Topology-aware Optimization of Communications for Parallel Matrix Multiplication on Hierarchical Heterogeneous HPC Platforms

Tania Malik, Vladimir Rychkov, Alexey Lastovetsky, Jean-Noël Quintin

Heterogeneous Computing Laboratory
University College Dublin, Ireland

Heterogeneity in Computing Workshop
Phoenix-Arizona, USA
19-25 May, 2014

Outline

- Motivation
- Problem Formulation
- Topology-aware Communication Optimization Approach
    - Cost function
    - Heuristic
- Experiments
- Conclusion

Introduction

- For efficient execution of data-parallel applications on HPC platform:
  - Balance the load between processors
  - **Optimize communication cost**
- Communications on heterogeneous platform involve:
  - Multiple message hops
  - Non-optimal routes
  - Traffic congestion
  - Significantly affect performance

Introduction

- For efficient execution of data-parallel applications on HPC platform:
  - Balance the load between processors
  - **Optimize communication cost**
- Communications on heterogeneous platform involve:
  - Multiple message hops
  - Non-optimal routes
  - Traffic congestion
  - Significantly affect performance
- **With topology information, communication operations can be optimized**

Topology-Aware Optimisation of Communications

- Number of topology-aware MPI collective operations have been proposed for optimal scheduling of messages
  - Improves communication performance
  - Non-intrusive to source code

Topology-Aware Optimisation of Communications

- Number of topology-aware MPI collective operations have been proposed for optimal scheduling of messages
  - Improves communication performance
  - Non-intrusive to source code
  - **Applicable to collective operations only**
  - **Does not affect point-to-point exchanges**

What To Do

- To address the problem of communication optimization in such data-parallel MPI applications, must take into account:
  - Topology information
  - Application communication flow

What To Do

- To address the problem of communication optimization in such data-parallel MPI applications, must take into account:
    - Topology information
    - Application communication flow
- Choose specific parallel application
    - Matrix multiplication based on the Scalable Universal Matrix Multiplication Algorithm (SUMMA)

What To Do

- To address the problem of communication optimization in such data-parallel MPI applications, must take into account:
  - Topology information
  - Application communication flow
- Choose specific parallel application
  - Matrix multiplication based on the Scalable Universal Matrix Multiplication Algorithm (SUMMA)
- Target dedicated heterogeneous HPC platforms with network hierarchy
  - Interconnected clusters

Problem Formulation

- Select parallel matrix multiplication application for heterogeneous platform based on SUMMA
  - SUMMA originally designed for homogeneous platform
  - Communication flow consists of multiple broadcasts
- Assuming workload is already balanced
  - Existing load balancing algorithm are oblivious to network topology
- Rearrange existing heterogeneous data partition based on network topology and application communication flow

Problem Formulation

- Select parallel matrix multiplication application for heterogeneous platform based on SUMMA
    - SUMMA originally designed for homogeneous platform
    - Communication flow consists of multiple broadcasts
- Assuming workload is already balanced
    - Existing load balancing algorithm are oblivious to network topology
- Rearrange existing heterogeneous data partition based on network topology and application communication flow
    - **Approach is non-intrusive to the source code but application-specific**

Communication Flow of Heterogeneous SUMMA

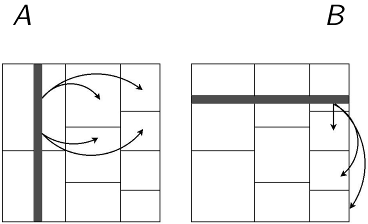


Figure : Communication flow of heterogeneous SUMMA: one-to-all

Load Balancing

- Number of partitioning algorithms exist for efficient load balancing
  - **Column-Based Partitioning**

    (Kalinov and Lastovetsky 1999) (KL)
  - **Minimising Total Communication Volume**

    (Beaumont, Boudet, Rastello, Robert, 2001) (BR)
  - **1D Functional Performance Model-based Partitioning**

    (Lastovetsky, Reddy, 2007) (FPM1D)
  - **2D Functional Performance Model-based Matrix Partitioning Algorithm**

    Clarke, Lastovetsky, Rychkov, 2011 (FPM-BR)

Communication Flow of Heterogeneous SUMMA

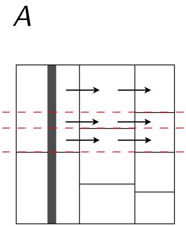


Figure : Communication flow of heterogeneous SUMMA implementing FPM-BR: ring

Comparison of some SUMMA-based algorithms

Table : Comparison of some SUMMA-based algorithms

Algorithm	Data partitioning	Communication vol.	Communication flow
SUMMA	homogeneous	–	broadcasts
BR	constant speeds	min	nb-p2p one-to-all
FPM-BR	speed functions	min	nb-p2p one-to-all/ring

Matrix Partitioning Algorithm

- FPM-BR algorithm:
  - Balances the workload
  - Minimizes the total volume of communication

Matrix Partitioning Algorithm

- FPM-BR algorithm:
  - Balances the workload
  - Minimizes the total volume of communication
- **However, none of the Matrix Multiplication load balancing algorithms takes into account the underlying networks topology**

Matrix Partitioning Algorithm

- FPM-BR algorithm:
  - Balances the workload
  - Minimizes the total volume of communication
- **However, none of the Matrix Multiplication load balancing algorithms takes into account the underlying networks topology**
- Goal is to reduce communication cost of the parallel application that implements the FPM-BR matrix multiplication algorithm

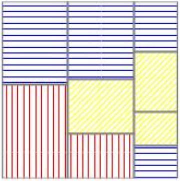
Matrix Partitioning Algorithm

- FPM-BR algorithm:
  - Balances the workload
  - Minimizes the total volume of communication
- **However, none of the Matrix Multiplication load balancing algorithms takes into account the underlying networks topology**
- Goal is to reduce communication cost of the parallel application that implements the FPM-BR matrix multiplication algorithm
- **Rearrange existing heterogeneous data partition based on network topology and application communication flow**

Exhaustive Search Partitions

- Performed exhaustive search with all possible arrangements of rectangles
  - Found several arrangements that reduced and increased communication cost

Exhaustive Search Partitions



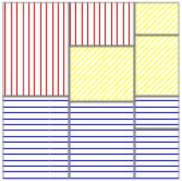


Figure : Communication optimal arrangements

Exhaustive Search Partitions

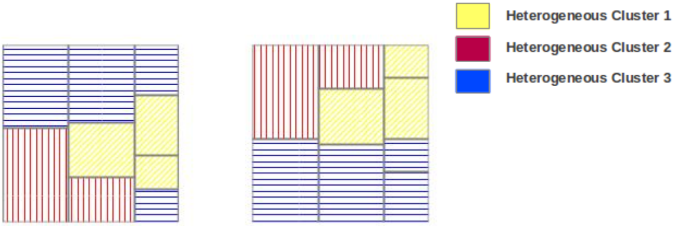


Figure : Communication optimal arrangements

Exhaustive Search Partitions

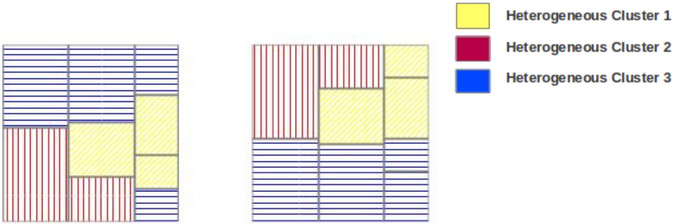


Figure : Communication optimal arrangements

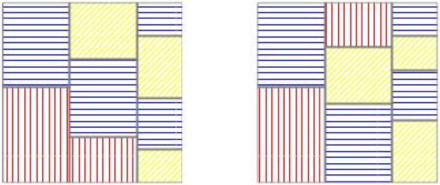
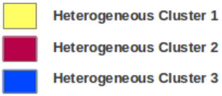


Figure : Worst case arrangements

Exhaustive Search Partitions



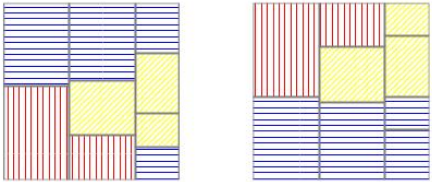


Figure : Communication optimal arrangements

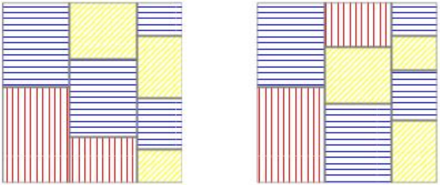
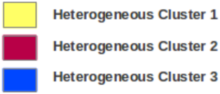


Figure : Worst case arrangements

- Observed regularity in the comm-optimal arrangements related to the topology
  - Rectangles were grouped by clusters
  - Less inter-cluster comm.

Exhaustive Search Partitions



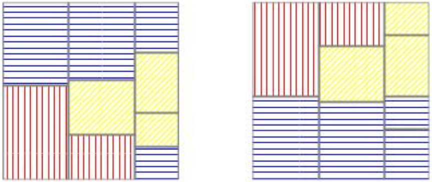


Figure : Communication optimal arrangements

- Observed regularity in the comm-optimal arrangements related to the topology
  - Rectangles were grouped by clusters
  - Less inter-cluster comm.

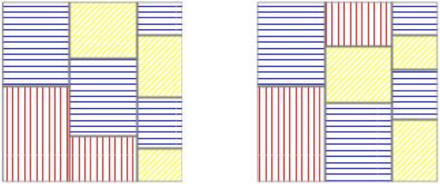


Figure : Worst case arrangements

Table : Exhaustive search experimental results

	Cost		Exec time (sec)	
	Worst case	Optimal	Worst case	Optimal
Exhaustive search	89.80	73.59	6.00	2.78

Search Space Size

- Column widths are different:
  - Cannot move a rectangle to another column unless the whole columns are interchanged
- In column, no restrictions on interchanges of rectangles

Search Space Size

- Column widths are different:
  - Cannot move a rectangle to another column unless the whole columns are interchanged
- In column, no restrictions on interchanges of rectangles
- **Let**
  - c be the number of columns
  - r_i be the number of rectangles in column i, $1 \leq i \leq c$
  - Number of combinations will be equal to the product $r_1! \times \ldots \times r_c!$

NP-Complete

- **Which arrangement of rectangles is communication-optimal?**
    - NP-complete problem

NP-Complete

- **Which arrangement of rectangles is communication-optimal?**
  - NP-complete problem
- **Exhaustive search can be avoidable**
  - By applying some heuristic that efficiently finds a near optimal arrangement

Requires to estimate the communication cost incurred by each data partitioning

Cost Function

- Based on observation from exhaustive search
  - Propose cost function for FPM-BR

Ring Communication flow
Two level network Hierarchy

Cost function for Matrix A

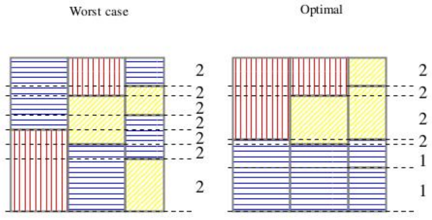


Figure : Inter-cluster Communication related to matrix A

Cost function for Matrix A

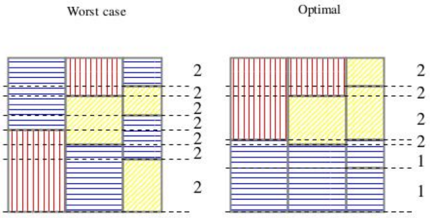


Figure : Inter-cluster Communication related to matrix A

- **Let**
- $o=$ Overlaps of matrix rectangles
- $h=$ No. of inter-cluster Communication
- $v=$ Height of overlap
- $cost_A = \sum_{i=1}^{o} h(i) \times v(i)$

Cost function for Matrix A

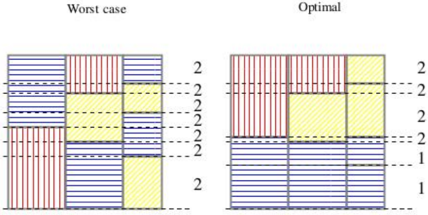


Figure : Inter-cluster Communication related to matrix A

- **Let**
- $o=$ Overlaps of matrix rectangles
- $h=$ No. of inter-cluster Communication
- $v=$ Height of overlap
- $cost_A = \sum_{i=1}^{o} h(i) \times v(i)$
- Worst case: $2 \times (11+3+3+3+4+2+6) = 64$
- Optimal: $1 \times (6+8) + 2 \times (1+9+2+6) = 50$

Cost function for Matrix B

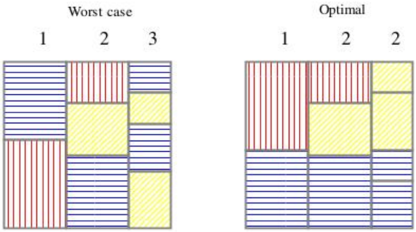


Figure : Inter-cluster Communication related to matrix B

Cost function for Matrix B

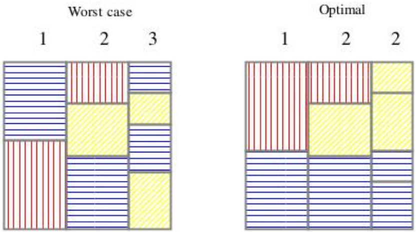


Figure : Inter-cluster Communication related to matrix B

- **Let**
- c=Total columns
- h= No. of inter-cluster Communication
- v= Column width
- $cost_B = \sum_{i=1}^{c} h(i) \times v(i)$

Cost function for Matrix B

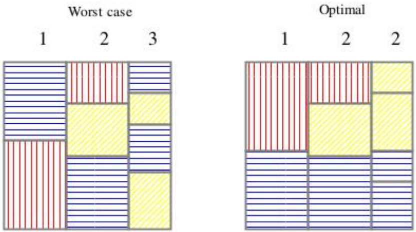


Figure : Inter-cluster Communication related to matrix B

- **Let**
- c=Total columns
- h= No. of inter-cluster Communication
- v= Column width
- $cost_B = \sum_{i=1}^{c} h(i) \times v(i)$
- Worst case: $(1 \times 12) + (2 \times 12) + (3 \times 9) = 63$
- Optimal: $(1 \times 12) + (2 \times 12) + (2 \times 9) = 54$

Cost function for M Arrangement

- Use Euclidean norm
  - Represent combined cost and can be used to compare any two arrangements
- $\|(cost_A(M), cost_B(M))\|$
  - Worst case: $\sqrt{64^2 + 63^2} = 89.80$
  - Optimal case: $\sqrt{50^2 + 54^2} = 73.59$

Cost function for M Arrangement

- Use Euclidean norm
  - Represent combined cost and can be used to compare any two arrangements
- $\|(cost_A(M), cost_B(M))\|$
  - Worst case: $\sqrt{64^2 + 63^2} = 89.80$
  - Optimal case: $\sqrt{50^2 + 54^2} = 73.59$
- finding the communication-optimal arrangement can be formulated as minimization of the Euclidean norm:
  - $\|(cost_A(M), cost_B(M))\| \rightarrow \min$

Cost function for M Arrangement

- Use Euclidean norm
  - Represent combined cost and can be used to compare any two arrangements
- $\|(cost_A(M), cost_B(M))\|$
  - Worst case: $\sqrt{64^2 + 63^2} = 89.80$
  - Optimal case: $\sqrt{50^2 + 54^2} = 73.59$
- finding the communication-optimal arrangement can be formulated as minimization of the Euclidean norm:
  - $\|(cost_A(M), cost_B(M))\| \rightarrow \min$
- Use cost function in Heuristic

Heuristic for the Communication-Optimal Arrangement

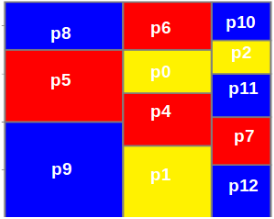
- Propose heuristic to avoid too many combination

Heuristic for the Communication-Optimal Arrangement

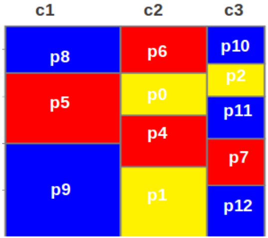
- Propose heuristic to avoid too many combination
  - Permutation based on groups

Requires to test $g_2! + \ldots + g_c!$ arrangements of submatrices

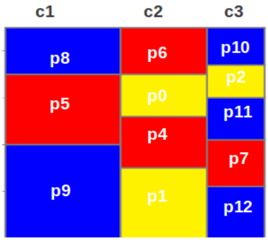
Heuristic for the Communication-Optimal Arrangement

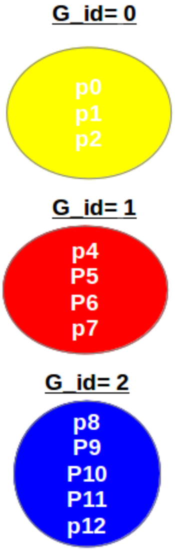


Heuristic for the Communication-Optimal Arrangement

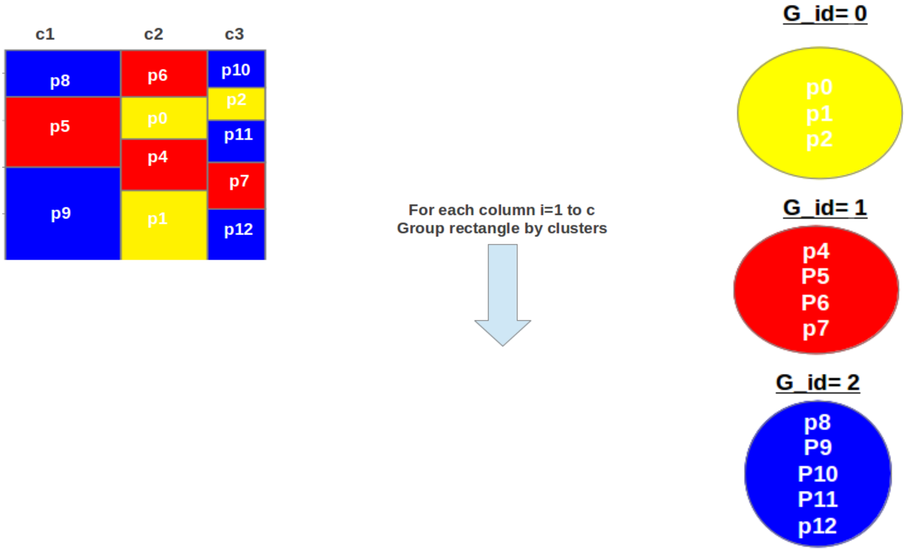


Heuristic for the Communication-Optimal Arrangement

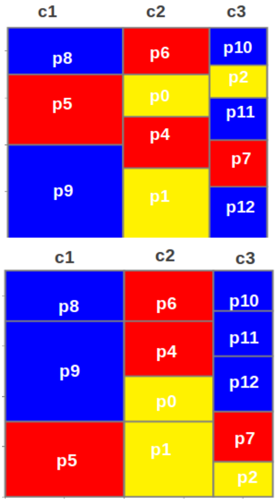




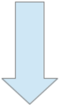
Heuristic for the Communication-Optimal Arrangement

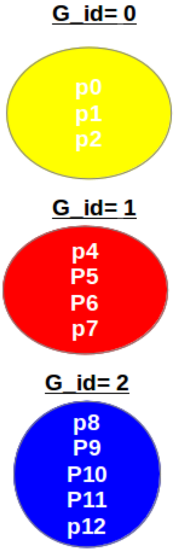


Heuristic for the Communication-Optimal Arrangement



For each column i=1 to c
Group rectangle by clusters





Heuristic for the Communication-Optimal Arrangement-2

- Accept c_1 as optimal order

Heuristic for the Communication-Optimal Arrangement-2

- Accept c_1 as optimal order
- Generate group permutations $g_i!$

Heuristic for the Communication-Optimal Arrangement-2

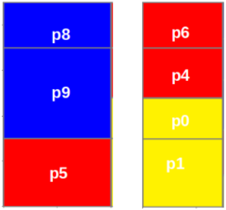
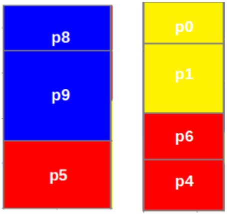


Figure :
Permutation order
k=1

- Accept c_1 as optimal order
- Generate group permutations g_i!

Heuristic for the Communication-Optimal Arrangement-2

- Accept c_1 as optimal order
- Generate group permutations $g_i!$



Heuristic for the Communication-Optimal Arrangement-2

- Accept c_1 as optimal order
- Generate group permutations $g_i!$
- For each permutation $k = 1$ to g_i
- Find k that has minimum cost function for extended sub-matrix

Heuristic for the Communication-Optimal Arrangement-2

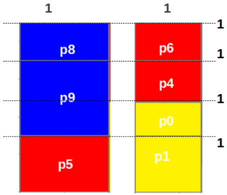


Figure : Permutation order k=1

- Accept c_1 as optimal order
- Generate group permutations g_i!
- For each permutation $k = 1$ to g_i
- Find k that has minimum cost function for extended sub-matrix

Heuristic for the Communication-Optimal Arrangement-2

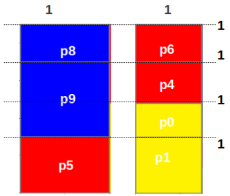


Figure :
Permutation order
k=1

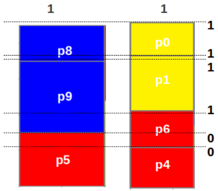


Figure :

- Accept c_1 as optimal order
- Generate group permutations $g_i!$
- For each permutation $k = 1$ to g_i
- Find k that has minimum cost function for extended sub-matrix

Heuristic for the Communication-Optimal Arrangement-2

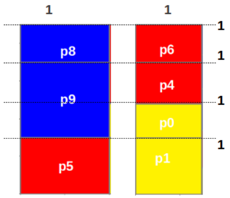


Figure : Permutation order k=1

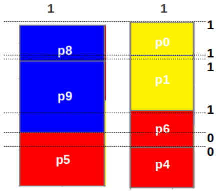


Figure :

- Accept c_1 as optimal order
- Generate group permutations g_i!
- For each permutation $k = 1$ to g_i
- Find k that has minimum cost function for extended sub-matrix
- Cost function for k1=45 and k2=35

Heuristic for the Communication-Optimal Arrangement-2

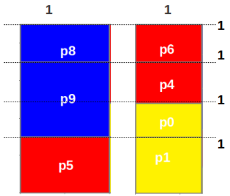


Figure :
Permutation order
k=1

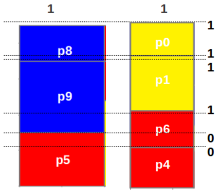


Figure :

- Accept c_1 as optimal order
- Generate group permutations g_i!
- For each permutation $k = 1$ to g_i
- Find k that has minimum cost function for extended sub-matrix
- Cost function for k1=45 and k2=35
- Add minimum k to resulting arrangement

Heuristic for the Communication-Optimal Arrangement-3

- Repeat the same steps for all c column

Heuristic for the Communication-Optimal Arrangement-3

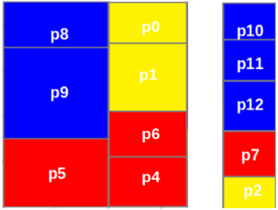


Figure : Permutation order k=1

- Repeat the same steps for all c column

Heuristic for the Communication-Optimal Arrangement-3

- Repeat the same steps for all *c* column

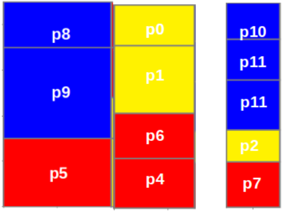


Figure : Permutation order k=2

Heuristic for the Communication-Optimal Arrangement-3

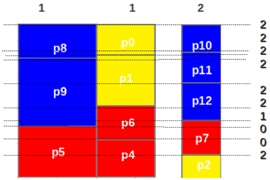


Figure : Permutation order k=1

- Repeat the same steps for all *c* column

Heuristic for the Communication-Optimal Arrangement-3

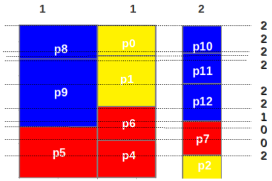


Figure : Permutation order k=1

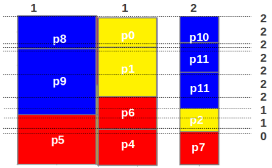


Figure : Permutation order k=2

- Repeat the same steps for all *c* column

Heuristic for the Communication-Optimal Arrangement-3

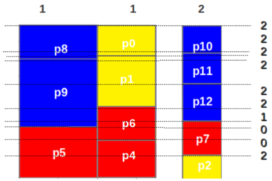


Figure : Permutation order k=1

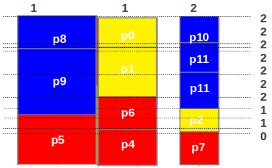


Figure : Permutation order k=2

- Repeat the same steps for all *c* column
- Cost function of k1=74 and k2=65
- Choose k2 as optimal order

Heterogeneous Inter-Cluster Experiments

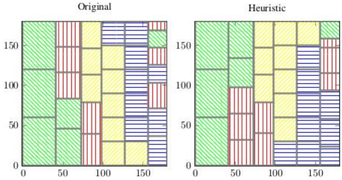


Figure : Matrix partitioning for 32 nodes

Heterogeneous Inter-Cluster Experiments

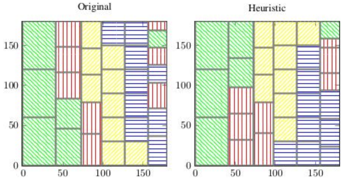
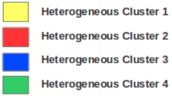


Figure : Matrix partitioning for 32 nodes



Heterogeneous Inter-Cluster Experiments

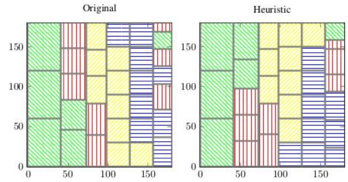


Figure : Matrix partitioning for 32 nodes

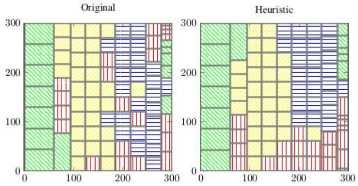
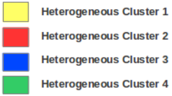


Figure : Matrix partitioning for 90 nodes



Heterogeneous Inter-Cluster Experiments

Table : Inter-cluster experimental results

Nodes	Cost		Exec time (sec)		Ratio
	Orig	Heuristic		Orig	Heuristic
16	533	432	58.00	42.58	1.36
32	868	710	119.30	88.30	1.35
90	1719	1263	400.80	297.83	1.34

Homogeneous Inter-Node Experiment

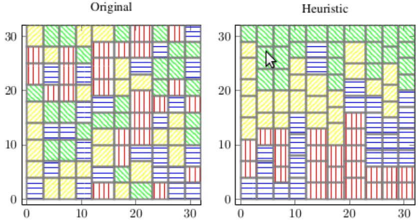


Figure : Partitioning for 4 homogeneous multi-core nodes

Homogeneous Inter-Node Experiment

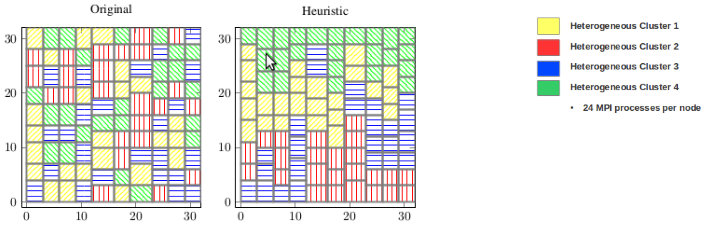


Figure : Partitioning for 4 homogeneous multi-core nodes

Homogeneous Inter-Node Experiment

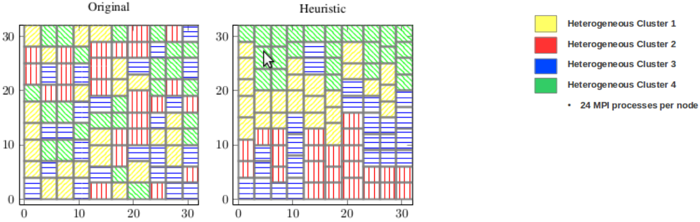


Figure : Partitioning for 4 homogeneous multi-core nodes

Homogeneous Inter-Node Experiment

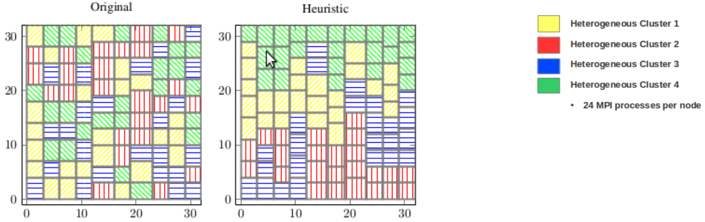


Figure : Partitioning for 4 homogeneous multi-core nodes

Table : Homogeneous inter-node experimental results

Nodes	Cost		Exec time (sec)		Ratio
	Orig	Heuristic		Orig	Heuristic
4	336	199	3.85	3.17	1.21

Conclusion

- Heuristic approach for combinatorial problem
- Prediction is based on topology and Communication flow
- Minimize inter-cluster communication cost