Framework for Adaptivity in Peer-to-Peer Systems

Valentin MESAROS

Department of Computer Science
Université catholique de Louvain
1348 Louvain-la-Neuve, Belgium

email: `valentin@info.ucl.ac.be`

May 29, 2001

1 Introduction

The present proposal represents the author's admission submission; i.e., the second step of the Ph.D. program at the *Faculté de Science Appliquées*, *Université catholique de Louvain*, Louvain-la-Neuve, Belgium. It contains five sections, as follows: We begin by describing the research context; i.e., peer-to-peer systems, and highlight the need for adaptivity in distributed computing. We continue with a succinct overview of the state of the art in adaptive systems. Section 3 describes the objectives and motivations of the project, and its novelty with respect to the state of the art. In Section 4 we present the *society* model, representing, in broad strokes, our architectural vision of a peer-to-peer system. We conclude by presenting our ongoing research and studies.

1.1 Peer-to-peer systems

Recently, researchers from different fields (e.g., distributed computing, networking) have been giving a paramount attention to peer-to-peer systems[1], considered as possible solution treasure to problems like scalability, fault tolerance, availability, and load balancing.

A peer-to-peer (P2P) system forms a logical network at the application level. A such network has the involved applications acting as peers (i.e., logical nodes), and the connections between them considered as links. The underlying physical network[2] is used for interchanging messages between the peers.

Within a P2P system, the peers are said to be equal; e.g., they can play any of the following roles: client, server, and router. Thus, a peer $\mathcal{A}$ can be consumer with respect to a peer $\mathcal{B}$, and producer with respect to an other one $\mathcal{C}$, at the same time, thus naturally providing the primitives for information deployment in the system. By allowing a peer to play any of these roles, a new type of network is formed, necessitating a new computing paradigm (i.e., the P2P computing paradigm). This paradigm deals with algorithms for information and peers management providing solutions to the mentioned problems.

[1] However, the idea of peer-to-peer systems is not that new. The very first networks were built with this concept in mind; an example application is DNS [?].

[2] By physical network we understand the routing at the network level of the OSI seven-level model [?].

The current P2P systems are quite different in functionality and purpose, but they all adhere to the same main concepts; i.e., make the system more robust by better employing the edges of the network and propagating the information gradually, rather than connecting directly to a single failure point, thus increasing accessibility, scalability, and fault tolerance. Here are some of the current systems: Gnutella [?] – file sharing system, Freenet [?] – distributed storage and retrieval system addressing concerns of privacy and availability, Chord [?] – distributed lookup service, Tapestry [?] – an infrastructure for fault tolerant wide-area location and routing, Gridella [?] – system enabling distributed search and replication, Jxta [?] – common platform for building a wide range of distributed services and applications.

1.2 The need for adaptivity

We are long past the period when the machines, had mostly the same computation capabilities, running the same transport protocol stack, and using the same kind of network. With the advent of the distributed computing, the researchers have begun to better notice the discrepancy between the machines and their systems. That is, the machines are no longer thought of as isolated, but they are designed to inter-operate.

The recent advances in electronics and telecommunications have, somehow, overtaken the distributed computing researchers. The variety of machines (e.g., mainframes, personal digital assistants, smart-phones), the transport environments diversity (e.g., WAN, WLAN, GPRS, bluetooth), and the increase of the application demands (e.g., multimedia content, take part in vast application level networks), have required that the set of machines no longer run[3] the same operating systems, use the same transport protocols, and run the same applications. Moreover, as interconnected computing becomes more accessible, researchers must reconsider problems (which had been, more or less, ignored) like security, limited resources, and intermittent connection.

Hence, the common conclusion was that there was a need for new computing paradigms (e.g., ubiquitous/pervasive computing [?, ?], P2P computing [?]), and adaptability at different levels of the computing systems [?, ?, ?].

2 State of the art in adaptive systems

There has been done a lot of work in the field of adaptability in centralized and distributed systems. As described previously, adaptability was required as part of the solution to the problem of disparity between different components of a computing system. The adaptability is considered to be applied at different levels of a computing system. We classify it in four categories regarding with the entities that have been studied for adaptation; i.e. application, protocols, operating system, and network.

Adapt the application The application can adapt to the underlying network fluctuations and to the running environment (e.g., processing resources). If the adaptation is entirely the responsibility of individual applications, it is called *laissez-faire*. Instead, if the application can instrument the underlying subsystem and intervenes only when required, the adaptation is said to be *application-aware*; an example is the Odyssey system [?].

Adapt the protocols The transport and distributed protocols of a computing system can adapt both, to the network and the running environment, and to the application employing them.

[3] At least in their static form.

In the first case, the protocols can adapt to the network characteristics such as bandwidth, jitter, and bit error rate. In order to implement such an adaptive system, different configurable systems have been proposed such as x-kernel [?], Horus [?], and Coyote [?]. They all address mainly the same problems, but differ in their flexibility to configure.

In the second case, the application provides the underlying protocols with information about its behavior [?], allowing them to adapt adequately.

If the adaptation is solely the responsibility of the underlying subsystem, it is said to be *application-transparent.*

Adapt the OS The assumption of current operating systems is that a computer's hardware and software configuration will not change substantially while it is running. That is not reasonable for the variety of computing gadgets now being used. Researches on "micro-kernel" operating systems, like Amoeba [?], can be part of the solution.

Adapt the network A recent study [?] defines the real goal of the current adaptive networking as providing good end-to-end service, where the end points are located in applications. Ongoing research in active networks [?] are promising, providing the possibility to program the underlying physical networks in order to satisfy communication requirements of highly distributed systems. Since this technology is not yet ready for large scale, researchers have thought of partitioning the application functionality between the client, server, and intermediate deployed active proxies (i.e., adaptor modules) in order to alter the form of data (e.g., data distillation, compression, encryption). There are two proxy-based models: the *last-mile* approach [?], where there is only one proxy located close to the host to be adapted to, and the *distributed* approach [?], where the adaptors are deployed in the network by a planning algorithm.

To our knowledge, in the field of P2P computing, there are only two very recent ongoing research projects that approach our beliefs for P2P systems; namely, MAPS [?] and brocade [?]. MAPS is an architecture which extends the existing P2P infrastructure in order to improve media service over heterogeneous peer networks. Precisely, by being aware of the overlay infrastructure[4], and peer's resources, it accommodates and accelerates multimedia peer applications. Brocade is an overlay to be layered on top of the existing P2P systems, that exploits knowledge of underlying network characteristics. Precisely, it uses a hierarchical schema for routing messages in the system. That is, the messages are tunneled to, and exchanged between landmark nodes (supernodes), thus reducing the network latency.

3 Objectives and motivations

Our main objective is to study the adaptability issues in the P2P systems and to develop a framework where the tasks of a P2P system can be defined, system-wide deployed, and well mapped to the peers' *profiles*[5]. Additional objectives include the development of algorithms for task deployment throughout the system and adaptive algorithms acting at different levels of the system.

Although a lot of work has been done in the field of adaptability for distributed systems, none claims to solve the complete problem. Moreover, they are mainly concentrated on point to point services, and did not consider the distributed system as a whole; still, their results are valuable. We intend to study the adaptability in P2P computing at different levels of the system.

[4]That is, the logical network formed by the peers at the application layer.

[5]See Section 4 for the definition of the peer's *profile* which is part of our vision of a P2P system.

The P2P systems are composed of applications running on different machines and gadgets, static or mobile, having different capabilities and resources, and employing a large variety of data transport environments [?, ?]. Under such conditions, where heterogeneity is the rule, not the exception, we believe that the adaptability will play a key role.

Distributed computing and P2P computing are quite similar. Nevertheless, one of the most dramatical differences between them is the fact that, in P2P computing, global invariants have to be maintained by only having access to local neighborhood information, and acting locally. We will focus our research on the adaptability of the computing system as a whole. Roughly, we can define our research as finding answers to the following two questions:

- *When should a P2P system react?*
- *How should a P2P system adapt?*

However, before answering these questions, an indirect one, related to the system representation, must be addressed. Since we consider P2P computing, we can not design a central approach. At the other extreme, we can have each peer acting individually, relying on information measured to, or provided by its neighbors. A third approach would be a hybrid between the central and the distributed ones. That is; forming groups of peers and having groups acting individually, relying on information provided by other groups.

Coming back to our research questions, the first one is about the possible situations when a P2P system would react, as a first step of the adaptive process. A P2P system could react when different conditions are fulfilled such as: peer (dis)connection, and change in peers' profile. However, a system that is highly agile may suffer from instability. Thus, the reacting conditions must be well defined. Furthermore, the metrics leading to condition triggering must be reliable.

The second question is about the ways a system should change to different reacting conditions. A system can change its neighborhood, for instance, when it discovers a closer peer (in number of network hops), or it can change the deployment of some of its tasks to peers with currently available resources. Moreover, a peer host can make use of most of the adaptive techniques described in Section 2 in order to adapt its underlying protocols. A resulting challenge is to orchestrate and to compose the adaptations at different levels of the system so that they produce the desired effect.

4 The *society* peer-to-peer model

The current P2P systems are premised on the fact that (a) all peers are assumed to *fully* participate, and that (b) all peers *are* equal (i.e., they have the same capabilities and abilities). The first premise specifies that each peer in the system must participate and serve the system objectives, for the well-functioning of the system[6]. The second premise, specifies that all peers can, and should be ready to play any role the system would request them to play; e.g., client, server, or router.

We believe that these premises are too rigid to be applied to the variety of computing machines and running environments. Furthermore, we argue that whereas all peers *may* be equal, not all *can* be, and that the choice of whether, or not, to participate should be left at the peers' discretion. Indeed, there may be situations where some peers are not equal with the others, for instance, because they can not (e.g., due to their resource constraints), or because they don't want (e.g., due to "personal reasons", like trust) to participate as servants of the system. Nevertheless, they may still be interested in the systems' services.

[6]For instance, in Gnutella the peers are imposed to play actively, by publishing what they have downloaded, mainly, in order to hide bad system designs (i.e., the Gnutella's policy against "free-riders").

Our vision of a P2P system is an organized society where every individual – alias peer – can have its contribution and responsibility to the well-functioning of the system as a whole. As within a society, peers, or groups of peers, are assigned different jobs (i.e., tasks). We simply characterize peers by their *profile*: *potential*, *qualification*, and *interest*.

potential The potential of a peer represents the resources (e.g., bandwidth, CPU, storage capacity) that a peer contains and can provide for achieving the system's objectives. The connectivity and data transport capabilities may also characterize the peer's potential.

qualification The architectural computing model of a peer (e.g., the operating system, protocol stack, safety schema) represents its qualification degree.

interest The interest of a peer represents its willingness to participate or collaborate with the system.

Depending on its *qualities*, and in collaboration with the system, a peer negotiates its place in the society; i.e., finding the right job for the right *man*. As in a real society, in a P2P system, the peers differ by their qualities, and, with respect to certain tasks, certain peers could be more suitable than others. Related with this, a new challenge arises: how should we deal with partial-reliable information; i.e., the information that peers advertise may not reflect their actual profile.

Another issue about our P2P system vision is related to peer grouping. There are three reasons that make us believe that dynamically organized hierarchical architectures for P2P systems will be necessary. (a) The system demands on a participating peer varies. (b) The profiles of collaborative peers are very diverse. This will lead to groups of peers specialized for different services[7]. (c) The number of peers in P2P systems will grow, leading to increasing latency in the case of gradually information propagation (i.e.; before reaching its destination, the information has to pass through too many peers).

We believe that the *society* model can lead to novel functionality and interesting behaviors for P2P. For instance, the potential quality of a peer may affect the well-functioning of the system, leading to an adaptation. That is, the fact that a peer does not have enough resources (*thin* peer), can make the system take action adequately: (a) connect to a more powerful node, and (b) still use the thin peer for less powerful computations and services. Furthermore, this model will remove the responsibility from the applications, facilitating their programming.

Ex 1 Freenet [**?**] is a distributed storage and retrieval system. One of its advantages is related to privacy in the system; i.e., one does not know who the requester and the provider of an exchanged file are. The privacy is attained, mainly, because the file is not directly downloaded from the source, but it is passed through the P2P network, thus not distinguishing between peers with different potentials.
A more efficient model would adapt to the peers' profile, and, thus, use all peers (i.e., even the thin ones) for file location, and, as much as possible, use only the suitable peers for file downloading.

Ex 2 Gridella [**?**] is a system enabling distributed search and replication. Each peer in the system is responsible for part of the overall tree, and the peer workload differs from one peer to another. An adaptive system would deploy the workload with respect to peers' profile.

[7]Jxta [**?**], also, has a similar concept.

5 Ongoing research

During the first doctorate year we have been working within the PIRATES[8] project. Our main research was based on the Mozart [?] system communication layer; precisely, on the development, integration, and implementation of a new transport protocol (i.e., Multichannel Data Protocol – `MDP`) to the Mozart system.

Beginning this year, we have begun working within the PEPITO[9] international project. Our research within this project is based on the distributed subsystem of P2P systems. Currently, we are working on the design and development of a decentralized algorithm [?] for mapping the logical network routing of P2P systems to the physical network routing.

The choice of the present thesis subject emerged from the combination of the experience we had with the PIRATES project and the questions the PEPITO project is challenged to answer.

During the period June 2001 – May 2002 the author attended the 13th International Lipari School[10] for Computer Science, and two courses at INGI/INFO/UCL; namely, "Structures des langages algorithmiques" in the first semester, and "Théorie des langages: syntaxe et sémantique" in the second semester.

References

[1] The Gnutella Protocol Specification v0.4. `http://www.clip2.com/GnutellaProtocol04.pdf`.

[2] Jxta v1.0 protocols specification. `http://www.jxta.org`, June 2001.

[3] Mozart system. `http://www.mozart-oz.org`, May 2002.

[4] K. Aberer, M. Punceva, M. Hauswirth, and R. Schmidt. Improving data access in p2p systems. *IEEE Internet Computing Journal*, 6(1), January/February 2002.

[5] B. Badrinath, A. Fox, L. Kleinrock, G. Popek, P. Reiher, and M. Satyanarayanan. A conceptual framework for network and client adaptation. *IEEE Mobile Networks and Applications*, 5(4), December 2000.

[6] N. Bhatti, M. Hiltunen, R. Schlichting, and W. Chiu. Coyote: A system for constructing fine-grain configurable communication services. *ACM Transactions on Computer Systems*, 16(4), November 1998.

[7] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proc. of ICSI Workshop on Design Issues in Anonymity and Unobservability*, July 2000.

[8] E. Oram et al. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies.* O'Reilly & Associates, 2001.

[9] A. Fox, S. Gribble, Y. Chawathe, and E. Brewer. Adapting to network and client variation using active proxies: Lessons and perspectives. *A special issue of IEEE Personal Communications on Adaption*, 1998.

[10] N. Hutchinson and L. Peterson. The x-kernel: An architecture for implementing network protocols. *IEEE Transactions on Software Engineering*, 17(1), January 1991.

[11] R. Lienhart, M. Holliman, Y. Chen, I Kozintsev, and M. Yeung. Improving media services on p2p networks. *IEEE Internet Computing Journal*, 6(1), January/February 2002.

[12] V. Mesaros and L. Onana. Mapping the logical network routing to the physical network routing. To be presented at the 2nd PEPITO Workshop, July 2002.

[8] PIRATES' objectives are about developing methods and tools for network-transparent dependable distributed programming, see `http://www.info.ucl.ac.be/people/PVR/distribution.html` .

[9] PEPITO is a European research project focused on P2P applications, platforms, languages, and theory, see `http://www.sics.se/pepito` .

[10] Lipari international sumer school `http://lipari.cs.unict.it` .

[13] P. Mockapetris and K. Dunlap. Development of the domain name system. In *Proc. of ACM SIGCOMM*, April 1988.

[14] S. Mullender, G. van Rossum, A. Tanenbaum, R. van Renesse, and H. van Staveren. Amoeba: A distributed operating system for the 1990s. *IEEE Computer*, 23(5), May 1990.

[15] B. Noble, M. Satyanarayanan, D. Narayanan, J. Tilton, J. Flinn, and K. Walker. Agile application-aware adaptation for mobility. In *Proc. of the 16th ACM Symposium on Operating Systems Principles*, October 1997.

[16] 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*, 6(1), January/February 2002.

[17] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. Technical Report UW-CSE-01-06-02, University of Washington, July 2001.

[18] M. Satyanarayanan. Fundamental challenges in mobile computing. May 1996.

[19] M. Satyanarayanan. A catalyst for mobile and ubiquitous computing. *IEEE Pervasive Computing*, 1(1), January/March 2002.

[20] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. Technical Report TR-819, MIT, March 2001.

[21] A. Tanenbaum. *Computer Networks*. Prentice Hall, 1996.

[22] D. Tennenhouse, J. Smith, W. Sincoskie, D. Wetherall, and G. Minden. A survey of active network research. *IEEE Communications Magazine*, 35(1), January 1997.

[23] R. van Renesse, K. Birman, R. Friedman, M. Hayden, and D. Karr. A framework for protocol composition in Horus. In *Proc. of the 14th ACM Symposium on Principles of Distributed Computing*, August 1995.

[24] T. Watson. Application design for wireless computing. In *Proc. of Workshop on Mobile Computing Systems and Applications*, December 1994.

[25] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3), September 1991.

[26] M. Yarvis, P. Reiher, and G. Popek. Conductor: A framework for distributed adaptation. In *Proc. of the 7th Workshop on Hot Topics in Operating Systems*, March 1999.

[27] B. Zhao, Y. Duan, L. Huang, A. Joseph, and J. Kubiatowicz. Brocade: Landmark routing on overlay networks. In *Proc. of the 1st Workshop on Peer-to-Peer Systems*, March 2002.

[28] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, U. C. Berkeley Technical Report, April 2000.