รายการอ้างอิง

1. Foster, I and Kesselman, C. 1998. The Grid: Blueprint for a New Computing Infrastructure. California: Morgan Kaufmann.
2. Foster, I. 2002. The Grid: A New Infrastructure for 21st Century Science. Physics Today 55, 2: 42-47.
3. Milojicic, D.S., et al. 2002. Peer-to-peer computing. Technical Report HPL-2002-57, Hewlett Packard Laboratories.
4. Andy, O. 2001. Peer-To-Peer, Harnessing the Power of Disruptive Technology. O'Reilly.
5. Moore, D. and Hebeler, J. 2001. Peer-to-Peer: Building Secure, Scalable, and Manageable Networks. (n.p.): McGraw Hill.
6. Intel 2001. Peer-to-Peer-Enabled Distributed Computing. Intel White Paper.
7. Napster. 2001. The Napster home page [Online]. Avialable from: http://www.napster.com.
8. Gnutella. 2001. The Gnutella home page [Online]. Avialable from: http://gnutella.wego.com.
9. KaZaA. 2001. The KaZaa home page [Online]. Avialable from: http://www.kazaa.com.
10. B. Cohen. 2003. Incentives build robustness in BitTorrent. Proceedings of the First Workshop on the Economics of Peer-to-Peer Systems, Berkeley, California.
11. SETI@home. 2001. The SETI home page [Online]. Avialable from: http://setiathome.ssl.berkeley.edu
12. Mersenne Prime Search. The great internet mersenne prime search [Online]. Avialable from: http://www.mersenne.org
13. Foster, I. and Kesselman, C. 1998. The Globus Project: A Status Report. Proceedings of IPPS/SPDP '98 Heterogeneous Computing Workshop, pp. 4-18.
14. Foster, I. and Kesselman, C. 1997. Globus: A Metacomputing Infrastructure Toolkit. International J. Supercomputer Applications 11(2) : 115-128.
15. Sun Microsystems, Inc. JXTA v2.3.x: Java Programmer's Guide, January 21, 2005.

16. Kato, D. 2002. GISP: Global Information Sharing Protocol - A Distributed Index for Peer-to-Peer Systems. Proceedings of 2nd International Conference on Peer-to-Peer Computing, Sweden.

17. Kato, D. (n.d.): GISP Specification (protocol version 3.4 beta4) [Online]. Avialable from: http://gisp.jxta.org/gisp-spec-3.4.txt

18. Foster, I., Kesselman, C., Nick, J. and Tuecke, S. 2002. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Proceedings of the Open Grid Service Infrastructure WG, Global Grid Forum. June 22, 2002

19. Foster, I., Kesselman, C., and Tuecke, S. 2001. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications 15(3).

20. Lee. C., Kesselman, et al. 1998. A. The Quality of Service Component for the Globus Metacomputing System. Proceedings of IWQoS '98, pp. 140-142.

21. Jacob, B. 2004. Grid computing: What are the key components? Taking advantage of Grid computing for application enablement [Online]. Available from: http://www-106.ibm.com/developerworks/grid/library/gr-overview.

22. The Globus Alliance. 2004. The Globus Project [Online]. Available from: http://www.globus.org

23. Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S. and Kesselman, C. Grid Service Specification. Open Grid Service Infrastructure WG, Global Grid Forum, Draft 2, 7/17/2002.

24. Foster, I., Kesselman, C., Nick, J. and Tuecke, S. 2002. Grid Services for Distributed System Integration. Computer 35(6).

25. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E. (Editors). 2000. Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation 6 October 2000.

26. Clarke, I. 1999. A Distributed Decentralized Information Storage and Retieval System, unpublished report, Division of Informatics, University of Edinburgh. freenet.sourceforge.net/freenet.pdf.

27. Clarke, I., I., Miller, S.G., Wong, T.W., Sandberg, O. and Wiley, B. 2002. Protecting Free Expression Online with Freenet. IEEE Internet Computing 6(January-February): 40-49.

28. Clarke, I., Sandberg, O, Wiley, B. and Hong, T.W. 2001. Freenet: A Distributed Anonymous Information Storage and Retrieval System. Designing Privacy Enhancing Technologies. Proceeding of International Workshop on Design Issues in Anonymity and Unobservability, LNCS 2009, ed. by H. Federrath. Springer, New York.

29. FreeNet. 2001. The FreeNet home page [Online], Avialavle from: http://freenet.sourceforge.net and http://www.freenetproject.org.

30. Groove Networks. 2000. Introduction to Groove. Groove Networks White Paper.

31. Groove Networks. 2000. Connection Age. Groove Networks White Paper.

32. Groove Networks. 2001. Groove Networks Product Backgrounder. Groove Networks White Paper. www.groove.net/pdf/

33. Groove Networks. 2001. Groove and .NET Brief. Groove Networks Brief. www.groove.net/feature/webandpeer/groove_and_dotnet.pdf

34. John Deep. Aimster, John Deep v. RIAA [Online]. Avialable from: www.aimster.org.

35. Bolcer, G. ET AL. 2000. Peer-to-Peer Architectures and the Magi Open Source Infrastructure. Endeavors technologies White Papers.

36. Bolcer, G, 2001. Magi: An Architecture for Mobile and Disconnected Workflow. White paper (www.endeavors.com)

37. SoftWax. 2001. SoftWax Technologies [Online]. Avialable from: http://www.softwax.com.

38. iMesh. Over 15,000,000 Music downloads and MP3's - 100% legal [Online]. Avialable from: http://www.imesh.com/

39. Traversat, B. Arora, A. Abdelaziz, M. Duigou, M. Haywood, C. Hugly, J. Pouyoul, E. Yeager, B. 2002. Project JXTA 2.0 Super-Peer Virtual Network. Available:http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf

40. Avaki Corporation. 2001. Avaki 2.0 Concepts and Architecture. White paper. www.avaki.com/papers/AVAKI_concepts_architecture.pdf.

41. Entropia. Entropia - PC Grid Computing [Online]. Avialabe from: www.entropia.com.

42. Publius Censorship Resistant Publishing System. Publius Home Page [Online]. Avialable from: http://www.cs.nyu.edu/waldman/publius.

43. The Free Haven Project. The Free Haven Project [Online]. Avialable from: http://www.freehaven.net/

44. Jabber SoftwareFoundation. 1999. Jabber: Open Instant Messaging and a Whole Lot More, Powered by XMPP [Online]. Avialable from: http://www.jabber.org.

45. Hall Gailey, J. 2001. Introducing .NET My Services. MSDN Library note [Online]. Avialable from: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/myservintro.asp.

46. Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S. 2001. A Scalable Content-Addressable Network. Proceedings of the SIGCOMM, pp. 161-172.

47. OpenCOLA. 2001. OpenCOLA 2001 [Online]. Avialable from: http://www.opencola.com

48. Waterhouse, S., Doolin, D.M., Kan G. and Faybishenko, Y. 2002. Distributed Search in P2P Networks. IEEE Internet Computing 6 (January-February): 68-72.

49. Yahoo. 2001. Yahoo! [Online]. Avialable from: http://www.yahoo.com.

50. AOL. 2001. AOL [Online]. Avialable from: http://www.aol.com.

51. Buzzpad. 2001. Buzzpad [Online]. Avialable from: http://www.buzzpad.com

52. Williams Rice, A. and Mahon B. 2000. Peer Networking. Deutshe Banc Alex. Brown White Paper.

53. Quazal. Quazal Net-Z: Simplifying Multiplayer Game Development [Online]. Avialable from: www.quazal.com.

54. CenterSpan. 2001. Scour Exchange [Online]. Avialable from: http://www.centerspan.com.

55. Stoica, I., Morris, R., Karger, D., Kaashoek, M. F. and Balakrishnan, H. 2001. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. Proceedings of ACM SIGCOMM, San Diego, California.

56. SHA-1. 1997. American National Standards Institute, American National Standard X9.30.2-1997: Public Key Cryptography for the Financial Services Industry - Part 2: The Secure Hash Algorithm (SHA-1).

ภาคผนวก

ภาคผนวก ก
เค้าร่างเอ็กซ์เอ็มแอลสำหรับเมสเสจจีไอเอสพี

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:gisp="http://gisp.jxta.org/protocol/3.4/beta4"
      targetNamespace="http://gisp.jxta.org/protocol/3.4/beta4"
      elementFormDefault="qualified">

 <xsd:simpleType name="pidType">
  <xsd:annotation>
   <xsd:documentation>
    hex string representation of 160bit integer
    example="123456789abcdef0123456789abcdef012345678"
   </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
   <xsd:pattern value="([a-z0-9])+(/([a-z0-9])+)?" />
  </xsd:restriction>
 </xsd:simpleType>

 <xsd:attributeGroup name="extensibleAttributes">
  <xsd:annotation>
   <xsd:documentation>
    All simple element should be extended to refer this attribute
            group.
   </xsd:documentation>
  </xsd:annotation>
  <xsd:anyAttribute namespace="##other" />
 </xsd:attributeGroup>

 <xsd:complexType name="extensibleType" abstract="true">
  <xsd:annotation>
   <xsd:documentation>
    All complex element should extend this type so that any ##other
            element is allowed.
   </xsd:documentation>
  </xsd:annotation>
```

```
<xsd:sequence>
  <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
 </xsd:sequence>
 <xsd:anyAttribute namespace="##other" />
</xsd:complexType>

<xsd:element name="message">
 <xsd:annotation>
  <xsd:documentation>
   Top element for the GISP message
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:complexContent>
   <xsd:extension base="gisp:extensibleType">
    <xsd:sequence>
     <xsd:element ref="gisp:src" minOccurs="0" />
     <xsd:element ref="gisp:seen" minOccurs="0"
           maxOccurs="unbounded" />
     <xsd:element ref="gisp:peer" minOccurs="0"
           maxOccurs="unbounded" />
     <xsd:element ref="gisp:search" minOccurs="0"
           maxOccurs="unbounded" />
     <xsd:element ref="gisp:insert" minOccurs="0"
           maxOccurs="unbounded" />
     <xsd:element ref="gisp:query" minOccurs="0"
           maxOccurs="unbounded" />
     <xsd:element ref="gisp:result" minOccurs="0"
           maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" />
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
</xsd:element>

<xsd:element name="src">
 <xsd:annotation>
  <xsd:documentation>
```

```
    Source Address of the message
   </xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
   <xsd:simpleContent>
    <xsd:extension base="xsd:string">
     <xsd:attributeGroup ref="gisp:extensibleAttributes" />
    </xsd:extension>
   </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

<xsd:element name="seen">
 <xsd:annotation>
  <xsd:documentation>
   Address that is already received by the source peer of the
           message.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:complexContent>
   <xsd:extension base="gisp:extensibleType">
    <xsd:attribute name="message" type="xsd:string"
           use="required" />
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
</xsd:element>

<xsd:element name="peer">
 <xsd:annotation>
  <xsd:documentation>
   Peer information
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:complexContent>
   <xsd:extension base="gisp:extensibleType">
    <xsd:sequence>
```

```
      <xsd:element ref="gisp:pid" />
      <xsd:element ref="gisp:addr" />
      <xsd:element ref="gisp:ttl" />
      <xsd:element ref="gisp:sent" minOccurs="0"
            maxOccurs="unbounded" />
     </xsd:sequence>
    </xsd:extension>
   </xsd:complexContent>
  </xsd:complexType>
</xsd:element>

<xsd:element name="pid">
 <xsd:annotation>
  <xsd:documentation>
   pid for the peer
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:simpleContent>
   <xsd:extension base="gisp:pidType">
    <xsd:attributeGroup ref="gisp:extensibleAttributes" />
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
</xsd:element>

<xsd:element name="addr">
 <xsd:annotation>
  <xsd:documentation>
   Address of the peer
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:simpleContent>
   <xsd:extension base="xsd:string">
    <xsd:attributeGroup ref="gisp:extensibleAttributes" />
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
```

```
</xsd:element>

<xsd:element name="ttl">
 <xsd:annotation>
  <xsd:documentation>
   Milliseconds to live of peer or pair data
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:simpleContent>
   <xsd:extension base="xsd:integer">
    <xsd:attributeGroup ref="gisp:extensibleAttributes" />
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
</xsd:element>

<xsd:element name="sent">
 <xsd:annotation>
  <xsd:documentation>
   Sent address of peer or pair data
  </xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
   <xsd:simpleContent>
    <xsd:extension base="xsd:string">
     <xsd:attributeGroup ref="gisp:extensibleAttributes" />
    </xsd:extension>
   </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

<xsd:element name="search">
 <xsd:annotation>
  <xsd:documentation>
   Search for peer
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
```

```
   <xsd:complexContent>
    <xsd:extension base="gisp:extensibleType">
     <xsd:sequence>
      <xsd:element ref="gisp:pid" minOccurs="0" />
      <xsd:element ref="gisp:limit" minOccurs="0" />
     </xsd:sequence>
    </xsd:extension>
   </xsd:complexContent>
  </xsd:complexType>
</xsd:element>

<xsd:element name="limit">
 <xsd:annotation>
  <xsd:documentation>
   Limit number for searching peer
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:simpleContent>
   <xsd:extension base="xsd:integer">
    <xsd:attributeGroup ref="gisp:extensibleAttributes" />
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
</xsd:element>

<xsd:element name="insert">
 <xsd:annotation>
  <xsd:documentation>
   A pair data
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:complexContent>
   <xsd:extension base="gisp:extensibleType">
    <xsd:sequence>
     <xsd:element ref="gisp:pid" />
     <xsd:element ref="gisp:xml" />
     <xsd:element ref="gisp:ttl" />
     <xsd:element ref="gisp:sent" minOccurs="0"
           maxOccurs="unbounded" />
    </xsd:sequence>
   </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
</xsd:element>

<xsd:element name="xml">
 <xsd:annotation>
  <xsd:documentation>
   XML data of the pair
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:sequence>
   <xsd:any namespace="##any" />
  </xsd:sequence>
  <xsd:attributeGroup ref="gisp:extensibleAttributes" />
 </xsd:complexType>
</xsd:element>

<xsd:element name="query">
 <xsd:annotation>
  <xsd:documentation>
   Query for pair
  </xsd:documentation>
 </xsd:annotation>
 <xsd:complexType>
  <xsd:complexContent>
   <xsd:extension base="gisp:extensibleType">
    <xsd:sequence>
     <xsd:element ref="gisp:pid" />
     <xsd:element ref="gisp:qstr"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" />
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
```

```
</xsd:element>

 <xsd:element name="qstr">
  <xsd:annotation>
   <xsd:documentation>
    XPath of the query
   </xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
   <xsd:complexContent>
    <xsd:extension base="gisp:extensibleType">
     <xsd:attribute name="type" type="xsd:string" use="required" />
    </xsd:extension>
   </xsd:complexContent>
  </xsd:complexType>
 </xsd:element>

 <xsd:element name="result">
  <xsd:annotation>
   <xsd:documentation>
    Result of query
   </xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
   <xsd:complexContent>
    <xsd:extension base="gisp:extensibleType">
     <xsd:sequence>
      <xsd:choice maxOccurs="unbounded">
       <xsd:element ref="gisp:xml" />
       <xsd:element ref="gisp:peer" />
     </xsd:choice>
     </xsd:sequence>
     <xsd:attribute name="query" type="xsd:string" />
    </xsd:extension>
   </xsd:complexContent>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

ภาคผนวก ข

บทความที่ได้รับการตีพิมพ์

Integrating Peer-to-Peer File Sharing into the Grid

Kittisak Thanomtheeranant and Veera Maungsin
Scientific Parallel Computer Engineering Lab
Department of Computer Engineering, Faculty of Engineering,
Chulalongkorn University, Bangkok 10330, Thailand
kittisak.tha@chula.ac.th and veera.m@chula.ac.th

Abstract

Grid and peer-to-peer technology are two different approaches with the same goal to create large-scale distributed computing systems whose data sharing is one of the most important functionalities. While most data grid implementations concern with sharing large volumes of data stored at servers, peer-to-peer file sharing systems mostly deal with a large number of relatively small files stored in PCs at the edge of the internet. With currently available grid technology, home and office PCs are not to be connected to the grid. Therefore, a huge amount of aggregated storage that is potentially available via peer-to-peer file sharing is not yet accessible to the grid.

This paper proposes an integration of data grid and peer-to-peer file sharing. The objective is to gather free disk space on computers that are not connected to the grid and make it available to grid applications. The integrated system exploits gateway nodes that are members of both grid and peer-to-peer systems. The current design is asymmetric such that the peer-to-peer file sharing system is treated as a grid resource. Grid users and applications can put and get files on the virtual storage in the peer-to-peer network. The virtual storage consists of directory-based shared disk space on peer nodes. The system is implemented on top of Globus and JXTA as grid and peer-to-peer middleware respectively. The peer-to-peer file sharing system is modified from Jnushare, that uses GISP (Global Information Sharing Protocol) for its scalable indexing mechanism based on distributed hash table.

1 Introduction

Grid and peer-to-peer technology are two different approaches with the same goal to create large-scale distributed computing systems whose data sharing is one of the most important functionalities. Grid technology aims to create large-scale distributed computing systems consisting of a large set of computing resources that lie across geographical locations and administrative domains. The grid is created for solving hard problems that need extensive computing power or a massive amount of data, or both. Therefore, grid resources are usually high-performance computers and large file servers. The terms "computational grid" and "data grid" are used to emphasis the purposes of grid projects. Here we focus on the data grid.

Peer-to-peer technology nowadays targets at personal computers and devices that arbitrarily connect to the internet. The most popular peer-to-peer application is file sharing. There are many widely-used file sharing programs including Napster, Gnutella, and KaZaa. Peer-to-peer computing is also a very interesting approach for performing large-scale distributed computing. The best known example is Seti@home (analyzing radio signal from the sky) and GIMPS (finding the largest prime number). Due to the widespread use and fast growing performance and storage capacity of personal computers, peer-to-peer network is potentially the largest pool of computing power and data storage.

However, with currently available grid technology, home and office PCs are not to be connected to the grid. Therefore, a huge amount of aggregated storage that is potentially available via peer-to-peer file sharing is not yet accessible to the grid.

This paper proposes an integration of peer-to-peer file sharing into data grid. The objective is to gather free disk space on computers that are not connected to the grid and make it accessible to grid users and applications. The basic idea is to have some gateway nodes that are members of both grid and peer-to-peer network. The peer-to-peer file sharing system is treated as a grid resource. Grid users and applications can put and get files on the virtual storage in the peer-to-peer network.

This paper is organized as follow. Section 2 gives some background information. The design and implementation are described in section 3. Conclusion and future work are discussed in section 4.

2. Background

2.1 Data Grid with Globus

A grid is a large-scale geographically distributed hardware and software infrastructure composed of heterogeneous networked resources owned and shared by multiple administrative organizations which are coordinated to provide transparent and pervasive computing support to a wide range of applications [1, 2]. In grid terminology, each machine connected to the grid is called a "host" and a group of hosts can form a "virtual organization". A grid computing system that deals with large amounts of distributed data is called a "data grid" [3].

Globus Toolkit [4, 5] is the de facto standard grid middleware. It consists of a set of components that implement basic services, such as security, resource allocation, resource management, data management, and communications. To support data grid, Globus Toolkit provides components for data transfer and data replication [6].

The most commonly used grid data transfer protocol is GridFTP [7], which is provided by Globus Toolkit. GridFTP is a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks. The GridFTP protocol is based on FTP. Globus Toolkit provides a server implementation called *globus-gridftp-server*, a command line client called *globus-url-copy*, and a set of development libraries for custom clients. The globus-url-copy program supports a wide range of protocols including http, https, ftp, gsiftp, and local file access.

2.2 Peer-to-Peer File Sharing with JXTA

Peer-to-peer (P2P) architecture is a class of systems and applications that employ distributed resources to perform some function in a decentralized manner. The resources encompass computing power, data (storage and content), network bandwidth, and other services. The function of a peer-to-peer system can be distributed computing, data/content sharing, communication and services. Although decentralization of resources, algorithms, data, and meta-data is the ideal concept for such systems, retaining centralization in some parts of the systems and applications is often found in existing implementations. Typical P2P systems reside at the edge of the Internet or in ad-hoc networks [8]. Each entity is called a "peer" and can act as server and client. Many peers can form a "peer group".

JXTA is an open peer-to-peer computing platform developed by Sun Microsystems. It provides a set of protocols, libraries and services as building blocks for developing various peer-to-peer applications on peer groups. The JXTA protocols standardize how peers discover each other, self-organize into peer groups, advertise and discover services, communicate and monitor each other, etc. JXTA is independent of programming languages, and independent of transport protocols. The protocols can be implemented in Java, C/C++, Perl, and other languages. They can be implemented on top of TCP/IP, HTTP, Bluetooth, and other transport protocols [9]. JXTA has an extensive use of "advertisements" for announcing and discovering peers, peer-groups, and services.

Jnushare [10] is a data sharing application that employs JXTA's Content Management System (CMS) for uploading and downloading data content and uses Global Information Sharing Protocol (GISP) [10] for searching meta-data. GISP is a protocol for building and searching a distributed hash table (DHT). A DHT consists of (key, value) pairs. For reliability, each pair data can be replicated and distributed onto several peers depending on a hash function that is applied on the key. Therefore, each peer is responsible for a fraction of the hash table that can be overlap with other peers. For a file sharing application, the key can be the logical path name of the file and the value can be the peer identification and location where the file is. When a file is shared, a new entry of DHT is created and distributed on several peers. When the file is searched for, the hash function is used to find a peer that keeps the table entry of that file. After retrieving the table entry, the peer identification and location of the file is then used for retrieving the file content.

3. System Design and Implementation

3.1 Design Decisions

In order to integrate peer-to-peer network and grid, their differences must be taken into account. Firstly, while most data grid implementations concern with sharing large volumes of data stored in servers, peer-to-peer file sharing systems mostly deal with a large number of relatively small files stored in PCs. The number of peers in the peer-to-peer network can also be much larger than hosts in the grid. The presence of these peers can be arbitrary and thus the availability of their resources is very dynamic.

Therefore, three design decisions have been made from the start. Firstly, individual peers will not be accessed directly from the grid. In the peer-to-peer network, each individual peer is one resource and can be access directly by other peers. However, making every single peer accessible to grid hosts is not very useful. We therefore adopt the idea of cluster computing in which only the master node can directly communicate outside the cluster. A peer-group is analogous to a cluster where most peers are like compute nodes. Some peer can be chosen to act like a master node, or a "super-peer". A peer-group can provide some collective services, such as providing a virtual data storage. Therefore, a peer-to-peer file sharing system can act like a single host on the data grid.

We have also decided that, for now, the integration between peer-to-peer file sharing and data grid is asymmetric. It means that a peer-group acts as a grid resource and its service is available to all grid hosts. However, a grid host is not a peer in the peer-to-peer network.

The third decision is that the integration must be transparent. Grid users and applications can put and get files on the peer-to-peer virtual storage just like other grid hosts. It should not introduce new concepts or new commands. Instead, regular GridFTP commands will be supported.

3.2 System Architecture

The integrated system consists of three types of entities, namely grid hosts, peers, and gateways. Grid hosts are where users and applications work on. Peers join together as a peer-group. These peers run a file sharing program and make some disk space available to the file sharing network, creating a virtual storage. Gateways are members of the grid and the peer-group at the same time. In other words, they run both Globus and JXTA. A gateway provides an access point from grid hosts to the peer-to-peer virtual storage. Figure 1 depicts the abstract model.

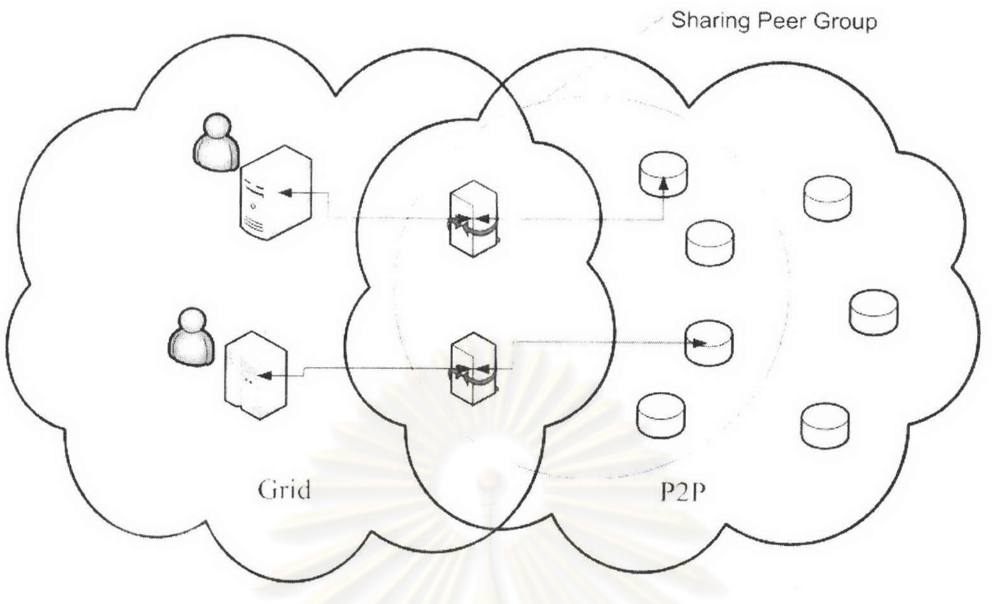

Fig. 1. The abstract model of data grid and peer-to-peer file sharing integration

3.3 Peer-to-Peer Virtual Storage Mechanism

When a grid user or application wants to put a file on the virtual storage, the file is transferred to the gateway using the grid file transfer protocol. The gateway then puts the file into a shared directory using our modified version of the Jnushare peer-to-peer file sharing system. Normally Jnushare only distributes the hash table, but in our system the file owner is the gateway and the peers that are responsible for the table entry will also automatically get the file from the gateway. A gateway stores advertisements of the files on the virtual storage. Therefore, a listing of the files can be acquired from the gateway.

When a grid application subsequently acquires the file from the virtual storage, the gateway retrieves the file from one of the peers that keep the file and sends it back to the grid application.

After transferring the file to other peers, the gateway can remove the file from its storage. However, the gateway can keep the file in its cache storage for future access. It can remove the least recently used file when the space is required for a new file. The gateway with large cache storage can reduce communication with other peers and thus the respond time for accessing frequently used files.

When a peer disconnects from the peer-group, other peers can be chosen, based on the hash function again, to be responsible for the files previously are on that peer. The number of peers that keep a copy of a file affects the reliability and performance of the system. The optimal number depends on many factors such as the size of the file and the characteristics of a particular peer-to-peer network including dynamicity, bandwidth, and available storage space.

Figure 2 depicts the interaction between grid and peer-to-peer components. A grid host acting as a client performs GridFTP operations with the gateway which is also a grid host. The gateway provides GridFTP service to other grid hosts. If the operation involves the peer-to-peer virtual storage, the gateway acts as a peer and translates the operation into a peer-to-peer file sharing operation via the JXTA protocol. When the peer-to-peer operation is completed, the result is returned to the grid client via the GridFTP protocol.

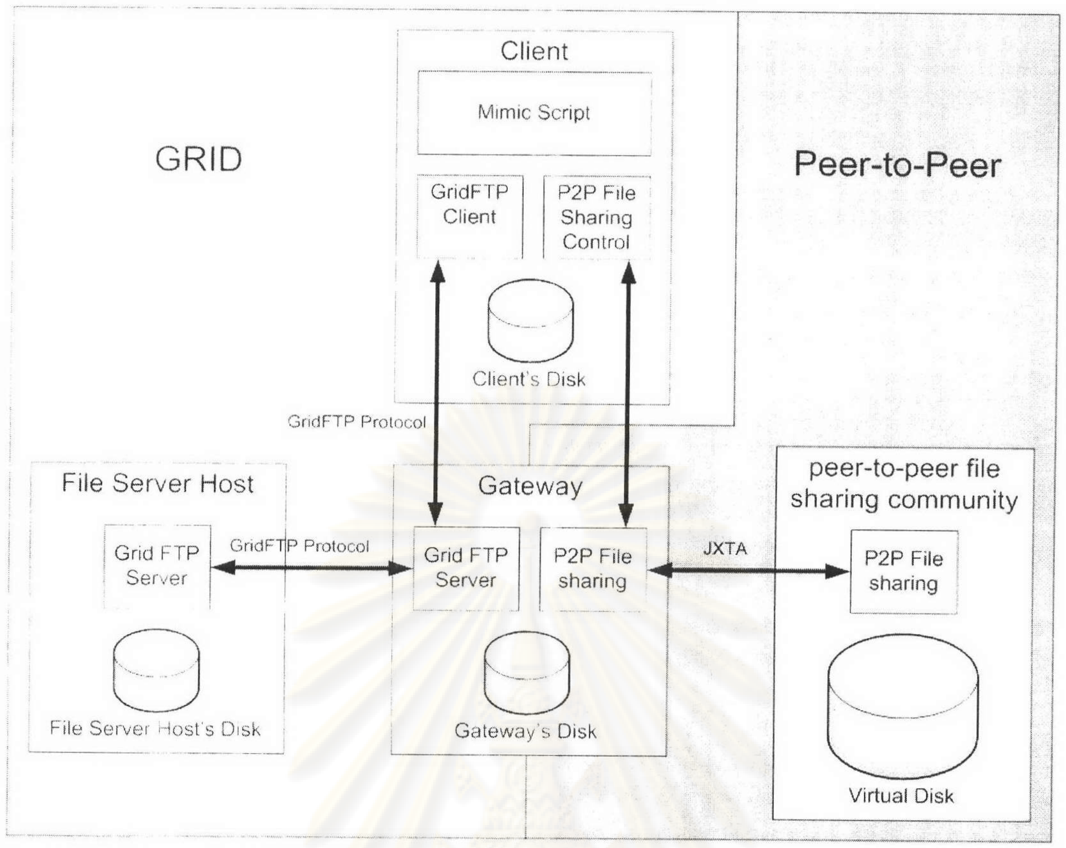

Fig. 2. The interaction in grid and peer-to-peer virtual storage system integration

In order to make the peer-to-peer virtual storage easily accessible to grid users and applications. we create a script on top of the GridFTP client command, *globus-url-copy*. The new script has the same name for convenience. The purpose is to make globus-url-copy support a new protocol called *"p2p"*, in addition to "http", "ftp", "file", etc. Therefore, if the URL of the source file in the file transfer command begins with *"p2p://"*, the script will send a command to the gateway to get the file from the peer-to-peer virtual storage. When the file is available on the gateway local storage, it will be transferred to the destination with the regular GridFTP operation.

If the URL of the destination in the file transfer command begins with "p2p://", the file will be transferred to the gateway's local storage with the regular GridFTP operation. When the file is available on the gateway local storage, it will be put in the peer-to-peer virtual storage with the previously explained mechanism.

4. Conclusions and Future Work

This paper proposes a model of a peer-group that provides a collective service, similar to the concept of cluster computing, that can be integrated into the grid. The model is applied to a peer-to-peer virtual storage system and it is successfully implemented based on Globus, GridFTP, JXTA, GISP, and Jnushare.

There is a lot of potential for future work on the system. The first one is to support a file system structure with better access control. Secondly, the DHT mechanism to select peers to handle files should also take into account the available storage space and network availability. Finally, some mechanism is needed to prevent the gateway becoming a bottleneck.

5. References

[1] I. Foster. C. Kesselman (eds.), *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann, San Francisco, USA (1998).

[2] I. Foster. C. Kesselman, S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, International Journal of Supercomputer Applications, (2001), pp 200-222.

[3] A. Chervenak. I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, *The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets*, Journal of Network and Computer Applications, 23:187-200, 2001.

[4] I. Foster. C. Kesselman, *Globus: a Metacomputing Infrastructure Toolkit*, International Journal of Supercomputer Applications, (1997), pp 115-128.

[5] I. Foster. C. Kesselman, *The Globus Project: a Status Report*, Proceedings of IPPS/SPDP'98 Heterogeneous Computing Workshop (1998), pp 4-18.

[6] W. Allcock. J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, S. Tuecke, *Data Management and Transfer in HighPerformance Computational Grid Environments*, Parallel Computing, 2001.

[7] G. Cawood, *GridFTP Report*, Sun Data and Compute Grids, epcc, (2003).

[8] D. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, Z. Xu. *Peer-to-Peer Computing*, HP Labs Technical Report HPL-2002-57, (2002).

[9] Sun Microsystems, *JXTA v2.3.x: Java Programmer's Guide*.

[10] K. Daishi. *GISP: Global Information Sharing Protocol - A Distributed Index for Peer-to-Peer Systems*, Proceedings of 2nd International Conference on Peer-to-Peer Computing (P2P'02), (2002), pp 65-72.

ประวัติผู้เขียนวิทยานิพนธ์

นายกิตติศักดิ์ ถนอมธีระนันท์ สำเร็จการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อวันที่ 5 เมษายน พ.ศ. 2545 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2545

งานวิจัยที่ได้รับการตีพิมพ์ในการประชุมวิชาการมีด้วยกัน 1 ชิ้น โดยบทความต่างๆ ได้แนบไว้หลังวิทยานิพนธ์นี้แล้ว

1. Thanomtheeranant, K., and Maungsin, V., "Integrating Peer-to-Peer File Sharing into the Grid", 10th Annual National Symposium on Computational Science & Engineering (ANSCSE10), Thailand, 22-24 March 2006.