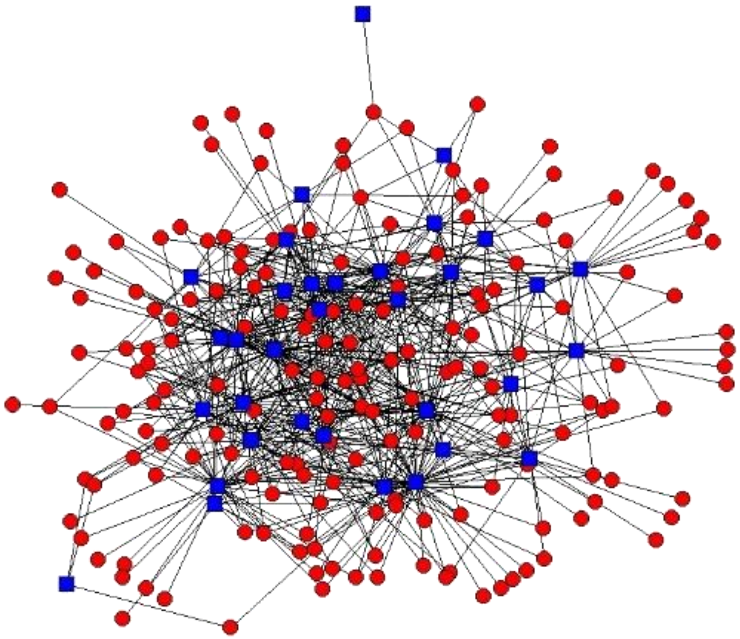
High-Performance Distributed RMA Locks

PATRICK SCHMID, MACIEJ BESTA, TORSTEN HOEFLER

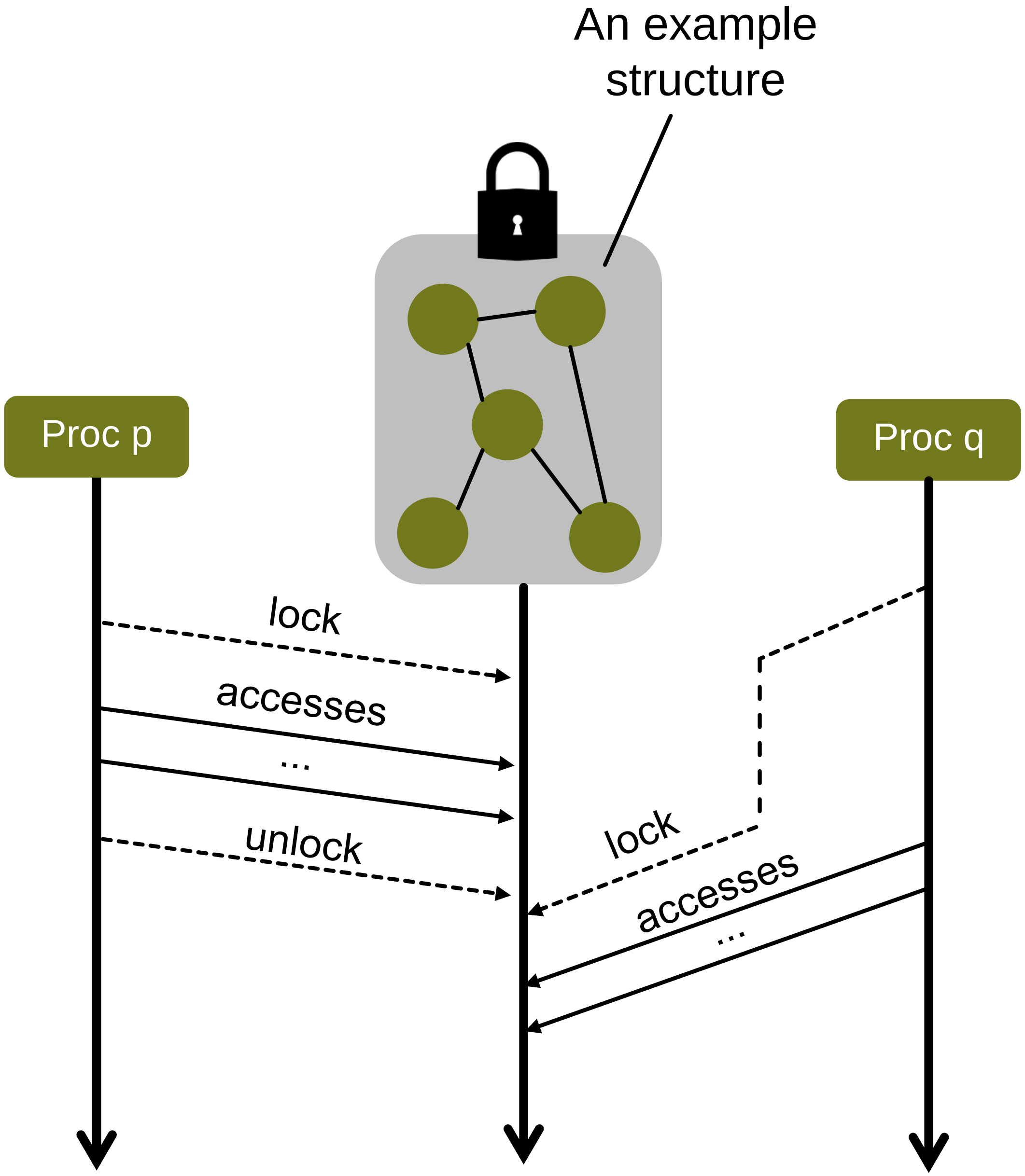
NEED FOR EFFICIENT LARGE-SCALE SYNCHRONIZATION



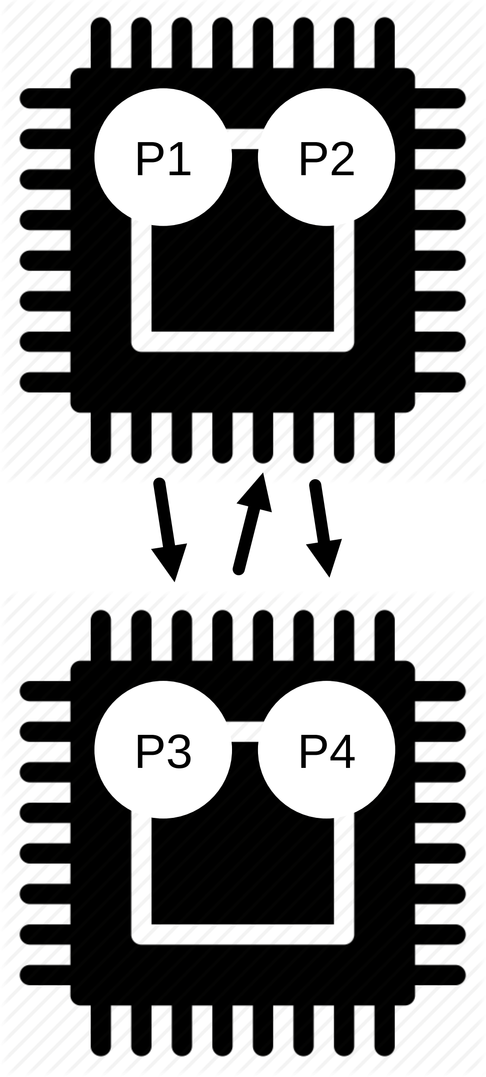
Locks

✓ Inuitive semantics

✗ Various performance penalties

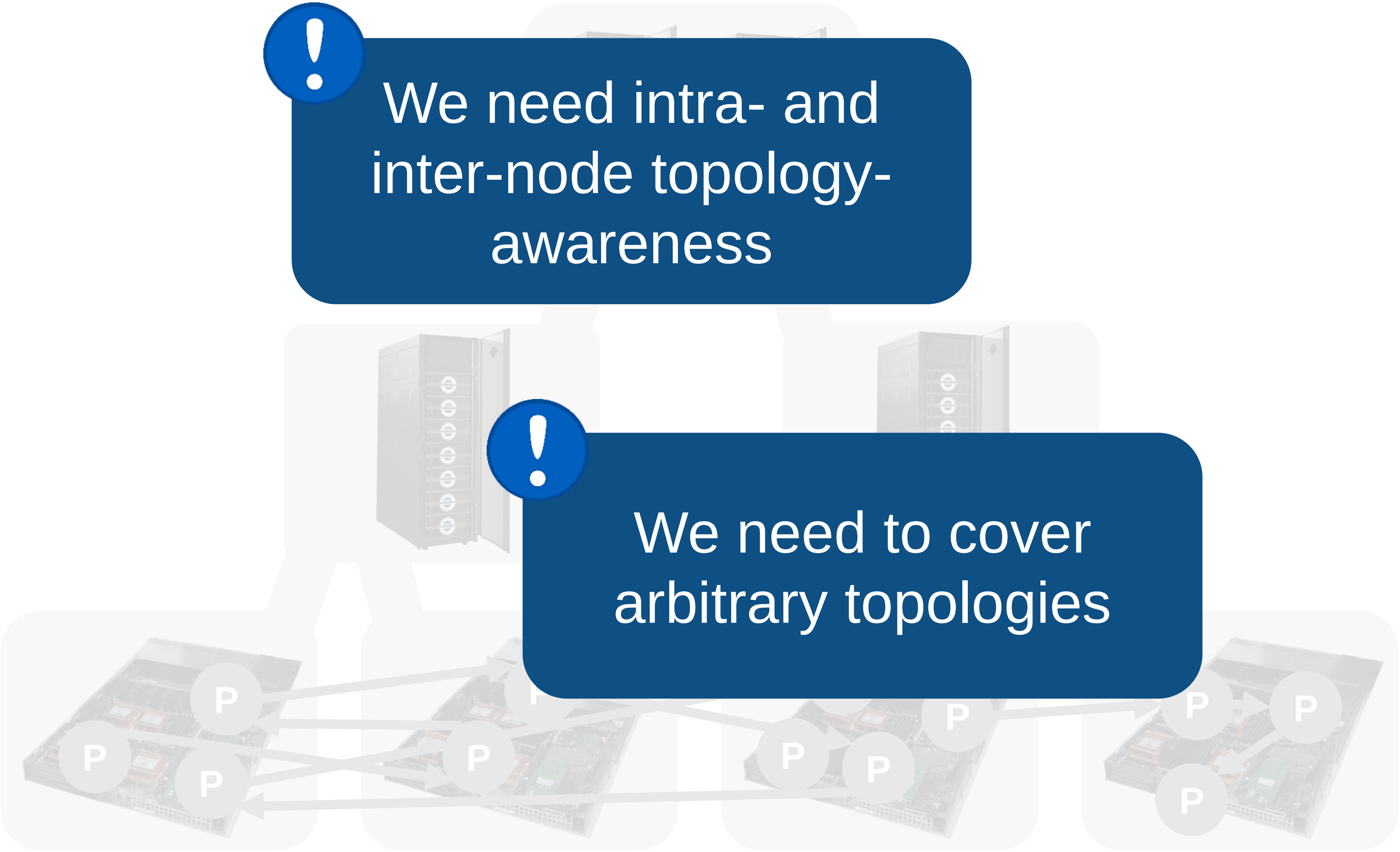


LOCKS: CHALLENGES



Calciu et al.: NUMA-aware reader-writer locks, PPoPP'13

Locks: Challenges



We need intra- and inter-node topology-awareness

We need to cover arbitrary topologies

Locks: Challenges

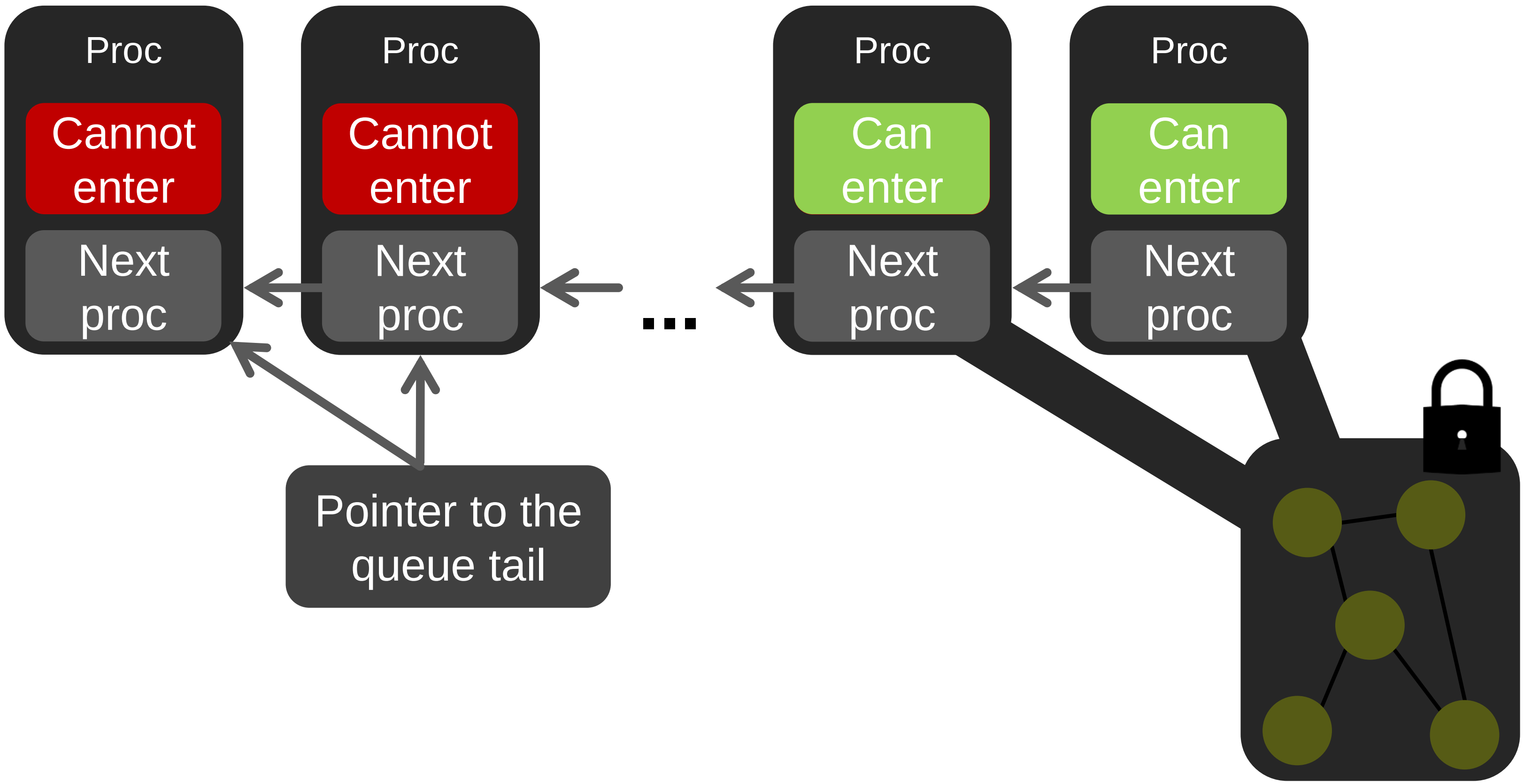
We need to distinguish between readers and writers

We need flexible performance for both types of processes

[1] V. Venkataramani et al. Tao: How facebook serves the social graph. SIGMOD'12.

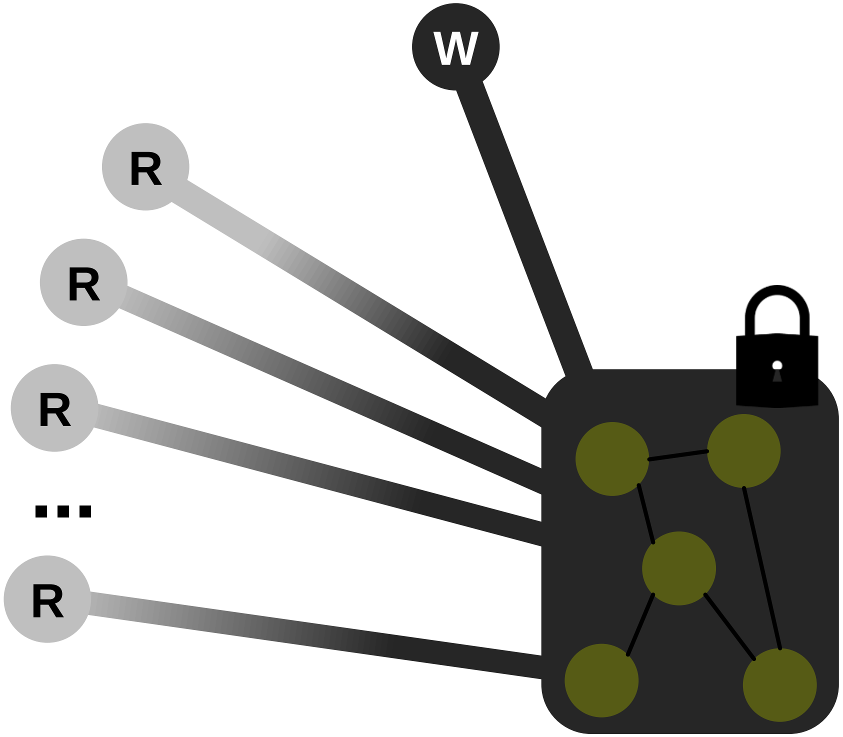
What will we use in the design?

WHAT WE WILL USE
MCS Locks



Mellor-Crummey and Scott: Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors, ACM TOCS'91

What we will use
Reader-Writer Locks

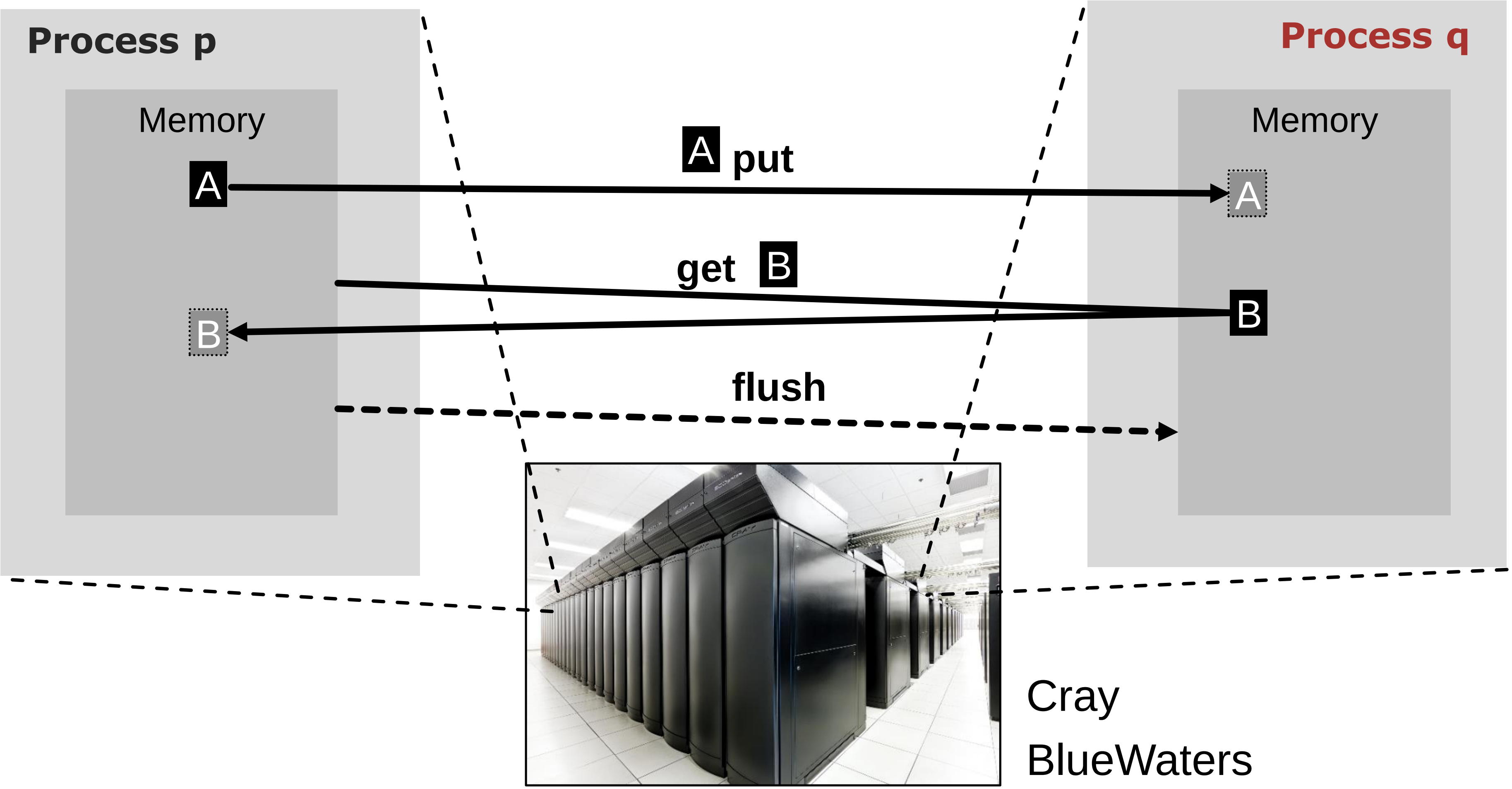


How to manage the design complexity?

How to ensure tunable performance?

What mechanism to use for efficient implementation?

Remote Memory Access (RMA) Programming

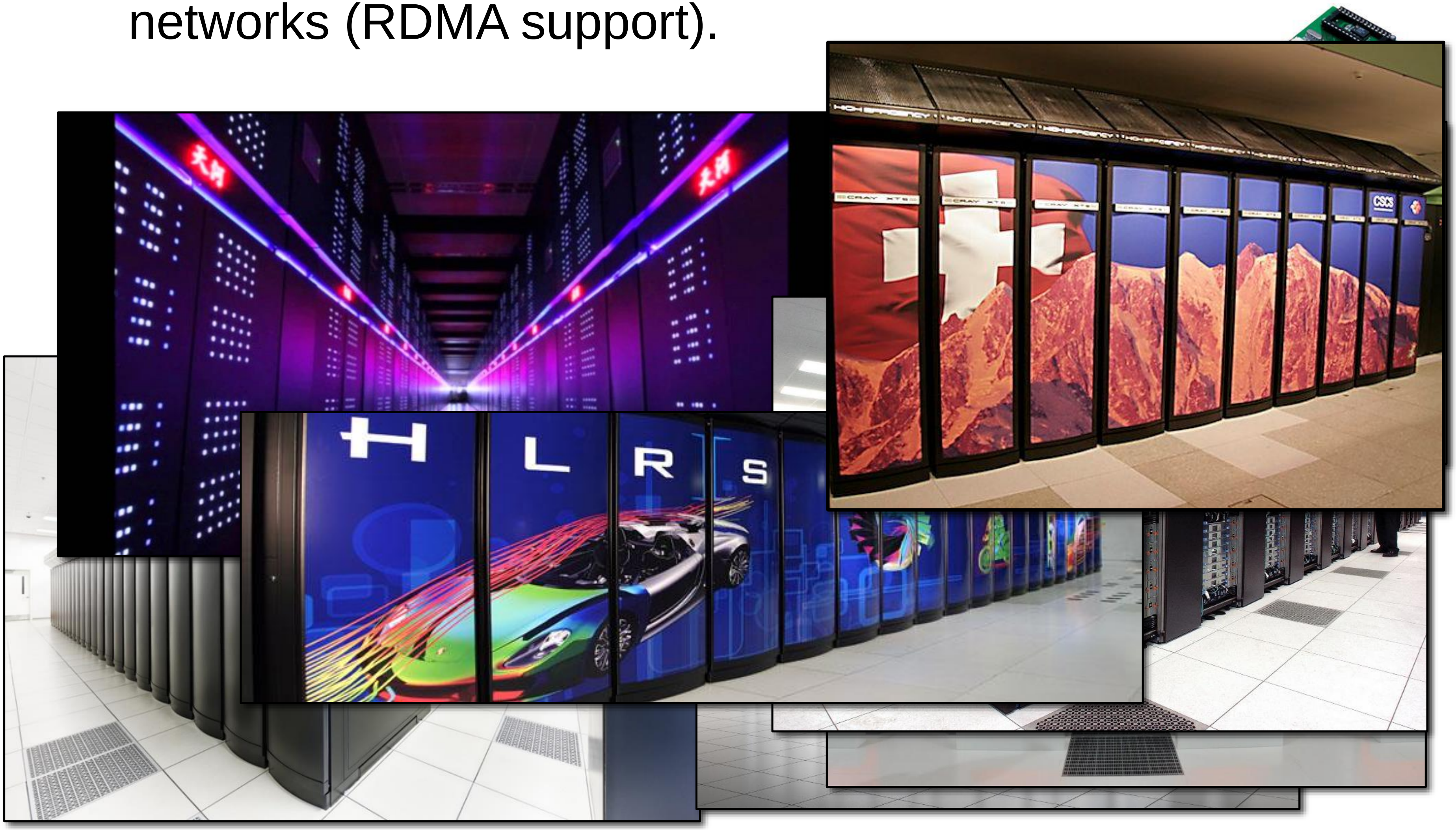


Cray BlueWaters

TH, J. Dinan, R. Thakur, B. Barrett, P. Balaji, W. Gropp, K. Underwood: Remote Memory Access Programming in MPI-3, ACM TOPC'15

REMOTE MEMORY ACCESS PROGRAMMING

- Implemented in hardware in NICs in the majority of HPC networks (RDMA support).



How to manage the design complexity?

How to ensure tunable performance?

What mechanism to use for efficient implementation?

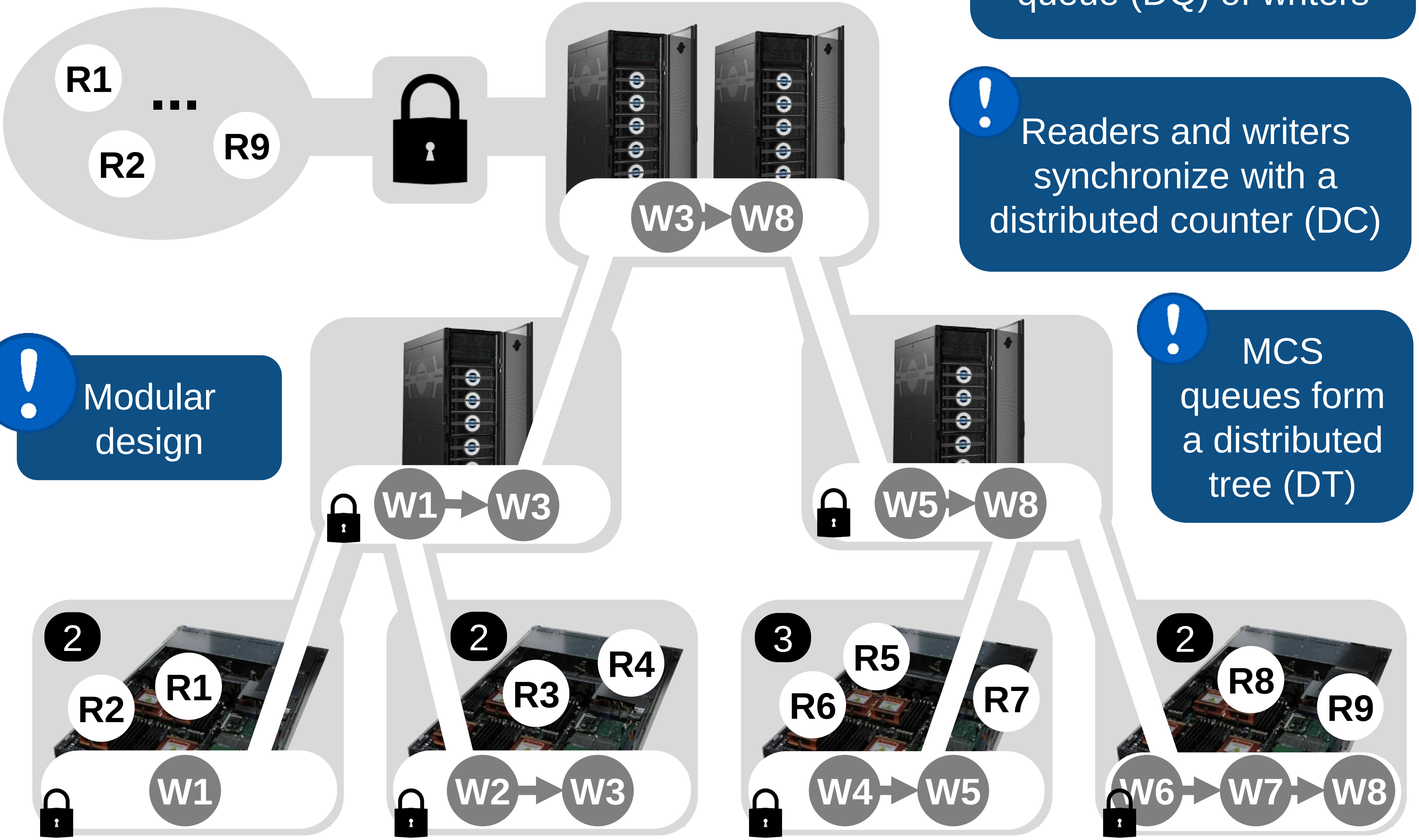
How to manage the design complexity?

Each element has its own distributed MCS queue (DQ) of writers

Readers and writers synchronize with a distributed counter (DC)

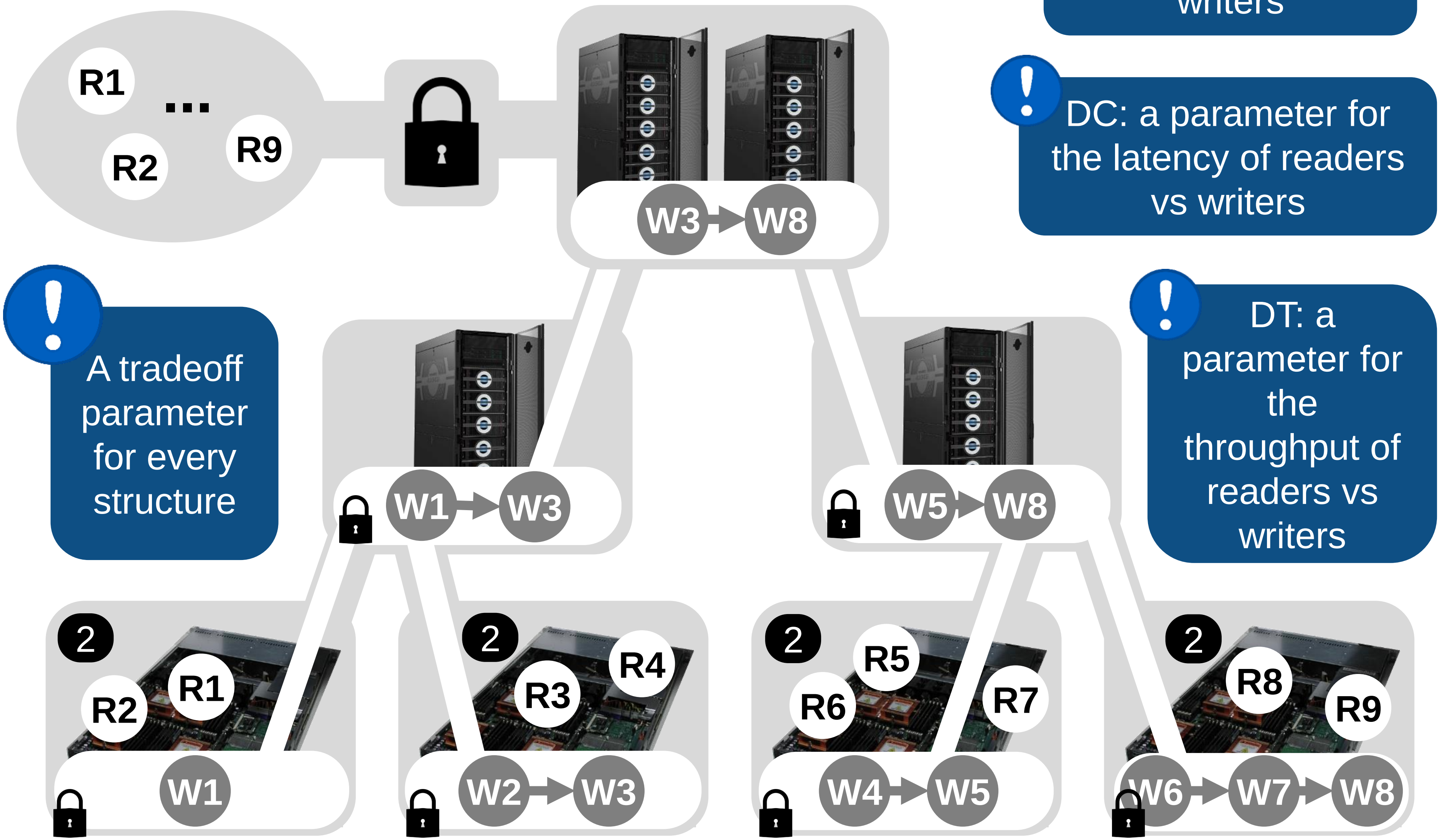
MCS queues form a distributed tree (DT)

Modular design



P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC'16, best paper

How to ensure tunable performance?



Each DQ: fairness vs throughput of writers

DC: a parameter for the latency of readers vs writers

A tradeoff parameter for every structure

DT: a parameter for the throughput of readers vs writers

P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC'16, best paper

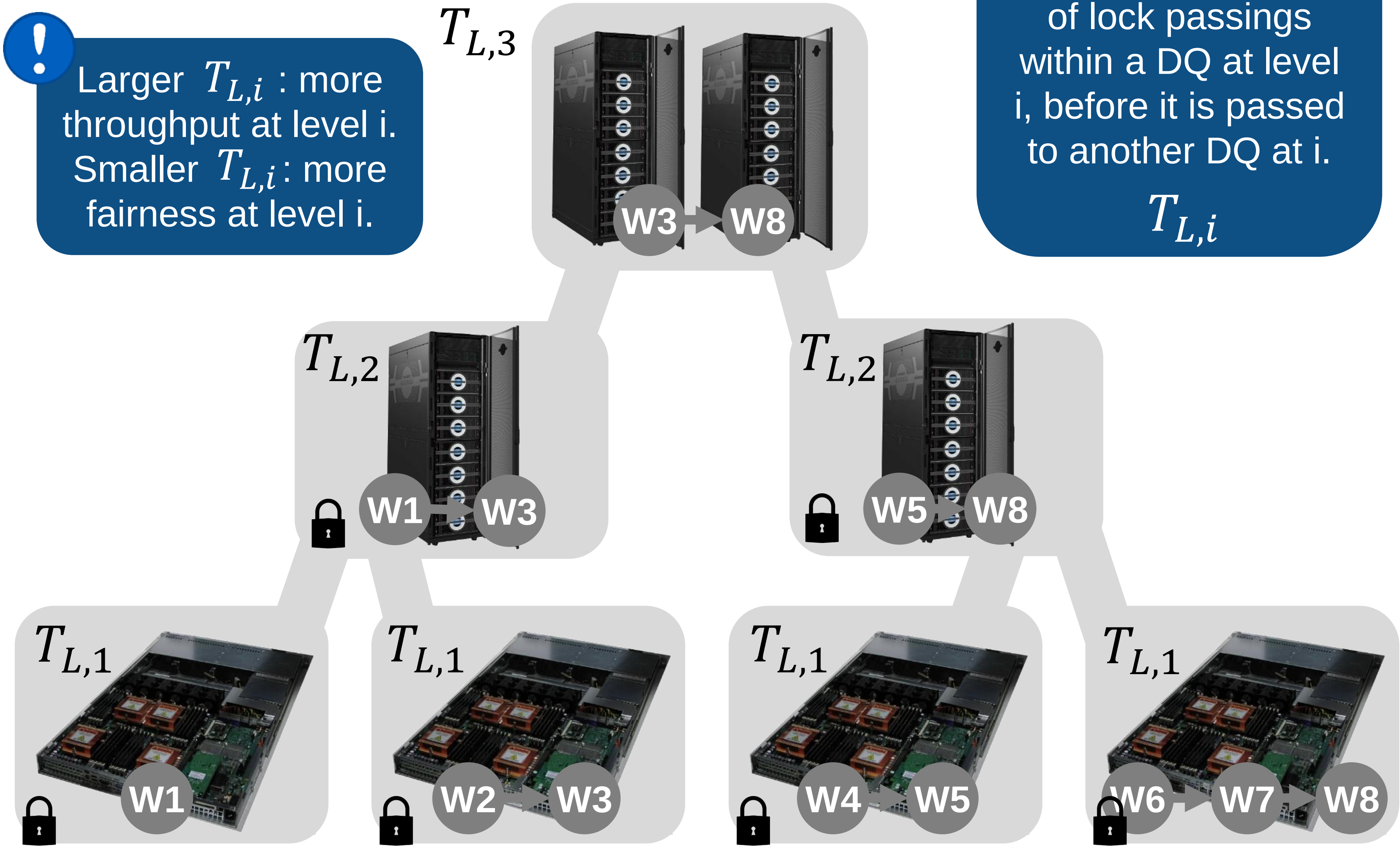
Distributed MCS Queues (DQs)

Throughput vs Fairness

Larger $T_{L,i}$: more throughput at level i. Smaller $T_{L,i}$: more fairness at level i.

Each DQ: The maximum number of lock passings within a DQ at level i, before it is passed to another DQ at i.

$T_{L,i}$

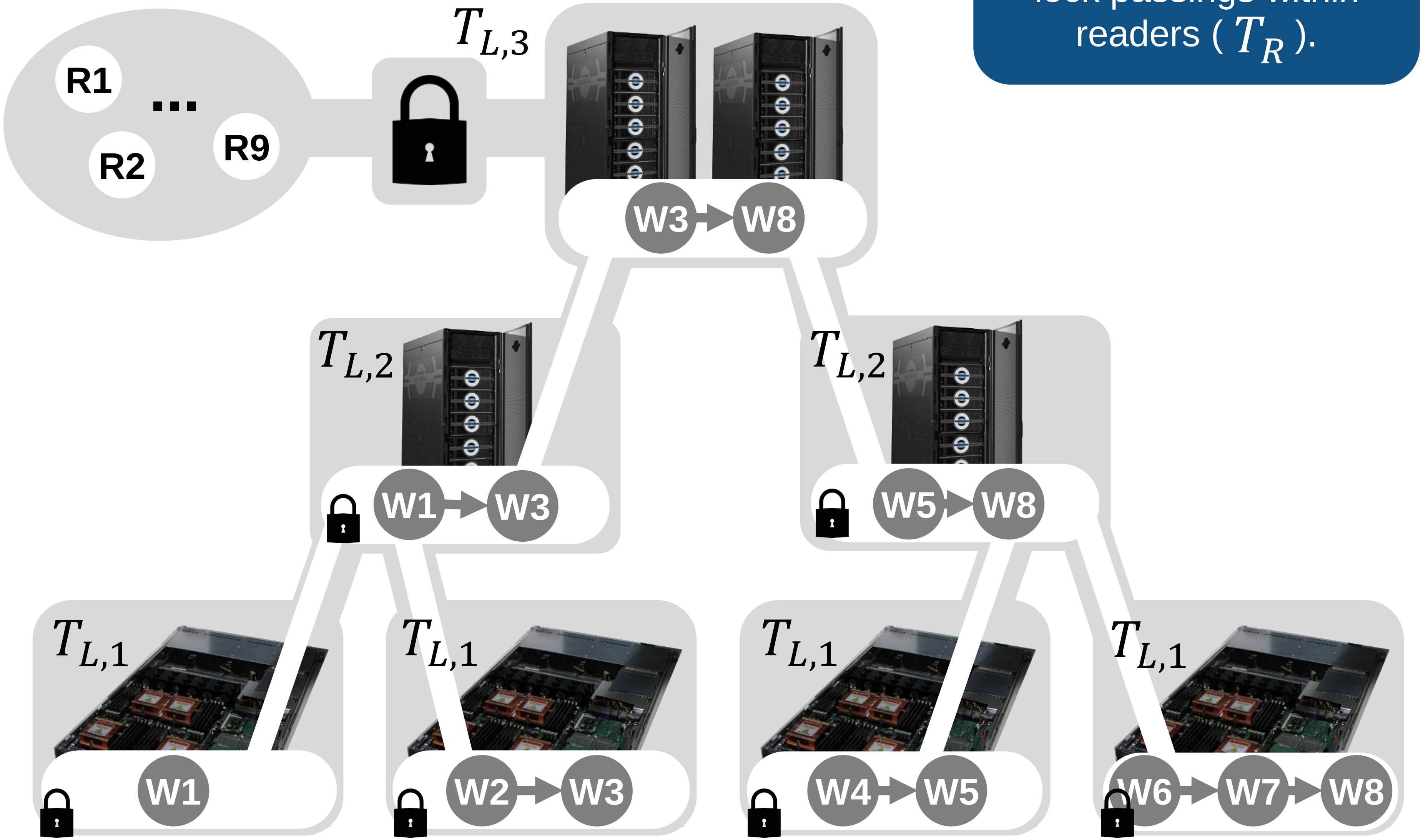


P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC'16, best paper

DISTRIBUTED TREE OF QUEUES (DT)

Throughput of readers vs writers

DT: The maximum number of consecutive lock passings within readers (T_R).



P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC'16, best paper

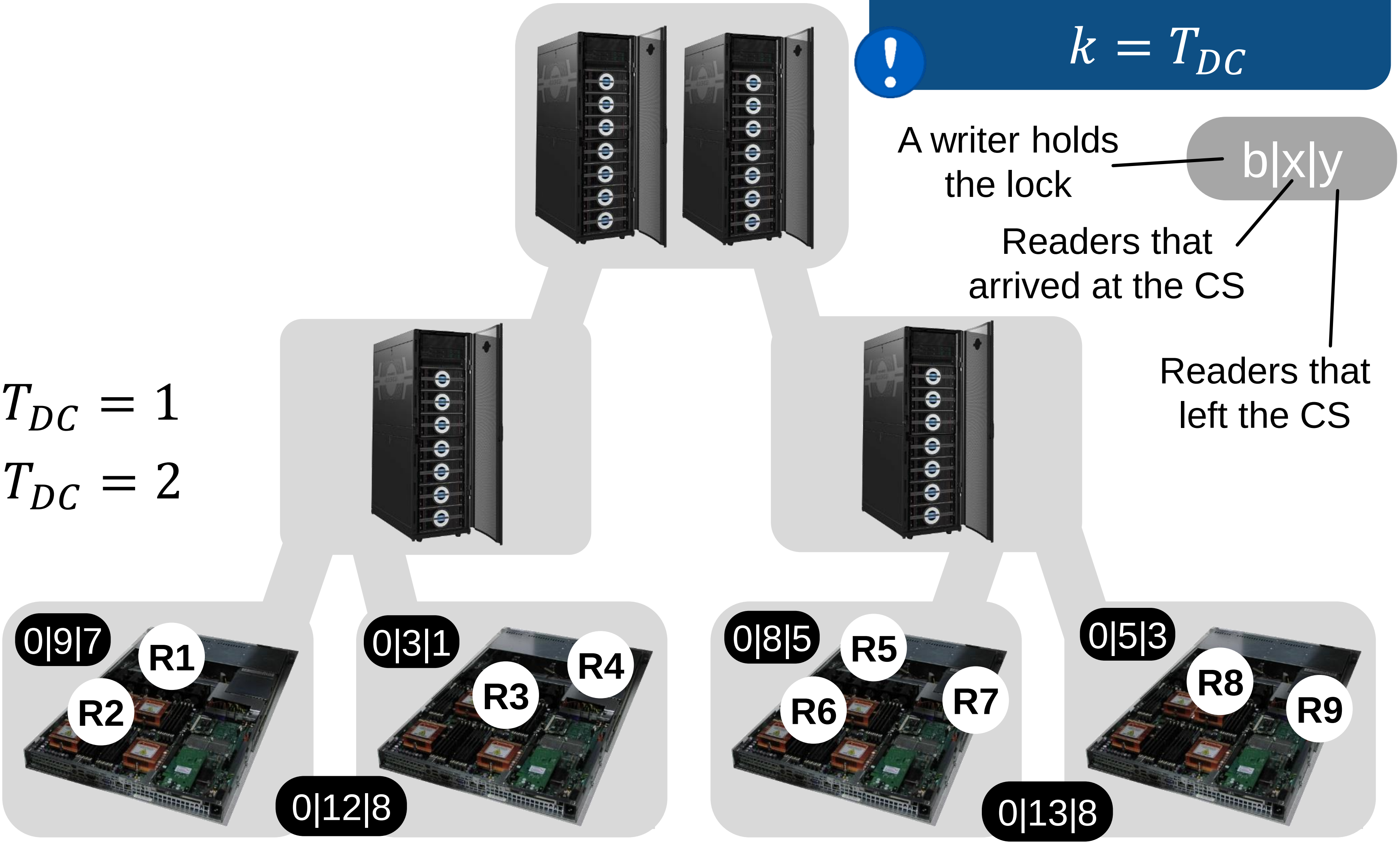
Distributed Counter (DC)
Latency of readers vs writers

DC: every *k*th compute node hosts a partial counter, all of which constitute the DC.

$$k = T_{DC}$$

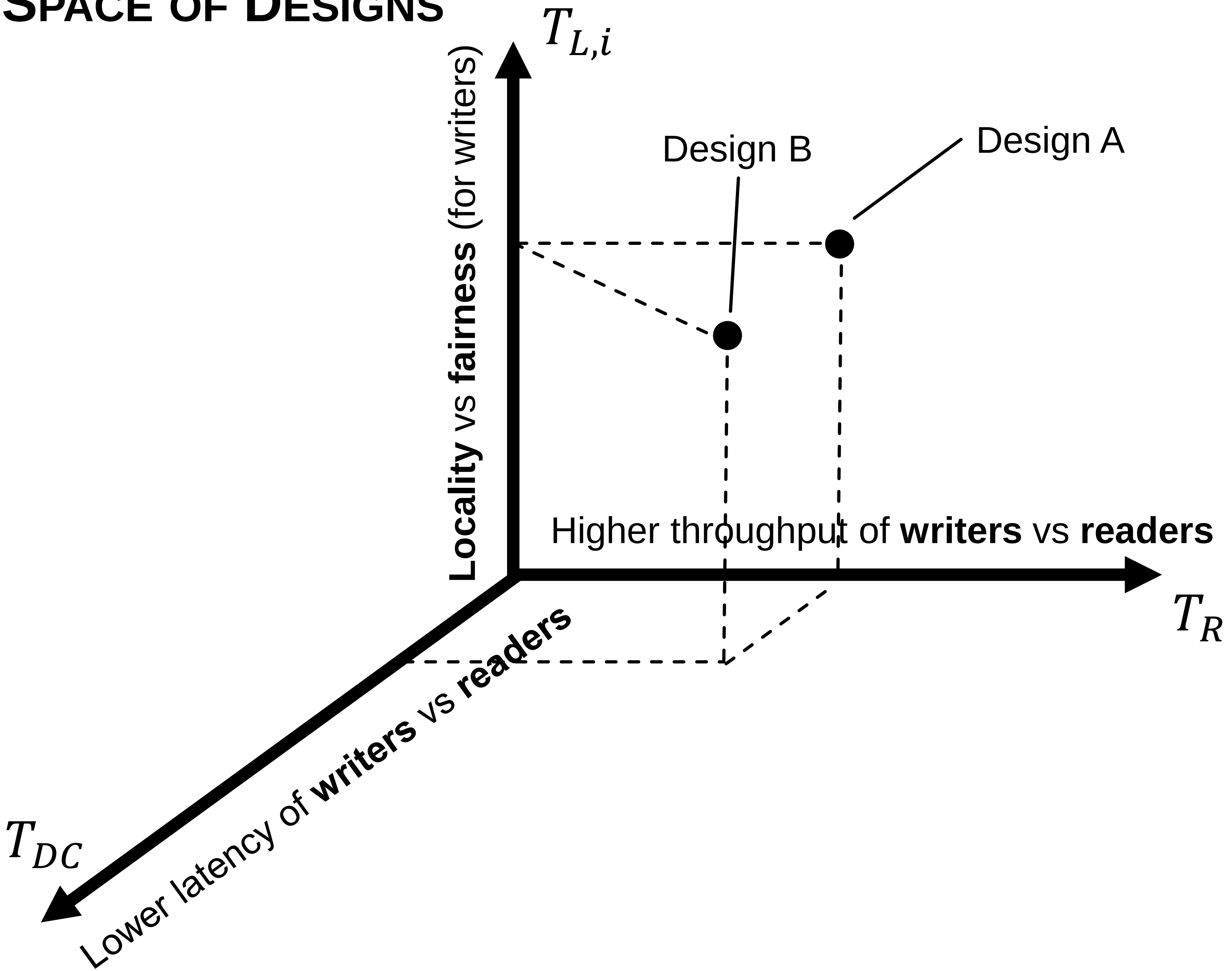
$T_{DC} = 1$

$T_{DC} = 2$



P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC'16, best paper

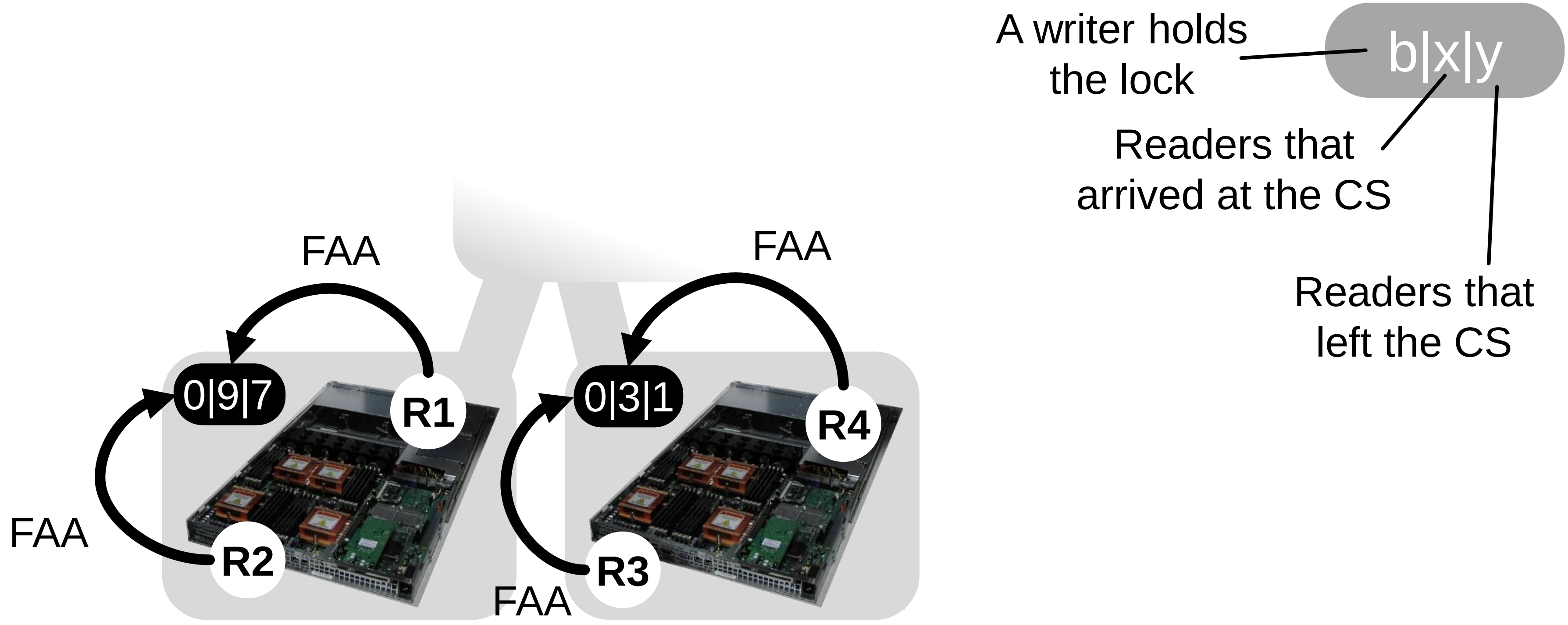
THE SPACE OF DESIGNS


$T_{L,i}$
Locality vs **fairness** (for writers)
Design B
Design A
Higher throughput of **writers** vs **readers**
T_R
T_{DC}
Lower latency of **writers** vs **readers**

P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC'16, best paper

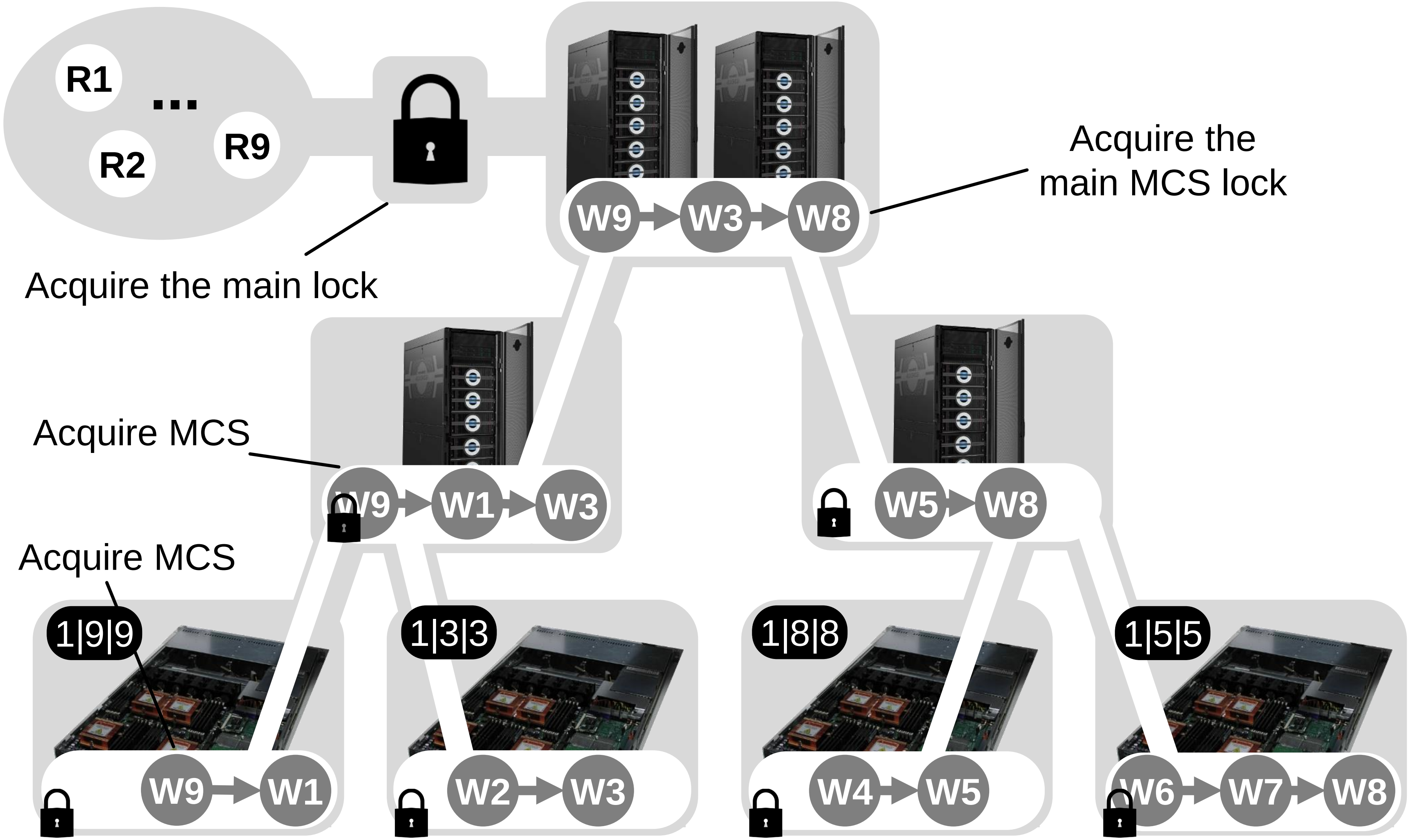
Lock Acquire by Readers

! A lightweight acquire protocol for readers: only one atomic fetch-and-add (FAA) operation



P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC’16, best paper

Lock Acquire by Writers



Acquire the main MCS lock

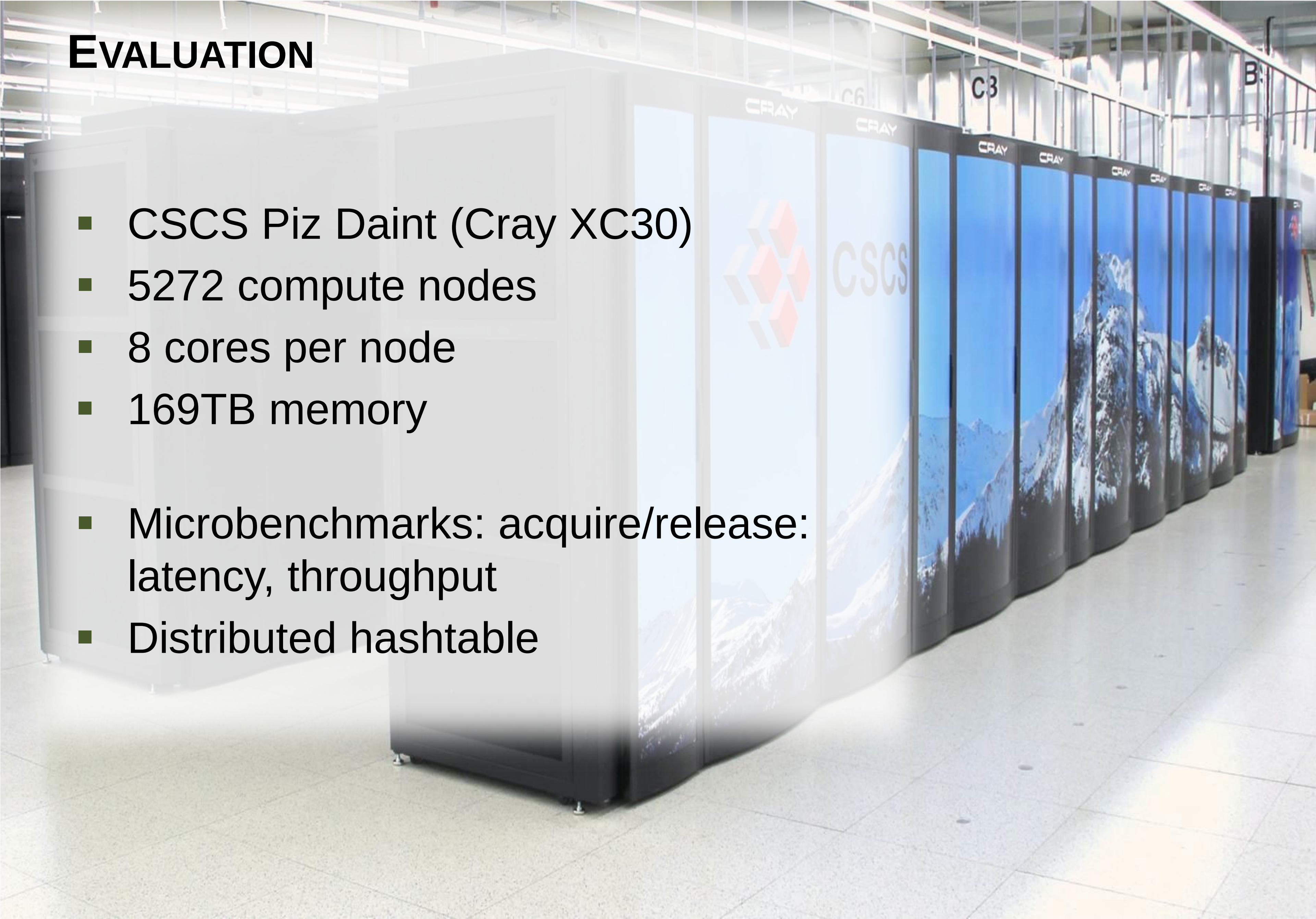
Acquire the main lock

Acquire MCS

Acquire MCS

P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC’16, best paper

Evaluation



- CSCS Piz Daint (Cray XC30)
- 5272 compute nodes
- 8 cores per node
- 169TB memory

- Microbenchmarks: acquire/release: latency, throughput
- Distributed hashtable

EVALUATION

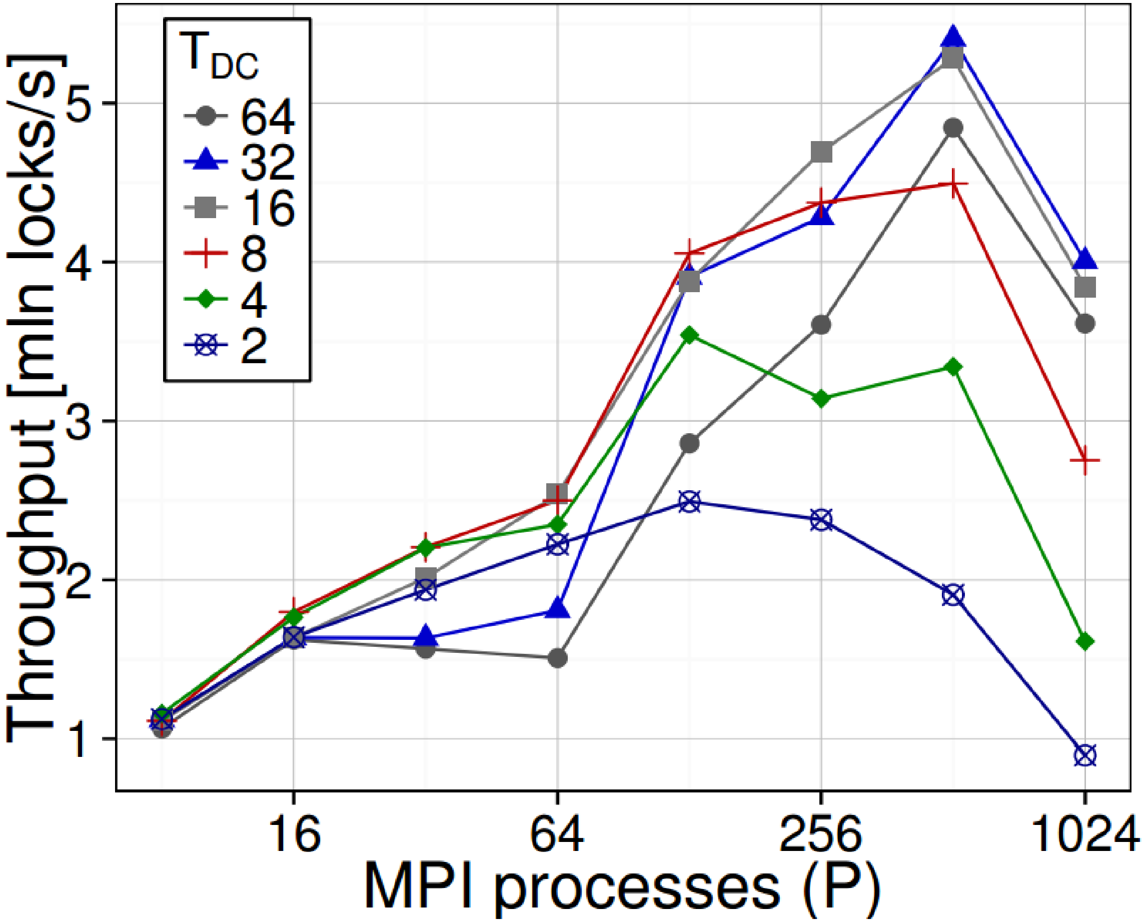
DISTRIBUTED COUNTER ANALYSIS

0|9|7 0|3|1

0|12|8

Throughput, 2% writers

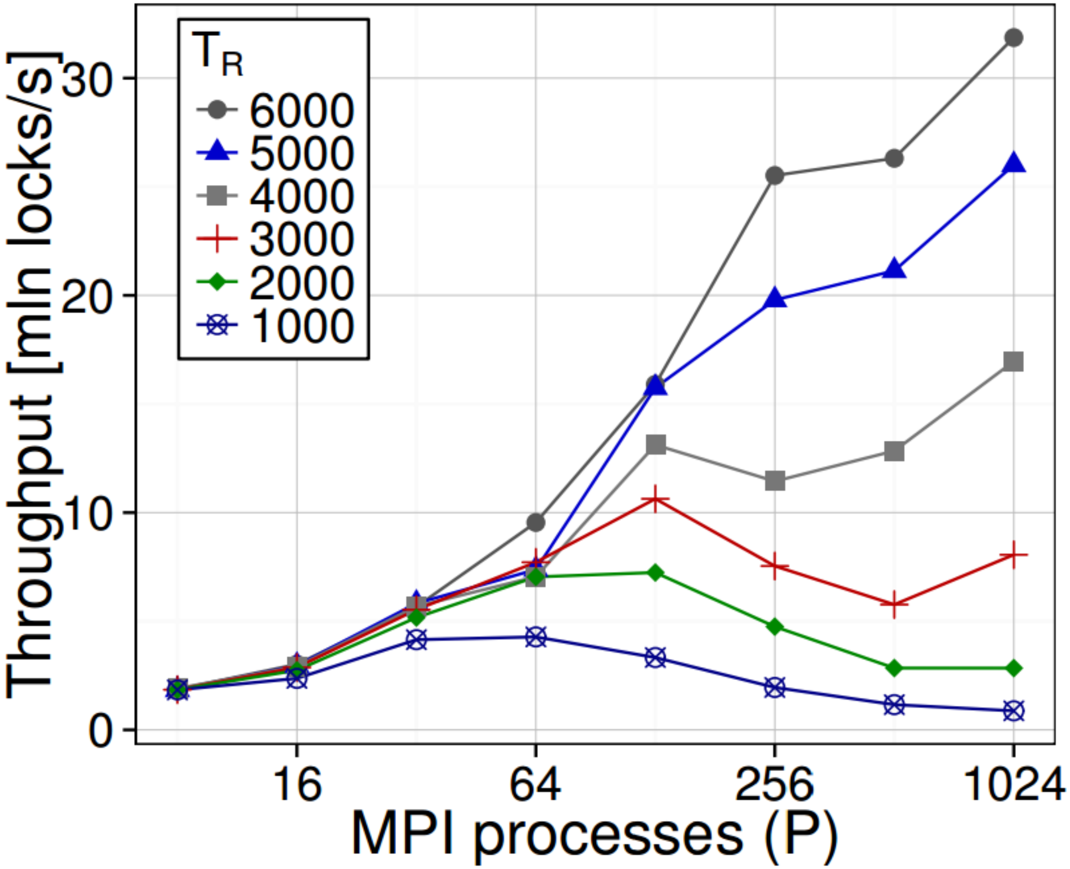
Single-operation benchmark



P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC’16, best paper

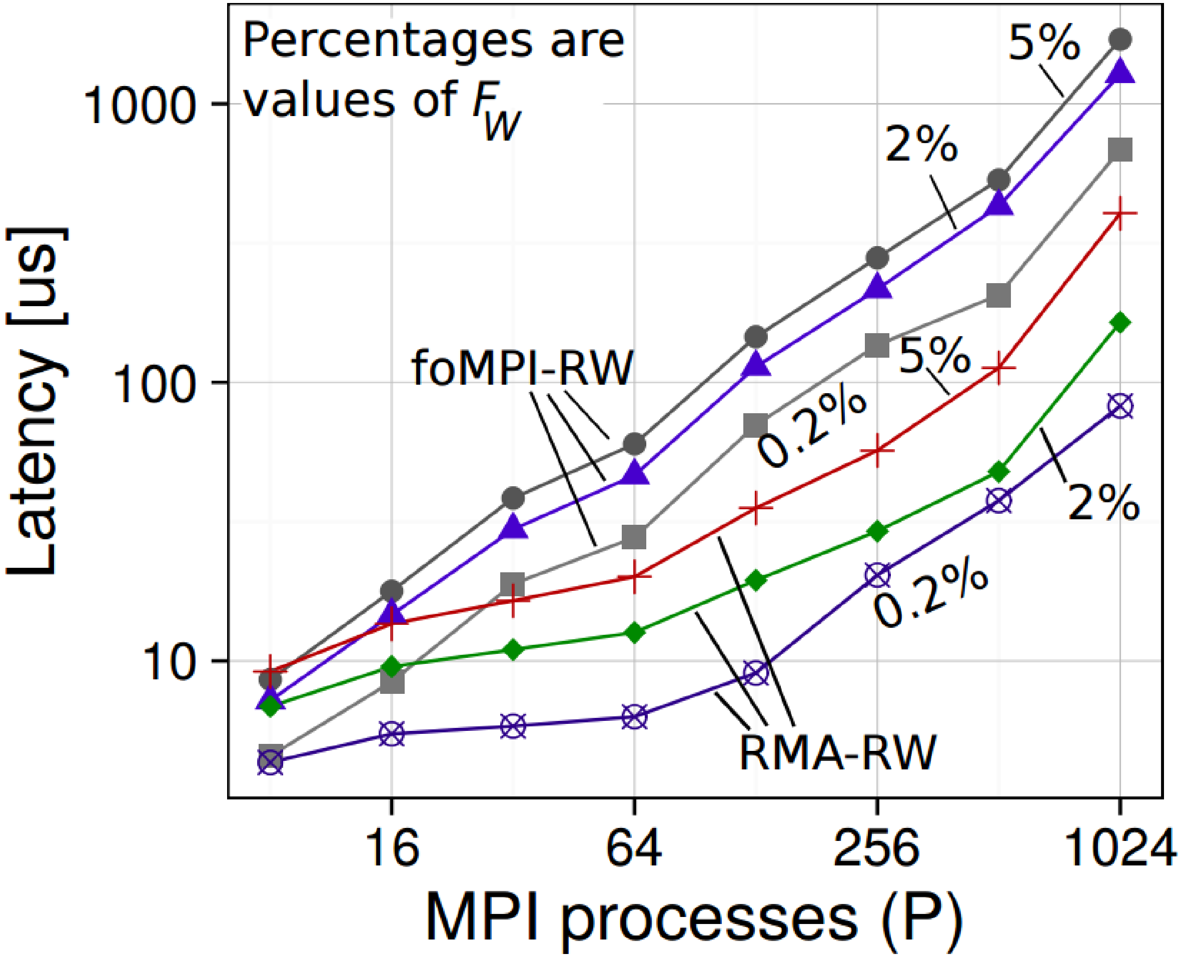
EVALUATION
READER THRESHOLD ANALYSIS

Throughput, 0.2% writers,
Empty-critical-section benchmark



P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC'16, best paper

EVALUATION
COMPARISON TO THE STATE-OF-THE-ART

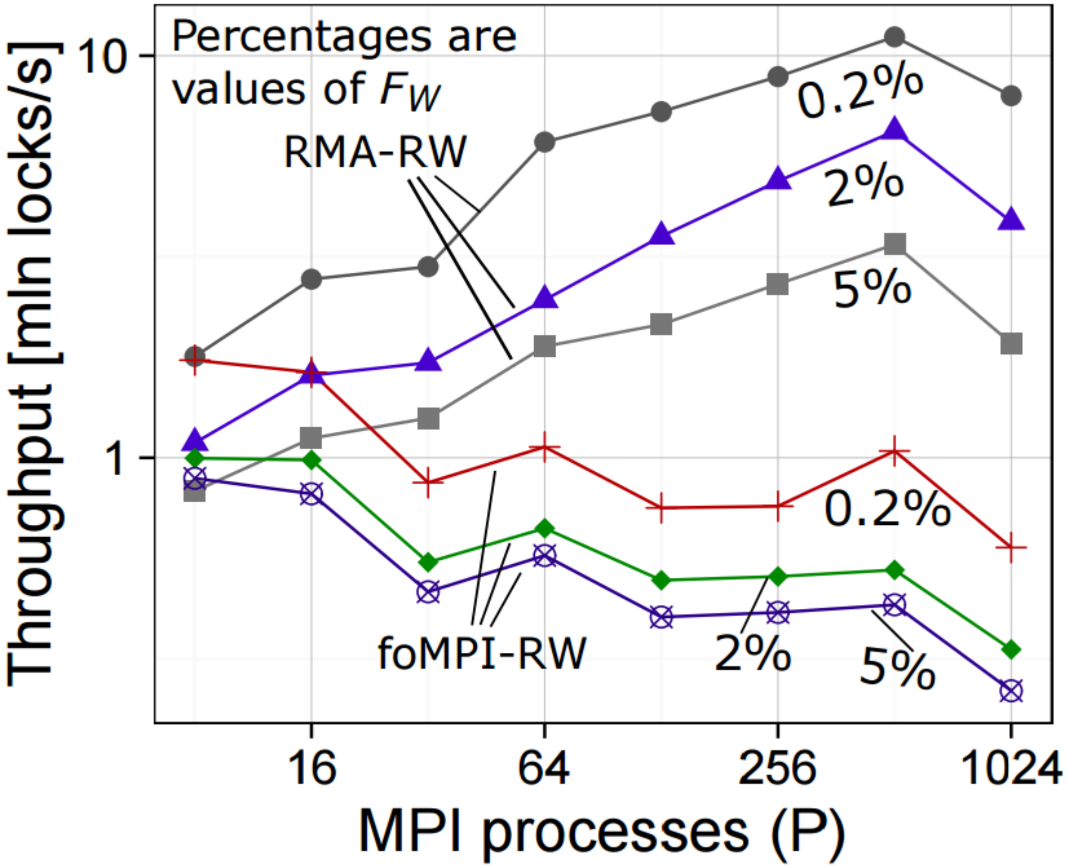


[1] R. Gerstenberger et al. Enabling Highly-scalable Remote Memory Access Programming with MPI-3 One Sided. ACM/IEEE Supercomputing 2013.

EVALUATION

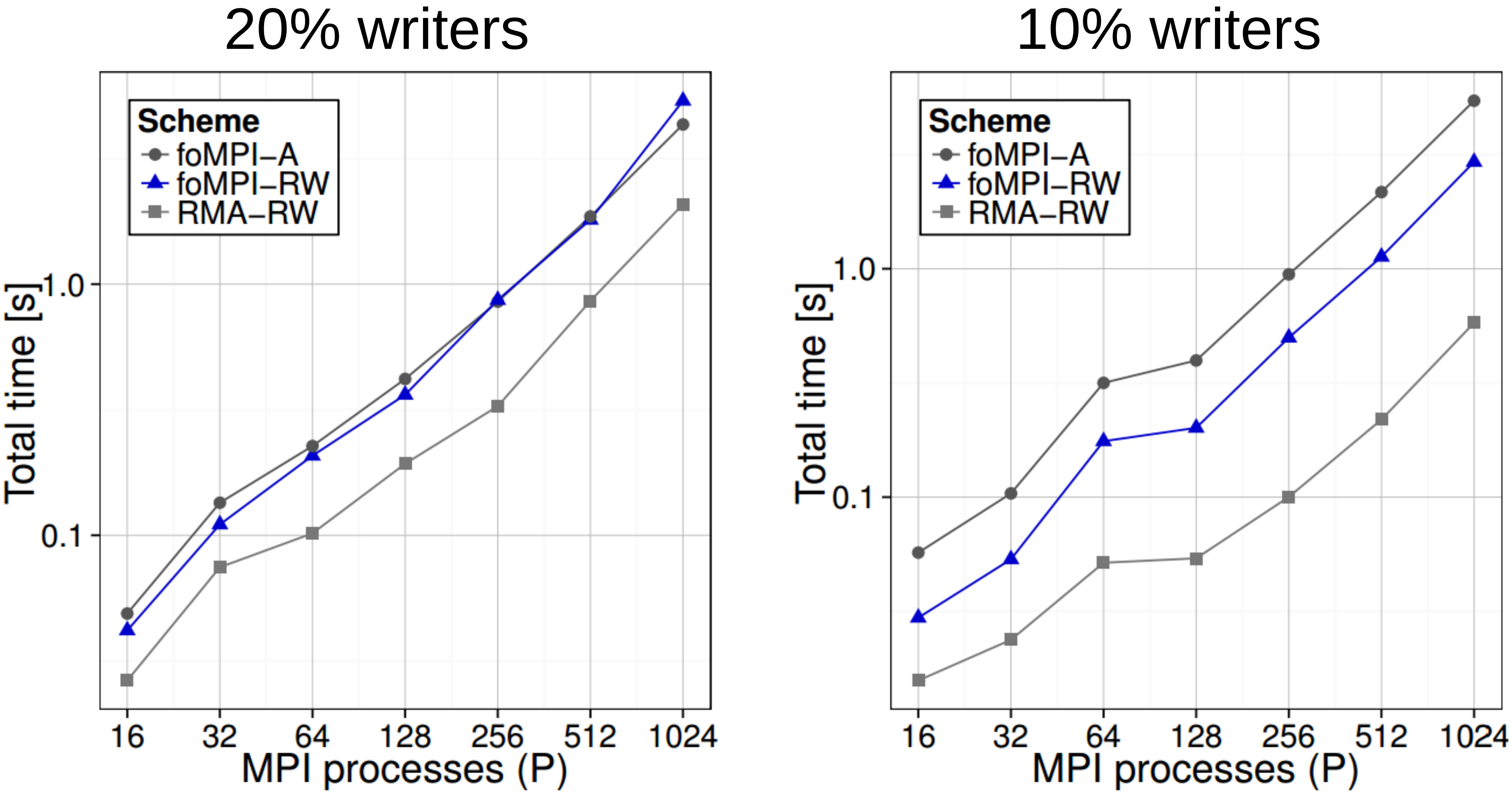
COMPARISON TO THE STATE-OF-THE-ART

Throughput, single-operation benchmark



[1] R. Gerstenberger et al. Enabling Highly-scalable Remote Memory Access Programming with MPI-3 One Sided. ACM/IEEE Supercomputing 2013.

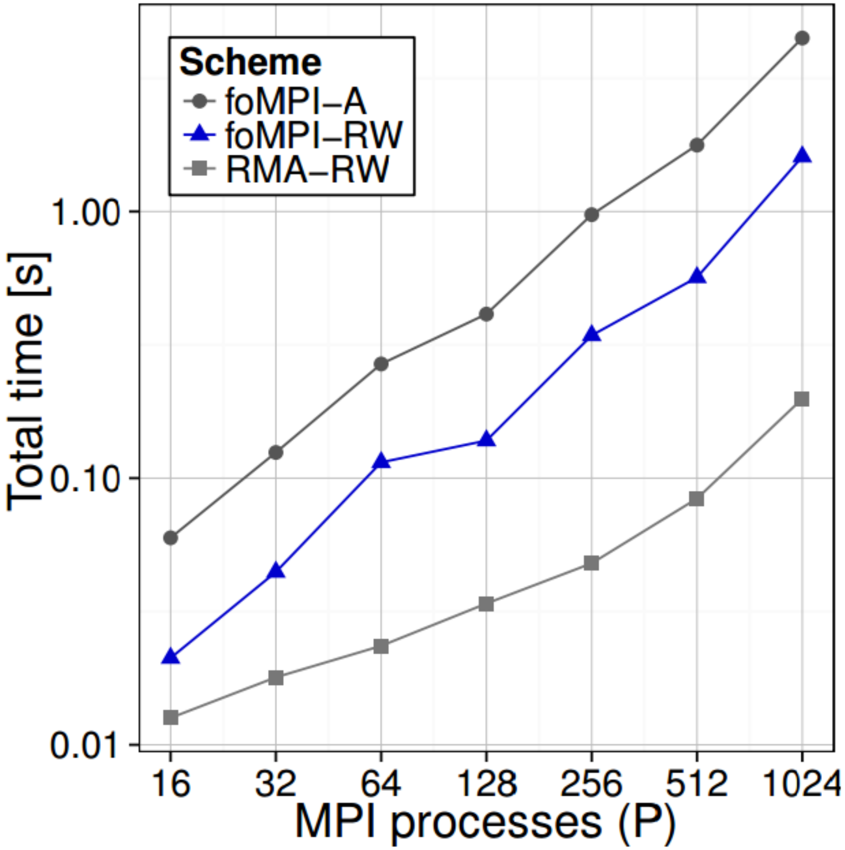
EVALUATION
DISTRIBUTED HASHTABLE



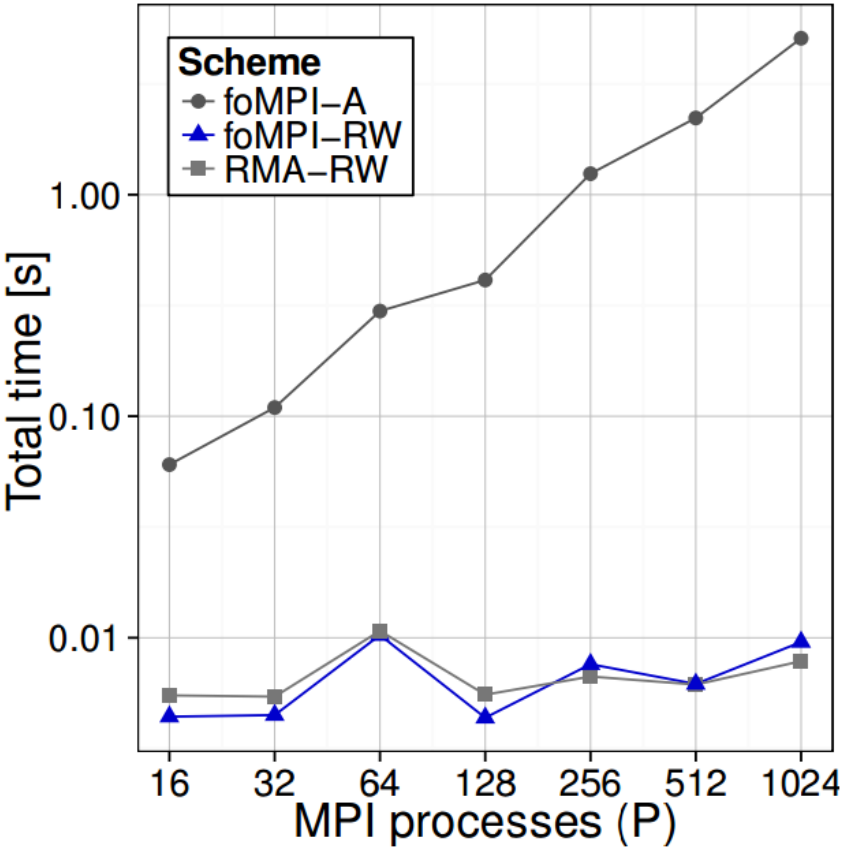
[1] R. Gerstenberger et al. Enabling Highly-scalable Remote Memory Access Programming with MPI-3 One Sided. ACM/IEEE Supercomputing 2013.

EVALUATION
DISTRIBUTED HASHTABLE

2% of writers



0% of writers

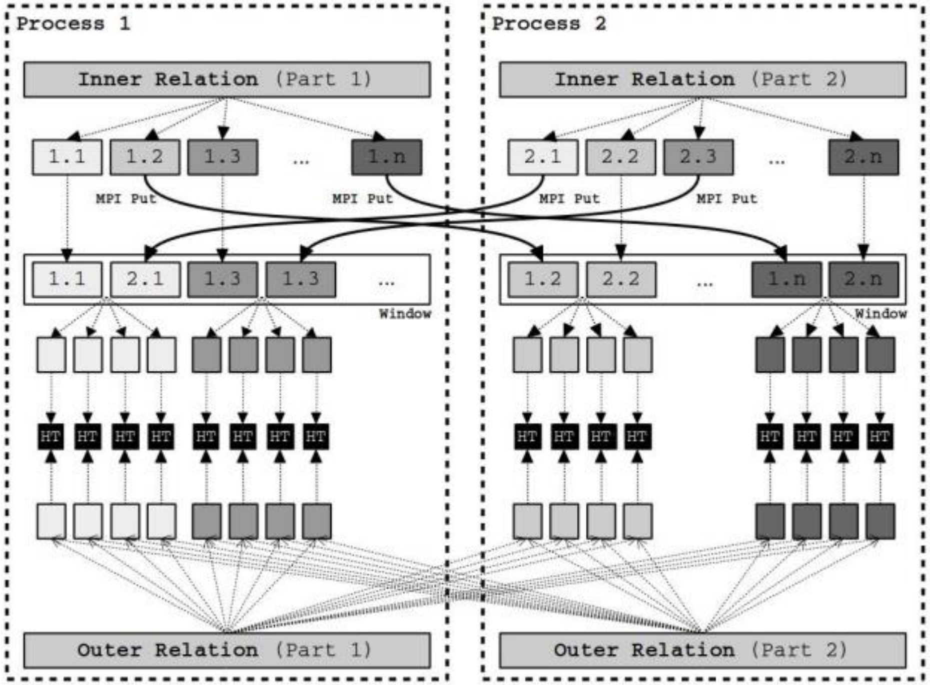


[1] R. Gerstenberger et al. Enabling Highly-scalable Remote Memory Access Programming with MPI-3 One Sided. ACM/IEEE Supercomputing 2013.

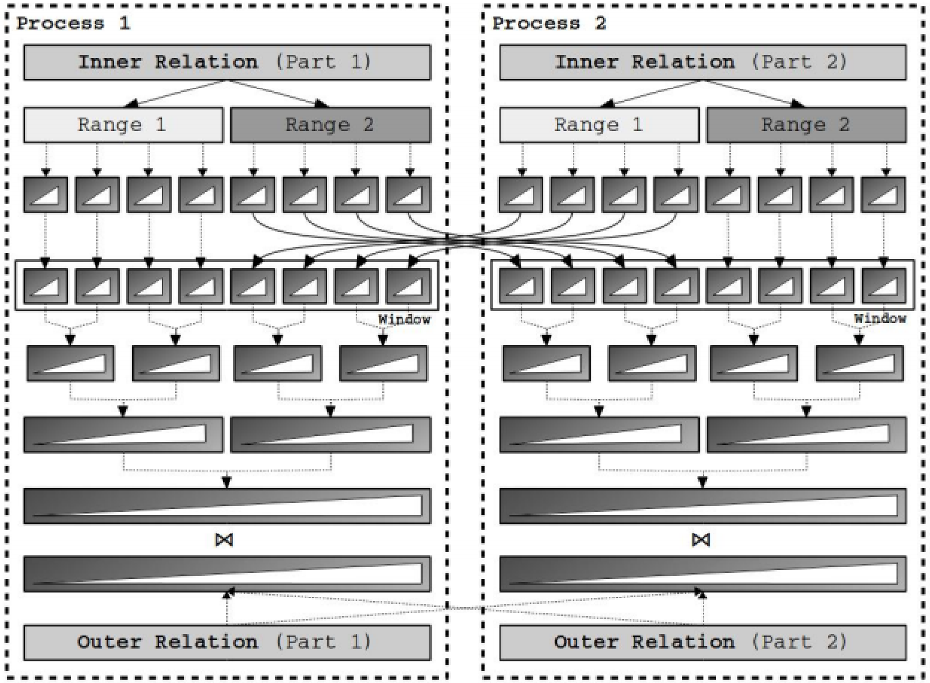
Another application area - Databases

- **MPI-RMA for distributed databases?**

Hash-Join



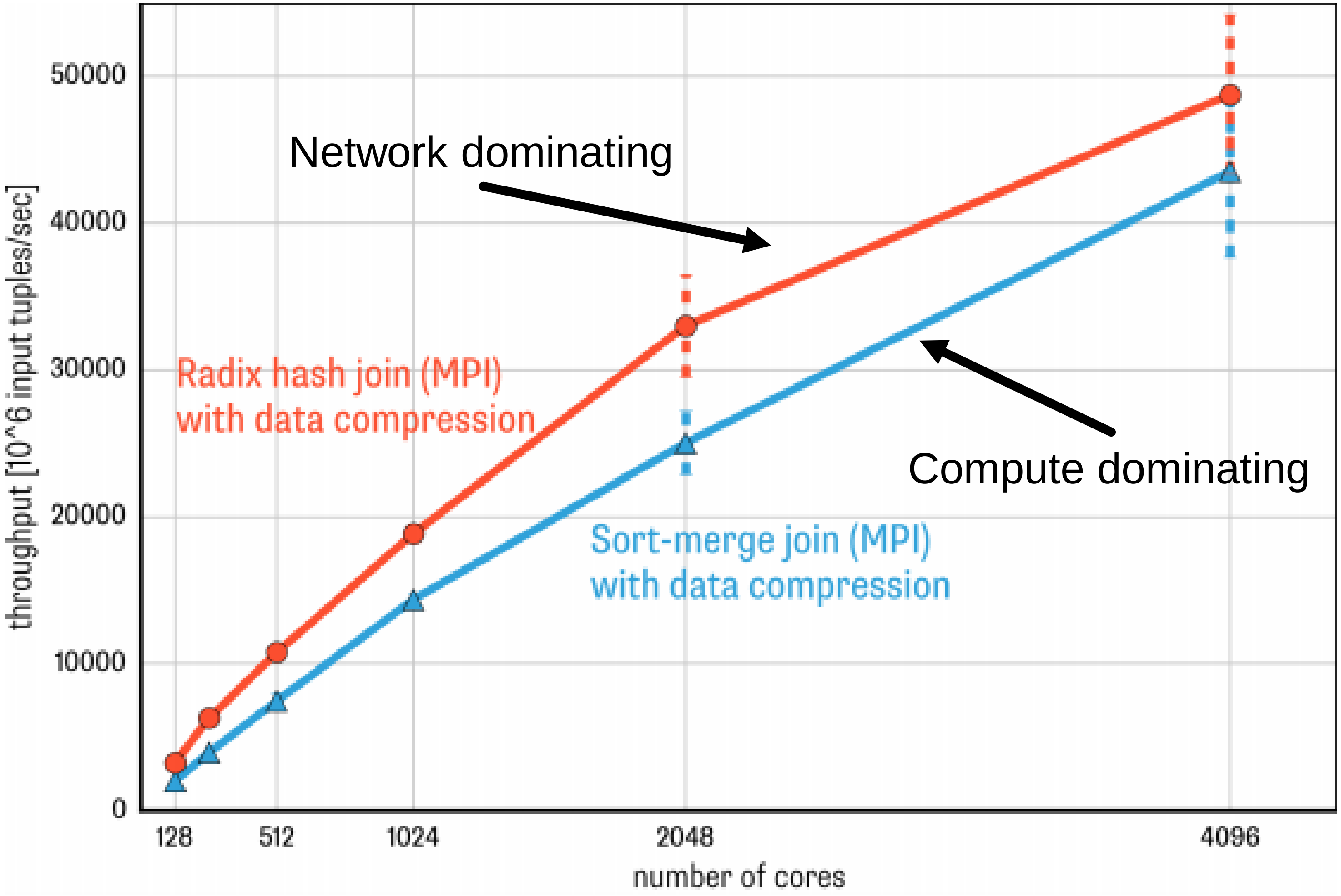
Sort-Join



C. Barthels, et al., TH: Distributed Join Algorithms on Thousands of Cores presented in Munich, Germany, VLDB Endowment, Aug. 2017

Another application area - Databases

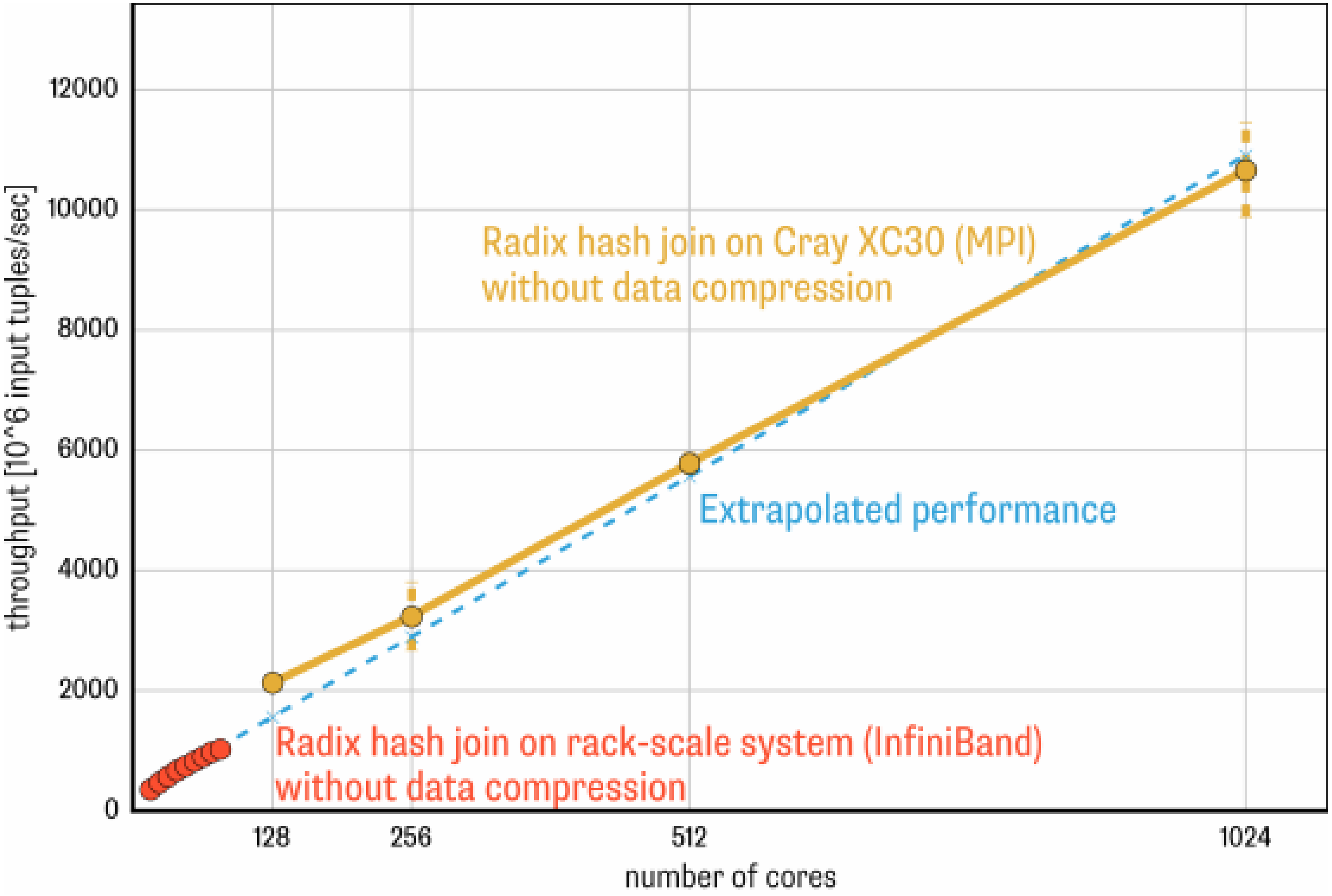
- **MPI-RMA for distributed databases on Piz Daint**



C. Barthels, et al., TH: Distributed Join Algorithms on Thousands of Cores presented in Munich, Germany, VLDB Endowment, Aug. 2017

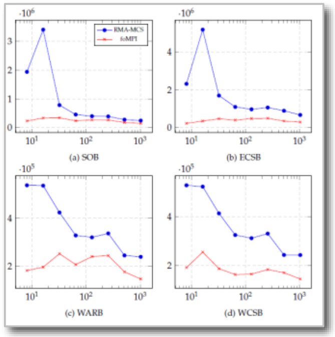
Another application area - Databases

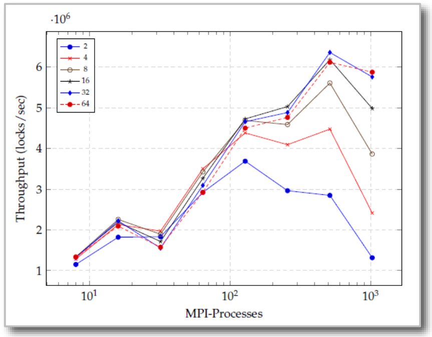
- **MPI-RMA for distributed databases on Piz Daint**

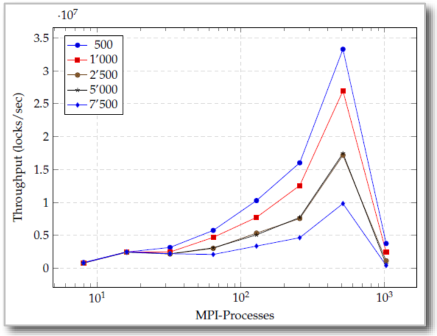


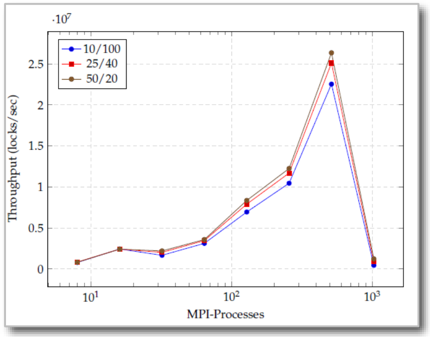
C. Barthels, et al., TH: Distributed Join Algorithms on Thousands of Cores presented in Munich, Germany, VLDB Endowment, Aug. 2017

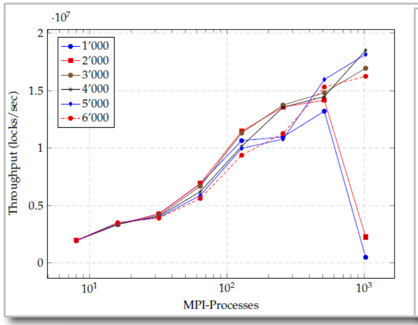
Other Analyses

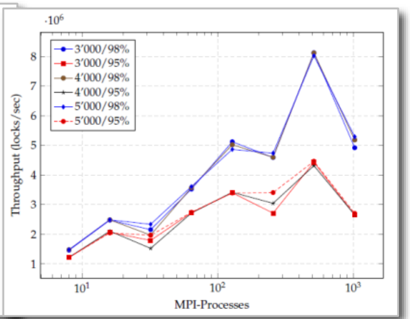


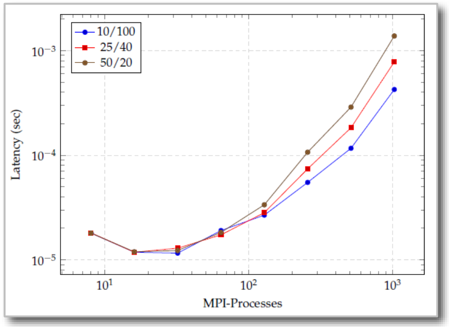


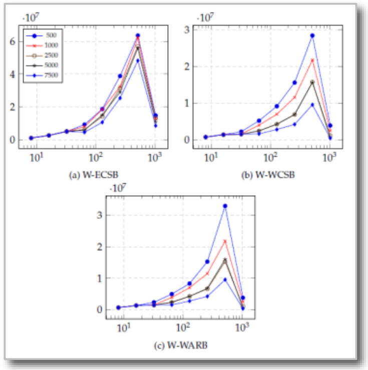


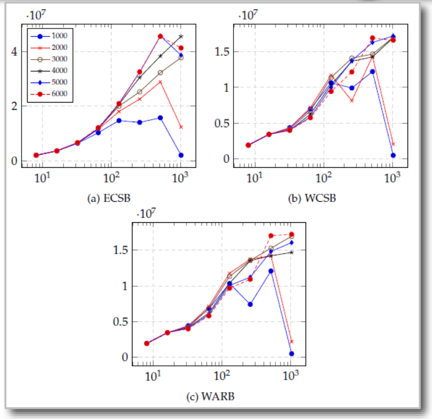


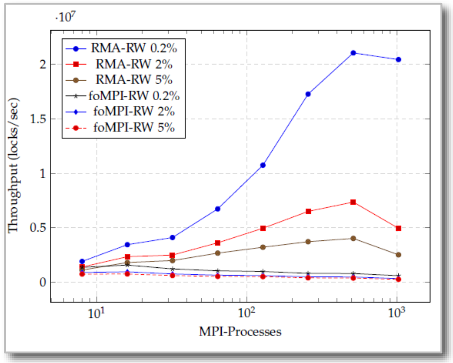


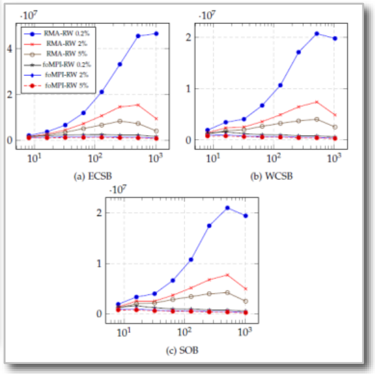


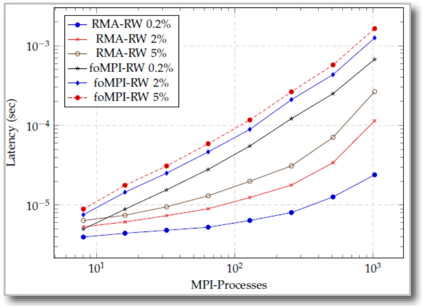






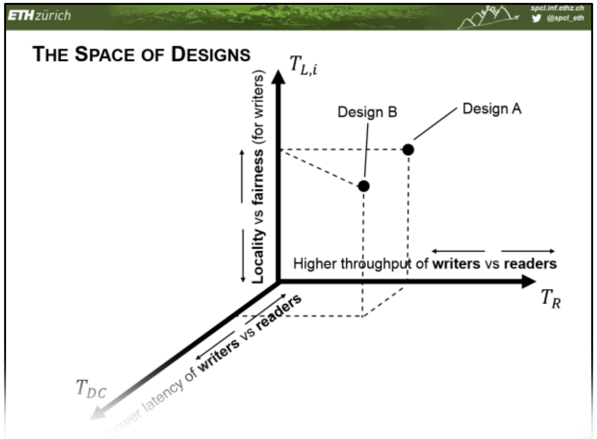






P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC'16, best paper

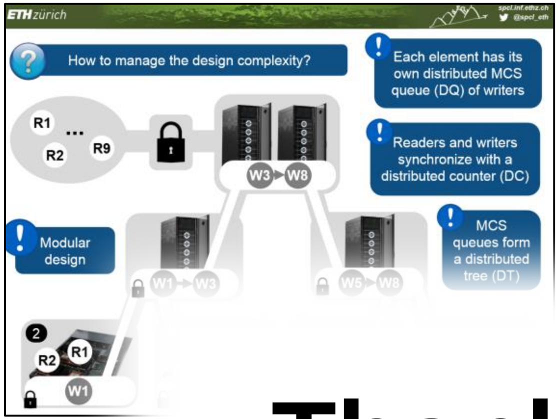
Conclusions



ETH zürich spcl.inf.ethz.ch @spcl_eth

THE SPACE OF DESIGNS

$T_{L,i}$
Locality vs fairness (for writers)
Design B
Design A
Higher throughput of writers vs readers
T_R
T_{DC}
Lower latency of writers vs readers



ETH zürich spcl.inf.ethz.ch @spcl_eth

How to manage the design complexity?

Each element has its own distributed MCS queue (DQ) of writers

Readers and writers synchronize with a distributed counter (DC)

MCS queues form a distributed tree (DT)

Modular design

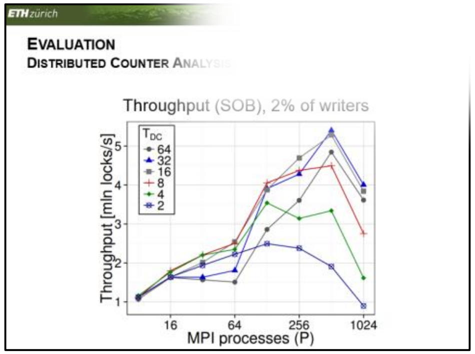
ible

es

Modular d

corre

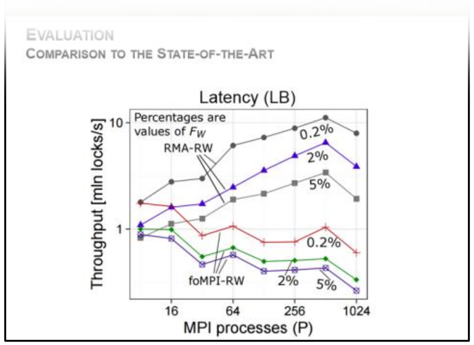
Thank you for your attention



ETH zürich

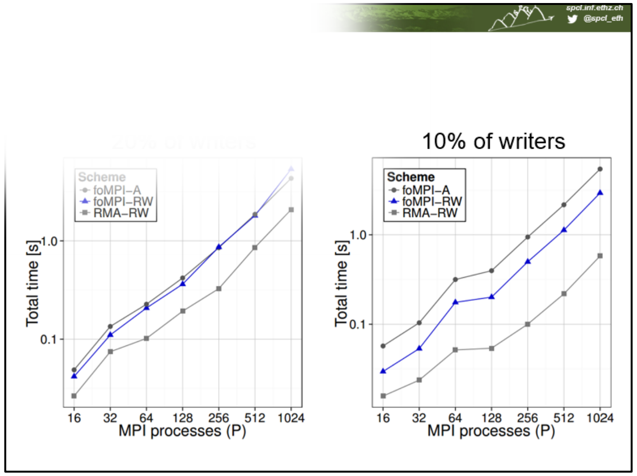
EVALUATION
DISTRIBUTED COUNTER ANALYSIS

Throughput (SOB), 2% of writers



EVALUATION
COMPARISON TO THE STATE-OF-THE-ART

Latency (LB)



spcl.inf.ethz.ch @spcl_eth

10% of writers

Improves latency and throughput over state-of-the-art

Accelerates distributed hashtabled

P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC'16, best paper