TECHNISCHE UNIVERSITÄT DRESDEN

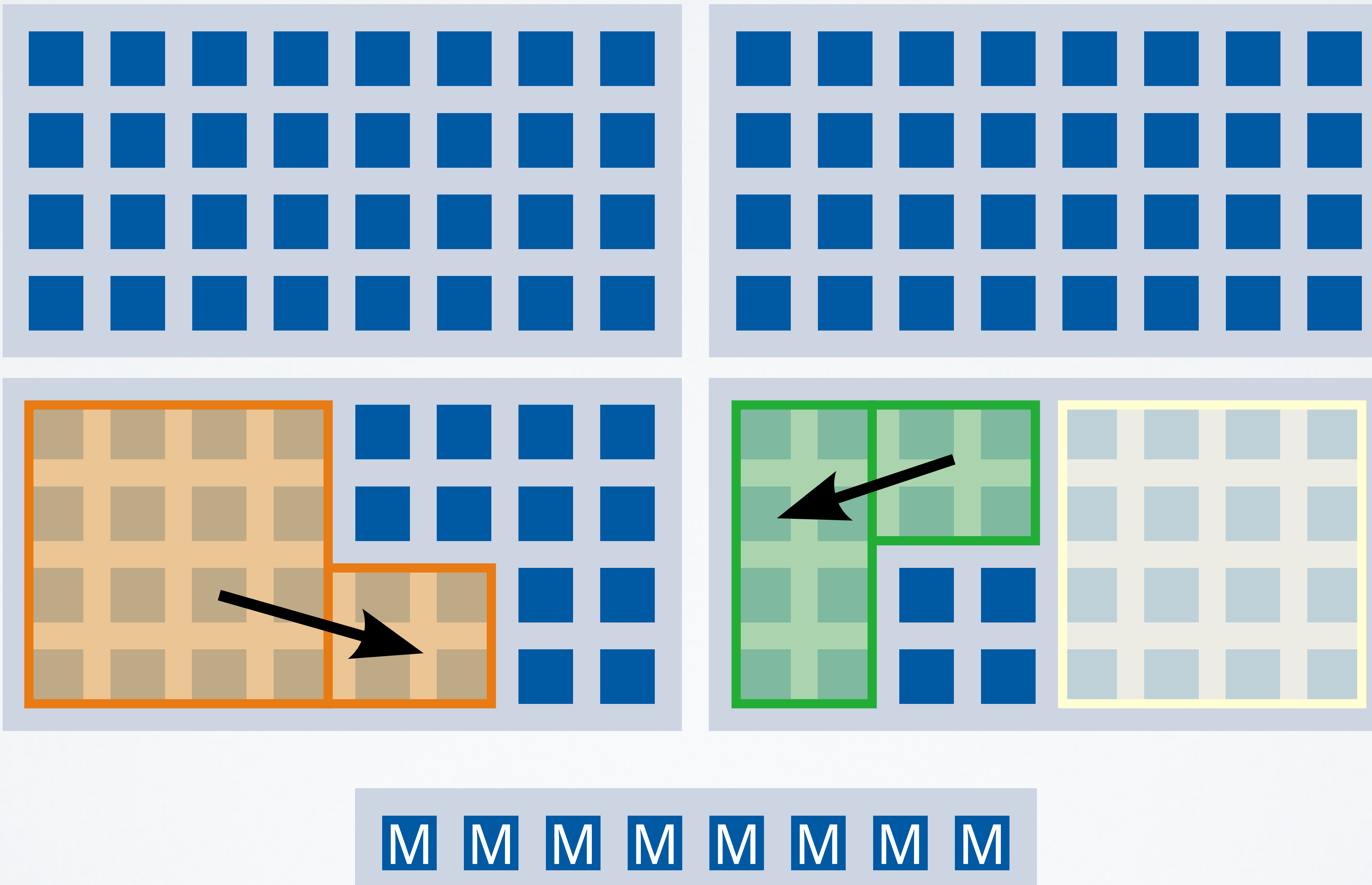
The Hebrew University of Jerusalem

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

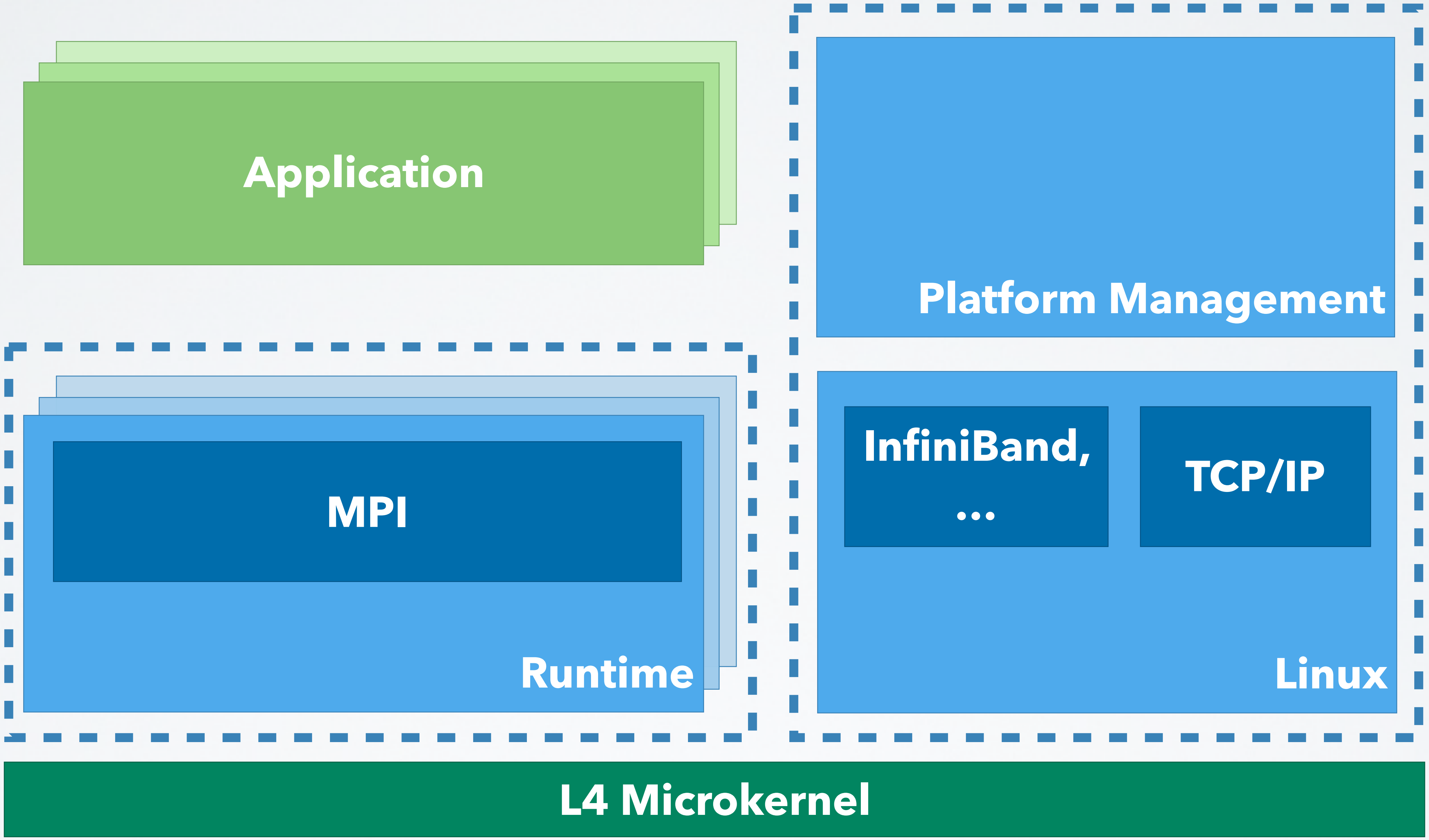
Amnon Barak	Hebrew University Jerusalem (HUJI)
Hermann Härtig	TU Dresden, Operating Systems Group (TUDOS)
Wolfgang E. Nagel	TU Dresden, Center for Information Services and HPC (ZIH)
Alexander Reinefeld	Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

CARSTEN WEINHOLD, TU DRESDEN

SYSTEM MODEL

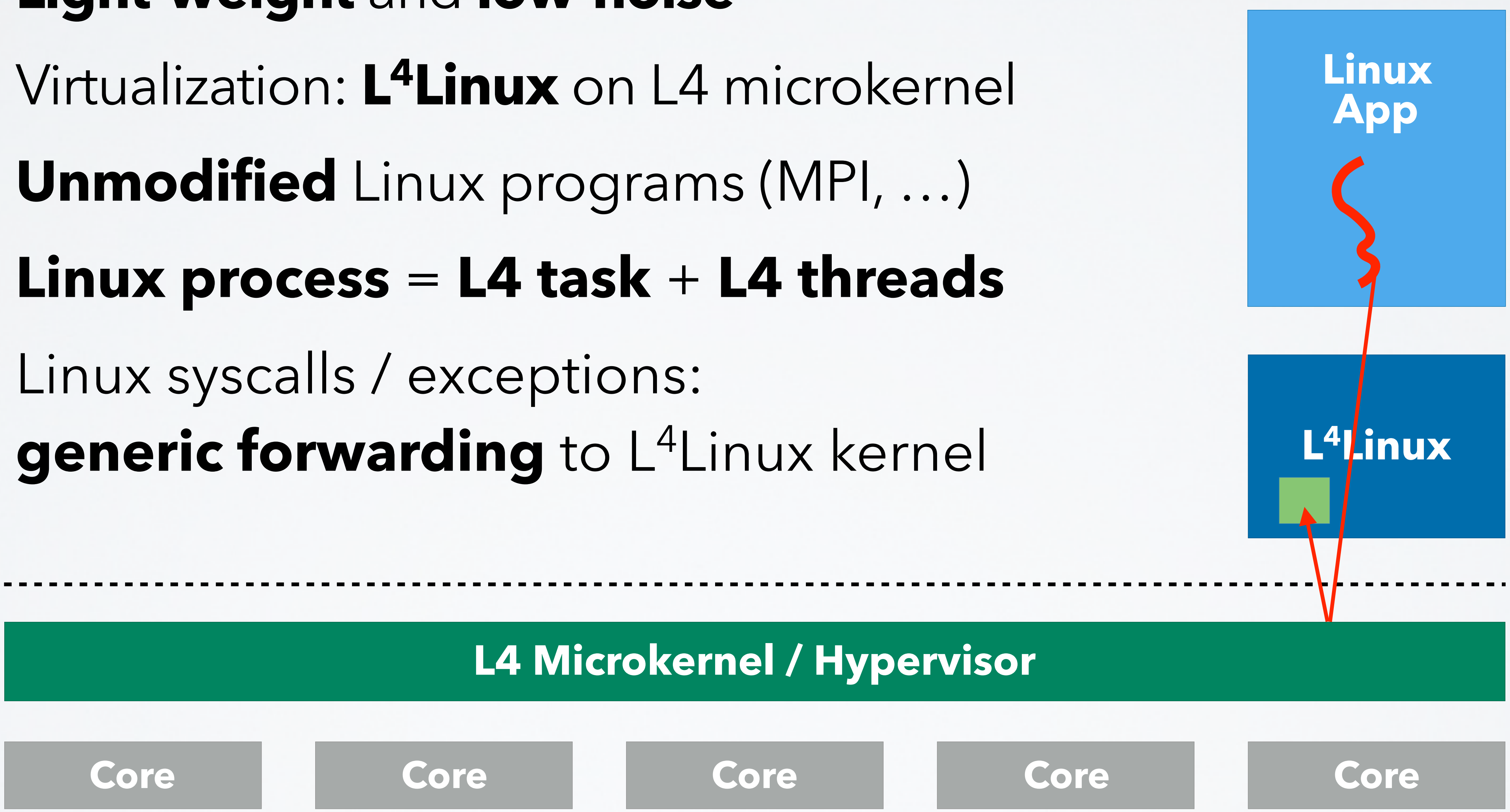


NODE ARCHITECTURE



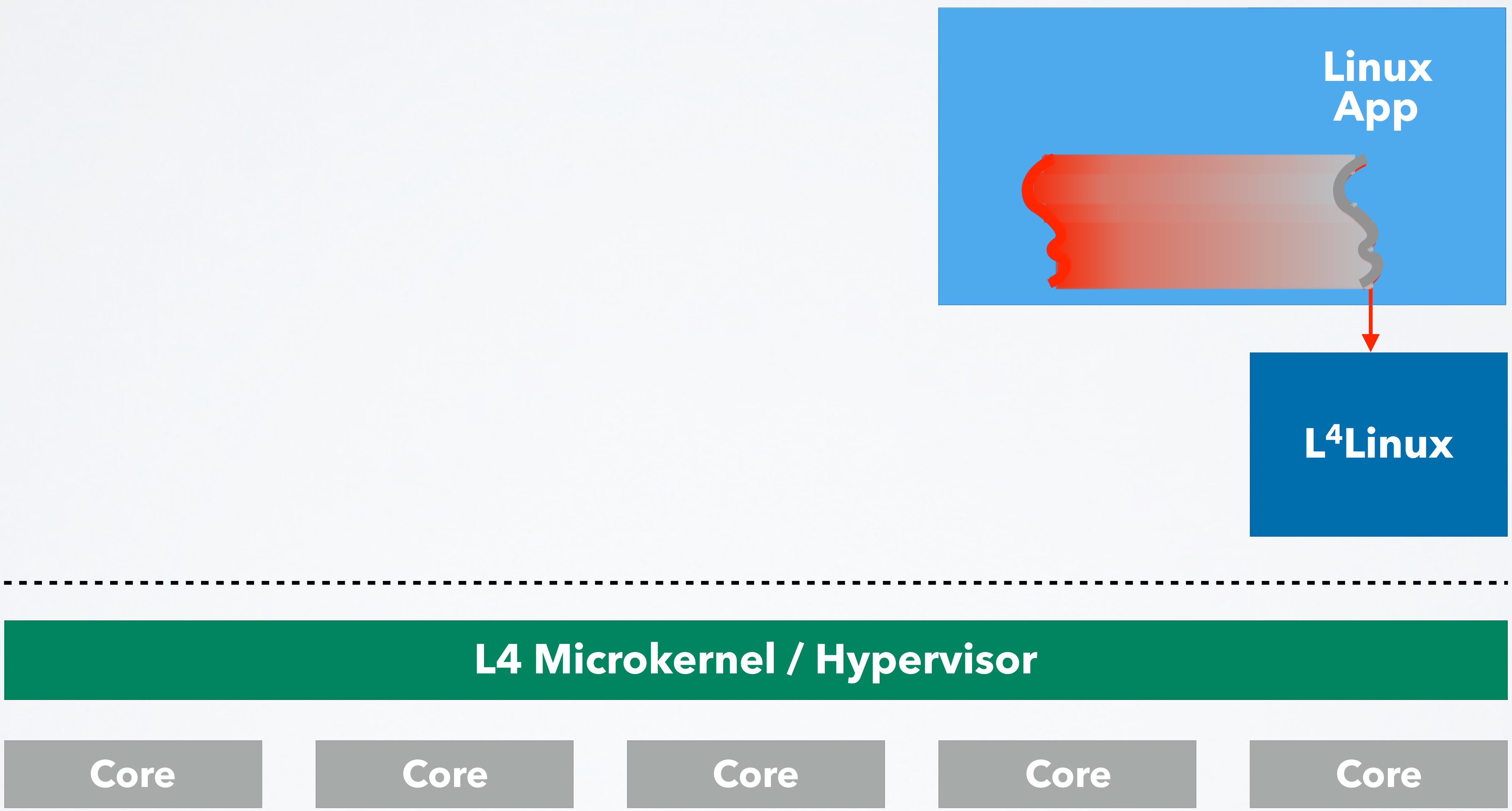
L4 + L^4LINUX

- **L4 microkernel** controls the node
- **Light-weight** and **low-noise**
- Virtualization: **L^4Linux** on L4 microkernel
- **Unmodified** Linux programs (MPI, …)
- **Linux process** = **L4 task** + **L4 threads**
- Linux syscalls / exceptions:
  generic forwarding to L^4Linux kernel



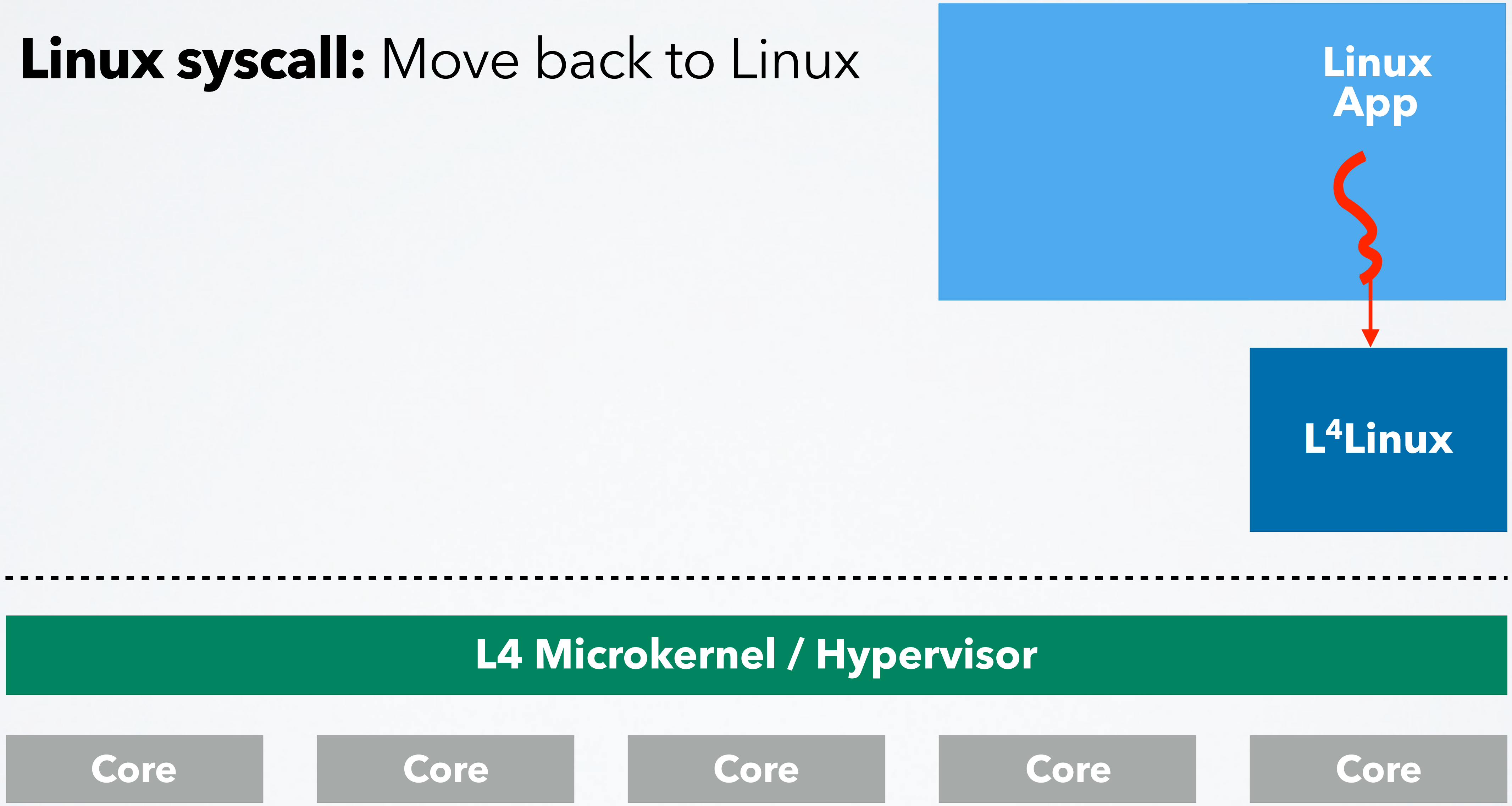
DECOUPLED EXECUTION

- **Decoupling:** move Linux thread to new L4 thread on its own core



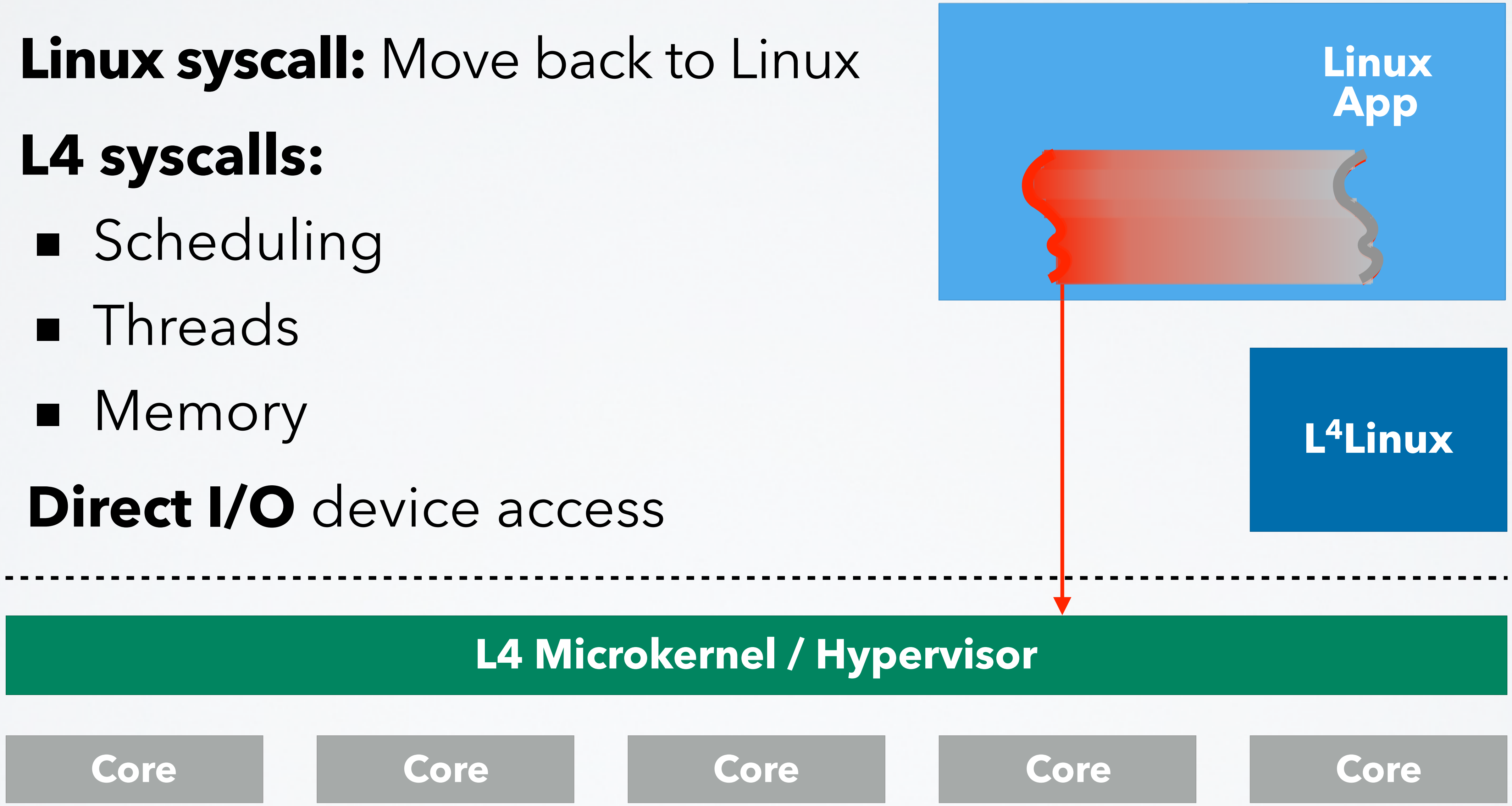
DECOUPLED EXECUTION

- **Decoupling:** move Linux thread to new L4 thread on its own core
- **Linux syscall:** Move back to Linux

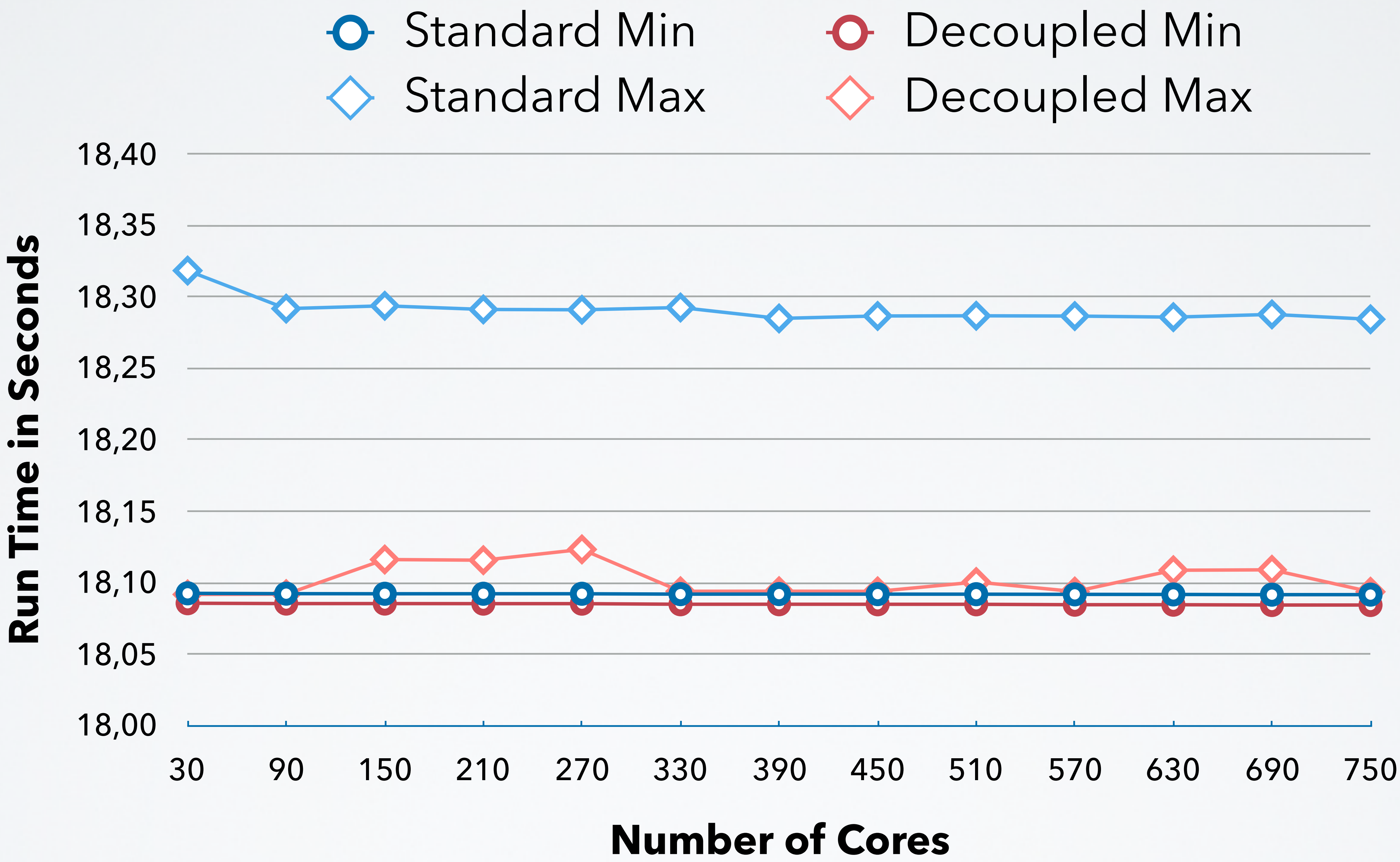


DECOUPLED EXECUTION

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

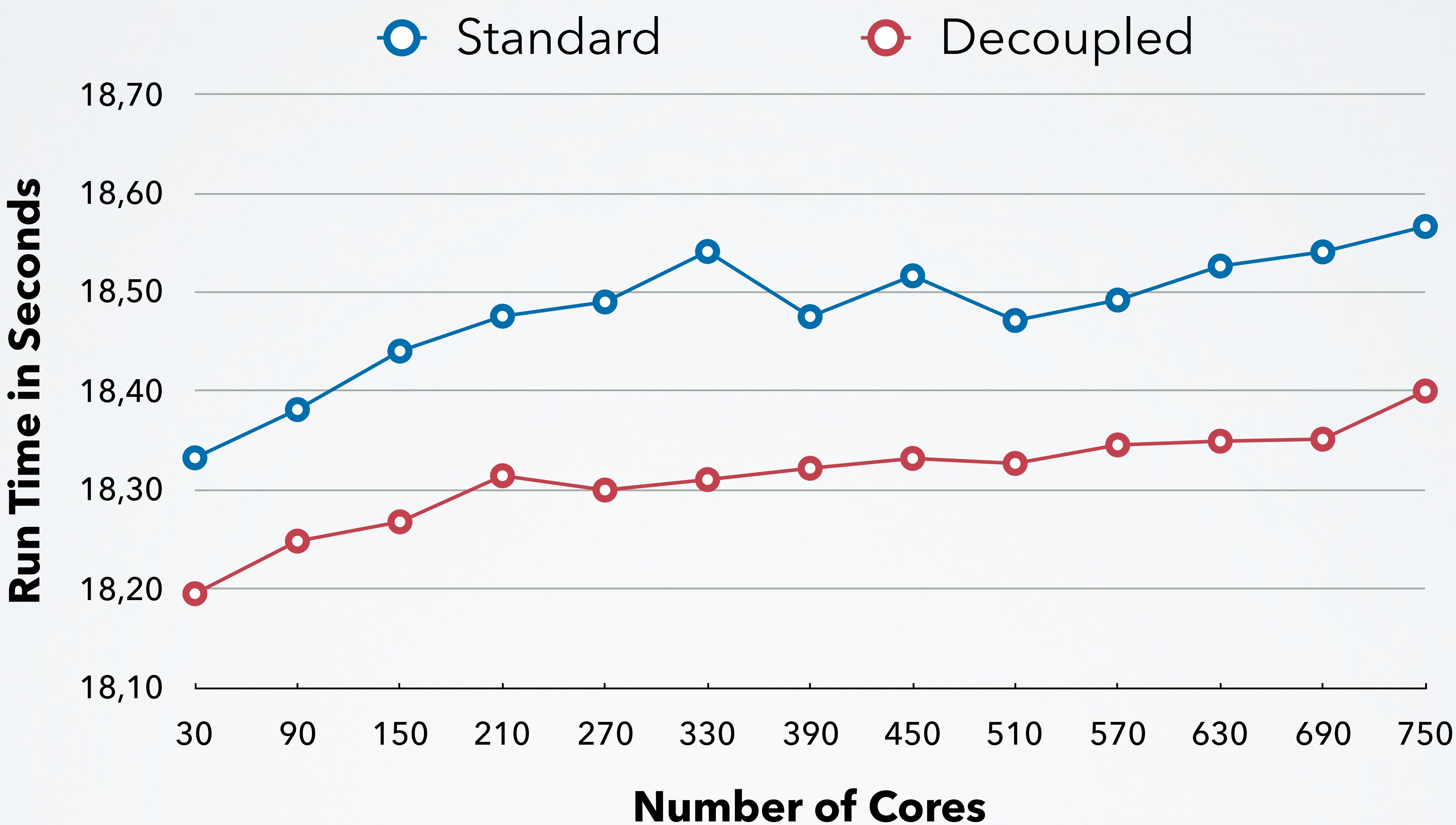


DECOUPLING: EP



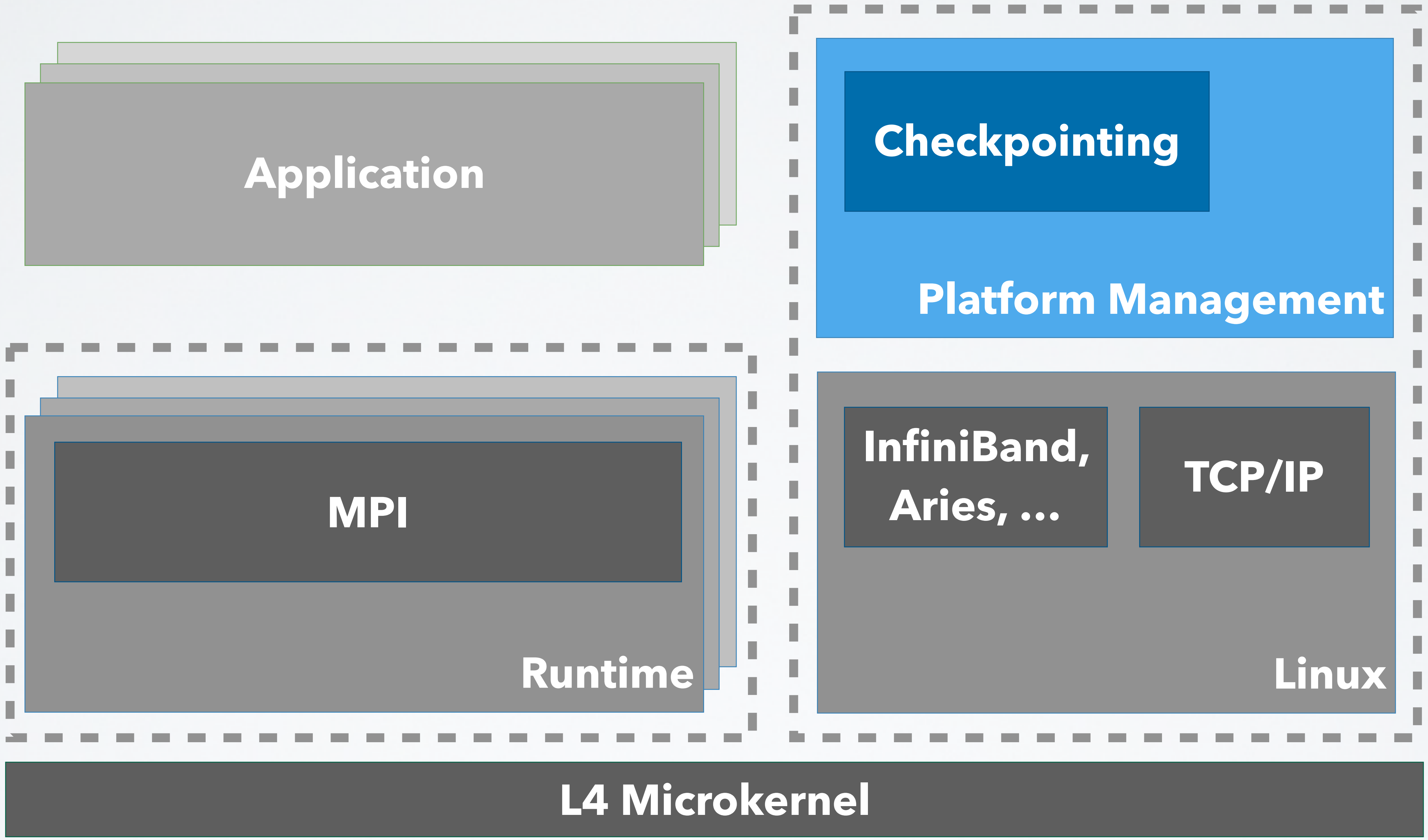
Adam Lackorzynski, Carsten Weinhold, Hermann Härtig, "Decoupled: Low-Effort Noise-Free Execution on Commodity Systems", ROSS 2016, June 2016, Kyoto, Japan

DECOUPLING: BSP



Adam Lackorzynski, Carsten Weinhold, Hermann Härtig, "Decoupled: Low-Effort Noise-Free Execution on Commodity Systems", ROSS 2016, June 2016, Kyoto, Japan

NODE ARCHITECTURE

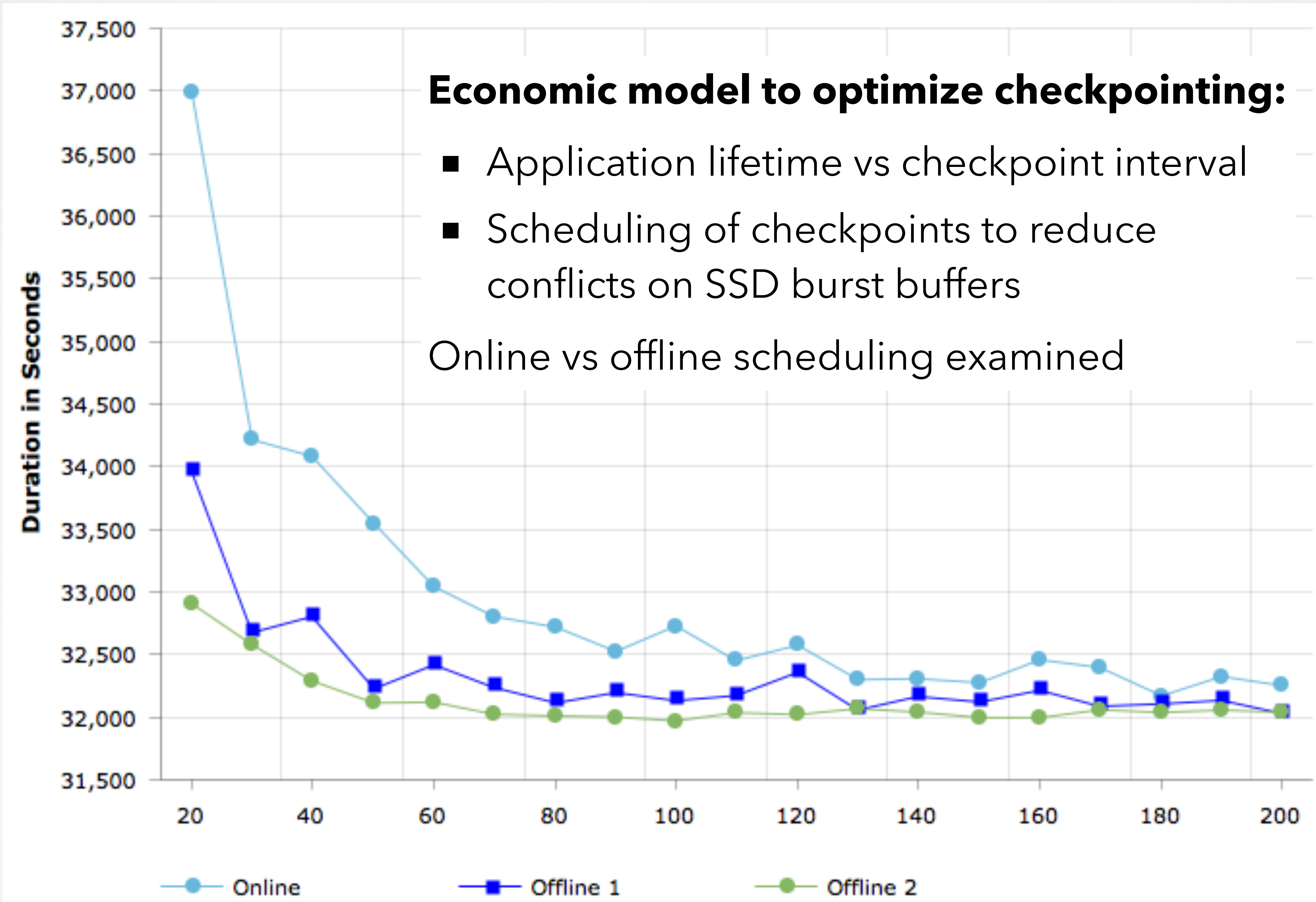


CHECKPOINTING

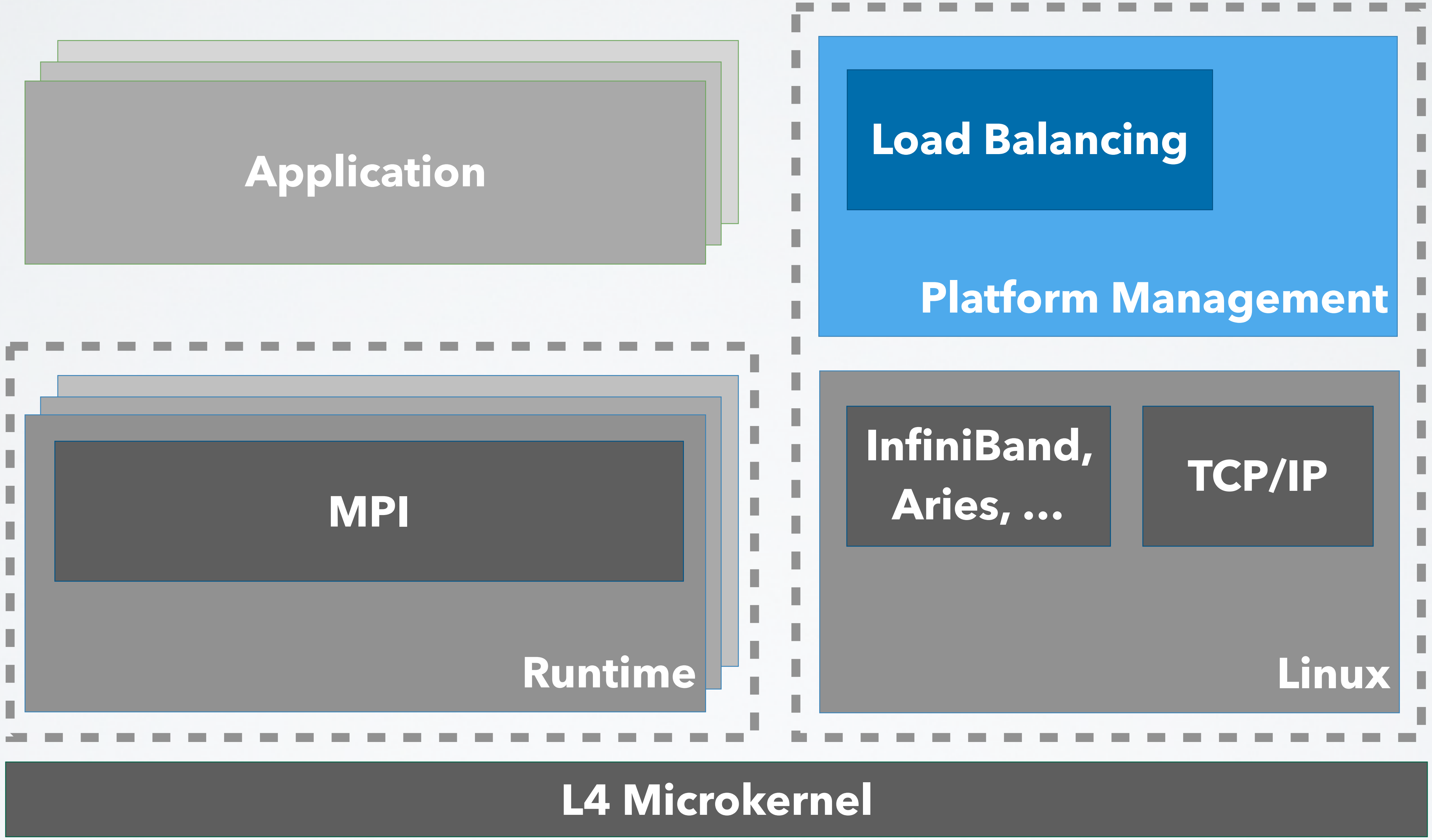
Economic model to optimize checkpointing:

- Application lifetime vs checkpoint interval
- Scheduling of checkpoints to reduce conflicts on SSD burst buffers

Online vs offline scheduling examined

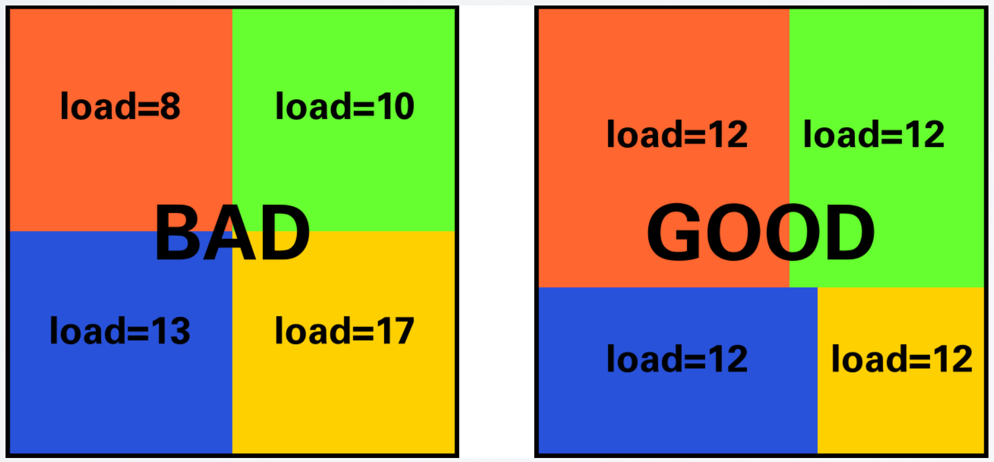


NODE ARCHITECTURE



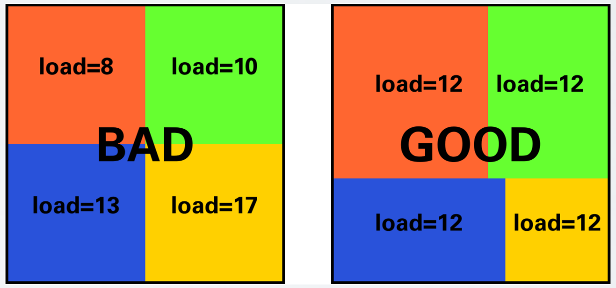
LOAD BALANCING

- Balance workload

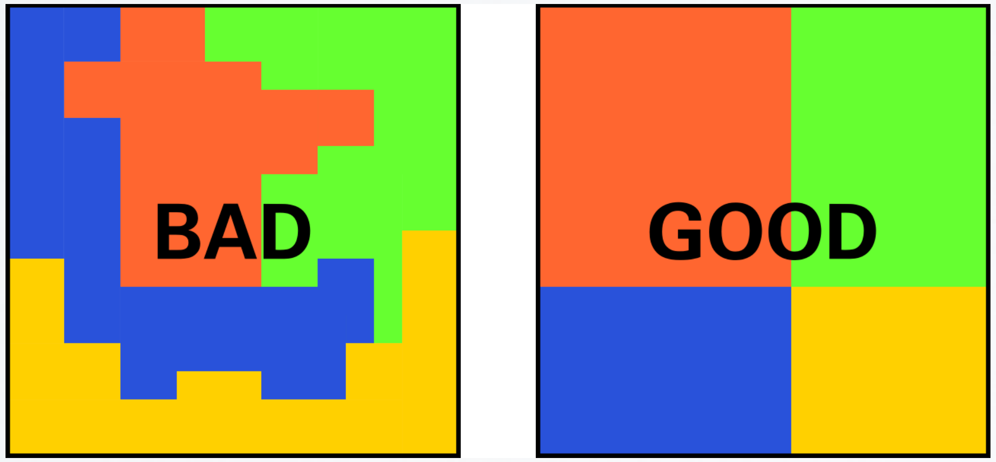


LOAD BALANCING

- Balance workload

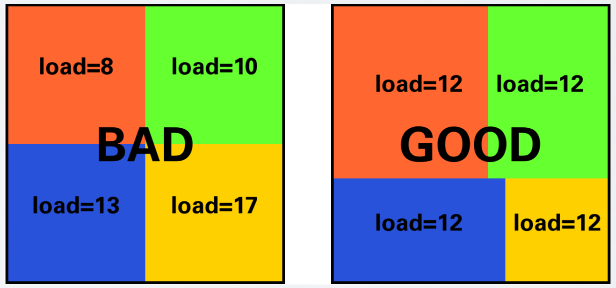


- Minimize communication between partitions

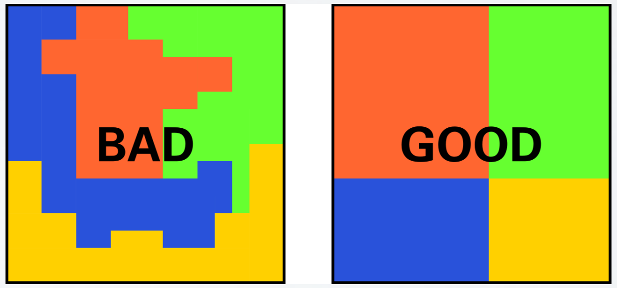


LOAD BALANCING

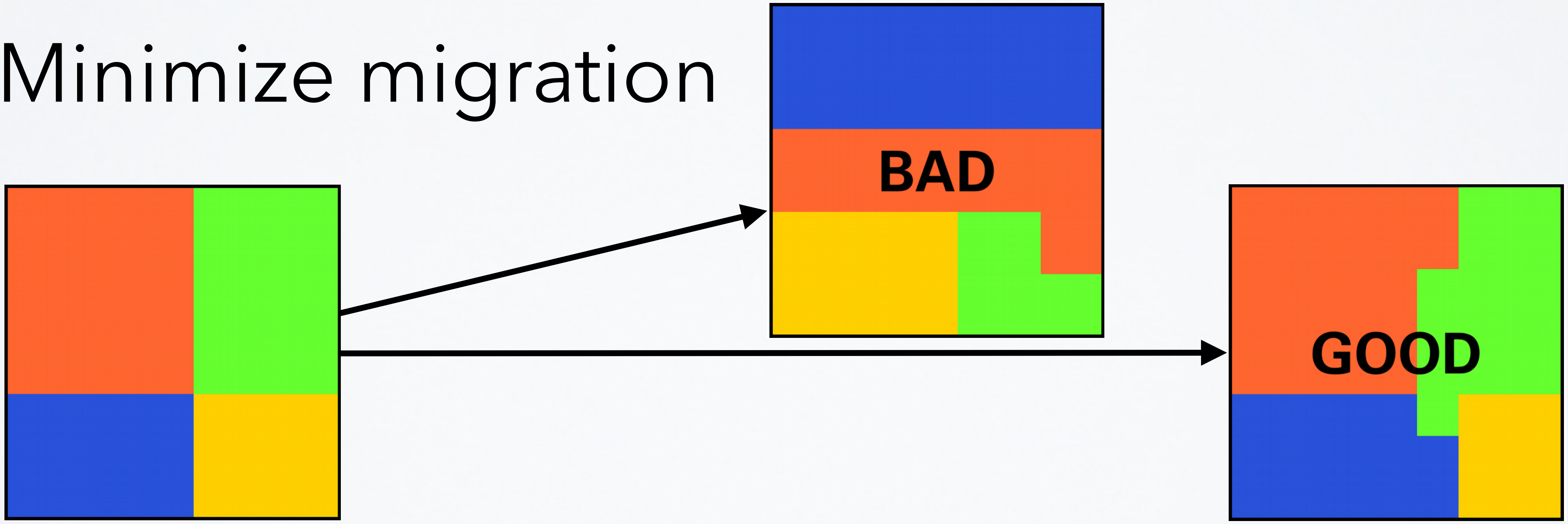
- Balance workload



- Minimize communication between partitions

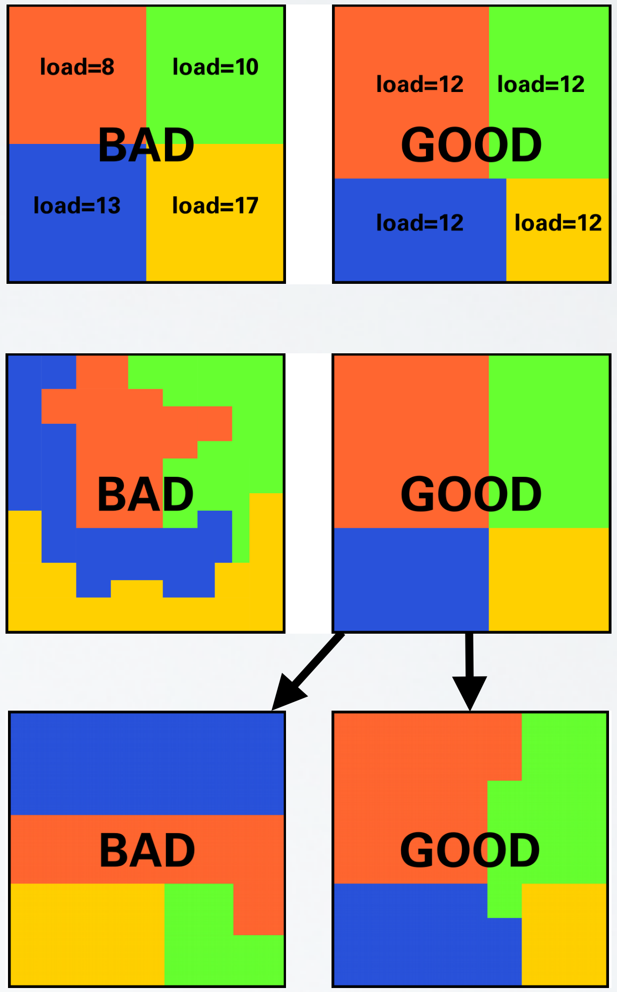


- Minimize migration

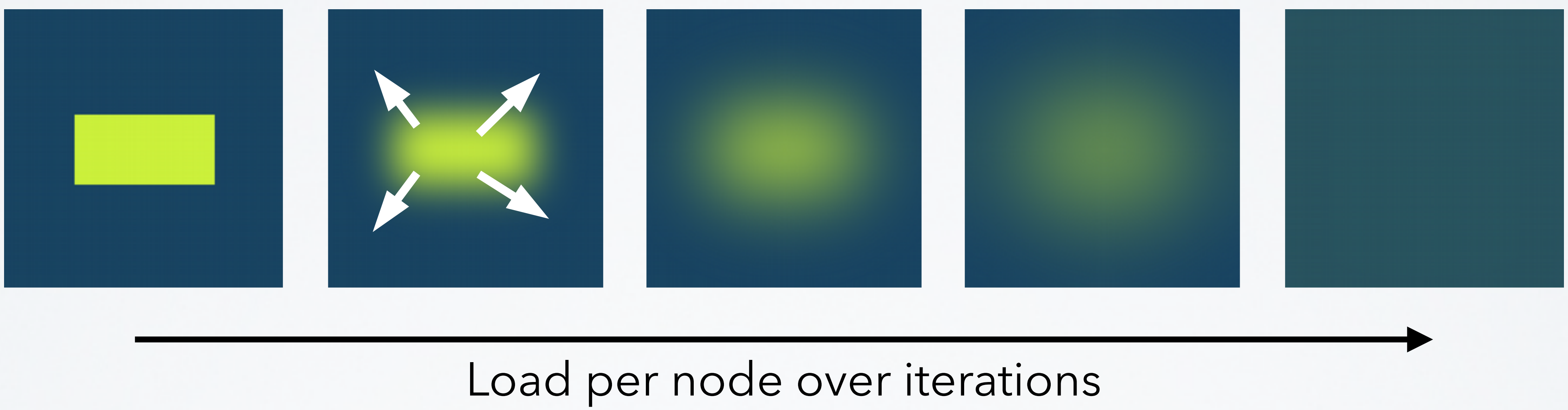


LOAD BALANCING

- Balance workload
- Minimize communication between partitions
- Minimize migration
- Compute new partitions fast



DIFFUSION

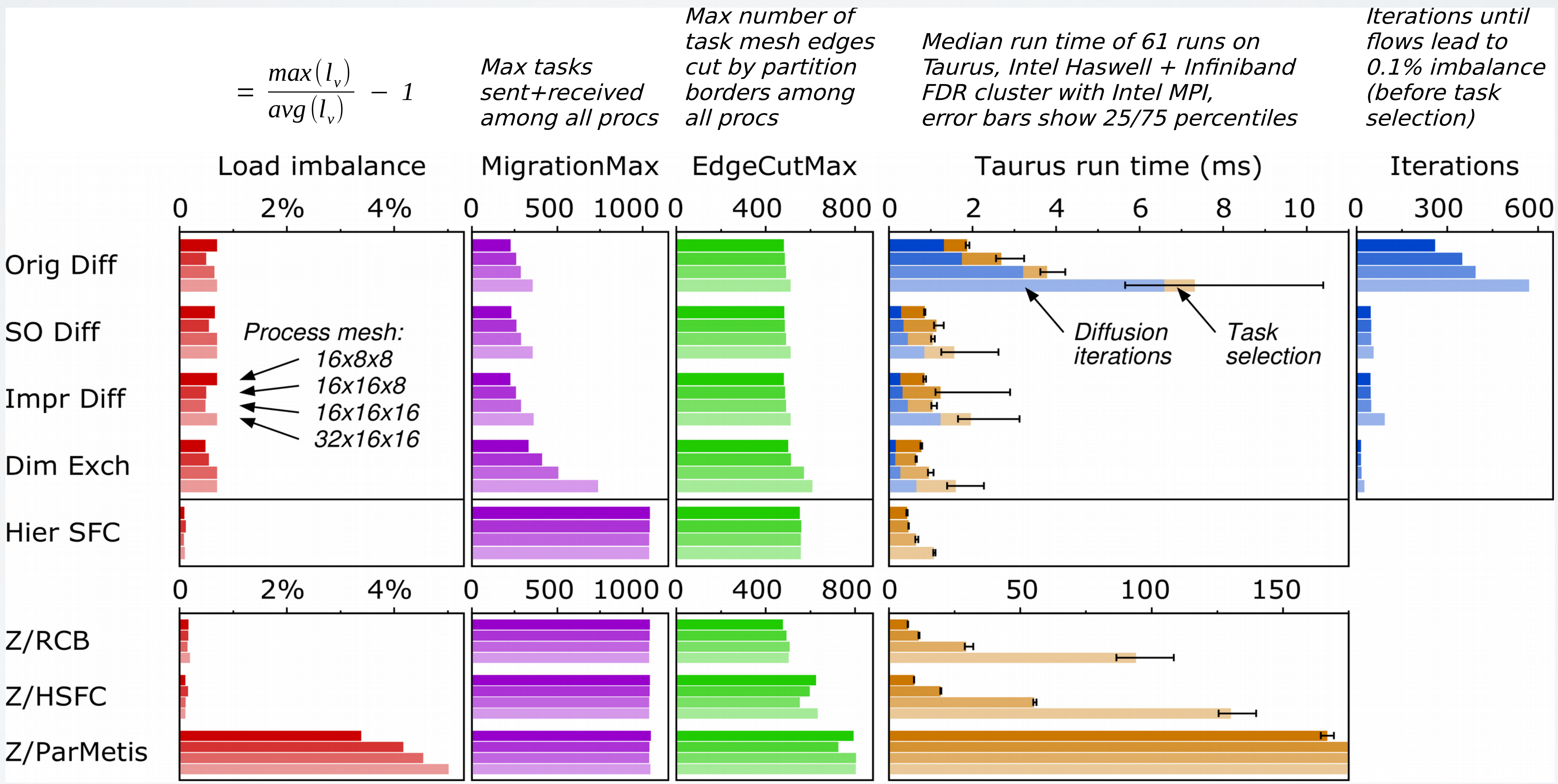


TECHNISCHE UNIVERSITÄT DRESDEN

ZIB

The Hebrew University of Jerusalem

DIFFUSION RESULTS



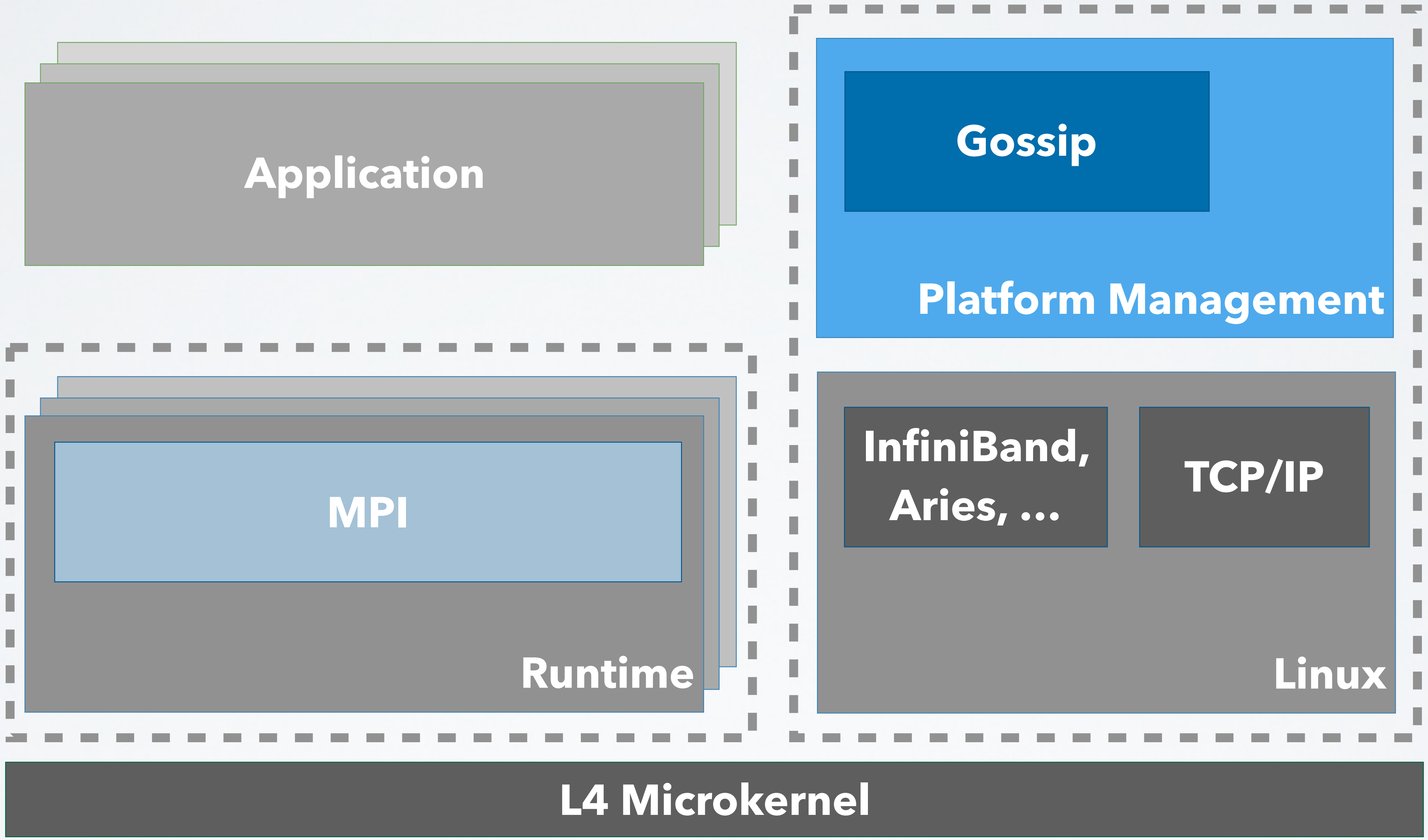
Diffusion leads to smallest migration

Diffusion achieves very good edge cut

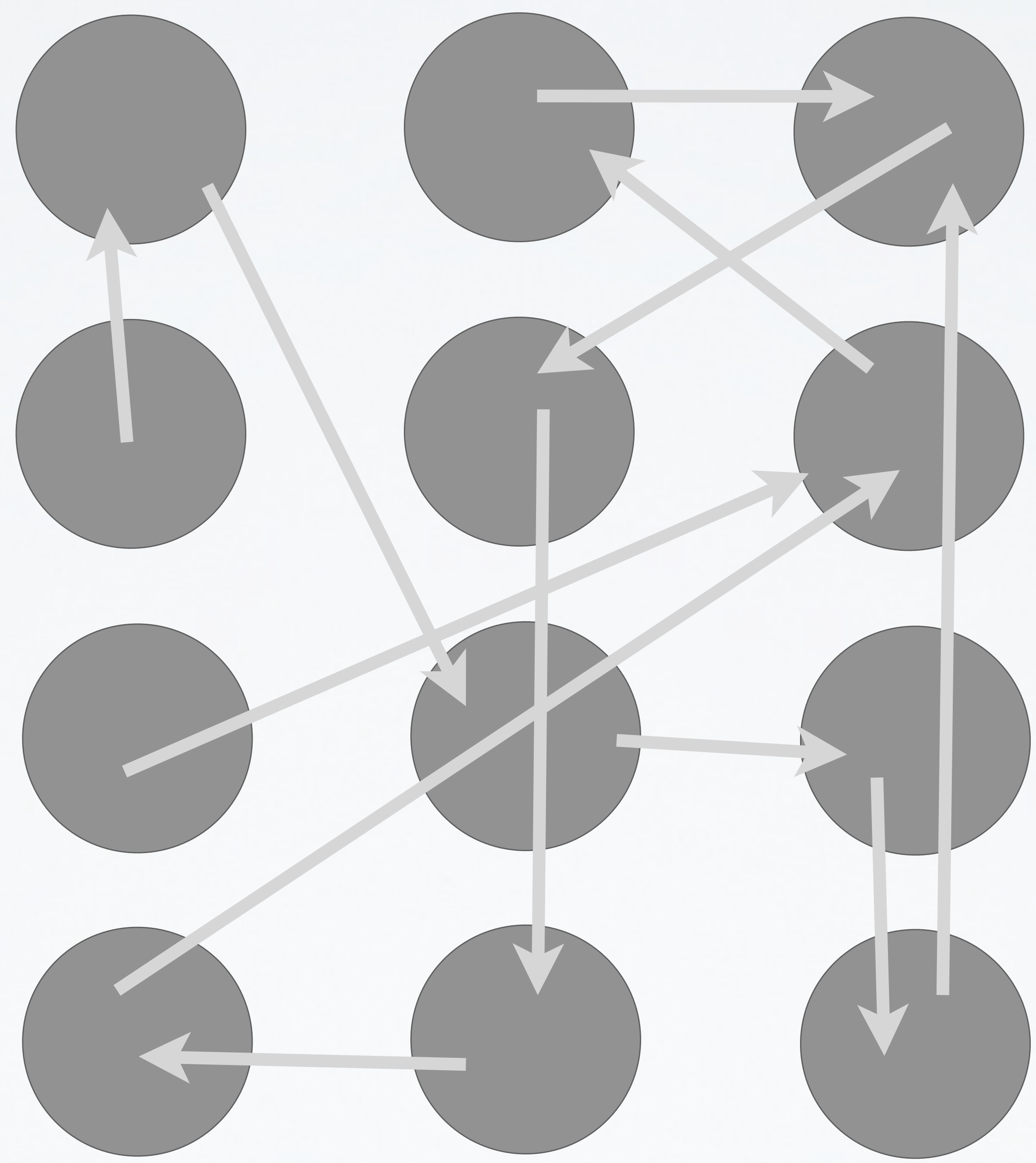
Diffusion run time ~2 ms for 8192 processes, Zoltan much slower

Matthias Lieber, Kerstin Gößner, Wolfgang E. Nagel, "The Potential of Diffusive Load Balancing at Large Scale", EuroMPI 2016, June 2016, Edinburgh, United Kingdom

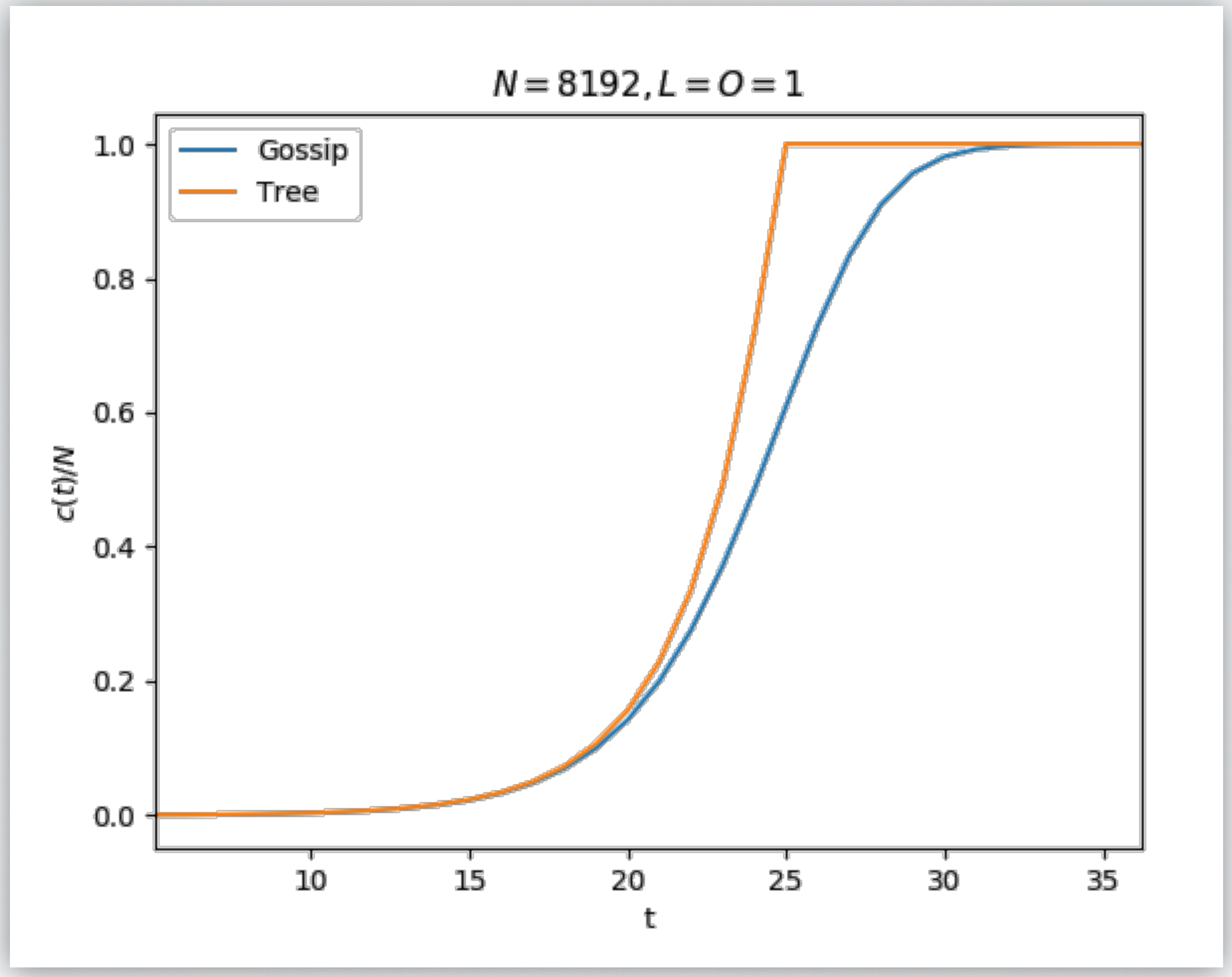
NODE ARCHITECTURE



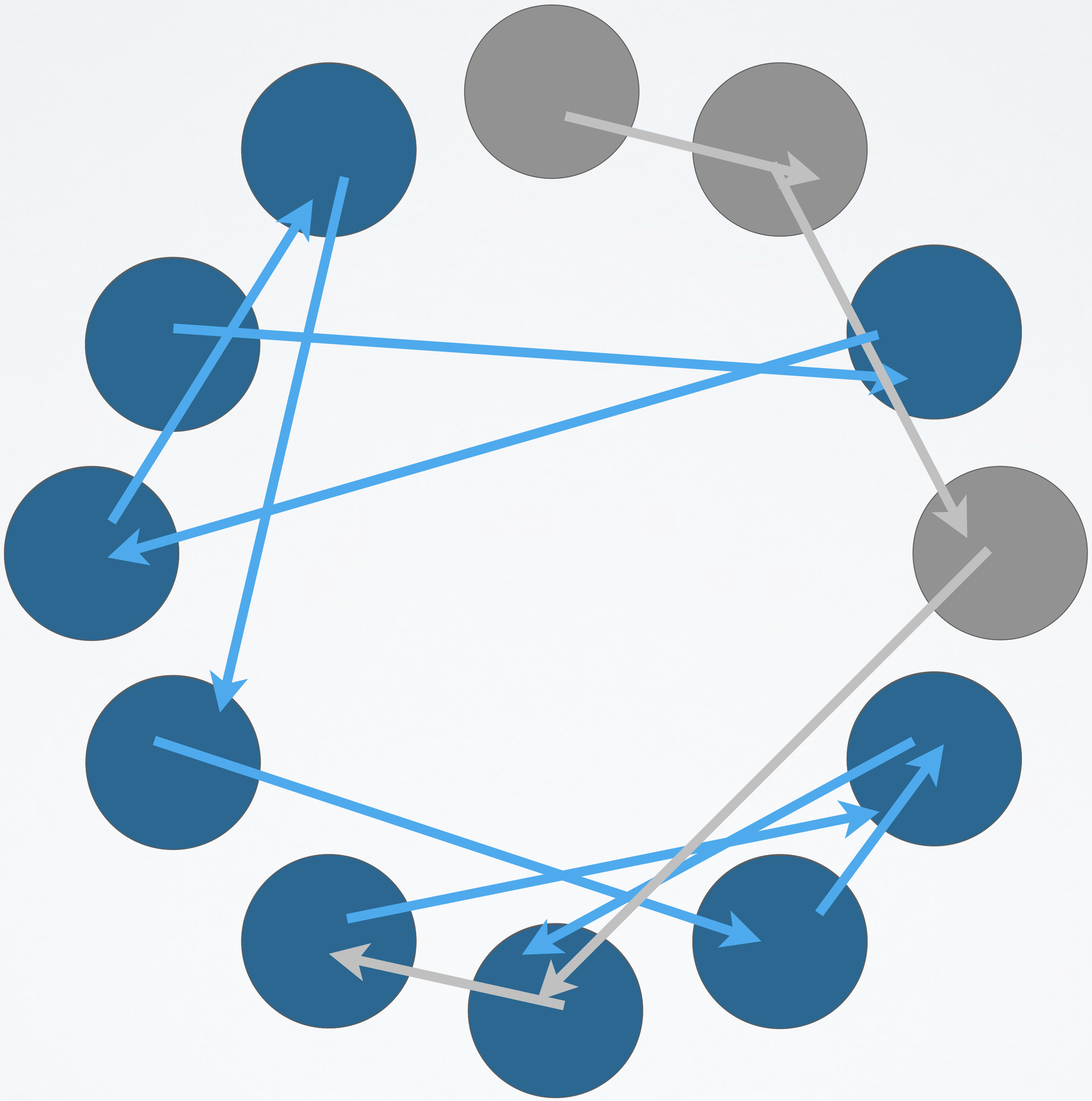
GOSSIP



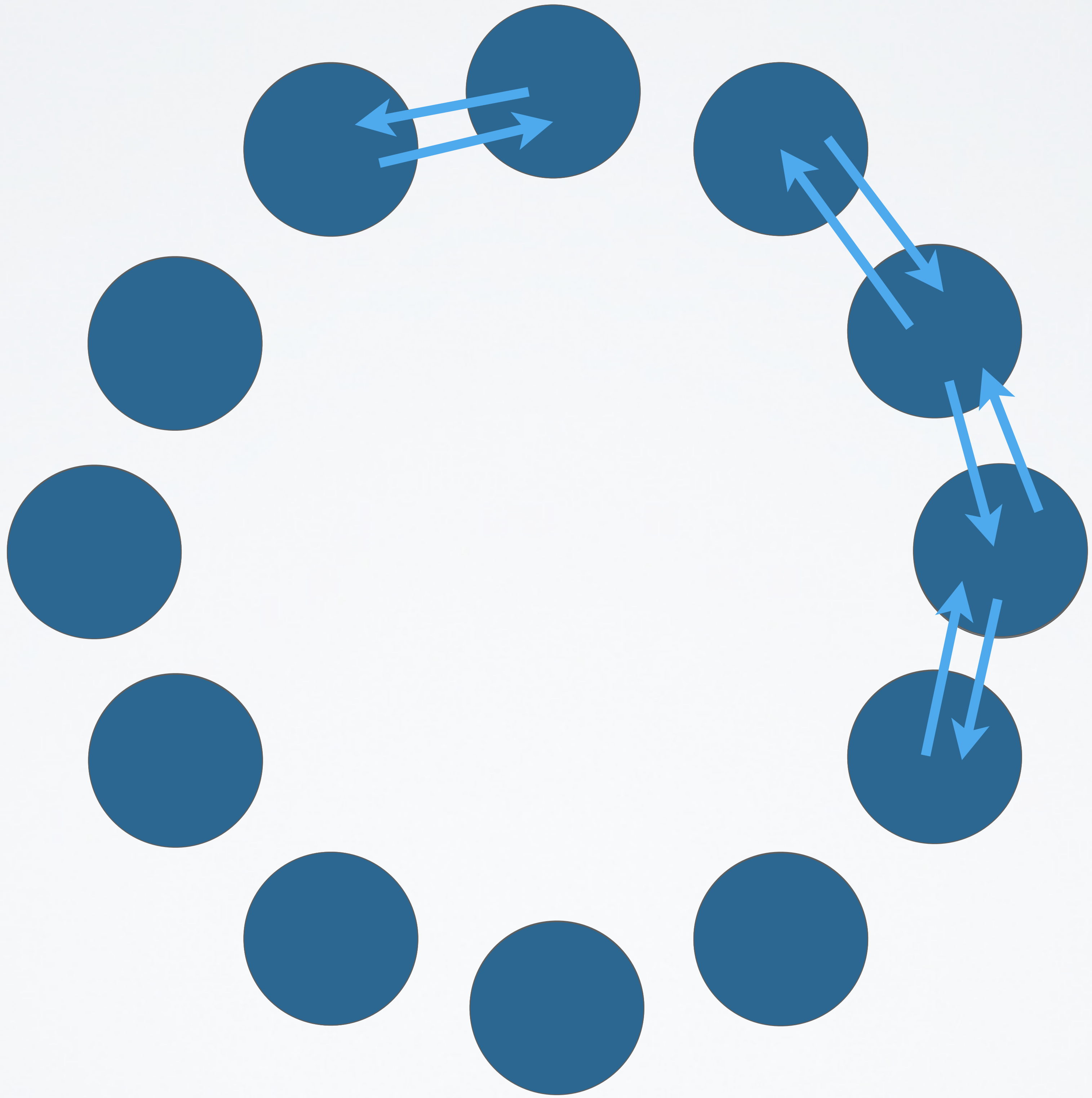
GOSSIP



STEP 1: GOSSIP



STEP 2: CORRECTION



FT COLLECTIVES

- **Two-stage algorithm:** gossip + correction
- **Main advantage:** scalability and resilience (continues to work in presence of failures)
- **Works for:** fault-tolerant broadcast
- **Next step:** extend to operations that include barrier semantics
- **Future:** use in MPI?

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

SUMMARY

- **Decoupled threads:** reduced noise
- **Checkpointing:** Economic model
- **Diffusion:** may be efficient alternative
- **Corrected Gossip:** fault-tolerant broadcast
- **Work in progress:** integrate monitoring + gossip + decision making + migration

Deutsche Forschungsgemeinschaft
SPPEXA German Priority Programme 1648
Software for Exascale Computing