An Introduction

to

Caching Technology for e-learning Environments

Equiinet Limited
Edison House
Edison Road
Dorcan
Swindon
SN3 5JX
UK

Tel : 01793 603700
Fax : 01793 603700

www.Equiinet.com
www.CachePilot.com

Contents

Introduction

The sheer scale of delivering sophisticated (and thus sizeable) e-learning content across large numbers of schools, to an increasingly large population of student PCs, puts tremendous demands on communications networks. The Wide Area Network bandwidth required by UK educational establishments to carry this and other information is an expensive commodity and must be used efficiently.

For e-learning material to be effectively delivered in academic establishments it should be presented in a coherent fashion to many students simultaneously - precisely when and where it is needed. It must be capable of being managed and updated easily by not only the content providers, but also by schools or colleges who will wish to add their own information and controls.

The requirement for high bandwidth Internet connectivity for schools has long been recognised, and the required moves by those who control budgets are already underway. Provision of high speed circuits capable of 2, 4, 8 or even 10 Megabits per second (Mbps) into schools are being planned and rolled out. However, while 10Mbps sounds like impressive performance, put in context of emerging e-learning content requirements, it is now recognised that this is insufficient for simultaneous delivery to even a few dozen students. For example, a single digital video stream alone could take 1.5Mbps!

Typical UK primary school PC population numbers are something in the order of 30 PCs per school, while secondary schools and colleges usually have in excess of 200 PCs. If sophisticated content is being simultaneously accessed by large numbers of pupils at multiple schools, the infrastructure between the origin web server containing the e-learning content and the individual pupil cannot cope. This is despite government-backed investment to significantly increase bandwidth capacity.

The solution is to bring the content needed for that individual school - and that particular day's classes for a given group of pupils - to be stored on campus ahead of when it is needed. Network performance beyond the school, and its actual connectivity bandwidth to the outside world, is then largely irrelevant.

Networking devices known as 'Web Caches' aim to bring content closer to the end user of the material. These devices have been available for a number of years and have been deployed successfully in the Internet infrastructure at ISPs, as well as by businesses on their own sites. UK educational bodies have also now recognised that this technology has significant advantages and benefits for their environment.

This paper describes how caching acts as a means of fast and efficient content delivery. It goes on to explain why simply deploying ever more bandwidth is not a practical proposition, and that to satisfy the higher demands placed by the latest e-learning applications, a significant part of the ultimate solution is deployment of a new generation of specialised 'smart caching appliances'.

What is Web Caching?

Traditional Web Caching is the technique of storing previously requested data - usually ftp and http web traffic – geographically closer to those who may require it in the future. For example, if a school has a cache, and one of its pupils requests a web page from a remote web site which has not been requested before, the cache will automatically add a copy of this page to its data store as it passes through. If another student then requests this same web page, it will be delivered directly from the cache, hence saving the delays of requesting and retrieving the data from the original site. Smart Caching offers additional benefits of having content for that day's lessons preloaded ready and waiting. This may be information that is provided by specialist content providers or public information defined as required by the local teaching staff. Smart Caches can do a variety of other important tasks including acting as a firewall, a WAN router and a Web content filter.

Why Web Cache?

There are a number major reasons why Web Caches are used, but put simply, the key elements are to improve performance and reduce costs. The top reasons most usually given are probably:

- To reduce the time it takes to view a web page (i.e. reduce latency)
- To reduce traffic and impact on both WAN and LAN Network Bandwidth**.**
- To reduce traffic impact on the Web Servers supplying the information.
- To allow use of rich new content for e-learning applications.

What does a Web Cache look like?

A Web Cache generally comprises powerful server-like hardware architecture with fast processor, disk and memory. Physically some manufacturer/suppliers will build a product out of standard PC components – but these devices are unlikely to have high performance characteristics. Indeed they still need a monitor, keyboard and mouse to operate!

A purpose-built Caching Appliance does not look physically like a standard PC, nor is it a conventional PC or server. It will have hardware components optimised exclusively for high performance caching applications. It will typically have at least two LAN (Local Area Network) ports and probably other interfaces for connecting to the Internet via various forms of WAN (Wide Area Network) services.

Software used on caching devices is typically quite specialised as it is tailored for the specific job – and doing it well. The underlying operating system for performance reasons will not be based on Microsoft software. Typically, open source software such as Linux or BSD may be used, or even proprietary operating systems. On top of the core operating system, a variety of addition proprietary functionality will be added by individual manufacturers, that defines what, how and when objects will be cached, as well as how that process is controlled and managed.

Where is Web Cache physically located ?

A Web Cache is deployed somewhere in the network infrastructure, physically located between Web Servers or '*origin servers*' and the PCs whose users need to access to the web content.

In an e-learning situation, a cache could be located say at an LEA, behind a firewall, at the gateway between all the connected schools and the public Internet. However, this is not optimal in terms of performance or cost – since it still means an LEA must necessarily invest heavily in onward communications bandwidth to each school.

The best location for a cache is actually sited on-campus, directly connected to the LAN of the school or college. It also provides the control and flexibility to tailor the exact content necessary for each individual school.

In addition, it is possible to interconnect caches so that they work in tandem for even further efficiencies. A centralised cache could work on behalf of many schools and each establishment additionally could deploy their own local cache on-site. Caches have their own communications protocol for operating in this hierarchical fashion – Internet Cache Protocol (ICP). Information is given below on this topic.

How Traditional Web Caches Work

Caching is in fact used in all areas of computing – at the CPU, in the disk controller, in the operating system, even in the standard browser software.

Web Caching works on the principle that any data that has been frequently accessed in the past is likely to be needed again. Storing this information in the RAM memory or on the disk of a local cache device eliminates the need for further requests for data to be directed to the server holding the data.

A Web Cache physically sits between Web servers or '*origin servers*' and a client or many clients (i.e. the student PCs), and watches requests for HTML web pages, images and files collectively known as '*objects*' come by, automatically saving a copy for itself.

Even though a whole page is stored in the cache, it is still stored as a number of discrete objects. If a user subsequently accesses objects from that page, they can be supplied individually from the cache. This typically happens as a user navigates a Web site, where pages accessed will contain different content, but will include many of the same objects.

Each time the user retrieves a new page, even though that page is not in cache, many of its common objects will be – logos, background images, etc. Couple this with the fact that the cache engine will usually pre-fetch all objects on a page in one hit (rather than in small groups like the browser), and it becomes apparent that a Web cache will improve the browsing experience even when requesting brand new pages.

Benefits for users

Caches reduce the time it takes to view a web page (latency) because the request is satisfied from the cache memory (which is closer to the requesting student) instead of the origin web server, it takes less time for the client to get the object and display it. This makes Web sites seem more responsive.

Benefits for network administrators

The automated nature of letting the cache collect objects and fill the cache storage up for itself is attractive to network administrators, as it imposes little in terms of management overheads.

Caches also reduce network traffic**,** as each object is only requested from the server once, it minimises the amount of bandwidth used by a client PCs, not only across the Internet but also lessens bandwidth overheads on the local educational establishment network. This saves money on bandwidth and networking equipments costs, and keeps the whole procedure of conducting e-learning more manageable.

Benefits for content and service providers

Additionally Caches can also reduce traffic and impact on the Web Servers supplying the e-learning content. Because caches can off-load much of the burden from the web server, the investment in server capacity can be much reduced.

In practice a request for a web page may be made to many caches en route, and indeed, delivered from one of these caches, not the actual web site, if the page had previously passed through one of them, as a copy would have been saved. The Internet is rich in caches as they save both time and bandwidth.

Smart Caching – What is it?

Before explaining some of the 'smarter' aspects of more advanced caching devices, it is important to review some of the limitations that they overcome versus previous generation traditional caches.

An obvious limitation of a traditional cache is that it stores data sequentially as and when it arrives, and not in any ordered form. So, even if a complete web site had been cached over a period of time, it would be interspersed with other sites and web pages on the same caching device, and introduce its own delays whilst searching for each page.

Indeed, it may be quicker to re-visit a web site than to search a 10Gbyte traditional cache for a specific object, especially considering the bandwidth now available to many schools. A typical e-learning site could comprise many tens of Gigabytes of data, far too big to fit into standard caching products, and far too big to be efficiently navigated, as a new search of this huge content would be required for each and every page requested.

It's relatively easy to spot a traditional cache from its smarter successors by looking at hardware specifications. Small disks mean that the traditional cache has no way of quickly accessing required objects, hidden amongst lots of other cached information. The larger the disk, the more content that is cached, which often can mean wasted time searching. Worst of all, after searching through a large cache of random objects, is to find the object is not present and the cache still needs to refer to the origin server for the object, having already created significant response degradation searching. Hence with this traditional type of product, somewhat ironically, it often works best with relatively modest amounts of RAM and disk.

These traditional caches are not web servers in their own right. They only have a necessarily limited selection (as explained above) of commonly used objects that make up a web page. Pull the plug on the traditional cache's link to the Internet and you have nothing useful to view. It still needs Internet connectivity to the origin server to work, in order to supply meaningful content.

Where smart caches differ is by replicating or 'mirroring' whole sections of a web site and then be completely autonomous from the origin server. In this instance the smart cache is acting as a local web server in its own right. Web content is copied into what is known as a 'Webshare' resident on the Smart Cache. Browsers on student PCs point at the Webshare to view the mirrored content. Physically disconnecting the WAN link on the connection to the Internet proves it's completely freestanding!

There are some practicalities to appreciate with this approach. Mirroring massive amounts of content, or even trying to replicate a complete remote origin web server is not practical. However, copying particular focused elements of an origin server and making sensible limits of how much is loaded is an extremely powerful approach. By specifying out of peak hours periods when this activity is conducted (e.g. overnight), makes this a very practical proposition and provides for a more permanent local copy of content than using a conventional caching approach.

These functions go way beyond the capability of standard traditional caching as discussed below.

Smart Caching Functionality: Pre-Caching and Web Mirroring

Smart Caches can ensure information is kept as fresh as possible by preloading when a change is likely and fetching a new copy ahead of time. Using such techniques, the cache attempts to ensure that out-of-date information is rarely, if ever, served to users.

On a Smart Caching device such as Equiinet's CachePilot there are two distinct storage areas which can be loaded ahead of when the content is actually needed to be viewed by the students. The first is a 'Cache' storage area into which information can be 'Pre-Cached' for future use and the second is a 'Web Mirroring' file storage area, which can be pre-loaded with complete pages.

The loading principles are similar in terms of administration commands, but how the information is preserved is quite different.

Pre-Caching - into the Smart Cache

As with a traditional caching device, there is storage area allocated on hard disk, which is used to cache objects. Without any administrator intervention this area will automatically be filled with objects that the connected PC users introduce into the cache by browsing the Internet.

When the cache area on the disk becomes full, the Cache disposes of the oldest and least frequently used objects, just keeping the freshest and most used items.

The Caching device will automatically copy the most frequently used objects from the disk into its RAM memory to provide the fastest possible response time, rather than always waiting to retrieve them from a relatively slow disk drive.

The disadvantage of this whole approach is that no performance benefit will occur until the first user activates the collection of objects, which

then reside in the cache for the benefit of all subsequent users. There is also no way to force objects to stay in cache. If they become the oldest and least used and the space is needed – they get evicted!

One way around this is to pre-load into cache the content for a particular lesson or subject matter, ahead of time. Smart Caches can be configured to preload all the objects for specified pages in an orderly structured fashion during off-peak times.

Overnight, the cache could be completely cleared of the previous day's contents and the specific URLs of the desired sites to be copied for tomorrow's lessons can be specified in a table. This timing control can be daily or even hourly.

Specifying a complete web site to be cached locally is usually not practical nor is it usually required. A content provider such as the BBC has a large array of servers containing vast amounts of content. Replicating entire content provider sites is therefore unrealistic.

Web site pages are arranged in an inverted tree structure. Specifying just the part of the site that you are interested in and then the depth of levels beyond the starting point that you wish to copy, makes pre-caching more realistic from a size perspective. This depth specification is known as a 'Recursion Level'.

Take the example of a hypothetical site called http://www.ourcontent.co.uk. There may be 1000 web pages on the site, made up of vast numbers of objects. By starting at a lower level such as http://www.ourcontent.co.uk/history/ we cut down to 10% the potential number of pages, and by being even more specific by starting at an even deeper level say: http://www.ourcontent.co.uk/history/1stCentury we perhaps cut out 95% of the origin site content. Similarly if we set a Recursion Level depth of say 2 or 3, we will not pull over the more obscure parts of the site. By employing both these methods we have kept the pre-caching task to a sensible size, while we have all the necessary content held locally.

One issue that needs to be given some thought is if the origin servers that hold the required data have traditional caching devices between them and your Smart Cache. Do these caches have the most up to date objects? One way around this dilemma is to force the Pre-Caching or Pre-loading process to only take data from the origin server. Alternatively, in your network, you may have a hierarchy of Smart Caches, whereby the device that is say at the LEA is caching on behalf of a community of schools. The LEA device assumes nothing, and only accepts content from the origin server. Caches at each school could then accept cached content from the LEA, happy that the freshness of this content was predetermined. This would cut down bandwidth requirements and response times at each school.

Web Pages from the origin server may have 'Requisites' which are non-HTTP files. For example, the origin server contains an HTTP web page with a list of educational documents, and the documents are stored on the origin server in PDF format. Traditional caches would store the HTTP page, but not the PDF documents – so most of the useful content remains remote. The Smart Cache can be told to load some or all of this PDF content so it is available locally.

Web Mirroring - Pre-Loading into the Smart Cache's WebShare

One of the main benefits of Web Mirroring is that even if the origin server content has not been written to be 'cache friendly' i.e. has very poor caching attributes, these can be overcome by taking all the objects of all the required pages and making them reside locally on the Smart Cache's WebShare.

Keeping Pre-Loaded Content Fresh

Pre-Loading into a WebShare overcomes issues of lack of permanency within the caching area, and also bypasses any inadequate settings by the web site authors of cache control headers. However, it is necessary to think through the issues of potentially holding stale information. The Smart Cache administrator needs to decide whether the data should be reloaded daily, weekly, monthly etc., to ensure the freshest content is available locally for students to access. This will be based on how frequently this type of information changes, and how often the authors are likely to create updates. The Cacheability Tool discussed later in this paper helps, by showing dates when individual pages have been altered by the authors.

Limiting the scope of Web Mirroring

Consider the issue whereby you have specified a particular area on an origin server you wish to Pre-Load on your Smart Cache. You are aware that you have to be cautious – you don't want, or need, to copy the whole site – so you specify the starting page and a sensible depth of pages below that starting point. However, there are issues of which you still need to be aware.

What should happen if the scope you have set links back up the web page tree to the home page of the site (as often happens) and therefore onward linking to all pages on the site? Also, what happens if the origin site has links to other sites, and that site also has external links to yet more sites? The danger is that very quickly you are mirroring much more content and far more sites than you ever expected.

Smart Cache controls give you the means to specify whether you should stay within the boundaries of the particular site you are mirroring. Also whether you stay within the boundaries of the starting URL and only

specified addresses below that. It is also possible to explicitly define and prevent individual directory structures from being copied.

Content that cannot or should not be Pre-Loaded by Mirroring

Origin server web pages that authors deliberately mark as stale, because they contain large numbers of dynamically changing objects, should not be Pre-Loaded into a WebShare without some thought as to the consequences of potentially showing out of date information. A discussion on Dynamic content and a Cacheability Tool that can help you decide on these issues follows below.

Other content that cannot / should not be Pre-Loaded includes e-commerce pages and anything using HTTPS for secure client server communication.

FTP Mirror and HTTP Mirroring

So far the discussion has concentrated on mirroring HTTP web pages. However it is possible to mirror non-HTTP information. This could be any file type capable of being transmitted by FTP (File Transfer Protocol). This would include Microsoft application files such as Word or Excel, and file types like PDF.

There are two type of FTP transfer mode, Active and Passive. Both of these can be handled.

Access Controls

There may be a variety of access controls on the Smart Cache, or a proxy server that the Smart Cache is onward linking to, or on the remote origin server holding the desired content. The content may be deliberately protected because it is of a proprietary nature or is licensed and the authors wish to control who downloads what. In this case whether the content is to be pre-cached or pre-loaded into a Webshare, the Smart Cache can automatically supply the required password for the remote origin server. In addition access control passwords for local Webshares and parent proxies can be defined for automatic presentation by the Smart Cache.

It may be required by the authors of the pages on the origin server holding the desired content, that users utilise a particular type of browser. Typically, this is to ensure that their specially formatted content can be viewed correctly. The Pre-Loading mechanism can emulate particular browsers to satisfy the access controls on the remote web site during its mirroring process.

Some origin web sites are only accessible via links from another specified site, known as a '*Referrer*'. The Smart Cache can deal with this situation if the Referrer needs to be specified.

Absolute and Relative Addressing

The authors of the origin server content may have used a mix of absolute and relative addresses on their site (fairly normal). Pages that are linked to each other with their relative positioning are easily copied and made operational on a Smart Cache. Absolute addressing will relate explicitly to the addressing situation at the origin server. When the site is mirrored onto a local Smart Cache it must convert all absolute addresses into meaningful relative addresses.

Fundamental Web Performance Issues

Many of the root cause problems in delivering sophisticated e-learning content are concerned with how Web pages are actually constructed - as a number of discrete objects. Every item on a Web page - be it a logo, menu button, picture or block of text - is a separate object. This requires a highly complex set of requests and responses between the client PC sitting in front of the student and the remote Web server, in order to correctly display a complete Web page.

Round trips between client and server are required for the initial identification of where the remote site is located from a network addressing perspective and setting up the underlying communications (DNS lookup, opening the TCP connection, and retrieving the Web page details).

After that, two round trips are required for each and every object on the page. Further delays are caused by Web browsers only being able to fetch objects in groups of a few at a time. With some poorly designed Web pages containing tens or even hundreds of small objects, more time is spent requesting and retrieving objects than on actual data transfer.

The best solution is to bring as much data as close to the user as possible – which is precisely the role of a caching device.

Cache Hierarchies

A single Web cache will reduce the amount of traffic generated by the clients behind it. Similarly, a group of Web caches can benefit by sharing another cache in much the same way.

By using the Internet Cache Protocol (ICP), Caches can be arranged in a hierarchy or mesh for additional bandwidth savings. In a cache hierarchy, one cache establishes peering relationships with its neighbour caches.

Caching devices such as Equiinet's CachePilot work transparently and intelligently so that no configuration is needed by the user, and little overhead is placed on the network administrator.

Local caching devices can be configured to work in conjunction with an ISP's cache or one located centrally at an LEA. Both devices are aware of the age and content of the information that is being carried by the co-operating device, considerably reducing the likelihood of having to go back to the originating server.

Where the time to live for Web page information is short, or where data is dynamic, CachePilot will refresh cache data regularly (or always go directly to the Web site), ensuring that stale data is never served from the cache.

Although Web caches use HTTP for the transfer of object data, inter-cache communications benefit from a simpler, lighter communication protocol. ICP is a lightweight message format primarily used in a cache mesh to locate specific Web objects in neighbouring caches. One cache sends an ICP query to its neighbours, and the neighbours send back ICP replies indicating a "*hit*" or a "*miss*".

Handling Static and Dynamic Content

Some web pages obviously contain dynamically-changing data that needs to be obtained directly from the origin server each time a request is made – typical examples would be pages that trigger database enquiries on availability levels, for perhaps currently available places on courses.

Most objects are fairly static, though, and can be stored in the cache for long periods of time in some cases. However, even relatively static pages will change occasionally, and many will have an expiration date stored in the HTTP header of the web page. This can be used by the cache to determine when a page needs to be refreshed from the origin server. An example of more static information might be a course syllabus, which would only be updated once per year.

Another technique available to the cache engine is to issue a "Get If Modified" request to the server each time an object is requested. Unlike a standard HTTP "Get", the "Get If Modified" request is fulfilled only if the object has been changed since the previous demand for the same object. When a cache receives a request for an object that it has already stored, it can thus ensure that if the cache's modification date for that object is older that the server's, the a new copy of the object is retrieved.

Testing how cacheable is a web site?

A number of tools are available to help indicate how cacheable a site may be. One such tool can be activated from : http://www.cachepilot.com/news/Tools.asp

Example of pages with low levels of Cacheability

The example shown below is the BBC learning site. The tool shows that the home page is not cacheable because the site authors have set none of the HTTP header information up, and they are also making use of 'Cookies'. (The red dot denotes low likelihood of being cached)

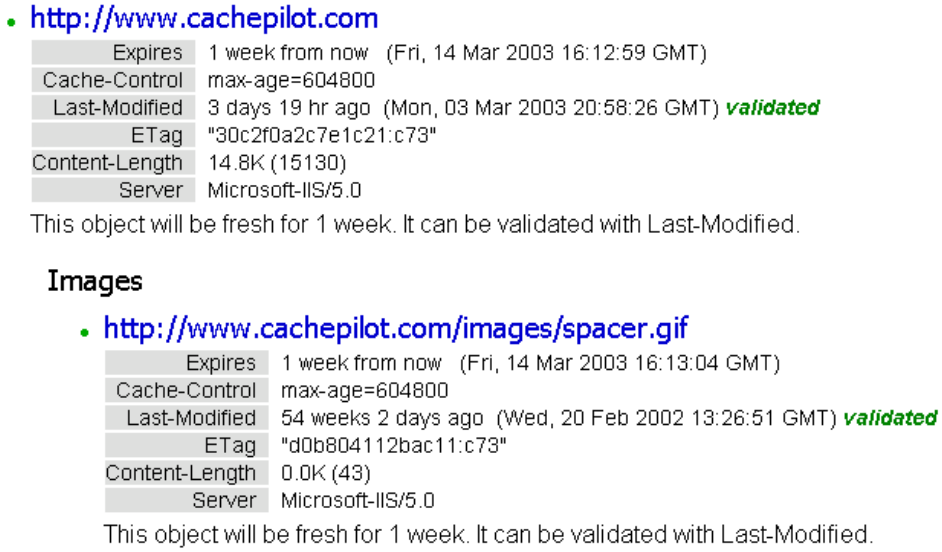
However, not all is lost when using a Smart Cache. Many of the objects on that page, and generally used by this particular site, are GIF images that are very rarely changed. As denoted by the green dot, these objects have a high likelihood of being cached.

Another alternative is to use the Smart Cache's 'Web Mirroring' functions rather than using pure caching. Web Mirroring (which is discussed above) gets around problems surrounding pages with poor caching attributes.

Example of pages with High levels of Cacheability

The example shown below is the Equiinet CachePilot site. The tool shows that the home page is indeed cacheable because all of the HTTP header information has been set up correctly by the site authors. (The

green dot denotes high likelihood of being cached). An Expiration date and Cache-Control information has been correctly specified.



How to Design Web Sites to be cacheable

As we have seen from the above example, the BBC pages are not particularly cacheable. Authors of web sites - particularly those involved with e-learning - need to design sites with caches in mind i.e. to be 'cache friendly'

To maximize the cacheability of your Web site, you should give Expires headers to all static-content elements (buttons, graphics, audio and video files, and pages that rarely change) so that they can be cached for weeks or months at a time. Dynamic elements or pages may be marked Cache-Control: no-cache if they change on every request, or perhaps Cache-Control: private if it's OK for the client browser to cache the item (such as for personalised resources that don't change often). When possible, all resources should have a Last-Modified date (so that caches can check for newer versions) and all resources should include Content-Length headers so that persistent connections are possible.

A technical paper covering these issues and more, has been written especially for web site authors and administrators and is available on: http://www.cachepilot.com/news/whitepapers.asp

How Smart Caches go beyond Pure Caching

True Smart Caching devices have a rich repertoire of capabilities to distribute content, but where they perhaps go beyond their simple 'smart' title is their ability to conduct a range of other key functions for the educational sector.

Diagram 1 shows a smart cache being deployed in a purely caching role. The browser on each of the PCs on campus points at the cache. If the web objects that are required to display the requested web pages are not currently present in the cache device, then the Cache communicates with the remote origin server connected somewhere on the Internet via a 'Gateway' device. This could be a single combined unit, which acts as a router, firewall, and filtering device such as Equiinet's NetPilot unit. Alternatively, it could be a combination of separate devices configured in tandem to do these same functions. The Cache will physically use a single LAN port - the PC's request will come in on the same port that the Cache uses to communicate with the Gateway.

Diagram 1 : Smart Cache deployed in pure caching mode

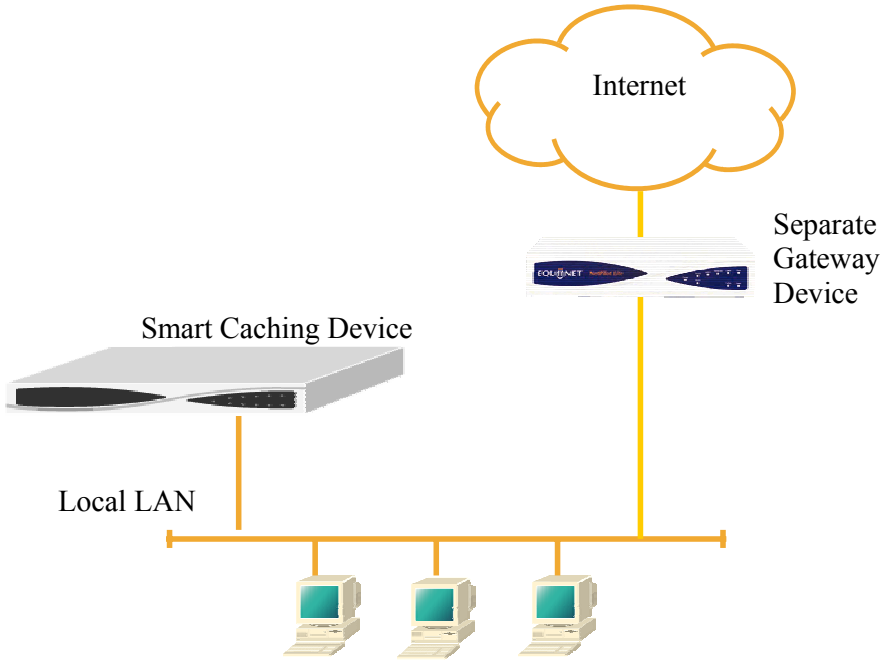
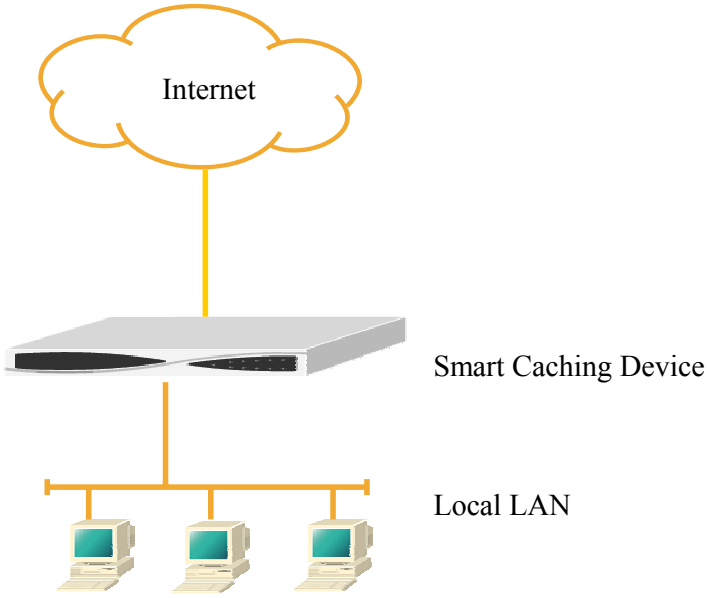


Diagram 2 below, shows a Smart Cache being deployed as both the Cache and the Gateway device. Physically, in this configuration the unit is utilising two ports. The incoming requests for web content come in via a LAN port. As before, if the Cache can satisfy all the requests for all the objects that make up all the requested pages it will do so locally, and not require any communication with the outside world. However, if it does indeed need to communicate with the remote origin server, it will

use a second port to directly communicate to the Internet and obtain the required information - without the need of a separate Gateway device.

Diagram 2 : Smart Cache deployed as combined Gateway and Cache



In practice what the Smart Cache is undertaking is a number of sophisticated tasks. The second port may be another LAN connection, or if the Smart Cache is very clever, it can talk directly to a wide area connection negating the need for a separate router device. Even smarter still, it can undertake all the duties of a firewall – in fact it has a completely integrated internal firewall, which protects the school or college from any malicious access from outside. Additionally, it can also conduct filtering duties checking that the Internet pages requested by the student are from a site that has 'appropriate' content.

This latter 'content filtering' task uses access controls and user profiles to define what is 'appropriate content'. Primary school pupils might have a profile that gives very restricted access, and older secondary school students while still being blocked from viewing obviously inappropriate categories of sites such as pornography, might be provided with much more freedom to browse. Yet further access profiles could be defined for teaching staff with even wider access allowed.

While nothing worthwhile is absolutely for free, these additional functions can be implemented with the only potential drawback being minor impact on processing performance of the Smart Cache depending on the number of locally attached PCs. Clearly the more tasks any device is given, the less free capacity it has. Configuring the right devices with the right capacity is still a requirement for Smart Caches, as it is for any networking or computer.

Comparing and Selecting Caching Products

Price, performance, features, service, suitability for educational applications, availability of e-learning content, etc. are all parameters that need to be evaluated.

Key caching functions that need to be evaluated for e-learning environments would comprise:

Structured data storage
The ability to faithfully replicate the directory and file structure of the source content, which will enable the speedy retrieval and delivery of the content without unnecessary searching.

Automatic content mirroring (data pull)
The caching appliance should be capable of automatically checking its content against a remote server containing the source content data, and copying any new or modified files and directories. This would usually be done on a daily or weekly basis, and data transfers actually occur during off peak periods such as overnight or during the weekend.

Central content distribution (data push)
The appliance should also be capable of receiving updates and modifications pro-actively sent to it from the remote server holding the source content data. This is a useful function where many local caching appliances under the control of a single authority are remotely updated by that authority.

Web serving
E-learning content is almost exclusively web-based, so a web server built into the appliance would ensure its delivery locally without additional hardware. This would also add resilience against network failure, as all local PCs would point directly to the server held locally on the appliance, which would in turn deliver the content also held locally as a replica of the source content.

Management statistics & logs
It is likely that both the controlling authority and the content provider would be interested in the access patterns and statistics of users, and these should be presented in a standard format ready for easy analysis. This data may also form the basis of charging for the service. It is also important that all updates are recorded, and errors reported if an update fails. These statistics and logs should be automatically transmitted to a central administrator on a regular basis.

Furthermore, where a single authority manages many caching appliances, messages relating to potential or actual problems should be automatically transmitted to a central administrator.

Storage Capacity for multiple content providers
It is likely that the combined content of providers of e-learning packages, such as Espresso and the BBC, will exceed 100Gbytes within the next three years, so a hard drive larger than this should be the minimum recommended for units presently being installed. The system must also be capable of being preloaded with content from all the required sources.

Performance
The appliance should also be capable of the simultaneous delivery of e-learning content, such as digital video, to many tens of users. A resilient and high performance operating system, such as Linux, would be preferable, as it would also enable more users to be served than from an equivalent platform running a Microsoft operating system.

Secure and automated Content Updates
The integrity of the e-learning package needs to be automatically secured by the updating process, to ensure that content is digitally signed off before it is added and capable of being accessed by students.

Caches – do they make financial sense?

Most of the caching on the Internet has been deployed to date, by large corporates and Internet Service Providers (ISP). It is true to say, that in both categories, not all of these organisations have deployed caching.

Why? Certainly caching with traditional devices used to be an expensive business. Purpose-built appliances with high performance processors and disk subsystems and lots of RAM running complex cache operating systems were not cheap.

The payback rationale for the ISP industry has been complex. Deployment of caching in the ISP market has little to do with altruism on their part – they are not looking to simply speed up access for their users.

The return on investment for businesses deploying caches has been clearer where data has been held remotely. The more US-based data a European business can store on this side of the link, the less expensive bandwidth is employed moving those same pages across day after day.

For educational establishments now looking for caching solutions today they have the benefit of choosing from new generations of Caching Appliance solutions at new price points. ISPs and business have been the trail blazers with early devices. Much has been learnt and new devices tailored for the e-learning market have been developed.

Using lower cost devices than those deployed by the ISP, educational users can now enjoy similar benefits in ensuring that regularly-accessed Web pages are stored on the local LAN, eliminating the need to use the limited bandwidth link to the Internet each time content is requested.

Caching Appliance Alternatives

Purchasing More Bandwidth

An obvious alternative to a cache is just to simply purchase more bandwidth. Prices are reducing and carriers would be prepared to do a deal for groups of organisations clubbing together to buy in bulk (as we have seen occur with some educational bodies already). However, we get the same well-known dilemma, as with adding a new lane on a motorway, that extra capacity rapidly becomes blocked especially at peak times. Temporary surges can swamp any portion of a network, regardless of the network's bandwidth.

Many primary schools are already increasing their bandwidth up to 2 Megabit per second leased line circuits (also known as MegaStream). However, they are also increasing their numbers of PCs. The effective throughput could be as low as 34K bits per second, i.e. half the speed of a modem the student would use at home!

Primary School Bandwidth Provision

Number of PCs	WAN Bandwidth In Megabits/ Sec	Effective Bandwidth after WAN Transmission Overheads in Megabits/ Sec	Effective throughput for at each student PC in Kbits/Second
30	2.048	1.74	58.03
40	2.048	1.74	43.52
50	2.048	1.74	34.82

Many secondary schools are increasing their WAN bandwidth capacity beyond 2 Megabits per second, to double or even quadruple that figure. However, this attempt to provide fast Internet access by just throwing bandwidth at the problem looks even more futile. This is clearly not the total answer.

Secondary School Bandwidth Provision

Number of PCs	WAN Bandwidth in Megabits/ Sec	Effective Bandwidth after WAN Transmission Overheads In Megabits/ Sec	Effective throughput for at each student PC in Kbits/Second
100	2.048	1.74	17.41
150	4.096	3.48	23.21
200	8.192	6.96	34.82

Caching can help prevent traffic jams by reducing or removing the need to get on the data highway. There is no need to travel huge Internet distances if the information is already located on your local campus.

Other Types of Caching

There are other types of caching options - on PCs and Servers - rather than on focused Caching Appliances

Some may hope that the local cache employed by the user's own PC browser as a way around problems of limited bandwidth. Unfortunately, this cache cannot be shared amongst several users on a network, and whilst some improvements may be seen in environments with small numbers of users (any form of caching being better than none), a number of heavy users will still cause duplication of data retrieved over the Internet link.

Another alternative that commercial organisations have tried but seems to have gained little headway in educational circles is the Microsoft offering known as Microsoft Internet Security and Acceleration (ISA) Server 2000, probably better known as Microsoft Proxy Server in its previous incarnation. This software is an attempt to put caching and Firewalling on a Windows 2000 server. Obviously it has the Microsoft name, but less compelling is the performance and price compared to current generation caching appliances, and lack of any significant features aimed at the e-learning environment.

About Equiinet's CachePilot

CachePilot™ has been designed as an all-in-one smart content delivery system for e-learning applications.

It has been developed to meet the growing demands of the education market, and it can store, update and deliver both static and dynamic content locally, enabling simultaneous access by multiple PCs.

E-learning packages are predicted to routinely exceed 100 Gbytes in size within three years, putting an enormous strain on data networks. It is vital that content delivery methods keep pace and CachePilot more than meets the challenge by incorporating a variety of clever techniques to ensure all steps are taken to avoid wasting valuable bandwidth or slowing down networks.

CachePilot is a Smart Caching appliance and it offers far more than just a traditional cache, and incorporates the Smart Caching capabilities described above – and more. CachePilot has a structured data store for easy retrieval of information, automatic mirroring to pull data, central distribution to push data, web serving, secure automatic updating and comprehensive management statistics and logs.

CachePilot has been specially designed to work alongside any of the ever-growing number of content providers, and has the flexibility and capacity to hold information from multiple sources. Third party content may be factory pre-loaded on to CachePilot. In addition, content may be pre-loaded by the CachePilot administrator in the form of replicated web sites. This allows lecturers to plan their lessons by pre-loading or pre-caching the required web content the night before the material is needed.

CachePilot products are powerful enough to handle data content of all shapes and sizes from any provider, using any speed and type of Internet WAN connectivity.

CachePilot includes a high-performance Smart Caching server - based on the popular Squid software - for web clients, supporting FTP, gopher, and HTTP data objects. Unlike traditional caching software, it handles all requests in a single, non-blocking, I/O-driven process.

CachePilot keeps Meta data and especially hot objects cached in RAM, caches DNS lookups, supports non-blocking DNS lookups, and implements negative caching of failed requests. It also supports SSL, extensive access controls, and third party content filtering from N2H2.

CachePilot will also try to ensure that it retrieves all objects associated with a web page in one hit when the page is first accessed, rather than in small groups in the manner of a standard Web browser. This means that as the browser requests objects on a page, they are already in cache, thus

speeding up data delivery on pages, which were not even in cache when first requested.

The integrity of the e-learning package is secured by CachePilot's proprietary automatic updating method, which ensures that content may be digitally signed off before it is added. Using a unique software upgrading method this solution provides peace of mind for content providers that rogue content cannot be added to their material without their approval.

CachePilot is available in two differing hardware form factors. CachePilot comes in an attractive enclosure for desktop or mounting on a shelf within a cabinet. CachePilot Enterprise is a 1U high enclosure rackmount device designed to fit in a cabinet. The Enterprise model also comes with the benefit of a faster caching engine and twin mirrored removable hard drives.

For more information see: http://www.CachePilot.com

Glossary

Automatic content mirroring (data pull)
The ability of a caching appliance to automatically check its content against a remote server containing the source content data, and copying any new or modified files and directories.

Boundary Caching
A unique Equiinet method of caching at the boundary or 'gateway' to a local area network, i.e. inside the Equiinet appliance.

Cache Hit Ratios
When a Cache finds a requested object of a web page in its memory, this is known as a 'Cache Hit'. When it finds it has to refer to the origin server to obtain the content, this is known as a 'Cache Miss'. To be effective the caching device should have a high ratio of hits to misses.

Central content distribution (data push)
Involves the Smart Cache being capable of receiving updates and modifications pro-actively sent to it from the remote server holding the source content data. This is a useful function where many local caching appliances under the control of a single authority are remotely updated by that authority.

Content Acceleration
Content acceleration involves smart caching to regulate the flow of this information without impacting the core network and without necessitating new bandwidth infrastructure.

Content Filtering
A facility to block or allow Internet sites and content from being accessed and viewed by an individual, a group of individuals, or all the connected users.

Domain Name Server (DNS)
A server that contains a database of host names and their corresponding IP addresses.

Dynamic web page content
Frequently changing *objects* within a web page that are flagged as always having to come from the origin server, i.e. objects that must not be cached – opposite to *static content*. Example - latest information from a constantly changing database.

File Transfer Protocol (FTP)
A service that supports the transfer of objects using TCP/IP between local and remote computers over the Internet.

Firewall
A firewall typically guards an internal network against malicious access from the outside. May also be configured to limit access to the outside from internal users.

Freshness
The length of time that an object can be cached and used without an up-to-date or revalidation check.

Gateway Device
An inter-networking device that joins two networks together. Specifically, a device that acts as the intermediary between the Local Area Network (LAN) and the WAN connection to the Internet. The Gateway may be physically be a Smart Cache, or is often a router unit with extra functionality.

Gateway Functionality
The Gateway device as a minimum will provide simple routing functionality, but often has much higher-level knowledge and capabilities with the applications being utilised. May also incorporate a *firewall*, access controls and *content filtering*.

Hierarchical Proxy Caching
The process of configuring Web servers to communicate with each other to determine whether documents missing from one cache might be present in another. In this environment an unlimited number of caches and users can cooperate using industry-standard Internet Cache Protocol (ICP) to speed access to shared content and eliminate redundant Web server requests.

HTML (Hypertext Markup Language)
The coding language used to create *Hypertext* documents. A set of markup symbols or codes is inserted in a file that tells a Web browser how to display a Web page's words and images for the user. The "hyper" in Hypertext comes from facility of HTML to additionally specify that a block of text, or an image, is linked to another file on the Internet

HTTP -- (HyperText Transfer Protocol)
The protocol for moving hypertext files across the Internet.

HTTP headers
A newer version of HTTP – version 1.1 – has headers that provide more control for caching content more efficiently.

HTTPS (Hypertext Transfer Protocol over Secure Socket Layer, or HTTP over SSL)
Web protocol built into browsers that encrypts and decrypts user page requests as well as the pages that are returned by the Web server.

Internet Cache Protocol (ICP)
The protocol used for querying proxy servers for cached objects.

ISP (Internet service provider)
A company that provides individuals and other organisations access to the Internet and other related services such as Web site building and virtual hosting.

LAN (Local Area Network)
A LAN spans a limited local geographical area such as a school campus. LANs typically use Ethernet technology and are usually an order of magnitude faster and

more capable than the WANs that are used to connect then to services such as the Internet.

Origin Server
The web server that hosts a particular resource, such as a web page.

Overlay Networking
The creation of an additional network to carry bandwidth-intensive applications and relieve the burden on a network. Overlay networks can be achieved using a number of different transmission media, from ISDN and ADSL to cable and satellite

Pre-Caching
Loading the cache storage area with content prior to when it is needed. Typically this is loading activity conducted out of prime time e.g. overnight when free bandwidth capacity is available.

Pre-Loading (of WebShare)
Also known as *Web Mirroring* see below.

Proxy
A device whose IP address is specified as a configuration option to the browser or other protocol program in order to handle Internet service requests between web clients and servers.

Proxy Auto-configuration (PAC) file
A file used to configure the IP address and port of the proxy server in a browser.

Proxy server
An intermediary server that accepts requests from clients and forwards them to other proxy servers, the origin server, or services.

Ready Video
Video-on-demand, stored within the Video Server.

Router
A device that physically joins multiple networks – maybe a combination of *LANs* and *WANs*. Also acts as a traffic forwarder – as it understands the where network traffic has come from and where it is destined.

Secure Shell (SSH)
A program to log into another computer over a network. SSH provides authentication and secure communications over insecure channels.

Secure Sockets Layer (SSL)
A security protocol used for sending encrypted information for Internet transactions and communications between client web browsers and web servers.

Smart Caching
Equiinet terminology to encapsulate all its caching techniques built into a modern caching solution for content delivery in educational environments. This also includes

a range of Internet *Gateway* functions as well as sophisticated pre-caching.

Smart Caching Protocol (SCP)
SCP a proprietary Equiinet protocol based on the draft RFC BTFTP, enabling Predictive and Reactive cashing and re-transmission techniques.

Static web page content
Objects within a web page that are flagged as rarely changing, i.e. objects that can be cached – opposite to *dynamic content.* Examples include logos and graphics.

Structured data store
The ability to faithfully replicate the directory and file structure of the source content, which will enable the speedy retrieval and delivery of the content without unnecessary searching.

Traditional Caching
Functionality provided by a first generation caching device. Will have the ability to automatically cache HTML and FTP. It may have some ability to preload but only into cache, and has none of the content control and storage, and additional access control and gateway features of a *Smart Cache*.

Uniform Resource Locator (URL)
The address of a file or resource accessible on the Internet, which contains the name of the protocol required access the resource, a domain name that identifies a specific computer on the Internet, and a description of a file location on the computer.

WAN (Wide Area Network)
A WAN spans a large geographic area, and is typically used to interconnect multiple LANs. WANs typically use communications media such as leased lines, ADSL, ISDN etc., which are all relatively slow compared to *LAN* technology.

Web Mirroring
The ability to replicate entire web pages from an origin server and place them into a local *WebShare* area on the Smart Cache. Issues such as creating a localised relative web page addressing structure, rather than an absolute or relative addressing structure relating to the origin server are also resolved.

Web Server
A server that retrieves and delivers requested web pages to users who enter the URL in a web browser.

WebShare
An area reserved on the Smart Cache reserved for storage of local content. This may be pre-loaded web pages from a specialist content provider, or it may be content created as a result of *Web Mirroring*.

Acknowledgements

This document references :

'Boundary Caching' - First published August 1999 (V1.0), by The NSS Group.

Full version is available on :
http://www.netpilot.com/uploads/97/Boundary_Caching_WP.pdf

'Caching Tutorial for Web Authors and Webmasters' – Published 1998-2003 by Mark Nottingham

Full version is available on:
http://www.cachepilot.com/news/whitepapers.asp