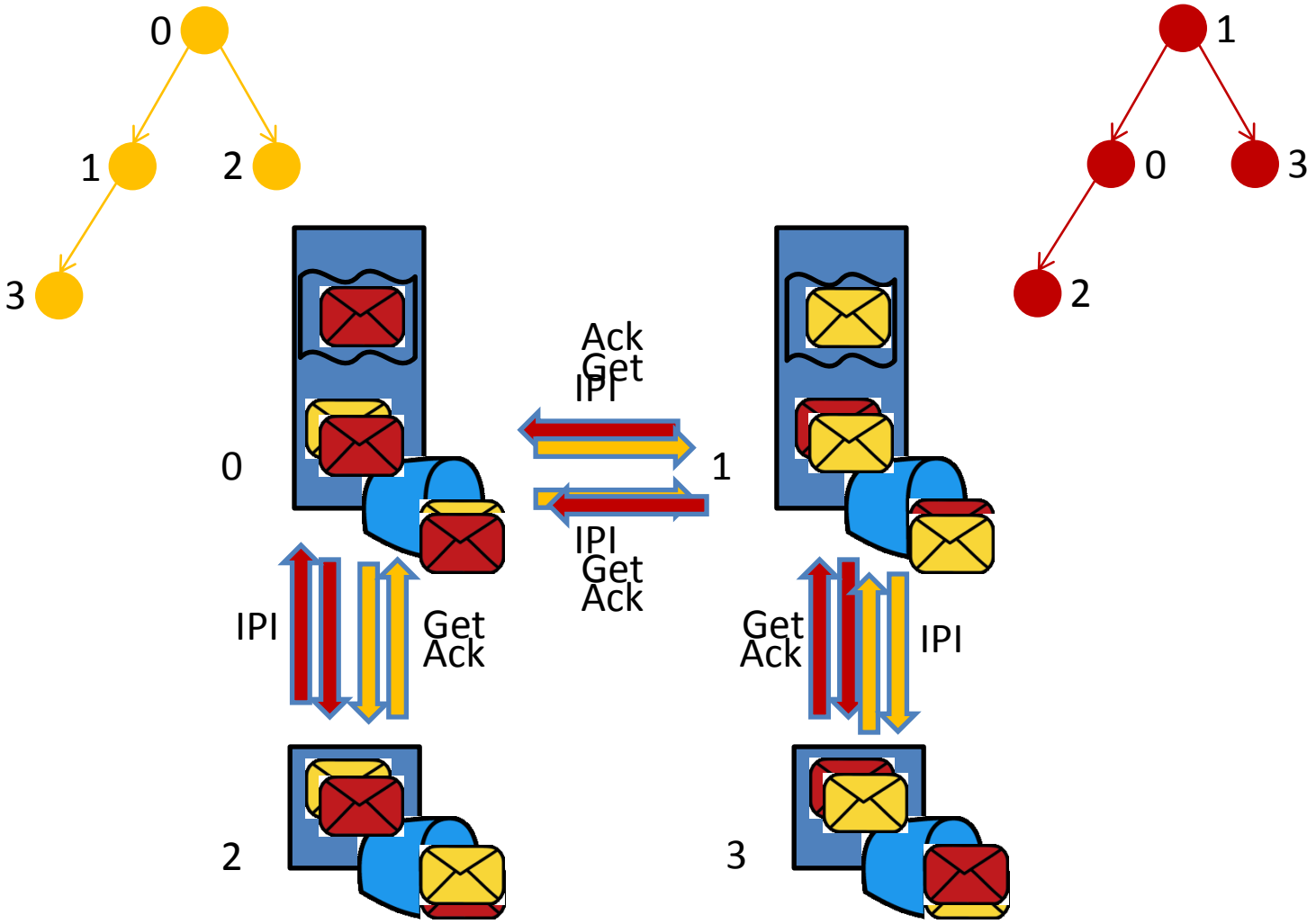
Cores check for Acks by polling.

Issue with Concurrent Broadcasts



Trivial solution: using locks -> reduces concurrency ☹

Solution Preserving Parallelism

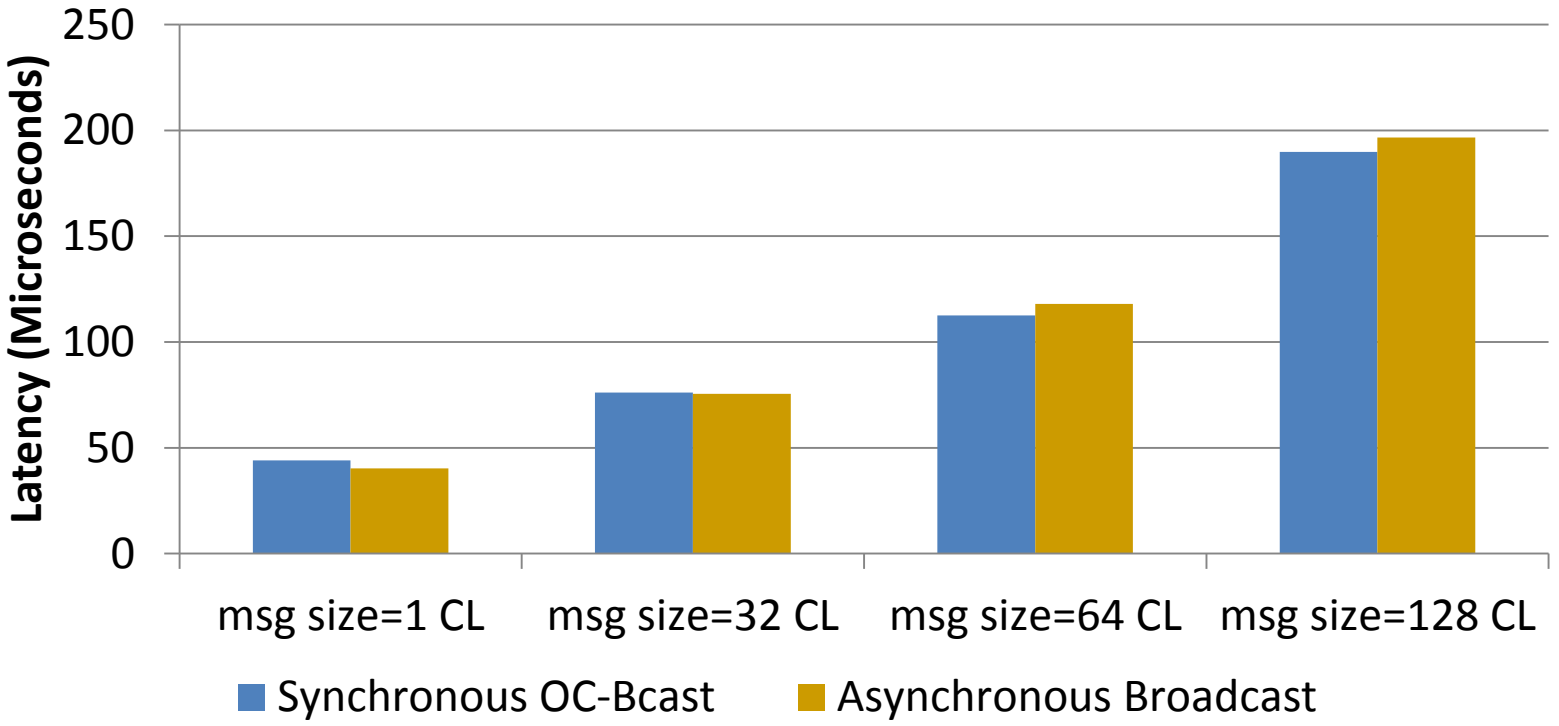


Experiments

- Setup: 533 MHz tile frequency, 800 MHz mesh and DRAM frequency. sccKit 1.4.1.3 running a custom version of sccLinux.
- The kernel of every core runs a special kernel module to convert interrupts to UNIX signals.
- The algorithms are implemented using our user-space interrupt library.
- Metrics: latency, throughput
- Variables: k, # of sources, msg size

Latency Evaluation

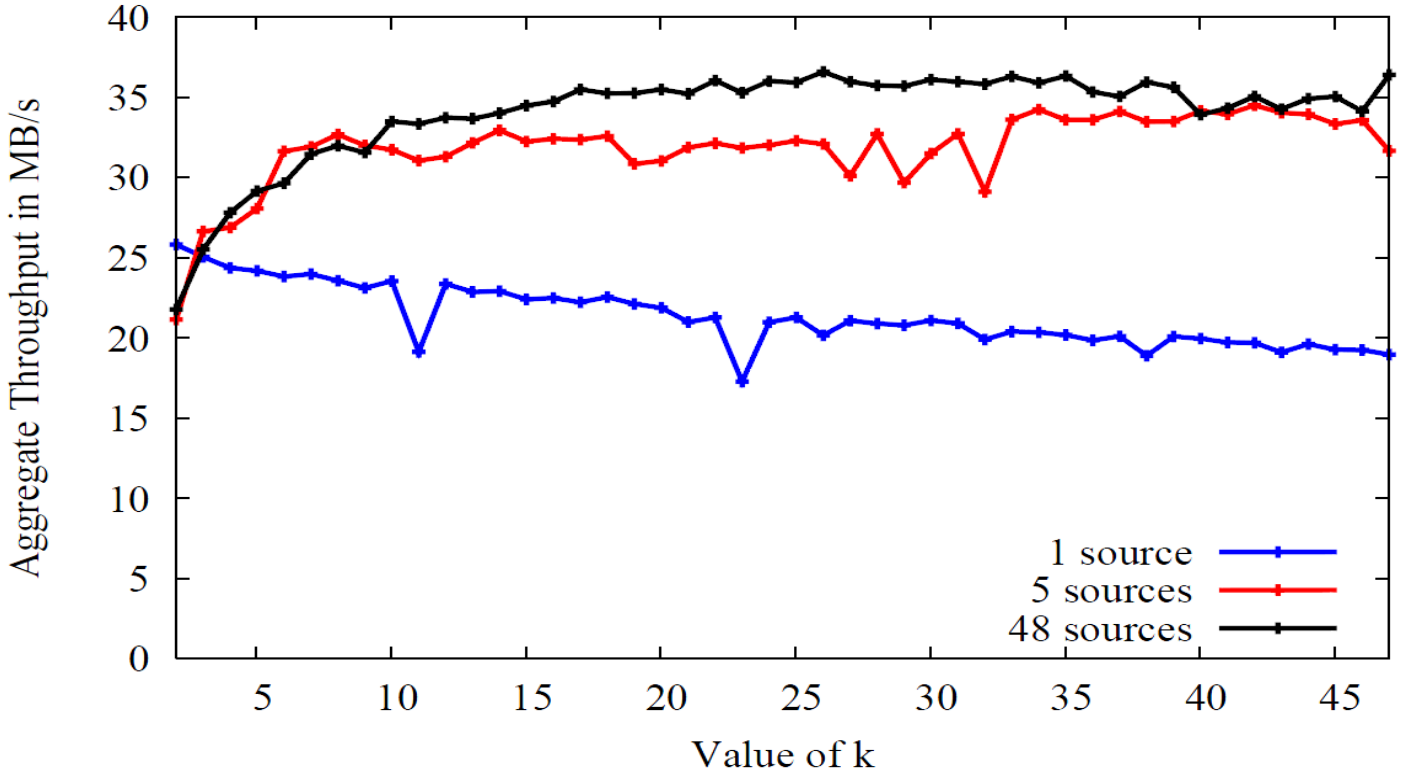
→ k=47, 1 source, different message sizes (CL: Cache Lines)



Comparable latencies

Throughput Evaluation

→ Different k and # of sources, msg size = 128 CL, each source repeats broadcasting messages (pipeline)



68% of maximum MPB-to-memory throughput (55 MB/s)

Summary

- We provide an efficient asynchronous broadcast using IPIs.
- We also provide a user space IPI library to implement asynchronous algorithms on the SCC.

Takeaway Message

Asynchronous group communication can be implemented efficiently using parallel IPIs on many-cores.