DPHPC group: Kosta Stojiljkovic, Han Yao Choong, Langwen Huang, Benjamin Langer

PROJECT PROPOSAL - Parallelization of popular genome sequencing algorithms

In bioinformatics, algorithms that compare to strings for similarity play a decisive role in genome sequencing. Although today's research is dominated by programs like BLAST that produce approximate solutions on large datasets, they fundamentally rely on dynamic algorithms such as:

1. Smith-Waterman algorithm[1]
2. Needleman-Wunsch algorithm[2]
3. Hirschberg's algorithm[3]

These three, closely related algorithms all fall under the category of dynamic programming algorithms for finding the edit distance of two strings (i.e. a measure for their dissimilarity). Our project will consist of exploring different methods for parallelization of these fundamental algorithms while using different techniques (MPI, threading, combination etc.).

In a second step, our research will try to look into the more advanced methods utilized in bioinformatics. The widely used BLAST toolbox[4] (which relies on heuristics due to the generally large data amounts) and minimap[5] which specializes in processing long and noisey (i.e. error-prone) sequences will both serve as a benchmark and loose point of reference for our implementation.

Where possible we will try to relate our work to existing research on parallelization of the aforementioned algorithms and benchmark against those projects if applicable.[6]

Rough project timeline (Oct 11, 2019 - Dec 17, 2019) - 10 weeks

- **Weeks 1-3 (Oct 14 - Oct 31):** Research into dynamic programming parallelization and current implementations and parallelization of algorithms of interest
- **Weeks 4-8:** Work/testing of different implementations
- **Weeks 9-10:** Benchmarking different implementations and prepare for the presentation
- **Until Jan 17, 2020:** Preparing the final report

---

[1] https://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm
[2] https://en.wikipedia.org/wiki/Needleman%E2%80%93Wunsch_algorithm
[3] Hirschberg, D.S., Algorithm for Computing Maximal Common Subsequences, 1975, http://www.mathcs.emory.edu/~cheung/Courses/323/Syllabus/DynProg/Docs/Hirschberg-LCS-1975.pdf
[4] https://blast.ncbi.nlm.nih.gov/Blast.cgi
[5] Li, H., Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences, 2017, https://arxiv.org/pdf/1512.01801.pdf and https://github.com/lh3/minimap2
[6] E.g. Rashid, N.A. et al., Parallel Homologous Search with Hirschberg Algorithm: A Hybrid MPI-Pthreads, 2007, Solutionhttp://eprints.usm.my/9497/1/Parallel_Homologous_Search_with_Hirschberg_Algorithm_A_Hybird_MPI-Pthreads_Solution.pdf