Designing Peer-to-Peer Systems for Business-to-Business Environments

Davide Carboni, Massimo Deriu, Cristian Lai and Antonio Pintus
CRS4 (Center for Advanced Studies, Research and Development in Sardinia)
{dcarboni,mderiu,clai,pintux}@crs4.it

Abstract

This paper describes the design of a peer-to-peer system integrated in a larger framework for the automatic content production, formatting, distribution and delivery over multiple platforms called AXMEDIS (E.U. IST-2-511299).

One of the goals of the project is the reduction of costs and, among the others, the adoption of a collaborative environment based on a virtual database as an abstraction of a multitude of objects shared in a large network of content producers/distributors/aggregators. The peculiar properties of this system are the automation of P2P related operations, the professional query user interface based on Dublin Core and available rights of target objects, and the preemptive exclusion of uncertified participants.

1. Introduction

Peer-to-Peer technologies have drawn a lot of attention in the last years because of the large adoption by end users in file-sharing communities. The P2P can be considered an "old" novelty and not a revolutionary idea and often the distinction between P2P and traditional distributed systems is not so evident. Old UNIX tools like "*talk*" or the Web itself were conceived to make Personal Computers directly interconnected and able to exchange data. Although the P2P paradigm was potentially available from the beginning of the Internet era, only with Napster this technology gained the worldwide popularity. As everybody knows, file sharing is mainly used for illegal distribution of copyrighted contents, so the peer-to-peer technology is often referred as the technology of "pirates". The social and legal implications of the misuse of the technology are beyond the scope of this paper, but it is important to avoid that this misuse casts a bad shadow on a technology which has the potential to leverage the cooperation in community-based processes, to reduce costs in delivery of data and services by exploiting the resources at the edge of the Internet, and to make the net a place with more freedom for the users.

This work mainly covers the aspect of cost reduction in the production and delivery of digital contents in B2B environments. This reduction can be achieved by a more effective use of computing resources in an overlay network of content producers, distributors, and aggregators.

The resources that can be subject to "effective" use are the CPU, the disk space, and the bandwidth.

Most of the time, the CPU of our Personal Computers are "idle". This occurs when he CPU is not performing a computation in the background and in the foreground the user is involved in an interaction where the computer reaction is one or two order of magnitude faster than the human reaction. Seti@home[1] and distributed.net [2] are success stories of CPU sharing over a network of volunteers.

The disk space is another resource that can be easily shared because the cost-per-giga is dramatically dropped and even a low-end machine is equipped with tens or even hundreds of gigabytes. This fact allows overlay networks like Chord[3] and Freenet[4] to place replicas of files according to specific needs and strategies.

The bandwidth is another critical resource that is often a bottleneck in the traditional client-server distribution. P2P systems like Bittorrent[5] implement a set of algorithms that balance the load of distribution among a network of downloaders.

2. Requirements

Most of the popular P2P networks were born for illegal file-sharing purpose so among their features are user-identity hiding, resistance to the censorship, scalability toward huge amount of users, etc. Most of these features are not relevant for a B2B environment like AXMEDIS[6], where on the contrary the main requirements are the full automation of activities, the integration with a secure environment, the enforcement of digital rights and the availability of a professional query support.

Our P2P system is not a stand-alone application, but it is part of the larger AXMEDIS framework which provides a complete business process definition, defines a digital object schema, including support for DRM and extensible metadata.

The section §3 describes the automatic support for P2P related activities. Section §4 describes the design of the Query User Interface (QUI) and finally section §5 introduces some features related to the overlay network such as indexing/searching, discovery of peers, handshaking and object downloading.

3. Automating Publication and Loading of Objects

In the AXMEDIS framework the entire lifecycle of digital contents is governed by rules named "AXCP rules" (AXMEDIS Content Production rules). Rules are schedulable tasks associated to one or more queries in a way that at a given time, a given operation is performed on the objects belonging to the actualized queries.

The areas involving such rules are automatic content production, automatic content protection, automatic content adaptation, and automatic content publication in the P2P network. The last area is the topic of interest of this section and involves content sharing and distribution among content producers and distributors.

The P2P architecture of AXMEDIS contains a number of components which supports searching, indexing, loading, and updating objects. These components are:

- AXDB, which is the logical storage of objects either produced by the peer or for which he has the rights to include them in his production process.
- AXDBOUT, which is the logical storage of objects that a peer makes available to the P2P network. This component resolves remote queries and can be either local or remote depending on the nature of the peer (see §5 for details).
- AXDBIN, which is the logical storage of objects downloaded from remote peers but which are not yet loaded in the AXDB because the peer has not the rights to include these objects in his production process either because the peer must obtain a license or simply because the user discarded the objects after a preliminary preview.
- Publishing and monitoring module, which is mainly an event-based system built on top of the P2P network. Once a peer loads an object it is automatically subscribed as listener for "update" events. On the other hands, once a peer who has published an object releases a new version of the same object, this module broadcasts this "update" event to all the registered listeners.

Three different types of schedulable tasks are defined in the P2P architecture of AXMEDIS:

- Publication, to publish a selection of objects making them available in the P2P network.
- Loading, to load a selection of objects from AXDBIN to AXDB.
- Downloading, to download a selection of objects from the P2P network to the AXDBIN.

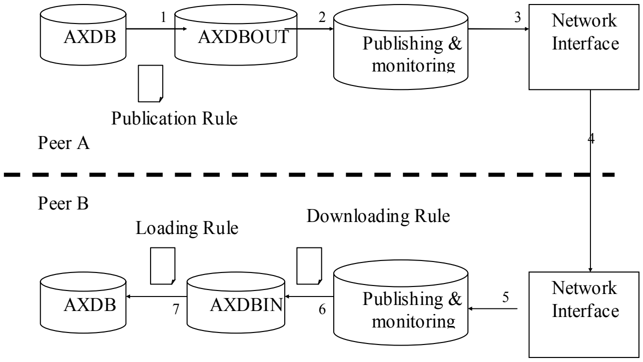

Figure 1. Steps in the publish-to-load process

Figure 1 shows the steps involved in the publish-to-load process that are described below:

1. when a publication rule is executed a selection of objects is published. Objects are indexed in the AXDBOUT
2. a publication event is produced
3. the publication event is sent to the P2P network
4. a node B interested in such event is notified
5. the event is enqueued in peer B
6. when a Downloading rule is executed, the P2P system checks if some objects are missing in AXDBIN and eventually starts a download

7. when a Loading rule is executed checks if in AXDBIN there are some objects that need to be reloaded in AXDB

4. The Advanced Query User Interface

The user interface for a P2P system in a B2B environment should permit to query both on document metadata as defined in the Dublin Core[8] specification and on available rights of target objects. Moreover, it should provide the functionality to save queries and to compose them using logical operators in order to build queries with arbitrary complexity. In our design we identified the Query User Interface (QUI) as a critical component in the whole AXMEDIS platform. The QUI is composed by two main components: the Flat QUI and the Tree QUI.

The Flat QUI presents a set of editable data pertaining to the Right Expression Language[9] and the Dublin Core Metadata. Each set is showed in a different form, generated on the fly according to a configuration file that can be edited by an expert operator in order to modify the number, the position, the editable data and, if necessary, to add new metadata. The Flat QUI is provided by default with a configuration file with most common metadata.

The Dublin Core form and the Available Rights form are presented into separated tabs of a tabbed panel. The elements in a tab are logically related by the AND operator. The two tabs are related each other by the AND operator. The logical structure of a flat query is:

```
DC_Query = (field1 OP1 value1) AND
... (fieldN OPN valueN)

AR_Query = (field1' OP1 value1') AND
... (fieldN' OPN valueN')

Flat_Query = AR_Query AND DC_Query
```

Where DC_Query is the query on Dublin Core metadata and AR_Query is the query on "available rights" on target objects.

The Tree Query User Interface allows the user to build composite queries of arbitrary complexity using logical operators. The composite query is obtained putting together flat queries saved into the database. An example of logical expression built with the Tree QUI (Figure 2) is the following:

```
Query1 OR Query3 OR (Query2 AND
Query7) OR Query4
```

A technical query user interface needs a query specification language shared among peers and components able to interpret this language and produce query-results in remote databases. The AXMEDIS platform provides this query language which is based on XML so that queries can be both passed as arguments in Web Services invocations and embedded in messages of the P2P protocol.

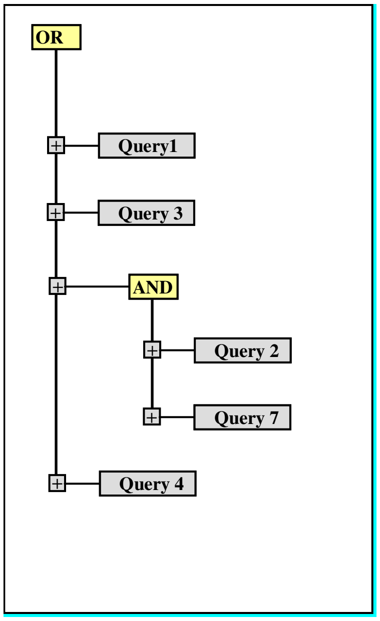

Figure 2. Mock-up of the tree GUI for editing complex Boolean expression in the Query User Interface.

5. The Topology of the Overlay Network

The good choice of the topology of an overlay network has an immediate impact on performances, security, scalability, robustness and fault-tolerance of a P2P system. Often, the choice is based on the appropriate trade-off between centralized and decentralized services which results in a hybrid P2P architecture.

Decentralization contributes to the extensibility and fault-tolerance. On the other hand, a partial centralization makes the network more coherent than a purely decentralized system, since there are fewer hosts that are holding authoritative data. A remarkable property is the scalability of hybrid topologies as proved by large file sharing networks like e-Donkey2000 and Gnutella2.

In our design the network is thus composed by leaf nodes and "super-peer(s)" nodes that centralize the services of query resolution and indexing.

Leaf nodes can connect to one super-peer but cannot accept connections from other peers.

Indexing/searching services are delegated to super-peers which collect queries from their leaves and forward them to the other super-peers that can give the appropriate query results. The query results are sent back to the searching peer who can initiate a download session, on demand, contacting directly the leaf node storing the requested content. This implies that a leaf node should accept incoming connections only for uploading contents to a requester.

A new peer which wants to connect to the network must perform the following steps:

1. *discovery*: using a well-known set of bootstrap URLs the peer gets a list of AXMEDIS super-peers.
2. *Handshake*: the peer tries to connect to a super-peer; if the TCP/IP connection is established then the two peers must handshake. During the handshake the peers must verify each other identities and the compliancy of their implementation with an AXMEDIS tool certification in order to prevent malicious peers to access to unlicensed exercises. So, the peer which wants to connect to the network must have a certificate, released and granted by a reliable certification authority. Once the identity of a peer is verified it is accepted in the network. A super-peer can reject the incoming connection if it has already reached the maximum number of connected nodes. In this last case, the connecting node should iteratively attempt to connect to another super-peer. A node is elected to be a super-peer taking into account several features like capability to receive incoming connections, to maintain several connections, to maintain caches with an acceptable level of performance.

Regarding download services, they cannot be centralized in super-peers. In fact, if all contents moving from a peer to another would be relayed by super-peers the bandwidth used these ones would be huge. Thus, the download of an object is performed by a direct connection requested by the downloader to the storing peer.

In the case of popular downloads a Bittorrent-like strategy can be exploited in order to obtain better performances, to share the workload among concurrent downloader peers, and to keep statistics of downloads at global level deploying a limited number of tracker nodes in the network.

Adopting a BitTorrent-like service, a content file may be broken into smaller pieces and the pieces are randomly distributed among the peers in the network as long as they try to get the entire content. To efficiently download the file, it is important to design the protocol such that each peer is matched with others who have the pieces of the file that it needs and further, to ensure that the downloading bandwidth of each peer is fully utilized[7].

Bittorrent is not exactly a file-sharing system, but rather a download-session-sharing system, as only the original source is really publishing a content file, the other downloaders can only send pieces as long as they are running their download session. In the AXMEDIS platform the nature of digital contents is deeply dependent on the protection model used in the B2B distribution. Thus, the same object may be either distributed in a protected but unique binary file for all recipients or protected and encrypted in a different binary form for each different recipient. In this last case, any download strategy based on file replicas or torrents becomes useless.

6. Conclusion

In this paper we have sketched the design of a P2P content distribution system for the AXMEDIS B2B environment. The main benefits of the proposed architecture are in the automation of P2P related tasks, in the professional query user interface based on Dublin Core and available rights of target objects, and in the preemptive exclusion of malicious participants. Some of these features are the direct result of the integration with the larger AXMEDIS platform which provides a structured query language, a framework for schedulable tasks, and a robust data model for delivering protected objects and enforcing digital rights. Although the main components and solutions have been already identified the implementation of the system is still in its early stage and some of the choices are still subject to modifications according to real tests and to simulations that will be performed on the system.

7. References

[1] "SETI@home", *http://setiathome.ssl.berkeley.edu/*, University of California. Space Sciences Laboratory,1999.

[2] “Distributed.net”, *http://www.distributed.net*

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

[4] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. “Freenet: A Distributed Anonymous Information Storage and Retrieval System”. *Lecture Notes in Computer Science*, volume 2009, year 2001.

[5] B. Cohen, “Incentives Build Robustness in BitTorrent”, *Proceedings of the Workshop on Economics of Peer-to-Peer Systems.* 2003.

[6] “Automating Production of Cross Media Content for Multichannel Distribution”, *http://www.AXMEDIS.org*

[7] Dongyu Qiu, R. Srikant.“ Modeling and performance analysis of BitTorrent-like peer-to-peer networks”. *ACM SIGCOMM, Proceedings of the 2004 conference on applications, technologies, architectures, and protocols for computer communications*, Volume 34, Issue 4, ACM Press.

[8] “Dublin Core Metadata Element Set” - Version 1.1, 20-12-2004 http://dublincore.org/documents/dces/index.shtml

[9] “MPEG REL”, ISO/IEC FDIS 21000-5: Information Technology — Multimedia Framework —
Part 5: Rights Expression Language, August 2003.