

A Tool for Isolating Performance in General-Purpose Operating Systems

Valéria Quadros dos Reis
vreis@inf.puc-rio.br
Renato Cerqueira
rcerq@inf.puc-rio.br

Computer Science Department, PUC-Rio

VI Workshop on Middleware for Grid Computing

Contents

- 1 Introduction
- 2 CPUReserve
- 3 Experimental Evaluations
- 4 Usage Cases
- 5 Conclusion
- 6 Future Work

Motivation

Processing Reservation is an issue desired in many scenarios:

- Grid Computing
- On-demand Computing
- Multimedia Applications

Motivation

- General-Purpose Operating Systems do not provide efficient mechanisms for processing reservation.
- Alternatives:
 - Kernel Modification
 - Need for recompiling
 - Virtualization
 - Overhead for instancing and managing the VMs
 - User-level reservation managers
 - Guarantees performance isolation for soft real-time applications
 - The best approach for our needs

DSRT (Dynamic Soft Real Time Scheduler)

- Project was discontinued;
- Communication done by shared memory;
- Manages processing, but not processors;
- Does not treat idle processing;
- No separation between the reservation mechanism and the reservation policies;
- Highly complex code.

CPUReserve

A new tool for processing reservation at user-level. CPUReserve can be used in scenarios of:

- Distributed Computing;
- Grid and On-demand Computing;
- Flexible resource usage;
- Opportunistic Computing;
- Multi-processed architectures.

Execution

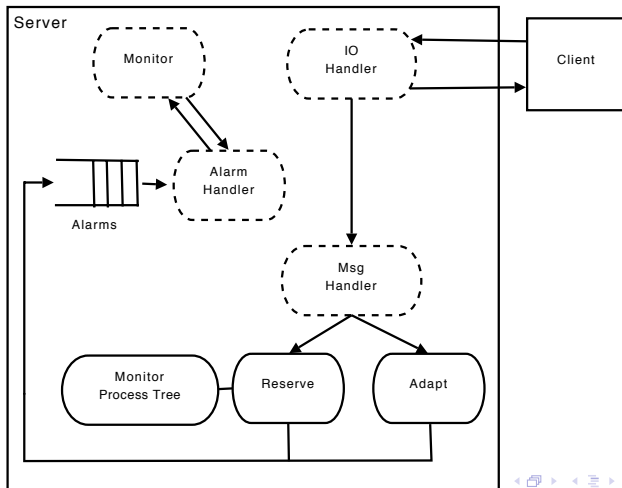
Server

```
./server <port> <procs_bit_mask> <system_limit>
```

Client

```
./client <mac:port><period><slice><cons><exec><params...>  
./adapt <mac:port><period><slice><cons><pid>
```

Architecture



Implementation Details

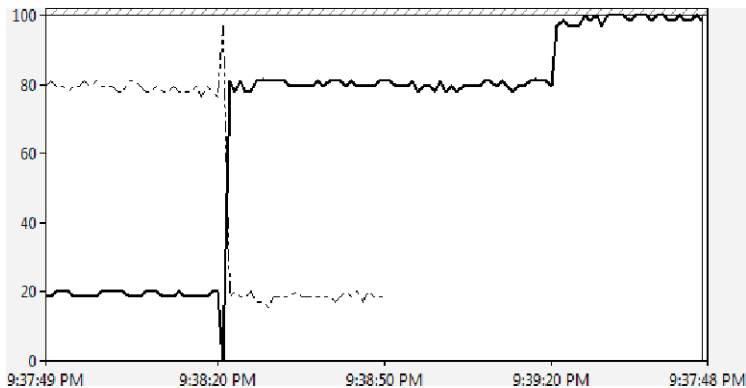
- Use of system calls in order to enforce new scheduling rules;
- Scheduling policies used:
 - SCHED_RT to enforce execution
 - SCHED_OTHER to explore idle processing
- The server process is executed with the highest priority to real-time processes;
- Client processes are executed with the second highest priority to real-time processes;
- Example: An application should be executed with 500 ms out of every 1000 ms.

CPUReserve's Effectiveness

How do applications with and without reservation behave when they are executed concurrently?

- Two busy-waiting applications:
 - App1 being serviced without reservation, that is, in a best-effort way;
 - App2 being initially serviced with 20% reservation in the non-working-conserving mode.

CPUReserve's Effectiveness



CPUReserve's Scalability

Estimate the amount of resources the CPUReserve server consumes to support an increasing number of client processes.

Testbed

Intel Centrino Duo 1,66 GHz, 1 G memory and running kernel 2.6.24

Parametrization

- Server initialised with 2 processors;
- Clients request reservations of 10 ms out of every 1000 ms.

Experimental Evaluations

Scalability Results:

- Server consumes less than 1% of CPU, 18 MB of virtual memory, and 732 KB of resident memory;
- As the number of clients increases:
 - CPU and memory consumption increases linearly;
- It was not possible to make experiments with over 50 clients.

Limitations

- Limited by the Operating System;
- Can interfere on or be interfered by other processes running out of CPUReserve;
- Cannot reserve all the machine's processing capacity.

Usage Cases

- To reserve processing in the execution nodes of CSBase;
- To measure the accuracy of some benchmarks used in CSBase.
 - Is the benchmark result consistent when the workload of the machine being measured varies?
 - Experiments: execution of the benchmark with reservation varying from 10 to 100% of a machine's processing capacity.

CPUReserve and Virtualization

- CPUReserve can be used with different virtualization techniques in order to constrain:
 - The processing of new virtual machines;
 - CPUReserve server triggers a new instance of KVM;
 - The processing of running processes inside a VM.
 - Xen 3.2 initialised with SEDF scheduler;
 - CPUReserve running inside a Xen domain.

Conclusion

- CPUReserve is a light and easy-to-use tool which provides application performance isolation without the need for kernel recompiling or for virtualization overload;
- Tests have shown that the CPUReserve server consumes few resources; consequently, it can support a great number of clients simultaneously;
- CPUReserve can be used in modern computational environments such as multi-institutional opportunistic grids with multi-processed architecture machines.

Future Work

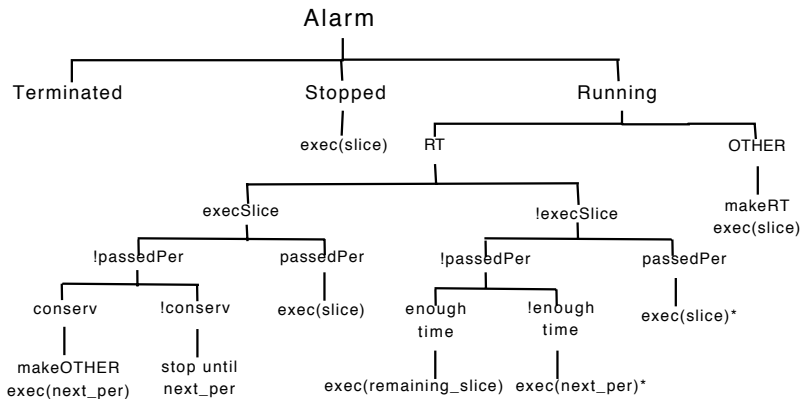
- The study of machine learning techniques to parametrize reservations;
- The insertion of new kinds of resource reservation – disk, memory, and network;
- The development of new policies for resource sharing.

Contacts

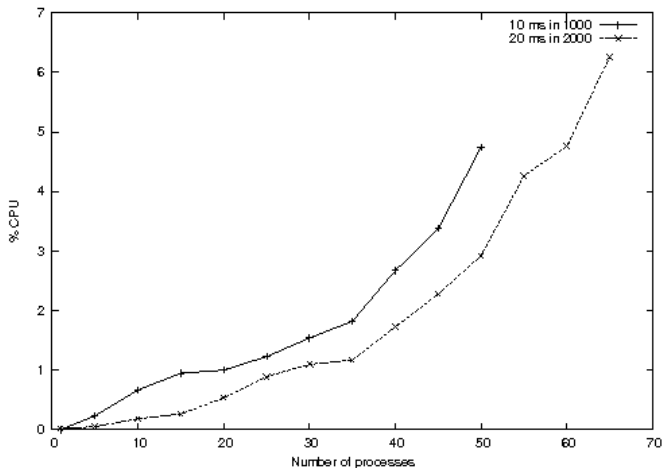
Valéria Quadros dos Reis
vreis@inf.puc-rio.br

Renato Cerqueira
rcerq@inf.puc-rio.br

Decision Tree



Experimental Evaluations



CPUReserve and Xen

