# Resource and Service Discovery in the iGrid Information Service

Giovanni Aloisio, Massimo Cafaro, Italo Epicoco, Sandro Fiore, Daniele Lezzi, Maria Mirto, and Silvia Mocavero

Center for Advanced Computational Technologies,
University of Lecce/ISUFI, Italy
{giovanni.aloisio, massimo.cafaro, italo.epicoco, sandro.fiore, daniele.lezzi, maria.mirto, silvia.mocavero}@unile.it

**Abstract.** In this paper we describe resource and service discovery mechanisms available in iGrid, a novel Grid Information Service based on the relational model. iGrid is developed within the GridLab project by the ISUFI Center for Advanced Computational Technologies (CACT) at the University of Lecce, Italy and it is deployed on the European GridLab testbed. The GridLab Information Service provides fast and secure access to both static and dynamic information through a GSI enabled web service. Besides publishing system information, iGrid also allow publication of user's or service supplied information. The adoption of the relational model provides a flexible model for data, and the hierarchical distributed architecture provides scalability and fault tolerance.

## 1 Introduction

The grid computing paradigm is widely regarded as a new field, distinguished from traditional distributed computing because of its main focus on large-scale resource sharing and innovative high-performance applications [1]. The grid infrastructure can be seen as an ensemble of Virtual Organizations (VOs), reflecting dynamic collections of individuals, institutions and computational resources [2].

In this context, achieving flexible, secure and coordinated resource sharing among participating VOs requires the availability of an information rich environment to support resource and service discovery, and thus decision making processes. Indeed, we think of distributed computational resources, services and VOs as sources and/or potential sinks of information. The data produced can be static or dynamic in nature, or even dynamic to some extent. Depending on the actual degree of dynamism, information is better handled by a Grid Information Service (static or quasi-static information) or by a Monitoring Service (highly dynamic information).

In this context, information plays a key role, therefore in turn Grid Information Services are a fundamental building block of grid infrastructure/middleware. Indeed, high performance execution in grid environments is virtually impossible

O. Gervasi et al. (Eds.): ICCSA 2005, LNCS 3482, pp. 1–9, 2005.

without timely access to accurate and up-to-date information related to distributed resources and services. Our experience has shown that using manual of default configurations hinders application performance, and the lack of information about the execution environment prevents design and implementation of grid-aware applications. As a matter of fact, an application can react to changes in its environment only if these changes are advertised. Therefore, self-adjusting, adaptive applications are natural consumers of information produced in grid environments. Neverthless, making relevant information available on-demand to consumer applications is actually nontrivial, since information can be (i) diverse in scope, (ii) dynamic and (iii) distributed across one or more VOs. Moreover, obtaining information about the structure and state of grid resources, services, networks etc. can be challenging in large scale grid environments.

The rest of the paper is organized as follows. We recall related work in Section 2, and discuss resource and service discovery mechanisms available in iGrid in Section 3. Finally, we conclude the paper in Section 4.

## 2 Related Work

### 2.1 Resource Discovery

Historically, flooding protocols [3] have been used for resource discovery in networks; later, gossiping protocols such as [4] have been shown to be more efficient for information dissemination. Recently, the problem of efficient resource and service discovery received a great deal of attention. Essentially, the current focus is on structured and/or unstructured peer-to-peer systems. Iamnitchi and Foster [5] propose some heuristic solutions for decentralized distributed resource discovery, and define a taxonomy based on membership protocol, overlay construction, preprocessing and request processing. Butt et al. [6] describe the use of Condor pools coupled with the Pastry [7] distributed hash table algorithm, and utilize the peer-to-peer overlay network to attain scalable resource discovery.

Other approaches are based on the notion of non uniform information dissemination, such as the work of Iyengar et al. [8]. The key idea is to update replicated information services using non uniform instead of full dissemination of information. This leads to reduced information replication overhead, maintaining anyway accurate information at locations where it is most likely to be needed. The authors observe that grid resources share some properties of sensor networks, and thus tend to be of more interest to nearby users, because the overhead in starting a job and transferring related data and results increases with the distance to the resources. The dissemination protocols proposed work by propagating resource state information more aggressively and in more detail to nearer information repositories than they do to farther ones. This leads to the Change Sensitive Protocol and Prioritized Dissemination Protocol. Maheswaran et. al. [9] share the same concept, introducing in their work the notion of grid potential, which is used to weight a grid resource capability with its distance from the application launch point.

The majority of the works deals with variation of distributed hash table algorithms for building scalable infrastructures. We recall here CAN [10], Chord [12], Kademlia [13], Tapestry [11], Viceroy [14]. The CONE data structure by Bhagwan et al. [15], and the peer-to-peer information retrieval system designed by Schmidt et al. [16] using Hilbert Space Filling Curve and the Chord overlay network topology, specifically address distributed resource discovery. We conclude this subsection citing the Globus Toolkit Metadata Directory Service (MDS v2) [17], an LDAP based approach characterized by mechanisms and protocols for storing information and building scalable distributed collections of servers. No specific organization strategies, overlay topologies, or dissemination protocols have been specified. This service has been deployed and used on many grid projects, both for production and development. However, the performances of the MDS Information Service were not satisfactory enough, and the Globus project provided users with a new version of this service, available in the Globus Toolkit version 3.x and called Monitoring and Discovery Service (again, MDS v3). Version 3.x of the Globus Toolkit has been based on the Open Grid Service Infrastructure (OGSI) and the Open Grid Service Architecture [19] specifications. The MDS has been developed using Java and the Grid Services framework [18], so that, again, its performances are not satisfactory enough. The MDS service will change once again in the Toolkit version 4, and it will be based on the emerging Web Service Resource Framework (WSRF) specification [20].

It is important to understand that while many versions of the Globus Toolkit exist, only version 2.x and version 3.x pre-OGSI have been widely deployed for use in production grids and testbeds. Indeed, version 3.x based on OGSI has been targeted only by developers, since this version was known to be replaced next year by the new WRSF based Toolkit, which will also offer the possibility to develop grid services and clients using C/C++ besides Java.

The GridLab project started adopting and extending initially the Globus MDS [32], but we decided to move from LDAP to the relational model, in order to overcome the Globus MDS shortcomings: a data model better suited for frequent reading, not able to cope with frequent updates, a weak query language missing key operations on data (e.g., there is no *join*), and performances. The relational data model allows us to model information and store data in tabular form, relationships among tables are possible, typical relational DBMS are strongly optimized for both fast reading and writing, and finally a good query language, namely SQL, allows complex operations on data, including joins.

### 2.2 Service Discovery

Historically, service discovery has been initially addressed by the ANSA project [21], an early industry effort to define trading services to advertise and discover relevant services. As an evolution, we recall here the CORBA trading service [22]. Recently, yet another industry standard is emerging in the context of web services, UDDI [23]. This is a centralized registry storing WSDL description of business oriented web services. A grid oriented approach, called Web Services Inspection Language (WSIL) [24] has been proposed by IBM specifically for grid services, as envisioned by the OGSA architecture.

The Web Service Discovery Architecture (WSDA) proposed by Hoschek [25], specifies communication primitives useful for discovery, service identification, service description retrieval, data publication as well as query support. The individual primitives when combined and plugged together by clients and services yield a wide range of behaviors. The author also introduces an hyper registry, a centralized database node for discovery of dynamic distributed content. This registry supports XQueries over a tuple set from a dynamic XML data model. The architecture includes a Unified Peer-to-Peer Database Framework (UPDF) and a corresponding Peer Database Protocol (PDP).

The Jini Lookup Service [26] is a centralized service approach which exploits multicast and soft state protocols. However, it is worth noting here that the protocols rely on Java object serialization, therefore the service usefulness is limited to Java clients. The Java JXTA peer-to-peer network is a notable effort; we recall here its stateless, best-effort protocols for ad hoc, pervasive, and multi-hop P2P computing. JXTA includes, among others, the following protocols. The Endpoint Routing Protocol allows discovering routes from one peer to another peer, given the destination peer ID. The Rendezvous Protocol provides publish and subscribe functionalities within a peer group. The Peer Resolver Protocol and Peer Discovery Protocol allow publishing advertisements and simple queries that are unreliable, stateless, non-pipelined, and non-transactional.

The Service Location Protocol (SLP) [27] can be used to advertise and query the location, type and attributes of a given service; it uses multicast and soft state protocols. A recent extension, called Mesh Enhanced Service Location Protocol (mSLP) [28], increases scalability through the use of multiple cooperating directory agents. The Service Discovery Service (SDS) [29] supports XML based exact match queries, and is based on multicast and soft state protocols too.

## 3 Resource and Service Discovery in iGrid

iGrid is a novel Grid Information Service developed within the European GridLab project [30] by the ISUFI Center for Advanced Computational Technologies (CACT) at the University of Lecce, Italy. An overview of the iGrid Information Service can be found in [33]; here we delve into details related specifically to resource and service discovery mechanisms available.

iGrid distributed architecture is shown in Fig. 1. The architecture is based on iServe and iStore GSI [31] enabled web services. An iServe collects information related to the computational resource it is installed on, while iStore gathers information coming from trusted, registered iServes. The current architecture resembles the one adopted by the Globus Toolkit MDS, therefore iStores are allowed to register themselves to other iStores, creating arbitrarily complex distributed hierarchies. Even though this architecture proved to be effective to build scalable distributed collections of servers, neverthless we are already investigating peer-to-peer overlay networks based on current state of the art distributed hash table algorithms in order to improve iGrid scalability. The implementation includes system information providers outputting XML, while trusted users

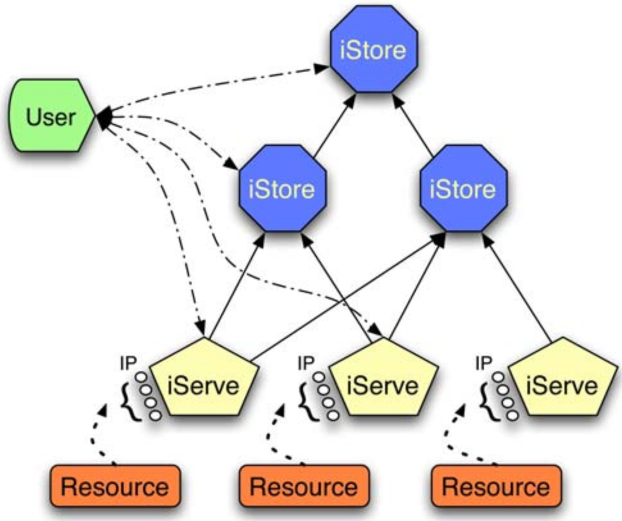


**Fig. 1.** iGrid hierarchical architecture

and/or services can publish information simply calling a web service registration method.

Resource discovery using the iGrid Information Service is based on the availability of the following information (not exaustive):

**System** operating system, release version, machine architecture etc;

**CPU** for CPUs, static information such as model, vendor, version, clock speed is extracted; the system also provides dynamic information such as idle time, nice time, user time, system time and load;

**Memory** static information such as RAM amount and swap space is available. Dynamic information related to available memory and swap space is published too;

**File Systems** static as well dynamic information is extracted, such as file system type, mount point, access rights, size and available space;

**Network Interfaces** network interface names, network addresses and network masks;

**Local Resource Manager** the information belonging to this category can be further classified as belonging to three different subclasses: information about queues, jobs and static information about Local resource Management System (LRMS). Some examples of extracted information are: LRMS type and name; queue name and status, number of CPU assigned to the queue, maximum number of jobs that can be queued, number of queued jobs, etc; job name, identifier, owner, status, submission time etc. Currently information providers for OpenPBS and Globus Gatekepeer are available, with LSF planned;

**Certification Authorities** certificate subject name, serial number, expiration date, issuer, public key algorithm etc.

**Virtual Organization** information related to VO can be used to automatically discover which resources belong to a given VO; we have VO name, resource type, help desk phone number, help desk URL, job manager, etc.

Of course, this set of information is not meant to be static, the iGrid schema will continue to evolve and will be extended to support additional information as required by the GridLab project or iGrid users.

One of the most important requirements for grid computing scenarios is the ability to discover services and web/grid services dynamically. Services in this context refers to traditional unix servers. The iGrid system provides users and developers with the following functionalities: register, unregister, update and lookup. More than one instance for each service or web service can be registered. The following information is available for services: logical name, instance name, service description, default port, access URL, distinguished name of the service publisher, timestamps related to date of creation an date of expiration of the published information.

For web services, relevant information includes logical name, web service description, WSDL location (URL), web service access URL, distinguished name of publisher and timestamps related to date of creation and date of expiration of the published information.

Information related to firewalls is strictly related to service information. As a matter of fact, before registering a service, developers will query iGrid to retrieve the range of open ports available on a specified computational resource. This is required in order to chose an open port, allowing other people/services to connect to a registered service. The information available includes firewall hostname, open ports, time frame during which each port (or a range of ports) is open, the protocol (TCP/UDP) used to connect to these ports, the distinguished name of the firewall administrator, and timestamps related to date of creation and date of expiration of the published information.

iGrid uses a push model for data exchange. Indeed, system information (useful for resource discovery) extracted from resources is stored on the local database, and periodically sent to registered iStores, while user and/or service supplied information (useful for service discovery) is stored on the local database and immediately sent to registered iStores. Thus, an iStore has always fresh, updated information related to services, and almost fresh information related to resources; it does not need to ask iServes for information. The frequency of system information forwarding is based on the information itself, but we also allow defining a per information specific policy. Currently, system information forwarding is based on the rate of change of the information itself. As an example, information that does not change frequently or change slowly (e.g. the amount of RAM installed) does not require a narrow update interval. Interestingly, this is true even for the opposite extreme, i.e., for information changing rapidly (e.g., CPU load), since it is extremely unlikely that continuous forwarding of this kind of information can be valuable for users, due to information becoming quickly inaccurate. Finally, information whose rate of change is moderate is forwarded using narrow update intervals. Allowing user's defined policies provides a forwarding protocol similar to the Prioritized Dissemination Protocol, whilst our update frequencies closely resemble the way the Change Sensitive Protocol works.

We have found that the push model works much better than the corresponding pull model (adopted, for instance, by the Globus Toolkit MDS) in grid environments. This is due to the small network traffic volume generated from iServe to iStore servers: on average, no more than one kilobyte of data must be sent. Moreover, we tag information with a time to live attribute that allows iGrid to safely removes stale information from the database when needed. For instance, when users search for data, a clean-up operation is performed before returning to the client the requested information, and during iGrid system startup, the entire database is cleaned up. Therefore the user will never see stale information.

Finally, it is worth recalling here that the performances of iGrid are extremely good, as reported in [33] .

## 4 Conclusion

We have described iGrid, a novel Grid Information Service based on the relational model, and its mechanisms for resource and service discovery. The GridLab Information Service provides fast and secure access to both static and dynamic information through a GSI enabled web service. Besides publishing system information, iGrid also allows publication of user's or service supplied information. The adoption of the relational model provides a flexible model for data, and the hierarchical distributed architecture provides scalability and fault tolerance. The software, which is open source, is freely available and can be downloaded from the GridLab project web site [34].

## Acknowledgements

We gratefully acknowledge support of the European Commission 5th Framework program, grant IST-2001-32133, which is the primary source of funding for the GridLab project.

## References

1. Foster, I., Kesselman C.: The Grid: Blueprint for a new Computing Infrastructure, Morgan Kaufmann (1998)
2. Foster, I., Kesselmann, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Intl. Journal of High Performance Computing Applications **15**(3) (2001) 200–222. http://www.globus.org/- research/papers/anatomy.pdf.
3. Harcol-Balter, M., Leighton, P., and Lewin, D.: Resource discovery in distributed networks. In Proc. of ACM Symp. on Principles of Distributed Computing (1999) 229–238.

4. Li, L., Halpern, J., and Haas, Z.: Gossip-based ad hoc routing. In IEEE Infocom (2002) 1707–1716.
5. Iamnitchi, A., and Foster. I.: A Peer-to-Peer Approach to Resource Location in Grid Environments. Proceedings of the 11th Symposium on High Performance Distributed Computing (HPDC'02) (2002) 419.
6. Butt, A.R., Zhang, R., and Hu, Y.C.: A Self-Organizing Flock of Condors. In Proceedings of IEEE/ACM SC2003 (2003).
7. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware) (2001) 329–350. http://freepastry.rice.edu
8. Iyengar, V., Tilak, S., Lewis, M.J., and Abu-Ghazaleh, N.B.: Non-Uniform Information Dissemination for Dynamic Grid Resource Discovery. IEEE NCA 2004 (2004) 97–106.
9. Maheswaran, M., and Krauter, K.: A parameter-based approach to resource discovery in grid computing system. In Proceedings of the First IEEE/ACM International Workshop on Grid Computing (2000) 181–190.
10. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. in Proceedings of ACM SIGCOMM MANCA VOLUME(2001) PAGINE.
11. Hildrum, K., Kubiatowicz, J., Rao, S., Zhao, B.: Distributed Object Location in a Dynamic Network. In Proceedings of 14th ACM Symp. on Parallel Algorithms and Architectures (SPAA'02) (2002) 41–52.
12. Stoica, I. et al: Chord: A scalable peer-to-peer lookup service for Internet applications. In Proceedings of ACM SIGCOMM (2001) 149–160. http://www.pdos.lcs.mit.edu/chord.
13. Maymounkov, P., Mazieres, D.: Kademlia: A peer-to-peer information system based on the XOR metric. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems, Springer-Verlag (2002). http://kademia.scs.cs.nyu.edu/
14. Malkhi, D., Naor, M., Ratajczak, D.: Viceroy: A scalable and dynamic emulation of the butterfly. In Proceedings of ACM Principles of Distributed Computing(PODC'02) (2002) 183–192.
15. Bhagwan, R., Varghese, G., and Voelker, G.M.: Cone: Augmenting DHTs to Support Distributed Resource Discovery. In Proceedings of CS2003 (2003).
16. Schmidt, C., and Parashar, M.: Flexible Information Discovery in Decentralized Distributed Systems. In Proceedings of HPDC (2003) 226–235.
17. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing, IEEE Press (2001) 181.
18. Foster, I., Kesselmann, C., Nick, J., Tuecke, S.: Grid Services for Distributed System Integration. Computer. **35**(6) (2002) 37–46.
19. Foster, I., Kesselmann, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed System Integration. Technical Report for the Globus project. http://www.globus.org/research/papers/ogsa.pdf
20. The WSRF specification. http://www.globus.org/wsrf/specs/ws-wsrf.pdf
21. Beitz, A., Bearman, M., and Vogel, A.: Service Location in an Open Distributed Environment. In Proc. of the 2nd International Workshop on Services in Distributed and Networked Environments (1995) 28.
22. Object Management Group. Trading Object Service. OMG RPF5 Submission, May 1996.

23. UDDI Consortium. UDDI: Universal Description, Discovery and Integration. http://www.uddi.org.
24. IBM. Web Services Inspection Language (WS-Inspection) 1.0 specification. http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html.
25. Hoschek, W.: Peer-to-Peer Grid Databases for Web Service Discovery. Concurrency: Pract. Exper. (2002) 1–7.
26. Waldo, J.: The Jini architecture for network-centric computing. Communications of the ACM **42**(7) (1999) 76–82.
27. Guttman, E.: Service Location Protocol: Automatic Discovery of IP Network Services. IEEE Internet Computing Journal **3**(4) (1999) 71–80.
28. Zhao, W., Schulzrinne, H., and Guttman, E.: mSLP - Mesh Enhanced Service Location Protocol. In Proc. of the IEEE Int. Conf. on Computer Communications and Networks (ICCCN'00) (2000).
29. Czerwinski, S.E., Zhao, B.Y., Hodes, T., Joseph, A.D., and Katz, R.: An Architecture for a Secure Service Discovery Service. In Fifth Annual Int. Conf. on Mobile Computing and Networks (MobiCOM'99) (1999) 24–35.
30. The GridLab project. http://www.gridlab.org
31. Foster, I., Kesselmann, C., Tsudik G., Tuecke, S.: A security Architecture for Computational Grids. Proceedings of 5th ACM Conference on Computer and Communications Security Conference (1998) 83–92.
32. Aloisio G., Cafaro M., Epicoco I., Lezzi D., Mirto M., Mocavero S.: The Design and Implementation of the GridLab Information Service. Proceedings of The Second International Workshop on Grid and Cooperative Computing (GCC 2003)Lecture Notes in Computer Science, Springer-Verlag **3032** (2004) 131–138.
33. Aloisio, G., Cafaro, M., Epicoco, I., Fiore, S., Lezzi, D., Mirto, M. and Mocavero, S.: iGrid, a Novel Grid Information Service. To appear in Proceedings of the first European Grid Conference (EGC) 2005.
34. The iGrid Web Site. http://www.gridlab.org/WorkPackages/wp-10.