Load Balancing using High Performance Computing Cluster Programming

Dr. Sumit Srivastava[1], Mr. Pankaj Dadheech[2] and Mr. Mahender Kumar Beniwal[3]

[1] Information Technology Department, Rajasthan Technical University, Poornima College of Engineering, Jaipur, Rajasthan 302025/Sitaputa, India

[2] Computer Science & Engineering Department, Rajasthan Technical University, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur, Rajasthan 302025/Jagatpura, India

[3] Computer Science & Engineering Department, Rajasthan Technical University, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur, Rajasthan 302025/Jagatpura, India

Abstract

High-performance computing has created a new approach to science. Modeling is now a viable and respected alternative to the more traditional experiential and theoretical approaches. High performance is a key issue in data mining or in image rendering. Traditional high performance clusters have proved their worth in a variety of uses from predicting the weather to industrial design, from molecular dynamics to astronomical modeling. A multicomputer configuration, or cluster, is a group of computers that work together. A cluster has three basic elements—a collection of individual computers, a network connecting those computers, and software that enables a computer to share work among the other computers via the network. Clusters are also playing a greater role in business. Advances in clustering technology have led to high-availability and load-balancing clusters. Clustering is now used for mission-critical applications such as web and FTP servers. For permanent clusters there are, for lack of a better name, cluster kits, software packages that automate the installation process. A cluster kit provides all the software you are likely to need in a single distribution. Cluster kits tend to be very complete. For example, the OSCAR distribution. Open Source Cluster Application Resources is a software package that is designed to simplify cluster installation. A collection of open source cluster software, OSCAR includes everything that you are likely to need for a dedicated, high-performance cluster. OSCAR takes you completely through the installation of your cluster. In this Paper with the help of Open Source Cluster Application Resource (OSCAR) cluster kit, attempt to setup a high performance computational cluster with special concern to applications like Integration and Sorting. The ease use of cluster is possible globally and transparently managing cluster resources. Cluster computing approach nowadays is an ordinary configuration found in many organizations to target requirements of high performance computing.

Keywords: *Clustering, Performance analysis, Web clustering, Workload characterization, High Performance.*

1. Introduction

Computing speed isn't just a convenience. Faster computers allow us to solve larger problems, and to find solutions more quickly, with greater accuracy, and at a lower cost. All this adds up to a competitive advantage. In the sciences, this may mean the difference between being the first to publish and not publishing. In industry, it may determine who's first to the patent office.

Traditional high-performance clusters have proved their worth in a variety of uses—from predicting the weather to industrial design, from molecular dynamics to astronomical modeling. High-performance computing (HPC) has created a new approach to science—modeling is now a viable and respected alternative to the more traditional experiential and theoretical approaches. High performance is a key issue in data mining or in image rendering. Advances in clustering technology have led to high-availability and load-balancing clusters. Develop the algorithms for the Sorting that are faster and more accurate than they were ever before. The problem with the huge data sorting is that it takes a lot of time and as a result a large amount of money is required. This huge money is just wasted in the sorting of data and not in any useful work so, it is highly required that this time be as less as possible. So we came up with the some new methods to lower that Complexity. The approach we followed was High Performance Computing.

2. Background

The field of High Performance Computing is the most popular for getting faster results for any application. The applications designed using High Performance Computing concepts are Numerical Integration, Quick Sorting. Application for Numerical Integration is designed for calculating the integration of function cos(x) under the limits 0 to pi/2. Such integration application can be easily executed over several nodes simultaneously so as to reduce the work load on single computer for calculating the integration of function completely.

2.1 The main Applications we took up as the Research were:

- Numerical Integration of function cos(x) for limits 0, pi/2
- Quick Sorting

The strength and weakness of each algorithm and methodology is to be analyzed and then new methodologies and algorithms are to suggested for doing the above stated work. This methodology could be evaluated in certain measures like:

- Sorting any amount of Data
- Complexity be as less as possible
- Fast in response and output
- Reliable

The basic unit of a cluster is a single computer, also called a "node". Clusters can grow in size - they "scale" - by adding more machines. The power of the cluster as a whole will be based on the speed of individual computers and their connection speeds are. In addition, the operating system of the cluster must make the best use of the available hardware in response to changing conditions. This becomes more of a challenge if the cluster is composed of different hardware types (a "heterogeneous" cluster), if the configuration of the cluster changes unpredictably (machines joining and leaving the cluster), and the loads cannot be predicted ahead of time.

High performance is a key issue in data mining or in image rendering. Advances in clustering technology have led to high-availability and load-balancing clusters. OSCAR is a package of RPM's, Perl-scripts, libraries, tools, and whatever else is needed to build and use a modest-sized Linux cluster. OSCAR is an Open Source project. Every component within OSCAR is available under one of the well known Open-Source licenses (e.g. GPL). The goal of OSCAR is making clusters easy to build, easy to maintain and easy to use. In other words, OSCAR contains the resources need to apply cluster computing to High Performance Computing problems.

3. Implementation and Applications

The numerical Integration algorithm can divide a big problem of integration into several smaller problems of integration. For decomposing the problem, the area under the curve of the function has to be divided into rectangles. These smaller segments of problem can be distributed over the cluster (i.e. the nodes) so that they can be executed simultaneously and hence result can be produced in a shorter time than computing the whole problem on a single machine. Using the concept of Message Passing Interface, we can also select the number of processes in which we want our problem to be decomposed and executed. On increasing the number of processes, the result can be generated by the processors more accurately.

Another application which can be designed using MPI libraries is Quick sort algorithm in which sorting of the numbers can be done using the Divide and Conquer Rule. In Quick sort, the array of numbers is divided into partitions according to the position of the pivot element. The elements lesser than pivot come before it and elements greater than pivot comes after it. Thus the partitions can be distributed over cluster and this quick sort mechanism can be called recursively so as to sort the partitions. Using HPC, these problems can be executed in a shorter time and more accurately.

The major difficulty in parallel programming is subdividing problems so that different parts can be executed simultaneously on different machines. MPI is use to determine how a problem can be broken into pieces so that it can run on different machines.

With most parallelizable problems, programs running on multiple computers do the bulk of the work and then communicate their individual results to a single computer that collects these intermediate results, combines them, and report the final results. As the program executes on each machine, it will first determine which computer it is running on and, based on that information, tackle the appropriate part of the original problem. When the computation is complete, one machine will act as a receiver and all the other machines will send their results to it. For this approach to work, each executing program or process must be able to differentiate itself from other processes.

To calculate the area under a curve, we use numerical integration method. This problem can be solve by parallel calculations because it can be easily decomposed into parts that can be shared among the computers in a cluster. The numerical integration method used in this problem is data decomposition. Each process has a different set of bounds, so the area that each calculated is different but the procedure is the same.

To multiply two m*n matrices, we use decomposing the problem. It is essential to realize that there are a number of trade offs that must be balanced when dividing a problem. The idea is to break the algorithm into pieces of code or tasks based on the data required by that task. The best solution usually lies somewhere between maximizing concurrency and minimizing communication.

3. Results and Discussions

The process of parallel algorithm design can be broken into several steps. First, we must identify the portions of the code that can, at least potentially, be executed safely in parallel. Next, we must devise a plan for mapping those parallel portions into individual processes or onto individual processors. After that, we need to address the distribution of data as well as the collection and consolidation of results. OSCAR is designed with high-performance computing in mind. Basically, it is designed to be used with an asymmetric cluster. Generally decomposition strategies fall into two different categories- data decomposition or data partitioning, and control decomposition or task partitioning. With data decomposition, the data is broken into individual chunks and distributed among processes that are essentially similar. With control decomposition, the problem is divided in such a way that each process is doing a different calculation.

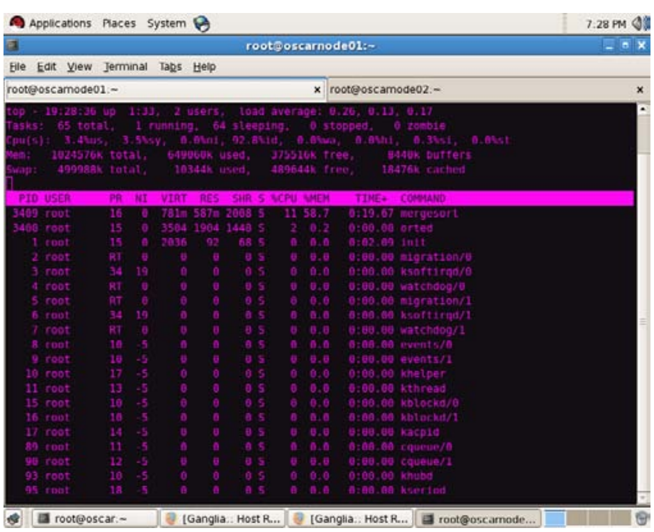

Fig. 1 Result on Node 1.

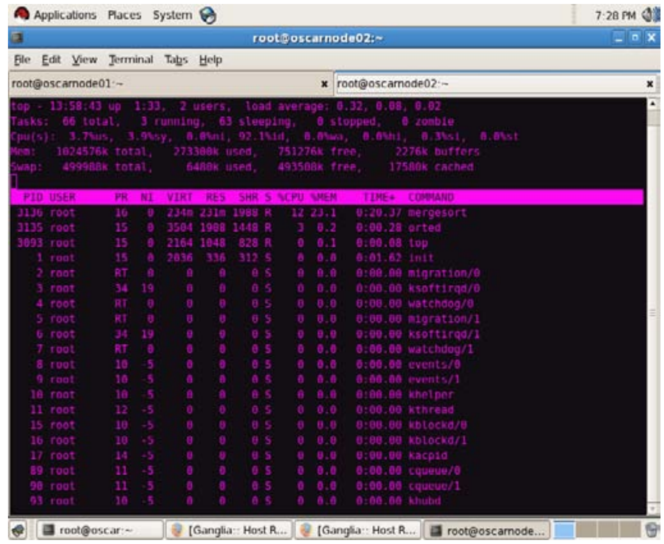

Fig. 1 Result on Node 2.

4. Conclusions

The basic objective of this paper is to setup a high performance computational cluster with special concern to applications like Integration and Sorting. By using a cluster kit such as OSCAR, the setting up a high performance cluster, could be performed and with the help of message passing libraries like LAM/MPI, the designing numerical integration and quick sort applications, is achieved. Finally, by designing test cases and checking the performance on two client nodes. A High Performance computational cluster using Open Source Cluster Application Resource Programming has achieved with better CPU utilization.

5. Future Work

Future work includes the formal analysis of existing code for automated conversion to a parallel version for cluster implementation. The research presented here is a work-in-progress. Our goal continues to be to meet the needs of performance tuning using high performance cluster programming with better CPU utilization.

Acknowledgments

Authors would like to thank the Department of CS/IT, of Poornima College of Engineering, Jaipur and Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur for providing computational facilities and Prof. C. M. Choudhary & Prof. Anil Chaudhary for their guidance.

References

[1] Michael D. Kane, John A. Springer, "Integrating bioinformatics, distributed data management, and distributed computing for applied training in high performance computing**",** Proceedings of the 8th ACM SIGITE conference on Information technology education, Destin, Florida, USA, Pages: 33-36 Year of Publication: 2007 ISBN:978-1-59593-920-3

[2] Ian Gorton, Daniel Chavarria-Miranda, Manoj kumar Krishnan, Jarek Nieplocha, **"**A High-Performance Event Service for HPC Applications", Proceedings of the 3rd International Workshop on Software Engineering for High Performance Computing Applications, Washington, DC, USA, Page: 1, Year of Publication: 2007, ISBN:0-7695-2969-0

[3] Sadaf R. Alam, Jeffrey S. Vetter, Pratul K. Agarwal, Al Geist, **"**Performance characterization of molecular dynamics techniques for biomolecular simulations", Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel programming, New York, USA, Pages: 59 – 68, Year of Publication: 2006, ISBN:1-59593-189-9

[4] Volodymyr Kindratenko, George K. Thiruvathukal, Steven Gottlieb, "High-Performance Computing Applications on Novel Architectures", Computing in Science & Engineering, Pages: 13 – 15, Volume : 10 , Issue:6, Nov.-Dec. 2008, ISSN : 1521-9615

[5] James C Phillips, John E. Stone, **"**Probing Biomolecular Machines with Graphics Processors", Queue Bioscience**,** New York, NY, USA, Pages: 30, Volume 7 , Issue 9, October 2009, ISSN:1542-7730

[6] Chrisila Pettey, Ralph Butler, Brad Rudnick, Thomas Naughton, "Simple maintenance of Beowulf clusters in an academic environment", Journal of Computing Sciences in Colleges, Pages: 208 – 214, Volume 18 , Issue 2, December 2002

[7] Adit Ranadive, Mukil Kesavan, Ada Gavrilovska, Karsten Schwan, "Performance implications of virtualizing multicore cluster machines", Proceedings of the 2nd workshop on System-level virtualization for high performance computing, Glasgow, Scotland, Pages: 1-8, 2008, ISBN:978-1-60558-120-0

[8] Rubing Duan, Radu Prodan, Thomas Fahringer, "Performance and cost optimization for multiple large-scale grid workflow applications", Proceedings of the 2007 ACM/IEEE conference on Supercomputing, Reno, Nevada, Article No.: 12, 2007, ISBN:978-1-59593-764-3

[9] Michael A. Bauer, "High performance computing: the software challenges", Proceedings of the 2007 international workshop on Parallel symbolic computation, London, Ontario, Canada, Pages: 11 – 12, 2007, ISBN:978-1-59593-741-4

[10] Marsha Meredith, Teresa Carrigan, James Brockman, Timothy Cloninger, Jaroslav Privoznik, Jeffery Williams, "Exploring Beowulf clusters", Journal of Computing Sciences in Colleges, Pages: 268 – 284, Volume 18, Issue 4 April 2003, ISSN:1937-4771

[11] Lamia Youseff, Rich Wolski, Brent Gorda, Chandra Krintz, "Evaluating the Performance Impact of Xen on MPI and Process Execution For HPC Systems", Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, Page: 1, 2006, ISBN:0-7695-2873-1

[12] Xiao Qin, Hong Jiang, Adam Manzanares, Xiaojun Ruan, Shu Yin, "Dynamic load balancing for I/O-intensive applications on clusters", ACM Transactions on Storage (TOS), Article No.: 9, Volume 5 , Issue 3, November 2009, ISSN:1553-3077

[13] Kathryn Mohror, Karen L. Karavanic, "A study of tracing overhead on a high-performance linux cluster", Proceedings of the 12th ACM SIGPLAN symposium on Principles and practice of parallel programming, San Jose, California, USA, Pages: 158 – 159, 2007, ISBN:978-1-59593-602-8

[14] Becker, Donald J., Thomas Sterling, Daniel Savarese, John E. Darband, Udaya A. Ranawak, and Charles V. Packer. "BEOWULF: A parallel workstation for scientific computation." Proceedings of the 24th International Conference on Parallel Processing, Volume I. Boca Raton, Fla.: CRC Press, 1995.

[15] Fang, Yung-Chin; Tau Leng, Ph.D.; Victor Mashayekhi, Ph.D.; and Reza Rooholamini, Ph.D. "OSCAR 1.1: A Cluster Computing Update." Dell Power Solutions, Issue 4, 2001.

[16] Hsieh, Jenwei, Tau Leng, and Yung-Chin Fang. "OSCAR: A Turnkey Solution for Cluster Computing." Dell Power Solutions, Issue 1, 2001.

[17] Vallee G, Scott S.L, Morin C, Berthou J.-Y, Prisker H, "SSI-OSCAR: a cluster distribution for high performance computing using a single system image", Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on Issue Date: 2001 On page(s): 22 - 25 ISSN : 1550-5243 Print ISBN: 0-7695-2343-9 INSPEC Accession Number: 8598350

[18] Luethke, Brian, Thomas Naughton, and Stephen L. Scott. "C3 Power Tools: The Next Generation." Paper to be presented at the Austrian-Hungarian Workshop on Distributed and Parallel Systems (DAPSYS 2002), Linz, Austria, September-October 2002.

[19] Naughton, Thomas, Stephen L. Scott, Brian Barrett, Jeff Squyres, Andrew Lumsdaine, and Yung-Chin Fang. "The Penguin in the Pail-OSCAR Cluster Installation Tool." Paper presented at the World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, Fla., July 2002.

[20] Leangsuksun, C. ; Shen, L. ; Tong Liu ; Hertong Song ; Scott, S.L. ; "Availability prediction and modeling of high mobility OSCAR cluster" Cluster Computing, 2003. Proceedings. 2003 IEEE International Conference on Issue Date: 1-4Dec.2003 On page(s): 380 - 386Print ISBN: 0-7695-2066-9, INSPEC Accession Number: 7947717

[21] Limaye, K. ; Leangsuksun, B. ; Munganuru, V.K. ; Greenwood, Z. ; Scott, S.L. ; Libby, R. ; Chanchio, K. ; "Grid aware HA-OSCAR" High Performance Computing Systems and Applications, 2005. HPCS 2005. 19th International Symposium on Issue Date : 15-18 May 2005 On page(s): 333 - 339 ISSN: 1550-5243 Print ISBN: 0-7695-2343-9INSPEC Accession Number: 8598352

[22] Greidanus, P. ; Klok, G.W. ; "Using OSCAR to Win the Cluster Challenge" High Performance Computing Systems and Applications, 2008. HPCS 2008. 22nd International Symposium on Issue Date : 9-11 June 2008 On page(s): 47 –51 ISSN : 1550-5243 Print ISBN: 978-0-7695-3250-9 INSPEC Accession Number: 10067160

[23] Mattson, T.G. ; "High performance computing at Intel: the OSCAR software solution stack for cluster computing" Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on Issue Date : 2001 On page(s): 22 – 25 Print ISBN: 0-7695-1010-8 References Cited: 7 INSPEC Accession Number: 6957990

[24] Laxman Nathawat ; "Tools and techniques for building High Performance Linux clusters" Source Journal of Computing Sciences in Colleges archive Volume 20, Issue 6 (June 2005)Pages: 55 - 56 Year of Publication: 2005 ISSN:1937- 4771

[25] Kimura, K. ; Kodaka, T. ; Obata, M. ; Kasahara, H. ; "Multigrain parallel processing on OSCAR CMP" This paper appears in: Innovative Architecture for Future Generation High-Performance Processors and Systems, 2003 Issue Date : 17July2003 On page(s): 56 - 65 Print ISBN: 0-7695-2019-7 INSPEC Accession Number: 7979833

Dr. Sumit Srivastava received his PhD degree from University of Rajasthan, Jaipur, India. And he has received his M.Tech from Rajasthan Vidyapeeth, Jaipur and M.C.A. from Birla Institute of Technology, Mesra, Ranchi. He has more than 10 years of experience in teaching. He is currently working as an Associate Professor in the Department of Information Technology, Poornima College of Engineering, Jaipur. To his credit, he has more than 15 publishing in the proceedings of the reputed National & International conferences. He has 8 publications in various International Journals. He is also guiding thesis of many students of M.tech affiliate to Rajathan Technical University, Kota. His research interest include Statistical method in Data Mining, Grid & Cluster Computing and Network Security.

Mr. Pankaj Dadheech received his B.E. in Computer Science & Engineering from University of Rajasthan, Jaipur, India. He is currently working as a Senior Lecturer in the Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur. He has presented 7 papers in various National & International conferences. He is a member of many Professional Organizations like the IEEE and the IEEE Computer Society, CSI, & ACM. His area of interest include High Performance Computing, Security Issues in Grid Computing and Information Security.

Mr. Mahender Kumar Beniwal received his M.Tech in Computer Science from JRN, Rajasthan Vidhyapeeth University, Udaipur, India. He is pursuing his PhD at Suresh Gyan Vihar University, Jaipur. . He is currently working as a Reader in the Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur. He is a member of the IEEE and the IEEE Computer Society. His area of interest include Internet Technology, Internet Security Applications, OOPs, Network Programming, Software Engineering, Web Technology.