Intelligent Power Management over large Clusters

Clive Gerrard[1] Paul Haldane[1] Sindre Hamlander[2] Stephen McGough[1]
Paul Robinson[2] Dave Sharples[1] Dan Swan[1] Paul Watson[1] Stuart Wheater[2]

[1]Newcastle University, Newcastle upon Tyne, NE1 7RU, UK
[2]Arjuna Technologies, Herschel Building, Newcastle upon Tyne, NE1 7RU, UK
Email:{c.gerrard, paul.haldane, stephen.mcgough, dave.sharples, d.c.swan, paul.watson}@newcastle.ac.uk,
{sindre.hamlander, paul.robinson, stuart.wheater}@arjuna.com

Abstract

There is a growing tension, within large organisations such a Universities, between the desire to perform large amounts of computational processing and the desire to reduce power consumption by switching off computers. Through careful management of computing resources it is possible to maximise computer usage whilst minimising power consumption though this can be costly in terms of human effort. We present here our work with the Agility Cloud Computing platform to provide intelligent control over a University-wide Condor system which works to reduce power consumption without adversely affecting the users of Condor. This system also provides auditing of the power usage, which can be used to determine the power efficiency of the Condor system.

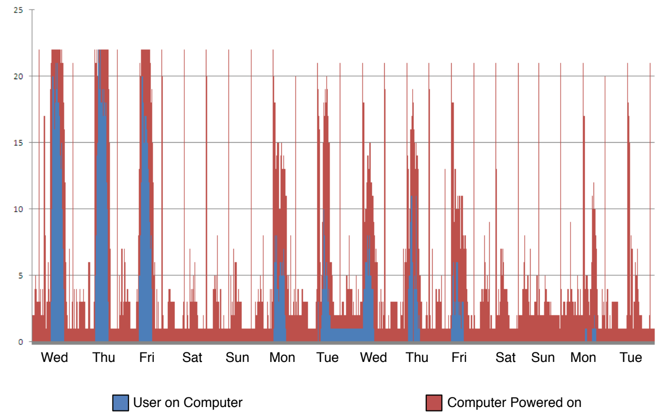

Figure 1: Example cluster usage

1 Introduction

There is currently a clash between two groups of society over the use of computing facilities within large organisations such as Universities. On the one hand you have those who have large computational tasks to perform, preferably within a confined time scale. Whilst on the other you have the green and sustainability members of society who seek to reduce power consumption, and hence lower carbon output, by turning computers off. Both groups can provide persuasive arguments as to why their needs should be met, thus a compromise is required to best satisfy both groups.

Figure 1 reflects the use of a single open access cluster within the University running Windows 7 for the last three days of an academic term followed by eleven days into the holiday period. The blue area indicates computers which are being used by users whilst the red area shows computers which are powered up. The Windows 7 computers are configured to power down a short time after a user logs out, hence the red bars being higher than the blue bars. Some of the red area indicates system maintenance. At present Condor is not run on our Windows 7 cluster just the Windows XP clusters. However, as can be seen from the graph the number of computers awake and able to accept jobs at the weekend or out of term time is low.

An ideal solution would be that computational resources were only powered up when they were needed and immediately powered down when not. This, however, is difficult to achieve given the uncertainty of when work will arrive. This situation is further compounded by the fact that the major computational resource at Newcastle is also the student computer clusters with jobs being submitted through Condor [3] (a distributed job submission and monitoring service). Here both students sitting down at computers and jobs sent through Condor can keep a particular computer active.

Like many universities Newcastle currently supports course-grained power-managment of its computers. Computers are set to power down if they are idle for a given period of time. Where idle is defined as no active user and no Condor jobs running on the computer. This is good in terms of power management though can lead to resource starvation for Condor if work arrives after the majority of the computers have powered down.

An administrator could simply rectify this situation by noticing that there was work awaiting execution and powering up enough computers to deal with the backlog. This could also allow for the tailoring of this procedure to account for different classes of Condor users with different priorities on their work. Users could be grouped into those that are allowed to power up computers, as perhaps they are willing to pay for extra electricity incurred or their work is seen as significant enough by the University, and those users who are only allowed to use free computational cycles on computers that are still turned on. However, providing this level of service would be costly in terms of human effort. We are developing an intelligent service which can take the role of the administrator and apply appropriate policies over the collection of Condor enabled computers. This will exploit the new features of the Condor system such as persistent ClassAds and Rooster. Our intelligent service is based around the Arjuna Agility Cloud Computing Platform.

2 Arjuna Agility Cloud Computing platform

Agility Cloud Computing platform [1] is a solution provided by Arjuna Technologies Ltd. designed to define, implement and enforce Service Level Agreements (SLA) between different services on the Internet. These SLA's are defined by sets of policies defined in each Agility partner taking part in the negotiation. The policies can be influenced by environmental conditions such as the type of user seeking the SLA or the time at which the SLA will be active. This allows Agility to act in an intelligent manner based on circumstances. The policy rules within Agility are provided in a programmatic format with the ability to add and remove rules easily. Thus the system can be easily configured and modified as our understanding of the way the system should react improves.

3 Power management information

In order to sensibly select between computers based on the electrical power consumed while performing a piece of work it is necessary to know something about the power consumption of the individual computers available. However, to model this precisely is a difficult task as each piece of software will place different loads on various parts a computer than others. A computationally intense job may place the processor under full load whilst a data intensive tasks may use the local hard disk (or network) more than the processor. These will each give different power consumption profiles. This is further compounded through the new multi-core processors which have significantly different electrical power profiles depending on the number of active cores.

At present we are working with the assumption that an active piece of work will produce a constant power load on the computer during execution. We are therefore interested in the following information: The number of flops per second provided per watt of electricity on each core within a computer and the Power Usage Effectiveness (PUE) of the computer within its environment. Computational power (flops) can be assessed in various different ways and some ways are more appropriate than others depending on what you are using it for. The flops measure is pretty ambiguous - factors such as cache coherence, precision etc. are also important. We are evaluating benchmarking tools (such as LINPACK [2]) to measure the performance of our computers.

PUE is the ratio of total amount of power used by a computer facility to the power delivered to computing equipment. In a data centre this gives a value greater than one with the added energy consumption normally coming from cooling. Within a cluster room the PUE value can be less than one with the heat generated offsetting the heating required for the room. This information needs to be pre-determined through experimentation and added to the ClassAd for the computer so that Condor can use it to make decisions on which computer is best to use.

4 Overall Architecture

The Condor system provides much of the core architecture and is capable of selecting computational resources based on priorities specified by the user. The latest releases of Condor comes with Rooster and persistent ClassAds. Persistent ClassAds allows Condor to retain information on computers which have powered down whilst Rooster allows Condor to wake up such a powered down computer. Our aim here is not to re-develop these services available through Condor, but rather to augment them to provide intelligent management of the Condor system.

In Figure 2 we illustrate our architecture. The user is provided with equivalent versions of the normal Condor commands thus making our additions transparent. The only changes they can optionally make is to indicate within their Condor submit script, what the priority of the Job is using the Condor Rank element. The current Policy realises high priority by powering up new machines.

The architecture is split into two domains, the Client domain and the Condor domain. The Client domain represents the End User and is responsible for submitting Jobs. The Condor domain represents the Condor Pool and is responsible for processing the jobs. An Agility server is located at each domain and is used to form a SLA between the two parties. Policy code is situated within each Agility server which monitors and reacts to changes in the SLA.

The Agility policies can also be used to modify the request coming from the user. For example, policy in the Condor domain can realise the low priority by re-writing the Job requirements to ensure only currently powered on machines are selected. Furthermore, Policy in the Condor domain can check that the user is allowed to request high priority and demote the job to low priority if they are not. The Agility policy will also specify a ranking of potential matches within Condor (using the Rank element) for the user's job based around the anticipated power usage, thus favouring more efficient computers. This is

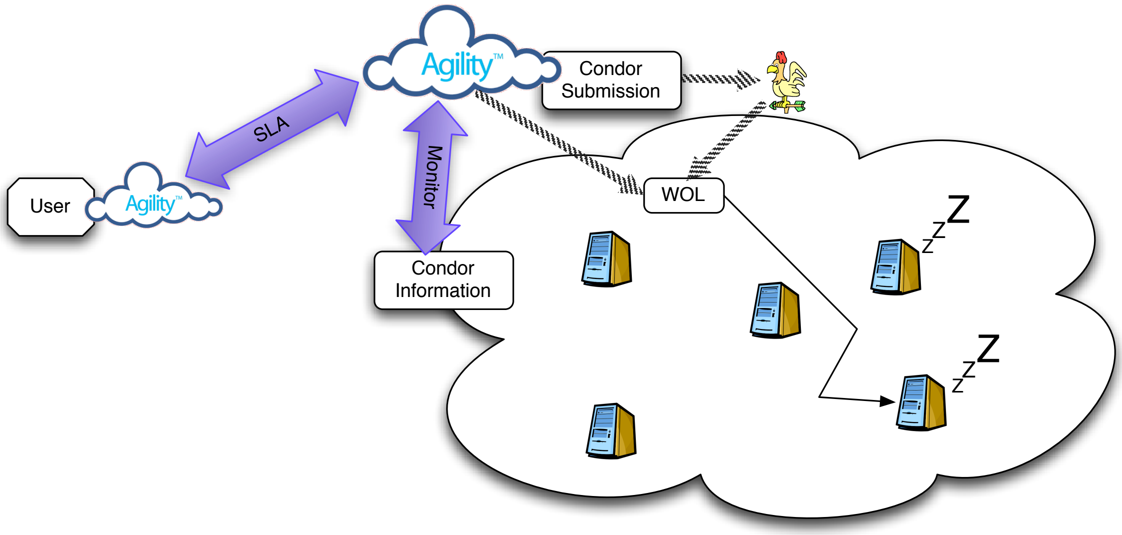

Figure 2: The Condor Agility Architecture

defined in terms of the information in section 3:

$$Rank = kflops/(PUE * watts)$$

If the user has already specified a Rank element within their submission script this needs to be merged in with our Rank equation. This does provide a small restriction to the user as they need to make their Rank equation produce values within a range which will neither obliterate the equation above nor be obliterated by it.

Once an agreement has been reached the submission service will submit the work to Condor and return job information to the user. The user is then able to monitor their jobs in the normal manner. At the same time the Agility policy in the Condor domain will monitor the work within the Condor system and invoke policies as appropriate over the work. Current policies include monitoring and flagging up rogue jobs which users try to submit without going through the correct submission channels and monitoring of waiting work within the Condor system. In the situation where a large amount of work is awaiting execution, though none have the ability to wake up a computer to start execution, Agility policy can take the decision to wake up computers to reduce the backlog. We are seeking to develop further policies to greater improve the ability of our Condor system to better process users work while keeping power consumption to a minimum.

5 Auditing

We now have the ability to derive an estimate for the level of power consumption within a Condor enabled computer. This allows us to produce an estimate for the amount of power used through a Condor submission. This can be defined as:

$$Power\_used = PUE * watts * (end\_time - start\_time)$$

This equation assumes that the computation was successful on the first computer that it ran on. However, the execution may have run unsuccessfully on other computers or it may have partly completed on a different computer before being check-pointed and moved to the computer where it finished (this case only applies to non-Windows jobs). In the latter case the power used should reflect the sum of all these individual computations. In the former case it is an open question as to whether the user should be accounted for these failed runs. Especially as this figure may end up counting against them in some financial sense. Information can then be grouped on a per user, per group, per school or per university level for further analysis.

6 Conclusion

In this work we have shown how a Condor cluster can be augmented with an intelligent service which works to minimise power consumption whilst trying to have minimum impact on the users of the system. The system is also capable of auditing the power consumption which is a powerful tool for showing the merits of using a job submission system like Condor.

References

[1] Arjuna Technologies Ltd. Arjuna Agility. `http://www.arjuna.com/agility`.

[2] Jack J. Dongarra. *The LINPACK Benchmark: An explanation*. LNCS 297.

[3] Condor Team. Condor Project Homepage. `http://www.cs.wisc.edu/condor`.