15-418 2021 Spring

Diego San Miguel (dsanmigu)
Nathan Ang (nathanan)

Distributed Machine Learning Using MPI and CUDA

Final Project Checkpoint

April 25th 2021

What We've Done So Far

Our goals for the first week of our project were as follows:

- Conduct research on N-bounded delay BSP parallelism
  - Reach goal: Schedule a meeting with 15-440 Professor Heather Miller to speak in depth about BSP parallelism and all-ringreduce
- Research how to get an MPI program running on multiple GHC machines rather than multiple different cores of the same GHC machine
  - Reach goal: Successfully configure an MPI program to do this
- Find a dataset to test our system on

We conducted more research on N-bounded delay BSP parallelism mostly by searching on google and finding examples of this algorithm implemented in the real world. We ended up not scheduling a meeting with Heather Miller since we felt that it was not necessary and we had the knowledge we needed to move forward with this project.

Getting MPI to run on multiple GHC nodes proved to be very difficult. Finding the MPI flags necessary to get a job to run on multiple hosts was fairly easy, however, the GHC compute cluster seemed to interfere with the requests and we ended up not being able to find a workaround. For now, we are confined to going multicore on a single node. We are either going to have to find a workaround in the GHC system, spend money on AWS VM credits, or pivot our project to permanently using a singular node.

Finding a dataset was quite easy. We created a Kaggle account, and after some digging, landed on this dataset about rain patterns in Australia. The dataset is fairly large (13MB) which will allow us to examine the speedup of our system in realistic machine learning scenarios. The data is also very easy to sort through; most features are float values and there are only a handful of features. The labels are simply Y or N, indicating whether or not it rained on a given day. This means we can likely just train a logistic regression model or very simple neural network. This will also help us spend less time on the machine learning side, and more time on improving speedup.

Our goals for the second week of our project were as follows:

- Find an implemented neural network to test the system with
- Implement all-ring reduce using MPI
- Begin researching how to implement N-bounded delay BSP through MPI
    - Reach goal: Get a starter version of this working

Initially we were thinking of using a Python machine learning library and then communicating between the C++ and Python programs through some pipes. This proved to be quite difficult for us so instead we did research on what the best C++ machine learning libraries were. We settled on Pytorch which has a beta C++ frontend. We've been able to get a basic deep neural network working locally and have found out how to access the models parameters.

All-ring reduce has been fully implemented in C++ using OpenMPI. We tried to make the implementation as generic as possible for whatever obstacles lie ahead. The ring reduce function takes in a function pointer for the reduction function that you want to use. The ring reduce function can also handle arrays of parameters of any reasonable size. The parameter arrays are partitioned into N pieces (N = number of nodes in the system) and then passed around between nodes in the reduce and sharing phases.

Major Problems

Not being able to get MPI to run on multiple GHC nodes has been a massive problem for us so far that may cause a large pivot in the direction of this project. The whole point of getting MPI to run on multiple nodes was so that we could get two levels of parallelism. We first split the training data amongst N nodes, and then each node can do the training in parallel, be it with the CPU or GPU.

If we can find a way to get this done we will be able to throw away this pytorch model we have been using and implement our own simple logistic regression model using CUDA (OpenMP will be a backup in case this proves to be too difficult.

If this MPI problem cannot be fixed, then we will have to resort to finding an alternative compute cluster and using whatever parallel programming framework is available on it.

We also are having trouble setting parameters within PyTorch's C++ due to minimal documentation and online resources on this topic.

Modified Remaining Goals

Bare Minimum:

- Implement Bounded-delay BSP
- Connect ringreduce program to the pytorch model
- Measure speedup of the program relative to a single core pytorch model training

Aiming for:

- Implement Bounded-delay BSP
- Get MPI to run on multiple nodes in the GHC cluster
- Create a simple logistic regression model in CUDA
- Measure speedup of the program relative to a single core running the logistic regression

Schedule

April 26th - May 2rd:

- Connect pytorch model to our ring reduce program
- Implement bounded delay BSP
- Measure speedup of program and construct graphs showcasing performance

May 3th - May 10th (Reach goals):

- If MPI is working on multiple nodes
    - Create logistic regression model using CUDA (or OpenMP if necessary)
- If MPI is not working on multiple nodes:
    - Find a compute cluster alternative
    - Create logistic regression model using whatever parallel programing framework available on alternative compute cluster

May 11th - May 13th (Report Due):

- Create and host a website through github pages
- Finish project report on hosted website