Date: October 9, 2020

Prepared by: Peter Muchina

This report is to provide an update on the Parallelizing long-read de novo genome assemblers on mixed architectures (CPU/GPU) with OpenACC and CUDA project. This project will result in the development of new algorithms for accelerated long-read de novo assembly on GPU architectures.

Summary of work in progress:

From 2-9th of October, i have completed the following:

i. Looked into communication between Parabricks and Assemblers.
ii. Reading on GPU-Accelerated Large-Scale Genome Assembly
iii. Developed simple python algorithms for genome assembly

Communication between Parabricks and Assemblers

Parabricks is a computational frameworks that supports genomic application from DNA to RNA. It uses NVIDIA's CUDA, HPC, AI and data analytics to build GPU accelerated libraries, pipelines and reference application work-flows for analysis. Clara Parabricks zeros down support new application development to address the needs of genomic labs. Major essence of the pipelines built is to optimize, acceleration, accuracy and scalability.

Features:

a) CudaAligner: Myer's and Hirchberg's algorithms
b) CudaPOA: python API and optimized implementation of partial order alignment(POA)
c) Cuda support

Interesting applications that fall in line with the project are:

Raven: which is an application for de novo assembly of long uncorrected reads that utilize cudaAligner (accelerated alignment) and cudaPOA (accelerated polishing)

Racon: SIMD-accelerated, POA based, stand alone consensus module. It is also based of long-read data. Coupled with miniasim enables genome with similar or better quality than many methods and is a much faster.

GPU-Accelerated Large-Scale Genome Assembly

Based on a paper that develops a GPU-accelerated genome assembler, called LaSAGNA. This tool seeks to address the fundamental limitation of GPUs in large-scale genome assembly. It can assemble datasets with billions of sequences using a single GPU by building string graphs from approximate all-pair overlaps. LaSAGNA significantly reduces the memory requirement by employing a semi-streaming approach that minimizes the number of disk accesses based on the available memory. It can also run on multiple GPUs across multiple compute nodes to expedite the assembly pipeline. Among many other contributions,LaSAGNA can build an approximate overlap graph from a real-world human genome sequence dataset (several hundred GB in size) using a single GPU equipped with only 6 GB device memory.

Developed simple python algorithms for genome assembly

With the help of Ben Langmead tutorials, i was able to develop algorithms for overlap finding. Two algorithms:

1) The shortest common superstring(scs): Based on brute force. Not a very efficient solution as it could be very slow for large chunks of read.
2) Greedy shortest common string: Much faster but comes with an accuracy cost.

Tasks to be accomplished by 16th October 2020:

a. Polish on the python algorithms and try them on a larger data set.
b. Continue with the literature review

Conclusion:

The project tasks are on track to be completed by the scheduled date. The project currently has no obvious obstacles or issues, but if they should arise they will be promptly raised to the supervisor.