

# KidneyGrid: A Grid Platform for Integration of Distributed Kidney Models and Resources

Xingchen Chu<sup>1</sup>, Andrew Lonie<sup>2</sup>, Peter Harris<sup>3</sup>, S.Randall Thomas<sup>4</sup>, Rajkumar Buyya<sup>1</sup>

Grid Computing and Distributed Systems Lab<sup>1</sup>  
Dept. of Computer Science and Software Eng.  
The University of Melbourne, Australia

Faculty Information Technology (IT) Unit<sup>3</sup>,  
Faculty of Medicine, Dentistry and Health Sciences  
The University of Melbourne, Australia

Department of Information Systems<sup>2</sup>  
The University of Melbourne, Australia

IBISC<sup>4</sup> (Informatiques, Biologie Intégrée et  
Systèmes Complexes) FRE 2873  
CNRS/Université d'Evry - Val d'Essonne  
91000 Evry, France

{xcchu,rbuyya,alonie,pjharris}@unimelb.edu.au, randy.thomas@ibisc.fr

## Abstract

In this paper, we present a grid computing platform that provides experimental scientists and analysts with access to computational simulations and knowledge databases hosted in separate laboratories around the world involved with human and animal kidney research. No single laboratory can develop these resources in isolation and the community of users should no longer need to be dependent upon the specific programming environment in which applications have been developed. This is a major innovation in life science and exploits the power of existing and planned high-bandwidth communications networks for collaborative research and for shared access to knowledge resources. The innovation is developed within a specialist community of renal scientists but will be transferable to any other field of research requiring interaction between published literature and databases, theoretical models and simulations and the formulation of effective experimental design.

## Categories and Subject Descriptors

C.2.4 [Computer Communication Networks]: Distributed Systems – *Distributed Applications*, J.3 [Computer Applications]: Life and Medical Sciences – *Biology and genetics*

## General Terms

Design, Experimentation

## Keywords

Kidney Modeling, Grid Computing, Grid Portal, Web Service Resource Framework, Visualization

## 1. INTRODUCTION

Mathematical modeling has been used successfully in analyzing the huge volume of biomedical data resulting from such ventures as the cardiac modeling. In particular, the approach has been very

"Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

4<sup>th</sup> Workshop on Middleware for Grid Computing, Nov. 27, 2006  
Melbourne, Australia, © 2006 ACM, ISBN:1-59593-581-9. \$5.00"

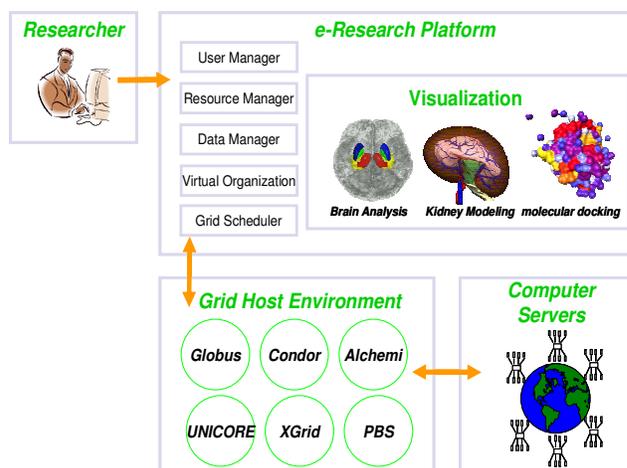


Figure 1. Typical e-Science/e-Research platform.

successful in modeling heart physiology, and mathematical modeling has always played a key role in research on kidney physiology. A large number of mathematical models and resources describing aspects of kidney physiology exist, in different formats and simulation environments around the world; however they are generally proprietary and incompatible with each other, making access and integration of models and resources extremely difficult and time-consuming. A grid computing based approach to integrating and representing distributed renal models and resources will provide researchers with ready access to a collection of interactive models and resources that minimize the requirement for specialist programming or computational modeling expertise on the part of the experimenter.

A variety of legacy kidney models exist in the research community, which represent many different implementation techniques (different modes of interaction – command line, graphical interface, command file driven) and they run on several different platforms. It is very difficult to (i) accelerate the development process for each model since they are distributed among diverse computing platforms and geographical areas, and (ii) manage the community of users and the information provided by the models. Grid Computing [1] not only provides powerful computation facilities as a means for researchers to do existing research faster, but it also promises them a number of other

facilities such as working in collaborative environments, reducing costs, and gaining access to an increased number of resources and instruments, allowing for more advanced research to be carried out by a wider community of users. Furthermore, many of the complexities arising with grid computing have been isolated and solved by technologies including Grid middleware [2], resource schedulers [3], virtual organization [4], data management [5] and resource management and information services. Advanced integration of these technologies generates the basic services of community infrastructure for life science and other e-Science/e-Research areas, as shown in Figure 1.

In this paper, we describe the development and deployment of an interactive web-based portal for quantitative renal physiology modeling at the international level, with the primary aim of providing general access to a repository of legacy renal models at all scales and to the published measurements that underpin the parameter values needed for the development of such models. Specifically we are able to provide an infrastructure with the following features:

- Additional functionality through web-based access to published quantitative measurements on all aspects of renal physiology and anatomy (QKDB) [30] and to the legacy of hypothesis-based mathematical models based on these data,
- A collaborative environment to encourage and assist the entry of new workers into modeling, particularly those with less well developed skills and expertise in programming and computational biology.

The infrastructure includes an interactive web interface to a collection of distributed legacy models at all levels of kidney physiology, with, for each model: documentation, physiological context, easily interpreted output, a statement of model limitations, interactive exploration, and user-customization of selected parameter values. A 3D kidney virtualization GUI is used to facilitate interaction with the resource.

The rest of the paper is organised as follows. In Section 2, we describe related work in the e-Science and life science areas. The design issues of our infrastructure are introduced in Section 3 and in Section 4, we present details of the design and implementation of our KidneyGrid. Evaluation of one scenario we have developed which integrates a legacy model of renal medulla called Kidney Simulation (KSim) under our environment is described in Section 5. Finally, a brief conclusion and plans for future work are presented in Section 6.

## 2. RELATED WORK

Many efforts have been made to develop e-Science platforms and infrastructure in the grid environment. A portal for Grid-enabled physics, in particular astrophysics and high energy particle physics was proposed and implemented in [6]. It was able to host user-level services such as simulation, or handling analysis of job configuration, job submission and monitoring. The Australian BioGrid Portal [7] was another portal-based solution enabling bio-scientists to perform drug-lead exploration in an efficient and inexpensive manner on national and international computing Grids. It provides a complete molecular docking e-Science platform for the use of docking software, from user management to experiment composition, execution and monitoring over grid resources, and finally visualization of simulation results. A Grid

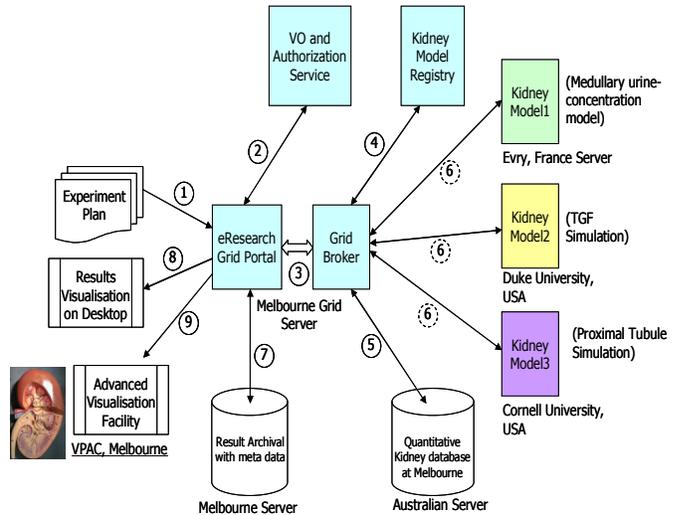


Figure 2. A Grid-based eResearch Architecture for Integration of Distributed Kidney Models.

platform for distributed analysis of neuroimaging data was introduced in [8], which was able to compose and formulate MRI data analysis as a parameter sweep application and perform the experiments on the Grid resources. It substantially reduced the total time needed to analyze MRI data compared with previous solutions. As reported in [9], a chemistry portal was developed for Grid-enabled molecular science, and similar to the BioGrid [7], also adopted a portal-base solution for executing experiments over Grid resources for quantum chemistry code. Importantly, the solution [9] unlike [6] [7] [8], did not utilize a user-level Grid middleware such as a resource scheduler [3]. The advantage of utilizing the user-level resource scheduler is that it decouples the system and underlying grid middleware like Globus [2] and makes the system applicable for other grid resources without the support of Globus. In this work, we take an approach similar to that described in [6] [7] [8] and [29], but focus heavily on the utilizing Web services-based technologies to enhance interoperability with other resources/platforms.

## 3. SYSTEM OVERVIEW

### 3.1 Architecture

We aim to develop a life science infrastructure and collaborative computing environment that enables (a) formation of a distributed community of developers of kidney models, resources, and users as a virtual organization, (b) coupling of distributed kidney models provided by different development communities, and (c) Web portal-based interaction between various models allowing aggregation and advanced visualization. A service-oriented Grid-based architecture for realization of this life science environment for distributed kidney modeling is shown in Figure 2. The entire infrastructure helps in overcoming many of the limitations we faced in our early proof-of-concept of the web-based portal for kidney models, in which all interactions with remote models were manual and hard-coded; communication was mediated through specific protocol calls (e.g. scripted telnet sessions), or results files were simply uploaded to the portal server using HTTP or FTP, and new resources and users were added manually. To this end, the proposed development of a grid-based infrastructure for

managing the distributed components enables a vastly simplified interaction mechanism for both model ‘suppliers’ and users of the system.

High-level interactions between various entities, components, resources, and human participants and the channels of control (shown in Figure 2) are described below:

1. User initiates session with portal by selecting modeling resources required and ‘planning’ experiment with the resources.
2. User obtains authorization through VO and authorization service.
3. Portal contacts Grid Broker to request resources (Remote/Local Kidney Database or remote simulations).
4. Grid Broker queries the Registry to identify reference to resource. Registry returns references to resources required and Broker returns reference plus any important control data to the Portal.
5. Broker opens communication channel to remote resource and sets up experiment information. Note that resources may be locally based (e.g. in the local mirror of the Kidney database) or distant.
6. More commonly, simulations will be hosted on remote sites, running on proprietary systems. The Grid Broker will manage communications with the remote resource through a combination of communications and/or implementation of service-based wrappers at the remote resource. A basic assumption of our system is that as little as possible should be required to be implemented on the remote resource, so communication brokering will most likely be tailored to individual model resources with most data processing of result sets occurring at the portal.
7. User session state, including current state of experiment, is saved to local database.
8. Results are visualized on the users’ desktop in a browser.
9. Alternatively, for complex data sets we will make use of external visualization resources, in particular the VPAC Advanced Visualization Facility in Melbourne.

## 3.2 Issues to be addressed

In the long term, we aim to develop a platform for ‘plug and play’ modeling, with a standard basis of representation for all models (preferably an XML-based open modeling markup language such as CellML [10]) and with similarly standardized data representation. We have made initial proof-of-concept progress toward this goal by developing a CellML based implementation of a published model (although the input and output data formats are still ‘proprietary’ to some extent) and rendering the results in an open-standards based 3D interface [12]. The implementation is based on the model described in [11]; currently we have two cell types working with advection-diffusion spatial modeling and have demonstrated agreement of the published numerical output [12]. Outputs of this model, including solute concentrations along the length of the distal tubule, are rendered on a browser based 3D nephron representation derived from real microCT data, which we intend to develop into the primary graphical interface for interaction with KidneyGrid resources data representation.

It is important to include sufficient material for the site to be interesting and useful once it is opened to the general community. The models we will implement in the core collection cover a wide

range of scales and represent some of the classic studies in the field. The list includes:

- Distal tubule reabsorption [11].
- Proximal tubule reabsorption [13] [14]
- Tubulo-glomerular Feedback [15] [16]
- Glomerular filtration [17]
- Medullary models of the urine concentrating mechanism:
  - simple central core model [18] [19]
  - multi-individual-nephron flat model [20]
  - shunted-multi-nephron flat model [21]

It should be noted that the listed legacy kidney models and resources represent many different implementation techniques (different modes of interaction – command line, graphical interface, command file driven) and they run on several different platforms. There are currently no standards for data representation, interaction or communication.

The issues of standards and interoperability could be addressed by recoding all the models in a common format as discussed above, but this long term and resource intensive solution is probably not feasible without direct and significant input by each of the model developers. A more feasible approach is to develop interaction ‘wrappers’ around the current model implementations and within the central portal convert the output so that diverse result sets can be rendered on a common interface. This preferred approach requires infrastructure ‘middleware’ to manage the communication channels, data transfer channels, user session state, user authentication etc within the context of a central portal.

The basic principle is to create a web page that presents the user (i.e., the visitor to the website; also called the “client”) with an annotated list or a visual interface for selection of one of the available models. The user is then presented with a more detailed description of the chosen model, including literature references, and is given the opportunity to either look at the results of some pre-configured simulations to “get the feel” of the model, or to alter particular parameter values (within a constrained range representing physiologically reasonable values) in order to customize the model as required. Upon submitting the new parameter values, the user will be served (after a wait that depends on the execution time of the model calculations) with the results and may choose among several styles of presentation, such as a simple table, graphical output, or in some cases more visually meaningful presentations, or may request to download the results as a table for post-treatment in a spreadsheet or graphing program of their own.

## 4. DESIGN AND IMPLEMENTATION

The primary design principle of our KidneyGrid is to maximize the reuse of existing software components, tools, services and frameworks provided by the open source community for generic tasks, while concentrating our efforts on the integration of those technologies as well as providing the service wrapper for legacy kidney models. Table 1 provides details on key technologies used in the development our KidneyGrid environment. Thus, we are able leverage various existing Grid technologies and build on our experience in the development of Grid-based platforms for e-Science applications in life sciences area.

<i>Component</i>	<i>Technology Used</i>	<i>Comments</i>
Portal	Gridsphere Framework [22]	Reusable web components. JSR 168 portlet standard compliant. Open source and widely used for grid portal development.
Virtual Organization (User Authentication)	MyProxy [23]	Only consider the authentication for the VO, may extend to completed VO system such as VOMS
Grid Resource Broker	Gridbus Broker V3.0 [3]	Provide ability to support Web Service Resource Framework (WSRF), various scheduling algorithms and resources such as Globus and Alchemi
Service Wrapper for Legacy Kidney Model	Web Service Resource Framework (WSRF) [24]	WSRF provides stateful web service which is able to deployed on the Globus toolkit 4 (gt-4) container
Visualization	VPAC Advanced Visualization Facility in Melbourne	Provide Java Applet that can be easily integrated into web portals
Core Grid Middleware	Globus Toolkit 4, Alchemi, UNICORE	Provide supports for integration of grid resources accessible through different low level grid middleware

**Table 1. Key Technologies Chosen.**

## 4.1 Portal Development

*Gridsphere* was chosen as the portal development framework because (i) it is open source and JSR 168 compliant which means we could easily reuse the components in other portal containers supporting this standard, (ii) it supports the Java Server Facers (JSF) standard which could potentially be used within a visual page editor tool to accelerate the development of web pages, (iii) it contains common portal components such as user management and localization support which we could reuse directly, and (iv) it is mature and widely used by other grid portal related projects.

## 4.2 Legacy Application Integration

The most important feature of our KidneyGrid is the integration with legacy kidney models developed by different programming languages and platforms. Our approach is to provide both language and platform neutral and interoperable wrappers for each model and those wrappers are responsible for the inter-communication between our infrastructure and the legacy models. Furthermore, the wrapper should be able to be deployed inside a grid environment such as Globus [2] and Alchemi [25]. The wrapper should also be capable of storing the states of the input, output and execution information about the kidney modeling as some of the model simulations last long time.

The Web Service Resource Framework (WSRF) [24] provides a service oriented architecture (SOA) solution to keep states as separate resources called WS-Resource, which do not require the web services to keep track of the resource identifier explicitly via passing it as parameters each time. The complete separation of web services and state information as WS-Resources (i) retains the relatively similar scalability of stateful web services compared with plain stateless web services, and (ii) fulfils the requirements of keeping state information of grid applications. WSRF relies on a set of specifications including WS-ResourceProperties, WS-ResourceLifetime, WS-ServiceGroup, WS-BaseFaults, WS-Notification and WS-Addressing.

The wrapper service for the legacy kidney models utilizes the WSRF, which implements one kidney WS-Resource for each model representing the state information such as input and output

parameters, and execution outcomes. Since a given simulation of the kidney modeling may not respond immediately, we also utilize the notification mechanism in WSRF provided by WS-Notification to notify the client once the simulation has finished and to collect the result. For each wrapper service, we implement a factory service that is responsible for creating the kidney resources and returning the handler of the actual service that performs the modeling simulation. As mentioned, the legacy kidney models use various programming languages. We therefore implement an executor written in Java for each model, which simply creates a runtime process for the modeling program and passes the parameters or composes the input files at runtime. The wrapper service actually utilizes the executor and passes the required parameters.

## 4.3 Grid Resource Broker

Consider the scenarios that some of the legacy models are only compatible with the UNIX environment and some are only executable under Windows. In these cases, we require different grid host environments, for example, Globus [2] for UNIX and Alchemi [25] for Windows and it is very important to utilize a user-level middleware to decouple the management work such as job composition, scheduling, monitoring and result collection from the underlying grid middleware. Gridbus Broker [3] is a user-level meta-scheduler that mediates access to distributed resources and provides facilities for all the aspects of the application, credential and service management. We adopt version 3.0 of the Gridbus Broker, which works with remote services that are compatible with WSRF-based web services compatible broker. In addition, broker services can be invoked as Web services; i.e., broker can be deployed separately on other servers instead of the same server on which our portal is running.

## 4.4 Experiment Preparation and Execution

Preparation of a virtual experiment is managed via a web page that allows the user to select and adjust various parameters including specified values and permitted ranges of values for their selected kidney models. The information is parsed and the grid jobs are created dynamically for the broker to dispatch to the resources. Once users submit their experimental plans, the Gridbus Broker is responsible for scheduling the submission of

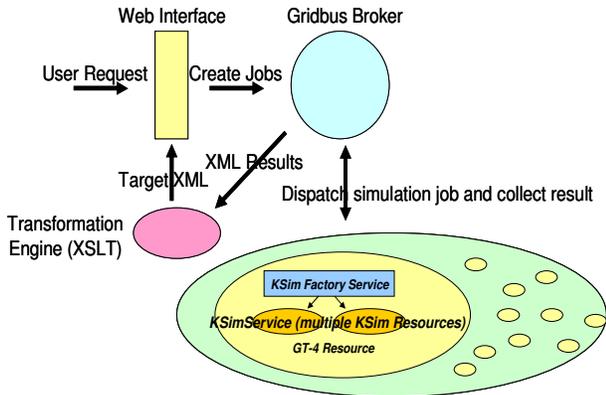


Figure 3. Workflow of KSim Service.

those jobs generated by different parameters and monitoring their execution on remote grid resources. We have implemented a generic monitoring service that extends the monitoring service in BioGrid [7] and is compatible with the Gridbus Broker 3.0 API.

#### 4.5 Visualization

For users to gain a better understanding of modeling simulation outcomes, visualization facilities are crucial and essential for almost every life science application. In our KidneyGrid, we utilize existing visualization facilities provided in VPAC Melbourne, which is designed to extend mozCmgui [26] to render a 3D kidney image by reading a XML file conforming to a standard XML Schema. As various kidney modeling programs are used it is not always possible to generate standard XML files. We therefore need to transform the diverse result formats to the standard that the visualization tool can read. XSLT [27] can be used as the transformation engine to accomplish the task and for each kidney model and we will create a stylesheet specifying the rules for conversion to the standard format.

### 5. EVALUATION

We have successfully implemented a wrapper service for KSim an implementation of a medullary kidney model implemented in the Fortran language [21]. Figure 3 shows the basic workflow required by users performing virtual experiments with this model. The KSim Factory Service and KSimService are two WSRF services deployed in the Globus Toolkit 4.0 (GT-4) container.

To access the portal, authentication and authorization are required to obtain the credentials for users utilizing the remote computing resources from the VO system, as shown in Figure 4. Currently, we simply create a mapping between the user and the credential in

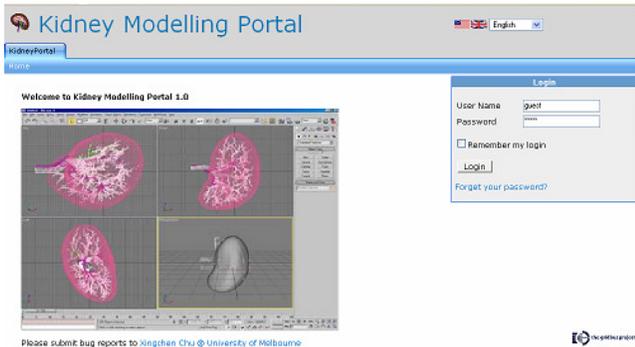


Figure 4. Access Page to KidneyGrid.

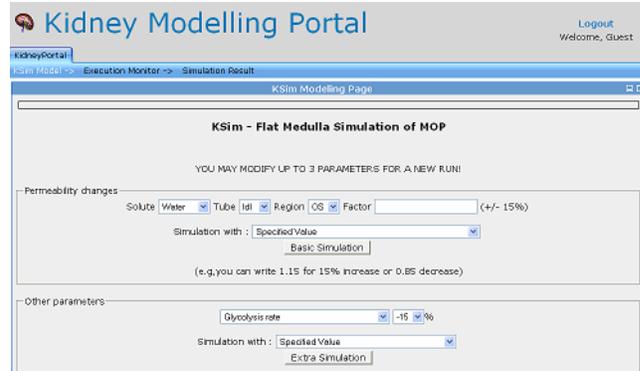


Figure 5. KSim Experiment Plan Page.

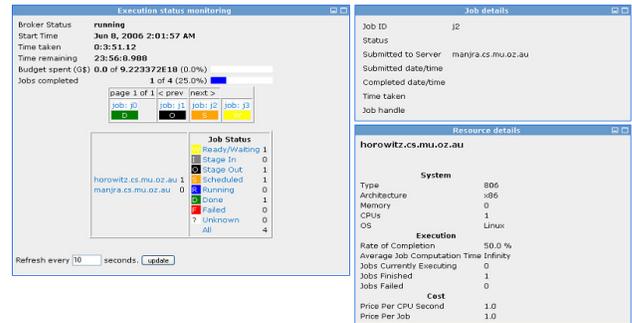


Figure 6. Execution Monitor.

the database, we may apply MyProxy [23] (use Grid Portlet directly in our portal) or our own VO implementation supporting various authentication mechanisms in the future. Once users have gained the access to the portal, an experiment plan page is rendered for them, in which the user can change the parameters accepted by KSim, as demonstrated in Figure 5. The user can specify values for each parameter or make generic choices for parameters to simulate a range of values for that parameter at one time. As soon as the user submits the experimental plan, the Gridbus Broker commences scheduling and managing the jobs generated from different parameter settings. Since an experiment is unlikely to complete immediately, the user is able to check the status of the simulation using the execution monitor, as shown in Figure 6. This is a user-friendly facility to provide feedback of the simulation run on the remote grid resources. It not only provides the current status of each simulation on the remote resources, but also the resource information as well as the time and budget spent for the experiment if applicable (user has specified the QoS Parameters). The user can then interact with the simulation results using the visualization applet shown in Figure 7, this

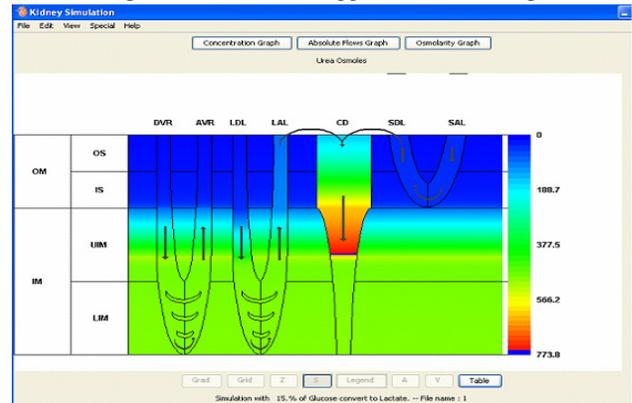


Figure 7. KSim Visualization Applet.

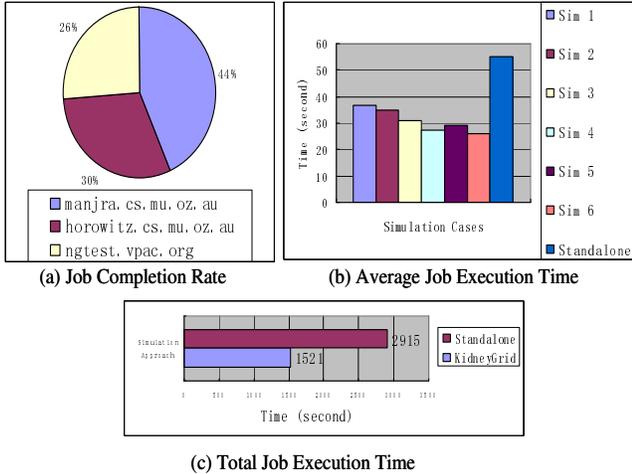


Figure 8. Evaluation of Grid Jobs over Resources.

applet was adapted from Randy Thomas KSim France [28].

As KSim accepts four basic types of parameter including solute, tube, region and factor, and extra special cases with factors, the possible simulation cases can be expressed as the equation:

$$T = C_s^1 \times C_t^1 \times C_r^1 \times C_{f_n}^1 + C_x^1 \times C_{f_x}^1$$

$T$  is the total number of cases,  $s$ ,  $t$ ,  $r$ ,  $x$  stand for solute, tube, region and extra parameter respectively,  $f_n$  and  $f_x$  are the acceptable factors for basic and extra simulation. Depending on the value for each parameter specified in KSim, it may generate a total of 513 simulation cases. To test the system performing multiple simulations, we deployed the KSim WSRF wrapper service onto three computer servers which have a GT-4 container installed on each server. As our portal is only capable of generalizing one parameter per experiment, we have conducted 6 experiments which choose generic value for one parameter at a time and have generated 53 simulation cases in total. The Gridbus Broker then scheduled 53 jobs (one for each simulation) over three grid resources.

Figure 8(a) demonstrates the jobs completed in each of 3 server. Two of the server 'horowitz' and 'ngtest' have a similar proportion of jobs completed (28% each) compared with the third server 'manjra', which has completed 44%. This is because we have configured a working queue with a limit of 4 jobs on 'manjra' which means there are 4 jobs running in parallel on manjra once scheduled. According to Figure 8(b), we have conducted 6 parameter sweep simulations, the average execution of each simulation using our KidneyGrid infrastructure over 6 simulation cases is about half of the execution time if we run the simulation sequentially on one machine, which indicates our approach has reduced almost 50% of the execution time to run the KSim simulation. Figure 8(c) also shows that the total execution time has been decreased about 50% using our approach. The execution time in our experiment includes all the aspects of network latency, broker overhead and actual job execution on the resource. We have also found that the overall time can be further reduced by (i) running the job manager system such as PBS with queueing supports, (ii) reducing the overhead (10 to 12 seconds for each job) of the broker to schedule, and (iii) adding more grid resources that can host the simulation models. Our experiment shows that our KidneyGrid provides improved throughput from

experiment plan to result visualization by distributing kidney modeling simulations over grid resources that have largely reduced the complexity of the usage and the computation time for batch experiments.

## 6. CONCLUSION AND FUTURE WORK

In this paper, a life science infrastructure to grid-enable distributed legacy applications and resources utilizing service oriented architecture is presented. We focus on grid-enabling various kidney models within this infrastructure, demonstrating the ability to compose, schedule, monitor and visualize the result of the application. The use of the grid resource broker not only simplifies the development of application, credential and resource management, but also decouples our platform from the underlying grid middleware. WSRF makes the state-oriented web services applicable for grid application, which provides a lightweight solution to integrate legacy applications by implementing service wrappers for them. Our project is still in the early stages of development, and much still needs to be accomplished. Wrappers are required for the kidney models on our list so that the successful development and deployment of the wrapper service for KSim model can be applied to other kidney models. Also we need to create the stylesheet to format each model result into the standard representation once the 3D visualization tool is available. As XSLT only works with XML input, we should utilize other transformation tools to handle non-XML input generated by the legacy kidney models. Other ongoing work includes the integration of other services for resource discovery and management, and MyProxy or VOMS for VO authentication and authorization.

## 7. REFERENCES

- [1] I. Foster and C. Kesselman. *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann Publishers, 1999.
- [2] I. Foster and C. Kesselman. *The Globus Project: A Status Report*, Proceedings of IPPS/SPDP'98 Heterogeneous Computing Workshop, 1998, pp. 4-18.
- [3] S. Venugopal, R. Buyya and L. Winton. *A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids*. Proceedings of the 2nd International Workshop on Middleware for Grid Computing October 18, 2004, ACM Press, 2004, USA.
- [4] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, A. Gianoli, K. Lorentey, and F. Spataro. *Voms: An authorization system for virtual organizations*. In Proceedings of the 1st European Across Grids Conference, Santiago de Compostela, Feb. 2003.
- [5] C. Baru, R. Moore, A. Rajasekar, and M. Wan. *The SDSC Storage Resource Broker*, in Proceedings of CASCON'98, Toronto, Canada, Nov 1998.
- [6] B. Beeson, S. Melnikoff, S. Venugopal, and D. G. Barnes. "A Portal for Grid-enabled Physics," in *Proceedings of the 2005 Australasian Workshop on Grid Computing and e-Research*, Newcastle, Australia, January 2005.
- [7] H. Gibbins, K. Nadiminti, B. Beeson, R. Chhabra, B. Smith, and R. Buyya. *The Australian BioGrid Portal: Empowering*

- the Molecular Docking Research Community*, Proceedings of APAC 2005 Conference, Sept. 2005, Gold Coast, Australia.
- [8] S. Kolbe, T. Ma, W. Liu, W. S. Soh, R. Buyya and G. Egan. *A Platform for Distributed Analysis of Neuroimaging Data on Global Grids*. Proceedings of the 1<sup>st</sup> International Conference on e-Science and Grid Computing, Dec. 5-8, 2005, Melbourne, Australia.
- [9] Z. Zhou, F. Wang and B.D. Todd. *Development of Chemistry Portal for Grid-enabled Molecular Science*. Proceedings of the 1<sup>st</sup> International Conference on e-Science and Grid Computing, Dec. 5-8, 2005, Melbourne, Australia.
- [10] CellML, <http://www.cellml.org/>.
- [11] H. Chang. and T. Fujita. *A numerical model of the renal distal tubule*. Am J Physiol 276(6 Pt 2): F931-51, 1999.
- [12] A. J. Lonie, C. Stevens and P. Harris. *LB520: Computer modelling of kidney function*. Experimental Biology 2004, Washington D.C.
- [13] A. M. Weinstein. *A mathematical model of the rat proximal tubule*. Am J Physiol 250(5 Pt 2): F860-73, 1986.
- [14] S. R. Thomas and G. Dagher. *A kinetic model of rat proximal tubule transport-loaddependent bicarbonate reabsorption along the tubule*. Bull Math Biol 56(3): 431-58, 1994.
- [15] H. E. Layton, E. B. Pitman and L. C. Moore. *Bifurcation analysis of TGF-mediated oscillations in SNGFR*. American Journal of Physiology 261(5 Pt 2): F904-19, 1991.
- [16] N. H. Holstein-Rathlou, A. J. Wagner and D. J. Marsh. *Tubuloglomerular feedback dynamics and renal blood flow autoregulation in rats*. Am J Physiol 260: F53-68, 1991.
- [17] W. M. Deen, C. R. Robertson and B. M. Brenner. *A model of glomerular ultrafiltration in the rat*. Am J Physiol 223(5): 1178-83, 1972.
- [18] D. M. Foster and J. A. Jacquez. *Comparison using central core model of renal medulla of the rabbit and rat*. AJP 234(5): F402-F414, 1978.
- [19] J. L. Stephenson, R. Mejia and R. P. Tewarson. *Model of solute and water movement in the kidney*. Proc Natl Acad Sci USA 73(1): 252-6, 1976.
- [20] P. Lory. *Effectiveness of a salt transport cascade in the renal medulla: computer simulations*. AJP 252 (Renal 21): F1095-F1102, 1987.
- [21] S. Hervy and S. R. Thomas. *Inner medullary lactate production and urine-concentrating mechanism: a flat medullary model*. Am J Physiol Renal Physiol 284(1): F65-81, 2003.
- [22] J. Novotny, M. Russell, O. Wehrens O. GridSphere: a portal framework for building collaborations, *Concurrency and Computation: Practice & Experience*. 16(5):503-513, 2004. <http://www.gridisphere.org>.
- [23] MyProxy, <http://grid.ncsa.uiuc.edu/myproxy/>.
- [24] Web Service Resource Framework, [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsrf](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf).
- [25] A. Luther, R. Buyya, R. Ranjan, and S. Venugopal, Alchemi: A .NET-Based Enterprise Grid Computing System, Proceedings of the 6th International Conference on Internet Computing, June 27-30, 2005, Las Vegas, USA.
- [26] mozCmgui, <http://www.cmiss.org/ReleaseCenter/mozcmgui>
- [27] XSLT, <http://www.w3.org/TR/xslt>
- [28] KSim, <http://www.lami.univ-evry.fr/~srthomas/kidneysim/>
- [29] W. Sudholt, K. Baldrige, D. Abramson, C. Enticott, and S. Garic. "Application of Grid computing to parameter sweeps and optimizations in molecular modeling". In Proceedings of the 4<sup>th</sup> International Conference on Computational Sciences, June 6 – 9, 2004, Krakow Poland.
- [30] QKDB, <http://www.lami.univ-evry.fr/~srthomas/qkdb/>