18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Implementing a large scale social simulation using the New Zealand BeSTGRID computer network: A case study

Walker, L.K. [1]

[1] *Department of Statistics and Department of Sociology, The University of Auckland, Auckland, New Zealand*
Email: lk.walker@auckland.ac.nz

The Modelling Social Change (MoSC) project is a large scale social simulation project which investigates changes in the social structure of New Zealand by simulating inter-ethnic cohabitation patterns using models populated with unit-level census data. This paper is presented as a case study of the implementation of this simulation project on a high power cluster within the New Zealand BeSTGRID (www.bestgrid.org) computer network by a researcher with a social science (rather than computer science) background. The aims of this paper are two-fold. Firstly, it aims to encourage other researchers to investigate the potential of implementing their own research in a grid based environment by highlighting the benefits that were gained from using grid computing. Secondly, it aims to inform the managers and systems administrators of grid-based systems of the some of the difficulties which a new user may face and how they may be of more assistance in introducing new researchers to the technology.

The MoSC research project applies computer based social simulation techniques to cohabitation and demographic data from the New Zealand Census in order to test models of New Zealand's social structure in the rapidly changing demographic and economic conditions of the period 1981 to 2006. The central research question of the project focuses on whether the social structure of partnerships in New Zealand, as reflected in the distribution of inter-ethnic marriages and the choice of cohabitation partners, has became more highly stratified and segregated over this period. In addition it also examines what factors and/or social processes have lead to the variability (or lack of variability) in the distribution of these inter-ethnic partnerships through the sensitivity testing of the simulation model and the use of "feedback loops" to model recursive social processes.

The simulation was written in Java and run on the Auckland cluster of the BeSTGRID computer network; a system with 80 CPUs, 160 Gigabytes of memory and 2500 Gigabytes of hard drive space. The processing power of the cluster allowed the simulation to be run at a city level, with unit data that provided demographic information for all of the single eighteen to thirty year olds listed in the census in the Auckland, Wellington and Canterbury regions. The use of such a large population dataset represents a huge advance over most social simulation experiments that tend to be run using small samples of data. The cluster also allowed for the parallel processing of code which provided the opportunity to run an evolutionary optimisation algorithm across the parameter space in order to find optimal combinations of the partnering parameters.

The implementation of the simulation on the grid wasn't completely trouble-free. Being a "guinea pig" (a new user with a project that was very different from the other users) created some frustration as many of the systems and processes for using the grid were not set up with social science users in mind. Stepping up from basic Java programming on a single machine to parallel processing on a complex system was a steep learning curve and increased the reliance of the project on the technical staff who look after the cluster.

Overall, the application of a large scale social simulation to grid technology has been a positive one. It has allowed the creation of simulation models on a much larger scale than are typically seen in the area of social simulation and has opened the door for future social science research using grid technology.

Keywords: *Grid computing, social simulation, partnership matching, BeSTGRID network, census data*

1. INTRODUCTION

Social simulation is a different paradigm from many other forms of simulation. Although it can be used for prediction, the main focus of the technique is to capture the underlying social processes by which social phenomena are occurring rather than simply reflecting the transition probabilities of the various states of the model (Gilbert & Troitzsch, 2005). The Modelling Social Change (MoSC) project is a social simulation project examining patterns of inter-ethnic cohabitation in New Zealand.

This paper will describe the philosophy behind the MoSC research and the evolution of the computational requirements of the project. A brief description of the methodology will be followed by some details on the experience of a social science researcher running simulation models on a grid-based system. It will highlight some of the benefits of the technology, such as the hugely increased processing speed, the ability to run processes in parallel and the programming support that was provided. It will also describe some of the difficulties encountered during the research process such as dealing with the new technology and a system that was designed with computer science (rather than social science) users in mind.

There are two main aims to this paper. The first is to provide some details about the MoSC project and the benefits of using cluster/grid computing resources in order to encourage other researchers to explore how this technology could benefit their own research. Whilst some disciplines such as bioinformatics have been very quick to embrace this technology, feedback from the grid administrators suggests that in some disciplines, resources are being under-utilised simply because of an unwillingness of researchers to experiment with the new technology. The second goal of this paper is aimed at the managers and systems administrators of cluster and grid systems. As a new user with a social science background rather than a computer science one, I found that I faced a steep learning curve in order to utilise the technology for my research. I hope that by describing my experiences, these stakeholders will gain a greater understanding of the perspective of a new user.

2. THE MODELLING SOCIAL CHANGE PROJECT

The MoSC project applies social simulation techniques to New Zealand census data in order to test a model of New Zealand's social structure through the examination of patterns of inter-ethnic cohabitation between 1981 and 2006. The central research question is whether the social structure – as reflected in the distribution of matching socio economic and ethnic - choices of co-habitation partner across households – became more highly stratified and segregated over this period. The census provides data on these dimensions of social stratification that are both fully representative and available at five-yearly intervals.

Inter-ethnic cohabitation is of interest because it demonstrates the existence of interaction across ethnic boundaries and indicates that members of different ethnic groups consider each other to be social equals (Kalmijn, 1998). A follow on effect of inter-ethnic marriage is that children of mixed ethnicity couples are less likely to define themselves within a single ethnic group, further reducing cultural distinctions and social boundaries between the groups. This can become a recursive process, with the partnership decisions of one generation helping to influence the next. Hypothetically, as the rate of inter-ethnic partnership increases it helps to normalize it, which will in turn see it become more prevalent. In addition, inter-ethnic partnerships may also be an indicator of a lack of availability of potential partners of the same ethnicity (Blau, 1977). These recursive properties are not easily modelled by traditional statistical analysis but have the potential to be examined by social simulation.

Gaining an understanding of the social processes that drive partnership choice help to provide an insight into the way in which society is shaped. Descriptive statistics, log-linear models and logistic regression provide some information about partnering patterns but they do not allow for the examination of possible phenomena such as emergent properties and micro-macro links in the behaviour of individuals. Social simulation helps to fill in these gaps by modelling sociological rules and recursive relationships in order to provide a deeper understanding of the partnering process.

2.1. Goals of the Simulation Modelling

The key measures of interest for the study are the patterns of inter-ethnic cohabitation. These are modelled using non-ethnicity based micro level variables, observations by agents of their macro environment and a random stochastic component.

There are two main goals of the empirically based simulation model:

1. To find the weighted combination of factors that produces the most similar set of inter-ethnic cohabitation patterns to those that have actually occurred.
2. To examine the effect on inter-ethnic cohabitation patterns of varying the relative importance of the four factors.

The first goal focuses on the accurate prediction of the patterns of inter-ethnic partnership. It uses an evolutionary optimisation method to find the combination of factors that produce similar results to what was observed in the actual census cohorts. Since each census is treated as a separate cohort, five sets of weights are produced for each of the three regions. These weights are observed independently and also as a collective to examine whether there are any time dependent trends occurring.

By comparison, the second goal examines some "what if" scenarios, examining some sets of parameter weights such as a one hundred percent weighting on a single factor to see what patterns of ethnicity occur in these circumstances. Although these scenarios don't provide as much explanatory power, they are of sociological interest and also provide a form of sensitivity testing for the model.

Beyond examining the patterns in weights, the other phenomenon interest is whether a relationship between micro level behaviour and macro level patterns can be observed. This is where the decisions of micro-level agents are influenced by macro level observation. When these decisions are collated they form the macro level at the next time step, which will then impact on the decisions of the following set of micro agents.

This simulation model provides the opportunity to examine abstract and empirical sociological models of partnership choice and examine social mechanisms such as emergence and downward causation. These mechanisms focus on the linkages between the micro-level and macro-level, where changes in the behaviour of the agents (micro-level), impact on the society as a whole (macro-level), which in turn impact on the decision of future individual agents in the system.

2.2. Census Data

The simulation is populated with unit level data from the New Zealand Census of Population and Dwellings in order to create a simulation environment which closely resembles the three different regions of New Zealand at each census time point. Due to privacy considerations, the data has a limited level of detail for each individual agent. However, the agents in the simulation have not been reverse-engineered from frequency tables in order to match marginal distributions; they are the actual unit level records for the sub-population that is being studied.

The agents in the model represent all of the single eighteen to thirty year olds from the Auckland, Wellington and Canterbury territorial regions for each five year census period from 1981 to 2001 (the 2006 data is available but is not used as a simulation input since it does not have a subsequent census to benchmark against for the evolutionary algorithm). These regions were chosen for two reasons. Firstly, they are three of the largest territorial regions, which alleviated some of the privacy concerns associated with the use of unit level data. Secondly, the three regions represent three different levels of ethnic diversity, from the high level of ethnic diversity in Auckland to the lower levels in Canterbury and Wellington.

The age grouping of eighteen to thirty year olds was used in order to create a fixed cohort that could be compared from one census to the next. The eighteen to thirty year old group at one census would become the twenty three to thirty five year old group at the next one, allowing inter-census comparisons to be made and validation of the results to be performed.

2.3. The Simulation Routine and the Evolutionary Algorithm

The simulation routine uses a matching algorithm where the agents form social networks of the opposite sex. At each time step, they will examine the agents in their network and allocate each a score. Starting from the highest score, the *n* highest ranked couples are paired off and removed from the system, where *n* is equal to the total number of actual partnerships over that census period divided by the number of time steps. It is based around the DYNASIM (Zedlewski, 1990) and APPSIM (Bacon & Pennec, 2007) models, although also borrows facets from other published studies. The scoring mechanism that is used to match the agents uses a combination of age similarity, educational similarity, a macro-based measure of the proportion of inter-ethnic couples who had formed in the last time step and a random stochastic component. The simulation was run with different weights providing differing levels of importance for each of the four

components. In addition, an evolutionary algorithm was applied to the simulation in order to try to find optimal combinations of the weights.

A ($\mu + \lambda$) evolutionary algorithm (Luna *et.al.*, 2008) is employed to search for the optimal set of weights for each census dataset. Starting from a random set of weights, the algorithm creates multiple sets of nearby (perturbed) weights and then simultaneously runs the simulation with each. The algorithm finds the weights combination with the lowest total squared deviance of partnership frequencies relative to the actual census and then uses that as the starting point for the next perturbed set of weights. The process repeats until no further movement would result in better predictions or a maximum number of iterations are reached. This method produces a more efficient search of the parameter space for the optimal set of weights for the scoring mechanism than relying on a brute force sweep of all of the possible combinations.

The simulation programme itself is approximately 600 lines of Java code, with 120 lines of XML and Unix code for the evolutionary algorithm and a further 370 lines of XML and Unix code required to schedule the simulation on the Grid and run the simulation multiple times in parallel on different cores.

2.4. The BeSTGRID Network

"Grid computing can be defined as coordinated resource sharing and problem solving in dynamic, multi-institutional collaborations. More simply, Grid computing typically involves using many resources (compute, data, I/O, instruments, etc) to solve a single, large problem that could not be performed on any one resource. (Nabrzyski, Schopf, & Weglarz, 2004)

Simulating the interactions between hundreds of thousands of agents requires a significant amount of computing power. Although early simulations were run on a desktop PC and then on a high end server, the Auckland cluster of the BeSTGRID computer network provided the high end computing power required to run the census based simulations. With access to multiple processors the stand alone simulations are supplemented with a parallel optimisation routine that searches across multiple sets of weights for the scoring routine in order to search for an optimal combination.

Moving to a Grid based computer system provided two key advantages over working with a single machine. Firstly, even when working with a single core (CPU) of the Grid there was a significant increase in performance relative to the desktop PC and the server for running the equivalent set of code. Secondly, and more importantly, the multiple processors of the Grid allowed for parallel processing. This is where code can be split up and run across a number of CPUs in order to provide reduced running times and increased computational efficiency (Parry, Evans, & Heppenstall, 2006). In the case of this research, it meant that multiple simulations could be run simultaneously through an evolutionary algorithm in order to efficiently search for optimal sets of weights.

The BeSTGRID computer network (https://www.bestgrid.org) is a national eResearch project which was started as a Tertiary Education Commission funded project. It includes shared computational resources made up of powerful computing clusters at several New Zealand universities including the University of Auckland. The Auckland cluster features five systems of two nodes, each of which is powered by two quad core Xeon 2.8 GHz processors, providing a total of 80 cores, together with 250 gigabytes of disk space and 16 gigabytes of memory for each node. The 80 cores allow for appropriately written code to be processed in parallel, greatly improving computational performance.

The grid was accessed using a secure shell client (http://www.ssh.com) and secure FTP. A proxy was set via the Grisu client (http://grisu.arcs.org.au/downloads/beta/webstart/) so that the grid could be accessed from multiple machines. The simulations were written in Java and operated via JDK 1.5 on the Auckland cluster of the grid via XML and Unix scripts.

3. THE BENEFITS OF GRID COMPUTING

Implementing the MoSC simulations on a grid-based system has reaped a number of benefits. The biggest benefit was the increased processing power and the ability run code in parallel. However, there have been other benefits gained from experimenting with this technology. One unforeseen benefit was the ideas and programming advice of the programming and administrative staff of the grid. They quickly became a valuable resource and their collaboration helped to enhance the project far beyond what was originally proposed. The experience has also provided new skills for several researchers and opened up further research possibilities using the grid.

3.1. Processing Power and Parallelisation

The grid was able to provide high power processing across up to eighty cores (CPUs). Even when running code on a single core, jobs would run more quickly than on a standalone desktop machine. Single runs of the simulation code were able to be compiled and executed up to fifty percent faster using a core of the cluster than they were on a mid-level laptop, even before parallel processing. The efficiency became even greater once the simulation process was set up to run multiple jobs in parallel. A single run of the simulation run would take between ten minutes and an hour on a laptop, depending on which dataset was used as the input. This was reduced to approximately six minutes for the smallest dataset through to about twenty five minutes for the largest. These improvements were in part due to the faster processors and in part thanks to some coding improvements suggested by the BeSTGRID staff (see Section 3.2 for further details).

The simulation was initially written using threading to take advantage of parallel processing; however, this had to be abandoned because the final algorithm required an evaluation of pairings that had to be conducted in order and therefore couldn't be split into separate threads. Murata *et.al.* (2004) and Arikawa and Murata (2007) are amongst numerous researchers who have extolled the virtues of parallel processing. They were able to find optimal solutions to complex problems significantly more quickly and efficiently using grid technology. In this study, the evolutionary algorithm that is used to find optimal combinations weights in the scoring mechanism of the simulation could not have been run without the ability to simultaneously evaluate different sets of parameters at each step of the process. This allowed for convergence to optimal solutions rather than relying on a brute force parameter sweep.

The use of the Grid allowed different combinations of parameters to be tested simultaneously and the ability to use the evolutionary algorithm. On a single machine, the evolutionary algorithm would have involved running 245 simulations one after the other, whereas with the Grid, this is brought down to ten runs of seventeen perturbed parameter combinations followed by five runs of fifteen perturbed parameter combinations, turning a process that would take days into one that took hours instead.

3.2. Programming Support

Moving from running the Java code on a single machine to running it on the grid required the ability to write parallel processing code, Unix and XML scripts and work remotely through an array of proxy servers and other security measures. The system administrators and programmers in the grid team were able to provide programming support and guidance in order to get the simulations loaded onto the grid and running quickly and securely. They were able to provide suggestions and feedback to the research group which helped to refine the simulation code and algorithms. At the time that this project started the Grid infrastructure was being underutilised, so programming assistance was very forthcoming as the staff sought to increase the number of users of the Grid, and in particular help new researchers to exploit the technology.

The XML scripting that was required for the evolutionary algorithm was largely written by these staff. It was comprised of approximately 370 lines of XML and Unix code that would load jobs and run them in parallel and a further 120 lines of code for the evolutionary algorithm. It meant that the full power of parallel processing could be utilised for finding optimal scoring weights. In addition to the more general assistance, it was the coding of these files that accelerated the progress of the research project and allowed me to make the most of the technology.

3.3. New skills and the potential for future research

In addition to benefiting the current research, the introduction to the grid has provided professional development in terms of the new skills and knowledge that have been acquired during the project. These have included programming skills, remote data management and dealing with remote hardware/software interfaces. The ability to use grid technology has also opened up new avenues of research for the MoSC project members. Having seen the processing power that is available via grid-based computing, team members have initiated a number of new simulation projects, including large scale simulations in residential segregation and national healthcare.

4. DIFFICULTIES ENCOUNTERED

Implementing the social simulation routines on the Auckland cluster has not been without some difficulties. The complexity of the system and the fact that some of the technology had not been thoroughly tested meant that things did not always run smoothly. The increased reliance on support staff was an unfamiliar and

slightly uncomfortable feeling for researchers who are used to working independently. One final difficulty, although it is one that is faced by all grid users, was the competition for computational resources at certain times.

4.1. Complexity

In order to log into the BeSTGRID system and run some Java code, a user needs to set up a proxy server which is done via software located on a machine in Australia which needs to be accessed remotely. They then need to log into a grid-client machine via secure-shell software, log into the proxy server and then use a combination of XML and Unix scripts to copy the Java file from the client machine to the grid machine and execute it. In order to run the code in parallel a further and more complex XML script is required. For a researcher who is new to this kind of technology, using a grid can be a complex and confusing process. Even with Java programming experience and a rudimentary knowledge of Unix, the learning curve for using the system was steep.

Having not come from a computer science background and therefore not possessing sufficient Unix and XML skills to be able to execute all of the grid related tasks that I required created a reliance on the grid staff. Many of the Unix and XML scripts for the parallel processing and evolutionary algorithms were written by the grid programming staff. They also provided ongoing assistance with Unix, proxy server problems and other troubleshooting. Thankfully there was low demand for their services at the time so they were able to provide me with a high level of assistance.

I trialled a primitive GUI which was put forward as an alternative interface for users with limited Unix experience. Unfortunately it was a beta version and still had a number of bugs, including not being able to run Java. This was one of a number of occasions where I was asked to trial new (generally beta) software for the grid to get a gauge of how user-friendly it was. Although I could see the benefit to my work and to future users, it did leave me feeling somewhat like a beta software "guinea pig".

4.2. Life as a computing "guinea pig"

The MoSC project was the first social science project to use the Auckland cluster. Prior to that, most of the users were from bioinformatics or computer science. Many of the bioinformatics users work on genetics research projects and typically run a few large jobs over long periods of time rather than the shorter repeated jobs seen in this project. Beyond this, the social simulation paradigm can be quite different to that of the more traditional scientific researchers, with more focus on sociological processes rather than a specific outcome such as a genome mapping. The differences in paradigm, coupled with the differences in computing experience provided new challenges to the systems administrators and at times led to some situations that administrators had not previously had to deal with.

One such example of this was a "ghost in the machine" situation. A simulation process that was not stuck in an infinite loop did not remove itself from the core that it was using at the conclusion of the simulation run. It could not be destroyed by the user or by the administrators and sat on the grid for a number of weeks before mysteriously disappearing. It was a problem that had not been encountered before and the staff were at a loss to explain it.

4.3. Competition for resources

The Auckland cluster of BeSTGRID has eighty available CPUs but individual users will often take advantage of parallel processing and use anywhere up to twenty or thirty cores. As the system is fairly new and the user-base is still growing there are currently no formal systems in place for scheduling or sharing CPU usage. Most of the time this is not a problem as there is sufficient slack in the system for the current level of usage, but if too many users schedule jobs without first checking on the current usage, traffic jams can occur. Schwiegelshohn and Yahyapour (2004) and Kokkinos *et.al.* (2008) highlight the importance of scheduling grid jobs in order to best utilise finite grid processing resources. As the demand for the resources grow, scheduling and sharing will need to be thought of in order to provide an equitable allocation of resources for all of the users.

5. CONCLUSIONS

High power computing systems are a costly capital investment so research institutions so they should be set up to get provide as much benefit as possible. The Auckland grid staff provided excellent assistance for the

MoSC project although they did find that working with social science researchers provided a different set of challenges from their normal users. It is important for systems administrators to be aware of the diversity of their users, particularly when they are deciding on user interfaces and operational procedures such as the level of general and specific computer skills that are needed to perform tasks using their grid. Intuitive processes and clear documentation help to make life easier for both parties and lead to more productive research.

Using the Auckland cluster of the BeSTGRID computer network or large scale social simulation has had benefits and pitfalls. Overall, the implementation of this social simulation project on a grid system has been a positive experience. It has provided the ability to use large datasets and computationally intensive algorithms and optimisation methods. The added computational power and parallel processing has enhanced one research project and provided the inspiration for several more.

Hopefully other social science researchers will take inspiration from this article and think about how their own research may be adapted to this kind of computational resource. In addition, I hope that the comments about the difficulties encountered will help to guide grid system administrators in dealing with new users who don't have a strong background in computer science.

ACKNOWLEDGMENTS

The author wishes to acknowledge funding from the Royal Society Marsden Fund and the co-operation of Statistics New Zealand in the provision of the data. The results presented in this study are the work of the author, not Statistics New Zealand and access to the data used in this study was provided by Statistics New Zealand in a secure environment designed to give effect to the confidentiality provisions of the Statistics Act 1975. In addition, the author would like to acknowledge all of the assistance provided by the University of Auckland BeSTGRID team in utilizing their computational resources.

REFERENCES

Arikawa, H., & Murata, T. (2007). Implementation issues in a grid-based multi-agent simulation system used for increasing labor supply. *The Review of Socionetwork Strategies, 1*(1), 1-13.

Bacon, B., & Pennec, S. (2007). APPSIM - Modelling Family Formation and Dissolution [Electronic Version], *Working Paper No. 4* from http://www.canberra.edu.au/centres/natsem/publications?sq_content_src=%2BdXJsPWh0dHAlM0ElMkYlMkZhbmltYWwuY2FuYmVycmEuZWR1LmF1JTNBNTgwJTJGbmF0c2VtJTJGaW5kZXgucGhwJTNGbW9kZSUzRHB1YmxpY2F0aW9uJTI2cHVibGljYXRpb24lM0QxMDQ5JmFsbD0x

Blau, P. M. (1977). *Inequality and heterogeneity: A primitive theory of social structure*. New York: Free Press.

Gilbert, N., & Troitzsch, K. G. (2005). *Simulation for the Social Scientist*. Maidenhead: Open University Press.

Kalmijn, M. (1998). Intermarriage and homogamy: Causes, patterns, and trends. *Annual Review of Sociology, 24*(1), 395-421.

Kokkinos, P., Christodoulopoulos, K., Doulamis, N., & Varvarigos, E. (2008). Quality of service scheduling of computation and communicaiton resources in grid networks. In J. Wong (Ed.), *Grid Computing Research Progress* (pp. 91-150). New York: Nova Science Publishers Inc.

Luna, F., Alba, E., Nebro, A. J., Mauroy, P., & Pedraza, S. (2008). Applying evolutionary algorithms to solve the automatic frequency planning problem. In A. Nayak & I. Stojmenovic (Eds.), *Handbook of Applied Algorithms* (pp. 271-286). New Jersey: John Wiley & Sons.

Murata, T., Kitano, H., Kadooka, Y., & Ukai, Y. (2004). Political multi-agent simulation with Grid computing [Electronic Version]. *RCSS Discussion Paper Series, 17* from http://www.rcss.kansai-u.ac.jp/DPS/pdf/dp017.pdf.

Nabrzyski, J., Schopf, J. M., & Weglarz, J. (Eds.). (2004). *Grid resource management: State of the art and future trends*. Norwell, MA: Kluwer Academic Publishers.

Parry, H. R., Evans, A. J., & Heppenstall, A. J. (2006). *Millions of agents: Parallel simulations with the RePast agent-based toolkit.* Paper presented at the Austrian Society for Cybernetic Studies, Vienna.

Schwiegelshohn, U., & Yahyapour, R. (2004). Attributes for communication between grid scheduling instances. In J. Nabrzyski, J. M. Schopf & J. Weglarz (Eds.), *Grid Resource Management* (pp. 41-52). Boston: Kluwer Academic Publishers.

Zedlewski, S., R. (1990). The development of the dynamic simulation of income model (DYNASIM). In G. H. Lewis & R. C. Michel (Eds.), *Microsimulation Techniques for Tax and Transfer Analysis*. Washington, DC: Urban Institute Press.