ISSN No. 0976-5697

Volume 4, No. 3, March 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

TECHNICAL NOTE

Available Online at www.ijarcs.info

An Approach For Parallel and Distributed Computing

Naazish Rahim
Computer Science & Engineering
CEC,Bilaspur Bilaspur,India
naazish.rahim786@gmail.com

Nikita Sharma
Computer Science & Engineering
CEC,Bilaspur Bilaspur,India
nikita.mar23@gmail.com

Sudha Tiwari
Information Technology
CEC,Bilaspur Bilaspur,India
sudhasatyen@gmail.com

Ashik hussain
Computer Science & Engineering
Govt. Engineering College Ajmer, Rajasthan
ashik.hussain07@gmail.com

Abstract: Decrease in hardware costs and advances in computer networking technologies have led to increased interest in the use of large-scale parallel and distributed computing systems. One of the biggest issues in such systems is the development of effective techniques/algorithms for the distribution of the processes/load of a parallel program on multiple hosts to achieve goal(s) such as minimizing execution time, minimizing communication delays, maximizing resource utilization and maximizing throughput. The inter-process communication amongst the various processes during the computation is the major factor for the high speed distributed computing. In this paper, the performance of the most common several parallel processing tools are discussed and compared. PVM is a software infrastructure that emulates a generalized distributed memory in heterogeneous networked environments. Now parallel MATLAB is most widely used computing environment in the distributed parallel computing. Parallel and Distributed computing techniques have proved to be effective in tackling the problem with high computational complexity in a wide range of domains, including areas of computational bioengineering

Keywords: Distributed Computing, Parallel, shared memory, distributed memory)

I. INTRODAUCTION

The concept of "*Using more than one computation resource for solving certain time consuming problems*" seems more interesting and valid. However, coordination among multiple computation resources and intelligent work distribution among them place major challenges on system designer.

Distributed computing is the process of aggregating the power of several computing entities, which are logically distributed and may even be geologically distributed, to collaboratively run a single computational task in a transparent and coherent way, so that they appear as a single, centralized system.

Parallel computing is the simultaneous execution of the same task on multiple processors in order to obtain faster results. It is widely accepted that parallel computing is a branch of distributed computing, and puts the emphasis on generating large computing power by employing multiple processing entities simultaneously for a single computation task. These multiple processing entities can be a multiprocessor system, which consists of multiple processors in a single machine connected by bus or switch networks, or a multicomputer system, which consists of several independent computers interconnected by telecommunication networks or computer networks[1].

The main purpose of doing parallel computing is to solve problems *faster* or to solve *larger* problems. Parallel computing is widely used to reduce the computation time for complex tasks. Many industrial and scientific research and practice involve complex large-scale computation, which without parallel computers would take years and even tens of years to compute. It is more than desirable to have the results available as soon as possible, and for many applications, late results often imply useless results. A typical example is weather forecast, which features uncommonly complex computation and large dataset. It also has strict timing requirement, because of its forecast nature[2][3]..

II. PARALLEL AND DISTRIBUTED COMPUTING

Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them.[3] The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel.[4] Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

In parallel computing, all processors may have access to a shared memory to exchange information between processors.

In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

Examples of distributed systems and applications of distributed computing include the following:

a. Telecommunication networks.
b. Network applications.
c. Real-time process control.
d. Parallel computation

The discussion below focuses on the case of multiple computers, although many of the issues are the same for concurrent processes running on a single computer.

Three viewpoints are commonly used:

Parallel algorithms in shared-memory model

a. All computers have access to a shared memory. The algorithm designer chooses the program executed by each computer.
b. One theoretical model is the parallel random access machines (PRAM) that are used. However, the classical PRAM model assumes synchronous access to the shared memory.
c. A model that is closer to the behavior of real-world multiprocessor machines and takes into account the use of machine instructions, such as Compare-and-swap (CAS), is that of *asynchronous shared memory*. There is a wide body of work on this model, a summary of which can be found in the literature.

Parallel algorithms in message-passing model

a. The algorithm designer chooses the structure of the network, as well as the program executed by each computer.
b. Models such as Boolean circuits and sorting networks are used. A Boolean circuit can be seen as a computer network: each gate is a computer that runs an extremely simple computer program. Similarly, a sorting network can be seen as a computer network: each comparator is a computer

Distributed algorithms in message-passing model

a. The algorithm designer only chooses the computer program. All computers run the same program. The system must work correctly regardless of the structure of the network.
b. A commonly used model is a graph with one finite-state machine per node.

In the case of distributed algorithms, computational problems are typically related to graphs. Often the graph that describes the structure of the computer network *is* the problem instance. This is illustrated in the following example.

The Parallel and distributed Computing is used to divide jobs into different parts and by use of the distributed task we can execute several tasks simultaneously.

The difference between parallel and distributed computing is, in distributed computing the workers can perform the task but cannot communicate with each other, define any number of task and the task does not run simultaneously, where as in parallel computing the workers performs the task and communicate with each other, define one task in a job and all the task run simultaneously.

The distributed computing approach enable us to coordinate and execute independent operations simultaneously on a clusters of computers or in the same computers, speeding up execution of large jobs that's why parallel computing is taken[5][6].

The Client session gives job to the job manager or job scheduler. The job manager or schedulers creates the tasks. The task created by the job scheduler will be given to workers for task completion. Workers executes the task and returns it back to the job scheduler.

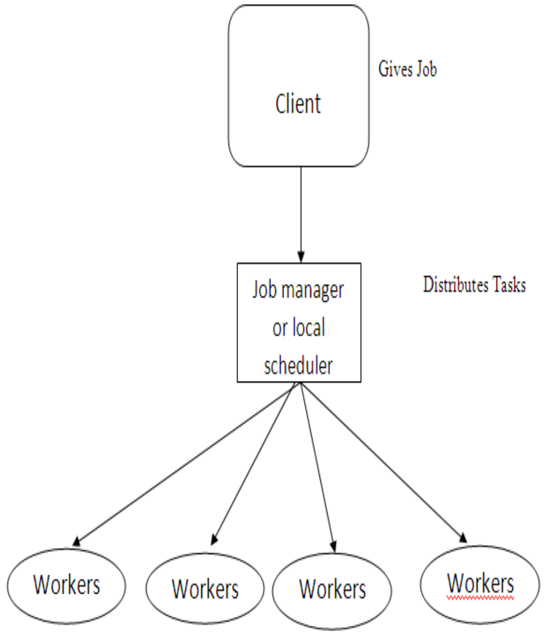

Figure: 1.1 Block Diagram of Distrubuted Computing Architecture

In simplest words, the parallelism is simultaneous processing by two or more processing units. The parallelism can be achieved by one of two main approaches:

a. Using parallel processing hardware and
b. Using distributed computing/processing system.

In the former approach, a computer system having multiple processors is used to accomplish computation task, while in latter approach, number of machines connected in network are used to accomplish computation task. However, depending upon the requirement of the application and available budget, the selection of architecture is done.

A parallel job consists of only a single task that runs simultaneously on several workers. More specifically, the task is duplicated on each worker, so each worker can perform the task on a different set of data, or on a particular segment of a large data set.

The workers can communicate with each other as each executes its task. In this configuration, workers are referred to as labs. The principle of creating and running parallel jobs is similar to distributed jobs:-

a. Find a scheduler.
b. Create a parallel job.
c. Create a task.
d. Submit the job for running.
e. Retrieve the results.

In our work we use local scheduler to simulate the result, by using distributed computing toolbox. This is the ability of the toolbox that it can run up to four workers on the client machine itself, using that we can run the parallel and distributed jobs without the clusters of computers. . The Figure 2 depicts the general Scenario of job manager, worker, and client[5][6].

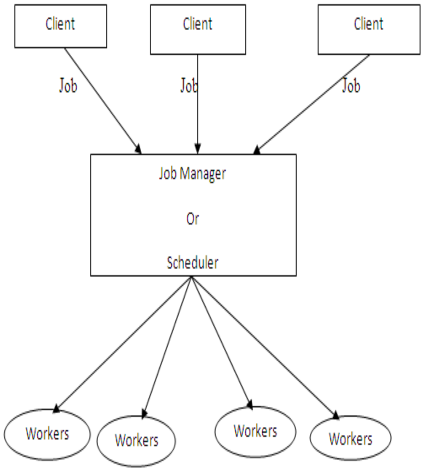

Figure: 1.2 The General Scenario of Job Manager, Worker, and Client

The distributed and parallel computing toolbox of the MATLAB provides the following properties they are as follows:-

a. ***Portability:*** - The system should be easily portable onto any parallel platform that allows MATLAB to run on each processor.
b. ***Programmability:-*** Users are provided the tools to write parallel code in MATLAB language. Parallel and distributed applications can be developed interactively in MATLAB environment.
c. ***Interactiveness :-*** Users are provided access to all processors in MATLAB environment, and this will lead parallel computation
d. ***Familiarity:*** - It is familiar to the student and ease of use.

III. PARALLEL MATLAB

It started in 1970s as an inactive interface to EISPACK[8],a set of eigenvalue and a linear system solution routines. In 1995 the use of parallel MATLAB was negligible in high performance parallel computing. Its interactive environment that provide high performance parallel computational routines making the drastic significant interest in the field of high performance computing .Interpreted built in language of MATLAB which is similar to C and the flexible matrix indexing are the innovative feature for matrix programming problem [9]. MATLAB user friendly feature and strong graphical capabilities makes it good data analysis tool. It Provide hooks to the java programming language, making integration with compiled program easy. With the help of parallel MATLAB we can implement task-parallel and data-parallel algorithms at a high level without programming for specific hardware and network architectures. A job is some large operation that we need to perform in MATLAB session. A job is broken down into segments called tasks. We decide how best to divide your job into tasks. We can divide our job into identical tasks, but tasks do not have to be identical. The MATLAB Distributed Computing Server product performs the execution of jobs by evaluating each of its tasks and returning the result to client session [10]. In file system based communication feature, MatlabMPI [11] implement basic MPI functions in parallel MATLAB using cross-mounted directories. Parallelism may achieve through polymorphism property of parallel MATLAB [12Maintaining the Integrity of the Specifications

IV. CONCLUSION

This survey provides a brief study of some research projects that are interested in providing parallel and distributed environments . The selection of the parallel programming tool through the comparison is the easiest way to select the best performing tool. In this paper the main features of the PVM, MPI and MATLAB are point out which are the most popular parallel programming tools in the distributed computing systems. . The evaluation of the performance of the any parallel high computing program may be achieved through the parallel MATLAB in the efficient manner. The performance analysis of any program by the researchers may be performed through the parallel MATLAB due to its user-friendly capability in the field of parallel computing. The research in this area still needs to address a number of open issues to be able to provide a robust, reliable and scalable parallel environment..

V. REFERENCES

[1] Harshad B. Prajapati and Dr. Sanjay K. Vij" Analytical Study of Parallel and Distributed Image Processing", 2011 International Conference on Image Information Processing (ICIIP 2011)

[2] W. Robitza , F. Schwarz , and D. Selig" Parallel and Distributed Computing - Opportunities and Challenges",

[3] Amit Chhabra, Gurvinder Singh, Sandeep Singh Waraich, bhavneet sidhu, and gaurav kumar" qualitative parametric comparison of load balancing algorithms in parallel and distributed computing environment", proceedings of world academy of science, engineering and technology volume 16 november 2006 issn 1307-6884

[4] Proceedings of world academy of science, engineering and technology volume 16 november 2006 issn 1307-6884.

[5] T. Mathworks., matlab distributed computing toolbox,system administrators guide. 2007, the mathworks, inc.,: natick, ma,.

[6] Ron choy and alan edelman” parallel matlab: doing it right”, *proceedings of the ieee*, vol. 93, no. 2, february 2005.

[7] Rafiqul Zaman Khan, Javed Ali, “ A Practical Performance Analysis of Modern Software used in High Performance Computing” , Proceeding of International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 3, March 2012.

[8] B.T. Smith, J.M. Boyle, J.J. Dongarra, B.S. Garbow, Y. Ilebe, V.C. Kelma, and C.B. Moler.,“Matrix Eigensystem Routines- EISPACK Guide”. Springer-Verlag, 2nd edition, 1976.

[9] C. Stephane, B. M. Francois “An Environment for High Performance MATLAB”.In Languages, Compilers,and Run-Time Systems for Scalable Computers, pp 27–40, 1998.

[10] R. Choy,“ parallel matlab survey”. Http://theory.lcs.mit.edu/cly/survey.html, 2001.

[11] Geist, J. A. Kohl, and P. M.Papadopoulos,“PVM and MPI: A Comparison of Features”, Calculateurs Paralleles, 8(2), 1996.

[12] R. Choy,a. Edelman“parallel matlab:doing it right”,massachusetts institute of technology ,cambridge,pp-10,november 2003.