GPMoo: Genomic Selection Related Analyses

Scott Winkleblack, Trevor DeVore, Dr. Chris Lupo
California Polytechnic State University, San Luis Obispo, CA
{swinkleb, tdevore, clupo}@calpoly.edu

Abstract

We present a tool, GenSel, which can be used to efficiently infer the effects of genetic markers on a desired trait or to determine the genomic estimated breeding values (GEBV) of genotyped individuals. To predict which genetic markers are informational, GenSel performs Bayesian inference using Gibbs sampling, a Markov chain Monte Carlo (MCMC) algorithm. Parallelizing this algorithm proves to be a technically challenging problem because there exists a loop carried dependence between each iteration of the Markov chain. Therefore, our approach seeks to exploit the fine-grained parallelism that exists within each iteration of the Markov chain. More specifically, we use CUDA to speed up the sampling of each genetic marker used in creating the model. Related works in parallelizing MCMC algorithms have resulted in speedups of 10’s to 100’s of times [1,2]. Achieving a speedup of similar magnitude will allow us to parallelize this algorithm to accommodate the expected increase in observations on animals and genetic markers per observation. Our current implementation executes up to 3 times faster than the optimized CPU implementation.

Algorithm

Gibbs Sampling

Basic statistical model:

\[ y = X\beta + u + \sum_{k=1}^{K} z_k a_k + e \]

\( y \) is a vector of observable traits
\( \beta \) is a vector containing the genetic fixed effects for a genetic marker
\( X \) is an incidence matrix that relates fixed effects to observable traits
\( z_k \) is a \( N \times 1 \) vector of the genotypes at marker \( k \)
\( a_k \) is the additive effects of marker \( k \)
\( e \) is a vector of residual effects

Results

The accuracy of the parallelized version of GenSel was verified against the original, serial implementation. All results were generated on a system with the following specifications:

- Intel Xeon E5-2650 x 2
- 8 cores
- 2 GHz
- 64 GB RAM
- Tesla K40
- 2880 cores
- 12 GB memory
- 6 GB memory
- Tesla K20X
- 2688 cores
- 6 GB memory

Future Work

- Multi-card CUDA Support
- Incorporate optimized CUDA libraries (CUBLAS and Eigen)
- Port more of the statistical analysis associated with MCMC to CUDA
- Further leverage the features supported by the Kepler architecture
  - Streams
  - Asynchronous memory transfer
  - Dynamic Parallelism

Works Cited
