eChallenges e-2009 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2009
ISBN: 978-1-905824-13-7

A Decentralized Wiki as Web 2.0 Enterprise Collaboration Platform

Udo BARTLANG[1], Christoph GERDES[1], Jörg P. MÜLLER[2]
[1]*Siemens Corporate Technology, Information and Communications,*
Otto-Hahn-Ring 6, Munich, D-81739, Germany
Email: {udo.bartlang.ext, c.gerdes}@siemens.com
[2]*Clausthal University of Technology, Department of Informatics,*
Julius-Albert-Str. 4, Clausthal-Zellerfeld, D-38678, Germany
Email: *joerg.mueller@tu-clausthal.de*

Abstract: Wikis are major applications of the Web 2.0; they are used for a large number of purposes, such as project documentation, and coordination, in open communities and in enterprises. At the application level, users are targeted as both consumers and producers of dynamic content. Yet today, this kind of peer-to-peer principle is not used: at the technical level, wikis are built using traditional client-server architectures. What is lacking is a generic platform that combines the scalability of the peer-to-peer approach with, e.g., a wiki's capability for consistent content management in a highly concurrent environment. This paper presents methods for a flexible content repository system that is intended to close the gap by using a hybrid peer-to-peer overlay to support scalable, fault-tolerant, and consistent data operations for the dynamic content of wikis. Major building blocks to enable peer-to-peer data management at the system's persistent storage layer are highlighted, and we show how these can be used to implement a peer-to-peer based wiki application, consisting of: (i) a peer-to-peer back-end to administrate a wiki's actual content resources; (ii) peer-to-peer service groups acting as indexing groups to implement a wiki's search index.

1. Introduction

Gartner [1] among others have observed converging key trends that drive the need for distributed content management, such as the growing role of working over the Internet and distributed collaboration within enterprises, which require sharing produced data. Especially the explosion of unstructured content complicates filtering, administration, and controlled exchange. Intra-enterprise knowledge management aims to facilitate and optimize retrieval, transfer, and storage of knowledge. However, the sole exchange of such content is difficult: inconsistencies in redundant content may lead to problems and require additional efforts [2]. The common practice in enterprises to employ various storage locations, e.g., an employee's local workstation, group storage devices, or intranet servers demand for content consolidation.

In recent work [3] the usage of specialized content repositories was advocated to enable the management of both structured and unstructured content. Such repository typically acts as a meta layer providing additional capabilities on top of traditional persistent data stores, such as database management systems. Today, design and implementation of content repositories are primarily based on a centralized architecture. Such client-server architectures are well suited for static networks and computing infrastructures, where the need for hardware resources can be predetermined. Considering, however, availability of crucial content, if the single server fails, the whole system service is no longer available, which is known as single point of failure.

In contrast, the peer-to-peer (P2P) paradigm offers a more flexible communication pattern migrating into more and more application domains. For instance, there has been significant increase of P2P-based systems regarding popularity and their employment for content distribution in the Internet [4]. The increase in storage capacities, processor power of commodity hardware, and technological improvements to network bandwidth accompanied by reduction of its costs foster decentralized solutions by pushing computer power to the edge of networks. E.g., today even commodity desktop machines are able to store huge amounts of content data and to act as a basis for building sophisticated computing infrastructures [5].

Today, enterprises are globally distributed over multiple locations; hence, the need for a shared platform arises to support collaborative knowledge management. A corporation organized in globally distributed business units can show complex hierarchies w.r.t. the number of domains or management of knowledge content. E.g., each of the participating departments may maintain its own view of the enterprise world.

The usage of a wiki promises to combine the sharing of intra-enterprise knowledge with low administration efforts. We argue that wikis as major applications of the Web 2.0 can be deployed to cope with the above mentioned issues. What is needed to realize this potential is a generic platform that combines the scalability of the P2P approach with a wiki's capability for consistent content management in a highly concurrent environment. To close the gap, we present relevant building blocks of our flexible content repository system which uses a hybrid peer-to-peer overlay to implement a decentralized wiki application: that is, in the context of this paper, we emphasize the technical perspective of the solutions to enable P2P data management at the system's persistent storage layer.

This paper is structured as follows. Section 2 describes the objectives of a P2P-based wiki. Our applied methodology is given and the major P2P developments are introduced in Section 3. Subsequently, a technical description of the system's major building blocks is shown. In Section 5 the approach is evaluated; in Section 6 we conclude and discuss future areas of research.

2. Objectives

From a technical perspective, a wiki represents a network-based information collection. Figure 1 illustrates the basic architecture of a centralized wiki system. The core challenge for a decentralized wiki is to distribute content management functions and storage while preserving consistency in the face of concurrent requests.

The scenario assumes that more and more projects require collaboration of geographically distributed partners to exchange content data, or rather knowledge; these persons may belong to different departments, which demands for collaboration across hierarchical boundaries - which is difficult to achieve using centralized client-server based systems [2].

P2P content management shall simplify knowledge cooperation by administrating content in one virtual place. This way, it shall facilitate dissemination of content to all interested parties. Thereby, the inherent degree of distribution shall be transparent to users.

In addition, the scenario assumes that the amount of available content in an enterprise is growing permanently. This growth complicates management and maintenance of content. A state of the art approach uses a centralized architecture to implement the application logic of a wiki and to administrate its content. E.g., the usage of geographically distributed cache servers for content distribution may benefit basic read requests. However, update requests are crucial as they target the central database. In addition, the centralized architecture raises both technical and financial issues for its operator: (i) from a technical point of view, a centralized wiki architecture shows modest scaling, because of employed static, central components. This is especially the case in the face of large media data or great amount of

abrupt content requests, so called flash crowds. Thus, employing a single site would be a bottleneck for the system. (ii) From an economic point of view, the fully replication of content at all department sites is often neither practical nor cost-effective. However, centralized components would typically constitute the majority of costs of such system. This raises the question to spread infrastructure costs in a fair way among the departments. In addition, power consumption may impose a restriction to how large the central location is able to grow in size. In the following, we take the position that described issues can be solved by using a P2P-based repository to implement a decentralized wiki application.

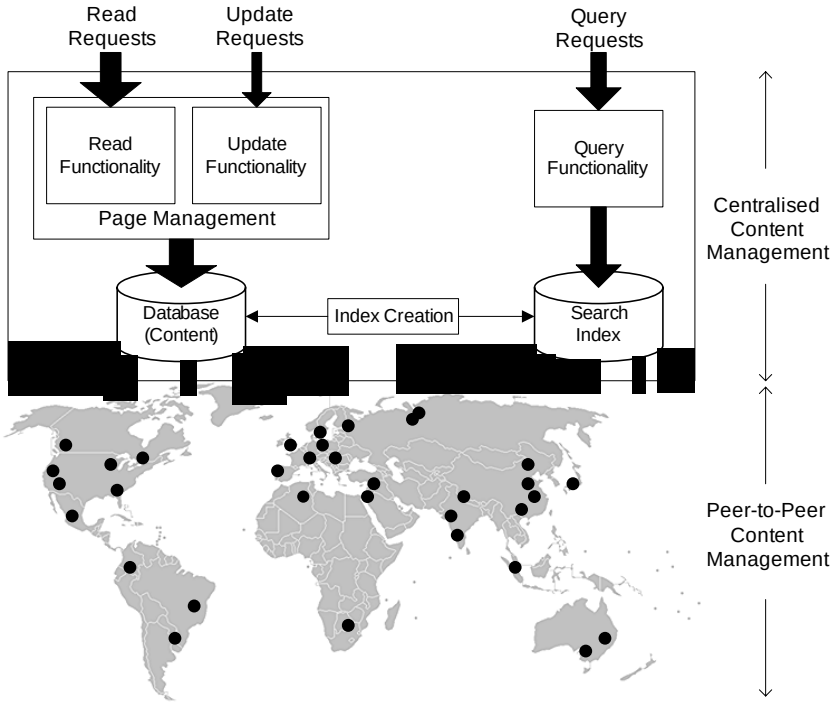

Figure 1: Towards a Decentralized Wiki Architecture

3. Methodology

We recommend using a specialized content repository architecture to enable the management of both structured and unstructured wiki content resources; employing dedicated content repositories is a change in perspective of content lifecycle management. Even with evolving efforts, however, to facilitate this shift of content management perspective today's content repositories are not flexible regarding the support of different content models, offered functionality as dynamic runtime reconfiguration, or distributed system models. E.g., despite the cognition to distinguish between types of content, explicitly known semantic of content data (e.g., its degree of importance) is neglected. But semantics of such knowledge regarding certain content types may be exploited, e.g., to optimize overall system performance.

This paper presents the method of using our flexible content repository system [3] to implement a P2P-based wiki engine achieving a more decentralized vision of a dynamic environment for the future Web 3.0. The P2P-based content repository system enables building the vision of an enterprise-wide wiki as a shared knowledge space and a shared structure of content organization: however, the system is not just limited to a wiki application. To be both scalable and showing good performance, its major functions are reconfigurable to enable a policy-based approach for content management. As most important feature the system supports fault-tolerant and consistent content management: as, once (important) content is stored to the system, it shall not be lost. This raises the

challenge to coordinate concurrent activity in a dynamic P2P environment and to protect the consistency of created artefacts to keep content up-to-date across geographically distributed locations.

We adopt a modular content repository approach considering horizontal and vertical system decomposition: horizontally, the degree of distribution of content repository functionality regarding the persistent storage support may vary. Vertically, different modules are responsible for different management tasks [3]: in the scope of this paper, however, we concentrate on the major building blocks to enable P2P data management at the system's persistent storage layer and how these building blocks can be used to implement a P2P based wiki application: (i) a P2P distributed hash table (DHT) back-end administrates a wiki's actual content resources. On top, (ii) P2P service groups act as indexing groups to implement a wiki's search index (see Figure 2).

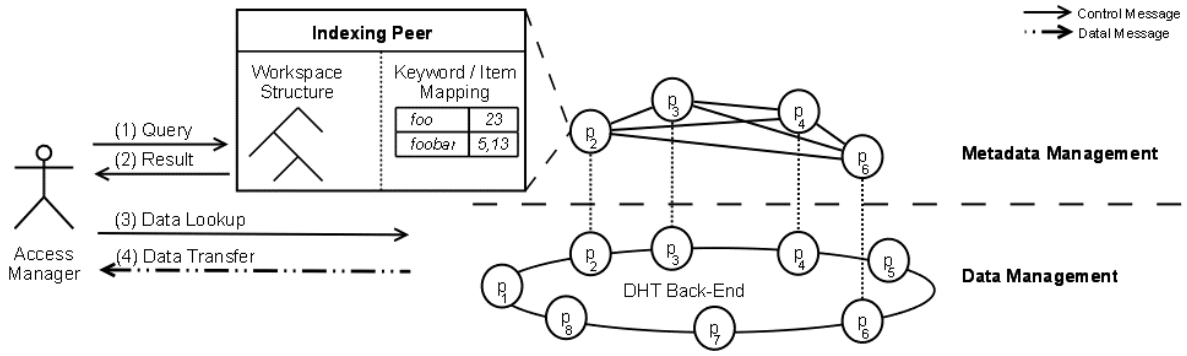

Figure 2: Content Retrieval in Case of a Hybrid P2P Overlay Network

At the P2P back-end level, the basic concept of a hybrid overlay is the basis to use P2P service groups (indexing peers) as a building block on top of a DHT (storage peers) to implement a persistent content storage using the decoupling of metadata management and data management. The reason of using two different roles of peers lies in the nature of DHTs: DHTs perform quite well for keyword-based put-get operations but diminish if executing more sophisticated queries [6]. P2P service groups provide a way to break the symmetry of peers and to exploit their diversity; they are introduced as a method to implement a distributed, replicated, and fault-tolerant search index. An important aspect of the concept is the establishment of a consistent inner-group communication mechanism.

Challenges for P2P solutions arise regarding the maintenance of data consistency:

- By using the DhtFlex algorithm we achieve consistent data operations for the DHT back-end.
- By combining distributed consensus algorithms with additional mechanisms we can implement a reconfigurable P2P group communication method enabling consistent data management in the face of dynamically changing memberships without service interruption.

4. Technology Description

P2P content management simplifies knowledge exchange by administrating content in one virtual place. This way, it facilitates dissemination of content to all interested users—the inherent degree of distribution is transparent to users. Our targeted scenario imposes different requirements: (i) a content view demands the system to deal with huge amounts of wiki pages and potentially large media files. (ii) A functional view characterizes the life cycle of wiki pages by continual modifications: new pages may be created; existing pages may be read or updated. The update of a page results in the creation of a new version. Change tracking is supported by push-based notifications, if changes apply to content of interest. The employment of an access control mechanism allows for user authentication to

support enterprise-wide or department-wide modification of content. As the editing of shared content may result in concurrent modification requests, locking enables exclusively blocking of content against undesirable update access. Query functionality shall support the searching for pages. (iii) A P2P view supposes the system to be self-organizing to handle dynamic arrivals and departures of peers; the system operates a decentralized method to determine the logical placement of content, as the physical location of content may change regularly. In the following, we explain how these requirements can be supported.

4.1 Reconfigurable P2P Service Groups

P2P service groups serve as method to implement a distributed, replicated, and fault-tolerant repository index. Hence, a P2P service group represents some kind of partitioning scheme of the world of peers; e.g., to foster performance, communication, or logical locality. Thereby, such P2P service group is reconfigurable: (i) peer group memberships can change dynamically at runtime; in addition, the offered service can be (ii) deployed and (iii) reconfigured applying some policy. The lifecycle management of these groups includes the discovery of suitable peers.

An important aspect of the concept is the establishment of a consistent inner-group communication mechanism. Therefore, such service group uses an encapsulated consensus module as inner-group communication component [8] to support the building of replicated state machines. The special aim of replicated P2P state machines is to benefit repository functions working at deep operational scope of a wiki's distributed content tree: the replication of relevant content item metadata on different peers is a useful redundancy for improving availability. But such multi-peer replication has the potential to increase performance, too; on the one hand, the selecting of a nearby group peer to serve a query request may result in shorter service time. On the other hand, fewer peers and communication messages may be involved in such query process within a group; e.g., no overlay lookup costs may be required to send messages between replicas.

Locally, each indexing peer maintains a view of the fault-tolerant index itself, and the data structures to actually create it. All such data structures shall reflect a historical record of critical metadata changes and are kept in an indexing peer's transient local memory and persistent local storage. The usage of such replicated index enables a reliable update of an indexing peer's state without the risking of inconsistencies in case of peer failures: therefore, the generic consensus module is used─accessed by the P2P group communication system. Hence, an indexing peer is able to restore its state by replaying the relevant data structures. In order to keep their history small, however, a checkpointing mechanism can be used if the size reaches a certain limit. Outdated peers may access up-to-date information by using some file transfer protocol for checkpoint exchange.

To increase availability and fault tolerance such workspace index is replicated among multiple indexing peers and a client's request is served only after flushing the corresponding record to disk, both locally and remotely─using the running consensus protocol instance. Here, several operations may be batched together to increase system throughput.

4.2 Flexible Atomic Data Management for the P2P Back-End

The P2P system supports complex queries using indexing peers. In addition, the DHT layer provides a basic put/get interface for keyword-based retrieval of the actual wiki content resources (e.g., textual, graphical, or video content). Combined with additional replication strategies, the DHT back-end promises high availability for published data resources; i.e., a certain data resource instance is replicated at different physically located peers, called its replication group. However, regarding the support for atomic data operations to ensure consistent wiki content, replication comes at the cost of maintaining data consistency: an

atomic data operation on a certain resource has to be consistently applied to all of its replicas; e.g., the update of a certain wiki page.

The system uses DhtFlex [8] to support flexible atomic data operations for replicated data. DhtFlex is a distributed algorithm that is trimmed for such a highly concurrent and fluctuating environment, where peers may fail with high rate. DhtFlex bridges the gap between the requirements of a DHT service and the benefits offered by a structured key-based routing overlay. A key concept of DhtFlex is to ensure consistent replication group configurations by using distributed consensus. DhtFlex serializes concurrent put and get requests over the master of a replication group accelerating these operations; in addition, it is able to deal with master failures by automated handovers—ensuring consistency. The approach uses techniques that extend a DHT in order to deal with the requirements emerging of supporting content repository functionality. DhtFlex supports both immutable as well as mutable data resources and offers flexible consistency strategies for atomic data operations. E.g., once a certain version of a wiki page is defined, it remains forever unchanged within the corresponding version chain.

A replicated content resource is independent of the peer on which it resides and may be regarded as virtual. This applied virtualization enables DhtFlex to employ structured overlay routing as partitioning strategy, e.g., applying the Chord protocol [9]. Thereby, DhtFlex manages all replication functions; the overlay is accessed only to conduct necessary information to construct a replication group.

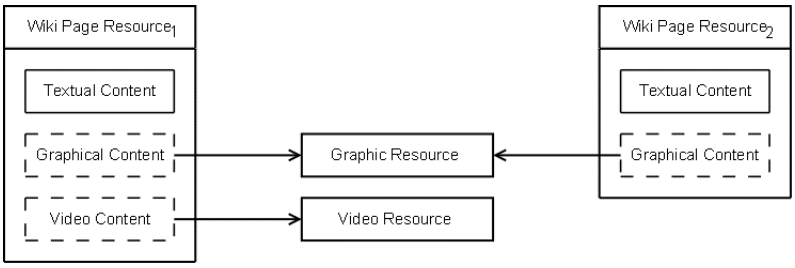

Figure 3: Mapping Wiki Contents to P2P Data Resources

Applying the described approach, the system enables (i) mapping wiki pages and (ii) different bundling of a page's data resources (as illustrated in Figure 14): e.g., (small-sized) textual content may be attached to a wiki page's representing content resource, on the one hand. On the other hand, different pages may share common (large-sized) multimedia contents, and different transport protocols may be used to retrieve them on demand.

The system supports flexible content repository functions in a P2P environment: DhtFlex enables to represent a wiki page as mutable data resource but to keep a single version of it as immutable resource. Resources can be locked to prevent undesirable update access. In addition, indexing peers can enable change tracking by supporting (deep) push-based notifications.

5. Results and Related Work

In the scope of this paper, we focus on the discussion of scalability properties for the P2P back-end and the P2P service groups. Indexing groups represent critical parts in the hybrid overlay: i.e., as these play a major role for supporting rich queries, the system shall select only peers showing good properties as group members; e.g., considering such peer's hardware resources high processor throughput, and large primary and secondary storage space is demanded. In addition, group members should communicate at high network connection speed to reduce message latencies (e.g., by being located in physically close distance).

Considering the DHT back-end, the total data load of the system is defined as the sum of the data loads of all participating peers. The data load of each peer refers to the amount

of data resources a peer is responsible to store locally. In the following, we show the evaluation of the distribution of 1000 data resources with varying replication factor ρ to a simulated network of 1000 peers. According to the wiki scenario the data resources represent the 1000 most viewed articles of the English version of Wikipedia [10] in August 2008 to indicate if DhtFlex's partitioning strategy is suited:

- SHA-1 is used as hash function to create both peer and data resource identifiers.
- Each peer is responsible for a certain (key) segment of the overlay: thereby, although the hash functions is used to achieve good distribution of peers in the overlay, the size of segments may vary.
- The following schema is applied to create the unique name for a content resource representing a certain Wikipedia page: $\underbrace{en.wikipedia.org/wiki/}_{\text{namespace}}.name_{article}$

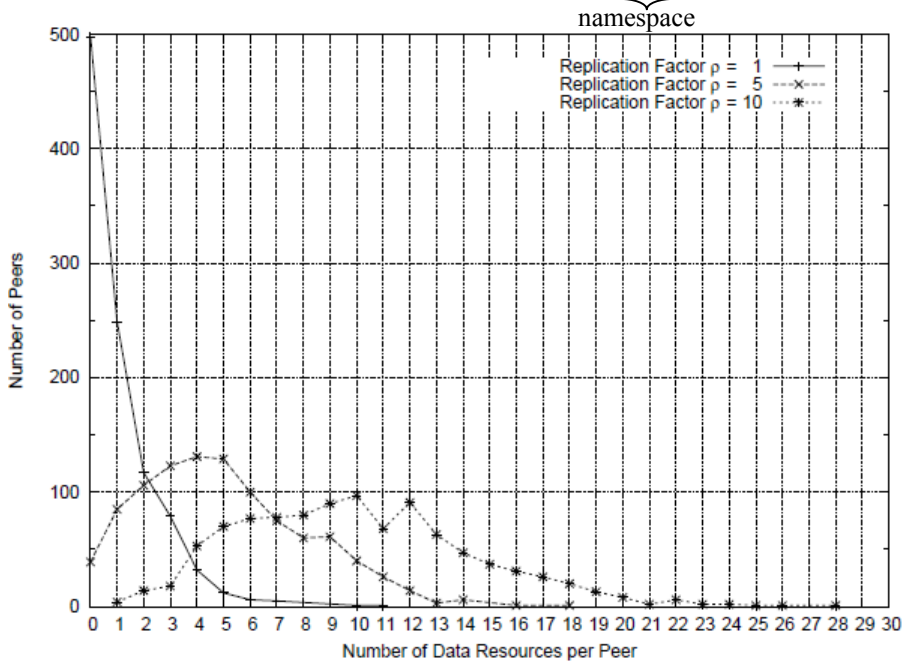

Figure 4: Data Distribution of 1000 Wiki Content Resources on 1000 Peers

Figure 4 shows the results of the experiments: in all cases, the theoretically expected value μ is achieved as kind of centre of distribution: e.g., μ=10 in case ρ=10 (10000 data resources). In general, the data load on each peer scales well with the number data resources and varying values of ρ: no hotspots are detected.

On the one hand, increasing the value of ρ seems to level data distribution; on the other hand, the number of peers not storing a resource may be decreased. DhtFlex can ensure load balancing by uniform distribution of peer and data resource identifiers, where the number of data resources stored at each peer is roughly balanced.

The scalability for indexing peers is indicated by a qualitative discussion of the applied data querying strategy: indexing peers support querying for data resources by their materialized (local) view of content metadata: (i) On the one hand, this may increase lookup performance in comparison to using the P2P back-end, exclusively─especially, if the addressed search space is large: e.g., the back-end is only used to retrieve actual content data (the query results); i.e., data transfer can be decoupled from metadata management. (ii) One the other hand, indexing peers are affected by more request load than usual (back-end) peers and need to ensure consistency of their local index.

Regarding related work, to our knowledge no content repository system has been proposed to be flexible enough to implement a P2P-based wiki combining the scalability of the approach with fault-tolerance and consistency properties. Considering related work for the future Web 3.0, for example, Urdaneta et al. [11] indicated a proprietary architecture of

a decentralized wiki engine using a gossiping protocol for the data management of dynamic content. However, their proposal is not evaluated and does not support all the functional properties (e.g., locking, observation) nor the non-functional properties (e.g., flexibility, consistency) of our approach. In addition, there exist several P2P-based systems to enable collaborative content distribution (e.g., to cache Web pages) [12]. But, all these systems focus only on static content and do not consider collaborative working on dynamic content.

6. Conclusions

This paper presented and evaluated a flexible P2P-based approach to build a decentralized wiki, which aims to reduce creation and maintenance costs. Presentation of content is decoupled from its background organization and storage location to support construction of web page instantiations on demand. The decoupling of design management, data structures, and content supports reuse of content. The system operates different functions to remove central components avoiding single points of failure. E.g., it uses content replication strategies to be less vulnerable to failure of individual network nodes and connections. The P2P system uses a hybrid overlay and a flexible approach to implement the required functions in the distributed environment. It is self-organizing to reduce administration and ad-hoc integration efforts; the approach offers the potential to reduce operating central resources like dedicated server hardware. However, the establishment of a hybrid approach lowers the degree of decentralization: i.e., more sophisticated strategies to select peers as indexing peers need to be developed. Additional future work will address the integration of semantic technologies at application level, e.g., Semantic Web technologies. As result, a decentralized semantic wiki would enable to establish a partnership between human and automated collaborators, a vision in future industry scenarios.

References

[1] GartnerConsulting. The emergence of distributed content management and peer-to-peer content networks. *Technical Report*, Gartner Group, January 2001.

[2] Schmücker J, Müller W. Praxiserfahrungen bei der Einführung dezentraler Wissensmanagement-Lösungen. *Wirtschaftsinformatik* 2003; 3:307-311.

[3] Bartlang U, Stäber F, Müller JP. Introducing a JSR-170 standard-compliant peer-to-peer content repository to support business collaboration. Expanding the Knowledge Economy: Issues, Applications and Case Studies, Information and Communication Technologies and the Knowledge Economy, vol. 4, Cunnigham P, Cunnigham M (eds.), IIM, IOS Press, 2007; 814-821.

[4]. True picture of P2P filesharing. Technical Report, CacheLogic Research, 2004.

[5]. Ghemawat S, Gobioff H, Leung ST. The Google file system. *SIGOPS Oper. Syst. Rev*. 2003; 37(5):29-43.

[6] Gerdes C, Bartlang U, Müller JP. Decentralised and reliable service infrastructure to enable corporate cloud computing. Collaboration and the Knowledge Economy: Issues, Applications and Case Studies, Information and Communication Technologies and the Knowledge Economy, vol. 5, Cunnigham P, Cunnigham M (eds.), IIM, IOS Press, 2008; 683-690.

[7] Reiser HP, Bartlang U, Hauck FJ. A reconfigurable system architecture for consensus-based group communication. *Parallel and Distributed Computing and Systems* (PDCS 2005), Zheng SQ (ed.), ACTA Press, 2005; 680-686.

[8] Bartlang U, Müller JP. DhtFlex: A flexible approach to enable efficient atomic data management tailored for structured peer-to-peer overlays. *ICIW*, vol. 3, Mellouk A, Bi J, Ortiz G, Chiu DKW, Popescu M (eds.), IEEE Computer Society Press, 2008; 377-384.

[9] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM Press, 2001; 149-160.

[10] Wikipedia, the free encyclopedia. http://www.wikipedia.org, 2009.

[11] Urdaneta G, Pierre G, van Steen M. A decentralized wiki engine for collaborative wikipedia hosting. *Proceedings of the 3rd International Conference on Web Information Systems and Technologies*, 2007.

[12] Freedman MJ, Freudenthal E, Mazières D. Democratizing content publication with coral. *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*, USENIX Association: Berkeley, CA, USA, 2004; 18.

