Motivation

• Increase in the data sizes forces scientific computing to execute parallel jobs.
• A good partitioning of the tasks to the parallel supercomputer cores becomes crucial to:
  • utilize computation and communication units better
  • use less energy
  • obtain shorter execution times
• Number of processors in supercomputers 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
• Not only a good partitioning of the tasks, but also a good mapping of them to the processors is crucial to obtain a better performance.
• This problem is called Mapping Problem

Models and Methods

• Computational tasks are represented using different models
  • Spatial Model: a geometric model
  • Connectivity-Based Models: graph model, hypergraph model
  • The Mapping problem is solved using any of the models
• Usually with a 2-phase approach:
  • First, a load balanced partition of the tasks is found
  • Then, the obtained parts are mapped to the cores of a supercomputer

Conclusions and Future Work

Load Balancing:

• Geometric partitioner
• A parallel multi-sectioning method
• Heuristics to minimize the data movements
• Connectivity-based, hypergraph, partitioner, UMPa
• The use of directed hypergraph models
• Methods for multi-objective hypergraph partitioning

Task Mapping:

• Task mapping using geometric model:
  • The use of geometric partitioning algorithm for task mapping
  • Heuristics to improve the quality further
• Task mapping using graph model:
  • Greedy mapping and refinement method
• Extending task mapping work using graph models in order to address:
  • Hierarchical architectures with different interconnection networks
  • Different routing mechanisms
• Studying 1-phase mapping solutions
• Different phases in 2-phase methods seek for different objectives
  • The first phase is not aware of the architecture
  • Global optima may not be found

Acknowledgements

This work was supported in part by the NSF grants OCI-0904809. We thank to Kamer Kaya, Bora Uçar, Karen Devine, Siva Rajamanickam, Vitus Leung, David Bunde, Stephen Oliver and Kevin Pedretti for useful discussions and their contributions.

References