Symbiotic Coupling of P2P and Cloud Systems: The Wikipedia Case

Lars Bremer and Kalman Graffi
Technology of Social Networks Group, University of Düsseldorf, Germany
Email: graffi@cs.uni-duesseldorf.de

***Abstract*—Cloud computing offers high availability, dynamic scalability, and elasticity requiring only very little administration. However, this service comes with financial costs. Peer-to-peer systems, in contrast, operate at very low costs but cannot match the quality of service of the cloud. This paper focuses on the case study of Wikipedia and presents an approach to reduce the operational costs of hosting similar websites in the cloud by using a practical peer-to-peer approach. The visitors of the site are joining a Chord overlay, which acts as first cache for article lookups. Simulation results show, that up to 72% of the article lookups in Wikipedia could be answered by other visitors instead of using the cloud.**

I. Introduction

The cloud offers elasticity, scalability, and robustness to failures. These properties address and solve the traditional shortcomings of centralized architectures. Previously, peer-to-peer (p2p) systems have been discussed in the last decade to reach same goals. Both paradigms fit to serve millions of users with desired content. However, success of p2p is found only in specific application areas. The cloud instead offers controlled quality of service, which comes with corresponding operational costs for storage and bandwidth. While p2p operates at "no" costs, one cannot match the quality of service of the cloud as all data serving relies on unreliable user devices.

In this paper we aim at combining the paradigms of the cloud and p2p systems symbiotically. The approach we present couples the p2p and cloud computing paradigms to combine the advantages of both. This system maintains most of the quality of service properties of the cloud and makes use of p2p functionality to reduce the operational costs. We make use of the fact, that users in a large-scale data-centered web-based or online applications are typically online for a while and can act as caches of other users interested in the same content. The main aspect of cost reduction is achieved by shifting the serving load from the cloud to other users in the system, users which are organized in a p2p network. Requests served by peers do not generate traffic load on the cloud, thus result in decreased operational costs for the main application or website provider. The users, on the other side, take the load to serve other nodes. This comes at only a small overhead at each user device. The overhead does not result in operational costs at the user side, as bandwidth is typically not fully utilized and often covered by Internet flatrate connections. Incentives to participate are given through less operational costs for the provider and might be implemented for users by e.g. less advertisements for actively participating users.

A. Goals of our Approach

The symbiotic combination of the p2p and cloud paradigm is modeled with the following requirements in mind.

- Application: We address the operation of websites which offer user generated content, e.g. like articles.
- Cost reduction: The costs of using the cloud should be reduced by lowering the traffic usage of the cloud. This is the main goal of our work.
- Scalability: A solution must be scalable and maintain a reasonable quality of service. The time for retrieving an article from the website should not exceed 1 sec.
- No modification of the website: The article serving website and its functionality must stay unmodified. Thus, the approach should only be an extension that brings the desired functionality of cloud cost reduction.
- Data / Article update: The articles on the website are assumed to be dynamic and frequently changed by visitors. These changes should be considered by the solution.

B. Use-Case

As a use-case for the symbiotic approach we focus on the collaborative Internet encyclopaedia Wikipedia. Wikipedia was selected as usage statistics are openly available and can be used for a proper workload model. However, the application scope is of course broader. Regarding Wikipedia, what if Wikipedia was hosted in the cloud, how much costs could be saved by using a p2p-assisted solution? For a valid evaluation model of the use-case, we investigate the current state of Wikipedia. The English Wikipedia consisted of 3.8 million articles in October 2011 according to [1]. Users edited those articles 3.4 million times in the same month. This means each article is edited about 0.9 times per month on average. However, frequently visited articles show a much higher frequency of edits. The article about the United States, for instance, was edited 84 times in October 2011. This is nearly one hundred times more than the average article. The website experiences about 11.3 million page views per hour according to [2]. The Wikimedia Foundation does not keep track of the number of visitors. For that reason third party data is used to estimate these numbers. Alexa Internet, a subsidiary company of Amazon that focuses on web traffic analysis, estimates four page views per user of Wikipedia.org [3]. Using this information, the number of unique users can be estimated. The average data transferred per page view is 19652 bytes. This number is the average data that was returned for a page view in the first days of October

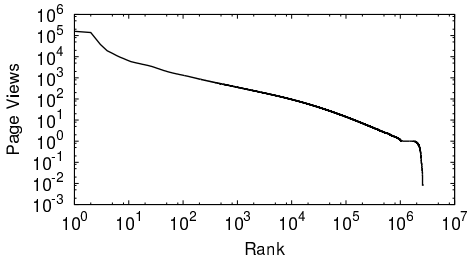


Fig. 1. Page View Distribution

2011. The mean session time is another important factor for evaluating the opportunities of a symbiotic model. Here, the mean session time is the average time a user participates in the symbiotic approach and runs the corresponding application. Since this number depends greatly on the application itself an estimation is hard. When the application needs a lot of resources and an active window most user will probably quit the application once an article was read. Instead, when the application runs in the background without much resource consumption, even in the case of a closed browser, the mean session time might be very high. Our investigation covers different mean session times of 30, 60, 90, and 120 minutes as suitable values.

One main attribute of Wikipedia is the distribution of article requests. To identify this distribution, data for page views of all articles in the English Wikipedia were evaluated using hourly records of page views from the 1st to the 5th of October 2011. Rank i describes the ith most popular article in this statistic. The resulting distribution is illustrated in Figure 1 on a logarithmic scale on both axes. The figure shows a steeply declining curve. The most visited article, for instance, is requested more than 10^5 times within one hour. The article with rank 100 is only visited a bit more than 10^3 times which is 100 times less. Articles with rank $500,000$ to $2,000,000$, on the other hand, are only visited about once each hour. This attribute of the Wikipedia page view distribution is heavily used in our symbiotic approach presented here.

II. Related Work

P2P technology comes with the economical advantage of distributing operational load to the users. Examples on how p2p resulted in cost reduction in traditional markets are presented in [4]. In recent years much research was undertaken to reduce the resource consumption of traditional client-server data access with p2p functionality.

In the field of peer-assisted video on demand, several solutions have been introduced, e.g. as in [5], [6], or [7]. The use case of video on demand is static and does not consider updates of the content. Several other papers like [8] and [9] address the idea of a symbiosis of a server infrastructure and p2p from a theoretical point of view. In this paper, however, we apply our approach to a practical use case. In [9], Kumar and Ross focus on the fundamental challenges of the symbiotic file distribution. On the other hand, in Quota [8], Liu et al. analyze the possibilities and trade-offs concerning server design in a hybrid environment. While both papers provide an interesting theoretical insight they fall short in providing a detailed protocol description. Approaches leveraging users' resources through hybrid architectures have been proposed in [10]. In these architectures, both servers and peers coexist.

Two examples that explicitly deal with a combination of the cloud and p2p are Cloudcast, introduced in [11], and the CS-P2P Stream Cloud Architecture as described in [12]. Cloudcast uses a system of a storage cloud and peers, in which the cloud is not used much different than peers. This design choice leads to a constant number of cloud accesses and consequentially only very little traffic. Cloudcast is based on the epidemic protocol introduced by Demers in [13] for information dissemination and the peer sampling protocol Cyclon presented in [14]. Using those, the model allows each peer to maintain a constant number of c references. The number of references pointing to a certain peer is expected to be c as well, since they are randomly chosen. Also only c references point to the cloud, due to the choice of same treatment of cloud and peers. This is the main reason why only a constant number of read and write operations at the cloud can be achieved using Cloudcast in order to reduce traffic costs.

The CS-P2P Stream Cloud uses a different approach of offering different types of quality of service that can be priced accordingly. The system assumes a cloud environment for streaming data. The best quality is achieved by users that receive the streamed data directly from the cloud. Those, so called first level clients, themselves provide the streaming information to higher level clients in a p2p fashion. This allows the provider to charge users that do not require any interaction with the cloud. Cloudcast and the CS-P2P Stream Cloud both reduce costs by allowing users to consume the service provided by peers. However, both assume a system where data is pushed into the network and to all participating users by the service provider. In this paper, we assume a passive role for the cloud in the Wikipedia case where an article is pulled from the network when requested by a peer.

Other architectures that fit the use-case of Wikipedia more are symbiotic solutions for online hosting like FS2You [15]. The architecture is built to reduce the costs for online storage providers, such as one-click hoster, where users can upload and publish their data. FS2You, focuses on bandwidth reduction to reduce the costs. The developed protocol is already implemented and deployed. This allowed the authors to gather real life evaluation data that show the performance and bandwidth reduction. The example of FS2You furthermore shows the user's acceptance of a hybrid architecture. The architecture consists of two types of servers. One type is used to maintain bootstrapping data and information about which peers in the network can provide a certain file. The second type of servers are replication servers that store content to prevent a fatal loss of some data objects when all peers containing that file leave the system. FS2You uses a very different protocol despite the similarities with our objectives. While FS2You uses a centralized unstructured p2p system we go one step further and propose the use of a structured p2p system that allows to lookup data objects from other peers without any cloud

interaction. In FS2You however, all requests are firstly sent to a tracking server to obtain information about the topology and location of requested data. Additionally, peers have to periodically communicate with the server to update their status and knowledge of the topology.

Work related to peer assisted cloud services, like in [16] and [17]. [16] motivate the usage of randomized p2p approaches for data backup. [17] uses a hybrid approach with tracking servers and decentralized gossip methods for constructing and maintaining the p2p overlay. This results in a dedicated server plane for the management of the peers. Our approach does not require such an extra layer. The authors of [18] motivate the replacement of such peer-supporting servers by a cloud of residential gateways. Such local gateways are located at the edge of the Internet, however, this is a drastic assumption requiring action by the ISPs. We advocate to use both central elements as well as elements at the edge of the Internet in order to fully utilize advantages at both sides without making changes to the Internet design. In [19] the use case of Dilbert's comic strips and the hourly News Update podcast from CNN is used for a combined p2p-supported cloud solution. In [20] the authors build an unstructured overlay with a focus on replica minimization with the help of a cloud. Another hybrid approach is the online data backup architecture proposed in [21]. The author focus on reducing the storage costs instead of costs for bandwidth. In the approach proposed in this paper; however, we focus on reducing the bandwidth and leave the storage requirements of the system untouched to be able to provide every single file even in times of great churn or other unintended behavior.

III. Peer-assisted Cloud: A Symbiotic Coupling

In this section, we introduce our symbiotic coupling of p2p networks with cloud-hosted applications or websites with the goal to reduce the costs of cloud vendors. First, we present the p2p model, the cloud model, and then the coupling solution. Please note, that our Wikipedia use case is just an example showing the potential of the symbiotic coupling.

A. Cloud Model

In the system model used here, we assume an article provider, i.e. the Wikimedia Foundation, acting as cloud vendor. As such, it relies on a cloud provider to offer the infrastructure that the Foundation uses to provide its own cloud service to its customers. This service, that allows browsing through Wikipedia articles, is offered as a *Software as a Service* (SaaS). On the other hand, we assume an *Infrastructure as a Service* (IaaS) provided by the cloud provider. The use of an *IaaS* provider has the advantage of not being forced to use a certain platform or software as it is the case in *PaaS* and *SaaS*. In addition to the article hosting service, we assume that the Wikipedia Foundation also runs a service to support the load reducing p2p overlay. Both services are strictly separated. Users are not required to participate in the p2p overlay; however, incentives are given to do so.

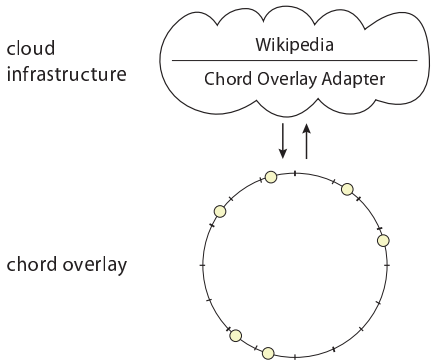


Fig. 2. Overview Cloud and Chord Overlay

B. Peer-to-Peer Model

In our system model, we consider a structured p2p overlay, e.g. as browser plugin, that allows the lookup of uniquely identifiable content. Chord, which was introduced in [22], is a prominent example for a key-based routing [23] overlay. Through consistent routing, which for example is not given in KAD (see [24]), one can quickly retrieve and also consistently update information in the network. In Chord, an identifier is assigned to each node and data element (e.g. articles in our case). The identifier of such an article is the hash value of the article name. Structured overlays organize the nodes within to maintain a clear indexing structure of object identifiers they are responsible for. Chord is used to index and look up Wikipedia articles in the p2p overlay with lookup and state complexity of $O(\log n)$, thus, satisfying the scalability requirement.

The previously mentioned service run in the cloud to support the p2p overlay is used as bootstrapping point for the nodes to enter the p2p overlay as well as an update instance for the peers that store article copies from the cloud. Figure 2 illustrates this connection.

C. Symbiotic Coupling Model

The main goal of the symbiotic coupling is to lower the bandwidth usage of the cloud hosted service to reduce the operational cost of the cloud vendor. For that reason, we assume that a client accessing the cloud to download an article, joins a p2p overlay simultaneously, Chord in this case. Clients offer downloaded articles to further clients, they act as a cache. New clients looking for articles first join the p2p overlay and start a lookup in the p2p overlay before querying the cloud. The cloud is only used to download articles from in case the initial lookup in the overlay was unsuccessful. Please note, we assume that clients are benign, we do not address security issues in this paper.

In the p2p overlay, a node responsible for an article stores references to nodes which previously queried for the specific article and downloaded it successfully. When a node n joins the overlay network it initially does not reference any article. In case another node s looks up a document that is assigned to the recently joined node, n will answer the request indicating that s has to obtain the article from the cloud because n has no information of any occurrences of the article in the overlay

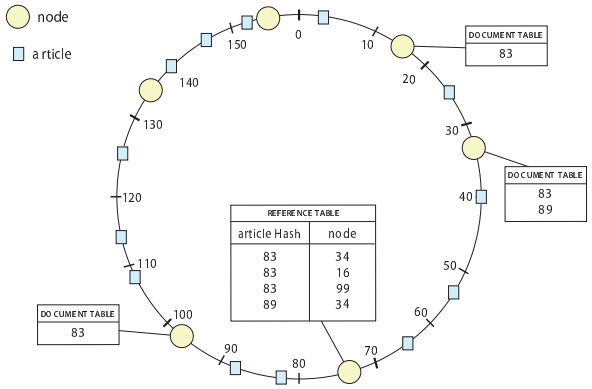


Fig. 3. P2P Overlay storing References to cached Articles from the Cloud

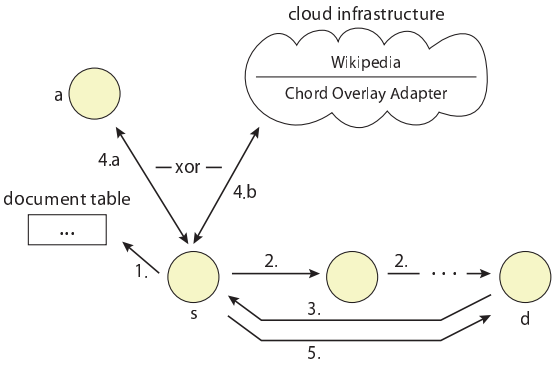


Fig. 4. Article Lookup when Article is in the Overlay

network. Additionally, n will add s to a list of references to nodes containing that article. If a third node requests the same document afterwards, n will reply with a message containing the reference to s. This is where the requester can download the article from without the need for any cloud interaction. To make use of the distribution of article requests, multiple reference per document can be maintained by one node. This results in a situation where nodes responsible for very frequently requested articles, have many references to other nodes containing that popular article. The consequence is, that requests to download an article can be evenly spread among those nodes that store the article. This caching can be done load- and capacity-aware.

Figure 3 illustrates an example of the p2p overlay in the symbiotic coupling. The figure shows the state after nodes 16, 34, and 99 requested articles from node 73. The requested articles are stored by the requesting nodes while the node responsible for the articles only maintains a table of references.

In case of article updates in the cloud one has to make sure that the p2p overlay does not spread outdated articles. For that, the second function of the cloud service is used: updating the status in the overlay. The cloud sends a message indicating the update to the node maintaining the references to the edited article. This node discards then the references it holds for the specific article. Now the node starts all over again and the updated version of the article is downloaded from the cloud in case it is requested.

D. Protocols for Symbiotic Coupling

Next, we describe the four most important functions of the symbiotic coupling model in more detail.

1) Join Operation: When a node joins the overlay network, it initially has no articles stored locally. First, the joining node contacts the cloud with a joining request to obtain bootstrapping information. The cloud answers this request with the IP address of a few peers that already joined the overlay network. When the joining node successfully received these addresses it contacts a node in the overlay network and triggers the Chord joining process.

2) Article Lookup: Once a node joined the overlay it queries articles first in the overlay, as depicted in Figure 4.

1) Assuming a peer s wants to download a certain article. First s checks its own document table for the occurrence of the article. If the node already contains it, no further actions are taken. In the more likely case that the node does not already contain the document, step 2 is executed.
2) In the second step, the routing table of the node is used to forward the request to the node d responsible for the object id, identified by the hash of the article's name.
3) In step 3, the request reaches the responsible node d in the overlay. This node either contains references to the requested article (step 4a) or it does not (step 4b). In the former case, d sends the first reference from the queue to node s. Then, the reference is placed to the last position of the queue. In contrast, when the queue is empty, s is notified that no reference was found.
4) In step 4 the query result is sent by the node being responsible for the article to the requesting node.

4a) In case s received a reference to a node a containing the requested article, s contacts this node. Then a answers with the requested article.
When s received reference to an offline node, d is informed of that issue. It then removes the reference of node a from the queue and starts again with step 3.

4b) When s receives a message indicating that the article is not in the overlay network, the article is obtained from the cloud instead. Since the cloud offers a reliable service, this lookup is assumed to be always successful.

5) When the download finished successfully, s informs d that it now has the article stored and d adds s to the list of references of that article.

3) Update Operation: Articles are edited directly in the cloud, just like the data is edited on the servers of the Wikimedia Foundation today. The p2p network does not support article editing itself. Thus, the cloud always holds the newest version of an article and can actively inform other nodes about edits. When an article a is edited, the cloud service therefore actively informs the responsible node for a in the overlay. Subsequently, the informed node empties the queue of references for that article. Now, a new request for a will be answered such that the requesting node obtains the article from the cloud. The responsible node for the references of

	Original Data	x0.001	x0.0005	x0.00025
Article Count	3,800,000	3,800	1,900	950
Page Views / Hour	11,300,000	11,300	5,650	2825
Reference Lookup		0.67s	0.97s	0.51s

TABLE I
DOWNSCALED DATA MODEL AND REFERENCE LOOKUP TIME RESULTS

Session Time	30 min	60 min	90 min	120 min
Setup: Scale Factor	x0.0005	x0.0005	x0.0005	x0.0005
Setup: Article Count	1,900	1,900	1,900	1,900
Setup: Peer Count	706	1,413	2,119	2,825
Downloaded from Peers	51.8%	61.0%	67.8%	72.4%

TABLE II
SIMULATION SERIES WITH SCALE FACTOR X0.0005

a builds up a new queue containing only nodes storing the newest version of the article.

4) Leave Operation: Each node n maintains references to nodes storing articles n is responsible for. Additionally n stores contact information of other nodes that hold references to n. Whenever a node leaves it uses this list of contacts to inform the other nodes to remove all references pointing to it. This keeps the list of references updated.

Having presented the protocols for the symbiotic coupling of p2p and cloud for a cost-reduced serving of Wikipedia articles, next we present the evaluation of the approach.

IV. EVALUATION

In the evaluation, we aim at measuring the ratio of queries served by the p2p overlay and the corresponding lookup performance. These metrics are influenced by the mean session time of the nodes as well as the size of the network. To obtain the results, we implemented and simulated the approach using the event-based p2p system simulator PeerfactSim.KOM, which is presented in [25], [26]. We used the Chord implementation as in [27], a bandwidth model for the globally distributed peers according the OECD Bandwidth Report from 2007 [28] and a delay model determined by GNP [29].

A. Simulation Setup

The simulation was executed with a downscaled model of the English Wikipedia with the factors x0.001, x0.005 and x0.00025. The number of articles and page views were scaled down whereas the number of page views per user and the average page size are kept the same since they are independent of the size of Wikipedia, see for that the description of the use case in Section I-B. The average page size is 19652 Bytes and the page views per user is 4 in all scenarios.

Two series of simulations were executed. The first series used a fixed scaling factor of 0.0005 times of the original size of Wikipedia. The effects of the scaling are depicted in Table I. Four setups were used with different node mean session times. We depict the simulation setups in Table II. The main goal of these simulations was to evaluate the performance and traffic costs of the symbiotic model and to get an idea of how the mean session time affects both. The second series of simulations uses different scaling factors to monitor the performance difference in large scaled systems while keeping the session time at a fixed value of 60 minutes.

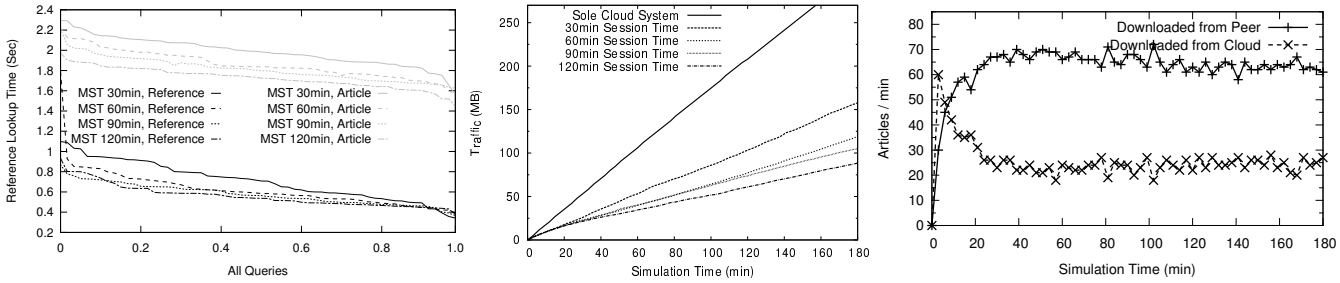
B. Evaluation Results

First, we present the results with regard to the lookup time, to show the applicability of our approach. Then, we present the costs saved in the cloud through our approach. Figure 5(a) shows the results of these simulations with a scaling factor of 0.0005 times of the original size of Wikipedia and different mean session times. The figure shows the time it takes to receive the reference to a node containing an article. The reference is looked up within 1 sec in nearly all cases, afterwards the 19kb file is quickly downloaded. Thus, the symbiotic approach is considered scalable, as the number of nodes (i.e. scale factor) does not significantly influence the overall lookup performance. The average reference lookup times for the other scaling factors do also not exceed 1 sec, they are depicted in Table I. A tolerable waiting time for an Internet user, however, is about 2 seconds according to [30], thus the performance is adequate.

Knowing that the lookup time is acceptably small, we investigate the traffic savings of our approach. For that we compare the number of article lookups from the cloud with the number of articles obtained from the p2p overlay. The largest effect is observable with a mean session time of 120 min in Figure 5(c). 72.4% of the articles are retrieved from the overlay. The figure shows a vast increasing number of articles downloaded from the cloud at the beginning of the simulation. This is an expected behaviour since the simulation starts without any articles being stored at any peer. However, about 10 minutes after the start of the simulation, the number of articles obtained from the overlay surpasses the number of articles downloaded from the cloud. This trend continues until a balanced state is reached. From this moment onwards, the rate of articles downloaded from the cloud and from the overlay shows no trend any more and maintains its level.

An overview for the savings corresponding to the mean sessions times 30, 60, 90, and 120 minutes is depicted in Table II. With mean session times between 30 min and 90 min, nodes go more frequently offline and drop their articles and reference list. This leads to more downloads from the cloud as valid article sources in the overlay are less frequently available. Consequently, in simulations with a lower session time articles are more often downloaded from the cloud than in an environment with a higher session time. The results obtained form the simulation show that after 90 minutes, 72.4% of the downloaded article are obtained from other peers when the session time is set to 120 minutes. In contrast, when the mean session time is 30 minutes, only 51.8% of the articles were downloaded from the overlay, as shown in Table II.

To compare the absolute traffic created at the cloud, each file transfer was summed up during the simulations. Figure 5(b) shows the development of this summed traffic. With scale factor x0.0005 and a mean session time of 120 min the sole cloud solution needs to serve 150 MB per hour. In the symbiotic approach the cloud serves only 55.25 MB per hour, thus nearly only one third. This shows that the costs savings are practically significant and thus relevant.



(a) Reference Lookup and Total Download Time (b) Traffic Load in the Cloud (c) Traffic Load Savings for Session Time 120 min

Fig. 5. Evaluation Results for the 4 Scenarios: Reference Lookup Time and Traffic Load Savings

V. Conclusion

Current cloud solutions provide desired storage and bandwidth capacity for fixed billing plans. In this paper we present a symbiotic approach, combining p2p and cloud technology. The presented solution offers the quality of service assured by the cloud paired with the low cost of a p2p system. This approach was implemented and simulated with a downscaled version of Wikipedia used as case study. The main idea of the approach is to reduce the traffic costs by allowing peers to obtain Wikipedia articles from other peers. Whenever a requested article is not found in the overlay network, it is obtained from the cloud. This makes each Wikipedia article accessible at any time due to the availability guarantees of the cloud. Evaluation indicates an average reference lookup time of below 1 second. While the mean session time of the nodes does not have impact on the download speed, it has a big impact on the ratio of articles downloaded from the cloud and those downloaded from other peers. With a mean session time of 120 minutes, for instance, 72% of the documents are downloaded from the p2p overlay and the traffic costs at the cloud are saved. The symbiotic approach can reduce the traffic needed for article lookups in case of Wikipedia up to 72%. This demonstrates how well the cloud and p2p systems can complement each other. As future work, we will focus on the elaboration of further use cases and an extension to delay tolerant networks for the case of intermitted connectivity.

References

[1] E. Zachte, "Wikipedia Statistics," http://stats.wikimedia.org/EN/TablesWikipediaEN.htm, March 2012.

[2] E. Zachte, "Page Views for Wikipedia, All Platforms, Raw data," http://stats.wikimedia.org/EN/TablesPageViewsMonthlyOriginalCombined.htm, March 2011.

[3] Alexa Internet, Inc, "Wikipedia.org Site Info," http://www.alexa.com/siteinfo/wikipedia.org, February 2011.

[4] N. Liebau, K. Pussep, K. Graffi, S. Kaune, E. Jahn, A. Beyer, and R. Steinmetz, "The Impact of The P2P Paradigm," in *Proc. of AMCIS '07*, 2007.

[5] K. R. C. Huang, J. Li, "Peer-Assisted VoD: Making Internet Video Distribution Cheap," in *Proc. of IPTPS '07*, 2007.

[6] H. S. Vaishnav Janardhan, "Peer assisted VoD for set-top box based IP network," in *Proc. of P2P-TV '07*, 2007.

[7] K. Graffi, S. Kaune, K. Pussep, A. Kovacevic, and R. Steinmetz, "Load Balancing for Multimedia Streaming in Heterogeneous Peer-to-Peer Systems," in *Proc. of ACM NOSSDAV '08*, 2008.

[8] F. Liu, Y. Sun, B. Li, and B. Li, "Quota: Rationing server resources in peer-assisted online hosting systems," in *Proc. of ICNP '09*, 2009.

[9] R. Kumar and K. W. Ross, "Optimal peer-assisted file distribution: Single and multi-class problems," in *Proc. of IEEE HOTWEBS '06*, 2006.

[10] A. Gharaibeh and M. Ripeanu, "Exploring Data Reliability Tradeoffs in Replicated Storage Systems," in *Proc. of HPDC '09*, 2009.

[11] A. Montresor and L. Abeni, "Cloudy Weather for P2P, with a Chance of Gossip," in *Proc. of IEEE P2P '11*, 2011.

[12] I. Trajkovska, J. S. Rodriguez, and A. M. Velasco, "A Novel P2P and Cloud Computing Hybrid Architecture for Multimedia Streaming with QoS Cost Functions," in *Proc. of ACM Multimedia '10*, 2010.

[13] A. J. Demers, D. H. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. E. Sturgis, D. C. Swinehart, and D. B. Terry, "Epidemic Algorithms for Replicated Database Maintenance," *Operating Systems Review*, vol. 22, no. 1, 1988.

[14] S. Voulgaris, D. Gavidia, and M. van Steen, "CYCLON: Inexpensive Membership Management for Unstructured P2P Overlays," *Journal on Network and System Management*, vol. 13, no. 2, 2005.

[15] F. Liu, Y. Sun, B. Li, B. Li, and X. Zhang, "Fs2you: Peer-assisted semipersistent online hosting at a large scale," *IEEE Transactions on Parallel Distributed Systems*, vol. 21, no. 10, pp. 1442–1457, 2010.

[16] L. Toka, M. Dell'Amico, and P. Michiardi, "Online Data Backup: A Peer-Assisted Approach," in *Proc. of IEEE P2P '10*, 2010.

[17] F. Liu, Y. Sun, B. Li, B. Li, and X. Zhang, "FS2You: Peer-Assisted Semipersistent Online Hosting at a Large Scale," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 10, 2010.

[18] A. Kermarrec, E. Le Merrer, N. Le Scouarnec, G. Straub, and A. Van Kempen, "Efficient Peer-to-Peer Backup Services through Buffering at the Edge," in *Proc. of IEEE P2P '11*, 2011.

[19] A. Montresor and L. Abeni, "Cloudy Weather for P2P, with a Chance of Gossip," in *Proc. of IEEE P2P '11*, 2011.

[20] A. M. H. Kavalionak, "P2P and Cloud: A Marriage of Convenience for Replica Management," in *Proc. of IEEE IWSOS '12*, 2012.

[21] L. Toka, M. Dell'Amico, and P. Michiardi, "Online data backup: A peer-assisted approach," in *Proc. of IEEE P2P '10*, 2010.

[22] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proc. of SIGCOMM '01*, 2001.

[23] F. Dabek, B. Y. Zhao, P. Druschel, J. Kubiatowicz, and I. Stoica, "Towards a Common API for Structured Peer-to-Peer Overlays," in *Proc. of IPTPS '03*, 2003.

[24] H. J. Kang, E. Chan-Tin, N. Hopper, and Y. Kim, "Why Kad Lookup Fails," in *Proc. of IEEE P2P '09*, 2009.

[25] K. Graffi, "PeerfactSim.KOM: A P2P System Simulator - Experiences and Lessons Learned," in *Proc. of IEEE P2P '11*, 2011.

[26] D. Stingl, C. Groß, J. Rückert, L. Nobach, A. Kovacevic, and R. Steinmetz, "PeerfactSim.KOM: A Simulation Framework for Peer-to-Peer Systems," in *Proc of HPCS '11*, 2011.

[27] A. Kovacevic, K. Graffi, S. Kaune, C. Leng, and R. Steinmetz, "Towards Benchmarking of Structured Peer-to-Peer Overlays for Network Virtual Environments," in *Proc. of IEEE ICPADS '08*, 2008.

[28] OECD, "OECD Bandwidth Report," 2007. [Online]. Available: http://www.oecd.org/sti/ict/broadband

[29] T. Ng, S. Eugene, and H. Zhang, "Global Network Positioning: A New Approach to Network Distance Prediction," *ACM SIGCOMM Computer Communications Review*, vol. 32, no. 1, pp. 61–61, 2002.

[30] F. F.-H. Nah, "A Study on Tolerable Waiting Time: How Long Are Web Users Willing to Wait?" in *Proc. of AMCIS '03*, 2003.