

A Cache Architecture for Modernizing the Usenet Infrastructure

Thomas Gschwind Manfred Hauswirth

{tom,M.Hauswirth}@infosys.tuwien.ac.at

*Distributed Systems Group
Technische Universität Wien*

Abstract

Current Internet users see the combination of World-wide Web (WWW) and electronic mail as a synonym for the Internet. While WWW is excellent for easy and fast dissemination of information and email facilitates efficient communication, a means for wide-spread, threaded discussion and interchange of information among large groups of users is still necessary. WWW has experienced a tremendous innovation boost during the last years resulting in an elaborate networking infrastructure. Usenet News, however, has only been slightly improved over its initial definition, which is no longer apt for the increased data volume. Concepts such as caching and proxies that have proven necessary and useful for other systems of comparable size have to find their way into Usenet News. When modernizing a widely used system, continuous operation must be guaranteed. In this paper, we present and analyze different approaches that can solve the scalability problems of the current Usenet News infrastructure.

1 Introduction

Usenet News [13] offers a world-wide discussion forum, which is divided into hierarchical newsgroups dedicated to defined topics, e.g., the `comp` hierarchy deals with computer related topics and the specific newsgroup `comp.lang.java` focuses on the Java programming language.

Users can access Usenet News, commonly denoted as News, via a news reader (client) which provides the user interface and manages the interaction with the news server. Users *subscribe* to a set of newsgroups based on their interests. After selecting a specific newsgroup, contributions (*articles, postings*) can be read by the user. If it is allowed by the news server, the user may submit replies to articles or post new ones. The News infrastructure then takes care of the world-wide distribution.

For focused discussions inside a newsgroup, postings can refer to each other by using article identifiers provided by the News system. News readers use these relations for convenient presentation of related articles (*threading*). Articles can also be posted to a set of newsgroups (*cross-posting*).

The provision of this blackboard functionality on a world-wide scale has made News a successful system that steadily grows in the number of newsgroups (currently over 55000 groups [15]) and users. At the infrastructure level this implies growing amounts of data that need to be distributed to an increasing number of computers. This growth steadily pushes the existing News infrastructure to its limits [19], [5].

Though flexible and scalable, News' underlying infrastructure may soon face scalability problems with coming requirements. In this paper we analyze the scalability problems of the current News system and present several approaches that can remedy these problems, while still providing continuous operation. The paper is structured as follows.

In Section 2 we start with a description of the existing News infrastructure which is then analyzed in Section 3 with focus on scalability issues. Based on this analysis we present possible solution strategies in Section 4 with the most promising solution using an access infrastructure of cache servers.

Section 5 presents the requirements for a cache server for News and gives a description of our NEWSCACHE implementation.

Section 6 gives statistical information that show how News can benefit from caching and discusses interoperability with exiting software. Related work is considered in Section 7 and we draw our conclusions in Section 8.

2 Current Infrastructure

To motivate the need for improvements and as a basis for the analysis in Section 3, this section presents the current News infrastructure.

Figure 1 shows a simplified News network. Current News software falls into two classes: news readers (clients in Figure 1) and news servers. News readers handle interactions with the user as described above (user interface, reading and posting of articles). The Network News Reader Protocol¹ (NNRP) is used for the interaction of news read-

¹NNRP is not a protocol of its own. Usually the subset of NNTP [12] used by news clients is referred to as NNRP.

ers with the News distribution infrastructure. Each news reader communicates with exactly one news server.

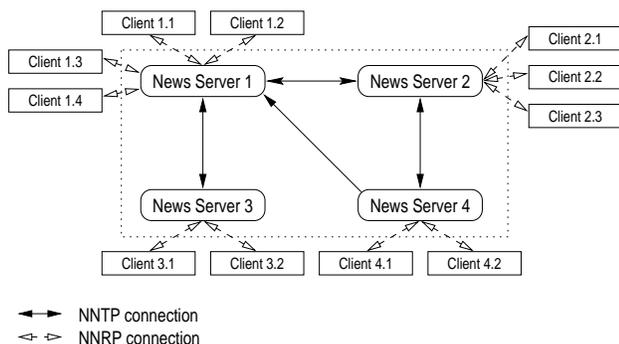


Figure 1: Simplified News network

The world-wide set of cooperating news servers makes up the distribution infrastructure of the News system. The Network News Transfer Protocol [12] (NNTP) is used to propagate articles among news servers. News servers receive articles from their clients and other news servers. Propagation of articles is commonly referred to as *news feeding*. Each news server defines which newsgroups it wants to hold, i.e., provide to its clients, and has a statically configured set of servers that constitute its news feed.

An article posted by a client is transferred via NNRP to its dedicated news server which is the client's access point to the News infrastructure. Each news server then keeps a copy of this article and propagates it to its neighboring news servers. This propagation finally results in a copy of each article on each news server in the infrastructure. Since no restriction exists on the topology and distribution paths of the News system, servers may receive an article multiple times. Hence, each article carries a globally unique identifier that allows news servers to identify duplicates. Causal ordering of articles, however, is not maintained by this distribution semantics: the different distribution paths of articles may cause that a reply reaches a news server before the article it refers to.

The data each news server holds is composed of several databases: article database, history database, and active database. The organization of these databases is simple. Usually the article database maps newsgroups and articles to the file system (C-News [6], INN [18], NNTP reference implementation). Each newsgroup is mapped to a directory with each article in a separate file. E.g., article 162 in `comp.lang.java` is stored as `comp/lang/java/162`. The history database records the article IDs and some administrative data of recently received articles in a history file, each line holding an entry for one article. The main purpose of the history database is

the identification of duplicate articles. The active database also consists of a single file that lists the newsgroups available on a news server. Each newsgroup is represented by a one line record holding the range of currently active article numbers and the moderation status of the group. Additionally, most servers keep an overview database [4] which is one of the few improvements that have been applied to the original News standard [12]: to allow news clients to quickly display a content listing of a newsgroup the overview database holds a short summary for each article (date, subject, author, etc.).

3 Infrastructure Problems

The existing News infrastructure has some severe drawbacks that have become obvious after being in operation for several years. These drawbacks are mainly related to scalability with respect to the increased amount of data that needs to be transferred by the News infrastructure. In the following we describe the problems that have motivated our approach.

An enormous redundancy is caused by copying each article of a newsgroup to all the news servers holding this newsgroup. This is a problem in the current infrastructure since a high percentage of newsgroups and articles will not be read by anyone, so distribution would not have been necessary [14]. Statistics of `news.tuwien.ac.at` show that about 66% of all newsgroups are not read [15].

News' distribution semantics stems from UUCP [16] which originally was used for article distribution. UUCP is batch-oriented and does not support on-demand connections like HTTP.

As described in [15] and [5] a typical news feed requires 2.5 GB/day. In terms of network bandwidth the permanently needed bandwidth to transfer the current amount of data is approximately 400 Kbit/s. This means 27% of the bandwidth of a T1 link (1.5 Mbit/s), which is the typical Internet link for many sites, is dedicated to the news service.

Another problem is the I/O load of the news server. On our university's news server, news clients retrieve over 2.5 GB of article data every day, excluding requests for active and overview records. This means that news clients retrieve more articles than are fed to the news server (the same article can be requested by different clients). By mapping this number onto the I/O load caused by the news feed and the clients, we can estimate that clients are responsible for over 50% of the I/O load on the news server. Thus, a higher degree of scalability can be achieved by separating communication with clients from server communication.

A backlog in the news feed may occur if either the above minimum bandwidth requirement is not met, the news server is not operable for some time, or no Internet

connectivity is available at all. If a backlog cannot be recovered, the quality of the news service has to be lowered either by decreasing the number of available newsgroups or by expiring articles earlier.

News' copy semantics also implies high disk space requirements, mainly caused by the filesystem-oriented organization of the article database: each article is stored in a separate file. Thus, considerable amounts of disk space—up to over 100% of the real data volume—are wasted [19].

Currently scalability problems are mainly managed by inflexible administrative measures. The approach presented in [5] suggests to replace one news server by a set of cooperating news servers where each news server holds part of the article spool. One dedicated news server is responsible for news distribution and the other servers handle client requests. This approach increases hardware expenses by approximately 100%, adds administrative overhead, and only provides a workaround for the problem. Our solution in contrast reduces both hardware requirements and administrative effort.

4 Solution Strategies

The News system steadily grows at a high rate. Statistical data on the growth of News in terms of disk space requirements and number of articles are given in [19]: by analyzing historical data of large news servers which show a doubling of article numbers nearly every 18 months, future numbers are extrapolated. In the current infrastructure these figures can easily be mapped onto network bandwidth requirements which are likely to grow at a similar rate. As noted above the current infrastructure is not apt in the long run for these rapidly growing requirements. The current infrastructure already starts to operate in the “red range” of its specifications and might soon be overloaded.

Improvements to the existing infrastructure have to be applied before this situation occurs. In the following sections we present several strategies that may solve this problem. The main focus will be on network bandwidth requirements, I/O load of the news server, and its compatibility to the existing News infrastructure. Each strategy has a special impact on the long-term development and requires a different spectrum of changes to existing configurations.

4.1 Central Server

In the central server model users are serviced by a single global news server. It is obvious that a central server without caching cannot replace the current News system. Without a caching infrastructure, the server would get overloaded and response times would be unacceptably slow. Another disadvantage is that this approach does not provide

all of the functionality that is possible. Organizations may want to hold private newsgroups that must not be stored on a foreign news server. Thus private groups are not supported by this approach.

4.2 Caching Infrastructure

By introducing caches into the News infrastructure articles are fetched on demand rather than propagated to all news servers. Given n_i is the number of accesses and sz_i is the size of article i , the reduction of network bandwidth can be approximated as

$$Reduction = \sum (n_i - 1) * sz_i \quad \text{[Bytes]}$$

Each requested article is stored locally by a news cache to satisfy further requests without having to contact the news server again. This eliminates additional transfers and decreases the load on the news server. Distribution and reading functionalities are separated. The application of caching is perfect for News because the content of an article does not change over time. Articles can only be added, deleted, and expired and their identifiers must not be reused.

Although a caching infrastructure does not solve all infrastructure problems, an appropriate caching scheme reduces the load on news servers and the network bandwidth requirements. In addition, a cache server can be used instead of a leaf node news server (a server that has only one link to another news server) which lowers the necessary hardware requirements for providing full access to the news system.

Caches can be employed for different access patterns as shown in Figure 2.

Usually networks connected to the Internet have fast local connections and a considerably slower link to the Internet. If this link is too slow for an intended news feed, News can either be accessed directly over the slow link or the feed must be cut down by reducing the number of available newsgroups.

Figure 2(a) gives the standard pattern of cache application for the News area: the cache is used to minimize traffic over a slow connection, which is the limiting resource. By introducing a cache server into this setting a virtual full feed is possible since transferred data is reduced to a minimum: only articles requested or posted by clients are transferred. Articles are retrieved only once and cached locally. Successive requests are satisfied by the cache via the high-speed link.

The cache server can also be used to reduce the load on heavily used news servers. Instead of directly connecting to the local news server, clients connect to a cache as shown in Figure 2(b). Clients may connect to whatever cache they

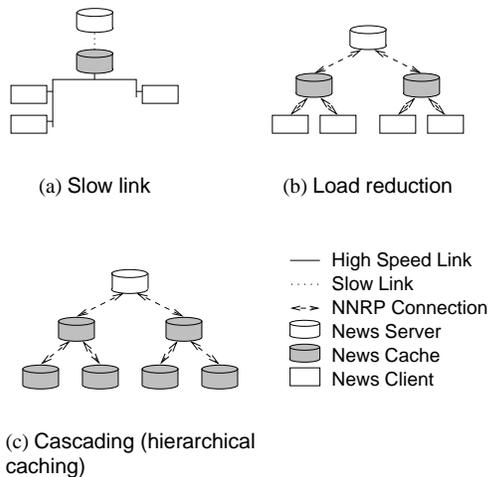


Figure 2: Architectural patterns

prefer provided that the caches are connected to the same news server because consistent access to articles (article numbering) is guaranteed. Round-Robin DNS [11] can be used for selecting caches and balancing their loads. This approach frees resources on the server and reduces operational problems.

Figure 2(c) finally gives a pattern that aims at large-scale application of caches, e.g., for national cache infrastructures. Hit rates can be increased and resource consumption can be reduced by cascading caches [7]. If a cache at a certain layer cannot satisfy a request, it forwards the request to its upstream cache and so forth. Since NEWSCACHE is an NNRP server and only uses NNRP for communication with a news server, setting up a tree is possible easily. This pattern, however, has to be used carefully since performance might degrade if the tree is too high.

A cache infrastructure cuts down on the required network bandwidth and the I/O load caused by news clients but does not attack the bandwidth requirements between different news servers.

4.3 Multicast Distribution

Multicast is a mechanism to send a packet to a set of machines without having to send it to each machine separately. An approach based on multicast to attack network bandwidth problems is given in [17]. Since it uses unreliable multicast for the article distribution, the current distribution infrastructure is still necessary. This approach is interesting but needs to be matured.

While multicast might be used to reduce the required network bandwidth for the article distribution between news

servers, it does not reduce the news server's I/O load. On the contrary, it imposes higher load on the news server, because the sender of all articles has to be verified [17].

4.4 Compressed Distribution

Network bandwidth could be decreased by distributing articles in a compressed form. This can be implemented on the client or on the server side. On the server side, however, this would impose additional I/O load because the articles must be compressed and decompressed. If news readers were responsible for (de)compression of articles, the news server would store the articles in compressed format and thus disk space would be conserved as well.

Due to compression keyword search is complicated. For this functionality, however, servers like *DejaNews* [2] can be used which provide a much better search facility anyway because news articles are indexed and stored over a much longer period of time.

Articles can already be compressed on the client side using MIME messages [9] and setting the content transfer encoding accordingly. Unfortunately this is not yet transparently implemented by news reading clients and would break many (older) clients that cannot handle MIME messages.

Although the required network bandwidth is reduced by this approach, the number of articles and their organization which are the main contributors for the I/O load are not changed.

4.5 Lightweight News Servers

A *lightweight news server* is a news server which stores only a subset of the available newsgroups. If news clients would be able to access News from different news servers, the administrator of a news server could select and store newsgroups of highest local interest only. All the other newsgroups would be retrieved from other news servers. This reduces the size of the database and the performance on the news server increases since fewer articles will have to be indexed. Another advantage is that only those newsgroups which are actually read will be transferred from the news feed and thus I/O load will be reduced.

This configuration, however, has a problem. Since a given news server would only store selected newsgroups, clients would have to access newsgroups from other news servers over a possibly slower connection. To overcome this problem a proxy/caching infrastructure is necessary.

Unfortunately, current news clients can only connect to one news server at a time. If different news servers shall be accessed, the user needs to switch by hand and maintain a set of configurations.

By introducing a proxy server, as shown in Figure 3, we can eliminate this drawback. The proxy server takes newsgroups from several news servers and merges them into one virtual news server that provides all newsgroups. This functionality is already included in our NEWSCACHE implementation. NEWSCACHE can act as a proxy and multiplex between a set of news servers that then appear to the client as a single server.

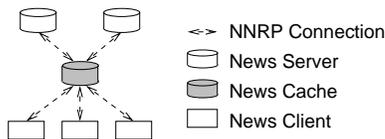


Figure 3: Server multiplexing

4.6 Distribution Backbone

Another approach is to setup a News distribution backbone, consisting of a subset of the currently available news servers. This subset would be in charge of article distribution. For news reading, other dedicated servers would exist. The distribution backbone consists of standard news servers using standard procedures for distribution (NNTP). The access infrastructure is implemented using caches (e.g., NEWSCACHE). Figure 4 depicts this architecture.

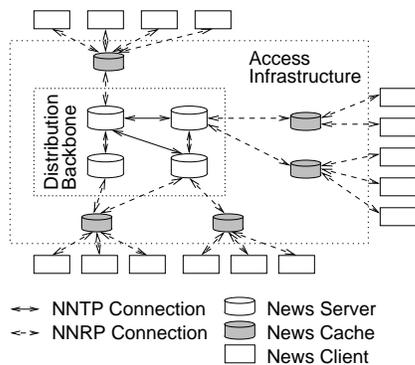


Figure 4: Enhanced News infrastructure

The advantage of this configuration is that the I/O load caused by news clients is reduced through the access infrastructure. Additionally, network bandwidth is conserved since the whole news spool is exchanged among fewer news servers.

4.7 Discussion

Table 1 compares the above approaches in respect to their feasibility and benefits.

The central server approach has only little complexity, but is inflexible. It cannot provide local newsgroups and from a political point of view, it would be impossible to find a place for this server.

The realization of the compressed distribution model is rather problematic because many older news readers do not support MIME (e.g., slrn, xrn, etc.). Thus users of such programs would be banned from reading News. On the other hand none of the currently available news readers supports transparent compression of news articles.

Until all news servers are capable of receiving news via multicast, the old distribution infrastructure has to be maintained for those news servers that do not support multicast. This also holds true if unreliable multicast should be used for news distribution. Using reliable multicast here is difficult because IP multicast [8], the only multicast protocol that is already used on a global scale, is unreliable.

A caching infrastructure can be used in addition to the other approaches. The advantage of caching is its upgrade path. As we will show in Section 5, it does not require any changes to the current news infrastructure. Wherever client load shall be decreased a cache server can simply be plugged in between the clients and the news server.

For the exploitation of the lightweight server model, the necessary access infrastructure has to be provided first. Then News administrators can start switching over by eliminating less frequently used newsgroups from their news servers. This is no problem because users can access all newsgroups using a multiplexing proxy server or a cache server.

The same accounts for the distribution backbone. With the availability of the access infrastructure, news servers whose main purpose is to distribute the load of news clients will be replaced by caches.

The most promising solution in our opinion is to implement the necessary access infrastructure first. Then the currently existing infrastructure will be replaced by a hybrid version of the distribution backbone and the lightweight server model. We especially favor the idea of a cache infrastructure since it can be used in combination with most of the other approaches.

5 Requirements for a News Cache Server

In this section we present the requirements for a cache server (based on our NEWSCACHE implementation). A detailed description of NEWSCACHE's prototype implementation can be found in [10].

Table 1: Comparison of different distribution infrastructures

	compatibility	bandwidth req.	I/O load	hardware req.
central	clients	higher	higher	higher
cache	clients and servers	lower at access in- infrastructure	lower at access in- infrastructure	lower
multicast	modified clients and servers	lower	higher	higher
compression	modified clients	lower	equal	lower
lightweight	proxy necessary	lower	lower	lower
backbone	clients and servers	equal	lower (no clients)	lower (fewer nodes)

To be an adequate solution, the cache server has to meet two main requirements:

- It has to fit into the existing News system seamlessly.
- It has to reduce the load on the news server caused by clients and decrease network bandwidth requirements.

5.1 Cache Accuracy

The cache server has to issue control messages to ensure that the cached data are accurate. Each article carries a per newsgroup article identifier. The article corresponding to an article identifier must not be changed and its identifier may never be reused. Thus, a cached article will remain valid forever and no cache consistency verification is necessary.

In addition to the article identifier, each article has a per newsgroup article number. The lowest and highest article numbers in a newsgroup are denoted low and high watermarks. Whenever a new article arrives on the news server, the news server increases the high watermark and uses this value for the new article's per newsgroup number. This allows clients to detect new articles simply by requesting the low and high watermark (using the `group` command).

Since no callback functionality is included in the news server, the news reading client has to poll at regular intervals whether new articles have been added. This means that the news cache has to check for new articles whenever a client requests a newsgroup even if another client has requested the same newsgroup just a moment before. If we apply a more relaxed policy for cache accuracy, we can further reduce the load of the news server: we assume that the information, we have retrieved from the news server will not change within a certain time interval t .

The result of a relaxed caching accuracy is that new articles will be delivered to the cache server's clients with an average delay of $t/2$. If the news cache should provide new articles immediately, the timeout must be set to zero. This might be interesting for newsgroups with high local

interest. Thus the timeout should be configurable on a per newsgroup basis.

5.2 The Client and Server Interfaces

Since NEWS-CACHE has to fit into the current News system seamlessly, it can only rely on interfaces that are already used in existing news readers and servers. This compatibility requirement implies that the client-side interface must use NNRP, because this protocol is used by news clients for article retrieval.

On the server side either NNRP or NNTP must be used. NNTP has several disadvantages for this purpose, since it was designed for the transmission of a full news feed and requires maintenance work on the cache's news server to establish the news feed. Hence, we have decided to use NNRP for NEWS-CACHE.

For the interaction among caches additional commands may be introduced (NNTP allows to define vendor-specific commands). Since these commands are only used between news caches they do not affect the compatibility with news clients and servers.

5.3 Replacement Strategy

The replacement strategy has to be distinguished from the article expiration. Article expiration means that an article is removed from the news server after a given period of time. This can be detected using the newsgroup's low and high watermark. The replacement strategy, however, decides which article to remove from the cache to make place for the new article if no more space is available in the cache area.

Several replacement mechanisms are applicable:

Least Frequently Used replaces those articles with lowest access rates first. This strategy assumes that frequently accessed articles are more likely to be accessed again.

Least Recently Used replaces those articles with the oldest access time stamps first. The assumption is that articles

which have not been accessed for a long period will not be accessed again.

Oldest Article First assumes that articles with older time stamps will be less probably accessed than newer ones. Additionally those articles would be expired by the news server earlier; consequently they would have to be removed from the cache anyway. In this context, oldest is meant relative to the expiration time of a specific newsgroup.

Biggest Article First assumes that the probability that one big article is referenced is lower than the probability that any of several smaller ones is accessed, which reuse the disk space of the big article. This strategy penalizes the *binary* newsgroups (e.g., the *alt.binaries* hierarchy).

At the moment we do not have enough evidence to judge which of the above strategies are the best. However, we assume that this may vary considerably according to user profiles and that a combination of the above strategies will prove to be optimal.

The current version of NEWS-CACHE uses LRU on a per newsgroup basis.

5.4 Handling News Postings

Postings submitted to the cache server may be passed on to the news server directly. If the news server is not available, postings are queued for later submission.

Another possibility is to store postings on the cache in a way that makes them available to the user immediately, even though they have not been passed on to the news server. This strategy, however, requires the cache server to use its own article numbering, since an article number must be assigned to each article that is presented to the user. This implies that the cache server has to maintain a mapping between local article numbers and news server article numbers. Additionally the user cannot switch between different NEWS-CACHES attached to the same news server because news clients usually use the article numbers to log articles read by the user. NEWS-CACHE uses the first approach.

5.5 Implementation

The architecture of NEWS-CACHE is divided into several components as shown in Figure 5. The *server module* serves as the communication facility between the database module, which stores the news articles and related information and the client module, which handles the communication with the news clients.

The server module is also responsible for the interaction with news servers. The *remote server* component implements a news server for accessing the news database of other news servers. This component allows developers

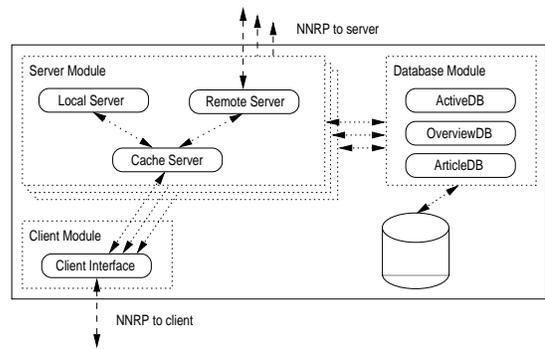


Figure 5: NEWS-CACHE architecture

to implement news readers without knowing the details of NNRP, because it encapsulates all communication actions.

The *local server* component implements a news server with a local news database and can be used as a base for a fully featured news server.

The *cache server* component combines the local and remote server components and implements the cache functionality. It uses the remote server to retrieve news from other news servers and uses the local server to cache received information. All types of information—articles, overview database, and active database—are cached and maintained by this component.

The client interface handles the connections to news readers. When a request is received from a news reader, the client interface chooses a server module based on the name of the current newsgroup. The client interface also translates the client's request into the cache server component's programmatic interface. The cache server component then checks whether the request can be fulfilled by the local server component. If so, the local server passes the information back to the cache server, which returns it to the client interface. Otherwise, a request is sent to the remote server component which in turn submits it to the news server. As soon as the request is completed by the news server, the result is passed on to the appropriate database component and to the cache server component which returns it to the client interface. Finally, the client interface translates the result into NNRP data format.

Since the same news database is shared by all server components and all processes of NEWS-CACHE running on one machine, the information requested from the news server is available to all other components and to all NEWS-CACHE processes as soon as it is passed to the database by the remote server component.

6 Evaluation

As explained in [14] the news server of our university has been switched to the lightweight news server model. If a newsgroup is not available users can request it to be added to the news server. A WWW interface for such requests exists. By using a cache server this could be implemented transparently and would not require any interaction from the user. In the average our news server spools one tenth of the theoretically possible news feed.

Figure 6 shows the percentage of accesses to the top n percent of the news groups (solid line) and the amount of disk space they use in percent of the size of the total news spool (dashed line). This shows that for instance, the 10% most frequently read newsgroups make up 97% of all the accesses but only 23% of the news spool. The values are taken from `news.tuwien.ac.at` which holds 60GB of article data and is accessed about 15000 times by 700 different hosts each day. [15]

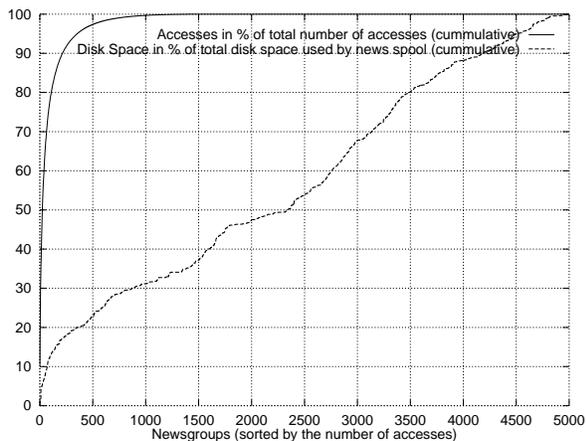


Figure 6: Accesses to the top n percent of the news groups and the amount of disk space they use

To evaluate the benefits of caching, we have setup NEWS-CACHE with a cache area of 3GB disk space and our university's news server as its upstream news server. An analysis of NEWS-CACHE's log files shows that even though the cache server was used by only 5% of the users (about 600 accesses by 60 different hosts each day), we got a hit rate of 25%. With more accesses to the cache we expect even higher hit rates, especially when people have to use the cache and cannot access the news server directly.

To ensure inter operability with existing News software, we have tested NEWS-CACHE with different versions of INN [18], which is by far the most frequently used server software. Some users of NEWS-CACHE reported problems with other news servers which have been fixed by now (e.g., AMU NEWS under VMS). Most of the problems

with other news servers originate from not implementing RFC977 or the new NNTP draft [4] precisely.

The most popular news readers have been tested with NEWS-CACHE, too. GNUS (Emacs) and TIN work without any problems. XRN works without problems but required some additional effort since it insists on optional features explained in [4]. Netscape's news reader works without any problems. Since it issues the GROUP command for every newsgroup to get a better estimation of the number of articles within the newsgroup, NEWS-CACHE's active database had to be optimized to make this delay small.

Currently, an increasing number of sites are using NEWS-CACHE and it is already included in the Debian Linux distribution [1].

7 Related Work

Related work focuses on statistics to forecast Usenet growth, on managing the resulting load on news servers, and on improving the News infrastructure and NNTP.

How resource requirements of News are monitored at the Stanford Linear Accelerator Center is presented in [19]. The goal is to forecast News growth and derive plans for hardware changes in order to guarantee stable operation of the news server.

An administrative ad hoc solution for growing resource requirements of News is given in [5]. It is intended for a news server at a single site, does not attack general News infrastructure problems, and requires significant hardware investments. The idea is to have a dedicated server that manages the news feed and several other servers that can access the news spool read-only offering read access to clients. Thus reading load is distributed and the server being in charge of the news feed is freed from read accesses. Additionally, the news spool disk space is partitioned among all servers. Access to the complete news spool is provided by extensive use of NFS cross-mounts.

NNTP-CACHE [3] takes a similar approach like NEWS-CACHE. While NEWS-CACHE uses a specialized hash table for storing the active database, NNTP-CACHE uses UNIX dbm format. For performance reasons both systems store the overview database in memory mapped databases using proprietary formats. Features of NNTP-CACHE not yet included in NEWS-CACHE are: censoring of articles, and forwarding of unknown commands. On the other hand NEWS-CACHE offers prefetching, off-line News reading, and UNIX inetd support which are not supported by NNTP-CACHE. Since no papers on NNTP-CACHE were available at the time of this writing, the above statements are based on NNTP-CACHE's implementation and our tests with it.

A networking-level approach based on multicast to at-

tack the article distribution problem is given in [17]. This approach has already been discussed in Section 4.3.

8 Conclusion

In this paper we have analyzed the current status of the News infrastructure and identified disk space consumption, enormous I/O load, and network bandwidth bottlenecks as its main problems.

We discussed several solution strategies like the lightweight server model, compressed article distribution, multicast, and a caching infrastructure. Additionally we presented an enhanced infrastructure using cache servers for future News distribution. This infrastructure fits seamlessly into the existing infrastructure without requiring modifications to existing software. As we explained, another advantage of this approach is that it does not impose new hardware requirements and that the administrative overhead is decreased.

It attacks the identified problems, supports flexible architectural usage patterns, is scalable, and can easily be adapted to future News standards. We believe that NEWS-CACHE or components that offer similar properties will soon be added to the existing News infrastructure since its scalability problems will get worse due to the continuing growth of Usenet. Future work on NEWS-CACHE will focus on improved storage models for News databases and management support.

A detailed description of the initial NEWS-CACHE prototype is given in [10]. The NEWS-CACHE software is available at <http://www.infosys.tuwien.ac.at/NewsCache/>.

Acknowledgments

We want to thank Michael Gschwind and Mehdi Jazayeri for their comments on draft versions.

References

- [1] Debian Linux Distribution (Hamm release). <http://www.debian.org/>.
- [2] The Discussion Network. <http://www.dejanews.com/>.
- [3] Julian Assange and Luke Bowker. NNTPCache. <http://www.nntpcache.org/>.
- [4] Stan Barber. Network News Transport Protocol, September 1998. Internet Draft, draft-ietf-nntpext-base-04.txt.
- [5] Nick Christenson, David Beckemeyer, and Trent Baker. A scalable news architecture on a single spool. ;*login.*; 22(3):41–45, June 1997.
- [6] Geoff Collyer and Henry Spencer. News Need Not Be Slow. In *Proceedings of the Winter 1987 USENIX Technical Conference*, 1987.
- [7] Peter Danzig, Michael Schwartz, and Richard Hall. A case for caching file objects inside internetworks. In *Proceedings of the ACM SIGCOMM 93 Conference*, pages 239–248, September 1998.
- [8] S. Deering. Host Extensions for IP Multicasting, May 1988. RFC1054.
- [9] Ned Freed, John Klensin, and Jon Postel. Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures, November 1996. RFC2048.
- [10] Thomas Gschwind. A Cache Server for News. Master's thesis, Technische Universität Wien, April 1997. <http://www.infosys.tuwien.ac.at/NewsCache/>.
- [11] Martin Hamilton and Russ Wright. Use of DNS Aliases for Network Services, October 1997. RFC2219.
- [12] Brian Kantor and Phil Lapsley. Network News Transfer Protocol - A proposed standard for the stream-based transmission of news, February 1986. RFC977.
- [13] David Lawrence and Henry Spencer. *Managing USENET*. O'Reilly & Associates, Incorporated, 1998.
- [14] Martin G. Rathmayer. Realisierung eines Bestellsystems für Newsgruppen an der TU Wien. *Pipeline*, (23), October 1997. In German.
- [15] Martin G. Rathmayer. Personal communication about access statistics of `news.tuwien.ac.at` (based on INN's log files), March 1998.
- [16] Ed Ravin, Tim O'Reilly, Dale Dougherty, and Grace Todino. *Using & managing UUCP*. O'Reilly & Associates, Incorporated, September 1996.
- [17] Heiko W. Rupp. A Protocol for the Transmission of Net News Articles over IP multicast, March 1998. Internet Draft, draft-rfced-exp-rupp-04.txt.
- [18] Rich Salz. InternetNews: Usenet Transport for Internet Sites. In *Proceedings of the Summer 1992 USENIX Conference*. USENIX, June 1992.
- [19] Karl L. Swartz. Forecasting disk resource requirements for a Usenet server. In *Proceedings of the Seventh System Administration Conference (LISA '93)*, pages 195–202. USENIX, November 1993.