Combining OpenMP, MPI, and Homomorphic Encryption for a privacy-preserving, parallelized GWAS

Project Update by
Noah L. Schrick
for CS-6643 Bioinformatics

Data

- Artificial Data
- 500,000 samples
- 10,000 variables
- ~ 4.5 GB

FHE Approach

- Used the implementation from [1]
- PALISADE
  - Open-Source Lattice Cryptography
  - Used the Residue Number System variant of the Cheon–Kim–Kim–Song (CKKS) HE scheme

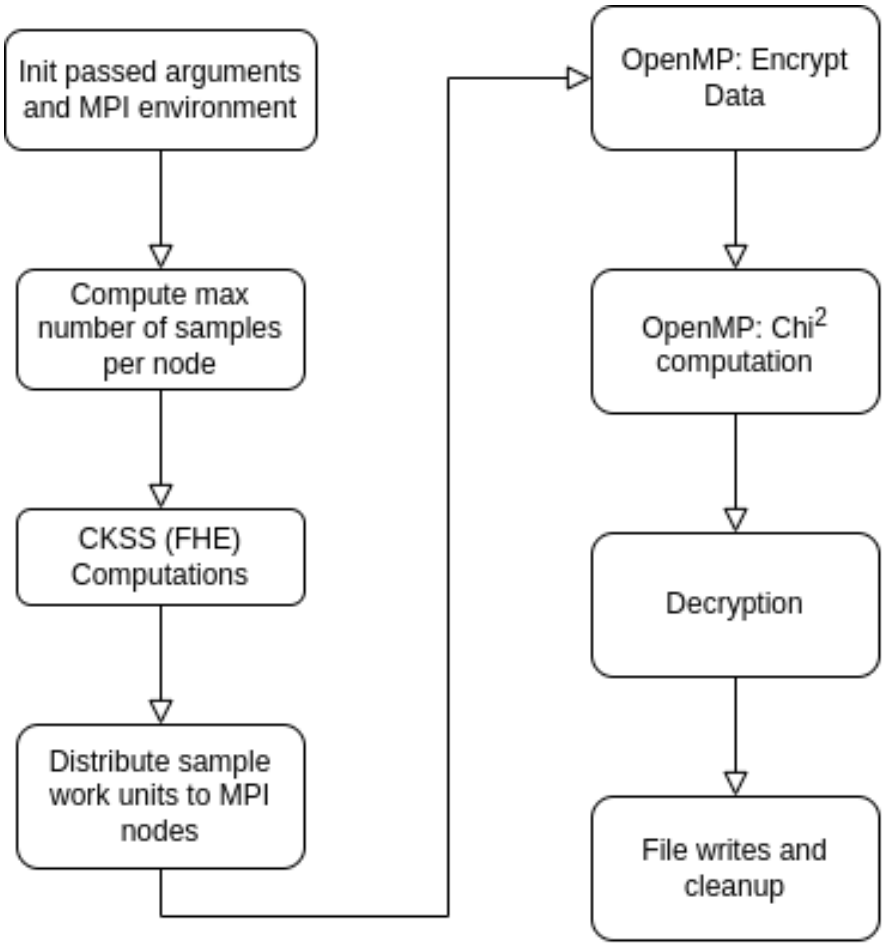
Approach Foundation

- Implemented in C++
- χ^2
- OpenMP Usage:
  - Encryption of ciphertexts, and
  - Over the number of variables
- MPI
  - Distribution of samples

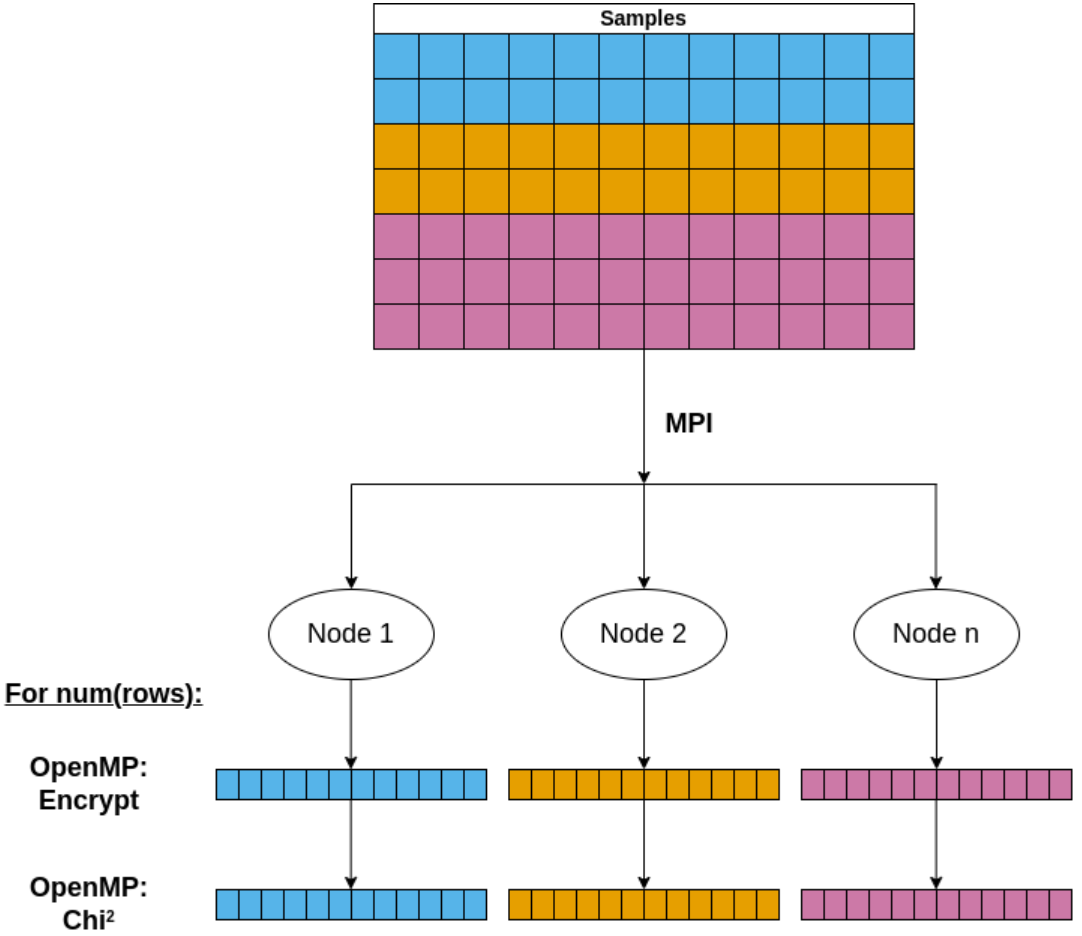
Scalability Challenges

- Memory Consumption
  - Unable to load all data into memory
  - Arbitrarily loading a constant set of data is not portable and stifles performance
  - Load a set of data proportional to memory capacity

Program Flow



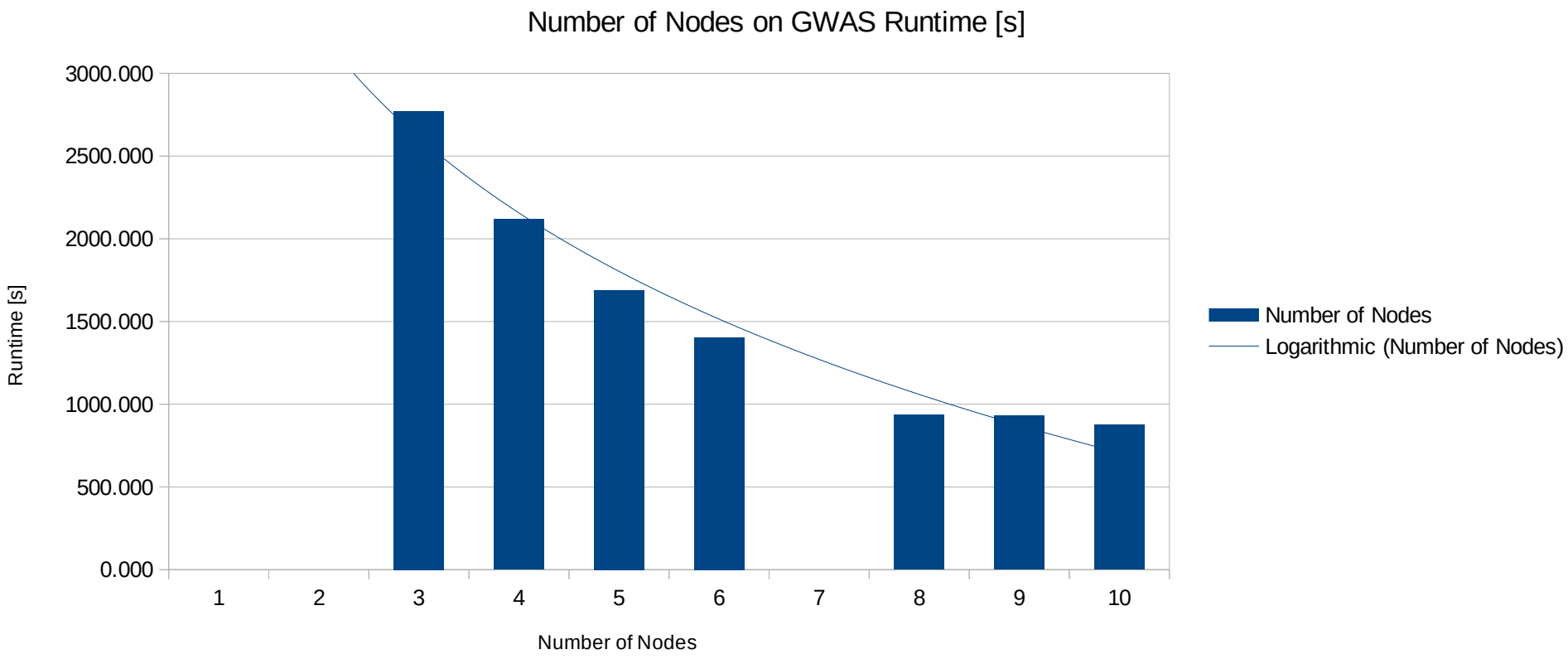
Parallelization Diagram



Preliminary Results

- Results are compared to the author’s results using OpenMP
- Results do not currently include the serial benchmarking, as this work was only intended as an expansion of the author’s original work
- All results are left independently. See Future Work section for intent to expand

Preliminary Results



Remaining Work

- Finalize Data Collection
- Speedup metrics using Amdahl’s
- Timing of individual tasks / speedup metrics of tasks

Future Work

- Build a map of the file and line positions
  - Requires additional pre-processing computation, but would reduce overall runtime.
- Meta-analysis of chi^2 results
- Vary problem sizes to examine efficiency of parallelization
  - Also examine scalability (strong vs. weak)

Update Presentation References

[1] M. Blatt, A. Gusev, Y. Polyakov, and S. Goldwasser, “Secure large-scale genome-wide association studies using homomorphic encryption,” Proceedings of the National Academy of Sciences, vol. 117, no. 21, pp. 11608–11613, May 2020, doi: 10.1073/pnas.1918257117