Nathan T. Weeks[1,2], Glenn R. Luecke[1], Brandon M. Groth[1], Marina Kraeva[3,4],
Li Ma[4], Luke M. Kramer[5], James E. Koltes[6], James M. Reecy[5]

1. High Performance Research Group, Department of Mathematics, Iowa State University
2. Department of Computer Science, Iowa State University
3. High Performance Computing Group, IT Services, Iowa State University
4. Department of Animal and Aviation Sciences, University of Maryland
5. Department of Animal Science, Iowa State University
6. Department of Animal Science, University of Arkansas

High Performance Epistasis Detection in Quantitative Trait GWAS for Intel Xeon Phi Clusters

Definitions

➜ **Genome-Wide Association Study (GWAS)**
  ◆ a statistical association of associating genetic variation, usually Single Nucleotide Polymorphisms (SNPs), with phenotypes (observable traits) in a group of individuals
➜ **quantitative trait**
  ◆ a phenotype that is expressed as a quantity (e.g., height, weight).
➜ **epistasis**
  ◆ the interaction of genes to determine a trait

Summary

- epiSNP[1] is a program for detecting pairwise epistasis in quantitative-trait GWAS using least squares analysis. This is is computationally expensive: $O(n^2)$ pairwise genetic marker tests (where n is the number of SNPs). Consequently...
- EPISNPmpi was created in 2008 to run on distributed-memory compute clusters. However...
- Sequencing and genotypic are getting much cheaper, meaning bigger data than EPISNPmpi was designed to handle. So...
- We optimized epiSNP to run much faser on host CPUs, as well as the Intel Xeon Phi coprocessor[2][3]. Specifically...
- Using a benchmark data set of 774,660 SNPs with 1,634 individuals and a single trait[4], we improved performance of epiSNP on 126 nodes of the Stampede supercomputer by an average of:
  ○ parallel performance (CPU only):: 16.7X
  ○ parallel performance (with 1 MIC per node): 29X
  ○ parallel performance (with 2 MICs per node): 39X

Problem

EPISNPmpi was adequate to handle what was considered large-scale GWAS 2008; however, it:
1. suffers from load imbalance (Figure 2)
2. can only be run on a certain ("triangular") number of MPI processes, placing restrictions on the number of processor cores/nodes it can utilize
3. cannot efficiently utilize the Intel Xeon Phi (Figure 2) due to constraint (2), in addition to memory constraints, and communication bottlenecks from multiple MPI processes on the same coprocessor
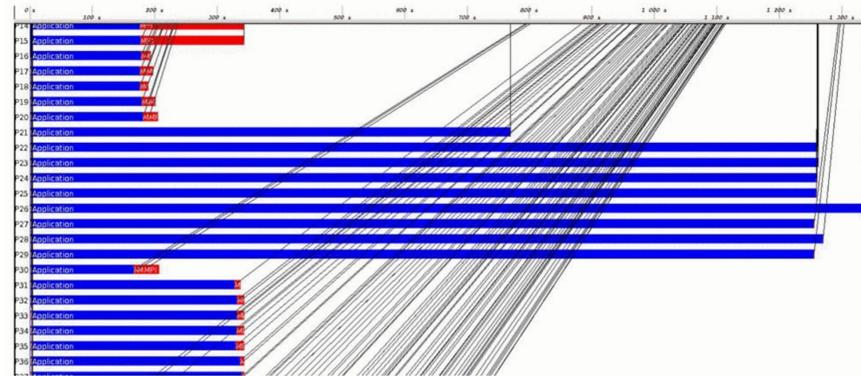4. has suboptimal serial performance

Figure 1. Intel Xeon Phi coprocessor (aka MIC). Specs:
- ~60 cores
- 4 hardware threads per core
- 8-16 GB GDDR5 memory
- 512-bit vector registers
- over 1 teraFLOPs double-precision peak performance

Figure 2. EPISNPmpi execution profiling revealed severe load imbalance. This Intel Trace Analyzer Event Timeline shows MPI ranks 14-37 (out of 56) Each horizontal bar is an MPI process (blue: program execution; red: MPI communication). Black lines connect communicating processes.

Figure 3. Stampede supercomputer at TACC (source: Texas Advanced Computing Center)

Solutions

- Code profiling to identify bottlenecks
  ○ **Result:** > 99% of serial runtime was in a single subroutine, identifying where to focus optimization efforts. Data-distribution of parallel execution resulted in severe load imbalance (Figure 2)
- Extensive serial optimizations (data-type changes, data-structure changes, dead-code elimination, reduce branching to help vectorization)
  ○ **Result:** serial performance is ~15X faster than EPISNPmpi
- Code modularized, simplified, and clarified with modern Fortran 2008 constructs.
  ○ **Result:** lines of source code reduced by ~⅓. Code now portable to any standards-compliant Fortran 2008 compiler.
- OpenMP-based multithreading adds shared-memory parallelism to every MPI process
  ○ **Result:** memory footprint greatly reduced and load balancing enhanced. Now runs on Intel Xeon Phi coprocessor, with performance about 1.5X that of host Xeon on Stampede
- Dynamic load balancing mechanism implemented using MPI-3 Remote Memory Access (RMA) between MPI ranks, and OpenMP within an MPI process
  ○ **Result:** allows flexible number of MPI processes, allowing arbitrary number of host CPUs and Xeon Phi coprocessors to be utilized

Figure 4 (right). Workflow diagram of original EPISNPmpi (left) and our optimized epiSNP (right). Red shading indicates the process utilizes shared-memory parallelism (OpenMP). Input files are read by rank 0 and broadcast to other ranks; this was parallelized using OpenMP so each thread read a separate input file. Profiling revealed that the single locus analysis wasn't a bottleneck, so this wasn't parallelized in the optimized epiSNP.
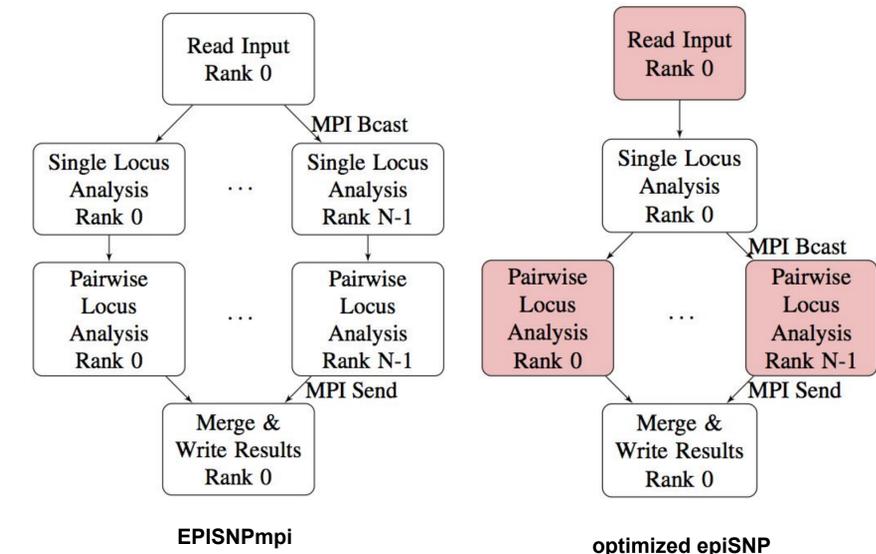

EPISNPmpi   optimized epiSNP

Table 1. Run time in minutes (Stampede)

nodes	EPISNPmpi	MPI3 + OpenMP	MPI3 + OpenMP 1 MIC	MPI3 + OpenMP 2 MIC
1	N/A	1421.89	767.63	505.01
2	N/A	659.90	361.19	250.05
4	N/A	325.97	178.10	129.88
8	2338.34	158.91	88.94	65.01
16	1360.06[1]	79.12	44.50	32.70
33	649.93	38.44	21.82	16.12
65	335.68	19.79	11.39	8.58
126	185.00	10.42	6.25	4.81

Table 2. Node hours (Stampede)

nodes	EPISNPmpi	MPI3 + OpenMP	MPI3 + OpenMP 1 MIC	MPI3 + OpenMP 2 MIC
1	N/A	23.70	12.79	8.42
2	N/A	22.00	12.04	8.34
4	N/A	21.73	11.87	8.66
8	311.78	21.19	11.86	8.67
16	362.68	21.10	11.87	8.72
33	357.46	21.14	12.00	8.87
65	363.65	21.44	12.34	9.29
126	388.50	21.89	13.13	10.10

Tables 1 and 2. Benchmark results. While table 1 lists the run time in minutes, table 2 lists the node hours consumed during the analysis, which corresponds to the allocation that a user is charged. No time was available for EPISNPmpi at 1, 2, or 4 nodes due to exceeding the queue wall time limit.

References

1. Ma L, Runesha HB, Dvorkin D, Garbe JR, Da Y. Parallel and serial computing tools for testing single-locus and epistatic SNP effects of quantitative traits in genome-wide association studies. BMC Bioinformatics. 2008 Jul 21;9:315. doi: 10.1186/1471-2105-9-315.
2. Luecke GR, Weeks NT, Groth BM, Kraeva M, Ma L, Kramer LM, Koltes JE, Reecy JM. Fast Epistasis Detection in Large-Scale GWAS for Intel Xeon Phi Clusters. Trustcomm/BigDataSE/ISPA, 2015 IEEE, vol.3, no., pp.228-235, 20-22 Aug. 2015. doi: 10.1109/Trustcom.2015.637
3. Weeks NT, Luecke GR, Groth BM, Kraeva M, Ma L, Kramer LM, Koltes JE, Reecy JM. High Performance Epistasis Detection in Quantitative Trait GWAS. International Journal of High Performance Computing Applications. [In Revision]
4. Kramer LM, Koltes JE, Fritz-Waters ER, Mayes M, Markey A, Saatchi M, Weeks NT, Reecy JM. 2014. Identification of Epistatic Interactions Among Fatty Acid Traits, in Angus Sired Beef Cattle. The proceedings, 10th World Congress of Genetics Applied to Livestock Production. 17-22 August 2014.