Task Mapping of Parallel Applications using Graph Partitioners

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

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
- Network links may be congested due to the heavy traffic
- Communication messages travel longer routes

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
- Minimize communication bottlenecks via task placement

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 [Deveci12], Scotch [Pedretti14]

  - To be used in a 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

Acknowledgments

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


Conclusions and Future Work

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