IOSR Journal of Computer Science (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727
PP 71-75
www.iosrjournals.org

Development of Parallel Processing Application For Cluster Computing Using Artificial Neural Network Approach

Minal Dhoke[1], Prof. Rajesh Dharmik[2]
[1]*IT Department, Yeshwantrao Chavan College of Engineering, India*
[2]*IT Department, Yeshwantrao Chavan College of Engineering, India*

ABSTRACT : *Computing Clusters (CC) consisting of several connected machines, could provide a high-performance, multiuser, timesharing environment for executing parallel and sequential jobs. In order to achieve good performance in such an environment, it is necessary to assign processes to machines in a manner that ensures efficient allocation of resources among the jobs. Here we present the algorithm that suited the best for cluster computation by the use of artificial neural network approach and find out the efficient output by using the gridsim. The Grid Modeling and Simulation (GridSim) toolkit provides a comprehensive facility for simulation of application scheduling in different Grid computing environments. The toolkit supports modeling and simulation of heterogeneous Grid resources (both time and space shared), users and application models. It provides primitives for creation of application tasks, mapping of tasks to resources and their management.*

Keywords *- Needleman-Wunsch, Parallel Algorithm, Parallelization, Gridsim toolkit.*

I. INTRODUCTION

Cluster computing can be described as a fusion of the fields of parallel, high-performance, distributed, and high-availability computing. Cluster computing has become a hot topic of research among academic and industry community including system designers, network developers, language designers, standardizing forums, algorithm developers, graduate students and faculties. Clusters also provide an excellent platform for solving a range of parallel and distributed applications in both scientific and commercial areas.

Artificial neural networks comprise a class of artificial intelligence that attempts to replicate, in simplified form, the way a biological brain works. The high degree of complexity present in artificialneural networks makes them computationally expensiveto simulate. In the past, there has been a large body ofwork considering techniques for exploiting the natural parallelisminherent in neural networks to improve their performance.

Most of this work has been centered on special purpose hardware implementations that provide a high degree of parallelism (such as massively parallel processors), or on mapping neural networks onto conventional shared memory multiprocessor parallel computers (SMP). In this work, we consider the implications of using a cluster computer.

The abundance of low cost computing resources in clusters and the increasing network bandwidth make it attractive to run parallel applications with cluster computing. We attempt to use efficiently the computational resources that are idle to obtain a system of high computing power with the existing machines. Due to the high communication cost in cluster computing, we are interested in designing parallel applications with low demand on communication. To this end, we propose a method to design parallel algorithms with the nice property of each processing having to communicate with only a few others. with the help of artificial neural network.

The rest of the paper depicts the following sections: Section 2 focuses on the Needleman Wunsch algorithm; Section 3 discusses the proposed work on gridsim; Section 4 we are coming up with conclusion.

II. NEEDLEMAN AND WUNSCH'S ALGORITHM FOR GLOBAL SEQUENCE ALIGNMENT

The algorithm consists of two parts: the calculation of the total score indicating the similarity between the two given sequences, and the identification of the alignment(s) that lead to score. Thus to compare two sequences, we need to find the best alignment between them, which is to place one sequence above the other making clear the correspondence between similar characters from the sequences [3]. In an alignment, gaps are inserted in arbitrary locations along the sequences so that they end up with the same size. Given an alignment
between two sequences s and t , a score can be associated for it as follows. For each column, we associate +1 if the two characters are identical, 0 if the characters are different and -1 if one of them is a space. The score is the sum of the values computed for each column. The maximal score is the similarity between the two sequences, denoted by sim(s,t) . Figure 1 shows the alignment of sequences s and t , with the score for each column[2].

A T - A A G T

A T G C A G T

+1 +1 -1 0 +1 +1 +1

Σ=4

Figure 1. Alignment between s and t.

For long sequences, it is unusual to obtain a local alignment. Instead, the global alignment algorithm is executed to calculate the fmal score between sequences; local alignment algorithm is executed to detect regions inside both sequences that are similar. Global alignment algorithms are executed for similarity and final score. Figure 2 illustrates this.

Needleman and Wunsch proposed an algorithm (NW) based on dynamic programming to solve the global alignment problem. The time and space complexity of this algorithm is O(mn) , where m and n are the lengths of the two sequences, and, if both sequences have approximately the same length, n, we get O(n2) [4]. The NW algorithm is divided into two parts: the calculation of the similarity array and the retrieval of the global alignments.

Sequence: Y

A		**A**	**T**	**G**	**C**	**A**	**G**	**T**
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
T	-2	0	2	1	0	-1	-2	-3
A	-3	-1	1	2	1	1	0	-1
A	-4	-2	0	1	2	2	1	0
G	-5	-3	-1	1	1	2	3	2
T	-6	-4	-2	0	1	1	2	4

Sequence **X**

Final score

Figure 2. Global alignment of two sequences with final score = 4

Using this algorithm in parallel, the reduction of time and space complexity is achieved. The results obtained to globally align real DNA sequences on two workstation of 12 cores each, present good speedup. We intend to implement this algorithm to compare very long DNA sequences (larger than lOMBP) in a computational grid.

III. IMPLEMENTATION OF NEEDLEMAN WUNSCH ALGORITHM ON GRIDSIM

Since the GridSim toolkit is an advanced and powerful simulation toolkit, its users will experience a high learning curve in order to utilize the toolkit functionalities for effective simulations. In addition, the users need to write Java code that use the toolkit packages to create the desired experimental scenarios. The GridSim toolkit provides a comprehensive facility for simulation different classes of heterogeneous resources, users, applications, and resource brokers. In Grid -like environment, resource brokers perform resource selection and aggregation depending on users requirements and hence they are *user-centric* in nature.

The gridsim has the facility so that the Users can be created with different requirements (application and quality of service requirements). These requirements include the baud rate of the network (connection speed), maximum time to run the simulation, time delay between each simulation, and scheduling strategy such as cost and/or time optimization for running the application jobs. The application jobs are modelled as Gridlets. The parameters of Gridlets that can be defined includes number of Gridlets, job length of Gridlets (in Million Instructions (MI)), and length of input and output data (in bytes). Each Grid user has its own economic requirements (deadline and budget) that constrains the running of application jobs. The Grid user can have the exact cost amount that it is willing to spend for the value-based option.

The multitasking and multiprocessing mode:

The Processing Elements (PEs) represent a fundamental unit in GridSim toolkit. Indeed it is through the PEs that GridSim can create machines, stitching together one or more PEs. Each PE has different speed and it is measured in MIPS or SPEC-like ratings. Moreover, all of the machines can be put together to create a grid resource. Thus we can obtain a grid resource that can be:

- a single processor;
- a multiprocessor, or better a shared memory multiprocessor (SMP);
- a distributed memory cluster of computers.

These Grid resources can simulate time- or space-shared scheduling depending on the allocation policy. To manage all the distributed memory multiprocessing systems (such as clusters), Grid- Sim uses a queuing system called a space-shared scheduler. This scheduler executes a gridlet by running it on a dedicated PE when allocated. The most common policies for allocating the resources in the space-shared systems are: first-come-first-served (FCFS), back filling, shortest-job-first-served (SJFS).

Multitasking and multiprocessing systems allow concurrently running tasks to share system resources such as processors, memory, storage, I/O, and network by scheduling their use for very short time intervals. In the real systems a detailed simulation of scheduling tasks would be very complex and very costly in terms of time. The events that simulate the execution of jobs, can be sent, received, or scheduled events by GridSim. The GridSim schedules self-events for simulating resource allocation depending on the scheduling policy and the number of jobs in the queue or in execution.

The algorithm will be implemented parallelly using gridsim simulator like in cluster computing environment. We can select the architecture, operating system, no. of machines on which the algorithm will work or distributed.

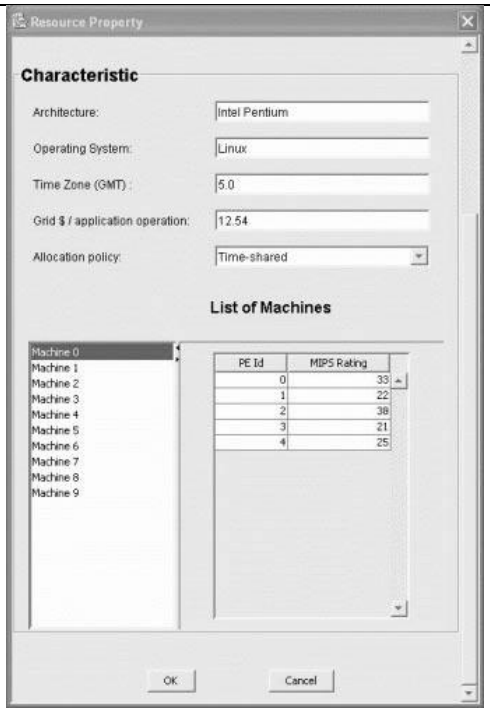

Fig. 3. Resource dialog to view Grid resource properties

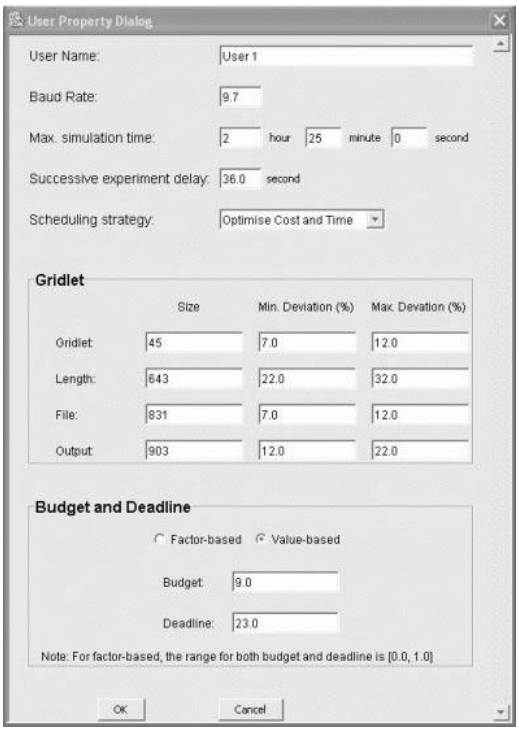

Fig. 4. User dialog to view Grid user properties

The input given through the gridsim will be distributed over the machines that are used by it with all its specifications as shown in the figures above, and the output will be created on the gridsim only.

The design space for computers that can be used for ANN is very large, and even if it seems futile to talk about an optimal design there are a number of interesting trade-offs to be made. One basic trade-of f is between *flexibility* and *speed*. That is, to increase speed we usually have to sacrifice

some flexibility. In one end of the spectrum we have ordinary general purpose computers and in the other end we might have analog or optical computers. As in ANN the neurons are used, likewise the processing elements are used in parallel computing clusters. ANN approach is used to reduce more time to calculate the results in our system.

IV. CONCLUSION

In this paper, we are proposing an algorithm that will be used in parallel computing to reduce time and space complexity. We are using Gridsim toolkit that is the toolkit for creating clusters and It incorporates features such as easy-to-use wizards that enables users to create simulation models, and automatic source code generation facility that outputs ready-to-run simulation scenario code. ANN approach is used for reducing time for the calculations.

REFERENCES

[1] http://www.gridbus.org/gridsim/
[2] Implementation of Parallel Algorithms on Cluster of Workstations. Shrimankar D D, Sathe S R. Department o/Computer Science and Engineering Visvesvaraya National Institute of Technology Nagpur, Maharashtra, lndia, 2012.
[3] Decypher Co. Decypher smith waterman solution, 2003
[4] S. F. Altschul et al. Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Research, 25(17):3389-3402, 1997.
[5] Pairwise Sequence Comparison [online], Lab of Bioinformatics, Institute of Computing Technology (ICT), Chinese Academia of Sciences (CAS). Available: http://www.bioinfo.org.cn/lectures/index-13.html
[6] Rong X, Jan 2003, Pairwise Alignment - CS262 - Lecture 1 Notes [online], Stanford University. Available: http://ai.stanford.edu/~serafim/cs262/Spring2003/Notes/1.pdf
[7] DNA Sequence Comparison [online], The BioWall by Swiss Federal Institute of Technology in Lausanne (EPFL). Available: http://lslwww.epfl.ch/biowall/VersionE/ApplicationsE/SequenceE.html
[8] Krishna N. and Akshay L. and Dr. Rajkumar B., 2002, Alchemi v0.6.1 Documentation [online], University of Melbourne. Available: http://alchemi.net/doc/0_6_1/index.html
[9] Local DNA Sequence Alignment in a Cluster ofWorkstations: Algorithms and Tools, Alba Cristina M. A. Melo, Maria Emilia M. T. Walter, Renata Cristina F. Melo, Marcelo N. P. Santana, Rodolfo B. Batista Department of Computer Science University of Brasilia, Brazil
[10] GridSim: A Toolkit for the Modeling and Simulation of Global Grids, Manzur Murshed, Gippsland School of Computing and IT Monash University, Gippsland Campus Churchill, Vic 3842, Australia Rajkumar Buyya School of Computer Science and Software Eng. Monash University, Caulfield Campus Melbourne, Vic 3145, Australia
[11] International Conference on Parallel and Distributed Systems. Extending GridSim with an Architecture for Failure Detection, 2007.