Influence of Replication on Availability within P2P Systems

Timo Warns[1], Wilhelm Hasselbring[1 2], and Mark Roantree[3]

[1] Software Engineering Group, Carl von Ossietzky Universität Oldenburg, Germany,
(timo.warns|hasselbring)@informatik.uni-oldenburg.de
[2] OFFIS Research Institute, Germany,
[3] Interoperable Systems Group, Dublin City University, Ireland,
mark@computing.dcu.ie

Abstract. An increasing number of digital library management systems is developed following P2P architectures to overcome the bottlenecks of client/server architectures. Usually, the participating peers are less dependable than traditional servers. Hence, a P2P system needs to deal with failures of single peers to avoid overall system failures. Replication is a means to improve availability of resources. We empirically investigate the influence of replication techniques on availability by simulations. We focus on voting-based replication control strategies which offer one-copy-serializability in the context of our XPeer architecture.

1 Introduction

Peer-to-peer (P2P) digital libraries are highly dynamic as they shall facilitate data sharing among users. The churn rate is high as their peers enter and leave the system frequently. Hence, the dependability of peers is worse than the corresponding characteristics of traditional servers. Particularly, current P2P systems employ *small-world topologies*, *cross-partition pointers*, *self-organization* and *periodic description updates* to improve dependability [10]. In summary, these means aim at maintaining the topology and communication among available peers in the presence of failing peers.

P2P systems offer services which are delivered by participating peers. The operation of a service requires access to resources, like databases. Usually, these resources are hosted by the peers which deliver the service. If the hosting peers fail, the service crashes jointly. Hence, means of retaining communication and topology do not suffice to improve dependability of P2P services. Means must address the resources hosted on peers as well.

Replication and caching are suitable means to improve availability of resources [6]. Digital library systems provide data and metadata. Usually, caching is preferred to replication for data like audio and video files as this kind of content changes rarely. However, metadata is subject to replication, because it is updated by users frequently, e.g., to rate content. For example, Kovács et al. describe a peer-to-peer digital library which caches content and replicates metadata [7].

P2P systems differ from traditional distributed systems. For example, intermediate peers influence the dependability of communication, and fault characteristics of peers are worse than the corresponding attributes of traditional servers. Therefore, replication has to be investigated specifically for P2P contexts, because solutions for traditional distributed systems may not be appropriate. Our work addresses this issue for the voting-based replication strategies *Read-One-Write-All* (ROWA) and *Majority Consensus* [9] as a first step. Particularly, we explore the availability of read and write operations on replicated resources compared to non-replicated resources.

2 Related Work

Vanthournot et al. propose small-world topologies, self-organization, and cross-partition pointers as means of fault-tolerance within P2P systems [10]. They simulate P2P systems to investigate the resulting dependability. They focus on failures of peers and connections, but omit the issue of resources on peers.

Replication based on rumour spreading is proposed for P-Grid [4]. These algorithms ease consistency conditions to improve availability and performance. They focus on analyzing the communication overhead.

3 The XPeer architecture

P2P systems which are required to be highly dependable in the presence of frequent updates under a sequential consistency model come into question for voting-based replication. Our current work includes the XPeer architecture for data integration within such systems.

XPeer is a logical super-peer architecture which is well suited for digital libraries [1]. It addresses the issue of querying distributed data in a large scale context by realizing an integrated schema. This schema is formed from heterogenous information sources by classifying data sources into domains and creating user profiles for query optimizations. Information sources are integrated using XPeer's novel concept of super-peer application in a database environment. Super-peers are used to integrate information sources from clusters of interest or similarity. However, these sources are prone to disappear in a P2P architecture, causing problems for the query service and the optimisation process. The Replication Service described here is used to extend the original architecture.

4 Method

The availability of reading a replicated resource must be considered separately from the availability of writing, because the behaviour of replication control strategies may differ for reading and writing. We evaluated the resulting values by discrete event-simulation of scenarios. We developed a simulation model

of peers following real-world implementations like Freenet [3] and incorporated techniques of replication.

A scenario consists of a P2P system with specific peers, connections, and their availabilities, a replication strategy and a distribution of replicas to peers. The set of all possible scenarios is infinite. We restrict ourselves to a subset of scenario classes for the investigation. The set of all scenarios can be characterized along five main dimensions: P2P architecture, P2P instantiation, faults, replication architecture, and replication deployment.

The P2P architecture describes the conceptual layout of a P2P system. The dimension is subdivided into degree of centralization, structure, and style of communication. The degree of centralization determines whether a system may have centralized elements, e.g., index servers. The topology of P2P systems may be structured, e.g., to a ring or small-world topology. The style of communication describes whether peers are able to communicate indirectly by relaying and forwarding messages. A P2P instantiation is a derivation of an actual P2P system from an architecture. It describes how many peers participate in the system and how they are connected. The faults dimension specifies the fault characteristics of peers and connections. We assume an exponential distribution of faults. For our purposes, the mean time to failure and the mean time to repair suffice, because the availability and reliability of peers and connections can be derived from these values [8]. The replication architecture describes the conceptual behaviour of replicas. Several classifications of replication are known [5, 12]. The replication instantiation specifies the actual number and distribution of replicas to peers.

5 Results

We derived 36 scenarios classes from the dimensions above. We chose decentralized and super-peer architectures with unstructured, mesh-like, and small-world topologies and indirect communication. The mesh-like topology is a special type of a structured topology, whereby each peer is connected to four neighbors to form a net. A small-world topology is characterized by short paths of intermediate peers between any pair of peers [11]. The number of peers is fixed to 49 for each scenario. The connections are chosen depending on the demands of the topology. The range of mean times to failure and mean times to repair is derived from observations of a real-world system [2]. Two simple types of weighted voting are considered for the replication architecture: Read-One-Write-All (ROWA) and Majority Consensus. The number of replicas is fixed to five for deployment. Their distribution to peers is managed in two ways: They are located on peers with best fault characteristics or are allocated according to a normal distribution. Each scenario class is simulated with a single read or a single write access to acquire the resulting availability. Additionally, each obtained P2P system is simulated with a non-replicated resource to be able to evaluate the relative influence of ROWA and Majority Consensus.

The results of the simulations for each scenario class are presented in table 1. The relative influence of the replication techniques compared to the correspond-

Table 1. Resulting Availability

Resulting Availability for Reading			Replication Control Strategy					
			None		Read-One-Write-All		Majority Consensus	
			Replica Distribution		Replica Distribution		Replica Distribution	
Structure	Fault Distribution		Best Peers	Normal Distr.	Best Peers	Normal Distr.	Best Peers	Normal Distr.
Unstructured	Uniform	Good	0.9993		0.9997		0.9995	
		Medium	0.9762		0.9871		0.9879	
	Normal Distr.		0.9470	0.9468	0.9775	0.9758	0.9781	0.9752
Mesh-Like	Uniform	Good	0.9998		0.9998		0.9996	
		Medium	0.9752		0.9886		0.9858	
	Normal Distr.		0.9692	0.9387	0.9740	0.9747	0.9780	0.9732
Super-Peer	Uniform	Good	0.9994		0.9997		0.9997	
		Medium	0.9628		0.9785		0.9761	
	Normal Distr.		0.9209	0.9099	0.9493	0.9486	0.9470	0.9299

Resulting Availability for Writing			Replication Control Strategy					
			None		Read-One-Write-All		Majority Consensus	
			Replica Distribution		Replica Distribution		Replica Distribution	
Structure	Fault Distribution		Best Peers	Normal Distr.	Best Peers	Normal Distr.	Best Peers	Normal Distr.
Unstructured	Uniform	Good	0.9992		0.9989		0.9995	
		Medium	0.9763		0.9296		0.9879	
	Normal Distr.		0.9470	0.9467	0.9499	0.6382	0.9781	0.9753
Mesh-Like	Uniform	Good	0.9996		0.9981		0.9998	
		Medium	0.9750		0.9271		0.9858	
	Normal Distr.		0.9690	0.9388	0.9220	0.6359	0.9778	0.9730
Super-Peer	Uniform	Good	0.9994		0.9983		0.9999	
		Medium	0.9627		0.9627		0.9760	
	Normal Distr.		0.9207	0.9098	0.6612	0.6106	0.9471	0.9297

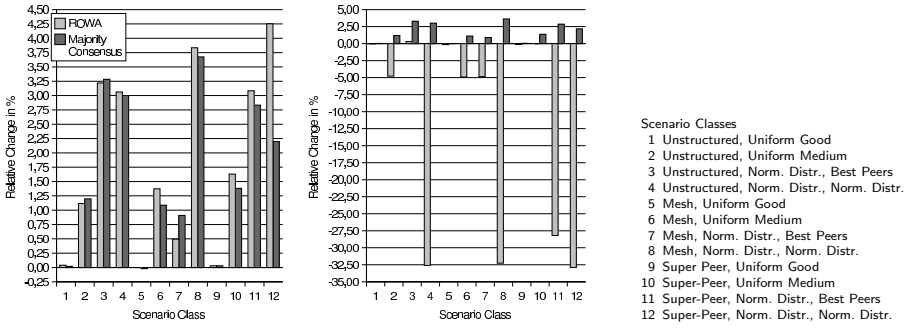

Fig. 1. Relative Change Reading / Writing

ing scenario classes without replication is illustrated in figure 1. The ROWA strategy requires access to a single replica for reading. We expected that the strategy heavily improves the availability of reading. It has its best influence for availability of reading for scenario class 12 with about 4.25%. The worst influence occurs for scenario class 5 where no influence was measured at all. Not surprisingly, it improves reading for most scenarios. However, for some scenario classes $(1, 5, 9)$ the influence is negligible.

The ROWA strategy requires access to all replicas for writing. Hence, all hosting peers must be available for executing a write operation successfully. We expected that the strategy heavily decreases availability of writing, which was confirmed with relative influences ranging from -32.89% to 0.15%.

The Majority Consensus strategy requires access to more than half of the replicas for reading and writing. We expected that the strategy improves both availabilities. The influence for reading was expected to be worse than the influence of ROWA, because access to more than one replica is required. It has its best influence for scenario class 8 with a relative improvement of about 3.68%. The worst influence occurs for class 5 where its influence is negligible. In general our expectation was confirmed as the strategy improves availability for reading and writing. The influence for read operations was worse than the influence of

ROWA. However, it exceeds ROWA for the scenario classes $2, 3$, and 7. It is interesting to see that it does not decrease availability significantly for any scenario class we chose.

A broader generalization of the simulation results is a topic for future work. Even small changes to the scenario may have high impact on the resulting values. However, our results already indicate at this stage that voting-based replication strategies are a favourable replication technique for P2P systems when high consistency is required. Even if the peers of the scenario classes had fault characteristics worse than traditional servers, Majority Consensus improves availability of reading and writing.

References

1. Z. Bellahsène and M. Roantree. Querying distributed data in a super-peer based architecture. In F. Galindo, M. Takizawa, and R. Traunmüller, editors, *Database and Expert Systems Applications, 15th International Conference, DEXA 2004*, volume 3180 of *Lecture Notes in Computer Science*, pages 296–305. Springer, 2004.
2. R. Bhagwan, S. Savage, and G. M. Voelker. Understanding availability. In F. Kaashoek and I. Stoica, editors, *Peer-to-Peer Systems II, Second International Workshop*, volume 2735 of *Lecture Notes in Computer Science*, pages 256–267. Springer, 2003.
3. I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In H. Federrath, editor, *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 / 2001 of *Lecture Notes in Computer Science*, pages 46–66. Springer, July 2000.
4. A. Datta, M. Hauswirth, and K. Aberer. Updates in highly unreliable, replicated peer-to-peer systems. In *23rd International Conference on Distributed Computing Systems (ICDCS 2003)*, pages 76–87. IEEE Computer Society, 2003.
5. S. B. Davidson, H. Garcia-Molina, and D. Skeen. Consistency in partitioned networks. *ACM Comput. Surv.*, 17(3):341–370, 1985.
6. W. Hasselbring. Federated integration of replicated information within hospitals. *International Journal on Digital Libraries*, 1(3):192–208, Nov. 1997.
7. L. Kovács, A. Micsik, M. Pataki, and R. Stachel. Collaboration of loosely coupled repositories using peer-to-peer paradigm. In M. Agosti, H.-J. Schek, and C. Türker, editors, *DELOS Workshop: Digital Library Architectures*, pages 85–92. Edizioni Libreria Progetto, Padova, 2004.
8. M. R. Lyu, editor. *Handbook of Software Reliability Engineering*. IEEE Computer Society Press and McGraw-Hill Book Company, 1996.
9. R. H. Thomas. A majority consensus approach to concurrency control for multiple copy databases. *ACM Trans. Database Syst.*, 4(2):180–209, 1979.
10. K. Vanthournot and G. Deconinck. Building dependable peer-to-peer systems. In *DSN 2004 Workshop on Architecting Dependable Systems*, 2004.
11. D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, June 1998.
12. M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso. Database replication techniques: a three parameter classification. In *Proceedings of 19th IEEE Symposium on Reliable Distributed Systems (SRDS 2000)*, pages 206–217. IEEE Computer Society, 2000.