On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing

Ian Foster [1, 2]   Adriana Iamnitchi [1]

[1] Department of Computer Science, University of Chicago, Chicago, IL 60615

[2] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439

1   Introduction

It has been reported [26] that life holds but two certainties, death and taxes. And indeed, despite much effort devoted to circumventing both phenomena, it does appear that any society—and in the context of this paper, any large-scale distributed system—must address both death (failure) and the establishment and maintenance of infrastructure (which we assert is a major motivation for taxes, so as to justify our chosen title!).

Two supposedly new approaches to distributed computing have emerged in the past few years, both claiming to address the problem of organizing large-scale computational societies: peer-to-peer (P2P) [15, 37, 50] and Grid computing [22]. Both approaches have seen rapid evolution, widespread deployment, successful application, considerable hype, and a certain amount of (sometimes warranted) criticism. The two technologies appear to have the same final objective—the pooling and coordinated use of large sets of distributed resources—but are based in different communities and, at least in their current designs, focus on different requirements.

In this position paper, we take some first steps towards comparing and contrasting P2P and Grid computing. Basing our discussion whenever possible on the characteristics of deployed systems, rather than the unverified claims abundant in the literature, we review their target communities, resources, scale, applications, and technologies. On the basis of this review, we draw some initial conclusions concerning their interrelationship and future evolution. In brief, we argue that (1) both are concerned with the same general problem, namely the organization of resource sharing within virtual communities; (2) both take the same general approach to solving this problem, namely the creation of overlay structures that co-exist with, but need not correspond in structure to, underlying organizational structures; (3) each approach has made genuine technical advances but also has—in current instantiations—crucial limitations, which we characterize (simplistically, but still, we believe, usefully) as "Grid computing addresses infrastructure but not yet failure; P2P addresses failure but not yet infrastructure"; and (4) the complementary nature of the strengths and weaknesses of the two approaches suggests that the interests of the two communities are likely to grow closer over time.

2   Defining Terms

The popularity of both Grid and P2P have led to a large number of (often contradictory) definitions. We assume here that Grids are sharing environments implemented via the deployment of a persistent, standards-based service infrastructure that supports the creation of, and resource sharing within, distributed communities. Resources can be computers, storage space, sensors, software applications, and data, all connected through the Internet and a middleware software layer that provides basic services for security, monitoring, resource management, etc. Resources owned by various administrative organizations are shared under locally defined policies that specify what is shared, who is allowed to share, and under what conditions. We call a set of individuals and/or institutions defined by such sharing rules a virtual organization (VO) [25].

We take as our definition of *P2P* a class of applications that takes advantage of resources—storage, cycles, content, human presence—available at the edges of the Internet [50]. Because accessing these decentralized resources means operating in an environment of unstable connectivity and unpredictable IP addresses, P2P design requirements commonly include independence from DNS and significant or total autonomy from central servers. Their implementations frequently involve the creation of overlay networks [54] with a structure independent of that of the underlying Internet. We opt for this definition rather than, as in the IPTPS announcement, "decentralized, self-organizing distributed systems, in which

all or most communication is symmetric," because it allows us to include large-scale deployed (albeit centralized) "P2P" systems (such as Napster and SETI@home) where much experience has been gained.

3 Comparing Grids and P2P

Current Grids provide many services to moderate-sized communities [20] and emphasize the integration of substantial resources to deliver nontrivial qualities of service within an environment of at least limited trust. For example, NASA's Information Power Grid links supercomputers at four NASA laboratories [27]. In contrast, current P2P systems deal with many more participants (e.g., hundreds of thousands in Gnutella as reported by LimeWire [32]) but offer limited and specialized services, have been less concerned with qualities of service, and have made few if any assumptions about trust.

These characterizations and examples might not suggest commonality of interest. Nevertheless, we shall argue that in fact the two environments are concerned with the same general problem, namely resource sharing within VOs that may not overlap with any existing organization. There are clearly both conceptual and concrete distinctions between the two types of system, which we attempt here to identify and illuminate, focusing as noted above on characteristics of deployed systems. We shall see that the distinctions seem to be the result of different target communities and thus different evolutionary paths. Grids have incrementally scaled the deployment of relatively sophisticated services and application, connecting small numbers of sites into collaborations engaged in complex scientific applications. As system scale increases, Grid developers are now facing and addressing problems relating to autonomic configuration and management. P2P communities developed rapidly around unsophisticated but popular services such as file sharing, and are now seeking to expand to more sophisticated applications as well as continuing to innovate in the area of large-scale autonomic system management. We would expect the definition of persistent and multi-purpose infrastructure to emerge as an important theme.

3.1 Target Communities and Incentives

The development and deployments of **Grid** technologies were motivated initially by the requirements of professional communities needing to access remote resources, federate datasets, and/or pool computers for large-scale simulations and data analyses. Initially developed to address the needs of scientific collaborations, commercial interest is growing. Participants in contemporary Grids thus form part of established communities that are prepared to devote effort to the creation and operation of required infrastructure, and within which exist some degree of trust, accountability, and opportunities for sanctions in response to inappropriate behavior. At the same time, the dynamic nature of VO existence and membership, and the often-limited engagement of VO participants, circumscribe the ability to impose solutions at individual sites (where local VO participants may have only limited authority) and VO-specific administration.

In contrast, **P2P** has been popularized by grass-roots, mass culture (music) file-sharing and highly parallel computing applications [4, 5] that scale in some instances to hundreds of thousands of nodes. The "communities" that underlie these applications comprise diverse and anonymous individuals with little incentive to act cooperatively. Thus, for example, we find that in file sharing applications, there are few providers and many consumers [2]; the operators of SETI@home [49] devote significant effort to detecting deliberately submitted incorrect results; and people tend to intentionally misreport their resources [46]. Thus, rule enforcing mechanisms as well as incentives for good behavior must be provided by the system (more in Section 3.5).

The need of alternative participation models based, for example, on payment, contracts, markets, and licensing is recognized in both systems [29, 31, 55] but is not yet standard practice.

3.2 Resources

In general, **Grid** systems integrate resources that are more powerful, more diverse, and better connected than the "desktop at the edge of the Internet" that constitutes a typical **P2P** resource. A Grid resource might be a cluster, storage system, database, or scientific instrument of considerable value that is administered in an organized fashion according to some well-defined policy. This explicit administration enhances the resource's ability to deliver desired qualities of service and can facilitate, for example,

software upgrades, but can also increase the cost of integrating the resource into a Grid. Diversity in architecture and policy makes the publication of resource properties important [16]. Explicit administration, higher cost of membership, and the stronger community links within scientific VOs mean that resource availability tends to be high and uniform. In contrast, P2P systems deal with intermittent participation and highly variable behavior: for example, in the case of Mojo Nation it is reported [55] that average connection time was only 28% and highly skewed (one sixth of nodes always connected).

Grids do not integrate only "high-end" resources: desktop systems with variable availability [35] form a major component of many contemporary Grids [12]. However, the ensemble of all such resources within a Grid are not treated as an undifferentiated swarm of global scope. Rather, they are aggregated within administrative domains via technologies such as Condor [33, 34] to create local resource pools that are integrated into larger Grids via the same Grid protocols as other computational resources.

Resources in Grids, traditionally from research and educational organizations, tend to be more powerful than home computers that arguably represent the majority of P2P resources (e.g., 71% of SETI@home systems are home computers [4]). The difference in capabilities between home and work computers is illustrated by the average CPU time per work unit in SETI@home: home computers are 30% slower than work computers (13:45 vs. 10:16 hours per work unit).

3.3   Applications

We see considerable variation in the range and scope of scientific **Grid** applications, depending on the interest and scale of the community in question. As three (real, not demonstration) examples, we mention the HotPage portal, providing remote access to supercomputer hardware and software [52]; numerical solution of the long-open 'nug30' quadratic optimization problem using hundreds of computers at many sites [7]; and the NEESgrid system that integrate earthquake engineering facilities into a national laboratory [19].

In contrast, **P2P** systems tend to be vertically integrated solutions to specialized resource sharing problems: currently deployed systems share either compute cycles or files. Diversification comes from differing design goals, such as scalability [41, 44, 51, 56], anonymity [13], or availability [13, 30].

One significant point of differentiation between applications on deployed Grid and P2P systems is that the former tend to be far more data-intensive. For example, a recent analysis of Sloan Digital Sky Survey data [6] involved, on average, 660 MB input data per CPU hour; the Compact Muon Solenoid [36] data analysis pipeline involves from 60 MB to 72 GB input data per CPU hour. In contrast, SETI@home moves at least four orders of magnitude less data: a mere 21.25 KB data per CPU hour. The reason is presumably, in part at least, better network connectivity, which also allows for more flexibility in Grid application design: in addition to loosely coupled applications [1, 10, 12], Grids have been used for numerical simulation [3, 43] and branch-and-bound-based optimization problems [7].

3.4   Scale and Failure

We can measure "scale" in terms of at least two different dimensions: number of participating entities and amount of activity. We discussed above the necessity of dealing with failure—seen as intermittent participation in collaboration—as imposed by resource and community characteristics. Dealing effectively and automatically with failure is both a consequence of and a prerequisite for scaling up in both dimensions.

The community orientation of scientific **Grid** communities means that they often involve only modest numbers of participants, whether institutions (tens), pooled computers (thousands), or simultaneous users (hundreds). For example, the high energy physics collaboration that shares and analyzes data from the D0 Experiment [17] spans 73 institutions in 18 countries, with thousands of scientists involved, of which hundreds access its resources (data and computers) simultaneously. The amount of activity, on the other hand, can be large. For example, during the first half of 2002, about 300 D0 users submitted 2.7 million requests and retrieved 824 TB of data. A consequence of these community characteristics is that early Grid implementations did not address scalability and self-management as priorities. Thus, while the design of core Grid protocols (as instantiated within the Globus Toolkit [21]) does not preclude scalability, actual deployments often employ centralized components. For example, we

find central repositories for shared data, centralized resource management components (such as the Condor Matchmaker [39]), and centralized (and/or hierarchical) information directories. This situation is changing, with much work proceeding on such topics as reliable and scalable management of large job pools, distributed scheduling[40], replica location [11], and discovery [16]. Overall, though, scalable autonomic management remains a goal, not an accomplishment, for Grid computing.

Far larger **P2P** communities exist: millions of simultaneous nodes in the case of file sharing systems [32, 47] and several million total nodes in SETI@home. The amount of activity is also significant, albeit, surprisingly, not always larger than in the relatively smaller-scale Grids: 1-2 TB per day in file sharing systems as of end of 2001 [47], amounting to less than half the data transferred in D0. This large scale has emerged from (and later motivated work on) robust self-management of large numbers of nodes. Over time, P2P systems have evolved from first-generation centralized structures (e.g., Napster index, SETI@home) to second-generation flooding-based (e.g., Gnutella file retrieval) and then third-generation systems based on distributed hash tables. First- and second-generation P2P collaborations have been characterized at the level of both individual nodes (behavior, resources [2, 46]) and network properties (topological properties [42], scale [32], traffic [47]), revealing not only general resilience but also unexpected emergent properties. Third-generation systems have been characterized primarily via simulation studies [41, 51] rather than large-scale deployments. Overall, it is clear that scalable autonomic management has been achieved to a significant extent in P2P, albeit within specific narrow domains.

3.5   Services and Infrastructure

The technologies used to develop Grid and P2P applications differ in both the specific services provided and in the emphasis placed on persistent, multi-purpose infrastructure.

Much work has been expended within the **Grid** community on both technical and organizational issues associated with creating and operating persistent, multi-purpose infrastructure services for authentication [24], authorization [38, 53], discovery [16], resource access, data movement, and so forth. (Perhaps because of the relatively self-contained nature of early Grid communities, less effort has been devoted to managing participation in the absence of trust, via accounting, reputation management, and so forth, although these issues are increasingly discussed.) We use the term *persistent* to indicate that services are operated by participants over extended periods as critical and often highly available infrastructure elements, like DNS servers; and *multi-purpose* to indicate that the same services are used for many different purposes (e.g., the same monitoring and discovery service [16] is used by a wide range of higher-level functions, such as computation scheduling, data replication, and fault detection). These services operate as overlays on resources and services maintained by participating institutions. Gatewaying from these overlay structures to local mechanisms and policies is a significant concern.

Many Grid communities use the open source Globus Toolkit [21] as a technology base. Significant effort has been channeled towards the standardization of protocols and interfaces to enable interoperability between different Grid deployments. The Open Grid Services Architecture (OGSA) [23] is such an effort: it integrates Grid and Web services technologies to define a service-oriented architecture within which all services adhere to a set of standard interfaces and behaviors (some required and some optional) for such purposes as service creation, registry, discovery, lifecycle, service data query, notification, and reliable invocation.

P2P systems have tended to focus on the integration of simple resources (individual computers) via protocols designed to provide specific vertically integrated functionality. Thus, for example, Gnutella defines its own protocols for search and network maintenance. Such protocols do, of course, define an infrastructure, but in general (at least for second- and third-generation systems) the persistence properties of such infrastructures are not specifically engineered but are rather emergent properties. Over time, experience with these emergent properties has revealed the need for new services, such as anonymity and censorship resistance [48], incentives for fair sharing and reputation management [14], and result checking [45]—important issues that have not tended to arise to date in Grid computing due to different underlying trust assumptions.

JXTA [28], XtremWeb [18], and BOINC [9] have been proposed as standard service infrastructure for P2P systems, but to date have seen little adoption and no interoperability. Thus, for example, a user

participating in Gnutella, KaZaA, SETI@home, and FightAIDSatHome must run four completely independent applications on their computer, each coded from scratch and running its own protocols over its own overlay networks. This lack of interest in standard infrastructure may perhaps derive from the simple nature of current P2P applications and the fact that a typical user does not encounter a substantial administrative burden even when running multiple applications. As functionality requirements increase, so presumably will the benefits of standard tooling and infrastructure.

While P2P and Grid service requirements overlap in many regards, there are also important distinctions. First, some services are specific to particular regimes: for example, mechanisms that make up for the inherent lack of incentives for cooperation in P2P. Second, functionality requirements can conflict: for example, Grids might require accountability and P2P systems anonymity. Third, common services may start from different hypotheses, as in the case of trust.

4 Future Directions

Grid and P2P are both concerned with the pooling and coordinated use of resources within distributed communities, and are constructed as overlay structures that operate largely independently of institutional relationships. Yet despite these commonalities, there are also major differences in communities, incentives, applications, technologies, resources, and achieved scale.

Nevertheless, we argue that the vision that motivates both Grid and P2P computing—that of a worldwide computer within which access to resources and services can be negotiated as and when needed—will only come to pass if we are successful in developing a technology that combines important elements of what we today call both P2P and Grid computing. This technology will address *failure* (death) at a fundamental level, using scalable self-configuring protocols such as those emerging from P2P research. It will also provide persistent and multi-purpose *infrastructure* (at some cost, justified because amortized over many uses and users), which like DNS and routing tables will be supported in an organized and distributed fashion and will exploit heterogeneity (whether naturally occurring or artificially imposed) in its environment to achieve goals of robustness, performance, and trust. Diverse discovery, negotiation, and maintenance protocols constructed on some common base will be used to deliver a wide spectrum of services and qualities of service.

The Grid and P2P communities are approaching this nirvana from different directions. Over time, the scale of Grid systems is increasing as barriers to participation are lowered and as commercial deployments enable communities based on purely monetary transactions. For example, the International Virtual Data Grid Laboratory [8] is deploying to scores of sites and many thousands of resources both nationally and internationally, and interest is growing in utility computing models [29] that establish sharing relationships based on commercial transactions rather than common interest. Both trends lead to a greater need for scalability, trust negotiation, self-configuration, automatic problem determination, and fault tolerance [11, 16]—areas where P2P has much to offer. OGSA definition work proceeding within the Global Grid Forum, and early releases of an OGSA-based Globus Toolkit 3, are stimulating much work on service definition and implementation.

Meanwhile, developers of P2P systems are becoming increasingly ambitious in their applications and services, as a result of both natural evolution and more powerful and connected resources. We suspect that the developers of such systems are going to become increasingly interested in standard infrastructure and tools for service description, discovery, and access, as well in standardized service definitions and implementations able to support different mixes of logical and physical organizations.

This analysis suggests to us that the Grid and P2P communities have more in common than is perhaps generally recognized, and that a broader recognition of key commonalities will tend to accelerate progress in both disciplines. Which is why we wrote this paper.

Acknowledgements

References

1. Abramson, D., Sosic, R., Giddy, J. and Hall, B., Nimrod: A Tool for Performing Parameterised Simulations Using Distributed   Workstations. In *Proc. 4th IEEE Symp. on High Performance Distributed Computing*, 1995.

2. Adar, E. and Huberman, B.A. Free Riding on Gnutella. *First Monday*, *5* (10), 2000.

3. Allen, G., Dramlitsch, T., Foster, I., Goodale, T., Karonis, N., Ripeanu, M., Seidel, E. and Toonen, B., Supporting Efficient Execution in Heterogeneous Distributed Computing Environments with Cactus and Globus. In *SC'2001*, 2001.

4. Anderson, D.P., Cobb, J., Korpella, E., Lebofsky, M. and Werthimer, D. SETI@home: An Experiment in Public-Resource Computing. *Communications of the ACM*, *45* (11). 56-61, 2002.

5. Anderson, D.P. and Kubiatowicz, J. The Worldwide Computer. *Scientific American* (3), 2002.

6. Annis, J., Zhao, Y., Voeckler, J., Wilde, M., Kent, S. and Foster, I., Applying Chimera Virtual Data Concepts to Cluster Finding in the Sloan Sky Survey. In *SC'2002*, 2002.

7. Anstreicher, K., Brixius, N., Goux, J.-P. and Linderoth, J.T. Solving Large Quadratic Assignment Problems on Computational Grids. *Mathematical Programming*, *91* (3). 563-588, 2002.

8. Avery, P., Foster, I., Gardner, R., Newman, H. and Szalay, A., *An International Virtual-Data Grid Laboratory for Data Intensive Science*, Technical Report GriPhyN-2001-2, 2001.

9. Berkeley Open Infrastructure for Network Computing.

10. Casanova, H., Obertelli, G., Berman, F. and Wolski, R., The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid. In *Proc. SC'2000*, 2000.

11. Chervenak, A., Deelman, E., Foster, I., Guy, L., Hoschek, W., Iamnitchi, A., Kesselman, C., Kunszt, P., Ripeanu, M., Schwartzkopf, B., Stockinger, H., Stockinger, K. and Tierney, B., Giggle: A Framework for Constructing Scalable Replica Location Services. In *SC'02*, 2002.

12. Chien, A., Calder, B., Elbert, S. and Bhatia, K. Entropia: Architecture and Performance of an Enterprise Desktop Grid System. *Journal of Parallel and Distributed Computing*, *To appear*,

13. Clarke, I., Sandberg, O., Wiley, B. and Hong, T.W., Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *International Workshop on Designing Privacy Enhancing Technologies*, Berkeley, CA, USA, 2000.

14. Cornelli, F., Damiani, E., Capitani, S.D., Paraboschi, S. and Samarati, P., Choosing reputable servents in a P2P network. In *International World Wide Web Conference*, Honolulu, Hawaii, USA, 2002.

15. Crowcroft, J. and Pratt, I., Peer to Peer: peering into the future. In *IFIP-TC6 Networks 2002 Conference*, Pisa, Italy, 2002.

16. Czajkowski, K., Fitzgerald, S., Foster, I. and Kesselman, C., Grid Information Services for Distributed Resource Sharing. In *10th IEEE International Symposium on High Performance Distributed Computing*, 2001.

17. The DZero Experiment. *http://www-d0.fnal.gov/*.

18. Fedak, G., Germain, C., Néri, V. and Cappello, F., XtremWeb : A Generic Global Computing System. In *Workshop on Global Computing on Personal Devices (CCGRID2001)*, Berlin, Germany, 2001.

19. Finholt, T.A., Wierba, E.E., Birnholtz, J.P. and Hofer, E., *NEESgrid User Requirements*, Technical Report NEESgrid-2002-xx, 2002.

20. Foster, I. The Grid: A New Infrastructure for 21st Century Science. *Physics Today*, *55* (2). 42-47, 2002.

21. Foster, I. and Kesselman, C. Globus: A Toolkit-Based Grid Architecture. in Kesselman, C. ed. *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999, 259-278.

22. Foster, I. and Kesselman, C. (eds.). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.

23. Foster, I., Kesselman, C., Nick, J. and Tuecke, S., *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, Technical Report Globus Project, 2002.

24. Foster, I., Kesselman, C., Tsudik, G. and Tuecke, S. A Security Architecture for Computational Grids. in *ACM Conference on Computers and Security*, 1998, 83-91.

25. Foster, I., Kesselman, C. and Tuecke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, *15* (3). 200-222, 2001.

26. Franklin, B., In a letter to Jean-Baptiste Leroy, 1789.

27. Johnston, W.E., Gannon, D. and Nitzberg, B., Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid. In *8th IEEE Symposium on High Performance Distributed Computing*, 1999.

28. JXTA. *www.jxta.org*.

29. Kenyon, C. and Cheliotis, G., Architecture Requirements for Commercializing Grid Resources. In *11th IEEE International Symposium on High Performance Distributed Computing*, 2002.

30. Kubiatowicz, J., Bindel, D., Chen, Y., Czerwinski, S., Eaton, P., Geels, D., Gummadi, R., Rhea, S., Weatherspoon, H., Weimer, W., Wells, C. and Zhao, B., OceanStore: An Architecture for Global-Scale Persistent Storage. In *Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000)*, 2000.

31. Lai, C., Medvinsky, G. and Neuman, B.C. Endorsements, Licensing, and Insurance for Distributed System Services. in *Proc. 2nd ACM Conference on Computer and Communication Security*, 1994.

32. Limewire. *http://www.limewire.com*.

33. Litzkow, M., Livny, M. and Mutka, M. Condor - A Hunter of Idle Workstations. in *Proc. 8th Intl Conf. on Distributed Computing Systems*, 1988, 104-111.

34. Livny, M. High-Throughput Resource Management. in Kesselman, C. ed. *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999, 311-337.

35. Mutka, M. and Livny, M. The Available Capacity of a Privately Owned Workstation Environment. *Performance Evaluation*, *12* (4). 269--284, 1991.

36. Negra, M.D., *CMS Collaboration*, Technical Report CERN LHCC 94-38, CERN, 1994.

37. Oram, A. (ed.), *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly, 2001.

38. Pearlman, L., Welch, V., Foster, I., Kesselman, C. and Tuecke, S., A Community Authorization Service for Group Collaboration. In *IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, 2002.

39. Raman, R., Livny, M. and Solomon, M., Matchmaking: Distributed Resource Management for High Throughput Computing. In *IEEE International Symposium on High Performance Distributed Computing*, 1998.

40. Ranganathan, K. and Foster, I., Decoupling Computation and Data Scheduling in Distributed Data Intensive Applications. In *International Symposium for High Performance Distributed Computing (HPDC-11)*, Edinburgh, UK, 2002.

41. Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S., A Scalable Content-Addressable Network. In *SIGCOMM Conference*, 2001.

42. Ripeanu, M., Foster, I. and Iamnitchi, A. Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design. *Internet Computing*, *6* (1). 50-57, 2002.

43. Ripeanu, M., Iamnitchi, A. and Foster, I. Performance Predictions for a Numerical Relativity Package in Grid Environments. *International Journal of High Performance Computing Applications*, *15* (4), 2001.

44. Rowstron, A.I.T. and Druschel, P., Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Middleware*, 2001.

45. Sarmenta, L.F.G. Sabotage-tolerance mechanisms for volunteer computing systems. *Future Generation Computer Systems*, *18* (4). 561 - 572, 2002.

46. Saroiu, S., Gummadi, P.K. and Gribble, S.D., A Measurement Study of Peer-to-Peer File Sharing Systems. In *Proceedings of Multimedia Computing and Networking (MMCN)*, San Jose, CA, USA, 2002.

47. Sen, S. and Wang, J., Analyzing Peer-to-Peer Traffic Across Large Networks. In *Internet Measurement Workshop*, Marseille, France, 2002.

48. Serjantov, A., Anonymizing Censorship Resistant Systems. In *1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*. Cambridge, MA, USA, 2002.

49. SETI@home: The Search for Extraterrestrial Intelligence. *http://setiathome.berkeley.edu*.

50. Shirky, C. What is P2P... and what isn't? *http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html*.

51. Stoica, I., Morris, R., Karger, D., Kaashoek, F. and Balakrishnan, H., Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *SIGCOMM Conference*, San Diego, CA, USA, 2001.

52. Thomas, M.P., Mock, S. and J., B., Development of Web Toolkits for Computational Science Portals: The NPACI HotPage. In *Ninth IEEE International Symposium on High Performance Distributed Computing*, 2000.

53. Thompson, M., Johnston, W., Mudumbai, S., Hoo, G., Jackson, K. and Essiari, A., Certificate-based Access Control for Widely Distributed Resources. In *8th Usenix Security Symposium*, 1999.

54. Touch, J. Overlay networks. *Computer Networks*, *3* (2-3). 115-116, 2001.

55. Wilcox-O'Hearn, B., Experiences Deploying A Large-Scale Emergent Network. In *1st International Workshop on Peer-to-Peer Systems   (IPTPS'02)*, Cambridge, MA, USA, 2002.

56. Zhao, B.Y., Kubiatowicz, J.D. and Joseph, A.D., *Tapestry: An infrastructure for fault-tolerant wide-area location and routing*, Technical Report CSD-01-1141, UC Berkeley, 2001.