HPC FOR HUDDERSFIELD

ROBUST MOULDABLE INTELLIGENT SCHEDULING USING APPLICATION BENCHMARKING FOR ELASTIC ENVIORNMENTS

University of HUDDERSFIELD
inspiring tomorrow's professionals

I. Kureshi, V. Holmes, D. Cooke, R. Allan, S. Liang, D. Gubb and Y. James
Computing and Engineering

Abstract

In a "Green IT" obsessed world system power efficiency in 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.*

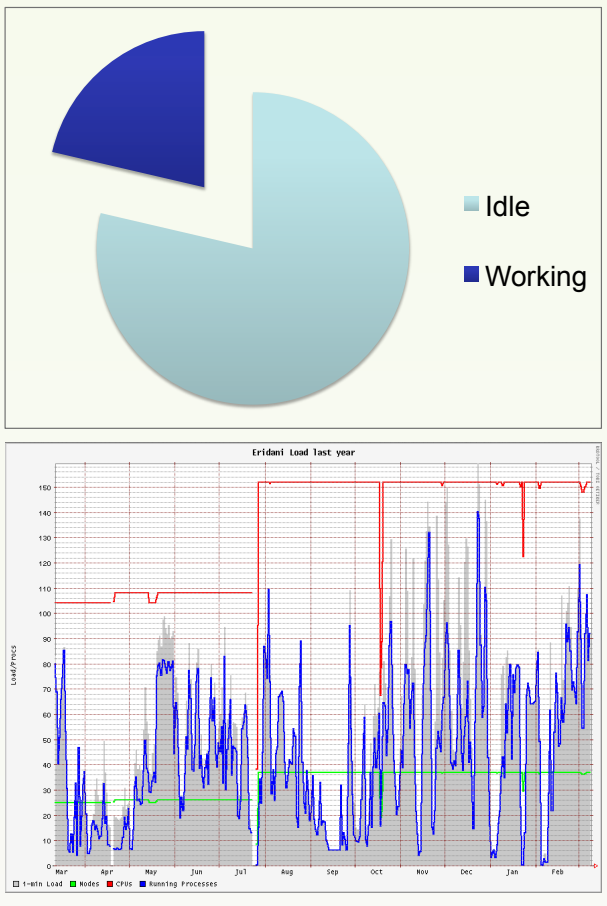

Fig 1: An overview of the utilisation of the Eridani cluster with out queue optimisation

- *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.*

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.

eridani.qgg.hud.ac.uk:

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	S	Elap Time
35609.eridani.qgg.hu	u0775731	paraul	d057mml30m	6825	6	1	--	10000	R	267:1
35610.eridani.qgg.hu	u0957831	serialul	producAREN	16473	1	4	--	336:0	R	70:22
35611.eridani.qgg.hu	u0957831	serialul	producPYY.	10885	1	4	--	336:0	R	67:00
35612.eridani.qgg.hu	u0957831	serialul	producSecA	12834	1	4	--	336:0	R	67:00
35613.eridani.qgg.hu	u0957831	serialul	producWWE.	12090	1	4	--	336:0	R	66:58
35712.eridani.qgg.hu	u0652238	parastd	dlpoly	15979	2	1	--	10000	R	97:22
35713.eridani.qgg.hu	u0652238	parastd	dlpoly	10831	2	1	--	10000	R	96:27
35714.eridani.qgg.hu	u0652238	parastd	dlpoly	29684	2	1	--	10000	R	92:47
35715.eridani.qgg.hu	u0652238	parastd	dlpoly	14024	2	1	--	10000	R	92:46
35717.eridani.qgg.hu	u0652238	parastd	dlpoly	20618	2	1	--	10000	R	92:40
35718.eridani.qgg.hu	u0652238	parastd	dlpoly	17286	2	1	--	10000	R	92:39
35719.eridani.qgg.hu	u0652238	parastd	dlpoly	17381	2	1	--	10000	R	92:37
35720.eridani.qgg.hu	u0652238	parastd	dlpoly	12990	2	1	--	10000	R	92:37
35721.eridani.qgg.hu	u0652238	parastd	dlpoly	13048	2	1	--	10000	R	92:35
35722.eridani.qgg.hu	u0652238	parastd	dlpoly	20898	2	1	--	10000	R	92:32
35723.eridani.qgg.hu	u0652238	parastd	dlpoly	14153	2	1	--	10000	R	92:32
35724.eridani.qgg.hu	u0652238	parastd	dlpoly	16928	2	1	--	10000	R	92:30
35725.eridani.qgg.hu	u0652238	parastd	dlpoly	18609	2	1	--	10000	R	88:14
35726.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35727.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35728.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35729.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35730.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35731.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35732.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35733.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35734.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35735.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35736.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35737.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35738.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35739.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35740.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35741.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35742.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35743.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35744.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35745.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35746.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35747.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35748.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--
35749.eridani.qgg.hu	u0652238	parastd	dlpoly	--	2	1	--	10000	Q	--

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

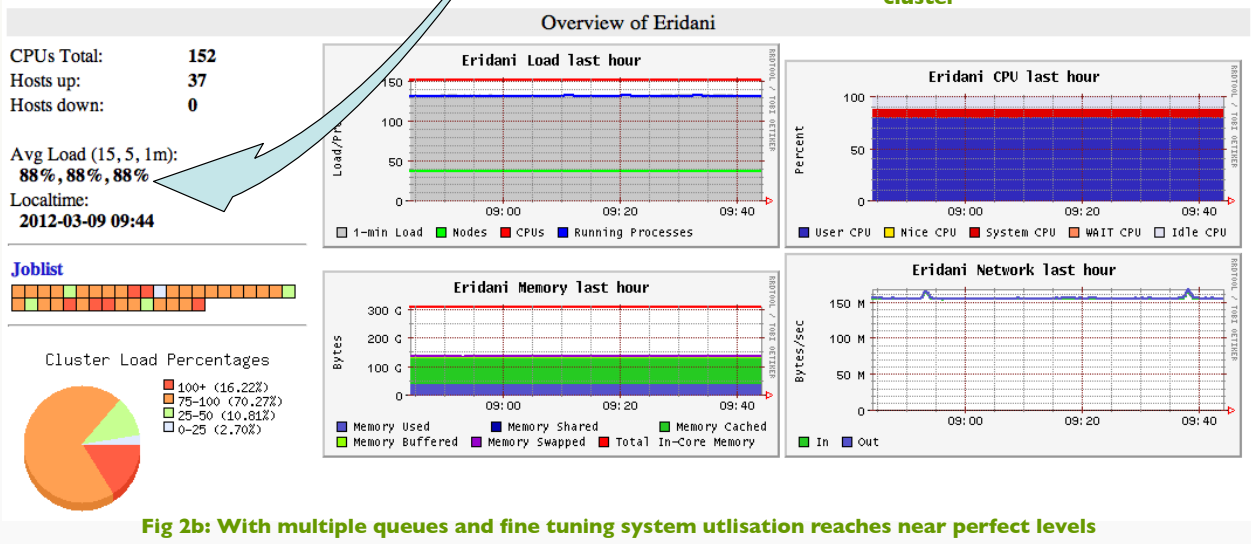

Fig 2b: With multiple queues and fine tuning system utlisation reaches near perfect levels