Team Members

1. Dylan Hutchison (ECE Senior)
2. Eric Cherin (ECE Senior)
3. Di Ren (CS Senior)
4. Xuelian Liu (Graduate Student)
5. Mojie Yao (Graduate Student)
6. Hefei Yang (ECE Senior)
7. Xin Li (ECE Senior)
8. Jaroor Modi (Very enthusiastic freshman)
9. ~~Brian Cesar-Tondreau (ECE Senior, joined a different SeniorD group but wants to contribute anyway)~~
10. Peiran Guan (already completed SeniorD; joined just for TG 421)
11. Hanyu Jiang (PhD student with Ganesan)

Advisor Dr. Narayan Ganesan

Heterogeneous Graph Library

We aim to advance the state of graph algorithm performance in two parts:

1. Implement graph algorithms on heterogeneous architectures, including CPUs, GPUs and FPGAs, as part of Dr. Narayan Ganesan's research cluster. We will benchmark their performances and discern guidelines for when algorithms perform well on these new architectures.
2. Partnering with MIT CSAIL, we will help develop a graph library for the Accumulo distributed database. Defining graph data structures, ingest and scan routines, and database-level computation are within scope, but the majority of this work will occur at CSAIL, not at Stevens.

We will link the two together by using graph data queried from Accumulo through the library, in cluster computation. **The target deliverable is a library companies will use to do graph analytics on a heterogeneous architecture**. Our pipeline will set a performance and design precedent for graph processing at scale.

From a business perspective, we will sell our library as a product along with a support agreement. For a higher price, we will offer our expert knowledge as consultants in our special niche field: *high performance analytics on heterogeneous architectures*.

1. Explain why the team picked the final idea. [1 Point]

We're exploiting an opportunity—to take graph algorithms and implement them on new, nontraditional architectures. The graph algorithms are already known, but the implementation on GPUs and FPGAs are novel. We expect a considerable performance increase over naïve implementations, which companies with large data sets and a thirst for competitive edge by analytics will pay for. Plus, we have the skills and technical background necessary to pull this project off.

Possible Schedule

September

- experiment with different graph data structures,
- need to fix a language, preferably with high GPU library support.
- Implement basic algorithm on a single-core CPU, such as shortest path or triangle counting. Take data from a text file.
- Play with CUDA, explore possible graph algorithm implementations

October-November

- Fix a data structure. Implement a graph algorithm without GPUs and implement with GPUs. Benchmark; compare performance.
- Do more as we have time. Options are open- look to FPGAs, look to more advanced algorithms, ….

December

- Package our algorithms into a library.

Spring

- Bring in Accumulo as a data source. Look to distributed graph algorithms on a cluster, taking advantage of GPUs/FPGAs on multiple nodes.
- Expand library for distributed graph processing.
- Dylan will assist in linking with the Accumulo graph library from MIT.

Team Roles

[consider Dylan as part of every project]

TG Business Modeling

- Lead: Peiran Guan
- Hefei, Eric
- Completing TG assignments, modeling our project as a business

Website and UI

- Lead: Xin Li
- Di
- Create a website advertising our project (description, team members)

Datasets / Applications

- Lead: Di Ren
- Jaroor, Xin, _

- Implement (or even better, make use of existing libraries) to generate fake data. Focus on power law graphs.
- Create (or even better, find existing) toy and real datasets.
- Ask interesting questions for each dataset. We will answer these questions using graph algorithms. Example here.

Graph Algorithm Prototyping

- Lead: Dylan Hutchison
- Di, Eric, Jaroor
- Implement graph algorithms in a high level language, single-threaded CPU simple case.

GPU/OpenCL

- Lead: _
- Xuelian, Yao, Brian
- Implement graph algorithms on GPUs using OpenCL. If possible, also bind to multiple threads on CPUs using OpenMP / pthreads / something similar, for comparison.

FPGA

- Lead: _
- Xuelian, Yao, Brian
- Implement graph algorithms on a FPGA.

Performance / Benchmarking

- Lead: Di Ren
- _
- Create a framework for easily benchmarking performance, for CPU, GPU, FPGA, across different sized datasets, different number of cores, etc. Input is a C executable / library, output are timing datasets and charts.

Database

- Lead: Dylan Hutchison
- Jaroor, _
- Bring our work to the Accumulo distributed database. Coordinate with MIT.

Related Work

- All
- Understand the landscape of related projects out there. When you find something that seems relevant, write it in the Related Work document with a description on how it is / is not relevant, how we compare in both approach and performance. Or just write in your journal.