Hierarchical Parallel Matrix Multiplication on Large-Scale Distributed Memory Platforms

Jean-Noël Quintin, Khalid Hasanov, Alexey Lastovetsky

Heterogeneous Computing Laboratory
School of Computer Science and Informatics, University College Dublin,
Belfield, Dublin 4, Ireland
http://hcl.ucd.ie

2013

Outline

Outline

Outline

Motivation

- Majority of HPC algorithms for scientific applications were introduced between 1970s and 1990s
- They were designed for and tested on up to hundreds (few thousands at most) of processors.
  - In June 1995, the number of cores in the top 10 supercomputers ranged from 42 to 3680 (see http://www.top500.org/)

Motivation

- Majority of HPC algorithms for scientific applications were introduced between 1970s and 1990s
- They were designed for and tested on up to hundreds (few thousands at most) of processors.
  - In June 1995, the number of cores in the top 10 supercomputers ranged from 42 to 3680 (see http://www.top500.org/)
- Nowadays, in June 2013, this number ranges from 147,456 to 3,120,000

Motivation

The increasing scale of the HPC platforms creates new research questions which needs to be solved:

- Scalability
- Communication cost
- Energy efficiency
- etc.

Introduction

We focus on the communication cost of scientific applications on large-scale distributed memory platforms.

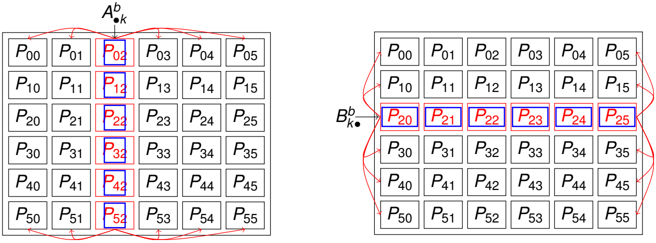
- Example application: Parallel Matrix Multiplication.
- Why Matrix Multiplication?
  - Matrix multiplication is important in its own rights as a computational kernel of many scientific applications.
  - It is a popular representative for other scientific applications
  - If an optimization method works well for matrix multiplication, it will also work well for many other relative scientific applications

Introduction

- Example algorithm:
  - SUMMA - Scalable Universal Matrix Multiplication Algorithm.
  - Introduced by Robert A. van de Geijn and Jerrell Watts. University of Texas at Austin, 1995.
  - Implemented in ScaLAPACK.

Outline

SUMMA



- Number of steps: $\frac{n}{b}$ ($n \times n$ - matrices, b - block size, $\sqrt{P} \times \sqrt{P}$ - processors grid, $P = 36$)
- The pivot column $A^b_{\bullet k}$ of $\frac{n}{\sqrt{P}} \times b$ blocks of matrix A is broadcast horizontally.
- The pivot row $B^b_{k\bullet}$ of $b \times \frac{n}{\sqrt{P}}$ blocks of matrix B is broadcast vertically.
- Then, each $\frac{n}{\sqrt{P}} \times \frac{n}{\sqrt{P}}$ block c_{ij} of matrix C is updated, $c_{ij} = c_{ij} + a_{ik} \times b_{kj}$.
- Size of data broadcast vertically and horizontally in each step: $2\frac{n}{\sqrt{P}} \times b$

Outline

Our Contribution

- We introduce application level hierarchical optimization of SUMMA
- Hierarchical SUMMA (HSUMMA) is platform independent optimization of SUMMA
- We theoretically and experimentally show that HSUMMA reduces the communication cost of SUMMA

SUMMA vs HSUMMA. Arrangement of Processors

P_{00}	P_{01}	P_{02}	P_{03}	P_{04}	P_{05}
P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
P_{20}	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}
P_{30}	P_{31}	P_{32}	P_{33}	P_{34}	P_{35}
P_{40}	P_{41}	P_{42}	P_{43}	P_{44}	P_{45}
P_{50}	P_{51}	P_{52}	P_{53}	P_{54}	P_{55}

SUMMA

SUMMA vs HSUMMA. Arrangement of Processors

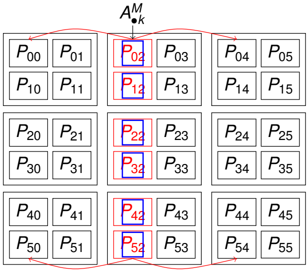
P_{00}	P_{01}	P_{02}	P_{03}	P_{04}	P_{05}
P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
P_{20}	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}
P_{30}	P_{31}	P_{32}	P_{33}	P_{34}	P_{35}
P_{40}	P_{41}	P_{42}	P_{43}	P_{44}	P_{45}
P_{50}	P_{51}	P_{52}	P_{53}	P_{54}	P_{55}

SUMMA

P_{00}	P_{01}	P_{02}	P_{03}	P_{04}	P_{05}
P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
P_{20}	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}
P_{30}	P_{31}	P_{32}	P_{33}	P_{34}	P_{35}
P_{40}	P_{41}	P_{42}	P_{43}	P_{44}	P_{45}
P_{50}	P_{51}	P_{52}	P_{53}	P_{54}	P_{55}

HSUMMA

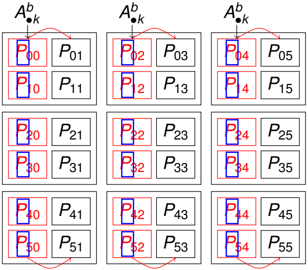
Horizontal Communications Between Groups in HSUMMA



- P - number of processors ($P = 36$)
- G - number of groups ($G = 9$)
- $\sqrt{P} \times \sqrt{P}$ - processors grid
- $\sqrt{G} \times \sqrt{G}$ - grid of processor groups
- M - block size between groups
- n/M - number of steps
- Size of data broadcast horizontally in each step: $\frac{n \times M}{\sqrt{P}}$

The pivot column $A_{\bullet k}^{M}$ of $\frac{n}{\sqrt{P}} \times M$ blocks of matrix A is broadcast horizontally between groups

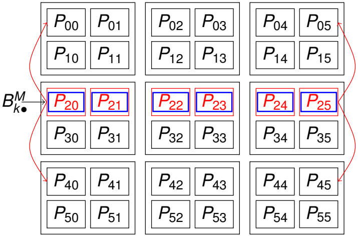
Horizontal Communications Inside Groups in HSUMMA



- $\frac{\sqrt{P}}{\sqrt{G}} \times \frac{\sqrt{P}}{\sqrt{G}}-$ grid of processors inside groups
- $b-$ block size inside one group
- $M/b-$ steps inside one group
- $n/M-$ steps between groups
- Size of data broadcast horizontally in each step: $\frac{n \times b}{\sqrt{P}}$

Upon receipt of the pivot column data from the other groups, the local pivot column $A^b_{\bullet k}$, $(b \leq M)$ of $\frac{n}{\sqrt{P}} \times b$ blocks of matrix A is broadcast horizontally inside each group

Vertical Communications Between Groups in HSUMMA



- P - number of processors ($P = 36$)
- G - number of groups ($G = 9$)
- $\sqrt{P} \times \sqrt{P}$ - processors grid
- $\sqrt{G} \times \sqrt{G}$ - grid of processor groups
- M - block size between groups
- n/M - number of steps
- Size of data broadcast vertically in each step: $\frac{n \times M}{\sqrt{P}}$

The pivot row $B_{k\bullet}^{M}$ of $M \times \frac{n}{\sqrt{P}}$ blocks of matrix B is broadcast vertically between groups

Vertical Communications Inside Groups in HSUMMA

- $\frac{\sqrt{P}}{\sqrt{G}} \times \frac{\sqrt{P}}{\sqrt{G}}-$ grid of processors
- $b-$ block size inside one group
- $M/b-$ steps inside one group
- $n/M-$ steps between groups
- Size of data broadcast vertically in each step: $\frac{n \times b}{\sqrt{P}}$

Upon receipt of the pivot row data from the other groups, the local pivot row $B^b_{\bullet k}$ of $b \times \frac{n}{\sqrt{P}}$, $(b \leq M)$ blocks of matrix B is broadcast vertically inside each group

Communication Model to Analyse SUMMA and HSUMMA

Time of sending of a message of size m between two processors:

$$\alpha + m\beta \qquad (1)$$

Here,

- α -latency
- β -reciprocal bandwith
- m -message size

General Broadcast Model to Analyse SUMMA and HSUMMA

We use a general broadcast model for all homogeneous broadcast algorithms such as

- flat
- binary
- binomial
- linear
- scatter-allgather broadcast

$$T_{bcast}(m,p) = L(p)\times\alpha + m\times W(p)\times\beta \qquad (2)$$

General Broadcast Model

$$T_{bcast}(m,p) = L(p)\times\alpha + m\times W(p)\times\beta$$

Assumptions:

- $L(1) = 0$ and $W(1) = 0$
- $L(p)$ and $W(p)$ are monotonic and differentiable functions in the interval $(1, p)$,
- their first derivatives are constants or monotonic in the interval $(1, p)$

SUMMA and HSUMMA with General Broadcast Model

- SUMMA:

$$T_S(n,p) = 2\left(\frac{n}{b}\times L(\sqrt{p})\alpha + \frac{n^2}{\sqrt{p}}\times W(\sqrt{p})\beta\right) \quad (3)$$

- HSUMMA:

$$T_{HS}(n,p,G) = T_{HS_l}(n,p,G) + T_{HS_b}(n,p,G) \quad (4)$$

Here $G \in [1,p]$ and we take $b = M$ for simplicity and

- T_{HS_l} is the latency cost:

$$T_{HS_l}(n,p,G) = 2\frac{n}{b}\times\left(L(\sqrt{G}) + L(\frac{\sqrt{p}}{\sqrt{G}})\right)\alpha \quad (5)$$

- T_{HS_b} is the bandwidth cost:

$$T_{HS_b}(n,p,G) = 2\frac{n^2}{\sqrt{p}}\times\left(W(\sqrt{G}) + W(\frac{\sqrt{p}}{\sqrt{G}})\right)\beta \quad (6)$$

SUMMA is a special case of HSUMMA when $G = 1$ or $G = p$.

Optimal Number of Groups in HSUMMA with General Broadcast Model

Derivative of the communication cost function of HSUMMA with general broadcast model:

$$\frac{\partial T_{HS}}{\partial G} = \frac{n}{b} \times L_1(p, G)\alpha + \frac{n^2}{\sqrt{p}} \times W_1(p, G)\beta \tag{7}$$

Here, $L_1(p, G)$ and $W_1(p, G)$ are defined as follows:

$$L_1(p, G) = \left(\frac{\partial L(\sqrt{G})}{\partial \sqrt{G}} \times \frac{1}{\sqrt{G}} - \frac{\partial L(\frac{\sqrt{p}}{\sqrt{G}})}{\partial \frac{\sqrt{p}}{\sqrt{G}}} \times \frac{\sqrt{p}}{G\sqrt{G}} \right) \tag{8}$$

$$W_1(p, G) = \left(\frac{\partial W(\sqrt{G})}{\partial \sqrt{G}} \times \frac{1}{\sqrt{G}} - \frac{\partial W(\frac{\sqrt{p}}{\sqrt{G}})}{\partial \frac{\sqrt{p}}{\sqrt{G}}} \times \frac{\sqrt{p}}{G\sqrt{G}} \right) \tag{9}$$

If $G = \sqrt{P}$ then $L_1(p, G) = 0$ and $W_1(p, G) = 0$. Thus, $\frac{\partial T_{HS}}{\partial G} = 0$

Optimal Number of Groups in HSUMMA with General Broadcast Model

- HSUMMA has extremum in $G \in (1, P)$
- $G = \sqrt{P}$ is the extremum point.
- Depending on α and β:
  - This extremum can be minimum which means HSUMMA always outperforms SUMMA.
  - Or maximum which means HSUMMA has the same performance as SUMMA.

Theoretical Prediction by Using Scatter-Allgather Broadcast

Algorithm	Comp. Cost	Latency Factor		Bandwidth Factor	
		inside groups	between groups	inside groups	between groups
SUMMA	$\frac{2n^3}{p}$	$(\log_2(p) + 2(\sqrt{p}-1)) \times \frac{n}{b}$		$4\left(1-\frac{1}{\sqrt{p}}\right) \times \frac{n^2}{\sqrt{p}}$	
HSUMMA	$\frac{2n^3}{p}$	$\left(\log_2\left(\frac{p}{G}\right) + 2\left(\frac{\sqrt{p}}{\sqrt{G}}-1\right)\right) \times \frac{n}{b}$	$\left(\log_2(G) + 2\left(\sqrt{G}-1\right)\right) \times \frac{n}{M}$	$4\left(1-\frac{\sqrt{G}}{\sqrt{p}}\right) \times \frac{n^2}{\sqrt{p}}$	$4\left(1-\frac{1}{\sqrt{G}}\right) \times \frac{n^2}{\sqrt{p}}$

Optimal Number of Groups with Scatter-Allgather Broadcast

$$\frac{\partial T_{HS_V}}{\partial G} = \frac{G - \sqrt{p}}{G\sqrt{G}} \times \left(\frac{n\alpha}{b} - 2\frac{n^2}{p} \times \beta \right) \tag{10}$$

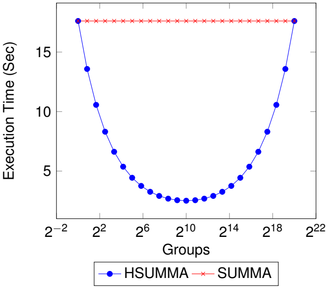
If $G = \sqrt{p}$ then $\frac{\partial T_{HS_V}}{\partial G} = 0$.

- If $\frac{\alpha}{\beta} > 2\frac{nb}{p}$ then $G = \sqrt{p}$ is the minimum of T_{HS}.
- If $\frac{\alpha}{\beta} < 2\frac{nb}{p}$ then $G = \sqrt{p}$ is the maximum of T_{HS}. In this case the function gets its minimum at either $G = 1$ or $G = p$.

Optimal Number of Groups with Scatter-Allgather Broadcast

Algorithm	Comp. Cost	Latency Factor: inside groups	Latency Factor: between groups	Bandwidth Factor: inside groups	Bandwidth Factor: between groups
SUMMA	$\frac{2n^3}{p}$	$(\log_2(p) + 2(\sqrt{p}-1)) \times \frac{n}{b}$		$4\left(1-\frac{1}{\sqrt{p}}\right) \times \frac{n^2}{\sqrt{p}}$	
HSUMMA	$\frac{2n^3}{p}$	$\left(\log_2\left(\frac{p}{G}\right) + 2\left(\frac{\sqrt{p}}{\sqrt{G}}-1\right)\right) \times \frac{n}{b}$	$\left(\log_2(G) + 2\left(\sqrt{G}-1\right)\right) \times \frac{n}{M}$	$4\left(1-\frac{\sqrt{G}}{\sqrt{p}}\right) \times \frac{n^2}{\sqrt{p}}$	$4\left(1-\frac{1}{\sqrt{G}}\right) \times \frac{n^2}{\sqrt{p}}$
HSUMMA($G=\sqrt{p}, b=M$)	$\frac{2n^3}{p}$	$(\log_2(p) + 4(\sqrt[4]{p}-1)) \times \frac{n}{b}$		$8\left(1-\frac{1}{\sqrt[4]{p}}\right) \times \frac{n^2}{\sqrt{p}}$	

Theoretical Prediction on Future Exascale Platforms by Using Scatter-Allgather Broadcast



- Total flop rate (γ): $1E18$ flops
- Latency: 500 ns,
- Bandwidth: 100 GB/s
- Problem size: $n = 2^{22}$,
- Number of processors: $p = 2^{20}$
- Block size: $b = M = 256$

Prediction of SUMMA and HSUMMA on Exascale.
(The parameters were taken from: Report on Exascale Architecture. IESP Meeting. April 12, 2012)

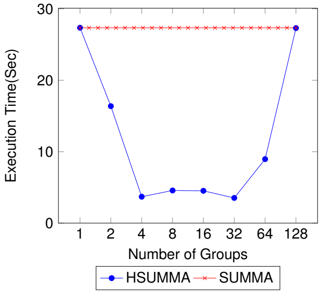
Outline

Experimental platforms

- The experiments were carried out on *Graphene cluster of Nancy site of Grid5000* platform,
- On *8, 16, 32, 64 and 128* cores and
- On IBM BlueGene on *1024, 2048, 4096, 8192 and 16384* cores

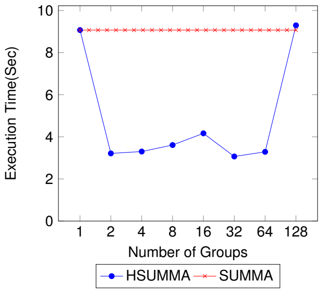
Outline

Summa vs HSUMMA on Grid5000 with MPICH



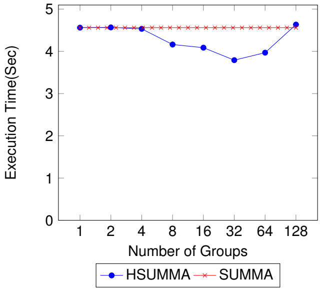
HSUMMA and SUMMA on Grid5000 with MPICH-2.
$b = M = 64, n = 8192$ and $p = 128$.
7.75 times reduction of the execution time.

Summa vs HSUMMA on Grid5000 with MPICH



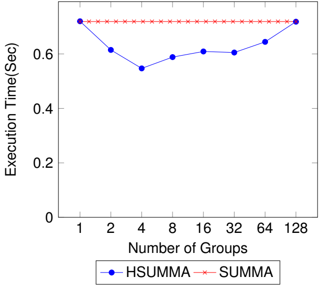
HSUMMA and SUMMA on Grid5000 with MPICH-2.
$b = M = 256$, $n = 8192$ and $p = 128$.
2.96 times reduction of the execution time.

Summa vs HSUMMA on Grid5000 with OpenMPI on Ethernet



HSUMMA and SUMMA on Grid5000 with OpenMPI on Ethernet. $b = M = 256$, $n = 8192$ and $p = 128$.
16.8 percent reduction of the execution time.

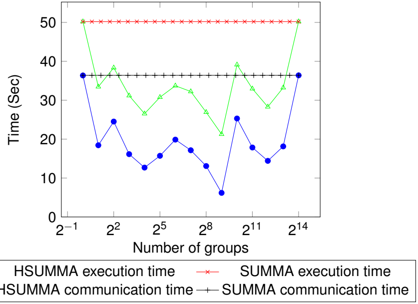
Summa vs HSUMMA on Grid5000 with OpenMPI on Infiniband



HSUMMA and SUMMA on Grid5000 with OpenMPI on Infiniband. $b = M = 256$, $n = 8192$ and $p = 128$.
24 percent reduction of the execution time.

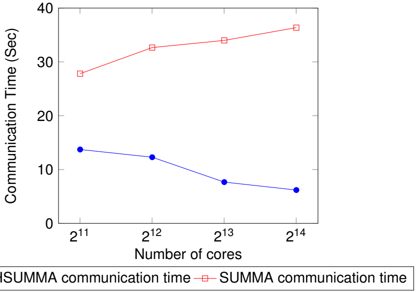
Outline

Summa vs HSUMMA on BlueGene



SUMMA and HSUMMA on BG/P. Execution and communication time. $b = M = 256$, $n = 65536$ and $p = 16384$. 2.08 times reduction of the execution time. 5.89 times reduction of the communication time.

SUMMA and HSUMMA Communication Time



SUMMA and HSUMMA on BG/P. Communication time.
$b = M = 256$ and $n = 65536$

Summary

Improvement over SUMMA:

- Hierarchical SUMMA has theoretically better communication time and thus less execution time than SUMMA
- 2.08 times less communication time on 2048 cores
- 5.89 times less communication time on 16384 cores
- 1.2 times less overall execution time on 2048 cores
- 2.36 times less overall execution time on 16384 cores

Questions?