University of Würzburg
Institute of Computer Science
Research Report Series

Performance Evaluation of a Reliable Content Mediation Platform in the Emerging Future Internet

Simon Oechsner and Phuoc Tran-Gia

Report No. 408 April 2007

Department of Distributed Systems, Institute of Computer Science
University of Würzburg, Am Hubland, 97074 Würzburg, Germany
{oechsner|trangia}@informatik.uni-wuerzburg.de

Performance Evaluation of a Reliable Content Mediation Platform in the Emerging Future Internet

Simon Oechsner and Phuoc Tran-Gia
Department of Distributed Systems, Institute of Computer Science
University of Würzburg, Am Hubland, 97074 Würzburg, Germany
{oechsner|trangia}@informatik.uni-wuerzburg.de

Abstract

In todays Internet, a trend towards distributed content delivery systems can be observed. These systems still have to offer the same functionality as centralized architectures, e.g. content localization. While several advantages like load distribution and cost reduction could be gained by using a decentralized distribution platform, they come with a tradeoff in resource consumption. In this paper, we analyze and evaluate a decentralized Content Distribution Application (CDA) with respect to the time needed to locate specific content. We introduce a node model based on queueing theory and provide methods to compute important characteristics like the mean search time. In this model, we do not only consider the external offered load, but also the internal traffic created as a consequence of decentralization.

1 Introduction

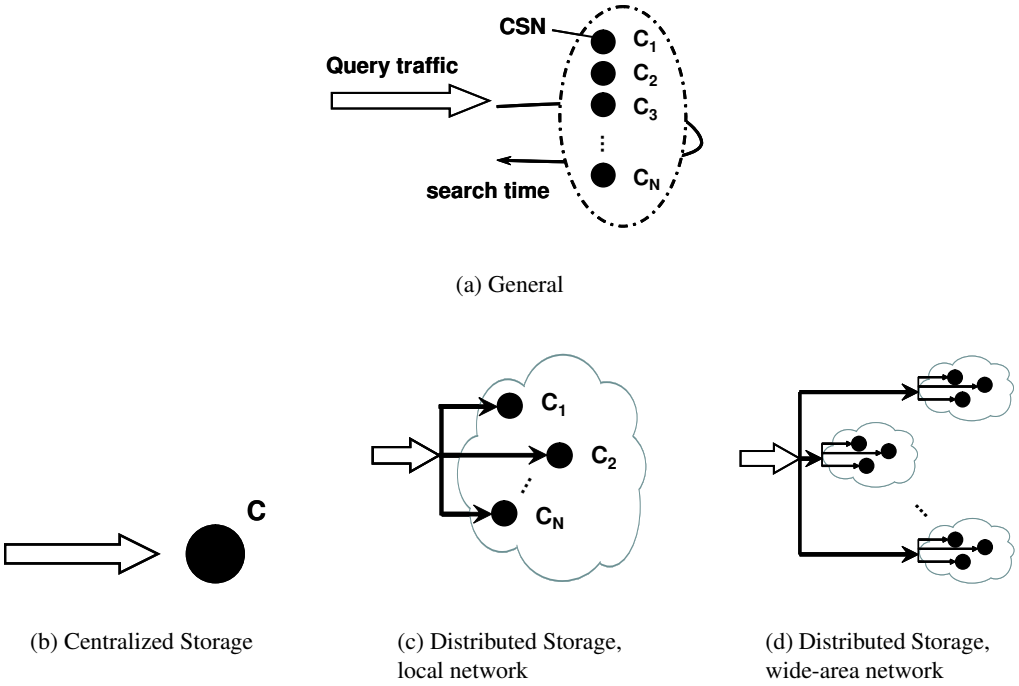
Content application and content providing applications are significant emerging services in the evolution towards the next-generation Internet. Two trends can be observed: the decentralization of the application in content providing and the emerging field of scientific computing. The reasons for these trends are manifold. It is estimated that the volume of information doubles every year, i.e. one has to deal with a factor of one thousand in ten years. The complexity of the geographical structure of the content in such applications is increasing while the dynamic of the information provided is more stochastic. Further on, security issues have to be considered to provide integrity of the content, which adds another dimension to the operation of future content providing applications.

In several current and future content delivery applications, a trend of decentralization can be observed. Large content (genetic research data, customer-related data of mobile network providers, etc), which used to be stored in one single location with standby facilities, is to be divided structurally and stored in highly distributed architectures. The design aims of such structures are to i) balance the query load, ii) minimize the search delay and incurred network traffic and iii) provide higher reliability and resilience, among others. These structures should support application-layer routing in future content providing networks.

This work was funded by Siemens AG, Berlin. The authors alone are responsible for the content of the paper.

The aim of this paper is to provide a performance discussion of content distribution platforms regarding search delay in a generic context. For this purpose we take into account a content providing and distribution platform (CDA: Content Distribution Application) with a Content C, which is segmented in a number of N Content Segments (CS) $C_1, C_2, \ldots, C_N$. These CS are hosted on Content Segment Nodes (CSN), where each node holds one or more CS (see Figure 1(a)). These nodes also serve as the interface to users of the CDA, accepting queries and locating content. The actual specific content, i.e., data documents, are part of the CS and in general assumed to be of much smaller size than a complete segment.

Therefore, the main functions of a CSN are: i) provide access to the content application platforms, ii) routing and lookup of requested Content Segments and iii) provide local data contained in the locally stored Content Segment to attached clients.



(a) General

(b) Centralized Storage

(c) Distributed Storage, local network

(d) Distributed Storage, wide-area network

Figure 1: System architecture

Figures 1(b) to 1(d) show different realizations of such an architecture. One alternative is the storage of the complete content on one central server (or several redundant ones), shown in Figure 1(b). In contrast to this, the architectures shown in Figures 1(c) and 1(d) distribute the Content Segments to different nodes. The difference between those two solutions lies in the geographical distribution of the CSNs. In this paper, we concentrate on the analysis of the architecture shown in Figure 1(c), i.e., a system where all nodes are connected to the same local network. Details about the considered architecture will be given in Section 3.

One reason why we consider this variant is the assumption that the described CDA is to be employed in a performance-sensitive, i.e., corporate environment. This also means that the performance of the system is of great importance while the system is dimensioned for a pre-defined grade of service. This is a big difference to 'best-effort' applications, where the partaking

nodes usually use only a fraction of their resources (especially processing power), and where virtually no hard timing constraints exist. These efficiency considerations make it interesting to conduct a performance evaluation. The remainder of the paper is organized as follows. In Section 4 we will show analytical results as well as a numerical example. We conclude this work in Section 5.

2 Related Work

In the past, several works have been published that consider efficient overlays for content storage and content distribution. Gupta et al. [1, 2] evaluate a fully meshed overlay architecture, which is also the basis for the architecture used in this paper. They show that the additional overhead needed to keep the complete routing table in each node up-to-date can be handled even for large overlays.

In [3], a different scheme based on tokens for maintaining the global routing table in a one-hop overlay was presented and compared to the mechanism of Gupta et al.. The authors showed that less bandwidth was used in their architecture, resulting in a more efficient system.

Another architecture and its analysis of a one-hop DHT was presented in [4]. Again, it was shown that this kind of system is indeed feasible in terms of message overhead and response times to keep the complete routing tables needed at each peer up-to-date.

However, while [1, 2, 3, 4] all analyze the bandwidth consumption at the nodes, they do not consider the processing load generated by the queries on the nodes themselves. Since we assume that the system considered here is positioned in a corporate environment, we conclude that the nodes of the system are not underutilized, but are dimensioned for the load offered by the database users. Therefore, we take a closer look at the conditions of the nodes themselves in this work.

In [5], an example of an application of the described CDA architecture is given in the form of a small-scale publish/subscribe system. It is also emphasized that the availability and usage of resources in a small-scale, highly utilized network is of critical importance for the performance of the system.

3 Architecture

The architecture considered in this paper consists of a number of CSNs that are physically fully meshed by a Local Area Network (LAN). When a CS is requested by a client, one of these lookup nodes is contacted. The choice of this initial node is arbitrary and for load distribution reasons assumed to be random. In reality, this could be accomplished by, e.g., a round robin load distributor.

Since we distributed the data to the CSNs, it is probable that the needed information is not stored on this node. Therefore, the node first queried has to identify the correct storage location of the data and request it from that node. Once this is done, either the first or the second node can forward it to the client that originated the query. However, in this work we assume the search process to be independent from the actual content delivery and therefore consider the search finished once the query reaches the node holding the requested content.

To ensure that the content segment is located quickly, we assume an addressing scheme for the nodes that is based on structured P2P algorithms, specifically Chord [6]. The basic principle is as follows: Each node gets an ID from a one-dimensional identifier space, typically of size $[0; 2^m - 1]$. The *<key,value>* pairs of data are also placed in that space, with the hashed key as an identifier. Therefore, the hash function that transforms the original information of the data into the lookup key also has a codomain of $[0; 2^m - 1]$. The data is now stored on the node which has the next higher ID than the key of the data set in a clockwise direction. This rule also determines and defines the Content Segments. Content belongs to the same segment if it is hashed to the same interval between two nodes in the identifier space.

When a piece of content in a CS is searched for, the node with the next higher ID to the hash value of the content searched for is polled for that information. In a traditional structured P2P overlay, this would be accomplished by an internal routing operation that forwards the message closer to the target node with each hop. However, since the number of nodes used in this architecture is small for such an overlay (in the range of 10^2 nodes instead of the typically assumed 10^5 or 10^6), we can hold the addresses of all nodes in the 'routing table' of every node and therefore reach the target of the internal lookup in one hop. This makes the lookup efficient in terms of hops. Of course, if a client had knowledge about the addressing scheme it could query the according node directly, but we assume that the CDA is transparent to a client in the sense that the internal structure is not known outside of the system.

We assume that the nodes are distributed uniformly over the identifier space, so that the CS each node has to store has approximately the same size. Furthermore, we assume that each CS is as popular as the others, resulting in an equal number of queries for each segment. In case this assumption should be invalid, i.e., if there are CS queried very frequently in comparison to others, these CS could be copied to all CSNs in order to cope with the high demand. However, we do not consider this case here.

In this paper, we will not evaluate the described architecture during node failures. However, since it has to be considered that one or several nodes of the network can fail, some kind of redundancy mechanism should be implemented. This can happen either in the form of partial redundancy like known from P2P systems, where nodes replicate their data to neighboring peers, or via full redundancy (i.e., the whole ring system is copied), resulting in several networks.

4 Performance Evaluation and Comparison

4.1 Overall System Model

To evaluate the presented architecture, we employ the system model shown in Figure 2. The total initial load offered to the system is created by the clients connected to the CSNs. This query arrival process is assumed to be Poisson with rate λ_0. We further assume an equal distribution of this load on each node in the system, resulting from the client model described in the last section. Each node is modelled by means of a $M/GI/1$ waiting system with an arrival rate λ_{node} and a processing time B with mean $E[B]$. Thus, the normalized offered load per node is $\rho = \frac{\lambda_0 E[B]}{N}$, which is the utilization of one node caused by initial queries. The waiting time W is implicitly given by the node model and parameters. Query forwarding duration over the network is modeled by the transmission time distribution T_T.

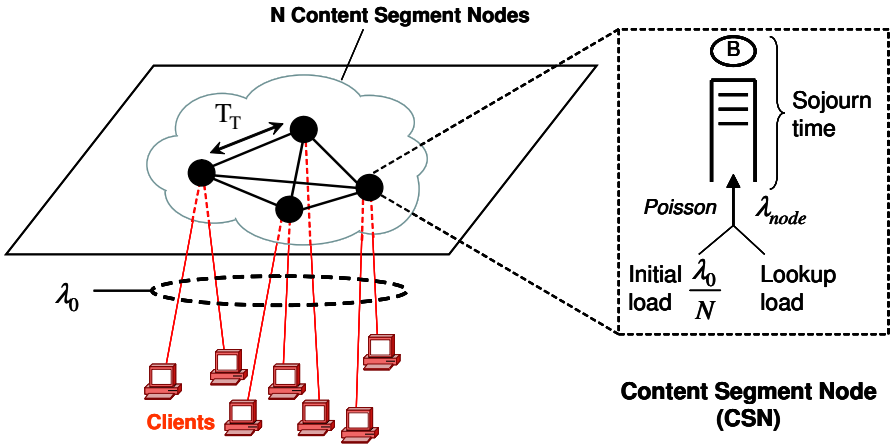


Figure 2: System model

The search procedure follows the phase diagram depicted in Figure 3. The first node is traversed in any case. With a probability $1 - p$, this node holds the queried data and the search is finished (the upper path of the diagram). With probability p however, the query has to be forwarded to the correct node, meaning one hop and an additional node traversal (corresponding to the lower path in the diagram). Since we do not consider content replication, the sum of all data is partitioned equally between the nodes. For a number N of nodes, therefore the probability to find the queried data in the first node is $\frac{1}{N}$ and the probability for an additional hop is $p = \frac{N-1}{N}$.

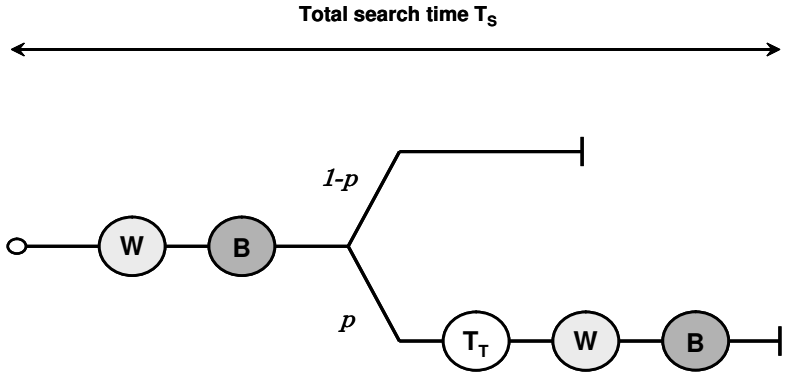


Figure 3: Phase diagram for the search procedure

An important aspect of the architecture under consideration is that not each query is answered by the first node it encounters, but may spawn a subsequent internal query. Therefore, the query arrival rate at one node is not only the total external rate on the system λ_0 divided by the number of nodes N, but additionally the re-routed queries from all other nodes. Since each node forwards queries with a rate of $p \cdot \frac{\lambda_0}{N}$, it receives the same amount for symmetry reasons and under our assumption that an equal share of the queries is forwarded to each node except the local node. We assume that the resulting total query flow arriving at one node is still Markovian with an adapted rate. We will later show simulation results that suggest that this assumption is valid.

Figure 4 shows the resulting traffic flows for one node. While this has no great impact on

systems like P2P content storage or distribution platforms, where node load is assumed to be low, it may have a significant influence on systems which operate with a high utilization. Since we assume that our architecture falls in the latter category, we have to evaluate the nodes using the total arrival rate per node λ_{node} instead of $\frac{\lambda_0}{N}$.

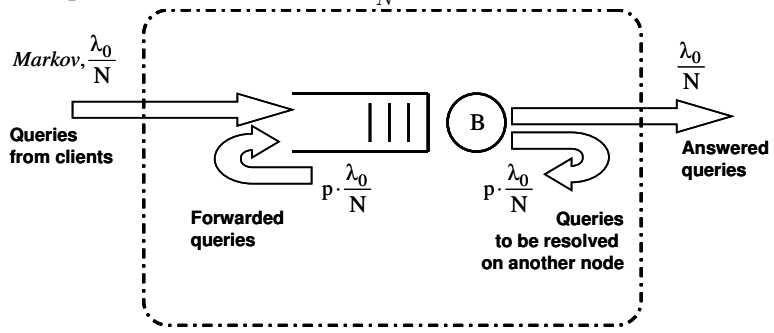


Figure 4: Model for the query traffic at a node

The normalized offered initial load of one node is $\rho = \frac{\lambda_0 E[B]}{N}$. With the given system model, the total load on one node then is this inital load plus the additional re-routed requests:

$$\lambda_{node} = \frac{\lambda_0}{N}(1+p) = \frac{\lambda_0}{N}(1+\frac{N-1}{N}). \tag{1}$$

With a service time B, the final node utilization ρ^* can be derived as

$$\rho^* = \lambda_{node} \cdot E[B] = \rho \cdot (1+p). \tag{2}$$

4.2 Analytical Approaches

In this section, we start to describe how different parameters affect the system performance. We focus on the search time, i.e., the time needed to locate a specific content segment, in the architecture described in Section 3. An approximate analysis of the search time will be presented, followed by numerical examples.

First, we will analyze the distribution function T_S of the total search time. Using an independent assumption of waiting times at the first and second nodes we arrive at the Laplace-Transform of the search time T_S:

$$\Phi_S(s) = (1-p) \cdot \Phi_W(s)\Phi_B(s) + p \cdot (\Phi_W(s)^2\Phi_B(s)^2\Phi_T(s)). \tag{3}$$

As a CSN is modelled with a $M/GI/1$ system, we can use the Pollaczek-Khintchine formula [7]

$$\Phi_W(s) = \frac{s(1-\rho)}{s - \lambda_{node} + \lambda_{node}\Phi_B(s)} \tag{4}$$

and finally obtain

$$\Phi_S(s) = (1-p)\frac{s(1-\rho)\Phi_B(s)}{s - \lambda_{node} + \lambda_{node}\Phi_B(s)} + p\left(\frac{s(1-\rho)\Phi_B(s)}{s - \lambda_{node} + \lambda_{node}\Phi_B(s)}\right)^2 \Phi_T(s). \tag{5}$$

In general it is numerically difficult to quickly obtain values in the time domain in order to gain basic insights into the system behaviour. Thus, we will first compute directly the mean value and variance of the search time.

From the phase diagram shown in Figure 3, the mean search time can be given as

$$E[T_S] = E[W] + E[B] + p(E[T_T] + E[W] + E[B]). \tag{6}$$

Again, under the assumption of a $M/GI/1$ system, we can express the mean waiting time in the queue of a node according to Takács [8] as

$$E[W] = \frac{\lambda_{node} E[B^2]}{2(1-\rho)},$$

which leads us to

$$E[T_S] = \frac{2N-1}{N}\left(\frac{\lambda_{node} E[B^2]}{2(1-\rho)} + E[B]\right) + \frac{N-1}{N} E[T_T]. \tag{7}$$

Note that we need the second moment of the service time distribution in (7) in order to compute the mean search time. The variance of the search time can also be derived from Figure 3 as

$$VAR[T_S] = VAR[W] + VAR[B] + p(VAR[T_T] + VAR[W] + VAR[B]), \tag{8}$$

where $VAR[W]$, $VAR[B]$ and $VAR[T_T]$ are the variances of the waiting time, service time and transmission time distribution, respectively. The second moment of the waiting time is

$$E[W^2] = 2E[W]^2 + \frac{\lambda_{node} E[B^3]}{3(1-\rho)}, \tag{9}$$

yielding

$$VAR[W] = \frac{3\lambda_{node}^2 E[B^2]^2 + 4(1-\rho)\lambda_{node} E[B^3]}{12(1-\rho)^2}. \tag{10}$$

Combining equations (8) and (10), we finally obtain the variance of the search time as

$$\begin{aligned} VAR[T_S] &= (1+p)\left(\frac{3\lambda_{node}^2 E[B^2]^2 + 4(1-\rho)\lambda_{node} E[B^3]}{12(1-\rho)^2} + E[B^2] - E[B]^2\right) \\ &\quad + p(E[T_T^2] - E[T_T]^2). \end{aligned} \tag{11}$$

This allows us to compute the variance of the search time from the first moments of the service time distribution and the transmission time distribution.

4.3 Parameter Studies

In this section, we will show some results derived from the formulas (7) and (11). First, we take a look at the effect of the variance of the service time of the nodes on the mean search time. Figure 5 shows the mean search times normalized by $E[B]$ for different utilizations and

coefficients of variation c_B of the service time. The parameter c_B gives us the second moment of the service time and also allows us to derive the third moment as described e.g., in [9]. We chose coefficients of variation that describe different distributions, ranging from deterministic ($c_B = 0$) to strongly varying processes ($c_B = 2$). We considered a smaller ($N = 5$) as well as a larger system ($N = 20$).

For some parameter settings, simulations have been conducted which do not assume that the combined arrival process at the nodes is Markovian, just the arrival rate of the initial queries is modeled with an exponential distribution. These results are given with confidence intervals for a 95% confidence level over 5 different runs. They suggest that our assumption about the node arrival process are valid and that the approximation is very accurate.

It should be noted that due to Equation (2), the node utilization in this model can be up to twice the utilization by external queries alone. Therefore, we will only cover the range $[0, 0.5]$ for ρ in the following analysis.

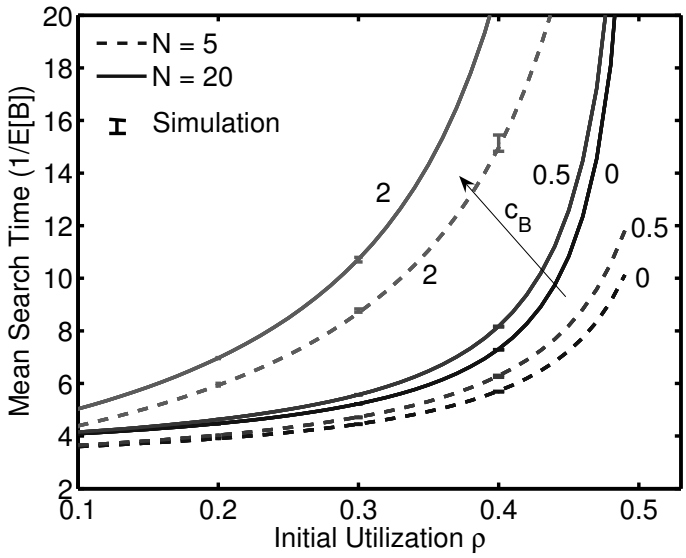


Figure 5: Effect of the coefficient of variation of the service time on the mean search times

The results show an increase in the mean search time for service processes with a higher variance for both system sizes. This is due to the longer waiting times created by the service time distribution. It can also be observed that the mean search times are lower for smaller systems. The reason for this is twofold. First, a lower number of nodes also leads to a smaller probability for a second hop, shortening the search time. Second, and also as a consequence of a lower probability p, a smaller system incurs less additional internal load and therefore shorter waiting times.

Figure 6 shows the coefficients of variation from the same experiments. A higher variance in the service time also leads to a higher variance in the total search time, as expected.

Another observation is that the coefficient of variation of the search time grows for higher utilizations and a low variance of the service time. This can be explained by the fact that ρ^* is close to 1 for these values of ρ and therefore the waiting times have a larger influence on the

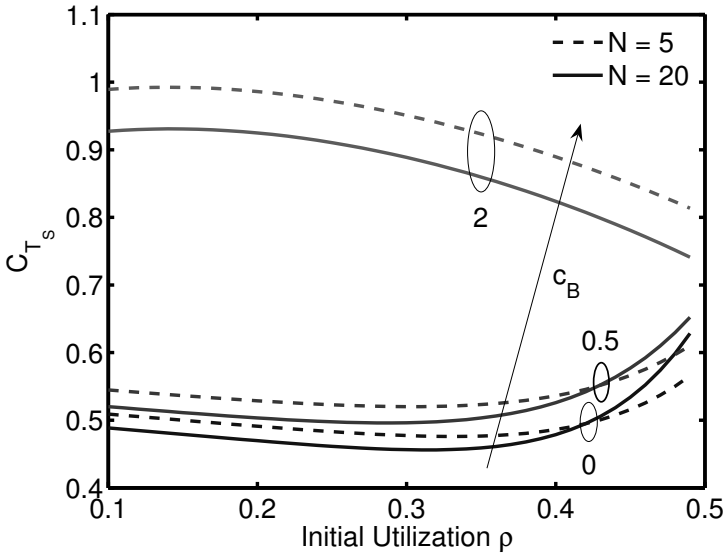


Figure 6: Effect of the coefficient of variation of the service time on the variance of the search times

total system behaviour. The same can be observed for a value of $c_B = 2$, where the effect on the coefficient of variation is reversed.

To see how the system size influences the search time, we take a look at the mean search time for different numbers of nodes. Figure 7 again shows the medium search times depending on the utilization of the system and for systems with 10, 20 and 30 nodes. We additionally compare these values for coefficients of the service time $c_B \in \{0, 1\}$. Again, we have validated these results with simulations for chosen parameter settings.

We can again observe an increase in the mean search time for larger systems due to the higher probability for an additional hop in systems with more nodes, which in turn leads to a larger fraction of searches that have to visit two nodes to find their results. Additionally, the higher internal traffic again leads to longer waiting times. Also, the results show longer search times for a higher coefficient of variation of the service time, as was to be expected from the results discussed above.

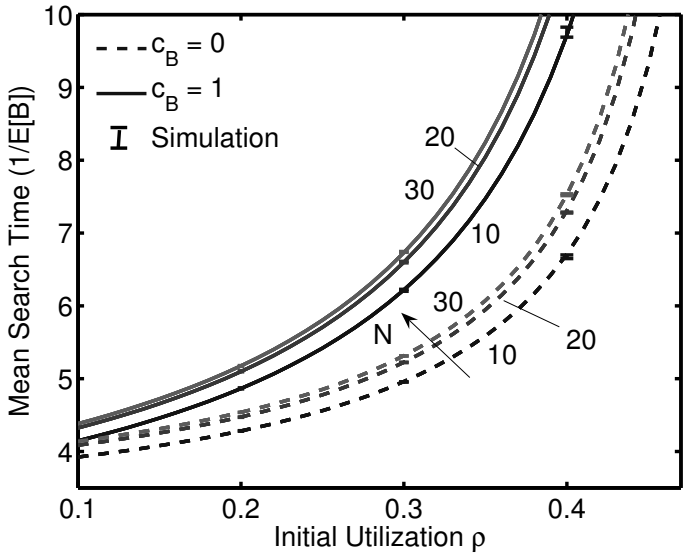


Figure 7: Effect of the system size on the mean search times

Similarly, the coefficients of variation of the search time from the same experiments, shown in Figure 8, exhibit the same behaviour as discussed in Figure 6.

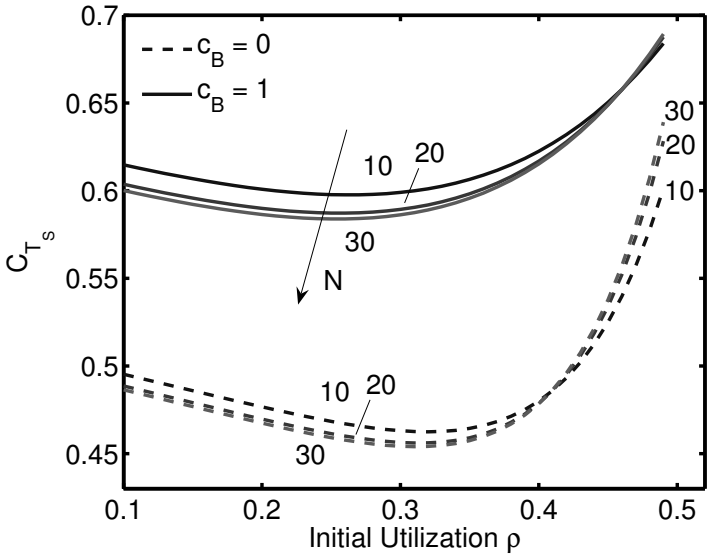


Figure 8: Effect of the system size on the variance of the search times

We can also observe a trend towards lower coefficients of variation for larger systems up to a certain point. A smaller system size leads to a higher variance, since a smaller amount of queries traverses two nodes. However, since we assume that the waiting and service time distributions of the two nodes are independent, the additional hop would lower the variance in the search time. This effect holds until the higher internal traffic in the larger system leads to longer waiting queues than in the smaller system, at least for small coefficients of variation of the service time.

4.4 Numerical Example

In this section, we present results taken from a numerical computation of the search times, using a time-discrete GI/GI/1 model as described in [10] with a Markovian arrival process with parameter λ_0 and a two-parameter representation of the service time. The reason why we use the GI/GI/1 computation algorithm instead of the analyzed M/GI/1 model is its general applicability, allowing for studies of systems with different arrival processes. However, in this work we have adapted it to fit the system model described in Section 4.1. The service time is thought to be distributed following a negative binomial distribution with a mean of $E[B]$ and coefficient of variation $c_B = 1$, while a deterministic distribution is used for $c_B = 0$.

In Figure 9, we take a look at the impact of the coefficient of variaton of the service time on the total search times for systems with 5 and 20 CSNs, respectively. The complementary cumulative distribution function (CCDF) of the total search time is shown for coefficients of variation $c_B \in \{0, 1\}$. The system utilization is set to $\rho = 0.3$.

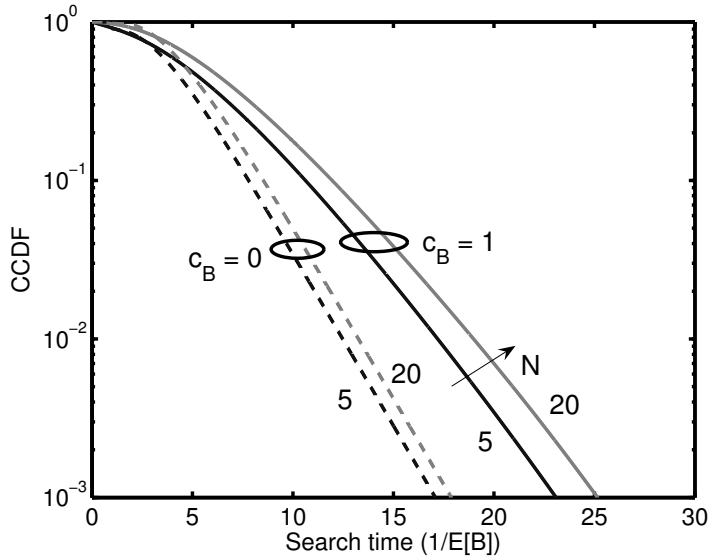


Figure 9: Effect of the coefficient of variation of the service time on the search times

We can observe that the higher variance of the service process leads to longer search times, due to the longer waiting times. This effect is stronger for larger systems, since a larger number of queries have to traverse two nodes instead of one.

5 Conclusion and Outlook

In this work, we described and evaluated a distributed, fully meshed content storage system. We provided an analysis of the search times in such a system and showed some of the most important parameters that influence it. In this context, we concentrated on the query load that the nodes in such a system have to handle, in contrast to the network traffic considerations made in other papers.

Most importantly, we considered in our analytical model the additional load generated by internal routing, which may be unimportant in distributed architectures with underutilized nodes, e.g., large scale P2P platforms, but have an impact on the dimensioning of dedicated systems. We showed that this additional load can not be neglected and has to be considered when designing the system.

While our assumption of a fully meshed DHT-like structure limits the direct applicability of the results to these systems, we believe that our approach of characterizing the load experienced by a node through modeling the traffic flows can be also used to analyze more general architectures, such as current overlay networks.

In the future, it could therefore be of interest to adapt the analytical model presented here to other distributed architectures, e.g., relaxing the assumption that the queried content is located assuredly after the internal hop. Also, a more detailed model of the query processing could be implemented, where the process times are not independent of the fact whether a node actually has the content or has to forward the query to another node.

Acknowledgement

The authors would like to thank Robert Henjes for his valuable comments and the insights gained in the fruitful discussions.

References

[1] A. Gupta, B. Liskov, and R. Rodrigues, "One hop lookups for peer-to-peer overlays," in *Proceedungs of the Ninth Workshop on Hot Topics in Operating Systems*, (Lihue, Hawaii), May 2003.

[2] A. Gupta, B. Liskov, and R. Rodrigues, "Efficient routing for peer-to-peer overlays," in *Proceedings of the First Symposium on Networked Systems Design and Implementation*, (San Francisco, CA), March 2004.

[3] B. Leong and J. Lik, "Achieving one-hop dht lookup and strong stabilization by passing tokens," (12th International Conference on Networks 2004 (ICON 2004), Singapore), November 2004.

[4] L. R. Monnerat and C. L. Amorim, "D1ht: A distributed one hop hash table," (Rhodes, Greece), IEEE International Parallel & Distributed Processing Symposium, April 2006.

[5] V. Muthusamy and H.-A. Jacobsen, "Small-scale peer-to-peer publish/subscribe," (San Diego, CA, USA), P2PKM'05: Peer-to-Peer Knowledge Management, July 2005.

[6] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, 2003.

[7] H. Takagi, *Queueing Analysis: A Foundation of Performance Evaluation*, vol. 1. Amsterdam, Netherlands: North-Holland, 1991.

[8] L. Takács, "A single server queue with poisson input," *Operat. Res.*, vol. 10, pp. 388–397, 1962.

[9] T. Raith, *Leistungsuntersuchung von Multi-Bus-Verbindungsnetzwerken in lose gekoppelten Systemen*. PhD thesis, Universität Stuttgart, 1986.

[10] P. Tran-Gia, "Discrete-time analysis technique and application to usage parameter control modelling in ATM systems," in *Proceedings of the 8th Australian Teletraffic Research Seminar*, (Melbourne, Australia), 12 1993.