# Caching @ Novell
## A Beigepaper

by Grettir Asmundarson (grettir@neticus.com)
Last Revised: January 31, 2000

A Service Of Novell's Information Services & Technology Global Technical Architecture Group

## About The “@ Novell” Series

Most documentation starts as hastily scrawled notes from sleep-deprived developers who weren’t necessarily hired for their keen communication skills. Those notes are then fleshed out by recently graduated English majors who have spent their last four years immersed in works of fiction. The results are then passed on to the marketing department whose job it is to make sure that no word or phrase, even if it’s true, will reflect unfavorably on the product (“I don't think that the word ‘Basic’ properly communicates the exciting nature of the product. Why don’t we call it ‘Visual Zesty!?!’”). It is then beset by lawyers who finish the job by making sure that they haven’t explicitly promised that the product will actually do anything.

By the time the documentation gets into your hands, it has been so sanitized for your protection and generalized beyond recognition that you usually have to go out and buy a 3rd-party manual (that was, more likely than not, written by the same non-technical technical writer who wrote the original documentation) in a vain attempt to get an unbiased, unexpurgated, and/or unfiltered view of just how you’re really supposed to use the stuff.

That’s where the “@ Novell” series comes in. Rather than the vague, generalized, and wholly fictional examples found in most documentation, we’re going to tell you exactly how we use our own products to run our own company. After all, we are not a small, tidy computing environment suitable for documentation. We are a big, sprawling, untidy environment made up of over 500 production servers and 20,000 workstations in 130 locations throughout the world. In other words, we’re probably an awful lot like you.

And it’s not that we’re necessarily any smarter than you are, we just have a distinct advantage. By the time you get your hands on one of our released products, we’ve already been using it to run our business for quite some time. For instance, a month before NetWare 5 shipped, well over half of our 500 production file servers had already been upgraded to NetWare 5. (Keep in mind that these were production servers. These were not test servers that we had safely tucked away in antiseptic labs. These were real-world servers in a real-world environment solving real-world problems.) And two months before NetWare 5 shipped, we’d already converted one of our buildings to IP Only. That means that we’ve probably gained some insights into implementing our products in a big, sprawling, untidy environment, and this paper is an attempt to share those big, sprawling, untidy insights with our customers.

But keep in mind that this document may be a little rough. It wasn’t conceived by a committee, written by a committee, or approved by a committee, so it hasn’t been edited, re-edited, tidied up, sanitized, and whitewashed. Don’t think of this as an official whitepaper. It’s more like a beigepaper.

## All Right, *You* Try Making Caching Funny!

It was like a bad dream. “We want you to write one of those beigepaper things you do entitled ‘Caching @ Novell.’ And we want it to make it funny like those other ones.”

Caching? Funny? You've got to be kidding me. As if I just can just reach into my enormous cache of caching jokes and pull out a couple of zingers:

> "…then, after a long pause on the other end of the technical support line, the customer said, "'Proxy cache?' I thought you told me to 'epoxy the cat!'"

Or how about:

> "…so she turned around and said 'I'll show you a SOCKS client!' And she punched him right in the nose.

It doesn't look very promising, does it? But we'll forge ahead and hope for the best.

## A Not-So-Silly Market Segment

A few years ago, our main Web site (www.novell.com) underwent a major face-lift. This was back in the days when every Photoshop user in the world simultaneously discovered beveled buttons and drop shadows. So, in a move that would have horrified Jakob Nielsen, our Web designers nearly quadrupled the size of every Web page on the site.

At the time, our Web servers were already straining under the existing load, so the chances of them being able to handle four times that load were pretty slim. The hardware geeks had the perfect solution. "What? You need to handle four times the load? Well, that's easy. We'll just quadruple the number of Web servers! Or, better yet," they salivated, "buy us new servers that are four times as powerful." That's a very typical hardware geek response: there's no problem that can't be solved by throwing more hardware at it.

But cooler (and smarter) heads prevailed when it was pointed out that we were in the process of developing a product, BorderManager, which was perfectly suited to this kind of problem. So, we dashed the hopes of the hardware geeks by not buying any new hardware at all and simply re-deploying two of our old NetWare servers as reverse proxies in front of www.novell.com.

In all honesty, I should admit that when I first heard that some of our brightest developers were working on "caching," I thought it sounded like a perfectly silly market segment to be dabbling in. I mean, come on! There wasn't a caching *market*, let alone a *segment*.

But when we flipped the switch on the new Novell site, just as we'd expected, our hits went from one million a day to four million a day…but the load on our Web servers actually *decreased*. And even though they were handling almost 90% of the requests, the BorderManager servers weren't even breathing hard.

From that moment on, I was a true believer. And I was reminded once again why Fortune magazine hasn't called me for an interview for their "100 Technological Visionaries" issue.

## Reverse

Since those humble beginnings, things really haven't changed that much other than the fact that our Web pages are bigger than ever. We've also increased the number of BorderManager servers that we use on www.novell.com, but not because our loads have increased (which they have). We could certainly get by with two or three reverse proxies, but since we have a bad habit of running alpha and beta code in our production environment, we need a few more to act as a cushion should something go wrong with the servers running alpha/beta code (which never happens, I promise).

We also have two reverse proxies located at Digital Island in Honolulu. Digital Island has some very fat pipes that are only a single hop from many major hubs, so those two reverse proxies handle most of the traffic to www.novell.com that is routed from the Far East and even parts of South America and Europe, depending on which route is closer/faster/more efficient[1].

Using reverse proxies means that we don't actually need to mirror our sites. Why go to the trouble of maintaining two different sets of data and managing multiple Web servers when a reverse proxy will take care of it for you…and do it faster?

Another advantage of using reverse proxies is that we are able to put the BorderManagers out in the DMZ and have the Web servers themselves located safely inside our firewall. When the Web servers resided inside the DMZ, we had to jump through all sort of hoops to allow the Web servers in the DMZ to securely communicate with our internal databases. But now that we don't have to jump through so many hoops to ensure secure communications, we're able to expose more services to the outside world that we might not be willing to expose otherwise.

To improve the performance of our reverse proxies, we allow persistent connections between clients and the BorderManager servers, and between the BorderManager servers and the Web servers. This can save time since the clients and servers don't have to keep closing and reestablishing TCP/IP connections. And these reverse proxies take advantage of "cache clustering," so if one of them falls over (and can't get up) another will simply pick up its IP address and service those requests along with its own.

We've also found reverse proxies to be quite helpful when we're rolling out a new Web site. We can build the site and do all of our testing on a different Web server. Then when the time comes to put the new site online, we just flip a switch and tell the BorderManagers to start pulling content from the new Web server. It's instantaneous. Should something go terribly wrong (which never happens, I promise) we could flip the switch again and have the reverse proxies pull data from the old Web servers again.

[1] Wouldn't it be nice if airlines routed *you* to Hawaii when it was closer/faster/more efficient.

## Forward

We have at least one forward proxy located at every site at Novell, no matter how small the site. This has done a great deal to cut down on our WAN traffic and has significantly accelerated the Internet access of the people in some of our far-flung field offices. Trust me, anything you do to significantly accelerate Internet access for people on the other end of a 56k WAN link makes you an instant hero.

We configure all of our browsers with the hostname “proxy” specified as the proxy server. We could hard code all of the browsers in Provo to use “proxy.provo.novell.com” but that means that if you go to Hong Kong and forget to reconfigure your browser, you’d be pulling information from a proxy server across a 56k WAN link, which sort of defeats the purpose. By simply specifying the hostname “proxy,” the client will append your DHCP-assigned sub-domain name to that hostname to come up with the fully-qualified DNS name. For instance, if you were in Hong Kong, the hostname “proxy” along with the DHCP-assigned sub-domain name “hkg.novell.com,” would become “proxy.hkg.novell.com.” which just happens to be the name of the local proxy server.[2]

We are actively also using transparent proxy in a number of situations. With transparent proxy you can insert the proxy server “in the flow” or tell your routers or Level 4 switches to send all outbound requests on port 80 to the BorderManager, and everyone gets accelerated without even knowing it. And if you’re doing content filtering, everyone gets filtered without even knowing it, too. (Although some people will probably find out very quickly…)

And speaking of centent filtering, I’m often asked if we’re doing any content filtering on our forward proxies here @ Novell. Even though we have the ability to do content filtering with BorderManager, we’re not currently employing it anywhere. Of course, that could change at the drop of a lawyer’s hat…

Another thing some folks might find interesting (in a caching hierarchy sort of way) is the fact that we’re using a CERN hierarchy internally, rather than using ICP. In a CERN hierarchy, there is a single parent and if a proxy server doesn’t find the requested page in its cache, it will go request it from the parent. If an ICP-based proxy server doesn’t have the requested page in its cache it goes out and queries all of its parents and its peers to see if they have a copy. If they don’t, then those parents and peers go query their parents and peers, and so on, and so one. This can make ICP a little “chatty” and, having had a little experience with “chatty” protocols in the past, we’ve chosen not to use it internally.

[2] We could jump through a number of different hoops attempting to reconfigure all of our clients/browsers/applications when someone travels to a new location, but why bother? Contrary to the current philosophy of a large portion of the industry, the simplest solution is almost always the best solution.

## Caching In A Cheerios Box

A while ago, Novell announced the release of the Novell Internet Caching System (ICS). Basically, ICS is a cache "appliance" that is so easy to use that you could put it in a box of Cheerios and have kids install it at their homes. They'd just open the box of Cheerios, pull out their shiny prize/cache appliance, plug it in, boot up, answer a few questions, and they'd be off and running.

Compaq, Dell, Fujitsu, Hitachi, IBM, NEC, and Toshiba (among others) have all announced or released products based on ICS. I've seen quite a few of them and they're pretty sexy (in a cache appliance sort of way).

The first thing might notice about ICS is that it is very un-NetWare-like. You might not even recognize it as NetWare. It doesn't use the traditional NetWare file system, or even NSS. It uses it's own file system that is optimized for writes, because on a forward proxy server writes are the bottleneck. And since this box is dedicated solely to caching, the caching process doesn't have to be well behaved and play well with others. It just takes over the box. It grabs every ounce of RAM it can and plows ahead. It is truly amazing in its single-minded efficiency[3].

To configure the box, you can either telnet in or you can use a Java applet to do the administration. And if you want to nuke the system, you can boot up off of the CD, re-image the appliance, and you can start from scratch in a matter of minutes. And code updates a breeze. You simply point the ICS box to a Web server that has the updated code and the ICS box will upgrade itself over the wire.

ICS was released after BorderManager 3.5, so you may notice that ICS has some capabilities that aren't yet available in BorderManager. For instance, ICS has sophisticated clustering capabilities that provide support for failover, cloning, and load balancing. It's sophisticated enough that it can measure the horsepower of the individual ICS appliances in the cluster and the different services that are being cached, and assign your most powerful ICS appliance to your most resource-intensive service. ICS also provides a different kind of load balancing by keeping track of the current number of "fills" from each origin server and will redirect requests to a less busy origin server if it needs to.

ICS also supports multi-homing, has support for WPAD, and has been "Akamai-zed" for your protection. Yes, folks, that's an adjective that I actually heard a marketing drone use. it basically means that ICS will interoperate with the Akamai network to provide even faster delivery of "Akamai-zed" content. And while we use BorderManager to cache most of the

[3] Speaking of single-minded efficiency, I have to say that the developers of BorderManager and ICS are amazing. I haven't seen a group of developers so dedicated to ruthless efficiency since the days when people used to brag about being able to shave yet another line of code from an assembly language program. I was once listening in on a conversation between two ICS developers when one of them suggested a change in the way the code was written. The other developer's eyes grew wide and (in an over-my-dead-body sort of way) he said, "But that would require me to 'go to disk'." Anything that would slow his code down was beyond the pale.

content on www.novell.com, we've recently deployed an ICS box as a reverse proxy for our download servers because ICS supports resumable downloads if you're using a download manager (like Netscape's SmartDownload or GetRight).

And last, but not least, ICS can use content filtering services like N2H2 or X-Stop…which is a feature you'd want if you were going to include it in boxes of Cheerios, right?

Of the proxy servers at our main sites, we're probably split about 50/50 between using BorderManager and ICS. With its advanced and single-minded caching feature set, ICS makes sense in a lot of situations. But BorderManager, with its caching, firewall, VPN, and RADIUS capabilities makes more sense in others. Everyone has specific needs and different tastes.

In other words, if you don't like Cheerios, we'll probably start shipping BorderManager in boxes of Wheaties.

## Acknowledgments

Nothing in this beigepaper represents original thought on my part. I couldn't have written a word without the generous help and input from everyone in Novell's IS&T Global Technical Architecture group. (I'd name them all individually but they'd probably get spammed.)

Send all comments, questions, corrections, and/or complaints to:

grettir@neticus.com

Tasty baked goods can be sent to:

Grettir Asmundarson
PRV-C-122
122 E. 1700 S.
Provo, UT 84606

And please note that Grettir Asmundarson is just a ridiculous pseudonym, so don't bother trying to call. You'll only confuse our receptionist.