Scooped, Again

Jonathan Ledlie, Jeff Shneidman, Margo Seltzer, John Huth
Harvard University
{jonathan,jeffsh,margo}@eecs.harvard.edu
huth@physics.harvard.edu

Abstract

The Peer-to-Peer (p2p) and Grid infrastructure communities are tackling an overlapping set of problems. In addressing these problems, p2p solutions are usually motivated by elegance or research interest. Grid researchers, under pressure from thousands of scientists with real file sharing and computational needs, are pooling their solutions from a wide range of sources in an attempt to meet user demand. Driven by this need to solve large scientific problems quickly, the Grid is being constructed with the tools at hand: FTP or RPC for data transfer, centralization for scheduling and authentication, and an assumption of correct, obediant nodes. If history is any guide, the World Wide Web depicts viscerally that systems that address user needs can have enormous staying power and affect future research. The Grid infrastructure is a great customer waiting for future p2p products. By no means should we make them our *only* customers, but we should at least put them on the list. If p2p research does not at least address the Grid, it may eventually be sidelined by defacto distributed algorithms that are less elegant but were used to solve Grid problems. In essense, we'll have been scooped, again.

1 Introduction

Butler Lampson, in his SOSP99 Invited Talk, stated that the greatest failure of the systems research community over the past ten years was that "we did not invent the Web" [39]. The systems research community laid the groundwork, but did not follow through. The same situation exists today with the overlapping efforts of the Grid and p2p communities. The former is building and using a global resource sharing system, while the latter is repeating their mistake of the past by focusing on elegant solutions without regard to a vast potential user community.

In 1989, Tim Berner-Lee's need to communicate his own work and the work of other physicists at CERN led him to develop HTML, HTTP, and a simple browser [4].

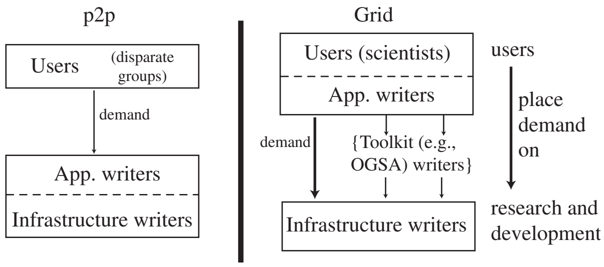

Figure 1: In serving their well-defined user base, the Grid community has needed to draw from both its ancestry of supercomputing and from Systems research (including p2p and distributed computing). P2p has essentially invented its user base through its technology.

While HTTP and HTML are simple, they have exhibited serious network and language-based deficiencies as the Web has grown [21]. These problems have been examined and patched to some extent, but this simple inelegant solution remains at the core of the Web.

A parallel situation exists today with the p2p and Grid communities. A large group of users (scientists) are pushing for immediately useable tools to pool large sets of resources. The Grid is currently building these tools. The Systems community is in danger of falling victim to the contemporary version of Lampson's admonishment if we do not participate in this process.

The Grid is an active area of research and development. The number of academic grids has jumped six-fold in the last year [43]. As Figure 1 depicts, the "customer base" of scientists, who are often also application writers, drive Grid developers to produce tangible solutions, even when the solutions are not ideal, from a computer systems perspective. Most current p2p users are people sharing files; sometimes these people are pursuing noble causes like anonymous document distribution, but more often they are simply trying to circumvent copyright, and rarely are they interacting with the systems' developers. P2p does not have the driving force that interactive users provide and therefore has focused on solutions that are interesting primarily from a research perspective.

The difficulty with this parallel development is not that it is wasteful, but that, without the p2p community's input, the Grid will most likely be built in a way incompatible and non-inclusive of many of p2p's strong points: search and storage scalability, decentralization, anonymity and pseudonymity, and denial of service prevention.

In the rest of this paper, we first discuss the charters of the two different communities. We introduce three fallacies that may have kept the communities separate. We then describe common problems being attacked by the two communities and compare solutions from each camp, and discuss problems that seem to be truly disjoint. We conclude with a call to action for the p2p community to examine the Grid needs and consider future research problems in that context.

2 Grids

2.1 What is the Grid?

Buyya defines the Grid as "a type of parallel and distributed system that enables the sharing, selection, and aggregation of resources distributed across multiple administrative domains based on their (resources) availability, capability, performance, cost, and users' quality-of-service requirements" [8]. The Grid is "distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and in some cases, high performance orientation" [27].

2.2 Goals of the Grid

The Grid aims to be self-configuing, self-tuning, and self-healing, similar to the goals in autonomic computing [2]. It aims to fulfill the vision of Corbato's Multics [13]: like a utility company, a massive resource to which a user gives his or her computational or storage needs. The Grid's goal is to utilize the shared storage and cycles from the middle *and* edges of the Internet.

2.3 Manifestations

Grids historically arose out of a need to perform massive computation. A manifestation of shared computation, Condor accepts compiled jobs, schedules and runs them on remote idle machines [42]. Exemplifying its focus on computation, it issues RPCs back to the job originator's machine for data. An example of computational middleware is Globus [23]; it "meta-schedules" jobs among Grids like Condor and ships host data between them using GridFTP, an FTP wrapper. Manifestations of Data Grids include the European Data Grid project [20].

The Grid community is currently authoring a Web Services-oriented API called Open Grid Services Architecture (OGSA) [24, 25]. The next generation of Globus is intended to be a reference implentation of this API.

3 P2P

3.1 What is P2P?

Unintentionally, Shirky describes p2p much like a Grid: "Peer-to-peer is a class of applications that take advantage of resources — storage, cycles, content, human presence — available at the edges of the Internet" [19]. Stoica *et al.* offer a more restrictive definition focusing on decentralized and non-hierarchical node organization [54].

P2p's focus on decentralization, instability, and fault tolerance exemplify areas that essentially have been omitted from emerging Grid standards, but will become more significant as the system grows.

3.2 Goals of P2P

The goal of p2p is to take advantage of the idle cycles and storage of the edge of the Internet, effectively utilizing its "dark matter" [19]. This overarching goal introduces issues including decentralization, anonymity and pseudonymity, redundant storage, search, locality, and authentication.

3.3 Manifestations

O'Reilly's p2p site [46] divides p2p systems into nineteen categories, primarily offering file sharing (*e.g.,* Gnutella [28], KaZaA [36]), distributed computation (*e.g.,* distributed.net [17]), and anonymity (*e.g.,* Freenet [12], Publius [44]). Seti@home is another p2p application, although the Grid community considers it one of theirs as well [51].

Prominant research instances of p2p include Chord, Pastry, and Tapestry [48, 53, 56]. File systems built on these include CFS, PAST, and Oceanstore [14, 18, 37]. File systems, however, are not applications and, while a variety of forms of storage have been built, there exists no driving application in this space.

4 Three Falacies That Have Kept the Communities Separate

This paper argues that the p2p and Grid communities are "natural" partners in research. The following are some objections and responses to this claim.

4.1 "The technical problems in Grid systems are different from those in p2p systems."

Conventional wisdom posits "the Grid is for computational problems" and "p2p is about file sharing." Historically, Grids have grown out of the computationally-bound supercomputers and local batch computation systems like Condor. As these localized systems have become linked to one another in the Grid, handling data has become a much larger problem. Some Grid-connected instruments (*e.g.,* specialized telescopes) focus purely on data production for others to use. Similarly, p2p is moving in the computation direction with efforts in desktop collaboration [31] and network computation [5].

Formally stated "open problems" papers from each camp exhibit a striking similarity in their focus on formation, utilization, security, and maintenance [16, 45, 49]. Conference proceedings echo this trend. Section 5 summarizes areas of active overlap.

4.2 "While the technical problems are similar, the architectures (physical topology, bandwidth availibility and use, trust model, etc.) demand that the specific solutions be fundamentally different."

Researchers familiar with both areas claim that the two will blend as they mature, even perhaps to the point of a merging of the two research communities [6, 22, 50]. Regardless of the veracity of this forecast, it indicates that researchers see good ideas in each community that can solve common problems.

This fallacy is application dependent. Some applications (*e.g.,* military missile calculation or totally anonymous file sharing) impose special requirements for assorted (not always technical) reasons. But even in these situations, a general awareness of technical approaches taken by the other community may help solve "physically private" problems.

4.3 "Grid projects do not have the flexibility to try new algorithms/ideas because they have to get real work done. P2P research is all about this flexibility."

P2p research is very flexible: one version can obsolete the previous and new algorithms can be developed without conforming to any standard. Within the emerging Grid standards, however, there is room for flexible research too.

Grid researchers recognize the need for test-beds as staging grounds for new applications and protocols [52]. Traditional Grid settings have been in university settings where support staff is on hand to test and deploy new software updates [34]. Moreover, as evidenced by the many toolkits and custom Grid implementations [23, 30], Grid users are willing to adopt different technologies to get their work done.

It seems natural for p2p researchers to develop algorithms either independently or within Grid test-beds, and then "publish" their prototypes within a Grid setting. For example, Systems researchers could build to a particular aspect of the OGSA, benchmark their solution to this API, and then release it as a fresh, improved implementation.

5 Shared Technical Problems

We list and compare the two communities' approaches to problems in four categories and conclude this section with problems that are not shared.

5.1 Formation

Topology formation and peer discovery deals with the problem of how nodes join a system and learn about their neighbors, often in an overlay network [38]. Membership protocols have been explored in both settings [34, 40]. While espousing the autonomous ideal, much Grid infrastructure is hardcoded and could benefit from the active formation found in p2p research prototypes.

5.2 Utilization

Resource discovery determines how we find "interesting items," which can include sets of files, computers, services (compute/storage services), or devices (such as printers or telescopes). Data is also searched within files [16] and across relational tuples [32]. Search requirements exhibit tradeoffs in expressiveness, robustness, efficiency, and autonomy. For some, but not all applications, the ubiquitous hash-based lookup schemes of the p2p world are appropriate.

Resource management and optimization problems deal with how "best" to utilize resources in a network. This category includes data placement, computation, fairness, and communication usage decisions such as, "Where/How do we distribute data/metadata?", "Who performs a certain computation?", "Which links do we use to transmit/receive information?", and "How can I speed up access to popular files?" Both communities have examined data replication and caching algorithms to increase performance [3, 12].

Scheduling and handling of contention has been examined in both communities. P2p has focused on bandwidth usage (*e.g.,* Gnutella's maximum connections) with solutions that are often resolved with dynamic programming at the node level. Grids are often centered around scheduling and use traditional scheduling techniques involving cost functions implemented by a central scheduler and often lack QoS or fairness guarentees [7]. Agoric solutions in Grids have been centralized thus far; distributed economic scheduling mechanisms may be more appropriate for fault tolerance reasons [55].

Load balancing/splitting schemes in both communities attempt to reduce the load of a particular file, link, or computation by breaking larger blobs into many smaller distributed atoms.

Popular p2p file sharing programs are trivial to join, use, and leave, with `get`, `put`, and `delete` as their primative operations. Grid users (scientists) cannot use some simple p2p solutions, mainly because what they are trying to do (*e.g.,* distributed code instrumentation) is complex. The lack of tools for application developers (and the complexity of the existing tools) is also a major stumbling block [49].

5.3 Coping with Failure

Partial and arbitrary failure must be addressed in any realistic distributed network. Most p2p systems punt on guarenteeing accessibility by accepting lossy storage in their model [12, 28]. If the ideas from p2p storage are going to be successfully applied to the Grid, p2p researchers need to consider revising the common loss model. They also should consider the order-of-magnitude of storage some Grid experiments will produce: some are expected to produce on the order of half a petabyte per month [47], about the current capacity of KaZaA. This data, however, cannot be lossy.

Traditional distributed systems techniques for dealing with failure often make assumptions on connectivity and complexity that may not be appropriate for traditional p2p systems or Grid systems, as both Daswani *et al.* [16] and Schopf *et al.* [49] note in their respective open problems papers. The same replication algorithms used to increase availability can also serve to ensure correctness in the face of data, computation, or communication failures.

Security-related research in the two communities includes authenticity issues (such as verification of data or computation, or handling man-in-the-middle style attacks), availibility issues (surviving denial of service attacks), and authorization issues (such as access control), but research from p2p would help address some DoS problems introduced by the Grid's frequent reliance on centralized structures [1, 9]. Chang *et al.* examine enforcing safety with certifying compilers in a Grid environment [10].

These aspects have been identified and explored in the context of data sharing p2p systems [16]. Computational p2p systems have similar problems: more than fifty percent of the Seti@Home's project resources were at one time spent dealing with security problems, including the problem of "cheating" on the distributed computation [35].

The Grid community faces the same security problems [49]. The work on the Grid Security Infrastructure [26] addresses the problem of authentication, but inter-testbed authentication remains to be resolved [33]. Focus on decentralized authentication schemes like SPKI may be a step in the right direction [11].

5.4 Maintenance

In the areas of deployment and managability, p2p has essentially no standards or APIs; with the possible exception of Gnutella, each version obsoletes the previous one. Many Grid papers profess the need for a standardized programming interface [49], and by necessary, the Grid is being forced to standardize and the OGSA is beginning to help. Similar efforts in p2p standardization are the Berkeley BOINC [5], Google Compute [29], and overly standardization [15].

5.5 Disjoint Problems

Not every problem in "pure" p2p research has an analog in the Grid community. Anononymity (often grouped with security issues) for instance, may not be so useful, but a middleground of pseudonymity may exist. Anonymizing systems are important and offer a prime example of a consideration left out of the emerging Grid standards: for example, they are currently being used to promote free speech in China [41].

6 Call to Action

How do we avoid being scooped again? Given the large degree of commonality between the two worlds, "coming up to speed" on the Grid is not a difficult undertaking. As a community and individuals, we must familiarize ourselves with the set of problems the Grid is addressing, so we can identify the areas where we have solutions or are actively working on solutions. We must also work to understand Grid users' day to day needs; how robust must a solution be in order to be appropriate for deployment on a Grid? We should understand the standards they are developing and to which they expect all systems to comply (OSGA). We can make a significant contribution helping them create a standard that allows the evolution and experimentation that "outside" researchers can provide. Perhaps the answer lies in a directive as simple as: "Find a user and figure out what that user needs."

References

[1] A. Adya, W. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In *OSDI '02*, Boston, MA, 2002.

[2] Autonomic computing. `http://www.research.ibm.com/autonomic/`.

[3] William H. Bell, David G. Cameron, Luigi Capozza, A. Paul Millar, Kurt Stocklinger, and Floriano Zini. Simulation of dynamic grid replication strategies in optorsim. In *Grid 2002*, November 2002.

[4] Tim Berners-Lee and R. Cailliau. WorldWideWeb: Proposal for a HyperText project, 1990.

[5] Berkeley BOINC. `http://boinc.berkeley.edu`.

[6] Scott Bradner. The rest of peer-to-peer. *Network World Fusion*, October 2002.

[7] R. Buyya, H. Stockinger, J. Giddy, and D. Abramson. Economic models for management of resources in peer-to-peer and grid computing. In *ITCom 2001*, August 2001.

[8] Rajkumar Buyya. Grid computing info centre. `http://www.gridcomputing.com`, 2002.

[9] M. Castro, P. Drushel, A. Ganesh, A. Rowstron, and D. Wallach. Secure routing for structured peer-to-peer overlay networks. In *OSDI '02*, Boston, MA, 2002.

[10] Bor-Yuh Evan Chang, Karl Crary, Margaret DeLap, Robert Harper, Jason Liszka, Tom Murphy VII, and Frank Pfenning. Trustless Grid Computing in ConCert. In *Grid Computing - GRID 2002, Third International Workshop*, November 2002.

[11] Dwaine Clarke, Jean-Emile Elien, Carl Ellison, Matt Fredette, Alexander Morcos, and Ronald L. Rivest. Certificate chain discovery in SPKI/SDSI. *Journal of Computer Security*, January 2001.

[12] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore Hong. Freenet: A distributed anonymous information storage and retrieval system. `http://freenetproject.org/cgi-bin/twiki/view/Main/ICSI`, 2001.

[13] F.J. Corbato and V.A. Vyssotsky. Introduction and overview of the multics system. In *Proceedings of AFIPS FJCC*, 1965.

[14] Frank Dabek, M. Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica. Wide-area cooperative storage with CFS. In *SOSP '01*, October 2001.

[15] Frank Dabek, Ben Zhao, Peter Druschel, and Ion Stoica. Towards a common API for structured peer-to-peer overlays. In *IPTPS '03*, Berkeley, CA, February 2003.

[16] Neil Daswani, Hector Garcia-Molina, and Berverly Yang. Open problems in data-sharing peer-to-peer systems. In *9th International Conference on Database Theory*, January 2003.

[17] Distributed.net. `http://distributed.net`.

[18] Peter Druschel and Antony Rowstron. PAST: A large-scale, persistent peer-to-peer storage utility. In *SOSP '01*, October 2001.

[19] Andy Oram (ed.). Gnutella. In *Peer-to-Peer: Harnessing the power of disruptive technologies*. O'Reilly & Associates, 2001.

[20] European union data grid project. `http://eu-datagrid.web.cern.ch/eu-datagrid/`, 2001.

[21] Roy Fielding. Representational state transfer: An architectural style for distribtuted hypermedia interactive (research talk), 1998.

[22] John Fontana. P2p getting down to some serious work. *Network World Fusion*, August 2002.

[23] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications and High Performance Computing*, Summer 1997.

[24] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid ser-

vices architecture for distributed systems integration. http://www.globus.org/research/papers/ogsa.pdf, 2002.

[25] I. Foster, C. Kesselman, J.M. Nick, and S. Tuecke. Grid services for distributed systems integration. *IEEE Computer*, 35(6), 2002.

[26] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *ACM Conference on Computers and Security*, November 1998.

[27] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid. *The International Journal of Supercomputer Applications*, 15(3), Fall 2001.

[28] Gnutella. Gnutella protocol specification v0.4. http://www.clip2.com/GnutellaProtocol04.pdf, 2001.

[29] Google compute. http://toolbar.google.com/dc/.

[30] Andrew S. Grimshaw and William A. Wulf. The legion vision of a worldwide virtual computer. *Communications of the ACM*, January 1997.

[31] Groove networks desktop collaboration software. http://www.groove.net/, 2002.

[32] M. Harren, J. Hellerstein, R. Huebsch, B. Loo, S. Shenker, and I. Stoica. Complex queries in DHT-based peer-to-peer networks. In *IPTPS '02*, March 2002.

[33] M. Humphrey and M. Thompson. Security implications of typical grid computing usage scenarios. In *HPDC 10*, August 2001.

[34] Adriana Iamnitchi and Ian Foster. On fully decentralized resource discovery in grid environments. In *International Workshop on Grid Computing*. IEEE, November 2001.

[35] Leander Kahney. Cheaters bow to peer pressure. http://www.wired.com/news/technology/0,1282,41838,00.html, 2001.

[36] KaZaA. http://www.kazaa.com.

[37] John Kubiatowicz, David Bindel, Yan Chen, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westly Weimer, Christopher Wells, and Ben Zhao. Oceanstore: An architecture for global-scale persistent storage. In *ASPLOS-IX*, November 2000.

[38] Shay Kutten and David Peleg. Deterministic distributed resource discovery. In *PODC 2002*, July 2000.

[39] Butler Lampson. Computer systems research: Past and future (invited talk). In *SOSP '99*, December 1999.

[40] Jonathan Ledlie, Jacob Taylor, Laura Serban, and Margo Seltzer. Self-organization in peer-to-peer systems. In *10th EW SIGOPS*, September 2002.

[41] Jennifer 8. Lee. Guerrilla warfare, waged with code. *New York Times*, October 2002.

[42] Michael Litzkow, Miron Livny, and Matt Mutka. Condor - a hunter of idle workstations. In *Proceedings of the 8th International Conference of Distributed Computing Systems*, June 1988.

[43] Om Malik. Ian foster = grid computing. *Grid Today*, October 2002.

[44] Aviel D. Rubin Marc Waldman and Lorrie Faith Cranor. Publius: A robust, tamper-evident, censorship-resistant, web publishing system. In *9th USENIX Security Symposium*, August 2000.

[45] Andy Oram. Research possibilities in peer-to-peer networking. http://www.openp2p.com/lpt/a/1312, 2001.

[46] O'Reilly p2p directory. http://www.openp2p.com/pub/q/p2p_category, 2002.

[47] Petascale virtual-data grids. http://www.griphyn.org/projinfo/intro/petascale.php.

[48] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware*, November 2001.

[49] Jennifer Schopf. Grids: The top ten questions. *Scientific Programming*, 10(2), 2002.

[50] Second IEEE international conference on peer-to-peer computing: Use of computers at the edge of networks (p2p, grid, clusters). www.ida.liu.su/conferences/p2p/p2p2002, September 2002.

[51] Seti@home. http://setiathome.ssl.berkeley.edu.

[52] H. J. Song, X. Liu, D. Jakobsen, R. Bhagwan, X. Zhang, Kenjiro Taura, and Andrew A. Chien. The microgrid: a scientific tool for modeling computational grids. In *Supercomputing*, 2000.

[53] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM '01 Conference*, August 2001.

[54] Ion Stoica, Robert Morris, David Liben-Nowell, David Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. Research report, MIT, January 2002.

[55] Michael Stonebraker, Paul M. Aoki, Robert Devine, Witold Litwin, and Michael A. Olson. Mariposa: A new architecture for distributed data. In *ICDE*, February 1994.

[56] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Research Report UCB/CSD-01-1141, U.C. Berkeley, April 2001.