Dhruv Jain

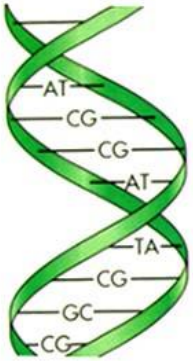
Amit Jaiswal

Graphical Processing Units

As accelerators for

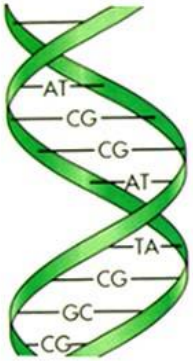
NEXT-GENERATION SEQUENCING (NGS)

Applications



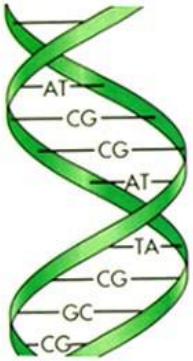
Brief Introduction

- Sequence Alignment
  - Discipline of Bioinformatics which concerns itself with arranging sequences of DNA, RNA or protein to identify regions of similarity.
  - Usually a small query sequence aligned against large reference gnome
- Next Generation Sequencing
  - Reads small pieces between 20 and 1000 bases, depending on the technology used.
  - So, there is a need to speed up alignment to save processing time
- Use of Fast alignment algorithms
- GPUs to achieve parallelization of sequencing



Methodology

- **Profiling**:
  - Code of the open source software tools obtained from web
  - Profiled on running on real data. Compute intensive kernels have been identified
- **High Level Performance Estimation**:
  - Rough estimate of the resources consumed by the kernel on the GPU
    - Estimation of the number of functional units to exploit parallelism.
    - Estimation of communication interface and memory hierarchy
- **Selection of desired algorithm**
  - Based on the performance estimation
- **Hardware-Software Co-design**
  - Code divided into specific parts to execute on GPU; Processor



Profiling Results

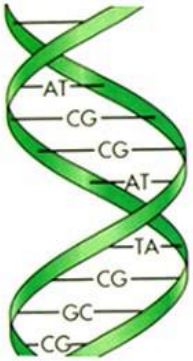
Chromosome	Size of the reference genome(MB)	Algorithm	No. of bp in the query sequence	Time taken to index(s)	Time taken to align(s)
10	131.8	maq	70	40.4	741.6
		soap	70	194.1	79.4
		bwa*	70	232.4	315.2
		bwa-sw*	500	232.4	8838.5
19	57.5	maq	70	31.5	865.2
		soap	70	78.2	84.2
		bwa	70	92.7	359.4
		bwa-sw	500	92.7	8030.5
X	151	maq	70	41.1	802.7
		soap	70	225.8	86.4
		bwa	70	267.5	342.6
		bwa-sw	400	267.5	11269.8

Profiling Results of BWA

Algorithm	Function	% Contrib	Remarks
bwa-index	bwa_index	100	It calls a lot of functions each consuming a significant amount of time.
bwa	bwa_cal_sa_reg_gap	99.1	bwa_cal_sa_reg_gap is a sub-function of
	bwa_aln_core	100	bwa_aln_core which constitutes a critical part of it
bwa-sw	bsw2_aln_core	99.6	The functions above it constitute a lot of code but take very less time.

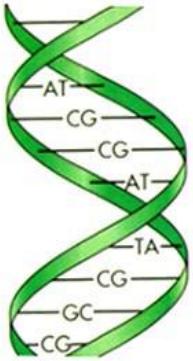
Chose BWA because:

- Likely to show an increase on GPU
  - BWT already implemented and showed 10x improvement in time (MuMmerGPU)
  - Evident from the above results (Amdahl's law)
- Popular (highly cited and widely used)



BWA algorithm in brief

- Constructs a suffix array interval (BWT) from ref gnome and aligns short query sequence to it (Li, Durbin, Sanger-UK, 2009)
- Initial construction of suffix array takes large time linear in length of query sequence
- Alignment time linear w.r.t the length of query sequence using backward search
  - Independent of the size of the ref. gnome



Implementation Strategy

- Transferring BWT - encoded reference sequence and sequence reads from disk to GPU
- CUDA thread assignments
  - Modifications
    - BWA utilizes a time-efficient BFS for backward search which would require a lot of memory in GPU
      - DFS search strategy for CUDA
  - Long sequence reads will be divided into short fragments and aligned with multiple DFS kernel runs
- After the kernel finishes, the alignment data will be copied back to the host for output

Results

- Similar alignment accuracy as BWA

	BWA	GPUbwa
Map percentage	89%	88%
Error	0.05%	0.05%

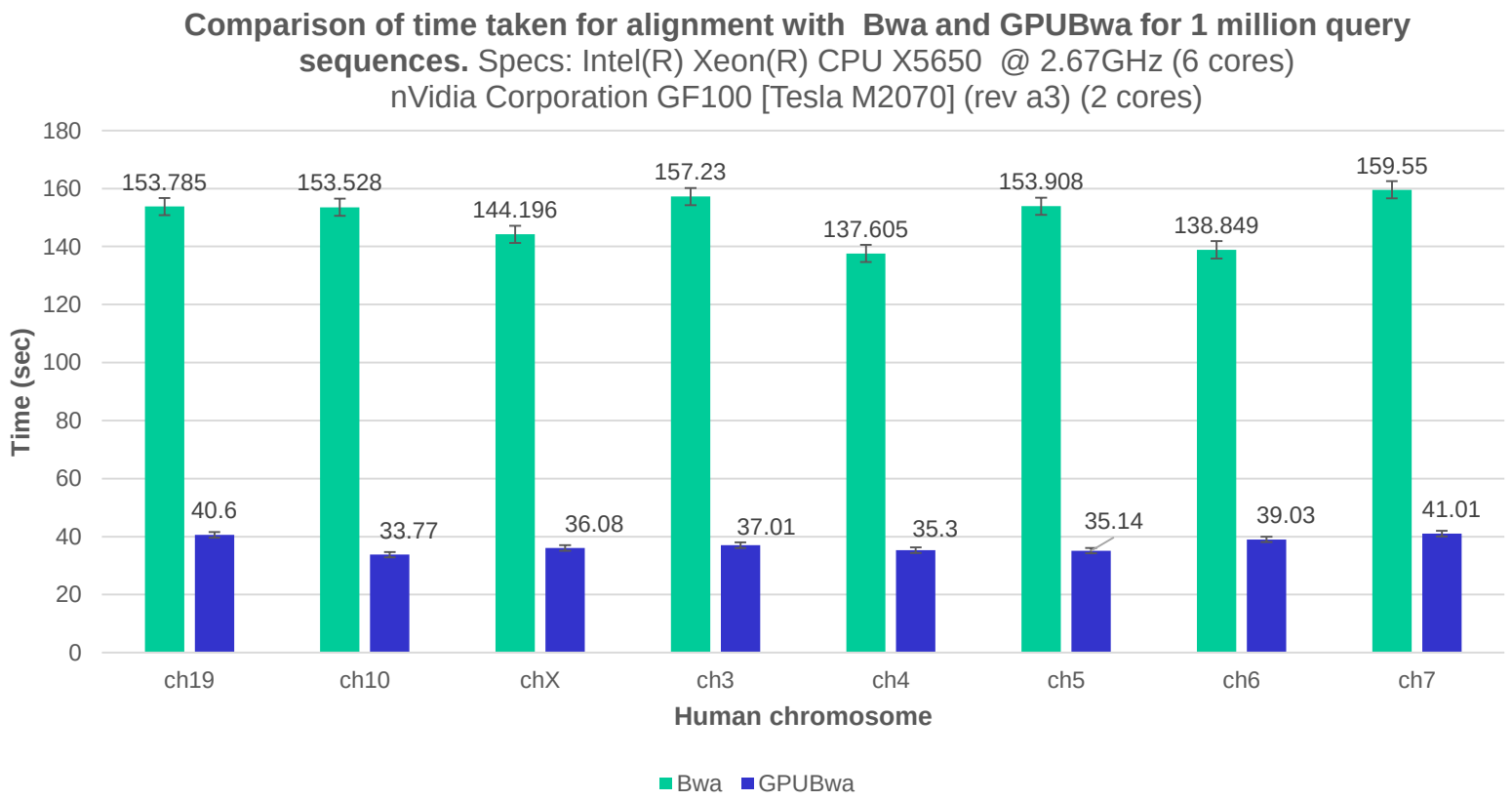
Accuracy comparison of BWA and GPUBwa for random generated short reads (70 bp) of *D. Melanogaster* via wgsim

	BWA	GPUbwa
Map percentage	89.95%	90.01%
Error	0.06%	0.04%

Accuracy comparison of BWA and GPUBwa for random generated short reads (70 bp) of *C. Elegans* via wgsim

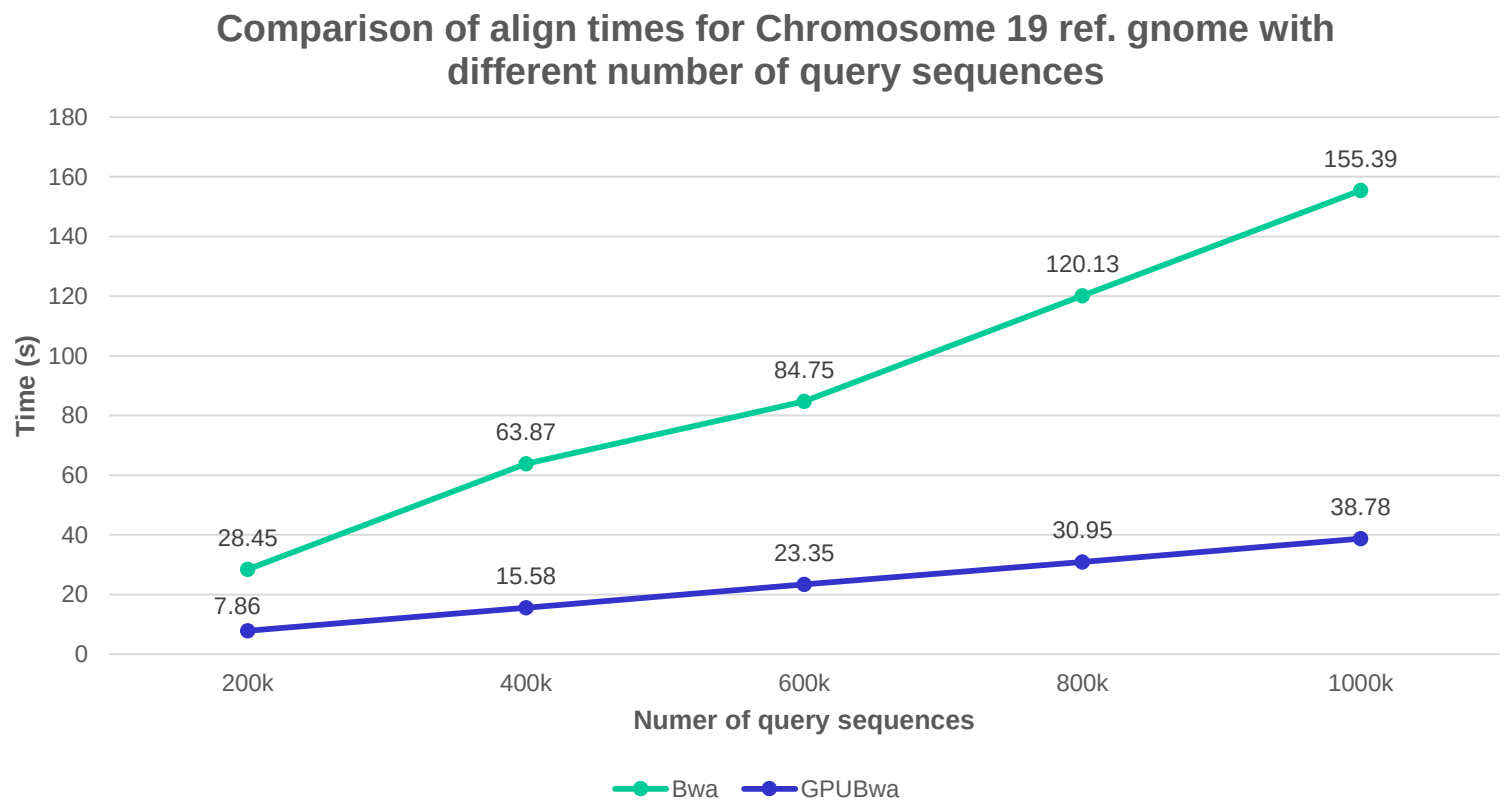
Results

- Performs faster than Bwa *(Avg. speedup = 4.04)*



Results

- Scales better with the number of query seqs.



Further scope

- Implementation of indexing to construct BWT on GPU
- Analysis of scalability with multi-GPU systems
- Future optimizations like indels (inexact alignment), splitting on-host