

# Putting Social Science Applications on the Grid

Rob Crouchley<sup>1</sup>, Ties van Ark<sup>1</sup>, John Pritchard<sup>1</sup> John Kewley<sup>2</sup>, Rob Allan<sup>2</sup>, Mark Hayes<sup>3</sup> and Lorna Morris<sup>3</sup>

<sup>1</sup>e-Social Science Centre of Excellence and Collaboratory for Quantitative e-Social Science, University of Lancaster

<sup>2</sup>e-Science Centre, CCLRC Daresbury Laboratory

<sup>3</sup>Cambridge e-Science Centre, University of Cambridge

E-mail address of corresponding author: [r.crouchley@lancs.ac.uk](mailto:r.crouchley@lancs.ac.uk)

## Abstract.

As e-Social Science develops, there will be a growing user base of social researchers who are keen to share resources and applications in order to tackle some of the large-scale research challenges that confront us. They will be aware of the potential of e-Science technology to provide collaborative tools and provide access to distributed computing resources and data. However social scientists are not ideally catered for by the current Grid middleware and often lack the extensive programming skills to use the current infrastructure to the full and to adapt their existing “heritage” applications.

In late 2003, a lightweight client toolkit that is easily installable, yet provides extensible access mechanisms to Grid resources was seen as a possible solution. By implementing a client-side polling strategy, problems associated with institutional firewalls for Grid protocols can be reduced. This is an alternative strategy to a Grid portal, easing the access problems while still sharing the same underlying services and infrastructure. A prototype library called GROWL: *Grid Resources on Workstation Library* was developed to use a client-server model to interface to existing Grid Services from applications written in C, C++, Fortran and R. Together with its associated wrapper services, GROWL is now being further developed into a number of demonstrators as part of the JISC-funded Virtual Research Environment (VRE) Programme by a collaboration from CCLRC Daresbury Laboratory and the Universities of Lancaster and Cambridge [1].

This paper describes the GROWL library and how it could be used to Grid-enable some computationally demanding statistical models for e-Social Science applications.

## 1. Background

The need for lightweight client toolkits was articulated in late 2003. Chin and Coveney [2] listed a number of problems associated with middleware packages then existing which were seen as a barrier to the uptake of Grid technologies, even in the computational sciences such as chemistry and physics:

*“We have attempted to make use of Grid technologies for serious scientific work, as part of the EPSRC-funded Reality Grid e-Science pilot project. 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 [3] who gives a cogent argument for the benefits for programming projects adopting a design philosophy of simplicity in cases where “the right thing is too complex” and conclude:

*“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 write the prototype GROWL: *Grid Resources On Workstation Library* [4] toolkit. This toolkit provides an easy-to-use client-side interface and is being further extended in the JISC-funded VRE project, in collaboration with the Universities of Cambridge and Lancaster. See project Web site at <http://www.growl.org.uk>.

We note there is discussion in the Applied Computer Science community on End-User Development, i.e. tools that empower users to create their own software solutions. Methods, techniques and tools are discussed which support non-programmers to adapt their applications in an intuitive way [8]. We hope that the GROWL toolkit will be seen to enhance this possibility as far as Grid computing is concerned.

It is also of possible interest to note that because of the distributed nature of the team working on this project we are adopting a very modular approach [9]. Modules are mostly independent so that the end user can select which ones are appropriate for compilation and linking into the final application. It also helps considerably with software engineering and management of the project.

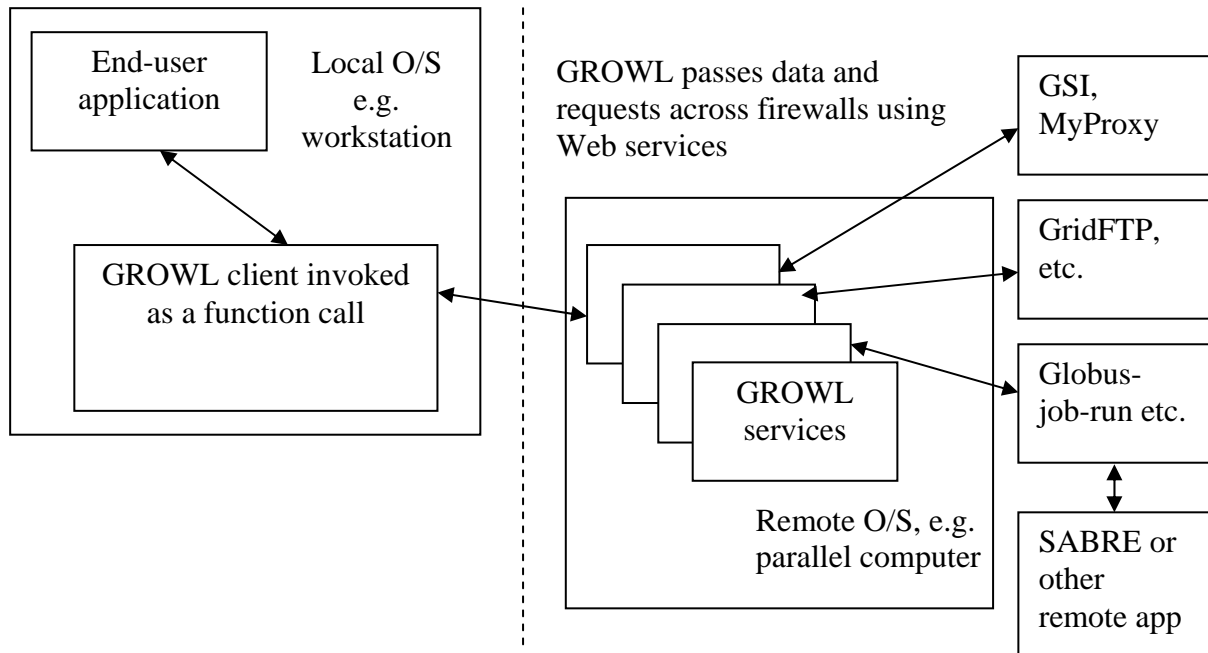
## **2. Introduction to GROWL**

Our aims are 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 refer to the difficulties identified, as the “client problem” and suggest a solution building upon the existing prototype GROWL library to produce a truly lightweight extensible toolkit which complements other solutions.

Most developers of scientific software get optimal performance and functionality by linking their bespoke applications to one or more specialised and highly tested libraries, e.g. for numerical algorithms, visualisation or data management. GROWL adopts the same model and extends it to the scenario of Grid computing. GROWL has an easy-to-use client-side application programming interface (API) to a variety of open source libraries and services. GROWL uses basic Web Services Technologies (SOAP, WSDL and UDDI), for communication although this is hidden from the user by the library interface.<sup>1</sup> Some of the applications we are considering such as Stata [17] provide facilities for plug-ins to be used to provide additional functionality such as Grid-enabling via GROWL. In this way they can continue to be used as before, but have the capability to call remote high-performance algorithms on parallel computers or access remotely-stored data when necessary.

---

<sup>1</sup> Please see documentation on Web Services for definitions of these abbreviations.

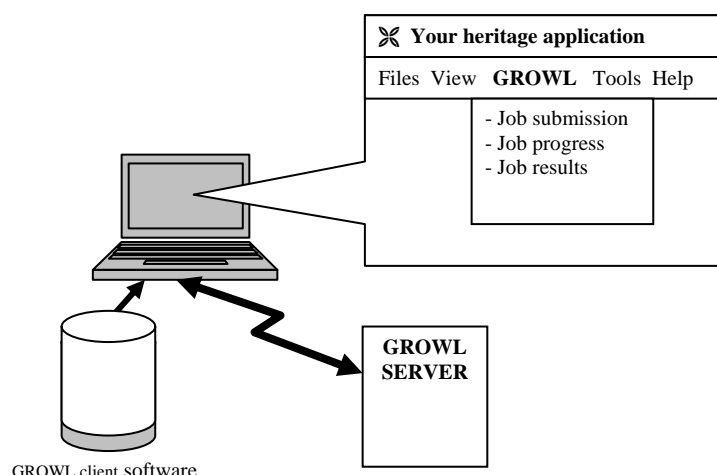


**Figure 1:** GROWL architecture with Globus showing client and server

On the server side GROWL accesses application wrappers and services, in the main developed in other projects. Typically these might be wrappers to the Globus v2 toolkit [20] for large-scale Grid resources. Globus itself has a C API, which we are using, but requires holes to be “punched” through firewalls for inter-institution communications with its own protocol and security interface. Containing this on the server side enables system administrators to keep a close control of any potential security risks. The list of required services is still being worked on, but is likely to be derived from software such as HPCPortal, DataPortal, InfoPortal (all from CCLRC), SRB (the Storage Resource Broker from San Diego Supercomputer Centre) [18], Condor [19] and NetSolve, the distributed numerical linear equation solver library from Jack Dongarra’s group in Tennessee. Alongside these associated services is a procedure for enabling a particular service to be linked into the infrastructure, whether it be written in C, C++, Perl, Python, PHP, R or Java. This is illustrated in Figure 1. The basic services in GROWL such as authentication can also be used on the server side simply by installing the GROWL library there too. We term this usage a “GROWL Grid”.

To the end user however, GROWL is a purely client side Grid programming environment, it does not help the user create Web or Grid services, nor does it help the user put their computer on the Grid. The software and functionality is kept to a minimum for ease of installation and use on a workstation or PC. For social scientists, particularly those engaged in statistical computing, this will be provided through APIs. Those already familiar with Stata or R can then immediately become part of a large Virtual Research Environment. The graphical user interface R-Commander will also be used by extending its menu capability to include GROWL functions as indicated in Figure 2.

Particular requirements of the project are that GROWL be “lightweight” and “extensible”. Lightweight implies that the GROWL should be extremely easy to install, with functionality minimal but sufficient for the user requirements we will identify in target application areas. Extensible means that it should be possible to easily extend GROWL to provide interfaces to additional middleware services or to use additional security mechanisms such as Shibboleth and PERMIS.



**Figure 2:** GROWL the end user/ client side

## 2.1 e-Social Science Applications

We are developing and evaluating GROWL for Social Science statistical modelling applications. In this work we will:

- provide Web service wrappers to parallel statistical modelling libraries and software, e.g. the SABRE [5] package (some of this work is already ongoing) and some key routines from the Harwell Subroutine Library (HSL), [6] in the first instance, so that they can be used from within standard applications such as R and Stata;
- implement GROWL functionality in existing GUI interfaces for the social science community;
- use GROWL client-side tools to make SABRE and HSL functionality available via the ReDReSS Web portal for training purposes.

The *OGSA Component-Based Approach to Middleware for Statistical Modelling* (OGSA) e-Social Science pilot demonstrator project is porting the open-source SABRE statistical analysis package to a parallel computing Grid-based environment. The statistical modelling package SABRE, is used for analysing work/ life history data,

The *Collaboratory for Quantitative e-Social Science* (CQeSS) is the first node of the National Centre for e-Social Science. CQeSS held an Agenda Setting Workshop on 6<sup>th</sup> April to identify requirements and other issues important to e-Science uptake [7]. We are now seeking to apply the GROWL middleware to wrap the statistical modelling methods available in SABRE as well as other computationally-demanding models and to make these developments available in the distributed environment as a componentised R library and as a Stata “plug-in”. The HSL routines have been used to solve the linear algebra equations that occur in the estimation of fixed effects linear models, see [10].

The free-to-use R language and environment provides a wide variety of useful statistical and graphical techniques (linear and non-linear modelling, statistical tests, time series analysis, classification and clustering). R is an interpreted functional language. It is a GNU open-source version of S, originally from Bell Labs, see <http://www.r-project.org/>. R is very much a

vehicle for newly-developing methods of interactive data analysis and particularly used in statistical computing applications.

Middleware extensions to R are being added using Web services in the GROWL library and appropriate additional wrappers for R clients and services; this has already been demonstrated for the simplest cases. This pilot project will play a key role in the development of some of the middleware components and services appropriate to quantitative e-Social Science.

Stata is very widely used in the Social Sciences. Stata is a commercial, statistical package that contains several hundred statistical tools. These include random effect survival models, models for the analysis of panel data, such as generalized estimating equations, as well as models with sample selection and classical time series models.

In release 8.1 of Stata the "plug-in" feature was introduced. It provides a way of adding to the Stata command set a user-written procedure in a compiled language, in this case C. This has the potential to greatly increase the speed with which certain computations are done in Stata's own interpreted language. It is described at the URL <http://www.stata.com/plugins/>.

Although the interface provided is for procedures written in C, the mixed language programming environment in the Solaris operating system running on the HPC at Lancaster allows a C "wrapper" procedure to call an application written in a different language, e.g. Fortran in the case of SABRE. A Stata plug-in written in Fortran within a C wrapper has already been tested and found to work as expected.

SABRE has already been implemented as a package within R using a similar external language interface (In the case of R there are explicit macro interfaces for code written in C or Fortran [15]). Work undertaken previously, to generalise the SABRE code so that the same base code could be compiled both as a stand-alone package and as a function callable from R, should facilitate the easy implementation of SABRE as a Stata plug-in. The basic requirements, apart from the ability to dynamically load in the compiled Fortran code, are a means to transfer datasets to and from the host package, Stata in this case, and a means of displaying log file messages while running within the host package.

Stata provides these facilities although there are reservations about the efficiency with which large datasets can be transferred given the one element at a time interface provided.

## **2.2 Other applications**

In addition to Social Science applications, GROWL will be evaluated for use by BioInformatics, Physics and Chemistry researchers to ascertain its value as a generic Virtual Research Environment. See the Web site for further information about these application areas.

## **3 The GROWL Toolkit and Services**

Our initial aim is to produce a number of demonstrators using the GROWL toolkit and to show that they are suitable for all the application fields. It should be possible to install the GROWL library quickly on a variety of client workstations running Linux or a similar UNIX-like operating systems and Windows XP, with a minimum of additional software and to access a range of basic functionality. Requirements identified so far include:

- ability to access basic Grid services including authentication, file transfer, resource discovery and job submission. This is done by providing wrappers to the Globus v2 toolkit;
- provide client and wrappers to existing VRE resources and services developed in e-Science or JISC projects. Examples might include cross search facilities for data discovery such as X-grain and SPP (the latter was a portal interface to the RDN network including the SOSIG data collection for social science);
- integration of new common services (such as Condor, NetSolve and SRB) into GROWL;
- produce clients enabling ease of use of the National Grid Service and other HPC resources on the Grid.

Prototype interfaces for GROWL services have already been created and can be called from C or R language programs and there is a prototype for creating C, Perl and R services. The aim of this work is to make it easy to call remote functions from an R or Stata script.

The underlying gSOAP C/ C++ library for Web services [16] is used which can also be called from Fortran applications. Wrappers in GROWL will be designed to interface to a variety of services as illustrated in Figure 1 above. Given a C language header file defining the functions to be called remotely, the gSOAP system automatically creates a set of code stubs and libraries containing the necessary client and server parts of the Web service. It also creates a WSDL file, which can be stored in a registry (such as UDDI) for discovery of the service by a third party. Because Web services are “language agnostic”, interfaces can be generated in a variety of ways for Java, Perl, PHP, Python, or C. We chose the C version on the client side because of the ease with which it can be used in our target applications. Whilst this may not permit dynamic service discovery or “late binding” in the way that Java can, we do not envisage using these capabilities in the near future as semantic descriptions of the services we need are not available in a form which can allow them to be automatically consumed.

Installation of the client software is extremely easy and fulfils our criterion of being lightweight. The required modules can be selected in a configuration file; currently these include authentication, resource discovery, file upload/transfer and job submission. A `makefile` includes the possibility to download gSOAP or to use an already compiled version and then creates all the necessary stubs and library objects using the GROWL source, header files and the gSOAP `soapcpp2` compiler. We are providing GROWL with a set of test programs and examples showing how it can be linked to the end-user application. This simply requires referencing one header file and linking to the GROWL library. For other details of how GROWL works, see the Web site where there are a variety of draft papers.

#### **4. The use of GROWL by Social Scientists**

At the beginning of the GROWL VRE project we envisaged functionality which would be appropriate to requirements in estimating complex statistical models using the Grid. The Grid provides the technology for us to address the computational limitations present in statistical software, such as that occurring in the estimation of fixed and random effects models on large data sets, i.e. those with  $>10^6$  cases and  $>100$  covariates. There are many other potential statistical models applications, e.g. in the analysis of financial time series and data mining but these are beyond the scope of this paper.

Figure 3 shows a typical scheme in which a number of data sources are harvested to create

working data sets, perhaps under different conditions of a hypothesis. The working data may be re-combined with additional raw data following analysis so that a cycle of data harvesting, merging and analysis is created. The components can be linked together using the Grid and naturally involve access to data sources and high-performance computers.

We will illustrate how we will use GROWL for the estimation of fixed and random effects models in Stata and R. These models are the two main statistical approaches to the analysis of recurrent events and each addresses explicitly the issue of dependence between an individual’s apparently separate responses. Random effects models are estimated from the marginal likelihood obtained by integrating the individual specific random effects out of the model. The class of fixed effect models we consider here explicitly estimate the individual specific effects. This is equivalent to generating dummy variables for each individual (case) and including them in the regression to control for these unobserved but fixed effects.

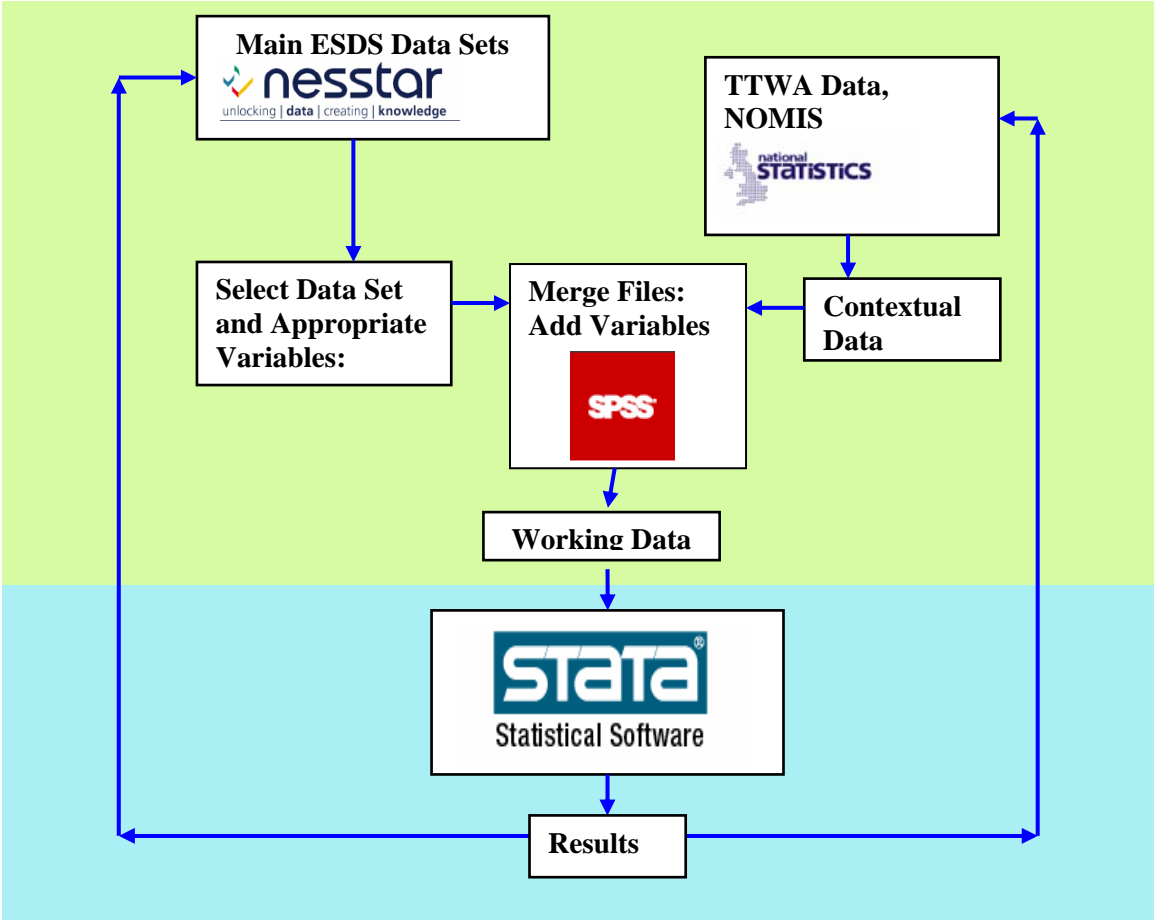


Figure 3: The Analysis Cycle

4.1 Random Effects Models

For substantive reasons, social scientists need the desirable features of random covariate parameters (i.e. acknowledging more stochastic complexity) and multi-process capability (i.e. acknowledging the interdependencies between different aspects of behaviour).

SABRE has been shown to be over 65 times faster than the commercial package Stata on large data sets on a single processor of our Dual Sun Blade 1000/2000 systems. Furthermore,

the parallel version of SABRE is 15 times faster on 16 processors than the serial version of SABRE. Thus we consider SABRE to be a good platform from which to start.

We are now developing a multivariate random effects version of SABRE. SABRE can now estimate a correlated bivariate random effects model using 2-dimensional Gaussian quadrature. There are many situations like this in which social scientists need to jointly estimate models for several outcome types and allow for dependence/correlation across outcome types. The outcomes can be of many types, e.g. continuous, ordered and unordered categorical outcomes, duration (hazard), and counts.

**Illustrative Example:** Labour market behaviour and social exclusion.

Social exclusion is an important economic, political and social problem that renders a substantial proportion of the population disadvantaged, dis-enfranchised and dis-affected. In previous work we sought to assess how social exclusion arises in the context of labour market transition behaviour, and to analyse the determinants of exclusion. To do this, we estimated a multi-state, multi-spell, competing risks model and identified five states: high-skilled employment, intermediate-skilled employment, low-skilled employment, unemployment and out of the labour market. The data used for estimating our model are extracted from the first seven waves of the British Household Panel Survey, which refer to the period 1992-97. We showed that there are a substantial number of workers trapped in a vicious circle of low-skilled employment, unemployment and inactivity. Consequently, this group are more likely to suffer social exclusion.

Our previous analysis however assumed that the risks were independent and we ignored the initial conditions problem [11, 12]. Ideally, we would estimate this model in a simultaneous framework, but the computational burden using serial software is too great. The extended parallel version of SABRE will allow for dependence between the competing risks, and to control for the so-called “initial conditions problem”. This would require multivariate integrals and the simultaneous estimation of over 250 parameters.

**Illustrative Example:** Young people’s routes into Higher Education

Young people’s routes to Higher Education have been examined using information on their experiences of education, training and work, as well as their aspirations, family and personal circumstances, collected as part of Cohort 3 of the Youth Cohort Study (YCS) of England and Wales. In the YCS, data on young people is collected at three sweeps: when respondents are aged 17, 18 and 19, but attrition is very high. For example, the number of respondents participating in YCS Cohort 3 drops from 14,000 in sweep 1 to 8,000 in sweep 3. Previous work shows that dropout, or attrition, is informative, with the consequence that more able individuals are over-represented by sweep 3. A sequential random effects model has already been developed to take into account the residual heterogeneity between students [13]. The parallel extended version of SABRE will allow us to estimate a joint sequential model for staying on at school at each age with a sub-model for dropout. This approach will allow us to take into account informative dropout, whilst estimating the effects of potentially significant explanatory variables describing the individual (e.g. gender, ethnicity, or marital status), their parents (e.g. education, socio-economic status and economic activity) and family circumstances (e.g. housing tenure, number of siblings). Such a model will require the evaluation of multivariate integrals and the simultaneous estimation of over a hundred or more parameters.

## 4.2 Fixed Effect Models

The estimation of fixed effects models are also computationally demanding, the larger the number of cases the larger the number of parameters that need to be estimated, various approximations have been proposed [14], but concern has been expressed about their adequacy.

### **Illustrative Example:** Linked employer-employee models

To understand the operation of the labour market we need to acknowledge the fact that labour market outcomes (e.g. wages) are driven by the decisions of both firms and workers. Thus a regression model of wages needs to contain terms that allow for both employer and employee observed and unobserved (fixed) effects. An analysis which does not acknowledge this matching will give only a partial picture of the determinants of a person's wages. Acknowledging both sets of fixed effects is computationally demanding. For instance, estimating the fixed effects for the IAB (Institut für Arbeitsmarkt- und Berufsforschung, Nürnberg) labour market wage data has 5.1 million worker years ( $N^*$ ) of data on individuals employed at 1,821 firms ( $J$ ), with 64 covariates ( $K$ ). The Least Squares Dummy Variables (LSDV) estimator of the fixed effects regression model has the best properties, but the estimated fixed effects are inconsistent, but unbiased. There are other estimators. Some data sets have tens of thousands of firms, or even hundreds of thousands and the estimators data matrix can be prohibitively large for software packages that store all the data in memory, e.g. R and Stata.

We are investigating the possibility of going parallel by adapting the HSL subroutines for solving the large sparse linear equations that occur in the estimation of these models. These adaptations will be made available to R and Stata via GROWL plug-ins.

## 5. The Benefits of using GROWL

In addition to accessing the significant increase in computational power required to make progress with the more demanding scientific research as described above, GROWL offers the social scientist some additional distinct benefits:

- GROWL will be an easily installable toolkit. The installation will involve minimal efforts from system administrators and can be completed by the users;
- GROWL will be made accessible from within existing applications to provide these with increased computational power;
- GROWL will negotiate institutional firewalls through the use of Web service protocols, thereby removing thresholds for the use of "Grid power" and further limiting the need for involvement of system administrators. This is done at both the host institution and the remote institutions whose systems are used for providing computational power;
- The demand for in-depth computational knowledge to access Grid middleware is minimised;
- GROWL will be easily extendable with further functions and services.

## 6. Conclusions and Summary

We have taken the opportunity in this paper to explain the objectives of the GROWL project, its architecture and some sample Social Science applications.

The GROWL toolkit is still a working prototype and is currently being used by Daresbury and Lancaster in the OGSA e-Social Science pilot demonstrator project. A number of other prototype lightweight Grid toolkits are being developed worldwide and one brief of this project is to monitor this activity and contribute to any development of standards. In particular we cite gLite from the EU EGEE project (<http://glite.web.cern.ch/glite/>) and implementations of specifications developed in the GGF SAGA and DRMAA research groups (<https://forge.gridforum.org/projects/saga-rg/> and <http://www.drmaa.org/>). This project will be represented at SAGA workshops. Wrappers to services supported by the UK's Open Middleware Infrastructure Institute (OMII: <http://www.omii.org.uk/>) will be addressed as these emerge and are taken up on the UK Grid.

GROWL provides scientists who have a limited interest in and knowledge of computing with an easy vehicle to improve their access to computational power. For the first time it will be within the grasp of many social scientists to massively increase the computational power from their within their existing applications and models. This will make it feasible, cheap and efficient, to increase the size and complexity of models, the amount of data processed and to achieve results in a much reduced time frame. GROWL will provide tools to increase the effectiveness of large scale research and the efficiency of the researchers involved, while at the same time reducing costs by allowing continued use of existing often expensive applications by using plug-ins. Thereby GROWL will enable many quantitative social scientists to make the step to using e-Science technology to solve their problems.

## Acknowledgements

We acknowledge receipt of a grant from JISC under the VRE Programme to develop the GROWL toolkit. The prototype was developed at the CCLRC e-Science Centre with funding from OST. We acknowledge funding for related projects which will be either using GROWL or contributing services for e-Social Science as follows: CQeSS (ESRC), OGSA (ESRC), NCeSS (ESRC).

## References

- [1] JISC Virtual Research Environments (VRE) Programme: [http://www.jisc.ac.uk/index.cfm?name=programme\\_vre](http://www.jisc.ac.uk/index.cfm?name=programme_vre)
- [2] J. Chin and P.V. Coveney *Plea for Lightweight Middleware* <http://www.realitygrid.org/lgpaper.html>
- [3] R.P. Gabriel *The Rise of "Worse is Better"* <http://www.jwz.org/doc/worse-is-better.html>
- [4] GROWL project Web site: <http://www.growl.org.uk>
- [5] SABRE: <http://www.cas.lancs.ac.uk/software/sabre/sabre.html>
- [6] HSL: <http://hsl.rl.ac.uk/contentshsl2004.html>
- [7] CQeSS: <http://cqess.lancs.ac.uk>
- [8] *End-User Development*. Special Issue of the Communications of the ACM, September 2004, Volume 47, Number 9
- [9] J.D. Herbsleb and R.E. Grinter *Architectures, Coordination and Distance: Conway's Law and Beyond* (IEEE Software , September-October 1999) pp63-70

- [10] J.M. Abowd and R.H. Creecy *Computing Person and Firm Effects Using Linked Longitudinal Employer-Employee Data*, <http://instruct1.cit.cornell.edu/~jma7/abowd-creecy-kramarz-computation.pdf>
- [11] J.J. Heckman *Statistical Models for Discrete Panel Data*, in Manski, C.F. & McFadden, D, (eds.), *Structural analysis of discrete data with econometric applications*, (MIT Press, Cambridge, Mass. 1981)
- [12] J.J. Heckman *The Incidental Parameters Problem and the Problem of Initial Conditions in estimating a Discrete Time-discrete Data Stochastic Process*, In Manski, C.F. & McFadden, D, (eds), *Structural analysis of discrete data with econometric applications*, (MIT Press, Cambridge, Mass. 1981)
- [13] D. Stott, D.M. Berridge and D.M. Dos Santos *The ORDINAL command in SABRE Release 3.1 (1981)*, <http://www.cas.lancs.ac.uk/software/sabre3.1/sabre.html>
- [14] P. Davis *Estimating Multi-way Error Components Models with unbalanced Data Structure*, *Journal of Econometrics* (1982) 106, pp67-95.
- [15] *Writing R Extensions* (The R project) <http://cran.r-project.org/doc/manuals/R-exts.html>
- [16] R.F. van Englen *The gSOAP Stub and Skeleton Compiler for C and C++* (Florida State University, 2002). Available as HTML or PDF with gSOAP distribution.
- [17] Stata (<http://www.stata.com/>)
- [18] SRB (<http://www.sdsc.edu/DICE/>)
- [19] Condor (<http://www.cs.wisc.edu/condor/>)
- [20] Globus (<http://www.globus.org/>).