International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Performance Analysis of Clustal W Algorithm on Linux Cluster

Swati Jasrotia[1], Salam Din[2]

[1,2]School of Electrical Engineering and Information Technology, Punjab Agricultural University, Ludhiana, Punjab-141004, India

Abstract: ***Parallel version of bioinformatics applications can speed up the analysis of large-scale sequence data, especially about sequence alignments. Sharing of distributed idle computing resources and data with the use of parallel version software's on high performance clusters is an emerging step. Moreover Linux based clusters are replacing the mainframe systems / supercomputers because of its cost effectiveness. In this paper, we provide the performance results of parallel version of Clustal W on Linux cluster with the help of an open source rocks toolkit. An easy to deploy, contract / expand and manage, and scalable, distributed environment is proposed and built for bioinformatics applications. Our experimental results show that multiple sequence alignment of protein sequences with Clustal W software on rocks distributed environment gives high efficiency and speedup. It also shows cluster platforms are excellent alternatives to access to supercomputing, due to its price to performance ratio.***

Keywords: multiple sequence alignment; clustal w; linux cluster; high performance computing; parallel and distributed computing; speedup.

1. Introduction

The alignment of DNA or protein sequences is by far the most common task in Bioinformatics. Protein sequences can be related by homology with the help of multiple sequence alignment algorithms. It is a common occurrence that we only know the structure or function for one or two members of a group of protein sequences. Multiple alignments help in predicting the structure and functions of other members of a group of protein sequences [1]. The generated predictions about functional importance of specific sequences can then be tested experimentally.

The Clustal series of programs are widely used in molecular biology for the multiple alignments of both nucleic acid and protein sequences. The popularity of the programs depends on a number of factors: (i) accuracy of the results; (ii) robustness; (iii) portability (iv) user friendliness of the programs [2].

ClustalW method provides a command line interface for multiple sequence alignment. The alignment is achieved via three steps: pair-wise alignment; guide-tree generation; and progressive alignment [3]. The first of these stages uses the sequences to construct a distance matrix, which tabulates the similarity between every pair of sequences. The second stage involves the construction of a phylogenetic or guide tree, using the information in the distance matrix. This tree is used as input to the final stage, where the sequences are progressively aligned in an order specified by the guide tree to produce the final sequence alignment.

Parallel version of bioinformatics applications can speed up the analysis of large-scale sequence data. ClustalW-message passing interface (mpi) is a parallel implementation of ClustalW [4]. The pairwise alignments can be easily parallelized as many alignments are time independent on each other. However the progressive alignments are basically not parallelizable because of the time dependencies between each alignment.

High Performance Computing (HPC) clusters are used to run parallel programs for time-intensive computations. It comprises of a set of Massively Parallel Processors (MPPs) where each processor is referred as a node, having its own CPU, memory, operating system and input-output (I/O) subsystems. Cluster computing has emerged as a mainstream method for parallel computing in scientific and other applications domain. With the arrival of clustering technology and growing acceptance of open source software, supercomputers can now be created for a fraction of cost of traditional high-performance machines [5]. Clustering can be performed on various operating systems like Windows, Macintosh, Solaris etc., but Linux is the most popular operating system for developing cluster environment. Reason being, Linux runs on a wide variety of hardware and support n-number of applications and softwares, like parallel file systems and MPI implementations are freely available for Linux.

The idea of the Linux based PC cluster is to maximize the performance-to-cost ratio of computing by using low-cost commodity components and free-source Linux and GNU software to assemble a parallel and distributed computing system. So the objective of this paper is to implement the parallel version of ClustalW method on Linux cluster and analyse the performance results for speedup.

2. Materials and Methods

For our experiments, we used a 5 node PC cluster with 3.06 GHz Intel Core 2 Duo processors and 50 GB of disk storage per node. The NPACI rocks 6.0 (Mamba), an open source, Linux based software package is used to easily deploy, manage, maintain and scale the cluster.

2.1 Building HPC cluster using NPACI Rocks toolkit

NPACI Rocks is an open-source, Linux based software stack for building and maintaining high-performance clusters, and is available as a free download on the NPACI

Rocks Website [6]. One of the key ingredients of Rocks is the mechanism to produce customized distributions that defines the complete set of software for a particular node. Within a distribution, different sets of software can be installed on nodes by defining a machine specific Red Hat Kickstart file. Rocks use a MySQL database to define the global configurations (such as listing of all compute nodes for the host's database) and then generate database reports to create service-specific configuration files (e.g., DHCP configuration file, /etc/hosts file).

Rocks use a master node called front-end node for centralized deployment and management of a cluster. The frontend is installed with widely-used, standard software to support cluster application development and parallel application execution. The front-end node is installed with Rocks Base CD and Rocks Roll CDs. The insert-ethers utility is run on the front-end node to generate service specific configuration files (/etc/hosts and /etc/dhcpd.conf) into the database. For compute nodes installation, Preboot Execution Environment (PXE) is used, in which the nodes perform a network boot to obtain the operating system from the front-end node. When compute node is PXE booted, the DHCP request is sent to the front-end node and the MAC address of that node is stored into the database.

A Kickstart file is a text-based description of all the software packages and software configuration to be deployed on a node. Compute nodes use Kickstart's HTTP method to pull RPMs across the network. Rocks leverage this scripting feature to achieve 100% automatic configuration of compute nodes. By leveraging this installation technology, we can abstract out many of the hardware differences and allow the Kickstart process to auto detect the correct hardware modules to load (e.g., Ethernet interfaces and high-speed network interfaces). All the nodes are interconnected via a private 100 Mbps Ethernet network.

2.2 Sequence Alignment with ClustalW software on cluster

The Clustal W bioinformatics application software is distributed across the processors of the cluster using message passing interface (MPI) for communication. All the five compute nodes (total 10 processors) are running the job in parallel. The execution time is measured as the time between the start of the first process and the termination of the last process on the parallel and distributed processors.

As test data, we obtained protein sequence data sets from the National Center for Biotechnology Information (NCBI). We used three widely varying inputs, 100 sequences with sequence length of 500; 500 sequences with sequence length of 500; 1200 sequences with sequence length of 100. All the sequence data sets are mounted from the front-end node on to the compute nodes through secure shell login (ssh).

3. Results and Discussions

Our experiments measured the relative speedup (ratio of execution time on one processor and parallel p processors) as a function of the number of processors.

$$\text{Speedup} = T(1) / T(p) \qquad (1)$$

where T(1) is the execution time on one processor and T(p) is the execution time on p processors.

Figure 1-3 shows the relative speedup for the different phases of the algorithm (pair wise alignment and multiple alignments) as well as the overall speedup, on three different set of sequences. The speedup for pair-wise alignment is recorded the highest, thus giving an average speed up of 8.5 over ten processors. The average speed up for multiple alignment and overall alignment is 1.5 and 5.5 over ten processors. The poor speedup for multiple alignments is affecting the speedup for overall alignment because of poor load balancing in the third stage of the Clustal W algorithm [7] [8].

From the table given below, we concluded that for 1200 sequences with 100 characters, an overall speedup of 2.6 with ten processors can be achieved. Whereas for 100 sequences with 500 characters, and for 500 sequences with 500 characters, even greater overall speedup is possible, which in the two cases were 6.1 and 7.9 with ten processors.

Table 1: Relative speedup for three datasets

Sequence sets	pair_align speedup	multiple_align speedup	Overall speedup
100 seq, 500 char	9	1.6	6.1
500 seq, 500 char	8.8	1.7	7.9
1200 seq, 100 char	7.8	1.3	2.6
Average	8.5	1.5	5.5

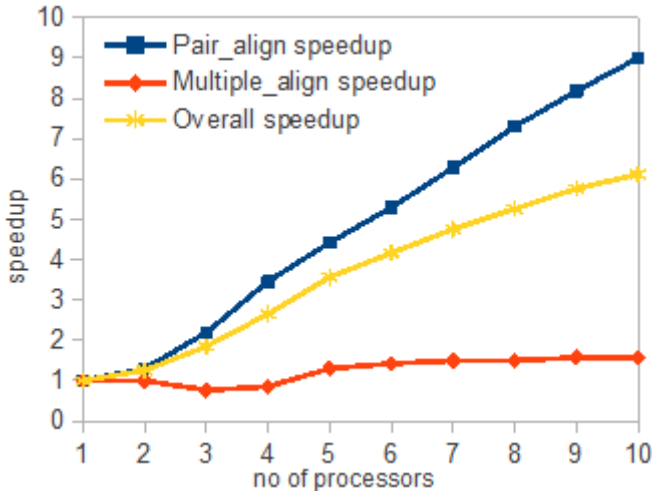

Figure 1: Speedup for 100 sequences with 500 characters each

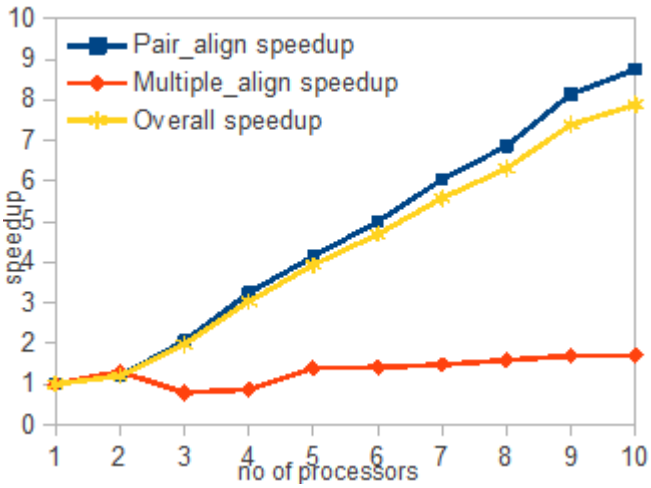

Figure 2: Speedup for 500 sequences with 500 characters each

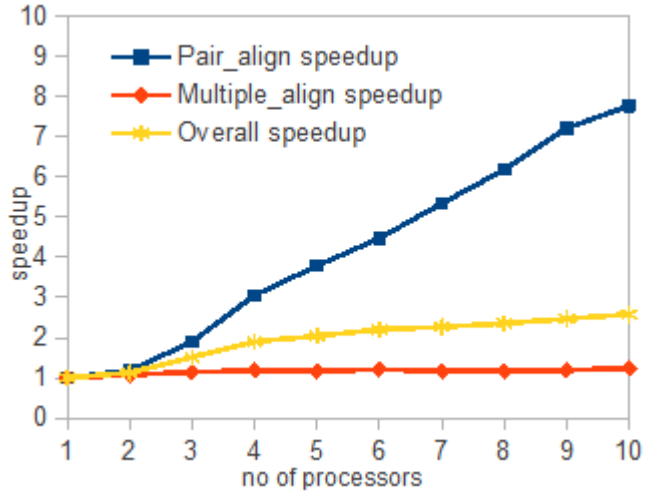

Figure 3: Speedup for 1200 sequences with 100 characters each

4. Future Prospects

Significant speedup can be achieved on lengthy multiple alignments using the inexpensive PC clusters. Further use of high speed Ethernet networks or infiniband can add up even more speedup to the cluster. Linux clusters provide us with a platform, which is an excellent alternative to access to supercomputing.

References

[1] Xiong J., Essential Bioinformatics, Cambridge University Press, UK, 2006.
[2] Chenna R. (2003): Multiple sequence alignment with the clustal series of programs. *Oxford Univ Press,*vol-31, pp. 3497-3500
[3] Thompson J. D., Higgins H. G. and Gibson T. J. (1994) Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research,* vol-22, pp. 4673-80.
[4] Li K. B.(2003). ClustalW-MPI: ClustalW analysis using distributed and parallel computing. *Oxford Univ Press,* vol-19, pp. 1585-86.
[5] Buyya R., Jin H. and Cortes T. (2002) Cluster Computing, *Elsevier Science B.V.*, pp. 5-8.
[6] Gupta R., Fang Y. C. and Hussain M. (2005) Streamlining Beowulf Cluster Deployment with NPACI Rocks. *Dell Power Solutions* : pp. 111-114
[7] Datta A. and Ebedes J. (2004) Multiple sequence alignment in parallel on a workstation cluster, *Oxford Univ Press,* vol- 20, pp. 1193-95.
[8] Yang C. T., Kuo Y. L. (2003) Apply Parallel Bioinformatics Applications on Linux PC Clusters. Tunghai Science, pp. 125-141

Author Profile

Swati Jasrotia received the B.Tech. degree in Information Technology from Punjab Technical University in 2011. Now she is pursuing her M.Tech degree in Computer Science & Engineering from Punjab Agricultural University. Her area of dissertation is Parallel and Distributed computing.

Salam Din is working as an Associate Professor in School of Electrical Engineering and Information Technology, Punjab Agricultural University and is guiding M.Tech students in Dissertation/Thesis.