Parallel and Scalable Tensor Tools for Modern Computing Systems

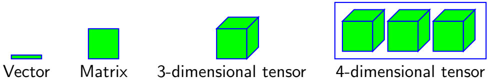
Suraj Kumar

ROMA Team
LIP and Inria, ENS Lyon

AG du LIP
November 29, 2022

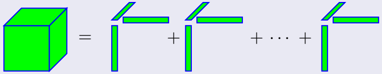
Tensors and their uses

- **Neuroscience**: Neuron × Time × Trial
- **Media**: User x Movie x Time
- **Ecommerce**: User x Product x Time
- **Social-Network**: Person x Person x Time x Type

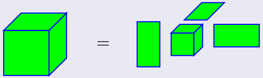


- High dimensional tensors: Neural network, Molecular simulation, Quantum computing
- People work with low dimensional structure (decomposition) of the tensors

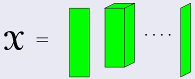
Canonical decomposition



Tucker decomposition



Tensor-train decomposition

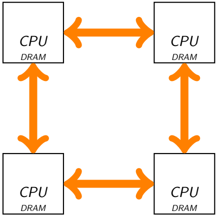


Importance of communication in high performance computing

- Gaps between computation and communication costs growing exponentially

	Annual improvements
Time-per-operation	59 %
Network-bandwidth	26 %
Network-latency	15 %

Source: Getting up to speed: The future of supercomputing (observed from 2004)



- **Research goal**: Scalable and communication optimal tools for tensor computations

Communication Optimal Algorithms for Multiple Tensor-Times-Matrix Computation

Hussam Al Daas[1], Grey Ballard[2], Laura Grigori[3], Suraj Kumar[4], and Kathryn Rouse[5]

[1]Rutherford Appleton Laboratory, UK

[2]Wake Forest University, USA

[3]Inria Paris, France

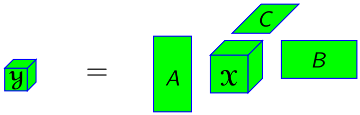
[4]Inria and ENS Lyon, France

[5]Inmar Intelligence, USA

AG du LIP

November 29, 2022

Higher-order SVD (HOSVD) to compute Tucker decomposition



Algorithm 1 3-dimensional HOSVD Algorithm($\mathcal{X}$)

1: Obtain factor matrices A, B and C from the matrix representations of the input tensor $\mathcal{X}$
2: $\mathcal{Y} = \mathcal{X} \times_1 A^\mathsf{T} \times_2 B^\mathsf{T} \times_3 C^\mathsf{T}$
3: Return $\mathcal{Y}$, A, B, C

- $\mathcal{X}$, $\mathcal{Y}$: 3-dimensional input and output tensors (or arrays) & A, B, C: matrices
- $\times_i$: tensor contraction along the ith dimension (similar to matrix multiplication)
- Multiple Tensor-Times-Matrix (Multi-TTM) computation: $\mathcal{Y} = \mathcal{X} \times_1 A^\mathsf{T} \times_2 B^\mathsf{T} \times_3 C^\mathsf{T}$

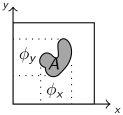
Communication lower bounds and communication optimal algorithms

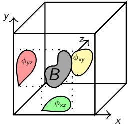
Settings

- P number of processors
- Each processor performs (asymptotically) equal amount of 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
- Focus on bandwidth cost (volume of data transfers)

Approach to obtain communication lower bounds

- Loomis-Whitney inequalitiy: for $d-1$ dimensional projections
  - For the 2d object A, $\phi_x\phi_y \geq Area(A)$
  - For the 3d object B, $(\phi_{xy}\phi_{yz}\phi_{xz})^{\frac{1}{2}} \geq Volume(B)$





- Hölder-Brascamp-Lieb (HBL) inequality – generalization for arbitrary dimensional projections
  - Provide exponent for each projection

Constraints for parallel load balanced matrix matrix multiplication

- $C = AB$ with $A \in \mathbb{R}^{n_1 \times n_2}$, $B \in \mathbb{R}^{n_2 \times n_3}$, and $C \in \mathbb{R}^{n_1 \times n_3}$

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

- ϕ_A, ϕ_B, ϕ_C: projections of computations on arrays A, B, C
- From Loomis-Whitney/HBL inequality: $\phi_A^{\frac{1}{2}}\phi_B^{\frac{1}{2}}\phi_C^{\frac{1}{2}} \geq$ number of multiplications per processor $= \frac{n_1n_2n_3}{P}$
- Extra constraints: $\frac{n_1n_2}{P} \leq \phi_A \leq n_1n_2$, $\frac{n_2n_3}{P} \leq \phi_B \leq n_2n_3$, $\frac{n_1n_3}{P} \leq \phi_C \leq n_1n_3$

Optimization problem and communication lower bounds

$$\begin{aligned}
&\textit{Minimize } \phi_A + \phi_B + \phi_C \quad \text{s.t.}\\
&\phi_A^{\frac{1}{2}}\phi_B^{\frac{1}{2}}\phi_C^{\frac{1}{2}} \geq \frac{n_1 n_2 n_3}{P}\\
&\frac{n_1 n_2}{P} \leq \phi_A \leq n_1 n_2\\
&\frac{n_2 n_3}{P} \leq \phi_B \leq n_2 n_3\\
&\frac{n_1 n_3}{P} \leq \phi_C \leq n_1 n_3
\end{aligned}$$

Amount of array accesses $= \phi_A + \phi_B + \phi_C$

- Estimate the solution and prove optimality using all Karush–Kuhn–Tucker conditions are satisfied
- For $n_1 \leq n_2 \leq n_3$,

P: 1 — $\frac{n_3}{n_2}$ — $\frac{n_2 n_3}{n_1^2}$

From 1 to $\frac{n_3}{n_2}$: $\phi_A = n_1 n_2$, $\phi_B = \frac{n_1 n_3}{P}$, $\phi_C = \frac{n_2 n_3}{P}$

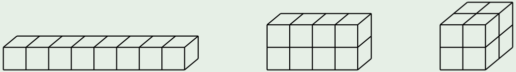
From $\frac{n_3}{n_2}$ to $\frac{n_2 n_3}{n_1^2}$: $\phi_A = \phi_B = \left(\frac{n_1^2 n_2 n_3}{P}\right)^{1/2}$, $\phi_C = \frac{n_2 n_3}{P}$

Beyond $\frac{n_2 n_3}{n_1^2}$: $\phi_A = \phi_B = \phi_C = \left(\frac{n_1 n_2 n_3}{P}\right)^{2/3}$

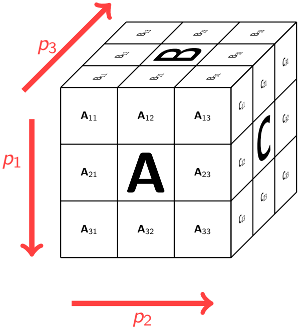
- Communication lower bound $= \phi_A + \phi_B + \phi_C -$ data owned by the processor $= \phi_A + \phi_B + \phi_C - \frac{n_1 n_2 + n_2 n_3 + n_1 n_3}{P}$

Design of communication optimal algorithms for $C = AB$

Arrangements of 8 processors

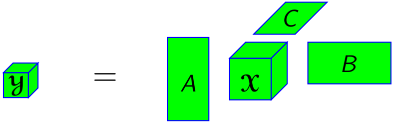


- P is organized into $p_1 \times p_2 \times p_3$ logical grid
- Select p_1, p_2 and p_3 based on the communication lower bounds
- Gather A on the set of processors along each slice of p_3
- Gather B on the set of processors along each slice of p_1
- Perform local computation
- Perform reduce operation along p_2 to obtain C



Outline

3-dimensional Multi-TTM computation



- $\mathcal{Y} = \mathcal{X} \times_1 A^{\mathsf{T}} \times_2 B^{\mathsf{T}} \times_3 C^{\mathsf{T}}$
- $\mathcal{X}$, $\mathcal{Y}$: 3-dimensional input and output tensors
- A, B, C: matrices
- $\times_i$: analogous to matrix multiplication

- TTM-in-Sequence approach (used in TuckerMPI library): $\mathcal{Y} = \left(\left(\mathcal{X} \times_1 A^{\mathsf{T}}\right) \times_2 B^{\mathsf{T}}\right) \times_3 C^{\mathsf{T}}$
- Our All-at-Once definition with $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{Y} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$, $A \in \mathbb{R}^{n_1 \times r_1}$, $B \in \mathbb{R}^{n_2 \times r_2}$, $C \in \mathbb{R}^{n_3 \times r_3}$

$$\text{for } \{n_1', n_2', n_3', r_1', r_2', r_3'\} = 1{:}\{n_1, n_2, n_3, r_1, r_2, r_3\}$$
$$\mathcal{Y}(r_1', r_2', r_3') += \left(\mathcal{X}(n_1', n_2', n_3') * A(n_1', r_1') * B(n_2', r_2') * C(n_3', r_3')\right)$$

- Establish lower bounds on data transfers based on the geometry of computations
- Design a 6-dimensional parallel algorithm
  - Select right parameters based on lower bounds to achieve communication optimality

Solving optimization problems 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

$$
\begin{aligned}
\text{Minimize } \phi_{\mathcal{X}} + \phi_{\mathcal{Y}} + \phi_1 + \phi_2 + \phi_3 \ \text{ s.t.}& \\
\phi_{\mathcal{X}}^{1-a}\phi_{\mathcal{Y}}^{1-a}\phi_1^a\phi_2^a\phi_3^a \geq \frac{n_1 r_1 n_2 r_2 n_3 r_3}{P}& \\
\frac{n_1 n_2 n_3}{P} \leq \phi_{\mathcal{X}} \leq n_1 n_2 n_3& \\
\frac{r_1 r_2 r_3}{P} \leq \phi_{\mathcal{Y}} \leq r_1 r_2 r_3& \\
\frac{n_1 r_1}{P} \leq \phi_1 \leq n_1 r_1& \\
\frac{n_2 r_2}{P} \leq \phi_2 \leq n_2 r_2& \\
\frac{n_3 r_3}{P} \leq \phi_3 \leq n_3 r_3& \\
0 \leq a \leq 1&
\end{aligned}
$$

Divide the problem into two parts

Matrix part

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

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

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

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

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

Tensor part

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

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

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

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

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 Karush–Kuhn–Tucker conditions)

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

P: $1 \;\longrightarrow\; \frac{n_3 r_3}{n_2 r_2} \;\longrightarrow\; \frac{n_2 r_2 n_3 r_3}{n_1^2 r_1^2} \;\longrightarrow\; P$

For $1 \leq P \leq \frac{n_3 r_3}{n_2 r_2}$:

$$\phi_1 = n_1 r_1, \quad \phi_2 = n_2 r_2, \quad \phi_3 = \frac{n_3 r_3}{P}$$

For $\frac{n_3 r_3}{n_2 r_2} \leq P \leq \frac{n_2 r_2 n_3 r_3}{n_1^2 r_1^2}$:

$$\phi_1 = n_1 r_1, \quad \phi_2 = \phi_3 = \left(\frac{n_2 r_2 n_3 r_3}{P}\right)^{1/2}$$

For $P \geq \frac{n_2 r_2 n_3 r_3}{n_1^2 r_1^2}$:

$$\phi_1 = \phi_2 = \phi_3 = \left(\frac{n_1 r_1 n_2 r_2 n_3 r_3}{P}\right)^{1/3}$$

P: $1 \;\longrightarrow\; \frac{n_1 n_2 n_3}{r_1 r_2 r_3} \;\longrightarrow\; P$

For $1 \leq P \leq \frac{n_1 n_2 n_3}{r_1 r_2 r_3}$:

$$\phi_y = r_1 r_2 r_3, \quad \phi_x = \frac{n_1 n_2 n_3}{P}$$

For $P \geq \frac{n_1 n_2 n_3}{r_1 r_2 r_3}$:

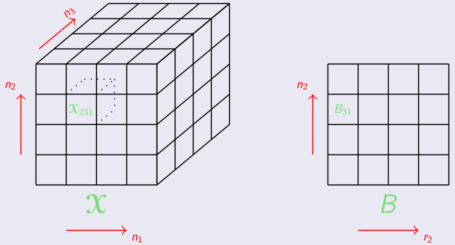
$$\phi_x = \phi_y = \left(\frac{n_1 r_1 n_2 r_2 n_3 r_3}{P}\right)^{1/2}$$

Communication lower bound $= \phi_x + \phi_y + \phi_1 + \phi_2 + \phi_3 - \frac{n_1 n_2 n_3 + r_1 r_2 r_3 + n_1 r_1 + n_2 r_2 + n_3 r_3}{P}$

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 $B_{31} = B(2\frac{n_2}{p_2} + 1 : 3\frac{n_2}{p_2}, 1 : \frac{r_2}{q_2})$ is distributed evenly among processors $(*, 3, *, *, 1, *)$



6-dimensional algorithm to compute Multi-TTM

Algorithm 1 3-dimensional Parallel Atomic Multi-TTM

Require: $\mathcal{X}$, A, B, C, $p_1 \times p_2 \times p_3 \times q_1 \times q_2 \times q_3$ logical processor grid

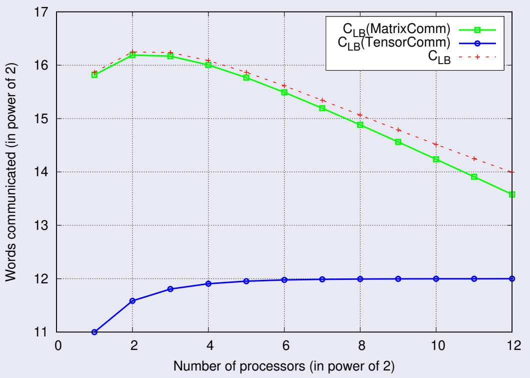
Ensure: $\mathcal{Y}$ such that $\mathcal{Y} = \mathcal{X} \times_1 A^\mathsf{T} \times_2 B^\mathsf{T} \times_3 C^\mathsf{T}$

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: $\mathcal{X}_{p_1' p_2' p_3'} = \text{All-Gather}(\mathcal{X}, (p_1', p_2', p_3', *, *, *))$
4: $A_{p_1' q_1'} = \text{All-Gather}(A, (p_1', *, *, q_1', *, *))$
5: $B_{p_2' q_2'} = \text{All-Gather}(B, (*, p_2', *, *, q_2', *))$
6: $C_{p_3' q_3'} = \text{All-Gather}(C, (*, *, p_3', *, *, q_3'))$
7: //Perform local Multi-TTM computation in a temporary tensor $\mathcal{T}$
8: $\mathcal{T} = \text{Local-Multi-TTM}(\mathcal{X}_{p_1' p_2' p_3'}, A_{p_1' q_1'}, B_{p_2' q_2'}, C_{p_3' q_3'})$
9: //Reduce-scatter the output tensor in $\mathcal{Y}_{q_1' q_2' q_3'}$
10: $\text{Reduce-Scatter}(\mathcal{Y}_{q_1' q_2' q_3'}, \mathcal{T}, (*, *, *, q_1', q_2', q_3'))$

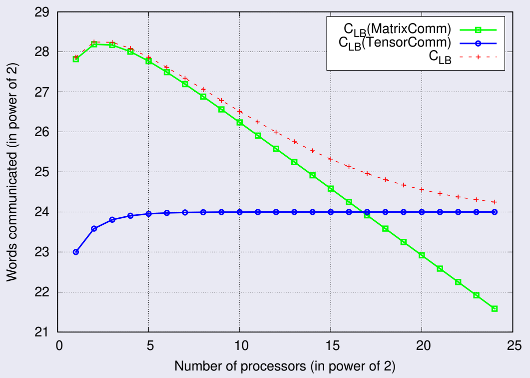
Outline

Communication lower bound (C_{LB}) distributions

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



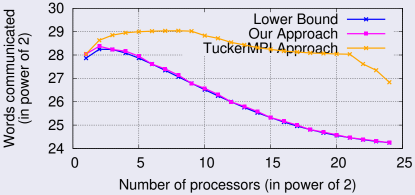
$n_1 = n_2 = n_3 = 2^{20}, r_1 = r_2 = r_3 = 2^8$



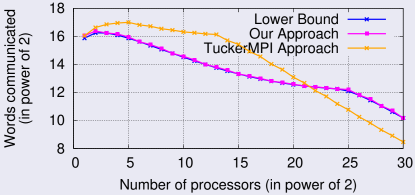
- Matrix communication costs dominate when P is much less than $\frac{n_1 n_2 n_3}{r_1 r_2 r_3}$

Performance comparison of our algorithm

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



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



- Typical scenarios in data compression problems
- For small P, our approach communicates much less than the state-of-the-art approach

Outline

Conclusion and future work

Conclusion

- Communication lower bounds and optimal algorithms for All-at-Once Multi-TTM
- Comparison of our approach with the TTM-in-Sequence approach
- Our algorithm communicates much less data than TTM-in-Sequence for small P

Future Work

- Detailed study of what scenarios are favorable for our approach
- Combine both All-at-Once and TTM-in-Sequence approaches
- Extend our framework for other linear algebra computations

Thank You!