



THE WHITE ROSE GRID e-Science Centre

Exludus MultiCore Optimizer and SGE

Introduction

Multicore systems offering more than two cores per CPU are now increasingly common and the core cost is reducing. However, systems are often still resource constrained as the cost of providing each core with the same amount of memory per core is not changing as rapidly. This means it is important to utilise the memory in a multicore system efficiently, and schedule to maximise resource usage. Coarse-grained methods at the job dispatch level of scheduling are often not optimally efficient. The current challenge is to support multicore systems with scheduling that maximises both core utilisation and memory utilisation to make best use of the capital investment in the equipment.



MultiCore Optimizer (MCOPT) is a product from Exludus Inc. in versions for both Linux and Windows that modifies the existing operating system scheduling to better optimise for resource usage, particularly memory usage.

Simulated Load

The load was based on a synthetic program which:

- Allocates an amount of memory randomly between a lower and upper limit.
- Manipulates this memory performing dummy calculations.
- Runs for a random period of time between an upper and lower limit between 3 and 20 minutes, uniformly distributed, based on the run time on a single processor of the experimental system with no other load.

A number of examples of the synthetic program were run.

The number of simultaneous instances that could be run was varied between 8 and 32 by varying the number of execution slots in Sun Grid Engine provisioned on each node in the system.

Different memory maximums and minimums were used for portions of this load, corresponding to a relatively light memory load and a more challenging load. As the number of slots used increased the total potential memory used increased.

Experimental System

Four Supermicro based compute nodes with 2 quad core Intel Xeon E5335 each, running at 2GHz (total 8 cores per machine) with 16GB RAM running Scientific Linux 4.5, Sun Grid Engine 6.2u4 scheduling system.

Control System

Sun Grid Engine using consumable memory, a common set up within the HPC community. The consumable memory for a particular job was set to be the largest the job might grow to.

Total Memory Footprint

The total memory footprint imposed on the system depends on:

- The memory footprint per job.
- The number of slots provisioned on the system.
- The number of simultaneous jobs of each memory footprint run by the scheduler.

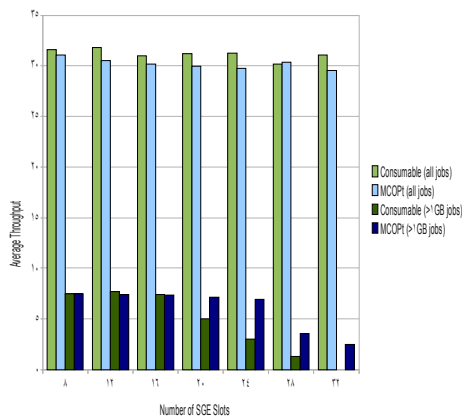
Small Memory Case

The average total memory load demanded (assuming that the demanded job mix could be run) varies from 2.81GB (at 4 slots) to 22.5GB (at 32 slots).

The overall throughput with and without MCOPT remains the same even at high slot counts, but the number of the highest memory jobs completed is greater with MCOPT, although there is some degradation. At 32 slots SGE without MCOPT runs no jobs in the 1GB to 2GB memory range at all. The peak throughput, at 8 slots, is similar for both systems as the system is not memory constrained at this number of slots.



Average Throughput Per Hour (Small Memory Case)



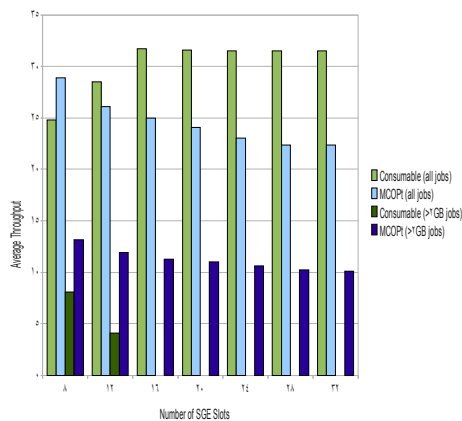
Large Memory Case

The average total memory load demanded (assuming that the demanded job mix could be run) varies from 8.25GB (at 4 slots) to 66GB (at 32 slots).

MCOpt again manages to keep the target throughput for high memory jobs much higher than when using basic SGE with consumable memory, although the overall throughput does dip a little at higher core counts.

At 16 slots and above vanilla SGE runs no jobs in the 2GB to 4GB range at all, whereas MCOpt provides the overall best throughput in this range (at 8 cores) and performance in this memory range remains good out to 32 slots.

Average Throughput Per Hour (Large Memory Case)



Conclusions

- Good overall system utilisation when memory demands are high, balanced

across user demands.

- Peak throughput is not negatively affected.
- Throughput of high memory jobs is improved over the control.
- Users do not need to specify memory usage requirements.
- Additional cost.
- Increase in complexity due to an additional software product installation.

Bibliography

- J. Krallmann, U. Schwiegelshohn, R. Yahyapour, "On the Design and Evaluation of Job Scheduling Algorithms", in IPPS/SPDP'99 Workshop: Job Scheduling Strategies for Parallel Processing, Springer Verlag Lecture Notes in Computer Science LNCS 1659, April 1999, pp 17-42
- Sun Microsystems, "Sun Grid Engine", <http://gridengine.sunsource.net/>
- P M Dew, J G Schmidt, M Thompson, and P Morris, "The White Rose Grid: Practice and Experience" in Proc. UK e-Science All Hands Meeting, Sept. 2nd-4th, 2003
- J. Schmidt, S. Clark, "The White Rose Grid Experience", JSCR, September 2005
- A. Turner, "The White Rose Grid: Experiences and 'Wish List'", Sun Grid Engine Workshop, Regensburg, Germany, October 2003
- 1. B. Marchand, "Multi-Core Processing Advancements via Optimized System Resource Allocation and Capacity Management", Excludus Technologies Inc., Canada, January 2008.

Further Information

Aaron Turner

aaron@cs.york.ac.uk

<http://www.wrg.york.ac.uk>