HBaaS: Heterogeneous-accelerated Bioinformatics-as-a-Service

Early Project Proposal, 13 October 2014

Senior Design Group 3
Stevens Institute of Technology
Sponsor: MIT Lincoln Laboratory and CSAIL

Team: Dylan Hutchison, Eric Cherin, Xin Li, Hefei Yang
"We pledge our honor that we have abided by the Stevens Honor System."

Advisor Dr. Narayan Ganesan

Collaborators: Di Ren, Xuelian Liu, Mojie Yao,
Jaroor Modi, Peiran Guan, Hanyu Jiang

Executive Summary (start in Fall, Flesh out in Spring)

Modern applications call for solutions that handle “big data,” datasets that span multiple machines and cannot fit in main memory, and “big compute,” computation patterns that tax even the most advanced processors. Bioinformatics is no exception.

We present HBaaS, Heterogeneous-accelerated Bioinformatics-as-a-Service. Our platform leverages heterogeneous computer architectures to provide sequence matching and motif finding as a web service for users of bioinformatics data. We tackle big data by means of the Accumulo distributed database, and big compute by means of GPUs. Their integration delivers top-tier performance for our chosen applications.

Table of Contents

Section – I: Project Definition and Plan (initial in Fall, refined in Spring)

I.1 Mission Statement

We aim to create an online Bioinformatics-as-a-Service (BaaS) platform, a web service providing sequence and motif alignment and search for proteins, accelerated by GPUs and the Accumulo DB. Our platform will run on a cluster of machines equipped with GPUs. Each machine will host data as part of the Accumulo distributed database and leverage their attached GPUs to accelerate parallel computation and data retrieval.

I.2 Background

Bioinformatics-as-a-Service (BaaS)

Bioinformatics is the use of mathematics and computer science to organize, analyze, and store the data generated in life science research and by the health industry. Bioinformatics often involves the development of software tools to understand biological processes through applications such as data mining, sequence analysis, gene and protein expression, protein structure modeling, and network and systems biology. Examples of bioinformatics applications include RNA and DNA sequencing, genome alignments and assemblies, mutation identification, microarrays, and database creation and management.

The very large amount of data that is collected in biological studies such as genomics or protein analysis must be processed using computers. Computer scientists combine algorithms, statistics, and mathematics, and engineering in order to make sense of the information.

HMMER is used for searching sequence databases for homologs of protein sequences, and for making protein sequence alignments. It implements methods using probabilistic models called profile hidden Markov models (HMMs).

Compared to BLAST, FASTA, and other sequence alignment and database search tools based on older scoring methodology, HMMER aims to be significantly *more* accurate and *more* able to detect remote homologs because of the strength of its underlying mathematical models. In the past, this strength came at significant computational expense, but in the new HMMER3 project, HMMER is now essentially as fast as BLAST.

We will call the HMMER algorithm on data queried from Accumulo using local GPUs.

Graphic Processing Unit (GPU)

A graphics processing unit (GPU) is designed to rapidly manipulate and alter memory to in order to accomplish a particular task. GPUs are commonly used in devices that require image processing such as computers, smartphones, and work stations. GPUs have a highly parallel structure which makes them very effective for processing data that can be done in parallel.

We will use GPUs for accelerated computation on data queried from accumulo.

Apache Accumulo

The Apache Accumulo sorted, distributed key/value store is a robust, scalable, high performance data storage and retrieval system. Apache Accumulo is based on Google's BigTable design and is built on top of Apache Hadoop, Zookeeper, and Thrift. Apache Accumulo features a few novel improvements on the BigTable design in the form of cell-based access control and a server-side programming mechanism that can modify key/value pairs at various points in the data management process.

In short, Accumulo is our database, delivering fault tolerance, distribution and availability for huge amounts of protein sequence and model data. Cell-based security may gain relevance if we need to restrict access to certain gene data.

I.3 Stakeholder List

Our target users range among physicians working in personalized medicine, forensic scientists working in crime scene identification, epidemiologists working in disease identification, genealogy consultants working in kinship analysis, biologists working in research generally, and other groups who use protein model-to-sequence scoring in their daily work. These users need scoring information fast, so that they may prescribe correct medication and catch criminals early, provide expedited genealogy services and more rapidly discover scientific advances.

Stakeholder Group	Stakeholder's Subgroup	Areas of Interest
Project Team	Senior Design Group 3 Project Advisor Other Team Members	Great speedups in protein searching compared with current application
Customers	Biologists Physicians Forensic scientists Epidemiologists	Efficient and accurate protein model-to-sequence scoring Easy-to-use UI Reliable resources
Competitors	HMMER BLAST	More market share Better application performance
Scientists	Bioinformatic Scientists	Improvements in bioinformatic searching
Project Sponsor	SIT MIT Lincoln Laboratory and CSAIL	Cost, revenue and profit Students with potential and talent

I.4 Analysis of Stakeholder Needs

I.5 Project Scope and Resources

- Project objectives

  We aim to provide two service classes: *hmmsearch* and *hmmbuild*.
  - In *hmmsearch*, a user provides HMMs (Hidden Markov Models) representing protein sequence motifs, and requests the top-scoring sequences for each model, from a database of protein sequences stored on our Accumulo platform.
  - In *hmmbuild*, a user specifies a range of sequences and requests the HMM motif that best represents the consensus between the sequences.

- Goals

We will compare our performance to the existing HMMER search application online at http://hmmer.janelia.org/. We aim to show great speedups by leveraging GPUs and the Accumulo DB.

We envision the following outline of our service:

1. A user submits a query via our web interface.
2. Our web server backend creates a customized query-and-compute request for the Accumulo database and launches it.
3. In parallel, each machine on the Accumulo database searches through its locally stored protein sequences. The top scoring results are returned to the web server backend.
4. The web server backend forwards the results to display on the web interface.

- Resources
  - Stevens research cluster of 8 machines, all equipped with GPUs
  - Matlab Parallel Computing Toolbox, license courtesy of Stevens

I.6 Project Plan

- Install Accumulo on a local research cluster
- Create iterators within Accumulo that control HMMER computation on GPUs.

- Create the back end of a web server that accepts web requests, constructs lookup and computation queries, calls Accumulo and returns the results, ideally asynchronously and with a separate thread for each request
- Create a front end query interface for a web browser and add visualizations
- Benchmark and test
- Test scalability on large clusters at the MIT Lincoln Laboratory

Section – II: Design, Evaluation & Optimization (initial in Fall, refined in Spring)

II.1 Requirements

Users must be able to ingest data and complete queries via our web service without error. Users should be able to do so in parallel.

II.2 Constraints and Assumptions

The web service will not process more requests than we can handle with our cluster resources. We will know this constraint better once we start benchmarking.

We will not store more data than we have hard disk space, about 1TB on Dr. Ganesan's server.

We assume data and queries are in correct format. We don't want to error check every sequence, model and query as this costs development time and possibly performance.

II.3 Applicable Codes/Standards/Regulations

OpenCL is an API industry standard for GPU computation. CUDA is another API specific to NVIDA. We aspire to conform to OpenCL to guard against vendor lock-in but will use CUDA at first since it has greater support (more libraries, more examples, more of our team has background knoweldge in CUDA)

II.4 Concept Development and Selection

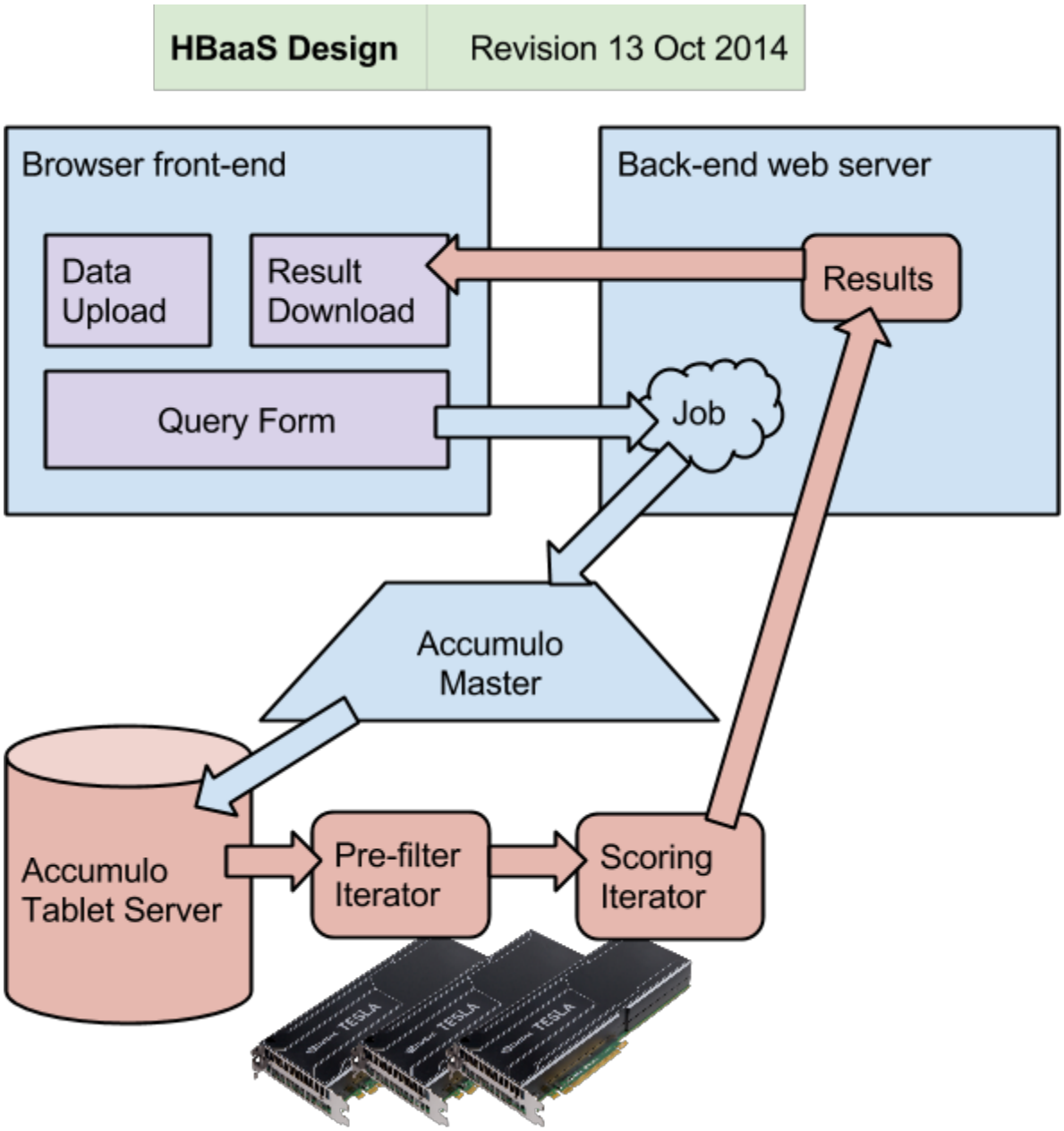
Our original project idea was to create a distributed graph library, offering graph algorithms on the Accumulo database that leverage local GPUs. The GPUs accelerate computation and Accumulo enables scalability to big datasets.

After investigating the graph library idea, we decided that it was too abstract. The team had trouble connecting with the project idea and motivation.

This sparked our shift to specialize on an application in bioinformatics. Dr. Ganesan and other students had some experience with biological computation, and everyone found the application more inspiring.

We are still toying with choice of web server. We chose Accumulo for its performance, for its extensibility to call GPUs within iterators, and for compatibility with our sponsor.

II.5 Preliminary (Fall) and Detailed Design (Spring)



II.6 Design Evaluation Methods

Benchmarking is the name of the game. We will measure performance in terms of run times and plot the scalability of run times against query size, motif HMM size, and number of computing resources available.

II.7 Design Evaluation Report: Performance, Reducibility and Cost

We will compare against HMMER's web service at <http://hmmer.janelia.org/search/hmmsearch> as a baseline. We expect performance improvements as a result of using Accumulo and GPUs.

II.8 Design Revisions and Optimizations (Fall/Spring)

II.9 Final Design Specification with BOM (spring)

Section – III: Entrepreneurship & Business Development (primarily TG course requirements)

III.1 Business Objectives and Risks (Fall)

- Business Objectives:
    - become the fastest bioinformatics service
    - Promote heterogeneous architecture as a solution for speeding up real world processes
- Risks:
    - Time put into project
    - Electricity bills for running GPUs
    - matlab and other software costs money

III.2 Competitive Intelligence: Market Analysis (Fall)

Our competitors are other companies offering data analytics, such as Sqrrl and Argyle data. Their products are Sqrrl Enterprise and ArgyleDB, respectively. We target algorithms for which GPUs and FPGAs offer exceptional speedups beyond what can be provided by a traditional database. This is particularly relevant for problems with computational bottlenecks, not just data bottlenecks. (big compute in addition to big data)

One other dynamic is the target customer base. Argyle's prime customers are financial companies in the context of fraud detection. Sqrrly's prime customers are in the healthcare industry, and other industries that need a strong privacy / access control layer on their data.

We aim to target companies whose analytic interests are scalable on GPUs and FPGAs. We know this is true of graph algorithms and we also know that there are a large range of companies looking to GPU/FPGA solutions since they reached the limits of how well they can compute via ordinary CPU means. As we develop the project, we will gain a better understanding of what applications benefit most from our approach.

We picked a target application to show the potential of our graph library: protein motif scoring, the scoring of possible motif models for given protein sequences. The real world use case is in personalized medicine: matching each person's protein sequences to an array of drugs that act on them through different motifs.

Two competitors for protein scoring are HMMER and BLAST, two tools affiliated with the NCBI. We aim to implement HMMER on GPUs within a database, so we expect our performance to scale much higher than vanilla HMMER. We're not sure where we will stand with respect to BLAST.

III.3 Lean Canvas Business Model (Fall)

- **Problem**:
    - Need better graph algorithm performance for analytics
    - Would like to use FPGAs and GPUs to improve performance, but don't know how

- **Customer Segments**: Companies that have large data sets that want insight on data such as real time analytics.

- **Unique Value Proposition:** Create a library that speeds up searches and improves performance in very large databases

- **Solution:**
    - Use our library that we developed
    - Use our expertise to support companies that would like to speed up their analytic processes

- **Channels:** publish performance numbers in industry journals, directly market to companies, and use word of mouth in our specialized community.

- **Cost Structure:** Pay salaries and purchase software licenses and hardware.

- **Revenue Streams**: Companies must pay consulting fees and support agreements. Since we aim to open-source our library, primary revenue is by companies paying for a support agreement for us to help them setup, use, and maintain our library.

- **Key metrics**: The graph library can be tested for performance by looking at floating point operations per second and benchmarking for time performance. We can view performance graphically by looking at scalability curves (computation time vs. number of GPUs, CPU cores, nodes, …)

- **Unfair Advantage**: This is a niche field so developing algorithms and a library from scratch will take a lot of resources and time.

III.4 Financial Analysis (Spring)

III.5 Intellectual Property (Spring)

Section – IV: Results

IV.1 Conclusions

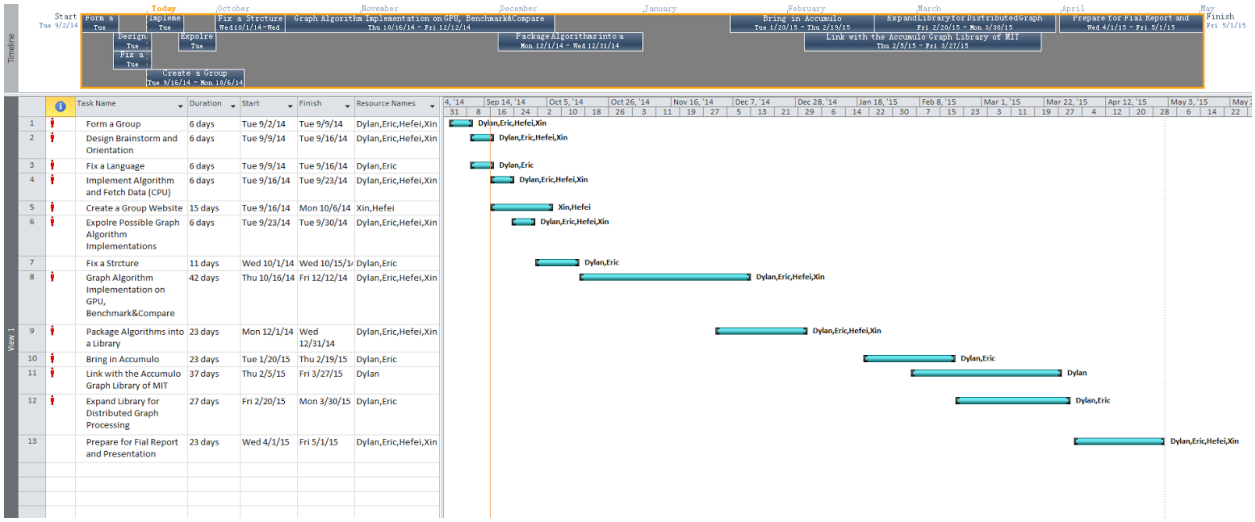
IV.2 Recommendations

Appendices

A Team organization chart

Team Member	Major	Role
Dylan Hutchison	BE Computer Engineering, MS Computer Science, MS Applied Mathematics	Team leader, Accumulo
Eric Cherin	BE Electrical Engineering MS Computer Engineering	Accumulo
Xin Li	BE Computer Engineering	Web server front-end
Hefei Yang	BE Electrical Engineering	Web server
Di Ren	BS Computer Science	Web server back-end
Mojie Yao	ME Computer Engineering	GPU
Xuelian Liu	ME Computer Engineering	GPU
Jaroor Modi	BS Computer Science	Theory and Algorithms
Peiran Guan	BE Computer Engineering	Business Modelling
Hanyu Jiang	PhD Computer Engineering	HMMER Support

B Project Gantt Chart



C Prototyping and Testing Budget (projected – Fall, Actual –Spring)

D Design Documents: Drawings, Layouts, Analysis reports

Our code is online here: https://github.com/Stevens-GraphGroup/

E Team Logistics Systems

We used several services to communicate and store information:

- Github - repository to share code and the website
- Todoist - online task manager for productivity
- Google Drive - file storage and synchronization service

F References cited

http://www.census.gov/cgi-bin/sssd/naics/naicsrch?code=511210&search=2012

http://www.census.gov/eos/www/naics/

http://www.supercomp.org/

http://thedataweb.rm.census.gov/TheDataWeb_HotReport2/econsnapshot/2012/snapshot.hrml?NAICS=511210

http://www.ibisworld.com/industry/default.aspx?indid=1239

http://sqrrl.com/product/sqrrl-enterprise/

http://www.argyledata.com/product/

http://www.ncbi.nlm.nih.gov/

http://blast.st-va.ncbi.nlm.nih.gov/Blast.cgi

http://hmmer.janelia.org/

https://accumulo.apache.org/

http://blast.st-va.ncbi.nlm.nih.gov/Blast.cgi