

Artificial Intelligence and Grids: Workflow Planning and Beyond

Yolanda Gil, Ewa Deelman, Jim Blythe, Carl Kesselman, Hongsuda
Tangmunarunkit

USC / Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

{gil, deelman, blythe, carl, hongsuda}@isi.edu

IEEE Intelligent Systems, special issue on E-Science, Jan/Feb 2004.

Abstract

Grid computing is emerging as key enabling infrastructure for science. A key challenge for distributed computation over the Grid is the synthesis on-demand of end-to-end scientific applications of unprecedented scale that draw from pools of specialized scientific components to derive elaborate new results. In this paper, we outline the technical issues that need to be addressed in order to meet this challenge, including usability, robustness, and scale. We describe Pegasus, a system to generate executable grid workflows given a high-level specification of desired results. Pegasus uses Artificial Intelligence planning techniques to compose valid end-to-end workflows, and has been used in several scientific applications. We also outline our design for a more distributed and knowledge-rich architecture.

1. Introduction

Grid computing (see attached Grid Computing callout) is emerging as a key enabling infrastructure for a wide range of disciplines in science and engineering including Astronomy, High Energy Physics, Geophysics, Earthquake engineering, Biology and Global Climate Change [1-3]. By providing fundamental mechanisms for resource discovery, management and sharing, Grids enable geographically distributed teams to form dynamic multi-institutional *virtual organizations* whose members use shared community and private resources to collaborate on the solutions to common problems. This provides scientists with tremendous connectivity across traditional organizations and fosters cross-disciplinary, large-scale research. The most tangible impact of Grids to date may be the seamless integration and access to high-performance computing resources, large-scale data sets and instruments as enabling technologies for advanced scientific discovery. However, scientists now pose new challenges that will require a significant shift to the current Grid computing paradigm.

First and foremost, significant scientific progress can be gained through the synthesis of models, theories, and data contributed across disciplines and organizations. The challenge is to *enable the synthesis on-demand of end-to-end scientific applications of unprecedented scale that draw from pools of specialized scientific components to derive elaborate new results*. Consider, for example, a physics-related application for the Laser Interferometer Gravitational Wave Observatory (LIGO) [4], where instruments collect data

that needs to be analyzed in order to detect gravitational waves predicted by Einstein's theory of relativity. To do this, scientists run pulsar searches in certain areas of the sky for a time period, where observations are processed through Fourier transforms and frequency range extraction software. The analysis may involve composing a workflow of hundreds of jobs and executing them on appropriate computing resources on the Grid, often spanning several days and necessitating failure handling and reconfiguration to handle the dynamics of the Grid execution environment.

Second, the impact of scientific research can be significantly multiplied by broadening the range of applications that it can potentially support beyond science-related uses. The challenge is to *make these complex scientific applications accessible outside the scientific community*. In earthquake science, for example, integrated earth sciences research for doing complex probabilistic seismic hazard analysis can have greater impact, especially when it is used to mitigate the effects of earthquakes in populated areas. Many potential users of scientific models lie outside the scientific community. These users include safety officials, insurance agents and civil engineers that need to evaluate the risk of earthquakes of certain magnitude ranges at potential sites. There is a clear need to isolate the end users from the complexity of the requirements to set up these simulations and execute them seamlessly over the Grid.

In this paper, we begin by discussing the issues that need to be addressed in order to meet the above challenges. We then give an overview of our work to date in Pegasus, a planning system integrated in the Grid environment that takes a user's high level specification of desired results and generates valid workflows that take into account the available resources and submits the workflow for execution on the Grid. We end the paper with our vision for a more distributed planning architecture with richer knowledge sources, and a discussion of the relevance of this work to enable the full potential of the Web as a globally connected information and computation infrastructure.

2. Challenges for Robust Workflow Generation and Management

In order to develop scalable robust mechanisms to address the complexity of the kinds of Grid applications envisioned by the scientific community, we need expressive and extensible ways of describing the Grid at all levels as well as flexible mechanisms to explore tradeoffs in the Grid's complex decision space that incorporate heuristics and constraints into that process. Specifically, the following issues need to be addressed:

Knowledge capture. High-level services such as workflow generation and management systems are starved for information and lack expressive descriptions of entities in the Grid, their relationships, capabilities, and tradeoffs. Current Grid middleware simply does not provide the expressivity and flexibility necessary to make sophisticated planning and scheduling decisions. Something as central to the Grid as resource descriptions are still based on rigid schemas. Although higher-level middleware is under development [2, 5], Grids will have a performance ceiling determined by the limited expressivity and amount of information and knowledge available to make intelligent decisions.

Usability. The exploitation of distributed heterogeneous resources is already a hard problem, much more so when it involves different organizations with specific use policies and contentions. All these mechanisms need to be managed, and sadly today the burden falls on the end users. Even though users think in much more abstract, application-level terms, today's Grid users are required to have extensive knowledge of the Grid computing environment and its middleware functions. For example, a user needs to know how to find the physical locations of input data files through a replica locator, understand the different types of job schedulers running on each host and their suitability for certain types of tasks, and consult access policies in order to make valid resource assignments that often require resolving denial of access to critical resources. Users should be able to submit high-level requests in terms of their application domain. Grids should provide automated workflow generation techniques that would incorporate the

knowledge and expertise required to access Grids while making more appropriate and efficient choices than the users themselves. The challenge of usability is key because it is an insurmountable barrier for many potential users that today shy away from Grid computing.

Robustness. Failures in highly distributed heterogeneous systems are commonplace. The Grid is a very dynamic environment, where the resources are highly heterogeneous and shared among many users. Failures can result from the common hardware and software failures but also from other modes where the policy usage for a resource is changed making the resource effectively unavailable. Worse yet, while the execution of many workflows spans days they incorporate information upon submission that is doomed to change in a very dynamic environment like the Grid. Users today are required to provide details about which replica of the data to use or where to submit a particular task, sometimes days in advance. The user's choices made at the beginning of the execution may not yield good performance further into the run. Even worse, the underlying execution system may have changed so significantly (due to failure or resource usage policy change), that the execution can no longer proceed. Without having knowledge about the history of the workflow execution, the knowledge of the underlying reasons for making particular refining and scheduling decisions, it may be impossible to rescue the execution of the workflow. Grids need more information to ensure proper completion, including knowledge about workflow history, the current status of their subtasks, and the decisions that led to their particular design. The gains in efficiency and robustness of execution in this more flexible environment, especially as applications scale in size and complexity, could be enormous.

Access. The multi-organizational nature of the Grid makes access control a very important and complex problem. The resources need to be able to handle users who belong to different groups, with most likely different access and usage privileges. Grids provide an extremely rich and flexible basis to approach this problem through authentication, security, and access policies both at the user-level and organization-level. Today's resource brokers schedule tasks on the Grid and give preference to jobs based on their predefined policies and those of the resources they oversee. But as the size and number of organizations supported by the Grid grows and users start to be more differentiated (considering the needs of students versus those of scientists), these brokers will need to consider complex policies and resolve conflicting requests from its many users. New facilities are needed to support advance reservations to guarantee availability, and provisioning of additional resources for anticipated needs. Without a knowledge-rich infrastructure, fair and appropriate use of Grid environments will not be possible.

Scale. Today, typical scientific applications on the Grid run over a period of days and weeks and process terabytes of data, and will need to be up to petabytes in the near future. Even the most optimized application workflows carry with them a great danger of lacking in performance when they are actually executing. Such workflows are also fairly likely to fail due to simple circumstances such as for example the lack of disk space. The large amounts of data are only one of the characteristics of such applications. The scale of the workflows themselves also contributes to the complexity of the problem. To perform a meaningful scientific analysis, many workflows, on the order of hundreds of thousands may need to be executed. These various workflows may be coordinated to result in more efficient and cost-effective use of the Grid. Therefore, there is a need to manage complex pools of workflows that balance the access to resources, adapt the execution of the application workflows to take advantage of newly available resources, provision or reserve new capabilities if the foreseeable resources are not adequate, and repair the workflows in case of failures. The scientific advances enabled by such a framework could be enormous.

In summary, Grids today use syntax or schema-based resource matchmakers, algorithmic schedulers, and execution monitors for scripted job sequences which attempt to make decisions with limited information about a large, dynamic, and complex decision space. Clearly, a more flexible and knowledge-rich Grid infrastructure is needed.

3. Pegasus: Generating Executable Grid Workflows

Our focus to date has been workflow composition as an enabling technology that can publish components and compose them together into an end-to-end workflow of jobs to be executed on the Grid. Our approach to this problem is to use Artificial Intelligence planning techniques, where the alternative possible combinations of components are formulated in a search space with heuristics that represent the complex tradeoffs that arise in Grids.

We have developed a workflow generation and mapping system, Pegasus [6, 7, 8, 9, 10], that integrates an AI planning system into a Grid environment. In one of the Pegasus configurations, a user submits an application-level description of the desired data product. The system then generates a workflow by selecting appropriate application components, assigning the required computing resources and overseeing the successful execution. The workflow can be optimized based on the estimated runtime. We tested the system in two different gravitational-wave physics applications where it generated complex workflows of hundreds of jobs that were submitted for execution on the Grid over several days [8].

We cast the workflow generation problem as an AI planning problem in which the goals are the desired data products and the operators are the application components [9, 10]. An AI planning system typically receives as input a representation of the current state of its environment, a declarative representation of a goal state, and a library of operators that can be used to change the state. For each operator there is a description of the states in which the operator may legally be used, called preconditions, and a concise description of the changes to the state that will take place, called effects. The planning system searches for a valid, partially-ordered set of operators that will transform the current state into one that satisfies the goal. The parameters for each operator include the host where the component is to be run while the preconditions include constraints on feasible hosts and data dependencies on required input files. Thus the plan returned corresponds to an executable workflow, assigning components to specific resources that can be executed to provide the requested data product.

The declarative representation of actions and search control in domain-independent planners is convenient for representing constraints such as computation and storage resource access and usage policies as well as heuristics such as preferring a high-bandwidth connection between hosts performing related tasks. In addition, planning techniques can provide high-quality solutions, in part because they can search a number of solutions and return the best ones found, and use heuristics that are likely to guide the search to good solutions.

Pegasus takes a request from the user and builds a goal and relevant initial state for the AI planner using Grid services to locate relevant existing files. Once the plan is completed, Pegasus transforms it into a directed acyclic graph to be passed to DAGMan [11] for execution on the Grid.

Pegasus is being used to generate executable grid workflows in several domains [7], including genomics, neural tomography, and particle physics. One of the applications of the Pegasus workflow planning system is to analyze data from the Laser Interferometer Gravitational-Wave Observatory (LIGO) project, the largest single enterprise undertaken by the National Science Foundation to date, aimed at detecting gravitational waves. Gravitational waves, though predicted by Einstein's theory of relativity, have never been observed experimentally. Through simulations of Einstein's equations, scientists predict that those waves should be produced by colliding black holes, collapsing supernovae, pulsars, and possibly other celestial objects. With facilities in Livingston, Louisiana and Hanford, Washington, LIGO joined gravitational-wave observatories in Italy, Germany and Japan in searching for these signals.

The Pegasus planner that we have developed is one of the tools that scientists can use to analyze data collected by LIGO. In the Fall of 2002, a 17-day data collection effort was held, followed by a two-months run in February of 2003, with additional runs to be held throughout the duration of the project. Pegasus was used with LIGO data collected during the first scientific run of the instrument, which targeted a set of locations of known pulsars as well as random locations in the sky. Pegasus generated end-to-end

4. Future Grid Workflow Management

We envision many distributed heterogeneous knowledge sources and reasoners, as illustrated in Figure 2. The current Grid environment contains middleware to find components that can generate desired results, to find the input data that they require, to find replicas of component files in specific locations, to match component requirements with resources available, etc. This environment should be extended with expressive declarative representations that capture currently implicit knowledge, and should be available to various reasoners distributed throughout the Grid.

In our view, workflow managers would coordinate the generation and execution of pools of workflows. The main responsibilities of the workflow managers are 1) to oversee their assigned workflows development and execution, 2) to coordinate among workflows that may have common subtasks or goals, and 3) to apply fairness rules to make sure the workflows are executed in a timely manner. The workflow managers also identify reasoners that can refine or repair the workflows as needed. One can imagine deploying a workflow manager per organization, per type of workflows or per group of resources whereas the many knowledge structures and reasoners will be independent from the workflow manager mode of deployment. The issue of workflow coordination is particularly crucial in some applications, where significant savings result from the reuse of data products from current or previously executed workflows.

Users provide high-level specifications of desired results and possibly constraints on the components and resources to be used. The user could for example request a pulsar search to be conducted on data collected over a given period of time. The user could constrain the request further by stating a preference for using Teragrid resources or certain application components with trusted provenance or performance. These requests and preferences will be represented declaratively and made available to the workflow manager. They will form the initial smart workflow. The reasoners indicated by the workflow manager will then interpret and progressively work towards satisfying the request. In the case above, workflow generation reasoners would invoke a knowledge source that has descriptions of gravitational-wave physics applications to find relevant application components, and would refine the request by producing a high-level workflow composed of these components. The refined workflow would contain annotations about the reason for using a particular application component and indicate the source of information used to make that decision.

At any point in time, the workflow manager can be responsible for a number of workflows, in various stages of refinement. The tasks in a workflow do not have to be homogeneously refined as it is developed, but may have different degrees of detail. Some reasoners will specialize in tasks that are in a particular stage of development, for example a reasoner that performs the final assignment of tasks to the resources will consider only tasks within the smart workflow that are “ready to run”.

The reasoners would generate workflows that have executable portions and partially specified portions, and iteratively add details to the workflows based on the execution of their initial portions and the current state of the execution environment. This is illustrated in Figure 3.

Users can find out at any point in time the state of the workflow and can modify or guide the refinement process if desired. For example, users can reject particular choices of application components made by a reasoner and incorporate additional preferences or priorities.

Knowledge sources and intelligent reasoners should be accessible as Grid services [12], the widely adopted new Grid infrastructure supported by the recent release of the implementation of the Open Grid Services Architecture (OGSA). Grid services build on web services and extend them with mechanisms to support distributed computation. For example, Grid services offer subscription and update notification functions that facilitate the handling of the dynamic nature of the Grid information. They also offer guarantees of service delivery through service versioning requirements and expiration mechanisms. Grid services are also implemented on scalable robust mechanisms for service discovery and failure handling. The Semantic Web, semantic markup languages, and other technologies such as web services [13-17] offer critical capabilities for our vision.

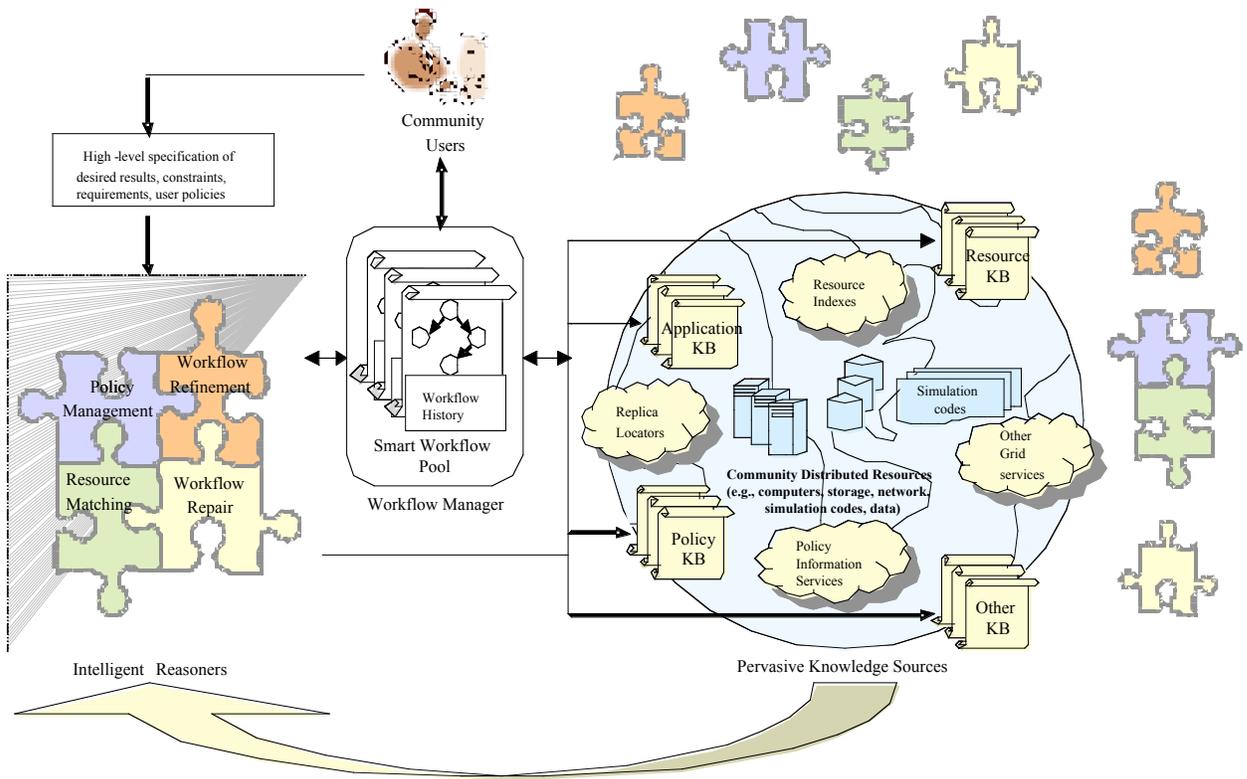


Figure 2: Distributed Grid Workflow Reasoning.

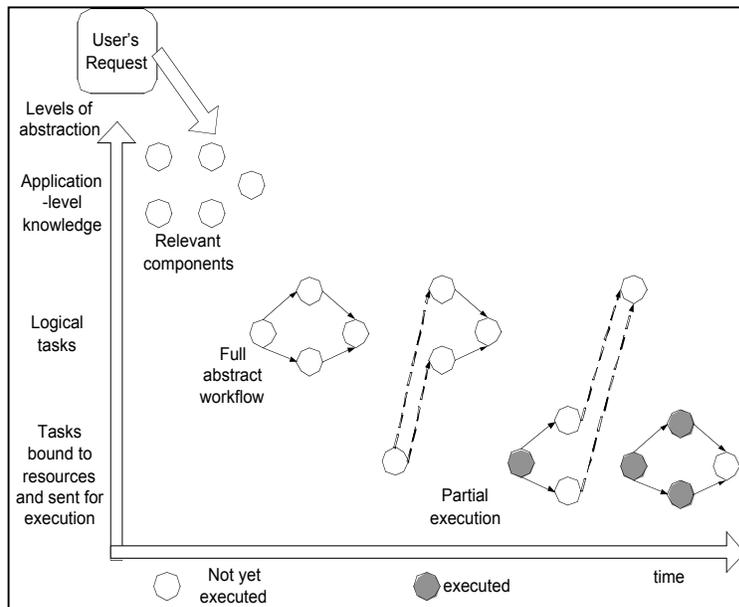


Figure 3: Workflows Are Incrementally Refined Over Time.

5. Related Work

Although scientists naturally specify application-level, science-based requirements, the Grid today dictates that they make quite prosaic decisions, (for example, which replica of the data to use, where to submit a particular task,) and that they oversee workflow execution often over several days when changes in use policies or resource performance may render the original job workflows invalid. Recent Grid projects focus on developing higher-level abstractions to facilitate the composition of complex workflows and applications from a pool of underlying components and services, such as the GriPhyN Virtual Data Toolkit [2] and the GrADS dynamic application configuration techniques [18]. The GriPhyN project is developing catalogs, planners and execution environments to enable the virtual data concept, as well as the Chimera system [1] for provenance tracking and virtual data derivation. There is no emphasis in automated application-level workflow generation, execution repair, or optimization. IVDGL [19] is also centered in data management uses of workflows and also not addressing automatic workflow generation and management. The GrADS project has investigated dynamic application configuration techniques that optimize application performance based on performance contracts and runtime configuration. However, these approaches are based on 1) schema-based representations that provide limited flexibility and extensibility, and 2) algorithms with complex program flows to navigate through that schema space.

MyGrid is a large ongoing UK-funded project to provide a scientist-centered environment to data management for Grid computing, and that shares with our approach the use of a knowledge-rich infrastructure that exploits ontologies and web services. Some of the ongoing work is investigating semantic representations of application components using semantic markup languages such as DAML-S [20], and exploiting DAML+OIL and description logics and inference to support resource matchmaking and discovery. Our work is complementary in that myGrid does not include reasoners for automated workflow generation and repair.

AI planning techniques have been used to compose software components [21, 22] and web services [23, 24]. However this work does not as yet address key areas for Grid computing such as allocating resources for higher quality workflows and maintaining the workflow in a dynamic environment. Distributed planning and multi-agent architectures will be relevant to this work in terms of coordinating the tasks and representations of the different reasoners and knowledge sources. Approaches for building plans under uncertainty, e.g. [25, 26] will be important for handling the dynamics of Grid environments.

6. Conclusions

More declarative, knowledge-rich representations of computation and problem solving will result in a globally connected information and computing infrastructure that will harness the power and diversity of massive amounts of on-line scientific resources. Our work contributes to this vision by addressing two central issues: 1) what mechanisms can map high-level requirements from users into distributed executable commands that pull large numbers of distributed heterogeneous services and resources with appropriate capabilities to meet those requirements? 2) what mechanisms can manage and coordinate the available resources to enable efficient global use and access given the scale and complexity of the applications that will be possible given this highly distributed heterogeneous infrastructure? The result will be a new generation of scientific environments that can integrate diverse scientific results whose sum will be orders of magnitude more powerful than its individual ingredients. The implications will go beyond science and into the realm of the Web at large.

Acknowledgments

We thank Gaurang Mehta, Gurmeet Singh, and Karan Vahi for developing the Pegasus system. We also thank Adam Arbre, Kent Blackburn, Richard Cavanaugh, Albert Lazzarini, and Scott Koranda. The visualization of LIGO data was created by Marcus Thieboux using a picture from the Two Micron All Sky Survey NASA collection. This research was supported in part by the National Science Foundation under grants ITR-0086044 (GriPhyN) and EAR-0122464 (SCEC/ITR), and in part by an internal grant from the Information Sciences Institute.

References

- [1] J. Annis, Y. Zhao, et al., "Applying Chimera Virtual Data Concepts to Cluster Finding in the Sloan Sky Survey," Technical Report GriPhyN-2002-05, 2002.
- [2] P. Avery and I. Foster, "The GriPhyN Project: Towards Petascale Virtual Data Grids," Technical Report GriPhyN-2001-15, 2001.
- [3] E. Deelman, K. Blackburn, et al., "GriPhyN and LIGO, Building a Virtual Data Grid for Gravitational Wave Scientists," In Proceedings of the 11th Intl Symposium on High Performance Distributed Computing, 2002.
- [4] A. Abramovici, W. E. Althouse, et al., "LIGO: The Laser Interferometer Gravitational-Wave Observatory (in Large Scale Measurements)," *Science*, vol. 256, pp. 325-333, 1992.
- [5] T. H. Jordan, C. Kesselman, et al., "The SCEC Community Modeling Environment--An Information Infrastructure for System-Level Earthquake Research." <http://www.scec.org/cme/>
- [6] E. Deelman, J. Blythe, Y. Gil, and C. Kesselman, "Workflow Management in GriPhyN," in Grid Resource Management, J. Nabrzyski, J. Schopf, and J. Weglarz editors, Kluwer, 2003.
- [7] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M. Su, and K. Vahi. Pegasus: Mapping Scientific Workflows onto the Grid. To appear in the Proceedings of the Second European Across Grids Conference, 2004.
- [8] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, A. Lazzarini, A. Arbre, R. Cavanaugh, S. Koranda. "Mapping Abstract Complex Workflows onto Grid Environments," *Journal of Grid Computing*, vol. 1, pp. 25-39, 2003.
- [9] J. Blythe, E. Deelman, Y. Gil, C. Kesselman, A. Agarwal, G. Mehta, K. Vahi. "The Role of Planning in Grid Computing," In Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS), 2003.
- [10] J. Blythe, E. Deelman, Y. Gil, C. Kesselman, "Transparent Grid Computing: a Knowledge-Based Approach," In Proceedings of the National Conference on Intelligent Applications of Artificial Intelligence (IAAI), 2003.
- [11] J. Frey, T. Tannenbaum, et al., "Condor-G: A Computation Management Agent for Multi-Institutional Grids.," *Cluster Computing*, vol. 5, pp. 237-246, 2002.
- [12] I. Foster, C. Kesselman, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Globus Project 2002.
- [13] "Web Ontology Language (OWL), Reference Version 1.0." <http://www.w3.org/TR/2002/WD-owl-ref-20021112/>
- [14] DAML, 2003. <http://www.daml.org/ontologies>
- [15] O. Lassila and R. R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification," W3C Recommendation, World Wide Web Consortium, 22 February 1999. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>
- [16] E. Christensen, F. Curbera, et al., "Web Services Description Language (WSDL) 1.1," W3C Note, 15 March 2001. <http://www.w3.org/TR/wsdl>
- [17] F. Leymann, " Web Service Flow Language (WSFL 1.0)," 2001. [http:// www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf](http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf).

- [18] F. Berman, A. Chien, et al., "The GrADS Project: Software Support for High-Level Grid Application Development," *International Journal of High Performance Computing Applications*, vol. 15, 2001.
- [19] P. Avery, I. Foster, et al., "An International Virtual-Data Grid Laboratory for Data Intensive Science," Technical Report GriPhyN-2001-2, 2001.
- [20] A. Ankolekar, M. Burstein, et al., (The DAML Services Coalition). "DAML-S: Web Service Description for the Semantic Web." In Proceedings of the First International Semantic Web Conference (ISWC), 2002).
- [21] S. A. Chien and H. B. Mortensen, "Automating Image Processing for Scientific Data Analysis of a Large Image Database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 854-859, 1996.
- [22] K. Golden, "Universal Quantification in a Constraint-Based Planner," In Proceedings of the Sixth International Conference on Artificial Intelligence Planning Systems, 2002.
- [23] S. McIlraith and T. Son, "Adapting Golog for Composition of Semantic Web Services," In Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002), 2002.
- [24] D. McDermott, "Estimated-Regression Planning for Interactions with Web Services," In Proceedings of the Sixth International Conference on Artificial Intelligence Planning Systems, 2002.
- [25] J. Blythe, "Decision-Theoretic Planning,," *AI Magazine*, vol. 20, 1999.
- [26] C. Boutilier, T. Dean, et al., "Planning under uncertainty: structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 11, 1999.

CALLOUT: GRID COMPUTING

Grid computing is promising to be the solution to many of today's science problems by providing a rich, distributed platform for large-scale computations, data and remote resource management. The Grid enables scientists to share disparate and heterogeneous computational, storage and network resources as well as instruments to achieve common goals. Although the resources in the Grid often span across organizational boundaries, the Grid middleware is built to allow users to easily and securely access them. The current de-facto standard in Grid middleware is the Globus Toolkit. The toolkit provides fundamental services to securely locate, access and manage distributed shared resources. Globus Information services facilitate the discovery of available resources. Resource management services provide mechanisms for users and applications to schedule jobs onto the remote resources, as well as a means to manage them. Security is implemented using the Grid Security Infrastructure, which is based on the public key certificates. Globus data management services such as the Replica Location Service and GridFTP can be used to securely and efficiently locate and transfer the data in the wide area.

Many projects around the world are undertaking the task of deploying large-scale Grid infrastructure. Among projects in the US are: the International Virtual Data Grid Laboratory (iVDGL), the Particle Physics Data Grid (PPDG) and the Teragrid. In Europe project such as the LHC Computing Grid Project (LCG), the Enabling Grids for E-science and industry in Europe (EGEE) initiative and projects under the UK E-science are building the necessary infrastructure to provide a platform for scientists from various disciplines of Physics, Astronomy, Earth Sciences, Biology and others.

Although the basic Grid building blocks are being widely used, higher-level services dealing with application-level performance and distributed data and computation management are still under research and development. Among projects in the US addressing such issues are the Grid Physics Network (GriPhyN) project, the National Virtual Observatory (NVO), Earth System Grid (ESG), the Southern California Earthquake Center (SCEC) ITR project, and others. In Europe, much research is being carried within the UK-E-science projects, the EU GridLab project and others.

Currently, Grid computing is undergoing a fundamental change; it is shifted toward the Web service paradigm. Web services define a technique for describing accessible software components (i.e., services), methods for discovering them and protocol for accessing them. Grid services extend the Web service models and interfaces to support distributed state management. Among the necessary extensions are the ability to manage transient services and their lifetime, and the ability to introspect the characteristics and states of the services. Grid services can be dynamically created and destroyed. Web services, and therefore Grid services, are neutral to programming language, programming model and system software.

Another important aspect of Grid Services is the support they are receiving from the wide Grid community. Meetings such as the Global Grid Forum bring together a broad spectrum of researchers and developers from academia and industry with the goal of sharing ideas and standardizing interfaces.

The tremendous advances in Grid computing research are possible because of international collaboration and the financial support of a multitude of funding agencies, the National Science Foundation, the Department of Energy, the National Aerospace Agency, and others in the US as well as the European Union and the UK government in Europe; and governments in Asia and Australia.

For more information about the Grid, and the related projects, please refer to the following publications and web sites:

- [1] I. Foster and C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure," Morgan Kaufmann, 1999
- [2] I. Foster, C. Kesselman, et al., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International Journal of High Performance Computing Applications*, vol. 15, pp. 200-222, 2001.
- [3] I. Foster, C. Kesselman, et al., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Globus Project 2002.
- [4] I. Foster, C. Kesselman, et al., "Grid Services for Distributed System Integration," *Computer*, vol. 35, 2002.
- [5] Enabling Grids for E-science and industry in Europe: egee-ei.web.cern.ch/egee-ei/New/Home.htm
- [6] Earth Systems Grid: <http://www.earthsystemgrid.org>
- [7] Global Grid Forum: www.globalgridforum.org
- [8] The Globus Project: www.globus.org
- [9] The Grid Physics Network project: www.griphyn.org
- [10] International Virtual Data Grid Laboratory: www.ivdgl.org
- [11] LHC Computing Grid Project: lcg.web.cern.ch/LCG
- [12] National Virtual Observatory: www.us-vo.org
- [13] Particle Physics Data Grid: www.ppdg.net
- [14] Southern California Earthquake Center: www.scec.org