Journal of Computational Science xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Workflow optimization of performance and quality of service for bioinformatics application in high performance computing

Rashid Al-Ali, Nagarajan Kathiresan*, Mohammed El Anbari, Eric R. Schendel, Tariq Abu Zaid

Biomedical Informatics Research Division, Sidra Medical and Research Center, Doha 26999, Qatar

ARTICLE INFO

Article history:
Received 14 July 2015
Received in revised form 21 February 2016
Accepted 4 March 2016
Available online xxx

Keywords:
High performance computing
BWA-MEM algorithm
Quality of service
Next generation sequencing
Scalability
Application performance and parallel efficiency

ABSTRACT

Nowadays, High Performance Computing (HPC) systems commonly used in bioinformatics, such as genome sequencing, incorporate multi-processor architectures. Typically, most bioinformatics applications are multi-threaded and dominated by memory-intensive operations, which are not designed to take full advantage of these HPC capabilities. Therefore, the application end-user is responsible for optimizing the application performance and improving scalability with various performance engineering concepts. Additionally, most of the HPC systems are operated in a multi-user (or multi-job) environment; thus, Quality of Service (QoS) methods are essential for balancing between application performance, scalability and system utilization. We propose a QoS workflow that optimizes the balancing ratio between parallel efficiency and system utilization. Accordingly, our proposed optimization workflow will advise the end user of a selection criteria to apply toward resources and options for a given application and HPC system architecture. For example, the BWA-MEM algorithm is a popular and modern algorithm for aligning human genome sequences. We conducted various case studies on BWA-MEM using our optimization workflow, and as a result compared to a state-of-the-art baseline, the application performance is improved up to 67%, scalability extended up to 200%, parallel efficiency improved up to 39% and overall system utilization increased up to 38%.

1. High performance computing for bioinformatics

Due to various advancements in next-generation sequencing technologies (e.g. Illumina, SOLiD), larger volumes of genome data are being produced every year at a lower cost [1]. New functional variants are being discovered due to this ever-growing availability of genome data [2]. However, the analysis applications required for these discoveries typically are performance limited due to their compute and memory-intensive operations [3]. This paper addresses these challenges by optimizing genome alignment applications that are commonly hindered when using traditional High Performance Computing (HPC) systems. To overcome the traditional system limitations, HPC systems are becoming popular in bioinformatics for providing faster genome alignment by utilizing high-throughput and parallel-processing techniques [4,5], referred as "HPC for Bioinformatics" [6]. Thus, large-scale genome analysis can be parallelized to achieve empirically faster results by using HPC architectures, but those gains still have much room for improvement.

Nowadays, multi-core HPC systems used in genome sequencing still have no optimal choice of workflows based on application characteristics, in terms of accuracy, performance and optimal selection of computing resources. Generally, the application performance is dependent on various factors like complexity of the algorithm, application design, data distribution methods, communication cost, workflow dependency conditions, software stack (e.g. compilers, Message passing Interface (MPI)/Thread libraries) and hardware limitations [7]. To achieve the optimal performance of any application, it is necessary to understand the application characteristics and the performance bottlenecks.

Most bioinformatics applications are written in multi-threaded programming models that do not scale well in the modern multi-core HPC systems [3,8]. For example, when a modern GATK Haplotype caller application [9] is executed on a 32-cores HPC system with core steps of 2, 4, 8, 16 and 32, the performance improvement was expected as execution time keep reducing with the increased number of cores. However, when beyond 8 cores, the performance was not improving in relation. Therefore, the optimal computing resources should be selected based on the scalability

* Corresponding author.
E-mail address: nkathiresan@sidra.org (N. Kathiresan).

http://dx.doi.org/10.1016/j.jocs.2016.03.005

limitations to achieve the better performance. Alternatively, the utilization of the HPC system is very poor (about 25% of resource utilization) for the above GATK Haplotype caller workload. In this case, it is necessary to improve the system utilization by concurrently submitting more similar workloads within the HPC system. We conducted various case studies of different ways of parallelization and their performance impacts are summarized in [8]. As of now, tools are unavailable to balance between application performance, scalability and system utilization and hence we introduced the Quality of Service (QoS) factor, which uses an effective ratio between parallel efficiency and system utilization. We designed this QoS as an optimization workflow to provide the best performance and linear scalability with optimal set of resources.

2. Literature review

In the last few years, hash table based algorithms [10] (e.g. BLAST, SOAP, SeqMap, etc.) and prefix/suffix trees based algorithms (e.g. FM-Index, BWA-MEM, BWT-SW) are commonly used in genome mapping in bioinformatics research [1]. Burrows-Wheeler Aligner (BWA) is the most popular genome mapping software widely used in human genomic sequencing [11–13]. The BWA-backtrack, BWA-SW and BWA-MEM are three different algorithm versions of BWA. The BWA-SW and BWA-MEM algorithms are supported for long-reads (70 bp–1 Mbp) human genome sequences. Unlike the other algorithms, the BWA-MEM provides fast and accurate alignment for sequence reads and support for long-query and split-alignment in the human genome sequencing [14].

BWA-MEM, BWA-BT, Bowtie2, SMALT and MOSAIC are some of the widely used aligner tools. The aligned reads in BWA-MEM and SMALT are greater than 99%, where the execution time of SMALT is 3 times slower than BWA-MEM. The BWA-MEM and Bowtie2 execution times are relatively comparable to each other but, the Bowtie2 aligned reads are relatively good (98.27%) compared to BWA-MEM (99.10%) [15].

A new MICA aligner is optimized to take advantage of Intel's Many Integrated Core Architecture (MIC), which is 4.9 times faster in execution time compared to the BWA-MEM algorithm [16]. The Regional Hashing-based Alignment Tool (rHAT) produces accurate aligned reads, correctly aligned bases and excellent execution time [17]. In this paper, we compared the rHAT algorithm, even though it uses Hash-Indexing, in order to understand the computational limitations of the BWA-MEM. Overall, the new implementations of aligners, MICA and rHAT, are compared to BWA-MEM [16,17]. The aligned reads and aligned bases are comparatively similar to each other, but the BWA-MEM algorithm failed to produce better execution time due to CPU limitations. Hence, we are optimizing the BWA-MEM algorithm using "data-parallel and concurrent parallelization" [3].

The implementations [15–17] discussed prior are focused in reducing execution time and not the optimal selection of the computational resources. The utilization of the resources is equally important in a multi-user environment, and the performance is not always ideal when all the computing resources are utilized [8]. To address these challenges, we proposed an optimal workflow for bioinformatics applications that will give a better suggestion to balance between application performance, scalability and system utilization, referred to as "best QoS".

3. Workflow optimization for bioinformatics applications

We present a systematic sequence of approaches called "workflow optimization" for the bioinformatics applications on the HPC system. The workflow is developed based on experience and various performance engineering concepts. When the application source code is available, compiler optimization techniques are used to improve the application performance [18]. We used 4 sets of compiler optimization flags: default optimization (-O3/-O2 flag), vectorization, Single Instruction Multiple Data (SIMD) based tunings and architecture aware optimizations (e.g. AVX, AVX2, -qarch=pwr8). For every change in the compiler flags, a different versions of application binary is created and run with a subset of genome data to measure the application execution time, HPC system efficiency and resource utilization. The best set of results is referred as "un-optimized" and it is a "baseline" reference for our workflow optimization.

The application profile based analysis is used to optimize the licensed applications because of its pre-compiled binaries. By using the flat profile (e.g. using GNU profiler and Intel Vtune), the performance bottleneck of both types (source code and licensed based) of applications are analyzed [19]. Based on the flat profile results, the relationship between application instructions and low-level characterizations (e.g. cache miss, translation lookaside buffer (TLB) miss, etc.) are studied. Accordingly, the application is tuned (e.g. parallelize the instruction set, change the order of execution of the instruction to take benefit of the cached registered entries, etc.) and optimized to make use of low-level hardware features. Additionally, the genome data is equally partitioned into independent chunks and equal to the number of cores in the HPC system. The optimized binary, which is used as the "baseline" reference, is concurrently executed with independent chunks of genome data and then measurements are taken of the application execution time (last concurrent job completion time), HPC system efficiency and resource utilization. This set of performance number are referred as "concurrent parallelization" [3].

Due to the larger volume of genome data, the cache miss/translation lookaside buffer (TLB) misses are possible in genome alignment. Hence, the genome data is partitioned into independent multiple chunks (not necessarily equal to the number of cores) based on the level of cache misses. The optimized binary is executed in a multi-threading mode with every independent chunk of data. During binary execution, the number selection of multi-threads is determined for providing the best scalability factor. Accordingly, the system utilization is calculated. The overall execution time is sum of all the execution time of independent chunks of data processing time and this method is referred as "data-parallelization".

Most of the multi-threaded applications are affected by shared memory contentions. As a result, scalability limitations and poor system utilization are observed. To address these challenges, "data-parallel with concurrent parallelization" is introduced [3]. In this method, optimal number of cores is selected based on the scalability limitation using the data-parallelization concept. The genome data is independently partitioned into an optimal number of cores. The optimal binary is executed with independent partition of data concurrently across all and multi-threading is used, which is equal to the number of optimal number of cores. Additionally, hyper threading (HT) or simultaneous multi-threading (SMT) enabled options are studied to bring the best performance improvement when the application is not affected with shared memory contention.

Fig. 1 provides a workflow performance optimization overview of bioinformatics applications represented by step-by-step flowchart model. Additionally, we summarized our method of optimization in the automated scripting (Algorithm 1), which is described as follows:

Notations and assumptions:

1. The HPC system $C = \{C_1, C_2 \ldots C_n\}$ has 'n' cores.
2. The genome data $D = \{D_1, D_2 \ldots D_m\}$ can be partitioned into 'm' independent chunks.

Please cite this article in press as: R. Al-Ali, et al., Workflow optimization of performance and quality of service for bioinformatics application in high performance computing, J. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.jocs.2016.03.005