Rateless Codes for Sparse Matrix and Non-Linear Computations

Project Web Page

sophiesmith.me/research-proposal.html

Project Description

In computations on large-scale parallel and distributed systems, straggling nodes (workers used for computation which unpredictably fail or slow down) serve as a performance bottleneck. Over the years, researchers have developed various techniques to handle this slowdown including data replication, work reassignment, or usage of other coding techniques. One of the techniques used, traditional fixed-rate erasure coding, involves a lot of redundant computation. Thus, researchers have started to move towards approaches in rateless fountain coding strategy where the codes no longer have a fixed proportion of useful data in the associated data stream by which the information is sent. Rather, information is sent at a fast rate with the understanding that not all will be received, and the computations are instead structured to return correct results despite this data loss. Particularly within the Optimization, Probability, and Learning (OPAL) Lab at Carnegie Mellon University, work has been done to explore the impact of rateless coding techniques on the latency and load-balancing of matrix-vector multiplication. Based on previous successes of this approach to adapt to varying node speeds and to achieve good load-balancing and a limited number of redundant computations, the OPAL Lab intends to extend the approach to other computational environments including sparse matrix-vector multiplication (SpMV) and additional non-linear computations such as neural network inference. I will be working with Professor Gauri Joshi (Lead of OPAL Lab) and her Ph.D. student Ankur Mallick to analyze and implement these further applications for rateless coding techniques.

Project Goals

The intended project is composed of two parts, effectively the 75% goal and the 100% goal. The first goal is to design rateless codes applicable for sparse matrix computations and to ensure these codes do not drastically increase the density of the matrix rows. Following the completion of this step, the overall goal would be to extend this work into developing additional rateless coding techniques for non-linear computations. This would enable the development of effective codes for use in machine learning inference. If both of these goals are achieved, then for the additional goal (125%), we could develop rateless coding techniques for other computation types or other applications of non-linear computation to machine learning. The details of the extended project are fairly vague at this point, as this goal could be determined depending on which research conclusions show promise or potential for additional depth from the first two goals.

Milestones

For this project, I'll be collaborating with the Ph.D. student, Ankur Mallick, who has done the initial work with rateless coding applications to matrix multiplication. For the first technical milestone, I hope to meet with him to discuss initial steps for extending this project and determine the process for me to get all necessary background information for the project. Then, after this meeting, for the remainder of this semester and over Winter Break, I can read papers or other resources he suggests to gain the necessary background. Thus, at the beginning of the Spring semester, I'll begin on the implementation and actual research project.

The intended biweekly milestones for this project are listed below.
January 27- By this point, I plan to meet with Professor Joshi and Ankur Mallick multiple times to discuss the beginning steps for the rateless coding on sparse matrices project. This will include setting up necessary coding environments and discussing general ideas for the approach to take for this project in terms of implementation and project direction.

February 10- At this point, I will have all the necessary background information and environments set up. For this milestone, I plan on designing the majority of the model for sparse matrix rateless coding and to begin setting up the analysis and experimental evaluation of performance.

February 24- The goal is that by this point I will finish the theoretical evaluation of sparse matrix multiplication and resulting density analysis. I also hope to set up the environments for experimental evaluation and begin on the performance measurements.

March 16- At this time, I hope to complete the 75% goal of sparse matrix analysis. The final step which needs to be done by this point is to complete the benchmarking and performance analysis on parallel and distributed coding environments.

March 30- After hopefully finishing the 75% goal, I hope to begin on the rateless coding applied to non-linear computation environments project. For the first milestone of this goal, I plan on narrowing down the problem space to a particular non-linear or machine learning inference application. Once the problem space is determined, I'll also begin the design of rateless coding applications and experiment structure to this space.

April 13- Ideally, I will finish the majority of design and implementation for the rateless coding application to the non-linear environment by this time.

April 27- At this point, I hope to complete the implementation and benchmarking for the non-linear application project. Thus, with the remainder of the semester after this point, I can focus on the analysis and formalization of the work I've completed throughout the semester.

Literature Search

With the prevalence of computation in increasingly parallel and distributed environments, efforts have been focused on determining ways to limit the negative impact of straggler nodes. Previous approaches have included task replication and alternative load balancing strategies. Task replication methods focus on creating back-up tasks or copies of the task assigned to a particular worker node and assigning this task to additional workers. Thus, rather than waiting on one particular worker to finish the computation, the result used is the result produced by the fastest worker node. Additionally, load balancing techniques involve dynamically moving tasks from slower workers to faster workers to decrease computation time. While both of these approaches are sufficient to decrease the negative impact of straggling nodes on latency, they require excessive amounts of centralized control and unnecessary replication of tasks [1, 3].

Based on concepts in algorithmic fault tolerance, researchers have begun trying to lessen unnecessary waste in previous approaches and utilize concepts of erasure coding to improve performance in the midst of straggling workers. Initially designed to handle packet losses in unreliable communication protocols, erasure codes were designed to recover lost information through handling limitless amounts of packet losses or data erasures. Researchers have been able to adapt these techniques to handle computation through rapid data communication without worrying about the server not receiving information [2, 3].

Rateless codes are a modified version of erasure coding suited for data transmission at a non-fixed rate. Recent applications of these models to scenarios including matrix-vector multiplication have demonstrated promise as rateless coding schemes achieve low-latency and at least 3x speedup over alternative uncoded implementations. Particularly, rateless coding is beneficial due to its near-perfect load balancing, limited redundant computations, and tolerance to straggling nodes [3]. Thus, efforts in research are experimenting with alternative rateless coding implementations and applications. My proposed research topic will expand upon previous efforts in the field by applying rateless coding techniques to other computation applications including sparse matrices and non-linear computations.

Resources Needed

There are no additional resources needed for this project at the time, as all the resources necessary for this research are accessible through the lab I'll be working with. Previous rateless coding techniques use iMac Desktops for parallel computing measurements, AWS workers for distributed computing measurements and AWS Lambda for serverless computing measurements. Additionally, they use Dask as software for parallel computing in Python which is openly available for use. OPAL Lab currently has access to all these environments and software.

References

[1] Das, Anindya B., et al. "C3LES: Codes for Coded Computation That Leverage Stragglers." *2018 IEEE Information Theory Workshop (ITW)*, 2018, doi:10.1109/itw.2018.8613321.

[2] Mallick, Ankur, and Gauri Joshi. "Rateless Codes for Distributed Computations with Sparse Compressed Matrices." *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, doi:10.1109/isit.2019.8849306.

[3] Mallick, Ankur, et al. "Rateless Codes for Near-Perfect Load Balancing in Distributed Matrix-Vector Multiplication." *CoRR*, abs/1804.10331, 2018.