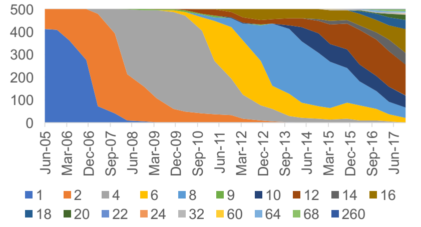
Fast and generic concurrent message-passing

Hoang-Vu Dang, *Advisor: Prof. Marc Snir*

Department of Computer Science, College of Engineering, University of Illinois at Urbana-Champaign

MOTIVATIONS

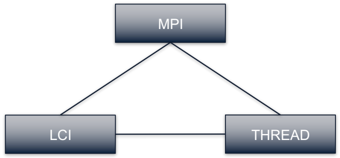
- Clusters and supercomputers have increasing core numbers and are more heterogeneous
- Explicit data movement becomes more important to performance
- There is growing interest in high-performance for non-traditional scientific applications: machine-learning, data/graph analytics
- Message-Passing Interface (MPI) is being used, but the performance is not ideal – especially with high thread counts



The number of machine with higher number of cores/socket increases each year in the TOP 500 supercomputers list

CONTRIBUTIONS

- Study and evaluation of MPI semantics and performance for emerging applications and architectures
- Design and Implementation of LCI, a low-level and efficient communication interface targeting multi-threaded, event-driven, heterogeneous frameworks
- Development of new thread synchronization and scheduling techniques for efficient inter-operation between threads and communication runtimes



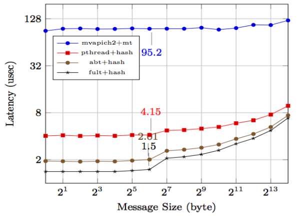
MPI performance and analysis [EuroMPI'16 best-paper, CCGrid'17]

Case study and implementation with MPICH 3.1 performance with threads:

- MPI_THREAD_MUTLIPLE performs poorly with high thread contention
- Cooperative scheduling techniques improve latency by 3x
- Advanced lock with unbounded-bias improves message rate by 4x
- Implementations are being incorporated into MPICH [mpich/pull/3068]

Design and implementation of message-passing point-to-point:

- MPI relaxation of wildcard matching
- Efficient low-contention tag-matching using hash-table
- Dedicated communication server minimizes data movement
- User-Level tasking minimizes thread synchronizations



[EuroMPI'16] **Hoang-Vu Dang**, Marc Snir, and William Gropp. **"Towards millions of communicating threads."**
[CCGrid'17] **Hoang-Vu Dang**, Sangmin Seo, Abdelhalim Amer, and Pavan Balaji.**"Advanced Thread Synchronization for Multithreaded MPI Implementations.**

LCI: generic and low-overhead communication interface [IPDPS'18, PLDI'18]

LCI design principles are to decouple:

- producer-consumer matching: tag, un-tag, one-sided, two-sided
- completion events and progress: completion queue, completion signal
- fatal-error and recoverable errors: retry when recoverable
- high-level, low-level features: maintains simple network facing primitives

	bfs	cc	pagerank	sssp
LCI	**1.17**	**2.41**	**89.72**	**2.46**
IntelMPI-Probe	1.41	2.95	174.67	2.94
MVAPICH2-Probe	1.40	2.93	177.72	2.82
OpenMPI-Probe	1.33	2.99	171.57	2.82
IntelMPI-RMA (+1.4)	**1.06**	2.36	**87.84**	**1.93**
MVAPICH2-RMA (+1.8)	1.14	**2.29**	93.53	2.13
OpenMPI-RMA (+1.2)	1.21	2.34	93.74	2.25

LCI improves the state-of-the-art performance for graph frameworks

- D-Galois: deals with issues with flow-control and data management
- Gluon: deals with issues with heterogeneity in computing architecture

[IPDPS'18] **Hoang-Vu Dang**, Roshan Dathathri, Gurbinder Gill, Alex Brooks, Nikoli Dryden, Andrew Lenharth, Loc Hoang, Keshav Pingali, and Marc Snir. **"A lightweight communication runtime for distributed graph analytics."**
[PLDI'18] Roshan Dathathri, Gurbinder Gill, Loc Hoang, **Hoang-Vu Dang**, Alex Brooks, Nikoli Dryden, Marc Snir, and Keshav Pingali, **"Gluon: A communication optimizing framework for distributed heterogeneous graph analytics"**

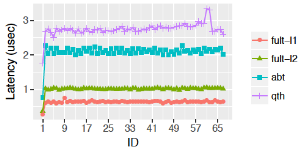
FULT/PPL: Fast synchronizations for communication [ICPP'18, ESPM2'15]

Schedule/de-scheduling tasks quickly is needed for distributed events:

- Communication server receives messages and signals waiting threads
- Signal/wait performance is critical for the performance of communication with large number of threads.

FULT is a Fast User-Level Threading scheduling technique:

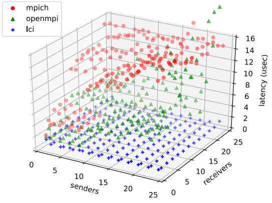
- Each work queue of a worker is a bit-vector
- Hierarchical bit-vectors for millions of tasks per node
- Load-balancing using work-stealing, highly scalable synchronizations
- Performance improvement upto 6x vs Argobots and Qthreads.



[ICPP'18] **Hoang-Vu Dang**, and Marc Snir, **"FULT: A Fast User-Level Thread Scheduling using bitvectors"**
[ESPM2'15] Alex Brooks, **Hoang-Vu Dang**, Nikoli Dryden, Marc Snir, **"PPL: An abstract runtime system for hybrid parallel programming"**

CONCLUSIONS

- MPI performance is lagging behind due to the changes in architecture and usage patterns
- Performance of message-passing can be improved with better data structures and relaxation in semantics
- LCI represents a clean ground-up design, very low-overhead and highly integrated with threads
- FUTL is a thread scheduling technique and library for scalable communication synchronization
- Future work: a standard LCI API, new micro-benchmarks, integration MPI + OpenMP



LCI+FULT outperforms MPICH and OpenMPI on OpenMP using multi-threaded point-to-point varying number of sender or receiver threads.

ACKNOWLEDGEMENTS

NSF

XSEDE
Extreme Science and Engineering Discovery Environment

Argonne NATIONAL LABORATORY

Galois

CONTACTS AND LINKS

Hoang-Vu Dang: hdang8@illinois.edu

LCI: https://github.com/danghvu/LCI

UIUC-HPC: https://github.com/uiuc-hpc

D-Galois: http://iss.ices.utexas.edu/?p=projects/galois

I ILLINOIS