15-418 Project Checkpoint

Yijun (Jack) Dong, Enzhe Lu
yijund, enzhel

April 25, 2017

TITLE:

CuGB: Parallelizing Gradient Boosting on GPU *by Jack Dong and Enzhe Lu*

PROGRESS SUMMARY:

In the passing weeks, we have searched and learned both the algorithm and the xgboost tools. We also finished a simple test script in Python to check the correctness and the speed up of our code. We leveraged the xgboost python package and their demo datasets to be our baseline and our testing datasets. We have also done some research on the previous year project, but we found the code in the previous years is poorly documented and lack description to run them. Thus we decided to drop them from our reference and only use the xgboost as our primary reference.

We are currently writing a sequential gradient boosting in c++. We have finished constructing the decision tree, but still in the process of building gradient boosting part of the algorithm.

GOALS AND DELIVERABLES:

We believe we are able to achieve the goals we planned in our project proposal. If we have significant speed up during our project, we would like to combine our code with xgboost in order to make some contribution to the open source community. Here's a revised list of the goals we plan to achieve:

1. Successfully implement a parallel version of gradient boosting that can run on GPU.
2. Improve the performance of the GPU gradient boosting to achieve significant speedup from the serial xgboost version.

PLAN TO SHOW:

At the parallelism competition, we would like to show a graph that presents the speedup of our parallel gradient boosting run on GPU against the serial xgboost version run on CPU. In addition, we would like to show a second graph that displays the error rate of two versions on test datasets. If time permits, we could show a small demo of classifying a smaller dataset using the serial and parallel gradient boosting and comparing the runtime and error rate.

PRELIMINARY RESULTS:

We have finished a sequential version Decision-tree construction, which is only a starter code of the whole project.

ISSUES AND CONCERNS:

We are worried about the data structure overhead in the Cuda. Since Cuda does not support the standard library in C++, we have to use thrust as alternative library. However, both of us haven't work with thrust before. We are not sure about the data structure communication overhead in the thrust library. This might reduce our performance.

REVISED SCHEDULE:

Week 1 (April 16th): Done
Successfully run reference implementations to provide baseline and develop test harnesses for the project. Study gradient boosting in depth, and explore opportunities for parallelization.

Week 2 (April 23rd): In progress (working on parallelized version)
Develop a first and naive version of parallelized gradient boosting with CUDA. Test for correctness and performance against baseline implementation.

Checkpoint (April 25th): In progress (working on parallelized version)
Finish baseline, test harnesses and naive version of parallelized gradient boosting with CUDA.

Milestone 1 (April 27th):
Successfully finished sequential version of gradient boosting algorithm, and explore thrust library.

Milestone 2 (April 30rd):
Develop a first and naive version of parallelized gradient boosting with CUDA. Test for correctness and performance against baseline implementation.

Milestone 3 (May 1th):
Explore all potential opportunities for parallelism in the code. Develop a second approach of parallelization.

Milestone 4 (May 7th):
Continue to optimize code by testing performance and finding bottlenecks in the implementation. Potentially explore a third approach of parallelization, if a reasonable one exists.

Milestone 5 (May 12th):
Finish final writeup and prepare for the presentation.