TECHNISCHE UNIVERSITÄT DRESDEN

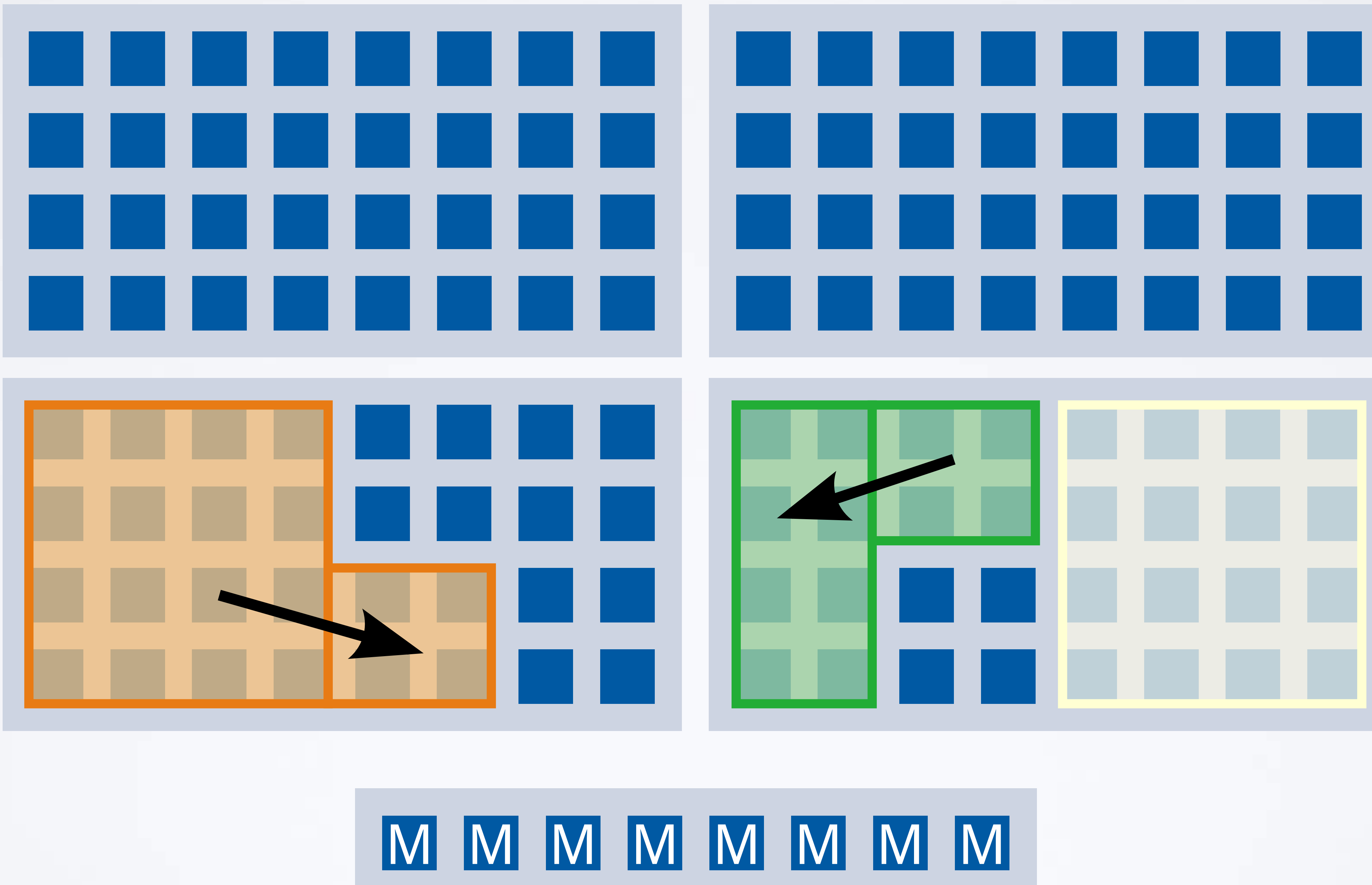
The Hebrew University of Jerusalem

FFMK: A FAST AND FAULT-TOLERANT MICROKERNEL-BASED SYSTEM FOR EXASCALE COMPUTING

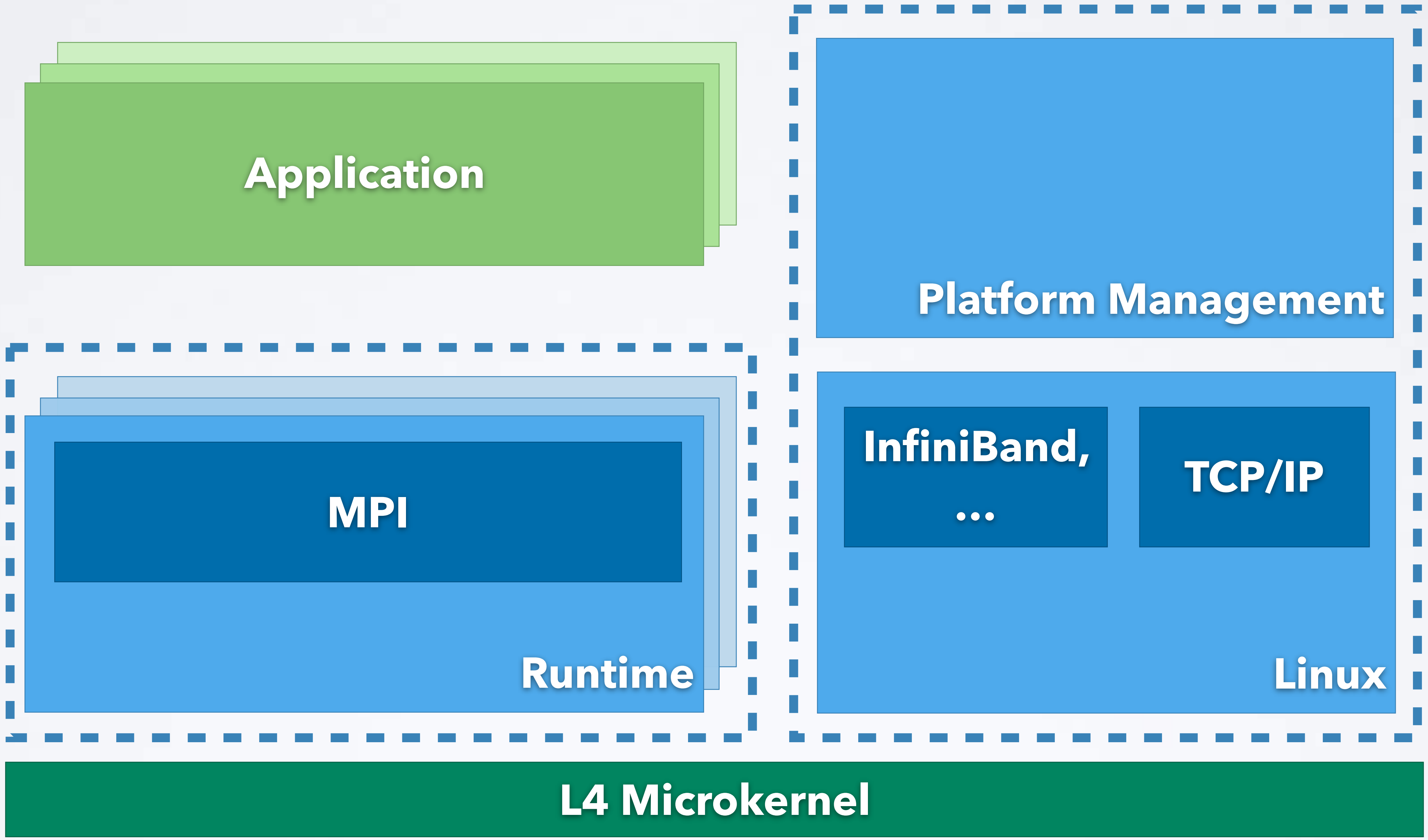
Amnon Barak	Hebrew University Jerusalem (HUJI)
Hermann Härtig	TU Dresden, Operating Systems Group (TUDOS)
Wolfgang E. Nagel	TU Dresden, Center for Information Services and HPC (ZIH)
Alexander Reinefeld	Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

CARSTEN WEINHOLD, TU DRESDEN

SYSTEM MODEL

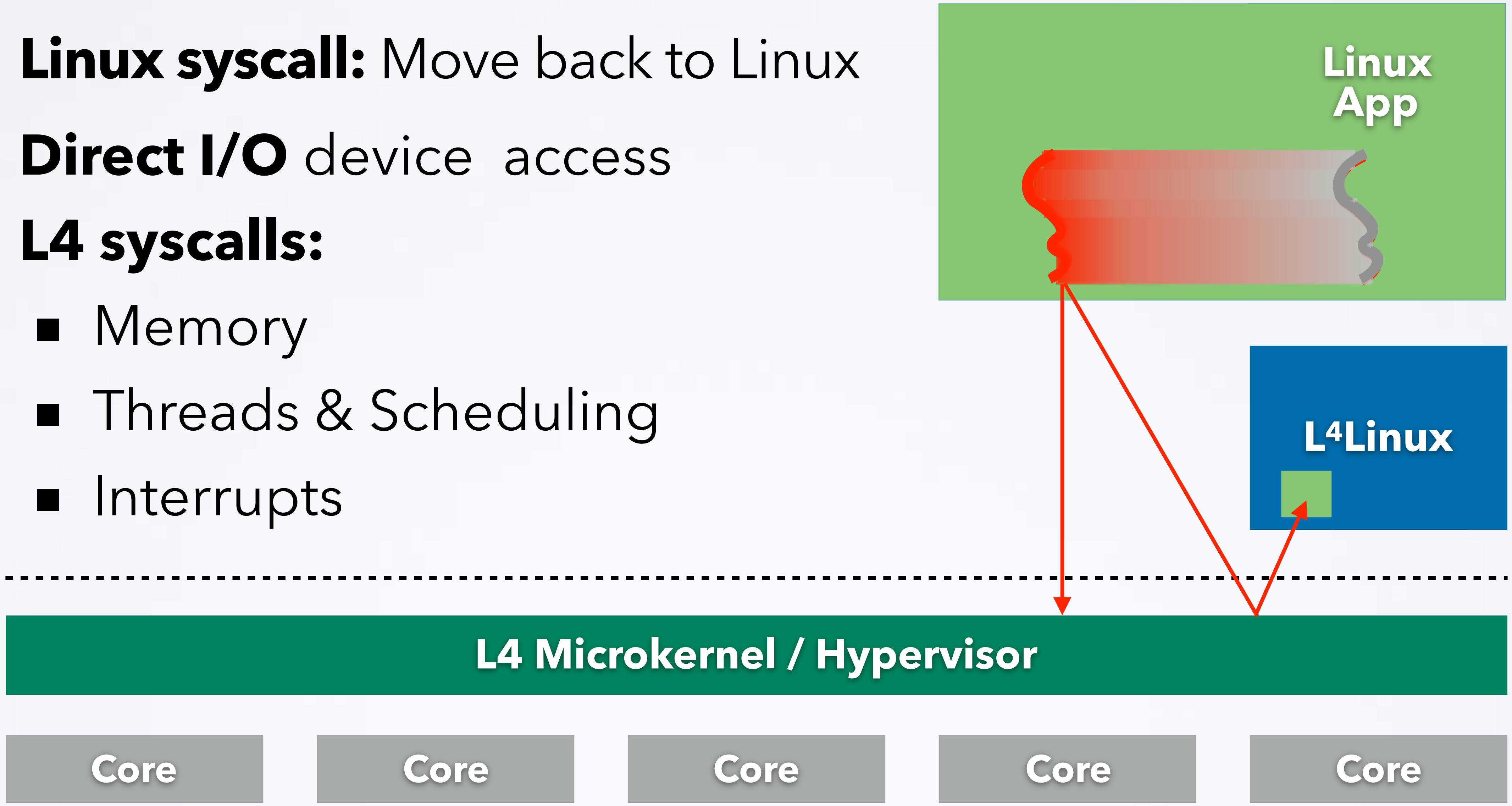


NODE ARCHITECTURE



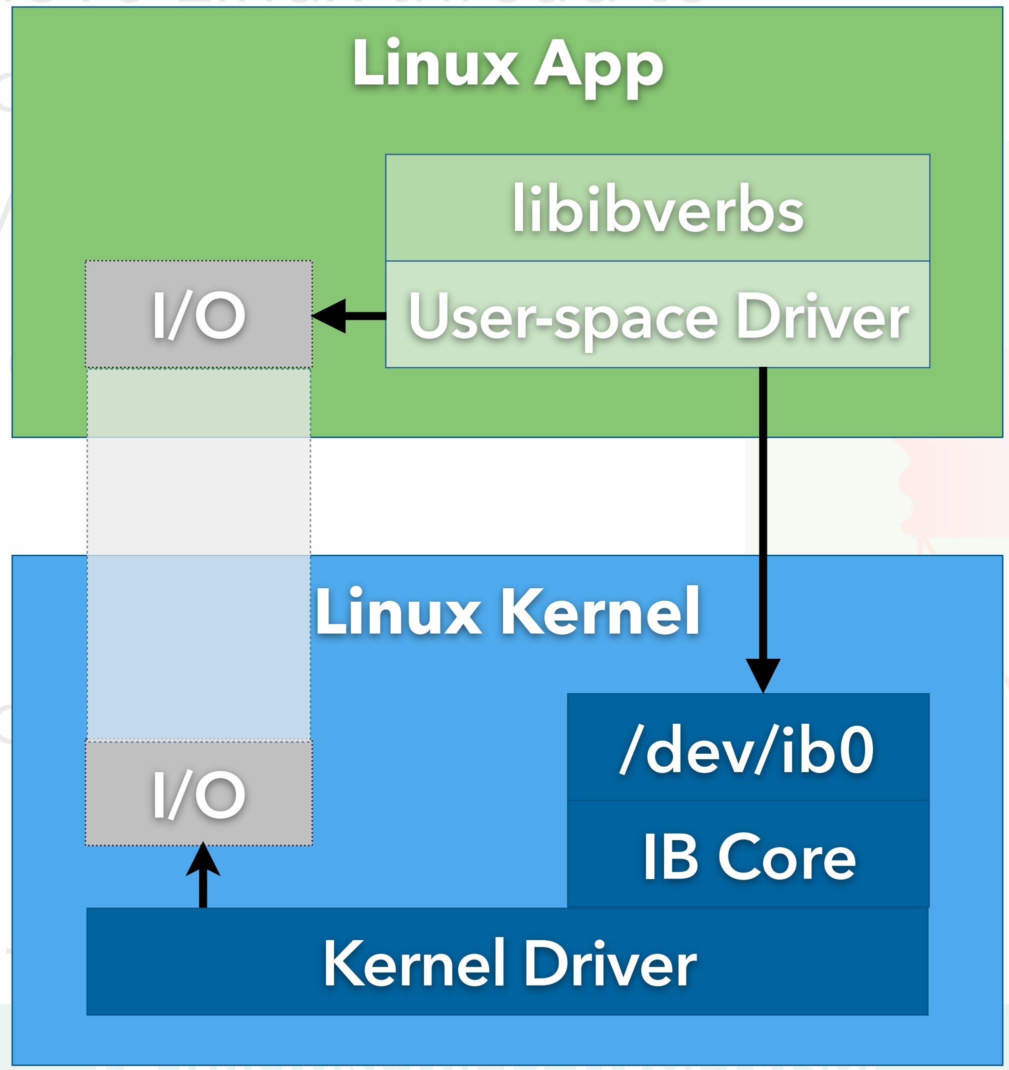
DECOUPLED THREADS

- **Decoupling:** move Linux thread to new L4 thread on its own core
- **Linux syscall:** Move back to Linux
- **Direct I/O** device access
- **L4 syscalls:**
  - Memory
  - Threads & Scheduling
  - Interrupts

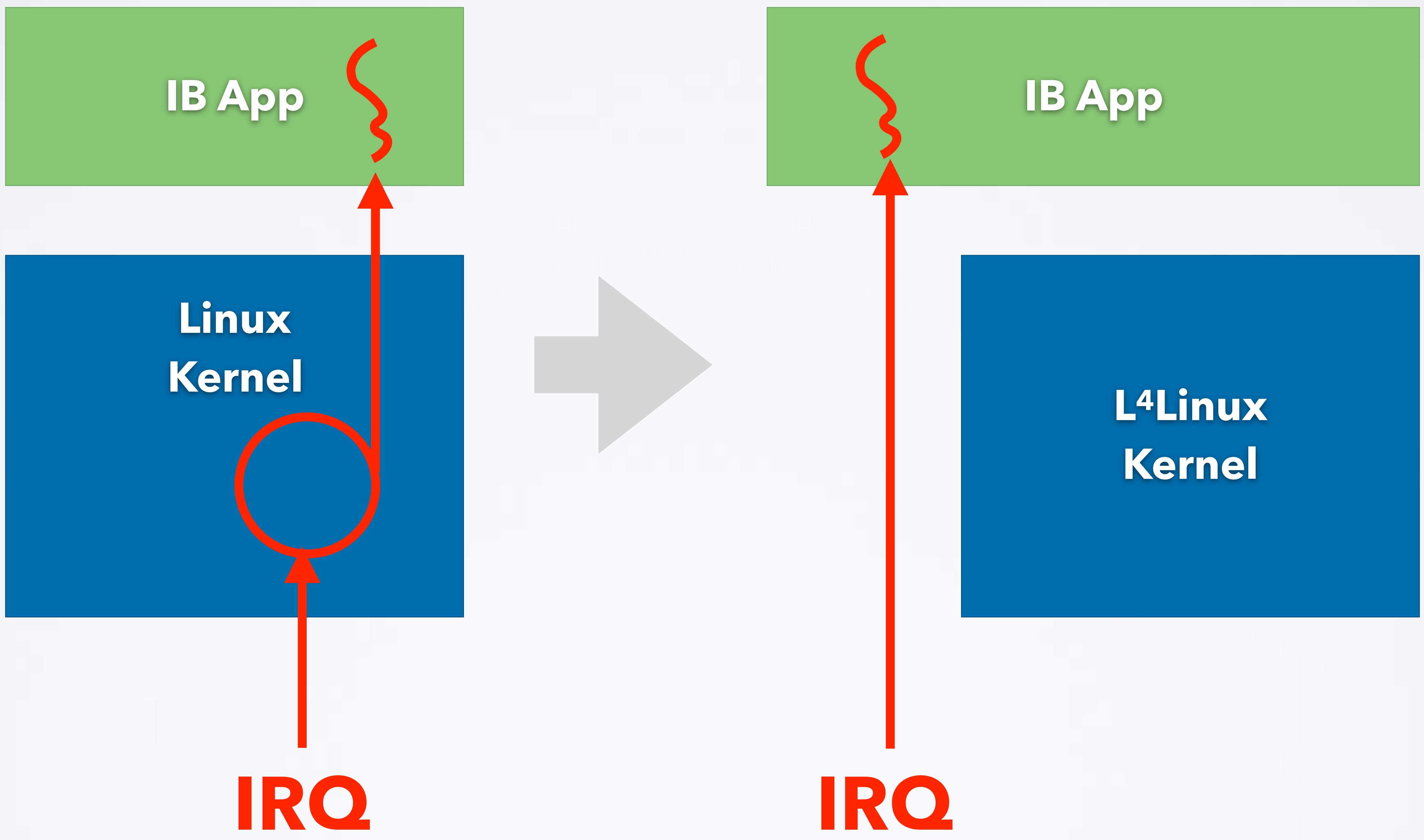


DECOUPLED THREADS

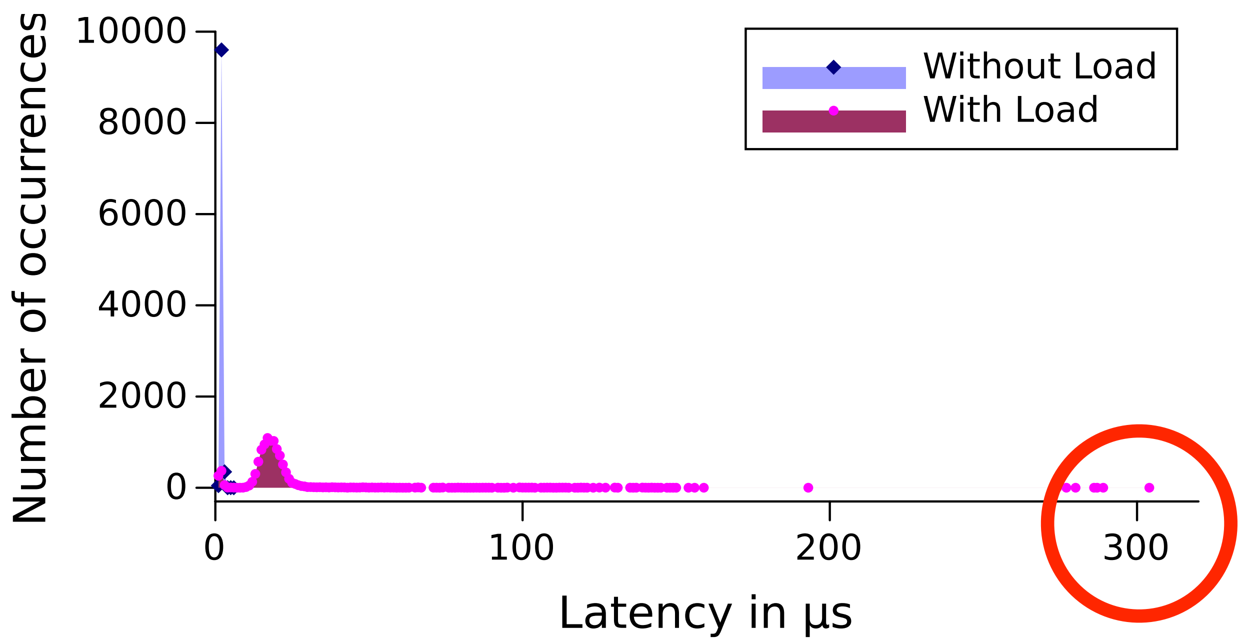
- **Decoupling:** move Linux thread to new L4 thread c
- **Linux syscall:** M
- **Direct I/O** devi
- **L4 syscalls:**
  - Memory
  - Threads & Sc
  - Interrupts



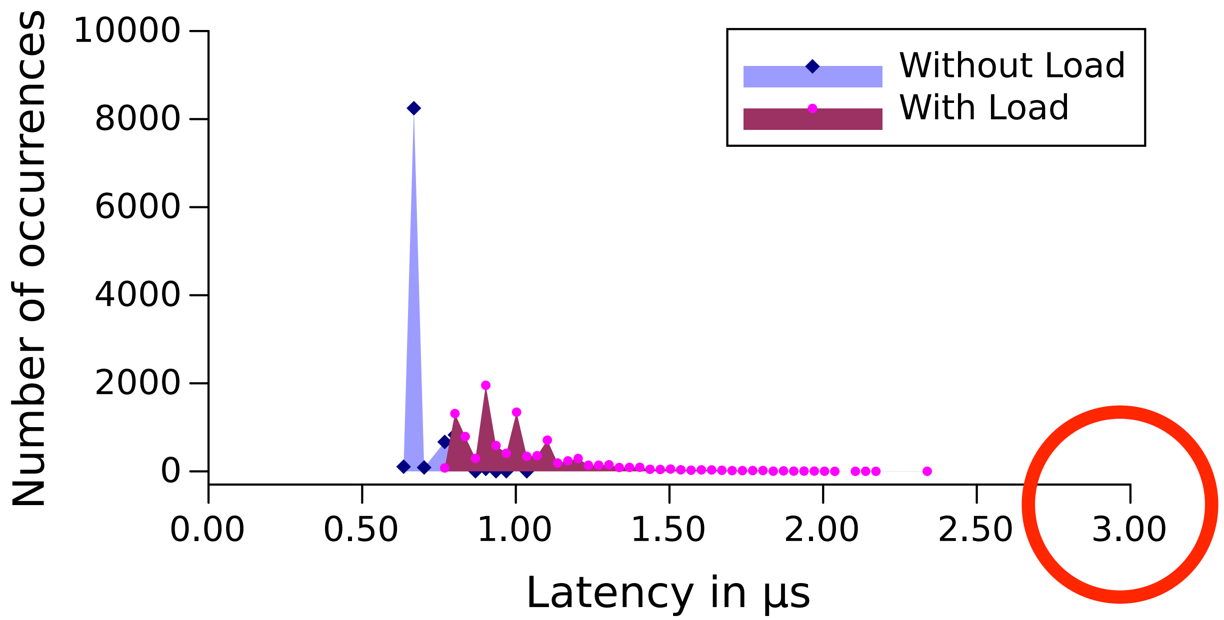
IRQ FASTPATH



WAKE FROM IRQ



Linux



L4

Work in progress: User-space handling of InfiniBand HCA interrupts

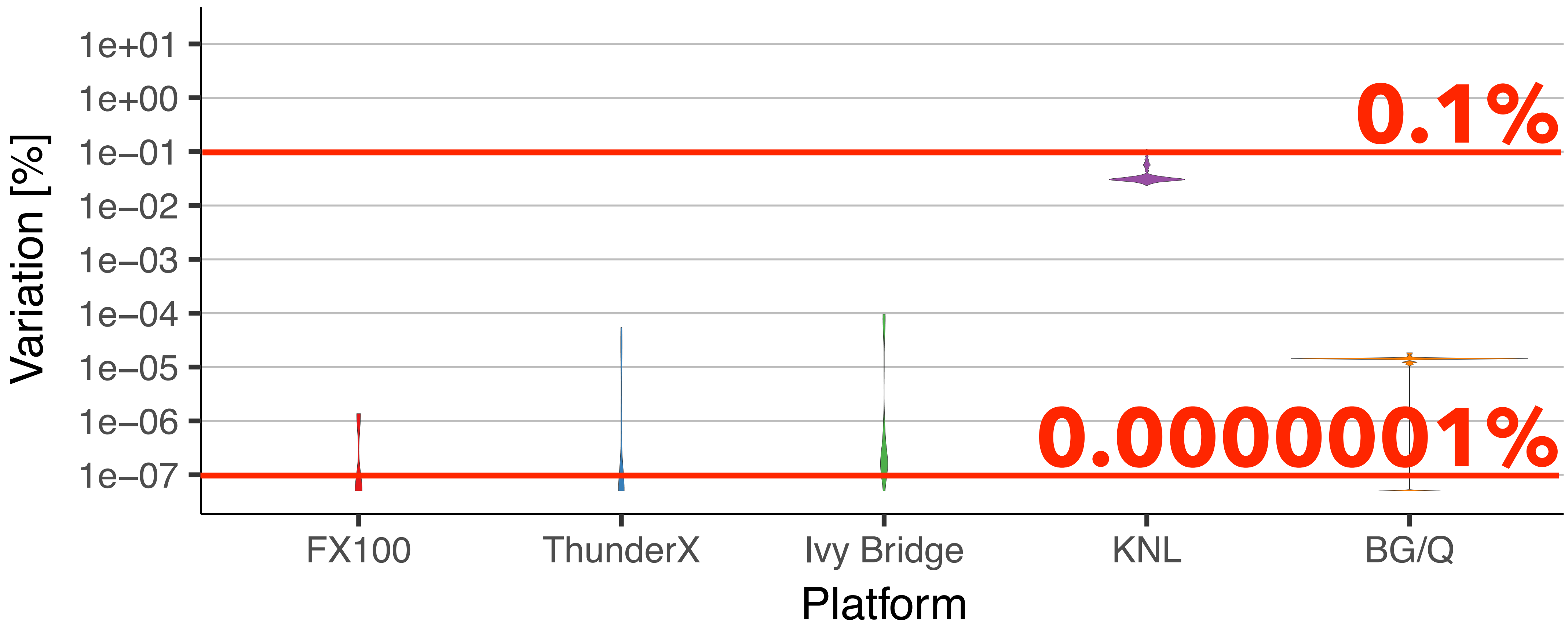
Adam Lackorzynski, Carsten Weinhold, Hermann Härtig, „Predictable Low-Latency Interrupt Response with General-Purpose Systems", OSPERT 2017, Dubrovnik, Kroatia, June 2017

COLLABORATION

- PhD student: internship at RIKEN, Japan
- Comparative study:
  - Hardware performance variation
  - 5 different CPU architectures
  - Light-weight kernel (McKernel)

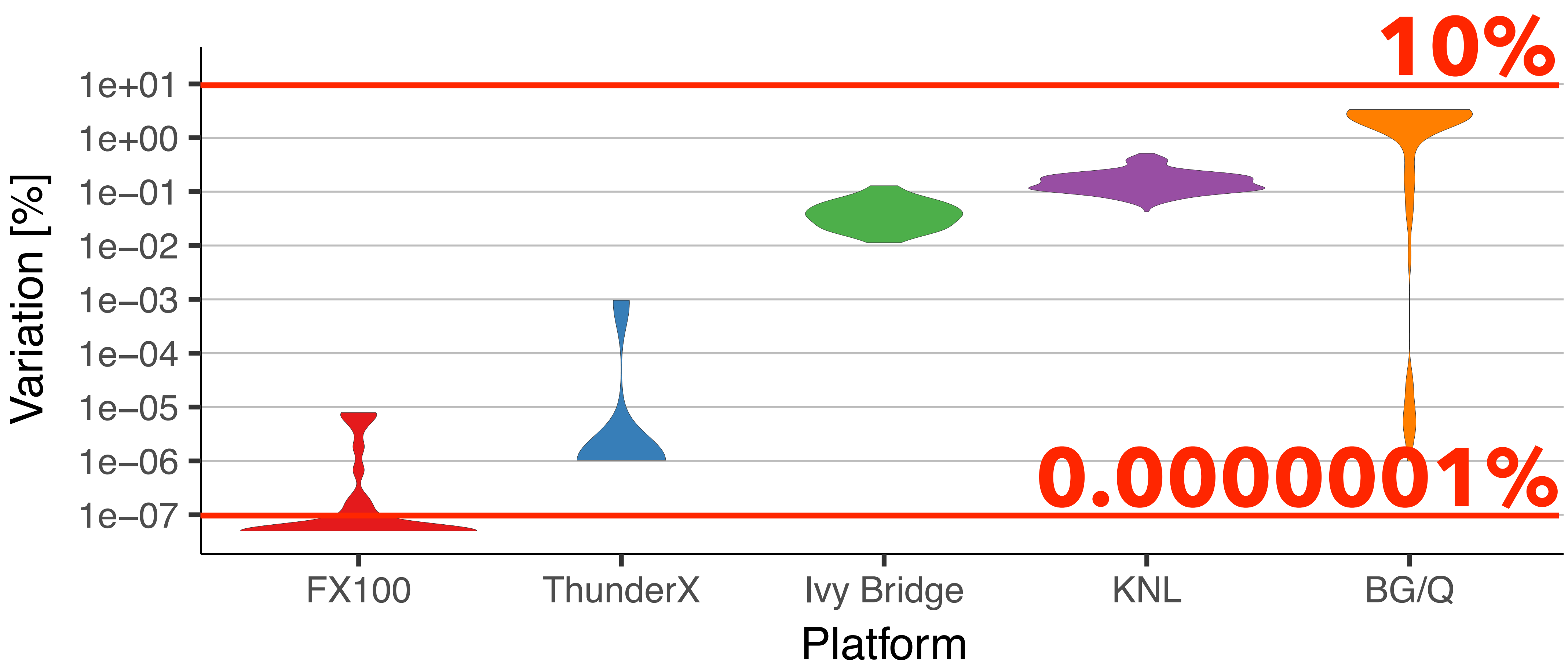
Hannes Weisbach, Brian Kocoloski, Hermann Härtig, Yutaka Ishikawa, Balazs Gerofi, „Hardware Performance Variation: A Comparative Study using Lightweight Kernels", ISC'18, Frankfurt, Germany, June 2018

FWQ BENCHMARK



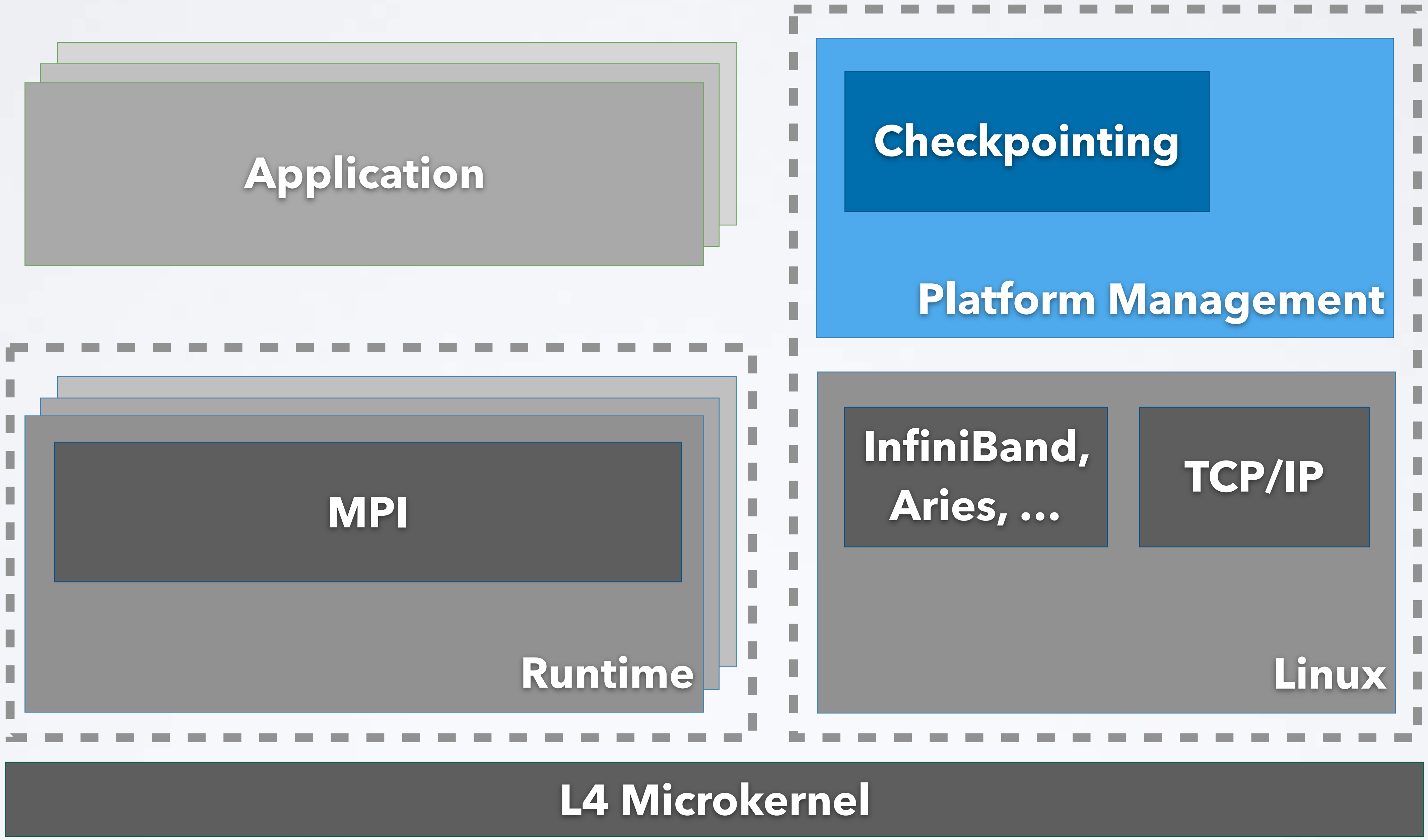
Hannes Weisbach, Brian Kocoloski, Hermann Härtig, Yutaka Ishikawa, Balazs Gerofi, „Hardware Performance Variation: A Comparative Study using Lightweight Kernels", ISC'18, Frankfurt, Germany, June 2018

DGEMM BENCHMARK

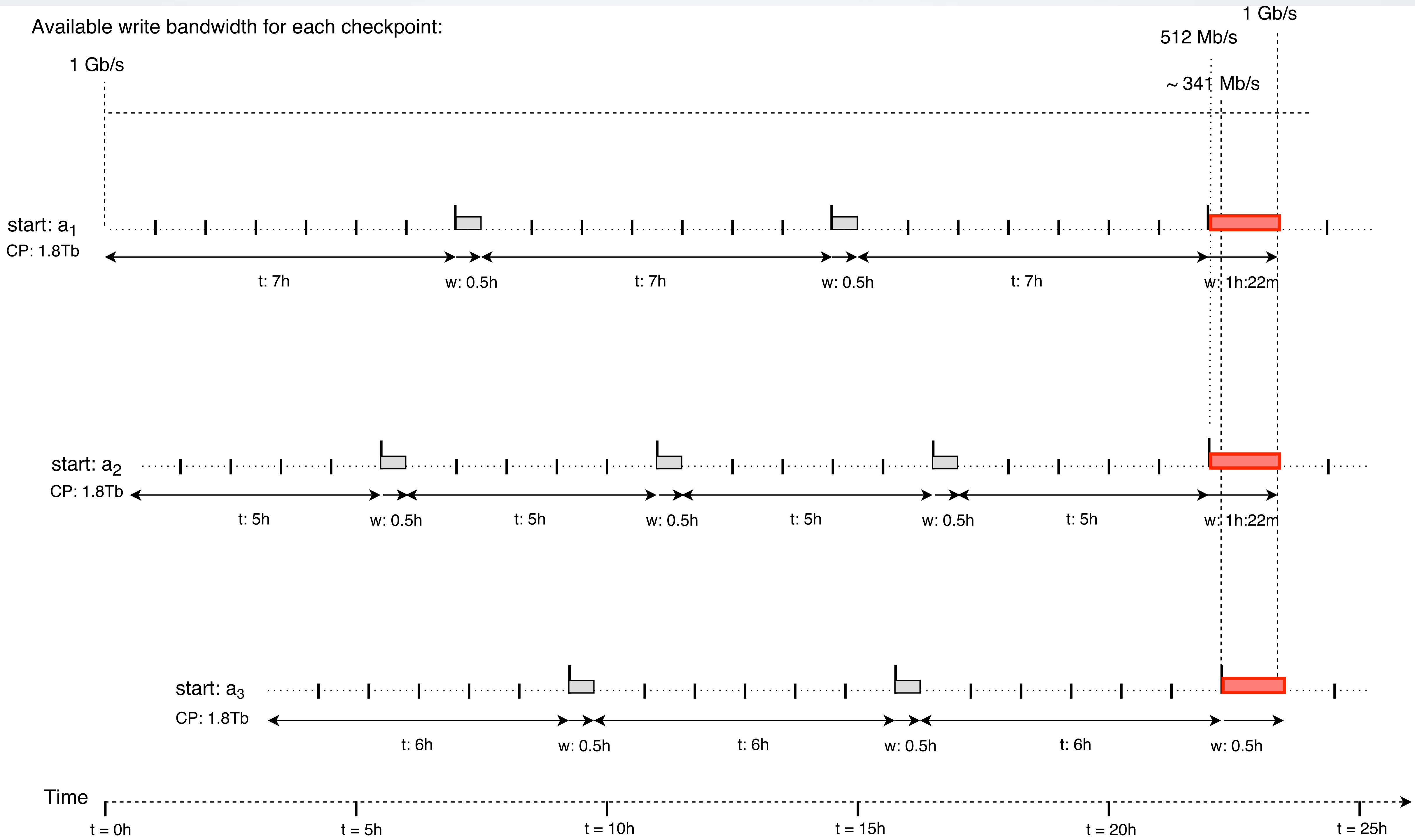


Hannes Weisbach, Brian Kocoloski, Hermann Härtig, Yutaka Ishikawa, Balazs Gerofi, „Hardware Performance Variation: A Comparative Study using Lightweight Kernels", ISC'18, Frankfurt, Germany, June 2018

NODE ARCHITECTURE

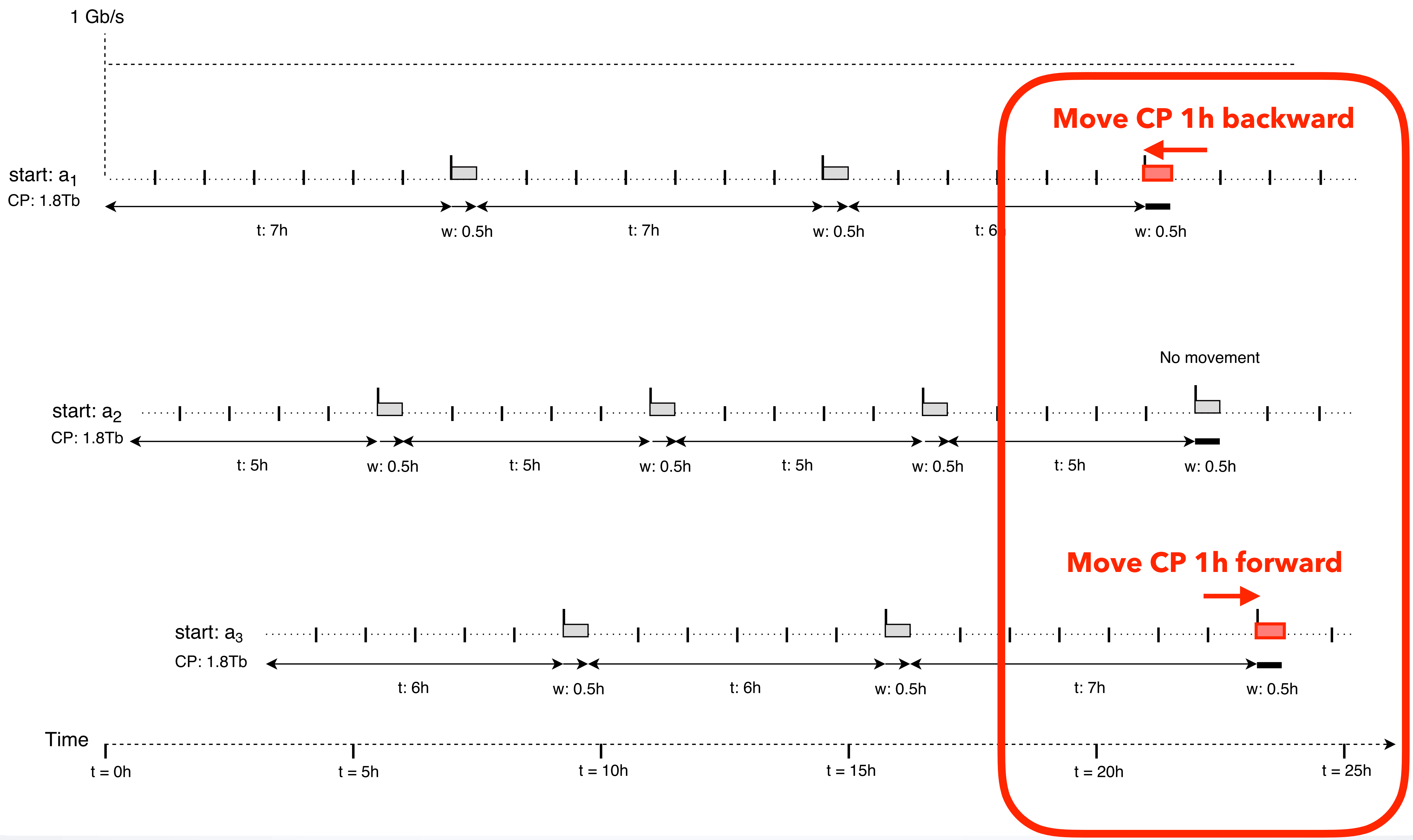


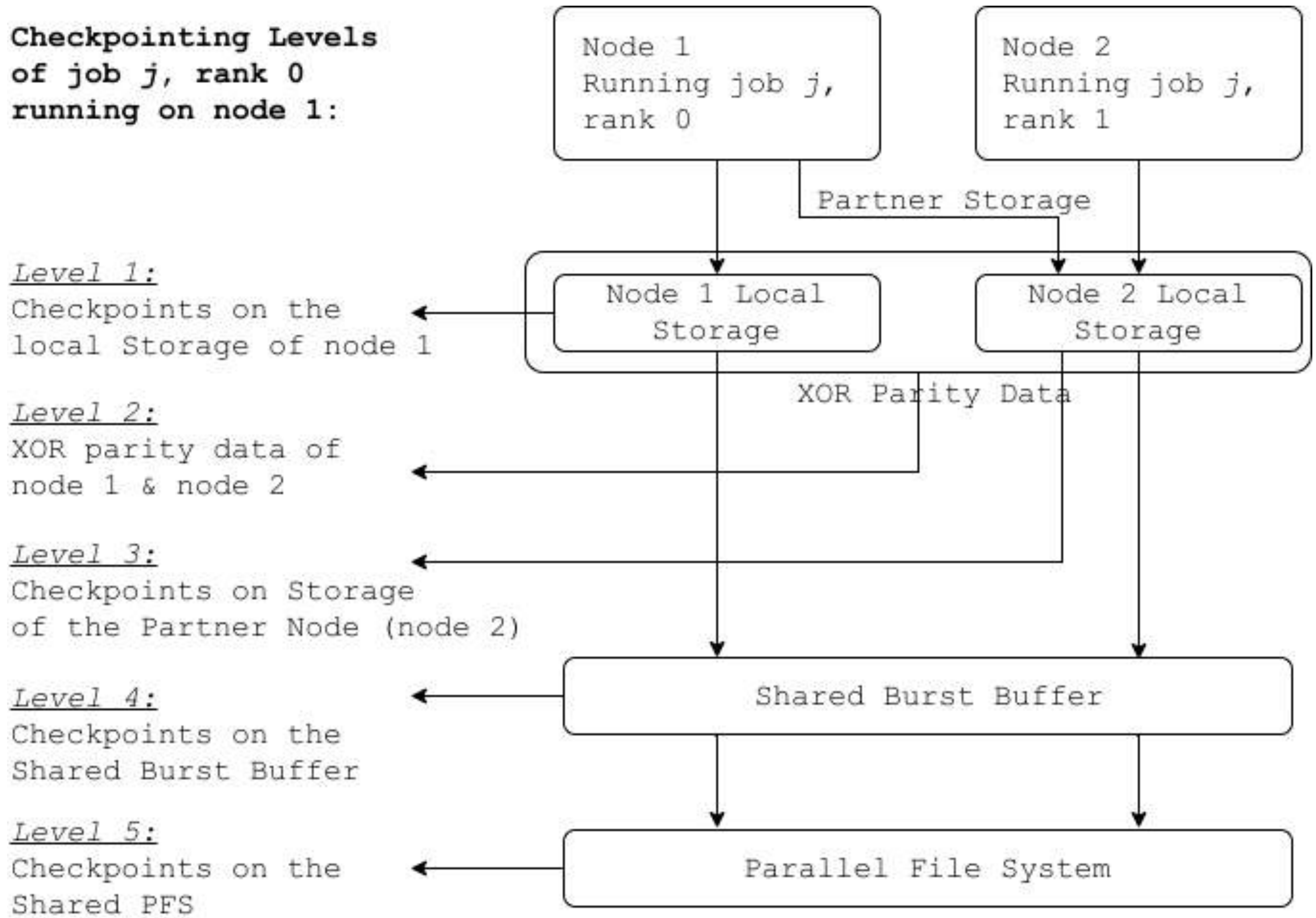
COORDINATED C/R



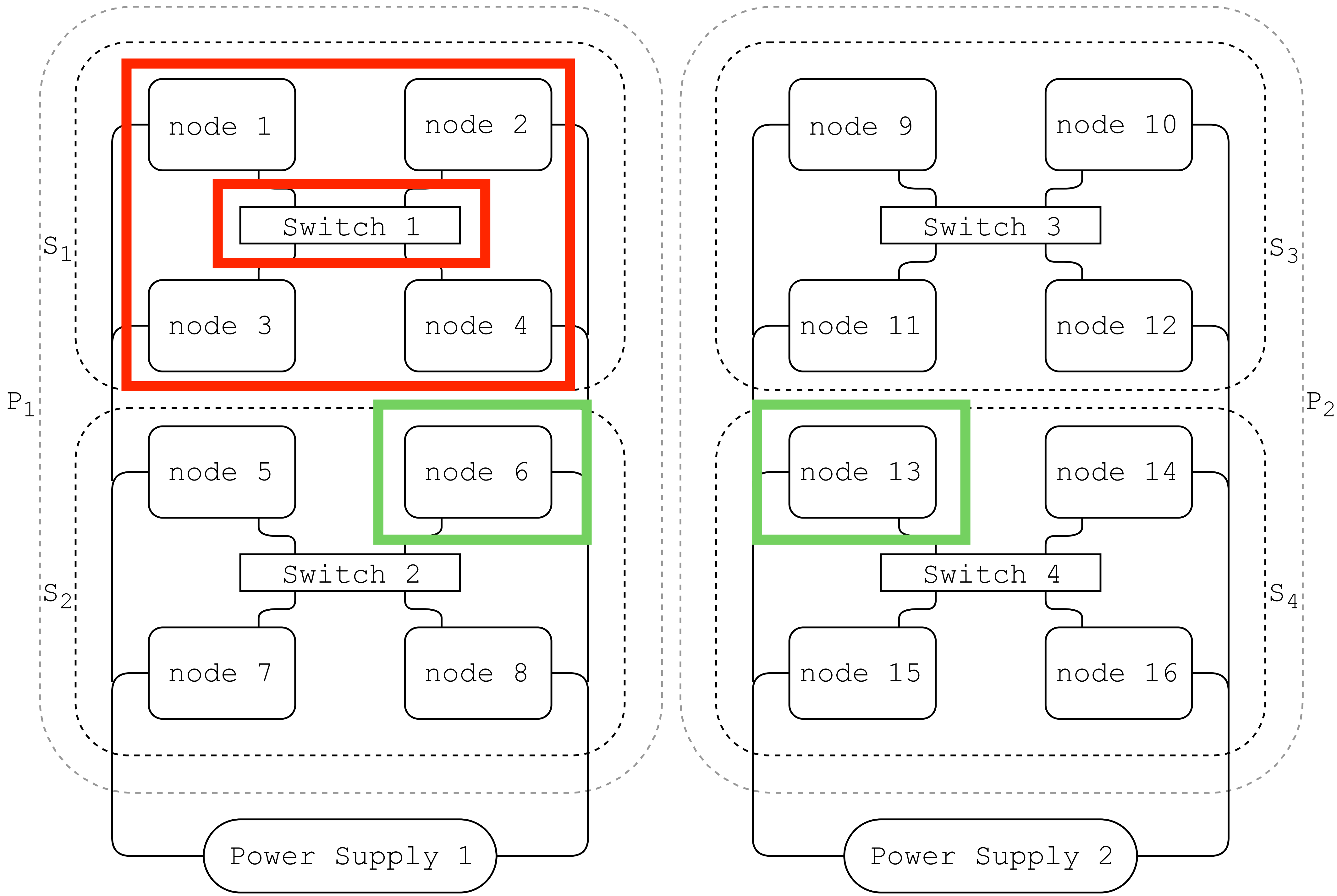
COORDINATED C/R

Available write bandwidth for each checkpoint:

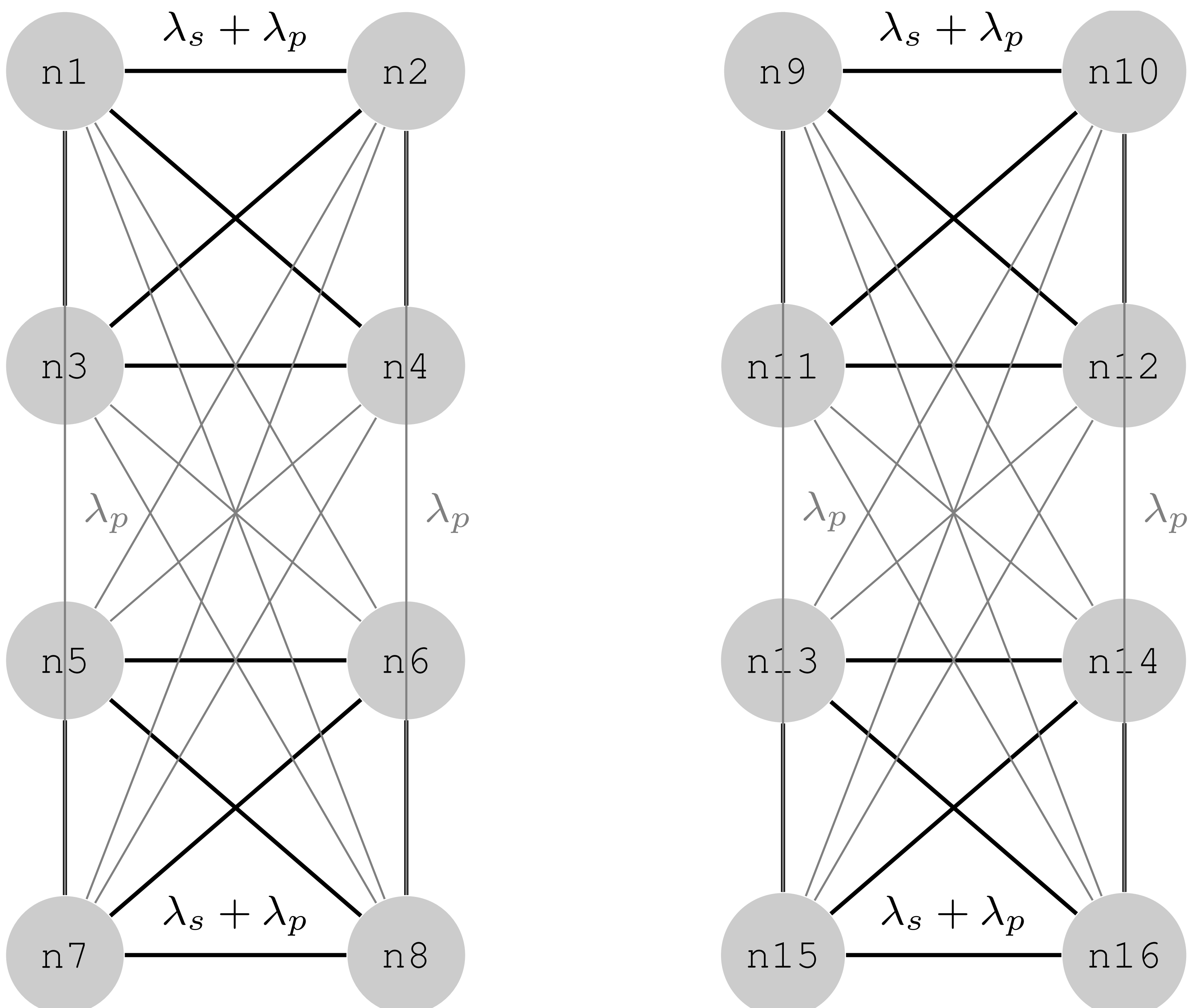



Checkpointing Levels
of job j, rank 0
running on node 1:
Node 1
Running job j,
rank 0
Node 2
Running job j,
rank 1
Partner Storage
Level 1:
Checkpoints on the
local Storage of node 1
Node 1 Local
Storage
Node 2 Local
Storage
XOR Parity Data
Level 2:
XOR parity data of
node 1 & node 2
Level 3:
Checkpoints on Storage
of the Partner Node (node 2)
Level 4:
Checkpoints on the
Shared Burst Buffer
Shared Burst Buffer
Level 5:
Checkpoints on the
Shared PFS
Parallel File System

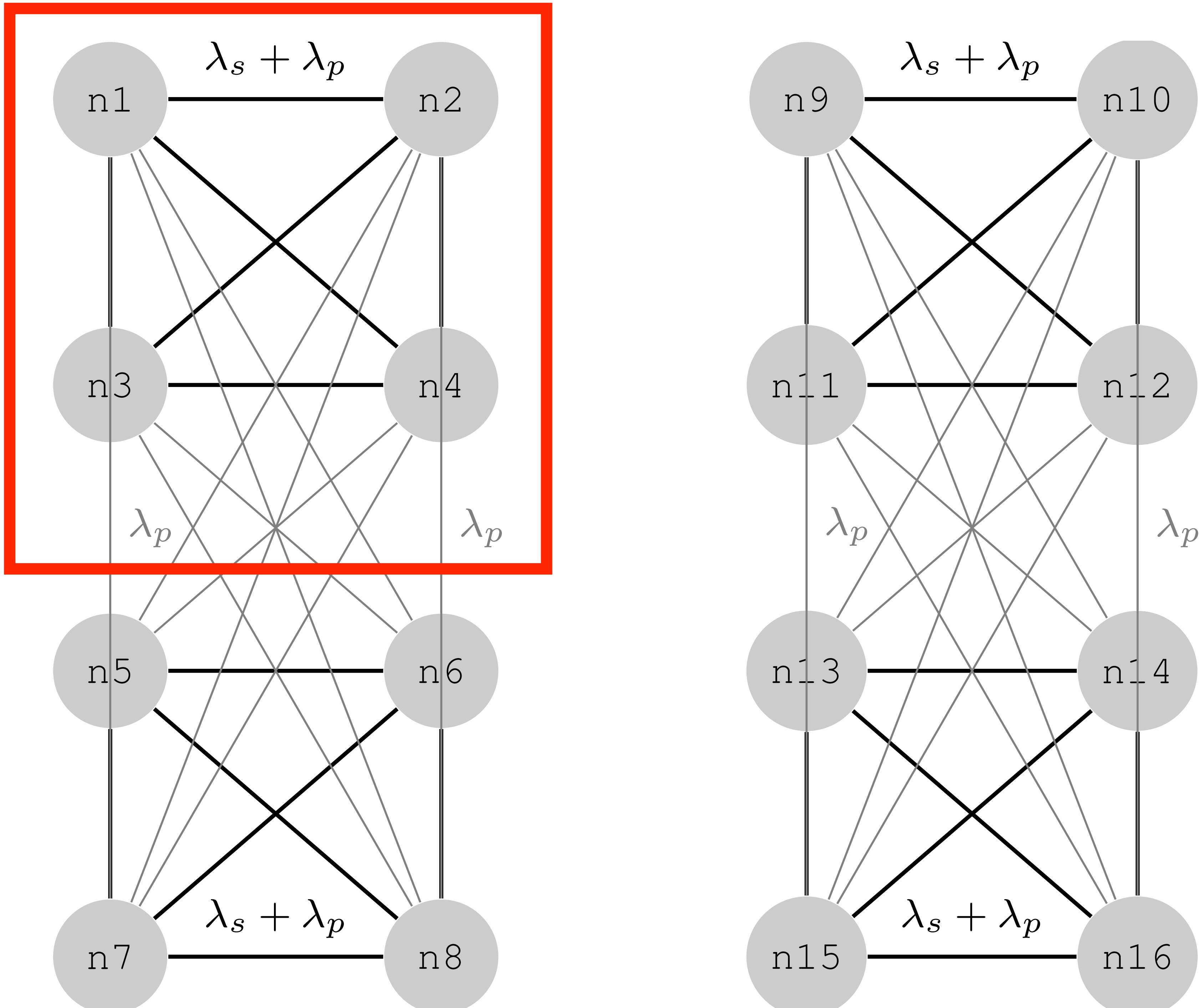
CORRELATED FAILURE



CORRELATED FAILURE



CORRELATED FAILURE



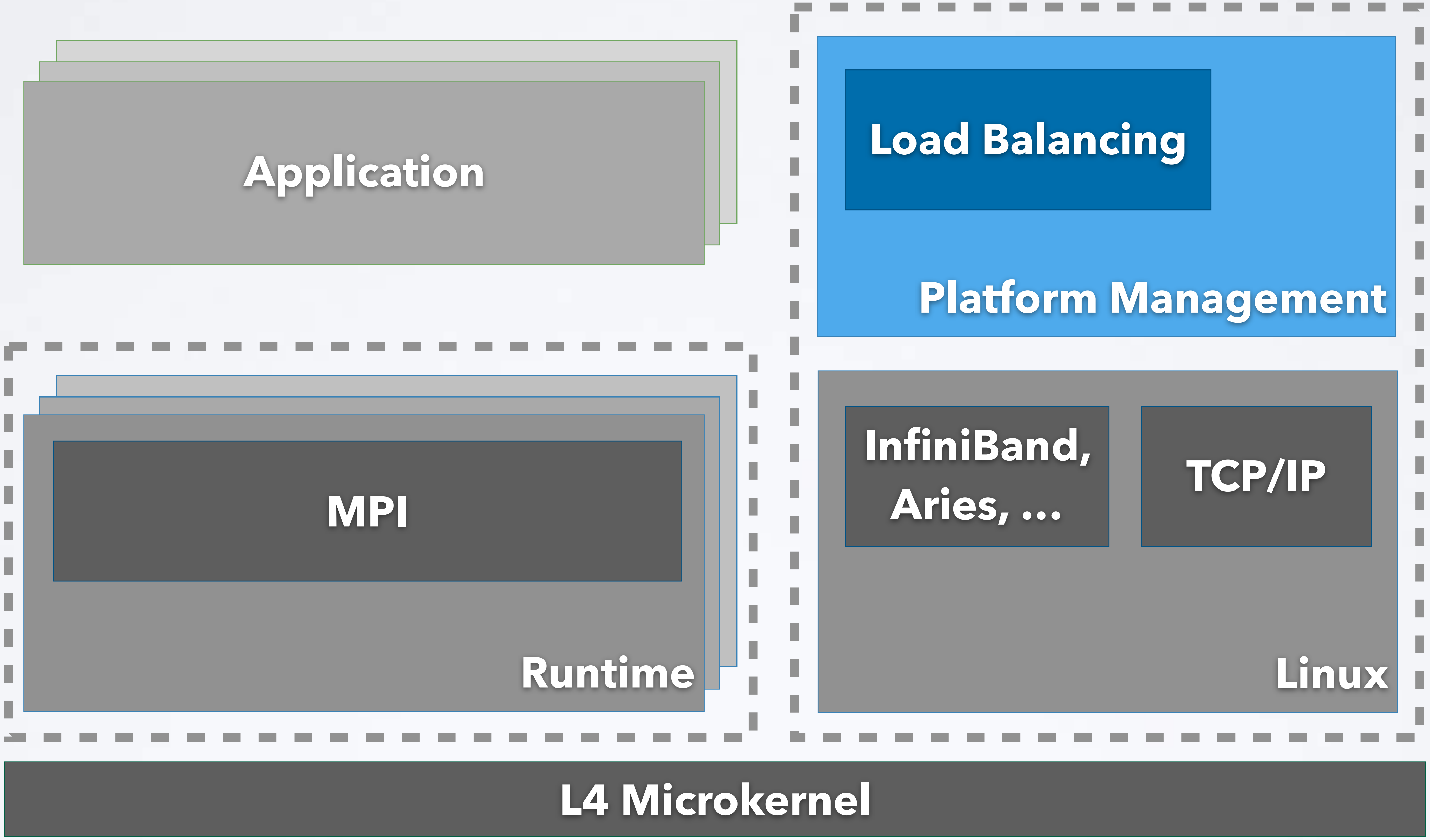
Graph problem:

- Find disjoint independent sets
- Find dominating subgraphs („least correlated nodes“)

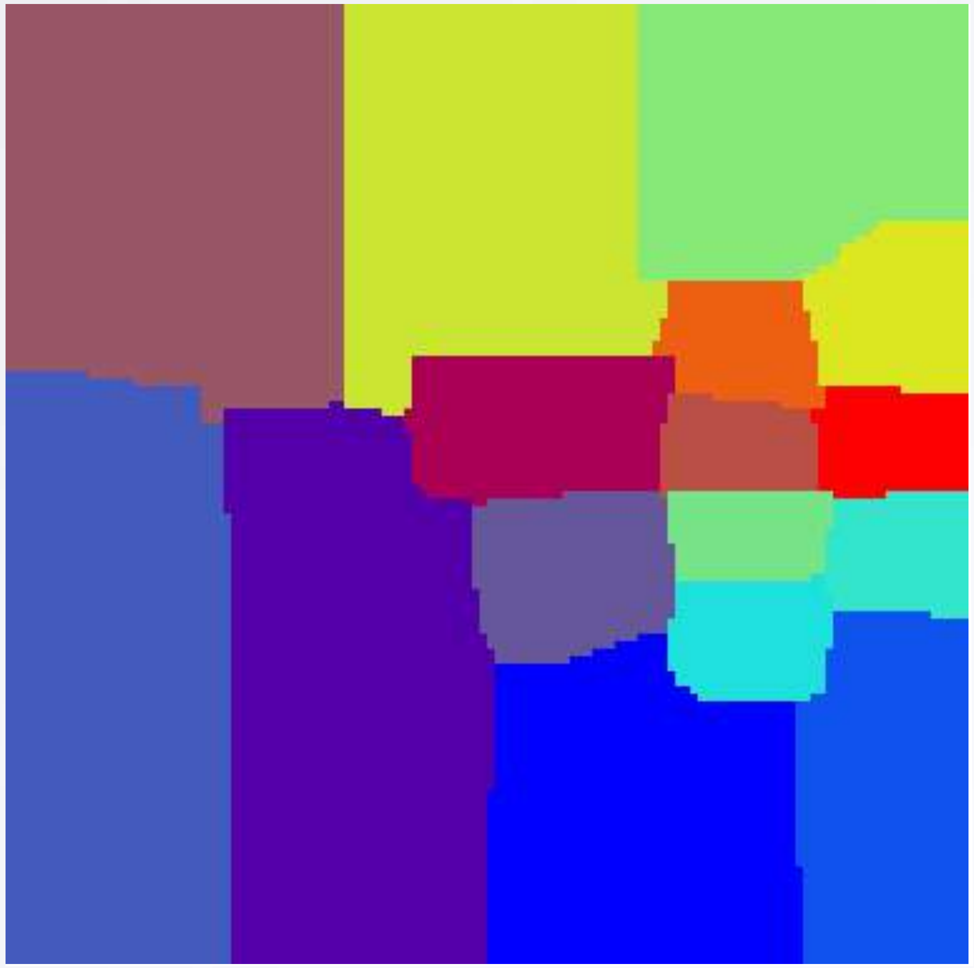
Optimization problem:

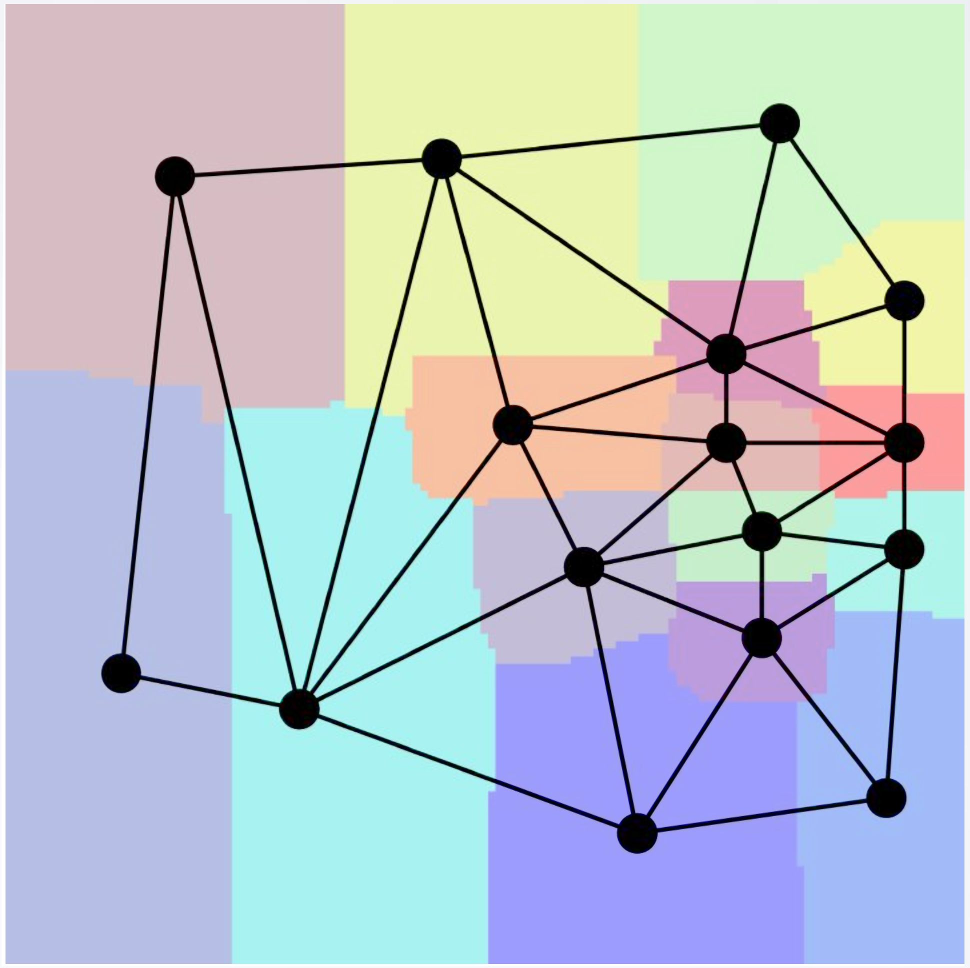
- least correlated nodes for checkpoint distribution
- Consider: job run time, C/R cost, MTTI
- Minimize run time

NODE ARCHITECTURE



DIFFUSION



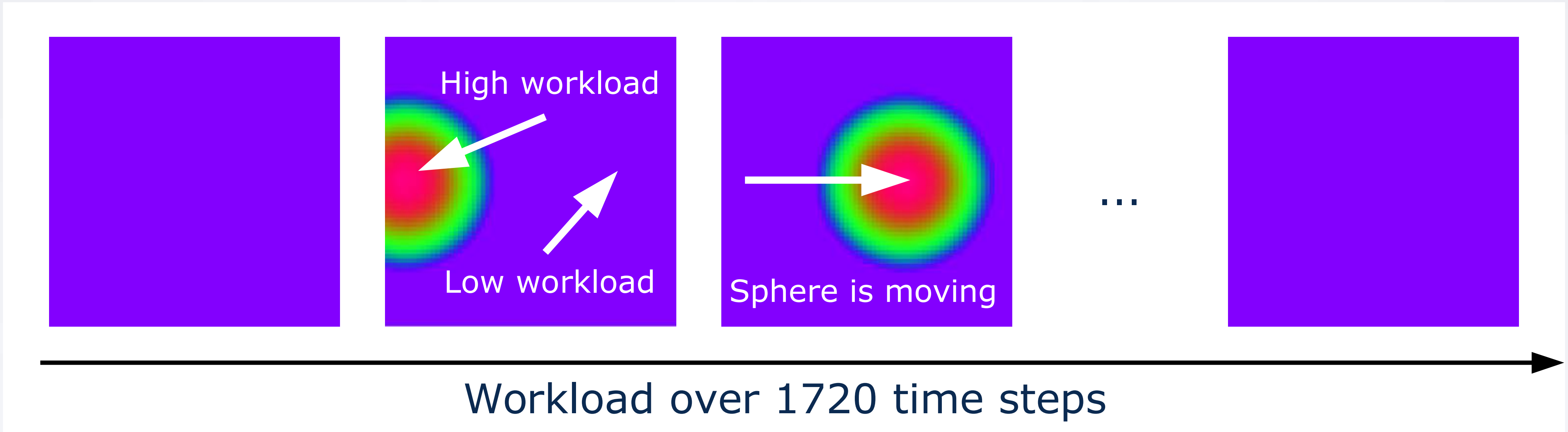


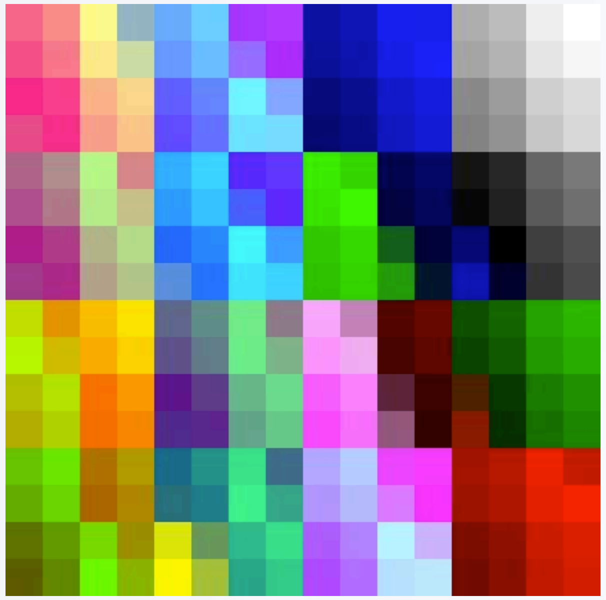
Diffusion graph topology from application topology

Diffusion coefficient weighted by interface length:

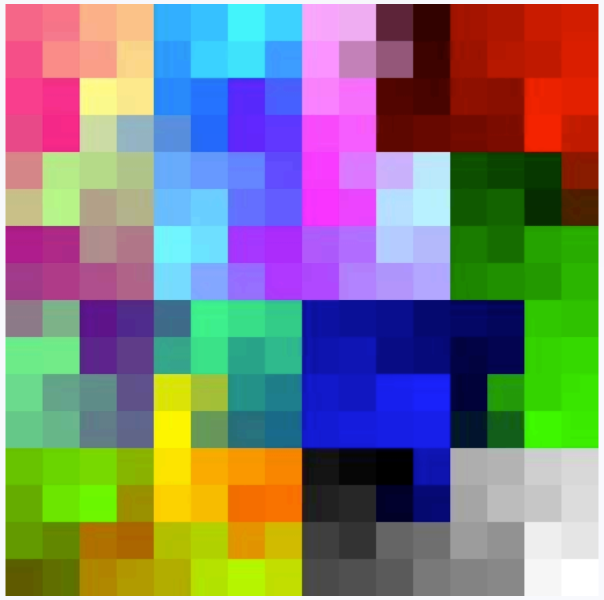
- Tasks migrated between neighbor partitions
- Better partition shape

DIFFUSION EXAMPLE

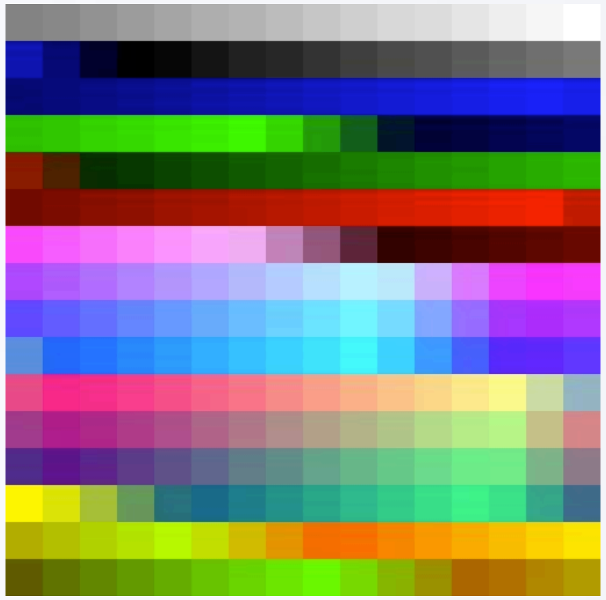




Zoltan

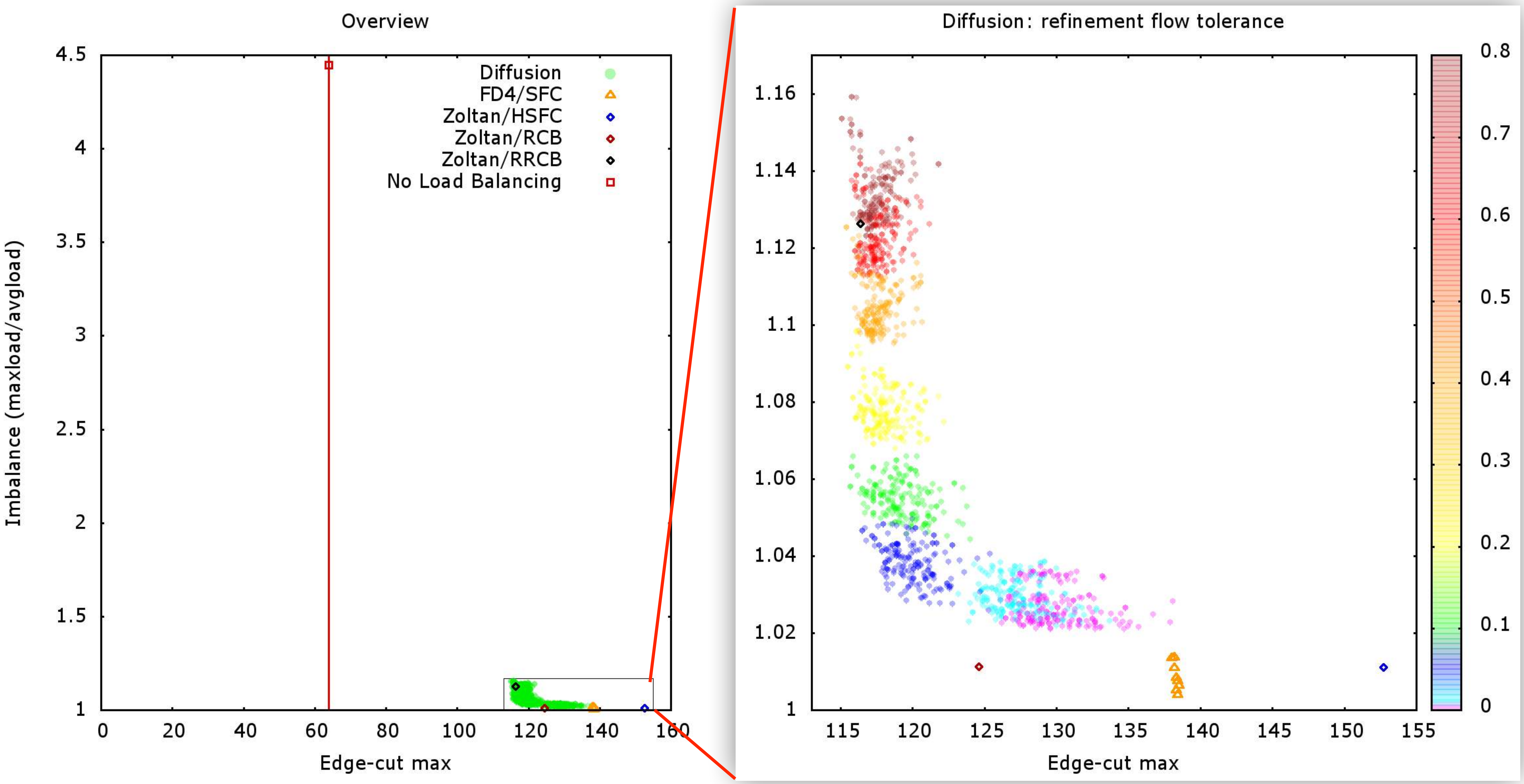


Space-filling Curves



Diffusion

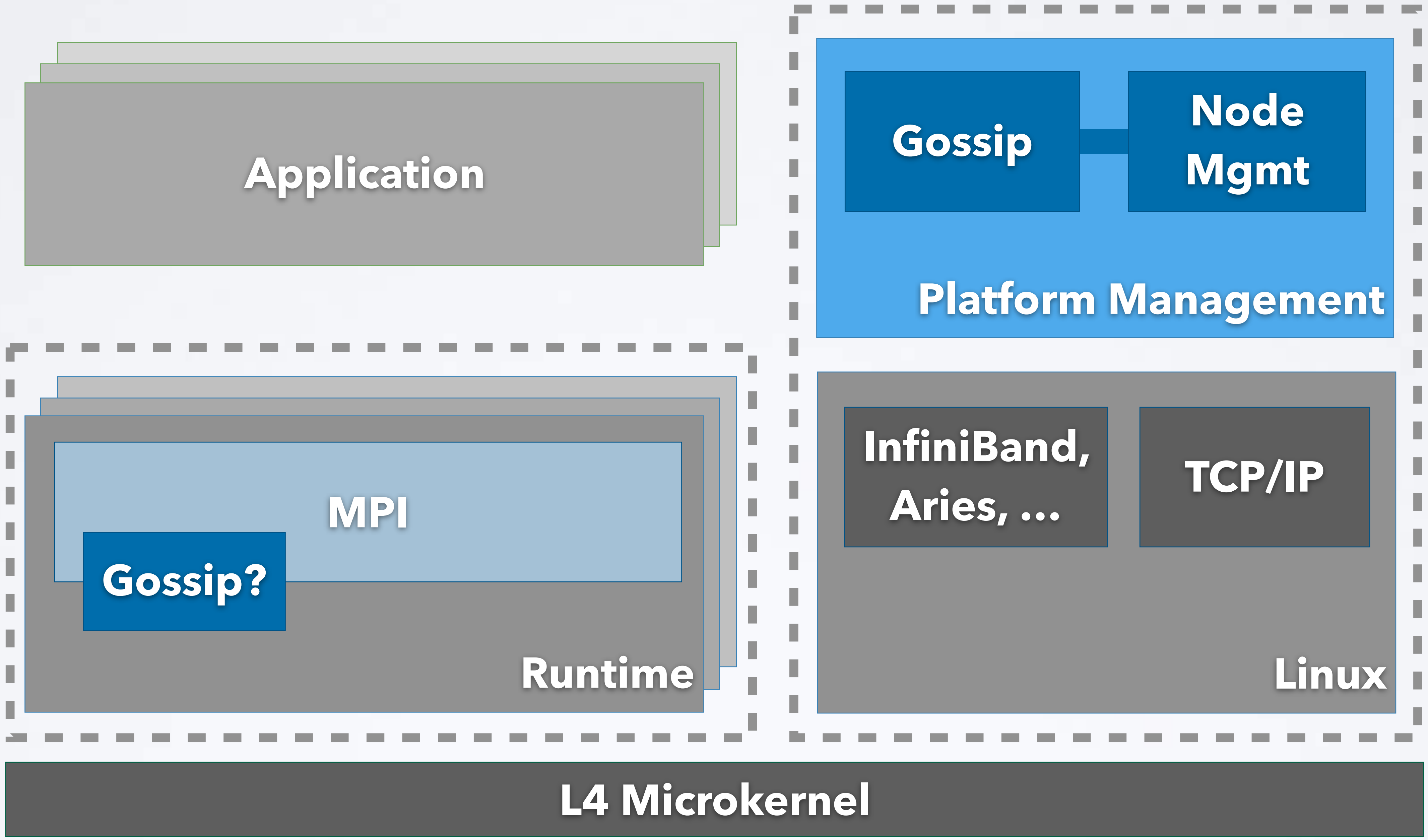
DIFFUSION RESULTS



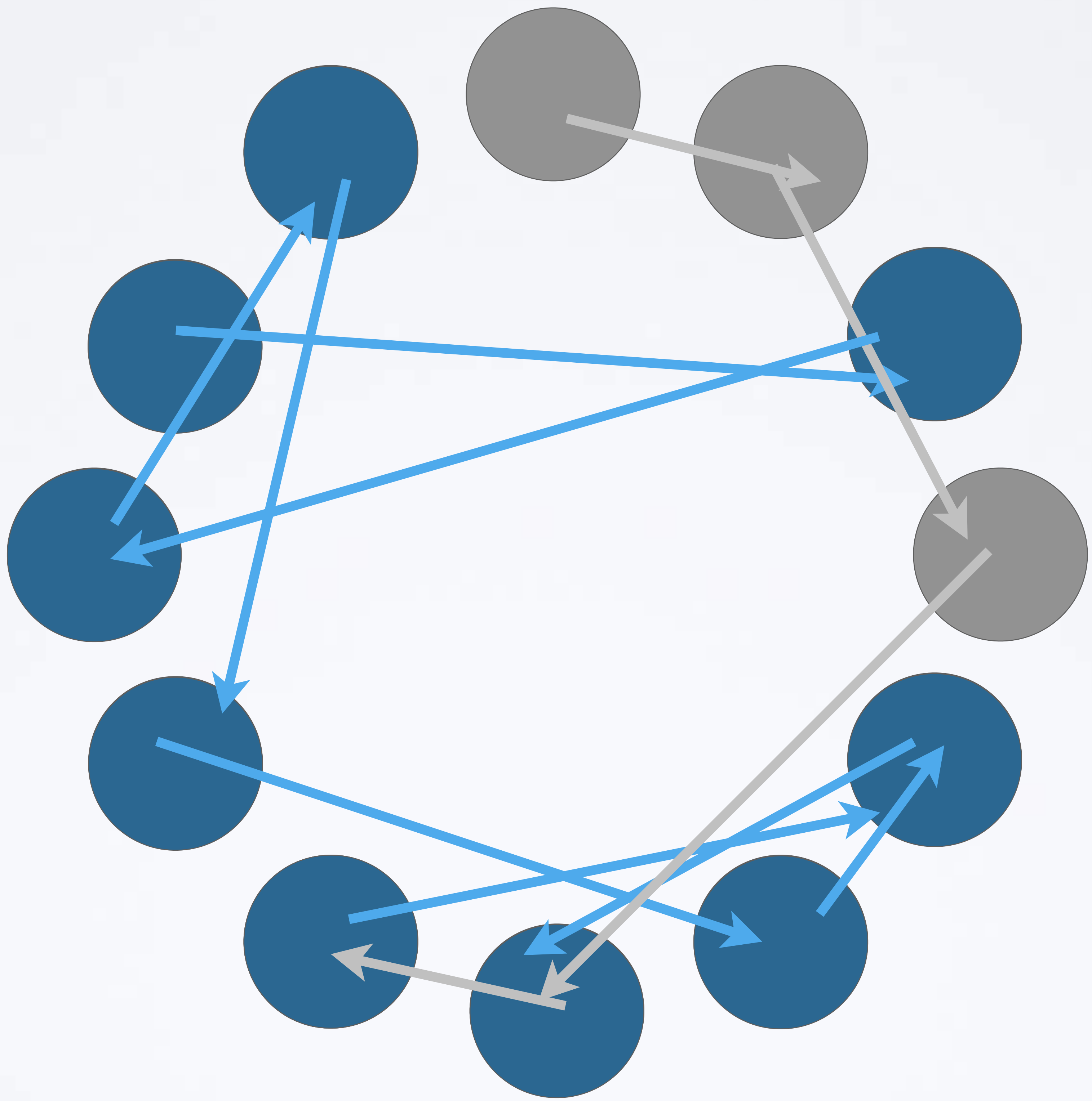
DIFFUSION SUMMARY

- **Best method to reduce:**
  - Migrations (less data movement)
  - Edge cut (less communication)
- **Load balance** good, but not superior
- **Flexible:** uses communication graph specific to application

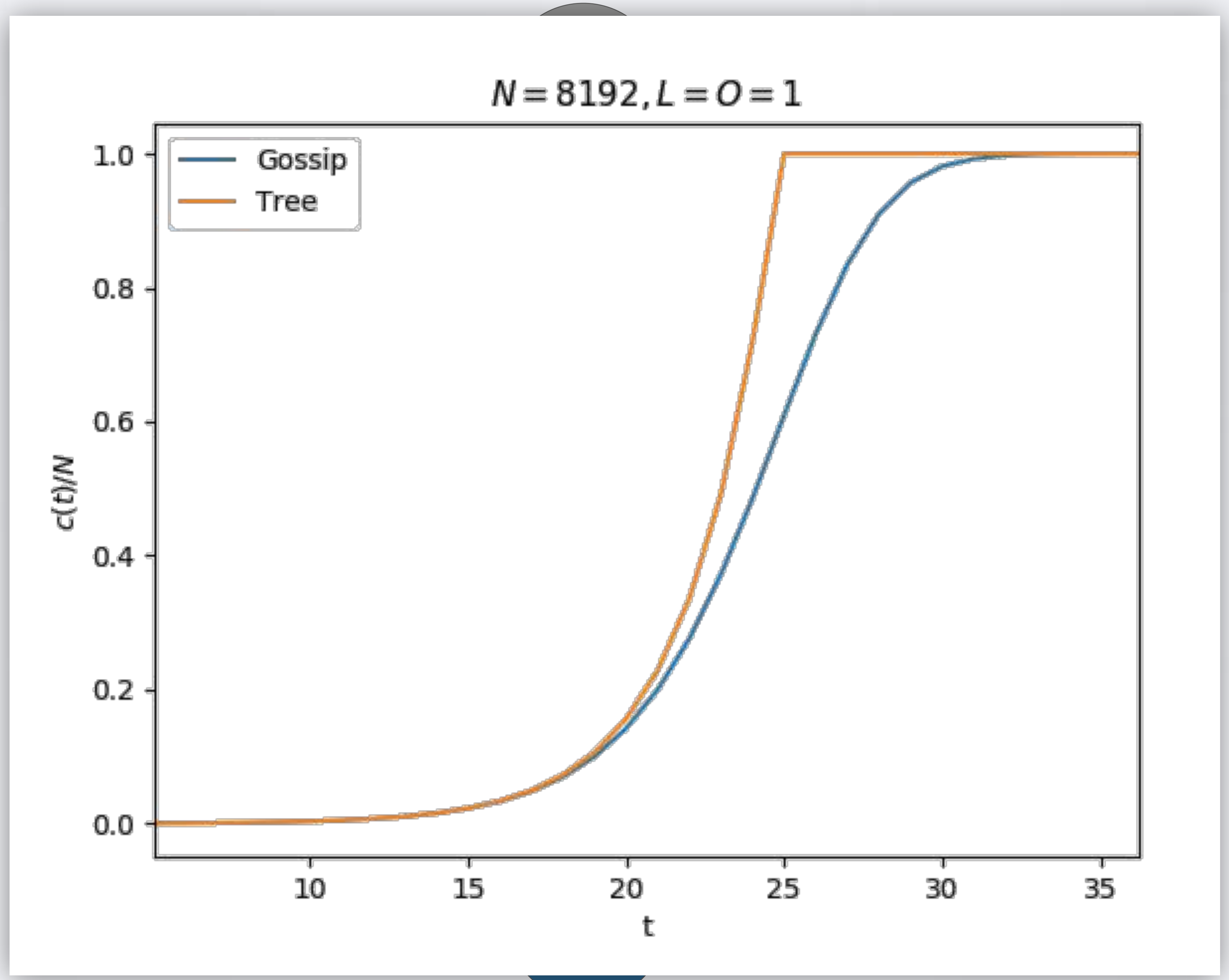
NODE ARCHITECTURE



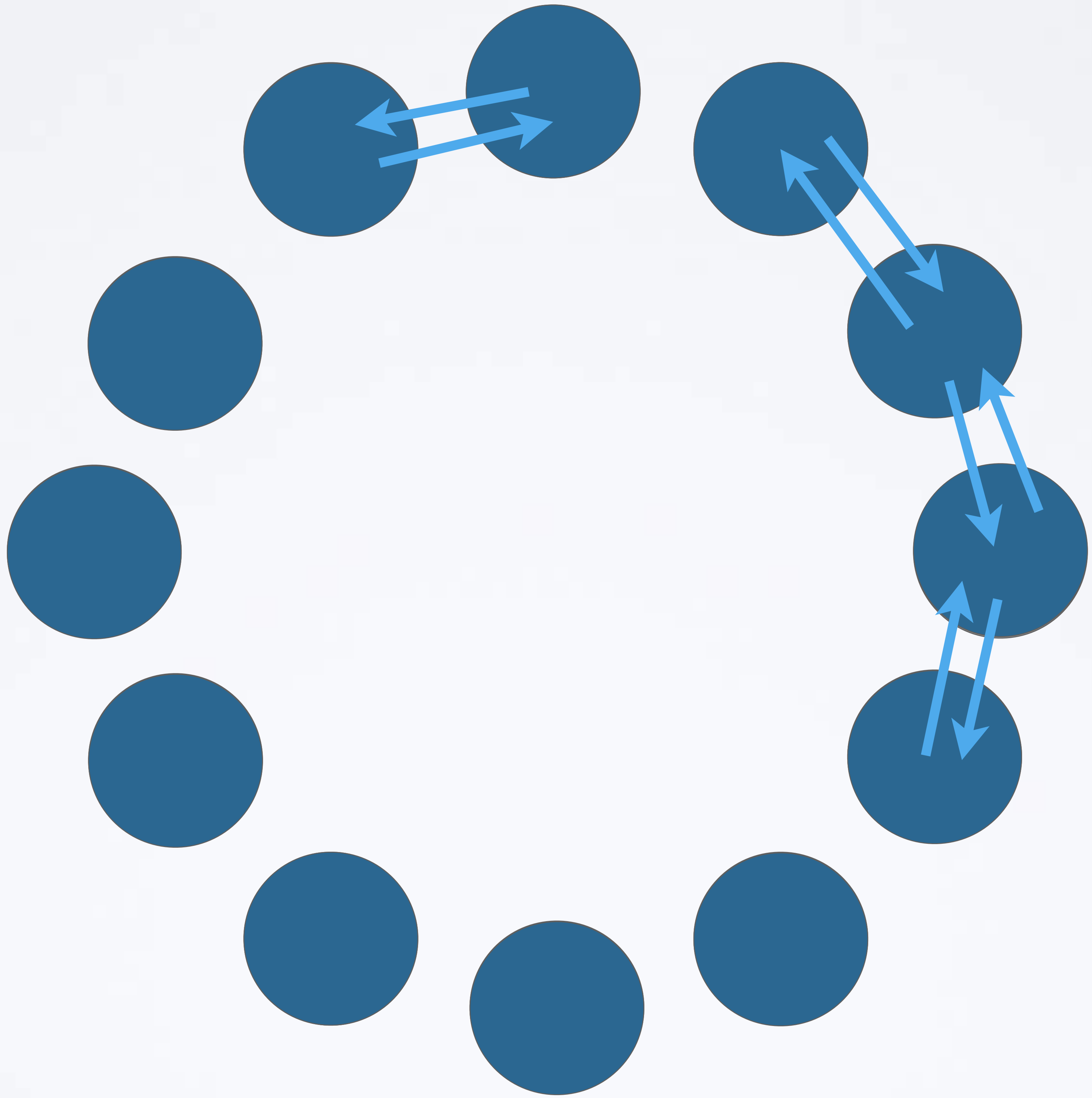
STEP 1: GOSSIP



STEP 1: GOSSIP



STEP 2: CORRECTION



FT COLLECTIVES

- Fault-tolerant **broadcast:** published[*]
- Fault-tolerant **Reduce** + **Allreduce**, collectives with builtin **fault-detection**
  - Formal analysis, measurements show: log-scalable, sturdy in most cases
- Resiliency for **tree-based collectives:**
  - Succeed / complete with failing nodes
  - Latency comparable to non-ft algorithms

[*] Torsten Hoefler, Amnon Barak, Amnon Shiloh and Zvi Drezner, "Corrected Gossip Algorithms for Fast Reliable Broadcast on Unreliable Systems", IPDPS'17, Orlando, FL, USA

SUMMARY

- **Decoupled interrupts:** faster wakeup
- **Checkpointing:** Global optimization
- **Diffusion:** Promising
- **Corrected Gossip & Trees:** fault-tolerant collective operations (maybe for MPI)
- **Integrated:** gossip + decision making
- **WIP:** integrate monitoring + migration

DFG Deutsche Forschungsgemeinschaft
SPPEXA German Priority Programme 1648
Software for Exascale Computing