

# A RESTful Web Service for High Performance Computing based on Nimrod/G

Jan Hettenhausen<sup>1</sup>, Andrew Lewis<sup>2</sup>, Blair Bethwaite<sup>3</sup>, Heidi Perrett<sup>4</sup>, Amanda Miotto<sup>5</sup>

<sup>1</sup>Griffith University, Brisbane, Australia, [j.hettenhausen@griffith.edu.au](mailto:j.hettenhausen@griffith.edu.au)

<sup>2</sup>Griffith University, Brisbane, Australia, [a.lewis@griffith.edu.au](mailto:a.lewis@griffith.edu.au)

<sup>3</sup>Monash University, Melbourne, Australia, [blair.bethwaite@monash.edu](mailto:blair.bethwaite@monash.edu)

<sup>4</sup>Griffith University, Brisbane, Australia, [h.perrett@griffith.edu.au](mailto:h.perrett@griffith.edu.au)

<sup>5</sup>Griffith University, Brisbane, Australia, [a.miotto@griffith.edu.au](mailto:a.miotto@griffith.edu.au)

## INTRODUCTION

The Computational Resource Framework (CRF) is a project being undertaken at Griffith University to make High Performance Computing (HPC) enabled scientific software available via a range of web portals. It addresses the problem that the use of HPC resources currently still needs a high level of expertise with the Unix command line interface, creating a technical boundary that limits uptake among researchers. As pointed out by Cholia et al. [2], this hinders cross domain research and either excludes many researchers or makes them rely on the few researchers in their group who are comfortable with the current HPC system.

A major technical challenge of the CRF project was to expose the HPC infrastructure at Griffith University in a way that allows easy deployment of new front-end portals. In addition, there was a desire to integrate external HPC resources, such as HPC resources from other institutions and cloud resources from, for example NeCTAR or commercial services. A lot of this functionality could be achieved by using Nimrod/G [1], a Grid and Cloud aware distributed computing middleware with interfaces into a wide variety of HPC resources, such as Globus, PBS, EC2 and OpenStack. NIMROD/G also allows cost and time-based scheduling of jobs, which can be defined via a simple declarative language. Nimrod/G's main interface is a command line interface with separate instances of Nimrod/G and the Nimrod/G database running for each user.

To allow easy programmatic access to the HPC infrastructure a key part of the CRF project was to develop a state-of-the-art API through which portals and other applications could programmatically access HPC jobs and resources. We implemented this in the form of a RESTful web-service that maps all relevant functionality of Nimrod/G to appropriate web resources. The main challenge on the front-end side of the API was to define a sensible RESTful API to manage jobs, resources and users. Other challenges arose from the integration with Nimrod/G as this system had to be implemented to allow access to each users database and start and control instances of Nimrod/G in their user context, while at the same time presenting all this information via a single API.

## IMPLEMENTATION

Nimrod/G currently features a command line interface as the primary frontend. Via this interface a user can create, monitor and control all jobs. A separate PostgreSQL database is maintained for each user, holding information about the users jobs and resources. A user's jobs are controlled by a separate instance of Nimrod/G for each user.

To use Nimrod/G as the middleware for the CRF we needed to develop a web-service that could seamlessly integrate with the existing Nimrod/G system. This required us to develop a way to access each users PostgreSQL database and run commands in the context of each user. To minimise the security risks from giving a service such broad access to user accounts, we developed a two-tier system comprising a background daemon with an internal API and a web service front end that can access this daemon via secure authentication. Both services can run as different, non-privileged users and only the background daemon requires access to other user accounts. Access to user accounts is limited via the *sudoers* file on Linux/Unix, giving the process only access to certain users and certain commands. An advantage of this approach is that no component of the web-service requires super-user rights at any time. Nimrod/G calls are seamlessly handled by a component within the background daemon that executes calls to the Nimrod/G database and other Nimrod/G components in the context of the respective user.

Both services were developed in Python using the event driven network engine Twisted. In addition the WSGI web-framework Flask in combination with the authentication framework repose who were used to develop the frontend API. Communication between the two components is secured with SSL Client and Server authentication.

## API METHODS

Currently the following resources and methods are available via the HPC web-service:

### Authentication

This service has been designed for use by portals or portal like systems. It is therefore mainly focussed on authenticating portals via API keys and allowing them access to a distinct group of users. However, this service was implemented in a flexible and extensible manner so that extensions to authentication and authorisation of individual users could be trivially implemented.

### Job Management

The API allows complete management of jobs, allowing creating, monitoring, controlling and deleting jobs. Compute resources can either be assigned manually or automatically based on job type. In addition, numerous metadata fields can be stored and managed per experiment.

### HPC Resource Management

Nimrod/G allows users to add a wide variety of grid and cloud resources. Due to the structure of Nimrod/G these are managed on a per user basis. Users resources are exposed via the web-service, which, in addition to the information maintained by Nimrod/G, maintains a database mapping these resources to global resource information. This way the web-service makes it possible to query for resources that can run a specific type of software or support a particular library (e.g. GPU, MPI). If desired suitable resources can automatically be added to a job.

### File Management

The web-service offers full support for uploading and downloading input and output files for experiments. We are currently in the process of replacing most of this functionality with the integrated rule-based data management system iRODS. iRODS will allow us to more comprehensively maintain data and meta-data and process and distribute it according to pre-defined rules.

### Account Information

The web-service allows read access to information about all users with access to the HPC resources. Actual account information is available as read-only data for security reasons. User creation and change of user information is available via the admin console.

## FUTURE WORK

The current API is targeted at programmatic access from a portal or similar service. Future extensions might include the option to handle sessions via cookies and incorporate a purely Javascript based interface for either full or admin access.

## REFERENCES

1. Abramson, D., Giddy, J. and Kotler, L. "High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?", International Parallel and Distributed Processing Symposium (IPDPS), pp 520- 528, Cancun, Mexico, May 2000
2. Cholia, Shreyas, David Skinner, and Joshua Boverhof. "NEWT: A RESTful service for building High Performance Computing web applications." *Gateway Computing Environments Workshop (GCE), 2010*. IEEE, 2010.
3. Hedges, M., Hasan, A., & Blanke, T. (2007, November). Management and preservation of research data with iRODS. In *Proceedings of the ACM first workshop on CyberInfrastructure: information management in eScience* (pp. 17-22). ACM.

## ABOUT THE AUTHORS

<sup>1</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.

<sup>2</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>3</sup> Mr Blair Bethwaite is a Monash Computer Science graduate who joined the lab in 2006. At the time he was also working with Monash's Optimization and Constraint Solving group in conjunction with the Monash library, and in e-Learning. Now a full-time MeSsAGE Lab employee for three years Blair has contributed to a variety of projects within the lab. He has broad experience in Grid and distributed computing and has helped apply and evolve the Nimrod tools to numerous driving applications. Blair also does much of the user support, testing and system administration within the group.

<sup>4</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>5</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.