



University of HUDDERSFIELD

University of Huddersfield Repository

Kureshi, Ibad, Holmes, Violeta, Cooke, D., Allan, R., Liang, Shuo and Gubb, D.

Robust mouldable intelligent scheduling using application benchmarking for elastic environments

Original Citation

Kureshi, Ibad, Holmes, Violeta, Cooke, D., Allan, R., Liang, Shuo and Gubb, D. (2012) Robust mouldable intelligent scheduling using application benchmarking for elastic environments. In: Proceedings of The Queen's Diamond Jubilee Computing and Engineering Annual Researchers' Conference 2012: CEARC'12. University of Huddersfield, Huddersfield, p. 156. ISBN 978-1-86218-106-9

This version is available at <http://eprints.hud.ac.uk/id/eprint/13492/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

<http://eprints.hud.ac.uk/>

ROBUST MOULDABLE INTELLIGENT SCHEDULING USING APPLICATION BENCHMARKING FOR ELASTIC ENVIRONMENTS

I.Kureshi, V. Holmes, D. Cooke, R. Allan, S. Liang, and D. Gubb
University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK

ABSTRACT

In a green IT obsessed world hardware efficiency and usage of computer systems becomes essential. There is a multiplier effect when this is applied to High Performance Computing systems. With an average compute rack consuming between 7 and 25kW it is essential that resources be utilised in the most optimum way possible. Currently the batch schedulers employed to manage these multi-user multi-application environments are nothing more than match making and service level agreement (SLA) enforcing tools. System Administrators strive to get maximum “usage efficiency” from the systems by fine-tuning and restricting queues to get a predictable performance characteristic, e.g. any software package running in queue X will take N number of cores and run for a maximum of T time. These fixed approximations of performance characteristics are used then to schedule queued jobs in the system, in the hope of achieving 100% utilisation. Choosing which queue to place a job in, falls on the user. A savvy user may use trial an error to establish which queue is best suited to his/her needs, but most users will find a queue that gives them results and stick to it – even if they change the model being simulated. This usually leads to a job receiving either an over or under allocation of resources, resulting in either hardware failure or inefficient utilisation of the system. Ideally the system should know how a particular application with a particular dataset would behave when run.

Benchmarking Schemes have historically been used as marketing and administration tools. Some schemes like Standard Performance Evaluation Corporation (SPEC) and Perfect Benchmark used “real” applications with generic datasets to test a systems performance. This way a scientist looking for a cluster computer could ask questions such as “How well will my software run?” rather than “How many FLOPS can I get out of this system?” If adapted to include an API to plug in any software to benchmark and to pass results to other software, these toolkits can be used for purposes other than sales and marketing. If a job scheduler can get access to performance characteristic curves for every application on the system, optimal resource allocation and scheduling/queuing decisions can be made at submit time by the system rather than the user. This would further improve the performance of Mouldable schedulers that currently follow the Downey model. Along with the decision-making regarding resource allocation and scheduling, if the scheduler is able to collect a historic record of simulations by the particular users, then further optimisation is possible. This would lead to better and safer utilisation of the system. Currently AI is used in some decision making in Mouldable schedulers. Given a user inputted variance of resources required the scheduler makes a decision on resource allocation by selecting from the available range. If the user supplied range is incorrect, the scheduler is powerless to adapt, and on a next run cannot learn from previous mistakes or successes.

This project aims to adapt an open-framework benchmarking scheme to feed information to a job scheduler. This job scheduler will also use gathered heuristic data to make scheduling decisions and optimise the resource allocation and the system utilisation. This work will be further expanded to include elastic or even shared resource environments where the scheduler can expand the size of its world based on either financial or SLA driven decisions.

Keywords scheduler, Mouldable benchmarking, elastic cloud, HPC, batch processing, cluster, grid