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
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
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**
Motivation

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

- Topology information
- Application communication flow
Motivation

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)
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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Data partitioning</th>
<th>Communication vol.</th>
<th>Communication flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMA</td>
<td>homogeneous</td>
<td>–</td>
<td>broadcasts</td>
</tr>
<tr>
<td>BR</td>
<td>constant speeds</td>
<td>min</td>
<td>nb-p2p one-to-all</td>
</tr>
<tr>
<td>FPM-BR</td>
<td>speed functions</td>
<td>min</td>
<td>nb-p2p one-to-all/ring</td>
</tr>
</tbody>
</table>
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.
Problem Formulation

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
Problem Formulation

Exhaustive Search Partitions

Figure: Communication optimal arrangements
Problem Formulation

Exhaustive Search Partitions

Figure: Communication optimal arrangements

Figure: Worst case arrangements
Problem Formulation

Exhaustive Search Partitions

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

Figure: Communication optimal arrangements

Figure: Worst case arrangements
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

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
<th>Exec time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst case</td>
<td>89.80</td>
<td>6.00</td>
</tr>
<tr>
<td>Optimal</td>
<td>73.59</td>
<td>2.78</td>
</tr>
</tbody>
</table>

Tania Malik (UCD HCL)
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
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
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 $A$

Let $o$ = Overlaps of matrix rectangles
$h$ = No. of inter-cluster Communication
$v_i$ = Height of overlap

$$\text{cost}_A = o \sum_{i=1}^{h} (v_i) \times (h)$$

Worst case: $2 \times (11 + 3 + 3 + 3 + 4 + 2 + 6) = 64$
Optimal: $1 \times (6 + 8) + 2 \times (1 + 9 + 2 + 6) = 50$

Figure: Inter-cluster Communication related to matrix $A$
Let

- \( o \) = Overlaps of matrix rectangles
- \( h \) = No. of inter-cluster Communication
- \( v \) = Height of overlap

\[
\text{cost}_A = \sum_{i=1}^{o} h(i) \times v(i)
\]

Figure: Inter-cluster Communication related to matrix A
Cost function for 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

\[
\text{cost}_B = \sum_{i=1}^{c} h(i) \times v(i)
\]

Worst case: \((1 \times 12) + (2 \times 12) + (3 \times 9) = 63\)

Optimal: \((1 \times 12) + (2 \times 12) + (2 \times 9) = 54\)

Figure: Inter-cluster Communication related to matrix \(B\)
Let

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

Figure: Inter-cluster Communication related to matrix $B$
Cost function for 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
  - \( \| (\text{cost}_A(M), \text{cost}_B(M)) \| \)
    - 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:
  - \( \| (\text{cost}_A(M), \text{cost}_B(M)) \| \rightarrow \min \)
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:
  - \[ \| (cost_A(M), cost_B(M)) \| \to \text{min} \]
- Use cost function in Heuristic
Propose heuristic to avoid too many combination
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

Tania Malik (UCD HCL)

Topology-aware Communication Optimization

IPDPS 2014
Heuristic for the Communication-Optimal Arrangement

G_id= 0
- p0
- p1
- p2

G_id= 1
- p4
- P5
- P6
- p7

G_id= 2
- p8
- P9
- P10
- P11
- p12
Heuristic for the Communication-Optimal Arrangement

For each column i=1 to c
Group rectangle by clusters
Heuristic for the Communication-Optimal Arrangement

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

$G_{id}=0$
- $p0$
- $p1$
- $p2$

$G_{id}=1$
- $p4$
- $P5$
- $P6$
- $p7$

$G_{id}=2$
- $p8$
- $P9$
- $P10$
- $P11$
- $p12$
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

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

Figure:
Permutation order $k=1$
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

- 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

- 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

- 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

- 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 $k_1=45$ and $k_2=35$
- Add minimum $k$ to resulting arrangement

Figure: Permutation order $k=1$
Heuristic for the Communication-Optimal Arrangement-3

- 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 $k=1$
Repeat the same steps for all $c$ column

Figure: Permutation order $k=2$
Heuristic for the Communication-Optimal Arrangement-3

Repeat the same steps for all $c$ column

Figure: Permutation order $k=1$
Heuristic for the Communication-Optimal Arrangement-3

- Repeat the same steps for all $c$ column

Figure: Permutation order $k=1$

Figure: Permutation order $k=2$
Heuristic for the Communication-Optimal Arrangement-3

- Repeat the same steps for all $c$ column
- Cost function of $k_1=74$ and $k_2=65$
- Choose $k_2$ as optimal order

Figure: Permutation order $k=1$

Figure: Permutation order $k=2$
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

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Cost Orig</th>
<th>Cost Heuristic</th>
<th>Exec time (sec) Orig</th>
<th>Exec time (sec) Heuristic</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>533</td>
<td>432</td>
<td>58.00</td>
<td>42.58</td>
<td>1.36</td>
</tr>
<tr>
<td>32</td>
<td>868</td>
<td>710</td>
<td>119.30</td>
<td>88.30</td>
<td>1.35</td>
</tr>
<tr>
<td>90</td>
<td>1719</td>
<td>1263</td>
<td>400.80</td>
<td>297.83</td>
<td>1.34</td>
</tr>
</tbody>
</table>
### Experimental Result

#### Homogeneous Inter-Node Experiment

**Figure:** Partitioning for 4 homogeneous multi-core nodes

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Cost</th>
<th>Exec time (sec)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orig</td>
<td>336</td>
<td>199</td>
<td>3.85</td>
</tr>
<tr>
<td>Heuristic</td>
<td>3.17</td>
<td>1.21</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Homogeneous inter-node experimental results

---

Tania Malik (UCD HCL)
Homogeneous Inter-Node Experiment

Figure: Partitioning for 4 homogeneous multi-core nodes
Experimental Result

Homogeneous Inter-Node Experiment

Figure: Partitioning for 4 homogeneous multi-core nodes
Experimental Result

Homogeneous Inter-Node Experiment

Figure: Partitioning for 4 homogeneous multi-core nodes

Table: Homogeneous inter-node experimental results

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Cost Orig</th>
<th>Cost Heuristic</th>
<th>Exec time (sec) Orig</th>
<th>Exec time (sec) Heuristic</th>
<th>Ratio Orig/Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>336</td>
<td>199</td>
<td>3.85</td>
<td>3.17</td>
<td>1.21</td>
</tr>
</tbody>
</table>
Heuristic approach for combinatorial problem
Prediction is based on topology and Communication flow
Minimize inter-cluster communication cost