Resource Discovery Infrastructure

Wolfgang Hoschek, Karlo Berket and Deb Agarwal
Lawrence Berkeley National Laboratory

Abstract

For large distributed multi-disciplinary science collaborations it is a difficult problem to find data, results, and resources. Typically, defining centralized storage and managing the resources as a single domain solves this problem. But this solution scales poorly and does not allow for opportunistic use of resources and data repositories. As a much more scalable alternative, we propose to research, design, develop, evaluate and benchmark a reusable software infrastructure for Peer-to-Peer resource discovery, thereby enabling a range of innovative research directions building on it. The infrastructure will address a number of scalability problems in a general way. It will provide flexible and uniform transport-independent resource discovery mechanisms to reduce both the client and network burden in multi-hop P2P systems. We hope that this research and resulting infrastructure will enable new types of DOE science applications.

Further future work will focus on enabling heterogeneous P2P networks of services for science applications, integrating many distinct infrastructures and application components. For example, such heterogeneous P2P networks provide an innovative and reusable productivity tool suite by integrating collaborative P2P chat, service discovery, databases, network monitoring, data sharing tools as well as end-to-end security mechanisms.

Motivation

Modern scientific research is conducted by large distributed multi-disciplinary teams that are cooperating to conduct experiments, simulations, and achieve results. Such collaborations are often globally distributed and multi-institutional. A difficult problem for these teams is often finding and organizing data, results, and resources. Distributed applications used in modern science are characterized by large scale, heterogeneity, lack of central control, multiple autonomous administrative domains, unreliable components and frequent dynamic change. In such applications it is desirable to maintain and query dynamic information about active participants such as services, resources and user communities in a timely manner. Peer-to-Peer file sharing systems, instant messaging services, monitoring infrastructures, resource brokers, job schedulers and flexible bootstrap configuration systems all share these requirements.

For example, physics science users would like to easily share information such as Grid input and output files with collaborators, independent of the churn rate of the collaboration. For good decision-making, a resource broker service would like to find the available job execution services, their capacity and utilization. Further, it needs to find the storage services that have given input files, the storage services that can store output files for the user, as well as find the access control policies and bandwidth between all involved services.

Administrators and performance analysts would like to find and display the average latency and bandwidth between all or a subset of Grid nodes over the last 24-hour period. In case of an epidemic or environmental disaster, a future emergency response team would like to quickly mine the vast data of a large variety of active sensors (wind, earth vibrations, highway traffic), and passive databases (historic sensor data, population statistics, immunization records).

The use cases above share many commonalities, but their current implementations differ in that they utilize a variety of heterogeneous database technologies, data formats, query and response languages, routing strategies and network protocols. They also exhibit different degrees of node churn: some participants are almost always available (e.g. core servers), while others frequently come and go (e.g. laptop users.

Proposal

We propose a flexible, powerful and extensible resource discovery and data integration framework that may be used by a wide variety of scientific applications. This framework should meet the following requirements:

1) transport-independence
2) work in multi-hop system (peer-to-peer, SOAP)

3) provide effective and scalable I/O abstractions
4) allow for incremental queries that can gather results sequentially rather than all at once
5) support any query language, including XQuery, SQL, XPath and regular expressions
6) supports different models of control, efficiency and security
7) enable smart dynamic routing (allow a client to select the data and peers the query should be applied to)

As nodes (and neighbors) dynamically join and leave the network, the P2P topology changes in an adaptive manner over time. We propose to develop dynamic algorithms to bootstrap and adapt the topology in order to optimize given metrics such as overall and individual response time (latency), reliability (redundancy, reachability), message count and bandwidth consumption. A goal for these algorithms is to acknowledge heterogeneous node roles (e.g. router, database server, file server, network monitoring node, chat user), heterogeneous capabilities (e.g. different data sources and query languages, with different policies), and wildly heterogeneous capacities of individual nodes (e.g. orders of magnitude difference in bandwidth, latency, storage space).

Conclusions

The abstractions that this framework can support are quite powerful. In particular, the use cases outlined in the motivation can be implemented by application-specific plug-ins once the framework is in place. For example, the resource broker use case could use an XQuery language module, in conjunction with an XML module, flooding as a routing strategy, a simple XPath query for neighbor selection, and a topology adaptation algorithm directed towards high availability.

Such an infrastructure is critical to achieve a functioning adaptive Grid and to enable geographically separated scientists to more effectively work together as a team. Such work is also critical in subsequently enabling heterogeneous P2P networks of services for science applications, integrating many distinct infrastructures and application components into innovative and reusable productivity tools.

References

[1] Wolfgang Hoschek. Peer-to-Peer Grid Databases for Web Service Discovery. Journal of Concurrency and Computation: Practice and Experience (CCPE), 2002. Wiley Press.

[2] Peer-to-Peer I/O (P2PIO) Protocol Specification, K. Berket, A. Essiari, D. Gunter, W. Hoschek, 2003, http://dsd.lbl.gov/firefish.

[3] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In Workshop on Design Issues in Anonymity and Unobservability, 2000.

[4] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In ACM SIGCOMM, 2001.

[5] Steven Waterhouse. JXTA Search: Distributed Search for Distributed Networks, 2001. White Paper, http://www.jxta.org.

[6] Brian Tierney, Ruth Aydt, Dan Gunter, Warren Smith, Valerie Taylor, Rich Wolski, and Martin Swany. A Grid Monitoring Architecture. Technical report, Global Grid Forum Informational Document, January 2002.

[7] Jason Lee, Dan Gunter, Martin Stoufer, and Brian Tierney. Monitoring Data Archives for Grid Environments. In Proc. of the Int'l. IEEE/ACM Supercomputing Conference (SC 2002), Baltimore, USA, November 2002. IEEE Computer Society Press.

[8] Karl Czajkowski, Steven Fitzgerald, Ian Foster, and Carl Kesselman. Grid Information Services for Distributed Resource Sharing. In Tenth IEEE Int'l. Symposium on High-Performance Distributed Computing (HPDC-10), San Francisco, California, August 2001.