

# Distributed Computing Systems: P2P versus Grid Computing Alternatives

A. Cabani\*, S. Ramaswamy+, M. Itmi\*, S. Al-Shukri+, J.P. Pécuchet\*

**Abstract** -Grid and P2P systems have become popular options for large-scale distributed computing, but their popularity has led to a number of varying definitions that are often conflicting. Taxonomies developed to aid the decision process are also quite limited in their applicability. While some researchers have argued that the two technologies are converging [1], in this paper, we develop a unified taxonomy along two necessary distributed computing dimensions and present a framework for identifying the right alternative between P2P and Grid Computing for the development of distributed computing applications.

## 1. INTRODUCTION

A distributed computing system is defined as a collection of independent computers that appear to their users as a single computing system [2]. Distributed software systems are increasingly being used in modern day software systems development to tackle the issues of geographically separated work groups and increasing application complexities. Often this constitutes the interconnection of autonomous software residing on individual machines through communication networks to enable their users to cooperate and coordinate to successfully accomplish their objectives.

The widespread need for a distributed system based solution is due to the need for resource sharing and fault-tolerance. Resource sharing implies that a distributed system allows its resources - hardware, software and data - to be appropriately shared amongst its users. Fault-tolerance means that machines connected by networks can be viewed as redundant resources, a software system could be installed on multiple machines to withstand hardware faults or software failures [3].

For a distributed system to support active resource sharing and fault-tolerance within its multitude of nodes, it needs to possess certain key properties. These include openness and transparency. Openness in a distributed system is achieved by specifying its key interface elements and making it available to other software developers so that the system can be extended for use. Distributed systems generally tend to provide three forms of transparency. These include: (i) Location transparency, which allows local and remote information to be accessed in a unified way; (ii) Failure

transparency, which enables the masking of failures automatically; and (iii) Replication transparency, which allows duplicating software/data on multiple machines invisibly.

To be able to demonstrate the above properties, in turn such a distributed system must provide support for concurrency and be built on a scalable architectural framework. Concurrency refers to the simultaneous processing of requests to multiple interconnected machines / networks. Scalability refers to the adoption of an interconnection network architecture that allows for seamless extendibility to a large number of machines and/or users to support the needs of increased processing power requirements.

In this paper, we compare and contrast two currently popular approaches for distributed computing applications: Grid and P2P approaches. The objective of both P2P and grid computing is the collective, coordinated use of a large number of resources scattered in a distributed environment. However the user communities that have adopted and popularized these two approaches are vastly different, both in terms of their user-level requirements as well as the architectural design of the systems themselves.

This paper is organized as follows: Section 2 and 3 present a brief overview of Grid and P2P computing. Section 4 presents the unified taxonomy along two necessary distributed computing dimensions. Section 5 compares and contrasts Grid and P2P computing using a set of commonly desired criteria for a distributed computing solution. Section 6 concludes the paper.

## 2. GRID COMPUTING

According to IBM's definition [4]: "A grid is a collection of distributed computing resources available over a local or wide area network that appear to an end user or application as one large virtual computing system. The vision is to create virtual dynamic organizations through secure, coordinated resource-sharing among individuals, institutions, and resources. Grid computing is an approach to distributed computing that spans not only locations but also organizations, machine architectures and software boundaries to provide unlimited power, collaboration and information access to everyone connected to a grid."

Another definition, this one from The Globus Alliance is [5]: "The grid refers to an infrastructure that enables the integrated, collaborative use of high-end computers, networks, databases, and scientific instruments owned and managed by multiple organizations. Grid applications often involve large amounts of data and/or computing and often require secure resource sharing across organizational

\* LITIS Laboratory EA4051, INSA-Rouen, 76131 Mt. St-Aignan Cedex, France. This work was supported, in part, by INSA-Rouen - for supporting Dr. Ramaswamy as a visiting professor at INSA during June 2006.

Email: [adnane.cabani@insa-rouen.fr](mailto:adnane.cabani@insa-rouen.fr) / [itmi@insa-rouen.fr](mailto:itmi@insa-rouen.fr) / [jean-pierre.pecuchet@insa-rouen.fr](mailto:jean-pierre.pecuchet@insa-rouen.fr)

+ Knowledge Enterprises for Scalable Resilient Infrastructures (KESRI), Computer Science Department, Univ. of Arkansas at Little Rock, AR 72204 USA. This work was supported, in part, by an NSF MRI Grant #CNS-0619069 and the Graduate Institute of Technology at UALR.

Email: [srini@acm.org](mailto:srini@acm.org) / [shalshukri@ualr.edu](mailto:shalshukri@ualr.edu)

boundaries, and are thus not easily handled by today's Internet and Web infrastructures."

Grid computing evolved out of the scientific realm where there was a need to process and analyze increasingly colossal quantities of data such as those needed to perform weather or climatic forecasts [6, 7], to model and calculate the aerodynamic behavior of a plane, in genomics [8], etc. Such application specific needs were not only at odds with the wait times for technological evolution but also were highly intolerant to the advances attained by means of Moore's law [9] with respect to performance (doubling every eighteen months), storage (doubling every twelve months), and networks (double every nine months). Since the speed of network performance outpaces that of processors and storage, it has led to the increasing interconnection of geographically scattered resources by means of a dynamic network that accumulates the capacities of calculation, storage, etc.

Grid computing [10-12] thus allows for better resource sharing between users in an institution to resolve a common problem. From a business perspective, the purpose of grid computing is to minimize the time to market, thereby profiting from the infrastructure costs incurred. Grid computing offers its users' access to processing across diverse storage structures which are transparently distributed geographically. It is based on the concept of on demand data processing wherein a user pays according to their needs and resource consumption.

### 3. PEER-TO-PEER COMPUTING

As contrasted from grid computing, peer-to-peer computing refers to a network of equals that allows two or more individuals to spontaneously collaborate without necessarily needing any centralized coordination [13]. P2P computing was made famous by a music sharing frenzy generated by Napster, which was initially a server based centralized architecture. Other P2P systems have since appeared without the limitation imposed by a centralized server. Here, a user seeking a file (song, video, software) sends their query which is incrementally forwarded by the nodes of the network, thereby creating an ad-hoc chain between the requester's PC requester to the supplier's PC – culminating in the transfer of the requested file. Examples include Limewire [14], Kazaa [15], eDonkey [16], BitTorrent [17]; which are some popular P2P systems. When designing P2P applications, it is important to assume that peers are untrustworthy [18]. While they are designed to interconnect and communicate with each other they can join or quit dynamically from the P2P network. When a node quits, there will be communication failures. This makes the development of P2P applications a very challenging task.

While P2P technology can be applied to many application domains, most current utilization is customer targeted with the primary focus on file sharing. These systems allow files to

be easily shared and quickly propagated through the Internet without powerful host servers. Other applications include:

1. Personal productivity applications: Collaboration between individual users, i.e. sharing address books, schedules, notes, chatting, etc. allows improvements in productivity. Connecting such desktop productivity software systems together enables collaborative e-business communities to form for flexible, productive, and efficient working teams. For example, Java developers have used OpenProjects.net to collaborate. On a broader scale, hundreds of thousands of users use instant messaging, one of the most popular P2P applications to date.
2. Enterprise resource management: These systems allow the coordination of workflow processes within an organization thereby leveraging the existing network infrastructure for improvements in business productivity. For example, Groove [19] enables an aerospace manufacturers to post job order requests to partner companies and route the completed requests from one department to the next.
3. Distributed computation: A natural extension of the Internet's philosophy of robustness through decentralization is to design peer-to-peer systems that send computing tasks to millions of servers, each one possibly also being a desktop computer.

In [20, 21] the authors present a taxonomy for P2P applications and distinguish three specific categories of P2P applications. The specific classes include:

1. Parallel applications: In such applications, a large calculation is split into several small independent entities that can be executed independently on a large number of peers - SETI@Home [22], genome@Home, etc. Another possibility is making the calculation of the same operation but with different data sets or parameters. Such computation kind called a parametric study computation. Example: fluid dynamics simulation. The goal is to solve computational problems and cycle-sharing. P2P cycle-sharing and grid computing are converged but its origins are different [11, 23]. In P2P cycle-sharing the whole application runs on each peer and no need communication between the peers.
2. Content and file management: Content encapsulates several types of activities and refers to anything that can be digitized; for example, messages, files, binary software. It essentially consists of storing, sharing and finding various kinds of information on the network. The main application focus is content exchange. Such projects aim to establish a system of files for distribution within a community

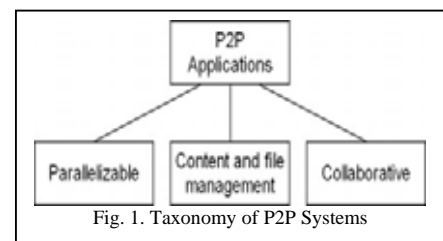


Fig. 1. Taxonomy of P2P Systems

and examples include CAN [24] and Chord [25]. Other such applications are in distributed databases and distributed hash tables.

3. Collaborative: Collaborative P2P applications allow users to collaborate, in real time, without relying on a central server to collect and relay information. Such applications are characterized by ongoing interactions and exchanges between peers. Typical applications include: instant messaging (AOL, YM!). Some games (DOOM) have also adopted the P2P framework.

#### 4. A REVISED & UNIFIED TAXONOMY

There are several problems with the taxonomy presented in Fig. 1. While the taxonomy is very simplistic and coarse, there are three specific drawbacks. These include:

1. First, it lumps all parallel applications together into one classification. It is not in sufficient detail to help distinguish the application driven needs for distribution. Parallel applications may involve the distribution of the application itself over multiple nodes – as in cluster computing - to the locations where data is stored, or vice-versa – the distribution of the data elements to the applications reside – as in Grid computing. Computationally both of these types lead to completely different costs if not evaluated effectively. For the purposes of this paper, we concentrate on the distribution of the data elements.
2. A second issue is the need (or the lack of need) for synchronization. This issue is completely ignored in the present taxonomy. In fact, not all types of parallelization suit P2P applications. Whether the nodes are tightly coupled or loosely coupled is a very important criteria for the choice (or rejection) of a P2P solution. For tightly coupled applications, P2P is a bad implementation choice.
3. Another highly related issue that is ignored by this taxonomy is the issue of bandwidth disparity between member nodes. Certain studies have [26, 27] showed that P2P systems are extremely heterogeneous, but Grid computing systems tend to be more homogeneous in the composition of their nodes.

For these reasons, we propose a more well-refined and unified taxonomy as shown in Figure 2. This taxonomy is clearly defined

along two dimensions. The first dimension relates to the requirements and capabilities of the underlying infrastructure, these include the architecture, the application domain and the high level of interaction required between the nodes. The second dimension relates to the various applicative constraints that are ignored by the existing taxonomy. While these criteria are independent by themselves, as seen from Figure 2, from the perspective of an application they are highly interdependent. These criteria include:

- Interconnectivity: P2P networks distinguish themselves by the presence of a volatile connectivity. Every peer can join or quit the system without any notice. Therefore, tightly or loosely coupled have an influence direct on expected results and techniques used for synchronization are not the same [28].
- Data size: The rate and size of data that is transferred between the nodes is an important discriminator for making an appropriate choice. Existing optimization techniques on P2P networks do not adequately address large scale keyword searches since the bandwidth required by such searches exceeds the internet’s available capacity. Additionally, P2P-based systems thrive on low latency.
- Bandwidth: Available bandwidth is another critical criterion that needs to be considered.

This new taxonomy is helpful: the choice between grid computing and P2P is easier for developers. This requires three steps:

- After defining his application’s objective, a developer chooses the application’s kind (multicomputers, content and file management or collaborative). At this stage, the first dimension (requirements and capabilities) is fixed.

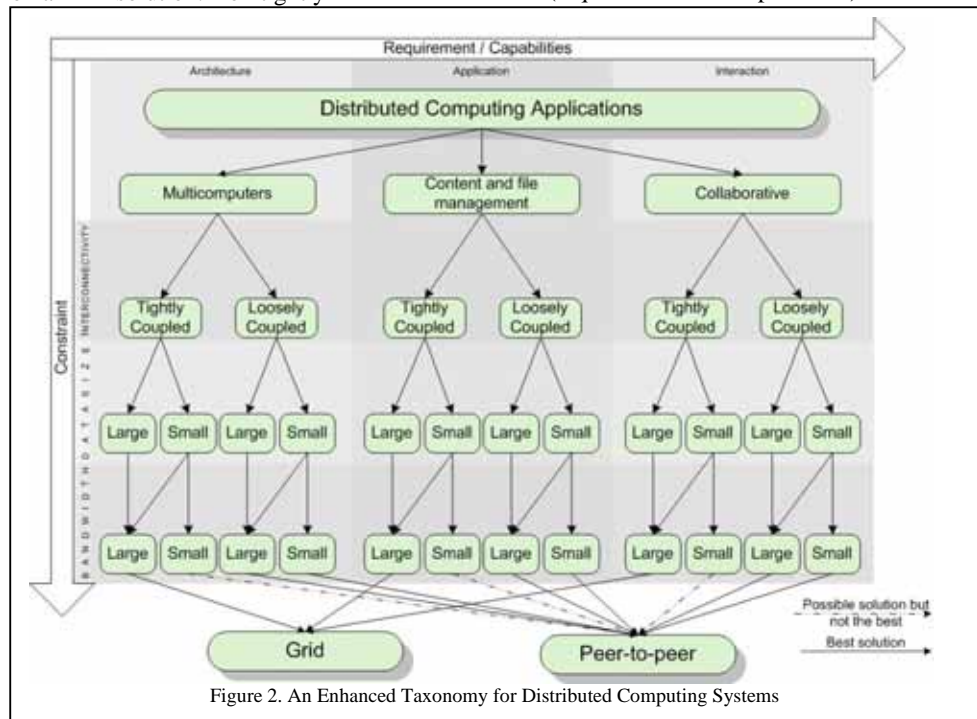


Figure 2. An Enhanced Taxonomy for Distributed Computing Systems

- Taking into account the three above criteria, he/she determines the application's specifications.
- Then, he/she chooses the adopted technology: grid computing or P2P.

#### 5. GRID VERSUS P2P COMPUTING: A FEATURE-BASED ANALYSIS

While it is clear that Grid and P2P computing are two promising approaches to distributed computing, they are very different and their differences are often misunderstood [29]. In a pure P2P system clients and servers work together and are indistinguishable from one another, this is not the case with Grid computing. One of the main P2P characteristics is that once the initial step is completed, data exchange is strictly and directly between peers. This property is completely absent in grid computing. Recently, P2P based Grid Computing systems that combine the advantages offered by P2P systems to scale up grid-based distributed computing systems have been proposed. Such systems enable the creation of a PC based grid architecture that address data overload problems caused by an enormous amount of access from clients by allowing for 'localized' data sharing by PCs through P2P communication mechanisms [30]. However, this may be largely dictated by several application domain characteristics. Hence in this section, we compare and contrast Grid and P2P computing using a set of commonly desired features of distributed computing solution. In [31] a similar attempt to identify and characterize the differences has also been reported – the characteristics include population, ownership discovery, user and resource management, resource allocation & scheduling, interoperability, single system image, scalability, capacity, throughput speed (lat. bandwidth). Figure 3 compares and contrasts P2P and Grid computing using several technical and economic characteristics. Some of the notable ones include the following:

1. *Decentralization*: Decentralization allows for flexibility and unlike client-server systems, does not suffer from single points of failure, wherein the server quickly becomes a bottleneck. Hence distributed systems that are scalable and resilient in unstable environments are very important. Decentralization allows us to move resources closer to where are accessed thereby decreasing response times and reducing, or even eliminating, network latency. This also allows better utilization of network capacity. Features that support decentralization include: Distribution of control, complete local autonomy and the ability orchestrate dynamic on the fly interactions.
2. *Cost and efficiency*: Performance-wise, networks are evolving at a faster rate than hardware and more PCs are connected to Web via broadband networks. This allows for better exploitation of these resources that were previously unrecognized. This has led to greatly increasing three important parameters in modern computing: storage, bandwidth and computing resources. This takes more sense if we know that it becomes easier to interconnect hundreds of million computers worldwide to

form a network with the global revolution of the internet. Robert Metcalfe formulated an empirical law allowing measuring the utility of a network. Utility of a network =  $k \times N^2$ . In 1999, the Reed Law [32] adds a human dimension to the technological dimension. The utility of large networks, particularly social networks, can scale exponentially with the size of the network. The reason for this is that the number of possible sub-groups of network participants is  $2^N - N - 1$ , where N is the number of participants. All of these laws prove that number of connection peers is very important for increasing utility of a network. In P2P systems the number of nodes can reach hundreds of millions allowing increased network utility.

3. *Pervasive computing*: With P2P systems, it's possible to connect any machine with processor to network (PDA, cell phone, GPS...). That is there is a need for heterogeneous computing that is flexible enough to support new communication protocols for exchange of information in support of pervasive computing needs. Some work has been done on developing conceptual models for data management in pervasive computing environments based on cross-layer interaction between data management and communication layers [33].
4. *Target communities and incentives*: Although Grid technologies were initially developed to address the needs of scientific collaborations; commercial interest is growing [1]. 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. In contrast, P2P has been popularized by grassroots, mass-culture (music) file-sharing and highly parallel computing applications [14, 34] 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.
5. *Resources*: In general, Grid systems integrate resources that are more powerful, more diverse, and better connected than the typical P2P resource [1]. 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.
6. *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 [1].
7. *Keyword Searching*: Current P2P searching techniques in unstructured (examples: Gnutella and KaZaa [15, 35]) and structured P2P systems (examples: CAN [24], Chord [25], Pastry [36], Tapestry [37], and SkipNet [38]) include query flooding and inverted list intersection. In [39] the authors present a summary of techniques for unstructured P2P networks, while in [40] the authors present a search technique based on VSM (Vector Space Model) and LSI (Latent Semantic Indexing) for structured P2P systems. In

[41], the authors identify storage and bandwidth as two limiting constraints on full text keyword searching on P2P systems. They suggest the use of a combination of optimizations and compromises to make this P2P searching feasible. Some hybrid schemes[42, 43] have been proposed for digital libraries but are not directly applicable to P2P systems. In [44] the authors propose a hybrid index multi-level partitioning scheme on top of structured P2P networks and indicate achieving a good tradeoff between partition-by-keyword and partition-by-document schemes.

8. *Scale, Security and failure:* Scalable autonomic management clearly has been achieved to a significant extent in P2P, albeit within specific narrow domains [1]. In [45] the authors analyze Gnutella's P2P topology graph and evaluate generated network traffic. They suggest that P2P systems must exploit particular distributions of query values and locality in user interests. They also suggest replacing traditional query flooding mechanisms with smarter and less expensive routing and/or group communication mechanisms. They report that Gnutella follows a multi-modal distribution, combining a power law and a quasi-constant distribution. Which makes the network as reliable as a pure power-law network when assuming random node failures, and makes it harder to attack by a malicious adversaries.
9. *Consistency Management:* Current P2P systems are based predominantly on sharing of static files. However, for using peer-to-peer networks in grid computing systems they will need to support sharing of files that are frequently modified by their users. Consistency has been studied for web caching [46, 47]. In [48] the authors present three techniques for consistency management in P2P systems: push (owner initiated) and pull (client initiated) and a hybrid push and adaptive pull technique.
10. *Services and infrastructure:* P2P systems have tended to focus on the integration of simple resources (individual computers) via protocols designed to provide specific vertically integrated functionality.

## 6. CONCLUSIONS

In this paper our major contribution is an expanded taxonomy for the classification of distributed computing systems and highlighting commonalities and differences between Grid and P2P computing alternatives using a set of commonly desired features. Using these features, we have clearly identified and clarified issues to be addressed and the appropriate selection of the right alternative between P2P versus Grid Computing solution for the development of distributed computing applications.

## 7. REFERENCES

- [1] I. Foster and A. Iamnitchi, "On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing," *Lecture Notes in Computer Science*, vol. 2735, pp. 118-128, 2003.
- [2] A. S. Tanenbaum, *Distributed Operating Systems*: Prentice Hall, 1994.
- [3] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*: Addison-Wesley, 2001.
- [4] L.-J. Zhang, J.-Y. Chung, and Q. Zhou, "Developing Grid computing applications, Part 1 : Introduction of a grid architecture and toolkit for building grid solutions," 2005.
- [5] "Globus Alliance," <http://www.globus.org>.
- [6] "Earth System Grid," <http://www.earthsystemgrid.org/>.
- [7] "LEAD," <https://portal.leadproject.org/gridsphere/gridsphere>.
- [8] "Genome@home, distributed computing," <http://genomeathome.stanford.edu/>.
- [9] R. Hiremane, "From Moore's Law to Intel Innovation - Prediction to Reality," in *Technology@Intel Magazine*, 2005.
- [10] M. D. Stefano, *Distributed Data Management for Grid Computing*: Wiley InterScience, 2005.
- [11] A. Iamnitchi and I. Foster, "A Peer-to-Peer Approach to Resource Location in Grid Environments," in *Symp. on High Performance Distributed Computing*, 2002.
- [12] I. Foster and C. Kesselman, *The Grid, Blueprint for a New Computing Infrastructure*: Morgan Kaufmann, 1998.
- [13] D. Schoder and K. Fischbach, "Peer-to-peer prospects," *Commun. ACM*, vol. 46, pp. 27-29, 2003.
- [14] "Limewire," <http://www.limewire.com/>.
- [15] "Kazaa," <http://www.kazaa.com/>.
- [16] "eDonkey," <http://prdownloads.sourceforge.net/pdonkey/>.
- [17] "BitTorrent," <http://www.bittorrent.com/protocol.html>.
- [18] M. Senior and R. Deters, "Market Structures in Peer Computation Sharing," in *Second International Conference on Peer-to-Peer Computing (P2P'02)*, Linköping, Sweden, 2002, pp. 128-135.
- [19] "Groove," <http://www.groove.net>.
- [20] D. S. Milojevic, V. Kalogeraki, R. Lukose, K. Nagaraja, J.

Feature / Criteria	Grid Computing	P2P Computing
<b>G E N E R A L C R I T E R I A</b>		
Goal	Virtual organization	Virtual system
Role of entities	Grid server	Peer both server and client
Number of entities	10 – 1000 users	Millions of users
Node	Dedicated	Not dedicated
Class	Pre-organized	Self-organized
<b>T E C H N I C A L C R I T E R I A</b>		
Structure	Static hierarchical	Fully distributed and dynamic
Fully decentralized	No	Yes
End-to-end connectivity	No	Yes
Scalability	Limited	Unlimited
Control mechanism	Central	Fully distributed
Connectivity	Static high speed	In/out anytime
Availability	High	Volatile
Failure risk	High	Low
Resources	More powerful	Less powerful
Resources discovery	Static central registration in a hierarchical fashion	Limited addition of a new peer on the network
Location transparency	Yes	No
Ad-hoc formation	No	Yes
<b>E C O N O M I C C R I T E R I A</b>		
Communities	Established communities (closed)	Anonymous individuals (opened)
Participants	Registered	Voluntary
Reliability	Guaranteed trust	Partially (no trust peers)
Standards	Yes	No
Security	Secure	Insecure
Applications	Scientific – Data intensive	Compute cycles or file sharing

Fig. 3. A Comparative Criteria Driven Evaluation

- Prune, B. Richard, S. Rollins, and Z. Xu, "Peer-to-Peer Computing," HP, External HPL-2002-57, 2002.
- [21] D. Barkai, "Technologies for sharing and collaborating on the Net," in *First International Conference on Peer-to-Peer Computing (P2P'01)*, 2001, pp. 13-28.
- [22] "SETI@home," <http://setiathome.ssl.berkeley.edu/>.
- [23] D. Talia and P. Trunfio, "Toward a synergy between P2P and grids," in *IEEE Internet Computing*, vol. 7, 2003, pp. 96, 94 - 95.
- [24] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *ACM SIGCOMM 2001*, San Diego, California, United States, 2001, pp. 161-172.
- [25] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications," in *ACM SIGCOMM 2001*, San Diego, California, United States, 2001, pp. 149-160.
- [26] S. Saroiu, K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Multimedia Computing and Networking*, San Jose, California, United States, 2002.
- [27] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *ACM SIGOPS Operating Systems Review*, vol. 35, pp. 131-145, 2001.
- [28] V. K. Garg, *Concurrent and Distributed Computing in Java*: John Wiley & Sons, 2004.
- [29] D. Barkai, *Peer-to-Peer Computing: Technologies for Sharing and Collaborating on the Net*: Intel Press, 2001.
- [30] H. Sunaga, T. Oka, K. Ueda, and H. Matsumura, "P2P-Based Grid Architecture for Homology Searching," in *Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05) - Volume 00*, 2005, pp. 148-149.
- [31] R. Buyya, "Convergence Characteristics for Clusters, Grids, and P2P networks," in *2nd Intl Conf. on Peer-to-Peer Comp.* Linköping, Sweden, 2002.
- [32] "ReedsLaw," <http://www.reed.com/Papers/GFN/reedslaw.html>.
- [33] F. Perich, "On Peer-to-Peer Data Management in Pervasive Computing Environments," UMBC, 2004.
- [34] D. Abramson, R. Sasic, J. Giddy, and B. Hall, "Nimrod: A Tool for Performing Parameterized Simulations Using Distributed Workstations," in *Fourth IEEE International Symposium on High Performance Distributed Computing (HPDC-4 '95)*, 1995, p. 112.
- [35] "Gnutella," <http://www.gnutella.com>.
- [36] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large scale peer-to-peer systems", IFIP/ACM Middleware. 2001. ." *Lecture Notes in Computer Science*, pp. 329-351, 2218.
- [37] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," UC Berkeley, Tech. Report UCB/CSD-01-1141 Apr. 2001.
- [38] N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman, "SkipNet: A Scalable Overlay Network with Practical Locality Properties," in *4th USENIX Symp. on Internet Technologies and Systems (USITS '03)*, Seattle, WA, USA, 2003.
- [39] B. Yang and H. Garcia-Molina, "Efficient Search in P2P Networks," in *22nd IEEE International Conference on Distributed Computing Systems (IEEE ICDCS'02)* Vienna, Austria: Computer Society, 2002.
- [40] C. Tang, Z. Xu, and M. Mahalingam, "PeerSearch: Efficient information retrieval in peer-to-peer networks," in *Proceedings of HotNets-I, ACM SIGCOMM*, 2002.
- [41] J. Li, B. T. Loo, J. Hellerstein, F. Kaashoek, D. R. Karger, and R. Morris, "On the Feasibility of P2P Web Indexing and Search," in *2nd Intl. Workshop on P2P Systems*, 2003.
- [42] O. Sornil and E. A. Fox, "Hybrid partitioned inverted indices for large-scale digital libraries," in *Proc. of the 4th Int. Conf. of Asian Digital Libraries*, India, 2001.
- [43] C. Badue, R. Baeza-Yates, B. Ribeiro-Neto, and N. Ziviani, "Distributed query processing using partitioned inverted files," in *9th String Proc. and Info. Ret. Symp. (SPIRE)*, 2002.
- [44] S. Shi, G. Yang, D. Wang, J. Yu, S. Qu, and M. Chen, "Making Peer-to-Peer Keyword Searching Feasible Using Multi-level Partitioning," in *3rd Int. Workshop on Peer-to-Peer Systems*, San Diego, USA, 2004.
- [45] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design," *IEEE Internet Computing Journal*, vol. 6, 2002.
- [46] V. Duvvuri, P. Shenoy, and R. Tewari, "Adaptive Leases: A Strong Consistency Mechanism for the World Wide Web," in *Proc. of the IEEE Infocom'00*, Israel, 2000.
- [47] J. Yin, L. Alvisi, M. Dahlin, and C. Lin, "Hierarchical Cache Consistency in a WAN," in *Proc. of the USENIX Symp. on Internet Technologies*, Boulder, CO, 1999.
- [48] J. Lan, X. Liu, P. Shenoy, and K. Ramamritham, "Consistency Maintenance in Peer-to-Peer File Sharing Networks," in *Proc. of 3rd IEEE Workshop on Internet Apps*, 2002.