Efficient Message Passing Support for Asynchronous Many-Task Systems

Jiakun Yan[1], Hartmut Kaiser[2], Marc Snir[1]

[1]Department of Computer Science,, University of Illinois at Urbana-Champaign

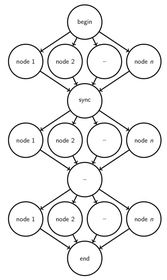
[2]Center of Computation and Technology, Louisiana State University

Motivation

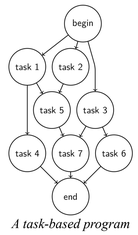
Modern Parallel Architecture:
- Increased intra-node parallelism.
- Increased heterogeneity.
- Powerful Interconnect.

Task-based Programming Model:
- Programmers decompose their programs into **tasks** along with their **dependencies**.
- The runtime will handle the **mapping**, **scheduling,** and **data movement**.
- E.g. HPX, Legion, PaRSEC.
- Communication layer: MPI/GASNet.

New Communication Pattern:
- Multithreaded.
- Irregular destinations.
- Small messages.



A BSP program

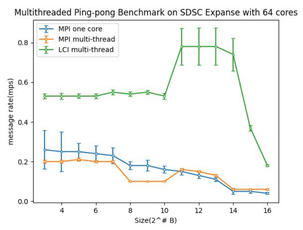


A task-based program

Lightweight Communication Interface (LCI) [1]

A low-level communication library.
- Intended user: high-level library developers.
- Like UCX/Libfabric/GASNet, as opposed to MPI.

Designed with task-based runtime as target clients.
- Applies to other irregular applications such as graph analysis/sparse linear algebra.



Major features:

- **Flexible communication primitives and signaling mechanisms.**
- **Better multithreaded performance.**
- **Explicit user control of communication behaviors and resources.**

HPX [2]

Task-based programming model with implicit task dependency graphs.
- Users invoke tasks like sequential code, and the runtime analyzes data usage and builds a task dependency graph.

Its communication layer implements a "parcelport" interface.
- A parcel consists of a small buffer (control data + small arguments) and optionally a few large zero-copy buffers.
- Current implementations: TCP, MPI, libfabric (work in progress).

The LCI Parcelport for HPX

Dedicated LCI progress threads:
- Better cache locality, cleaner worker thread behavior, more responsive and consistent network behavior.
- Instead of implicit background works in every MPI calls.

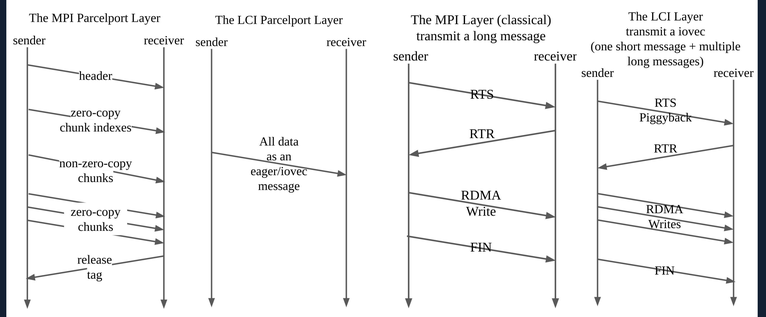
One-sided put operation:
- Only send. No receive needs to be posted.
- Receiver will get the references to the data from LCI **completion queues**.
- Instead of pre-posting multiple receives, probing, and then posting more follow-up receives in the MPI parcelport.

Directly put an "iovec" message:
- Send one eager message and multiple rendezvous messages in one call.
- Instead of multiple send/recv in the MPI parcelport.

Use Completion queues to deliver messages completion information:
- Instead of repeated MPI_Test for lots of MPI requests.

No mutex locks on HPX/LCI message passing path:
- Instead of multiple lock-protected layers in the HPX/MPI message passing path: parcel queues, connection cache, pending request store, and internal lock in MPI.

Other Optimizations:
- Thread-local backlog queues with aggregation.
- Separate network resources/progress threads for eager/rendezvous messages.



References

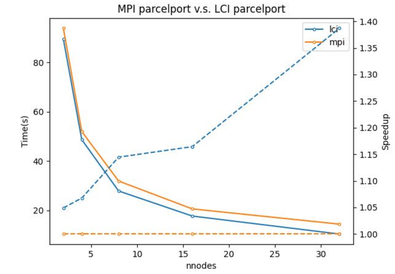
1. Dang, Hoang-Vu, Marc Snir, and William Gropp. "Towards millions of communicating threads." Proceedings of the 23rd European MPI Users' Group Meeting. 2016. (https://github.com/uiuc-hpc/LC/tree/dev-v1.7)
2. Kaiser, Hartmut, et al. "HPX: A task based programming model in a global address space." Proceedings of the 8th International Conference on Partitioned Global Address Space Programming Models. 2014. (https://github.com/STEllAR-GROUP/hpx)
3. Marcello, Dominic C., et al. "octo-tiger: a new, 3D hydrodynamic code for stellar mergers that uses hpx parallelization." Monthly Notices of the Royal Astronomical Society 504.4 (2021): 5345-5382. (https://github.com/STEllAR-GROUP/octotiger)

Evaluation

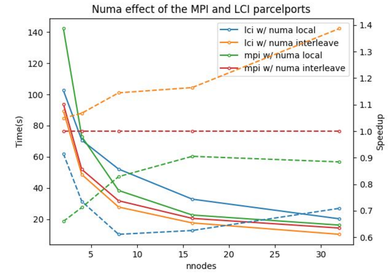
Platform: SDSC Expanse
- AMD EPYC 7742, 2 sockets, 4 NUMA nodes per socket, 128 cores (per node)
- 256GB Memory per node.
- HDR InfiniBand Interconnect.

Application: Octo-Tiger
- Astrophysics program simulating the evolution of star systems.
- Based on the fast multipole method on adaptive Octrees.

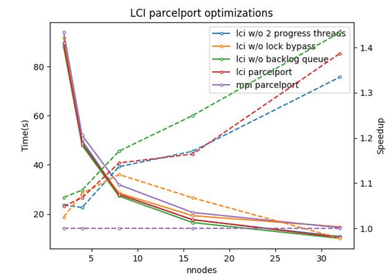
Results:
- LCI v.s. MPI:
  - The LCI parcelport surpassed the MPI parcelport by more than 40%.



- NUMA effects:
  - NUMA effects are significant.
  - LCI is more sensitive to NUMA effects than MPI.



- Specific optimizations:
  - Removing locks on the communication path helps. (~40% speedup)
  - Using separate resources helps. (~4% speedup)
  - The backlog queue doesn't
  - help. (~3% slowdown)



NSF

ILLINOIS