Web & Grid Technologies in Bioinformatics, Computational and Systems Biology: A Review

Azhar A. Shah[1], Daniel Barthel[1], Piotr Lukasiak[2,3], Jacek Blazewicz[2,3] and Natalio Krasnogor[*,1]

[1]*School of Computer Science, University of Nottingham, Jubilee Campus, NG81BB, Nottingham, UK*

[2]*Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland*

[3]*Institute of Bioorganic Chemistry, Laboratory of Bioinformatics, Polish Academy of Sciences, Noskowskiego 12/14, 61-704 Poznan, Poland*

Abstract: The acquisition of biological data, ranging from molecular characterization and simulations (e.g. protein folding dynamics), to systems biology endeavors (e.g. whole organ simulations) all the way up to ecological observations (e.g. as to ascertain climate change's impact on the biota) is growing at unprecedented speed. The use of computational and networking resources is thus unavoidable. As the datasets become bigger and the acquisition technology more refined, the biologist is empowered to ask deeper and more complex questions. These, in turn, drive a runoff effect where large research consortia emerge that span beyond organizations and national boundaries. Thus the need for reliable, robust, certified, curated, accessible, secure and timely data processing and management becomes entrenched within, and crucial to, 21st century biology. Furthermore, the proliferation of biotechnologies and advances in biological sciences has produced a strong drive for new informatics solutions, both at the basic science and technological levels. The previously unknown situation of dealing with, on one hand, (potentially) exabytes of data, much of which is noisy, has large experimental errors or theoretical uncertainties associated with it, or on the other hand, large quantities of data that require automated computationally intense analysis and processing, have produced important innovations in web and grid technology. In this paper we present a trace of these technological changes in Web and Grid technology, including details of emerging infrastructures, standards, languages and tools, as they apply to bioinformatics, computational biology and systems biology. A major focus of this technological review is to collate up-to-date information regarding the design and implementation of various bioinformatics Webs, Grids, Web-based grids or Grid-based webs in terms of their infrastructure, standards, protocols, services, applications and other tools. This review, besides surveying the current state-of-the-art, will also provide a road map for future research and open questions.

Keywords: Semantic web, web services, web agents, grid computing, middleware, *in silico* experiments.

1. INTRODUCTION

'*The impact of computing on biology can fairly be considered a paradigm change as biology enters the 21st century. In short, computing and information technology applied to biological problems is likely to play a role for 21st century biology that is in many ways analogous to the role that molecular biology has played across all fields of biological research for the last quarter century and computing and information technology will become embedded within biological research itself*' [1].

As a visualization of the above referred conclusion regarding the embedding of computing and information technology (IT) in biological research, one can look at the current state-of-the-art in web and grid technologies as applied to bioinformatics, computational biology and systems biology. The World Wide Web or simply the *web* has revolutionized the field of IT and related disciplines, by providing information-sharing services on top of the internet. Similarly, grid technology has revolutionized the field of computing by providing location-independent resource sharing-services such as computational power, storage, databases, networks, instruments, software applications and other computer related hardware equipment. These information and resource-sharing capabilities of web and grid technologies could upgrade a single user computer into a global supercomputer with vast computational power and storage capacity. The so called upgraded web and grid-enabled global super computer would make itself a potential candidate to be used in resource-hungry computing domains. For example, it could be used to efficiently solve complex calculations such as parameter sweep scenario with Monte Carlo simulation and modeling techniques, which would normally require several decades of execution time on a traditional single desktop processor.

A quick look at the literature reveals that web and grid technologies are continuously being taken up by the biological community as an alternate to traditional monolithic high performance computing mainly because of the inherent nature of biological resources (distributed, heterogeneous and CPU intensive), smaller financial costs, better flexibility, scalability and efficiency offered by the web and grid-enabled environment. An important factor that provides the justification behind the ever growing use of web and grid technologies in life sciences is the continuous and rapid increase in biological data production. About eight years ago it has been approximated that the amount of information produced by a single gene laboratory could be as higher as 100

*Address correspondence to this author at the ASAP Group, School of Computer Science, University of Nottingham, Jubilee Campus, NG81BB, Nottingham, UK; Tel: +44 115 8467592; Fax: +44 115 8467066; E-mail: Natalio.Krasnogor@Nottingham.ac.uk

1574-8936/08 $55.00+.00

terabytes, which is equivalent to about 1 million encyclopedias [2]. Although being extensive, these data also differ in terms of storage and access technologies and are dispersed throughout the world. Currently, there are no uniform standards, or at least not yet been adopted properly by the biological community as a whole, to deal with the diverse nature, type, location and storage formats of this data. On the other hand, in order to obtain the most comprehensive and competitive results, in many situations, a biologist may need to access several different types of data which are publicly available in more than 700 [3] biomolecular databases. One way to handle this situation would be to convert the required databases into a single format and then store it on a single storage device with extremely large capacity. Considering the tremendous size and growth of data, this solution seems to be infeasible, inefficient and very costly. The application of Web and Grid technology provides an opportunity to standardize the access to these data in an efficient, automatic and seamless way through the use of appropriate DataGrid middleware technologies. These technologies include grid middleware specific Data Management Services (DMS), distributed storage environments such as Open Source Grid Services Architecture-Data Access and Integration (OGSA-DAI) (http://www.ogsadia.org) with Distributed Query Processing (OGSA-DQP), Storage Resource Broker (SRB) [4] and IBM DiscoveryLink [5] middleware etc.

Furthermore, it is also very common for a typical biological application that involves very complex analysis of large-scale datasets and other simulation related tasks to demand for high throughput computing power in addition to seamless access to very large biological datasets. The traditional approach towards this solution was to purchase extremely costly special-purpose super computers or dedicated clusters. This type of approach is both costly and somewhat limited as it locks the type of computing resources. Another problem associated with this approach would be that of poor utilization of very costly resources, i.e. if a particular application finishes its execution then the resources could remain idle. Grid technology on the other hand provides more dynamic, scalable and economical way of achieving as much computing power as needed through computational grid infrastructures connected to a scientist's desktop machine. There are many institutional, organizational, national and international Data/Computational/Service Grid testbeds and well established production grid environments, which provide these facilities free of charge to their respective scientific communities. Some of these projects include Biomedical Research Informatics Delivered by Grid Enabled Services(BRIDGES) (http://www.brc.dcs.gla.ac.uk/projects/), Enabling Grids for E-scienceE project (EGEE) (http://public.eu-egee.org), Biomedical Informatics Research Network project (BIRN) (http://www.nbirn.net), National Grid Service UK (http://www.ngs.ac.uk), OpenBioGrid Japan [6], SwissBioGrid [7], Asia Pacific BioGrid (http://www.apgrid.org), North Carolina BioGrid (http:// www.ncbiotech.org), KidneyGrid [8], Virtual Laboratory for drug design on World Wide Grid [9] etc. All these projects consist of an internet based interconnection of a large number of pre-existing individual computers or dedicated clusters located at various distributed institutional and organizational sites that are part of the consortium.

Some other large scale bioinformatics grid projects have provided a platform where a biologist can design and run complex *in silico* experiments by combing several distributed and heterogeneous resources that are wrapped as web-services. Examples of these are myGrid [10, 11], BioMOBY [12-14], Seqhound (http://www.blueprint.org /seqhound) and Biomart (http://www.biomart.org) etc., which allow for the automatic discovery and invocation of many bioinformatics applications, tools and databases such as EMBOSS [15] suite of bioinformatics applications and some other publicly available services from the National Center for Biomedical Informatics (NCBI) (http://www.ncbi. nlm.nih.gov) and European Bioinformatics Institute (EBI) (http://www.ebi. ac.uk). These projects also provide some special toolkits with necessary application programming interfaces (APIs), which could be used to transform any legacy bioinformatics application into a web-service that can be deployed on their platforms. The availability of these BioGrid projects brought into sharp focus the need for better user interfaces as to provide the biologist with easier access to these web/grid resources. This has led to the development of various web based interfaces, portals, workflow management systems, problem solving environments, frameworks, application programming environments, middleware toolkits, data and resource management approaches along with various ways of controlling grid access and security. This review attempts to provide an up-to-date coherent and curated overview of the most recent advances in these technologies as applied to life sciences. The review aims at providing a complementary source of additional information to some previous reviews in this field such as [16, 17].

The organization of the paper is as follows: *section 2* presents an overview of the state-of-the-art on web and grid technologies focusing on the direction of progress of each technology and how they are becoming part and parcel of life sciences. *Section 3* describes the architecture, implementation and services provided by a selection of *'flagship'* projects. The idea is to present some success stories with brief architectural details that could provide the basic building blocks to a researcher interested in the further exploration and utilization of web and grid technologies in life sciences. *Section 4* presents the analysis of the reviewed literature with a clear indication of certain key open problems within the existing technological approaches and provides a roadmap and open questions for the future. Finally, *section 5* provides the concluding remarks.

2. STATE-OF-THE-ART OVERVIEW

Among the many advances that the computational sciences have provided to the life sciences, the proliferation of web and grid technologies is one of the most conspicuous. Driven by the demands of biological research, these technologies have moved from their classical and somewhat static architectures to more dynamic and service-oriented ones. The direction of current development in these technologies is coalescing towards an integrated and unified Web-based grid service [18] or Grid-based web service environment (Fig. **2**). Accompanying this rapid growth, a huge diversity of approaches to implementation and deployment routes have been investigated in relation to the use of various innovative web and grid technologies for the solution of problems related to life sciences. This section provides an

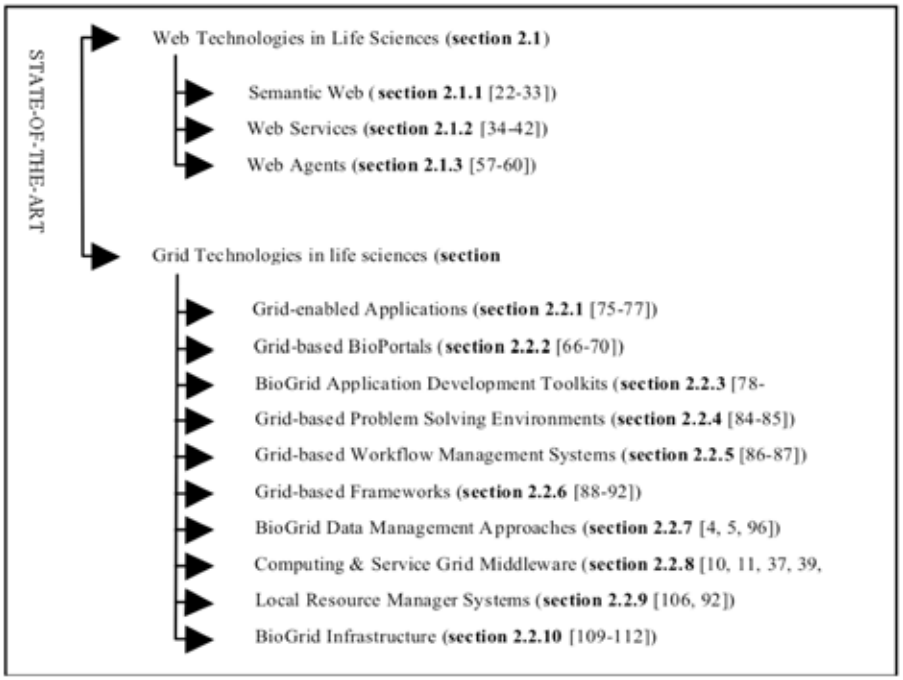


Fig. (1). Hieratical Organization of the state-of-the-art overview.

overview of some of these works through a hierarchal organization as illustrated in Fig. **1**.

2.1. Application of Web Technologies in Life Sciences

As can be observed from Fig. **2**, currently there are three main thrusts in the development of web technologies: Semantic-web, Web-services and web-agents. The continuous growth of these technologies takes on a converging path giving rise to agent based semantic web services and web portals. The use and effect of these technologies in relation to life sciences are analyzed in the following three subsections:

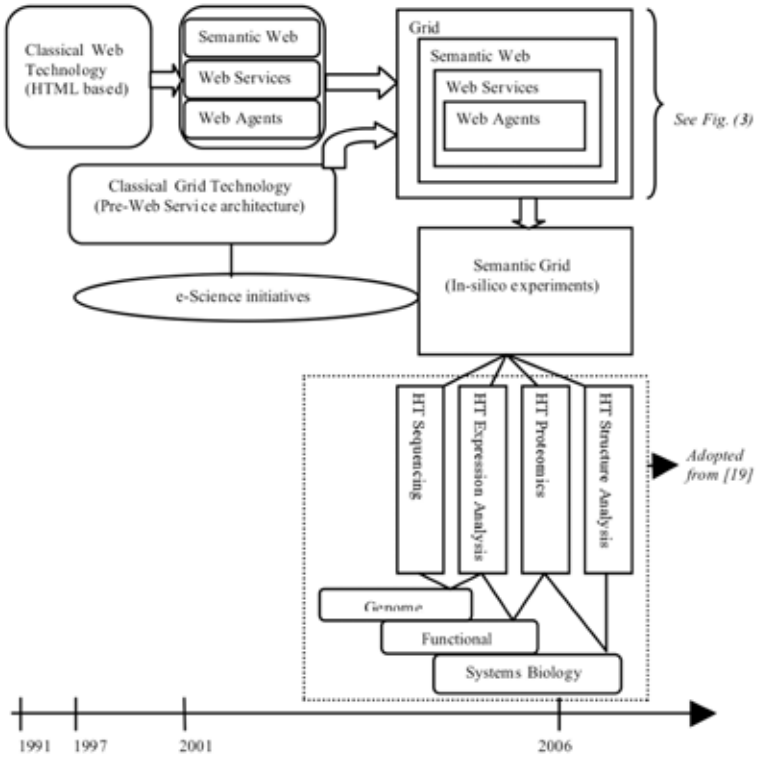


Fig. (2). Review of technological infrastructure for life sciences: Classical HTML based web started in 1991 and traditional Globus based grid was introduced by Ian Foster in 1997. With the introduction and development of semantic web, web-services and web agents in and after 2001, the new web and grid technologies are being converged into a single uniform platform termed as 'service-oriented autonomous semantic grid' that could satisfy the needs of HT (high throughput) [19] experimentations in diverse fields of life sciences as depicted above.

2.1.1. Semantic Web Technology

One of the most important limitations of the information shared through classical web technology is that it is only interpretable by human and hence it limits the automation required for more advanced and complex life science applications. Therefore, the basic purpose of semantic web technology is to eliminate this limitation by enabling the machine (computer) to interpret/understand the meaning (semantics) of the information and hence allow artificial intelligence based applications to carry out decisions autonomously. It does so by adding some important features to the basic information-sharing service provided by the classical web technology. These features provide a common format for interchange of data through some standard languages and data models such as XML (eXtensible Markup Language), RDF (Resource Description Framework) along with several variants of schema and semantic based markup languages such as, Web Ontology Language (OWL) and Semantic Web Rules-Language (SWRL) etc. Each of these models and languages has its own benefits and limitations. Therefore, a particular model or language being highly useful for the solution of one problem at some point in time may not be suitable for another problem or even the same problem at another point in time. For example, Wang *et al.* [21] argues that although initially XML was used as a data standard for platform independent exchange and sharing of data, however, because of its basic syntactic and document-centric nature, it was found limited, especially for the representation of rapidly increasing and diverse *'omic' data.* Therefore, currently RDF along with some new variants of OWL such as OWL-Lite, OWL-DL and OWL-Full are being adopted for implementation that would free the end-user (biologist) from performing manual invocation of analysis tools and interpretation of partial results needed for further execution of remaining software components in a complex *in silico* experiment. It is therefore, we find various ways towards the use of semantic web for life sciences mainly focusing on data/application integration, data provenance, knowledge discovery, machine learning and mining etc. For example, Sooho *et al.* [22], discussed the development of a semantic framework based on publicly available ontologies such as *GlycO* and *ProPreO* that could be used for modeling the structure and function of enzymes, glycans and pathways. This framework uses a sublanguage of OWL called OWL-DL [23] to integrate extremely large (~500MB) and structurally diverse collection of biomolecules. Biological database integration as discussed here, also encounter the problem of inconsistencies between databases. Therefore, there have also been certain efforts for providing some external semantic-based tools for the measurement of the degree of inconsistencies between different databases. One such effort is discussed by Chen *et al.* [24]. It describes an ontology-based method to determine if two databases are compatible. The database compatibility determination is based on the results of semantically matching the reference attributes though a mathematical function. Several practical examples have been demonstrated with encouraging results. This measure can be used as criteria to decide whether a particular database can be integrated or not and hence making the process of integration more predictable, efficient and reliable.

The autonomous and uniform integration, invocation and access to biological data and resources as provided by semantic web have also created an environment that supports the use of *in silico* experiments. Proper and effective use of *in silico* experiments requires the maintenance of user specific provenance data such as record of goals, hypothesis, materials, methods, results and conclusions of an experiment. For example, Zhao *et al.* [25] showcased the design of a RDF based provenance log for a typical *in silico* experiment that performs DNA sequence analysis as a part of myGrid [10, 11] middleware services. The authors have reported the use of Life Science Identifiers (LSIDs) for achieving location independent access to distributed data and metadata resources. Similarly, RDF and OWL have been used for associating uniform semantic information and relationships between resources, while Haystack [26], a semantic web browser, has been used for delivering the provenance-based web pages to the end-user. The use of RDF model as compared to XML provides more flexible and graph-based resource description with location independent resource identification through URIs (Universal Resource Identifier).

There are various other significant contributions that illustrate the use of semantic web technology for the proper integration and management of biological data. For example, The Gene Ontology Consortium [27, 28], make uses of semantic web technologies to provide a central gene ontology resource for unification of biological information. Ruttenberg *et al.* [29, 30] used OWL-DL to develop a data exchange format that facilitates integration of biological pathway knowledge. Similarly, Whetzel *et al.* [31] and Navarange *et al.* [32] used semantic web to provide a resource for the development of tools for microarray data acquisition and query according to the concepts specified in Minimum Information About a Microarray Experiment (MIAME) standard [33]. Further information about some other semantic web and ontology based applications and tools for life sciences is presented in Table **1**.

2.1.2. Web Service Technology

Web-service technology further extends the capabilities of classical web and semantic web by allowing information and resources to be shared among machines even in a distributed heterogeneous environment (such as a grid environment). Therefore, applications developed as web services can easily *interoperate* with peer applications. Web services are defined through Web Service Description Language (WSDL) and deployed and discovered through Universal Description, Discovery and Integration (UDDI) protocol. They can exchange XML based messages through Simple Object Access Protocol (SOAP) over different computer platforms. Furthermore, with the introduction of Web Service Resource Framework (WSRF), now web services have become more capable of storing the state information during the execution of a particular transaction. These features of web-services have made them extremely important to be applied to life science domain. Today many life science applications are being developed as web services. For example, the National Center for Biotechnology Information (NCBI) provides a wide range of biological databases and analytical tools as web services such as all the Entrez e-utilities

Table 1. Semantic-Web and Ontology Based Resources

Semantic Web Based Application/Tool	Full Name and Source	Usage
BioPAX [43, 44]	Biological Pathway Exchange http://www.biopax.org/	Data exchange format for biological pathway data
MGED [45, 46]	Microarray for Gene Expression Data http://www.mged.org/	Data standard for Systems Biology
TAMBIS [47]	Transparent Access to Multiple Bioinformatics Information Sources http://img.cs.man.ac.uk/tambis	Biological Data Integration
caCORE SDK [48]	Software Development Kit for cancer informatics http://ncicb.nci.nih.gov/infrastructure/cacoresdk	Semantically integrated bioinformatics software system
AutoMed Toolkit [49]	AutoMatic Generation of Mediator Tools for Heterogeneous Data Integration http://www.doc.ic.ac.uk/automed/	Tools for assisting transformation and integration of distributed data
Gaggle [50]	Gaggle http://gaggle.systemsbiology.org/docs/	An integrated environment for systems biology
EcoCyc [51]	Encyclopedia of Escherichia coli K-12 Genes and Metabolism http://ecocyc.org/	Molecular catalog of the E. coli cell
SBML [52]	Systems Biology Markup Language http://sbml.org/index.psp	Computer-readable models of biochemical reaction networks.
CellML [53]	Cell Markup Language http://www.cellml.org/	Storage and exchange of computer-based mathematical models for biomolecular simulations
OBO [54]	Open Biomedical Ontology http://obo.sourceforge.net/	Open source controlled-vocabularies for different biomedical domains
GO [28]	Gene Ontology http://www.geneontology.org/	Controlled-vocabulary for genes
GMOD [55]	Generic Model Organism Database http://www.gmod.org/home	An integrated organism database
PSI-MI [56]	Proteomics Standards Initiative: Molecular Interactions http://psidev.sourceforge.net/	Data standard for proteomics

including EInfo, ESearch, EPost, ESummary, EFetch, ELink, MedLine, and PubMed. Similarly, the European Institute for Bioinformatics (EBI) provides many biological resources as web services such as SoapLab, WSDbfetch, WSfasta, WSBLast, WSInterProScan, EMBOSS amongst others. A comprehensive list of all publicly available and accessible biological web services developed by different organizations, institutions and groups can be found at myGrid website (http://taverna.sourceforge.net).

All these web services can be used as part of complex application specific programs. IBM provides WebSphere Information Integrator (WS II) as an easy way for developers to integrate individual web-service components into large programs. As an example, North Carolina BioGrid (NC BioGrid) [34] in collaboration with IBM uses web services to integrate several bioinformatics applications to high performance grid computing environment. The NC BioGrid also provides a tool (*WSDL2Perl)* to facilitate the wrapping of Perl based legacy bioinformatics applications as web services. Other open source projects that provide registry, discovery and use of web services for biosciences include BioMOBY [12-14], myGrid [10, 11], and caBIG (https://cabig.nci.nih.gov/) etc.

The integration and interoperatibility of distributed and heterogeneous biological resources through web services have also opened an important niche for data mining and knowledge discovery. For example, Hahn U *et al.* [35] introduced web based reusable text mining middleware services that could be used for medical knowledge discovery. The middleware provides a Java based API for clients to call searching and mining services. Similarly, Hong *et al.* [36] used web services for the implementation of a microarray data mining system for drug discovery. Due to the success of web services to provide flexible, evolvable and scalable architectures with interoperability, between heterogeneous applications and platforms, the grid middleware is also being transformed from its pre-web service versions to the new ones based on web services. There are several initiatives in this direction such as Globus [37], an open source grid middleware, which has adopted web service architecture in its current version of Globus Toolkit 4; the EGEE project is also moving from its pre-web service middleware LCG2 [38] to a new web service based middleware named gLite [39]; and similarly Imperial College E-science Network Infrastructure (ICENI) [40] and myGrid are also adopting the web services through the use of *Jini*, *OGSI*, *JXTA* and other technologies [41, 42].

2.1.3. Agent-Based Semantic Web-services

Agents are described as software components that exhibit autonomous behavior and are able to communicate with their peers in a semantically defined high-level language such as FIPA-ACL (Foundation of Intelligent Physical Agents-Agents Communication Language). Since the main focus of agent technology is to enable the software components to perform certain tasks on behalf of the user, this somewhat relates and supports the goals of web-services technology and hence the two technologies have started converging to-

wards the development of more *autonomous web-services* that exhibit the behavior of both web-services as well as agents. There have been many attempts regarding the use of agents in bioinformatics, computational biology and systems biology. For example, Merelli *et al.* [57], reported the use of agents for the automation of bioinformatics tasks and processes such as phylogenetic analysis of diseases, protein secondary structure prediction, stem cell analysis, and simulation among others. In their paper, the authors also highlight some key open challenges in agents research: analysis of mutant proteins, laboratory information management system (LIMS), cellular process modeling and formal and semi-formal methods in bioinformatics. Similarly, the use of mobile agents for the development of a decentralized, self-organizing peer-to-peer grid computing architecture for computational biology has been demonstrated in [66], which we have selected as one of our case studies and briefly described in *section 3.2*. Similarly, *Luc et al.* [58], suggested the use of agents in myGrid [10, 11] middleware in order to best fit the ever-dynamic and open nature of biological resources. In particular, the authors propose the use of agents for '*personalization, negotiation and communication'*. The personalization agent can act on behalf of the user to automatically provide certain preferences such as the selection of preferred resources for a workflow based *in silico* experiment and other user related information. The user-agent can store these preferences and other user related information from previously conducted user activities and thus freeing the user from tedious repetitive interactions. The user-agent could also provide a point of contact for notification and other services requiring user interaction during the execution of a particular experiment. Other experiences related to the use of agents for biological data management and annotations have also been discussed in [59, 60].

2.2. Application of Grid Technologies in Life Sciences

Since its inception, the main focus of grid technology has been to provide platform independent global and dynamic resource-sharing service in addition to co-ordination, manageability, and high performance. In order to best satisfy these goals, its basic architecture has undergone substantial changes to accommodate other emergent technologies. As shown in Fig. **2**, the grid has moved from its initial static and pre-web service architecture to a more dynamic Web Service Resource Framework (WSRF) based Open Grid Service Architecture (OGSA) [61]. This architecture combines existing grid standards with emerging Service Oriented Architectures (SOAs) and web technologies in order to provide an innovative grid architecture known as *service-oriented semantic grid*. The main characteristics of this service-oriented semantic grid would be to maintain intelligent agents that could act as software services (grid services) capable of performing well-defined operations and communicating with peer services through uniform standard protocols. This paradigm shift in the grid's architecture is deemed to have a significant impact on the usability of bioinformatics, computational biology and systems biology. The impact is also evident from the very large number of literary contributions that report the experiences of using grid technologies for these domains. We have tried to summarize the findings of some of these contributions under the relevant categories of a typical BioGrid infrastructure. Some important components of a typical BioGird infrastructure are shown in Fig. **3**, and are briefly described in following ten subsections:

2.2.1. Grid-Enabled Applications and Tools

The actual process of developing (writing the code), deploying (registering, linking and compiling), testing (checking the results and performing debugging if necessary) and executing (scheduling, coordinating and controlling) an application in a grid-based environment is far from trivial. Mainly, the difficulties faced by developers arise because of incapability of traditional software development tools and techniques to support the development of some sort of virtual application or workflows, whose components can run on multiple machines within heterogeneous and distributed environment like grid [62]. Despite these difficulties, there are several grid-enabled applications for life sciences [17], mostly developed by using either standard languages such as Java along with message passing interfaces (e.g. MPICH/G) or web services. For example, Jacq *et. al.* [63] reported the deployment of various bioinformatics applications on the European Data Grid (EDG) testbed project. One of the deployed applications was *PhyloJava, a* GUI based application that calculates the polygenetic trees of a given genomic sequence using *fastDNAml* [64] algorithm. This algorithm uses *bootstrapping,* which is a reliable albeit computationally intensive technique that calculates the consensus from a large number of repeated individual tree calculations (about 500-1000 repeats). The *gridification* of this application was carried out at a granularity of 50 for a total of 1000 independent sequence comparison jobs (20 independent packets of 50 jobs each) and then merging the individual job results to get the final *bootstrapped* tree. The selection of the appropriate value of *granularity* depends upon the proper consideration of the overall performance because highly parallelized jobs can be hampered by resource brokering and scheduling times, whereas poorly parallelized jobs would not give significant CPU time gain [63]. The execution of this gridified application on the EDG testbed required the installation of a Globus [37] based EDG user interface on a Linux RedHat Machine, the use of Job Description Language (JDL) and the actual submission of the parallel jobs through Java Jobs Submission Interface (JJSI). It is reported that the gridified execution of this application provided 14 times speedup compared against a non-grid based standalone execution. The deviation in gain from ideal (speed up of 20) is considered to be the effect of network and communication overhead (latencies). Similarly, the gridification of other applications such as a grid-enabled bioinformatics portal for protein sequence analysis, grid-enabled method for securely finding unique sequences for PCR primers, and grid-enabled BLAST for orthology rules determination has also been discussed in [63] with successful and encouraging results. A brief description of some other grid-enabled applications is presented in Table **2**. Still there might be many other legacy applications that could take advantage of grid based resources; however, the migration of these applications to grid environment requires more sophisticated tools than what is currently available [65].

2.2.2. Grid-based BioPortals

As mentioned above, the actual process of developing and deploying an individual application on grid requires sig-

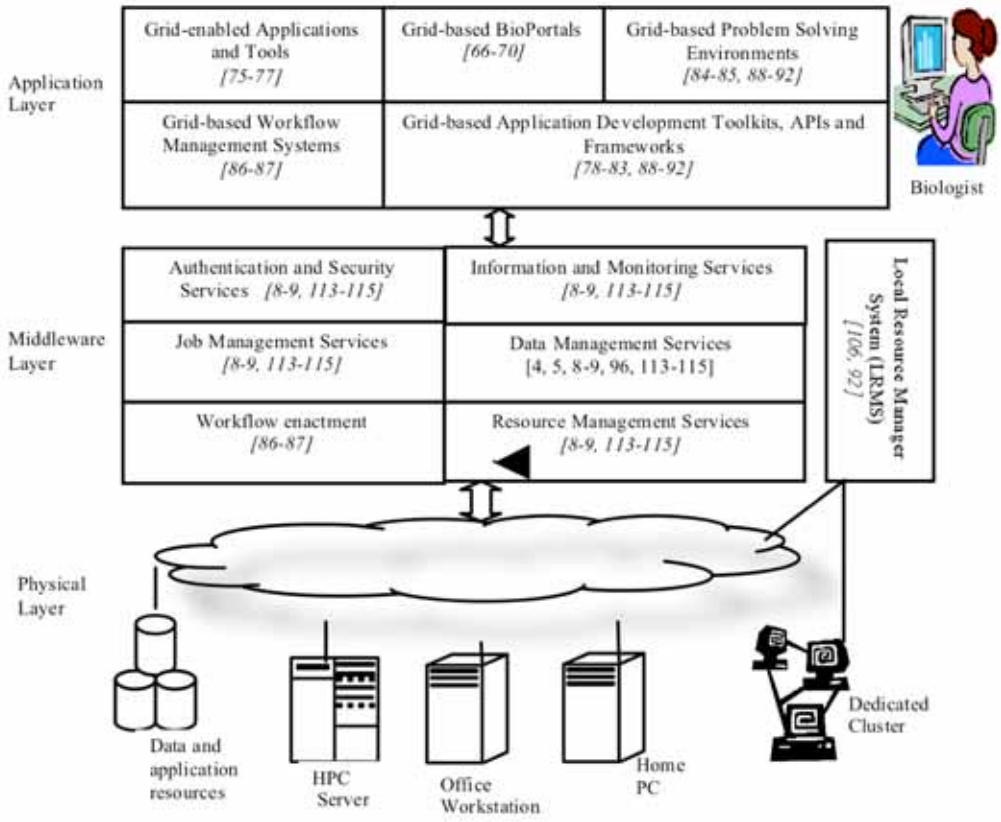


Fig. (3). Major components of a generic BioGrid infrastructure: application layer services help the user to design, execute, monitor and visualize the output of grid-enabled applications; middleware services provide access and management services for the use of physical layer resources; each site at physical layer has usually a pool of compute elements managed by some local resource manager system (LRMS) such as Sun Grid Engine (SGE), Portable Batch System (PBS), Load Sharing Facility (LSF), and Condor etc. (see subsections *2.2.1 to 2.2.10* for further details).

nificant level of expertise and considerable period of time. This issue hinders the usage of available grid infrastructures. Therefore, in order to enhance the use of different grid infrastructures, some individuals, groups, institution and organizations have started to provide the most frequently used and standard domain specific resources as grid-enabled services, which can be accessed by any or authenticated researcher through a common browser based single-point-of-access, without the need of installing any additional software. In this context a grid portal is considered to be an extended web-based application server with the necessary software capabilities to communicate with the backend grid resources and services [66]. This type of environment provides full level of abstraction and makes it easy to exploit the potential of grid seamlessly.

Grid-based portals are normally developed using some publicly available grid portal construction toolkits such as GridPort Toolkit [67, 68], NinF Portal Toolkit [66], GridSphere (http://www. gridsphere.org), IBM WebSphere (http://www-306.ibm.com/software/websphere) etc. Most of these toolkits follow the Java portlet specification (JSR 168) standard and thus make it easy for the developer to design the portal front-end and connect it to the backend resources through middleware services. For example, GridSphere [66], enables the developer to specify the requirements of the portal front-end (e.g. authentication, user interaction fields, job management, resources etc) in terms of an XML based file, which automatically generates a JSP file (through Java bases XML parser), that provides an HTML based web page for front-end. Similarly, it provides some general-purpose Java Servlets that can communicate to grid-enabled backend applications and resources though Globus based *GridRPC* mechanism. The toolkit also helps the developer for the gridification of applications and resources needed at the backend. It is because of this level of ease for the creation of grid-based portals that in [69] it is claimed that *'portal technology has become critical for future implementation of the bioinformatics grids'*. Another example is that of BRIDGES (http://www.brc.dcs.gla.ac.uk/projects/bridges) project, which provides portal-based access to many biological resources (federated databases, analytical and visualization tools etc) distributed across all the major UK centers with appropriate level of authorization, convenience and privacy. It uses IBM WebSphere based portal technology, because of its '*versatility and robustness'*. The portal provides a separate workspace for each user that can be configured by the user as per requirements and the configuration settings are stored using session management techniques. This type of environment can help in many important fields of life sciences such as the field of exploratory genetics that leads towards the understanding of complex disease phenotypes such as heart disease, addiction and cancer on the basis of analysis of data from multiple sources (e.g. model organism, clinical drug trials and research studies etc). Similarly [70] presents an-

Table 2. Grid-Enabled Applications

Grid-Enabled Application Task and source	Grid Middleware and Application Level Tools, Services and Languages	Effect of Gridification
GADU/GNARE [75] Task: Genome Analysis and Database Update http://compbio.mcs.anl.gov	◆ Globus Toolkit and Condor/G for distributing DAG based workflows ◆ GriPhyN Virtual Data System for workflow management. ◆ User interface to standard databases (NCBI, JGI etc.) and analysis tools (BLAST, PFAM etc.)	◆ Analysis of 2314886 sequences on a single 2GHz CPU can take 1061 days ◆ A grid with an average of 200 nodes took only 8 days and 16 hours for the above task.
MCell [76] Task: Computational biology simulation framework based on Monte Carlo algorithm http://www.mcell.cnl.salk.edu/	◆ Globus GRAM, SSH, NetSolve, PBS for remote job starting/monitoring ◆ GrdiFTP and scp for moving application data to grid ◆ Java based GUI, Relational Database (Oracle), Adoptive scheduling	◆ A typical r_disk MCell simulation on a single 1.5 GHz CPU can take 329 days ◆ A grid with an average of 113 dual CPU nodes took only 6 days and 6 hours for the above task.
Grid Cellware [77] Task: Modeling and Simulation for systems biology http://www.cellware.org	◆ Globus, Apache Axis, GridX-Meta Scheduler ◆ GUI based jobs creation editor ◆ Jobs mapped and submitted as web services	◆ Different stochastic (Gillespie, Gibson etc.), deterministic (Euler Forward, Runge–Kutta) and MPI based swarm algorithms have been successfully implemented in a way to distribute their execution on grid nodes.

other instance of system that is easy to use, is scalable and extensible, providing among others, secure and authenticated access to standard bioinformatics databases and analysis tools such as nucleotide and protein databases, BLAST [71], CLUSTAL [72] etc. A common portal engine was developed with the reusable components and services from Open Grid Computing Environment Toolkit (OGCE) [73] that combine the components of different individual grid portal toolkits. This common portal engine was integrated with a biological application frame work by using PISE [74] (web interface generator for molecular biology). The portal provides access to around 200 applications related to molecular biology and also provides the way to add any other application through the description of a simple XML based file.

2.2.3. BioGrid Application Development Toolkits

Although some general purpose grid toolkits such as Globus, COSM (http://www.mithral.com/projects/cosm) and GridLab (http://www.gridlab.org), provide certain tools (APIs and run time environments) for the development of grid-enabled applications, they are primarily aimed at the provision of low level core services needed for the implementation of a grid infrastructure. Therefore, it seems to be difficult and time consuming for an ordinary programmer to go through the actual process of developing and testing a grid enabled application using these toolkits; instead, there are some simulation based environments such as EDGSim (http://www.hep.ucl.ac.uk/~pac/EDGSim), extensible grid simulation environment [78], GridSim [79] and GridNet [80], that could be used at initial design and verification stage.

As different application domains require certain specific set of tools that could make the actual process of grid-enabled application development life-cycle (development, deployment, testing and execution) to be more convenient and efficient. One such proposal for the development of a *'Grid Life sciences Application Developer (GLAD)'* was presented in [81]. This publicly available toolkit works on top of the *ALiCE* (Adaptive scaLable Internet-based Computing Engine), a light weight grid middleware and provides a Java based grid application programming environment for life sciences. It provides a list of commonly used bioinformatics algorithms and programs as reusable library components along with other software components needed for interacting (fetching, parsing etc) with remote distributed and heterogeneous biological databases. The toolkit also assists in the implementation of task level parallelism (by providing effective parallel execution control system) for algorithms and applications ranging from those having regular computational structures (such as database searching applications) or irregular patterns (such as phylogenetic tree) [82]. Certain limitations of GLAD include the non-conformance of AliCE with OGSA standard and the use of socket based data communication, which might not be good for performance of critical applications. Another grid application development toolkit for bioinformatics that provides high level user interface with a problem solving environment related to biomedical data analysis has been presented in [83]. The toolkit provides a Java based GUI that enables the user to design a Direct Acyclic Graph (DAG) based workflow selecting a variety of bioinformatics tools and data (wrapped as java based JAX-RPC web services). The toolkit also enables the user to assign appropriate dependencies and relationships among the components of the workflow. Once the workflow is submitted on the grid, the scheduler would use the information from dependencies and relationships to determine the order of execution of individual components.

2.2.4. Grid-based Problem Solving Environments

The Grid-based Problem Solving Environment (PSE) is another way of providing a higher level of interface such as graphical user interface or web interface to an ordinary user so that he/she could design, deploy and execute any grid-enabled application related to a particular class of specific domain and visualize the results without knowing the underlying architectural and functional details of the backend resources and services. In fact, grid-based PSE brings the grid application programming at the level of drawing, that is, instead of writing the code and worrying about the compiling and execution, the user can just use appropriate GUI compo-

nents provided by the PSE to compose, compile and run the application in a grid environment. PSEs are developed using high level languages such as Java and are targeted to transform the user designed/modeled application into an appropriate script (distributed application or web service) that could be submitted to a grid resource allocation and management service for execution and on completion of the execution, the results are displayed through appropriate visualization mechanism.

There are several different grid-based PSEs available for bioinformatics applications e.g. *Cannataro M et al.* [84] described the design and architecture of a PSE (*Proteus*) that provides an integrated environment for biomedical researchers to search, build and deploy distributed bioinformatics applications on computational grids. The PSE uses semantic based ontology (developed in DAMIL+OIL language (http://www.daml.org)) to associate the essential meta-data such as *goals* and *requirements* to three main classes of bioinformatics resources such as data sources (e.g. SwissProt and PDB database), software components (e.g. BLAST, SRS, Entrez and EMBOSS an open source suite of bioinformatics applications for sequence analysis) and tasks/processes (e.g. sequence alignment, secondary structure prediction and similarity comparison) and stores this information in a meta-data repository. The data sources are specified on the basis of kind of biological data, its storage format and the type of the data source. Similarly, the components and tasks are modeled on the basis of the nature of tasks, steps and order in which tasks are to be performed, algorithm used, data source and type of the output etc. On the basis of this ontology, the PSE provides a dictionary (knowledge-base) of data and tools locations allowing the users to compose their applications as workflows by making use of all the necessary resources without worrying about their underlying distributed and heterogeneous nature. The modeled applications are automatically translated into grid execution scripts corresponding to GRSL (Globus Resource Specification Langue) and are then submitted for execution on grid through GRAM (Globus Resource Allocation Manager). The performance of this PSE was checked with a simple application that used TribeMCL (http://www.ebi.ac.uk/research/cgg/tribe), for clustering human protein sequences, which were extracted from SwissProt database by using *seqret* program of the EMBOSS suite and compared *all against all* for similarity though BLAST program. In order to take advantage of the grid resources and enhance the performance of similarity search process, the output of *seqret* program was spilt into three separate files in order to run three instances of BLAST in parallel. The individual BLAST outputs were concatenated and transformed into a Markov Matrix required as input for TribeMCL. Finally, the PSE displayed the results of clustering in an opportune visualization format. It was observed that total clustering process on grid took 11h50'53'' as compared to 26h48'26'' on standalone machine. It was also noted on the basis of another experimental case (taking just 30 protein sequences for clustering) that the data extraction and result visualization steps in the clustering process are nearly independent of the number of protein sequences (i.e. approximately same time was observed in the case of all protein *vs* 30 protein sequences). Furthermore, BLAST computation takes more time as compared to TribeMCL (for *all against all* case BLAST took 8h50'13'' while TribeMCL took 2h50'28'') [17]. Another PSE for bioinformatics has been proposed in [85]. It uses Condor/G for the implementation of PSE that provides an integrated environment for developing component based workflows through commonly used bioinformatics applications and tools such as *Grid-BLAST, Grid-FASTA, Grid-SWSearch, Grid-SWAlign and Ortholog-Picker* etc. Condor/G is an extension to grid *via* Globus and it combines the inter-domain resource management protocols of Globus Toolkit with intra-domain resource management methods of Condor to provide computation management for multi-institutional grid. The choice of Condor/G is justified on the basis of its low implementation overhead as compared to other grid technologies. The implementation of a workflow based PSE is made simple by the special functionality of Condor meta-scheduler DAGMan (Directed Acyclic Graph Manager), which supports the cascaded execution of programs in a grid environment. The developed prototype model was tested by integrated (cascaded) execution of the above mentioned sequence search and alignment tools in grid environment. In order to enhance the efficiency, the sequence databases and queries were splited into as much parts as the number of available nodes, where the independent tasks were executed in parallel.

2.2.5. Grid-Based Workflow Management Systems

As discussed in the context of PSE, a workflow is a process of composing an application by specifying the tasks and their order of execution. A grid-based workflow management system provides all the necessary services for the creation, execution and visualization of the status and results of the workflow in a seamless manner. These features make workflows ideal for the design and implementation of life science applications that consist of multiple steps and require the integrated access and execution of various data and application resources. Therefore one can find various domain specific efforts for the development of proper workflow management systems for life sciences (Table **3**).

There have been several important demonstrations of different types of life science applications on grid-based workflow management systems. For example, the design and execution of a tissue-specific gene expression analysis experiment for human have been demonstrated in a grid-based workflow environment called '*WildFire'* [86]. The workflow takes as an input 24 compressed *GeneBank* files corresponding to 24 human chromosomes and after decompression, it performs exon extraction (through *exonx* program) from each file in parallel resulting in 24 FASTA files. In order to further increase the level of *granularity*, each FASTA file is split into five sub-files (through *dice* script developed in Perl), making a total of 120 small files ready for parallel processing with BLAST. Each file was matched against a database of transcripts ('*16, 385 transcripts obtained from Mammalian Gene Collection'*). The execution of this experiment on a cluster of 128 Pentium III nodes took about 1 hour and 40 minutes, which is reported to be 9 times less than the time required for the execution of its sequential version. The iteration and dynamic capabilities of *WildFire* have also been demonstrated through the implementation of a swarm algorithm for parameter estimation problem related to biochemical pathway model based on 36 unknowns and 8 differential equations. Similarly, the effectiveness of Taverna

Table 3. Grid-Based Workflow Management Systems

Workflow Management System Ref.	Supported Grid Middleware Technologies and Platforms	Main Features
Wildfire [86] http://wildfire.bii.a-star.edu.sg/	◆ Condor/G, SGE, PBS, LSF ◆ Workflows are mapped into Grid Execution Language (GEL) script ◆ Platform: Windows and Linux	◆ GUI-based drag-and-drop environment for workflow construction through EMBOSS Suite of tools ◆ Supports complex operations such as iteration and dynamic parallelism ◆ Open source and extensible
Taverna [87] http://taverna.sourceforge.net/	◆ myGrid middleware ◆ SCULF language for workflows ◆ Workflows are mapped into web services ◆ Platform: cross platform	◆ GUI-based workbench for creating In-silico experiments using EMBOSS suite of tools, NCBI, EBI, DDBj, SoapLab, BioMoby and other web services. ◆ Currently does not support complex operations. ◆ Open source and extensible
ProGenGrid [93] http://www.cact.unile.it/projects/	◆ Globus Toolkit4.1 ◆ GridFTP and DIME for data transfer. ◆ iGrid information service for resource and web service discovery. ◆ Java Axis and gSOAP Toolkit	◆ UML-based editor for workflow construction, execution and monitoring ◆ RASMOL for visualization ◆ AutoDoc for drug design

[87] workflow has been demonstrated by construction of a workflow that provides genetic analysis of the *'Graves'* disease. The demonstrated workflow makes use of Sequence Retrieval System (SRS), mapping database service and other programs deployed as SoapLab services to obtain information about candidate genes, which have been identified through Affymetrix U95 microarray chips as being involved in Graves' disease. The main functionality of the workflow was to map a candidate gene to an appropriate identifier corresponding to biological databases such as Swiss-Prot and EMBL in order to retrieve the sequence and published literature information about that gene through SRS and MedLine services. The result of *tBLAST* search against the PDB provided identification of some related genes, whereas the information about the molecular weight and isoelectric point of the candidate gene was provided by the *Pepstat* program of EMBOSS suite. Similarly, Taverna has also been demonstrated with the successful execution of some other workflows for a diversity of *in silico* experiments such as pathway map retrieval and tracking of data provenance.

2.2.5. Grid-Based Frameworks

In software development, a framework specifies the required structure of the environment needed for the development, deployment, execution and organization of a software application/project related to a particular domain in an easy, efficient, standardized, collaborative, future-proof and seamless manner. When becoming fully successful and widely accepted and used, most of these frameworks are also made available as Toolkits. There are some general frameworks for grid-based application development such as Grid Application Development Software (GrADS) [88], Cactus [89] and IBM Grid Application Development Framework for Java (GAF4J) (http://www.alphaworks.ibm.com/tech/GAF4J). Similarly, some specific grid-based frameworks for life sciences have also been proposed and demonstrated such as *Grid Enabled Bioinformatics Application Framework* (GEBAF) [90], that proposes an integrated environment for grid-enabled bioinformatics application using a set of open source tools. The open source tools used in GEBAF include Bioperl Toolkit [91], Globus Toolkit [37], Nimrod/G [92] and Citrina (database management tool (http://www.gmod.org/citrina)). The framework provides a portal based interface that allows the user to submit a query of any number of sequences to be processed with BLAST against publicly available sequence databases. The user options are stored in a hash data structure by creating a new directory for each experiment and a script using the BioPerl::SeqIO module, dividing the user query into sub-queries each consisting of just a single sequence. The distributed query is then submitted through Nimrod/G plan file for parallel execution on the grid. Each grid node maintains an updated and formatted version of the sequence database through Citrina. The individual output of each sequence query is parsed and concatenated by another script that generates the summary of the experiment in the form of an XML and Comma Separated Value (CSV) files containing the number of most significant hits from each query. The contents of these files are then displayed through the result interface. A particular demonstration for BLAST was carried out with 55,000 sequences against *SwissProt* database. With 55,000 parallel jobs, the grid has been fully exploited within the limits of its free nodes and it has been observed that the job management overhead was low as compared to the actual search time for *BLASTing* of each sequence. Although summarizing thousands of results is somewhat slow and nontrivial, its execution time remains insignificant when compared with the experimenting time itself. The developed scripts were also tested for reusability with other similar applications such as ClustalW and HMMER, with little modification. A web service interface has been proposed for future development of *GEBAF* in order to make use of other bioinformatics services such as Ensemble. Similarly, for better data management, Storage Resource Broker (SRB) middleware is also proposed as an addition for the future.

GEBAF is not the only grid-enabling framework available. For example, Asim *et al.* [94] described the benefits of using the Grid Architecture Development Software (GrADS)

framework [88] for the gridification of bioinformatics applications such as FASTA [77]. Though there already existed an MPI-based master-slave version of the FASTA but it used a different approach: it kept the reference database at the master side and made the master responsible for equal distribution of database to slaves and the subsequent collection and concatenation of the results. In contrast to that, GrADS based implementation makes reference databases (as a whole or as a portion) available at some or all of the worker nodes through database replication. Thus, the master at first sends a message to workers for loading their databases into memory and then it distributes the search query and collects the results back. This type of *data-locality* approach eliminates the communication overhead associated with the distribution of large scale databases. Furthermore, through the integration of various software development and grid middleware technologies (such as *Configurable Object Program*, *Globus Monitoring and Discovery Service (MDS) and Network Weather Service (NWS))*, GrADS framework provides all the necessary user, application and middleware services for the composition, compilation, scheduling, execution and real time monitoring of the applications on a grid infrastructure in a seamless manner.

2.2.6. BioGrid Data Management Approaches

It is a well recognized fact that most of the publicly available biological data originates from different sources of information, i.e. it is *heterogeneous* and is acquired, stored and accessed in different ways at different locations around the world, i.e. it is *distributed* [95]. The heterogeneity of data may be *syntactic* i.e. difference in file formats, query languages and access protocols etc, *semantic* i.e. genomic and proteomic data etc, or *schematic* i.e. difference in the names of database tables and fields etc. In order for this heterogeneous and distributed data to be accessed in a uniform and federated environment, one has to make use of appropriate web and grid technologies to form an intermediatary bridge (Fig. **4**). The gridification of biological databases and applications is also motivated by the fact that their number, size and diversity are growing rapidly and continuously. This makes it impossible for an individual biologist to store a local copy of any major databases and execute either data or computer-intensive application in a local environment.

This inability of locality also demands for the grid-enablement of the resources. However, an important factor that hinders the deployment of existing biological applications, analytical tools and databases on grid-based environments is their inherent pre-grid design (legacy interface). This is so because the design suits the requirements of a local workstation environment in terms of input/output. The gridification of such applications requires a transparent mechanism to connect local input/output with a grid-based distributed input/output through some intermediatary tools such as grid middleware specific Data Management Services (DMS) and distributed storage environments. One such example of biological data management in grid environment has been discussed in [96]. It provides a transparent interface for legacy bioinformatics applications, tools and databases to be connected to computational grid infrastructures such as EGEE, without incurring any change in the code of these applications. Authors have reported the use of modified Parrot [97] as a tool to connect a legacy bioinformatics application to the EGEE database management system. The EGEE database management system enables location and replication of databases needed for the management of very large distributed data repositories. With Parrot-based connection, the user is freed from the overhead of performing file staging and specifying in advance an application's data need. Rather, an automated agent launched by the Parrot takes care of replicating the required data from the remote site and supplying it to the legacy application as it would have been accessing data with local input/output capabilities. The agent resolves logical file name to the storage file name, selects the best location for replication and launching the program for execution on the downloaded data. For the purpose of demonstration, authors have reported the deployment (virtualization) of some biological databases such as.

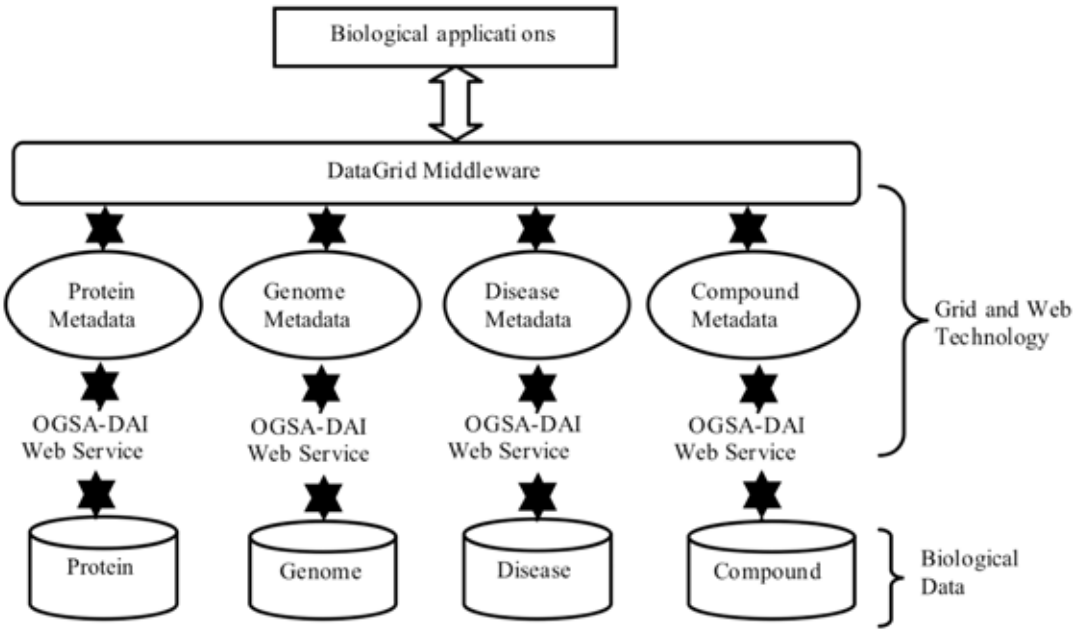


Fig. (4). Major architectural components of a biological DataGrid environment (reproduced from [6] and annotated).

Swiss-Prot and TrEMBL [98] by registering these databases with the replica management service (RMS) of EGEE. Similarly, programs for protein sequence comparison such as BLAST [71], FASTA [99], ClustalW [72] and SSearch [100] have been deployed by registering them with the experiment software management service (ESM). The deployed programs were run on a grid environment and their access to the registered databases was evaluated by two methods: replication (by copying the required database directly to the local disk) and remote input/output (attaches the local input/output stream of the program to the copied data in cache or on-the-fly mode). The evaluation of these methods showed that both methods perform similarly in terms of efficiency e.g. on a database of about 500,000 protein sequences (205 MB), each method takes about 60 seconds for downloading from any grid node and about four times less than this time in case the data node is near the worker node. It is important to note, however, that the replication method creates an overhead in terms of free storage capacity on the worker node. This problem may be particularly if the size of the database to be replicated is too high or if the worker node has many CPUs sharing the same storage and each accessing a different set of databases. This is the reason why remote input/output method overweighs the replication.

However, the real selection of any method depends on the nature of program (algorithm). For compute-intensive programs such as SSearch, remote input /output is always better (as it works on copying progressive file blocks) whereas for data-intensive programs such as BLAST and FASTA, the replication method may work better [96].

Similarly, there are some other DataGrid projects which provide an integrative use of these highly heterogeneous and distributed data sources in an easy and efficient way. A biologist could take advantage of some specific Data Grid infrastructure and middleware services such as BRIDGES (http://www.brc.dcs.gla.ac.uk/projects/bridges), BIRN (http://www.nbirn.net) and various other European Union Data Grid projects e.g. EU-DataGrid (http://www.edg.org/) and EU-DataGrid for Italy (http://web. datagrid.cnr.it/Tutorial_Rome) etc. These Data Grid projects make use of standard middleware technologies such as Storage Resource Broker (SRB), OGSA-DAI and IBM Discovery Link. Some of the important features of these DataGrid middleware technologies are listed in Table **4**.

2.2.8. Computing and Service Grid Middleware

In the same way that a computer operating system provides a user-friendly interface between user and computer hardware, Grid middleware provides important services needed for easy, convenient and proper operation and functionality of grid infrastructure. These services include access, authentication, information, security and monitoring services as well as data and resource description, discovery and management services. In order to further reduce the difficulties involved in the process of installation, configuration and setting-up of grid middleware, there have also been proposals for the development of Grid Virtual Machines (GVM) and Grid Operating Systems [101-105]. The development of specific grid operating systems or even embedding grid middleware as a part of existing operating systems would greatly boost-up the use of grid computing in all computer related domains, but this is yet to be seen in the future. The important features of currently used computing and service grid middleware are listed in Table **5**.

2.2.9. Local Resource Management System (LRMS)

The grid middleware interacts with different clusters of computers through Local Resource Management System (LRMS) also known as Job Management System (JMS). The LRMS (such as Sun Grid Engine, Condor/G and Nimrod/G) is responsible for submission, scheduling and monitoring of jobs in a local area network environment and providing the results and status information to the grid middleware through appropriate wrapper interfaces. Some of the important features [106] of commonly used LRMS software are listed in Table **6**.

2.2.10. BioGird Infrastructure

Mostly, BioGrid infrastructure is based on the simple idea of cluster computing and is leading towards the creation of a globally networked massively parallel supercomputing infrastructure that connects not only the computing units along with their potential hardware, software and data re-

Table 4. DataGrid Middleware Technologies

DataGrid Technology	Main Features and Services
SRB (Storage Resource Broker) [4] http://www.sdsc.edu/srb/	◆ Hierarchical logical name space for managing distributed heterogeneous databases ◆ Data transfer, replication, publication, sharing and preservation services. ◆ Callable library functions for applications programs ◆ Runs on Windows, Linux and Unix platforms and is freely available for academic use
OGSA-DAI (Open Grid Services-Data Access and Integrations) http://www.ogsadai.org.uk/	◆ Web service based data access and integration ◆ Supports two main web service specifications namely web services interoperability (WS-I) and web services resource framework (WSRF). ◆ Runs on all platforms and is freely available as open source.
IBM DiscoveryLink [5] http://www-306.ibm.com/software/data/integration/	◆ Suite of wrappers for relational and non-relational databases ◆ Federated view of distributed heterogeneous resources ◆ Runs on Windows, Linux, Unix, and IBM Aix platforms and is free for authorized academic use

Table 5. Computing and Service Grid Middleware

Grid Middleware Goal and Developer	Brief Description of Architecture and Services
Globus Toolkit GT4 [37] Goal: To provide a suit of services for job, data, and resource management. Developer: Argonne National Laboratory, University of Chicago and other partners http://www.globus.org/	◆ OGSA-WSRF based architecture ◆ Credential management services (MyProxy, Delegation, SimpleCA) ◆ Data management services (GridFTP, RFT, OGSA-DAI, RLS, DRS) ◆ Resource management services (RSL and GRAM) ◆ Information and monitoring services (Index, Trigger and WebMDS) ◆ Instrument management services (GTCP) ◆ Platform: Linux
LCG-2/ gLite [39] Goal: Large scale data handling and compute power infrastructure Developer: EGEE project in collaboration with VTD, US and other partners. http://glite.web.cern.ch/glite/	◆ LCG-2: a pre-web service middleware based on Globus 2 ◆ gLite: an advanced version of LCG-2 based on web services architecture ◆ Authentication and security services (GSI, X.509, SSL, CA) ◆ Information and monitoring services (Globus-MDS, R-GMA, GIIS, BDII) ◆ Resource management services (GUID, SURL) ◆ Data management services (WMS, SLI) ◆ Platform: Linux and Windows
UNICORE [108] Goal: Light weight grid middleware UNICORE : Fujitsu Lab EU, UniGrid: EU Funded Project http://www.unicore.eu/	◆ UNICORE: a pre-web service grid middleware based on OGSA standard ◆ UNICORE 6: a web service-based and OGSA compatible advanced version of UNICORE ◆ Security services (X5.09, CA, SSL/TLS) ◆ Execution management engine (XNJS) ◆ Platform: Unix/Linux platform
myGrid [10, 11] Goal: Service-oriented in-silico environment for life sciences. ServiceGrid @ University of Manchester, UK http://www.mygrid.org.uk/	◆ OGSA and web service-based architecture ◆ Semantic web-based annotation, ontologies and discovery management ◆ Talisman: user interface and workbench for in-silico experiment design ◆ Platform: Linux, Windows, and Mac
ABCGrid [107] Goal: Easily installable and simple grid setup for bioinformatics Center for Bioinformatics, Peking University http://abcgrid.cbi.pku.edu.cn/	◆ Java based client server implementation with a package of three independent and easy to install programs; ABCUser, ABCMaster and ABCWorker ◆ Platform: Windows, Linux/Unix, Mac OS X platform

sources, but also expensive laboratory and industrial equipment, and ubiquitous sensor device in order to provide unlimited computing power and experimental, setup required for modern day biological experiments. Moreover, this infrastructure is also being customized in a way that it becomes easily accessible by all means of an ordinary general purpose desktop/laptop machine or any type of handheld devices. Some of the major components of a generic BioGrid infrastructure has been illustrated in Fig. **3**. The overall architectural components are organized at three major levels (layers) of services. The focus of application layer services is to provide user-friendly interfaces to a biologist for carrying out the desired grid-based tasks with minimum steps of usability and interaction (enhanced automation and intelligence). Similarly, the focus of grid middleware services is to provide seamless access and usability of distributed and heterogeneous physical layer resources to the application layer services. In the following sections we discuss various contributions related to the development and use of some of these services at both application and middleware level.

The design and implementation of a typical BioGrid infrastructure vary mainly in terms of the availability of resources and demands of the biological applications that are supposed to use that particular grid. There are many infrastructures starting from an institutional/organizational grids consisting of simple PC based clusters or combination of clusters [109-112] to national and international BioGrid projects with different architectural models and for appropriate problem handling. The models include Computing Grid architecture (providing basic services for task scheduling, resource discovery, allocation and management etc), Data Grid architecture (providing services for locating, accessing, integrating and management of data), Service Grid architecture (services for advertising, registering and invocation of resources) and Knowledge Grid architecture (services for sharing collaborative scientific published or unpublished data). The infrastructure details of some major BioGrid projects are presented in Table **7**. It may be observed that the same infrastructure may be used to serve more than one application models based on the availability of some additional service and resources.

3. SOME FLAGSHIP BIOGRID PROJECTS

We present here some selected flagship case studies which have elicited a positive public response from bioscientists for their special role and contribution to the life science domain. The description of most important imple-

Table 6. Job Management Systems (Local Resource Manager)

Local Resource Manager	General features Platform, GUI and APIs	Job support Job description, type and MPI support
Sun Grid Engine 6 Open Source developed by Sun Microsystems http://gridengine.sunsource.net	◆ Platform: Solaris, Apple Macintosh, Linux and Windows platform ◆ User friendly GUI and portal ◆ DRMAA API ◆ Integration with globus through GE-GT Adopter	◆ Shell scripts for job description ◆ Supports standard and complex job types with arrays and workflows ◆ Provides modules for integration with MPI ◆ 5 Million jobs on more than 10,000 hosts
Condor –G (version 6.8.2) Open source developed by University of Wisconsin http://www.cs.wisc.edu/condor	◆ Solaris, Apple Macintosh, Linux, Windows platform ◆ Web-service interface ◆ DRMAA API ◆ Condor-G is Globus enabled and allows jobs to be sent to other resource managers	◆ Classified Advertisements (*classads*) for job description ◆ Supports standard and complex jobs with arrays and workflows ◆ Provides modules for integration with MPI
Nimrod-G 3.0.1 [92] Open source research prototype developed by Monash University other flavors: Nimrod/O, Netsolve, Active Sheets http://www.csse.monash.edu.au/~davida/nimrod/nimrodg.htm	◆ Platform: Linux, Solaris, Mac with x86 and sparc architecture ◆ Provides web portal and API ◆ Supports integration with Globus, Legion, Condor, NetSolve and others	◆ Nimrod Agent Language for job description ◆ Uses GRAM interfaces to dispatch jobs to computers

mentation strategies along with some main services is provided with the help of appropriate illustrations.

3.1. EGEE Project

The *Enabling Grids for E-science in Europe* (EGEE) project evolved as a testbed from its precedor *European Data Grid* (EDG) project and completed its first phase during 2004-2006 by providing support to only three scientific applications. The second phase of this project (2006-2008) has evolved from the testbed to a pilot production project that runs more than 20 applications and projects related to different domains of science and industry ranging from high-energy physics to life science and nanotechnology [96]. For example, BioInfoGrid (http://www.bioinfogrid.eu/), WISDOM (http://wisdom.eu-egee.fr) and EMBRACE (http://www.embracegrid.info) are some of the major biomedical applications that are continuously making use of the EGEE infrastructure. It is during the 2nd phase that the project has

Table 7. BioGrid Infrastructure Projects

BioGrid Project	Grid Technologies and Infrastructure	Main Applications
Asia Pacific BioGrid http://www.apgrid.org	◆ Middleware: Globus1.1.4 ◆ Job managers: Nimrod/G, LSF. SGE ◆ Size: 5 nodes, 25+ CPUs, 5 sites	◆ FASTA, BLAST, SSEARCH, MFOLD ◆ Virtual Lab DOCK, EMBASSY, PHYLIB ◆ EMBOSS suite of applications
Open BioGrid Japan OBIGRID Japan [6] http://www.obigrid.org	◆ Middleware: Globus 3.2. ◆ Ipv6 for secure communication ◆ VPN over internet for connecting multiple sites ◆ Size: 363 nodes, 619 CPUs, 27 sites	◆ Workflow based distributed bioinformatics environment ◆ BLAST search service ◆ Genome annotation system ◆ Biochemical network simulator
Swiss BioGrid [7] http://www.swissbiogrid.org	◆ Middleware: NorduGrid's ARC and GridMP ◆ Infrastructure consists of heterogeneous hardware platforms including both clusters and Desktop-PC grids	◆ High throughput compound docking into protein structure binding sites ◆ High throughput analysis of proteomics data.
Enabling Grids for E-sciencE (EGEE) [113] http://www.eu-egee.org	◆ Middleware: gLite ◆ Size: More than 30,000 CPUs and 20 Petabytes storage. ◆ Maintains 20,000 concurrent jobs on average. ◆ Connects more than 90 institutions in 32 countries world wide	◆ WISDOM: drug discovery application ◆ GATE: radio therapy planning and medical tomography application ◆ SiMRI3D: parallel MRI simulator ◆ GPS@: Grid Protein Sequence @Analysis and other applications
North Carolina BioGrid http://www.ncbiotech.org	◆ Avaki data grid middleware with Virtual File System across grid nodes. ◆ Heterogeneous standalone and clustered processors with a variety of operating systems and cluster management systems	◆ Bioinformatics datasets and applications installed on native file system and shared across the grid. ◆ Unified view of data and computers by making them appear local

been renamed as *Enabling Grid for E-scienceE* in order to enhance its scope from European to international level [113]. At this time, the overall EGEE infrastructure consists of more than 30,000 CPUS with 20 petabytes of storage capacity provided by various academic institutes and other organizations and industries around the world in the form of high-speed and high-throughput compute clusters, which are being updated and interoperated through its web-service based light-weight, more dynamic and inter-disciplinary grid middleware named *gLite [39]*. Like EGEE, gLite middleware also builds on a combination of various other projects including *LCG-2* (http://cern.ch/LCG), *DataGrid* (http://www.edg.org), *DataTag* (http://cern.ch/datatag), *Globus Alliance* (http://www.globus.org), *GriPhyN* (http://www.griphyn.org) and *iVDGL* (http://www.ivdgl.org). Technical description of some of the gLite services is presented in the following sections:

3.1.1. Authentication and Security Services

EGEE uses Grid Security Infrastructure (GSI) for authentication (through digital X.509 certificate) and secure communication (through SSL: Secure Socket Layer protocol with enhancements for single sign-on and delegation). Therefore, in order to use the EGEE grid infrastructure resources, the user has to register first and get a digital certificate from appropriate Certificate Authority (CA). When the user signs in with the original digital certificate which is protected with a private key and a password, the system then creates another passwordless temporary certificate called *proxy certificate* that is then associated with every user request and activity.

3.1.2. Information and Monitoring Services

The gLite 3 uses Globus MDS (Monitoring and Discovery Service) for resource discovery and status information. Additionally, it uses Relational Grid Monitoring Architecture (R-GMA) for accounting and monitoring. In order to provide more stable information services, the gLite Grid Information Indexing Server (GIIS) uses BDII (Berkeley Database Information Index Server) that stores data in a more stable manner than original Globus based GIIS.

3.1.3. Data Management Services

As in traditional computing, the primary unit of data management in EGEE grid is also the file [39]. gLite provides a location independent way of accessing files on EGEE grid through the use of Unix based hierarchical logical file naming mechanism. When a file is registered for the first time on the grid, it is assigned a GUID (Grid Unique Identifier that is created from User Unique Identifier; MAC address and a time stamp) and it is bound with an actual physical location represented by SURL (Storage URL). Once a file is registered on the EGEE grid, it cannot be modified or updated because the data management system creates several replicas of the file in order to enhance the efficiency of subsequent data access. Thus, updating any single file would create the problem of data inconsistency which has not as yet been solved in EGEE.

3.1.4. Workload Management System (Resource Broker)

The new gLite based work load management system (WMS or resource broker) is capable of receiving even multiple inter-dependent jobs described by Job Description Language (JDL) and it dispatches these jobs to most appropriate grid sites (selection of appropriate grid site is based on the dynamic process of *match-making*) and then keeps track of the status of the jobs and retrieves the results back when jobs are finished. While dispatching the jobs, the resource broker uses Data Location Interface (DLI) service to supply input files along with job to the worker node.

3.2. Organic Grid: Self Organizing Computational Biology on Desktop Grid

The idea of Organic Grid [114] is based on the decentralized functionality and behavior of self organizing, autonomous and adaptive organisms (entities) in natural complex systems. The examples of natural complex systems include functioning of biological systems and behavior of social insects such as ant and bees. The idea of the organic grid leads towards a novel grid infrastructure that could eliminate the limitations of traditional grid computing. The main limitation of traditional grid computing lies in their centralized approach. For example, a Globus based computational grid may use a centralized meta-scheduler and thus it would be limited to smaller number of machines only. Similarly, Desktops Grid Computing based on distributed computing infrastructure such as BOINC may use centralized master/slave approach and thus would be only suitable for coarse-grained independent jobs only. The idea of Organic Grid is to provide a 'ubiquitous' type peer-to-peer grid computing model capable of executing arbitrary computing tasks on a very large number of machines over network of any quality, by redesigning the existing desktop computing model in a way that it supports distributed adaptive scheduling through the use of mobile agents. In essence, it means that a user application submitted on such type of architecture would be encapsulated in some type of a mobile agent containing the application code along with the scheduling code. After encapsulation, the mobile agent can decide itself (based on its scheduling code and the network information) to move to any machine that has appropriate resources needed for the proper execution of the application. This type of mechanism provides the same type of user-level abstractness as provided by traditional Globus-based grid but additionally it builds on decentralized scheduling approach that enables the grid to span to very large number of machines in a more dynamic peer-to-peer computing model. The use of mobile agents (which are based on RMI mechanism that is built on top of client/server architecture) as compared to their alternate service based architecture, provides higher level of ease and abstractedness in terms of validation (experimentation) of different types of scheduling, monitoring and migration schemes. Although the project uses a scheduling scheme that builds on tree-structured overlay network, it is made adaptive based on some value of application specific performance metric. For example, the performance metric for a data-intensive application such as BLAST would give high consideration to bandwidth capacity of the communication link before actually scheduling the job on a particular node. Similarly, it will select a high-speed node for another application that comes under the class of compute-intensive applications. Furthermore, in order to provide uninterrupted execution with dynamic and transparent migration features, the project makes use of strongly mobile agents instead of

traditional weakly mobile agents (Java based mobile agents that cannot access their state information). Following the common practice in grid computing research, the proof-of-concept has been demonstrated with the execution of NCBI BLAST (that falls in the class of independent task application) on a cluster of 18 machines with heterogeneous platform and ranked under the categories of fast, medium and slow through the introduction of appropriate delays in the application code. The overall task required the comparison of a 256 KB sequence against a set of 320 data chunks each of size 512 KB. This gave rise to 320 subtasks, each responsible for matching the candidate 256 KB sequence against one specific 512 KB data chunk. The project successfully carried out the execution of these tasks and it has been observed that by adopting the scheduling according to the dynamics of the architecture greatly improves performance and quality of results. The project is being further extended to provide the support for different categories of applications and enabling the user to configure different scheduling schemes for different applications through some easy to use APIs.

3.3. Advancing Clinico-Genomic Trials on Cancer (ACGT)

ACGT is a Europe wide integrated biomedical grid for post-genomic research on cancer (http://www. eu-acgt.org) [115]. It intends to build on the results of other biomedical grid projects such as caBIG, BIRN, MEDIGRID and My-Grid. The project is based on open source and open access architecture and provides basic tools and services required for medical knowledge discovery, analysis and visualization. The overall grid infrastructure and services are aimed to provide an environment that could help scientists to:

a. Reveal the effect of genetic variations on oncogenesis

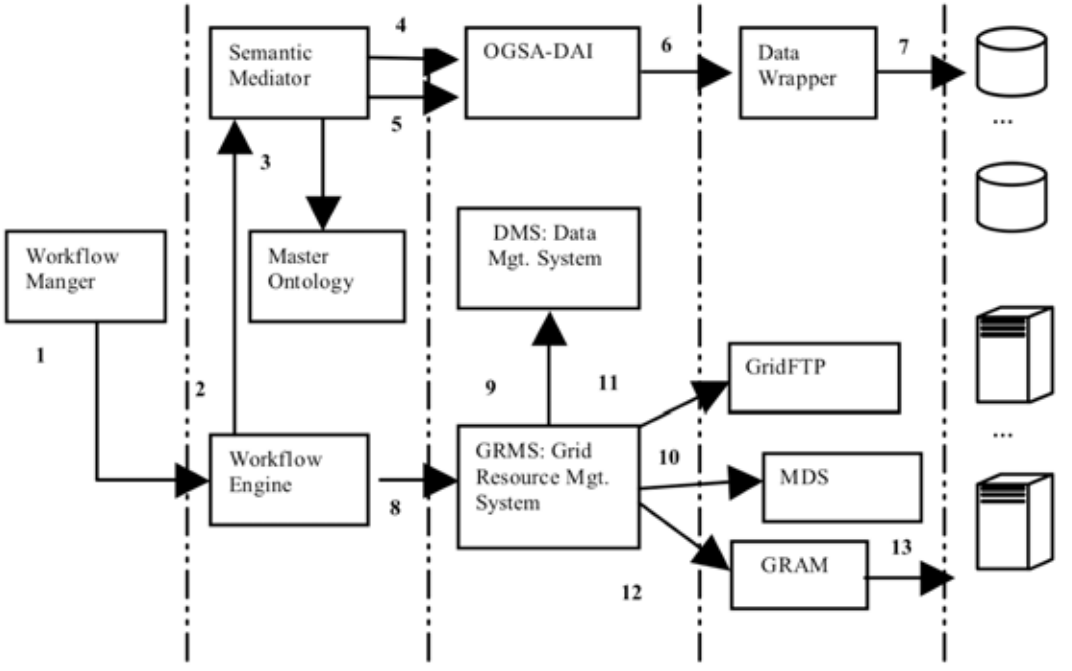
b. Promote the molecular classification of cancer and development of individual therapies

c. Modeling of *in silico* tumor growth and therapy response

In order to create the required environment that supports the implementation of these objectives, ACGT focuses on the development of a virtual web that interconnects various cancer related centres, organizations and individual investigators across the Europe through appropriate web and grid technologies. Mainly, it uses semantic web and ontologies for data integration and knowledge discovery and Globus toolkit with its WS-GRAM, MDS and GSI services for cross organization resource sharing, job execution, monitoring and result visualization. Additionally, ACGT also uses some higher level grid services from Gridge framework developed at Poznan Supercomputing and Networking Centre (PSNC) [116]. These services include GRMS (Grid Resource Management System), GAS (Grid Authorization System) and DMS (Data Management System). These additional services provide the required level of dynamic and policy-based resource management; efficient and reliable data handling; and monitoring and visualization of results. Fig. **5** provides a usage scenario of these services in the context of ACGT environment.

3.4. KidneyGrid

KindneyGrid project [8] uses web service-based technologies to develop a collaborative e-Research platform in the form of a virtual organization that interconnects and provides interaction among different entities (e.g. kidney model



1. Solve Problem; **2**. Prepare Data; **3**. Get Ontology Context; **4**. Get Molecular Data; **5**. Get Clinical data; **6-7**. Query Database; **8**. Submit Computation; **9**. Get Data Location; **10**. Find Available Resources; **11**. Transfer Data; **12**. Submit Job to Resource; **13**. Run Job [116]

Fig. (5). ACGT integrated environment usage scenario (reproduced from [116]).

developers, resources and users) related to the study of human and animal kidneys. Fig **6** shows the basic architecture of the system along with the list of typical interactions among its components. The interaction among heterogeneous legacy components and the overall functionality of the project has been achieved by using following web and grid technologies [8]:

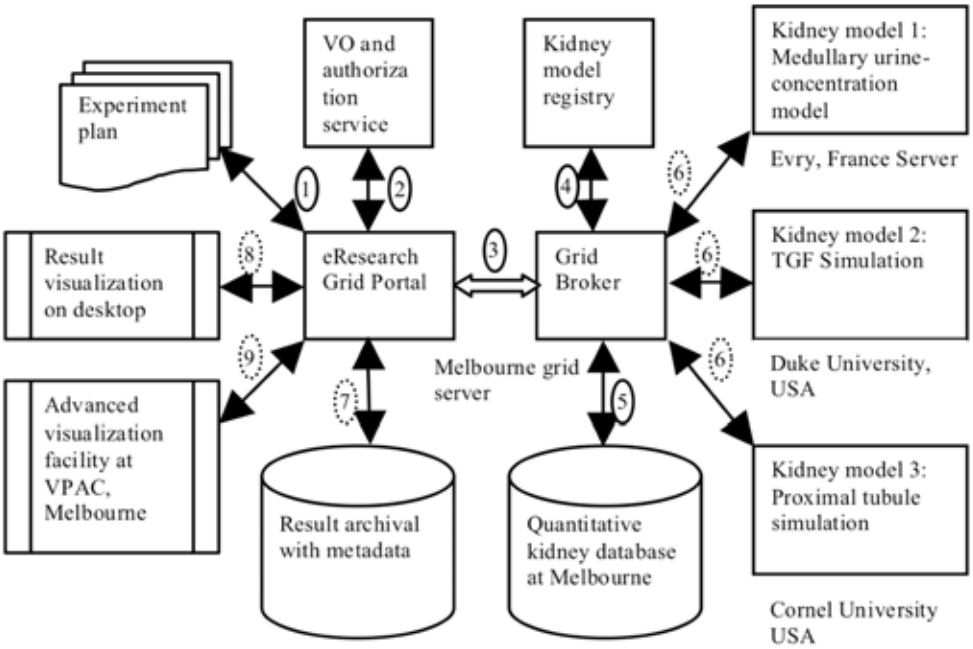
- Gridsphere Framework for portal development
- MyProxy for virtual organization
- Gridbus as application level meta-scheduler (resource broker)
- WSRF for the development of wrappers that provide interaction among legacy resources and application and
- GT4, Unicore, Alchemi as core grid middleware at different sites

The project has been tested and evaluated by the implementation of a medullary kidney model. It allows the user to compose, run and monitor an *in silico* experiment by selecting appropriate kidney model and required parameters through a web-based user interface. The results of the experiment are also displayed with the help of suitable visualization tools.

3.5. Virtual Laboratory

Virtual laboratory project [9] develops an integrated e-Research environment that could provide the computational power and fast access to databases required for molecular modeling-based drug design. The process of molecular modeling requires each target protein to be screened out against all the molecules stored in a particular chemical database (CDB). Given the huge size of CDBs (millions of compounds in a single database), it requires decades of years on a single computer to complete a single docking experiment. This time could be reduced to a single day or even less than that, with a grid based architecture. As a testbed, the Virtual Laboratory project interconnects geographically distributed resources at three main sites i.e. Monash University, Melbourne, Australia; AIST, Tokyo, Japan; and Argonne National Lab., Chicago, USA. It uses Nimrod parameter modeling tools to adopt DOCK molecular modeling software to run as a parametric sweep application in the grid environment. DOCK jobs are scheduled on the grid by using Nimrod/G resource broker. Each resource on the grid is accessed *via* Globus where as GRACE software toolkit has been used for resource trading. Similarly, some intelligent tools were used for chemical database management.

The virtual laboratory architecture has been tested with the molecular docking of some 200 molecules from *aldrich 300* CDB using the 3D structure of *endothelin converting enzyme* (ECE) as a target receptor that is involved in hypotension. With the value of range parameter 'ligand-number' from 1 to 200 and step size of 1, there were 200 docking jobs. The common input files and executables required for these jobs were pre-staged on all the grid resources at different sites using *globus-rcp* command. These jobs were successfully scheduled and executed on the avail-



1. User request; *2.* Authorization; *3.* Portal asks for resources; *4.* Broker verifies with registry; *5.* Broker provides connectivity to resources if they are available in the local mirror; *6.* Broker provides connectivity to remote resources; *7.* User and experiment state information; *8.* Results on desk top; *9.* External visualizatio n resources [8]

Fig. (6). Architecture of the KidneyGrid project: steps 1-9 illustrate a typical user/component interaction and activities in the system (Reproduced from [8]).

able resources using deadline and budget constraints optimization schemes for meeting the user requirements regarding the completion time and affordable amount of money (see [117,118]). As a result of these experiments, it has been proven that economy driven and service-oriented architectures provide the best utilization of grid resources.

4. CONCLUSIONS AND FUTURE RECOMMENDATIONS

This paper presents a scrupulous analysis of the state-of-the-art in web and grid technology for bioinformatics, computational biology and systems biology in a manner that provides a clear picture of currently available technological solutions for a very wide range of problems. While surveying the literature, it has been observed that there are many grid-based PSEs, Workflow Management Systems, Portals and Toolkits under the name of Bioinformatics but not as many for Systems or Computational Biology. However, in each case a mix of projects and applications has been found overlapping from bioinformatics to computational and systems biology. Thus, the paper provides a synthetic overview of several major contributions under each technological category in a way that could help and serve a wide range of individuals and organizations interested in:

- Setting up a local, enterprise or global IT infrastructure for life sciences.
- Solving a particular life science related problem by selecting the most appropriate technological options that have been successfully demonstrated and reported herein.
- Migrating already existing bioinformatics legacy applications, tools and services to a grid-enabled environment in a way that requires less effort and is motivated with previous related studies provided herein.
- Comparing a newly developed application/ service features with those currently available.
- Starting a research and development career related to the use of web and grid technology for biosciences.
- Based on the analyses of the state-of-the-art, we identify below some key open problems:
- The use of semantic web technologies such as domain ontologies for life sciences is still not at its full level of maturity, perhaps because of semi-structured nature of XML and limited expressiveness of ontology languages [28].
- Biological data analysis and management are still quite difficult jobs because of the lack of development and adaptation of optimized and unified data models and query engines.
- Some of the existing bioinformatics ontologies and workflow management systems are simply in the form of Directed Acyclic Graphs (DAGs) and their descriptions are lacking expressiveness in terms of formal logic [86].
- Lack of open-source standards and tools required for the development of thesaurus and meta-thesaurus services [22].
- Need for appropriate query, visualization and authorization mechanism for the management of provenance data and meta-data in *in silico* experiments [10, 86].
- Some of the BioGrid projects seem to be discontinued in terms of information updating. This might arise from funding problems or difficulties associated with their implementation.
- There is a lack of domain specific mature application programming models, toolkits and APIs for grid-enabled application development, deployment, debugging and testing.
- There still seems to be a gap between the application layer and middleware layer of a typical BioGrid infrastructure because existing middleware services do not fully facilitate the demands of applications such as there is no proper support in any grid middleware for automatic application deployment on all grid nodes.
- It is not trivial to deploy existing bioinformatics applications on available grid testbed (such as NGS, EGEE etc), as this requires the installation and configuration of specific operating system and grid middleware toolkits, which is not, from a biologist end-user point of view, a trivial task.
- It has been observed that there are still many issues with grid based workflow management systems in terms of their support for complex operations (such as loops), legacy bioinformatics applications and tools, use of proper ontology and web services etc. [86].
- The job submission process on existing grid infrastructures seems to be quite complex because of inappropriate maturity of resource broker services.
- Lack of appropriate implementation initiative regarding knowledge grid infrastructure for life sciences.

These facts provide evidence that web and grid technologies are still moving from an infant to a semi-mature state in terms of their proper application and use in life sciences. We believe that much work is to be done in easing the task of exploiting these technologies properly. In particular we recommend:

- Automation of the grid deployment process for legacy applications
- Domain specific Grid application programming models with appropriate toolkits and libraries
- More user friendly interfaces with dynamic problem solving environments, portals and workflows
- Complete virtualization of all resources needed for the development, deployment and execution of an application
- Application of agent technology to support modeling and simulation of complex biological processes
- A single unified grid middleware that could provide the services for Data, Computing, Service and Knowledge Grid
- Bringing the grid middleware at the level of operating system in terms of its ease of use, efficiency and reli-

ability. It should be an operating system for a word wide virtual supercomputer.

- Implementation of more dynamic service and knowledge grid infrastructures for life sciences.
- Development of sophisticated data management services for aggregation, integration, query, visualization and inference of complex biological knowledge and data

Needless to say that the actual list of key open problems is far larger, but the above mentioned are some of the key issues that most directly affect the biologist.

ACKNOWLEDGEMENTS

N. Krasnogor acknowledges the BBSRC for project BB/CS11764/1 and EPSRC for projects GR/T07534/01 and EP/D061571/1; Azhar A. Shah acknowledges the University of Sindh, Pakistan for scholarship SU/PLAN/F.SCH/794; Jacek Blazewicz and Piotr Lukasiak acknowledge the KBN for partial research grant. Finally all the authors acknowledge the anonymous reviewers for their valuable suggestions for the improvement of this paper.

ABBREVIATIONS

BIRN = Biomedical Informatics Research Network

BRIDGES = Biomedical Research Informatics Delivered by Grid Enabled Services

DAG = Direct Acyclic Graph

DAI = Data Access and Integrator

DMS = Data Management Service

DQP = Distributed Query Processor

EBI = European Bioinformatics Institute

EDG = European Data Grid

EGEE = Enabling Grid for EscienceE

EMBOSS = European Molecular Biology Open Software Suite

EMBRACE = European Model for Bioinformatics Research and Community Education

GEBAF = Grid Enabled Bioinformatics Application Framework

GLAD = Grid Life Science Application Developer

GSI = Grid Security Infrastructure

GUI = Graphical User Interface

JDL = Job Description Language

LCG = LHC Computing Grid

LHC = Large Hadron Collider

LIM = Laboratory Information Mgt. System

LRMS = Local Resource Management System

LSF = Load Sharing Facility

LSIDs = Life Science Identifiers

MIAME = Minimum Information About a Microarray Experiment

NCBI = National Center for Biomedical Informatics

NGS = National Grid Service

OGSA = Open Grid Service Architecture

OWL = Web Ontology Language

PBS = Portable Batch System

PSE = Problem Solving Environment

RDF = Resource Description Framework

RFTP = Reliable File Transfer Protocol

SGE = Sun Grid Engine

SOAP = Simple Object Access Protocol

SRB = Storage Resource Broker

SWRL = Semantic Web Rules-Language

UDDI = Universal Description, Discovery and Integration

WFMS = Workflow Management System

WISDOM = Wide *In Silico* Docking On Malaria

WSDL = Web Service Description Language

WSRF = Web Service Resource Framework

XML = eXtensible Markup Language

REFERENCES

[1] Wooley JC, Lin HS, Catalyzing Inquiry at the Interface of Computing and Biology, National Academic Press, Washington, DC 2005.

[2] Malakoff D. NIH urged to fund centres to merge computing and biology. *Science* **1999**; 284: 1742.

[3] Galperin MY. The Molecular Biology Database Collection: 2005 update. *Nucleic Acids Res* **2005**; 33: D5-24.

[4] Baru C, Moore R, Rajasekar A, Wan M, The SDSC storage resource broker, In: Proceedings of CASCON'98 Conference, Toronto, Canada, 1998.

[5] Haas LM, Schwarz PM, Kodali P *et al.* DiscoveryLink: a system for integrated access to life sciences data sources. *IBM Systems J* **2001**; 40: 464-88.

[6] Susumu D, Kazutoshi F, Hideo M, Haruki NShinji S. An Empirical Study of Grid Applications in Life Science: lessons learnt from *Biogrid* project in Japan. *Int J Inf Tech* **2005**; 11: 16-28.

[7] Podvinec M, Maffioletti S, Kunszt P *et al.*, The SwissBioGrid Project: objectives, preliminary results and lessons learned, In: Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing (e-Science 2006) – Workshop on Production Grids. IEEE Computer Society Press, 2006; 148-56.

[8] Chu X, Lonie A, Harris P, Thomas SR, Buyya R, KidneyGrid: a grid platform for integration of distributed kidney models and resources, In: Proceedings of the 4th International Workshop on Middleware for Grid Computing (MGC 2006). ACM Press, NY 2006; 194: 5-12.

[9] Buyya R, Branson K, Giddy J, Abromson D. The Virtual Laboratory: enabling molecular modelling for drug design on the World Wide Grid. *Concurrency Computat: Pract and Exper* **2003**; 15:1-25.

[10] Stevens RD, Robinson AJ, Goble CA. myGrid: personalised bioinformatics on the information grid. *Bioinformatics* **2003**; 19 Suppl 1: i302-04.

[11] Goble C, Wroe C, Steven R, myGrid consortium. The myGrid project: services, architecture and demonstrator, In: Proceedings of the UK e-Science All Hands Meeting, Nottingham, UK, 2003.

[12] Wilkinson MD, Links M. BioMOBY: an open source biological web services proposal. *Brief Bioinform* **2002**; 3: 331-41.

[13] Wilkinson M, Schoof H, Ernst R, Haase D. BioMOBY Successfully Integrates Distributed Heterogeneous Bioinformatics Web Services: the *PlaNet* exemplar case. *Plant Physiol* **2005**; 138: 5-17.

[14] Kerhornou A, Guigo R. BioMoby web services to support clustering of co-regulated genes based on similarity of promoter configurations. *Bioinformatics* **2007**; 23: 1831-33.
[15] Rice P, Longden I, Bleasby A. EMBOSS: the European molecular biology open software suite. *Trends Genet* **2000**; 16: 276-83.
[16] Guillermo PT, Oscar P, Emilio LZ, A survey on grid architectures for bioinformatics, In: Martino BD, Dongarra J, Hoisie A, Yang LT, Zima H Eds, Engineering The Grid: Status and Perspective. American Scientific Publishers, 2006; 109-121.
[17] Krishnan A. A survey of life sciences applications on the grid. *New Generation Computing Archive* **2004**; 22: 111-26.
[18] Naseer A, Stergioulas LK, Integrating Grid and Web Services: a critical assessment of methods and implications to resource discovery, In: Proceedings of the 2nd Workshop on Innovations in Web Infrastructure, Edinburgh, UK, 2005.
[19] Luck R, Networking requirements of the life sciences community, http://www.geant2.net/upload/pdf/Lueck_Rupert.pdf [Accessed: August 2007].
[20] Cheung KH, Smith AK, Yip KYL, Baker CJO, Gerstein MB, Semantic web approach to database integration in the life sciences, In: Baker C, Cheung K Eds, Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences. Springer, NY 2007; 11-30.
[21] Wang X, Gorlitsky R, Almeida JS. From XML to RDF: how semantic web technologies will change the design of 'omic' standards. *Nat Biotechnol* **2005**; 23: 1099-103.
[22] Sahoo SS, Thomas C, Sheth A, York WS, Tartir S, Knowledge Modelling and its Application in Life Sciences: tale of two ontologies, In: Proceedings of the 15th International Conference of World Wide Web, Edinburgh, UK, 2006.
[23] Horrocks I, Patel-Schneider PF, Harmelen FV. From SHIQ and RDF to OWL: the making of a web ontology language. *J Web Semantics* **2003**; 1: 07-26.
[24] Chen YP, Chen Q, Analyzing inconsistency toward enhancing integration of biological molecular databases, In: Proceedings of the 4th Asia-Pacific Bioinformatics Conference, Taipei, Taiwan, 2006.
[25] Zhao J, Wroe C, Goble C *et al*., Using semantic web technologies for representing e-science provenance, In: Sheila MA, Dimitris P, Frank H Eds, The Semantic Web - ISWC 2004. Springer-Verlag, 2004; LNCS 3298: 92-106.
[26] Quan D, Karger DR, How to make a semantic web browser, In: Proceedings of the 13th International World Wide Web Conference, ACM Press, NY 2004; 255-65.
[27] The Gene Ontology Consortium. Creating the Gene Ontology Resource: design and implementation. *Genome Res* **2001**; 11: 1425-33.
[28] The Gene Ontology Consortium. The Gene Ontology (GO) project in 2006. *Nucleic Acids Res* **2006**; 34: D322-28.
[29] Ruttenberg A, Rees J, Luciano J, Experience using OWL DL for the exchange of biological pathway information, In: Proceedings of the Workshop on OWL Experiences and Directions, Galway, Ireland, 2005.
[30] Ruttenberg A, Rees J, Zucker J, What BioPAX communicates and how to extend OWL to help it, In: Proceedings of the Workshop series on OWL: Experiences and Directions, Manchester, UK, 2006.
[31] Whetzel PL, Parkinson H, Causton HC *et al*. The MGED Ontology: a resource for semantics-based description of microarray experiments. *Bioinformatics* **2006**; 22: 866-73.
[32] Navarange M, Game L, Fowler D *et al*. MiMiR: a comprehensive solution for storage, annotation and exchange of microarray data. *BMC Bioinformatics* **2005**; 6: 268-78.
[33] Brazma A, Hingamp P, Quackenbush J *et al*. Minimum Information About a Microarray Experiment (MIAME) - toward standards for microarray data. *Nat Genet* **2001**; 29: 365-71.
[34] Altunay M, Colonnese D, Warade C, High throughput web services for life sciences, In: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05). IEEE Computer Society, DC 2005; 1: 329-34.
[35] Hahn U, Wermter J, Blasczyk R, Horn PA. Text Mining: powering the database revolution. *Nature* **2007**; 448: 130.
[36] Gao HT, Hayes JH, Cai H. Integrating biological research through web services. *Computer* **2005**; 38: 26-31.
[37] Foster I. Globus Toolkit Version 4: Software for service-oriented systems. J Computer Sci and Tech **2006**; 21: 513-20.
[38] EGEE-II overview paper, http://www.dcc.ac.uk/events/policy-2006/EGEE-II%20overview %20paper.pdf [Accessed: August 2007].
[39] gLite3 user guide, https://edms.cern.ch/file/722398/1.1/gLite-3-User Guide.html [Accessed: August 2007].
[40] Furmento N, Hau J, Lee W, Newhouse S, Darlington J, Implementations of a service-oriented architecture on top of Jini, JXTA and OGSA, In: Proceedings of the UK e-science All Hands Meeting, Nottingham, UK, 2003.
[41] Wietrzyk B, Radenkovic M, Semantic life science middleware with web service resource framework, In: Proceedings of the 4th All Hands Meeting, Nottingham, UK, 2005.
[42] Radenkovic M, Wietrzyk B, Life science grid middleware in a more dynamic environment, In: Meersman R, Tari Z, Herrero P *et al*. Eds, Proceedings of OTS Workshops. Springer-Verlag, 2005; LNCS 3762: 264-73.
[43] Luciano JS. PAX of mind for pathway researchers. *Drug Discov Today* **2005**; 10: 937-42.
[44] Joshi-Tope G, Gillespie M, Vastrik I *et al*. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res* **2005**; 33: D428-32.
[45] Brazma A, Robinson A, Cameron G, Ashburner M. One-stop shop for microarray data. *Nature* **2000**; 403: 699-700.
[46] Brazma A. On the importance of standardisation in life sciences. *Bioinformatics* **2001**; 17: 113-4.
[47] Stevens R, Baker P, Bechhofer S *et al*. TAMBIS: transparent access to multiple bioinformatics information sources. *Bioinformatics* **2000**; 16: 184-89.
[48] Komatsoulis GA, Warzel DB, Hartel FW *et al*. caCORE version 3: implementation of a model driven, service-oriented architecture for semantic interoperability. *J Biomed Inform* **2007** (To appear).
[49] Maibaum M, Zamboulis L, Rimon G *et al*. Cluster based integration of heterogeneous biological databases using the AutoMed toolkit, In: Proceedings of the 2nd International Workshop on Data Integration in the Life Sciences, San Diego, USA, 2005.
[50] Shannon PT, Reiss DJ, Bonneau R, Baliga NS. The Gaggle: an open-source software system for integrating bioinformatics software and data sources. *BMC Bioinformatics* **2006**; 7: 176-89.
[51] Karp PD, Riley M, Saier M *et al*. The EcoCyc and MetaCyc databases. *Nucleic Acids Res* **2000**; 28: 56-65.
[52] Rodriguez N, Donizelli M, Novere LN. SBMLeditor: effective creation of models in the Systems Biology Markup Language (SBML). *BMC Bioinformatics* **2007**; 8: 79-87.
[53] Nickerson D, Hunter P, Using CellML in computational models of multi-scale physiology, In: Proceedings of the 27th IEEE Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS 2005). IEEE Press, 2005; 6096-99.
[54] Bard JB, Rhee SY. Ontologies in Biology: design, applications and future challenges. *Nat Rev Genet* **2004**; 5: 213-22.
[55] Stein LD, Mungall C, Shu S *et al*. The Generic Genome Browser: a building block for a model organism system database. *Genome Res* **2002**; 12: 1599-610.
[56] Hermjakob H, Montecchi-Palazzi L, Bader G *et al*. The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data. *Nat Biotechnol* **2004**; 22: 177-83.
[57] Merelli E, Armano G, Cannata N *et al*. Agents in bioinformatics, computational and systems biology. *Brief Bioinform* **2007**; 8: 45-59.
[58] Luc M, Simon M, Carole G, On the use of agents in a bioinformatics grid, In: Proceedings of the 3rd International Symposium on Cluster Computing and the Grid. IEEE Computer Society, DC 2003; 653-61.
[59] Bryson K, Luck M, Joy M, Jones D. Agent interaction for bioinformatics data management. *Applied Artificial Intelligence* **2001**; 15: 917–47.
[60] Decker K, Zheng X, Schmidt C, A multi-agent system for automated genetic annotation, In: Proceedings of the 5th ACM International Conference on Autonomous Agents. ACM Press, NY 2001; 433-40.
[61] Foster I. Service-oriented science. *Science* **2005**; 308: 814-17.
[62] Yen E, Lee HC, Ueng W, Lin S. Building Grid-enabled applications in bioinformatics and digital archive, http://www2.twgrid.org/event/isgc2004/ presentation/0728/ GridAPs.pdf [Accessed: January 2007].

[63] Jacq N, Blanchet C, Combet C. Grid as a bioinformatics tool. *Parallel Computing* **2004**; 30: 1093 -107.

[64] Felsenstein J. Evolutionary Trees from DNA Sequences: a maximum likelihood approach. *J Mol Evol* **1981**; 17: 368-76.

[65] Kommineni J, Abramson D, GriddLeS enhancements and building virtual applications for the GRID with legacy components, In: Proceedings of the European Grid Conference (EGC 2005). Springer-Verlag, 2005; LNCS 3470: 961-71.

[66] Nakada H, Sato M, Sekiguchi S. Design and Implementations of Ninf: towards a global computing infrastructure. *Future Generation Computer Systems* **1999**; 15: 649-58.

[67] Thomas M, Mock S, Boisseau J *et al*., The GridPort toolkit architecture for building grid portals, In: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-01), San Francisco, 2001.

[68] Novotny J. The Grid portal development kit. *Concurrency and Computat: Pract and Exper* **2002**; 14:1129–44.

[69] Chongjie Z, Kelley I, Allen G. Grid Portal Solutions: a comparison of the GridPortlets and OGCE. *Cocurrency and Computat: pract and exper* **2007**; 19: 1739-48.

[70] Ramakirshnan L, Reed MSC, Tilson JL, Reed DA, Grid portals for bioinformatics, In: Proceedings of the 2nd International Workshop on Grid Computing Environments (GCE'06), Tampa, USA, 2006.

[71] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic Local Alignment Search Tool. *J Mol Biol* **1990**; 215: 403-13.

[72] Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* **1994**; 22: 4673-80.

[73] Alameda J, Christie M, Fox G *et al*. The Open Grid Computing Environments Collaboration: portlets and services for science gateways. *Concurrency and Computat: Pract and Exper* **2007**; 19: 921-42.

[74] Letondal C. A Web interface generator for molecular biology programs in Unix. *Bioinformatics* **2001**; 17: 73-82.

[75] Sulakhe D, Rodriguez A, D'Souza M *et al*. GNARE: automated system for high-throughput genome analysis with grid computational backend. *J Clin Monit Comput* **2005**; 19: 361-70.

[76] Casanova H, Berman F, Bartol T. The Virtual Instrument: support for grid-enabled Mcell Simulations. *Int J High Perform Computing Appl* **2004**; 18: 3-17.

[77] Dhar PK, Meng TC, Somani S *et al*. Grid Cellware: the first grid-enabled tool for modelling and simulating cellular processes. *Bioinformatics* **2005**; 21: 1284-87.

[78] Lu E, Xu Z, Sun J, An extendable grid simulation environment based on GridSim, In: Proceedings of the 2nd International Workshop on Grid and Cooperative Computing (GCC 2003). Springer-Verlag, 2004; 3032: 205-08.

[79] Buyya R, Murshed M. GridSim: a toolkit for the modelling and simulation of distributed resource management and scheduling for Grid computing. *Concurrency and Computat: Pract and Exper* **2002**; 14: 1175-1220.

[80] Lamehamedi H, Shentu Z, Szymanski B, Deelman E, Simulation of Dynamic Data Replication Strategies in Data Grids, In: Proceedings of the 17th International Symposium on Parallel and Distributed Processing (IPDPS 2003). IEEE Computer Society, DC 2003; 100-02.

[81] Teo YM, Wang X, Ng YK. GLAD: a system for developing and deploying large-scale bioinformatics grid. *Bioinformatics* **2005**; 21: 794-802.

[82] Trelles O. On the parallelization of bioinformatics applications. *Briefings in Bioinformatics* **2001**; 2: 181–94.

[83] Nobrega R, Barbosa J, Monteiro AP, BioGrid Application Toolkit: a Grid-based problem solving environment tool for biomedical data analysis, In: Proceedings of the VECPAR-7th International Meeting on High Performance Computing for Computational Science, IMPA, Brazil, 2006.

[84] Cannataro M, Comito C, Schiavo FL, Veltri P. Proteus: a grid based problem solving environment for bioinformatics- architecture and experiments. *IEEE Computational Intel. Bull.* **2004**; 3: 07-18.

[85] Choong HS, Byoung JK, Gwan SY, A model of problem solving environment for integrated bioinformatics solution on grid by using Condor, In: Proceedings of the 3rd International Conference on Grid and Cooperative Computing (GCC 2004). Springer-Verlag, 2004; LNCS 3251: 935-38.

[86] Tang F, Chua CL, Ho LY *et al*. Wildfire: distributed, Grid-enabled workflow construction and execution. *BMC Bioinformatics* **2005**; 6: 69-78.

[87] Hull D, Wolstencroft K, Stevens R *et al*. Taverna: a tool for building and running workflows of services. *Nucleic Acids Res* **2006**; 34: W729-32.

[88] Berman F, Chien A, Cooper K. The GrADS Project: software support for high-level grid application development. *Int J High Perform Computing App* **2001**; 15: 327-44.

[89] Goodale T, Allen G, Lanfermann G *et al*., The cactus framework and toolkit: design and applications, In: Proceedings of the 5th International Conference on High Performance Computing for Computational Science, Springer-Verlag, 2003; LNCS 2565: 15-36.

[90] Moskwa S, A Grid-enabled Bioinformatics Applications Framework (GEBAF), Masters Thesis, *School of Computer Science*, The University of Adelaide, 2005.

[91] Stajich JE, Block D, Boulez K *et al*. The bioperl toolkit: Perl modules for the life sciences. *Genome Res* **2002**; 12:1611-18.

[92] Abramson D, Buyya R, Giddy J. A computational economy for grid computing and its implementation in the Nimrod-G resource broker. *Future Generation Computer Systems* **2002**; 18: 1061-74.

[93] Aloisio G, Cafaro M, Fiore S, Mirto M, A WorkFlow management system for bioinformatics grid, In: Proceedings of the Network Tools and Applications in Biology (NETTAB) Workshops, Naples, Italy, 2005.

[94] Yarkhan A, Dongarra JJ. Biological sequence alignment on the computational grid using the GrADS framework. *Future Generation Computer Systems* **2005**; 21: 980-86.

[95] Sinnott R, Atkinson M, Bayer M *et al*. Grid services supporting the usage of secure federated, distributed biomedical data, In: Proceedings of the UK e-Science all hands meeting, Nottingham, 2004.

[96] Blanchet C, Mollon R, Thain D, Deleage G, Grid deployment of legacy bioinformatics applications with transparent data access, In: Proceedings of IEEE Conference on Grid Computing 2006. IEEE Computer Society, 2006; 120-27.

[97] Thain D, Livny M. Parrot: an application environment for data intensive computing. *Scalable Computing: Pract and Exper* **2005**; 6: 9-18.

[98] Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res* **2000**; 28: 45-8.

[99] Smith TF, Waterman MS. Identification of Common Molecular Subsequences. *J Mol Biol* **1981**; 147: 195-97.

[100] Pearson WR. Searching Protein Sequence Libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genom* **1991**; 11: 635-50.

[101] Krauter K, Maheswaran M, Architecture for a grid operating system, In: Buyya R, Baker M, Proceedings of the 1st IEEE/ACM International Workshop on Grid Computing, Springer Verlag, 2000; LNCS 1971: 65-76.

[102] Padala P, Wilson JN, GridOS: operating system services for grid architectures, In: Proceedings of the High Performance Computing (HiPC 2003). Springer-Verlag, 2003; LNCS 2913: 353-62.

[103] Huh EN, Mun Y, Performance analysis for real-time grid systems on COTS operating systems, In: Proceedings of the International Conference on Computational Science (ICCS 2003). Springer-Verlag, 2003; LNCS 2660: 482-90.

[104] Talia D, Towards GRID Operating Systems: from GLinux to a GVM, In: Proceedings of the Workshop on Network Centric Operating Systems, Bruxelles, Belgium, 2005.

[105] Grimshaw AS, Natrajan A, Legion: Lessons learned building a grid operating system, In: Proceedings of the IEEE, 2005; 93: 589-603.

[106] Imamagic E, Radic B, Dobrenic D, Job Management Systems Analysis, In: Proceedings of the *6th CARNET Users Conference*, Zagreb, Croatia, 2004.

[107] Sun Y, Zhao S, Yu H, Gao G, Luo J. ABCGrid: application for bioinformatics computing grid. *Bioinformatics* **2007**; 23: 1175-77.

[108] Benedyczak K, Wronski M, Nowinski A *et al*, UNICORE as Uniform Grid Environment for Life Sciences, In: Proceedings of the European Grid Conference (EGC 2005). Springer-Verlag, 2005; LNCS 3470: 364-73.

[109] Yang CT, Kuo YL, Li KC, Gaudiot JL, On design of cluster and grid computing environment toolkit for bioinformatics applications, In: Proceedings of 6th International Workshop on Distributed Computing (IWDC 2004). Springer-Verlag, 2004; LNCS 3326: 82-87.

[110] Yang CT, Hsiung YC, Kan HC, Implementation and evaluation of a Java based computational grid for bioinformatics applications, In: Proceedings of the 19th IEEE International Conference on Advanced Information Networking and Applications (AINA 2005), IEEE Computer Society, 2005; 1: 298-303.

[111] Li KC, Chen CN, Liu CC, Chang CF, Hsu CW. PCGrid: integration of college's research computing infrastructures using grid technology, In: Proceedings of the National Computer Symposium (NCS'2005), Tainan, Taiwan, 2005.

[112] Chen CN, Li KC, Tang CY, Lin YL, Wang HH, Wu TY, On design and implementation of a bioinformatics portal in cluster and grid environments, In: Proceedings of the 7th International Meeting on High Performance Computing for Computational Science (VECPAR), IMPA, Brazil, 2006.

[113] Gagliardi F, Jones B, Grey F, Begin MEHeikkurinen M, Building an Infrastructure for Scientific Grid Computing: status and goals of the EGEE project. *Philos Transact A Math Phys Eng Sci*, **2005**; 363: 1729-42.

[114] Chakravarti AJ, Baumgartner G, Lauria M, The Organic Grid: self-organizing computational biology on desktop grids, In: Zomaya AY Eds, Parallel Computing for Bioinformatics and Computational Biology: Models, Enabling Technologies and Case Studies, John Wiley & Sons, 2005; 671-703.

[115] Desmedt C, Nabrzyski J, Tsiknakis M. A semantic grid infrastructure enabling integrated access and analysis of multilevel biomedical data in support of post-genomic clinical trials on cancer. *IEEE Transactions on Inform Tech in Biomedicine* **2007**; to appear.

[116] Pukacki J, Nabrzyski J, Stroiński M, Programming grid applications with Gridge, In: Nabrzyski J, Stroiński M Eds, Grid Applications – New Challenges for Computational Methods, 2006; 12: 10-20.

[117] Buyya R, Giddy J, Abramson D, An evaluation of economy-based resource trading and scheduling on computational power grids for parameter sweep applications, In: Proceedings of 2nd Workshop on Active Middleware Services (AMS 2000) in conjunction with HPDC 2000, Pittsburgh, USA, 2000.

[118] Buyya R, Date S, Mizuno-Matsumoto Y, Venugopal S, Abramson D. Neuroscience Instrumentation and Distributed Analysis of Brain Activity Data: a case for eScience on global grids. *Concurrency Computat: Pract and Exper* **2005**; 17(15): 1783-98.

Received: February 22, 2007 Revised: May 30, 2007 Accepted: May 30, 2007