Adaptive replication strategies and software architectures for peer-to-peer systems

Ludger Bischofs[1,2], Simon Giesecke[1], Wilhelm Hasselbring[1,2], Heiko Niemann[1], and Ulrike Steffens[2]

[1] Carl von Ossietzky University of Oldenburg, Software Engineering Group, 26111 Oldenburg (Oldb.), Germany,
{*giesecke,niemann*}@informatik.uni-oldenburg.de
[2] Kuratorium OFFIS e.V., Escherweg 2, 26121 Oldenburg (Oldb.), Germany,
{*ludger.bischofs,ulrike.steffens*}@offis.de

1 Introduction

The use of replication techniques for data in distributed information systems aims at improving non-functional properties of these systems. Particularly, availability, reliability, and performance should be increased. When considering heterogeneous, autonomy-preserving systems such as distributed digital libraries, the replication techniques employed should be able to adapt to varying properties of the individual systems at run-time.

Adaptive replication strategies need to be realized by a software architecture which provides the context for their implementation. Digital library systems whose information providers are organizationally closely related, for example under the umbrella of a single institution, should be tightly coupled, in order to allow for a high degree of consistency to be maintained. Other digital libraries, such as those of separate publishers, should be coupled less tightly, to allow for retaining their autonomy. This can be realized by a replication strategy based on peer-to-peer (P2P) techniques which combines intra-institutional and inter-institutional replication strategies. The overall replication strategy is evaluated and optimized at run-time, such that domain-specific requirements are incorporated into a multi-dimensional problem model.

2 Data replication strategies

Data *replication* aims at increasing availability, reliability and performance of data accesses by storing data redundantly [1–3]. A copy of a replicated data object is called a *replica*. Replication ensures that all replicas of one data object are automatically updated when one of its replicas is modified. Replication involves conflicting goals with respect to guaranteeing consistency, availability and performance [4]. An improvement of one of these properties usually implies a degradation of the others.

Replication in a distributed system is realized by a *replication strategy*, which is often controlled by a *replication manager*. The quality of a strategy can be measured by the *correctness criterion* it safeguards. The strictest criterion is that of *one-copy-serializability*: The concurrent execution of a set of distributed transactions is one-copy serializable, if its effect is equivalent to the sequential execution of the transactions on a non-replicated database [5]. Besides this criterion a number of other correctness criteria have been defined, which are less strict and thus easier to fulfil.

Replication can be performed in an *eager* or *lazy* way [6]. In the case of eager replication, when one replica is modified by a transaction, the other replicas of the concerned data object are updated within the original database transaction, as opposed to lazy replication where only the originally accessed replica is updated within the original transaction, while the other replicas are updated in separate transactions. The classification into *synchronous* and *asynchronous* replication strategies is essentially equivalent to this classification. Combinations of synchronous and asynchronous replication have also been studied [7, 8].

The concept of an adaptive replication manager was first proposed in [9]. Its goal is to achieve an optimal trade-off between tight and loose coupling of information systems [10], which is realized by dynamically switching participating systems between being synchronously and asynchronously updated.

3 Peer-to-peer systems

Peer-to-peer systems are distributed systems that follow a communication model of equal nodes (peers) which communicate directly with each other. In the following we introduce different styles of peer-to-peer architectures before we describe the specific benefits of replication within peer-to-peer architectures.

3.1 Styles of peer-to-peer architectures

The class of peer-to-peer architectures can roughly be divided into pure and hybrid peer-to-peer architectures. *Pure peer-to-peer architectures* are completely decentralized. Each peer in the network is equipped with both client and server functionality and the architecture does not contain any central servers at all.

Hybrid peer-to-peer architectures combine characteristics of pure peer-to-peer architectures and client/server architectures. Some services are offered by servers and are therefore centralized, whereas other services are based on the peer-to-peer communication model. Hybrid architectures can further be subdivided into *centralized architectures* and *super-peer architectures*. In centralized architectures there are nodes with pure server functionality which offer exclusive services. Super-peer architectures are also called *hierarchical architectures* and combine the concepts of decentralized and centralized architectures. A super-peer is a peer which acts as a server for a set of ordinary peers and usually interacts with other super-peers. Ordinary peers are typically organized in clusters together with a single super-peer. Super-peer architectures are particularly suited to support both intra- and inter-institutional cooperation because they allow for representing the structures of an institution within the super-peers' clusters as well as the structures among different institutions by connecting the different super-peers accordingly.

3.2 Peer-to-peer replication

Today's replication strategies often depend on centralized solutions. Replicating heterogeneous, autonomy-preserving information systems, however, requires flexible and well-coordinated approaches. Intra- and inter-institutional replication should be addressed by an integrated approach, where institutions are coupled via a suitable peer-to-peer infrastructure.

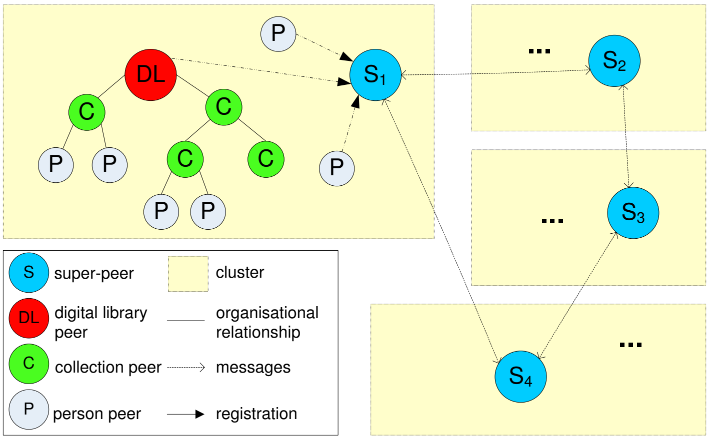

Fig. 1. Organization-oriented super-peer architecture for digital libraries

In figure 1, an organization-oriented super-peer architecture reflecting the requirements of different cooperative digital library systems is shown. This architecture is one example of how to model specific organizational requirements by extending peer-to-peer architectures by an additional layer which provides information on the organizational context (cf. [11–13]). The architecture forms a basis for intra- and inter-institutional integration and replication of distributed data resources and services. Digital libraries together with specialized subcollections as well as individual users are represented by different types of peers. The connections among these peers as shown in the upper right of figure 1 are only one example of how a library may be organizationally structured. Other, non-hierarchical structures are also supported. Documents and also services may be controlled by the digital library as superordinate institution, by single collections and also by single person peers who e.g. can offer their individual metadata or link lists to interesting web documents.

The intra-institutional structure as described above is complemented by an inter-institutional coupling with the help of super-peers. The super-peers impose a partition into clusters of organizationally closely related peers, where one cluster may comprise one or more institutions. Super-peers have more responsibilities than ordinary peers. A super-peer stores metadata concerning the structure of the institutions under its auspices and of the data resources and services it provides. Thus, its main task is to mediate between the different clusters.

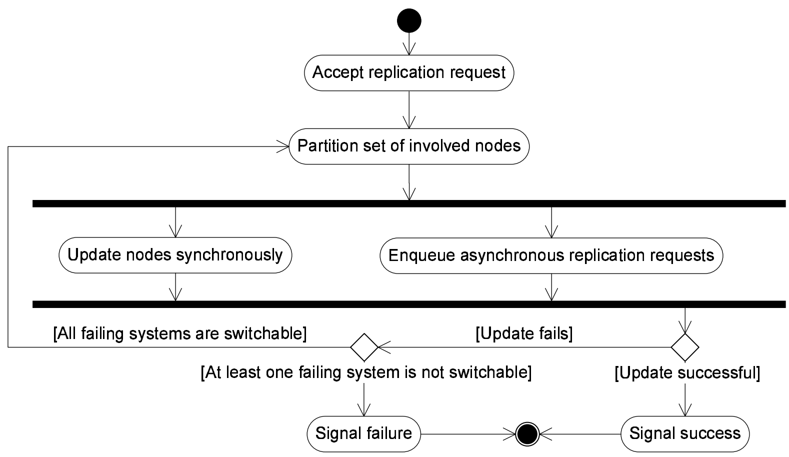

Fig. 2. Adaptive replication strategy

The structure given by an organization-oriented super-peer network can serve as foundation for the realization of adaptive intra- and inter-institutional replication strategies. Within a single institution the overall information system is usually composed of multiple component information systems, which must replicate data permanently due to high availability requirements. An advanced replication strategy is required to facilitate both high consistency and autonomy of the component information systems, each of which must not be vitally impaired when other systems fail. An optimal trade-off between these conflicting requirements can be reached by combining synchronous and asynchronous replication into an *adaptive replication strategy*. Each system may be replicated either synchronously or asynchronously at any time. Switching between synchronous and asynchronous replication should be configurable and adaptive with respect to the current network configuration status. The configuration is performed using a system of rules, which is continuously evaluated at run-time to ensure adaptivity. The correctness criterion of the replication strategy is evaluated by means of an appropriate consistency measure, i.e. a measure of the probability of consistent accesses.

A transaction—or, in this context, a replication request—is initiated by any node connected to the peer-to-peer network and is sent to the replication manager. In figure 2, the process realizing an adaptive replication strategy is illustrated for one replication request. By evaluating the system of rules, the involved nodes are split into two groups, those that are to be updated synchronously and asynchronously, respectively. After that, both groups of nodes are processed. The nodes in the synchronous group are directly updated, while the update requests for the nodes in the asynchronous group are enqueued into a message queue. All node updates are performed in parallel within one transaction. If all synchronous updates were successful, the processing of the replication request is completed and positively acknowledged. If a synchronous update fails, the replication manager checks whether the failing systems may be switched to asynchronous update mode. If this is the case, the corresponding systems are switched and the processing of the replication request is restarted. If some system could not be switched, the transaction fails.

Asynchronous replication requests in the message queue are continuously processed in an independent thread of execution. If a system to be updated is not available, the corresponding request stays in the queue until the update has successfully been performed.

4 Peer-to-peer replication for digital libraries

Today's digital library systems cooperate in manifold ways. They exhibit a varying internal organizational structure, e.g. given by the introduction of specialized subcollections or by extraction of project-specific reference libraries [14]. Furthermore, different business models also including the ability to count the cost of library access have to be taken into account [15]. Against this background, the ability to map intra- and inter-institutional requirements to the underlying digital library system as presented above is of paramount importance. Libraries including resources other than traditional library documents like e.g. resource collections in e-science [16] or health information systems [11, 17, 13, 10] emerge. This calls for the determination of detailed strategies not only for searching these collections [18] but also for replication of both resources and services as claimed in this paper.

References

1. Bernstein, P.A., Goodman, N.: The failure and recovery problem for replicated databases. In: Proceedings of the Second Annual ACM Symposium on Principles of Distributed Computing, New York, NY, ACM Press (1983) 114–122
2. Mustafa, M., Nathrah, B., Suzuri, M., Osman, M.A.: Improving data availability using hybrid replication technique in peer-to-peer environments. In: Proc. 18th International Conference on Advanced Information Networking and Applications (AINA'04), IEEE CS Press (2004) 593–598
3. Loukopoulos, T., Ahmad, I.: Static and adaptive data replication algorithms for fast information access in large distributed systems. In: Proc. 20th International Conference on Distributed Computing Systems (ICDCS 2000), IEEE CS Press (2000) 385–392

4. Hasselbring, W.: Federated integration of replicated information within hospitals. International Journal on Digital Libraries **1** (1997) 192–208
5. Traiger, I.L., Gray, J., Galthier, C.A., Lindsay, B.G.: Transactions and consistency in distributed database systems. ACM Transactions on Database Systems **7** (1982) 323–342
6. Gray, J., Helland, P., O'Neil, P., Shasha, D.: The dangers of replication and a solution. SIGMOD Rec. **25** (1996) 173–182 Proceedings of the SIGMOD 1996 conference.
7. Lubinski, A., Heuer, A.: Configured replication for mobile applications. In Barzdins, J., Caplinskas, A., eds.: Databases and information systems, Dordrecht, Niederlande, Kluwer Academic Publishers (2001) 139–151
8. Röhm, U., Böhm, K., Schek, H.J., Schuldt, H.: FAS – A Freshness-Sensitive Coordination Middleware for a Cluster of OLAP Components. In: 28th International Conference on Very Large Data Bases (VLDB 2002). (2002) 754–765
9. Niemann, H., Hasselbring, W., Hülsmann, M., Theel, O.: Realisierung eines adaptiven Replikationsmanagers auf Basis der J2EE-Technologie. In: Datenbanksysteme für Business, Technologie und Web (BTW). Volume 26 of GI-Edition – Lecture Notes in Informatics., Bonn, Bonner Köllen Verlag (2003) 443–452
10. Niemann, H., Hasselbring, W., Wendt, T., Winter, A., Meierhofer, M.: Kopplungsstrategien für Anwendungssysteme im Krankenhaus. Wirtschaftsinformatik **44** (2002) 425–434
11. Bischofs, L., Hasselbring, W., Schlegelmilch, J., Steffens, U.: A hierarchical super peer network for distributed artifacts. In Agosti, M., Schek, H.J., Türker, C., eds.: DELOS Workshop: Digital Library Architectures. (2004) 105–114
12. Bischofs, L., Hasselbring, W.: A hierarchical super peer network for distributed software development. In: Proc. Workshop on Cooperative Support for Distributed Software Engineering Processes (CSSE 04). (2004) 99–106
13. Bischofs, L., Hasselbring, W., Niemann, H., Schuldt, H., Wurz, M.: Verteilte Architekturen zur intra- und inter-institutionellen Integration von Patientendaten. In: Tagungsband der 49. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie (GMDS 2004). (2004) 87–89
14. Schmidt, J.W., Schröder, G., Niederée, C., Matthes, F.: Linguistic and Architectural Requirements for Personalized Digital Libraries. International Journal of Digital Libraries **1** (1997) 89–104
15. Weber, R.: Chablis - Market Analysis of Digital Payment Systems. Institutsbericht, Technische Universität München, Institut für Informatik (1998)
16. Frommholz, I., Knezevic, P., Mehta, B., Niederée, C., Risse, T., Thiel, U.: Supporting Information Access in Next Generation Digital Library. In: Sixth International Workshop on Digital Library Architectures, Cagliari, Italy, Edizioni Progetto Padova (2004) 49–60
17. Wurz, M., Brettlecker, G., Schuldt, H.: Data stream management and digital library processes on top of a hyperdatabase and grid infrastructure. In: DELOS Workshop: Digital Library Architectures. (2004) 37–48
18. Klas, C.P., Fuhr, N., Schaefer, A.: Evaluating Strategic Support for Information Access in the DAFFODIL System. In: Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2004). Volume 3232 of Lecture Notes in Computer Science., Springer (2004) 476–487