High-Performance Distributed RMA Locks

Patrick Schmid, Maciej Besta, Torsten Hoefler
NEED FOR EFFICIENT LARGE-SCALE SYNCHRONIZATION
Locks

- Inuitive semantics
- Various performance penalties

An example structure:

- Proc p
- Proc q

Activities:
- lock
- unlock
- accesses
- ...

L

LOCKS
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
We need to distinguish between readers and writers

We need flexible performance for both types of processes

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

Process p

Memory

A

B

Process q

Memory

A

B

A put

get B

flush

Cray BlueWaters

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

DT: a parameter for the throughput of readers vs writers

A tradeoff parameter for every structure

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 $i$. $T_{L,i}$

$T_{L,1}$

$T_{L,2}$

$T_{L,3}$

W1

W1

W2

W2

W3

W3

W4

W4

W5

W5

W6

W6

W7

W7

W8

W8

W8

Throughput vs Fairness

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 ($T_R$).
**DISTRIBUTED COUNTER (DC)**

Latency of readers vs writers

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

\[ k = T_{DC} \]

A writer holds the lock \( b|x|y \)

Readers that arrived at the CS

Readers that left the CS

\[ T_{DC} = 1 \]

\[ T_{DC} = 2 \]

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)

Higher throughput of writers vs readers

Design A

Design B

$T_{DC}$

Lower latency of writers vs readers

$T_{R}$

P. Schmid, M. Besta, TH: High-Performance Distributed RMA Locks, ACM HPDC’16, best paper
A lightweight acquire protocol for readers: only one atomic fetch-and-add (FAA) operation

A writer holds the lock
Readers that arrived at the CS
Readers that left the CS

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

Acquire the main lock

Acquire MCS

Acquire the main MCS lock

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

Throughput, 2% writers

Single-operation benchmark

Throughput [mln locks/s] vs. MPI processes (P)

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

EVALUATION
COMPARISON TO THE STATE-OF-THE-ART

Throughput, single-operation benchmark

Percentages are values of $F_W$

EVALUATION
DISTRIBUTED HASHTABLE

20% writers

10% writers

EVALUATION
DISTRIBUTED HASHTABLE

2% of writers

0% of writers

Another application area - Databases

- MPI-RMA for distributed databases?

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

Modular distributed RMA lock, correctness with SPIN. Parameter-based design, feasible with various RMA libs/languages.

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

Accelerates distributed hashtabled.

Thank you for your attention.

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