1-1-2021

Optimizing Sparse Tensor Computations via Orderings and Multilayered Data-Structures

Joshua D. Booth

Optimizing Sparse Tensor Computations via Orderings and Multilayered Data-Structures

Joshua Dennis Booth
Assistant Professor
Department of Computer Science

Tech-Hall N358
256.824.6433
joshua.booth@uah.edu

RCEU21-CS-JDB-01

RCEU21-CS-JDB-01 Joshua Dennis Booth

Project Description: The importance of tensor computations has grown to new levels of importance in all areas of computer and data science that depend on machine learning, such as deep neural networks, due to tensors being used as a primary data packaging device by tools like TensorFlow. New research has focused on sparse tensors, i.e., tensors where the number of nonzero elements is on the order of the dimension of tensor ~ O(n), as opposed to dense tensor, i.e., tenors where all elements are treated as nonzero elements, due to concerns with data storage, computational intensity, and overfitting of model parameters. Additionally, many new neural network architectures, such as graph neural networks, are better represented by sparse tensors. However, sparse tensors, like sparse matrices (i.e., 2D tenors), can suffer from performance issues on both multicore and graphic processing units (GPU) systems. The most common way to reduce poor performance is to find an ordering (i.e., a set of locations where nonzero elements should be placed in the tensor) that better fits the architecture's memory access pattern constraints and may result in faster convergence of iterative methods. Better orderings can result in up to 4-10*x* faster computations, and thus can speedup the many applications that make calls to sparse tensor packages. Finding the best ordering and the corresponding data-structure is NP-Hard, and a good metric to even evaluate an ordering and data-structure for a given architecture is an open problem.

This work proposes the exploratory work of trying several of the current sparse tensor orderings and data structure designed for multicore and GPU systems on several different architectures available at Pittsburg SuperComputing Center (PSC), Texas Advance Computing Center (TACC), and Alabama Supercomputer Authority (ASC) to judge their relative "goodness". From this initial data, a metric will be designed that integrates ordering distance (i.e., the distance in memory nonzero elements are stored)

and target architecture. Additionally, the multilevel caching techniques that Dr. Booth used for sparse matrices will be converted to a tensor ordering and data-structure method and evaluated.

Some initial work related to this was carried out by Dr. Booth and an undergraduate research student at Franklin & Marshall College, which shows that there does exist a large gap in performance for one tensor operation on one system. This work will extend far past this work but can use some of the initial structure, such as tensor operation and a test set of tensors, as a starting point.

The resulting findings will have an impact across multiple areas that use tensors and will help experts better select orderings and data-structures for their computational needs.

Student Duties, Contributions, and Outcomes: *Specific duties:* The student's duties will include reviewing literature about sparse tensors, tensor operations, and parallel computing. In the first three weeks (week 1-3), the student will be asked to review and summarize three journal papers written about sparse tensors and tensor operations, and the student will be asked to learn the following under the guidance of several tutorials posted by Lawrence Livermore National Labs and XSEDE: login, learn to

submit jobs, compile already written code, review basic Linux operations for our targeted hardware. In the next three weeks (week 4-6), the student will review one journal paper per week and will move into the testing phase. The testing phase will involve compiling and running the sparse tensor ordering and data structure libraries already available in open-source locations. Orderings that are not available, but critical to exploration, will be programmed by the student in a high-level langue, such as Python or Matlab, and then ran in parallel using the framework provided by Dr. Booth. Data will be collected in CSV format and analyzed in Microsoft Excel. In the next two weeks (week 7-8), the student will try to make the needed modification to Dr. Booth's past sparse matrix method to use for sparse Tensors. This will include both programming in a high-level language and coding in the low-level language C. Additionally, the student will start to analyze the results from the past runs using Microsoft Excel and will compare their results to those in literature. They will begin a Latex write up that will include a detailed background section related to the journal papers they reviewed. They will also log any of their findings from the data analysis. In the last two weeks (9-10), the student will continue their Latex write-up, compare analyzed data that now include Dr. Booth's method, and will prepare a poster in Microsoft PowerPoint explaining their work.

Tangible Contribution: The student will produce an online repository (GitHub) with programs, data, and writeups. The goal is to turn this poster into a piece to submit at either SuperComputing's ACM Undergraduate Poster Section or SIAM PP's poster section. The choice of the venue will be determined by the outcome of the experiments.

Student Specific Outcomes: The student will gain a deep insight into the importance of tensors and tensor operations (which is something no undergraduate course in CS or MATH at UAH offers). They will also gain experience working on high-performance supercomputers. Even though this project does not require the student to do parallel computing, the ability to use these machines is an important milestone for anyone looking to enter the field of high-performance computing. Additionally, they will learn the important lesson of how to read and review a research journal, and how to start writing their own in Latex.

Student Selection Criteria: The student must have complete course work in either CS 221 or CPE 212 and MA224. They must be comfortable working in a terminal environment (e.g., bash shell).

Project Mentorship: The student will meet with Dr. Booth twice a week to discuss the progress of the project. Ad-hoc meetings will be set up as needed if issues come up. Dr. Booth will regularly monitor the student's repo to judge if the student is having issues. Dr. Booth currently has an undergraduate research student that will be moving into their MS under Dr. Booth, and he will ask this student to aid in some of the "getting started" questions about the computer systems. Ideally, the student will be able to work in a lab in the CS department, while Dr. Booth works in his office. If questions come up that need a quick response, the student can pop into his office.

Safety and Contingency Plan: The student will be required to take the XSEDE video tutorial on good practices for shared resource computing. The tutorial explains that these devices are shared and that all work is monitored, i.e., there is no expectation of privacy. Additionally, any illegal behavior, such as doing calculations for nuclear devices, running spamming attacks, password cracking, etc, will be reported to the proper authority. Additionally, all repositories the student writes to will not be open to public view (only the student and Dr. Booth will have access) to keep the privacy of the student. Ideally, research will happen on campus in Tech Hall. However, if face-to-face meetings are not able to happen, meetings will be done via Zoom or Microsoft Teams. Overall, this will have no real impact on the project, though may slow it down and be inconvenient for both the student and Dr. Booth.