Application-Aware Resource Allocation in General Purpose Operating System

SHRUTI MEHTA
School of Electrical Engineering and Computer Science
Center for Innovation in Technology for Learning

BHAVIN MEHTA
Department of Mechanical Engineering

Ohio University, Athens, Ohio 45701, USA

Abstract
The Real-Time (RT) class on the Solaris operating system was evaluated to study the feasibility and effects of change in priority and time quantum within an RT class for competing processes. Several experiments were conducted and the results will be compared and discussed in the paper. The application of real-time process control will be discussed in context of a virtual manufacturing environment and e-Commerce infrastructure.

Key-Words: Resource Allocation, Virtual Manufacturing, Real-time Priority, eCommerce, Real-time Class, Time Quantum

1 Introduction

To meet the service demands created by the Internet's exponential growth, operators are scrambling to deploy application-level services, including Web caches, commerce servers, and intelligent transformation proxies for mobile thin clients and other Web Services. On one hand, the Internet's growth rate places unprecedented scalability and robustness demands on these services; on the other hand, that same growth rate demands that new services be developed, deployed, and evolved. Coping with this uncertainty requires the ability to retrieve and present data at varying degrees of fidelity without compromising the quality of service. In this paper we present application-aware adaptation using real-time priorities and process control as a solution to this problem. The real-time class on a general-purpose operating system, like Solaris, will be evaluated by running experiments with different priorities and time-quanta for competing processes and comparing their results. The results will be beneficial in understanding the performance of the real-time priorities and time-quanta for several competing applications running on a server. For example, to design and develop a new product in a Virtual Manufacturing environment there are several material processes like Extrusion, Rolling, Forging, Casting, Machining etc., which have to be analyzed. The simulation time varies by process and product complexity. Depending on the product, some processes might be connected and the simulation/analysis has to be performed in a particular order. Resource allocation on the server is very important to obtain optimum performance and attain new product development in a timely and cost effective fashion. The essence of our solution is first step in a collaborative partnership between applications and the operating system. This can be achieved by monitoring the level and quality of resources and by informing applications about any relevant changes in their environment and scheduling them according to their priority and process time requirements. Environmental changes include memory allocation, CPU usage, priority, time-quanta and even the server it-self in a distributed environment. Applications must be agile, that is able to receive events in an asynchronous manner and react appropriately. There is a need for a central point for managing resources and authorizing any application initiated request for their use.

2 Motivation

An International Data Corporation (IDC) survey on application availability showed that the application downtime is caused by four main reasons:

Table 1: Survey Results

% Of Downtime	Reason
44.6%	Systems Failure
28.5%	Application Failure
18.2%	Network Outage
8.7%	Administration

Addressing a single component of the environment is not going to help ensure ones application service level. One needs to address all of the Network, Application and System components. As seen from the table 1, system level failure is the major contributing factor in the application downtime and thus becomes a critical variable, which is being considered in this research.

One needs to ensure optimum uptime, availability, and performance of the entire virtual manufacturing or e-business application environment: network, systems, and applications for any integrated system. We need to provide solutions that actively monitor, control, and assure service levels of applications with timeliness and cost effective solution. By combining service level monitoring with service level control, one can take real-time, automated control of the various components of the simulations in the Virtual Manufacturing environment or e-business infrastructure. Multi-tier intelligence and control are essential at the application and system layers because they monitor the availability and performance of mission-critical components.

A real-time resource manager is one in which the requirements for correct operation and efficient service include timeliness of simulations or order processing in the virtual manufacturing or e-Commerce applications. Typically, the timeliness requirement is expressed as a constraint or a deadline on the execution of processing activity. The processing activity may be static or dynamic. A static real-time server has a constant or bounded processing demand, whereas a dynamic real-time server does not. To meet real-time deadlines, it is important to insure that sufficient processing resources are allocated to the application. Two approaches for resource allocation are possible. With the static allocation approach, the allocation of resources never changes after system design. While this approach has the advantage of being supported by well-established engineering methods and tools, it is inflexible and thus it is not ideal for dynamic real-time systems. Dynamic allocation approaches adjust the allocation of computing resources as the processing demands of the real-time application change.

3 Background

The Time-Share (TS) class on general purpose Unix operating system is designed to maximize the system throughput in a multi-user, time-sharing environment, which meant lowering the priorities of resource intensive jobs automatically. This can be a problem with applications like simulations in virtual manufacturing and eCommerce were timeliness is important for different application connectivity. The scheduler for those application designs needs a firm control of the execution priority of selected processes. The operating system controls the scheduling of priorities within the TS class. In contrast to that, the Real-time (RT) class allows user controllable scheduling of priorities and time quanta. This is ideal for systems developing and deploying real-time applications. The demand for real-time systems that can effectively operate within dynamic environments is increasing in many applications. The RT class on the Solaris system was evaluated to study the feasibility and effects of change in priority and time quantum within an RT class for competing processes. The real-time capabilities in the Solaris Operating Environment are derived from the fundamental capabilities of its innovative architecture, including a fully preemptable multithreaded kernel, priority-based scheduler, and precision timers and clocks. Together, these and other features assure bounded response time and enable a wealth of real-time application possibilities.

3.1 Solaris Scheduler

Solaris kernel implements multiple scheduling classes, where each class has a defined range of priorities. There are four types of classifications that can be used when managing the scheduling priorities of a process or set of processes. These classes are the real-time (RT), System (SYS), Interactive (IA) and Time-Share (TS). RT (priority

100-159) class threads provide support for applications that require minimal dispatch latency. RT class threads are the highest priority threads on a Solaris system, with the exception of interrupt threads, and they implement a fixed-priority scheme.

The different scheduling classes may implement a dispatch table for defining time quantum at a given priority level. From Figure 1, one can see that there are two ways to view priorities: the global priority, which is a unique priority across all loaded scheduling classes, and a per-class range of priorities (TS/IA priorities range from 0 to 59, SYS priorities range from 0 to 39, etc.). If the RT class is loaded, RT priority 0 corresponds to global priority 100. In Solaris, the higher the number, the higher is the priority. If RT class is configured on the system, any running real-time process should obtain the services of the CPU before a process that belongs to another class. Solaris implements a queue of queues, in which each CPU has a set of queues, one queue for every priority. Each priority value that is configured into the system has a separate scheduling queue that the system's process scheduler manages. Processes with the same priority value share the same scheduling queue or in other words, every thread at a given priority is placed on a linked list in a processor's queue.

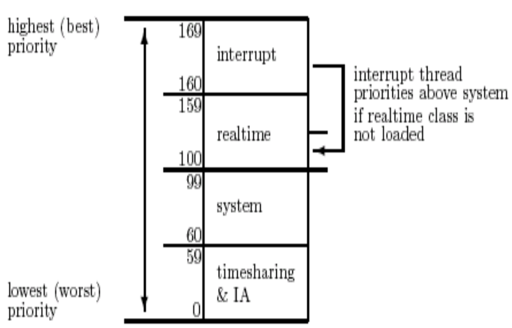

Fig. 1 Scheduling classes

An RT thread runs until it completes, voluntarily sleeps (issues a blocking system call), or is preempted due to a higher priority thread. The *priocntl*(1) command works with RT class threads and manages scheduling properties of process. In the RT class the global priorities ranges are from 0 to 59. *priocntl*(1) can be used to change the priority of an RT class thread, or alter the allotted time quantum. The time quantum value determines the maximum time that a running process, which has not entered a resource or event wait state (**sleep**), may run. Note that regardless of the time quantum specified, if another process at a higher priority makes a request to run, a running process may be preempted before receiving its full time quantum. A process that is preempted by a higher priority process remains first its scheduling queue with the remainder of its specified time quantum still available. In order to change the priority or time quantum setting of a real-time process the process invoking the *priocntl* function must have super user privileges or must itself be a real-time process whose real or effective user ID matches the real or effective user ID of the target process. The real-time priority and time quantum are inherited. The table 2 shows the experimental runs to demonstrate the effect of change in priority and time quantum for two competing processes within the RT class.

Table 2: Experimental setup

Experiment Number	Application	Priority	Time Quantum	Comments
1	P1	20	600	Default time-quanta (TQ)
	P2	20	600	
2	P1	20	100	Same priority & TQ
	P2	20	100	
3	P1	20	100	P2 has higher priority
	P2	30	100	
4	P1	20	100	P2 has higher TQ
	P2	20	200	
5	P1	30	100	P1 has higher priority while P2 has higher TQ
	P2	20	200	
6	P1	30	200	P1 has higher priority and Higher TQ
	P2	20	100	

3.2 Virtual Manufacturing Environment

The main objective of this research is to create a virtual manufacturing environment for the Internet that would allow users around the world to design dies and molds and analyze the flow of material inside the dies or forming processes like Extrusion, Rolling, Forging, Casting, machining and Sheet Metal Forming.

- Creation of a Web interfaces to bring together all the die design and analysis modules

- Creation of an AI shell for selecting an appropriate manufacturing process
- Addition of a drafting tool on the WWW for product geometry input
- Implementation of the Extrusion Process (STREAM, & SHEAR) on the WWW and addition of a feeder plate module in SHEAR
- Implementation of the Drawing die design package on the web
- Implementation of Rolling (profile and ring) on the web
- Implementation of Forging Process on the web
- Design of Casting and Sheet Metal forming Processes on the web (Future)

The research is part of the Virtual Material Processing (VMP) project underway at Ohio University (OU). The primary objective of this project was to develop a methodology and software that will use Artificial Intelligence (AI), Simulation and Modeling coupled with VR (virtual reality concepts) for the automated design, analysis and visualization of various material forming processes. The environment is designed to be platform independent and accessible over the World Wide Web (WWW). The objective is to recommend a suitable manufacturing process for achieving the final product shape by providing the user the option of designing the appropriate dies and molds, and then carrying out the analysis using Slab method, Upper-Bound or Finite Element method and visualizing the manufacturing process in an immersive and interactive environment. The user can get full control of the process using a haptic (feedback) control and head mounted display (HMD), with an option to modify the process as it proceeds (under development). The researcher can evaluate competing processes and also learn about forming processes through the web-based tutorials and training material. It is an ideal tool for research, experimentation and training. Tools like Java, CGM, VRML, and other tool-kits were used to develop the interfaces and programs to run under Netscape and Internet Explorer. The programs/packages for die design, slab and upper bound solutions for flow analysis and material selection, developed in-house, are being used to support the real-time virtual material processing environment. The environment is fully distributed and can be accessed from anywhere in the world using the Internet. The user accesses the VMP system using a web browser on their desktop, provides the necessary data which is send to the web server, which in-turn sends it to the appropriate solver on the Silicon Graphics machine, Sun Server, or the Supercomputer. The graphical and numerical results are displayed on the user's desktop/browser. The user has the choice of saving the data in the database and/or downloading it for further analysis, report generation and confidentiality. Resource management tools will be very beneficial in optimally executing the simulation modules with-in the virtual manufacturing environment and dynamically updating the priorities, and time-quanta of the analysis process when required. Just like express mail, the user could request a fast analysis at a higher cost; the resource management module would then allocate a higher priority and a larger time-quantum and/or change the priority class to accommodate the express analysis request.

4 Test-bed Configuration

The experiments were conducted using single processor Sun Ultra 10 workstation. The workstation was running Solaris 8 Operating System

Two identical programs, P1 and P2, were developed as competing processes. Each program had one outer loop were the counter was set to max integer. Within the loop the statements were designed to be CPU intensive. The timestamp was recorded in an array within the loop using the gettimeofday() function. The timestamp array was printed out to a file at the end of the program, outside the loop; to eliminate any I/O related premature swapping of the processes. The timestamp was used to determine the swapping between the two competing processes, P1 and P2. The controller program that was running at the higher priority than two applications was changing the RT priority and time quantum

5 Observations

The above experiments showed that it is possible to change/control the priority and time quantum of competing processes in the Real-time class. A process with higher priority executes before the process with lower priority, even if the time quanta for the lower priority process has not expired and was submitted earlier. Two processes with the same priority run for their allocated time quantum in the order of execution. The priority and time quanta along with memory allocation and CPU time can be implemented with-in a virtual manufacturing environment. Depending on the type of analysis, the complexity of the product design and the priority/timeliness of the application, the simulation can be placed in a real-time class and the allocated time-quanta can be increased or decreased.

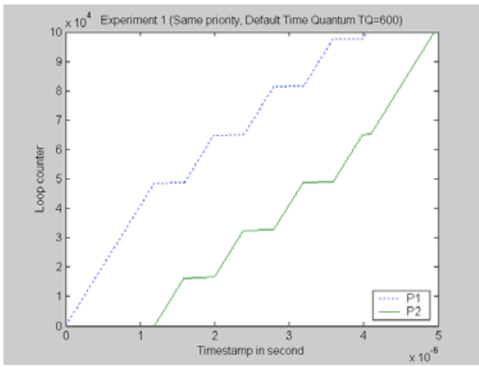

Fig 2 Experiment 1

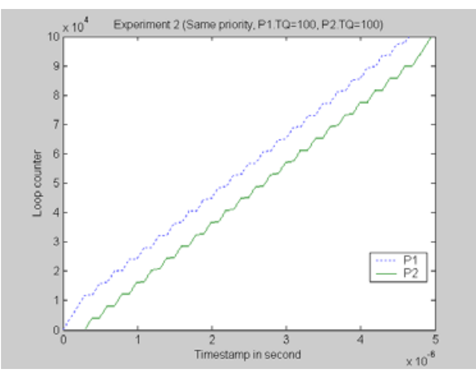

Fig. 3 Experiment 2

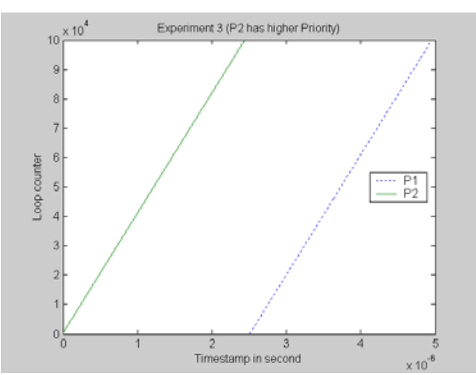

Fig. 4 Experiment 3

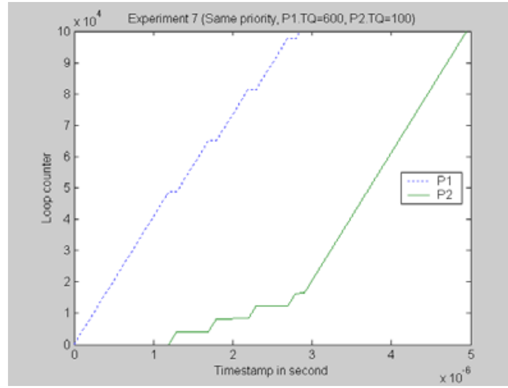

Fig. 5 Experiment 4

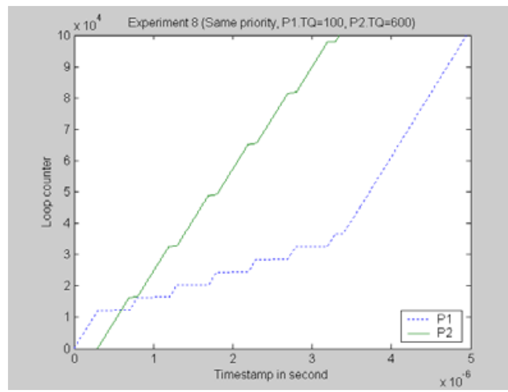

Fig. 6 Experiment 5

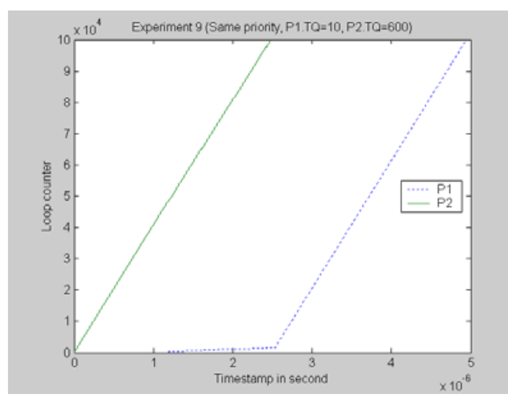

Fig. 7 Experiment 6

Figures 2-7 show the results of the experiments. Timestamp vs loop counter were plotted to illustrate the dynamic change in priorities and Time-Quanta (TQ). The steps in the graphs indicate swapping of process as time-quanta expires. In experiment 1, the two processes were started at same priority and default time quantum of 600ms. One can see from figure 1 there was process swapping at every 600 ms. In all the cases the 1st process gets double the time-quantum, which should be taken into account when developing the scheduler. In experiment 2 the TQ was reduced to 100ms and one can see from figure 3 that the swapping is more frequent. In experiment 3 p2 has a higher priority than p1 so p2 runs till completion and then p1 starts. In experiment 4, 5 and 6 the priority is same but the TQ is different. The intersection in figure 6 is because p1 was started first but p2 has a higher TQ. In figure 7, p1 has a very small TQ and thus the swapping is frequent till p2 is complete after which p1 gets all the process time.

6 Concluding Remarks

This paper presents preliminary experimental results of variation in priority and time-quantum with-in real-time scheduling class for two competing processes in a general purpose Solaris operating system. As seen from the observations, the RT class allocates double time-quantum for the first process in the first instance, which has to be accounted for. The results also indicate that changing priority and time-quanta can be accomplished dynamically and does effect the swapping of processes. In future work we plan to study the effectiveness of our approach within the virtual manufacturing environment and eCommerce applications and extending this approach to other operating systems like windows 2000, Linux etc.

References:

[1] D. A. Lawrence, J. Guan, S. Mehta & L. R.Welch, Adaptive scheduling via feedback control for dynamic real-time systems, *The 20th IEEE international performance, computing and communications conference (IPCCC 2001),* Phoenix, April 2001.

[2] Y. Zhou, L. R. Welch, E. Huh, C. Cavanaugh, C. Alexander, D. A. Lawrence, S. Mehta, Important Considerations for Execution time Analysis of Dynamic, Periodic Processes, *International Parallel & Distributed Processing Symposium*, San Francisco, April 2001.

[3] B. V. Mehta, R. Banga, J. Gunasekera, Virtual Material Processing on the WWW: Cold Rolling, *18th ASME International Computers in Engineering Conference*, Las Vegas, Nevada, Sept. 12-16, 1999.

[4] B. V. Mehta, CyberForm: Web-based Material Processing--Demo, *Proceedings of IPMM 2001*, Vancouver, BC, Canada, July 2001.

[5] B. V. Mehta, Web-based and Distributed Material Processing, *Proceedings of the 7th International Conference on High Performance Computing*, Bangalore, India, Dec. 2000.

[6] B. V. Mehta, J. S. Gunasekera, S. Shivananda, Virtual Material Processing (VMP) on the WEB: Extrusion Die Design, *Journal of Manufacturing Systems*, Vol. 29, No. 3, 1999.

[7] http://www.sun.com/