Invited Paper:

Peer-to-Peer Systems: The Present and the Future

Jorge Ardenghi and Javier Echaiz

Laboratorio de Investigación en Sistemas Distribuidos (LISiDi)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur – Bahía Blanca, Argentina
T.E.: +54 291–4595135 Fax: +54 291–4595136
{jra,je}@cs.uns.edu.ar

ABSTRACT

Nowadays *Peer-to-Peer* (P2P) systems became an important part of Internet, millions of users have been attracted to use their structures and services. The popularity of Peer-to-Peer systems speed up academic research joining researchers from systems, networking and theory. The most popular P2P applications support file-sharing and content distribution, new applications are emerging in different fields, Internet telephony is an example.
This paper discusses the issues of P2P systems such as characteristics, structures, protocols, drawbacks, open problems and futures fields of development.

Keywords: distributed systems, peer-to-peer, algorithms, performance design, grid computing,peer-to-peer.

1. INTRODUCTION

During years and today the client-server paradigm is the *battle horse* of the most users applications. In the last years there is a new paradigm that is emerging, *peer-to-peer* (P2P) mainly supporting applications providing file-sharing, content exchange like music, movies and programs, but have also successfully implemented distributing computing and Internet-based telephony. A refined definition of the ***Peer-to-Peer*** is : "***A Peer-to-Peer [P2P] systemm is a self organizing system of equal, autonomous entities (peers) which aims for the shared usage of distributed resources in networked environment avoiding central services***"[21]. It is possible to say that peer-to-peer is a system with completely decentralized self-organization and resource usage.
Due to principles design, completely decentralized and self-organizing - opposed to client-server paradigm - the peer-to-peer concept emerges as the design of the future. From the point of view of the peer-to-peer concepts there are different challenges, e.g. resilient and scalable distributed systems and new services. Statistics establish that 50 per cent of Internet traffic obeys to peer-to-peer applications, in some cases up to 75 per cent. The growing of Internet, users and bandwidth, is requiring an increase of a diverse wealth of applications. The client-server paradigm requires a great effort and resources to meet these challenges. Internet-based applications identify three main characteristics:

- Scalability.
- Security and reliability.
- Flexibility and quality of services.

It is difficult for client-server based applications to meet the evolution of Internet. The client-server centralized approach is one of the main constrains (resource bottleneck), it is easily attacked and difficult to modify due its placement within the network infrastructure.
All of above expressed is indicating that there is a bias of paradigm, from client-server schemes to peer-to-peer schemes.

2. UNSTRUCTURED PEER-TO PEER SYSTEMS

Was the first generation of peer-to-peer based file sharing, which used an unstructured approach. Napster [11] was one of them with a strategy based in a metaserver and servers for looking up the location of data items, after that the data was transferred directly between peers. Gnutella use a flooding technique, a query is sent to all the peers in the system until the required data of peer is found. Peer-to-peer networks do not rely on a specific infrastructure offering transport services. Based on TCP or HTTP connections, peer-to-peer system forms an *overlay structure* focusing on content allocation and distribution. In standard client-server systems content is stored and provided by a central server. Peer-to-peer are highly decentralized and locate a desired content at some peer and provide the corresponding IP address of that peer to the searching peer. The download of that content is initiated using a separate connection. In client-server system the server provides services or contents (webserver, time server), clients only request content or service from the server. In peer-to-peer systems all resource are provided by peers, playing role of clients or/and servers, this is expressed by the term **servent** (first syllable of the term server and the second of the term client). There was in the first generation of peer-to-peer systems some ones that used a centralized approach. The server is still available, however contrary to the client-server approach this server only stores IP address of peers where some content is available, reducing the load of the server (Napster [11] is an example).

Gnutella 0.4 and Freenet were decentralized approach in replacement of the centralized scheme above presented. These schemes rely on flooding the desired content identifier over the network, reaching a large number of peers. Peers which share content will respond to the requesting peer. An important drawback is the large generation of traffic by flooding the request. To avoid this situation,Gnutella 0.6 introduces a hierarchy of nodes called *superpeers*, which store the content available at the connected peers together with their IP addresses. The main mission of these superpeers is reduce hops in the process of searching, reducing the traffic in the network.

The above schemes are *unstructured peer-to-peer* because the content stored on a given node and its IP address are unrelated and do not follow any structure. Examples of unstructured peer-to-peer systems are Napster, Gnutella [11, **?**], FastTrack, eDonkey, Freenet.

3. STRUCTURED PEER-TO PEER SYSTEMS

The challenge of develop scalable unstructured Peer-to -Peer applications put in attention the research community.

Due the advantages and possibilities of decentralized self-organizing systems, researchers focused on approaches for distributed, content-addressable data storage so called *Distributed Hash Tables* (DHT). These were developed to provide distributed indexing, scalability, reliability and fault tolerance.

Using DHT a data item can be retrieved from the network in a complexity of $O(logN)$. The underlying network and the number of peers in a structure approach can grow without impact on the efficiency of the distributed application; there is a contrast to the previously describes unstructured peer-to-peer applications which usually exhibit, at best, linear search complexity.

Four of the most interesting and representative mechanisms for routing messages and locating data for structured content distribution systems are:

- Freenet [6, 7] is a loosely structured system that uses file and node identifier to produce an estimate of where a file may be located, and a chain mode propagation approach to forward queries from node to node.
- Chord is a system whose nodes maintain a distributed routing table in the form of an identifier circle on which all nodes are mapped and an associated finger table is built.
- CAN is a system using n-dimensional cartesian coordinate space to implement the distributed location and routing table, each node is responsible for a zone in the coordinate space.
- Tapestry (and Pastry and Kadmelia [13]) are based on plaxton mesh data structure, which maintains pointers to nodes in the network whose IDs match the elements of a tree-like structure or ID prefixes up to a digit position.

4. SELF ORGANIZATION

Under the term self-organization it is possible consider autonomy, self-maintenance, optimization, adaptability, rearrangement, reproduction or emergence.

4.1. Definitions

System: A system is a set components that have relations between each other and form a unified whole. A system distinguishes itself from its environment.

Complexity: This term is used to denote the existence of system properties that make it difficult to describe the semantics of a systems overall behavior in an arbitrary language, even if complete information about its components and interaction is known .

Feedback: The return to the input of a part of the output of a machine, system or process (as for producing changes in an electronic circuit that improve performance or in an automatic control device that provide self-corrective action).

Emergence: Refers to unexpected global system properties, not present in any of the individual subsystems, that emerge from component interactions [5].

Complex Systems: Complex systems are systems with multiple interacting components whose behavior cannot simply inferred from the behavior of the components [20].

Criticality: An assembly in which a chain reaction is possible is called critical, and is said to have obtained criticality.

Hierarchy: In this context hierarchy is defined as a rooted tree.

Heterarchy: A heterarchy is a type of network structure that allows a high degree of connectivity. by contrast, in a hierarchy every node is connected to at most one parent node and zero or more childs nodes. In heterarchy, however a node can be connected to any of the surrounding nodes.

Stigmergy: Stigmergy defines a paradigm of indirect and asynchronous communication mediated by an environment.

Perturbation: A perturbation is a disturbance which causes an act of compensation, whereby the disturbance may be experienced in a positive or negative way.

4.2. Characteristics of self-organization

Based on above definitions, self-organization of systems could be characterized as follow:

Self-determined Boundaries: The border between system and environment is defined *by the system* itself.

Independence of identity and structure: The distinction between identity and structure allows to explain flexibility and adaptability.

Maintenance: A self-organizing system must try to maintain itself.

Feedback and heterarchy: If a system is perturbed, it try to restructure to maintain itself, so it need cross-linked relations with its neighborhood.

Self-determined reaction to perturbation: A self-organizing system reacts when a perturbation occurs, but it needs metrics for detecting and evaluating the perturbation.

These characteristics of self-organizing systems can be extended to P2P systems establishing several basic criteria such as boundaries, reproduction, mutability, organization, metrics and adaptivity; and criteria for autonomy as feedback, reduction o complexity, randomness, self-organized criticality and emergence. Besides the degree of conformance to these criteria, every system has an **identity** or a main purpose that is essential characteristic of the system. The identity of a P2P system is imposed from outside (the developers) and it is not self-determined.

5. APPLICATION AREAS

Peer-to-peer is an alternative for managing different types of resources as information, files bandwidth, storage and processor cycles.

5.1. Information

In this section is explained how P2P networks is deployed in areas of information.
Presence Information: Presence information is very important in P2P applications. It provides information about which peers and resources are available. This is relevant for the self-organization of the system. The use of information is also important to share processor cycles because the system knows which processor is overload and which one not. The peers are agents of information for the others peers.
Document management: usually documents systems are centrally organized, this allows share storage, management and use of data. A great effort is necessary to create a centralized index of relevant documents. The experience shows that documents created in a company are distributed among the desktop PCs without a central repository having any knowledge of their existence. In this case, the P2P networks are very useful.
Collaboration: P2P permits management of documents at level of closed working groups.

5.2. Files

A characteristic of file-sharing is that sometimes peers ares client when they download files and sometimes servers when they upload files (*sevents*). A central problem in P2P systems is the searching of the contents or files required (lookup problem)[4]. In the context of file-sharing, three different models have been developed: the flooding request model (Gnutella) [16, 17], the centralized directory model (Napster) and document routing model (Freenet) [6, 7, 14] .

5.3. Bandwidth

The traffic on networks is constantly rising, mainly in large volume of multimedia data, file-sharing, so the effective use of the bandwidth has suffer an important increment. When data are centralized and a spontaneous increment of demand arises, the bandwidth becomes a bottleneck. P2P approach increases load balancing without any kind of additional administration, by taking advantage of transmission routes which are not fully exploited. This concept is applied in the areas of streaming. A shared use of the bandwidth is also very well exploited splitting big files in smaller blocks which are downloaded by the requesting peers, BitTorrent [8] is an implementation using this principle.

5.4. Storage Space

With P2P storage networks, only a portion of the disk space available on desktop PC will be used. A P2P storage network is a cluster of computers, based on existing networks, which share all the storage available in the network. Examples are PAST [18], Pasta [15], CFS [9], Oceanstore [12], Farsite [1], and Intermemory [10].

5.5. Processor Cycles

There are requirements for high performance computing, at the same time there is computing power unused, this an incentive for using P2P applications to bundle that computer power. In this way it is possible to achieve computing power more cheap than a supercomputer can provide. This is effected by forming a cluster of independent, networked computers, in which a single computer is transparent and all the networked nodes are merge into a single logical computer. An example is SETI@home [2].

6. APPLICATIONS BASED ON PEER-TO-PEER

Some applications based on P2P follows:

6.1. Application-Layer Multicast

In the early days the size of Internet , certainly limited, permitted broadcasting a single packet to every possible node. In the present Internet, this technique of broadcasting is very expensive. Now is necessary a selective broadcast, such multicast. In this field P2P technology has helped, in his unstructured networks, to reach *unlimited* scalability.

6.2. GRID Computing

The basic objective of GRID computing is to support resource sharing among individuals and institutions (organizational units), or resource entities within a networked infrastructure. Grids are structured and has standards, but not capacity of self-organizing, fault tolerance and scalability. On the other hand P2P systems are self-organizing, fault tolerance, react very well a transient populations of peers but is lack of standards. All the efforts of researching in these fields is in merge the best of the two worlds. Indeed the question of how the two concepts converge is still open [3].

7. SUMMARY: THE PRESENT AND THE FUTURE

There was a lot of work did and there is a lot of work to do in the field. It is possible to classify and summarize all the activities in applications and research, present and future.

7.1. Applications

7.1.1. The Present

From 2004 up today

- Support for different communications forms
- Telephony.
- Streaming
- Scalable and flexible naming systems.
- Personal communications (e.g.e-mail).
- Interorganization resource sharing.
- Context/content aware routing.

7.1.2. The Future

Challenges in the future of applications

- Video conference.
- Distribution of learning material.
- Location-based services in Mobile Ad Hoc Networks (MANET), distributed and centralized.
- Context aware service.
- Trustworthy computing.

7.2. Drawbacks

Reasons against peer-to-peer.

7.2.1. The Present

Up today.

- Law suits against users.
- Software patents.
- Intellectual properties.
- P2P requires *flat rates* access.
- Still low bandwidth end nodes.
- Digital right management.
- Best effort service insufficient for most applications.

7.2.2. The Future

- Lack of trust.
- Commercialization as the end of P2P.
- P2P integrated into other topics.

7.3. Research Focus

What are the present research efforts and what the researching work to do.

7.3.1. Nowadays

Actually points of researching.

- Semantics integration of different information types in the specific peer-database.
- Quality of services criteria (consistency, availability, security, reliability).
- Legacy support in overlays.
- P2P and non-request reply interactions.
- highly adaptive DHTs.
- Overlay optimization.
- P2P signaling efficiency.
- Data dissemination.
- Resource allocation (mechanism and protocols)and guaranteeing quality of services in
- P2P systems.
- Self determination of information source.
- Accounting incentive.
- Realistic P2P simulator.
- Decentralize reputation mechanism.
- Semantics queries.
- Efficient P2P content distribution.
- Content-based search queries, metadata.
- Reduction of signaling traffic.
- Data-centric P2P algorithm.
- Content management.
- Application/data integration.
- Security trust, authentication transmission.
- Incentive market mechanism.
- Reliable messaging.
- P2P in mobile cellular/ad-hoc.

7.3.2. Future Challenges

- Anonymous but still secure e-commerce.
- Interoperability and/vs standards.
- ReaP2P for bussiness information systems.
- Real time P2P data dissemination.
- P2P file systems.
- Concept of trust and dynamic security.
- Dynamic content update.
- Distributed search mechanism.
- P2P technologies in MANET.
- Mobile P2P.
- Intelligent search.
- Service differentiation.
- P2P-GRID integration.

Certainly there is a lot of work to do, this paper has not conclusions (nothing is over) because all is just beginning. The fields of applications is huge. There are excellent readings [23, 3, 22] that should be used for researching and teaching.

8. REFERENCES

[1] A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, and J. R. Douceur. *FARSITE: Federated, Available and Reliable Storage for an Incompletely Trusted Environment*, 2002. http://Research.microsoft.com/sn/Farsite/OSDI20002.pdf

[2] D. Anderson. *SETI@home*. chapter 5, pp 67-76. OReally,2001.

[3] S. AndroutsellisTheotokis and D. Spinellis. *A Survey of Peer-to-Peer Content Distribution Technologies*. ACM Computing Surveys, Vol. 36(4), 2004.

[4] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris and I. Stoica. *Looking up Data in P2P Systems*. Communications of the ACM, 46(2), 2003.

[5] J. L. Casti. *Complexity*. Enciclopaedia Britannica. 2005

[6] I. Clarke. *Freenets Next Generation Routing Protocol*. 2003. http://freenet.sourceforge.net/ index.php?page=ngrouting.

[7] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley. *Protecting Free Expression Online with Freenet*. IEEE Internet Computing, 6(1), pp 40-49, 2002.

[8] B. Cohen. ,Incentive to Build Robustness in BitTorrent. Workshop on Economics of Peer-to-Peer Systems, 2003.

[9] F. Dabek, M.F. Kasshoek, D. Karger, R. Morris, and I. Stoica. *Wide-area Cooperative Storage with CFS*. Proceedings of the 18th ACM Symposium on Operating Systems Principles. pp 202-215, 2001.

[10] A. Goldberg and P. Yianilos. *Forwards an Archival Intermemory*. Proceedings of the IEEE International Forum on Research and Technology Advances in Digital Libraries. pp 147-156, 1998.

[11] A. Kim and L. Hoffman. *Napster and other Internet peer-to-peer applications*. George Washington University, 2002,citeseer.ist.psu.edu/kim01pricing.html.

[12] J. Krubiatowicz, D. Bindel, Y. Chen et al. *OceanStore: An Architecture for Global Scale Persistent Storage*. Proceedings of the 9th International Conference on Architecture Support for Programming Languages and Operating Systems. 2000.

[13] P. Maymounkov and D. Mazieres. *Kademlia: A peer-to-Peer Information System Based on the XOR Metric*. International Workshop on Peer-to-Peer Systems. (IPTPS02), 2002.

[14] D. S. Milojicic, V. Kalogeraki,, R. Lukose, K. Nagaraja and J. Pruyne. *Peer-to-Peer Computing*.HP, Technical Report, HPL-2002-57, 2002. http://www.hpl.hp.com/techreports/2002/HPL-2002-57.pdf.

[15] T. Moreton, I. Pratt, and T. Harris. *Storage, Mutability and Naming in Pasta*. 2002. http://www.cl.cam.ac.uk/users/tlh20/papers/mph-pasta.pdf.

[16] M. Ripeanu. *Peer-to-Peer Architecture Case Study: Gnutella Network*. Proceedings of the IEEE 1st International Conference on Peer-to-Peer Computing, 2001.

[17] M. Ripeanu and I. Foster. *Mapping the Gnutella Network: properties of Large-scale Peer-to-Peer Systems and Implications for System Design*. IEEE Internet Computing, 6(1), 2002.

[18] A. Rowstron and P. Druschel. *Storage Management and Caching in PAST, a Large-scale, Persistent Peer-to-Peer Storage Utility*. 18th ACM SOSP01. 2001.

[19] S. Saroiu, K. P. Gummadi and S.D. Gribble. *Measuring and analyzing the characteristics of Napster and Gnutella hosts*. Multimedia Systems, 9(2), 2003. pp 170-184, Springer-Verlag.

[20] F. Schweitzer. *Coordination of Decisions in Spatial Multi-Agents Systems*. International Workshop on Socio- and Econo-Physics. 2003.

[21] R. Steinmetz and K. Wehrle. *Peer-to-Peer-Networking and -Computing*. Informatik-Spectrum, 27(1). Springer. 2004.

[22] R. Steinmetz and K. Wehrle (Eds). *Peer-to-Peer Systems and Applications*. Lecture Notes in Computer Science, LNCS 3485, Springer. 2005.

[23] J. Van Der Merwe, D. Dawound, S. Mc Donald. *A Survey on Peer-to-Peer Key Management for Mobile ad hoc Networks*, ACM Computing Survey, 39(1), 2007