Computational Challenges of Next Generation Sequencing Pipelines Using Heterogeneous Systems

Ernst Joachim Houtgast*†[1],
Vlad-Mihai Sima†, Koen Bertels*,
Zaid Al-Ars*

* *Computer Engineering Lab, TU Delft, Mekelweg 4, 2628 CD Delft, The Netherlands*
† *Bluebee, Laan van Zuid Hoorn 57, 2289 DC Rijswijk, The Netherlands*

ABSTRACT

We are rapidly entering the era of genomics. The dramatic cost reduction of DNA sequencing due to the introduction of Next Generation Sequencing (NGS) techniques has resulted in an exponential growth of genetics data. The amount of data generated, and its associated processing into useful information, poses serious computational challenges. Here, we give a brief introduction of NGS, show a typical NGS processing pipeline, and show the associated challenges from a computational perspective. A case study is presented where one component of the NGS processing pipeline is accelerated: BWA-MEM, the de-facto industry-standard for the mapping stage. This is a first step in achieving a fully heterogeneously accelerated NGS pipeline.

KEYWORDS: BWA-MEM; FPGA; GPU; Next Generation Sequencing

1 Introduction

With the introduction of Next Generation Sequencing (NGS) techniques, the cost of sequencing complete genomes has fallen dramatically and has reached a point where it is becoming a feasible method to use in a wide variety of applications, such as medical diagnosis and forensics. Figure 1 illustrates how this cost reduction has even outpaced Moore's law, resulting in turn in an enormous growth of sequenced DNA data. The amount of data generated is projected to rival, if not overtake, other Big Data fields, such as astronomy and streaming video services [SLF+15]. However, this data is not very usable in its raw form of millions of *short reads*, short DNA fragments of usually only a few hundred base pairs. An example of a typical processing pipeline is given in Figure 2. First, these raw reads need to be reassembled into a complete genome using a *mapping* tool. Short read mapping finds the most likely place on a reference genome that a read originates from. Then, after obtaining the complete genome, the reads are *sorted* and mutations as compared to a reference genome are discovered during a *variant calling* phase. Many such mutation sites are annotated and knowledge about specific mutations can help in establishing a medical diagnosis.

[1]Corresponding author: ernst.houtgast@bluebee.com

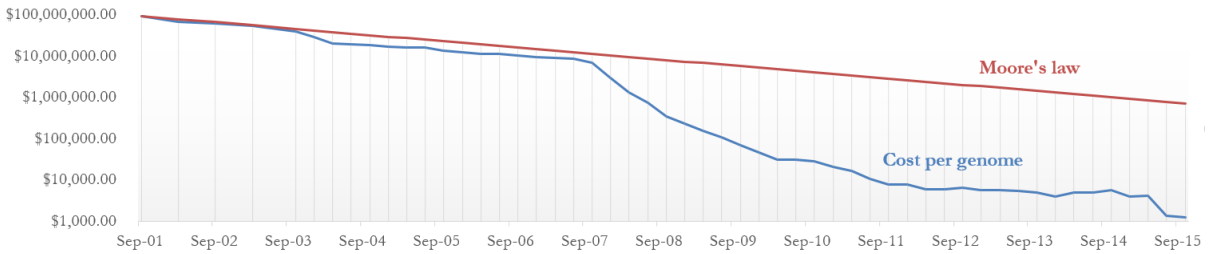

Figure 1: DNA sequencing costs are falling at a pace faster than Moore's law (from [Wet16]).

All the steps of the NGS pipeline operate on huge data sets, requiring large amounts of processing by complex algorithms. A sequencing run on an Illumina HiSeq X, a state-of-the-art NGS sequencer, produces data in the order of 1.2 TB every two days. For cancer data sets, this data requires multiple days of processing, even on high performance computing clusters, taking over 3,000 CPU-core hours of processing time. The extreme scale of data and processing requires enormous computing capabilities to make the analysis feasible within a realistic time frame. Heterogeneous computing holds great potential to provide large advantages in processing speed and power-efficiency. Power-efficiency is becoming at least as important as raw performance, as it is an important driver to overall data center cost.

2 NGS Computational Challenges

NGS processing pipelines come in many forms and shapes, depending on the specific use case. However, there are a number of traits that all such pipelines share. These characteristics make them pose unique challenges when targeted for acceleration efforts. The two most important ones are outlined below:

Extreme-Scale Data Size: The data size that these processing pipelines deal with are of an enormous magnitude. As an example, a single human genome contains three billion base pairs (A, C, G or T). The sequencer also provides a quality score for each base, which indicates the confidence with which the nucleotide was read. Finally, as only short fragments are sequenced and this data often contains errors, it is common practice to read the genome multiple times, a *coverage* of 30x or more being typical. This results in a compressed output size of around 100 GB. A single sequencer is able to process multiple samples in parallel.

An advantage is that this huge amount of data typically coincides with an abundance of parallelism. For example, in the case of short read mapping tool BWA-MEM, the mapping of short reads is independent, and hence these can be mapped in parallel.

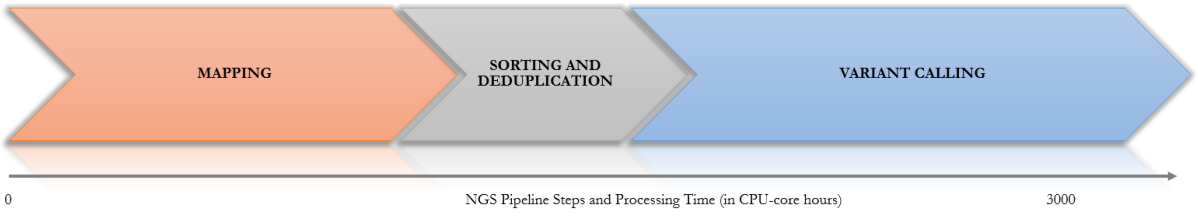

Figure 2: Processing time per NGS pipeline stage for a 30x coverage cancer NGS DNA data set with three tumor samples and one normal sample. Time is expressed in CPU-core hours.

Complex Multikernel Algorithms: Most of the tools in the pipeline are complex algorithms, consisting not just of a single phase that dominates execution time, but instead containing a number of time-consuming steps. For example, BWA-MEM processing is spread over three distinct stages, making acceleration of this algorithm more challenging, as not only does it require the adaptation of multiple separate algorithms, but also care has to be taken to not shift the bottleneck to another part of the application, limiting the benefit of any potential speedup as per Amdahl's law. This makes it quite difficult to obtain larger performance gains.

3 Heterogenous Acceleration of the NGS Pipeline

As the growth of genomic data is rapidly outpacing the increase in computational capabilities of a typical processor, it is clear that other means have to be used to meet the increased processing demand. As heterogeneous systems hold a great potential for large advantages in speed and efficiency, compared to traditional forms of computing, it has been our strategy to accelerate the individual components of the NGS pipeline into such systems. As a staple tool within NGS pipelines, the widely used BWA-MEM short read mapping tool was an interested candidate for acceleration. Our experiences are briefly discussed below.

3.1 Case Study: Accelerated BWA-MEM

The goal of the BWA-MEM algorithm is to find the best mapping of a short read onto a reference genome [Li13]. It makes use of the Seed-and-Extend paradigm (see Figure 3), a two-step method consisting of an Exact Matching phase and an Inexact Matching phase. First, exactly matching subsequences of the read and reference are identified. A single short read can have many such seed locations identified. Then, these seeds are extended using an algorithm similar to the widely-used Smith-Waterman algorithm, using a scoring system that awards matches and penalizes mismatches, insertions and gaps. The highest scoring match is chosen as final *alignment*. Figure 2, which contains a typical flow of an NGS pipeline, shows that BWA-MEM contributes about 36% to the overall processing time of the entire pipeline. Therefore, it is an important target for acceleration to reduce the overall time, cost and energy of processing NGS data sets.

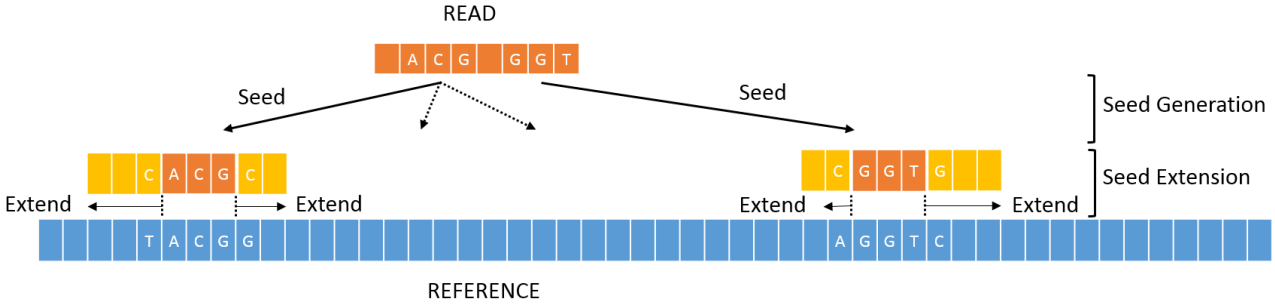

Figure 3: BWA-MEM uses the Seed-and-Extend paradigm. First, seed locations on the reference are found, which are subsequently extended using a Smith-Waterman-like algorithm.

3.2 Accelerated Implementation Results

We have implemented two heterogeneously accelerated version of BWA-MEM: an FPGA-accelerated version utilizing a single Xilinx Virtex-7 FPGA on the Alpha Data add-in card [HSM+16], and a GPU-based implementation using CUDA [HSBAA16]. Both versions offload the Seed Extension phase, which takes between 40%-50% of execution time, on the accelerator, accelerating the Smith-Waterman-like dynamic programming routine using a systolic array as detailed in [HSBAA15]. Thus, the implementations are able to achieve an up to two-fold speedup in overall application-level performance over the software-only implementation. This is the maximum theoretically achievable speedup when accelerating only this one BWA-MEM program kernel, as per Amdahl's law. This translates into tens of hours of time saved for real-world data sets.

4 Future Outlook

Acceleration of BWA-MEM is just one step of providing a fully accelerated NGS pipeline. There are many more steps that require large amounts of processing time. Examples include the various tools for variant calling, or for imputation. The bioinformatics software community needs to step up to this challenge, in order to be ready for the surge in demand for processing power as a result of ubiquitous generation and use of genomics data.

References

[HSBAA15] EJ Houtgast, V Sima, KLM Bertels, and Z Al-Ars. An FPGA-Based Systolic Array to Accelerate the BWA-MEM Genomic Mapping Algorithm. In *International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation*. IEEE, 2015.

[HSBAA16] EJ Houtgast, V Sima, KLM Bertels, and Z Al-Ars. GPU-Accelerated BWA-MEM Genomic Mapping Algorithm Using Adaptive Load Balancing. In *Architecture of Computing Systems–ARCS 2016*, pages 130–142. Springer, 2016.

[HSM+16] EJ Houtgast, V Sima, G Marchiori, KLM Bertels, and Z Al-Ars. Power-Efficient Accelerated Genomic Short Read Mapping on Heterogeneous Computing Platforms. In *Proc. 24th IEEE International Symposium on Field-Programmable Custom Computing Machines*, Washington DC, USA, May 2016.

[Li13] Heng Li. Aligning Sequence Reads, Clone Sequences and Assembly Contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997*, 2013.

[SLF+15] ZD Stephens, SY Lee, F Faghri, RH Campbell, C Zhai, MJ Efron, R Iyer, MC Schatz, S Sinha, and GE Robinson. Big Data: Astronomical or Genomical? *PLoS Biology*, 13(7), 2015.

[Wet16] KA Wetterstrand. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP). `http://www.genome.gov/sequencingcostsdata/`, 2016. Accessed: 2016-05-30.