POSTER: A Parallel Sparse Tensor Benchmark Suite on CPUs and GPUs

Jiajia Li
Jiajia.Li@pnnl.gov
Pacific Northwest National Laboratory
Richland, WA, USA

Mahesh Lakshminarasimhan
maheshl@cs.utah.edu
University of Utah
Salt Lake City, UT, USA

Xiaolong Wu
wu1565@purdue.edu
Purdue University
West Lafayette, USA

Ang Li
ang.li@pnnl.gov
Pacific Northwest National Laboratory
Richland, WA, USA

Catherine Olschanowsky
catherineolschan@boisestate.edu
Boise State University
Boise, ID, USA

Kevin Barker
Kevin.Barker@pnnl.gov
Pacific Northwest National Laboratory
Richland, WA, USA

Abstract

Tensor computations present significant performance challenges that impact a wide spectrum of applications. Efforts on improving the performance of tensor computations include exploring data layout, execution scheduling, and parallelism in common tensor kernels. This work presents a benchmark suite for arbitrary-order sparse tensor kernels using state-of-the-art tensor formats: coordinate (COO) and hierarchical coordinate (HiCOO). It demonstrates a set of reference tensor kernel implementations and some observations on Intel CPUs and NVIDIA GPUs. The full paper can be referred to at http://arxiv.org/abs/2001.00660.

CCS Concepts • **Computing methodologies → Shared memory algorithms**; **Massively parallel algorithms**; • **Mathematics of computing → Mathematical software performance**.

Keywords sparse tensors, benchmark, GPU, roofline model

1 Introduction

Tensors (as multi-dimensional arrays) especially sparse tensors, are utilized by a large number of critical applications that span a range of domain areas, which include quantum chemistry, healthcare analytics, social network analysis, data mining, signal processing, machine learning, and more. Operations on sparse tensors tend to dominate the execution-time of these applications. Understanding the performance characteristics of different implementation approaches is of paramount importance. This paper presents a benchmark suite specifically for that purpose. The suite provides implementations of common tensor kernels using state-of-the-art sparse tensor data structures and a variety of real sparse tensors as its input dataset.

Given the heterogeneity in available hardware resources for high performance computing (HPC), it is non-trivial to answer questions about the potential for sparse tensor algorithms to be efficiently ported to various hardware. The difficulty of planning for the irregular parallelism that results from operating on sparse data structures is compounded by the availability of Graphics Processing Units (GPUs), vectorizing units, Field Programmable Gate Arrays (FPGAs), and potentially Tensor Processing Units (TPUs).

Optimizing the performance of tensor applications is challenging due to several application characteristics, named in [1, 3, 4]: *curse of dimensionality, mode orientation, tensor transformation, irregularity, and arbitrary tensor orders (or dimensions).* Beyond these, challenges associated with all benchmarks also apply, which include *completeness, diversity, extendibility, reproducibility, and comparability* across implementations. Comparisons across research groups and optimizations will be improved by using a standard set of kernels and inputs.

2 Contributions

Our benchmark suite consists of a set of reference implementations from various tensor applications, each of which show different computational behavior. We implement two sparse tensor formats: the most popular and mode-generic coordinate (COO) format and a newly proposed, more compressed hierarchical coordinate (HiCOO) format [5] to represent general, arbitrary sparse tensors. Beyond the implementation diversity, platform and workload (or input) diversity is also critical to gain insights from a benchmark suite. We implement the same set of tensor kernels on CPUs and GPUs to provide a good understanding for users. Different inputs of an algorithm usually obtain different performance due to their diverse data sizes and patterns. This phenomenon is

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

PPoPP '20, February 22–26, 2020, San Diego, CA, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6818-6/20/02.
https://doi.org/10.1145/3332466.3374513

more obvious for sparse problems because their algorithm behavior largely depends on the features of data. Besides, our benchmark suite can easily adopt new sparse tensor kernels and data representations.

The contributions of this work include:

- reference implementations for five tensor kernels [1, 3, 4]: tensor element-wise (Tew), tensor-scalar (Ts), tensor-times-vector (Ttv), tensor-times-matrix (Ttm), and matriced tensor times Khatri-Rao product (Mttkrp), in COO and HiCOO formats on CPUs and GPUs;
- Roofline performance models for one multicore Intel CPU and one NVIDIA GPU platforms to analyze the tensor kernels; and insights gained from experiments and analysis of the performance.

3 Experimental Results

We perform experiments on an Intel Xeon Gold 6126 multicore server platform with 56 physical 2.6GHz-cores distributed on four sockets and an NVIDIA Tesla V100 GPU with 5120 1.53GHz-cores. Figure 1 plots the Roofline models for the two platforms with DRAM and last-level cache (LLC) bandwidth tested from the Empirical Roofline Tool (ERT) [6], and the theoretical peak SP performance and DRAM bandwidth (not cache-aware) for reference. The dataset, described in Table 1, uses sparse tensors derived from real-world applications [2, 7].

Table 1. Description of sparse tensors.

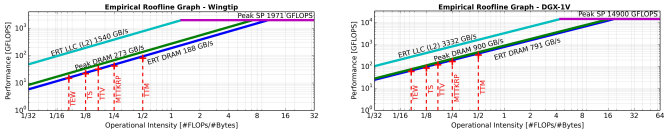
No.	Tensors	Order	Dimensions	#Nnzs	Density
r1	vast	3	$165K \times 11K \times 2$	26M	6.9×10^{-3}
r2	nell2	3	$12K \times 9K \times 29K$	77M	2.4×10^{-5}
r3	choa	3	$712K \times 10K \times 767$	27M	5.0×10^{-6}
r4	darpa	3	$22K \times 22K \times 24M$	28M	2.4×10^{-9}
r5	fb-m	3	$23M \times 23M \times 166$	100M	1.1×10^{-9}
r6	fb-s	3	$39M \times 39M \times 532$	140M	1.7×10^{-10}
r7	flickr	3	$320K \times 28M \times 1.6M$	113M	7.8×10^{-12}
r8	deli	3	$533K \times 17M \times 2.5M$	140M	6.1×10^{-12}
r9	nell1	3	$2.9M \times 2.1M \times 25M$	144M	9.1×10^{-13}
r10	crime4d	4	$6K \times 24 \times 77 \times 32$	5M	1.5×10^{-2}
r11	uber4d	4	$183 \times 24 \times 1140 \times 1717$	3M	3.9×10^{-4}
r12	nips4d	4	$2K \times 3K \times 14K \times 17$	3M	1.8×10^{-6}
r13	enron4d	4	$6K \times 6K \times 244K \times 1K$	54M	5.5×10^{-9}
r14	flickr4d	4	$320K \times 28M \times 1.6M \times 731$	113M	1.1×10^{-14}
r15	deli4d	4	$533K \times 17M \times 2.5M \times 1K$	140M	4.3×10^{-15}

Observation 1: *Achieved performance is diverse and hard to predict, which varies with the dimension sizes and non-zero patterns of tensors, platforms, and data formats.*

Observation 2: *Performance is generally below the Roofline performance calculated from main/global memory bandwidth except for some small tensors fitting into caches or algorithms with good data locality thus making a good use of caches.*

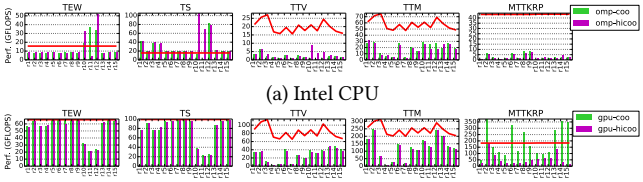
Observation 3: *It is hard to obtain good performance efficiency for non-streaming kernels on multi-socket CPU machines because of NUMA effect, which might be even harder than on GPUs.*

Observation 4: *HiCOO algorithms is faster than or similar to COO counterparts because of its better local locality and smaller memory footprint, except Mttkrp on GPUs where load imbalance and lower parallelism play more important roles.*



(a) Intel CPU (b) NVIDIA GPU

Figure 1. Roofline models marked with the operational intensities of tensor kernels.



(a) Intel CPU

(b) NVIDIA GPU

Figure 2. Single-precision performance of tensor kernels on the Intel CPU and NVIDIA Tesla V100 platforms with the Roofline performance marked as a line.

4 Conclusion

This paper presents a benchmark suite targeting sparse tensor kernels, which are memory bound and often dominate application performance. It identifies important kernels and data representations and provides reference implementations to aid the community in effectively sharing and comparing performance and optimization results. This benchmark suite is a continuous effort: more operations and complete tensor methods, data representations, platforms will be included.

Acknowledgments

This research was funded by the US Department of Energy, Office for Advanced Scientific Computing (ASCR) under No. 66150: "CENATE: The Center for Advanced Technology Evaluation" and the Laboratory Directed Research and Development program at PNNL under contract No. ND8577. Pacific Northwest National Laboratory (PNNL) is a multiprogram national laboratory operated for DOE by Battelle Memorial Institute under Contract DE-AC05-76RL01830.

References

[1] Andrzej Cichocki. 2014. Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions. *CoRR* abs/1403.2048 (2014).

[2] Inah Jeon et al. 2015. HaTen2: Billion-scale Tensor Decompositions. In *IEEE International Conference on Data Engineering (ICDE)*.

[3] T. Kolda et al. 2009. Tensor Decompositions and Applications. *SIAM Rev.* 51, 3 (2009), 455–500. https://doi.org/10.1137/07070111X

[4] Jiajia Li. 2018. *Scalable tensor decompositions in high performance computing environments.* Ph.D. Dissertation. Georgia Institute of Technology, Atlanta, GA, USA.

[5] Jiajia Li et al. 2018. HiCOO: Hierarchical storage of sparse tensors. In *Proceedings of the ACM/IEEE International Conference on High Performance Computing, Networking, Storage and Analysis (SC)*. Dallas, TX, USA.

[6] Yu Jung Lo et al. 2015. Roofline Model Toolkit: A Practical Tool for Architectural and Program Analysis. In *High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation*. Springer International Publishing, Cham, 129–148.

[7] Shaden Smith et al. 2017. FROSTT: The Formidable Repository of Open Sparse Tensors and Tools. http://frostt.io/