

Cardiff University's Condor Pool: Background, Case Studies, and fEC

James Osborne¹, Alex Hardisty²

1. Information Services, Cardiff University, 50 Park Place, Cardiff, CF10 3AT, Wales UK
2. Welsh e-Science Centre, Cardiff University, 5 The Parade, Cardiff, CF24 3AA, Wales UK

Abstract

Cardiff University's Condor pool aims to cater for the high throughput computing needs of a wide range of users based in various schools across campus. The paper begins by discussing the background of Cardiff University's Condor pool. The paper then presents a selection of case studies and outlines our strategy for Condorising applications using submit script generators. Finally the paper presents the findings of a fEC exercise and outlines our policy for attributing directly allocated costs to a particular research project.

1. Background

Cardiff University's Condor pool is the third largest Condor pool in the UK with an average of around 800 execute nodes providing 500 gigaflops of computing power on-demand to our researchers, giving them a significant advantage over their competitors in other universities whilst at the same time saving between a quarter and a third of a million pounds on dedicated high throughput computing resources [1].

Cardiff University's Condor pool began as a pilot service back in April 2004 as an initiative sponsored by Information Services and the Welsh e-Science Centre after Dr Hugh Beedie of Information Services realised the machines they were using to provide an open access workstation service to support learning and teaching were being under utilised.

In April 2004 Condor 6.6.2 was deployed on a number of open access workstations running Windows NT using ZENworks application management tools. Condor 6.6.2 was configured and tested on a dedicated test cluster by Jon Giddy and Chris Tottle prior to deployment.

One of the applications Condorised during the pilot phase involved a colleague in the School of Biosciences, Dr Kevin Ashelford. His application used the university's Condor pool to perform a distributed search to identify corrupt records in a DNA database. Kevin would have had to run his application constantly meaning the elapsed wall-clock time would have been of the order of 28 months. Using Condor Kevin ran his application in parallel reducing the time taken to 18 days.

In April 2006 Condor 6.6.11 was deployed on a number of open access and schools donated workstations running Windows XP using

ZENworks application management tools. Condor 6.6.11 was also configured and tested on the dedicated test cluster, this time by Dr James Osborne prior to deployment. James was responsible for tightening up the security policy and defining an additional job control policy for long running jobs.

All applications that run on the open access workstations are tested on the dedicated test cluster prior to allowing them to run on the production system and long before we allow users to run their own applications allowing us to reduce the risk of applications causing problems on the production system.

2. Case Studies

The three most significant research projects in terms of their potential or actual consumption of Condor hours are presented in this section.

2.1 The School of Computer Science in Collaboration with Velindre Cancer Centre

BEAMnrc and DOSXYZnrc are applications that use Monte Carlo simulation to perform radiotherapy dose calculation [2].

Before Condorising these applications our colleagues in the School of Computer Science, Prof David Walker, and at the Velindre Cancer Centre, Dr Geraint Lewis and Mary Chin, were already investigating grid computing methods to reduce the time taken to simulate a single radiotherapy dose calculation.

Mary would have had to run BEAMnrc and DOSXYZnrc long enough to simulate 45 to 60 million X-ray events per radiotherapy dose calculation, each dose calculation taking between 2,430 and 3,240 hours to complete. Mary would have had to run BEAMnrc and DOSXYZnrc constantly meaning the elapsed

wall-clock time to simulate a single dose calculation would have been of the order of 3.5 to 4.5 months. In the life of a cancer patient months may be too long to wait hence the investigation of grid computing methods.

Using Condor Mary can run BEAMnrc and DOSXYZnrc in parallel reducing the time taken to simulate a single radiotherapy dose calculation to 36 hours.

During the first quarter of 2006 Mary used a total of 2,771 Condor hours which equates to 8% of the total Condor hours consumed. During the second quarter she used a total of 5,474 Condor hours which equates to 6% of the total Condor hours consumed. If we were charging Mary in line with our external fEC policy we would have charged her £55 in the first quarter and £109 in the second.

The Condorisation of BEAMnrc and DOSXYZnrc was performed by Mary Chin who is based at Velindre Cancer Centre.

2.2 The School of Biosciences

Structure is an application that uses Bayesian classification to assign individual genetic sequences to appropriate populations [3,4].

Before Condorising Structure our colleague in the School of Biosciences, Tim Bray, would typically run Structure on the same individual genetic sequence a total of twenty times, each run taking 12 hours to complete. Tim used to run Structure overnight meaning the elapsed wall-clock time to assign an individual genetic sequence to an appropriate population used to be of the order of 4 weeks.

Using Condor Tim can run Structure in parallel reducing the time taken to assign an individual genetic sequence to an appropriate population to 12 hours. In addition Tim can run Structure on multiple genetic sequences at the same time. Tim said "Condor has allowed submission of jobs in large numbers that will run in parallel as long as there are free machines available." Tim also said "In this way an analysis that may take a day for a single sequence has the potential to be finished with twenty repeats in the same period."

During the first quarter of 2006 Tim used a total of 5,915 Condor hours which equates to 5% of the total Condor hours consumed. During the second quarter he used a total of 15,221 Condor hours which equates to 17% of the total Condor hours consumed. If we were charging Tim in line with our internal fEC policy we would have charged him £89 in the first quarter and £228 in the second.

The Condorisation of Structure was performed by Steffan Adams who is based in the School of Biosciences. Steffan is responsible for a small Linux based Condor pool used by the school. Steffan ported his solution to the university's Windows based Condor pool because Tim was saturating the school's Condor pool.

2.3 The School of Optometry and Vision Sciences

Dammin and Gasbor are applications that use ab initio methods to build models of proteins using simulated annealing [5,6].

Before Condorising these applications our colleagues in the School of Optometry and Vision Sciences, Prof Tim Wess and Donna Lammie, would typically run Dammin or Gasbor on the ab initio data of a particular protein a total of twenty times, each run taking either 18 or 90 minutes to complete using Dammin or Gasbor respectively. Donna used to run Dammin or Gasbor during the day meaning the elapsed wall-clock time to build a model of a protein used to be of the order of either 1 or 4 days using Dammin or Gasbor respectively.

Using Condor Donna can run Dammin or Gasbor in parallel reducing the time taken to build a model of a protein using Gasbor to 2 hours. In addition Donna can run Dammin or Gasbor on multiple ab initio datasets at the same time. Donna said "Condor has proved invaluable to our research since the work is completed rapidly and efficiently."

During the first quarter of 2006 Donna used a total of 1,876 Condor hours which equates to 2% of the total Condor hours consumed. During the second quarter she used a total of 25,566 Condor hours which equates to 28% of the total Condor hours consumed. If we were charging Donna in line with our internal fEC policy we would have charged her £28 in the first quarter and £383 in the second.

The Condorisation of Dammin and Gasbor were performed by James Osborne who is based in Information Services. James is responsible for the university's Condor pool. James Condorised Dammin and Gasbor by writing a program to generate Condor submit scripts. Developing the submit script generator was done in such a way so that the generator could be quickly adapted to support other applications in the future. To date the generator has been adapted to support five additional applications. We briefly discuss the submit script generator in the next subsection.

2.4 Condorising Applications Using Submit Script Generators

The submit script generator allows us to rapidly Condorise applications concerned with data processing. We will use the Dammin and Gasbor applications to aid our discussions.

Dammin and Gasbor can be used to build models of proteins using simulated annealing. The input files for Dammin and Gasbor are produced by an application called Gnom which is used to filter the data captured by the experimental apparatus which bombards a sample of the protein under investigation with X-rays.

Dammin and Gasbor were originally designed as interactive applications asking the user a number of questions before processing the output from Gnom and generating a model of the protein that can be visualized. Dammin and Gasbor can also operate in batch processing mode by providing an answer file containing the answers the user would otherwise have had to supply during interactive mode.

When running Dammin, Donna would typically accept all the default answers to the questions Dammin asked except the name of the Gnom file, the name of the log file (used to log any errors), and the name of the project identifier (used to name the output file).

Using the submit script generator Donna does not have to answer a single question, the generator simply looks in the input directory for the Gnom file and generates twenty answer files containing the name of the Gnom file, the name of the log file in the format file0.log to file19.log, and the name of the project identifier in the format file0 to file19. The submit script generator then generates twenty batch files calling Dammin with one of the twenty answer files.

The submit script generator then builds the Condor submit script itself which in turn transfers copies of the Dammin binary, the Gnom file, the answer file, and the batch file to each execute node and tells Condor where to transfer the output files to on the submit node. The script generator is also capable of processing multiple Gnom files in the input directory.

When running Gasbor, Donna would typically accept all the default answers to the questions Gasbor asked except the name of the Gnom file, the name of the log file, the project identifier, and the number of residues in the asymmetric part.

Using the submit script generator Donna only has to answer one question, the number of

residues in the asymmetric part. The Gasbor submit script generator works in the same way as the Dammin submit script generator.

3. Full Economic Costing

A full economic costing of Cardiff University's Condor pool was conducted in line with various higher education funding council's requirements for full economic costing of research projects.

3.1 Terminology

The full economic costing (fEC) model divides costs into two types, indirect costs and direct costs. The model further divides direct costs into two subtypes, directly incurred costs and directly allocated costs [7].

Indirect costs are those costs incurred that are not directly related to any one project but costs that are necessary to support a given project. Directly incurred costs are those costs incurred for equipment or services related to a single project. Directly allocated costs are those costs incurred for equipment or services shared by a number of projects.

3.2 Indirect Costs

The indirect cost of the university's Condor pool is the cost of equipment, power, and staff required to provide execute nodes, submit nodes, and networking which are already provided by Information Services as part of the university's overall costs in providing an open access workstation service for students to support learning and teaching.

The cost of the open access workstation service includes: initial purchase of machines with three-year warranties updated on a four-year rolling cycle, the cost of power consumed, the cost of support staff required to maintain and update the machines, and an element of cost for networking and central data storage.

The indirect cost of the university's Condor pool is recovered via the indirect charge in £ per FTE per year added to the full economic costing of every research project.

3.3 Direct Costs

The direct cost of the university's Condor pool is the cost of equipment, power, and staff required to provide the central manager and Condor support services beyond those provided by Information Services as part of the open access workstation service.

The direct equipment cost is the cost of the central manager which includes: initial purchase of the machine with a three-year warranty

updated on a four-year rolling cycle, as well as an annual racking fee that includes: rack space, uninterruptible power supply, air conditioning, and network connection. The cost of the central manager is £1,560 pa.

The direct power cost is significantly less than the cost of running the pool at maximum capacity which, based on current market prices, would be £49,640 pa.

The direct staff cost is currently £35,000 pa which includes one member of full-time staff and an element of cost for administration.

Currently the direct cost of the university's Condor pool is met by Information Services and the Welsh e-Science Centre. In the future the direct cost of the university's Condor pool will become a directly allocated cost shared between the research projects using the Condor service.

We briefly discuss how to attribute directly allocated costs to a particular research project in the next subsection.

3.4 Attributing Directly Allocated Costs to Particular Research Projects

Directly allocated costs can be attributed to a particular research project using the accounting information collected by the central manager.

A value for directly allocated costs can be calculated by dividing the maximum directly allocated costs of the Condor pool by the maximum number of Condor hours available which gives us a cost of 1.5 pence per Condor hour.

Calculating the directly allocated equipment and staff costs is trivial, however calculating the power cost is a little more involved. To do this we measured the power consumed by a number of different open access workstations of various specifications.

We measured the power consumption of each sample machine in three different states for a total of fifteen minutes. A sample machine in the first state, IDLE, was simply turned on with nobody logged in. A sample machine in the second state, MAX CPU, was running a Condor job that called the CPUSoak program provided with the Condor toolkit. A sample machine in the third state, MAX DISK, was busy copying and deleting an ISO file over and over.

We then calculated the combined additional power consumption of each sample machine whilst running at MAX CPU and MAX DISK.

We then used census information to calculate the cost of running the pool for an hour and divided that by the number of machines in the pool giving us an average cost of 0.5 pence per Condor hour.

We recommend that future research projects that wish to use the university's Condor pool try to estimate, with our assistance, the number of Condor hours needed to satisfy their computational requirements in order that directly allocated costs can be included and subsequently recovered via the fEC of their own research projects.

Our internal charging policy is to charge 1.5 pence per Condor hour whereas our external charging policy is to charge 2.0 pence per Condor hour.

Acknowledgements

We are grateful to those users who provided case studies: Steffan Adams, Kevin Ashelford, Tim Bray, Mary Chin, Donna Lammie, Geraint Lewis, David Walker, and Tim Wess. We are also grateful to Information Services and the Welsh e-Science Centre for funding this research.

References

- [1] M. Litzkow, M. Livny, and M. Mutka. Condor – A Hunter of Idle Workstations. In *Proceedings of 8th IEEE International Conference on Distributed Computing Systems 1988 (ICDCS8)*, pages 104-111, San Jose, California, USA, June 1988. IEEE.
- [2] M. Chin, G. Lewis, and J. Giddy. Implementation of BEAMrc Monte Carlo Simulations on the Grid. In *Proceedings of 14th International Conference on the Use of Computers in Radiation Therapy 2004 (ICCR2004)*, Seoul, Korea, May 2004.
- [3] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of Population Structure Using Multilocus Genotype Data. *Genetics*, 155(2):945-959, 2000.
- [4] G. Evanno, S. Regnaut, and J. Goudet. Detecting the Number of Clusters of Individuals Using the Software Structure: a Simulation Study. *Molecular Ecology*, 14:2611-2620, 2005.
- [5] D. I. Svergun. Restoring Low Resolution Structure of Biological Macromolecules from Solution Scattering Using Simulated Annealing. *Biophysical Journal*, 76:2879-2886, 1999.
- [6] D. I. Svergun, M. V. Petoukhov, and M. H. J. Koch. Determination of Domain Structure of Proteins from X-Ray Solution Scattering. *Biophysical Journal*, 80:2946-2953, 2001.
- [7] The Joint Costing and Pricing Steering Group. Internet. <http://www.jcpsg.ac.uk/>, July 2006.