



## **Griffith University Computational Resource Framework.**

### Author

Lewis, Andrew, Perrett, Heidi Olwen, Miotto, Amanda, Hettenhausen, Jan

### Published

2013

### Conference Title

eResearch Australasia 2013

### Rights statement

© The Author(s) 2013. The attached file is reproduced here in accordance with the copyright policy of the publisher. For information about this conference please refer to the conference's website or contact the authors.

### Downloaded from

<http://hdl.handle.net/10072/59330>

### Link to published version

<http://conference.eresearch.edu.au/>

### Griffith Research Online

<https://research-repository.griffith.edu.au>

# Griffith University Computational Resource Framework

**Andrew Lewis<sup>1a</sup>, Heidi Perrett<sup>1b</sup>, Amanda Miotto<sup>1c</sup>, Jan Hettenhausen<sup>1d</sup>**

<sup>1</sup> Griffith University, Brisbane, Australia, {a.lewis, h.perrett, a.miotto, j.hettenhausen}@griffith.edu.au

## BACKGROUND

In today's technology-driven environment, it is increasingly common for research methods to require the use High Performance Computing (HPC) services. Discoveries in areas such as genomics, protein folding, computational chemistry, criminology and statistical analysis are using larger and more complex calculations than ever before. Due to the complexity and size of these jobs, the resources needed to run them efficiently exceeds the capacity of a traditional desktop. In many cases, a single job submission can take months to process and will produce output files in excess of a gigabyte. Several state and national organisations provide services specifically for large-scale processing such as Queensland Facility of Advanced Bioinformatics (QFAB), Queensland Cyber Infrastructure Foundation(QCIF), National eResearch Collaboration Tools and Resources (NeCTAR) and Research Data Storage Infrastructure (RDSI).

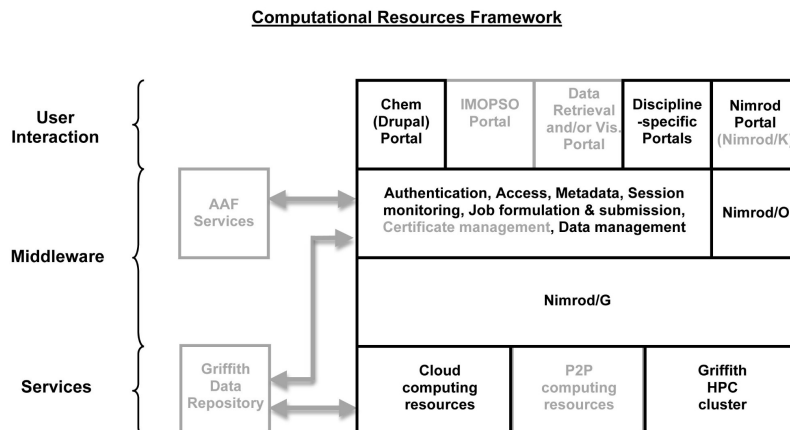
Although there is demand to use HPC resources, a researcher often encounters a steep learning curve to be able to use them and has to use interfaces aimed at people with a technical background. This combination can act as a roadblock and limits the use and uptake of these services. This results in the majority of HPC resources being used by a minority of researchers. Our Computational Resource Framework provides researchers with an intuitive, user-friendly system that allows effortless use of HPC systems via a simple, web-based interface. Researchers can now access the available pool of computing power and process their jobs in a more accessible, efficient, secure manner.

## COMPUTATIONAL RESOURCE FRAMEWORK

The Computational Resource Framework (CRF) is a solution that provides seamless, user-friendly access, submission and file storage for multiple HPC and cloud resources. The front-end is composed of several submission portals. As with the resources, the portals have been developed to be flexible and extendible. These flexible components are held together by a common middleware. This is composed of NIMROD/G and a web service to communicate between authenticated web portal's and NIMROD/G. NIMROD/G is a Grid and Cloud aware version of the Nimrod distributed computing middleware (1). It manages the job submissions and communicates with the resources a user can submit to. Finally, all of the submission and output files are stored using the integrated Rule-Oriented Data-management System (iRODS), a community-driven, data grid software solution (2). The web portal, middleware (web service, NIMROD/G, iRODS) and resources (HPC and cloud) are all hosted on separate servers and all components of this solution are open source, including the CRF code.

The front-end submission portal has been implemented to use several job submission methods. The primary submission method is via a web interface that contains discipline-specific portals. We opted for a web portal as it provides fast, easy access (anywhere, anytime), no software has to be installed (OS independent) and it is a familiar delivery method (minimises user learning curve). These portals have been developed to either suit a researcher's custom submission needs, or

to accommodate the broader community with a generic submission portal for tools such as Matlab and R.



The web interface uses LDAP authentication and taxonomy-based roles to restrict login and portal access for each user. Once a user has logged in they can select a portal for job submission or view the list of jobs they have already submitted. Each portal contains one or more links to the associated webforms that accept job submissions of a specific type. The system currently supports job submissions in Matlab, R and NAMD, with development in Octave, AMBER, Autodock and Gromacs underway.

The webform submission page has been constructed to allow upload of required and optional files as well as user entered metadata such as job name and job description. Once the user has uploaded and entered the relevant information, they submit the webform and the job is sent away. The user can then navigate to the 'My Jobs' page which displays a list of all their jobs and some summary details. A user can also view specific details of each job, including a list of downloadable files (input and output). Once a job has returned, successful or not, an email is sent to the user.

From the user perspective the submission process is straightforward. However, once the job is submitted a myriad of events takes place. A custom script has been developed for each submission type that processes and submits the files stored in the iRODS repository at upload, as well as user-defined metadata, to NIMROD/G on the middleware server. The web server scripts interact with NIMROD/G via a web service. The web service only accepts authenticated requests from authenticated hosts. Through the web service, the job files are packaged and submitted to NIMROD/G with metadata, a NIMROD/G job is created and the job is associated with one or more resources. The job then begins processing.

The processing is handled by NIMROD/G, which is grid and cloud aware and allows for cloud bursting. Because of these features we have been able to use resources such as the Griffith HPC cluster as well as NeCTAR's research cloud. These can be added as separate resources, or using Nimrod/G's economic scheduling capabilities, which allow cloud bursting for job overflow (e.g. if a job takes too long it will fire up a NeCTAR instance). Once the job returns from processing the output files are stored in the same iRODS repository as the input files and the job owner is notified by email.

The incorporation of an iRODS repository allows persistent storage and management of large job input and output files over multiple servers. It is highly configurable and easily extensible and performs well in large-scale testing with ~50 million files and ~250 million annotations (metadata items) (2).

The Computational Resource Framework (CRF) is a solution that allows researchers to conduct HPC job submission in a straightforward, easily accessible manner. It does this by providing

simple web portals for job submission that hide the complexity and technicality traditionally associated with HPC.

## **REFERENCES**

1. Nimrod/G. Retrieved May 12th, 2013, from <http://messagelab.monash.edu.au/NimrodG>.
2. What is iRODS?. Retrieved May 12th, 2013, from <https://www.irods.org>.

## **ABOUT THE AUTHOR'S**

<sup>1a</sup> Dr Andrew Lewis is a Senior Research Specialist in eResearch Services and an Adjunct Senior Lecturer in ICT at Griffith University. Prior to this appointment he worked in industrial applied research with BHP Billiton. His research interests include: parallel optimisation algorithms for large numerical simulations, including evolutionary programming, particle swarm and ant colony systems, multi-objective optimisation techniques for engineering design, and parallel, distributed and grid computing methods. He has numerous publications in computational optimisation and has been involved with high performance computing research and applications for over 20 years.

<sup>1b</sup> Ms Heidi Perrett is an eResearch Web and Software Developer in Griffith University's eResearch Services group. She graduated with a double degree in Science and IT majoring in Biological Sciences and Information Systems. She has been a lead developer on a number of systems utilizing open source software platforms for eResearch support of researchers in all disciplines. She has developed discipline-specific portals and tools using Drupal, Wordpress, LimeSurvey, CiviCRM, OME and a variety of scripting languages.

<sup>1c</sup> Ms Amanda Miotto is an eResearch Support Specialist in Griffith University's eResearch Services group. She graduated with a Bachelor of Science majoring in Bioinformatics before gaining hands on experience developing software systems in support of a range of projects in genomics, stem cell research, systems biology and microbiology. Amanda writes in php, python, sql, and R primarily and has worked on projects using tools such as UCSC Genome Browser mirror, AISRAP websites, Limesurvey, Cytoscape, Galaxy, Chipster and OMERO for Griffith's Adult Stem Cell Data Capture Project. She also acts as a liaison for research groups such as the Genomic Research Centre to assist in providing computing and storage needs.

<sup>1d</sup>Mr Jan Hettenhausen is a Senior eResearch Web and Application Developer in Griffith University's eResearch Services. He has graduated with Masters of Information and Communication Technology and is currently undertaking a PhD in Computer Science entitled "Interactive Multi-Objective Particle Swarm Optimisation". He is an experienced developer in Python and other languages and has been developing software for research professionally as well as for his PhD research.