Agent-Based Modeling Supporting the Migration of Registry Systems to Grid Based Architectures

Martin E. Cryer, BSc (Hons), BS1, Lewis Frey, PhD1,2
1University of Utah, Salt Lake City, UT
2Huntsman Cancer Institute, University of Utah, Salt Lake City, UT

Abstract
With the increasing age and cost of operation of the existing NCI SEER platform core technologies, such essential resources in the fight against cancer as these will eventually have to be migrated to Grid based systems. In order to model this migration, a simulation is proposed based upon an agent modeling technology. This modeling technique allows for simulation of complex and distributed services provided by a large scale Grid computing platform such as the caBIG™ project’s caGRID. In order to investigate such a migration to a Grid based platform technology, this paper proposes using agent-based modeling simulations to predict the performance of current and Grid configurations of the NCI SEER system integrated with the existing translational opportunities afforded by caGRID. The model illustrates how the use of Grid technology can potentially improve system response time as systems under test are scaled. In modeling SEER nodes accessing multiple registry silos, we show that the performance of SEER applications re-implemented in a Grid native manner exhibits a nearly constant user response time with increasing numbers of distributed registry silos, compared with the current application architecture which exhibits a linear increase in response time for increasing numbers of silos.

Introduction
With an agent-based model it is possible to reasonably and accurately model distributed computer system architectures such as exhibited with Grid computing and client-server system platforms[1, 2]. With such models it is also possible to illustrate scaling effects for various distributed computer system architectures as a result of making topology changes to a Grid computing network, for instance by adding additional database silos to the system.

In the healthcare field existing translational registry silos are a valuable resource, likely to significantly contribute to the future success of personalized medicine initiatives. Existing translational registry silos are implemented with client-server system architectures, such as the Surveillance, Epidemiology and End Results (SEER) registries[3]. The registries provide data regarding cancer incidence, survival and mortality in the US. More recent trends have been to provide database silos based within a Grid computing architecture system, such as provided by the caBIG™ project. Grid systems make both database and analysis software available in a distributed manner across a number of loosely coupled nodes, interfacing via a number of well-defined, semantically equivalent, software services.

In the existing, or legacy, client-server architecture systems, translational registry silos cannot be readily utilized in a standalone or isolated manner due to issues of security, semantic translation and extensive requirements for application interfacing between client and server systems. Such resources in the future should be readily available to multiple translational research teams of investigators in a secure and federated structure, as provided by the caBIG™ Grid system. The data storage requirements of translational registry data silos are under tremendous growth pressure with the incorporation of not only expanding genomic data volumes, but also with the inclusion of scanning imaging data and its extremely large data set sizes. The pressure on legacy client-server architecture derives not only from requirements for more widespread multi-project access in a semantically standardized manner, but also from significantly increased data volumes. Grid based systems such as caBIG™ not only provide a potential architectural solution to the distributed data access requirements of registry data silos, but for the future translational research requirements of personalized medicine[4].

Agent-based models of potential system implementations will allow legacy database based project architects to make informed decisions regarding registry data silo expansion and adoption. Designers can choose as whether to augment their existing systems under existing architectures for the short term, or move directly to caBIG™ based Grid systems, based upon model predicted future requirements for performance, computational services and increased repository capacity. The modeling of potential target systems will contribute to risk reduction in planning such migrations.

This paper investigates potential scenarios for migrating to a series of distributed registry data silos.
The first scenario models a legacy client server system that has been migrated to a single Grid node and the original client-server applications expanded to retrieve data from a variety of additional databases on other Grid nodes, via Grid level services. It is assumed in this model that this application migration is very primitive in nature and the application retrieves data in a sequential manner from remote registry data silo Grid nodes. This simulation would represent the performance of an application port from a legacy client server system to a Grid without making fundamental changes to the application architecture yet still allowing the benefit of being able to perform more complex translational analysis.

The second scenario provides for a more complex adaption or re-write of the applications and analysis software to take true advantage of the parallel access technology and full semantic equivalence available from the caBIG™ Grid. The caBIG™ project’s development tools and runtime environments allow for the building of applications that result in Grid certified applications and services[5] ensuring a high degree of reliable interchange of data. In this scenario, each Grid node supports some component of the analysis software, coordinated by the client workstation and associated Grid workflow software[6, 7]. Each Grid node also supports a separate registry data silo utilized by the translational analysis application software. The Grid nodes hosting the analysis software components and the database silos may, or may not, be collocated. In this model it is assumed that analysis applications and registry data silos are hosted on unique Grid nodes. The Grid platform has a flexible and extensible architecture, allowing for significant increases in capacity and performance for more linear increases in cost.

Although there is prior work on the integration of Grid based and agent-based systems[8], as well as efforts on various modeling techniques for certain characteristics of Grid based systems[9], a literature review has shown little work in using agent-based modeling techniques to simulate Grid performance.

**The caGRID and SEER Platforms**

The caGRID platform provides for a Grid wide common security infrastructure. However, within this Grid wide security mechanism, security access for locally owned resources are locally controlled and maintained. Therefore, local resources and services can be secured for local access only; access has to be explicitly granted by the owner of a resource to other Grid users. The Grid requires data services to present data to service clients according to a standardized data definition enabling a universal understanding of the information being transferred. The caGRID services were designed to provide data storage and searching services, along with distributed analytical services, for data sets not dissimilar to those used within NCI SEER systems. NCI SEER systems could be potentially readily ported to caBIG™ Grid platforms for future project expansion as illustrated in, Figure 1.

A newly introduced caGRID application service, such as that presented by a hypothetical integration of the NCI SEER system to a caGRID platform would make use of a number of caGRID services, including the security services components, at a minimum. Additionally, services such as the workflow service would likely be integrated into the application as the functionality of the NCI SEER client applications were expanded over time.
computational analysis in order to achieve the required levels of performance. The system must become flexible enough to move the statistical analysis of the cancer public health data from desktop client systems to larger, more powerful parallel systems comprising of multiple Grid nodes.

**Methodology**

During the model execution, co-incidental workflow within the simulation is monitored in order to calculate increased user response time latencies due to the multiple workflow threads attempting to simultaneously access common or shared resources within the system. Such delays are calculated from the interaction between agents in the model representing the functionality of logical software and hardware subsystems and the agents representing the workflow of the application itself. In Figure 2, the interaction between two application workflows and a shared Grid node can be observed as two workflow agents interact spatially and temporally with the agents representing the Grid node’s environment.

![Diagram](image)  
**Figure 2:** Workflow agents and node agents determine delay upon coincidental workflows contending for resources.

Each node along with its interconnect fabric, is modeled using a number of agents to represent the functionality and performance of the node and its network traffic to other nodes. Such a representation can model not only native Grid nodes but also NCI SEER nodes as ported to the caGRID network. At the highest level within each node being modeled is a group of agents responsible for the simulation of message requests from other nodes to handle remote connection requests. The model passes client requests from other systems to the Grid Application agents for processing. These agents in turn utilize Grid Remote Resource Services [1, 10] agents to obtain valid connections to the required targets servers.

The modeling engine used for the various simulations performed was the AnyLogic™ modeling package supplied by XJ Technologies[11]. This modeling package comprises of an interactive graphical tool for creating models of agents with their associated state diagrams. The model run time engine provides a number of inbuilt statistical analysis functions and distributions as well as output graph generation.

The model is executed in multiple passes, for each of the two defined scenarios, each set of passes for a varying number of Grid nodes. The first scenario comprises a straightforward migration of a legacy registry silo system to a single caBIG™ Grid node. The application has been modified to allow for a number of remote Grid node registry data silos to be sequentially accessed by a simple single threaded analysis application. The second scenario comprises a re-architected application based around one or more Grid nodes and one or more registry silos, utilizing all the features available from an application built using the caBIG™ development toolkits. For a single node, both scenarios should provide roughly the same results, as they are functionally equivalent for such a configuration. As a client-server system is logically equivalent to a Grid node client system with a single analysis application Grid node and a single registry data silo Grid node, it is assumed the client-server and Grid architectures have roughly equivalent single node performance in this configuration.

![Diagram](image)  
**Figure 3:** AnyLogic(tm) state engine diagram for a simple Grid node.
The model is repeatedly executed for both scenarios, with a varying number of registry data silos in turn. In this simulation, 1, 3, 5 and 10 database nodes are utilized. Each of the node components is represented internally within the model as agents, each with a unique state engine. There is a state engine for each type of component within the system and the state engine is replicated for each occurrence of a particular node type. In this model, there are agents representing the application clients that initialize and control the system, the Grid node servers which obtain data items from the various registry data silos, the access control server which handles security administration and the database servers which house the registry data silo services. Agents are programmed utilizing a flow chart type style mechanism, as illustrated in Figure 3, where the block components represent agent states and the flow lines between blocks represent state transitions[12]. State transitions can be dependent upon timeouts, variables, messages and other randomized events.

Results

The results were collected from a number of model execution runs, with a combination of configurations and numbers of registry data silo nodes. The mean user response time for each of two scenarios (a single legacy ported application node versus a native Grid application) was recorded against four differing Grid registry data silo node topologies (1, 3, 5 and 10 node configurations). The dependent variable is the transaction duration time. The independent variables are the scenario under test and the registry data silo configurations. Graphing the raw data for each of the two scenarios results in a response time or transaction duration graph for each of the 1, 3, 5 and 10 registry data silo configurations. The results for both scenarios are illustrated in Figure 4 and Figure 5. In Figure 4, it can be seen that the average response time for the scenario of a single node supporting a minimal port of a legacy application coupled with sequential access to 3 registry data silos or database nodes, is approximately 17 seconds. This contrasts with the application response time for the scenario supporting a Grid native version of the application retrieving data from 3 registry data silos in parallel, being approximately 10 seconds, in Figure 5. The horizontal line within both graphs denotes the average response time.

Plotting the mean values for user response time duration against each scenario of registry data silos, we can combine the graphs to display the likely latency times depending upon the application configuration, as in Figure 6. The dependent variable is the mean transaction duration. The independent variables are the scenario under test and the registry data silo configuration. The increased number of sample points in the second scenario, Figure 4 versus Figure 5, is due to the fact the data were recovered with a fixed duration model and not for a fixed number of events, the faster scenario merely processed more events.

Discussion

It can be seen from Figure 6, scenario 1, that the user response time for this system decays linearly, due to the fact that application recovers data from a number of Grid-based databases serially, before a single threaded analysis of the data at the end of the data
recovery phase. This would be a typical result for an existing application being enhanced to obtain additional data from remote translational registry silos without re-architecting the entire legacy system, hence using a serial data recovery technique. In scenario 2, an approximation of the application provided as a native Grid application in a caBIG™ Grid environment, the databases are accessed and analyzed in parallel, hence the user response time remains consistent across increasing numbers of database nodes. Realistically, such levels of gain over the reconfiguration of existing client-server systems to a single Grid node would not be as large as observed due to concurrent access to the database nodes from other unrelated client system nodes and the inability to completely parallelize the translational analysis functions. From the simulation results it is possible to project for what level of required future enhanced application performance is possible on an enhancement of the existing legacy platform, against the requirements for fundamental architectural change required in order to scale registry systems to utilize increasingly available translational resources.

Our results show that Grid based systems have a potentially greater headroom for future application load growth over existing single system registries. Modeling large-scale systems with agent-based modeling simulations could become a useful tool in supporting production Grid system architectural design verification. Future work will involve calibrating our model results against real world systems using performance data recorded from locally available caBIG™ Grid nodes incorporating Grid enabled versions of the SEER cancer registry.

Conclusion

Future NCI systems should be able to take advantage of direct data transmission from electronic patient records and associated personal electronic health record systems supported by clinicians, hospitals and clinics, for a reduction in media format conversion and manual data format translation costs from the NCI’s perspective. As clinicians and medical institutions become more integrated components of NCI’s systems, the cost burden for providing accurate cancer public health data will likely shift from the NCI to the data providers. Modeling systems have the potential to provide more accurate risk assessment, performance profile estimation and failsafe capability prediction, for such a migration to Grid based systems.

Acknowledgements

This project was funded by NLM training grant 1T15LM07124-10. The authors would also like to thank Dr. Antoinette Stroup of the Utah Cancer Registry, University of Utah, for her guidance and assistance. The Utah Cancer Registry is funded by contract N01-PC-35141 from the NCI with additional support from the Utah State Department of Health and the University of Utah. Dr. Lewis Frey is funded as the Principle Investigator of the NCI caBIG™ project at the University of Utah.

References

1. Cancer Biomedical Informatics Grid, caBIG(tm): caGRID 1.2. Cancer Biomedical Informatics Grid 2008 [cited 2008 10/24/2008]; Available from: https://cabig.nci.nih.gov/workspaces/Architecture/caGrid/
2. Dewire DT. Client/server computing. New York: McGraw-Hill; 1993.
3. The caBIG pilot phase : report, 2003-2007. [Washington, D.C.]: National Cancer Institute; 2007.
4. Molidor R, Sturm A, Maurer M, Trajanoski Z. New trends in bioinformatics: from genome sequence to personalized medicine. Experimental Gerontology. 2003;38(10):1031-6.
5. Cancer Biomedical Informatics Grid, caBIG(tm): Compatibility and Certification. Cancer Biomedical Informatics Grid 2008 [cited 2008 10/24/2008]; Available from: https://cabig.nci.nih.gov/guidelines_documentation/
6. Oinn T, Greenwood M, Addis M, et al. Taverna: lessons in creating a workflow environment for the life sciences. CONCURRENCY AND COMPUTATION. 2006;18(10):1067.
7. Phillips J, Chilukuri R, Fragos G, Warzel D, Covitz PA. The caCORE Software Development Kit: Streamlining construction of interoperable biomedical information services. BMC Med Inform Decis Mak. 2006;6(2):1472-6947.
8. Foster I, Jennings NR, Kesselman C. Brain Meets Brawn: Why Grid and Agents Need Each Other. 2004.
9. Bagchi S, Hung E, Iyengar A, Vogl N, Wadia N. Capacity planning tools for web and grid environments. 2006: ACM Press New York, NY, USA; 2006.
10. The Cancer Biomedical Informatics Grid (caBIG): infrastructure and applications for a worldwide research community. Stud Health Technol Inform. 2007;129(Pt 1):330-4.
11. Official AnyLogic Website. X. J. Technologies; 2007.
12. Wiguna R. Model-Driven Design using XJ Technologies AnyLogic.