From the Editor-in-Chief...

Peering at Peer-to-Peer Computing

Munindar P. Singh • *singh@ncsu.edu*

How would you like to share files with another user without having to explicitly place them in a designated external location? The recent successes of (and controversies surrounding) Napster, Gnutella, and FreeNet have drawn attention to peer-to-peer computing, which allows precisely such interactions between information and service providers and their customers. Let's take a brief look at peer-to-peer computing, or P2P for short, and its main variants—both those that are popular and those that ought to be.

P2P can be defined most easily in terms of what it is not: the client-server model, which is currently the most common model of distributed computing. In the client-server model, an application residing on a client computer invokes commands at a server. In P2P, an application is split into components that act as equals. The client-server model is simple and effective, but it has serious shortcomings, which I will discuss shortly.

P2P can be defined most easily in terms of what it is not: the client-server model.

P2P is by no means a new idea. The distributed computing research community has studied it for decades. Networks themselves demonstrate P2P in action: Ethernet is nothing if not a P2P protocol, and network routing operates through routers acting as peers with other routers. The difference in the recent focus on P2P seems to be that it has finally caught the imagination of people building practical systems at the application layer. And for good reason.

Going Beyond Client-Server

The centralization of information on servers makes for performance bottlenecks and for overall system susceptibility to single-point failure. Cluster computing and even content networking are inspired by the idea of preserving the logical centrality of servers while replicating them to build system redundancies sufficient to sustain higher performance and to support graceful functional degradation under failure.

The implicit claim behind P2P is that these redundancies aren't enough. A political motivation for P2P is that as long as a system retains some centralization, it can be controlled from that central location. Some people think that Napster can be shut down simply because it has a server and the site running that server can be sued. P2P approaches without this type of central server, such as FreeNet, would be much harder to shut down. The notion of technology to help resist repressive regimes is appealing, although I would worry if the freedom were used mainly for piracy.

A technical motivation for P2P is that in client-server computing, the control rests entirely in the client; the server merely responds to requests. By requiring all control to reside on the client, the client-server model forces applications to be structured in such a way that coordination between their components is rigid. Each client interacts with the server independently of other clients. This property is codified in the traditional *transaction* model, which is great for isolation (as among bank accounts), but not so great for collaboration.

Richer Interaction Models

P2P can support richer models of interaction than client-server. These models take three main forms:

- *Symmetric client-server.* Each party can query the other, thereby giving each power over the other at different times. The idea of symmetry is appealing in general, but this particular form, which is often touted as an explanation of P2P's inherent power, doesn't look much beyond client-server.

- *Asynchronous.* If the client needs to know of changes observed by the server, it must poll the server to learn of them. As peers, computations naturally communicate asynchronously. This is the original form of P2P. The request-response paradigm corresponds to "pull" communication, while asynchronous communication corresponds to "push." Unfortunately, push communication got a lot of bad press with applications that place their entire intelligence on the server (pushing) side. Pushing ads is another name for spamming.
- *Federation of equals.* When we apply asynchrony in settings where all parties have an equal share in and equal control over a computation, we can create applications of a performance and usability that is simply unattainable by client-server—applications that offer symmetry not only at the level of communication, but also in terms of decision-making. Such applications are at the heart of solutions for virtual enterprises and truly flexible B2B interoperation. This would require making the reasoning and policies of interaction explicit and enabling each party to take and concede the initiative as it sees fit. In my view, only intellectual equals can be true peers.

Today's P2P systems involve the first two variants. While useful, these systems provide limited abstractions for applications that go beyond file sharing. Only when we support flexible interactions will we achieve true P2P and realize the benefits of dynamic interoperation that the Internet enables.

How to Reach IC

Articles

We welcome submissions about Internet application technologies. For detailed instructions and information on peer review, *IEEE Internet Computing*'s author guidelines are available online at http://computer.org/internet/edguide.htm.

Letters to the Editor

Please send letters, including reference to articles in question, via e-mail to internet-computing@computer.org.

Reuse Permission

For permission to reprint an article published in *IC*, contact William J. Hagen, IEEE Copyrights and Trademarks Manager, IEEE Service Center, 445 Hoes Lane, Piscataway, N.J. 08855-1331; w.hagen@ieee.org. Complete information is available at http://computer.org/permission.htm. To purchase reprints, see http://computer.org/author/reprint.htm.

IEEE INTERNET COMPUTING
IEEE Computer Society Publications Office
10662 Los Vaqueros Circle, PO Box 3014
Los Alamitos, CA 90720-1314

EDITOR-IN-CHIEF
Munindar P. Singh • singh@ncsu.edu

ASSOCIATE EDITOR-IN-CHIEF
Robert Filman • rfilman@arc.nasa.gov

EDITORIAL BOARD
Salah Aidarous • salah@rss.dl.nec.com
(IEEE Communications Society Liaison)
Donna Auguste • donna@freshwater.com
Miroslav Benda • miro@amazon.com
Elisa Bertino • bertino@dsi.unimi.it
Scott Bradner • sob@harvard.edu
Fred Douglis • douglis@research.att.com
Stuart I. Feldman • sif@us.ibm.com
Li Gong • li.gong@sun.com
Seif Haridi • seif@it.kth.se
Michael N. Huhns • huhns@sc.edu
Gail E. Kaiser • kaiser@cs.columbia.edu
Leonard Kleinrock • lk@cs.ucla.edu
Doug Lea • dl@altair.cs.oswego.edu
Frank Maurer • maurer@cpsc.ucalgary.ca
Charles E. Perkins • charliep@iprg.nokia.com
Charles J. Petrie • petrie@nrc.stanford.edu
(EIC emeritus)
Agostino Poggi • poggi@ce.unipr.it
Anthony Michael Rutkowski • amr@chaos.com
Ravi Sandhu • sandhu@gmu.edu

STAFF
Senior Acquisitions Editor: **Linda World**
lworld@computer.org
Associate Editor: **Joan Taylor**
jtaylor@computer.org
Staff Editor/IC Online Design: **Steve Woods**
swoods@computer.org
Magazine Assistant: **Hazel Kosky**
hkosky@computer.org
Art Director: **Joseph Daigle**
Cover Design: **Carmen Flores-Garvey**
Graphic Artist: **Ken Duckworth**
Contributing Editors: **David Clark, Keri Schreiner, Martin Zacks**
Publisher: **Angela Burgess**
Membership/Circulation Marketing Manager: **Georgann Carter**
Business Development Manager: **Sandy Aijala**
Advertising Supervisor: **Marian Anderson**

CS MAGAZINE OPERATIONS COMMITTEE
Sorel Reisman (Chair), James H. Aylor, Jean Bacon, Thomas J. Bergin, Wushow Chou, George Cybenko, William I. Grosky, Steve McConnell, Daniel E. O'Leary, Ken Sakamura, Munindar P. Singh, James J. Thomas, Yervant Zorian

CS PUBLICATIONS BOARD
Rangachar Kasturi (Chair), Jake Aggarwal, Laxmi Bhuyan, Lori Clarke, Mike T. Liu, Sorel Reisman, Gabriella Sannitti diBaja, Sallie Sheppard, Mike Williams, Zhiwei Xu