Biomed Tech 2013; 58 (Suppl. 1) © 2013 by Walter de Gruyter · Berlin · Boston. DOI 10.1515/bmt-2013-4309

GENOME ASSEMBLY FRAMEWORK ON MASSIVELY PARALLEL, DISTRIBUTED MEMORY SUPERCOMPUTERS

Friedrich Menhorn[1, 2], Matthias Reumann[2, 3]

[1]Department of Computer Science, TU München, Germany
[2]Department of Computing and Information Systems, University of Melbourne, Carlton, VIC, Australia
[3]IBM Research - Australia Laboratory, Carlton, VIC, Australia

menhorn@in.tum.de

Abstract: *Genome Assembly describes the process of assembling a long Deoxyribonucleic acid sequence out of next generation sequencing (NGS) data. Computational resources can become a bottleneck or large scale routine use. We propose a genome assembly framework for massively parallel, distributed memory supercomputers. Our frameworks builds on the simple idea to equally distribute the number of reads to each processor. Each processor holds the whole reference genome. Each processor aligns the short reads independently and sends the reads back to root processor together with the corresponding position and the whole genome can be aligned. We run our alignment framework on up to 8,196 processors of the IBM Blue Gene/Q "Avoca" at the Victorian Life Science Computation Initiative. The results show that more than 6 Million reads of over 324 Million nucleotides can be assembled in under 20 minutes versus previously requiring hours. Thus, our framework allows fast assembly of NGS data.*

Keywords: *Computation Biology, Genome Assembly, Supercomputing*

Introduction

Next generation sequencing (NGS) has made feasible the use of sequencing in healthcare. However, routine use of NGS in a clinical setting on day to day basis has not yet been achieved due to lack of automatised workflows as well as the need for experienced bioinformaticians to carry out the analysis. Further, computational resources can become the bottleneck when scaling up NGS to large sample sizes.
In NGS, the genome is divided in smaller partitions of about 50 to 250 base pairs with one base pair representing a Deoxyribonucleic acid (DNA) molecule. These so called reads are aligned to a reference genome. The alignment process is computationally expensive. However, the large number of short reads (millions for bacteria) can be distributed to a parallel system. Thus, the objective of this study was to build a framework for massively parallel, distributed memory supercomputers to carry out to speed up genome assembly to a reference genome, which we anticipate to be the most common form of assembly in high throughput mode.

Methods

Our frameworks builds on the simple idea to equally distribute the number of reads to each processor. Each processor then requires to hold the whole reference genome for the problem to become pleasantly parallel. Then, each processor can use the chosen alignment algorithm independently to find the proper position for its specific reads. After finishing, it sends the reads back to root processor together with the corresponding position and the whole genome can be aligned.
As an example system for massively parallel, distributed memory supercomputers we use the IBM Blue Gene/Q, a supercomputer "Avoca" at the Victorian Life Science Computation Initiative with peak performance of 838.86 teraFLOPS [1]. Avoca holds 4,096 compute nodes and a total of 65,536 PowerPC based 1.6GHz cores each with four hardware threads. Each compute node holds 16GB of RAM (1GB per core). The assembly framework is build using the Message Passing Interface (MPI) for inter-process communication at present. While we currently have not implemented a hybrid model, shared memory parallelism can easily be introduced in our framework using e. g. OpenMP for a single process per compute node that could yield up to 262,144 threads for the full Avoca system.
We use a slightly modified master-worker communication where process 0 and 1 are the root processors, thereby controlling the other processors. At the beginning, the root processors read the raw NGS output file. In parallel, the other processors read the file which contains the sequence of the reference genome. By using MPI_Scatter, we are able to distribute an equal number of reads m to each processor. The number of reads x for processor i of a given number of processors n are calulated through

$$x_i = \lceil \frac{m}{n-2} \rceil, i = 2 \dots n \quad (1)$$

Note that this might not yield an integer value and care needs to be taken to adjust for rounding and distributing the right number of reads to each processor.
After having obtained its corresponding reads, each processor uses the chosen string pattern matching algorithm to find the alignment in the reference genome. A wide variety of pattern matching and alignment algorithms exist. We arbitrarily chose the Knoth-Morris-Pratt [2] and the Boyer-Moore [3] algorithms to compare performance of different alignment algorithms in our framework. However, any pattern matching algorithm could be implemented and inserted in the proposed framework. The algorithm returns an

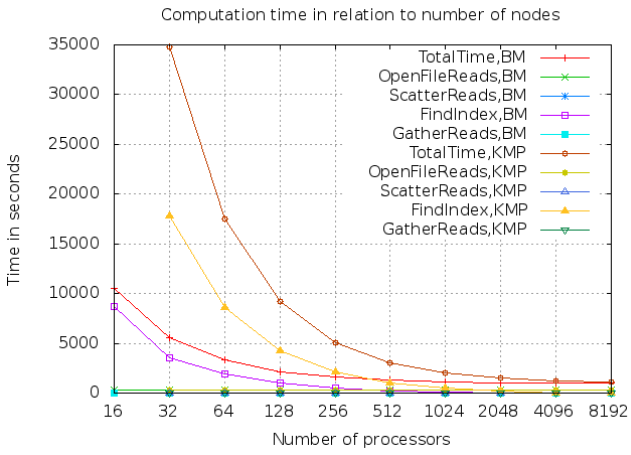

Figure 1: **Results of the framework**: Execution times for the each steps in the execution and the total runtime of the program in relation to the number of processors used. FindIndex: time needed to find the DNA reads in the reference genome with Boyer-Moore (BM) or Knuth-Morris-Pratt (KMP) alignement. ScatterReads/GatherReads: MPI overhead; OpenFileReads: I/O time for sequence files.

index which represents the distance from the first character in the reference genome using the MPI_Gather routine. Based on the returned indices, the root processors assemble the reads. In the last two steps, both processors merge their sequences and processor 0 compares it to the reference genome, thereby deriving the actual match.
We test our genome assembly framework with a NGS data set of Streptococcus pneumoniae. It consists of a reference genome of 2,221,315 nucleotides [4] and two files each containing 3,253,173 DNA reads with a length of respectivly 54 nucleotides [5]. This data set was arbitrarily chosen but offers fast prototyping and performance investigation in a real world setting rather than simulated data.

Results

Fig. 1 demonstrates that the run time of our program decreases almost exponentially. On 32 processors, the assembly finishes within 1.6 hours for the Boyer-Moore algorithm and 9.6 hours for the Knoth-Morris-Pratt algorithm. These times are reduced to less than 20 minutes for both algorithms on 8,196 processors. The duration of the index finding algorithm extends asymptotically towards zero for a increasing number of processors. On 8,192 processors it needs not even a second computation time in both cases.

Discussion and Conclusion

Our results show that our framework yields strong scalability that allows different alignment algorithms to be compared on large-scale parallel computers. Kalyanaraman et al. [6] published a genome assembly framework on the IBM BlueGene/L with 1,024 processors (750 MHz PowerPC). They were able to partition more than 1.6 million fragments of over 1.25 billion nucleotides total size of maize genome into genomic islands in under 2 h. Moretti et al. introduced a different approach on the parallelisation of the assembly by using a campus grid. They used the Smith-Waterman algorithm to align 8 million reads of length 750 base pairs. They required about 18 hours on a the campus grid [7] for the whole genome assembly.
To conclude, our framework allows highly parallel genome assembly. While direct comparison with other methods is not straight forward due to use of different compute systems and data, the proposed framework suggests to yield fast genome assembly that could overcome the computational bottleneck for usage in routine NGS assembly workflows.

Acknowledgement

This research was supported by a Victorian Life Sciences Computation Initiative (VLSCI) grant number 1049 on its Peak Computing Facility at the University of Melbourne, an initiative of the Victorian Government, Australia. We thank Prof. Justin Zobel, Department of Computing and Information Systems, University of Melbourne, to host this student project as well as Dr. Kelly Wyres, IBM Research - Australia Laboratory for selecting the genome sequence used in this study.

Bibliography

[1] Victorian Life Sciences Computation Initiative, “PCF Hardware: IBM Blue Gene/Q - Avoca,” April 2013.

[2] D. E. Knuth, J. H. Morris Jr., and V. R. Pratt, “Fast pattern matching in strings,” *Siam J. Comput.*, vol. 6, June 1977.

[3] R. S. Boyer and J. S. Moore, “A fast string searching algorithm,” *Communications of the ACM*, vol. 20, October 1977.

[4] National Center for Biotechnology Information, “Streptococcus pneumoniae atcc 700669, complete genome,” April 2013.

[5] European Nucleotide Archive, “Read: ERR026191 : Illumina Genome Analyzer II paired end sequencing,” April 2013.

[6] A. Kalyanaraman, S. Emrich, P. Schnable, and S. Aluru, “Assembling genomes on large-scale parallel computers,” *Journal of Parallel and Distributed Computing*, vol. 67, pp. 1240–1255, 2007.

[7] C. Moretti, M. Olson, S. Emrich, and D. Thain, “Highly scalable genome assembly on campus grids,” *Many-Task Computing on Grids and Supercomputers*, no. 12, 2009.