Communication Optimal Algorithms for Multiple Tensor-Times-Matrix Computation

Hussam Al Daas[1], Grey Ballard[2], Laura Grigori[3], Suraj Kumar[3], and Kathryn Rouse[4]

[1]Rutherford Appleton Laboratory, UK

[2]Wake Forest University, USA

[3]Inria Paris, France

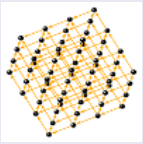
[4]Inmar Intelligence, USA

ROMA Working Group
December 7, 2021

My Past Experience

Parallelization in Polyhedral Model

- Linked-list operations
- Improved spatial locality
- Parallelization using OpenMP



Seismic Imaging on GPUs

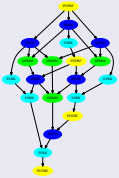
$$H_1 = \sin^2\theta\cos^2\phi\frac{\partial^2}{\partial x^2} + \sin^2\theta\sin^2\phi\frac{\partial^2}{\partial y^2} + \cos^2\theta\frac{\partial^2}{\partial z^2} + \sin^2\theta\sin 2\phi\frac{\partial^2}{\partial x\partial y} + \sin 2\theta\sin\phi\frac{\partial^2}{\partial y\partial z} + \sin 2\theta\cos\phi\frac{\partial^2}{\partial x\partial z}$$

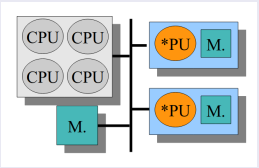
$$H_2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} - H_1$$

Schedulers for Blue Gene Supercomputers

- GASNET API
- Unbalanced Tree Search benchmark
- Comparison to Charm++

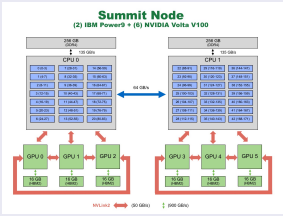
Scheduling on Heterogeneous Platforms





Molecular Simulations on Supercomputers

- NWChemEx Project
- TAMM library
- Hartree Fock and CCSD applications



Communication Optimal Algorithms for Multiple Tensor-Times-Matrix Computation

Hussam Al Daas[1], Grey Ballard[2], Laura Grigori[3], Suraj Kumar[3], and Kathryn Rouse[4]

[1]Rutherford Appleton Laboratory, UK

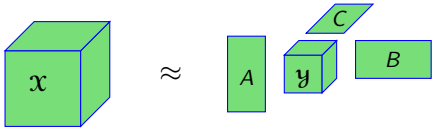
[2]Wake Forest University, USA

[3]Inria Paris, France

[4]Inmar Intelligence, USA

ROMA Working Group
December 7, 2021

Higher-order SVD (HOSVD) to compute Tucker decomposition



Algorithm 1 HOSVD Algorithm($\mathcal{X}$, R_1, R_2, R_3)

1: $A \leftarrow R_1$ left singular vectors of $\mathcal{X}_{(1)}$
2: $B \leftarrow R_2$ left singular vectors of $\mathcal{X}_{(2)}$
3: $C \leftarrow R_3$ left singular vectors of $\mathcal{X}_{(3)}$
4: $\mathcal{Y} = \mathcal{X} \times_1 A^\mathsf{T} \times_2 B^\mathsf{T} \times_3 C^\mathsf{T}$
5: Return $\mathcal{Y}$, A, B, C

- $\mathcal{X}$, $\mathcal{Y}$: 3-dimensional input and output tensors (or arrays) & A, B, C: matrices
- $\mathcal{X}_{(i)}$: matricization of $\mathcal{X}$ (ith dimension represents rows and remaining dimensions represent columns)
- Multiple Tensor-Times-Matrix (Multi-TTM) computation: $\mathcal{Y} = \mathcal{X} \times_1 A^\mathsf{T} \times_2 B^\mathsf{T} \times_3 C^\mathsf{T}$
  - To obtain full tensor, $\mathcal{X} = \mathcal{Y} \times_1 A \times_2 B \times_3 C$

Outline (Lower Bounds and Communication Optimal Algorithms)

Assumptions

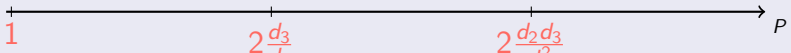
- P number of processors
- Each processor performs (asymptotically) equal amount of operations
- No redundant operations
- One copy of data is in the system
  - $1/P$th amount of inputs (before the computation) and output (after the computation) on each processor
- Each processor has enough memory

Outline (Lower Bounds and Communication Optimal Algorithms)

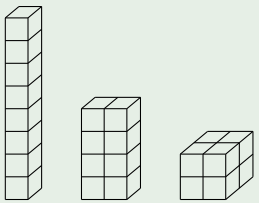
Existing Lower Bounds for Matrix Matrix Multiplications

- $C = AB$, where $A \in \mathbb{R}^{n_1 \times n_2}$, $B \in \mathbb{R}^{n_2 \times n_3}$, and $C \in \mathbb{R}^{n_1 \times n_3}$
- Let $d_1 = \min(n_1, n_2, n_3) \leq d_2 = median(n_1, n_2, n_3) \leq d_3 = \max(n_1, n_2, n_3)$

Existing Communication Lower Bounds (CARMA [IPDPS 2013])



Arrangements of 8 processors

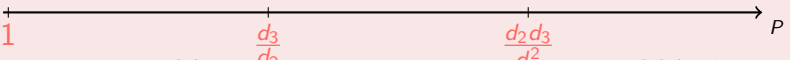


Existing Lower Bounds for Matrix Matrix Multiplications

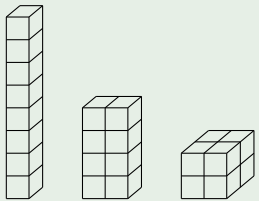
- $C = AB$, where $A \in \mathbb{R}^{n_1 \times n_2}$, $B \in \mathbb{R}^{n_2 \times n_3}$, and $C \in \mathbb{R}^{n_1 \times n_3}$
- Let $d_1 = \min(n_1, n_2, n_3) \leq d_2 = median(n_1, n_2, n_3) \leq d_3 = \max(n_1, n_2, n_3)$

Existing Communication Lower Bounds (CARMA [IPDPS 2013])

Our Communication Lower Bounds



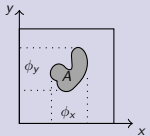
Arrangements of 8 processors



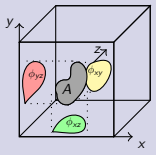
Loomis-Whitney & Hölder-Brascamp-Lieb inequalities

Size of $d-1$ dimensional projections (Loomis-Whitney inequalitiy)

- 2-dimensional object A and its 1-dimensional projections ϕ_x, ϕ_y
- $\phi_x \phi_y \geq Area(A)$



- 3-dimensional object A and its 2-dimensional projections: ϕ_{xy}, ϕ_{yz}, ϕ_{xz}
- $(\phi_{xy}\phi_{yz}\phi_{xz})^{\frac{1}{3-1}} \geq Volume(A)$



Hölder-Brascamp-Lieb (HBL) inequality – Generalization of Loomis-Whitney inequality

$$\mathbf{\Delta} = \begin{array}{c} \\ i \\ j \\ k \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$$

for $i = 1{:}n_1$, for $k = 1{:}n_2$, for $j = 1{:}n_3$

$$C[i][j] += A[i][k] * B[k][j]$$

- Find $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$ such that $\mathbf{\Delta}.\mathbf{x} \geq \mathbf{1}$, $\mathbf{1}$ is vector of all ones
- ϕ_A, ϕ_B, ϕ_C: projections of computations on arrays A, B, C
- HBL inequality: $\phi_A^{x_1} \phi_B^{x_2} \phi_C^{x_3} \geq$ Amount of computations
- To make inequality tight select $\mathbf{x}$ such that $\mathbf{1}^T\mathbf{x}$ is minimum $=> x_1 = x_2 = x_3 = \frac{1}{2}$

Constraints for Matrix Multiplications

$$\text{for } i = 1{:}n_1, \text{ for } k = 1{:}n_2, \text{ for } j = 1{:}n_3$$
$$C[i][j] += A[i][k] * B[k][j]$$

- Total number of multiplications $= n_1 n_2 n_3$
- Consider a processor which performs $\frac{n_1 n_2 n_3}{P}$ amount of multiplications
- Optimization problem: $\textit{Minimize } \phi_A + \phi_B + \phi_C$ s.t.

$$\phi_A^{\frac{1}{2}} \phi_B^{\frac{1}{2}} \phi_C^{\frac{1}{2}} \geq \frac{n_1 n_2 n_3}{P}$$

Extra constraints (our contributions)

- Each element of A (resp. B) is involved in n_3 (resp. n_1) multiplications
  - To perform at least $\frac{n_1 n_2 n_3}{P}$ multiplications: $\phi_A \geq \frac{n_1 n_2}{P}, \phi_B \geq \frac{n_2 n_3}{P}$
- Each element of C is the sum of n_2 multiplications, therefore $\phi_C \geq \frac{n_1 n_3}{P}$
- Projections can be at max the size of the arrays: $\phi_A \leq n_1 n_2, \phi_B \leq n_2 n_3, \phi_C \leq n_1 n_3$

Optimization Problem to Compute Communication Lower Bounds

- Projections (ϕ_A, ϕ_B, ϕ_C) indicate the amount of array access
- Communication lower bound $= \phi_A + \phi_B + \phi_C -$ data owned by the processor

$$\text{Minimize } \phi_A + \phi_B + \phi_C \quad \text{s.t.}$$

$$\phi_A^{\frac{1}{2}} \phi_B^{\frac{1}{2}} \phi_C^{\frac{1}{2}} \geq \frac{n_1 n_2 n_3}{P}$$

$$\frac{n_1 n_2}{P} \leq \phi_A \leq n_1 n_2$$

$$\frac{n_2 n_3}{P} \leq \phi_B \leq n_2 n_3$$

$$\frac{n_1 n_3}{P} \leq \phi_C \leq n_1 n_3$$

Generalized version (in terms of d_1, d_2, d_3)

$$\text{Minimize } \phi_1 + \phi_2 + \phi_3 \quad \text{s.t.}$$

$$\phi_1^{\frac{1}{2}} \phi_2^{\frac{1}{2}} \phi_3^{\frac{1}{2}} \geq \frac{d_1 d_2 d_3}{P}$$

$$\frac{d_1 d_2}{P} \leq \phi_1 \leq d_1 d_2$$

$$\frac{d_1 d_3}{P} \leq \phi_2 \leq d_1 d_3$$

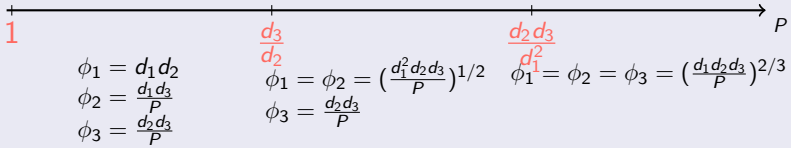
$$\frac{d_2 d_3}{P} \leq \phi_3 \leq d_2 d_3$$

$$d_1 \leq d_2 \leq d_3$$

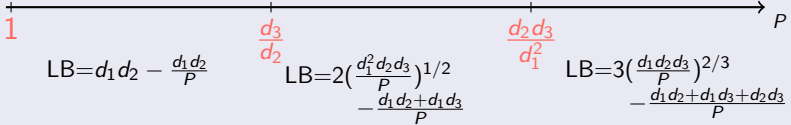
Amount of Accesses and Communication Lower bounds

- Estimate the solution based on Lagrange multipliers
- Prove optimality using all KarushKuhnTucker (KKT) conditions are satisfied

Amount of accesses $=\phi_1 + \phi_2 + \phi_3$



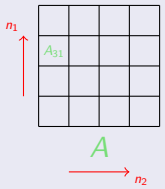
Communication Lower Bounds (Amount of Data Transfers)



Design of Communication Optimal Algorithms

Data Distribution (P is organized into a $p_1 \times p_2 \times p_3$ grid)

- p_1, p_2, and p_3 evenly distribute n_1, n_2, and n_3
- Each processor has $\frac{1}{P}$th amount of input and output variables
- $A_{31} = A(2\frac{n_1}{p_1} + 1 : 3\frac{n_1}{p_1}, 1 : \frac{n_2}{p_2})$ is evenly distributed among $(3, 1, *)$ processors
- $B_{12} = B(1 : \frac{n_2}{p_2}, \frac{n_3}{p_3} + 1 : 2\frac{n_3}{p_3})$ is evenly distributed among $(*, 1, 2)$ processors



Algorithm 2 $C = AB$ Matrix Multiplication Algorithm

1: (p_1', p_2', p_3') is my processor id
2: //All-gather input matrices A and B
3: $A_{p_1'p_2'}$ = All-Gather(A, $(p_1', p_2', *)$)
4: $B_{p_2'p_3'}$ = All-Gather(B, $(*, p_2', p_3')$)
5: T = Local-Matrix-Multiplication($A_{p_1'p_2'}, B_{p_2'p_3'}$) // Local matrix multiplication in a temporary
6: Reduce-Scatter($C_{p_1'p_3'}$, T, $(p_1', *, p_3')$) // Reduce-scatter the output

Cost Analysis and Open Questions

Cost Analysis

- Total amount of multiplications per processor $= \frac{n_1 n_2 n_3}{p_1 p_2 p_3} = \frac{n_1 n_2 n_3}{P}$
- Total data transfers $= \frac{n_1 n_2}{p_1 p_2} + \frac{n_2 n_3}{p_2 p_3} + \frac{n_1 n_3}{p_1 p_3} - \frac{n_1 n_2 + n_2 n_3 + n_1 n_3}{P}$

Open Questions

- How to select p_1, p_2, p_3?
- Are communication lower bounds are achievable for all matrix dimensions?

Outline

1. For Matrix Matrix Multiplications

2. For Multi-TTM Computation
   - Simulated Experiments
   - Conclusion

Multi-TTM Computation

- $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$: input tensor, $\mathcal{Y} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$: output tensor
- $\mathbf{A}^{(k)} \in \mathbb{R}^{n_k \times r_k}$: factor matrix of the kth mode

2-dimensional Multi-TTM ($\mathcal{Y} = \mathcal{X} \times_1 \mathbf{A}^{(1)^\mathsf{T}} \times_2 \mathbf{A}^{(2)^\mathsf{T}}$)

TTM-in-Sequence approach (used in Tucker-MPI): $\mathbf{Y} = \mathbf{A}^{(1)\mathsf{T}}\mathbf{X}\mathbf{A}^{(2)}$

All-at-Once approach (our contribution):

for $n_1' = 1{:}n_1$, for $n_2' = 1{:}n_2$

for $r_1' = 1{:}r_1$, for $r_2' = 1{:}r_2$

$$\mathcal{Y}(r_1', r_2') = \mathcal{Y}(r_1', r_2') + \left(\mathcal{X}(n_1', n_2') * \mathbf{A}^{(1)}(n_1', r_1') * \mathbf{A}^{(2)}(n_2', r_2')\right)$$

3-dimensional Multi-TTM ($\mathcal{Y} = \mathcal{X} \times_1 \mathbf{A}^{(1)^\mathsf{T}} \times_2 \mathbf{A}^{(2)^\mathsf{T}} \times_3 \mathbf{A}^{(3)^\mathsf{T}}$)

All-at-Once 3-dimensional Multi-TTM Computation

for $n_1' = 1{:}n_1$, for $n_2' = 1{:}n_2$, for $n_3' = 1{:}n_3$
  for $r_1' = 1{:}r_1$, for $r_2' = 1{:}r_2$, for $r_3' = 1{:}r_3$

$$\mathcal{Y}(r_1', r_2', r_3') = \mathcal{Y}(r_1', r_2', r_3') + \left(\mathcal{X}(n_1', n_2', n_3') * \mathbf{A}^{(1)}(n_1', r_1') * \mathbf{A}^{(2)}(n_2', r_2') * \mathbf{A}^{(3)}(n_3', r_3')\right)$$

$$\mathbf{\Delta} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{1}_3 & \mathbf{0}_3 \\ \mathbf{I}_{3\times3} & \mathbf{0}_3 & \mathbf{1}_3 \end{bmatrix}$$

- Total number of $4-$ *array* operations $= n_1 r_1 n_2 r_2 n_3 r_3$
- $\mathbf{\Delta}$ is not full rank
  - Consider each vector $\mathbf{x}$ such that $\mathbf{\Delta} \cdot \mathbf{x} = \mathbf{1}$, $\mathbf{x}$ is of the form $[a\ a\ a\ 1\text{-}a\ 1\text{-}a]^\mathsf{T}$ and $0 \leq a \leq 1$
- $\phi_\mathcal{X}$, $\phi_\mathcal{Y}$: tensor projections & ϕ_1, ϕ_2, ϕ_3: matrix projections
- From HBL, $\phi_\mathcal{X}^{1-a}\phi_\mathcal{Y}^{1-a}\phi_1^a\phi_2^a\phi_3^a \geq$ Amount of computations

Solving Optimization Problem to Compute Lower Bounds

- Select a processor which performs $\frac{n_1 r_1 n_2 r_2 n_3 r_3}{P}$ amount of $4-$ *array* operations
- After applying lower and upper bounds for each projection, we need to solve the following optimization problem

$$\text{Minimize } \phi_{\mathcal{X}} + \phi_{\mathcal{Y}} + \phi_1 + \phi_2 + \phi_3 \text{ s.t.}$$

$$\phi_{\mathcal{X}}^{1-a} \phi_{\mathcal{Y}}^{1-a} \phi_1^a \phi_2^a \phi_3^a \geq \frac{n_1 r_1 n_2 r_2 n_3 r_3}{P}$$

$$\frac{n_1 n_2 n_3}{P} \leq \phi_{\mathcal{X}} \leq n_1 n_2 n_3$$

$$\frac{r_1 r_2 r_3}{P} \leq \phi_{\mathcal{Y}} \leq r_1 r_2 r_3$$

$$\frac{n_1 r_1}{P} \leq \phi_1 \leq n_1 r_1$$

$$\frac{n_2 r_2}{P} \leq \phi_2 \leq n_2 r_2$$

$$\frac{n_3 r_3}{P} \leq \phi_3 \leq n_3 r_3$$

Divide the problem into two parts

Matrix part

$$\text{Minimize } \phi_1 + \phi_2 + \phi_3 \quad \text{s.t.}$$

$$\phi_1\phi_2\phi_3 \geq \frac{n_1 r_1 n_2 r_2 n_3 r_3}{P}$$

$$\frac{n_1 r_1}{P} \leq \phi_1 \leq n_1 r_1$$

$$\frac{n_2 r_2}{P} \leq \phi_2 \leq n_2 r_2$$

$$\frac{n_3 r_3}{P} \leq \phi_3 \leq n_3 r_3$$

Tensor part

$$\text{Minimize } \phi_{\mathcal{X}} + \phi_{\mathcal{Y}} \quad \text{s.t.}$$

$$\phi_{\mathcal{X}}\phi_{\mathcal{Y}} \geq \frac{n_1 r_1 n_2 r_2 n_3 r_3}{P}$$

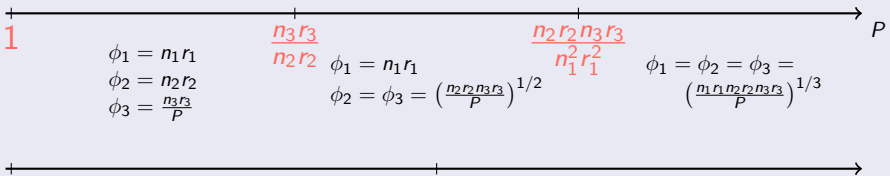
$$\frac{n_1 n_2 n_3}{P} \leq \phi_{\mathcal{X}} \leq n_1 n_2 n_3$$

$$\frac{r_1 r_2 r_3}{P} \leq \phi_{\mathcal{Y}} \leq r_1 r_2 r_3$$

Amount of Accesses and Lower bounds

- We assume $n_1 r_1 \leq n_2 r_2 \leq n_3 r_3$ and $r_1 r_2 r_3 \leq n_1 n_2 n_3$
- Estimate solutions for both parts using Lagrange multipliers (optimality can be proven using KKT conditions)

Amount of accesses $= \phi_{\mathcal{X}} + \phi_{\mathcal{Y}} + \phi_1 + \phi_2 + \phi_3$

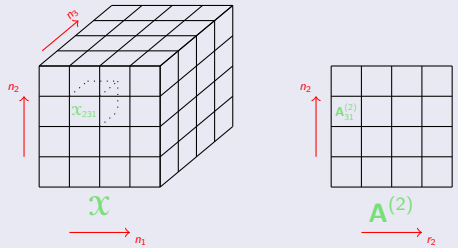


Communication lower bound $= \phi_{\mathcal{X}} + \phi_{\mathcal{Y}} + \phi_1 + \phi_2 + \phi_3 - \frac{n_1 n_2 n_3 + r_1 r_2 r_3 + n_1 r_1 + n_2 r_2 + n_3 r_3}{P}$

Design of Communication Optimal Algorithms

Data Distribution (P is organized into a $p_1 \times p_2 \times p_3 \times q_1 \times q_2 \times q_3$ grid)

- p_1, p_2, p_3, q_1, q_2, and q_3 evenly distribute n_1, n_2, n_3, r_1, r_2, and r_3
- Each processor has $\frac{1}{P}$th amount of input and output variables
- Subtensor $\mathcal{X}_{231} = \mathcal{X}(\frac{n_1}{p_1} + 1 : 2\frac{n_1}{p_1}, 2\frac{n_2}{p_2} + 1 : 3\frac{n_2}{p_2}, 1 : \frac{n_3}{p_3})$ is distributed evenly among processors $(2, 3, 1, *, *, *)$
- Submatrix $\mathbf{A}^{(2)}_{31} = \mathbf{A}^{(2)}(2\frac{n_2}{p_2} + 1 : 3\frac{n_2}{p_2}, 1 : \frac{r_2}{q_2})$ is distributed evenly among processors $(*, 3, *, *, 1, *)$



6-dimensional Algorithm to compute Multi-TTM

Algorithm 1 3-dimensional Parallel Atomic Multi-TTM

Require: $\mathcal{X}$, $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$, $\mathbf{A}^{(3)}$, $p_1 \times p_2 \times p_3 \times q_1 \times q_2 \times q_3$ logical processor grid

Ensure: $\mathcal{Y}$ such that $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{A}^{(1)^\mathsf{T}} \times_2 \mathbf{A}^{(2)^\mathsf{T}} \times_3 \mathbf{A}^{(3)^\mathsf{T}}$

1: $(p_1', p_2', p_3', q_1', q_2', q_3')$ is my processor id
2: //All-gather input tensor and matrices
3: $\mathcal{X}_{p_1'p_2'p_3'}$ = All-Gather($\mathcal{X}$, $(p_1', p_2', p_3', *, *, *)$)
4: $\mathbf{A}^{(1)}_{p_1'q_1'}$ = All-Gather($\mathbf{A}^{(1)}$, $(p_1', *, *, q_1', *, *)$)
5: $\mathbf{A}^{(2)}_{p_2'q_2'}$ = All-Gather($\mathbf{A}^{(2)}$, $(*, p_2', *, *, q_2', *)$)
6: $\mathbf{A}^{(3)}_{p_3'q_3'}$ = All-Gather($\mathbf{A}^{(3)}$, $(*, *, p_3', *, *, q_3')$)
7: //Perform local Multi-TTM computation in a temporary tensor $\mathcal{T}$
8: $\mathcal{T}$ = Local-Multi-TTM($\mathcal{X}_{p_1'p_2'p_3'}$, $\mathbf{A}^{(1)}_{p_1'q_1'}$, $\mathbf{A}^{(2)}_{p_2'q_2'}$, $\mathbf{A}^{(3)}_{p_3'q_3'}$)
9: //Reduce-scatter the output tensor in $\mathcal{Y}_{q_1'q_2'q_3'}$
10: Reduce-Scatter($\mathcal{Y}_{q_1'q_2'q_3'}$, $\mathcal{T}$, $(*, *, *, q_1', q_2', q_3')$)

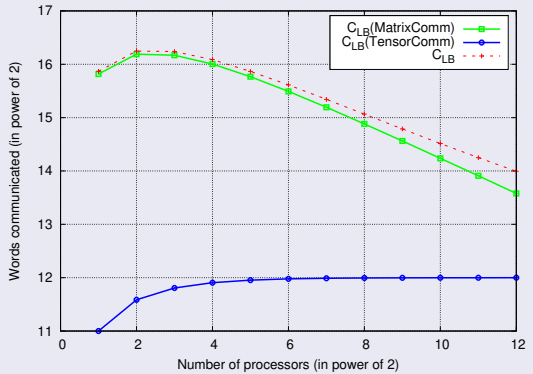
Cost Analysis of our Algorithm

- Total amount of 4-array operations per processor $= \frac{n_1 r_1 n_2 r_2 n_3 r_3}{p_1 q_1 p_2 q_2 p_3 q_3} = \frac{n_1 r_1 n_2 r_2 n_3 r_3}{P}$
- Data transfers happen only in All-Gather and Reduce-Scatter collective operations
  - Cost on Q processors is $(1 - \frac{1}{Q})w$, w is the amount of total data after All-Gather or before Reduce-Scatter operation
- Total data transfers $= \frac{n_1 n_2 n_3}{p_1 p_2 p_3} + \frac{r_1 r_2 r_3}{q_1 q_2 q_3} + \frac{n_1 r_1}{p_1 q_1} + \frac{n_2 r_2}{p_2 q_2} + \frac{n_3 r_3}{p_3 q_3} - \frac{n_1 n_2 n_3 + r_1 r_2 r_3 + n_1 r_1 + n_2 r_2 + n_3 r_3}{P}$
- p_1, p_2, p_3, q_1, q_2, and q_3 can be obtained based on lower bounds (Not today)

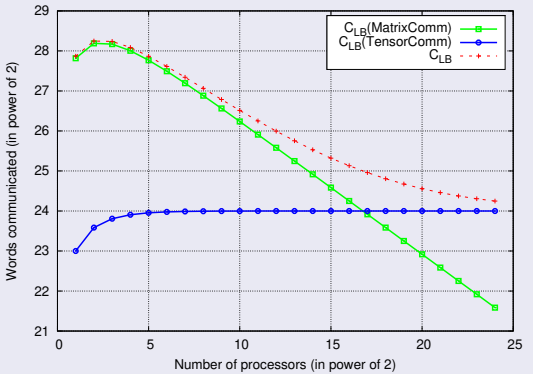
Outline

Communication Lower Bound (C_{LB}) distributions

$n_1 = n_2 = n_3 = 2^{12}, r_1 = r_2 = r_3 = 2^4$



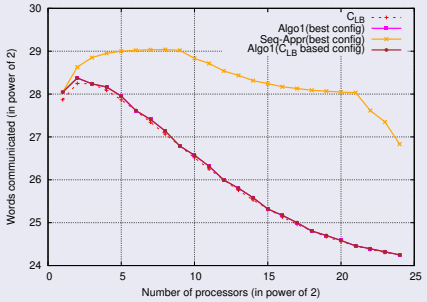
$n_1 = n_2 = n_3 = 2^{20}, r_1 = r_2 = r_3 = 2^8$



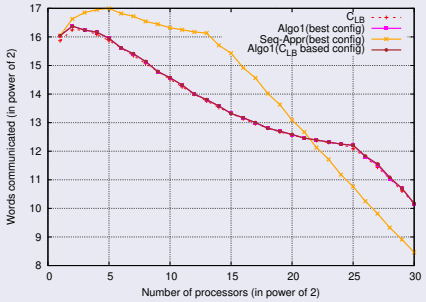
- Matrix communication costs dominate when P is much less than $\frac{n_1 n_2 n_3}{r_1 r_2 r_3}$

Performance Comparison of Our Algorithm

$n_1 = n_2 = n_3 = 2^{20}, r_1 = r_2 = r_3 = 2^8$



$n_1 = n_2 = n_3 = 2^{12}, r_1 = r_2 = r_3 = 2^4$



- C_{LB}: Communication lower bound, *Algo1(best config)*: Our algorithm with the best partition, *Algo1(C_{LB} based config)*: Our algorithm with the partition based on the lower bound, *Seq-Appr(best config)*: Multi-TTM computation used in Tucker-MPI with the best partition
- Our algorithm communicates much less than the approach in Tucker-MPI

Outline

Conclusion and Future Work

Conclusion

- Lower bounds for All-at-Once Multi-TTM
- Communication Optimal Algorithm for Multi-TTM
- Comparison of our algorithm with the TTM-in-Sequence Algorithm
- Our algorithm communicates much less data than TTM-in-Sequence for $P < \frac{n_1 n_2 n_3}{r_1 r_2 r_3}$

Future Work

- Detailed study of what scenarios are favorable for our algorithm
- Combine both All-at-Once and TTM-in-Sequence approaches to minimize communication lower bounds
- Extend our work for the full Tucker-decomposition algorithms