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


This version is available at http://eprints.hud.ac.uk/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/
In a “Green IT” obsessed world system power efficiency is key! While manufacturers are striving to reduce energy costs incurred by their hardware, IT managers are experimenting with ‘sleep states’ to conserve power on desktops. This applies even more to the world of HPC and other centralised On-Demand computing paradigms. Efforts need to be made to maximise the utilisation of the system, and to ensure that these systems are not sitting idle.

In an attempt to achieve the seemingly irreconcilable goals of maximizing usage and minimizing turnaround time this research aims to adapt existing scheduling tools and benchmarking suites to optimise the usage of a system while delivering a high Quality of Service (QoS) to the end users.

**Background**

- Benchmarking Schemes are generally used as marketing tools.
- Efforts to create “real” application benchmarks have allowed users to quantify their purchases in more relevant terms. But these schemes do not take into account dataset complexities or multi-user workloads.

**Objectives**

This research aims to set up a Resource and Job Management system to:

1. Utilise an adapted real application benchmarking tool which takes user specified datasets and workloads into account when calculating system performance characteristics.
2. Use the modified benchmarking scheme to feed this information back to the system rather than the salesman.
3. Contain an altered off-the-shelf scheduler which will use the benchmarking information to better allocate resource to a particular job, instead of information provided by the user. The jobs can then be moulded to fit space in the system with the aim of maximising the utilisation and improving TaT.
4. Gather heuristic data from previous jobs and use it to fine tune resource allocations.
5. Cope with shared and elastic environments and determine when it is better to scale beyond the fixed resources.

**Case Study**

The usage of the University of Huddersfield’s Eridani cluster has been analysed to find:

- With out resource and time dependant queues the cluster spends a large amount of time sitting idle waiting for resources to become available for wider jobs. (see Fig 1)
- With optimised queues usage efficiency greatly improves (Fig 2b) but a large back log is created (Fig 2a) and there is difficult to estimate time of completion.

The proposed system will avoid both situations.

- Resource Management Tools (RMT) are nothing more than match making and SLA enforcing tools.
- These RMTs require a significant effort to configure, and continuous on-going management to get maximum consumption and minimum turn-around-time (TaT).
- The RMTs rely on user prescribed parameters and these can lead to the system being utilised inefficiently.

**Fig 1:** An overview of the utilisation of the Eridani cluster with out queue optimisation

**Fig 2a:** A look at the backlog in the queues on the Eridani cluster

**Fig 2b:** With multiple queues and fine tuning system utilisation reaches near perfect levels