

On Performance of Caching Proxies

Alex Rousskov and Valery Soloviev
Computer Science Department
North Dakota State University
Fargo, ND 58105-5164
{rousskov,soloviev}@plains.NoDak.edu

1. Introduction

Web caching proxies are commonly used for handling Web traffic. They are designed to improve the request response time and reduced network bandwidth requirements. However, little is known about performance of real proxies.

We present a performance analysis of Squid, a state-of-the-art caching proxy. Squid is the most popular caching proxy within public domain [2]. Our instrumented version of Squid [1] measures per request network and disk activities. These detailed profiling allows for in-depth studying of major proxy components.

Our analysis is unique because it covers a variety of hardware, operating systems, caching hierarchy levels, and workloads rather than concentrating on a single proxy server. Data from all proxies were collected over a relatively recent and short time interval, September – October 1998, for increasing the relevance of comparison.

2. Methodology and Framework

Performance data was collected using a *patched* version of Squid caching proxy [1]. The patch enabled Squid to log detailed *per request* measurements of network and disk I/O activities. The instrumented version of Squid was run on 7 proxies. The proxies represented all levels of caching hierarchy starting with leaf university proxies and ending with the root proxy of an international hierarchy. We collected 18 days worth of logs. Off-line, we ran several analyzing scripts to extract useful statistics. *Percentiles* and distributions were used whenever possible. Medians were used as an estimate of an average value.

3. Results

The entire set of experiments is publicly available on the Web [1]. Here we present only a short and incomplete

summary of our observations. The reader is referred to [1] for definitions, graphs, explanations, related work, and discussion.

Traffic Patterns and Aggregate Performance

More than 50% of all network transfers are smaller than 1 KB. About half of the disk transfers (swap requests) are smaller than 2 KB. Median size of a cached file is about 3 KB. On average, proxies swap-out larger objects than they swap-in.

We have detected two types of proxy load. Root proxies experience relatively small variations in load. Other proxies have bell-shaped curves with highest load during the day and lowest load at night leaving resources for an optimization such as prefetching.

The ratio of *concurrent* misses to hits in the system may be as high as 7:1, which is much higher than the miss/hit ratio in the traffic. Misses are much slower than hits and that makes misses the major consumer of proxy resources.

Even a small increase in traffic intensity may lead to severe performance degradation. A modest 10-20% increase in incoming rate often led to at least 100% increase in the number of concurrent request on the root proxy.

Response Time and Hits

Decreasing the response time is one of major functions of a caching proxy. Our results show that average response time of a hit may be five times smaller than of a miss. The average improvement depends on the Document Hit Ratio (DHR) and varies from 28% on root proxies to 60% on leaves.

Savings in response time depend on the document size. Most savings come from hits smaller than the size of TCP socket output buffer. Large hits do not improve response time much. However, they are responsible for higher Byte Hit Ratio (BHR). Consequently, we have a fundamental tradeoff between improving response time (by caching smaller objects) and saving more bandwidth (by caching larger objects).

Interestingly, neither DHR nor BHR depend on the traffic intensity. Despite increase in the number of clients during peak load, hit ratios remain stable.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGMETRICS '98 Madison, WI USA
© 1998 ACM 0-89791-982-3/98/0006...\$5.00

Disk and IMS hits are responsible for more than 90% of all hits. Average share of IMS hits is large and increases with caching hierarchy level: about 14% on leaf, 23% on top level, and 28% on root proxy.

The percentage of negative hits is small. However, invalid requests (their results are negatively cached) may have very large response times. Thus, additional experiments are needed to show if response time reduction for invalid requests is worth caching them.

Network

It takes longer for a hit to connect to the original server (to send an IMS request) than for a miss. This effect is especially visible on leaf proxies.

On average, origin server replies for hits are 40% faster than misses on root proxy, 60% faster on top-level, and 70% on leaf. The IMS reply time is actually the same as miss reply time for very small documents. This indicates that network congestion dominates in server reply time.

It may seem that client connect time cannot depend on the result of the request (hit or miss) because the result is not known during connect phase. However, on most proxies, it takes longer for hits to connect. The root proxy has an opposite pattern.

Client connect time depends on time of day. This dependency is probably caused by inbound network congestion.

Disk Storage Subsystem

The actual number of swap-in and swap-out disk requests is determined by traffic intensity, DHR, and caching policy. The number of swap-in requests usually dominate on leaf proxies. The opposite is true for top level caches. On the root proxy, swap-in requests dominate half of the day! Moreover, the Swap-In/Swap-Out ratio significantly changes with time on all participating proxies. These changes make performance tuning harder because *static* optimization may not work.

The number of concurrent swap requests increases sharply during peak load. The increase is not proportional to the incoming swap requests rate. This is a direct effect of large queuing time of swap requests. We also observed drastic increases in response time that correspond to a higher number of concurrent swap requests. These peaks sharply increase the total response time for hits.

Disk subsystem utilization (measured as percent of time there is at least one swap request pending) often reaches 90% on root and top level proxies while leaf proxies have at most 40% utilization.

On most participating proxies, swap-out requests were somewhat faster than swap-in requests in spite of a larger average size.

Proxy Response Time Components

The relative and absolute contribution of each request processing stage towards total delay were analyzed to identify performance bottlenecks. The origin server reply time dominates in miss response time (50-60%). However, its relative contribution *decreases* during peak

loads. On root proxy, server reply time component becomes even less important than proxy connect stage (which sharply increases its relative value during peak loads). It often takes longer to send a small request to an origin server than to receive a potentially large reply.

The contribution of client connect stage is usually 10% or less, but it doubles during peak load. Interestingly, proxy reply time is responsible for a constant portion of the total delay regardless of the load.

Our analysis implies that connect times are most susceptible to traffic intensity. Using *persistent connections* (HTTP 1.1) is a promising way to reduce the impact of load on response time.

Disk delays contribute about 30% towards total hit response time. Thus, disk storage subsystem may be considered a potential bottleneck. Network performance is often beyond control of a caching proxy. On the contrary, disk I/O performance is isolated from external factors which makes it a promising target for an optimization.

4. Conclusions

We have studied performance of several installments of Squid caching proxy, using profiling of *per-request* network and disk activities. Many common performance patterns were detected across various proxy environments. By careful classifying requests, we were able to identify and quantify the degradation of network and disk storage subsystems during high load periods. By studying proxies from different levels of caching hierarchy, we analyzed the influence of cooperative caching on proxies, i.e. how a level of caching hierarchy affected proxy performance. We also demonstrated that various classes of requests had different impact on proxy resources, and optimization decisions must take this into account. Our data and analysis are essential in understanding, modeling, and enhancing performance of a proxy server.

5. Acknowledgments

We are very thankful to all cache admins who managed to squeeze time to run our experiments on their proxies and discuss the results: Henny Bekker (Utrecht University, the Netherlands), Edwin Culp (MexCom, Mexico), Brian Denehy (ADFA, Australia), Lars Slettjord (University of Tromsø, Norway), Ton Verschuren (SURFNet, the Netherlands), and Duane Wessels (NLNLR, the USA).

This work was supported by grants from NSF (OSR-95-53368 and RIA IRI-94-09845), ND EPSCoR, and NDSU PPRC.

References

- [1] Performance profiling patch and complete paper: <http://www.cs.ndsu.nodak.edu/~rousskov/research/cache/squid/profiling/>
- [2] Squid Internet Object Cache: <http://squid.nlnr.net/>