On P2P Collaboration Infrastructures

Manfred Hauswirth, Ivana Podnar
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

Stefan Decker
Digital Enterprise Research Institute
National University of Ireland
University Road, Galway, Ireland

Reporter: YuGuangCan
03/20/2006
1. Introduction

In this paper we argue that collaboration needs to get more flexible, make use of existing resources provided by the participants, should be easier to set up, and support virtual organizations and teams collaborating for limited time spans.
1. Introduction

As this implies that teams can rely on pre-existing infrastructure only to a limited extent, we believe that P2P systems have a lot to offer in this context as they specifically target the use of resources at the edge of the network.
1. Introduction

On the other hand, they may introduce new problems that do not exist in the server-based solutions since they are not as mature and a number of problems in P2P still awaits being solved.
2. Scenarios and Requirements

- Collaboration scenarios have recently been enriched by the proliferation of mobile devices with high-bandwidth network connectivity.

- Let us consider three different collaboration scenarios covering different time spans: ad-hoc, short-term, and long-term collaborations.
2. Scenarios and Requirements

- Ad-hoc collaborations offer temporary support possibly in a limited geographical area, and enable group members to flexibly interact and communicate with limited collaboration functionalities.

- Typical examples are meetings, conferences, or conventions where people meet, exchange contacts, ideas, and knowledge, but do not work together on a common product or to satisfy a mutual goal.
2. Scenarios and Requirements

- *Short-term collaborations* cover limited time spans, and enable *virtual organizations* and teams to collaborate in order to satisfy a *mutual objective*, e.g., create a project proposal, jointly work on a paper, or integrate software components.

- Such collaborations require knowledge and data exchange, and are based on a *trust relationship* between team members.
2. Scenarios and Requirements

- Long-term collaborations are set up by collaborating organizations and teams for longer time periods.
- As this type involves a longer time period and the involved people and utilized resources fluctuate less, it is necessary to offer flexible and versatile interaction and cooperation capabilities and services. Usually a centralized infrastructure is used for this type.
2. Scenarios and Requirements

- A single person may be involved in a set of different collaboration scenarios at the same time.

- Keeping in mind different collaboration scenarios, we can examine different technological strategies for providing support for collaboration: a standard centralized solution, a pure P2P approach, and a mixed model comprising peers that can rely on some pre-existing infrastructure.
2. Scenarios and Requirements

- The *centralized solution* is expensive as it requires continuous maintenance and relies on centralized collaboration servers. Groove.

- A *pure P2P approach* uses a P2P overlay network without need for extra infrastructure or setup since team members provide the computers and software, be suitable for ad-hoc and short-term collaborations.
2. Scenarios and Requirements

- The mixed model seems a feasible intermediate solution between the centralized and pure P2P approach. It uses a reliable super-peer network for storage while guaranteeing data availability.
3. Architecture

- We assume that a system uses a P2P overlay network, either a pure P2P solution, or a super-peer enhanced P2P overlay.

- Then a peer in the collaboration system would have the architecture shown in Figure 1.
3. Architecture

Figure 1. Architecture for P2P collaboration systems
3. Architecture

- Let us briefly discuss lower-level services and their relationship to P2P overlay since these represent the most challenging part of the architecture.

- The main issues to be solved by a distributed storage system are data availability, data consistency, data confidentiality and resilience against attacks.
3. Architecture

- Trust and reputation management are at the focus of current research in distributed systems.

- The **scale**, **distribution**, and characteristics of these systems dictate that also trust and reputation needs to be managed in a **distributed fashion** to avoid **single points of failures**.
3. Architecture

The publish/subscribe (P/S) service enables asynchronous and flexible group communication. It allows the system to inform interested and available users about changes in system, availability of new data, etc.
4. Distributed Search and Storage

- In this section we take a closer look at current, advanced P2P technologies to support distributed search and storage.
- Distributed search is a **prerequisite** for distributed storage as it offers the functionality which is usually provided by a **directory** in a standard file system.
- In the simplest case search is needed to discover a resource.
4. Distributed Search and Storage

- If a resource is large, then it may have been split into smaller segments which again would be stored on arbitrary peers.
- Search is needed to recover all parts to reconstruct the original resource.
- Also when replication comes into play, distributed search is relevant. It is necessary to discover replicas and provide support to keep them up-to-date.
4. Distributed Search and Storage

- On the other hand, the search mechanism is influenced by the replication mechanism as the consistency guarantees provided by the replication mechanism, determine the freshness of information the search mechanism can return.
4. Distributed Search and Storage

■ Thus having a distributed indexing system at hand enables the discovery and access to any resource in a distributed system.

■ Additionally these systems typically offer replication and load-balancing for the index information.

■ the location of any resource will be found at any time, but it may not be able to access it if the corresponding peer is offline.
4. Distributed Search and Storage

To increase *availability* of the data, everyone who has downloaded a resource, of course could again enter it in the index. This would already provide a *very simple notion* of distributed storage, without much *consistency* guarantees though.
To clarify functionalities and terminology, it is also important to stop viewing current P2P systems just as distributed database systems.

Typical P2P systems, such as Napster, Kazaa, eDonkey, Gnutella, or distributed hash tables, such as P-Grid, Pastry, or Chord, are *location* systems.
4. Distributed Search and Storage

- They basically provide a distributed index that can be maintained in the presence of changing peer populations, node failures, and network separations.

- To ensure fault-tolerance they use replication which is applied to the index, not to the indexed data itself.
4. Distributed Search and Storage

- However, replication is required to ensure data availability.
- Simple replication strategies on a per-le basis scale only for small le sizes up to maybe 100kB.
- With larger sizes and just a bit of dynamicity in the system (joining/leaving peers, mobility, etc.) the system would breakdown immediately.
4. Distributed Search and Storage

- The distributed index can be used to track replicas, but the **actual data** transfer defeats efficiency.

- To **remedy** this situation, distributed archival storage projects such as **OceanStore** thus split resources into **digestible pieces** and distribute them among the peers.
4. Distributed Search and Storage

- Coding strategies are needed to detect changed pieces, and the content must be encrypted which comes at the cost that standard data manipulation operations become very expensive.

- In terms of data manipulation operations, at the moment P2P systems work very well for read-only or rather static data sets.
4. Distributed Search and Storage

When it comes to updates, P-Grid is one of the very few systems (at least the only distributed hash table) that supports this.

In principle, data updates would be possible based on this functionality by sending notifications to data replicas.

However, consistency of updates has not been extensively studied for these systems, so they are theoretically not modeled and understood.
4. Distributed Search and Storage

- The key question for the distributed storage in a P2P-based collaboration system thus is, which data should be replicated.

- Just replicating all data typically defeats any distributed scheme since it is of no use to replicate data that nobody will ever access.
4. Distributed Search and Storage

Additionally, it is the question whether complete consistency is required all the time or some relaxed model would be tolerable.

We claim that in many scenarios a relaxed consistency model would suffice. We also claim that many successful systems do on-demand replication/caching, tolerate lack of information, and work with not up-to-data data.
4. Distributed Search and Storage

For a fully decentralized, distributed system, a possible solution with relaxed consistency could look as follows:

- When a peer downloads data, it becomes a replica.

- In case of an update, all replicas are being informed. If a replica wants to update, it may look for the highest version via the index and download from any source. Peers which are not online will learn about the update when they get online again.
4. Distributed Search and Storage

- Queries always include the version of the found data. The user may decide what to use.

- Conflicting updates require manual resolution. In fact there are no good automatic schemes that always work as this essentially boils down to the distributed cache reconciliation problem.
4. Distributed Search and Storage

- If data is unavailable (all replicas are offline), the requester can subscribe to the system to be notified when the data becomes available again.
5. Publish/Subscribe and Presence

Publish/subscribe (P/S) systems offer support for collaboration scenarios that deal with m:n communication with information coming from multiple and heterogeneous sources and targeting numerous users.
5. Publish/Subscribe and Presence

- P/S provides means to define expressive subscriptions describing information and data properties the subscriber is interested in.

- Subscriptions are used as data filters, because they are matched to data coming from various sources (publishers) prior to delivery to interested destinations (subscribers).
5. Publish/Subscribe and Presence

- A P/S system provides an efficient service which pushes the data at the time of its publication to interested subscribers.
- Communicating parties, publishers and subscribers, interact asynchronously by generating and consuming short information items, i.e., notifications.
5. Publish/Subscribe and Presence

The major challenge for P/S in such environments is related to the design of efficient routing strategies that can deal with network changes (.churn.) while preserving high-expressiveness of subscriptions and low latency for delivered data.
5. Publish/Subscribe and Presence

- P2P systems offer interesting properties P/S systems could **exploit**. The goal is to provide **infrastructure-less** P/S systems on top of P2P systems.

- However, currently there are no P/S systems that utilize P2P characteristics, such as **efficient routing**, **redundancy**, and **load balancing**, to build flexible and efficient P/S solutions.
5. Publish/Subscribe and Presence

- Presence service maintains and offers information on users' presence and contact information.

- This includes the current user communication capabilities and preferences with respect to the applied terminal, application, and user state.
5. Publish/Subscribe and Presence

- Therefore, presence implementation can largely rely on an existing P/S implementation taking into account an extension related to user privacy.

- Users need to be able to grant access to their presence information. Furthermore, the update and retrieval of this information must be secured and authenticated.
Thanks!