International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Industrial Automation and Computing (ICIAC- 12th & 13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

Development of parallel processing application for cluster computing using Artificial Neural Network approach

Minal Dhoke, Prof. Rajesh Dharmik
Yeshwantrao Chavan College of Engineering, Nagpur,India
Email: dhokeminal13@gmail.com, raj_dharmik@yahoo.com

ABSTRACT
Grid Computing is an emerging technology to provide high performance computing in a virtual organization composed of a large number of computers connected through web based technologies. We have implemented a parallel version of the Needleman-Wunsch algorithm for handling the DNA matching and alignment problem. We have presented the Needleman Wunsch algorithm for global alignments of sequences which obtains the best global alignments at the expense of very high computing power and huge memory requirements is integrated with the gridsim. Gridsim which supports the creation of *repeatable* and *controllable* Grid environments for quicker performance evaluation of scheduling strategies under different scenarios such as varying number of resources and users with different requirements.

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

I. INTRODUCTION

Cluster computing is a fusion of the fields of parallel, high-performance, distributed, and high-availability computing, which also provides an excellent platform for solving a range of parallel and distributed applications which is integrated with the Artificial Neural Networks approach to provide a high degree of parallelism. Parallel computing is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel").

The primary objective of this project is to investigate effective resource allocation techniques based on computational economy through simulation. We like to simulate millions of resources and thousands of users with varied requirements and study scalability of systems, algorithms, efficiency of resource allocation policies and satisfaction of users. We are also interested to explore how significantly the local economy and the global positioning (e.g., the time zone) of a particular resource play role in securing jobs under various pricing and demand/supply situations.

As such a large-scale simulation consumes large amount of computing power, we would like to use parallel and cluster computing systems. In our simulation we would like to model applications in the areas of biotechnology, astrophysics, network design, and high-energy physics in order to study usefulness of our resource allocation techniques. The results of our work will have significant impact on the way resource allocation is performed for solving problems on cluster and grid computing systems.

Grid systems are classified into two categories: compute and data grids. In compute grids the main resource that is being managed by the resource management system is compute cycles while in data grids the focus is to manage data distributed over geographic allocations. The architecture and the services provided by the resource management system are affected by the type of grid system.

II. Parallel 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.

		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

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.

IV. Building Simulations With Gridsim

The Java-based GridSim discrete event simulation toolkit provides Java classes that represent entities essential for application, resource modeling, scheduling of jobs to resources, and their execution along with management. In GridSim toolkit, we can create CPUs (also called *Processing Elements* (PEs)) with different MIPS (Million Instructions Per Second) or SPEC-like ratings. Then, one or more PEs can be put together to create a machine (a single CPU/SMP node). Such one or more machines can be put together to create a Grid resource. The resulting Grid resource can be a single processor, shared memory multiprocessors (SMP), or a distributed memory cluster of computers.

A Gridlet is a tiny GridApp that contains all information related to jobs and job execution management details such as jobs processing requirements, expressed in MIPS, disk I/O operations, the size of input files, etc. that help in computing execution time of remote resource and the size of output files.

In the output, we have got the gridlet status, no. of gridlets, their resource id,and the execution time.

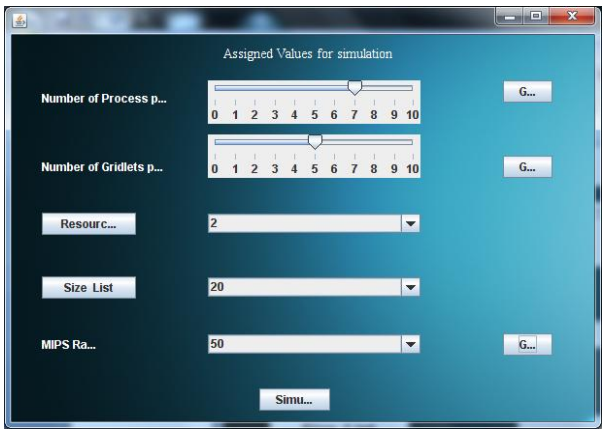

Figure 3. Assigning values to gridsim.

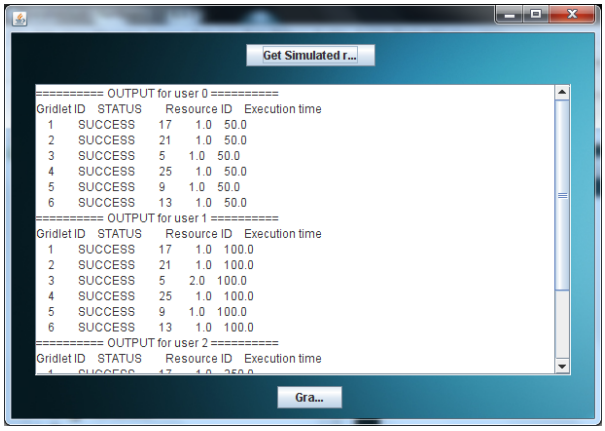

Figure 4. output of gridsim

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.

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.

V. 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.