Load Balancing and Task Mapping For Exascale Systems

Mehmet Deveci and Ümit V. Çatalyürek
The Ohio State University
{mdeveci, umit}@bmi.osu.edu

THE OHIO STATE UNIVERSITY
COLLEGE OF MEDICINE

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 usefull discussions and their contributions.

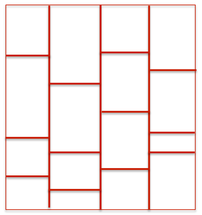
References

- "Exploiting Geometric Partitioning in Task Mapping for Parallel Computers", M. Deveci, S. Rajamanickam, V. Leung, K. T. Pedretti, S. L. Olivier, D. P. Bunde, Ü. V. Çatalyürek, K. D. Devine, 28th IEEE International Parallel and Distributed Processing Symposium, May 2014
- Technical Report, "Multi-jagged: A Scalable Multi-section based Spatial Partitioning Algorithm", M. Deveci, Ü. V. Çatalyürek, S. Rajamanickam, K. D. Devine, Sandia National Laboratories, SAND2012-10318C
- "Hypergraph partitioning for multiple communication cost metrics: Model and methods", M. Deveci, K. Kaya, B. Uçar, Ü. V. Çatalyürek, Journal of Parallel and Distributed Computing, 2014 (under review)

Spatial Models

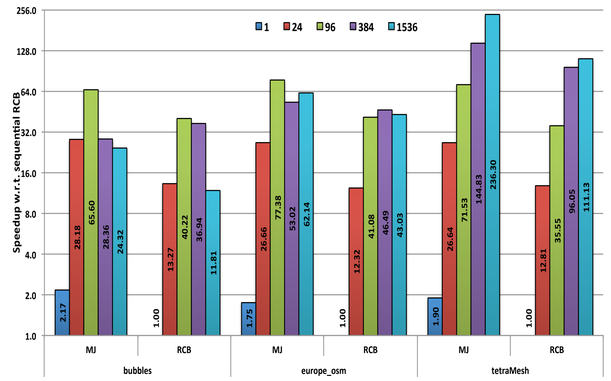
Geometric Partitioning:
- Given the set of points associated with coordinates and weights, distribute the coordinates into equally weighted parts based on the locality of the coordinates

Applications: The applications requiring geometric data locality
- Particle methods, crash simulations, direct volume rendering

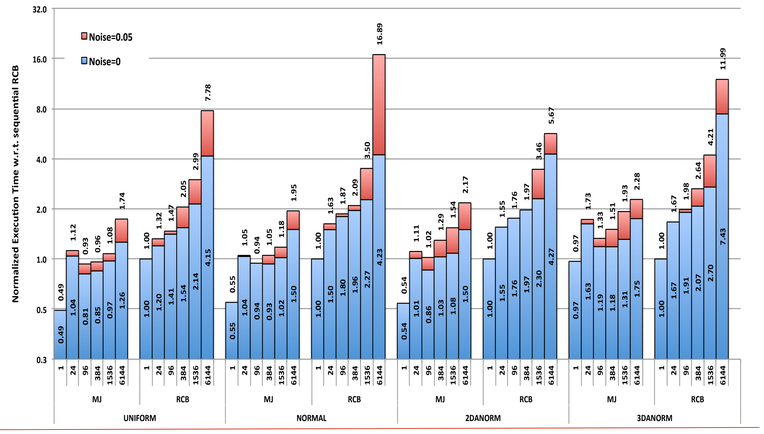


Geometric Partitioning

Parallel Multi-Jagged (MJ) Algorithm

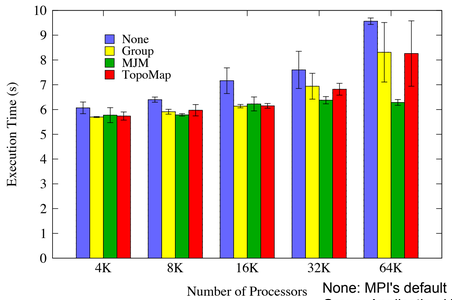


- Multi-sections instead of bisections
- Hybrid (OpenMP + MPI)
- Avoids migrations by checking:
  - Load Imbalance vs. migration cost
- Number of global messages
- If further partitionings will be communication bounded



Task Mapping

Objective: Map tasks that are "close" to each other to processors that are close in a mesh or torus network
- The machine topology and application's MPI procs are represented by coordinates
- MJM: MJ is used to consistently reorder both the MPI processes and the allocated cores. The ordering is used to construct the mapping
- Uses pre-processing heuristics to improve the quality:
  - Rotations of the coordinate dimensions
  - Transform the coordinates to account for wrap-around in each dimension
- Experiments for Minighost on Cielo Cray XE6 (3D Tourus)



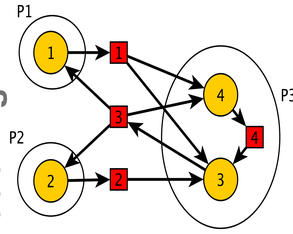
None: MPI's default
Group: Application Heuristic
MJM: Proposed method
TopoMap: [Hoefler & Snir 2011]

Connectivity-based Models

Problem: Distributing communicating tasks among processing units.
- Balanced load distribution
- Good communication pattern

Hypergraph Partitioning: Find a balanced partition of the tasks that minimizes multiple communication metrics

Directed Hypergraph Model

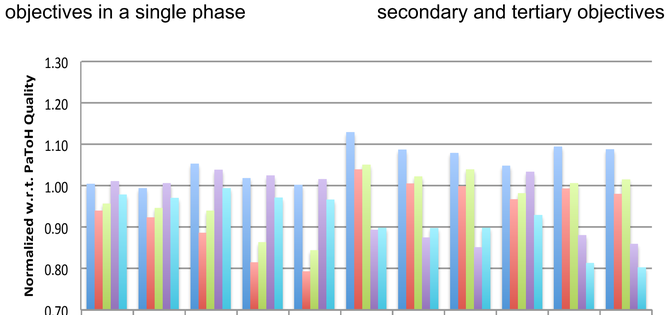


- Allows to formulate and minimize other communication metrics in addition to the total volume metric
- **TV**: Total communication volume
- **MSV:** Max. of the send vol. of processors
- **MSRV:** Max. of the communication (send/recv) vol. of the processors
- **TM**: Total number of mess. exchanged
- **MSM:** Max. of the send mess. of the procs
- **MSRM:** Max. of the send/recv mess. of the procs

Hypergraph Partitioning

UMPa Multiobjective Multilevel Hypergraph Partitioner

- K-way multi-level partitioner
- Uses directed hypergraph model
- Minimizes multiple communication objectives in a single phase
- Handles multiple metrics with a tie-breaking heuristic
  - Prioritizes metrics as primary, secondary and tertiary objectives



		+PW		+MSRV +TV	+PW		+TV	+PW		+TM +TV	+PW
	UMPa$_{TV}$		UMPa$_{MSV}$			UMPa$_{TM}$			UMPa$_{MSM}$		
TV	1.00	0.99	1.05	1.02	1.00	1.13	1.09	1.08	1.05	1.09	1.09
MSV	0.94	0.92	0.89	0.81	0.79	1.04	1.01	1.00	0.97	0.99	0.98
MSRV	0.96	0.95	0.94	0.86	0.84	1.05	1.02	1.04	0.98	1.01	1.01
TM	1.01	1.01	1.04	1.02	1.02	0.89	0.87	0.85	1.03	0.88	0.86
MSM	0.98	0.97	0.99	0.97	0.97	0.90	0.90	0.90	0.93	0.81	0.80

Task Mapping

Objective: Map the tasks to the processors to reduce both the congestion of the links and the hops that messages travel
- Uses graph model to represent communicating tasks and machine topology
  - Irregular applications and hierarchical networks can be represented
- Works in two phases:
  - Greedy growing phase reduces the hop metric
  - Refinement phase refines the solution to reduce the congestion
- Experiments on 42 graphs for Cielo Cray XE6 Supercomputer for K=1024, 2048, 4096, 8192, 16384

Table: Geometric means of the metrics (maximum congestion (MC), weighted hop (WH), total hop (TH), and maximum message congestion (MMC)) of the mapping algorithms normalized with respect respect to those of the default mapping of PaToH results are given below. In addition, the geometric means of the execution times of the mapping algorithms are given at the bottom.

		PaToH				$UMPa_{MM}$				$UMPa_{MV}$				$UMPa_{TM}$			
Metric	PPN	Default	UMPa	TOPOMAP	SCOTCH	Default	UMPa	TOPOMAP	SCOTCH	Default	UMPa	TOPOMAP	SCOTCH	Default	UMPa	TOPOMAP	SCOTCH
MC	1	1.00	**0.72**	0.95	0.95	1.01	0.75	0.94	1.06	1.01	0.75	0.99	1.06	1.01	0.75	0.94	1.06
	4	1.00	0.78	0.92	1.14	1.00	**0.75**	0.92	1.21	1.00	0.79	0.89	1.14	1.00	**0.75**	0.92	1.21
	16	1.00	0.57	0.92	1.12	1.01	0.58	0.96	1.22	1.03	**0.53**	0.98	1.23	1.08	0.58	1.03	1.10
WH	1	1.00	0.84	0.95	**0.83**	1.00	0.84	0.95	**0.83**	0.99	**0.83**	0.97	**0.83**	1.00	0.84	0.95	**0.83**
	4	1.00	0.75	1.00	1.21	1.00	0.75	1.00	1.20	1.00	**0.74**	1.00	1.21	1.00	0.75	1.00	1.20
	16	1.00	**0.74**	1.03	1.14	1.04	0.78	1.06	1.24	1.04	0.82	1.05	1.22	1.05	0.78	1.11	1.23
TH	1	1.00	0.83	0.94	0.82	0.87	0.73	0.82	**0.71**	1.00	0.83	0.97	0.82	0.87	0.73	0.82	0.71
	4	1.00	**0.73**	0.96	1.11	1.00	**0.73**	0.97	1.10	1.00	**0.73**	0.95	1.12	1.00	**0.73**	0.97	1.10
	16	1.00	0.73	1.00	1.08	0.96	**0.70**	0.95	1.06	0.99	0.76	0.98	1.07	0.97	0.71	0.97	1.05
MMC	1	1.00	0.90	0.90	0.82	0.83	0.80	0.79	**0.73**	0.98	0.89	0.93	0.86	0.83	0.80	0.79	**0.73**
	4	1.00	0.77	1.04	1.16	1.00	**0.76**	0.98	1.14	1.00	**0.76**	0.99	1.22	1.00	**0.76**	0.98	1.14
	16	1.00	0.99	1.13	1.33	1.04	**0.81**	1.11	1.40	1.02	0.87	1.03	1.23	1.12	0.94	1.17	1.16
Time	1	0.00	1.46	0.83	0.51	0.00	0.99	0.82	0.50	0.00	1.09	0.83	0.51	0.00	0.99	0.82	0.50
	4	0.00	0.31	0.55	0.21	0.00	0.50	0.55	0.21	0.00	0.30	0.56	0.21	0.00	0.50	0.56	0.21
	16	0.00	1.36	0.33	0.16	0.00	1.09	0.32	0.16	0.00	1.79	0.32	0.16	0.00	1.60	0.32	0.16