Task Mapping of Parallel Applications using Graph Partitioners

Mehmet Deveci[1,2], Abhinav Bhatele[2], Peter Robinson[2], Ümit V. Çatalyürek[1]
[1]The Ohio State University, [2]Lawrence Livermore National Laboratory

THE OHIO STATE UNIVERSITY
COLLEGE OF MEDICINE

Abstract

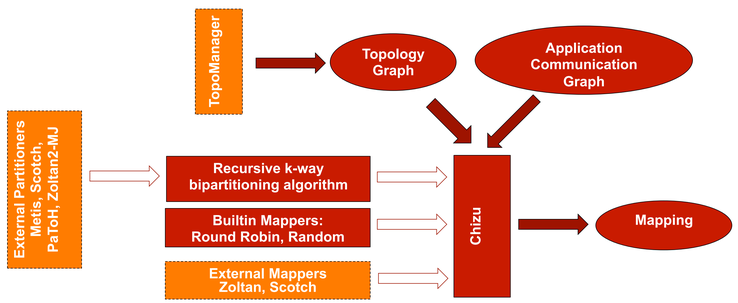
Communication time of parallel applications can be limited by various features of the interconnection networks such as latency or bandwidths of the links, and/or of the network card controllers. Topology aware task mapping methods that place MPI tasks on processors by exploiting information about the underlying network can help to avoid such limitations. In this work, we present a new framework for topology aware task mapping to reduce the applications' communication time.

Motivation

- The scale of the parallel applications and the number of processors in supercomputers have increased from O(100K) to O(1M)
  - large and hierarchical networks
  - sparse allocations where processors are spread further
  - communication messages travel longer routes
  - network links may be congested due to the heavy traffic
- A good partitioning and **mapping** of the tasks to the parallel supercomputer cores becomes crucial to:
  - utilize computation and communication units better
  - use less energy
  - obtain shorter execution times
- This problem is called **Mapping** Problem. The **aim of Chizu** is to improve the execution time by minimizing communication bottlenecks via task placement

Chizu Framework

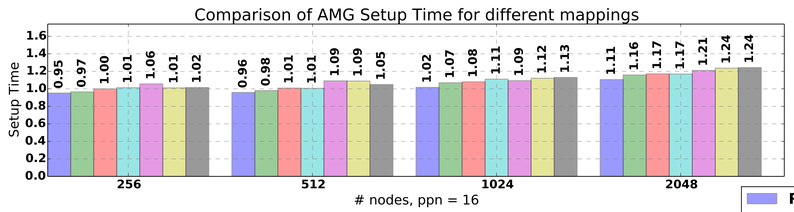
- Captures the underlying **topology** using **TopoManager**
- Provides various **mapping** algorithms:
  - Simple mapping methods such as:
    - **Round Robin** (RR), **Random**
  - Interfaces to existing mappers such as:
    - **Zoltan** [Deveci14] and **Scotch** [Pellegrini94]



  - Provides interfaces to:
    - Graph partitioners: **Metis** [Karypis99], **Scotch** [Pellegrini94]
    - Hypergraph partitioner: **PaToH** [Çatalyürek99]
    - Geometric partitioners: Zoltan **(MJ)** [Deveci12] of Trilinos

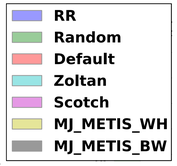
    to be used in **recursive k-way bipartitioning** algorithm to optimize for:
    - **Bandwidth** utilization
    - **Hop count** minimization
    - Other architecture specification metrics
- Can **simultaneously perform load balancing and task mapping** for parallel applications, as a side effect

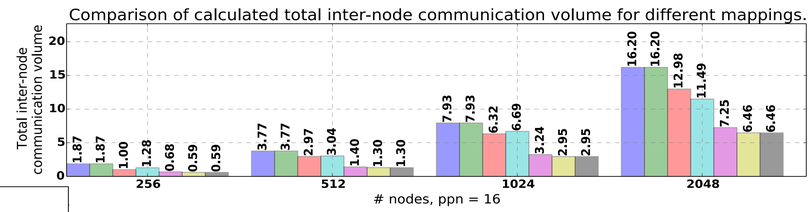
AMG Experiments



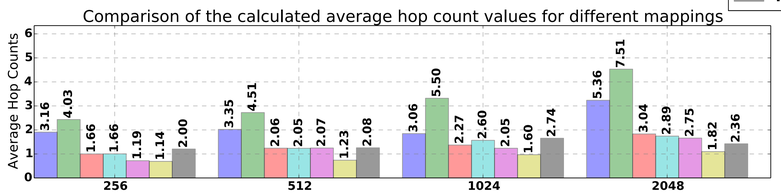
AMG2013 is a parallel algebraic multigrid solver for linear systems. Experiments for AMG on Vulcan (Blue Gene/Q at LLNL):

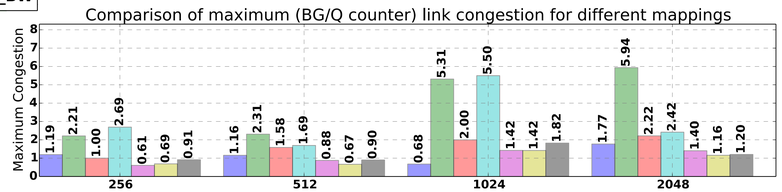
- Using 8 different mapping methods (Normalized w.r.t Default-256)
- On 256, 512, 1024, and 2048 nodes, with PPN=16



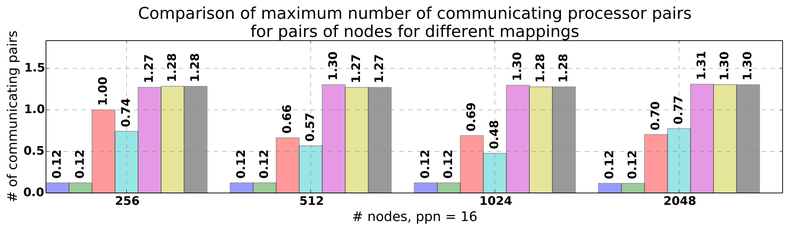


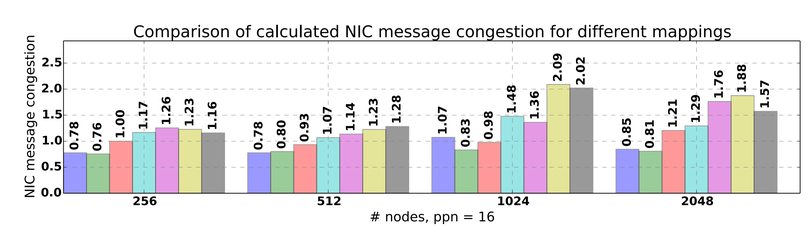
Traditional **hop count**, **max congestion**, and **inter-node communication volume** metrics do not correlate with the runtime results.





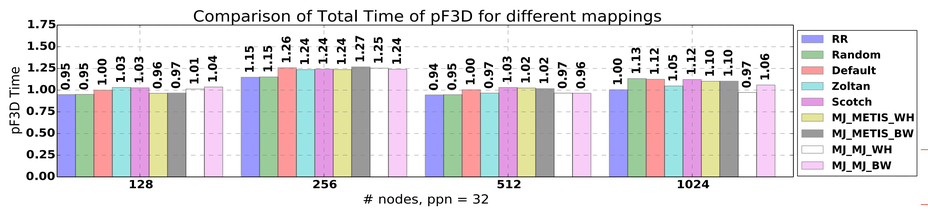
Bottleneck is likely to occur at the network controller (**NIC**) of the nodes. In BG/Q systems, there are 10 queues on the NIC, which are selected using a **static** algorithm (hop count % 10).





pF3D Experiments

pF3D is a multi-physics code used to study laser plasma-interactions in experiments conducted at the National Ignition Facility (NIF) at LLNL.



Conclusions and Future Work

- Implemented interfaces to new partitioning and mapping algorithms for the Chizu framework.
- Proposed a recursive k-way bi-partitioning algorithm that can be used for minimization of different mapping communication metrics.
- Studied the effectiveness of Chizu on the applications: AMG and pF3D; discussed the prediction capability of the traditional theoretical metrics.
- **Future Work:** Adding NIC congestion minimization metric to Chizu.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work was funded by the Laboratory Directed Research and Development Program at LLNL under project tracking code 13-ERD-055 (LLNL-POST-658094).

References

- [Deveci14] M. Deveci, S. Rajamanickam, V. Leung, K. T. Pedretti, S. L. Olivier, D. P. Bunde, Ü. V. Çatalyürek, K. D. Devine, Exploiting Geometric Partitioning in Task Mapping for Parallel Computers, 28th IEEE International Parallel and Distributed Processing Symposium, May 2014
- [Pellegrini94] F. Pellegrini., Static mapping by dual recursive bipartitioning of process and architecture graphs. Proceedings of SHPCC'94, Knoxville, Tennessee, pages 486-493. IEEE Press, May 1994.
- [Karypis99] George Karypis and Vipin Kumar, A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs", SIAM Journal on Scientific Computing, Vol. 20, No. 1, pp. 359—392, 1999
- [Çatalyürek99] Ü. V. Çatalyürek, Cevdat Aykanat, PaToH: a multilevel hypergraph partitioning tool, version 3.0, Bilkent University, Department of Computer Engineering, 1999
- [Deveci12] M Deveci, Ü. V. Çatalyürek, S Rajamanickam, KD Devine, Multi-jagged: A Scalable Multi-section based Spatial Partitioning Algorithm, Sandia National Laboratories, 2012

mdeveci@bmi.osu.edu