SERVICE ORIENTED P2P NETWORKS FOR DIGITAL LIBRARIES, BASED ON JXTA

Marco Pereira, Marco Fernandes, Joaquim Arnaldo Martins and Joaquim Sousa Pinto
IEETA - Instituto de Engenharia Electrónica e Telemática de Aveiro, Universidade de Aveiro, Portugal

Keywords: Peer-to-Peer, Digital Libraries, Service Oriented Architecture, Web Services.

Abstract: Digital libraries tend to be based on centralised models. This centralised approach has both advantages and drawbacks. Regarding the drawbacks we notice that the central server used is a single point of failure that can effectively render a digital library unusable if it fails. In this kind of approach we are also ignoring potential resources that are available in other computers on the network. This paper describes a peer-to-peer network architecture based on JXTA and on SOA (Service Oriented Architecture) that can be used as a support infrastructure for a digital library.

1 INTRODUCTION

Digital libraries can be seen as infrastructures used in the management, storage and distribution of digital content. As such they often employ centralised architectures in the support services that they require as well in the servers that are exposed to the outside. This approach is not without problems, as a failure in any centralised node can render the digital library unusable. Such failure can come from physical events such as power loss or from the inability to cope with increased demands for content.

We proposed that a peer-to-peer network could be used as underlying infrastructure for digital libraries (Fernandes et al., 2008a). With the use of a peer-to-peer network designed to take advantage of free resources that exist on computers (both storage resources and CPU cycles) it would be possible to create a scalable and reliable digital library, always taking into account that peer-to-peer networks suffer from specific problems. One of those problems is that searching for content, which is a key operation in digital libraries, on a peer-to-peer network is not an trivial task. Existing peer-to-peer networks search capabilities are rather limited (often they only allow querying for the file name or type) and depending on the underlying peer-to-peer architecture we can never be 100% sure that we have found all the desired contents. Metadata based search can be used to strengthen search capabilities and carefully chosen network architecture can be used to minimise the loss of search results.

In this paper we begin by introducing a peer-to-peer network architecture, we proceed to describe how Apache Lucene (LUC, 2008) can be used to provide rich search throughout the network and how we can leverage the peer-to-peer infrastructure to provide support services using a webservices based strategy; finally we present some test data from a reference implementation and draw some conclusions.

2 RELATED WORK

There are several projects that seek to harness the power of peer-to-peer networks and use it in digital libraries scenarios. In this section we reference some of those projects.

P2P-4-DL (Walkerdine and Rayson, 2004) is a plugin for the Lancaster P2P Application Framework (Walkerdine et al., 2008) that allows the creation of a digital library based on a peer-to-peer network. It uses a Napster-like system where references (comprised of author, title and keywords) are uploaded to an index peer. The references can then be used when searching the network for content. Since the references are constituted by fixed fields there is little flexibility in the search patterns that can be produced.

EDUTELLA (Nejdl et al., 2002) is a project that aims to provide RDF based metadata search on the JXTA peer-to-peer network, allowing for richer search experiences than the ones currently used on

Pereira M., Fernandes M., Arnaldo Martins J. and Sousa Pinto J. (2009).
SERVICE ORIENTED P2P NETWORKS FOR DIGITAL LIBRARIES, BASED ON JXTA.
In *Proceedings of the 4th International Conference on Software and Data Technologies*, pages 141-146
DOI: 10.5220/0002257501410146

most peer-to-peer applications. EDUTELLA is directed to the exchange of educational content on an internal level (inside an institution) and is presented as a service within the JXTA network. EDUTELLA is only responsible for handling metadata, leaving all the data collection and metadata extraction for available backends.

Freelib (Amrou et al., 2006) is an ongoing open-source project that proposes the usage of a peer-to-peer network based on the Symphony protocol (Manku et al., 2003) to build a digital library. Freelib tries to leverage access patterns to bring content to locations closer to where it is actually needed by directly connecting peers with common interests (or at least place them close enough for them to benefit from the small world effect (Kleinberg, 2000)). Freelib is content-oriented and does not provide mechanism to access support services that a digital library could require.

3 P2P ARCHITECTURE FOR DIGITAL LIBRARIES

For our digital library we want to avoid the centralised design that is the Achilles' heel of the traditional implementations and we also want to minimise loss of search results. With those two requirements we ruled out the possibility of implementing a peer-to-peer architecture based on a pure centralised approach (for example mandatory concentration of network search capabilities in only one peer) and on fully decentralised approach (due to the possible loss of search results). We choose to adopt an hybrid architecture in order to take advantage of the best traits of the centralised and decentralised approach. In our proposed architecture the function of the Super peer will be to collect individual peer indexes to help speed up the search process.

We must recognise that in some special situations it could be better to be in a decentralised network, without any Super peer, so the developed architecture allows for a peer to choose its operation mode. In our architecture a peer can choose to register with a Super peer using it to process queries, or it can remain independent and address queries to all neighbour peers that he knows. Using this strategy the network can survive the loss of all Super peers by having regular peers falling back to a decentralised architecture when such failure is detected. The index that each peer creates contains all the necessary information to retrieve content from that peer. When retrieving content from the network we can explore redundancy. If multiple peers have the same content we can transfer

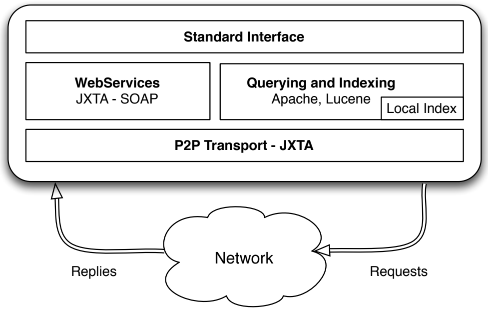

Figure 1: Internal components of a peer.

it in parallel from all of them, to speedup content retrieval process. The peer-to-peer network can be used for more than just storage and retrieval of content. Spare cycles can be used to provide services to the digital library. In order to implement the proposed architecture we use the JXTA (JXT, 2008a) peer-to-peer framework. With this framework we were able to implement the proposed features. JXTA creates an overlay network that makes peers appear to be directly connected with each other when in fact they are not. By using Apache Lucene to index files we create a rich search environment in the network (detailed in section 4). JXTA provides a mechanism for search of resources that can be used to find peers or services on the network. Using this mechanism (called advertisements) and the library JXTA-SOAP (JXT, 2008b) we were able to provide services by creating simple webservices. This process is detailed in section 5.

4 METADATA BASED SEARCH OVER P2P NETWORKS

As was said in section 1 search in peer-to-peer networks is usually limited to the name or type of a file. This is not a desired scenario in a network designed to support digital libraries so we decided to use an Apache Lucene based indexing system (Fernandes et al., 2008b). By using this system we can leverage Lucene's powerful full-text search exposing it to the network, thus creating a rich search environment where, for example, one could search for the author, the year of publication or even content of a book instead of searching for the name of the file that stores the book. This indexing scheme can be extended to handle every conceivable digital media.

Besides metadata specific to each content type we also keep generic information about a file that can be used to retrieve it. That information includes a hash (that allow us to identify the same file stored in different peers), an identifier of the peer that has the file, the physical file location on a peer, the file size and name. These management fields can be used as part of a query, along with content specific fields. There are scenarios where it can be interesting for a peer to index only specific content types (such as DICOM or PDFs files). To support this approach we use Java's reflection capabilities combined with a properties file. The file contains the mapping of the file extensions to an appropriate Document Handler class, and can be specified by any application that uses our library, implementing the necessary infrastructure for the creation of specialised peers.

When using a network with a Super peer, Lucene's generated indexes should migrate from the original peer to the Super peer. This is done to concentrate the search in a few peers, thus reducing the need for flooding the network with query messages. Index migration poses two potential problems: the migration can generate significative amounts of network traffic and migrated indexes can become out–of–date. This issue can be solved by propagating any changes in the indexes to the Super peers. In a digital library scenario where changes in the indexes are not frequent (and when happen they are scheduled) index migration will pose no major problems.

5 WEB SERVICES - SUPPORT OVER A P2P NETWORK

The architecture described in the above sections can handle content search and transfer. This means that we are using the storage resources available in the network, but we can also leverage the peer-to-peer infrastructure to use some of the available CPU cycles to provide support services to our digital library. By using the JXTA-SOAP library we can make our peers offer services using the known webservices interface. For the time being these services are only visible on the peer-to-peer network but they provide to the digital library load sharing capabilities and flexibility, as well as an added layer of reliability. Using this approach services can be replicated (each peer should have a pre-installed set of services), ensuring that there is always at least one peer offer a required service and creating an opportunity for sharing the workload among several peers. Each peer can offer a different set of services, and each service is announced to the network using standard JXTA advertisements (JXT, 2007), that have a WSDL (Web Services Description Language (Christensen et al., 2001)) file embedded. When a peer needs a service it can retrieve these advertisements from the network and choose a

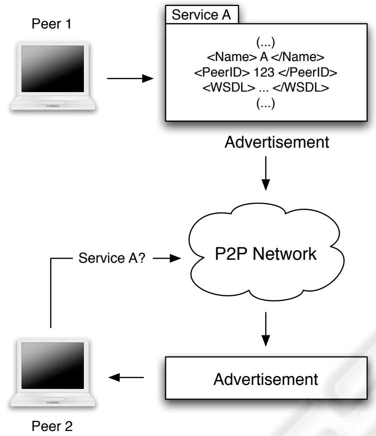

Figure 2: Service discovery.

peer (or several) that is providing the desired service. The service itself is described by the WSDL file embedded in the advertisements, but peers can use an extended set of attributes (such as service name or peer id) placed in the advertisement to discover the service (illustrated by Figure 2). A peer can query the network for services by name and then use the extended attributes to discover more information about the peer that is offering the service, ensuring that he can select the best peer (or peers) for the task. After this phase a peer follows regular procedures to call a webservice (with the difference that instead of having HTTP as transport protocol it has JXTA).

A problem with the current implementation is that only allows the use of byte arrays in SOAP messages for file transfers. This is not an efficient method of moving data and creates limits to the size of the file that can be transferred and is also a possible threat on the stability of the network, as it is possible to render a peer useless by forcing it to run out of memory. The solution would be the use of attachments, but we were unable to use them in the JXTA-SOAP context since when we added the attachment to the SOAP message it disrupted the JXTA message, causing an exception. Alternatives include contributions to the JXTA-SOAP library (to provide proper attachments support) or the use of other transfer mechanisms (such as the use of an reference that would enable the use of an intermediary FTP or HTTP server or even a native JXTA file transfer after the service has been requested). The last alternatives will break the web services facade we are striving to maintain.

6 TESTS

We have selected a series of tests to be conducted on a reference implementation of the peer-to-peer network architecture. These tests are supposed to reflect some operations that would be used on a daily basis on the network. We used a network of 7 computers (described in Table 1) in a local network (100 mbits) environment, with all computers connected to the same Ethernet switch (that provides connectivity with the rest of the laboratory's network). We could have chosen to host several peers on each computer but we believe that this would compromise the results of the service test. Please note that all graphics presented use a logarithmic scale on the Y axis for readability purposes.

6.1 Index Management

Index management can be seen as a two part process. The first part deals with index creation and update while the second part deals with Index migration.

Regarding index creation previous tests (Fernandes et al., 2008b) indicated that Lucene was the best option to provide advanced search and indexing capabilities to a peer-to-peer network. Initial content indexing is a task that should be made before exposing the peer to network. Since Lucene allows queries to be made while it is indexing content but offers no assurance that the contents being indexed will appear in the generated results it is important to plan when to index new content (thus updating the indexes) in order to provide the most complete result set to our network users. Focusing on index migration we can define two scenarios: initial index migration that occurs when a peer joins the network for the first time and elects to transfer its index to a Super peer and migration updates sent to the Super peer when new content has been indexed. We conducted an initial index migration test where indexes migrated from a peer to a Super peer. Index transfer was designed to mirror a regular file transfer (with minor differences on control issues). Test results (seen in Table 2) show a regular increase in migration time as the size of the index grows. Some indexes can achieve sizes in the GB magnitude. The migration of one of those indexes can cause some disruption on the network, so in order to avoid any problems migration times should be carefully chosen, and an indexing strategy that allows incremental updates must be adopted. In extreme cases where a peer has an existing index that is too large to be migrated we could implement a mechanism to allow the Super peer to forward queries to that peer. Based on test result coupled with the previous guidelines index migration will not be a problem in our peer-to-peer network.

Table 1: Peer specifications.

Peer	CPU	RAM	OS
1	Pentium III @ 863MHz	320Mb	OpenSuse 11.0
2	Pentium 4 @ 2.8GHz	512Mb	Windows Server 2003
3	Pentium 4 @ 2.4GHz	1Gb	Windows XP Pro
4	Pentium 4 @ 2.8GHz	2Gb	OpenSuse 11.0
5	Pentium 4 @ 3.2GHz	2Gb	Windows XP Pro
6	Core2 Duo @ 3.0GHz	2Gb	Windows Vista Basic
7	2x Quad Xeon @ 3.2GHz	2Gb	Windows Vista Business x64

Table 2: Time taken to transfer an index.

Index Size (Mb)	Time (s)
4×10^{-5}	0.58
0.86	0.87
1.7	1.17
18	7.53
53	19.78
100	38.67

6.2 Content Search

This test is designed to measure the time that a peer takes to find content on the network. The test is divided in two scenarios: in the first we use a Super peer to centralise all the searches and in the second we remove the Super peer in order to test search on the network when a decentralised architecture is used.

In both scenarios we start with searches that return no results and progress to obtaining each time more results. As expected in both scenarios the time taken to search the network increases as the number of retrieved results also increases. As we can see in Figure 3 the time taken to retrieve results from the network is acceptable. Since the results arrive in small units we can devise a strategy that would allow incremental display of results to a human user in order to increase the overall feeling of responsiveness of the peer-to-peer system.

Comparing both scenarios we can see that using a Super Peer presents better search times when there are only a few results to be sent over the network. This is behaviour can be explained by the overhead involved in establishing connections with other known peers, a network trait when using a decentralised architecture. As the number of results increase the time taken to perform the search in each scenario becomes similar. This can be explained by the increased effort of performing a centralised search and points out that Super peers must be chosen carefully, not by what they can

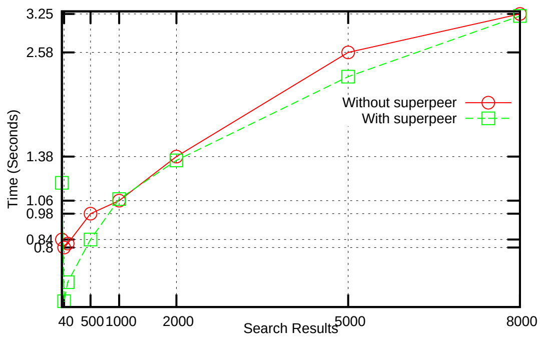

Figure 3: Time taken to search the network.

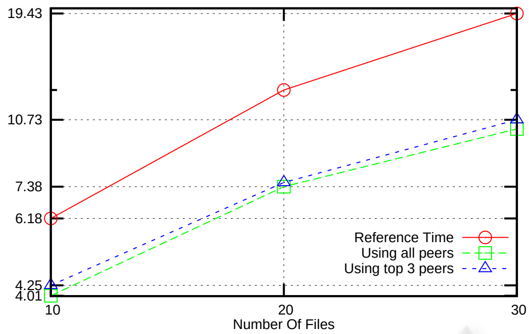

Figure 4: Conversion Times.

offer to the network but by their hardware capabilities (in an ideal scenario Super peers would be dedicated machines).

We can also discuss the results in the context of the actual network topology used. When in the presence of fast connections (as in this case) results show that a totally decentralised network will perform better than a network with a super peer (there is always the possibility of congestion if a large number of peers is present). Since the time taken to search the network in both scenarios appears to converge any network induced delay will have impact on the search time. This shows that when dealing with heterogeneous networks the best choice will be the use of an hybrid topology with carefully chosen Super peers that would ensure some degree of independence from peers on slower connections when searching the network.

6.3 TIFF to PNG Conversion Web Service

In order to test the peer-to-peer implemented webservice we designed a test to measure the time that takes a conversion service to run and return results. Due to constraints (described in section 5) the maximum size of a file (or group of files) is limited. The conversion service is available on six peers and the time taken to convert a batch of files is compared with a reference time obtained when converting the same files in sequence on only one computer (peer 2 described in Table 1). The test files are tiffs generated from the pages of a pdf, with an average size of 980kb.

When the conversion service is used the peer requesting the service is responsible for the distribution of the load as he wishes. In this case a simple strategy of statically assigning the files needing conversion to each peer prior to actually requesting the service, so that each peer had at least one file to convert. The downside of this strategy is that a slow peer potentially receives as much work as a fast peer, increasing the overall time of the conversion. We initially conducted two tests, one using only 3 peers (peers 5, 6 and 7 described in Table 1), selected because we consider that they possess the best hardware configurations and a test using 6 peers (peers 2 to 6 of Table 1). As we can see on Figure 4 the time taken to convert a given set of files using the distributed service was lower than the reference time. It is important to notice that increasing the amount of peers that offer the conversion service yields only marginal benefits when converting small amounts of files, since the overhead of the parallel service calls offsets most of the performance gains that would come from the increased number of peers.

Table 3: Time taken to convert files.

No. of files	Top 3 Peers (s)	With Peer 1 (s)	Ref. Time (s)
10	4.25	8.05	6.18
20	7.55	17.57	12.67
30	10.73	26.9	19.43

We also performed a test to illustrate the impact that a peer single peer can have when it is ill equipped to handle a service. In this test we repeated the first test, replacing peer 5 with peer 1. Table 3 presents the result of this test in comparison with the initial test. As we can see a single peer can have a relevant impact in the time that takes to execute a task. This result underlines the relevance of choosing an appropriate service replication strategy (since offering services in a peer that is ill equipped to handle them is counterproductive) and of the load distribution mechanism, that should take into account hardware specifications as well as the peer availability to actually execute the task. It is important to note that since all peers were on the same local network any transfer or network latency problems that could arise in WAN scenarios are not present.

7 CONCLUSIONS

In this article we proposed a peer-to-peer network that is able to support digital libraries and support services. The tests made with the reference implementation of our proposed P2P network architecture presented interesting results, that we believe that can be improved thus providing a strong incentive to continue this project. The possibility of using the network in both decentralised and hybrid architectures provides flexibility, and by choosing the right architecture according with the number of peers and connection type we can optimise network performance.

On the web services side, The performance gain is interesting, yet raw performance gains are always dependent of the degree of parallelism that a particular application supports. As such we prefer to point out that the added reliability, and the possibility of distributing the workload among different peers in scenarios where multiple services are available in the network are the key features that justify the use of P2P networks on digital libraries scenario. We consider that more work is required in order to provide an optimised strategy for choosing how to distribute the workload in a way that the potentialities of the peers with more resources are fully explored. In the future it could prove to be interesting the creation of a mechanism that would allow service migration, so that when a peer considers that it would benefit the network to replicate an existing service this could be done without human intervention. This is a real possibility due to the mechanisms that allow the discovery of new services.

Although we are aiming to use this peer-to-peer architecture on LANs we should never rule out the possibility of extending it to WANs. This implies that we should develop a way to reliably pass network boundaries and firewalls so that our peer-to-peer network can be used on larger scales. This is a scenario where super peers will play a fundamental role.

ACKNOWLEDGEMENTS

This work was funded in part by the Portuguese Foundation for Science and Technology grant SFRH/BD/23976/2005.
The authors would like to thank João Pereira (a30777@ua.pt) for his help and feedback during data collection.

REFERENCES

(2007). Jxta v2.0 protocol specification. Available from: https://jxta-spec.dev.java.net/.

(2008). Apache lucene. Available from: http://lucene.apache.org/java/docs/index.html.

(2008a). Jxta homepage. Available from: https://jxta.dev.java.net/.

(2008b). Jxta-soap project. Available from: https://soap.dev.java.net/.

Amrou, A., Maly, K., and Zubair, M. (2006). Freelib: Peer-to-peer-based digital libraries. In *AINA '06: Proceedings of the 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06)*, pages 9–14, Washington, DC, USA. IEEE Computer Society.

Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). Web services description language (wsdl) 1.1. Available from: http://www.w3.org/TR/wsdl.

Fernandes, M., Almeida, P., Martins, J. A., and Pinto, J. S. (2008a). Soppa: Service oriented p2p framework for digital libraries. In *ICEIS 2008: 10th International Conference on Enterprise Information Systems*, pages p. 215 – 219.

Fernandes, M., Martins, J. A., Pinto, J. S., and Almeida, P. (2008b). Search engines evaluation for p2p based digital libraries. In *EATIS 2008: Euro American Conference on Telematics and Information Systems*.

Kleinberg, J. (2000). The small-world phenomenon: an algorithm perspective. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170, New York, NY, USA. ACM.

Manku, G. S., Bawa, M., and Raghavan, P. (2003). Symphony distributed hashing in a small world. In *USITS'03: Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems*, pages 10–10, Berkeley, CA, USA. USENIX Association.

Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., and Risch, T. (2002). Edutella: a p2p networking infrastructure based on rdf. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 604–615, New York, NY, USA. ACM.

Walkerdine, J., Hughes, D., Rayson, P., Simms, J., Gilleade, K., Mariani, J., and Sommerville, I. (2008). A framework for p2p application development. *Comput. Commun.*, 31(2):387–401.

Walkerdine, J. and Rayson, P. (2004). P2p-4-dl: Digital library over peer-to-peer. In *P2P '04: Proceedings of the Fourth International Conference on Peer-to-Peer Computing*, pages 264–265, Washington, DC, USA. IEEE Computer Society.