JISC DEVELOPMENT PROGRAMMES

Project Document Cover Sheet

Final Report

Project

Project Acronym	GROWL	**Project ID**	None Assigned
Project Title	VRE Programming Toolkit and Applications (GROWL: Grid Resources on Workstation Library)		
Start Date	2005-02-01	**End Date**	2007-01-31
Lead Institution	University of Cambridge, e-Science Centre		
Project Director	Mark Hayes		
Project Manager & contact details	John Kewley, j.kewley@dl.ac.uk CCLRC e-Science Centre, Daresbury Laboratory, Keckwick Lane, Daresbury, Warrington, Cheshire, WA4 4AD tel: 01925 603513 fax: 01925 603634		
Partner Institutions	CCLRC Daresbury Laboratory, e-Science Centre University of Lancaster, e-Science Centre		
Project Web URL	http://www.growl.org.uk		
Programme Name	Virtual Research Environments (VRE) 5/04		
Programme Manager	Frederique Van Till Programme Manager (VRE) tel: 07875 33 8070 f.vantill@jisc.ac.uk		

Document

Document Title	GROWL: Final Report		
Reporting Period	Mar 2006 – Jul 2006		
Author(s) & Role	John Kewley (Project Manager), 01925 603513		
Date	2007-02-06	**Filename**	GROWL_Final-1.pdf
URL	www.growl.org.uk		
Access	☒ Project and JISC internal		☐ General dissemination

Document History

Version	Date	Comments

Table of Contents requested, but these are left off for now to save on printing

1 Acknowledgements

Note the name of the JISC programme, and that the project was funded by JISC. The project may also want to list the project partners and acknowledge any person or organisation that was helpful during the project or in writing the report.

The GROWL project is grateful for the funding it received from the JISC Virtual Research Environment (VRE) Programme. It was a collaboration between the University of Cambridge e-Science Centre (lead institute), the University of Lancaster Centre for e-Science and the CCLRC e-Science Centre at Daresbury Laboratory.

The project was initially an 18 month project but received a no-cost extension to 24 months due to staffing problems. The project partners would like to thank the JISC for their support and flexibility especially with respect to this extension. We also acknowledge the support and comments we received from the Tavistock Institute during their VRE Programme evaluation.

Special thanks are also due to the various users who provided input to our User Requirements and tested the resulting software.

2 Executive Summary

Summarise highlights of the project (one page), including aims/objectives, overall approach, findings, achievements, and conclusions. The full report may include technical terms, but try to keep the executive summary in **plain English.**

Grid computing promises great things for Computational Scientists and Social Scientists. Reality is that moving computational jobs from desktops and clusters to the Grid involves overcoming a variety of obstacles.

As Grid technology matures and emerges from e-Science research as production systems like the National Grid Service (NGS) [15], the North West Grid (NW-GRID) [14] and Campus Grids [6], its potential user-base will expand. Some of these user communities such as Bioinformatics and the Social Sciences rely on well-established software solutions such as R, Stata and Matlab. These applications do not easily fit into the typical grid environment. There is also a need for integrating heritage applications written in Fortran, C and C++ into the Grid in as seamless a way as possible.

Our aims in this project were to encourage the uptake of Grid-based computing and distributed data management, focusing on the issues which could either hinder or facilitate end-user application development. We refer to the difficulties identified as the “client problem” and suggest a solution to build upon the existing prototype GROWL library to produce a truly lightweight extensible toolkit which complements other solutions.

From our initial requirements gathering exercise and maybe even more so from out close inter-working with the quantitative and computational scientists, it has become clear that user’s needs vary considerably. Contributing factors include whether they are coming from a background where

cluster or parallel computing is the norm (in which case many aspects of the Grid paradigm are fairly accessible) and which environments they are comfortable (or indeed uncomfortable) using.

This led the development towards both the more simplistic GROWL Scripts for direct access to Grid resources and the development of the new GROWL server to support the SABRE-R project.

3 Background

Summarise the background to the project (and how it builds on previous work) and the need it for it (and why its important).

The need for lightweight client toolkits was identified in late 2003. Chin and Coveney [1] listed a number of problems associated with Grid middleware packages then existing which were seen as a barrier to the uptake of Grid technologies, even in the more established computational sciences such as chemistry and physics:

“We have encountered serious middleware-related problems which are hindering scientific progress with the Grid:

- *The existing toolkits have an excessively heavy set of software and administrative requirements, even for relatively simple demands from applications;*
- *Existing toolkits are painful and difficult to install and maintain, due to excessive reliance on custom-patched libraries, poor package management, and a severe lack of documentation for end-users;*
- *Existing standards bodies and the task forces within the UK e-Science programme are not engaging sufficiently with the applications community, and run a substantial risk of producing and implementing Grid architectures which are irrelevant to the requirements of application scientists.”*

They in turn cite Gabriel [2] who argues the benefits a simpler design philosophy in cases where the right thing is too complex, concluding:

“We argue that it is important to develop a simple, lightweight Grid middleware which is good enough for rapid adoption, rather than taking longer to develop a solution which will, supposedly, suit all needs. Such a toolkit must be:

- *substantially more portable, lightweight, and modular in design;*
- *produced in very close collaboration with application scientists;*
- *sufficiently well-documented that end-users will be able to port existing codes to use Grid techniques with the minimum of hassle.”*

It was this that prompted the Grid Technology Group at CCLRC Daresbury Laboratory to produce the prototype GROWL (Grid Resources On Workstation Library [3]). It was based on the C/C++ language and used the gSOAP library to create a Web service client-server model in a library which interfaces to existing services on the Grid.

4 Aims and Objectives

List the aim and objectives agreed at the start of the project, and note if they changed during the project.

The GROWL project aims were to encourage the uptake of Grid-based computing and distributed data management, focusing on the issues which may hinder or facilitate end-user application development. We proposed a solution that built upon the existing prototype GROWL library to produce a lightweight extensible toolkit which complements other solutions, utilising Web Service technologies (in particular gSOAP).

Lightweight implies that GROWL should be easy to install, with minimal but sufficient functionality for the user requirements we have identified in target application areas. It should also be possible to install the GROWL library quickly on a variety of client workstations running Linux or a similar UNIX-like operating system with a minimum of additional software.

Likewise any external dependencies should be minimised, so for instance there will be no requirement to perform the installation as `root` and only the parts of the library that are needed will be built and loaded.

Extensible means that it should be possible to easily extend the GROWL library to provide interfaces to additional middleware services (e.g. CONDOR, Netsolve, SRB) or to use additional security mechanisms (e.g. Shibboleth).

The project's intent was to fulfil these aims by meeting the following objectives:

1. Generate user requirements for a lightweight grid application toolkit for the three target user communities.
2. Produce GROWL toolkit: it should be possible to install the GROWL library quickly on a variety of client workstations running Linux or a similar UNIX-like o/s with a minimum of additional software.
3. Evaluate GROWL for Bioinformatics applications.
4. Evaluate GROWL for Physics and Chemistry applications.
5. Evaluate GROWL for Social Science statistical analysis applications.

5 Methodology

Summarise the overall approach taken and why this approach was chosen over other options considered. Then describe the methodology in more detail. Depending on the project, this might include the methodology for research you carried out, technical design or development, evaluation, etc. Finally, note any specific issues that had to be addressed by the methodology, e.g. standards, interoperability, scalability, etc.

A Web Service is an application accessible using standard Internet protocols. Web Services represent black-box functionality that can be reused without concern for how that service is implemented, or what language it is written in and are accessed via ubiquitous Web protocols (e.g. HTTP) and data formats such as SOAP, WSDL and UML.

GROWL uses gSOAP to generate the client-side of the Web Services which are then wrapped by the GROWL API. The gSOAP compiler tools simplify the development of Web services by their provision of C and C++ language bindings for stub and skeleton code generation. This results in a flexible framework that can be also be utilised from Fortran. This would allow a subsequent version of GROWL to provide a Fortran library rather than providing Fortran wrappers to the GROWL C library. The gSOAP stub compiler automatically does all the data conversion from user-defined C and C++ data types to equivalent XML data types and vice-versa.

Once this is done, calls made to the GROWL library are redirected to the appropriate service on the GROWL Server and from there to an appropriate machine to execute the job (as is shown in Figure **??**). It should be noted that since the GROWL Server is not over-constrained by institutional firewalls, it is accessible by submission and execute machines alike. This has the advantage that if the user's machine is unable to reach a given resource directly, the GROWL Server will act as a proxy for it. This, in a controlled way, circumvents the firewalls and means that the user can take advantage of Grid functionality without having to open any ports in his submission machine's firewall.

In order for the GROWL server to act as a proxy in this way, it must be able to act for the user. To ensure this is possible, GROWL provides an authentication service whereby users can upload a proxy to their GSI certificate to a MyProxy server and can then instruct the GROWL server to request a delegated certificate to act on their behalf.

GROWL functionality is therefore logically split into two main parts, the GROWL Client and the GROWL Server, corresponding to the client and server aspects of a Web Service. Note that additional parts of the GROWL Client are not actually web service clients (e.g., the remote file handling routines) and these can only be used if there is direct access from the client machine to the target resource.

The GROWL Client library is linked to the users application and makes Web Service calls to the GROWL Server. The GROWL Server is a set of GROWL services running on a well known machine, accessible (with respect to firewalls) to the users application machine, the execution resources and a MyProxy Server. For each supported Web Service on the GROWL Server there will be analogous client functionality in the GROWL Library. Note that a GROWL client library should be able to talk to more than one GROWL Server for different GROWL services.

6 Implementation

Describe how you planned and implemented the project work and the activities it involved. Depending on the project, this might cover technical development, processes, how you conducted user studies, etc. Include any problems or issues that arose and how you handled them, where readers can learn from your experience. Tell the story of what you did rather than listing workpackages.

Each site was responsible for collating user requirements from their respective user communities. These requirements were collated into a User Requirements document.

Lancaster were responsible for the GROWL Server and Daresbury for the GROWL Client.

Blah blah - need to add more to this

7 Outputs and Results

Explain the end result of the project work in an objective way. Depending on the project, it might include research results, findings, evaluation results, data, etc. If the project created something tangible like content, a portal, or software, describe it. Engage the reader, and avoid a long list of deliverables.

Mention different strands of work - all with common aim. Put this as positive spin rather than negative.

8 Outcomes

In this section, assess the value of the project work. List project achievements against the aims and objectives set. Summarise project outcomes and their impact on the teaching, learning, or research communities. Indicate who will benefit from the work, how, and why. Also comment on what you learned that may be applicable to other projects, e.g. whether the methodology worked.

8.1 Achievements against objectives

1. *Generate user requirements for a lightweight grid application toolkit for the three target user communities.*

   A User Requirements document has been produced following meetings with a variety of potential users, both individually and at workshops.

   The "Collaboratory for Quantitative e-Social Science (CQeSS)" project has expressed an interest in using GROWL for their work and Adam Braimah (DL) has produced an SRB client for use GROWL. He is currently investigating whether OGSA-DAI could be handled by GROWL.

   An interface to the GROWL SABRE service hosted at Lancaster is being provided as a free plug-in for the commercial statistics package Stata. This development is being undertaken in recognition of the need to make the Sabre service available from within the software tools most commonly employed by the social science and economics research communities.

   Since several CCLRC Staff have been using GROWL Scripts as an easy way to install the VDT globus tools, this work will be spun-off as a separate part of the GROWL client since it can be used independently of the Server and is being trialled for use by NGS, NW-GRID and SCARF clients.

2. *Produce GROWL toolkit: it should be possible to install the GROWL library quickly on a variety of client workstations running Linux or a similar UNIX-like o/s with a minimum of additional software.*

   The following groups in Computational Science and Engineering (CSE) at Daresbury Laboratory (in addition to those mentioned under Physics and Chemistry below) are using GROWL as an easy way to install VDT and for certificate transformation.

   - Computational Engineering Group (Simon Mizzi)
   - Band Theory Group (Martin Lüders)

- Advanced Research Computing Group (Martin Plummer)

Maybe mention Nick Harrison too with his Imperial College hat on!

3. *Evaluate GROWL for Physics and Chemistry applications.* The following groups are using GROWL Scripts within the CSE Department at Daresbury Laboratory:

   - Quantum Chemistry Group: Jens Thomas has used GROWL scripts as a globus client interface for their CCP1GUI [12]. This provides a graphical interface for community codes including GAMESS-UK [13].
   - Materials Science Group, Barry Searle is using the GROWL C interface to submit jobs to grid resources from DL Visualize [8].

4. *Evaluate GROWL for Social Science statistical analysis applications.*

   SABRE ... 2 paragraphs from Dan needed here

   Group at Bristol ... 1 paragraph from Dan needed here

5. *Evaluate GROWL for Bioinformatics applications.*

   Bioinformatics and Genomics Group (Micklem Lab), Department of Genetics, Peter Sykacek

   Also R work by Peter B

8.2 Workshop on Lightweight Grid Computing

Project members (with Peter Coveney of UCL) organised the Workshop on Lightweight Grid Computing in Hope, Derbyshire in May 2006 to discuss and share ideas on how we can give scientists simpler tools to utilise the Grid more effectively. A report of the event is in preparation. The objectives of the workshop were:

- To share ideas on the middleware needed to provide users with lightweight access to Grid Resources

- To look at ways to pool resources for future collaboration *(note that the use of portals to provide this capability was not considered since the workshop focused on programmatic and scripted interfaces to the Grid)*.

The event was considered a success and presentations are available from the event website [16].

8.3 NW-GRID Training

This should be somewhere else

Project members were instrumental in the running of the first NW-GRID training event. The goals of this course were to give participants:

- An understanding of the concepts of Grid Computing and e-Research

- An orientation to the NGS and NW-GRID and how to access them
- Sufficient practical experience to allow initial use of NW-GRID services in future

GROWL Scripts were used to show delegates how to install grid middleware, transform their certificates and run grid jobs with comparative ease.

9 Conclusions

Briefly summarise any conclusions that can be drawn from the project work.

- Good "demonstrator" that is being actively used, although in need of "hardening"
- GROWL Server ... *add something here*
- GROWL Scripts are in widespread use and are helping computational scientists run their jobs on the Grid.

10 Implications

Consider the future implications of your work and how others can build on it. What are the implications for other professionals in the field, for users, or for the community? What new development work could be undertaken to build on your work or carry it further?

Need for grid middleware developers to consider the client-side problem, whether firewalls, certificates or sheer amount of s/w to download, even VDT client.

Following our suggestions to make Grid Software easier to use, the following *changes have been accepted* by our 3rd party software suppliers:

- **gSOAP :** It is typical when gSOAP is used that either a client or a server is built, but not often both together. We requested that command-line options were provided for `soap2cpp` to produce client- or server-only stubs. The command-line options (`-C` and `-S`) have now been added to gSOAP for this purpose
- **MyProxy :** When the MyProxy command `myproxy-init` is used to upload a medium term proxy to a MyProxy server, it is frequently the case that the user will want to generate a local proxy straightaway afterwards. At our request, the MyProxy development team have made added `-l` option to `myproxy-init` to support this functionality.
- **GSI-enabled OpenSSL :** The ANL staff have enhanced their globus certificate handling commands to permit generation of proxies directly from the PKCS#12 (`.p12` or `.pfxA`) format certificates rather than relying of conversion to `.pem` format. MyProxy have also updated their code and we await a final bug to be fixed before this functionality can be utilised fully in the GROWL Scripts.

- **VDT :** When a client only download of VDT is requested, the amount of software that is additionally installed is considerable. The VDT team have agreed this is a dependency problem and they will look into it at some point.
- Condor - agreed the benefit of embedding GROWL Scripts functionality in the Condor-G submission back-end

11 Recommendations

Optional: List any specific recommendations for the teaching, learning, or research communities.

pass!

12 References

List any references to the work of others you have cited (e.g. articles, reports, studies, standards), and any explanatory notes. Provide URLs for any materials available on the web.

References

[1] Jonathan Chin and Peter Coveney, *"Towards tractable toolkits for the Grid: a plea for lightweight, usable middleware"*, June 2004,
http://www.realitygrid.org/lgpaper21.pdf

[2] R.P. Gabriel, *"The Rise of 'Worse is Better'"*
http://www.jwz.org/doc/worse-is-better.html

[3] The GROWL Project,
http://www.growl.org.uk/

[4] The R Project,
http://www.r-project.org/

[5] Nearest Neighbour

[6] UK CampusGrid meeting,
http://www.nesc.ac.uk/esi/events/556/

[7] Collaborative Computational Projects,
http://www.ccp.ac.uk/

[8] DL Visualize,
http://www.cse.clrc.ac.uk/cmg/DLV/

[9] Centre for Quantitative e-Social Science (CQeSS),
http://e-social-science.lancs.ac.uk/cqess.html

[10] SABRE,
http://www.lancs.ac.uk/staff/cpajp/sabre/

[11] SABRE in R,
http://www.ncess.ac.uk/research/pdp/#ogsa

[12] CCP1GUI,
http://www.cse.clrc.ac.uk/qcg/ccp1gui/index.shtml

[13] GAMESS-UK,
http://www.cfs.dl.ac.uk/gamess-uk/index.shtml

[14] NW-GRID,
http://www.nw-grid.ac.uk/

[15] NGS,
http://www.ngs.ac.uk/sites/

[16] Lightweight Grid Workshop,
http://tyne.dl.ac.uk/GROWL/Lightweight.shtml

[17] Department of Genetics, Bioinformatics and Genomics Group (Micklem Lab),
http://www.gen.cam.ac.uk/Research/micklem.htm

[18] Peter Coveney, Jamie Vicary, Jonathan Chin and Matt Harvey, *"Introducing WEDS: a WSRF-based Environment for Distributed Simulation"*, UKeS-2004-07, October 2004,
http://www.nesc.ac.uk/technical_papers/UKeS-2004-07.pdf

Optional Appendices

Include any appendixes that readers will find helpful to understand the work described or the results. For example, include a questionnaire if you conducted a survey, or technical details that support technical development carried out. A glossary of acronyms and technical terms is often helpful.