Forecasting Disk Resource Requirements for a Usenet Server

Karl L. Swartz
Standford Linear Accelerator Center

Forecasting Disk Resource Requirements for a Usenet Server†

Karl L. Swartz – Stanford Linear Accelerator Center

ABSTRACT

Three years ago the Stanford Linear Accelerator Center (SLAC) decided to embrace netnews as a site-wide, multi-platform communications tool for the laboratory's diverse user community. The Usenet newsgroups as well as other world-wide newsgroup hierarchies were appealing for their unique ability to tap a broad pool of information, while the availability of the software on a number of platforms provided a way to communicate to and amongst the computing community. The previous way of doing this ran only on the VM mainframe system and had become increasingly ineffective as users migrated to other platforms.

The increasing dependence on netnews brought with it the requirement that the service be reliable. This was dramatically demonstrated when the long-neglected netnews service collapsed under the load of the traditional fall surge in Usenet traffic and the site was without news service for a week while an upgraded system was installed. One result of that painful event was that efforts were made to forecast growth and the accompanying hardware requirements so that equipment could be acquired and installed before problems became visible to the users.

This paper describes the major on-disk databases associated with news software, then presents an analysis of the storage requirements for these databases based on data collected at SLAC. A model is developed from this data which permits forecasting of disk resource requirements for a full feed as a function of time and local policies. Suggestions are also made as to how to modify this model for sites which do not carry a full feed.

Why NetNews? Why Usenet?

The Stanford Linear Accelerator Center (SLAC) is a medium-sized national research laboratory. The lab's primary missions are research in elementary particle physics and development of new techniques in particle accelerators. These are large projects that often involve international collaborations and a diverse user community. This can present a formidable problem for communication with and amongst users, from discussions on the design of new experiments to progress reports on current experiments, as well as the more mundane but equally necessary announcements of network or server outages. There is also a tremendous need to stay in touch with what other researchers are doing.

In the eighties, the majority of computer users at SLAC logged onto an IBM mainframe running VM. The default user profile on VM brought up a VM News session upon login, where announcements could be made. Most users would see them since they logged into VM regularly. Discussions were handled by mailing lists maintained by LISTSERV, plus a home-grown VM conferencing system named CONSPIRE. (VM News was also developed at SLAC.) BITNET mailing lists provided the main contact with researchers at other sites.

There was also a smaller, though still sizable, group of users who used VAX/VMS systems most of the time, if not exclusively. These users tended to be excluded from VM News and from CONSPIRE, though they did make use of BITNET. DECnet mail with other physics sites was also available. This communications block posed some problems but was tolerated at the time.

Over the past few years the computing environment became far more diversified. Unix workstations began to appear, while the PCs, Macintoshes, and Amigas grew powerful enough that their users had dwindling need for VM. The number of users who did not use VM on a regular basis, if at all, increased until the VM-based communication model began to fail completely. A new model was needed that would permit users to read announcements and participate in discussions from whatever platform they were accustomed to using. The multitude of platforms made another home-grown solution undesirable. The software which supports Usenet (referred to hereafter as netnews software to distinguish it from the network itself), with ready availability of NNTP-based readers for a variety of platforms, seemed to solve the problem except for VM. The discovery of the PennState VM NetNews system completed the solution.

†This work supported by the United States Department of Energy under contract number DE-AC03-76SF00515, and simultaneously published as SLAC PUB-6353.

B News and NNTP software was built and installed on a Sun fileserver which had some spare disk space, and the PennState NetNews software, with NNTP software from Queen's University, was installed on VM.[1] Once the bugs were worked out of this system, NNTP-based readers were acquired for other platforms, often with help from interested users. Meanwhile, a local *slac* newsgroup hierarchy was being populated with new groups and users were being introduced to the new system.

There was also a great deal of interest in non-local groups, of course, i.e., Usenet. After a great deal of debate over proper use of government-funded equipment, the appropriateness of censorship in an academic/research community, and the feasibility of determining just what was and was not appropriate, it was decided to carry all groups and assume a mature and responsible user community.

Netnews flourished and mostly solved the communication problem, but created a new problem in that users came to expect it to be reliable. By early 1992 the ever-increasing growth of Usenet traffic had begun to severely strain the resources of the Sun which was trying to run netnews while also handling file service and a variety of other tasks. Spool areas would overflow and the now-obsolete B News software would collapse, causing substantial delays and user ire. The expiration noose would be tightened another notch, staving off disaster for a bit longer but also aggravating already irate users as articles expired before they could be read.

Work was begun to determine what equipment was needed to support netnews for the next few years, and to define requirements in the form of minimum expiration times. Eventually a SPARCserver 2 with 4.5 GB of disk space was ordered, arriving mere days too late to avert the collapse of SLAC's netnews service from the traditional September surge in Usenet traffic. While painful at the time, the degree of pain made it clear just how critical netnews had become to SLAC.

Popular reference books on managing Usenet are notably silent on the matter of forecasting resources. [1, 2] Therefore, a study of Usenet growth was begun and has continued, so that future growth needs can be anticipated and handled before another crisis occurs. This paper documents the current state of this ongoing study.

Organization of a Netnews Server's Data

A netnews server is composed of a number of databases, several of which have the potential to require a substantial amount of disk space and which fluctuate in size as a function of news activity. The three primary databases are the articles themselves, the history file, and the active file. Ancillary structures include thread databases, incoming and outgoing spool areas, and log files. While the following description is based on a Unix system running the C News software, most structures will likely be similar on other systems.

The article database is by far the largest portion of a news system. Unix's hierarchical directory structure is a handy analog to the newsgroup hierarchy, so the software uses a simple lexical mapping of newsgroup names into directory names (''.'' is mapped to ''/'') with each article stored in a separate file. For example, article 42 in group *news.announce.important* is stored as `news/announce/important/42` within the spool directory (often `/usr/spool/news`). Cross-posted articles are handled by links. These are normally hard links, though there is some support for falling back to symbolic links if a hard link fails.[2]

Storing each article in a separate file implies a tremendous number of relatively small files. If one eschews the apparently lightly tested symlink support – an option not likely to be available for large netnews servers for much longer – the use of links for cross-posted articles further implies that this entire database must reside within a single disk partition. Unfortunately, the combination of cross-posts, cancellations, varying expiration times for different newsgroups, and the common desire to be able to use grep and other Unix tools on the article database make an alternative implementation difficult and thus unlikely in the near future. (This is a recurrent thread in the *news.software.** newsgroups.)

The history file, typically located in `/usr/lib/news/history`, records the article ID of each article seen recently by the news system along with the time it was received, any explicit expiration time, and, if the article has not yet expired, a list enumerating the newsgroups the article appears in and its sequence number in each group. All of this is stored in a simple, albeit large, file with one article mentioned per line. In order to speed lookups by article ID, an index is maintained, typically using dbz.

The last of the primary news databases is the active file, typically `/usr/lib/news/active`, which lists each newsgroup known to the news system and, for each group, the range of article

[1] VM NetNews is a full news system, not an NNTP-based reader. There are now several NNTP readers available for VM, including a port of rn. These are being investigated, with the intention of consolidating on a single, UNIX-based netnews server.

[2] The documentation for the February 1993 C News Performance Release describes symlink support with uninspiring phrases like ''half-hearted code'' and ''this has not been tested much recently.'' [3] There appears to be more confidence in the symlink support in INN 1.4. [4]

numbers currently active along with the moderation or other status of the group. Like the history file this is a simple file. It is small, however, and so will not be considered further in this paper.

The largest of the ancillary structures is one or possibly several thread databases, which allow newsreaders to present related articles in a logical order, rather than in whatever order they happened to arrive on the netnews server. One example formed the heart of the *trn* newsreader. [5] This required about 5% of the space required for the articles, [6] a not insubstantial usage of space which was compounded by the development of several incompatible solutions for other newsreaders. Fortunately, a standard thread database is being adopted in the form of Geoff Collyer's Overview or NOV (News OverView). Unfortunately, the generality required to support the varying needs of different newsreaders, current and future, demands more space – approximately 10% of the space consumed by articles. [7]

Whatever threading mechanism is used, a separate file is maintained for each newsgroup, with configuration options to choose where these files are stored. One choice is to store them amongst the articles for the newsgroup, e.g., *News OverView*'s database for the *alt.sewing* newsgroup might reside in `/usr/spool/news/alt/sewing/.overview`. Since the article database is already quite large, and possibly constrained to a single partition, a more prudent choice is to use a separate directory tree, e.g., *trn* might store its database for the same newsgroup in `/usr/spool/threads/alt/sewing.th`.

The incoming spool area contains (batches of) articles that have been received but not yet processed by the news system. As long as the netnews server processes new articles expeditiously, this requires only modest amounts of space. (In fact, INN handles incoming NNTP traffic entirely within memory, making its space requirement zero.)

Outgoing spool space is even smaller, since it contains no articles, just references to them in the form of article IDs and/or pathnames, and these hopefully don't stick around very long. For transport mechanisms other than NNTP, e.g., uucp, space will also be required for batches queued for other systems. This paper doesn't consider this requirement, as most large sites are using NNTP these days.

Finally, there are the various log files, stored under `/usr/lib/news`. The main log records each article received by the system along with a timestamp and an indication of what was done with the article. Normally this log is restarted each night, with a few days worth of old logs kept around. There is also `errlog`, which hopefully is empty or nearly so, and perhaps `batchlog`, which should also be quite small.

Factors Influencing the Size of Databases

The size of the various databases described previously depends on a number of factors. Some, such as article life before expiration, are controlled by the news administrator. Others depend entirely on external influences and, over time, may vary.

By far the most ravenous consumer of disk space in netnews is the article database, so it makes sense to scrutinize most carefully those factors which affect the size of this database. The major ones are the average size of an article, and the rate at which new articles are received. Filesystem overhead can be significant as well, but this is a fairly simple function of the other two factors and, of course, the characteristics of the netnews server's system software.

Other factors include the size of article IDs and newsgroup names, cancellation rates, and cross-post rates. These play a comparatively minor role in the growth of disk consumption by netnews, as they are only slowly changing, and they have little effect on the article database. A cursory study of these factors – sufficient to establish some baseline data – is all that is attempted here.

Number of Articles Received per Week

From the beginning SLAC's netnews server was configured to keep one week's worth of old logs, and to generate a weekly traffic report based on these logs.[3] Many sites in the San Francisco Bay Area do this, and post the results to *ba.news.stats*. The SLAC reports were also archived for posterity, and this provided the raw material for studying the growth of news traffic. Comparison with selected reports from uunet [8] gives article counts of 85% to 90% for comparable periods, so this is representative of a typical large site that carries a full feed, but not every regional hierarchy carried by uunet.

Figure 1 shows the number of articles accepted per week by SLAC's netnews server. Note that this is *not* the number of articles received, as that number includes duplicates which have no effect on the amount of disk space required. Comparison of same-week data between two years suggests something close to a 67% annual growth rate. Aside from this rather stunning growth rate, the graph has a number of interesting features.

[3] The script to generate this report was written by Erik Fair and included in the B News distribution as `misc/report-awk`. The script was modified for C News, and C News itself was modified to provide some additional information in the log. This work was done by Larry Blair, with further modifications by Dave Rand. Karl Swartz made additional modifications and packaged everything up. This version is available via anonymous ftp from `ftp.slac.stanford.edu` in the file `pub/kls/news/c-news-report.tar.Z`.

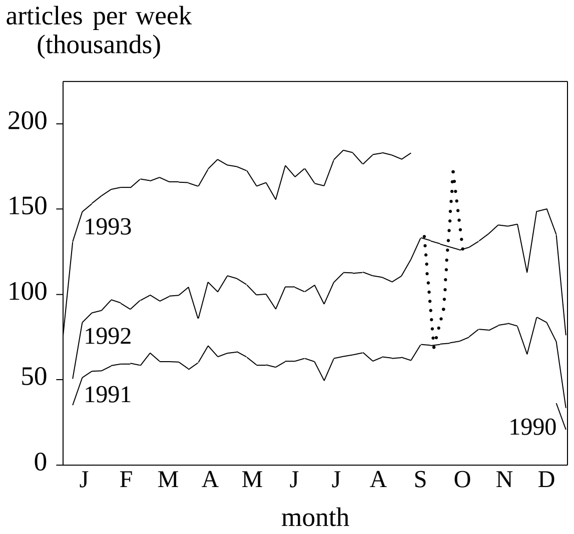

Figure 1: News articles accepted per week by SLAC's netnews server, with interpolation of data during the collapse of the old server and installation of the new one in September 1992

One noteworthy feature is the traditional September surge in traffic as many students return from summer vacations. The dotted line shortly after this onslaught in 1992 depicts the collapse of SLAC's old netnews server, along with the recovery after installation of the new server. (The analysis in this paper is based on the interpolated data for this period.)

Also interesting are the dips during holidays. Christmas/New Years was no surprise, nor was Thanksgiving, in late November. The dip in early July was a bit puzzling at first, but was quickly identified as the July 4th holiday – Independence Day in the United States. The size of the effect of a holiday unique to the U.S. did come as a bit of a surprise, given the worldwide nature of modern Usenet. (In retrospect, if this was surprising, Thanksgiving should have been as well.)

Other fluctuations appear to be correlated to certain annual events as well, but the evidence is weaker. Looking for many more patterns of this sort, while amusing, is probably no more productive than looking for patterns in clouds. Instead, the author turned to one of SLAC's physicists for some more scientific help.

A curve-fitting program (*Kaleidagraph*, running on a Macintosh), which uses the least fit squares fitting technique, was applied to the data. Several curves were identified that produced a good fit. Removal of five Christmas/New Years low points that were clearly not part of the overall trend fine-tuned the parameters of the curves a bit, producing the curves which are shown in Figure 2 along with their correlation coefficients (r). The equations for these two curves are

$$a = 49400 \times e^{0.0099w}$$

and

$$a = 49000 + 460w + 4.0w^2$$

where w is the number of the week relative to December 24, 1990.

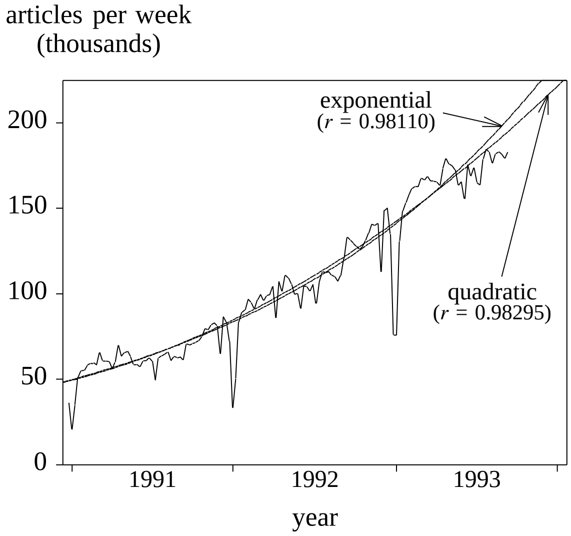

Figure 2: Exponential and quadratic curves fitted to actual number of articles per week minus five holiday low points

The adage that anything can be proven with statistics was reinforced when a least squares cubic fit was found that also seems to correlate well with the data. As seen in Figure 3, the long-rumored death of the net will occur on February 14, 1997 – no doubt a happy Valentine's Day for Usenet widows!

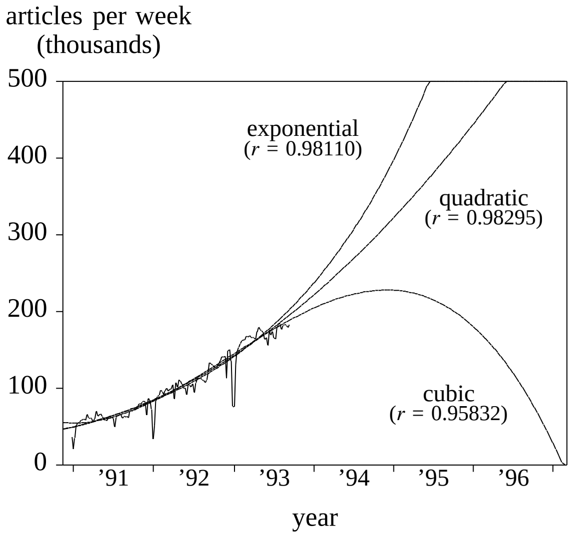

Figure 3: Death of the net predicted

There are several lessons here. One is that the sample is nowhere near large enough, and thus one should not depend on the resulting curves to accurately forecast growth too far in the future. Another is that good judgement and plain common sense needs to be applied to the interpretation of the results to avoid nonsensical predictions.

With this in mind, the exponential curve was selected for further use as being the closest to intuition. It's the most conservative choice in terms of disaster avoidance, as it forecasts the greatest disk needs. It is also remarkably close to the estimate of 67% annual growth, which a year ago had been estimated to be 65% and had predicted the past year's growth fairly accurately. In the interest of simplicity it was decided to just use the 67% growth equation

$$a = 50000 \times 1.67^{year-1991}$$

where fractional portions of a year are relative to January 1.

It's worth noting the rather large fluctuations in traffic levels. Because of this it isn't worth investing *too* much effort in precision when trying to forecast disk requirements – ballpark figures plus some extra space to accommodate peaks are sufficient to address the problem of knowing how much hardware will be necessary for the next year or so.

Size of Articles

The weekly reports generated by SLAC's netnews server don't include data on article size, and the only other local historical data was a single sample taken on March 30, 1992. There's little reason to expect much change, however, so data from several current sources were compared to establish a reasonable estimate of the current average size. This was then compared to the 1992 data, confirming the belief that article size is relatively stable over time.

First, `/usr/spool/news` on the SLAC server was scanned for all articles, ignoring non-article files such as those in the `out.going` directory, and being careful to note link counts to avoid multiply counting cross-posted articles. The result of this survey is shown in the first line of Figure 4. This result is biased because all groups aren't expired at the same rate. In particular, groups with very large articles such as those in the *alt.binaries.pictures* groups are expired much more quickly than other groups.

To compensate for this bias, several smaller scans were done of just the groups with non-default expiration periods, also shown in figure 4. All of these scans were done about twelve hours after the nightly *expire* run so the data represents a period equal to the expiration period plus one-half. Several SLAC hierarchies which are kept forever were ignored, since they don't apply to other sites. A weighted average was computed for the remaining data; this worked out to an average of 2,635.7 bytes per article.

For comparison, data from several sites were then harvested from *news.lists* for the August, 1993 timeframe. [8, 9] For the two weeks ending August 8, the average size of an article passing through uunet was 2641.5 bytes, while for the week ending August 7, the same metric for ira.uka.de was 2562.5 bytes, a difference of about 3%. For the six week period ending September 8, the average article size through uunet was 2586.2 bytes. These values agree fairly nicely with the numbers from the SLAC netnews server.

For the month of August, roughly the same period as the six week uunet data, [10] produced an average article size of only 2295.4 bytes. This is based on a survey of articles on disk, however, and thus is subject to the same expiration biases as the raw SLAC data, with which it compares quite closely.

The March 30, 1992 survey of SLAC's `/usr/spool/news` was done in the same manner as the current one, though no attempt was made to factor out expiration bias, since at that time all groups were being expired at nearly the same rate so expiration bias was minimal. The average article size at that time was 2,615.3 bytes, within the range of current sizes.

newsgroups	inodes	references	article size: total (KB)	article size: average	expiration
all articles	424,617	497,392	948,238.8	2,286.8	—
slac,bes,hepnet	7,482	8,049	36,371.0	4,977.8	never
alt.binaries.pictures	144	156	5,743.4	40,841.8	0
other *alt.binaries*	128	134	1,428.1	11,424.8	7
control	2,665	2,665	1,279.9	491.8	3
all others	414,198	486,388	903,416.4	2,233.5	17
weighted (excluding *slac,bes,hepnet*)				2,635.7	

Figure 4: Survey of SLAC's article spool area: number of inodes and references to those inodes, total and average file size, and expiration periods for specific groups

To produce a reasonable working estimate for average article size, the average was taken of the current SLAC value (2,635.7) and the six-week uunet value (2,586.2). This gives an average article size of 2,611 bytes.

Effect of File System Block Size

Since each article is relatively small and is stored in a separate file, the block size (or fragment size, where relevant) of the file system is important. Figure 5 shows the article sizes from the SLAC netnews server along with the space needed given various blocking factors. Running netnews on a machine with a large block size and no fragments, such as an RS/6000 running AIX, will waste more than half of the disk! Fortunately, many systems use a variant of the BSD fast filesystem, but care must still be taken to configure the filesystem with 512 byte fragments to make the best use of the space. Little effort has been spent on determining this inflation rate with great precision – assuming 512 byte fragments and rounding up to 12% gives a fairly conservative estimate.

block	average	total (KB)	inflation
1	2,286.8	948,238.8	0.0%
512	2,543.4	1,054,663.5	11.2%
1024	2,786.2	1,155,322.5	21.8%
4096	4,866.6	2,018,018.1	112.8%

Figure 5: Effect of blocking factor on article storage

Directories, Inodes, etc.

Given the large number of files, several other filesystem features are also worth mentioning. With nearly 500,000 directory entries for the articles, directories themselves consume a non-trivial amount of disk space. Filenames of articles are simply sequence numbers and thus are normally quite short. Until a group hits 10,000 articles only 12 bytes will be required with a BSD-style directory. Cross-posts require additional directory entries, adding about 17% based on the numbers in Figure 4. Figuring 16 bytes per article for directory entries accommodates both cross-posts and very active newsgroups.

The inodes themselves also take up considerable space, but they are pre-reserved, so as long as one looks at usable space after *newfs* is done, inodes need not be considered. Since most articles are small, few if any indirect blocks will be needed.

The Bottom Line for Articles

Putting all of this together, the space required for the articles can be described as a fairly simple function of *time* and the *expiration* rate:

$$\begin{aligned} bytes &= 50000 \times 1.67^{year-1991} / 7 \\ &\quad \times (2611 \times 1.12 + 16) \\ &\quad \times (E_{articles} + 1) \end{aligned}$$

One is added to the expiration rate assuming *expire* is run daily – an article arriving shortly after the time *expire* is run will get an extra day. Folding all the constants together we get

$$20{,}510 \text{ KB} \times 1.67^{year-1991} \times (E_{articles} + 1)$$

To put all this into perspective, a site expiring news after ten days will need about 1.67 gigabytes of usable storage just for the articles by the end of 1994.

The History File

The history file is composed of three tab-separated fields, containing the article ID, times relevant to the expiration process, and a list of where the article appears. For example:

```
<42@hh.uucp> 752520600~- news.answers/42
```

An article ID is composed of a host-specific part and the hostname. B News used a simple sequence number as the host-specific part, but most other news systems use more complicated and thus longer values. Today, most news systems have been converted so the size of this portion of the article ID is probably fairly constant. Likewise, the conversion to domain names caused an increase in article ID length, but again this probably has little additional impact today. Thus, a simple survey of SLAC's history file was done to establish an average size for the article ID. Figure 6 shows the breakdown by size; the average over 823,810 entries was 33.9 bytes.

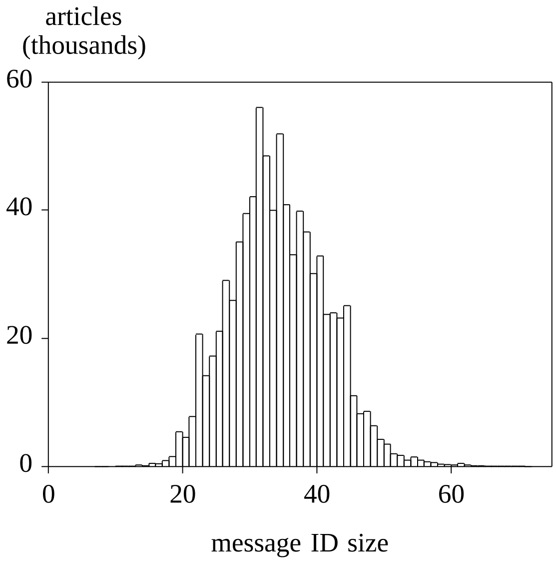

Figure 6: Article IDs by size (823,810 samples)

The second field is composed of two subfields separated by a tilde. The first part is the date and time the article was received by the local system in the normal UNIX form, i.e., the number of seconds since January 1, 1970 at midnight GMT. Normally, if the applicable expiration period is *n* days, *expire* will delete the article *n* days after it was received. If the article has an explicit expiration date, however, that date and time is stored in the second portion of the history entry's second field, and *expire* will try to use that as the expiration date for the article. (Explicit expirations may be overridden if they are too far in the future.) Few articles have explicit expiration dates, so this field of a history entry will usually be eleven bytes long, at least until September 9, 2001 (GMT).

The third field is a space-separated list enumerating the newsgroups in which the article appears, along with the article sequence number in each of those groups. This field and its leading tab will not be present if the article has expired. The number of entries with more than one subfield is just the cross-posting rate, which has already been estimated to be 17%. The size of each newsgroup/sequence number pair varies only slowly; it was determined by yet another survey of the SLAC history file to have an average size of 22.6 bytes.

The history expiration rate determines how long entries are kept in the history file. This is usually longer than the article expiration rate, so the oldest entries will lack the third field. If specified as being shorter than the article expiration rate, the article rate is used. (This case is not considered in the following calculations.) As with articles, one must be added to these rates to account for an extra day's worth of data, assuming *expire* is run once per day.

In addition to the history file itself there is also the index. Using *dbz* this is stored in `history.pag` (and `history.dir`, though that's tiny), and seems to stay around 10% of the size of the history file itself. Thus, at least for *dbz*, another 10% should be added to the size of the history file itself to allow space for this index.

Finally, when *expire* runs it creates a new history file and index, deleting the old ones only after installing the new ones. Thus the peak space requirement for the history file must account for two complete files and indexes.

Combining all of these factors, the overall space requirement for the history information can be stated as a function of the two expiration rates and time:

$$\begin{aligned} bytes = {} & 50000 \times 1.67^{year-1991} / 7 \\ & \times \left((E_{history}+1) \times (33.9 + 1 + 11) + (E_{articles}+1) \times (1 + 22.6) \times 1.17 \right) \\ & \times 1.10 \times 2 \end{aligned}$$

Log Files

The size of the log file depends upon various local configuration issues, including how many other sites a given article is fed to and how many duplicate articles are received. (With NNTP, duplicates should be zero or very nearly so.) A log at SLAC from a relatively busy mid-week day in early September contained 35,496 lines averaging 130.6 bytes long, a total of 4.4 megabytes. The number of lines can be expected to grow with the number of articles.

The logs compress quite well, often 3:1 or better, so if old logs are kept for a few days, compressing them will produce significant savings on space.

Overview and/or Thread Databases

Currently, SLAC is still using *trn* and its threading mechanism, which has used a bit less than 5% of the space required for articles, consistent with the author's projections. [6] This hasn't been studied in much detail since we plan on moving to *NOV*, for which the suggested 10% will be used as an initial estimate. [7]

Example

The following example demonstrates how these equations may be used to forecast the disk requirements for a netnews server through the end of 1994. The value for *year* is set to 1995, i.e., January 1, 1995. Articles in all newsgroups will be kept for 17 days, long enough for someone to go away for two weeks and not miss any articles, while history information will be kept for 30 days. The filesystems have a fragment size of 512 bytes. Article space is thus

$$\begin{aligned} & 50000 \times 1.67^4 / 7 \\ & \times (2611 \times 1.12 + 16) \\ & \times (17+1) \end{aligned}$$

or 2,804 megabytes. The space reserved by the BSD filesystem adds another 10% to this, bringing the total to 3,085 MB of usable disk space, well beyond the 2 GB limit of most BSD filesystems. The most expedient solution is probably a pair of 2 GB disks, using symlinks to split the articles over the two filesystems. This implies some effort at balancing newsgroups by volume, and hoping the symlink support really does work.

The history information will require

$$\begin{aligned} & 50000 \times 1.67^4 / 7 \\ & \times \left((30+1) \times (33.9 + 1 + 11) + (17+1) \times (1 + 22.6) \times 1.17 \right) \\ & \times 1.10 \times 2 \end{aligned}$$

or 224 megabytes.

This directory (`/usr/lib/news`) will also contain the logs. Assuming SLAC's log file is representative, and another 16 months will nearly double the size of the log, a daily log will need about 8.8 MB. Seven days of old logs, compressed to about one-third of their original size, adds another 20.5 MB. Adding a few megabytes for other files and the 10% reserved by the filesystem means a partition of about 300 MB is needed.

The *NOV* system will be used for threading; the 10% estimate implies 280 MB for this purpose. Leaving some room for error, and the filesystem's 10%, a 400 MB partition would probably be prudent.

Partial Feeds and Other Adjustments

The equations derived in this paper are based on carrying a full Usenet feed along with a number of other common hierarchies, e.g. *alt* and *gnu*. A uniform expiration rate across all newsgroups is also assumed. If these conditions are not true, some tuning will need to be done. A good source for raw data is the collection of monthly Usenet reports posted by Brian Reid, for example see [10].

Sites carrying a partial feed can identify major groups or hierarchies which they do not carry and, using the monthly Usenet reports, determine what fraction of the total articles are being carried, multiplying the computed article rate by this fraction. Depending on the groups, the average article size may also require some tuning – the data for this is also in the monthly reports.

Tuning for varying expiration rates can also be done using these monthly reports. This is essentially the reverse of the weighting done in Figure 4.

Conclusions

Historical data on system activity is quite valuable, as it can be analyzed to assist in planning for future system growth. One must be wary of trying to read too much into statistical results, however, especially on rather small samples. Common sense is needed to recognize unreasonable results.

With regard to Usenet, growth of the net can be predicted reasonably well, at least over the span of a year or two. With some analysis of how this growth influences the growth of various databases, disk requirements for a Usenet server can be forecast sufficiently to permit acquisition of new hardware or adjustment of administrative controls before problems arise.

References

1. Tim O'Reilly and Grace Todino, *Managing UUCP and Usenet, 10th ed.,* O'Reilly & Associates, Inc., Sebastopol, California, 1992.
2. Anatole Olczak, *Netnews Reference Manual, 2nd ed.,* ASP, Inc., San Jose, California, 1989.
3. Geoff Collyer and Henry Spencer, *Installing and Operating ''C News'' Network News Software*, February 1993.
4. Tom Limoncelli, ed., ''INN FAQ Part 1/3: General Information, how to compile, how to operate,'' *news.software.nntp (Usenet newsgroup)*, August 12, 1993.
5. Wayne Davison, mthreads(8) man page for trn 2.2.
6. Wayne Davison, README for trn 2.2.
7. Rob Robertson, ed., ''FAQ: Overview database/NOV General Information,'' *news.software.readers (Usenet newsgroup)*, September 8, 1993.
8. ''Total traffic through uunet for the last 2 weeks,'' *news.lists (Usenet newsgroup)*, biweekly report.
9. Matthias Urlichs, ''NewsStats, article counts,'' *news.lists (Usenet newsgroup)*, August 7, 1993.
10. Brian Reid, ''USENET READERSHIP SUMMARY REPORT FOR AUG 93,'' *news.lists (Usenet newsgroup)*, September 9, 1993.

Acknowledgements

Special thanks to Tom Glanzman for doing the curve fitting and for teaching me a bit about statistics. Other people who helped in various ways include Mark Barnett, George Berg, Chuck Boeheim, Les Cottrell, Krissie Griffiths, Don Pelton, Mike Sullenberger, Bebo White, Lois White, and others whose contribution is not diminished by my failure to remember them. My gratitude goes to all of them. Thanks, too, to Alexander for his patience – and for *not* finding another skunk.

Author Information

Karl Swartz is a member of the Server Systems Group within SLAC Computing Services at the Stanford Linear Accelerator Center, where he is the resident UNIX guru. Prior to joining SLAC, he worked at the Los Alamos National Laboratory on computer security and nuclear materials accounting, and in Pittsburgh at Formtek, a start-up that is now a subsidiary of Lockheed, on vector and raster CAD systems. He attended the University of Oregon where he studied computer science and economics. Karl instructs at high-speed driving schools and enjoys good food and good beer (though not while driving) and hikes on the beach with his Golden Retriever, Alexander. Reach him electronically at kls@chicago.com or kls@unixhub.slac.stanford.edu.