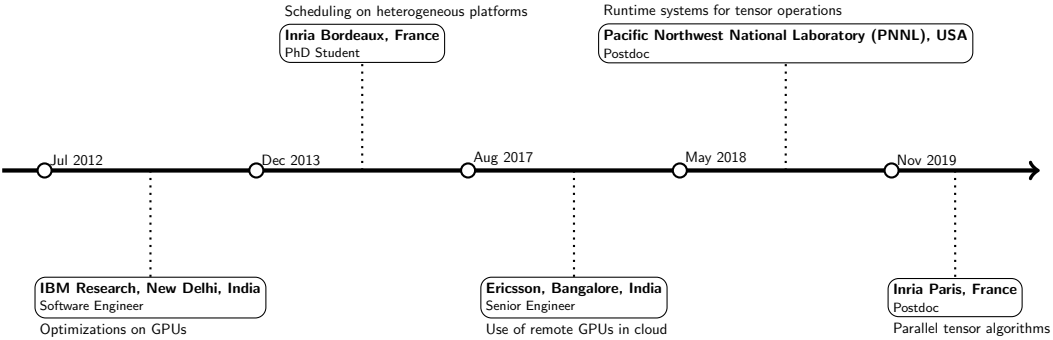
Scalable Tensor Algorithms for Modern Computing Systems

Suraj KUMAR

CNRS LIP/LaBRI Applicant

March 15, 2022

Resume in timeline

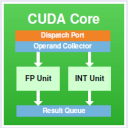


Advisors/Close collaborators

- PhD supervisors at Inria Bordeaux, France: Olivier Beaumont, Lionel Eyraud-Dubois, Samuel Thibault, Emmanuel Agullo
- Postdoc supervisors/Close collaborators: Marcin Zalewski (PNNL, USA), Sriram Krishnamoorthy (PNNL, USA), Laura Grigori (Inria Paris, France), Grey Ballard (Wake Forest University, USA)

Past research experience

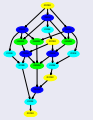
GPU algorithms

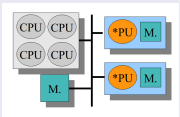


IBM Research, India, 2013

- Stencil computations

Scheduling on heterogeneous platforms

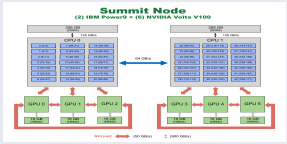




PhD thesis, Inria Bordeaux, France, 2017

- Dense linear algebra computations

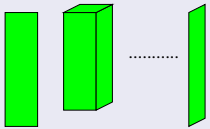
Fast molecular simulations



PNNL, USA, 2019

- Tensor computations, TAMM library

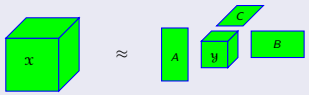
Parallel tensor approximations



Inria Paris, France, 2019

- Tensor-train decomposition

Communication optimal algorithms



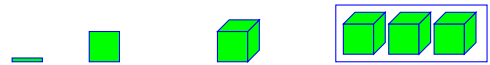
Inria Paris, France, 2019

- Multiple Tensor-times-matrix computations (to obtain $\mathcal{Y}$)

Expertise: Tensor computations, Parallel algorithms, Communication costs, Low-rank approximations

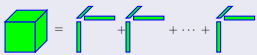
Tensors and their uses

- **Neuroscience**: Neuron $\times$ Time $\times$ 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

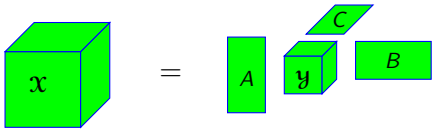
- Gaps between computation and communication costs growing exponentially

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

Source: Getting up to speed: The future of supercomputing

Goal : Scalable and communication optimal algorithms for tensor computations

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

1. For Matrix Matrix Multiplications
2. For Multi-TTM Computations

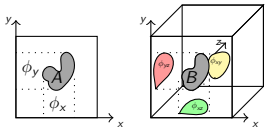
Settings

- 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)

- Submitted this work to SPAA 2022 conference
- Joint work with L. Grigori, G. Ballard, K. Rouse, H. Al Daas

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 $C \in \mathbb{R}^{n_1 \times n_3}$

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

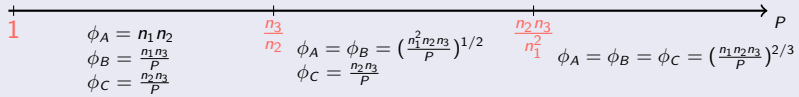
- ϕ_A, ϕ_B, ϕ_C: projections of computations on arrays A, B, C
- From Loomis-Whitney/HBL inequality: $\phi_A^{\frac{1}{2}} \phi_B^{\frac{1}{2}} \phi_C^{\frac{1}{2}} \geq$ number of multiplications per processor $= \frac{n_1 n_2 n_3}{P}$
- Our contributions: $\frac{n_1 n_2}{P} \leq \phi_A \leq n_1 n_2$, $\frac{n_2 n_3}{P} \leq \phi_B \leq n_2 n_3$, $\frac{n_1 n_3}{P} \leq \phi_C \leq n_1 n_3$

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 $= \phi_A + \phi_B + \phi_C$

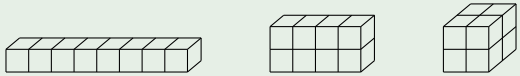
- Estimate the solution and prove optimality using all Karush–Kuhn–Tucker conditions are satisfied
- For $n_1 \leq n_2 \leq n_3$,



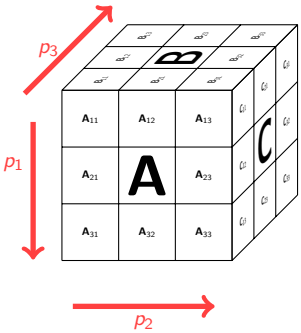
- Communication lower bound $= \phi_A + \phi_B + \phi_C -$ data owned by the processor $= \phi_A + \phi_B + \phi_C - \frac{n_1 n_2 + n_2 n_3 + n_1 n_3}{P}$

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



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

- TTM-in-Sequence approach (used in Tucker-MPI)
  - $\mathcal{Y} = \left(\left(\mathcal{X} \times_1 \mathbf{A}^{(1)^\mathsf{T}}\right) \times_2 \mathbf{A}^{(2)^\mathsf{T}}\right) \times_3 \mathbf{A}^{(3)^\mathsf{T}}$

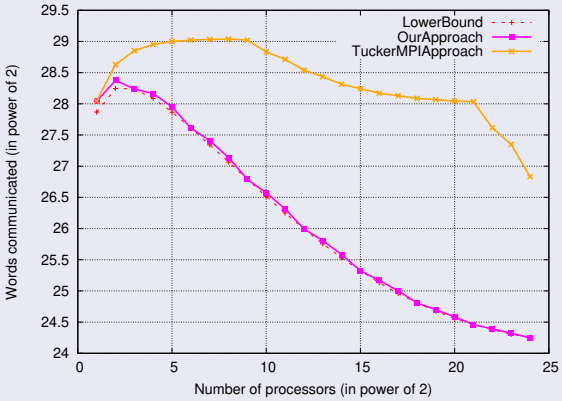
All-at-Once definition with $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{Y} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$, $\mathbf{A}^{(k)} \in \mathbb{R}^{n_k \times r_k}$ (our contribution)

$$\begin{aligned}
&\text{for } n_1' = 1{:}n_1, \text{ for } n_2' = 1{:}n_2, \text{ for } n_3' = 1{:}n_3 \\
&\quad \text{for } r_1' = 1{:}r_1, \text{ for } r_2' = 1{:}r_2, \text{ for } r_3' = 1{:}r_3 \\
&\qquad \mathcal{Y}(r_1', r_2', r_3') = \mathcal{Y}(r_1', r_2', r_3') \\
&\qquad\quad + \left(\mathcal{X}(n_1', n_2', n_3') * \mathbf{A}^{(1)}(n_1', r_1') * \mathbf{A}^{(2)}(n_2', r_2') * \mathbf{A}^{(3)}(n_3', r_3')\right)
\end{aligned}$$

- Applied our framework to compute lower bounds
- Designed similar communical optimal algorithms (though 6-dimensional)

Simulated performance comparison

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



- Typical scenarios in data compression problems
- Our approach communicates much less than the state-of-the-art approach (TuckerMPI)

Project: Scalable Tensor Algorithms for Modern Computing Systems

1. Design of Scalable Communication Optimal Algorithms for Tensors (Main Focus)
2. Extension of Existing Approaches/Algorithms (Short/Mid Term Research Plans)
3. Exploratory Topics (Mid/Long Term Research Plans)

Scalable algorithms for popular tensor operations

- Determine the communication lower bounds for tensor decompositions
- Analyse the popular decomposition algorithms and communications performed by them
- Propose new scalable communication optimal algorithms
  - If possible design tiles/tasks based algorithms
- Implement the proposed algorithms
- Same for manipulation operations of popular tensor representations
- Create a tensor library

Strassen's concepts to tensors

Matrix multiplication of $n \times n$ square matrices

- Complexity of traditional matrix multiplication is $\mathcal{O}(n^3)$
- Strassen's matrix multiplication
  - Expressed matrix multiplication as a tensor computation
  - Canonical rank of the tensor determines the complexity of the computation
  - Complexity is $\mathcal{O}(n^{2.81})$
- Plan to extend Strassen's concepts to tensor contractions

Contraction of a 3-dimensional tensor with a matrix

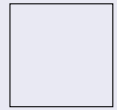
for $i_1 = 1 : n$ **do**
  for $i_2 = 1 : n$ **do**
    for $i_3 = 1 : n$ **do**
      for $j_2 = 1 : n$ **do**
        $\mathcal{G}(i_1, i_2, j_2) = \mathcal{G}(i_1, i_2, j_2) + \mathcal{A}(i_1, i_2, i_3) * B(i_3, j_2)$
      end for
    end for
  end for
end for

- Total $\mathcal{O}(n^4)$ operations
- Apply Strassen's algorithm for each i_1, total $\mathcal{O}(n^{3.81})$ operations
- Expressing as a canonical decomposition of $8 \times 8 \times 4$ tensor can further reduce the number of operations (first try)

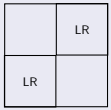
Hierarchical matrix concepts to tensors

Hierarchical matrices

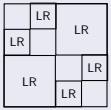
- Data sparse approximation of non-sparse matrices



Original Matrix



Step 1

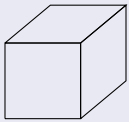


Step 2

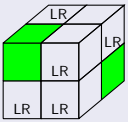
LR: low rank block

Tensors

- $f(i,j,k) = \frac{1}{|i-j|+|j-k|+|k-i|}$
- Value is small if difference of any pair is large
- Formalize and evaluate this approach for tensors



Original Tensor



Step 1

High dimensional tensor representations

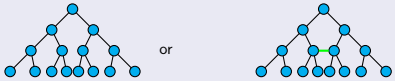
Notation: Tensors are denoted by solid shapes and number of lines denote the dimensions of the tensors. Connecting two lines implies summation (or contraction) over the connected dimensions.

- Adding tensors and applying an operator in Tensor-train representation

- Requires a truncation process which iterates over cores one by one – not suitable to work in parallel

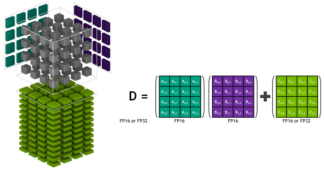
New tensor representations – suitable for parallelization

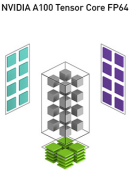
- Look at new representations in tree format



  - Data will be stored at the leaf nodes

Architecture aware algorithms and Training of tensorized neural networks





- Recent Nvidia GPUs have tensor cores to accelerate AI computations
- Design parallel algorithms which take architecture details into account
- Design parallel methods to train tensorized neural networks

Integration in the LIP/LaBRI laboratory

My research plans

- **Main focus**: scalable and communication optimal tensor algorithms
- **Short term plans**: Strassen's concepts to tensors, hierarchical matrix concepts to tensors
- **Long term plans**: new tensor representations, parallel training of tensorized neural networks

ROMA team (LIP laboratory)

- *Bora Ucar*: design of tensor compression and manipulation algorithms
- *Gregoire Pichon*: low-rank based methods
- *Anne Benoit, Loris Marchal, Yves Robert and Frederic Vivien*: scalability and scheduling aspects in the long term

SATANAS team (LaBRI laboratory)

- *Olivier Beaumont* and *Lionel Eyraud-Dubois*: parallel training of tensorized neural networks
- *Mathieu Faverge*: low-rank based methods
- *Abdou Guermouche, Samuel Thibault*: exploitation of maximum potential of HPC systems in the long term

Bringing additional skills

- High dimensional dense tensor computations, use of tensors in molecular simulations (for both teams)
- Communication lower bounds for linear algebra computations (for both teams)
- Scalable approaches for large HPC systems and GPU computations (only for the ROMA team)

Thank You!