

# **TO WHAT DEGREE WOULD BRIGHAM YOUNG UNIVERSITY-IDAHO BENEFIT FROM A WEB CACHE?**

by

Dallin C. Wilcox  
dcwilcox@gmail.com

# To What Degree Would Brigham Young University-Idaho Benefit from a Web Cache?

## Introduction

Brigham Young University-Idaho students and faculty alike have often had to wait excessively for web content to display. This situation can be time-consuming and becomes especially noticeable during presentations and demonstrations. At other times when trying to access resources last minute, students and faculty are unable to access the pertinent information in time. Individuals researching through web based tools and engines are forced to spend more time simply because they must wait on the browser to receive all of the information it is attempting to display.

Many people question the source of this problem and what steps have been taken to facilitate its resolution. Some speculate that the students in the dorms are using all the bandwidth, or come up with other even more foolish notions. However, without specialized knowledge or access to accurate information, most simply push the problem from their minds. Perhaps some think that the campus paper, *The Scroll*, may one day publish a piece that at least explains the problem if not generate enough interest to fix it. There are several solutions available, many of which are not feasible due to high cost and minimal improvement.

This paper will examine one possible solution, the web cache, to determine the degree of benefit that it could provide for BYU-Idaho. The paper will explore the implementation of a web cache at BYU-Idaho from several angles. First, a foundation will be built by explaining how a web page is displayed in a normal setting and the differences introduced with the implementation of a web cache. Second, the paper will explore the technical benefits that can be experienced with the introduction of a web cache in more detail. Third, the feasibility of implementing a web cache at BYU-Idaho will be covered. Finally, the information covered will be applied specifically to the BYU-Idaho network and users on the BYU-Idaho campus.

## The Web Cache Process

A basic understanding of a web page and the web cache process is necessary to understand the benefits that it can provide. This paper will begin by examining how a web page is displayed and then show how that process is altered by the introduction of a web cache.

Generally when a user accesses web content on the Internet, they most likely use a



program called a browser. When a user types in an address in the URL space or clicks on a link many things are happening behind the scenes to allow a web page to be displayed. A web page usually consists of several different elements known as objects. Objects include text, images, videos, embedded applications and other content (see Figure 1).

Figure 1

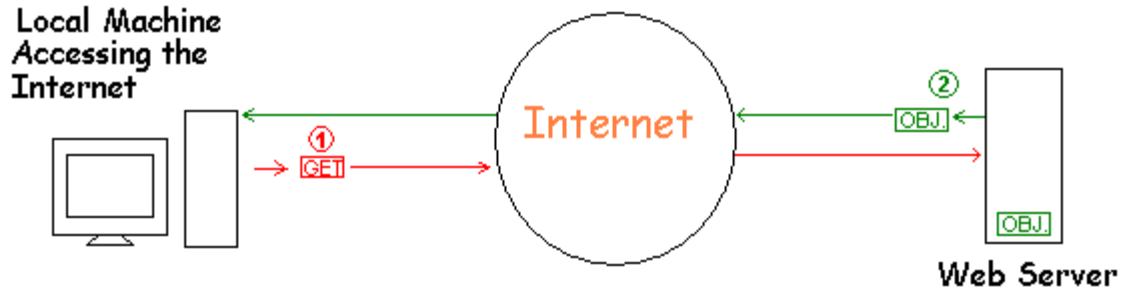


Figure 2

To initiate the display process, the browser sends a request called a GET to the web server that hosts the desired web page (see step 1 in Figure 2). Each object in the web page must be requested individually with a GET request that contains its specific address. As a response to the browser's first request, the server sends an object referred to (for the purposes of this paper) as the web page framework (see step 2 in Figure 2). This web page framework object includes the text and formatting code as well as references to the other objects contained in the page. The browser then requests and receives each object needed to display the page by repeating the request process and waiting to receive each object from the server. The reader has probably unknowingly witnessed this process when he or she has seen the text of a web page load before the images.

If a web cache is used, the process is similar but has a few important differences. A shared web cache is a server located on the same local network as the computers that are accessing web content. A web cache server's responsibility is to store a copy of objects and web pages following a set of guidelines established by the administrator. This storage process is known as "caching" hence the name web cache. The connection between the shared web cache server and the client computers is generally very fast compared to the connection to the Internet.

Imagine a small stream compared to a large river. The small stream carries less water and in this example at a slower speed. The large river moves at a faster pace and carries much more water. The water is the information and the stream and river are the links. The small stream represents the internet connection that is slower and carries less information. The large river represents the local connection that is many times faster and carries much more information. Applied to web caching, this means that computers on the local network can communicate with the web cache much faster through the river-like connection, than with the Internet as a whole through the stream-like connection. This is an important concept that the reader will see is the major reason why shared web caches can improve performance.

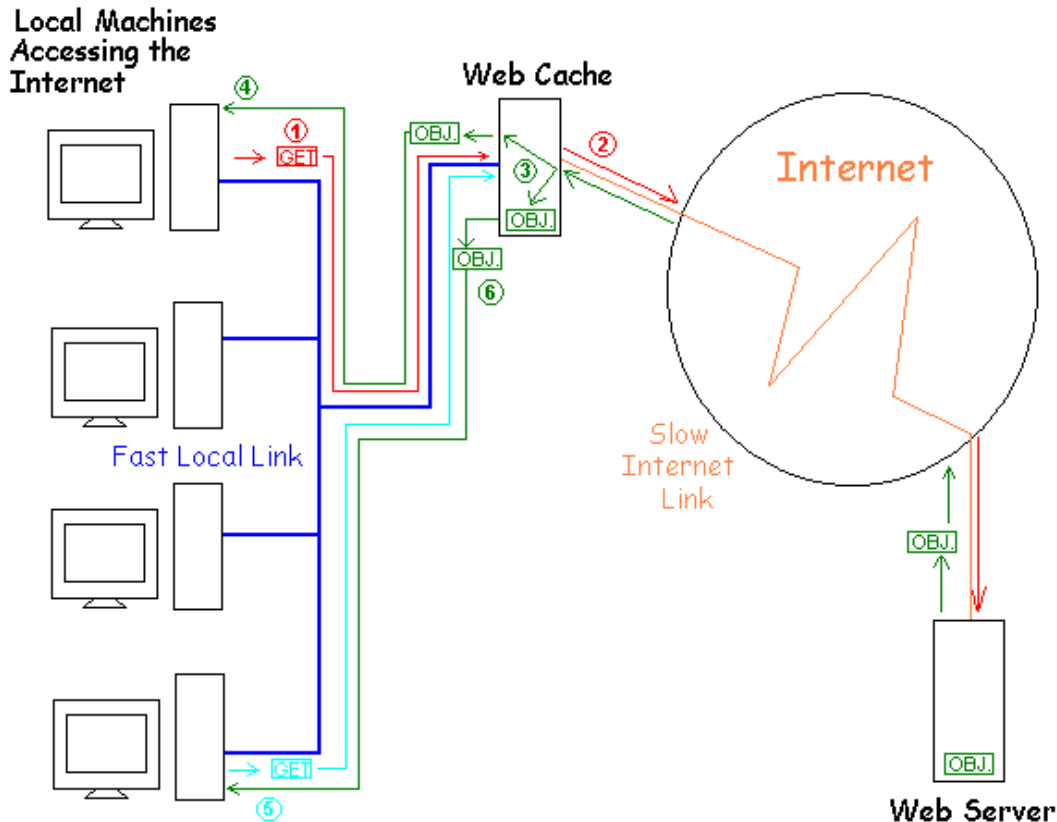


Figure 3

When a browser is configured to use a web cache, it sends the GET request to the web cache instead of the web server that hosts the desired web page or object (see step 1 in Figure 3). The web cache checks its storage for the requested content and may respond in various ways.

If the web cache does not find the requested content, it requests the content from the web server (step 2 in Figure 3). The web cache then stores a copy in its database and forwards a copy to the requesting browser over the fast local link (step 3 and 4 in Figure 3).

If the web cache encounters the desired content in its database, it still needs to verify with the real web server that the information is up to date because the web page may have changed since it was last accessed. The web cache may then send a conditional GET request to the web server on behalf of the browser. This conditional GET works differently than the normal GET request. It includes a section that contains the last modified date of the object stored in the web cache. The server will receive the conditional GET and if the object has not changed, it will send a very small fast reply that lets the web cache know that the object is still accurate. The web cache will then send a copy of the object from its storage over the fast local connection to the requesting browser (step 6 in Figure 3). If the content has changed, the web cache will update only the specific objects that have changed and store them for future use, then forward them to the requesting browser.

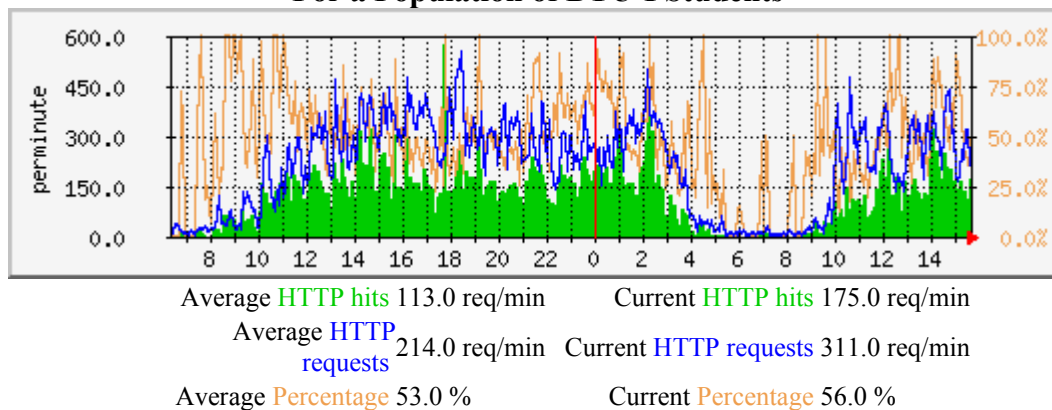
For the reader's better understanding, a web cache can be compared to a teacher's assistant in a university setting. In the university setting, consider a professor that has a very busy schedule and several hundred students. If a student has a question or concern outside of

normal class time, it may be difficult to contact the very busy professor (internet connection). Rather than waiting for the professor, the student asks the question to the teacher's assistant (web cache), who is readily available. If the teacher's assistant does not know the answer, he or she will ask the professor once, and then provide the answer to any future questioning students. If the teacher's assistant already knows the answer, he or she can simply respond to the student and allow the teacher to handle other more unique questions and situations.

### Technical Benefits Introduced by a Web Cache

Now that the basic process of web caching has been introduced, the technical benefits that a web cache provides can be explored. The main benefit is that the perceived wait time for a web page is significantly reduced. If an object can be located in the local database, it can be returned much more quickly than having to wait for the slower connection to the Internet to return the object. When a request can be satisfied by a web cache that request is known as a hit. A hit rate is the percentage of hits per requests. According to Michael Rabinovich and Oliver Spatschak, authors of *Web Caching and Replication*, between 20 and 45 percent of requests can be satisfied by a web cache, with some studies attaining hit rates of up to 50%. (2001)

**Proxy Hits/Requests: 'Daily' Graph (5 Minute Average)  
For a Population of BYU-I Students**



**Figure 4**

The requests and hit rates for HTTP (web) traffic of a population of approximately 400 BYU-Idaho students are represented Figure 4. The reader may note that hit rates of about 50% are attainable (see Average Percentage and Current Percentage in Figure 4). This is excellent, especially since not all objects are cacheable. Dynamic objects, for example are flagged so that they are not cached, allowing the web cache to store items more likely to be useful. Even though not all objects are cacheable, a web cache can still improve performance for non-cached objects. The more objects that can be accessed from the cache, the lower the number of objects that must be sent over the Internet connection, freeing the connection to access non-cached items more quickly. Also, if a particular web server happens to be slow or overwhelmed a web cache can hide sluggishness from the end user if the objects that server provides are stored in the web cache.

Another important fact should be considered when taking into consideration the BYU-Idaho network. A large percentage of the traffic generated by BYU-Idaho students is satisfied by the university's own web servers over a local high speed link. This would nullify some of the

improvement experienced by a web cache for any pages located within the byui.edu domain that are served on campus. For any other traffic, all the benefits described above would be valid.

The reader should note that there is a significant difference between the shared cache proposed by this document and each browser's individual cache. A browser's cache can also assist in improving perceived Internet access speeds, however only for the specific machine running the browser. With a shared cache, as soon as an object has been cached by a single request from any of the local computers, it is available to all other requesting users in the entire network. A simple example should show the difference. Suppose a very small network exists consisting of 5 computers running browsers and requesting the same web page. Suppose this page consists of 5 objects. If each of the 5 computers requests the five objects, 25 requests and 25 objects (most of which are repeats) must pass over the same shared Internet connection before the objects can be cached so that any machine can access them. With a shared web cache in place, only 5 requests and 5 objects must be sent before the web page is cached for all machines to access. As shown in this example, the larger the network, the greater the improvement that can be expected from the implementation of a web cache. With over 400 computers in the library alone, the improvement that could be experienced is evident. (Brigham Young University-Idaho, n.d., n.p.)

## Implementation

There are several commercial products available for use with varying costs and services provided. Squid is an open-source (free) implementation that can be easily installed on a Linux server with minimal processing power. (*Squid Web Proxy Cache*, 2006, n.p.) Servers that have been sold as surplus by BYU-Idaho could be used to implement Squid. (*Surplus List*, n.d., n.p) Due to the limited scope of this paper and varied opinions of information technology professionals, a dollar figure describing the cost of installation is intentionally omitted. Suffice it to say that it could be done relatively cheaply, especially if installed by existing staff.

The layout of the BYU-Idaho network also lends itself well to a distributed server approach with several web caches located strategically around the campus, or a rack in the server room located in the Kimball building could be provided specifically for this purpose. Implementing more than one web cache server increases installation complexity, but also increases the performance gains that could be experienced.

In line with copyright issues that are prevalent in today's Internet based media distribution network, the reader may wonder as to the legal ramifications that may occur when storing copies of copyrighted materials on BYU-Idaho owned servers. According to Wikipedia: "In 1998 the DMCA added rules to the United States Code (17 U.S.C. § 512) that largely relieves system operators from copyright liability for the purposes of caching." (2006, n.p.)

## Benefits for BYU-Idaho

A web cache can provide an economic benefit with regards to the cost of the internet connection. High-speed internet connections are very pricy. The more bandwidth (speed and information capacity) needed, the more expensive the connection.

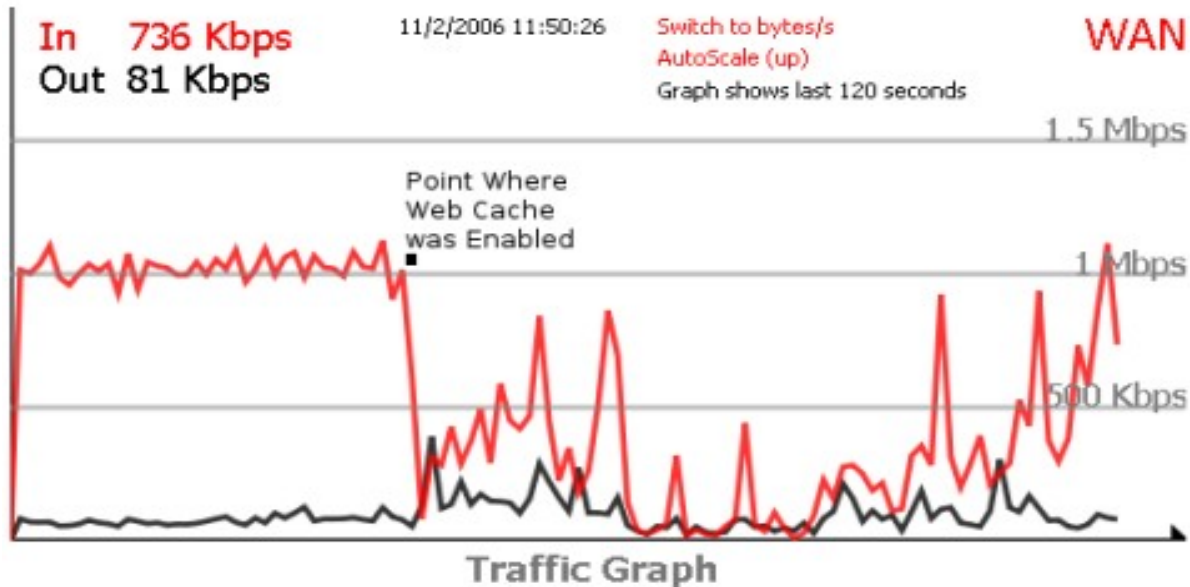


Figure 5

A web cache can optimize bandwidth usage, as shown in Figure 5. Figure 5 is a graph representing an Internet connection of 1.5 Mbps (megabit per second) with a 1 Mbps throttle set. The connection was maxed out at full usage until a web cache (with previously stored data) was enabled. As the reader may note the usage of the connection was reduced significantly due to the implementation of the web cache. These results were experienced by a population of BYU-Idaho students, and similar reductions could be experienced by the university. Bandwidth usage reduction is important because costly bandwidth upgrades can be avoided and more people can be provided better service with less bandwidth.

The reader should be well aware that the extensive wireless network already in place at the university draws more and more users every semester as more students are seen using laptops, personal data assistants, palm computers and other wireless devices that continue to place a load on limited network resources. According to research published by M.Granger Ali and Sajjad Zahir in spring of 2006, caching of web objects is deemed to be almost a “necessity” with wireless networks. (p.66) With the growing student population and accompanying growth of faculty and support staff to service it, BYU-Idaho’s Internet needs are only going to increase in the future. A web cache could help to supply this increased need for internet based resources.

## Conclusion

The implementation of a web cache on the BYU-Idaho campus would be a large benefit. Students and faculty could experience a moderate to significant improvement in display times for web pages. Although the unique nature of local servers for BYU-Idaho web content may diminish the improvement somewhat, an improvement could still be experienced. Implementation, although technical, could be completed by the university’s information technology employees at a minimal cost. Optimization of the existing internet connection resulting from the implementation of a web cache could produce savings now and in the future. A web cache would help the existing BYU-Idaho Internet infrastructure be more scalable and adaptable to the increase of students and faculty.

## References

- Ali, G. & Zahir, S. (2006, Spring). Performance evaluation for web applications with web caching in a distributed wireless system using Opnet(tm). [Electronic version]. *The Journal of Computer Information Systems*, 46 (3) 57. Retrieved December 14, 2006, from <http://wf2dnvr4.webfeat.org/>
- Brigham Young University-Idaho. *Surplus Material List*. (n.d.) Retrieved December 5, 2006, from <http://web.byui.edu/SurplusList/>
- Brigham Young University-Idaho. *Library Services*. (n.d.) Retrieved December 9, 2006, from [http://www.lib.byui.edu/Services\\_ContentRight.htm](http://www.lib.byui.edu/Services_ContentRight.htm)
- Kurose, J. & Ross, K. *Computer networking: A top down approach featuring the Internet*. (3rd ed.) (2005) Boston: Addison Wesley.
- Rabinovich, M. & Spatscheck, O. (2001). *Web caching and replication*. [Electronic version]. Boston: ProQuest Information and Learning Company.
- Squid Web Proxy Cache*. (2006) Retrieved December 5, 2006, from <http://www.squid-cache.org>
- Web Cache. (2006). Wikipedia, the free encyclopedia. Retrieved December 9, 2006, from [http://en.wikipedia.org/wiki/Web\\_cache](http://en.wikipedia.org/wiki/Web_cache)
- Wessels, D. (2001). *Web caching*. [Electronic version]. Sebastopol, CA: O'Reilly & Associates.

## Appendix: How to Configure a Squid Web Cache

This tutorial was made referencing squid-2.6.STABLE9 on a Fedora 6 Distribution.

For additional help check out the excellent wiki at:

[http://www.deckle.co.za/squid-users-guide/Main\\_Page](http://www.deckle.co.za/squid-users-guide/Main_Page)

Download Squid Cache from official site (this links to the tar file):

<http://www.squid-cache.org/Versions/v2/2.6/squid-2.6.STABLE9.tar.gz>

OR

Install squid using your package manager. I demonstrate using Fedora because that's the distro that I have.

On Fedora at the command prompt enter:

```
yum install squid
```

```
press y to install package when prompted
```

Before we edit the squid configuration file, a few guidelines from the Squid User's Guide are useful to review:

The default configuration file is probably right for 90% of installations - once you have Squid running, you should change the configuration file one option at a time. Don't get over-ambitious in your changes quite yet! Leave things like refresh rules until you have experimented with the basic options - what port you want your to accept requests on, what user to run as, and where to keep cached pages on your drives.

The Squid User's Guide includes lots of web caching history as well as some good pointers that cover things like replacing other types of web caches (Such as Microsoft's IIS proxy). I encourage the reader to access the user's guide for in depth configurations beyond the scope of this basic setup.

If you used the package manager in Fedora, the squid.conf file is located in this directory:  
/etc/squid/

The default directory according to the Squid User's Guide is:

```
/usr/local/squid/etc
```

If you can't find it, try a search. To search for the squid.conf change to the root (/) directory and enter the command:

```
find -name squid.conf
```

The location of the squid.conf will be output to the console if it is found.

Change directory using the cd command to the directory where your squid.conf is located.

I suggest making a backup as a good practice before modifying any configuration file.

Use the cp command (in the example I call my backup squid.conf.old):

```
cp squid.conf squid.conf.old
```

If we you up or something goes wrong, simply copy it back using the reverse:  
`cp squid.conf.old squid.conf`

Now open the squid.conf using your favorite editor. I prefer vi.  
 Command to open squid.conf using vi:  
`vi squid.conf`

Luckily for us the squid.conf file has been well documented. The documentation included can be very helpful.

One of the most important things to set is the cache directory. This will be the directory where squid will store cached files.

This is stored in the squid.conf file as *cache\_dir*.

Use the / key in vi to search:

`/cache_dir`

The default is:

`cache_dir ufs /usr/local/squid/cache/ 100 16 256`

`/usr/local/squid/cache` is the directory, the 100 represents 100 MB the size limit for this cache. The other two parameters specify the number of directories for squid to use. We won't worry about that for right now. If you have enough disk space and would prefer, I would recommend increasing the cache size.

In vi press i to insert and simply change the 100 to the size of your choice. If you want a gigabyte, simply add another zero.

Now lets set the *cache\_mgr* tag.

Do another search for *cache\_mgr* in vi:

`/cache_mgr`

Set the *cache\_mgr* tag to your email address.

Example:

`cache_mgr someone@somewhere.com`

Some other more advanced tags that you can set

*cache\_mem* – Sets the amount of RAM that squid can use.

*maximum\_object\_size* – Sets the maximum size for cached objects.

*minimum\_object\_size* – Sets the minimum size for cached objects.

Remember that if you mess up you can copy your old squid configuration file back and start over.

Try experimenting with some of the other settings in the configuration file and remember to have some fun!