Load and Traffic Balancing in Large Scale Cache Meshes

Christian Grimm, Helmut Pralle, Jens-S. Vöckler[1]

University of Hanover
Institute for Computer Networks and Distributed Systems
Schloßwender Straße 5
D-30159 Hannover

`{grimm,pralle,voeckler}@rvs.uni-hannover.de`

Abstract

The current document summarizes the experiences obtained during the maintaince and operations of the cache hierarchy within the German broadband research network. Installed at central nodes, the cache service is an integral part of the backbone infrastructure. Ten distributed cache servers are the building blocks of a large scale top-level cache hierarchy.

During the last 13 months of operation the cache service was subjected to different mesh designs and conceptions. Various approaches were aimed at improvements in the load and traffic balance among the caches. At the same time, the benefit of the mesh as whole was to be increased, as well. Due to the fact that the caches are used in a production-like environment, few of the variation manifested themselves in practical configurations.

Maintainers of similar cache meshes may benefit from our ideas and experiences related in this document for their own conceptual design, configuration, and hardware selection. The reader might want to investigate further publications describing the design of single web caching systems [1], [2], [7].

Keywords: design hierarchical cache mesh DFN B-WiN

Table of Contents

1 Introduction

This section starts with a brief introduction into the reasons for employing a cache. Afterwards the technical environment and administrational coordination in the current project are detailed.

1.1 The World-Wide Web, Caching and Cache Meshes

The paradigm of the World-Wide Web demands ever increasing resources from all networks participating in the Internet. The enormous growth of traffic is ascribed to the inflational use of multimedia content, although we are only at the start of an evolutionary process. Further stress is put on the networks by the increasing number of users browsing the multimedia content day by day.

More bandwith is the ultimate solution and eventually will render caching futile. But until we have enough capacity to satisfy every users' needs, caches are the most viable solution to counter network congestion. Furthermore, even sufficient bandwidth cannot overcome latency due to a finite speed of light. Therefore there are two good reasons to use a cache nowadays:

1. Webcaches reduce demands on the wide-area Internet capacity, thus cutting down costs of the providers and in the end making internet technology more affordable.
2. Webcaches reduce latency. The fundamental idea behind caching is to reduce the demands on a resource **and** the respective latency time by storing copies closer to the consumer.

1.2 The German Research Network

The German broadband research network (B-WiN) was launched in March 1996. It consists of central nodes in ten location distributed across Germany. All nodes participate in a ring structure employing 34 Mbps ATM technology. A few additional cross connections of the same capacity add a partial mesh structure. The capacity of some selected links was updated to 155 Mbps during the last months. The interconnections with other networks - foremost into the US - were significantly improved by dedicated high capacity links. At the writing of this document the B-WiN includes

- a dedicated 90 Mbps link to the US internet,
- an own point of presence in Washington D.C.,
- a gateway to TEN-34 at 45 Mbps and
- access to other national research networks, e.g. in Russia or to the science network of the People's Republic of China.

A map of the network is given in Figure 1.

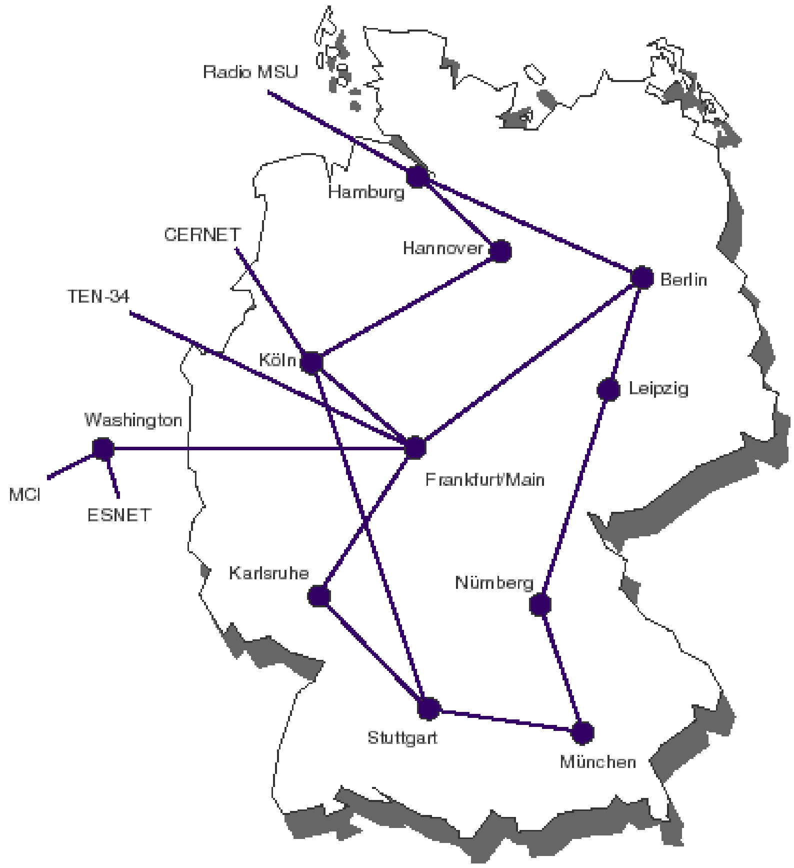

Figure 1: Map of German Reserach Network

1.3 Caching in B-WiN

In the beginning of 1996, the first German cache mesh was set up between universities of Bochum, Bonn and Frankfurt. During the following weeks, a large number of organisations joined in the expanding mesh.

Unfortunately, the mesh grew without almost any coordination, everyone was connected to everybody else. Soon after the launch, some of the larger caches were flooded by requests from other caches. As a result the flooded caches weren't even capable of operating satisfactorily for their own local clientèle. Since the own user group takes precedence over remote user groups, links in the mesh were bound to be torn down again.

Another atrocious effect was the formation of so-called cache loops by means of incorrectly configured parent relationships, e.g. cache A requests a document from B, B requests it from C, and C finally tries to obtain the document by asking A. A lesson learned was that parental relationships, visualized as edges of a directed graph, must not build circles, but have to form trees. Hierarchical depencies are inherent in this kind of structure.

1.4 The Project

To counter the effects of uncoordinated growth, overloaded links, and congested networks the project *Conception of a Caching Infrastructure in the German Research Network* was initiated in August 1996 by DFN Association. The main goals of the project were to

- coordinate a caching infrastructure among sites connected to B-WiN,
- clarify the potential of a top level cache hierarchy in B-WiN, and
- establish a mesh of ten top level cache servers as an ingetral part of the infrastructure in the backbone of B-WiN (the DFN cache service).

Ten SUN Ultra-2 servers were installed at the central nodes of B-WiN and connected to their respective Cisco 7513 router. Besides collecting router statistics and feeding news traffic, today the servers act as a cooperating cache mesh at the top level of a more complex hierarchy.

2 The Initial Setup

The DFN cache servers were installed in January 1997. Soon afterwards they were ready for operation and announced to the interested user group of peer cache administrators. Just a fortnight later, 45 caches were already using the DFN caches.

In order to enable the interested clientèle to test how they would benefit from the use of the DFN caches, no access controls were activated. If local cache administrators decided for a long term use of the DFN caches, they were asked to register their caches via email. At the time of this writing, over 120 local caches are registered. From all users of the B-WiN caches as seen in the cache logfiles, the amount of registered users exceed 80 %.

2.1 Conception

The initial configuration started out by distributing the various toplevel domains evenly all over the ten DFN Caches. Little consideration was given to redundancy with the exception of Cologne and Frankfurt. The .com domain was put on those particular hosts, because the transatlantic link arrives in Frankfurt. The distribution of toplevel cache domains is shown in table 1.

DFN Cache	Toplevel Domain
Berlin	.cz .hu .pl .ru .si .sk .su .tr
Cologne	.com
Frankfurt	.com
Hamburg	.cn .id .in .jp .kr .my .sg .th .tw .vn
Hanover	.dk .fi .gr .is .no .se .sp .pt
Karlsruhe	.at .be .lu .ch .fr .it .nl
Leipzig	.de
Munich	.net
Nuremberg	.edu .gov .int .mil .org .us
Stuttgart	.au .br .ca ie .mx .nz .uk .za

Table 1: Initial distribution of toplevel domains in DFN cache mesh

There were two levels of hierarchy within the DFN cache mesh. Each cache acted as sibling for a domain it was not responsible for. Thus the statistical chance of asking the correct cache for a given domain was 1:9 for all domains except .com. All the caches acting as sibling also saved of copy of the delivered object. Therefore the chance rose for obtaining popular objects without climbing up the hierarchy, though the particular drawbacks of this approach are detailed further on.

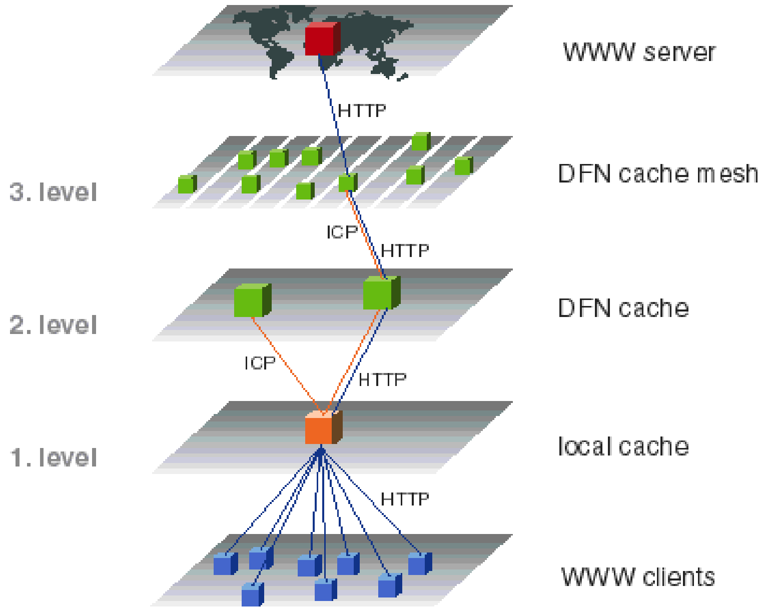

Figure 2: Initial cache hierarchy

2.2 Design Reasons

The reasons for choosing the particular design shown in figure 2 were few. Clearly, the design was a first approach. At the time, there were no other experiences of large scale cache meshes managed by a single institution available. As a result the approach from National Laboratory for Applied Network Research (NLANR) [5] was adopted.

Among the few advantages of the chosen approach, the structure is mutually fail-safe. A DFN cache on the second level can detect a dead parent on the third level, thus bypassing it.

2.3 Problems and Drawbacks

The first approach suffered from several drawbacks. Some of the shortcomings depend on by others.

- The shown distribution of DFN caches did not support any reasonable load and traffic balancing. The .com and .de caches experienced a strainful load whereas the other caches were almost idle. The situation is aptly shown in figure 3. Note the discrepancy in the output from .com and .de caches to the remaining caches[2].
- With the exception of the .com caches, no other DFN cache had a backup cache in case of failure. A failure in Munich rendered all requests destined to .net hosts virtually unusable. Popular documents were distributed evenly over the remaining caches.

  In the scenario, if Munich went back online, popular documents had to be fetched once more. Munich has to contact the origin site, even though the documents might be stored away on some other DFN cache.

- Even though the .com caches existed twice, frequent failures of the DFN cache in Cologne cache caused the cache in Frankfurt to overload. If a situation becomes desperate, it will deteriorate exponentially.
- An enduser who was directly connected to a local caches experienced in the case of cache misses three levels of caches until the origin site was reached. Each level implied a timeout of two seconds, adding up to six seconds.
- Every DFN cache was caching documents from every toplevel domain due to the extra level on which they act as siblings. Therefore similar sets of only the most popular documents were bound to be found on all caches. Not so popular documents were expired by the LRU algorithm exceedingly fast, recycling the majority of the disk space several times a day.

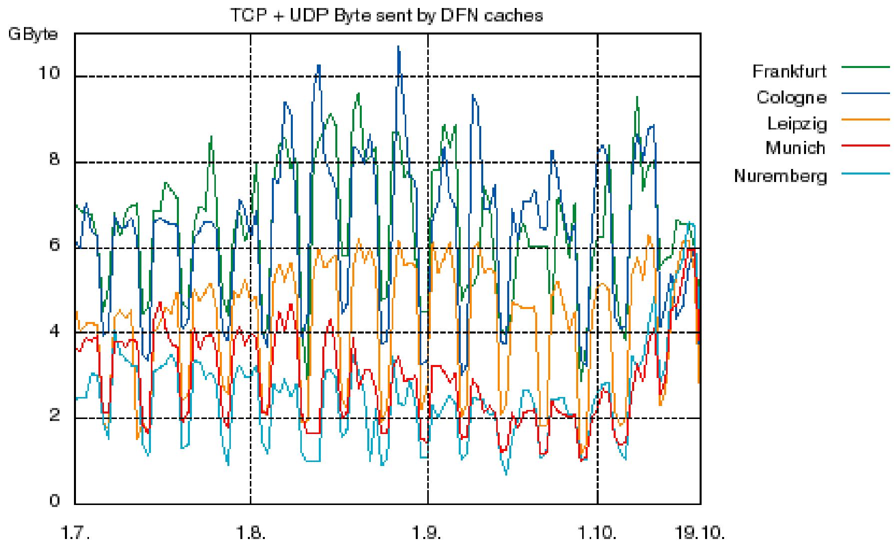

Figure 3: HTTP and ICP output of five DFN caches

3. The Redesign

The initial setup was not living up to the expectations of a cache. For obvious reasons a redesign was becoming more pressing each day. This chapter describes the research done to propose a well-designed concept. The proposal was accepted by peer reviewers without changes.

3.1 Real Life in WWW

In order to develope a sound concept for the redesign, the first step was to investigate the real dataflow within the mesh. The answers we were looking for derived from the question of what was *really* going on, and what do the users typically request.

Log file analysis was done with the help of the calamaris software ([4]) which provided the first usable numbers. Figure 4 shows the results from the local cache of the University of Hanover.

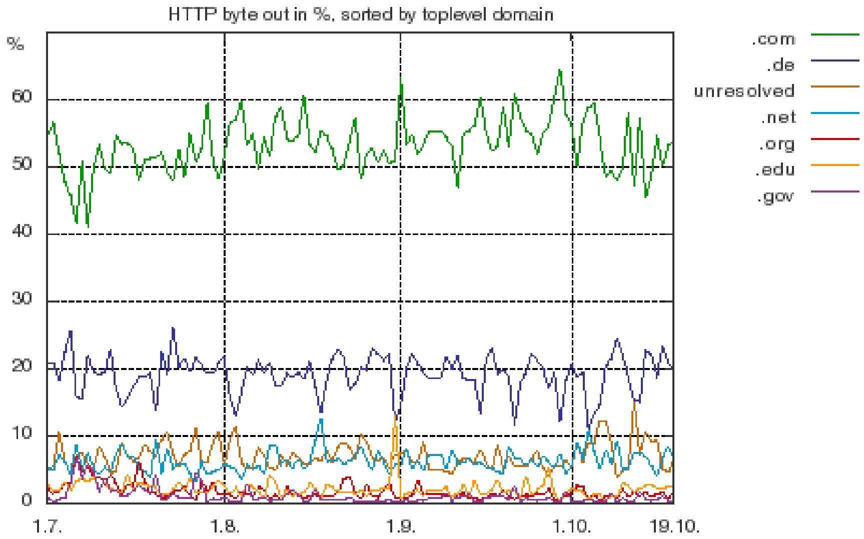

Figure 4: Bytes delivered by local cache, sorted by toplevel domain

The figure shows without doubt that more than 50 % of the overall web traffic was generated for objects in the .com domain. We can safely assume that over 90 % of all .com hosts can only be reached via the transatlantic link.

A less but still significant role are requests for .de objects. Usage statistics indicate that about 20 % was destined to hosts within the .de domain. At the time of the investigation this number implied that half of all objects had to be fetched either via the transatlantic or the European link. The leftover 30 % distribute over the remaining domains, including .net and .edu at various locations without regional focus.

3.2 The New Conception

The basic idea behind the new concept was to adopt the real life situation into the DFN cache mesh. Since 50 % of all traffic is destined to .com hosts, half of all DFN caches should handle .com traffic exclusively.

The open issue was how to handle the remaining 50 % of the B-WiN caches. After discussing the topic with other cache administrators we decided against a partitioning of the remainder. Rather we handled everything not destined for .com as one logical domain aptly named !.com. Therefore all caches were evenly partitioned between the two logical domains .com and !.com.

DFN Cache	Toplevel Domain
Berlin	.com
Cologne	!.com
Frankfurt	.com
Hamburg	!.com
Hanover	.com
Karlsruhe	!.com
Leipzig	!.com
Munich	!.com
Nuremberg	.com
Stuttgart	.com

Table 2: New distribution of toplevel domains in DFN cache mesh

A further benefit from the new scheme was the avoidance of parent relationships among the DFN caches. All caches within one group only maintain a sibling relationship to their neighbours. Thus the number of hops from an enduser to the origin site was reduced by one level. Currently, no interconnections between the two groups of caches exists. Figure 5 shows the different levels of the new cache hierarchy in B-WiN.

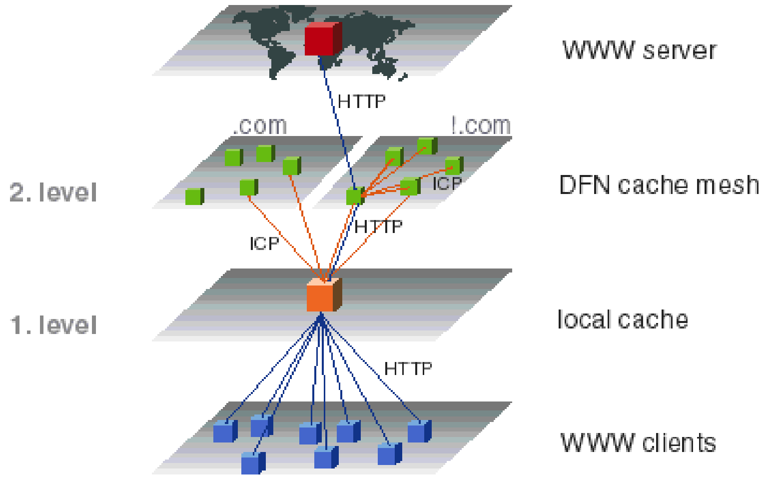

Figure 5: Reduced cache hierarchy in the new conception

3.3 The Benefit

Soon after the new configuration was started, the first effects confirming our decision could be observed.

- The cache hierarchy traversed from an end user to the origin site is reduced by one level, thus improving on the latency. Usually, the latency induced by host processing is larger by several orders of magnitude than the latency of a link.

- All caches at the next lower hierarchical levels have to configure **two** DFN caches from each group. If one cache fails, the other one from the group is still providing service. The likelihood of both caches failing is rather low; if both fail the complete network stands a good chance of being down, too.
- Due to the removal of one level in the hierarchy, the DFN caches only see peers in their own group, and no cache from the unrelated group. With the removal of the previous parental relationship, only hits from another cache are transferred via HTTP, or misses from the origin site. The amount of ICP traffic increased due to the higher number of caches responsible for a domain. With the emphasis on inter cache communication, ICP generates only 10 % of the volume HTTP is using.
- Most significant was the drastic increase in the output of the caches from below 30 GB/day to almost 60 GB/day. We saw a likewise drastic decrease of the inter cache communication from almost 15 GB/day to less than 5 GB/day. Figure 6 shows the changes during the switch-over to the new concept in detail. The new design was implemented at the start of the 42nd week.

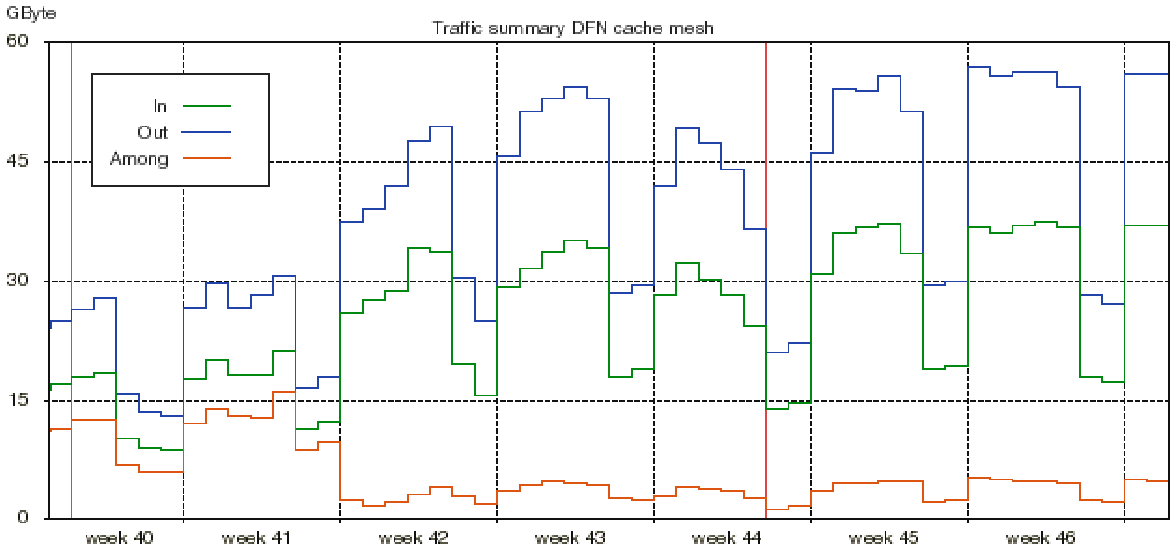

Figure 6: Traffic summary of DFN cache mesh

The increase in the input leads to the assumption that the previous scheme suffered latency from the additional layer in hierarchy. Also remember that the old scheme employed only three caches to handle over 70 % percent of the traffic, which is now distributed over *all* caches. The approximation towards a more homogenous output at an even higher level is illustrated in figure 7.

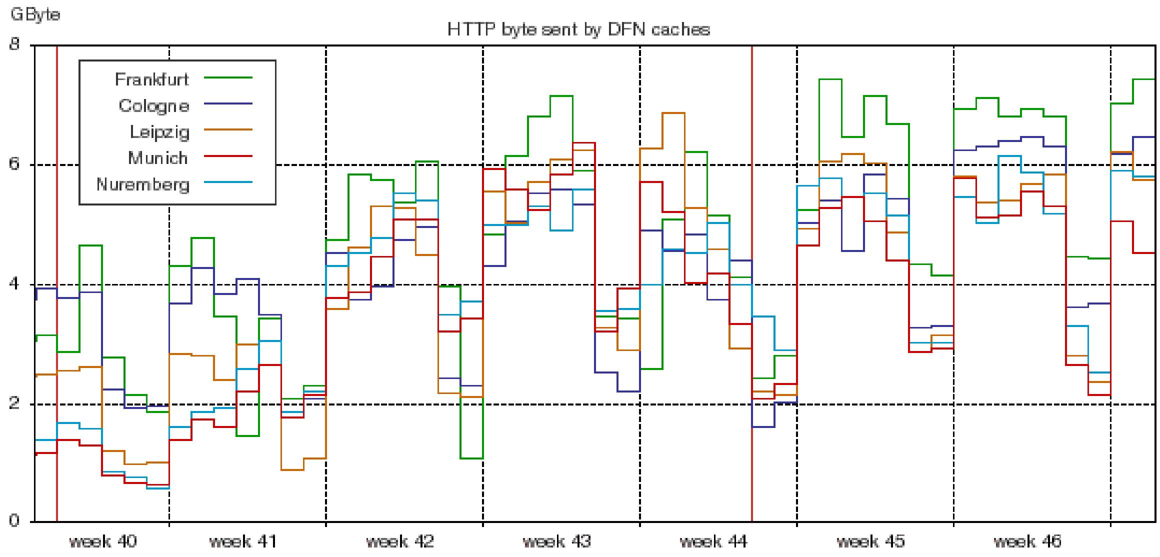

Figure 7: HTTP and ICP output of five DFN caches

3.4. Open issues

Even though the new design constitutes a major improvement, there are still a few shortcomings of the chosen approach.

- A significant drawback is the increased workload on all caches. Figure 8 illustrates the particular effect. At the moment we are at loss for an explanation due to the fact that our laboratory machines don't show such a strain even under a much heavier request load. A possible reason - not proven - could be the increase in ICP traffic.
- There do not exist any cross relations between the .com and !.com group. Thus the concept depends on the correct configuration of peer caches.
    - Even admins notoriously misconfigure their caches, thus defeating the concept.
    - The current design does not support cache software which cannot partition between the locigal domains, e.g. no support for the old CERN proxy.

  A reintroduction of interconnections between the groups results in the three hop situation again.

- If a cache gets a hit for a requested document on a peer cache, it will fetch and store the document. Therefore, after a finite time popular objects are stored on all caches within a group. We still have to be careful with our decision between the waste of disk capacity vs. the waste of B-WiN network capacity.

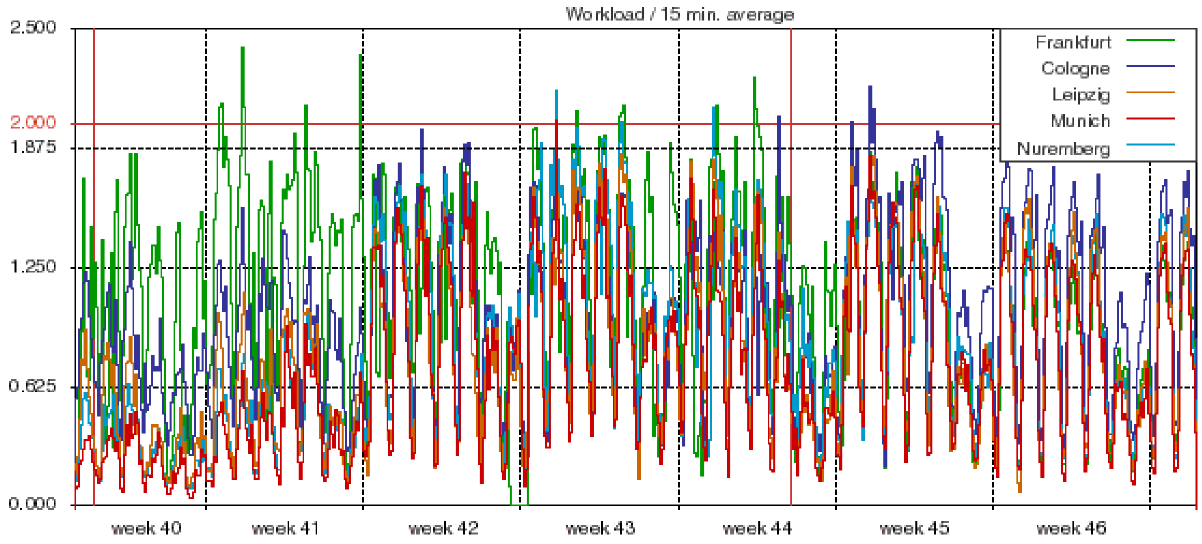

Figure 8: Workload on the DFN caches

Due to the increased efficiency of the design, as compared to the previous one, another drawback manifested itself. All caches lack main memory in particular. During the 44th week, some of the DFN caches started - among other contenders - *swapping* part of the cache process. Swapping drastically decreases the performance of a system. Unfortunately, plugging in more main memory was no option available to us.

The only way to get rid of this messy situation was to use the *no virtual memory (NOVM)* version of squid. The NOVM squid stores in-transient object directly on the disk. Furthermore, there does not exist a main memory cache of hot, that is popular, objects. Now, each request for an object has to be serviced from disk. This performance decrease is not as bad as swapping. Figure 6 indicates no decrease in the traffic summary.

4. Future Tasks

At the moment we see a more or less stable cache mesh in the B-WiN. The hardware won't scale any further, i.e. the current user group must not grow any more. We couldn't possibly handle more users, even if we would love to see all educational and research facilities using the DFN caches.

Additionally, router statistics show that majority of WWW traffic in the B-WiN bypasses the DFN caches. Apart from an ignorance of local institutions about the DFN caches, we must assume that institutions started to decide against the use of the DFN caches in favour of a direct connection to the origin site. This behaviour constitutes a further hint to the saturation of the service supplied by the DFN caches.

Calculations show that at peak times over 50 % of the link capacity between cache host and router connection is used by the cache. Any future demands cannot be satisfied by the current hardware. In order to attract new user to the DFN cache service, new scalable hardware has to be acquired, which surpasses all previously found bottlenecks. The more users are connected to the DFN caches, the better a hitrate we expect. Since the client institutions belong to the educational or research area, we can safely assume a more homogenous user group than a commercial internet provider might see.

The need for more powerful hardware is easily justified, if you compare the 60 GB/day through the DFN caches to the overall WWW traffic. Most of the 300 GB/day[3] WWW traffic on the transatlantic link is circumventing the caches. Easily much greater savings just on the transatlantic link can be imagined, but not with the current hardware.

Monitoring the caches revealed some stray users from foreign networks and endusers directly connecting to the DFN caches. Such attempts defeat their purpose. As a result, restrictive access controls have to be introduced. With the establishment of caching as a major service within the B-WiN, a policy-of-use must be developed, clearly stating the rights and responsibilities of each participating party - including well-configured peers and a peer maintained proxy-auto-configuration service for the endusers.

As physics shows, no object changes a plotted course without the influence of an exterior force. Thus a further method to enhance the interest in the DFN caches among the B-WiN clientèle is to employ traffic precedence in routers: WWW traffic to the DFN caches has a higher priority than other WWW traffic. Of course, only new hardware would be able to handle the demands of the clientèle.

There are still many areas of interest for future research. Some issues for investigation contain:

- inter cache communication between otherwise unconnected groups
- storing objects only once within a group (proxy_only option in squid)
- web cache routing based on AS numbers
- a completely unpartitioned cache modell

As the list of open issues for further research shows, there are still improvements of the current configuration to be expected. With the participation in the German exchange point DE-CIX in January 1998, almost all .de hosts can be reached at little cost in time and money. So far, this important change in network topology was not examined closely.

5. Conclusion

The steps described in the previous sections show the possibilities of improving performance and balancing through an intelligent conceptual design. As a guideline to the interested reader we suggest the following course of action:

- gather data from your caches

  It is all there! All you need to do is look into your logfiles.

- find out about the typical behaviour of your user group

  Every user group behaves differently and has different common interests. A cache should pay attention to the needs of its clientèle. Of course, the larger a user population grows, the more heterogenous the interests become. This is indeed a problem for the toplevel caches in a hierarchy.

- know your network topology and link specifications

  In order to plan, you have to know where it is possible to place a cache and what places would benefit most.

Even more important is some knowledge about the link specifications. Two examples:

  - Links with a high error rate might still be good for ICP (UDP) traffic. HTTP (TCP) traffic will more likely result in aborted transmission on such links.
  - Link capacity created from bundling lower capacity links won't improve your cache service, because each TCP connection is handled by the same low capacity flow. Thus the supremum of the datarate from and to your caches will never exceed the low capacity link datarate.

- distribute your caches according your findings

  Don't plan against your users.

- obey the principal of locality

  The basic idea behind caching is the concept of locality. If you are undecided about several possibilities, give priority to the one which employs a higher locality.

  Obviously there are many definitions possible for *locality*. For our purposes it is safe to assume that any object is local as soon as it reached the B-WiN.

- avoid experiments, plan!

  If experimenting with the structure, distribution, and configuration of your caches, try to predict the expected results **in advance!** That should help you to eliminate the need for over 90 % of all experiments. Your users will thank you.

6. References

[1] *Survey of caching requirements and specifications for prototype*, May 1997
http://www.cc.ruu.nl/~henny/desire/deliveries/del_41.html

[2] *Web caching architecture*, Mar. 1997
http://www.uninett.no/prosjekt/desire/arneberg/

[3] *Squid Internet Object Cache*, Feb. 1998
http://squid.nlanr.net/

[4] *Frutti de Mare, Tools for Squid*, Feb. 1998
http://www.detmold.netsurf.de/homepages/cord/tools/squid/Welcome.html.en

[5] *A Distributed Testbed for National Information Provisioning*, Feb. 1998
http://ircache.nlanr.net/

[6] *Building a Web Caching System - Architectural Considerations*, May 1997
http://www.terena.nl/conf/jenc8/papers/121.ps

[7] *Configuring Hierarchical Squid Caches*, Feb. 1998
http://squid.nlanr.net/Squid/Hierarchy-Tutorial/

[1] This work is supported by the German Academic Network Organisation (DFN-Verein) with funds from the Federal Ministry of Education, Science, Research and Technology (BMBF).

[2] The remaining five caches not shown in the picture are below the level of Nuremburg.

[3] The value of 300 GB/day is a rough estimate from short-term router statistics. More reliable numbers will be available by the time of the conference.