QoS Oriented Enhancement based on the Analysis of Dynamic Job Scheduling in HPC

Reshmi Raveendran, SBM College of Engineering and Technology, Dindigul, India
D. Shanthi Saravanan, PSNA College of Engineering and Technology, Dindigul, India

ABSTRACT

With the advent of High Performance Computing (HPC) in the large-scale parallel computational environment, better job scheduling and resource allocation techniques are required to deliver Quality of Service (QoS). Therefore, job scheduling on a large-scale parallel system has been studied to minimize the queue time, response time, and to maximize the overall system utilization. The objective of this paper is to touch upon the recent methods used for dynamic resource allocation across multiple computing nodes and the impact of scheduling algorithms. In addition, a quantitative approach which explains a trend line analysis on dynamic allocation for batch processors is depicted. Throughout the survey, the trends in research on dynamic allocation and parallel computing is identified, besides, highlights the potential areas for future research and development. This study proposes the design for an efficient dynamic scheduling algorithm based on the Quality-of-Service. The analysis provides a compelling research platform to optimize dynamic scheduling of jobs in HPC.

Keywords: Batch Processors, Dynamic Allocation, Evolutionary Algorithms, Quality-of-Service, Resource Allocation, Scheduling

1. INTRODUCTION

High Performance Computing (HPC) refers to the practice of aggregating computing power to a single problem in ways that would deliver much higher performance than that derived through a typical desktop computer or workstation. Usually such problems pose significant challenges in science, engineering and business applications. The purpose of having a high performance computing is to address these challenges by grouping of individual nodes that can work together to solve a larger problem much efficiently than any single computer can do.

A change is eminent in nature, and this seems certainly to be true, in the market of HPC. This market has progressively undergone rapid change of vendors, architectures, technologies and the usage of systems. The performance of microprocessors has been increasing in an exponential way for the last four decades and so the scope of computer’s processing ability does not have a fixed definition. A computer is considered to be high performance, if it uses...
Multiple processors (tens, hundreds or even thousands) connected together by some network to exceed the performance of a single processor. Use of multiple processors to enhance computing performance is known as parallel computing. Parallel computing includes several computations carried out simultaneously. In parallel computation, the jobs must be scheduled in a better way to achieve better results.

The first systematic approach to scheduling problem was undertaken in the mid of 1950s. In the second half of the seventies the introduction of vector computer systems marked the beginning of modern supercomputing. And in the eighties, the integration of vector computation in conventional computing environment became more important. Thereafter Massive Parallel Processing (MPP), which utilizes a large number of processors to perform a set of coordinated computations in parallel, became successful. The price/performance ratios contributed to the success. Later it was replaced by microprocessor based Symmetric Multiprocessor Systems (SMS) which are tightly coupled multiprocessor systems with a pool of homogenous processors running independently on different data and with the capability of sharing resources.

High performance computation accelerates calculations much faster than a conventional processor, thereby speeding up many time consuming computations. The objective of High Performance Center is not only to minimize the make span or waiting time but also to maximize the user’s satisfaction regarding the system behavior. Better job allocation gives better performance which leads to user satisfaction.

Scheduling the execution of parallel algorithms on parallel computer is an important and challenging area in current research. It is the job of a scheduler to determine when, where and how the given task should run and upon which the resource managers can be directed inappropriately. Parallel computers can be used to solve scheduling problems very fast. Although the importance of parallel scheduling algorithms has been widely recognized, only few results are obtained so far. Scheduling problem can be graphically depicted as in Figure 1.

Programming parallel applications is difficult and not worth the effort unless large performance gains can be realized. Scheduling is a key part of the workload management software which usually performs queuing, scheduling, monitoring, resource management and accounting. The critical part of scheduling is to balance policy enforcement with resource optimization in order to pick the best job to run. A scheduler selects the best job to run based on the defined policies and available resources, then executes the job and after completion, cleans up and loops for the next job. Individual nodes can be allocated with number of tasks which is less than its maximum load. The load of the processor is the sum of the processing times of the tasks assigned to it. A parallel computation consists of a number of tasks, \( T_1, T_2, T_3, \ldots, T_n \) that have to be executed by a number of parallel processors \( P_1, P_2, P_3, \ldots, P_m \). The task \( T_j \) requires processing time \( P_j \) and is to be processed sequentially by exactly one processor. The length of a schedule is the maximum load of any processor. The aim is to find a schedule with minimum length which powers better performance. I.e. a classical problem in scheduling theory is to compute a minimal length scheduler for executing ‘n’ unit length tasks on ‘m’ identical processors constrained by a precedence relation.

In parallel computation, the scheduling and the mapping tasks are considered to be the most critical problem. Parallel computation solves a problem by breaking it into subtasks and working on those subtasks at the same time. These application subtasks are assigned to parallel machines for execution, by an efficient scheduler that guarantees minimum execution time and satisfies load balancing between processors of parallel machines. High performance computing ensures better performance by the utilization of an efficient scheduling algorithm. These parallel machines infrastructure may be homogenous or heterogeneous. Homogenous systems use same machine power and performance, while heterogeneous infrastructure includes machines that differ in their performance, speed and interconnections.
Current Trends in Rough Set Flow Graphs
www.igi-global.com/chapter/current-trends-rough-set-flow/28471?camid=4v1a

Rhotrix Multiplication on Two-Dimensional Process Grid Topologies
www.igi-global.com/article/rhotrix-multiplication-two-dimensional-process/62995?camid=4v1a