# SCATE: SCAlable Tensor dEcompositions for data analytics

Décompositions de tenseurs évolutives pour l'analyse de données

CES46 > Axe E.05 > Calcul haute performance, Modèles numériques, simulation, applications

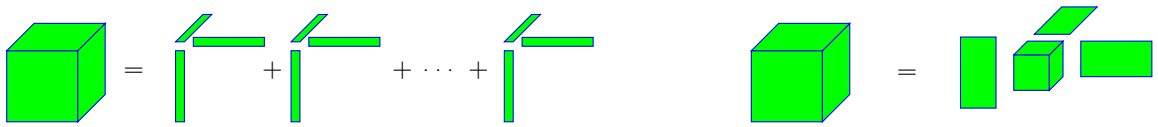
AAPG 2024 / JCJC 48 months

Suraj Kumar, ROMA project-team, Inria Lyon

## 1 Context, positioning and objectives

We produce approximately 2.5 quintillion bytes of data every day [1]. To analyze such massive amount of data, we need to design parallel and scalable approaches which can make efficient utilization of the modern computing systems. Tensors are multi dimensional arrays and used to store data in several domains [7], e.g., data mining, neuroscience and computer vision. Tensor decompositions help to identify inherent structure of data, achieve data compression and enable various ways of data analysis. Tensors are also used as operators to solve problems in applied mathematics, chemistry, physics, machine learning and many other fields [7, 8]. Working with tensors is tough due to their large computational effort and memory requirements (amount of memory and computations grow exponentially in the number of dimensions). It is therefore necessary to work with patterns of the tensor data. Using low dimensional structure of high dimensional data is a powerful approach in this context. Most tensor decompositions represent data in its low dimensional structures.

CP (also known as Canonical Polyadic or CANDECOMP or PARAFAC) and Tucker are the widely used tensor decompositions in the literature for data analytics. Both decompositions can be viewed as high order generalization of Singular Value Decomposition (SVD). CP decomposition is used to understand the latent components of the data. Tucker decomposition is considered to be more appropriate for compression and multi-modal data analysis.



(a) CP decomposition. (b) Tucker decomposition.

Figure 1: CP and Tucker decompositions of a 3-dimensional tensor.

Figure 1 shows visual representations of both tensor decompositions for a 3-dimensional tensor. CP decomposition represents a tensor with the sum of rank one tensors. Whereas, Tucker decomposition represents a tensor with multiple factor matrices and a much smaller core tensor. The number of dimensions for the core tensor and the original tensor are same. As mentioned earlier, tensors are used to store data in several domains. For example, in neuroscience, a single trial to observe behaviors of neurons over time can be stored in a matrix and data of multiple trials in a 3-dimensional tensor. CP decomposition is a popular method to analyse such neuroscience data [9, 6]. Similarly, a single experiment in combustion simulations or electron microscopy produces terabytes of data. For instance, tracking 64 variables on a 512×512×512 three-dimensional spatial grid for 128 time steps in a simulation requires to store $2^{40}$ entries. It is very difficult to transfer and analyze data of such experiments. Tucker decomposition is an appealing practice to compress such data before different types of analysis can be performed [2].

The data generated by many experiments is so big that it is impossible to perform a CP/Tucker decomposition without parallel computations. Therefore, it is important to focus on parallel algorithms for both decompositions. Thanks to recent advances in architectures, we witness many parallel high performance computing (HPC) systems which perform more than 1 million billion operations per second – Frontier at ORNL and Adastra at GENCI, for example. Such systems have a large number of computing units. In the last few decades, the rate of computations in these systems has improved drastically, but we have noticed relatively smaller improvement in the rate of data movement. The current HPC systems face bottleneck due to the large volume of communications [5]. Thus it is crucial to take communication costs into account while designing parallel algorithms. This approach also reduces energy consumption of the computations. GPUs deliver increased processing capabilities and superior energy efficiency compared to CPUs. Therefore, they have become a crucial element of many HPC systems over the past decade. The SCATE project addresses parallelization, communication costs and scalability of tensor decompositions on the modern HPC systems. More precisely, the high level aim of the project is to devise parallel and communication optimal tensor decomposition algorithms which scale well on both homogeneous and heterogeneous systems. Establishing communication lower bounds is also an important part of the project as it helps one to identify the more precise parallel algorithms that match the lower bounds.

Most existing tensor decomposition algorithms work with the matrix (2-dimensional) representations of the tensors at each step and rely on the parallelization of matrix operations. This approach neglects high dimensional properties of tensors and may not perform the computation efficiently. The high level idea of this project is to design approaches based on multidimensional properties of tensors. More precisely, we focus on improving the performance of Tucker and CP decomposition algorithms by taking multidimensional properties of tensors into account.

**Algorithms to compute Tucker decompositions:** Truncated higher-order SVD (HOSVD) is one of the popular algorithms for computing a Tucker decomposition, thanks to its quasi-optimal numerical approximation. In this algorithm, the core tensor is computed with the full-format original tensor and multiple tall and skinny factor matrices. For a 3-dimensional tensor $\mathcal{X}$, computation of the core tensor $\mathcal{Y}$ can be expressed in tensor notation as $\mathcal{Y} = \mathcal{X} \times_1 \mathsf{A}^{(1)^\mathsf{T}} \times_2 \mathsf{A}^{(2)^\mathsf{T}} \times_3 \mathsf{A}^{(3)^\mathsf{T}}$, where $\mathsf{A}^{(k)}$ is a tall-skinny factor matrix corresponding to mode $k$, and $\times_k$ denotes the tensor-times-matrix (TTM) operation in the $k$th mode [7]. This collective operation is known as the Multi-TTM computation [4] and is one of the main bottlenecks of the HOSVD algorithm. One approach to perform this computation is in sequence, i.e., $\mathcal{Y} = ((\mathcal{X} \times_1 \mathsf{A}^{(1)^\mathsf{T}}) \times_2 \mathsf{A}^{(2)^\mathsf{T}}) \times_3 \mathsf{A}^{(3)^\mathsf{T}}$. Another approach is to work with all the inputs at once, i.e., $\mathcal{Y}_{ijk} = \sum_{lmp} \mathcal{X}_{lmp} \cdot \mathsf{A}^{(1)}_{li} \cdot \mathsf{A}^{(2)}_{mj} \cdot \mathsf{A}^{(3)}_{pk}$. Daas et al. [4] show that the latter approach takes high dimensional properties of tensors into account and reduces communication significantly compared to the sequence approach for small and moderate number of processors. However there is not any clear winner for all settings. We can obtain better gains by a hybrid method which combines the best of both approaches. Furthermore, the amount of operations for the Multi-TTM computation depends on how factor matrices are operated with the input tensor. It is also important to model the computation-communication tradeoff in the combined hybrid method such that the overall completion time of the Multi-TTM is minimized.

In addition, HOSVD algorithm also computes SVD for each mode to obtain the corresponding factor matrix. Optimal data distributions to perform SVDs and the Multi-TTM computation may not be compatible and may require to perform extra communication. At last, our goal is to compute communication lower bound and communication-computation tradeoff for the complete HOSVD algorithm and devise a method which achieves the optimal completion time. As SVD is expensive, we also plan to look at other alternatives like randomized SVD or QR based column pivoting to compute factor matrices.

**Algorithms to compute CP decompositions:** Computing a CP decomposition involves solving a non-linear optimization problem to minimize the approximation error. The workhorse algorithm to compute this decomposition uses an alternating least squares approach. This works in multiple iterations. For a $d$-dimensional decomposition, in each iteration, $d$ matricized-tensor times Khatri-Rao product (MTTKRP) computations are performed. In this computation, a matrix representation of the original tensor is multiplied with the Khatri-Rao product of $d-1$ factor matrices. This is the bottleneck computation of the algorithm. Ballard et al. [3] show that the working with all the inputs at once without forming intermediates reduces communication costs significantly compared to the procedure where the Khatri-Rao product is computed first and the intermediate output is multiplied with a matricized representation of the original tensor. Certain operations can be reused among all MTTKRPs of a single iteration. But, their approach restricts the reuse of operations as intermediates are not formed. The amount of operations in the MTTKRP computation also depends on how factor matrices are operated with the original tensor. It is important to study tradeoff among communications, computations and reuse of intermediate results for the MTTKRP computations and design a method which achieves optimal completion time. Similar to the previous plan, at last, our goal is to devise a method for a single iteration of the algorithm which achieves optimal completion time. Another state-of-the-art approach to compute a CP decomposition is based on gradients. MTTKRP is also the bottleneck computation for this approach. Hence, improving the performance of MTTKRP will also accelerate gradient-based approach to compute the decomposition.

**Objectives:** The goal of the SCATE project is to devise Tucker and CP decomposition algorithms which scale well on the modern computing systems. The project first studies tradeoff among computations, communications and data reuse for the existing algorithms and proposes new methods whose completion times are optimal for homogeneous systems. After that, the proposed methods will be implemented and tested with various data-sets on GENCI computers. Significant amount of performance is produced by accelerators for the present HPC systems, including many at GENCI. So developing only homogeneous algorithms ignores too much computation power and may result in poor efficiency on heterogeneous systems. The project does address this concern and extends our framework for heterogeneous systems with CPUs and GPUs.

**Tasks:** The SCATE project has two tasks. **Task A** focuses on homogeneous systems and **Task B** targets heterogeneous systems. **Task A** is further subdivided into six main work packages. **WPA1** aims to obtain communication lower bounds for algorithms to compute Tucker and CP decompositions. **WPA2** analyzes tradeoff among computations, communications and data reuse for the tensor decomposition algorithms. **WPA3** designs new algorithms which achieve optimal completion time. **WPA4** implements the proposed algorithms for distributed memory homogeneous systems and validates the performance improvement on more than 5,000 cores and/or 512 nodes of GENCI computers. **WPA5** studies the numerical robustness of the proposed algorithms. **WPA6**, towards the end of the project, explores how to apply mixed precision arithmetic to further improve our algorithms.

Similarly, **Task B** is further subdivided into four main work packages. **WPB1** models communication costs among different processing units for a single node composed of CPUs and GPUs. **WPB2** aims to obtain communication lower bounds of tensor decomposition algorithms and **WPB3** proposes new algorithms which achieve optimal completion times for a single heterogeneous node. **WPB4** extends the proposed framework to multiple heterogeneous nodes.

We will begin with **Task A**. Our framework for **WPA1** and **WPA2** will be ready within a year. We will also have good understanding of structure of computations and communications for the decomposition algorithms by then. We plan to extend our framework for heterogeneous systems at this point and start **Task B**.

## 2 Consortium

**Scientific coordinator:** Suraj Kumar (PI) holds an Inria Starting Faculty Position since October 2022 in the ROMA Inria team, which is hosted at École Normale Supérieure de Lyon (ENS Lyon). He is an expert in tensor computations, communication avoiding algorithms and efficient utilization of resources on heterogeneous systems. He defended his thesis in April 2017 on Scheduling of Dense Linear Algebra Kernels on Heterogeneous Resources. After that, he joined Ericsson Research, India as an engineer and worked on the use of remote GPUs in cloud. He was a post-doctorate researcher at Pacific Northwest National Laboratory, USA from May 2018 to October 2019. He worked there on NWChemEx project, whose main goal was to run molecular simulations efficiently on exascale computers. He also worked at Inria Paris from November 2019 to September 2022 as a postdoctoral researcher on the design of parallel and communication optimal algorithms for Multi-TTM computations and tensor-train decompositions. He has collaborated with over 25 researchers all over the world. In particular he has an active collaboration with G. Ballard of Wake Forest University, USA, who is an expert in the design of communication-efficient algorithms, that will be beneficial to the implementation of the project. The PI will dedicate 75% of his research time on this project.

**Team:** The SCATE project includes L. Marchal (DR2, CNRS) and B. Uçar (DR2, CNRS) from the Inria ROMA team. We ask for a PhD student, 24 months of a postdoctoral researcher and two master interns. The PhD student will work with the PI and B. Uçar (PhD co-advisor) on **Task A** (**WPA1** to **WPA4**). The master interns will work on **WPA5** and **WPA6** with the PI. The postdoctoral researcher will work with the PI and L. Marchal on **Task B**, and will also help the PI to advise the PhD student. B. Uçar is an expert in tensor computations, while L. Marchal is an expert in memory aware computations and design of scheduling algorithms for heterogeneous resources.

**Growth of the PI:** We expect to publish the obtained results to the major venues (conferences/journals) of the field (SC, IPDPS, SIMAX, SISC). Furthermore we plan to organize a minisymposium on tensor computations at SIAM Conference on Computational Science and Engineering (CSE25). This will allow us to start more collaborations and fruitful discussions. Tensor operations are the basic ingredients of many molecular and quantum simulations. The approval of this proposal certainly will speed up our discussions with domain scientists and help us to establish new collaborations.

## References


[1] "How Much Data Is Created Every Day?" https://earthweb.com/how-much-data-is-created-every-day, accessed: 2023-03-18.

[2] W. Austin, G. Ballard, and T. G. Kolda, "Parallel tensor compression for large-scale scientific data," in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2016, pp. 912–922.

[3] G. Ballard, N. Knight, and K. Rouse, "Communication lower bounds for matricized tensor times khatri-rao product," in *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2018, pp. 557–567.

[4] H. A. Daas, G. Ballard, L. Grigori, S. Kumar, and K. Rouse, "Communication lower bounds and optimal algorithms for multiple tensor-times-matrix computation," 2022. [Online]. Available: https://arxiv.org/abs/2207.10437

[5] DOE ASCAC Subcommittee Report, "Top ten exascale research challenges," 2014, accessed: 2023-03-15. [Online]. Available: https://www.osti.gov/biblio/1222713

[6] D. Hong, T. G. Kolda, and J. A. Duersch, "Generalized canonical polyadic tensor decomposition," *SIAM Review*, vol. 62, no. 1, pp. 133–163, 2020.

[7] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[8] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[9] A. H. Williams, T. H. Kim, F. Wang, S. Vyas, S. I. Ryu, K. V. Shenoy, M. Schnitzer, T. G. Kolda, and S. Ganguli, "Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis," *Neuron*, vol. 98, no. 6, pp. 1099–1115.e8, 2018.