Friday, February 14, 2014

BLASTn Notes V-Day edition

Project goals

- Query for our program based on NCBI format
  - We want it to run fast as possible
  - We want to take any possible query and give same answers, faster.
  - We want to get an exact result.
- What parts of the various implementations do we like?
  - take those parts
  - leave out parts we don't like
- We should likely preprocess the database
  - How do we store it on our FPGA?
  - create database file
    - software routine to transfer/store database in hardware
    - Human genome database is 8GB? (nucleotide)
    - store hash table for database on FPGA?
    - NCBI site has 4^11 database of all possible entries
  - To-do: Look into format of databases
  - To-do: Run software and determine organization of database
- We could use two databases?
  - one for queries, nearest neighbor information
  - one for database, which strings include search term
  - Question: What is avg. length of a query? This number not official, find studies
- How are we going to handle our data?
- How are we going to transfer it
- How are we going to access it?

- We must consider the consequences of each question
  - Finite processing power
  - finite access to memory
  - finite resources
- We should try at first without much parallelism
  - will likely have to add parallel abilities afterward
- Designing an architecture requires defining the software/hardware partition
- We should try to run and profile NCBI Blast software for performance
  - Run different queries and lengths/numbers of queries
  - Benchmark with different genome databases
    - Run gprof on the software
    - See what parts are most time-consuming
  - Use an iterative method to increase speed
    - tackle most time-consuming methods
    - e.g, 70-80% was spent on scantask (finding the hits)
  - Is the throughput or the latency a problem?
    - If latency not a problem, we could provide a lot of pipes
    - send multiple queries at once on FPGAs
    - Likely that throughput can be added by multiple FPGAs, focus on latency for one
  - We should aim to process one query at a high rate
    - Is one user ever processing one query at a time?
    - Or does it run multiple jobs in a row?

Goals

- Goal #1: Do everything the NCBI software is doing
- Goal #2: Speed it up.
  - observe time/performance tradeoffs

  - use different amounts of queries and different query lengths
  - We should develop benchmarking system tool (use gprof)
    - compare our version vs their version

Tasks for next week

- Designing an Architecture (TASKS REQUIRED)
  - Profiling (what sections to speed up) - Taylor
  - Become expert on NCBI software operations - John
  - develop target specification
    - goal of the system
    - what speed can we hit
    - if having a question, refer to #3
  - generate benchmarks - Yash
    - write scripts to take in different queries
    - don’t focus on what queries yet, just a base script
    - work for NCBI software
    - Find average query length
  - analyze prior work (what parts we like and should use) - Nekhil
  - Propose architecture
  - consider and explain design decisions
  - Analyze scalability

Things to look into to help with above tasks

- Look into Mercury BLAST
  - 62 pages
  - NIH Public Access paper
- Herbordt BLASTn paper
  - Throughput was an issue

    - Bottleneck slowed it down
    - Pico can do 10x that speed, might want to look into
- Short read Archive
    - massive database of publicly available sequences
    - confusing to navigate
    - a lot of sample sequences to run through our profiler
    - be sure to note what we downloaded and provide link to study in case of publishing our work later on
- NCBI has the databases that we would want to analyze
    - Human Genome
    - E. coli
    - Mouse
    - Virus database desirable too
- 50 hours of work per week required for this project minimum