iShare – Bringing the TeraGrid to the User's Desktop

Ayon Basumallik
School of Electrical and Computer Engineering,
Purdue University
http://www.ece.purdue.edu/ParaMount
basumall@purdue.edu

Xiaojuan Ren
School of Electrical and Computer Engineering,
Purdue University
http://www.ece.purdue.edu/ParaMount
renx@purdue.edu

Rudolf Eigenmann
School of Electrical and Computer Engineering,
Purdue University
http://www.ece.purdue.edu/ParaMount
eigenman@purdue.edu

Sebastien Goasguen
Rosen Center for Advanced Computing,
Purdue University
sebgoa@purdue.edu

Abstract

iShare is an Internet-Sharing system that supports end-users as well as providers of computing resources (applications, data and hardware). iShare allows providers to disseminate resources and users to access these resources in a way that allows open participation. A fully decentralized organization for resource dissemination is enabled via the integration of a peer-to-peer (P2P) system and web standards such as XML and RDF. iShare has an open extensible architecture that allows different access mechanisms and protocols to be plugged in. It delivers a desktop-based environment for publishing and using remote resources, which decouples the computing environment perceived by end users from the underlying physical platforms.

This paper describes the iShare plug-ins, implemented using the Java Commodity Grid (CoG) Kit, that enable TeraGrid resources to be shared and used through the iShare desktop. TeraGrid resources are disseminated through the publication and discovery functionalities that are a part of iShare. For providing users access to TeraGrid resources, iShare allows certificate based authentication, remote job submission and file transfers. By supporting a variety of access protocols such as GridFTP and SRB, iShare allows users to access data collections across distributed and heterogeneous platforms. Together, these plug-ins enable the end-user to effectively discover and use TeraGrid resources from the desktop.

1. Introduction

The TeraGrid [2] has emerged as a valuable computational infrastructure in enabling scientific discovery. The TeraGrid comprehensively addresses the needs of today's computational scientists by providing high-performance computing cycles that enable more powerful simulation and numerical approaches; network and storage resources that accommodate larger datasets to allow modelling of scientific data at finer resolutions over larger meshes; and libraries and tools to facilitate the development of scientific applications.

To expose scientists to the plethora of possibilities provided by the TeraGrid, there is a need for tools and environments that allow them to easily interface with TeraGrid resources. While TeraGrid resources include software stacks for essential activities such as authentication, job submission and resource monitoring, directly learning and using these may be a daunting task for a large class of users – especially end-users habituated to a more windows desktop style interface.

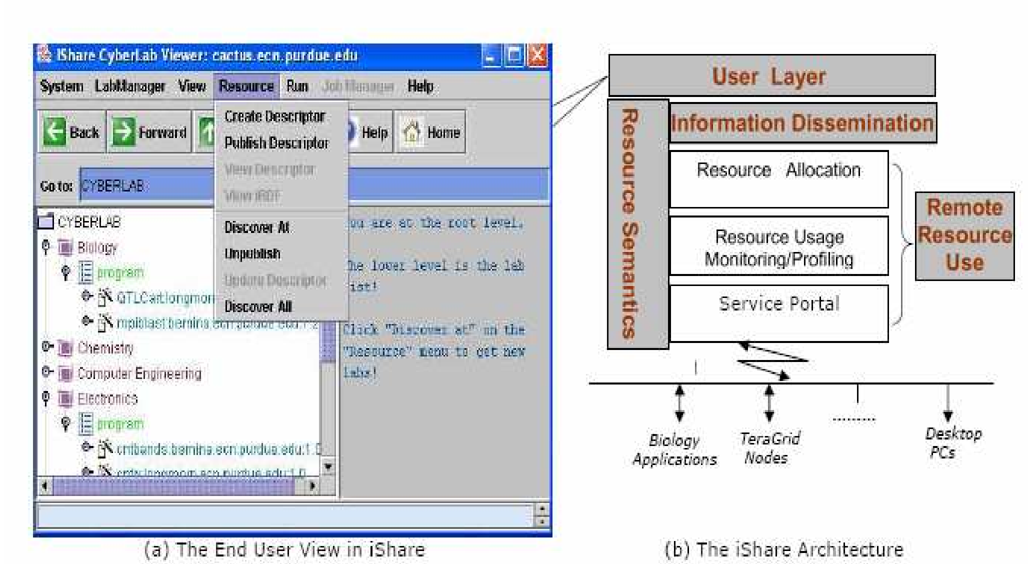
One commonly adopted approach to alleviate this problem is the development of web portals or web-based application servers. Portals provide an easier to use interface to applications deployed on computational grids. However, portals generally have a centralized architecture and thus may not be facile in terms of open participation of the end-user who wants to share his applications and datasets with other users. The TeraGrid, apart from being a valuable computational resource also holds promise in fostering greater collaboration between scientists. Thus, in the context of the TeraGrid, the lack of a simple methodology for sharing applications and data is a significant drawback.

A related observation is that scientific computations tasks often make use of a combination of grid and non-grid resources. For example, a user may want to run the computationally intensive part of his simulation on a high-performance computing platform on the TeraGrid but may do the pre and post-processing of his data using other resources. To be effective, the user's computing environment must seamlessly integrate the use of grid resources with the use of external resources.

In this paper, we present iShare – a decentralized peer-to-peer system for sharing applications, machines and data. iShare is a novel Internet sharing system that addresses the issues discussed in the previous paragraphs. Salient features of iShare include -

- *Intuitive Desktop User Interface :* iShare has graphical user interfaces that allow users to browse resources, run applications and see their output. It also has graphical interfaces that allow easy publishing and discovery of applications, machines and data.
- *Seamless Integration of Heterogeneous Resources :* iShare supports resources with a variety of access mechanisms. The iShare desktop itself is a lightweight portable client implemented in Java. Additionally it requires only user-level privileges for installation.

These features make iShare an effective computing environment for TeraGrid users, enabling them to discover and use TeraGrid resources from within an intuitive graphical desktop interface. The decentralized architecture of iShare, along with its publication and discovery interfaces, also allows resources to be disseminated and shared easily. In this paper, we describe the iShare system, along



(a) The End User View in iShare

(b) The iShare Architecture

Figure 1. The End User View and the Structure of iShare.

with the plug-ins, that effectively brings the TeraGrid to the user's desktop.

The rest of this paper is organized as follows – Section 2 provides an overview of the iShare system. Section 3 describes the iShare plug-ins for sharing, discovering and using TeraGrid resources. Section 4 concludes the paper and also discusses future work. Related work is discussed throughout the paper.

2. A Brief Overview of iShare

In this section, we present an overview of the iShare Internet sharing system [9]. To design iShare, we built on prior experience with the Purdue University Network Computing Hub, PUNCH [6, 7],in which two of our authors were involved. Among the motivations for the iShare design were to solve some of the weaknesses observed in PUNCH: adding new resources needed to go through a central point of management, and the requirements for adding new machine resources at times conflicted with the administrative rules of their owners. iShare removes these barriers by allowing resource providers to describe and publish their resources in an open manner: composing resource descriptors and posting them on the Web. To make the process easy, iShare provides rich tools for resource description, resource publication, and resource access management.

At a high level, iShare consists of the four main modules in Figure 1(b). (1) *Resource Semantics* contains tools for describing resources; (2) *Information Dissemination* provides functionality for publishing and discovering resources; (3) *Remote Resource Use* contains resource monitoring, resource allocation, and resource access (authentication and remote execution) capabilities. (4) *User Layer*, depicted in Figure 1(a), defines the interface to iShare users.

The following subsections describe these modules in greater detail.

2.1 Open Resource Semantics and Description

iShare supports three types of resources at an equal level: software services (application programs), service platforms (machines), and data. Available resources are organized into a hierarchical structure, grouping together semantically-related information. This organization enables an underlying P2P system to support resource dissemination in a decentralized manner.

To enable resource sharing, resource providers need to specify the semantics on what is shared, who is allowed to share, and how sharing occurs. iShare adopts the Resource Description Framework (RDF) [13] as the description language for resource semantics. RDF is an XML-based language for describing information contained in a Web resource. In iShare, RDF descriptors contain meta-data describing resource attributes. Typical resource meta-data for software services include location, connection protocol, execution syntax, pre (post)-process operations, and platform requirements. To assist the resource providers, iShare has a form-based GUI that allows RDFs to be created with ease.

A provider of resources can post their RDF files on a URL specified through iShare's publishing tool. All the posted RDF information is organized in a hierarchical way. At the highest level are the *Cyberlaboratories*, containing resources for a certain discipline. The hierarchical organization enables efficient resource discovery and resource matching.

New types of resources can be made to interoperate by adding "plug-ins" into iShare's modules for resource publishing, semantic parsing and translation, and remote execution. Section 3 describes the plug-ins that were implemented to connect TeraGrid resources (www.teragrid.org) to iShare.

2.2 Decentralized Information Dissemination and Discovery

A decentralized organization is necessary to achieve scalability in publication and discovery of information. iShare realizes decentralization through the integration of the Web infrastructure and a peer-to-peer (P2P) system. A provider of resources can post their availability on any web page or simply use iShare's publish functionality to publish resource descriptors.

Resources in iShare are hierarchically organized into *Cyberlaboratories*. The resources in one lab are semantically related in their functionalities, which we call *semantic locality*. Resources are described by meta-data, which form a tree representing the hierarchical name space. iShare realizes a decentralized directory service by distributing the hierarchical name space to the underlying P2P network, which supports the publishing and discovery process. Resources in an application discipline are discovered incrementally by traversing the hierarchical tree from its root Cyberlab. An end user with specific interests may request resource discovery under specific Cyberlabs.

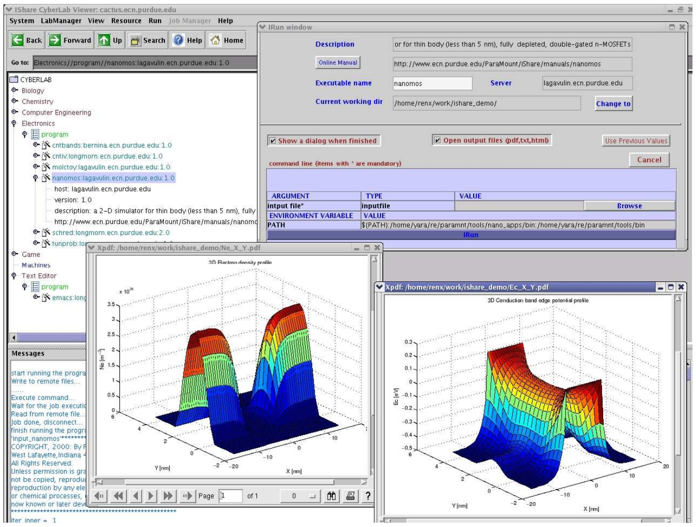


Figure 2. Remote Application Execution Using iShare.

2.3 Remote Program Execution

Mechanisms for remote resource utilization are at the core of Internet sharing systems. iShare builds on a large number of contributions in this area [4] for its remote file transfer and its remote program execution. iShare employs the state-of-the-art technologies in authentication, such as SSH, Public Key Infrastructure [8] and GSI [3].

iShare distinguishes between *pinned* and *unpinned* application resources. Pinned resources are those that are tied to a particular machine resource, while unpinned applications simply specify requirements (through the RDF) such as architecture or OS type, required runtime libraries and so on. For unpinned application resources, iShare performs matching of the application to a suitable machine resource.

Once iShare has an application resource and a matching machine resource, it establishes the user interface to a remote program. This involves the parsing of the machine and application RDFs to initiate suitable protocols for authentication, staging of data and executables and transferring input and output data between the execution platform and the user's desktop. Figure 2 shows a snapshot of remote execution through iShare. iShare supports remote execution of batch-oriented and fully interactive applications. For interactive applications, iShare integrates a Virtual Network Computing (VNC) interface.

3. iShare Plug-ins for TeraGrid Resources

The TeraGrid contains significant data, application and machine resources for the execution of science and engineering applications. We now describe an effort to make these resources available in iShare.

The base iShare system has modules for resource publishing, for resource semantics parsing, and for remote execution using SSH. We implemented three sets of plug-ins to extend these modules to allow TeraGrid resources to be accessed from iShare. Additionally, to allow easy access to the data resources on the TeraGrid, we created a general FTP client type user interface and implemented plug-ins that allow data to be accessed from GridFTP [12] servers and Storage Resource Brokers [10]. A schematic representation of these plug-ins and their functions is depicted in Figure 3.

3.1 RDF Extensions to Support Grid Applications and Machine Resources

The first set of plug-in extends the Resource Description format to accommodate grid resources. Apart from the usual parameters, specification of machine and application resources on a grid typically require additional details such as queue names and accounting information. We extended the iShare user interfaces to allow these parameters to be specified, wherever appropriate, during resource publication and remote execution. Additionally, we implemented two sets of plug-ins – the first extends the RDF to accommodate these additional details specified during resource publication and the second extends the RDF parser module to make appropriate use of these parameters for job submission on TeraGrid resources.

3.2 Remote Execution

iShare already has a remote execution module that uses SSH to run remote jobs. We implemented a set of plug-ins that extend this module for job submission to grid systems. For this implementation, we used the Java CoG kit [11]. The Java CoG kit provides a pure java implementation of a subset of Globus tools and libraries. Since iShare is also implemented in Java, the Java CoG kit provides a suitable set of libraries for developing the TeraGrid plug-ins.

For remote execution on the TeraGrid, two important functions that the plug-in needed to accomplish were (1) GSI based authentication using TeraGrid certificates and (2) Job submission using the Globus Resource Allocation Manager (GRAM) protocol. We implemented certificate management and authentication using the certificate management API provided by the Java CoG kit. For job submission, instead of creating a low level GRAM client with the *org.globus.gram* package, we used the *Task* abstraction [1] provided by the CoG kit. The use of abstraction enhances the stability of this iShare plug-in by decoupling it from the actual protocol implementation, thus insulating it from any protocol changes with future versions of GRAM and Globus. The use of the task abstraction also provides for a more simple, maintainable and reusable implementation of remote job submission on Grids.

3.3 Plug-Ins for Data Transfer

Data resources constitute an important class of TeraGrid resources. iShare enhances the end-user's computing experience by allowing seamless access to these data resources as well as non-grid data resources. Figure 4 shows a snapshot of the FTP user-interface provided by iShare. Using this interface, users may browse and transfer data between locations that have different access protocols (Secure FTP, GridFTP, SRB, local and network file systems).

To implement the GridFTP based transfers, we used the *FileResource* and *Task* abstractions of the Java CoG kit. As in the case of remote execution, the use of these abstractions enabled a cleaner, stable and reusable implementation. For implementing transfers from and browsing of data collections on Storage Resource Brokers (SRB), we used the Jargon [5] API.

We implemented a set of plug-ins to extend the iShare RDF to accommodate descriptions of these diverse data resources. A salient feature of this RDF plug-in is that it allows any number of format or item-specific attributes to be stored in the data descriptor. This provides the flexibility of using domain specific attributes in the future for enhanced functionality, such as transferring only those parts of a file matching the user's requirements, which is commonly desired for large datasets like climate modelling data.

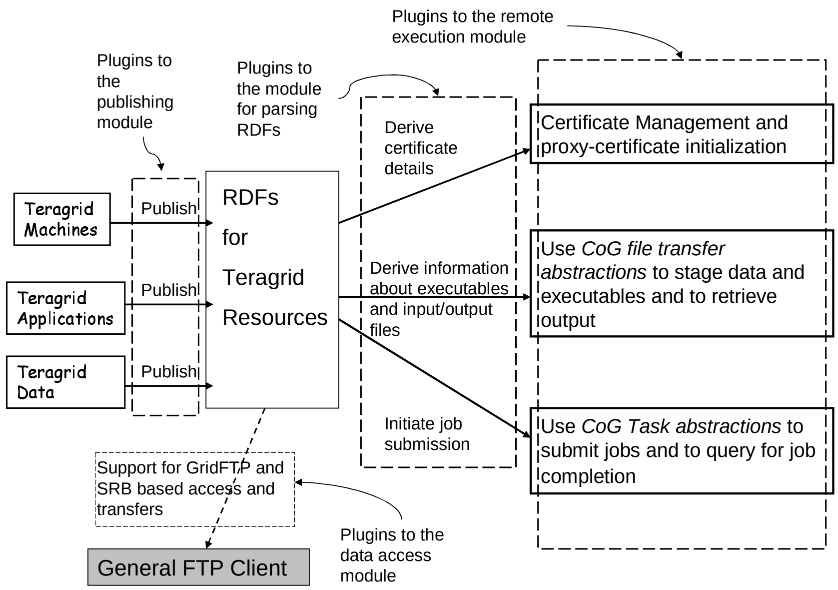


Figure 3. iShare Plug-ins to enable resource sharing on TeraGrid

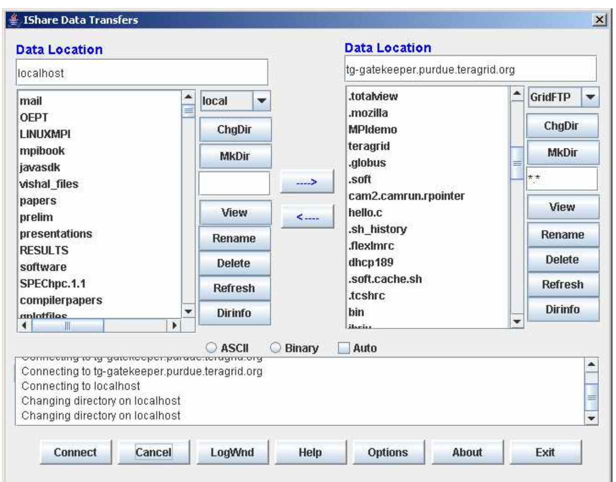


Figure 4. iShare Data Transfer Interface

4. Conclusion

In this paper, we have presented the iShare Internet Sharing system and the plug-ins that allow users to more effectively use TeraGrid resources from a graphical desktop interface. The TeraGrid plug-ins that we have developed are actually reusable for interfacing with any Grid system that uses GSI based authentication and Globus software for job management and data transfers. In our implementation, we have made use of Java CoG kit abstractions to make our plug-ins stable, maintainable and reusable. With these plug-ins, we have extended the existing resource dissemination, discover and remote resource usage functionalities of iShare to TeraGrid users.

For future work, we plan to incorporate the idea of *composition* into iShare. We envision that users will use iShare to compose published iShare resources into resource groups – combining workflows on application resources with machine and data resources. We are currently developing algorithms and additional iShare modules to this end. With its unique blend of end-user tools for sharing, discovery, remote usage and composition, iShare will continue to bring the TeraGrid to the user's desktop.

References

[1] K. Amin, M. Hategan, G. von Laszewski, and N. J. Zaluzec. Abstracting the Grid. In *Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2004)*, pages 250–257, La Coruña, Spain, 11-13 Feb. 2004.

[2] C. Catlett. The philosophy of TeraGrid: Building an open, extensible, distributed terascale facility. In *Proc. CCGRID*, 2002.

[3] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *CCS '98: Proceedings of the 5th ACM conference on Computer and communications security*, pages 83–92, New York, NY, USA, 1998. ACM Press.

[4] GGF. Global Grid Forum, 1998. http://www.ggf.org.

[5] JARGON. Java Api for Real Grids On Networks . http://www.sdsc.edu/srb/jargon/index.html.

[6] N. H. Kapadia and J. A. B. Fortes. PUNCH: An architecture for Web-enabled wide-area network-computing. In *Proc. Cluster Computing*, 1999.

[7] I. Park, N. H. Kapadia, R. J. Figueiredo, R. Eigenmann, and J. A. B. Fortes. Towards an integrated, Web-executable parallel programming tool environment. In *Proc. Supercomputing Conference*, 2000.

[8] PKI. Public Key Infrastructure, 1999. http://www.pkilaw.com/.

[9] X. Ren and R. Eigenmann. iShare - open internet sharing built on peer-to-peer and web. In *Proc. European Grid Conference*, Amsterdam, The Netherlands, Feb. 2005.

[10] SRB. The SDSC Storage Resource Broker. "http://www.sdsc.edu/srb/index.php/Main_Page".

[11] G. von Laszewski, I. Foster, J. Gawor, and P. Lane. A Java Commodity Grid Kit. *Concurrency and Computation: Practice and Experience*, 13(8-9):643–662, 2001.

[12] E. W. Allcock. Gridftp: Protocol extensions to ftp for the grid. http://www.globus.org/alliance/publications/papers/GFD-R.0201.pdf.

[13] W3C-RDF. Resource Description Framework (RDF): Concepts and Abstract Syntax, 2004. http://www.w3.org/TR/rdf-concepts.