Analysis of Scheduling Algorithms on a Parallel Programming Context

Peter N. Nyumu
Supervisor: Alfredo Goldman

IME - USP

November 17, 2009

Motivation

- Understand Scheduling of tasks in grid enviroment

Categories of scheduling algorithms

- Uniprocessor scheduling algorithms - involve one processor; Divided in two
- Multiprocessor they is more than one processor available to execute the jobs.

Scheduling in Operating System

- The scheduler is concerned mainly with:
- CPU Utilization - This is the capacity to keep the cpu busy as much as possible as long there is job to process.
- Throughput - A measure of work in terms of the number of processes that are completed per time unit.
- Turnaround time - The interval from the time of submission of a process to the time of completion.
- Waiting time - The sum of periods spend waiting in the ready queue.
- Response time - The time from the submission of a request until the first response is produced.

Scheduling in Operating System

- The best optimization occur when It is possible to maximize CPU utilization and throughput and to minimize turnaround time, waiting time, and response time.

Operating System Algorithms

- Round-Robin algorithm
- Weighted Round Robin Algorithm - weight factor
- Deficit Round Robin Algorithm - packets of different size
- Elastic Round Robin Algorithm - shared resources to multiple request
- Fair Queuing Algorithm - max-min criterion
- FCFS
- Shortest Job First

Operating System Algorithms

- Self-Clocked Fair Queuing Algorithm - virtual clock
- Start-time Fair Queuing Algorithm - schedule in increase of the start tag
- Weighted Fair Queuing Algorithm - support different bandwidth
- Earliest deadline first scheduling - earliest deadline has the highest priority
- Least Laxity First scheduling - higher priority to least(flexibility to schedule) laxity.
- Maximum Urgency First Algorithm - fixed and dynamic priority scheduling.

Operating System Algorithms

- Poorly documented algorithms;

Operating System Algorithms

- Poorly documented algorithms;
- Foreground-background
- Gang scheduling
- Highest Response Ratio Next
- Lottery Scheduling is a probabilistic scheduling algorithm

Brief look at Parallel computing

- In parallel computing the scheduling heuristics can be grouped in to two categories: online mode and batch-mode heuristics.
- Parallel computing rely on multithreading for efficient execution, thus relying on the schedule of the threads among the processors
- Two models used in parallel machine are Cilk and Kaapi, both of them use work stealing algorithm for executation.

Brief look at cluster computing

- Cluster computing is best characterized as the integration of a number of off-the shelf commodity computers and resources integrated through hardware, networks, and software to behave as a single computer
- The application model best suited for cluster enviroment is the Parallel Tasks (PT) model

Taxonomy of Grid Computing

- Local vs. Global
- Static vs. Dynamic
- Optimal vs. Suboptimal
- Approximate vs. Heuristic
- Distributed vs. Centralized
- Cooperative vs. Non-cooperative

Master-Slave Application on Grid Computation

- The master is responsible for decomposing the problem into small tasks, as well as for gathering the partial results in order to produce the final result of the computation.

Master-Slave Application on Grid Computation

- The master is responsible for decomposing the problem into small tasks, as well as for gathering the partial results in order to produce the final result of the computation.
- The slave processes execute in a very simple cycle: receive a message from the master with the next task, process the task, and send back the result to the master.

Master-Slave Application on Grid Computation

- The master is responsible for decomposing the problem into small tasks, as well as for gathering the partial results in order to produce the final result of the computation.
- The slave processes execute in a very simple cycle: receive a message from the master with the next task, process the task, and send back the result to the master.
- In evaluating a Master-Slave application, two performance measures of particular interest are speedup and efficiency.

Master-Slave Application on Grid Computation

- The most important objective functions: the minimization of the makespan (or total execution time), the minimization of the maximum response time (difference between completion time and release time), and the minimization of the sum of all response times.

Master-Slave Application on Grid Computation

- The most important objective functions: the minimization of the makespan (or total execution time), the minimization of the maximum response time (difference between completion time and release time), and the minimization of the sum of all response times.
- Have on-line scheduling problems; where release times and sizes of incoming tasks are not known in advance.

Master-Slave Application on Grid Computation

- How to adapt the OS algorithms in Master/Slave context

Master-Slave Application on Grid Computation

- How to adapt the OS algorithms in Master/Slave context
- For example:

Master-Slave Application on Grid Computation

- How to adapt the OS algorithms in Master/Slave context
- For example:
- Round Robin (RR) is the simplest algorithm. It simply sends a task to each slave one by one, according to a prescribed ordering. This ordering first chooses the slave with the smallest $w_i + c_i$, then the slave with the second smallest value, etc.
- No need of preemption

Master-Slave Application on Grid Computation

- How to adapt the OS algorithms in Master/Slave context
- For example:
- Round Robin (RR) is the simplest algorithm. It simply sends a task to each slave one by one, according to a prescribed ordering. This ordering first chooses the slave with the smallest $w_i + c_i$, then the slave with the second smallest value, etc.
- No need of preemption
- Other algorithms ;

Master-Slave Application on Grid Computation

- The problem ; $MS|On-line; r_i; w_j; c_j|C_{max}$

Master-Slave Application on Grid Computation

- The problem ; $MS|On - line; r_i; w_j; c_j|C_{max}$
- scheduling different-size tasks on a homogeneous platform reduced to a master and two identical slaves, without paying any cost for the communications from the master, and without any release time, is already an NP-hard problem

Master-Slave Application on Grid Computation

- OAR scheduler

Master-Slave Application on Grid Computation

- OAR scheduler
- Priority Queue, Use FCFS

Master-Slave Application on Grid Computation

- This work will continue; - simulating the algorithms in on-line, off-line or batch mode is the next step.

Master-Slave Application on Grid Computation

- This work will continue; - simulating the algorithms in on-line, off-line or batch mode is the next step.
- Check the results after two years...

Questions