Building Cyberinfrastructure for Bioinformatics Using Service Oriented Architecture

Wilfred W. Li, Sriram Krishnan, Kurt Mueller, Kohei Ichikawa, Susumu Date, Sargis Dallakyan, Michel Sanner, Chris Misleh, Zhaohui Ding, Xiaohui Wei, Osamu Tatebe, Peter W. Arzberger

***Abstract* — Cyberinfrastructure makes the development and deployment of bioinformatics applications easier by providing the framework and components that may be loosely coupled using service oriented architecture. Here we describe an end to end prototype environment that allows existing applications to run on the grid, taking advantage of open source software that provides a portal interface using GridSphere, with transparent GSI authentication using GAMA, a web service wrapper using Opal, a metascheduler using CSF4, a virtual filesystem using Gfarm, and a grid-enabled cluster environment using Rocks. Solutions to complex problems may be developed using workflow tools such as Kepler that coordinate different interoperable services. The availability of this type of cyberinfrastructure suggests that new applications should be designed with the grid in mind, using service oriented architecture for interoperability and efficiency. This approach may enable bioinformaticians to focus on their problems of interest, and make use of the emerging cyberinfrastructure through virtually "drag and drop" deployment.**

***Index Terms* — grid, bioinformatics, deployment, workflows, portals, metascheduler.**

I. Introduction

As genome sequencing technology has become a hot research area for nanotechnology, biology is undergoing the transformation from a mostly experimental science to a digital information science [1]. The complex data and computational challenges facing biology and medicine are enormous, especially from the realization that knowing the parts (genes and genomes) is far from the coveted future of personal, predictive, preventive, and participatory medicine [2] and there is significant gap to be bridged by using computational science in medicine [3]. The challenge to 21st century biology is to be able to develop a holistic view of humans and the environment, and develop software and technology for simulation based medicine [4]. The exciting development in biology parallels that of the grid or cyberinfrastructure in the computer science and engineering field. The Grid, or computational grid, refers to the hardware and software infrastructure that enables the solution of complex problems using many computers, including the sharing of distributed resources across institutions, or organizations [5, 6]. Cyberinfrastructure refers to "infrastructure built upon distributed computer, information and communication technology", which forms the foundation of a knowledge based economy [7]. Despite the subtle differences, the two terms are often used interchangeably.

As stated eloquently in the Atkins report, "Cyberinfrastructure makes applications dramatically easier to develop and deploy, thus expanding the feasible scope of applications possible within budget and organizational constraints, and shifting the scientist's and engineer's effort away from information technology development and concentrating it on scientific and engineering research. Cyberinfrastructure also increases efficiency, quality, and reliability by capturing commonalities among application needs, and facilitates the efficient sharing of equipment and services" [7].

Bioinformatics or computational biology, "the use of computer-driven methods to enable biological knowledge", is becoming as pervasive as what molecular biology is to experimental biology, an essential tool for any biological inquiries [8]. The developing grid or cyberinfrastructure must be able to support the needs of bioinformatics, through an iterative process of interdisciplinary collaborations between the scientists and infrastructure developers.

A number of international projects have formed over the years in the development of grid environment for biological, and environmental sciences, with strong bio- or eco-informatics components [9]. For example, the myGrid project [10, 11] in UK uses bioinformatics as a major focus in the development of semantic grid technology based on web services. As part of the myGrid project, Soaplab [12] is a CORBA based generic toolkit for deploying applications as web services; Taverna is capable of orchestrating web services based workflows with a strong emphasis on semantic service discovery [13]. The Japan BioGrid project [14] is among the early initiatives to address the grid computing needs in the life sciences, especially in the informatics support for biomolecular simulations using QM/MM techniques. The

Manuscript received December 28, 2005. WWL, SK, KM, CM, MS and PWA wish to acknowledge NIH NCRR P41 RR08605 to NBCR. WWL and PWA also acknowledge NSF Grant No. INT-0216895 and INT-0314015 to PRAGMA. X. Wei wishes to acknowledge Jilin University grants 419070200053 and 420010302338; and China NSF grant 60473487. K. Ichikawa, and S. Date are with CMC, Osaka University, Japan ({ichikawa,date}@ais.cmc.osaka-u.ac.jp). Z. Ding and X. Wei are with CCST, Jilin University, China (zding@sdsc.edu, weixh@jlu.edu.cn). O. Tatebe is with AIST, Japan (o.tatebe@aist.go.jp). S. Dallakyan, M. Sanner is with Scripps Research Institute (sanner@scripps.edu). W. W. Li, S. Krishnan, K. Mueller, C. Misleh and P. W. Arzberger are with University of California, San Diego, La Jolla 92093, USA. (phone: 858-822-0974; fax: 858-822-0861; e-mail: ({*wilfred,kurt,sriram,cmisleh}@sdsc.edu*; *parzberg@ucsd.edu*).

Pacific Rim Grid Middleware and Application Assembly (PRAGMA) [15] has been a catalyst for international collaboration and the development of a multinational grid testbed [16] and the sharing of valuable experiences in grid systems design and application deployment [17]. The National Biomedical Computation Resource (NBCR) at UCSD has also been actively developing cyberinfrastructure in support of multiscale modeling [18].

Here we describe the collaborative development of a prototype environment, or end to end cyberinfrastructure, for bioinformatics. The architecture is flexible, extensible, and generally applicable to many other application areas.

II. End to End Cyberinfrastructure

An end to end solution provides a comprehensive problem solving environment for users, with an intuitive user interface, transparent access to data, and underlying computation resources. It satisfies any visualization needs, and provides a personalized workspace for workflow composition and management of tasks. This type of complex environment is best developed using the service oriented approach [19].

A. Service oriented approach

The concept of service oriented architecture is not new. It differs from the component based approach in that services are independent autonomous units that fulfill requests through standardized protocols and frameworks [20]. Complex workflows may be developed using loosely coupled services at different end points. XML based web services, described in WSDL (Web Service Description Language), which communicate using SOAP, an XML-based messaging format, over HTTP protocol, is the predominant form of service in use today. The Globus toolkit 4 (GT4) [21], a popular middleware package comprised of software packages that mediate the interactions between user applications and the underlying raw compute resources, is a reference implementation of the Web Service Resource Framework (WSRF) [22]. A WS-Resource is a stateful web service that explicitly describes its operations using a Resource Property Document schema, which enables simplified interactions among services within the WSRF framework.

The adoption of service oriented architecture allows the autonomous evolution of underlying services without interruptions of the functionalities to be delivered. However, it does require the establishment of interoperable standards for success, as warned by the Atkins report: **"**Infrastructure offers a reference point for mediating the interaction among applications, defining common interfaces and information representations. The alternative of asking applications to interact directly with one another results in a combinatorial explosion of mutual dependencies, creating a house of cards that eventually falls of its own weight."

B. Key components

The architecture of the grid is described as an hour-glass model, in which a large number of user applications and a vast number of compute resources (fabric) are connected by a core set of middleware components. These include a small set of basic connectivity protocols that supports secure sharing of resources, and collective services that support the monitoring, scheduling and accounting of different resources [5]. The representative components are shown (Figure 1).

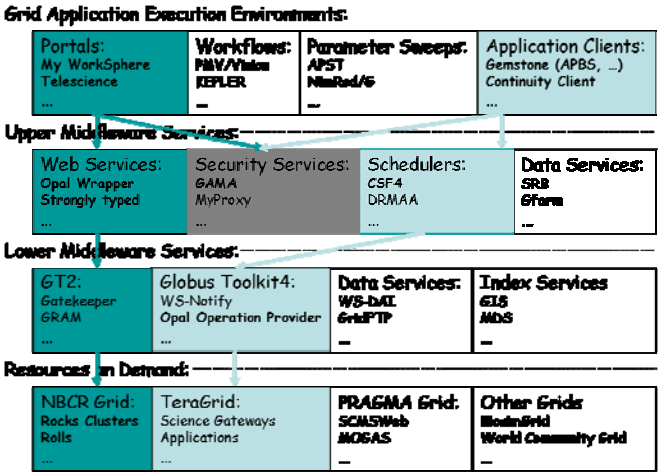

Figure 1. Selected components of the NBCR software service stack. Colored blocks and arrows indicate possible routes for distributed job execution.

Grid application execution environments [23] are designed to hide the execution details of user applications on distributed computation resources. These include web based portals, workflow management software packages, tools that help users automate a large number of parameter sweep jobs, desktop clients that are highly customized for individual applications, and other interactive environments. These tools may access different layers of the grid directly as necessary. The functionality, usability, flexibility, and stability of these tools are significantly enhanced by adopting a service oriented approach themselves and using loosely coupled services within standard frameworks.

Upper middleware services are those that make the development of distributed applications significantly easier, with support for higher levels of abstraction and standardization. For example, Opal based application web services provide job management and scheduling features based on the Globus toolkit. An application developer may begin using the grid quickly with the basic knowledge of web service development, as shown by the use cases later in this paper. Lower middleware services are those that have stabilized over the years and serve as the foundation for the development of more sophisticated and transparent modes of access. However, as often dictated by performance requirements, a user application may access lower layers directly. This is much less desirable unless the integration is based on the service oriented architecture.

In the following sections, we discuss the key components that we have developed or adopted as collaborations within

the context of bioinformatics grid requirements.

1) Opal – a simple yet powerful web service wrapper

Opal is developed by NBCR as a Java based toolkit that automatically wraps any legacy applications with a Web services layer that is fully integrated with Grid Security Infrastructure (GSI) based security, cluster support, and data management [24]. Opal is designed to make applications deployment as web services as simple as possible. This is critical because biologists/bioinformaticians should not have to spend much time to be able to take advantage of the grid computing power, but focus on making their applications able to solve more sophisticated biological modeling problems. Only a simple configuration file is required to deploy an existing application as a web service provider (Figure 2).

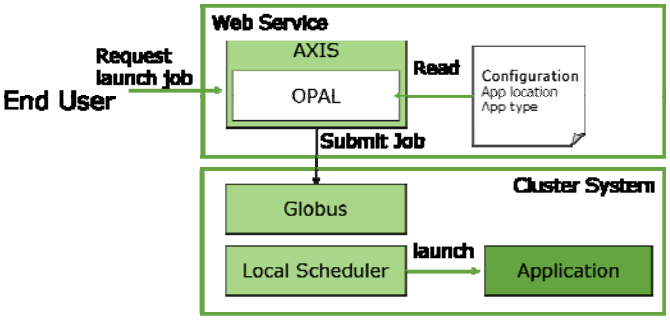

Figure 2. Opal allows rapid deployment of applications as web services using user provided configuration options

As Opal leverages the Globus toolkit for job scheduling, and utilizes resources such as Condor pools, clusters or even individual workstations, it is a natural extension to be able to deploy Opal based services as WSRF compliant web services, to leverage new features from GT4 and to interoperate with other WSRF services. Operation Provider is a Java class that provides a set of WSDL operations for a service. GT4 allows the composition of Java objects as web services [25]. This allows additional operations (services) to be added to the Opal operation provider, and the "flattened" service exposed through GT4 is a much more customized solution (Figure 3).

There are several important aspects of Opal that facilitate the development of the grid using service oriented architecture: 1) any application may be able to be offered as a service and the service is self contained independent of other services. Many providers of the same application service may be registered through a common registry or be discovered using search engines; 2) many Opal based services can be offered using the same computing resource, so that the local resource may be utilized to the maximal extent possible; 3) web service based workflow composition tools may orchestrate different service end points as long as the services can exchange messages effectively; 4) user interfaces to an application are no longer tied to a specific tool, or client, but may be exposed in any environment of choice. 5) Opal based services may leverage other compute resources as hosted applications services, e.g., deployment to large scale resources such as the TeraGrid through Science Gateways [26].

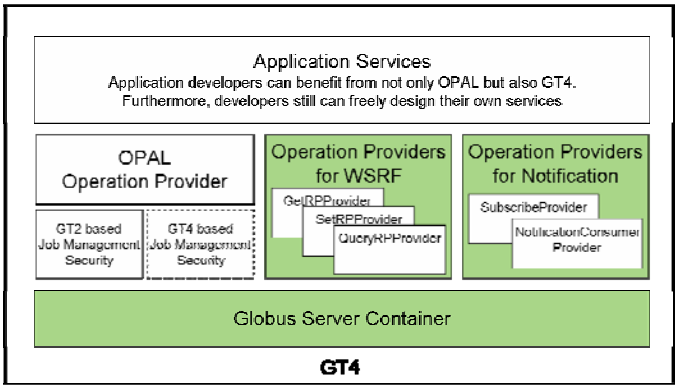

Figure 3. Opal based services can be deployed into GT 4 as Operation Providers and combined with additional desired methods or operations.

2) GAMA – security made simple

GAMA (Grid Account Management Architecture) is a GSI-based security service that manages X.509 user credentials on behalf of users, and supports SOAP-based applications [27]. The server component leverages existing software packages such as CAS [28], MyProxy [29], and CACL [30], with the GAMA version 2 (Figure 4) supporting other CA packages such as the NAREGI CA [31].

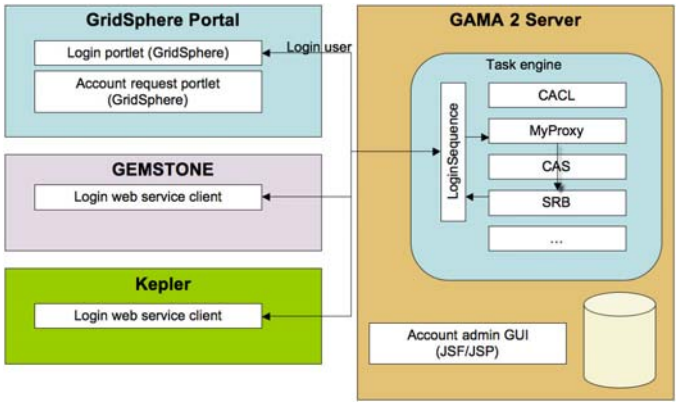

Figure 4. GAMA2 allows security and grid account management from various environments.

A portlet component or a JSP interface provides the administrative interface to the server. As security is a sensitive and critical issue in the production use of cyberinfrastructure, GAMA allows any organization to create their own certificate authority (CA), manage their user certificates, secure the SOAP communications using HTTPS and mutual authentication, and integrate seamlessly with portals, individual applications or rich clients. For example, GAMA is used by GridSphere [32], Gemstone [33], PMV/Vision [34]

and Kepler [35]. The Science Gateways in the TeraGrid project support project-based CAs to reduce the administrative overhead yet make the TeraGrid resources available on demand to more users with trusted project CAs.

3) Gfarm – virtual file system and computation grid

The Grid Datafarm [36] architecture is designed for global petabyte scale data-intensive computing, which provides a Grid file system with file replica management (Gfarm file system), and parallel and distributed data processing support for a set of files (*Gfarm file*). It provides scalable I/O bandwidth, and scalable parallel processing to exploit local I/O in a grid of clusters. The data is, physically replicated and dispersed among cluster nodes across administrative domains, where it can be accessed transparently from file replica locations via POSIX file I/O interface by data analysis tools.

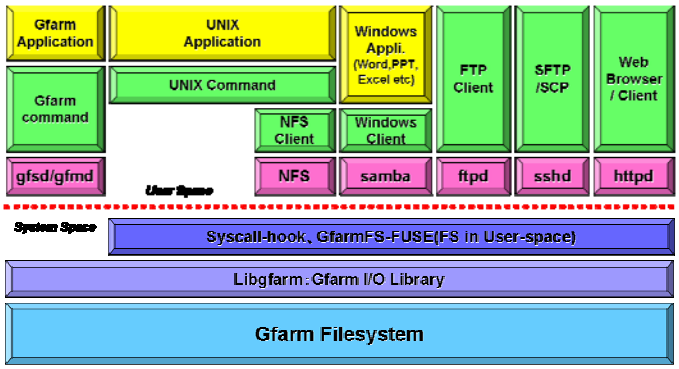

Figure 5. Gfarm supports distributed access of data and compute resources, including legacy applications without modifications.

Gfarm provides a system call hooking library which enables existing applications to run in Gfarm without code modification once the user environment is set up. This means that bioinformaticians may be able to leverage the grid computing power in Gfarm, without even realizing that they are using the grid (Figure 5). New features supported by Gfarm include Gfarm-FUSE [37], which removes the requirement for glibc-not-hidden package and the LD_PRELOAD settings for existing or non-grid-aware applications. Gfarm has been used successfully for a number of bioinformatics applications [38].

4) CSF4 – scheduling across resources

CSF4 (Community Scheduler Framework 4) [39] is the first meta-scheduler based on WSRF (Web Service Resource Framework). CSF is composed of a set of grid services based on Globus Toolkit 4 Java WS Core and can support grid-level scheduling. CSF4 supports both GT2 and GT4 based systems using the Java COG kit [40], and can schedule jobs with the proper user proxy delegation for Gfarm as a virtual filesystem for disparate clusters (Figure 6) [41]. CSF4 also supports the development of plug-in modules that enhance the performance of job execution through data aware scheduling mechanisms [42].

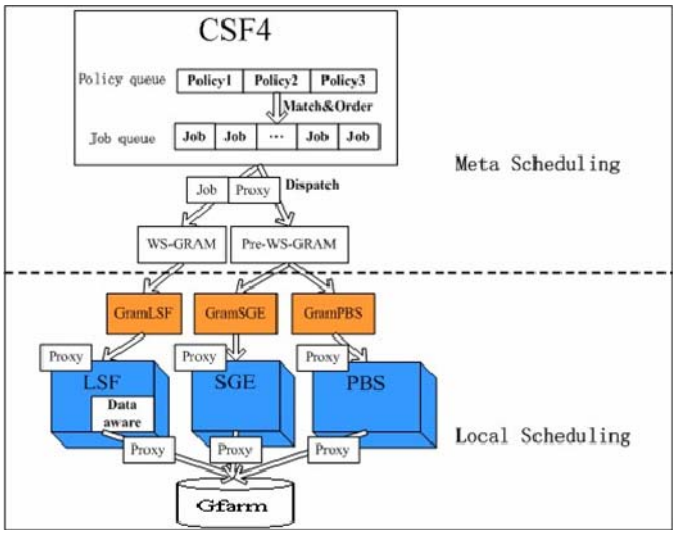

Figure 6. CSF4 schedules user tasks to different resources with different local schedulers, including the delegation of user proxies for Gfarm access.

5) GridSphere, Gemstone, and others as User Interfaces

The advantage of a service oriented approach is quite apparent in user interface development. Applications may be deployed once and accessed by many, not just different users, but also different modes of access using the same underlying SOAP protocol. GridSphere [32] is a portlet container that provides a basic set of grid portlets that support job execution, file management and user certificate management functionalities. There is a standard for portlet development (JSR168 at the moment) that is evolving to ensure interoperability among different containers.

Gemstone [43] is developed with major support from the NSF National Middleware Initiative, and has a basic built interface for Opal based web services as well as an application specific Opal based web service for PDB2PQR [44], a utility package for protein electrostatic surface calculations. Gemstone utilizes the open source Mozilla engine, and uses XML User Interface Markup Language (XUL) to describe user interface.

6) Rocks – Replicable Infrastructure

In an effort to make cyberinfrastructure more readily available to scientists and engineers, it is necessary to not only develop different middleware to support legacy applications, but also to make the different software packages easy to deploy into existing infrastructure. The Rocks cluster environment toolkit [45] has proven to be invaluable for NBCR to build the basic infrastructure, deploy our software stack, and make our infrastructure replicable by others. NBCR has contributed critically to the development of the Condor

[46] roll, which is a mechanism, similar to the RPM package manager though fully automated, for building reproducible cluster and grid environment. Other rolls are available from NBCR for APBS [47], MEME, GAMA, Continuity [48], AutoDock [49, 50], and PMV [48]. Additional rolls for SMOL [51], and FEtk [52] will be available soon.

III. Usage Scenarios in Bioinformatics

A. *Motif discovery in protein/DNA sequences*

MEME [53, 54] is a popular bioinformatics program for pattern recognition or motif identification in protein and DNA sequences. NBCR has hosted the MEME web server since 2000 [55], and is using MEME as one of the key collaborations and scientific drivers to build cyberinfrastructure for bioinformatics.

1) *Access MEME using My WorkSphere*

My WorkSphere is a GridSphere based portal environment that leverages JSR 168 compliant portlets to develop an effective work environment using open source technology [56]. MEME is deployed as a web service using the Opal toolkit in as little as an hour. A portlet that accesses the MEME Opal service is deployed into the GridSphere container with the standard HTML interface to MEME. The portlet takes advantage of Opal's data and job management features: a user's job output may be saved on the server for later retrieval, and any job status may be queried (Figure 7).

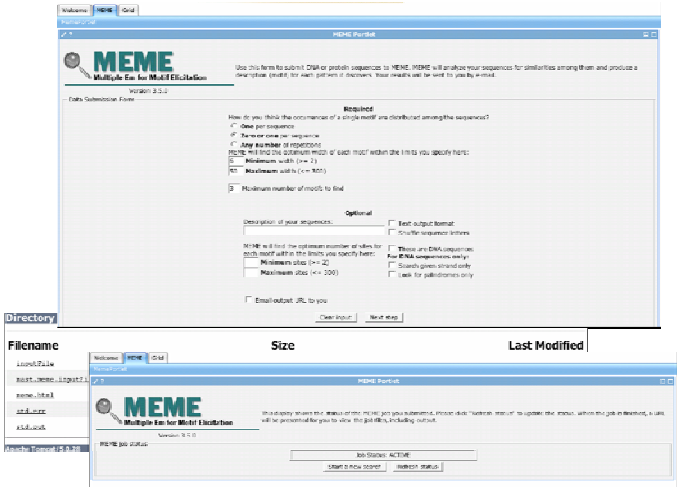

Figure 7. A portlet interface to the Opal based MEME web service that provides data management and job status report

2) *Workflow composition using MEME and MAST Opal services*

Kepler provides a visual programming environment for workflow composition. It supports the development of workflows based on web services, as long as the services are interoperable with the proper semantics understanding. MEME and MAST are deployed as Opal based web services and MEME output is used as input for MAST (Figure 8).

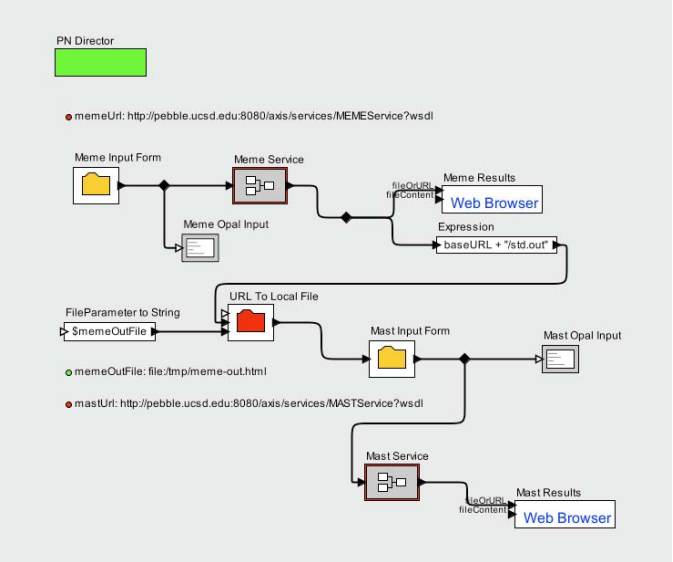

Figure 8. Opal based web services may be used to couple MEME and MAST analysis using KEPLER

Kepler uses the concept of Directors to define the type of workflow being implemented, e.g. communicating sequential processes, synchronous data flow, etc. It uses the concept of Actors to perform certain actions when triggered. Actors have input and output ports which are used to consume and produce data respectively. To enable the MEME and MAST workflow, a generic Opal client actor, separately developed, accesses an Opal service, launches a job, queries for job status, and finally returns job outputs. Since both the MEME and MAST services were implemented using Opal, the same actor could be used to access them. Other actors were developed for reading user parameters for MEME and MAST and creating the Web service inputs. Figure 8 shows how these actors are connected - user inputs for MEME are first read before invoking the MEME service. Once the MEME analysis is complete, the output files are downloaded, and used to create the inputs to MAST, along with user supplied parameters. Once the MAST analysis is complete, the final outputs are available for the user to download any time. Most of the actors used in this workflow were generic ones provided by Kepler. However, the ones described above had to be written from scratch. Packaging and deployment of custom actors for other users is still an open issue, and efforts are currently underway to handle the same.

B. *Simple, unified environment for proteome analysis*

Some bioinformaticians or computer scientists shun graphics based environments because they take a long time to develop, become overly complex quickly and take away the time to solve critical problems at hand. The current Gfarm, CSF4 environment has been used with iGAP (integrative

Genome Annotation Pipeline) [57] extensively and has proven to be a stable environment for production experiments. The PRAGMA Gfarm testbed has been used for the development of this environment. The system takes advantage of the distributed data storage properties of Gfarm to reduce unnecessary file transfers over the network. The work environment is exactly like the command line interface of a cluster environment, except that the actual resources are distributed across the world. Users may use local batch schedulers to submit jobs to a local cluster, but use CSF4 to submit jobs to all the clusters [41].

C. *Hybrid QM/MM studies*

As a collaborative effort under the PRIUS [58] project, located at Osaka University, which supports the internationalization of graduate education via internships, we are developing a grid system for bio-molecular simulation with the OPAL Operation Provider. Basically the application system consists of two programs (QM and MM) [14]. Each program simulates behavior of molecules in short time step, and transfers data to each other at every time step. If a new job is submitted for every time step, the time wasted to reschedule jobs is proportionate to the number of time steps. The OPAL Operation Provider is used to launch the QM or MM programs once, and the Opal WSRF service is now continuously available for any incoming requests. We are in the process of identifying a workflow management tool to coordinate the data transfers as additional operation providers.

D. *Molecular visualization and visual programming in Vision*

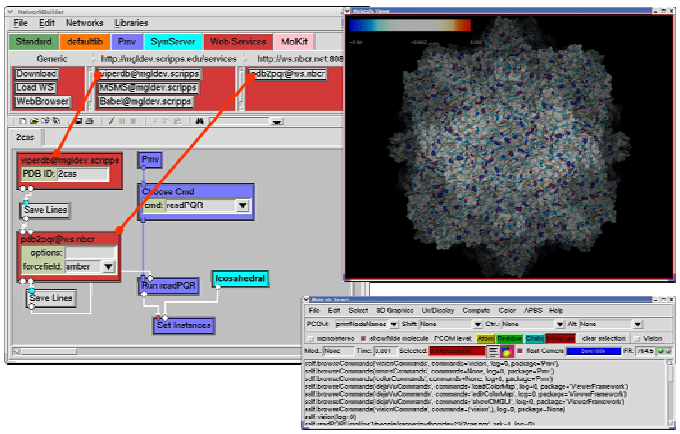

Figure 9. Users can access Opal based web services from Vision to retrieve database information or execute remote jobs, and visualize the results returned.

The Python Molecular Viewer (PMV) and the accompanying visual programming tool Vision are very popular tools for multiscale biomedical research [34], and many key packages have been reused in Continuity [48], a multiscale modeling package for the heart. Using Opal as the wrapper service for either an external application or a remote database application, a PMV/Vision user can access remote data or execute jobs and visualize the results right on his desktop (Figure 9).

IV. Summary and Discussion

There are a plethora of tools and innovative approaches to the development of cyberinfrastructure, as in the saying "All roads lead to Rome". However, the better and more robust approaches will always come out of close collaborations between computer scientists and biologists or other field specialists. The interactions will educate both groups to be fully aware of the requirements and challenges of the state of the art technology, and make routine use of the grid possible today. In addition, the development of new tools that support applications in different fields and through international collaborations greatly reduces the collective cost for global computational grids. The service oriented approach is gaining momentum and greatly facilitates the development of a knowledge based global economy.

A. *Collaborations needed to build robust grid computing platform*

As keenly observed in [59], there are conflicting requirements between biology and grid communities, with the former accustomed to a "one experiment at a time" approach, and the latter desiring systems that handle large number of jobs simultaneously. The reality is that current grid computing systems are just getting easier to learn, only relatively stable, and still limited by technologies available in hardware, software, and programming languages, as noted in [3]. Even with commodity clusters, the real work horse for many computational biologists to date, a user still has to be aware and design his applications not to overwhelm the network or file system I/O bandwidths. There are often so-called "non-malicious hackers" who cause serious disruption of services or "denial of service" attacks by not observing the limitations of the current system configurations. Therefore, it is a challenge to both communities to design better software and use them effectively. With these caveats in mind, one may attempt to select from the available tools, and build a robust platform to make routine use of the grid possible, through close collaborations with developers of all components involved where possible.

B. *System integration and interoperability*

The number of tools showed in Figure 1, along with those represented only as "…", offers many possible combinations to build end to end problem solving environments, often with overlapping features. For every tool listed, there are solid alternatives under different use cases, as often discussed in the publications for each tool mentioned. As shown in the usage scenarios, the ultimate choice of tools depends on the specific problems to be solved and the target audience of the designed environment. As demonstrated by the number of tools that are using Opal based services, the service oriented approach

provides the flexibility in customized front-end tools with transparent access to underlying distributed computation resources.

However, there are key challenges to services that can't be understood by machines, as the latter can process information much faster than the human brain. As outlined in the recently released NSF cyberinfrastructure Vision [60], strategic plans are needed for data, data analysis and visualization, with calls for a coherent data cyberinfrastructure in a "complex, global context". Necessary data standards, ontology, and tools for automated reasoning must be developed, to provide the semantic annotation for web services, and mechanisms for automated service discovery. NBCR has active research in this area, with several key applications such as PathSys [61], or OntoQuest [62].

C. *Application design considerations*

As discussed in the PRAGMA application deployment paper [17], there are many ways of application deployment on the grid, including the execution of existing applications without modifications in systems like Gfarm or Opal web service wrappers, or through the use of upper middleware such as APST [63, 64] or Nimrod/G [65], which enable legacy applications by automating the parameter sweep process. Other applications may use new programming models such as Ninf-G [66], MPICH-G2 [67], or MPICH-GX [68], which require more effort to redesign the applications, but could be as simple as recompilation with the MPICH-G2 library in the case of MPI-BLAST [69]. On the other hand, the communication overhead, and the network bandwidth heterogeneity, as well as the inherent instability of the grid mean that some applications need better fault detection and tolerance mechanisms built in, or they should not be run on the grid, or at least not yet without better execution environments.

D. *Workflow Management*

While a service oriented approach allows the applications to be designed as more independent "service units", there remains an efficient mechanism for management of complex workflows based on many services. While it is possible to encapsulate a predetermined workflow of coupled services, as in the case of online shopping networks, or a fixed procedure in parameter sweeps, the dynamic composition of web service based workflows remains a challenge. In addition, it is desirable to have standalone workflow execution engines that execute workflow plans on demand. This is an active research area, with many excellent presentations at a recent NETTAB workshop [70], as well as a GGF working group [71], and progress depends on data standards, web service standards, as well as workflow description language such as WSBPEL (Web Service Business Process Execution Language) defined through the Oasis standards consortium [72].

E. *Future works*

The usage of Opal based services can be significantly enhanced through the integration of XML Schema for description of data types, as well as use of OWL (web ontology language) [73] to annotate the services. Coupled with the availability of Opal WSRF operation providers and a generalized workflow management tool, the ability to compose complex workflows would be significantly enhanced. CSF4 based scheduling of web services or complex web service based workflows would enable the efficient sharing of grid resources. A grid portal interface to web services running on a virtual Gfarm file system with a metascheduler would significantly increase the usability of the grid to a large audience.

REFERENCES

[1] J. C. Wooley and H. S. Lin, "Catalyzing inquiry at the interface btween Computing and Biology," J. C. Wooley and H. S. Lin, Eds., . ed: National Academy of Science Press, Wahington, DC, 2005.

[2] L. Hood, "Foreword," in *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*, A. D. Baxevanis and B. F. F. Ouellette, Eds., 3rd ed. New York: Wiley-Interscience, 2004.

[3] PITAC, "Computational Science: Ensuring America's Competitiveness," in *National Coordination Office for Networking and Information Technology Research and Development*, P. s. I. T. A. Committee, Ed., 2005, http://www.nitrd.gov/pitac/reports/index.html.

[4] J. T. Oden, T. Belytschko, J. Fish, T. J. Hughes, C. Johnson, D. Keyes, A. Laub, L. Petzold, D. Srolovitz, and S. Yip, "Revolutionizing Engineering Science through Simulation," 2006.

[5] I. Foster and C. Kesselman, "The Grid 2: Blueprint for a New Computing Infrastructure," 2 ed. San Francisco: Morgan Kaufmann Publishers, Inc., 2004.

[6] S. L. Graham, M. Snir, and C. A. Patterson, *Getting Up to Speed: The Future of Supercomputing*: The National Academies Press, 2004.

[7] D. E. Atkins, K. K. Droegemeier, S. I. Feldman, H. Garcia-Molina, M. L. Klein, D. G. Messerschmitt, P. Messina, J. P. Ostriker, and M. H. Wright, "Revolutionizing Science and Engineering Through Cyberinfrastructure," National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure, Washington, D.C. 2003.

[8] J. C. Wooley, "Bioinformatics and Computational Biology: The Interface between Computing and the Biosciences," *Asia Pacific Biotech*, vol. 10, 2006.

[9] CTWatch, "Cyberinfrastructure Technology Watch," 2006, http://www.ctwatch.org.

[10] myGrid, "The myGrid Project," 2004, http://www.mygrid.org.uk.

[11] R. D. Stevens, A. J. Robinson, and C. A. Goble, "myGrid: personalised bioinformatics on the information grid," *Bioinformatics*, vol. 19, pp. i302-i304, 2003.

[12] SoapLab, "Soaplab," vol. 2006, 2004, http://sourceforge.net/projects/soaplab/.

[13] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, "Taverna: a tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, vol. 20, pp. 3045-54, 2004.

[14] H. Nakamura, S. Date, H. Matsuda, and S. Shimojo, "A challenge towards next-generation research infrastructure for advanced life science," *New Generation Computing*, vol. 22, pp. 157-166, 2004.

[15] P. W. Arzberger and P. Papadopoulos, "PRAGMA: Example of Grass-Roots Grid Promoting Collaborative e-Science Teams," in *CTWatch Quarterly*, vol. Feb 2006, 2006.

[16] C. Zheng, D. Abramson, P. W. Arzberger, S. Ayuub, C. Enticott, S. Garic, M. Katz, J.-H. Kwak, B. S. Lee, P. M. Papadopoulos, S. Phatanapherom, S. Sriprayoonsakul, Y. Tanaka, Y. Tanimura, O. Tatebe, and P. Uthayopas, "The PRAGMA Testbed: building a multi-application international grid," presented at CCGrid, Singapore, 2006.

[17] D. Abramson, A. Lynch, H. Takemaya, Y. Tanimura, S. Date, H. Nakmura, S. Hwang, K. Jeong, H.-C. Lee, C.-W. Wang, K. K. Baldridge, W. W. Li, and P. W. Arzberger, "Deploying Scientific Applications to the PRAGMA Grid testbed: Strategies and Lessons," presented at CCGrid, Singapore, 2006.

[18] W. W. Li, P. W. Arzberger, and A. McCulloch, "Developing End-to-End Cyberinfrastructure for Multiscale Modeling in Biomedical Research," in *CTWatch Quarterly*, 2006, pp. In Press.

[19] I. Foster, "Service-oriented science," *Science*, vol. 308, pp. 814-7, 2005.

[20] T. Erl, *Service-Oriented ARchitecture: a Field Guide to Integrating XML and Web Services*, 1st ed. Upper Saddle River: Prentice Hall, 2004.

[21] Globus, "The Globus Alliance," 2004, http://www.globus.org.

[22] WSRF, "Web Services Resource Framework," 2004, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf.

[23] H. Bal, H. Casanova, J. Dongarra, and S. Matsuoka, "Application-Level Tools," in *The Grid 2*, I. Foster and C. Kesselman, Eds., 2 ed. Amsterdam: Elsevier, 2004.

[24] S. Krishnan, B. Stearn, K. Bhatia, K. Baldridge, W. W. Li, and P. W. Arzberger, "Opal: Simple Web Service Wrappers for Scientific Applications," presented at International Conference for Web Services, 2006.

[25] J. Gawor and S. Meder, "GT4 WS Java Core Design," 2004, http://www.globus.org/toolkit/docs/development/wsrf/3.9.0/WSRFDesign.doc.

[26] TGSG, "Tera Grid Science Gateways," 2006, http://www.teragrid.org/programs/sci_gateways/.

[27] K. Bhatia, S. Chandra, and K. Mueller, "GAMA: Grid Account Management Architecture," presented at 1st IEEE International Conference on e- Science and Grid Computing, Melbourne, Australia, 2006.

[28] L. Perlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, "A Community Authorization Service for Group Collaboration," presented at IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[29] J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy," presented at High Performance Distributed Computing (HPDC), 2001.

[30] W. Link, "CACL, A CA System with Automated User Authentication," in *SDSC Technical Report*, 2003, http://www.npaci.edu/CA/cacl.pdf.

[31] NAREGI, "National Research Grid Initiative," 2006, http://www.naregi.org/index_e.html.

[32] Gridsphere, "GridSphere," 2004, http://www.gridsphere.org.

[33] K. Baldridge, K. Bhatia, J. P. Greenberg, B. Stearn, S. Mock, W. Sudholt, S. Krishnan, A. Bowen, C. Amoreira, and Y. Potier, "GEMSTONE: Grid Enabled Molecular Science Through Online Networked Environments," presented at Life Sciences Grid Workshop, Singapore, 2005.

[34] M. F. Sanner, M. Stolz, P. Burkhard, X.-P. Kong, G. Min, T.-t. Sun, S. Driamov, U. Aebi, and D. Stoffler, "Visualizing Nature at Work from the Nano to the Macro Scale.," in *Proteomics and Bioinformatics*, vol. 1, *: John Wiley & sons, Ltd., 2005, pp. 7-11.

[35] KEPLER, "The KEPLER project," 2006, http://kepler-project.org/.

[36] Gfarm, "Grid Data Farm," 2004, http://datafarm.apgrid.org/software/#download.

[37] FUSE, "Filesystem in USEr space," vol. 2006, 2006, http://fuse.sourceforge.net/.

[38] W. W. Li, C. L. Yeo, K. Jeong, S. Hwang, S. Date, J. Kwak, S. Sekiguchi, L. Ang, and P. W. Arzberger, "Proteome Analysis using iGAP in Gfarm.," presented at Life Sciences Grid Workshop, Singapore, 2005.

[39] CSF, "Community Scheduler Framework," vol. 2005, 2005, http://sourceforge.net/projects/gcsf/.

[40] COG, "Commodity Grid Kits," 2004, http://www-unix.globus.org/cog/.

[41] X. Wei, Z. Ding, W. W. Li, O. Tatebe, J. Jiang, L. Hu, and P. W. Arzberger, "Grid Infrastructure for Bioinformatics Applications Based on CSF4," *Future Generations of Computer Systems*, pp. In Press, 2006.

[42] X. Wei, J. Jiang, W. W. Li, O. Tatebe, G. Xu, L. Hu, and J. Ju, "Implementing Data Aware Scheduling and Data Management in Gfarm using LSF™ Scheduler Plugin Mechanism," *Future Generation of Computer Systems*, 2006.

[43] K. K. Baldridge, K. Bhatia, J. P. Greenberg, B. Stearn, S. Mock, W. Sudholt, S. Krishnan, A. Bowne, C. Amoreira, and Y. Potier, "Grid-Enabled Molecular Science through Online Networked Environments," presented at Life Sciences Grid Workshop, Singapore, 2005.

[44] T. J. Dolinsky, J. E. Nielsen, J. A. McCammon, and N. A. Baker, "PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations," *Nucleic Acids Res*, vol. 32, pp. W665-7, 2004.

[45] ROCKS, "Rocks Cluster Distribution," 2005, http://www.rocksclusters.org.

[46] M. Litzkow, M. Livny, and M. Mutka, "Condor - a hunter of idle workstations.," presented at Proceedings of the 8th International Conference of Distributed Computing Systems, 1988.

[47] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon, "Electrostatics of nanosystems: application to microtubules and the ribosome," *Proc Natl Acad Sci U S A*, vol. 98, pp. 10037-41, 2001.

[48] Continuity, "Continuity," 2006, http://www.continuity.ucsd.edu.

[49] G. M. Morris, D. S. Goodsell, R. Huey, and A. J. Olson, "Distributed automated docking of flexible ligands to proteins: parallel applications of AutoDock 2.4," *J Comput Aided Mol Des*, vol. 10, pp. 293-304, 1996.

[50] D. S. Goodsell, G. M. Morris, and A. J. Olson, "Automated docking of flexible ligands: applications of AutoDock," *J Mol Recognit*, vol. 9, pp. 1-5, 1996.

[51] Y. Song, Y. Zhang, T. Shen, C. L. Bajaj, J. A. McCammon, and N. A. Baker, "Finite element solution of the steady-state Smoluchowski equation for rate constant calculations," *Biophys J*, vol. 86, pp. 2017-29, 2004.

[52] FEtk, "Finite Element Toolkit," 2006, http://www.fetk.org.

[53] T. L. Bailey and C. Elkan, "The value of prior knowledge in discovering motifs with MEME," *Proc Int Conf Intell Syst Mol Biol*, vol. 3, pp. 21-9, 1995.

[54] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," *Proc Int Conf Intell Syst Mol Biol*, vol. 2, pp. 28-36, 1994.

[55] T. L. Bailey, N. Williams, C. Misleh, and W. W. Li, "MEME: Discovering and analyzing DNA and protein sequence motifs," *Nucleic Acids Res*, pp. In Press, 2006.

[56] MyWorkSphere, "My WorkSphere," 2006, https://www.nbcr.net/worksphere.

[57] W. W. Li, G. B. Quinn, N. N. Alexandrov, P. E. Bourne, and I. N. Shindyalov, "A comparative proteomics resource: proteins of Arabidopsis thaliana," *Genome Biol*, vol. 4, pp. R51, 2003.

[58] PRIUS, "Pacific RIm UniverSities," 2006, http://prius.ics.es.osaka-u.ac.jp/.

[59] K. Jeong, S. Jung, S. Hwang, and J. Lee, "An Overview of the MGrid Project," presented at HPC Asia, Singapore, 2005.

[60] NSFCC, "NSF's Cyberinfrastructure Vision for 21st Century Discovery," 2006, http://www.nsf.gov/od/oci/ci_v5.pdf.

[61] M. Baitaluk, X. Qian, S. Godbole, A. Raval, A. Ray, and A. Gupta, "PathSys: Integrating Molecular Interaction Graphs for Systems Biology," *BMC Bioinformatics*, pp. In Press, 2005.

[62] L. Chen, M. Martone, A. Gupta, and L. Fong, "OntoQuest: Exploring Ontological Data Made Easy," presented at Submitted, 2006.

[63] H. Casanova and F. Berman, "Parameter sweeps on the Grid with APST," in *Grid Computing: Making the Global Infrastructure a Reality*, F. Berman, G. C. Fox, and A. J. G. Hey, Eds. West Sussex: Wiley Publishers, Inc., 2003.

[64] A. Birnbaum, J. Hayes, W. W. Li, M. A. Miller, P. W. Arzberger, P. E. Bourne, and H. Casanova, "Grid Workflow Software for High-Throughput Proteome Annotation Pipeline," *Lecture Notes In Computer Science*, vol. In Press, 2004.

[65] D. Abramson, J. Giddy, and L. Kotler, "High performance parametric modeling with Nimrod/G: Killer application for the global grid?," presented at IPDPS, 2000.

[66] Y. Tanaka, H. Nakada, S. Sekiguchi, T. Suzumura, and S. Matsuoka, "Ninf-G: A Reference Implementation of RPC-based Programming Middleware for Grid Computing.," *J. of Grid Computing*, vol. 1, pp. 41-51, 2003.

[67] MPICH-G2, "MPICH-G2," 2006, http://www3.niu.edu/mpi/.

[68] MPICH-GX, "MPICH-GX: Extension of MPI functionality for the GRID ", vol. 2006, 2006, http://www.moredream.org/mpich.htm.

[69] A. Darling, "The Design, Implementation, and Evaluation of mpiBLAST,," presented at ClusterWorld, San Jose, 2003.

[70] NETTAB, "Workflows management: new abilities for the biological information overflow," 2005, http://www.nettab.org/2005/.

[71] WMRG, "Workflow Management Research Group," 2004, http://www.isi.edu/~deelman/wfm-rg/.

[72] OASIS, "OASIS Consortium," 2006, http://www.oasis-open.org.

[73] OWL, "Web Ontology Language," 2006, http://www.w3.org/2004/OWL/.