Performance Analysis of Scheduling Algorithms in Simulated Parallel Environment

Anupreet Kaur[1], Pawan Kumar[2]

[1]IITT College of Engineering, Pojewal, Nawanshahr, Punjab, India
anupreetbajwa.27@gmail.com.

[2]HOD and Dean Academics,
IITT College of Engineering, Pojewal, Nawanshahr, Punjab, India
pawanjal123@gmail.com

Abstract: ***Performance analysis of Scheduling Algorithms in Simulated Parallel Environment that enables to make the system to evaluate and compare the performance of various Scheduling Algorithms. Efficiency of scheduling algorithms is essential in order to attain optimal performance from parallel programming systems. The proposed work consider a connected set of workstations as a "pool of processors" to analyze the performance of scheduling algorithms. Scheduling algorithms select the best possible subset of workstations for a task to minimize its completion time. Virtualization of parallelization environment is carried out with the help of simulated program, which simulates all components of actual processors so as to give best possible outcome. Such simulated framework will provide the stage for the young researchers to model and evaluate their scheduling policies on virtual parallel environment. The paper presents first come first serve, smallest job first and round robin scheduling approaches where general problem is analyzed and modeled for a simple scenario. The performance of proposed schemes is evaluated and benefits are highlighted in terms of Response time, Turnaround time, Waiting Time, Total Turnaround time based on simulation study.***

Keywords: about four key words separated by commas**.**

1. Introduction

In parallel processing, the parallel portion of the application can be accelerated according to the number of processors allocated to it. In a homogeneous architecture, where all processors are identical, the sequential portion of the application will have to be executed in one of the processors, considerably degrading the execution time of the application. Two main distinguishing features of parallel versus sequential programming are program Partitioning and task scheduling. Both techniques are essential to high-performance computing on both homogeneous and heterogeneous systems. The partitioning problem deals with how to detect parallelism and determine the best trade-off between parallelism and overhead, which means finding the best grain size that maximizes parallelism while reducing overhead. After program partitioning, tasks must be optimally scheduled on the processors such that the overall make span of the parallel application is minimized. In general, the scheduling problem deals with choosing the order in which a certain number of tasks may be performed and their assignment to processors in a parallel/distributed environment.

2. Brief Literature Survey

We already have different Scheduling Algorithms, which is being run sequentially where different jobs run in a serial manner. Though the quality of parallelization has improved in the past several decades, fully automatic parallelization of sequential programs by compilers remain a grand challenge due to its need for complex program analysis and the unknown factors (such as input data range) during compilation. In parallelization we partition the computational work and associated memory among the different processors. The programming control structures on which parallelization places the most focus are loops, because in general, most of the execution time of a program takes place inside some form of loop. A parallelizing compiler tries to split up a loop so that its iterations can be executed on separate processors concurrently.

Actual experimentation on multiprocessor or parallel system [9] is still a costly and complex approach and moreover these systems are still out of reach to young researchers even doing research in higher education institutes like universities or technical colleges in developing country like India there may be several reasons for the non-availability of these systems.

3. Scheduling Algorithm

In the design of scheduling algorithms for efficient parallel processing, there are four fundamental aspects for the design of scheduling algorithms: Performance, Time complexity, Scalability and Applicability. By high performance we mean the scheduling algorithms should produce high quality solutions. The algorithms must be robust so that they can be used under a wide range of input parameters. Scheduling algorithms should have low time-complexity. The time complexity of an algorithm is an important factor so far as the quality of solution is not compromised. Parallel scheduling algorithms must be scalable. On the one hand, the problem should possess problem-size scalability, that is, the algorithms consistently give a good performance even for large input. On the other hand, the algorithms should possess processing-power scalability, that is, given more processors for a problem, the parallel scheduling algorithms produce solutions with almost the same quality in a shorter period of time. Scheduling algorithms could be used in practical environments. To achieve this goal one must take into account realistic assumptions about the program and multiprocessor models.

4. Methodology

Computer system users, administrators, and designers usually have a goal of highest performance at lowest cost. Modeling and simulation [12] of system design trade off is good preparation for design and engineering decisions in real world jobs. System Simulation is the mimicking of the operation of a real system, such as the day-today operation of a bank, or the value of a stock portfolio over a time period, or the running of an assembly line in a factory, or the staff assignment of a hospital or a security company, in a computer. Instead of building extensive mathematical models by experts, the readily available simulation software has made it possible to model and analyze the operation of a real system by non-experts, who are managers but not programmers. A simulation is the execution of a model, represented by a computer program that gives information about the system being investigated.

The simulation approach of analyzing a model is opposed to the analytical approach, where the method of analyzing the system is purely theoretical. As this approach is more reliable, the simulation approach gives more flexibility and convenience. The activities of the model consist of events, which are activated at certain points in time and in this way affect the overall state of the system. The points in time that an event is activated are randomized, so no input from outside the system is required. Events exist autonomously and they are discrete so between the executions of two events nothing happens.

In the field of simulation, the concept of "principle of computational equivalence" has beneficial implications for the decision-maker. Simulated experimentation accelerates and replaces effectively the "wait and see" anxieties in discovering new insight and explanations of future behavior of the real system.

5. Steps for the Working of Simulator

1. Initially value for the jobs is assigned.
2. Then information about number of processors is fed to the simulator as shown in snapshot.
3. On the basis of number of process arrived it equally divides the load among different processors.
4. When jobs are divided among processors they started giving response and at run time choose the scheduling algorithm according to requirement.
5. Then show the simulator and start the simulation where jobs start running and we get the response time, waiting time and Turnaround time. As information about Turnaround time of various jobs at any instant is available, total turnaround time of the processor is generated.

6. System Terminology

State: A variable characterizing an attribute in the system such as level of stock in inventory or number of jobs waiting for processing.

Event: An occurrence at a point in time, which may change the state of the system, such as arrival of a customer or start of work on a job [14].

Entity: An object that passes through the system, such as cars in an intersection or orders in a factory. Often an event (e.g., arrival) is associated with an entity (e.g., customer).

Queue: A queue is not only a physical queue of people, it can also be a task list, a buffer of finished goods waiting for transportation or any place where entities are waiting for something to happen for any reason.

Scheduling: Scheduling is an act of assigning a new future event to an existing entity.

Random variable: A random variable is a quantity that is uncertain, such as inter arrival time between two incoming flights or number of defective parts in a shipment.

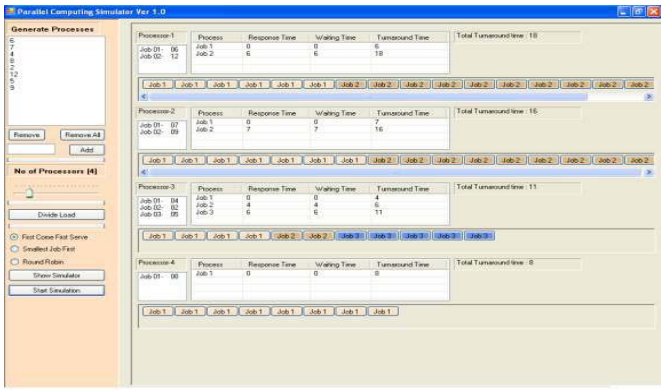

Figure 1: Simulator is started

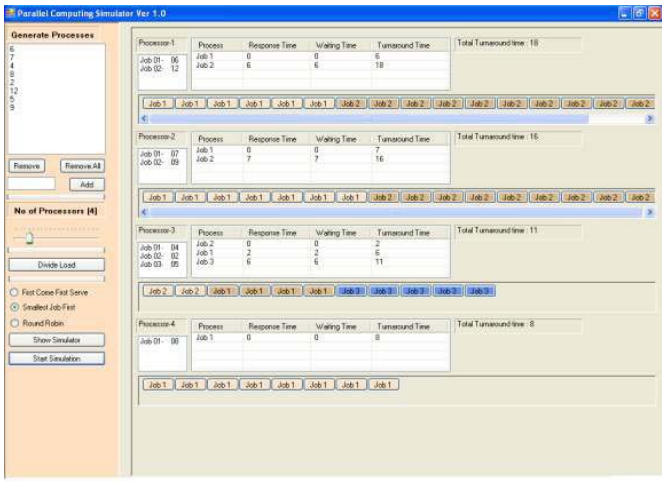

Figure 2: Smallest job first strategy has been chosen & simulation is generated.

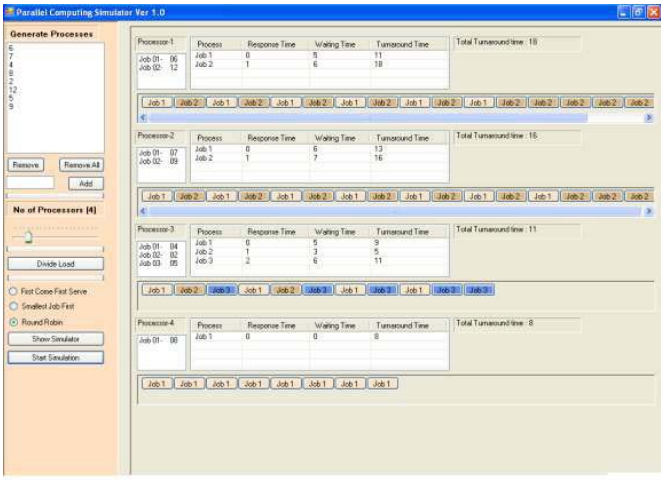

Figure 3: Round Robin strategy has been chosen and simulation is generated.

7. Performance Metrics of Parallel Systems

Speedup: Speedup Tp is defined as the ratio of the serial runtime of the best sequential algorithm for solving a problem to the time taken by the parallel algorithm to solve the same problem on p processor. The p processors used by the parallel algorithm are assumed to be identical to the one used by the sequential algorithm.

Cost: Cost of solving a problem on a parallel system is the product of parallel runtime and the number of processors used E = p.Sp

Efficiency: Ratio of speedup to the number of processors. Efficiency can also be expressed as the ratio of the execution time of the fastest known sequential algorithm for solving a problem to the cost of solving the same problem on p processors. The cost of solving a problem on a single processor is the execution time of the known best sequential algorithm.

Cost Optimal:A parallel system is said to be cost-optimal if the cost of solving a problem on parallel computer is proportional to the execution time of the fastest known.

8. Conclusion

The presented work demonstrated the advantages of deploying a scheduling algorithm method in a parallel system. It had presented an scheduling algorithm method and demonstrated its favorable properties, both by theoretical means and by simulations. The value of the proved minimal disruption property of the mapping adaptation has been demonstrated in the extensive set of simulations. Such a scheme is particularly useful in systems with many input ports and packets requiring large amounts of processing. With the proposed scheme, a kind of statistical multiplexing of the incoming traffic over the multiple processors is achieved, thus in effect transforming a network node into a parallel computer. The improvements of processor utilization decrease the total system cost and power consumption, as well as improve fault tolerance.

This paper presents a simulation environment developed with the aim to facilitate the research of multiprocessor systems as well as performance measurement of scheduling algorithms in developing countries. A simulator program was coded in VB.net to fulfil this purpose. In future the work done in this thesis can be extended by modeling many more scheduling algorithms in the developed environment. Effort will be done in future to validate the data captured by simulator with actual experimental setup.

References

[1] Almasi, G.S. and A. Gottlieb (1989), Highly Parallel Computing. Benjamin-Cummings publishers, Redwood City, CA
[2] Hillis, W. Daniel and Steele, Guy L., Data Parallel Algorithms Communications of the ACM December 1986
[3] Quinn Michael J, Parallel Programming in C with MPI and OpenMP McGraw-Hill Inc. 2004. ISBN 0-07-058201-7
[4] IEEE Journal of Solid-State Circuits:"A Programmable 512 GOPS Stream Processor for Signal, Image, and Video Processing", Stanford University and Stream Processors, Inc.
[5] Barney, Blaise. "Introduction to Parallel Computing", Lawrence Livermore National Laboratory
[6] Bill Dally, Stanford University: Advanced Computer Organization: Interconnection Networks
[7] Quinn Michael J, Parallel Programming in C with MPI and OpenMP McGraw-Hill Inc. 2004, ISBN 0-07-058201-7
[8] Albert Y.H. Zomaya, Parallel and distributed Computing Handbook, McGraw-Hill Series on Computing Engineering, New York (1996).
[9] Ernst L. Leiss, Parallel and Vector Computing, A practical Introduction, McGraw-Hill Series on Computer Engineering, New York (1995).
[10] Vipin Kumar, Ananth Grama, Anshul Gupta, George Karypis, Introduction to Parallel Computing, Design and Analysis of Algorithms, Redwood City, CA, Benjmann /Cummings (1994).
[11] X. Sun and J. Gustafson, "Toward a Better Parallel Performance Metric," Parallel Computing, Vol. 17, No. 12, Dec. 1991, pp. 1093-1109.
[12] Bossel H., Modeling & Simulation, A. K. Peters Pub., 1994
[13] Ghosh S., and T. Lee, Modeling & Asynchronous Distributed Simulation: Analyzing Complex Systems, IEEE Publications, 2000.
[14] Fishman G., Discrete-Event Simulation: Modeling, Programming and Analysis, Springer-Verlag, Berlin, 2001
[15] Woods R., and K. Lawrence, Modeling and Simulation of Dynamic Systems, Prentice

Author Profile

Anupreet Bajwa obtained her BTECH degree in computer science and Engineering from IITT College of engineering in 2009. She is currently a M.TECH student under the supervision of Dr.Pawan kumar HOD and Dean academics,IITT college of Engineering. Her research is Performance Analysis of Scheduling Algorithms in Simulated Parallel Environment.