IEEE distributed systems Expert-authored articles and resources ONLINE

IEEE DISTRIBUTED SYSTEMS ONLINE 1541-4922 © 2006 Published by the IEEE Computer Society
Vol. 7, No. 3; March 2006

Distributed Wisdom: Designing a Replication Service for Large Peer-to-Peer Data Grids

A. Vijay Srinivas, *Indian Institute of Technology Madras*
M. Venkateswara Reddy, *Indian Institute of Technology Madras*
D. Janakiram, *Indian Institute of Technology Madras*

We're part of a group that's realizing Vishwa (http://dos.iitm.ac.in/Vishwa), a peer-to-peer middleware architecture for developing grid applications. Two of us had a research brainstorming session with our advisor (the third author) during tea one fine day. We present parts of the session here to get across the key issues in building services for large-scale distributed systems.

VIJAY : Researchers are producing large amounts of scientific data—for instance, see the Grid Physics Network Project (http://www.griphyn.org). Distributed computations on this data must be scheduled, and this data must be available to a large number of scientists. So, there's a need to replicate the data at appropriate locations to handle node and network failures and minimize computation time, bandwidth, or both. We wish to develop a platform that could serve as a substrate for building the replication service required for large P2P data grids.

ADVISOR : This platform sounds interesting, but what are its key requirements?

VENKAT :

- *Scalability*. The platform must scale up to a large number (possibly millions) of nodes and data units (objects) in the system. The platform must also be scalable geographically; that is, it must work well over the Internet.

- *Dependability*. The Internet environment is dynamic, with nodes and networks prone to failures. So, the platform must allow applications to seamlessly adapt to failures.

- *Middleware reconfigurability*. The middleware (platform) components must also be adaptive to these dynamics.

- *Data consistency*. Data might be replicated for reasons of performance, fault tolerance, maintenance, and so on. So, the platform must ensure consistency of replicas. Metadata catalogs that store information about data (the data's location, description, and so on) must be maintained, and metadata might also be replicated. So, the platform must also ensure metadata consistency.

- *Efficient lookup*. The platform must provide efficient mechanisms for looking up objects and data.

ADVISOR : Distributed-shared-object spaces such as Orca and Linda provide a nice abstraction, comparable to distributed-object middleware, and they address issues relating to data consistency. Why can't they suffice as the required platform?

VIJAY : Existing shared-object spaces don't scale up owing to the presence of centralized components, their inability to handle failures, and inefficient object lookup and consistency mechanisms.[1] So, you can't use existing shared-object spaces to build services for large-scale distributed systems.

ADVISOR : Probably what you're addressing isn't just the abstraction but also the underlying realization of the abstraction. P2P systems have addressed scalability and fault tolerance quite well. Why can't you use P2P systems such as Gnutella, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf, (pdf) directly as the required platform?

VENKAT : Unstructured P2P systems such as Gnutella use a random graph as the overlay and use flooding or random walks to search for data. Although they can support complex queries, they can't provide deterministic retrieval time guarantees about the search.

ADVISOR : Structured P2P systems such as Tapestry[2] or Pastry (http://freepastry.org) use *Distributed Hash Tables* or other data structures as the overlay. Nodes form the

overlay on the basis of node identifiers, and the system gives objects identifiers from the same space. DHTs map an object identifier with a node identifier (id) that's responsible for information on that object. So, they provide $O(\log(n))$ guarantees for object searches. Why can't you use them directly to build the platform?

VIJAY: They're limited to exact queries (structured P2P systems don't allow complex queries such as "get all nodes with computing power greater than *aThreshold* and storage greater than *bThreshold*"). Structured P2P systems determine a node's neighborhood on the basis of node ids, whereas unstructured P2P systems allow any application-specific criteria for neighborhood formation. In grids, nodes must be able to form neighbors on the basis of their capabilities, which might be difficult with structured P2P systems. Furthermore, data placement might be constrained in large data grids. Unstructured P2P systems allow unconstrained data placement, so popular data gets placed on "good" nodes.

ADVISOR : Both structured and unstructured P2P systems have their pros and cons. Combining the two might be useful to handle complex queries and churn better in addition to providing $O(\log(N))$ guarantees.[3] (In P2P systems, node/network dynamics resulting in routing-table updates and/or data movement is known as churn) You should understand that our earlier design for the platform[1] is similar to superpeer-based loosely structured P2P systems (such as Kazaa, http://www.kazaa.com/us/index.htm). The main disadvantage of such systems is that handling superpeer failures isn't easy. So, we're looking at combining only structured and unstructured P2P systems now. How would you combine the two to build the required platform?

VENKAT : The unstructured layer restricts metadata replication to zones or clusters, making it easier to maintain metadata consistency. In addition, it allows neighborhood formation based on node capabilities (computing power, memory, data storage capability, and so on). The structured layer stores information required to recover from failures with $O(\log(N))$ guarantees.

VIJAY : However, we can't use the two-layer P2P routing substrate directly as the required platform. First, it doesn't address data management issues. Furthermore, it might not provide a good abstraction for programmers.

ADVISOR : We can develop a scalable, reconfigurable shared-object space over this two-layer architecture. It will incorporate mechanisms to maintain replicated data (and metadata). It must also provide a shared-object abstraction over a wide-area distributed system. But has anyone else tried this approach?

VENKAT : Gabriel Antoniu, Luc Bougé, and Mathieu Jan have attempted to unify P2P systems and distributed shared memory.[4] Their approach uses a JXTA (http://www.jxta.org) multicast primitive to ensure consistency. This might be unreliable. It also needs total ordering, so it might not scale up. And, like all DSMs, it might suffer from false sharing. (False sharing in DSMs refers to the problem of sharing unwanted data because the sharing granularity is at the level of pages. Shared object spaces allow an application to specify the granularity of sharing, thereby avoiding false sharing.)

VIJAY : Globe is a large-scale shared-object space,[5] but it uses a tree-based mechanism for object lookups. In contrast, we use DHTs for object lookup. Consequently, we can handle failures, while Globe can't.

ADVISOR : So, our approach of developing a replication service using a scalable, reconfigurable shared-object space realized over a two-layer P2P architecture appears unique. However, we need to benchmark the platform to test the performance. Also, we must look at key issues in designing the replication service itself, including data redundancy and efficient data placement strategies to minimize computation time and bandwidth.

Conclusion

We're investigating this idea and will tell you more after our next tea conversation. (A prototype of the Vishwa data grid has been developed since this discussion and is available for download, http://dos.iitm.ac.in/Vishwa.)

References

1. 1. A.V. Srinivas and D. Janakiram , "Scaling a Shared Object Space to the Internet: Case Study of Virat," (pdf) to be published in *Journal of Object Technology,*Sept./Oct. 2006; http://dos.iitm.ac.in/LabPapers/jot.pdf.
2. 2. B.Y. Zhao , et al., "Tapestry: A Resilient Global-Scale Overlay for Service Deployment,"*IEEE J. Selected Areas in Communications,* vol. 22, no. 1, 2004, pp. 41-53.
3. 3. M. Castro , M. Costa and A. Rowstron , "Debunking Some Myths about Structured and Unstructured Overlays,"*Proc. 2nd Usenix Symp. Networked System Design and Implementation,* Usenix, 2005.

4. 4. G. Antoniu , L. Bougé and M. Jan , "JuxMem: Weaving Together the P2P and DSM Paradigms to Enable a Grid Data-Sharing Service,"*Scalable Computing: Practice and Experience,* vol. 6, no. 3, 2005, pp. 45-55.
5. 5. M. van Steen , P. Homburg and A.S. Tanenbaum , "Globe: A Wide-Area Distributed System," http://csdl2.computer.org/persagen/ DLAbsToc.jsp?resourcePath=/dl/mags/pd/&toc=comp /mags/pd/1999/01/p1toc.xml&DOI=10.1109/4434.749137, *IEEE Concurrency,*Jan.-Mar. 1999,pp. 70-78.

A. Vijay Srinivas is a PhD research scholar in the Distributed and Object Systems Lab at the Department of Computer Science and Engineering of the Indian Institute of Technology Madras. Contact him at avs@cs.iitm.ernet.in.

M. Venkateswara Reddy is an MS research scholar in the Distributed and Object Systems Lab at the Department of Computer Science and Engineering of the Indian Institute of Technology Madras. Contact him at venkatm@cs.iitm.ernet.in.

D. Janakiram is a professor at the Department of Computer Science and Engineering of the Indian Institute of Technology Madras. Contact him at djram@cs.iitm.ernet.in.

Related Links

- DS Online's Peer to Peer Community, http://dsonline.computer.org/portal/site/dsonline/ index.jsp?pageID=dso_level1_home&path =dsonline/topics/p2p&file=index.xml&xsl =generic.xsl

- DS Online's Grid Computing Community, http://dsonline.computer.org/portal/site/dsonline/ menuitem.0e7741ff4cba82ff96d34f108bcd45f3/index.jsp ? &pName=dsonline_grid_test&

- "Heterogeneous Search in Unstructured Peer-to-Peer Networks" (pdf), http://csdl2.computer.org/comp/mags/ds/2005/02/o2001.pdf

- "Free Riding on Gnutella Revisited: The Bell Tolls?" (pdf), http://csdl2.computer.org/comp/mags/ds/2005/06/o6001.pdf

- "The Tech Hotlist: Grid Computing and P2P" (a review of From P2P to Web Services and Grids: Peers in a Client Server World, by Ian J. Taylor) (pdf), http://csdl2.computer.org/comp/mags/ds/2005/11/oy005.pdf

Cite this article: A. Vijay Srinivas, M. Venkateswara Reddy, and D. Janakiram, "Designing a Replication Service for Large Peer-to-Peer Data Grids," *IEEE Distributed Systems Online*, vol. 7, no. 3, 2006, art. no. 0603-o3002.