AUUGN

The Journal of AUUG Inc.

Volume 25 • Number 2
June 2004

Features:

News:

Regulars:

Handle with care and do not fold. This magazine contains fragile CD-Rs.

ISSN 1035-7521

Print post approved by Australia Post - PP2391500002

AUUGN

The journal of AUUG Inc.

Volume 25, Number 2
June 2004

Editorial

The new face of AUUGN

Greg Lehey <Greg.Lehey@auug.org.au>

It's already obvious: the AUUGN you're reading now looks very different from previous issues. The whole layout has changed.

That's not all that has changed: I regret to report that Con Zymaris has resigned as editor of AUUGN. Those of you who know Con will understand that he has a few more things to do than just edit AUUGN: go just about anywhere in the Open Source industry and you'll find that Con has been there before. It's not surprising that he has had to cut back on some activities, and unfortunately AUUGN is one of them. I'd like to thank Con for his revitalization of AUUGN over the last three years.

So who's the new editor? We don't have one. I'm doing this issue, but I don't intend to become the editor. Like Con, I've recently come to the conclusion that I don't have enough time for all the things I do. If you don't already know, you'll see in a couple of pages that I have resigned from the position of AUUG president, and I have also resigned from the FreeBSD Core team. These are consequences of this realization. Given the current pace of the industry and the lack of time available for volunteer activities, there's a good chance that nobody will have enough time to volunteer.

There's another factor too: in its meeting on 30 July 2004, the AUUG board of directors voted to stop production of the paper version of AUUGN by the end of the year. I am personally not sure that this is a wise move, and I voted against the motion. But ultimately it's up to you, the reader: do you still want a paper version of AUUGN? It's quite expensive: about half the membership fees go to printing (not producing) AUUGN. By stopping printing of the paper version, we could roughly halve our membership fees, or to put them to other good use.

Currently we're undecided whether to continue to publish AUUGN in electronic form. We will do so for a while, including it on the quarterly CD-Rs, which will continue. If you want a paper version, there are a couple of things you can do:

- Print out the PDF file on the CD-ROM.
- Get us to have it printed and bound for you.

Currently we have no idea what the latter option will cost, and we won't investigate unless somebody is interested enough in this alternative. So: if you're interested, please contact me (Greg.Lehey@auug.org.au) and tell me so.

Assuming that AUUGN continues publishing, I'd like to suggest an editorial team (dare I say "committee"?) of about four or five people who could share the load. If we go this route, I'd be prepared to be part of the team.

Another question is the content of AUUGN. Before Con came on board, it was mainly homegrown, and sparse. That changed dramatically with Con. Much of the content came from the web. I personally found this positive: Con searched the web for interesting content, so I didn't have to do it. Others asked whether we couldn't do more ourselves. AUUG creates a significant amount of original material through the papers at the annual conference. They're interesting, and most of the material isn't available anywhere else. One of the things that the board of directors has been discussing for some time is the republication of papers from the proceedings. We're doing that for the first time this quarter. I'd be very interested in feedback on the idea.

AUUG Membership and General Correspondence

The AUUG Secretariat

AUUG Inc
PO Box 7071
Baulkham Hills BC NSW 2153
Telephone: 02 8824 9511
or 1800 625 655 (Toll-Free)
Facsimile: 02 8824 9522
Email: auug@auug.org.au

AUUG Business Manager

Elizabeth Carroll
AUUG Inc
PO Box 7071
Baulkham Hills BC NSW 2153
busmgr@auug.org.au

AUUG Board of Directors
Email: auugexec@auug.org.au

President

Greg Lehey
PO Box 460
Echunga, SA, 5153
Bus. Tel (08) 8388 8286,
Mobile 0418 838 708,
Fax (08) 8388 8725
Greg.Lehey@auug.org.au

Vice-president

David Purdue
Sun Microsystems
Level 6, 476 St Kilda Road
Melbourne, VIC 3004
Phone: +61 3 9869 6412,
Fax: +61 3 9869 6288
David.Purdue@auug.org.au

Secretary

Adrian Close
Cybersource Pty.Ltd.
4, 10 Queen Street Melbourne VIC 3000
Business Telephone: +61 3 9621 2377
Business Fax: +61 3 9621 2377
Mobile: +61 417 346 094
adrian@auug.org.au

Treasurer

Gordon Hubbard
Custom Technology Australia Pty Ltd
3 Spring Street,
Bus Tel: 02 9659 9590,
Bus Fax: 02 9659 9510
Gordon.Hubbard@auug.org.au

Ordinary board members

Jonathon Coombes
Cybersite Consulting Pty Ltd
34 Newcastle Road, Wallsend NSW 2287
Business Telephone: +61 2 4965 6989
Jonathon.Coombes@auug.org.au

Andrew Frederick Cowie
Operational Dynamics
GPO Box 4339
Sydney, NSW, 2001
Telephone: +61-2-9977-6866
Andrew.Cowie@auug.org.au

Steve Landers
Digital Smarties
PO Box 717 Willetton WA 6155
Business phone: +61 8 9313 6868
Business fax: +61 8 9313 6077
Steve.Landers@auug.org.au

Stephen Rothwell
IBM Australia
Linux Technology Centre
8 Brisbane Ave
Barton ACT 2600
Business Telephone: +61 2 62121169
Stephen.Rothwell@auug.org.au

Michael Still
Phone: +61 414 382 568
mikal@auug.org.au

AUUG Incorporated gratefully acknowledges the support of its corporate sponsor:

IBM®

President's Column

Greg Lehey <Greg.Lehey@auug.org.au>

Change of guard

Two years ago, almost to my surprise, I became President of AUUG. This year I decided not to reapply for the position, and by the time you read this, David Purdue will (again) be President. I wish David every success.

There's an unwritten (almost unstated) rule that AUUG officeholders should keep their post for three years. Well, not so much a rule as a recognition that it makes good sense: it takes a while to get used to the office, and once you get into the swing of things, everything's easier.

As a result, the current situation reminds me of the time I visited the Great Wall of China at Badaling. It was bitterly cold, light snow was falling, the way was steep, and about two-thirds of the way up, I asked my guide if there was anything of great interest at the top. "No", she said. So we turned back. When we got to the bottom, she said (and this in 1997!): "Chairman Mao used to say that a Real Man would go to the top".

My current position with AUUG reminds me of that incident. I'm left with a bit of the feeling that I didn't go the distance. So why did I turn back? The job's getting harder as time goes on. Look round this edition of AUUGN. The good news is that almost all of it is original AUUG content, but the bad news is that about every other article bears my stamp.

This isn't a criticism of Con: he has done a valiant job. I suspect that part of his decision to give up editorship of AUUGN stems from the same problem. Fewer and fewer people are doing the work of running the organization. Only one person is paid to do AUUG work: Liz Carroll, our business manager, without whom we'd be completely stuffed. The rest of us do it when we have time. In the last few years, time has become an increasingly rare commodity, and it has shown both in the number of people available to do anything at all, and in the amount of time the remainder have left they need to do both the work they used to do in a more leisurely fashion, and also the work for those who no longer have any time at all—quite a problem.

What do we do? A lot of the problem is out of our hands. We're going through lean times, not for the first time in our history. Ask anybody else and you'll hear the same story. The good news is that such periods seldom last very long; until they're past, we'll have to muddle through as best we can. The more people who can help, the easier it will be.

From a personal point of view, I'm not leaving AUUG, nor even the Board of Directors. According to the constitution, in the coming year I'll be the Immediate Past President. Hopefully in the next twelve months I'll find enough time to contribute to AUUG. Why don't you do so too? AUUG lives from its events, but in the last few years their number has diminished. It only takes a couple of people with an idea, along with the help of Liz Carroll, to set up a one-day seminar on some topic of interest. Believe me, it's fun.

About AUUGN

AUUGN Editorial Committee

The AUUGN Editorial Committee can be reached at auugn@auug.org.au.

Send physical mail to the following address:

AUUG Inc
PO Box 7071
Baulkham Hills BC NSW 2153

Acting Editor: Greg Lehey

Contributors:

Thanks to the following people for contributions to this issue: Grant Allen <Grant.Allen@towersoft.com.au>, Frank Crawford <frank@crawford.emu.id.au>, Enno Davids <enno.davids@metva.com.au>, Ben Elliston <bje@air.net.au>, Greg Lehey <Greg.Lehey@auug.org.au>, Luke Mewburn <lukem@NetBSD.org> and Michael Paddon <mwp@qualcomm.com>

Public Relations and Marketing: Elizabeth Carroll busmgr@auug.org.au

AUUGN Submission Guidelines

Submission guidelines for AUUGN contributions can be obtained from the AUUG World Wide Web site at http://www.auug.org.au/publications/auugn/subguide.html.

AUUGN Back Issues

A number of back issues of AUUGN are still available. For price and availability please contact the AUUG Secretariat.

Conference Proceedings

A limited number of copies of the Conference Proceedings from previous AUUG Conferences are still available. Contact the AUUG Secretariat for details.

Mailing Lists

Direct enquiries regarding the purchase of the AUUGN mailing list to the AUUG Secretariat.

Disclaimer

Opinions expressed by the authors and reviewers are not necessarily those of AUUG Inc., its Journal, or its editorial committee.

Copyright Information

AUUGN is the journal of AUUG Inc., the organisation of UNIX™ and Open Source Professionals. Its interests include promoting knowledge and understanding of Open Systems, including, but not restricted to, the UNIX operating system, user interfaces, graphics, networking, programming and development environments and related standards.

Contribution Deadlines

Vol 25, No 3 (September 2004)	**15 August 2004**
Vol 25, No 4 (December 2004)	**15 November 2004**
Vol 26, No 1 (March 2005)	**15 February 2005**
Vol 26, No 2 (June 2005)	**15 May 2005**

Measuring and Tuning Apache

Enno Davids <enno.davids@metva.com.au>

ABSTRACT

Ever wondered what your web server is up to? If you've ever had to mind a large e-commerce site or even a smaller site where real money is at stake based on your administration skills then you've wondered how well or poorly your servers are performing. You may have sat in the days before Christmas as the load of marauding shoppers sweep past your site wondering how the load on the server compares to that of the day before. Will you have enough capacity or will things go wrong? Is there any tuning you can do to stay ahead of the load?

1. Introduction

Apache is one of the success stories of the modern Internet. At the time writing, the monthly Netcraft survey of the Web suggests that some 60-70% of all web sites are hosted on Apache Web servers and most of those on open source platforms of one sort or another.

Most of these web sites are of course not high volume, but some are. And it is for when you find yourself administering such a site that this paper is written.

This paper divides into a few basic sections. The first is a discussion of how to tune Apache for speed. Like most modern software Apache has lots of options that allows it to be deployed in many different ways. But, it's worth noting that not all of the options that are available make sense to use in all situations. Some are merely poor practice whilst others are downright bad practice.

The next section touches on the things we do in our web content that may or may not be wise. This includes static content, poorly written dynamic content and indeed options that may affect the delivery speed of either.

Finally we'll talk about measuring Apache performance. This is an area that frankly isn't talked about a great deal but should be. Notably most of the tools for measuring Apache performance are offline tools with little, almost nothing, available to provide real-time information. In an attempt to address this lack, I also present a simple Java applet that allows some insight into what your web server is up to.

Before we move on it may be worth briefly offering some advice on how not to read this paper, or perhaps how not to act on this paper. In particular, my advice is to read it all first, then perhaps install the Apache HitMeter applet and then before doing anything else establish a baseline for how fast or slow your web server runs as it is. This will involve you also perhaps doing some log post-processing to determine what the web server's typical workload looks like and will, if nothing else, allow you to judge whether you're getting any benefit from the tuning you're attempting. It may also allow you answer other questions like, how much headroom is there in your server capacity on the load it currently has to deal with, and when do you need to think about upgrades.

2. Tuning Apache

As we said up front, Apache, like much modern software, has lots of knobs you can twiddle with. Some can be beneficial in all environments, some in most and some will depend on what you are doing as to whether or not they offer an advantage or not. Let's look at some of the things you can consider.

2.1 Apache Modules

Apache is a modular web server these days. In its base it provides a framework for web serving functionality and then various extra features are attached to that base in a modular manner. Whilst a lot of software uses metaphors like this, it is good software engineering practice after all, in the case of modern versions of Apache it's also a physical reality of the way the software is structured. The extra functions are physically separate code modules which can be completely de-configured from the server if their function is not required.

This then is the first place to look for performance gains. There are lots of Apache modules available. Certainly you will not have a use for all of them and that's even if we're just talking about the base set of modules that come in the base server distribution. We can say this with certainty as some of them have overlapping function and it's a certainty that you don't need two, three of four ways of doing the same thing (with the inevitable opportunity for misconfiguration to result in behaviour you don't expect in some circumstances).

As it happens, each module you add to Apache will also cost you a slight performance penalty. So, it makes good sense to evaluate those modules with a much more jaundiced eye and weigh the benefits of possibly occasional use against the very real costs of their presence even when they're not used. The other obvious penalty for having modules in Apache is that they cost you memory. The more modules you add, the bigger the server binary image is when it's sitting in memory and the more main memory it will consume. This is true of both the code of the server (which is likely shared across all the server instances) and the dynamically allocated data the server uses. Now consider that a busy server can easily have hundreds of Apache processes and the problem is exacerbated.

It's also worth noting in passing that it's good security practice to eliminate unused modules from your server as that way you will avoid being compromised by any exploits which are built around functions of those modules. (i.e. if it's not in the server, an attacker can't exploit bugs in it...)

In older Apache installations, the module mix of the server was determined by the builder of the server software at compile time and was hard to change. Typically, some form of requirements analysis drove the compile time decision making process to include or exclude various modules. In more recent times, the dynamic linking/loading facilities of Apache have allowed modules to be dynamically loaded or excluded at run time from the *httpd.conf* file. This can take much of the perspiration out of the task of choosing what is and isn't needed in the server, allowing new modules to be added in as a requirement for them is identified and perhaps more significantly allowing modules for which the requirement has passed to be de-configured.

2.2 Customized Web Servers

One of the outcomes of this process of course is that some combinations of modules actively undermine each others performance profiles. Think here of the difference between a server which has been optimized for speed against say a server carrying the mod_perl module to support dynamic content. One is lean and mean and the other has a single additional module with a 5MB to 10MB memory footprint.

The solution may be to run two or more quite different instances of Apache with each tuned to some combination of requirements and resource use. Good examples of this might be:

- a web server with no CGI support and only static files for fast image serving
- a web server with only servlet support
- a web server with SSL support
- a web server with *mod_perl* or *mod_php* for dynamic pages

Indeed the top bullet point there is a favourite of sites that do a lot of graphic heavy web pages. (Yes they often have a lot of flesh tones in their imagery...) In some instances, people run completely different web server software in some roles, specifically because of a perception that they are better suited to those roles. To my mind, with proper tuning and on capable hardware Apache can achieve maximal performance for you and there is often no gain and considerable disadvantage to complicating your life with two different products (and two sets of bugs, and two sets of testing, etc...). For anecdotal evidence, entry level Sun UltraSPARC II based systems can easily saturate 100Mb/s Ethernet with Apache and unencrypted content. (More on this later...)

It also bears noting that multiple web server instances can be configured on single systems so long as different IP address or ports can be used. Or they may be spread across different host systems to isolate them. This latter allows overall system loads of the different tasks to be more easily determined and tracked.

2.3 Host Name Lookups

One of the earliest performance problems people identified for Apache was in its abilities to log accesses. Typically, each access to the web server, that is, each `HTTP GET/POST/HEAD` request is recorded as a single line of one or more log files. Each line identifies who the request comes from as well as other data of interest such as when the access was made, what was accessed, the success or failure of the access, how many bytes were transferred and possibly more...

The identity of the requester was written as the system which made the request. To do this the server had to do a reverse name lookup of the IP address of the network connection that request arrived on. This reverse lookup takes time and the server process/thread stalls while it completes.

To alleviate this problem, a directive HostNameLookup was added to the *httpd.conf* file allowing this behaviour to be turned off and on. Typically, on many intranets, where reverse lookups can be reasonably quick and loads are lighter it is left enabled and it is disabled for servers attached to the wider Internet where this may not be true. When the service is disabled, the same log information is still written, except the symbolic name is replaced by its IP address. This can result in significant improvement in the response times of Apache servers and the expected increase in total throughput this leads to.

To enable log files to be examined by humans, a program called *logresolve* is distributed as part of the Apache tools, which can post-process log files to replace the IP addresses with the more human friendly names that come out of the reverse DNS lookup. As this process is performed offline from the process of serving pages it does not affect the servers performance. Failing this, most log analysis packages will now also attempt to perform the IP address to name lookups if required.

Finally on the subject of name resolution we should briefly note that some configuration directives can also accept names rather than IP addresses as part of their syntax. Most of these are resolved once at start up and have little impact on the run time performance of the web server as a whole. Some though can cause greater penalties than might be immediately obvious. Of note here is the `Allow from xxx` and `Deny from xxx` syntax which can be used in Directory and Location contexts amongst others. Given a line like `Allow from metva.com` for instance, any access to the resources being thus protected will require a reverse DNS lookup of every access to see if it is in the allowed domain. In fact, the documentation tells us that a forward lookup of the resolved name is also performed as a double check. The use of IP addresses or ranges will not incur this performance penalty and may thus be preferred.

2.4 Server Pool Management

(It should be noted that this paper deals for the most part of with the 1.3.x stream of Apache releases. One of the notable differences in the newer 2.x server stream is the use of lightweight threads, which will likely invalidate much of the following discussion.)

Conventional server systems such as those managed by the *inetd* process, run by waiting on an open socket and when a client connects forking a new child to handle the request. By its nature this means that the request must wait while a new process is created and for the new process to run, handle the request and answer. Often times, the child process then exits, returning its resources to the system.

The problem with this approach is precisely the delays associated with starting a new process and the effect this has on the throughput the client process can achieve. The common solution is to pre-fork the server child process(es) before the requests arrive, and then hand off each request to an idle child process. That is, a pool of servers is maintained with a master to coordinate the processing of requests by the children.

Since its earliest versions, Apache has had the ability to manage such a pool of server processes. Rather than manage a fixed size pool of servers, Apache can maintain a variably sized pool where more servers are started as cumulative load increases and excess idles servers are shut down as this load decreases.

In fact, Apache performs this dynamic sizing of its server pool fairly well without a great need for tuning. There are some controls which can be tuned for sites/servers which are either exceptionally busy or exceptionally idle, but other than these degenerate cases not much need be done.

The most important controls are the MaxClients configuration and the depth of the TCP listen queue. These affect how much load an Apache instance can deal with. `MaxClients` limits how large Apache can grow the pre-forked server pool and together with the TCP listen queue depth this controls how many incoming connections the web server can accommodate before failing completely. Typically settings are chosen to reflect the maximal capacity of the server host rather than sizing things according to expected load or other such, likely poor, guesses.

2.5 Content Negotiation

One of the abilities Apache has picked up is an ability to negotiate content types with a client browser. Most typically, this is used to select between equivalent sets of related content and most often languages are the offerings that are selected between. (i.e. view a website in English or German or Japanese or whatever).

The problem of course is that negotiation costs time and this translates to an overall performance hit. Not only does negotiation cost time, but the server itself has to read the filesystem, often scan directories, on each access merely to determine which content types are available to offer.

To achieve higher performance then, the solution sadly, is to forgo the content negotiation abilities of the server and revert to offering single content websites. If multiple content types are catered for (and clearly this is a good thing) then checking once in a server side script and using redirects to send the user to a single content type site or indeed allowing the user to explicitly choose between content types themselves. This latter must usually remain an option anyway as few users set their browsers correctly to choose between multiple content types correctly and users may choose to sample other content types when they feel a need (e.g. to understand a poor translation...).

2.6 Page Access

It's worth briefly noting that Apache has some options which although very useful result in extra processing for each page hit and may need to be disabled for high performance servers.

Firstly, the *.htaccess* file allows the content provider(s) to override the access privileges of the content under each directory tree. This behaviour is enabled by the `AllowOverides` option to Apache. The difficulty though is that to process a request the server must now check for a *.htaccess* file (or whatever name has been configured) in each directory on the path to the page to be served and potentially process the contents of the file. As this processing is repeated essentially for each request, the overheads are quite high.

A similar issue exists with symbolic links. Apache may be configured to either follow symbolic links or not to. The latter is of course common advice for best security practice. The problem is though that in order to preclude symlinks the server must once again check each element of the path it is traversing to ensure that all are regular directories or a file (for the final element). Once again a high per request overhead. As ever, we must choose between more security or more performance.

2.7 Caching Pages

Newer releases of Apache offer an ability to cache specified content pages or components. To do this the `MMapFile` directive may be used to specify page elements which the server is to cache in memory. This content is then available for immediate use rather than needing to be fetched from the filesystem first.

To make use of this facility the *mod_mmap_static* Apache module must be loaded and then the `MMapFile` directive is available in the configuration file. Typical use is `MMapFile` <*path*> with a complete path to the file to be cached. Note though that if the file is updated, the Server must be restarted to refresh the in-memory copy.

To use this optimization well, the administrator must have some knowledge of the content that is being served. Typically most log analysis packages can give you a reasonable idea of which pages and page elements are accessed the most, which are those for which the most benefit can be obtained through caching. A good first approximation is to look for page elements that are common to many areas of your content such as organizational logos or navigation bars or parts thereof.

2.8 More Hardware?

A final and perhaps most obvious way of speeding up your web server is of course to buy a more capable server host system or to upgrade one or more portions of the host your server is currently using. Common solutions here are faster machines (e.g. higher clock rates), bigger machines (e.g. more CPUs), more machines (e.g. clusters) and off-board extras such as load balancers, crypto accelerators and reverse proxies.

The biggest caveat here is that as the number of systems increases the administrative load also rises. In fact, there can be real issues around the high workload involved in keeping multiple systems identically configured and the content on them synchronized. Once the number of systems is more than a couple, the implementation of automation to assist with this is highly desirable.

Not all of this needs to be bad news though. Clustered servers with redundant load balancing hardware can offer good fault resilience and may form the core of a high availability server farm.

2.9 Newer software

As noted earlier, this paper deals mostly with the 1.3.x release stream of Apache. There is good reason to expect that considerable performance gains will be realized by the change to the lightweight thread model that the Apache 2.x stream offers. In fact, there may well be other performance benefits which result from the re-engineering efforts being undertaken by the server development group.

The Apache Group now advises, I believe, that the the 2.x stream is ready for use in production environments. The sticking point, as is so often the case, may well be ‘other’ software which you are relying on which must work in conjunction with Apache which may not yet be aware of the 2.x stream servers. Examples here would be special purpose Apache modules, databases, servlet environments or similar.

3. Tuning our content

The next most obvious place to look at for performance of web based content is the content itself. Note that most often this is a case of not doing dumb things which adversely affect performance.

3.1 Static vs. Dynamic Pages

The first thing to consider is the difference between static and dynamic pages. Clearly many web sites use dynamically generated pages to provide interactivity of some sort. Static pages can of course be served very quickly, rates of 100’s of pages/sec are routine and with appropriate networks 1000 pages/sec may be achieved by a small server. Certainly, these rates are special cases for a LAN and for a real web site the external link bandwidth may well be the rate limiting factor.

Dynamic pages present a very different profile though. Often many factors interact to limit the generation of dynamic pages such as time to load an interpreter (e.g. *perl*), time to search a database, execution speed of a servlet and speed of middleware. All these often combine to drop dynamic page rates into the single figure page/sec regime for the same hardware.

There are a number of strategies which may yield improvement over these issues. The simplest is to pre-generate common dynamic pages, effectively making the content static once again. If some dynamicism is required then it may be sufficient to generate the page at regular intervals (say tracking a stock price at half hour intervals for instance). The biggest effect here by the way is that a single dynamic page generator runs periodically rather than one per server child process. To see the effect of this you need only consider the run time costs of 500 (say) copies of *perl* competing against each other.

The other cost here of course is that while Apache server processes are pre-forked, that command processors like *perl* are being forked at the time of the request again. The solution is to use a dynamic page generator which does not incur the process startup costs per page impression. Most typically, we would choose to use *mod_perl* or *PHP*, both of which exist as modules inside Apache and hence benefit from the pre-forking of servers ahead of their need once again.

There is of course some small irony here though that after we recommended removal of extra modules to trim the workload and size of the Apache binary and its in-memory footprint, we now add back in the biggest modules that are out there. The moral of this story is stay with static pages wherever you can.

3.2 SSL

SSL is one of those areas that concerns all e-commerce sites. Clearly, it’s not possible to credibly run an e-commerce site without the use of strong encryption. But at the same time strong encryption is by design a computationally intense task (to help resist brute force attacks). This then leads to the situation where on a busy e-commerce server, SSL can be the biggest consumer of the host servers computational resources. SSL hit rates which most servers can offer are generally at least an order of magnitude less than the rate that unencrypted pages can be served.

This then leads to a natural thought for encrypted content, which is to only encrypt those portions of a page that need to be protected. The plan here is to leave things like navigation bars and logos unencrypted because there is no benefit to delivering them in an encrypted manner. Unfortunately, most modern browsers, in an effort to protect their users against malicious content or content from incompetent developers will ‘warn’ their users through the agency of a dialog box when pages consisting of mixed, encrypted and plain content are delivered.

As the user is typically in the process of doing something where he or she is being asked to place trust in the web site owner, the effect of this is to unsettle the casual web surfer, typically at the exact moment when this is least desirable for the web site owner. Thus, this solution is not a practical one, as while it increases your system performance it typically also results in business being lost when some portion of the surfing community elects to abandon the transaction they had planned due to the now unsettled state they find themselves in.

It is however possible to deliver only those pages which require strong encryption in such an encrypted manner, leaving say the bulk of an e-commerce site unencrypted and only payment pages or pages exposing users confidential information protected (depending how strong the privacy laws of your jurisdiction are). Once again note that most modern browsers will alert their users when a transition to or from encrypted content occurs. Note also that some sites will find it desirable to deliver extra content encrypted. Most typically we see this in places like the page which renders the credit card entry HTML FORM for the user for instance. Nothing about this page itself requires encryption. The ACTION URL which the form data is sent to needs to be an SSL page, so that the form data is encrypted in transit from the browser to the server. But, web surfers are taught to check the padlock icon or similar status indicator which indicates the use of encryption prior to offering sensitive data such as their credit card number online. Whilst the page itself does not require SSL to be offered to the user, it typically must be so that the user can be re-assured that adequate encryption is being used to protect their private data. Similarly, the transition back to unencrypted pages must be planned so that it clearly takes place after the transfer of this data is complete. (Once again, if this were immediately after the user form data were received, then the user experience would be to submit their form with their credit card data only to see the browser dialog box warning of the transition back to plaintext content.)

Having worked out where and when to use encrypted content, it's worth making sure you don't make your lives any more difficult than necessary. This means that things like the SSL session cache must be sized suitably for the amount of traffic you expect to deal with. Larger sites should employ larger session caches than smaller ones. Sadly, there are no tools or statistics available to guide you in how much you need or how fully the currently configured cache size is being utilized.

In general it is best to err on the side of caution and specify more cache than you may feel is necessary simply because the optimization the cache offers is so great. An aside on SSL will perhaps explain this.

A generic SSL transfer, broadly, consists of two portions. A public key protected set of session keys and the bulk data that these session keys are protecting which typically uses a more efficient symmetric key cipher. The public key portion of the SSL processing is in fact the slowest of the conversation, with the symmetric cipher operating somewhat faster. This of course is the rationale of using the two ciphers to begin with. To further improve the performance of SSL though, a second and subsequent connection from the same browser can specify a session key which was already used in a previous SSL transfer. (With appropriate timeouts of course). Thus, the browser and server can avoid the costs of unnecessary public key cipher operations when they feel they do not need to do incur them. This is the purpose of the SSL session key cache and this of course is why a busy server should have a cache large enough to guarantee that the user who spends a few minutes hunting for his wallet or her purse can still benefit from the cached session key that was stored when their initial SSL transfers were done.

This is also why those of you running load balanced clusters of servers should ensure that you avail yourselves of 'sticky' connections where they are available, but only for SSL content. 'Stickiness' is the notion that rather than distributing load randomly across a server pool, that subsequent connections from the same clients on a network (i.e. the users browser or an ISPs proxy) should be made back to the same server host in a load balanced cluster. This once again ensures that the same session keys can be re-used. Note that absent this, your load balancer could direct each access to a different host incurring the longer, computationally more intense public key exchange of session keys each time. With even two servers only 'hidden' behind a load balancer, the browser could find itself discarding its session key on each access (because the servers would each report that the key being offered by the browser on this access was unknown in the cache of that server) and falling back to generate and exchange a new session key, with the performance hit this entails.

It should probably be noted that for non-encrypted content, stickiness is almost never a desirable feature, as it tends to defeat the ideal of distributing the load across your server pool evenly. Stickiness can result in single servers being overloaded whilst relatively more idle peers sit nearby, in the name of better SSL performance. Thus, if there's no SSL, even this justification is absent and stickiness should not be used.

Another potential gotcha in terms of SSL performance comes in the use of so called 'client side certificates'. This is an option in most modern web servers and certainly in *mod_ssl*, the Apache SSL implementation. These certificates allows a much stronger form of authentication between the web server and the web client system and for the encryption of the inbound and outbound data streams with different session keys.

The cost though is that the computationally more expensive operation of encrypting the session keys (in RSA the encryption operation is more expensive in computational terms than the decryption operation as it happens) is now performed by the server as well as the client system. In the case of the clients, this is no big deal as they need only do it once for their own data and as it happens each client is a computer of its own to do it. The server of course benefits from no such distributed computing scaling and in fact suffers greatly from it.

The server may also incur additional overhead attempting the verification task of checking certificate signing authority signatures for the client side certificates it is offered. This is good security practice of course but also incurs an overhead. Once again, stickiness in your load balancers can at least ameliorate some of this processing by (hopefully) making it necessary only once per client system.

3.3 Separate Virtual Hosts for different uses

This is the content side of the discussion we had earlier about optimizing instances of Apache for different tasks. By splitting your site into different pieces, such as graphics, SSL and the like, and fetching those pieces of content from servers which have been specialized for those tasks you may gain performance benefits. At the very least you may gain the ability to measure each of these operations separately, rather than simply seeing a aggregate single figure of server load.

Note though, that to provide tuned Apache's we typically cannot use the Virtual Host facilities of the server but must rather build completely separate instances of the server (which may still be on the same physical server host).

In contrast, the measuring process is already facilitated merely by having separate virtual hosts, either as separate Apache instance or simply as virtual hosts. It may in fact be a useful strategy to build your content with such a breakdown in mind with simple Apache Virtual Hosts while your site is small and then as it grows deploy separate Apache instances on perhaps separate hosts to spread the load. The fact that you have prepared things by breaking things down across the separate hosts and indeed had an ability to break out the loads into separate numbers will both allow you to measure the load more effectively and to facilitate the migration of that load to the new host instance when you decide to do this.

3.4 Impact of other infrastructure

It's worth noting that your ability to service web loads is almost certainly impacted by things other than just the capacity of your web servers. Thus it's worth taking a more holistic approach to the task of maximizing your performance for any given level of resourcing.

At the very least you should examine questions like, network congestion, both in your LANS, your DMZ and on your link to the outside world. Many people aren't even monitoring this sort of information. In most environments, the size of the link to the Internet at large, the size of your Internet connection is the single biggest rate limiting element. For any successful e-commerce site though, you may be losing trade by not being able to service customers.

Similarly, you will find it valuable to monitor load in critical pieces of infrastructure such as routers (do they need more memory or bigger CPUs?) and firewalls (likewise...).

Beyond the web servers themselves, you likely have backend databases and middleware servers which also participate in the running of the site and indeed in offering your service and transacting business. It's briefly worth noting here that the 'transaction' overhead of serving a web page is much lower than that of the typical database. This means that a small web server may as has been noted easily serve hundreds of pages a seconds whilst few but the very largest database servers can process hundreds of database updates a second. This means it pays to review the use of your database regularly as naively written web applications can easily offer more database load than can be comfortably served.

Another form of ever more common middleware server is the Java servlet engine. Java has captured a significant portion of the web e-commerce market as it is a high performance mechanism for delivering dynamically generated pages. The servlets themselves may be hosted on the same host as the web server or indeed on a different host. The loads offered by these servlet environments should also be tracked with care. Often the servlet host will have more middleware servlet hosts of its own to communicate with (often referred to as application servers in the industry nomenclature) and indeed may also be dealing with one or more databases or other data sources or data stores. Thus, communication issues should also be watched once again (network load, etc. but also use of sockets/file descriptors, TCP tuning and the like).

3.5 Dumb Content

As the administrator of a web server or servers you may or may not have any control over the content on your web servers. You almost certainly don't have the time to vet that content yourself. Typically content is updated seemingly randomly and seldom with any advance notice.

Sadly, it is possible for badly structured content to impact the performance of a web site significantly. In the simple case, relatively benign issues such as graphics being served from somewhere other than your lean/mean server optimized for graphics or SSL being served from an Apache not tuned for it. Dynamic pages where little need for dynamicism is evident or even 5 copies of the same thing, when only one is marked to be cached by the server for efficient access.

In extreme cases, poorly written or ill-conceived dynamic content can bring a system to its knees. A common example here is database access from a highly trafficked dynamic page where no attempt it made to use a persistent connection to the database. Those of you experienced with databases will be aware of the significant overhead associated with connecting to a database and disconnecting from it. Doing this per page hit can cost a lot of unnecessary resources when a persistent connection would both perform more efficiently and offer better user response times.

4. Measuring Apache

Having looked at our content and organized our web servers, lets take a look at how we can characterize the performance of an Apache web server and indeed tune it.

4.1 Log analysis

The first thing to do is to periodically perform some analysis of the log files that Apache keeps for you. Analyzing the logs will typically show you a number of features about the load your web server is processing and can be done easily either with one of the myriad open source log analysis packages like *webalizer* or *analog*, or indeed commercial packages like *WebTrends* can be used.

For web sites with any sort of geographic locality, the first thing that is noticed is that like a bricks and mortar shop, your website sees most of its traffic during the business day. It seems that it's a fact of life that currently most people do their web surfing from their places of work. If you offer a service which is tied to business hours in some manner this is even worse (e.g. stock information or the like).

The load graph for such sites is quite stark, with essentially no load in the early morning, a high load arriving the start of business hours running more or less constantly till midday when an upwards spike marks the highest load point for a typical day and then a relatively smooth decay away down to midnight. There is usually a small downward spike round the end of the business day as people commute back home and/or have their evening meals.

Sites without such locality, will typically see much smoother levels of constant load as web surfing communities all around the world come and go from their site (in overlapping versions of the pattern above as it happens).

Analysis of your logs can at least then tell you where your maximum periods of activity are, and when your idlest times are. This can be useful for running backups or doing system maintenance although it must be said, that even for sites with locality, there is seldom a time when no activity is taking place.

Statistics which may also be of use are hit rates per hour (not all hours are equal as we noted), rates of page failure per hour (where spikes reflect perhaps overloaded middleware or other infrastructure), dwell times per page for your users (are your pages too complicated) and page rates vs hit rates.

It's worth noting that a wealth of data can be extracted from your log files and it may even be worth writing some specialist log reduction programs in *perl* or a similar language suited to reportage to look at specific issues which may be of interest to you.

4.2 ApacheBench

One of the simplest questions raised once you start looking at your logs is how much capacity does you web server have? This is where this paper started in fact. Your Apache has some ability to service load that arrives from the Internet and when you're examining your log file the most obvious question is how much of that ability are you consuming and how close to running out are you?

Most people take a fairly rudimentary approach here of monitoring the hosts their web servers run on. This is of course a reflection of the fact that there are lots of great host monitoring tools out there and they may as well use them. But as we noted earlier, it's not all about the host, with innate questions of how much work your web server can do, how much network load it can generate and so on.

A simple first step is to characterize the performance of your web server. Fire a bunch of web requests at it and see how quickly it can respond. Apache is even distributed with a tool called *ApacheBench* (although the binary has the unassuming name of *ab* and it's not well publicized) that allows you to generate loads and fire them at an instance of Apache.

It's fairly straightforward, and indeed can be very useful in getting some idea of the raw abilities of a server. It's easy to use with a straightforward command line interface. Its drawbacks are that it only exercises a single URL and it can't do SSL.

4.3 httperf

There is also a tool out there called *httperf* which does much the same sort of thing as *ApacheBench* except it's more full featured. It has lots of options and is also much more anal about timing in its operation. This is of course a good thing in this sort of work.

The biggest disadvantage in the use of *httperf* is a somewhat cryptic command line interface. It's well worth the effort to come to grips with though. Like *ApacheBench*, *httperf* also cannot assist with benchmarking SSL pages nor does it allow you specify more than a single URL to test.

Of the two programs though, it's certainly worth using *ApacheBench* if for no other reason than you already have it if you have Apache. The extra facilities of *httperf* and its extra rigour in timing makes it well worth the effort of finding and installing though.

4.4 http://server/server-status

Having benchmarked your server and analyzed your log files you now have some ideas of what your site is capable of and indeed what it gets up to from day to day. What is still lacking though is some idea of what the load looks like at any given point in time. By its nature, log analysis tends to take place offline and often well after the fact. Small sites may in fact only be doing it at months end. So how can you tell what your web site is up to right now?

The simplest facility is of course to ask the web server(s) for some statistics about its operation. There is an Apache module called mod_status which allows the server to keep some basic stats and offer them on a web page. By default, this page is *http://yoursite/server-status*, although its name may be changed in the Apache configuration file.

The information presented is both summary, various totals for hits and bytes transferred and the like and an overview of the activity of the Apache server pool.

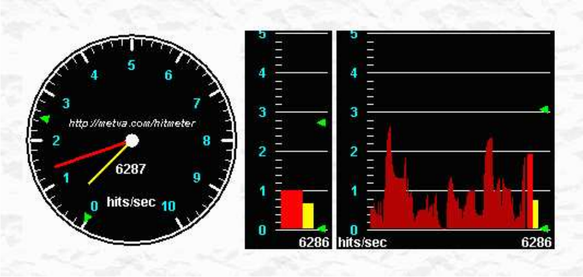
The biggest shortcoming of this data is that it's accumulated since the last server start or restart. So, when the status page offers a server hit rate, it is an average rate since the last server restart. As we noted though unless your site is one blessed with a relatively constant load, this is unlikely to be of great use to you.

4.5 HitMeter

A solution then is to use the status page to access the raw totals and display these in some more useful manner. My contribution to this then was to write a small Java applet which could repeatedly access the status page and display the differences in total hits between successive readings as a hits per second rate. The display, was in the form of a speedometer like dial in its original form although the applet can now also offer bar graph and stripchart style displays.

This then gives us a real-time or at least near real-time look at how hard the Apache web server it is sampling is working. And it does it without any modifications being required to Apache itself.

The applet can be viewed at *http://metva.com/hitmeter*, and I have made it open source under a BSD style license for those of you who'd like to avail yourselves of it.



In fact, the applet can also display other data, so long as it can download it from a page that 'looks like' the Apache server status page. Work is ongoing to make the data fetcher thread of the applet more flexible to allow it to read other non-HTTP data sources directly and to allow it to read data sources that look different from the Apache status page. It can be configured to display absolute data rather than differences (where data is already being presented as a rate) and can also in its current form display the Apache kb/s rate rather than the hit/s rate.

4.6 SSL benchmarking

It's worth briefly touching on the state of the art in benchmarking the SSL performance of web servers. In the open source arena, SSL benchmarking tools are still fairly thin on the ground. A number are said to be under development, indeed may by the time you read this have struggled out into the light of day, but by and large at the time of this writing, most SSL benchmarking still tends to be a little Mickey Mouse. (Often relying on scripts and tools like *curl* rather than something with the rigour of, say, *httperf*).

The reasons for this are fairly straightforward. SSL is hard. It's hard to do right, it's complex and likely it's only low reward. The organizations who need it may fall back to some commercial benchmarking packages which have addressed the need to offer something, albeit at an often hefty price.

Having said that, there is at least one OpenSSL solution said to be under development for instance. OpenSSL is a good choice as it will remove a lot of the need to re-invent the (cryptographic) wheel, although once again, a correct solution is still far from trivial.

One of the other things to watch in any benchmarking tool is the need to balance the use of short and long SSL handshakes. That is, the SSL connection that establishes a new session key (the long handshake) and the SSL connection which re-uses a previously established session key (the short handshake). As we noted, real SSL traffic is a mix of the two types of traffic with we would hope a bias toward the latter type, but we need to be able to adjust this bias in our benchmarking tool as it is difficult to know precisely what that bias is for our users. If you have to pick only one use the long handshake as it offers you a (very) worst case performance figure.

This is one of those areas worth pursuing if you do much SSL on your website precisely because it can be so rate determining for its performance. And as we all know, sluggish websites are not pleasant experiences and often people will avoid them wherever possible. If your livelihood depends on the SSL performance of your website (as it well may), then this alone could make or break you.

4.7 Other benchmarking products

As I noted above, there are a number of commercial offerings in the arena of benchmarking web applications/sites. The highest profile of these is probably *LoadRunner*. This is a great product although the cost of entry can be quite high for some. It works on a record/replay style model where a 'typical' workload is recorded from a live browser session and a script for a this typical workload is developed (with accesses, dwell times and other such features of real users). This script is then replayed in parallel to simulate the actions of multiple users accessing a website.

LoadRunner sees a fair bit of use in the commercial world. Clearly its focus is users rather than hits or pages (and we can make an argument that this is a much more sensible figure of merit to be concerned with) but this does mean both that it's hard to relate *LoadRunner* derived statistics to those a running web server offers or indeed that correlations between users and hits are that clear.

Other benefits of packages like this though are to offer an ability to regression test web applications and perform other quality assurance type operations (like running test cases or indeed the load testing that gives LoadRunner its name) which more conventional server only benchmarking tools with their focus on only one or at best a few URLs cannot offer.

5. Summary

In any large enterprise capacity planning and benchmarking are vital tools of the trade. Oddly, in the world of the web though there is only poor focus on the capacity of a web server and almost no focus on knowing how fully it is utilized.

Some of this is born of the fact that unlike traditional datacentre models, web based service models have no real control of how much load is presented to them. In some instances, attractive, compelling web sites get no traffic whilst simultaneously ugly, expensive and poorly implemented sites are being overwhelmed. The vagaries of search engines, online advertising, link exchanges and other means of getting the word out mean that at times offered load bears little relationship to the attractiveness of the offerings of your site as such.

Over and above this, successful advertising can bring waves of sudden load spilling in which you may not be prepared for (marketing people on occasions will run campaigns without advising the service delivery portions of an organization). And finally, as alluded to in the abstract of this paper, there are natural seasonal variations in load which may see see your comfortable performance headroom consumed and put your site at risk of complete failure, merely because of its own popularity.

All of these things suggest that in order to properly manage a site, you must have some idea of both its capabilities and of to what extent those capabilities are being used. Where possible, you can tune your environments to maximize their performance (in all but the most profligate environments you will likely be doing this anyway) and then monitor their performance in an ongoing manner to ensure that you have adequate performance headroom to meet any new load that may arrive.

One final word of caution then about predicting the nature of the load you may face. It's very hard to predict web loads. As I noted, it's not uncommon for sites to face sudden, unexpected and uncatered for loads arriving from the Internet. Anecdotal evidence abounds of large, well financed web sites which have manifestly failed in the past to scale to deal with the load offered them and no doubt more organizations will face the same problems in the future. All of these incidents represent lost revenue, and if they were properly managed sites, their current revenue should be sufficient to allow them to build infrastructure that can accept tomorrows loads. (More simply put, if you're just squeaking by, you're not doing it right...)

But crystal ball gazing is hard. It hard to know how much load is coming. Is Christmas twice as busy as August? What if it's suddenly three times as busy? You need to plan. The first step is know what your site is capable of today and knowing what it's being asked to deliver today. Then planning for tomorrow becomes easier...

This paper is reprinted with the author's permission from the proceedings of the AUUG 2003 conference, with subsequent updates. You can find the latest update at http://www.metva.com.au/users/enno/papers/hitmeter.php.

My Home Network

Frank Crawford <frank@crawford.emu.id.au>

Half way through the year and still going strong, or at least still going. Originally, I thought I'd just be writing about some network upgrades around my home network, but since I started planning this column I've been involved in a few more upgrades and changes. So, because of all these changes this column more than others will be a lot of bits and pieces.

Anyway, in roughly chronological order, just after the last column I invested in a number of network upgrades. Given the layout of my home, it was practical to install a Gigabit Ethernet switch linking up my main systems. Since the price of an 8 port unmanaged gigabit switch was just over $300, and I was forever waiting for local copies between systems, I jumped in and picked up a D-Link DGS-1008D switch from EverythingLinux.

After running my home network on 10Base-2 (Coax) which is really a half-duplex connection, it was an amazing upgrade, over 200 times the speed, although practically, I'm only using a 20 times improvement. However, it wasn't without its problems. The biggest one being that my older switch, with which I planned to link my old 10Base-2 network and the new 1000Base-T cabling, would not communicate. In fact this looks like it was an auto-negotiation problem, in that all connections to the DGS-1008D are autonegotiated, and as it is unmanaged, can't be changed, while the much older switch doesn't understand anything about autonegotiation. As such they can't communicate. After much playing around the simplest solution turned out to be to get a cheap auto-negotiating 10/100 Fast Ethernet switch for $25. As it turns out, this talked find to the Gigabit switch and to the 10Base-2 switch, and additionally gave me a point to connect other systems into the network.

Of course the final point about all this is that while the switch is capable of Gigabit performance (even so far as 2Gb/sec between ports) none of my servers currently support anything over 100Mb/sec and so don't stress the network. One curious point is how quickly the 8 ports on the switch were used, with three to the main servers, one to a wireless access point and two more to other Fast Ethernet switches. This only leaves two free ports, although it should be enough for mid-term expansion.

Since I already had a laptop (see my previous column), the next logical change was the installation of a wireless access point and a wireless card for the laptop. Of course, to give me more options for the future, I selected a D-Link DWL-7000AP, which is a tri-mode dualband wireless access point, providing 802.11a/b/g protocol support for network performance up to 54Mb/sec. Of course, the AP isn't the whole story, and again, this is where things sometimes come unstuck. Linux support for wireless cards is a bit problematic at present, as vendors bring out new chipsets, but won't open up the specifications. In most cases Linux drivers are well behind the release of these new chips, as they have to discover the operation by trial and error or by reverse engineering.

Initially I used an 802.11b PCMCIA card in my laptop, which was supported by the standard Orinoco chipset. Unfortunately, this only gave me a maximum of 11Mb/sec transfer rate. The obvious solution was to pick up a 802.11g PCMCIA card (or 802.11a, but 802.11g seems to be winning the battle) to connect at maximum speed. I was able to pick up a D-Link DWL-G650+ card, although at the time I assumed that it was one of the other D-Link varieties, namely the DWL-G650, DWL-650 or the DWL-650+, some of which are supported and others aren't. Luckily, the DWL-G650+ is a chipset that is being developed, although it is not fully supported (or even really working yet).

The chipset in the DWL-G650+ is a TI ACX111 chipset, an extension of the ACX100 chipset, supported through *http://acx100.sourceforge.net/*. This is developing a native driver fully supported by the Linux community and likely to be included in a future kernel tree. However, at this point it is not really functional and so I needed to find another solution.

This is where the Open Source solution shows its worth, as there are often multiple competing projects. In this case there is another related project, the NDIS wrapper which takes a very different approach. In this case, rather than trying to write a driver for the card, they have written a kernel module that can load MS Windows network driver API (Ndis) drivers. In fact it only needs to support enough of the Ndis API to get the cards working. The NdisWrapper project can be found at *http://ndiswrapper.sourceforge.net/* (isn't it amazing how many things point to SourceForge?).

To use the NdisWrapper you need the drivers supplied for an MS Windows installation, which are usually supplied with the card, and this does create a bit of controversy within the Open Source community, as there is one school of thought that objects to giving any form of support to proprietary drivers. Unfortunately, I'm from the pragmatic school of thought and use whatever works best. The NdisWrapper driver has a number of limitations, and doesn't provide all the hooks desired by Linux, in particular some statistics and similar peripheral items, so at present I intend to use the NdisWrapper until the native ACX111 driver is working.

Of course fixing up my laptop has only been a small part of the last few months, with the bulk of it being software upgrades of all my Linux systems. In mid May the Fedora community released Fedora Core 2 (FC2), the latest, "greatest" release.[1] While there are a number of changes in FC2, the biggest one is that the kernel is now Linux 2.6, and this is something I wanted to explore. However, while there were no other major updates, there were a number of minor ones, some of which caused me problems.

The first problem to occur was that their X11 base has moved from XFree86 to X.org. This has caused a couple of issues, including a Xkb error, with an easy fix. To correct this, just take your favourite editor and edit */etc/X11/XF86Config* and change the line that reads

```
  Option "Xkbrules" "xfree86"
```

to become

```
 Option "Xkbrules" "xorg"
```

Note, this doesn't happen with every installation, but did occur in two out of my five. I suppose it depends on the history of the X11 configuration.

Going on from there I found that the Synaptics touchpad wouldn't load or work. This is an understandable issue as the binaries have changed a bit with the X.org version. The easiest fix is to get the new source and compile up a new version (the older version won't compile due to some changes to header files). The full details are at *http://w1.894.telia.com/~u89404340/touchpad/*, and you can now find an RPM in a Fedora Repository (*http://www.fedora.us/*).

At this point the workstations were generally working well, although using the Fedora distributed kernel. I decided to compile my own 2.6 kernel up, both to address some issue, such as missing the FireWire driver and include some other options, such as support for the NTFS filesystem. It is supported through an RPM from *http://linux-ntfs.sourceforge.net/rpm/index.html*, but I'd prefer to be self-sufficient.

The steps here are fairly simple, but I did initially run into some major problems. While for upgrades in kernels in the same major revision (i.e. 2.4) it is sufficient to do a 'make oldconfig' to upgrade the configuration for updates, this is not sensible for a major revision as there are just too many incompatible changes. As such the best method is to take a known configuration and modify it to my requirements. The two reasonable candidates were the default for a Linux 2.6 kernel or the Fedora FC2 configuration (which is available in */boot/config-2.6.5-1.358*). In my case I took the default FC2 configuration and went to extend it. The biggest changes were to enable the NTFS filesystem (which was disabled due to Red Hat's concerns with Microsoft patents) and for my workstations to enable "Preemptible Kernel", which allows low priority processes to be preempted even if it is in kernel mode executing a system call. This is a new feature of the 2.6 Kernel, and is recommended for desktop, embedded or real-time system. Interestingly, FC2 doesn't enable it, even though they aren't officially for servers!

To make the kernel, I just copied the base configuration file to *.config* and then ran *make*gconfig to create an X11/GTK GUI for generating my new configuration file. Once generated all I should have needed to do was

```
 make clean install modules modules_install
```

(note 'dep' is no longer needed) to generate the final kernel. However, there is a problem with the default configuration files, they all have 'DEBUG_INFO' set, which saves debugging info in the kernel and modules. The effect of this was to increase the size of the kernel by a factor of over 100, and a normal 26MB kernel and modules (big enough anyway) to fill a 2GB partition. I can't believe that the Fedora group actually have that option enabled! After disabling this option things generally worked out.

[1] You'll find Fedora Core 2 CD-Rs in this issue –Ed

While things worked well enough for the desktop upgrades, I did come across a few more problems with my server. The biggest one was that IMAP server supported on Fedora Core 1 and previously in Red Hat was the ‘imap’ and has now been replaced by Cyrus-IMAPD. Unfortunately, there is an immense difference between the two, in particular, ‘imapd’ takes input from the standard mailfile, while Cyrus-IMAPD uses a maildir structure (i.e. each message is a separate file in a mail folder is a directory). In addition, Cyrus-IMAPD expects input fed through a special delivery program, which is to some extent incompatible with ‘procmail’, the standard program for mail filtering. All in all, while Cyrus-IMAPD is a much better IMAP server, it is a major change and is not suitably documented.

Before anything else, you need to start the Cyrus-IMAPD daemons, which can be done by:

```
/etc/init.d/saslauthd start
/etc/init.d/cyrus-imapd start
```

Obviously, "saslauthd" requires some certificate, but this appears to have been automatically created with the install procedure, so I won’t comment further.

To perform the conversion there are a number of steps to be taken. The first is to create the mailboxes for the users and their folders. Prior to creating the first users, I added the following lines in ‘/etc/imapd.conf’:

```
unixhierarchysep: 1
altnamespace: 1
```

The first line allows for Unix-style separators (‘/’) instead of news-style (‘.’). Also the folders are created a bit differently inside the Cyrus spool.

Without the second line, all IMAP folders must be created inside Inbox by the mail client. Adding the second line allows new folders to be created at the same level as Inbox.

To then create a user’s mailbox you need to run the ‘cyradm’ command as the "cyrus" user. Basically, run the following steps:

```
su - cyrus -c "cyradm localhost" << EOF
cm user/frank
setacl user/frank frank lrswipcd
EOF
```

This needs to be done for each user. Note also that "cyradm" asks for a password, which is the one for the local user ‘cyrus’.

The next step is to create the other user folders, however, this is a bit more complex, as it is necessary to discover what mailfiles are currently active folders. The easiest way, I found to do this, was to search for the file ‘.mailboxlist’ in each user’s home directory. This lists the mailboxes that are currently subscribed in some IMAP client. With this list, it is necessary to create additional mailboxes with commands similar to

```
cm user/frank/AUUGN
```

say to set up a folder named "AUUGN". There is no need to set ACLs as they are inherited from the parent.

While this creates empty mailboxes, for most people the preference is to keep all their old mail (otherwise why do they have folders anyway?). The recommended way to convert from one IMAP daemon to another is to set up two separate servers and then have the users select and drag from one set of folders to the others. For a small setup, such as a home, this is not a real possibility, especially when it is only after the upgrade that you find out the requirement. While there is no official upgrade path, there is a package that does help. Within the distributed FC2 RPMs, there is "cyrus-imapd-utils", which includes some Perl scripts to assist with the conversion. This package acted as a starting point, although there were a number of problems, incorrect paths, and lack of documentation to make it a non-trivial task.

The basic procedure I adopted was to convert the ‘.mailboxlist’ files into suitable input to the ‘inboxfer’ and ‘folderxfer’ scripts and then perform the copy. The format required for the ‘*xfer’ scripts is lines of the form:

```
username:Cyrus-mailbox-name:BSD-mailbox-name
```

for example for my earlier copy of AUUGN, I’d have:

```
frank:user/frank/AUUGN:/home/frank/Mail/AUUGN
```

This will create individual files within the Cyrus-IMAPD partition. By default, this is ‘/var/spool/imapd’, in a separate directory for each user. This procedure leaves the files owned by "root", so it is necessary to do a:

```
chown -R cyrus:mail /var/spool/imapd
chmod og= /var/spool/imapd
```

Finally, it is also then necessary to rebuild the Cyrus-IMAPD database by issuing the command:

```
su - cyrus \
   -c "/usr/lib/cyrus-imapd/reconstruct \
   -r user/frank"
```

which will recursively rebuild all mailboxes for the user `frank`. At this point all the mailboxes have been converted, however, it doesn't allow new mail to be delivered.

To allow new mail to be delivered, there are two possibilities, one to modify */etc/mail/sendmail.mc* to deliver directly by adding:

```
define(`confLOCAL_MAILER', `cyrusv2')dnl
define(`CYRUSV2_MAILER_ARGS', \
  `FILE /var/lib/imap/socket/lmtp')dnl
MAILER(cyrusv2)dnl
```

or to allow *procmail* to continue to work, add the following at the end of each user's *.procmailrc*:

```
:0W
| formail -I"From " | \
  /usr/lib/cyrus-imapd/deliver \
  -a $LOGNAME -m user/$LOGNAME

# If that fails - maybe because the user
# is out of quota, or the mailbox hasn't
# been created - then force a bounce
# (otherwise the message would get
# silently appended to
# /var/spool/mail/$LOGNAME).

# This is EX_CANTCREAT (Can't create output)
EXITCODE=73
:0
/dev/null
```

Obviously, much of this may need to be done while sendmail is stopped, otherwise mail may be lost.

The final issue that came out of the upgrade was that Red Hat/Fedora had decided to disable the ability of "xmms", the X Multimedia System, to play MP3 files. While it is easy to download a non-crippled version from the XMMS site (*http://www.xmms.org*), however a better solution is to download a plugin that restores the MP3 support. This plugin can be found at *http://rpm.livna.org/fedora/2/i386/RPMS.stable/ xmms-mp3-1.2.10-0.lvn.2.2.i386.rpm* and install the rpm, with

```
rpm -Uvh xmms-mp3-1.2.10-0.lvn.2.2.i386.rpm
```

This should work straight away, but if you want to be correct, you may want to go into xmms preferences and disable the Input plugin "MPEG Layer 1/2/3 Placeholder Plugin [librh_mp3.so]" and ensure that "MPEG Layer 1/2/3 Player 1.2.10" is enabled. While you are at it, check that the selected Output Plugin is suitable, such as "ALSA 1.2.10 output plugin [libALSA.so]" for the Linux 2.6 Kernel.

Finally, for a list of other FC2 changes and issues, I've found Fedora News Updates very handy. *http://fedoranews.org/colin/fnu/issue13.shtml*, in particular, also includes some other hardware specific issues and other things I haven't encountered.

Oh well, that will do for this issue, even though there are a couple of things I will keep for next time, but in the meantime, I'd better start preparing for AUUG2004, which will be held in Melbourne this year. Check out the AUUG website (*http://www.auug.org.au/*), and I hope to see you there.

The Design and Implementation of the NetBSD rc.d system

Luke Mewburn <lukem@NetBSD.org>

This paper was originally presented at the AUUG 2003 Conference "Open Standards, Open Source, Open Computing and is reprinted with the permission of the author.

For space reasons, this article does not include the references. See the original article at *http://www.mewburn.net/luke/talks/auug-2003/index.html* for the references and possible other updates.

Abstract

In this paper I cover the design and implementation of the *rc.d* system start-up mechanism that has been a part of NetBSD since NetBSD 1.5, which replaced the monolithic */etc/rc* start-up file inherited from 4.4BSD. Topics covered include a history of various UNIX start-up mechanisms (including NetBSD prior to 1.5), design considerations that evolved over six years of discussions, implementation details, an examination of the human issues that occurred during the design and implementation, and enhancements made since the initial integration.

Introduction

Three years ago NetBSD converted from the traditional 4.4BSD monolithic */etc/rc* start-up script to an */etc/rc.d* mechanism, where there is a separate script to manage each service or daemon, and these scripts are executed in a specific order at system boot.

This paper covers the motivation, design, and implementation of the *rc.d* system; from the history of what NetBSD had before to the system that NetBSD 1.5 shipped with in December 2000, and the enhancements made since then in NetBSD.

The changes were contentious and generated some of the liveliest discussions about any feature change ever made in NetBSD. Parts of those discussions will be covered to provide insight into some of the design and implementation decisions.

History

There is great diversity in the system start-up mechanisms used by various UNIX variants. A few of the more pertinent schemes are detailed below. As NetBSD is derived from 4.4BSD, it follows that a description of the latter's method is relevant. Solaris' start-up method is also detailed, as it is the most common System V UNIX variant.

4.4BSD

4.4BSD has a rather simple start-up sequence.

When booting multi-user, the kernel runs `init` (located in */sbin/init*), which spawns a shell (*/bin/sh*) to run */etc/rc*, which contains commands to check the consistency of the file-systems, mount the disks, start up system processes, etc. */etc/rc* invokes */etc/netstart* to configure the network and any associated services, and */etc/rc.local* (if it exists) for locally added services. After */etc/rc* has successfully completed, `init` forks a copy of itself for each terminal in */etc/ttys*, usually running */usr/libexec/getty* on them.

Administrative configuration of system services is controlled by editing the scripts (*/etc/rc*, */etc/rc.local*, */etc/netstart*). In some instances, only shell variables need to be changed, in others commands are added, changed, or removed.

4.4BSD has no specific shut down procedure. After `init` receives a `SIGTERM` signal it sends a `SIGHUP` signal to each process with a controlling terminal, which the process was expected to catch and handle appropriately. Ten seconds later, this is repeated with `SIGTERM` instead of the `SIGHUP`, and another ten seconds after that `SIGKILL` is sent. After all processes have exited or when thirty seconds had elapsed, `init` then drops to single user mode, reboots, or shuts down, as appropriate.

Solaris 9

Solaris is the most common System V variant, and serves as a good reference implementation of the System V *init.d* mechanism, as implemented by System V Release 4 (SVR4).

When running, the system can be in one of eight distinct run levels which are distinct states in which selected groups of processes may run. The run level may be changed at any time by a privileged user running the `init` with the run level as the argument, and the current run level may be determined at any time with the *who*-r command.

When the system is booted, the kernel runs `init` (located in */sbin/init*), whose purpose is to spawn processes defined in */etc/inittab* For each configuration line in */etc/inittab* that has a run level field ('rlevel') which matches the current run level, `init` starts the process defined on that line as per the given 'action' field. The different run levels are:

0 Shut down the operating system so that it's safe to turn off the power.
s Single user mode, with some file systems mounted.
1 Single user mode, with all file systems mounted. User logins are disabled.
2 Multi user mode, with all services running except NFS server daemons.
3 Multi-user mode with all services running. This is usually the default.
4 Currently unavailable.
5 Shut down the system and attempt to turn off the power.
6 Shut down the system to level 0, and reboot.

For a given run level *X*, a shell script */sbin/rcX* exists to control the run level change, and */etc/rcX.d* contains scripts to be executed at the change. */sbin/rcX* stops the services in the files matching */etc/rcX.d/K** in lexicographical order, and then starts the services matching */etc/rcX.d/S** in order.

To add a new service *foo* requires adding */etc/rcX.d/S*foo* in the appropriate run level to start the service, and then */etc/rcY.d/K*foo* in all the other run levels *Y* where the service is not to be run. Usually these files are actually links to the appropriate script in */etc/init.d* which implements the start up and shut down procedures for a given service.

To disable or remove a service *foo*, any files matching */etc/rc?.d/[KS]*foo* need to be removed.

NetBSD prior to 1.5

Prior to the release of NetBSD 1.3, NetBSD's start-up mechanism was similar to 4.4BSD's, with relatively minor changes, as described below.

NetBSD 1.3

In NetBSD 1.3 (released in January 1998), two major user-visible additions were made to the start-up system; */etc/rc.conf* and */etc/rc.lkm.*

/etc/rc.conf contains variables to control which services are started by */etc/rc* and */etc/netstart.* For each service *foo*, two variables may be provided:

foo Can be "yes" or "no" (or various other boolean equivalents). If set to "yes", the service or action relating to *foo* is started.
foo_flags Optional flags to invoke *foo* with.

The aim of */etc/rc.conf* was to separate the scripts that start services from the configuration information about the services. This allows updating of the start-up scripts in an operating system upgrade with less chance of losing site-specific configuration.

Similar */etc/rc.conf* functionality has been implemented in commercial UNIX and freely-available BSD derived systems, including current systems such as FreeBSD. By the time this change was considered for NetBSD, it had a reasonable number of users of the prior art to help justify its implementation.

/etc/rc.lkm was added to provide control over how load-able kernel modules (LKMs) are loaded at boot time. */etc/rc.lkm* is invoked at three separate stages during the boot process; before networking is started, before non-critical file systems (i.e., file systems other than */*, */usr*, */var*) are mounted, and after all file-systems are mounted. This complexity is required because an LKM may be located on a local or remote file system. The configuration file */etc/lkm.conf* controls behavior of */etc/rc.lkm.*

NetBSD 1.4

In NetBSD 1.4 (released in May 1999), two more additions were made; */etc/rc.shutdown* and */etc/rc.wscons.*

/etc/rc.shutdown is run at shut down time by `shutdown`. This occurs before the global `SIGHUP` is sent. This is useful because there are

some services that should be shut down in order (e.g., database-using applications before their databases) and some services that require more than SIGHUP for a clean shutdown.

/etc/rc.wscons was added to control how the *wscons* console driver was configured at boot time, and to allow manual reconfiguration. */etc/wscons.conf* controls this behavior.

Summary prior to NetBSD 1.5

At multiuser boot, init calls */etc/rc* to initialize the system. */etc/rc* calls */etc/netstart* to setup network services, */etc/rc.local* for local services, */etc/rc.lkm* to initialize load-able kernel modules, and */etc/rc.wscons* to configure the *wscons* console driver. The start-up of services is controlled by variables in */etc/rc.conf.*

At system shutdown time, shutdown calls */etc/rc.shutdown* to shut down specific services which have to be shut down before the global SIGHUP that init sends.

Design considerations

Over a six year period, various ideas on how to enhance the start-up system were floated on the public NetBSD mailing lists 'current-users' and 'tech-userlevel', as well as on the NetBSD developer-only mailing list.

There was no consensus on '*One True Design*'; there was too much contention for that. What is described below is an amalgamation of what a few developers felt was a reasonable analysis of the problems and feedback as well as the most reasonable solution to support the widest variety of circumstances.

Problems with the old system

The old system was perceived to suffer from the following problems:

- There was no control over the dependency ordering, except by manually editing */etc/rc* (and other scripts) and moving parts around.

  This caused problems at various times, in situations such as workstations with remotely mounted */usr* partitions, and these problems weren't completely resolved as was seen by observing various mailing discussions and a flurry of CVS commits to the source tree.

- It was difficult to control an individual service after the system booted (e.g., restart dhcpd, shut down a database, etc).

  Whilst some people suggested that a system administrator who couldn't manually restart a service was incompetent, this doesn't resolve the issue that typing "/etc/rc.d/amd restart" is significantly easier and less error-prone than finding the process identifier of amd, killing it, examining */etc/rc* for the syntax that amd is invoked with, searching */etc/rc.conf* for any site-specific options, and manually typing in the resulting command.

- It didn't easily cater for addition of local or third party start-up mechanisms, especially addition into arbitrary points in the boot sequence, including those installed by (semi-)automated procedures such as the NetBSD 'pkg' tools.

Requirements of the new system

Given the problems in the old system, and observations of what other systems have done, including those described above, the following design considerations were defined.

Some of these considerations were not determined during discussion prior to implementation, but were identified once users were actively using the implementation.

Dependency ordering

Dependency ordering is a strong requirement.

The following dependency ordering requirements were determined:

- Independence from lexicographical ordering of filenames.

  Some other systems (e.g., System V *init.d*) use an existing lexicographical ordering of filenames in a given directory, such as */etc/rc2.d/S69inet* occurring before */etc/rc2.d/S70uucp*, but experience has shown that this doesn't necessarily scale cleanly when adding local or third-party services into the order; often you end up with a lot of convoluted names that start with "*S99*".

- Ability to insert local or third-party scripts anywhere into the sequence.

  Some people proposed running */etc/rc.d/** out of */etc/rc.local*, and retaining the existing

/etc/rc semantics. This doesn't easily cater to a user who requires the ability to insert their own start-up items anywhere in the boot sequence (such as a cable modem authentication daemon required for networking).

- Not bloating */bin* and */sbin* on machines with small root (*/*) file-systems. Tools from */usr/bin* have to be avoided because */usr* might not be available early in the boot sequence.
- Use a dynamic dependency ordering.

  A lot of debate occurred regarding whether the dependency ordering is predetermined (e.g., by creating links to filenames or building a configuration file), or dynamically generated.

  A predetermined order may be more obvious to determine the order (using `ls` or examining the configuration file instead of invoking a special command), but it can be difficult to add a service in at a given point on a system because generally ordering is not based on services provided.

  A dynamic order may slow down boot slightly, but provides the flexibility of specifying start-up order in terms of dependencies.

  For example, if service *C* depends on *B* which depends on *A*, and I have a new service *D* to install that must start after *A* and before *C* then I want to specify it in these terms, without having to worry about whether it starts before *B*, after *B*, or simultaneously with *B*. There was some discussion about various methods in which to determine the dynamic ordering:

  - Using `make` and a *Makefile*.
  - Using `tsort`, `awk`, and a few shell commands
  - Providing a dedicated ordering tool which parsed the scripts for command directives in special comments to determine the order. If a script did not have a directive, it would be ordered last.

After various discussions and implementation tests, it was decided that a dedicated dynamic ordering tool, `rcorder` (described below) was the most appropriate mechanism; using `make` or `tsort` and `awk` would require moving those programs to */bin* ('bloating' the root file-system for machines with limited resources), and a dedicated tool could provide better feedback in certain error situations.

Manipulation of individual services

Most people seem to agree that the ability to manipulate an individual service (via a script) is one of the benefits of the System V *init.d* start-up mechanism. Having a script that allows direct starting, stopping, and restarting of a service, as well as other per-service options like 'reloading configuration files', significantly reduces system administration overheads.

Having the same script be used by the start-up sequence is also highly desirable, as opposed to using a monolithic */etc/rc* for booting and separate */etc/rc.d* scripts for manual control (which had been suggested).

It is interesting to note that some System V *init.d* implementations often start multiple services in the one file, which defeats the purpose of providing per-service control files. An example is Solaris' */etc/init.d/inetsvc*, which configures network interfaces, starts `named` and starts `inetd`.

Support third-party scripts

An important requirement is the ability to support third-party scripts, especially by allowing them to be inserted at any place in the boot sequence order.

The current system does support third-party scripts if they are installed into */etc/rc.d*. There has been discussion about allowing for different directories to be used for local and third-party scripts, in order to provide a separate 'namespace' to prevent possible conflicts with a local script and a future base system script, but so far none of the suggestions has been considered sufficiently complete to provide in the default system. This, however, does not prevent a site from implementing their own method.

Maintain /etc/rc.conf

/etc/rc.conf was introduced in NetBSD 1.3, and most users seem fairly happy with the concept.

One of the concerns about a traditional System V *init.d* style mechanism is that the control of service start-up is managed by the existing of a link (or symbolic link) from */etc/rc2.d/S69inet* to */etc/init.d/inetinit*, which is difficult to manage in a traditional configuration change management environment (such as RCS). Similar concerns exist regarding the suggestion of using mode bits on files in */etc/rc.d* to control start-up.

/etc/rc.conf was further enhanced as described below.

Promote code re-use

Traditional System V *init.d* implementations do not appear to re-use any code between scripts. From experience, maintaining local scripts in a traditional *init.d* environment is a maintenance nightmare. We achieved code re-use with common functions in */etc/rc.subr* which results in the average */etc/rc.d* script being a small (5-10 line) file.

Service shut down

The ability to shut down certain services at system shutdown time with */etc/rc.shutdown* was a useful feature of the previous system and of other systems, and it makes sense to retain this feature.

In the initial implementation, we reverse the dependency order, and shut down any services which are tagged with a "shutdown" keyword (see below) within the script. We may modify or enhance this behavior if observation of in-field use reveals a more complicated scheme is required.

Avoid mandatory run levels

We avoided the use of System V run levels (also known as run states or init states) and */etc/inittab*. This was the result of many discussions about the design, which can be summarized to:

- They're just too contentious; the */etc/inittab* concept had the least number of advocates. Many people expressed the opinion (both during the design phase and post implementation) that they don't mind the */etc/rc.d* idea but they do not think that an */etc/inittab*, run-levels or */etc/rcN.d* directories would improve things.
- There doesn't seem to be consistency between what each run-level means on various System V *init.d* implementations, or the exact semantics of what occurs at state change. Thus, using the argument of compatibility for system administration ease of use isn't as relevant. Some systems (such as HP/UX 10.x) treat these as levels, where a transition from level 4 to level 2 executes the shut down scripts in level 3 and then level 2. Other systems (such as Solaris) treat these as separate run states, where a transition to a level runs all the stop scripts in that level and then all the start scripts. This can be confusing to an administrator, as well as not necessarily providing the optimal behavior.
- If the ability to take the system from a given point in the order to another point in the order, then I feel that most people's requirements for what run-levels are touted to provide would be met. This is currently a work in progress - '`runrcto`'.
- Whilst */etc/inittab* provides for re-spawning of daemons, in practice very few daemons are actually started that way, and it's trivial to implement that feature in a few lines of shell script as a 'wrapper' to the start of the daemon.

Other issues

After various discussions, we settled on the name */etc/rc.d* instead of */etc/init.d*, because the implementation was different enough from the System V *init.d* mechanism that we decided not to confuse people expecting the exact System V semantics. Many system administrators may be used to referring directly to */etc/init.d/foo* or */sbin/init.d/bar* when manipulating a service; a symbolic link from */etc/init.d* or */sbin/init.d* to */etc/rc.d* on their systems could help retain their sanity.

The first implementation of */etc/rc.d* that I released for evaluation supported all three start-up schemes; the original monolithic */etc/rc*, a System V *init.d* (without run-levels), and the current */etc/rc.d*. These were all built from the same sources, and a command was provided to generate the style that an administrator preferred. After feedback and discussion, this functionality was abandoned, because:

- It is very difficult to support multiple ways of starting the system when users have problems or questions, especially so in a volunteer project.
- Two of the methods (*/etc/rc*, and System V *init.d*) do not have the ability to dynamically order the dependency list. In those situations, an administrator (or automatic application) would have to perform the extra step of 'rebuild order' upon installation.
- The source scripts had various constraints to ensure that they could work as part of */etc/rc* as well as acting as a stand-alone script in */etc/rc.d* or */etc/init.d*.

As architects of the NetBSD operating system, we have the responsibility to provide useful solutions to problems. In general, those solutions should be as flexible as possible, without introducing *unnecessary* flexibility, which will only cause confusion. Therefore, the alternative mechanisms were dropped.

That said, the current system is flexible enough that if a site decided to use a System V *init.d* approach, it is fairly trivial to populate */etc/rcN.d* with a symbolic link farm to files in */etc/rc.d* (using `rcorder` to build the dependency list), and modify */etc/rc* to run the scripts in */etc/rcN.d/* in lexicographical order, or to even implement a System V */etc/inittab* and run states.

Unfortunately, there is no easy solution for people who want to retain */etc/rc*. However, as NetBSD is an Open Source project and allows for public access to the CVS source code repository (via anonymous CVS as well as via a WWW front-end, nothing prevents users from reverting to the old style */etc/rc*.

Configuration improvements

The */etc/rc.conf* mechanism was enhanced in two ways:

1. The default configuration settings were moved from */etc/rc.conf* to */etc/defaults/rc.conf*, and */etc/rc.conf* sources the former. Site specific configuration overrides are placed in */etc/rc.conf*. This enables easier upgrades (both manual and automatic) of the default settings in */etc/defaults/rc.conf* for new or changed services.

   There was debate about this change, but a significant majority of users agreed with the change. Also, FreeBSD had made a similar change some time before, with a similar debate and outcome, and subsequent upgrade benefits observed which helped the case supporting the change.

2. An optional per-service configuration file in */etc/rc.conf.d/SERVICE* was added. This configuration file (if it exists) is read after */etc/rc.conf*, to allow per-service overrides. This optional functionality was added to allow automated third-party installation mechanisms to easily add configuration data.

   /etc/rc.conf.d/SERVICE may also contain variable assignments to override the variables defined in calling script (usually */etc/rc.d/SERVICE*), to provide an easy mechanism for a system administrator to override the behaviour of a given *rc.d* script without requiring the editing of the script. This feature has been used for purposes such as using */etc/rc.d/postfix* to start */usr/pkg/sbin/postfix* instead of */usr/sbin/postfix*, without editing the *rc.d* script.

   Migrating entirely away from */etc/rc.conf* to a multitude of */etc/rc.conf.d/SERVICE* files was considered, but no consensus was reached, and after a local trial, we decided that providing for the latter but retaining the former satisfies proponents of either side.

Thus, the order that configuration information for a given service *foo* is read in is as follows:

1. *foo* sources */etc/rc.conf*.
2. */etc/rc.conf* sources in */etc/defaults/rc.conf* (if it exists), and machine specific overrides of the defaults are added at the end of */etc/rc.conf*.
3. A per-service configuration file in */etc/rc.conf.d/foo* (if it exists) will be loaded. This allows for automated maintenance of */etc/rc.conf.d* configuration files, whilst retaining the popular */etc/rc.conf* semantics.

Implementation & aftermath

The system was implemented as described above in the design section, although the design was slightly fluid and did change as feedback was incorporated.

There are two elements to the post-implementation analysis; the human issues, and the technical details.

The human issues

There was a lot of feedback, debate, angst, flames, and hate-mail. The change has been one of the most contentious in the history of the project.

The first commits to the source code repository were made with the intention of providing a mostly complete implementation which was to be incrementally improved over a few months before the release of NetBSD 1.5.

Unfortunately, we made one of our largest implementation mistakes at this point; we didn’t warn the user-base that this was our intention, and the commits were seen as a ‘stealth attack’. This was partly because we felt that there had been enough debate and announcing our intentions would have delayed the project another few months for a rehash of the same debate (which had been going on for five years at that point).

After the initial implementation, various technical and ‘religious’ complaints were raised about the system. A summary of these is:

- "*The use of ‘magic’ functions [from /etc/rc.subr] is bad.*”

  It was felt that the code re-use that */etc/rc.subr* promotes was sufficiently worthy to justify its use. Additionally, a manual page was added describing the functions (see `rc.subr(8)`)

- "*Switching from /etc/rc is not the BSD way, ...*”

  This particular objection was expected; it’s a religious argument and the change was bound to annoy a certain section of the community.

  Robert Elz, a long time user and contributor to BSD, had a good point to make about ‘the BSD way’: “*[the BSD way is to] find something that looks to be better (in the opinion of the small group deciding such things), implement it, and ship it.*”

  In this case, the ‘small group’ was the NetBSD core team, who voted in unanimous agreement for the work, with the proviso that it would be tweaked and improved as necessary, which is what occurred.

- "*Why wasn’t a System V init.d implemented?*”

  This was covered above.

Because some of the detractors were quite vocal in the complaints, there was a perception for a time that the work was against a majority decision. This was far from the truth; many users and developers had become jaded with the discussion over the years and did not bother to argue in support of the change, since they agreed with it in principle, if not in implementation particulars. This was borne out by the level of support for the change in the time since implementation.

The technical details

The *rc.d* system comprises of the following components:

/etc/rc	System start-up script.
/etc/rc.shutdown	System shutdown script.
*/etc/rc.d/**	Individual start-up scripts.
/etc/rc.subr	Common shell code used by various scripts.
/etc/defaults/rc.conf	Default system configuration.
/etc/rc.conf	System configuration file.
*/etc/rc.conf.d/**	Per service config file.

/etc/rc

On system start-up, */etc/rc* is executed by `init`. If `init` is starting an automatic boot into multi-user mode, */etc/rc* is invoked with an argument of “autoboot”.

/etc/rc then calls `rcorder` to order the scripts in */etc/rc.d* that do not have a “nostart” `rcorder` keyword to obtain a dependency list of script names. */etc/rc* then invokes each script in turn with the argument of "**start**” to start the service.

The purpose of the “nostart” support is to allow (primarily third-party) scripts which are only to be manipulated manually (and not started automatically) to be installed into */etc/rc.d*. No scripts in the standard NetBSD distribution use this feature as yet.

/etc/rc.shutdown

At system shutdown, */etc/rc.shutdown* is executed by `shutdown`. `halt`, `reboot`, and `poweroff` do not call this script.

/etc/rc.shutdown then calls `rcorder` to order the scripts in */etc/rc.d* that have a “shutdown” `rcorder` keyword to obtain a dependency list of script names. This dependency list is then reversed, and */etc/rc.shutdown* then invokes each script in turn with the argument of “**stop**” to stop the service.

The rationale for this is that only a few services (such as databases) actually require a shutdown mechanism more complicated than the `SIGHUP` sent by `init` at shutdown time. Also, having every script perform “**stop**” slows down system shutdown as well as causing problems in other areas (such as cleanly un-mounting a ‘busy’ NFS mount once the networking services have been stopped).

/etc/rc.d/* scripts

The scripts in */etc/rc.d* are invoked by */etc/rc* (with an argument of "**start**") and */etc/rc.shutdown* (with an argument of "**stop**") in the order specified by `rcorder` to start and stop (respectively) a given service.

The following file naming conventions are used in */etc/rc.d/*:

ALLUPPERCASE — Scripts that are 'placeholders' or 'barriers', to ensure that certain operations are performed before others. In order of startup, these are:

- *`NETWORKING`* — Ensure basic network services are running, including general network configuration (`network`), and `dhclient`.
- *`SERVERS`* — Ensure basic services (such as `NETWORKING`, `ppp`, `syslogd`, and `kdc`) exist for services that start early (such as `named`), because they're required by `DAEMON` below.
- *`DAEMON`* — Before all general purpose daemons such as `dhcpd`, `lpd`, and ntpd.
- *`LOGIN`* — Before user login services (`inetd`, `telnetd`, `rshd`, sshd, and `xdm`), as well as before services which might run commands as users (`cron`, `postfix`, and `sendmail`).

foo.sh — Scripts that are to be sourced into the current shell rather than a subshell have a '*.sh*' suffix. Extreme care must be taken in using this, as the startup sequence will terminate if the script does.

bar — Scripts that are to be sourced in a sub shell.

Each script should support the following (mutually exclusive) arguments:

start — Start the service. This should check that the service is to be started as controlled by */etc/rc.conf*. Also checks if the service is already running and refuses to start if it is. This latter check is not performed by standard NetBSD scripts if the system is starting directly to multi-user mode, to speed up the boot process. If '**forcestart**' is given, ignore the */etc/rc.conf* check (but still determine if the service is already running).

stop — Stop the service if */etc/rc.conf* specifies that it should have been started. This should check that the service is running and complain if it is not. If '**forcestop**' is given, ignore the */etc/rc.conf* check and attempt to stop.

restart — Perform **stop** then **start**.

status — If the script starts a process (rather than performing a one-off operation), show the status of the process. Otherwise, it's not necessary to support this argument. Defaults to displaying the process ID of the service (if running).

rcvar — Display which */etc/rc.conf* variables are used to control the start-up of the service (if any).

Each script should contain `rcorder` keywords, especially an appropriate "PROVIDE" entry.

Other arguments for manual use by a system administrator (such as **reload**, etc) can be added on a per service basis. For example, */etc/rc.d/named* supports **reload** to reload `named`'s configuration files without interrupting service.

/etc/defaults/rc.conf, /etc/rc.conf, rc.conf.d/*

/etc/defaults/rc.conf contains the default settings for the standard system services, and is provided to facilitate easier system upgrades. End users should not edit this file.

/etc/rc.conf is the primary system start-up configuration file. It reads in */etc/defaults/rc.conf* (if it exists), and the end-user puts site-local overrides of these settings at the end of the */etc/rc.conf*. This makes it more obvious to differentiate be-

tween what is a system default and what is a site-local change, and provides similar functionality to FreeBSD's */etc/defaults* mechanism.

For a given service *foo*, it is possible to have a per-service configuration file in */etc/rc.conf.d/foo*, which is read after */etc/rc.conf.* This was provided to allow third-party installation tools to install a default configuration without requiring them to in-line edit */etc/rc.conf.*

Example */etc/rc.conf* entries for dhclient are:

```
dhclient=YES
dhclient_flags="-q tlp0"
```

To ensure that the system doesn't start into multi-user mode without the system administrator actually checking the configuration of the system, the variable **rc_configured** is set to "no" by default, and must be set to "yes" by the system administrator. If this is not set, the system will not boot into multi-user mode, and instead remain in single-user mode. The system installation tool `sysinst` makes this change for you when configuring a newly-installed system, but users performing manual installations or upgrades need to be aware of this.

As */etc/rc.conf* is a `sh` script, it is possible to put various shell commands into the script to conditionally set flags if necessary. Be aware, however, that if the script exits then any script that sources */etc/rc.conf* (such as the system boot scripts) will exit too. As */etc/rc.conf* may be loaded early in the boot sequence (possibly before */usr* is mounted), not all commands may not be available for use.

/etc/rc.subr

/etc/rc.subr is a shell script that's sourced by the */etc/rc.d* scripts. It contains 'helper' shell functions for commonly used operations:

- `checkyesno` *var*

  Return 0 if the variable *var* is set to `yes`, `true`, `on` or `1`. "yes", "true", "on", or "1". Return 1 if the variable *var* is set to `no`, `false`, `off` or `0`. Otherwise, warn that *var* is not set correctly. The values are case-insensitive.

- `check_pidfile` *pidfile procname* [*interpreter*]

  Parses the first word of the first line of *pidfile* for a PID, and ensures that the process with that PID is running and its first argument matches *procname.* Prints the matching PID if successful, otherwise nothing. If *interpreter* is provided, parse the first line of the file referenced by *procname*, ensure that the line is of the form:

  ```
  #! interpreter [...]
  ```

  and use *interpreter* with its optional arguments and *procname* appended as the process string to search for.

- `check_process` *procname* [*interpreter*]

  Print the PIDs of any process that are running with a first argument that matches *procname. interpreter* is handled as per `check_pidfile`.

- `load_rc_config` *command*

  Source in the configuration files for *command.* First, */etc/rc.conf* is sourced if it has not yet been read in. Then, */etc/rc.conf.d/command* is sourced if it is an existing file. The latter may also contain other variable assignments to override `run_rc_command` arguments defined by the calling script, to provide an easy mechanism for a system administrator to override the behaviour of a given *rc.d* script without requiring the editing of the script.

- `run_rc_command` *argument*

  Run the *argument* method for the current *rc.d* script, based on the settings of various shell variables. This is highly flexible, and supports many different service types. *argument* is argument describing the operation to perform (e.g., **start** or **stop**). The behavior of run_rc_command is controlled by shell variables defined before invoking the function.

- `run_rc_script` *file argument*

  Start the script *file* with an argument of *argument*, and handle the return value from the script.

  The execution of *file* depends upon the following checks:

  1. If *file* ends in '*.sh*', it is sourced into the current shell.
  2. If *file* appears to be a backup or scratch file (e.g., with a suffix of '~', '*#*', '*.OLD*', or '*.orig*'), ignore it.

3. If *file* is not executable, ignore it.
4. If the *rc.conf* variable **rc_fast_and_loose** is empty, source *file* in a sub-shell, otherwise source *file* into the current shell.

- `wait_for_pids` *pid* `[...]`

  Wait until all of the provided *pids* don't exist any more, printing the list of outstanding *pids* every two seconds.

In traditional System V *init.d* systems (e.g., Solaris), each script contains the code to determine if a script should be started or shut down, and often re-implemented the checks for a running process, etc. These scripts become difficult to maintain, and are often 1-2 pages long.

By using the functions in */etc/rc.subr*, the standard NetBSD rc.d scripts are quite small in comparison.

For example, the */etc/rc.d/dhclient* script (sans comments which aren't used by `rcorder`) is:

```
#!/bin/sh
#
# PROVIDE: dhclient
# REQUIRE: network mountcritlocal
# BEFORE:  NETWORKING
. /etc/rc.subr
name="dhclient"
rcvar=$name
command="/sbin/${name}"
pidfile="/var/run/${name}.pid"
load_rc_config $name
run_rc_command "$1"
```

It is not mandatory for scripts to use these functions. An ordinary shell script (with the appropriate `rcorder` control comment lines) which supports the arguments **start** and **stop** should work at system start-up and shutdown without modification. In order to be consistent with the existing rc.d scripts, in may help to also support **restart**, **status**, **rcvar** (if appropriate), as well as the '**force**' prefix.

rcorder

The ordering of the scripts in */etc/rc.d* is performed by `rcorder` (located in */sbin/rcorder*), which prints a dependency ordering of a set of interdependent scripts. `rcorder` reads each script for special comment lines which describe how the script is dependent upon other services, and what services this script provides.

Example `rcorder` comment lines for */etc/rc.d/dhclient* follow:

```
# PROVIDE: dhclient
# REQUIRE: network mountcritlocal
# BEFORE:  NETWORKING
```

In this case, `dhclient` requires the services '`network`' (to configure basic network services) and '`mountcritlocal`' (to mount critical filesystems that are required early in the boot sequence, usually */var*), provides the service '`dhclient`', and must run before the barrier script */etc/rc.d/NETWORKING*.

It is possible to tag a script with a keyword which can be used to conditionally include or exclude the script from being returned by `rcorder` in the result. */etc/rc* uses this to exclude scripts that have a "nostart" keyword, and */etc/rc.shutdown* uses this to only include scripts that have a "shutdown" keyword. For example, as `xdm` needs to be shut down cleanly on some platforms, */etc/rc.d/xdm* contains:

```
# KEYWORD: shutdown
```

The `rcorder` dependency mechanism enables third-party scripts to be installed into */etc/rc.d* and therefore added into the dependency tree at the appropriate start-up point without difficulty.

Enhancements since NetBSD 1.5

rc.d has been improved since its first release in NetBSD 1.5:

- Change various *rc.d* scripts to use `rcorder`'s "BEFORE" keyword, instead of having a 'barrier' script "REQUIRE" the script. This allows scripts to be removed without editing the barrier scripts.
- Various performance enhancements.
- Implement `wait_for_pids`, and use in the default **stop** method, rather than assuming that the process was stopped.
- Skip backup and scratch files in `run_rc_script`
- `run_rc_command` improvements:

  Add variables **procname** and **command_interpreter** to provide greater flexibility when starting-up programs which are interpreted scripts, and *argument*_**postcmd** which contains shell commands to be run if *argument*_**cmd** succeeded.

Expose various “internal” variables for use and manipulation by methods, including: **rc_arg**, **rc_flags**, **rc_pid**, **rc_fast**, and **rc_force**.

Future Work

I’d like to implement ‘`runrcto`’ to allow you to start up (or shut down) services from service *A* to service *B*. This would allow you to start in single user mode, and then start up enough to get the network running, or start all services until just before ‘multi user login’, or just those between ‘network running’ to ‘database start’, etc. This could be a fairly simple system, and would provide most of the functionality that most people seem to want run states for.

We also need a functional `chkconfig` command (similar to the equivalent command in IRIX), to manage */etc/rc.conf.d* by displaying a setting or changing its value. An incomplete implementation exists at this time.

I encourage other systems that are still using a monolithic */etc/rc* and who would like to resolve some of the similar issues NetBSD had, to consider this work. I would like to liaise with the maintainers of those systems to ensure as much code re-use as possible.

Conclusion

NetBSD 1.5 has a start-up system which implements useful functionality such the ability to control the dependency ordering of services at system boot and manipulate individual services, as well as retaining useful features of previous releases such as */etc/rc.conf*.

This work was extremely contentious and difficult to implement because of this contentious nature. The implementation phase did provide valuable insight into some of the difficulties involved in the design and development of large open source projects.

In the long run I believe that this work will have benefits for a majority of users, both in day-to-day operation of the system as well as during future upgrades from NetBSD 1.5 and later releases.

Availability

This work first appeared in NetBSD 1.5, which was released in December 2000. It was enhanced for NetBSD 1.6 and continues to evolve based on feedback from users. It is used by NetBSD pkgsrc on various platforms. It has since appeared in FreeBSD 5.0 albeit with minor changes that were required to suit their needs.

The CVSweb interface can be used to browse the work and its CVS history.

Acknowledgements

Many people contributed to the discussions and design of the current system.

However, some people in particular provided important elements: Matthew Green for finishing `rcorder` and providing the initial attempt at splitting */etc/rc* into */etc/rc.d*, and Perry Metzger for the idea of providing dependencies using a "PROVIDE" and “REQUIRE” mechanism, and for the initial `rcorder` implementation.

Minutes of AUUG board meeting, 19 February 2004

Location	IBM, Canberra

Attendees:

Elizabeth Carroll	EC
Jonathon Coombes	JC
Andrew Cowie	AfC
Gordon Hubbard	GH
Steve Landers	SL
Greg Lehey	GL
David Purdue	DP
Stephen Rothwell	SFR
Michael Still	MS

Meeting started at 10:00 am

1. Apologies

   Adrian Close

2. President's Report

   The highlight of this quarter was the "Linux and Open Source in Government" mini-conference at the Linux.Conf.Au. It marks our first conference in cooperation with Linux Australia, though the majority of the work was done by AUUG. The conference was a great success, in marked contrast to the relatively low-key Government thread at the 2003 annual conference, and we have high hopes of another good turnout at the 2004 annual conference.

   In addition, AUUG has been accepted as a member of the IT Council of South Australia, with myself as the current representative. This announcement was made only last week, and I have yet to participate in a meeting.

   Unfortunately, that was about all there was to report. At the Chapter Council in August we had agreed to sponsor some additional chapter events during the year. The Security Symposium will take place in Canberra on 20 February, and we have hopes for an Australian Open Source Symposium in Perth, but in particular the Installfests intended for the beginning of the University year have not occurred. We had hoped to gain numerous student members in this manner, so this is particularly unfortunate.

   At the last meeting, I reported on the Asian Open Source Symposium, in which we had hoped to participate. I regret to report that my name was taken off the mailing list with no explanation, and that further mail messages to the organizers have gone unanswered. It looks like they don't want us after all.

   Last quarter I wrote: "This quarter was marked by a surprising lack of activity, punctuated by a couple of highlights.". This seems to be the flavour of the year. At the last meeting we identified that action items were not being acted on. As a result, we changed from the previous online representation to a Wiki. Looking at the current state of that document, it seems that almost none of the items have been done. We seem to have serious issues with this particular matter, and it seems to be getting worse rather than better. I propose that we enforce the weekly revision of the action items, on which we had already agreed.

   On a personal note, I'm finding it difficult to find time for the office of president. I will not be nominating myself for any position on the board next year. I encourage other members of the board to take my place.

3. Secretary's Report

   Not available at time of meeting.

4. Treasurer's Report

   A copy of the Treasurer's report was submitted.

   GL asked about a comparison of accounts from this time last year against this time this year.

   John Lions account was discussed. An organisation needs to own this as it is in AUUG's name now, and we are receiving interest on this. The signatories to the account do not claim it as their asset, neither has AUUG to date. It is money that AUUG holds in trust. We need to look at this from a taxation point of view. AfC said that it could be looked as both an asset and liability. For the foreseeable future we should nominate the current signatories, who are Lucy Chubb, Greg Rose and possibly David Purdue. Therefore we appoint them until further notice to be the trustees. Motion by GL, seconded by GH, carried.

The address for the bank account needs to be changed as it is currently using the old AUUG PO Box and being re-directed.

Action: GH to organise change of address.

A comparison between the 2002 and 2003 accounts was looked at. It was noted that memberships were down. This will be discussed later.

Action: GH to check figures for the Security Symposium. There was no Security Symposium therefore that has reduced some of our figures. Some figures need to be looked at.

SL asked that we see conference budgets to get an idea of events.

Motion to accept, SL, seconded DP, carried. AfC dissented due to lack of confidence in the numbers. GH responds that due to recovery from previous years this is a work in progress. AfC is not confident in the way the figures are presented.

Action: GH, AfC and EC to liaise and come up with a new draft by 5 March.
Action: AfC to spend a day with EC and get up to speed on MYOB.

5. Business Manager's Report

   AUUG 2004

   Date and venue has been organised. Call for Papers has been sent out. Keynote speakers yet to be confirmed. Sponsorship packages to be confirmed before approaching potential sponsors.

   Security Symposium

   At time of writing this report, there are approximately 35 people who will be at the Security Symposium. The event is being sponsored by Fortinet.

   Other Events

   John Terpstra has approached AUUG regarding running a potential roadshow in June. The Board needs to make a decision regarding this, and possible planning needs to start.

   Memberships

   Membership renewals for December are still coming in. Those not received by end of February will be removed from the database.

   Electronic Services

   This is a topic that will be discussed at the meeting. I find that it is really in AUUG's interest to have online systems available for conference registration and memberships.

   Other

   These are the key areas that I have been addressing recently, in addition to general issues, most of which will be discussed under the other topics at the meeting.

   AfC asked about what we do about memberships that have not renewed. EC responded that they are followed up via email and phone, and that after 2 months membership benefits cease, thus allowing for late renewals to come in. It was agreed that we should let members know that benefits cease the moment their membership expires.

6. Minutes of previous meeting

7. Action Items

   Done—on wiki.

8. AUUG 2004 Conference

   It was agreed that this will be discussed on the conference call on Tuesday 24 February.

   - Speakers
   - Sponsors
   - Budget
   - Other

   Action: DP to include AUUG Board on the `auug2004prog` mailing list.

9. Events

   - Security Symposium

   Numbers have increased since time of writing of Business Manager's Report, close to 50 attendees.

10. Other—J Terpstra roadshow

    Action: GL to come up with suggestions and talk to John Terpstra by 29 February.
    Action: EC to tell J Terpstra next week that yes we will go ahead and that GL will be in touch by end of month. Financial issues to be run by GH.

11. Official AUUG position w.r.t. to OSS

12. OSS in Government 2005

    No decision made. Feeling to make LCA more technical. Regarding OCG, would AUUG be interested in running it in conjunction with LCA. Deferred.

    Action: MS to come up with more detail by the next meeting. Need to look at the revenue flow.

13. Chapters and Chapter Activities

    - AUUG promotional material at local meetings

      Action: GH to discuss AUUG merchandise by end of March.

14. Service of engrossed documents

    Done.

15. AUUGN

    GL is concerned about timeliness of AUUGN. Need to discuss the March issue.

16. Web pages—On-line services

    - Minutes

17. Web pages—On-line services

    - Minutes
    - Renewals and Memberships
    - Event Registration
    - Passwords—queries regarding use received from members

    On-line services were discussed, including on-line services required.

18. Australian Open Source Awards

    Michael Paddon has been offered the position of Chairperson, which he has accepted.

    Action: DP to chase MP on this.

19. Accounts

    Bank account signatories including nominated AUUG member (non-Board member) in case of emergency. It was agreed to leave the signatories as is, and update them once the new Board is in place.

    Signatories for John Lions account including change of address. Already discussed.

20. Linux Australia/AUUG cooperation including road shows

    Also per SL email, "Open Source Symposium that we've discussed for Perth mid year. In fact, it was always my intention to get the PLUG and SLPWA people involved—a collaboration with LCA would just make it easier."

    SL stated this might be a good chance to co-operate with LCA or someone else. For this we require a programme committee and venue. The next location will be in Perth. SL will be the programme chair. Look at dates for the Open Source Symposium.

    Action: SL to talk to SLUG and Slipway.
    Action: SL to discuss dates with EC.

21. OCG mini-conf for next LCA

22. IT Council organisation of a national conference on community ICT issues on May 3-4 at the Hyatt Regency, Adelaide.

    Action: GL to forward the announcement re IT Council organisation of a national conference on community ICT issues to auugannounce by Sunday.

23. Financial / Unfinancial members and receipt of AUUGN

    Motion by AfC that we do away with any grace period, members become unfinancial the moment their membership expires. Seconded MS, carried by majority.

24. Corporate Sponsorship

    In camera

25. SL: The role of the board vs executive. I think we clearly need to differentiate between the board members acting as a board (i.e. deciding direction) vs as an executive (putting the policy into practice). In addition, AUUG Board Members and general protocol.

26. AUUG and endorsing the spirit of commercial projects, including applications for funding

    Five of eight members voted "yes". There are obviously some details of the proposal that weren't clear, though it is believed that we all interpret them in the same way. We need to ratify this at the board meeting.

    All use of the AUUG logo must be approved by the AUUG Board.

    Action: GL to email the Board with the final wording.

    Moved by MS, seconded SFR, carried.

27. US$1000 sponsorship to Asiabsdcon

    GL stated they asked for help with the event, he does not recall being asked about the sponsorship. GL is not inclined to go ahead with this. The Board agrees.

28. Lions Award

    MS to take this over.

    Action: MS to liaise with EC about Lions Award by Wednesday and get letters out by then.

    Action: SFR to find out who in IBM they talk to about their student competition.

29. Elections and New Board

    There was discussion about who is thinking about being on the Board next year.

30. Other Business

    - CD's

      MS is currently following up cd's that went out with AUUGN with error.

      Looked at topics to go out with AUUGN.

      Action: MS to send a message to auugannounce to ask if it is viable to send out dvd's.
      Action: MS to find out cost of robot with dvd.
      Action: MS to look at other potential suppliers.

    - AUUG Board Member Liability

      Action: EC to follow up with Insurance Broker, check liability of Directors and General Board members. AUUG is incorporated in Victoria.

Meeting closed: 7 pm.

Next meeting: To be advised.

Living Subversively

Michael Paddon <mwp@qualcomm.com>

True Confessions of a CVS User Seduced by the Subversion Revision Control System

What is Subversion?

Subversion[1] is an open source revision control system, with the goal (as described by its creators) of becoming a "compelling replacement for CVS". Given the significant market share held by CVS *[2]* in both the open source and proprietary development communities, thise is an ambitious project indeed. Interestingly, many Subversion developers have previously participated in the CVS project, thus lending a degree of credibility to their goal.

The Subversion team does not claim to have met this goal. This paper was written with reference to Subversion 0.22, which was current in May 2003. By July 2003, the current version was 0.26, with a point release every 2 to 3 weeks being typical. Clearly, this is a work in progress, with full functionality promised for version 1.0.

However, the Subversion project itself has self hosted (i.e. the master source code has been stored and managed in a Subversion repository) since September 2001. It seems that the developers trust their system enough to put it into production use with code they care about. In many ways, this is the best possible advertisement for a revision control product.

Subversion is also in wide use by third parties. The project website lists 37 instances of Subversion installations, for projects ranging from small, single person efforts to large code bases with numerous developers. These examples probably represent only a small fraction of Subversion sites, and thus one can conclude that the tool has a sizable user community already.

[1] The Subversion project home page: *http://subversion.tigris.org/*

Why Switch From CVS?

CVS works well and is a mature product. Furthermore, in over a decade of heavy use, I have experienced no data loss or corruption whatsoever in using this tool. Anecdotal evidence suggests that my experience is typical. Why switch at all?

While CVS is extremely good at what it does, there are desirable features it does not have and behaviours it does not exhibit. Some issues of particular importance to me are:

- Incomplete support for directories. Anyone who has used CVS knows that directories are not version controlled (like files), leading to unwieldy workarounds that are not always effective.
- No support for file renaming. To rename a file, you have to first copy it and then remove the old version. This discards history information.
- Attic artifacts. Files that are removed are stored in a special directory called the "attic". The existence of such files should be invisible, but unfortunately make themselves felt under some circumstances as a side effect of the implementation.
- File based transactioning semantics. CVS operations occur at the level of file granularity. Unfortunately, this means that two users performing operations on large groups of files can interfere with each other and leave the repository in an inconsistent state.
- Rudimentary support for metadata. CVS supports metadata in only an *ad hoc* way (e.g. file access bits, RCS states, etc.), with no versioning whatsoever.
- Awkward handling of binary files. One has to manually remember to mark a file as binary or the file may be munged on checkout. Furthermore, the differencing algorithm used by RCS (which underlies CVS) is designed for line oriented text files, leading to significant storage wastage when a binary file changes slightly.

Subversion addresses all of these issues, and promises more besides. This was a compelling enough proposition to risk experimenting with a new, less mature tool, and to climb the inevitable learning curve that accompanies any novel technology.

Installation is Easy...

Installation of Subversion on Linux[1] was easy. Indeed, a version of the tool comes standard with RedHat 9, which was a pleasant surprise. The new user (or evaluator) would probably find this installation more than sufficient. However, I thought it desirable to upgrade to the latest and greatest at the time, which involved downloading several RPM's from the Subversion web site and installing.

The process was not entirely painless, as Subversion depends on the latest versions of several other libraries and tools (such as Apache's httpd). All the necessary RPM's are available from the web site, but the inevitable minor conflicts occurred when installing. While this is nothing an experienced systems administrator would have any trouble with, it is not recommended for the faint of heart.

Similar pre-built packages exist for Debian Linux, SuSe Linux, FreeBSD, MacOS X and Windows.

...Compilation is Harder

Unfortunately, Subversion was not available as a pre-built package on the my operating system of choice: OpenBSD. Therefore, compilation from scratch was necessary.

In order to build Subversion, the latest versions of the following packages were necessary:

- Apache httpd
- Berkeley DB (a database library)
- GNU libtool (a library linking tool)
- Neon (an HTTP and WebDAV client library)
- Subversion

The Subversion documentation does not recommend using older versions of these packages, and I decided to take this advice at face value. Unfortunately, this meant downloading, compiling and installing each package in the specified order. While this took some hours, the process itself was straightforward and yielded a working Subversion installation on the first pass.

[1] FSF purists should pipe this document through `sed 's:Linux:GNU/Linux:g'`.

Configuration

Subversion was easy to configure. Creating a repository was accomplished with a single *svnadmin* command. Once created, the repository may be accessed ia several different protocols (specified by URL), the simplest of which is direct file system access. Thus, `file` URLs may be used to access a repository on a locally mounted filesystem, with no further configuration necessary.

Remote access is supported via two mechanisms:

1. A standalone server, *svnserve*, accessed via a `svn` URL. *Svnserve* may be run as a daemon, started on demand by *inetd*, or launched from the command line. The latter mode is especially helpful if you intend to access Subversion through a *ssh* tunnel. Many existing CVS users will be comfortable with this model.
2. Through an Apache HTTP server, using the WebDAV protocols. This is more complex to set up, as appropriate stanzas must be added to the *httpd* configuration. However, the extra complexity has two major additional benefits:
    - A web portal for browsing the repository (somewhat akin to *cvsweb*).
    - Limited interoperability with third party WebDAV clients. At the moment this is of theoretical interest but, as WebDAV matures and becomes a more common protocol for managing object across the web, this may become a major benefit.

I chose to configure Apache as the primary repository access mechanism, since remote access was required and the capability to view the repository with a standard browser was highly appreciated.

Security

For most users of a revision control system, security is a key consideration. A body of source code has significant intrinsic value and dependable authorisation, access control and audit trails are mandatory.

Currently, Subversion implements no security whatsoever. In other words, it relies on the underlying transport to implement this functionality. For instance, `file` and `svn` URLs rely on the local file system security semantics, while `http` URLs rely on Apache's authorisation and access control mechanisms.

This approach allows for only very coarse security management: treating the entire repository as a unit, with the only choice being read-only versus read-write access.

In an environment where all or nothing access is acceptable, this is sufficient. Many small to medium size development groups already operate their CVS repositories in this manner, so Subversion's limitation may be quite acceptable. Larger (or more paranoid) groups, however, will find the lack of internal access controls problematical.

Access control lists are a documented feature, promised for Subversion 1.0. Right now, the best way to work around this lack of functionality is to create a separate repository for each security domain that requires compartmentalisation.

My reference configuration used standard Apache authentication mechanisms (passwords or client side certificates), combined with SSL transport to protect data in transit. This provided a convenient balance of security, logging and remote access within the deployment environment. As with all security issues, the reader is encouraged to design their own, tailored solution rather than copy the mistakes of others.

Tags and Branches

Subversion is similar in many ways to CVS; so much so that a CVS user can simply replace "`cvs`" with "`svn`" in their commands and proceed with few surprises, none nasty. However, one area where Subversion differs greatly is with tagging and branching, and a good understanding of the new approach is fundamental to how a repository is structured.

Most CVS users are aware that they can attach tags to files (or groups of files). This is most often used to label a "snapshot" of whatever is being developed for the purposes of communication, quality control and release engineering. Sophisticated users are aware that there are also special "branch" tags that allow files files to be developed along diverging paths.

All long term CVS users have probably been frustrated by the fact that tags themselves are not versioned. Thus, if a user mistakenly tags the wrong thing (or worse still, moves a tag incorrectly), it is difficult and perilous to undo the consequential damage. I have observed, firsthand, the difficultly that many developers have with branch tags in particular and the associated magic rules that apply to branch revision numbers.

Subversion has done away with all of this confusion... by removing all tagging functionality. While this has a certain minimalist charm, a quite reasonable question is: what do you use instead of tags? The answer is the “smart” copy.

Smart Copies

Perhaps more familiar to operating system aficionados as "copy-on-write", this mechanism allows a developer to copy an entire source tree quickly and with minimal overhead. The repository is aware of the relationship between the original and the copy and only one set of data is kept. Thenceforth, only the changes in either the original or the copy need to be stored.

Instead of tagging, you copy a tree. If your snapshot is left unchanged, then you have an analog to a traditional tag; if you change the tree, you have effectively branched. Of course the smart copy is fully versioned, unlike CVS tags.

Because snapshots and branches are now visible in the file system, this approach requires some forethought in designing the repository hierarchy. However, this is no more effort than what is required to design and manage a sustainable tagging regime. Indeed, a strong case can be made that smart copies are simpler, more visible and (most importantly) more intuitive than tags.

Repository Design

The Subversion documentation recommends structuring your repository with directories such as *trunk* for the main development trunk, *branches* for development branches, and *tags* for (you guessed it) tags. Furthermore, the documentation describes two basic approaches to creating this structure:

1. Repository wide, with project trees as children of exactly one set of these directories.

2. Per project, with each project tree containing its own set of these directories.

While none of this structure is mandated by the tool, having obvious and understandable conventions promotes better communication and teamwork. I adopted the conventions as recommended, and they have worked well. After some deliberation and experimentation, I felt that a per project model was superior for my particular needs, primarily because each project’s hierarchy is distinct and can be treated as a standalone unit. In addition, I created an additional topmost directory layer, in my repository to act as a superstructure. At the time of writing, this layer looked like:

- `README`: a file describing the repository structure and usage conventions.
- `admin`: a tree of repository administration files, scripts, etc.
- `hosts`: a tree, with a subdirectory for each machine, of host configuration files.
- `people`: a tree, with a subdirectory for each person, for personal files.
- `projects:` a tree, with a subdirectory for each project, for development files.
- `www`: the support web site served by the repository machine, containing documentation, Subversion clients, etc.

Obviously, each site’s repository design will be unique, however I found it worthwhile to design a structure to accommodate non project material as well as the traditional contents of a repository. This encourages *everything* to be stored under revision control, which I regard as highly desirable.

A Developer’s Perspective

As a developer, Subversion is easy to use. The online documentation is excellent, the tool acts predictably, and it is intelligently designed to minimise self inflicted injury. For developers with existing CVS experience, it is even easier since Subversion’s commands are both syntactically and semantically similar.

In practice, this means that you can pick up Subversion and use it immediately, without needing to read boring instructions. Every now and then, a quick dip into the functionally organised manual suffices to climb the learning curve quickly and painlessly.

Import Performance

If using the tool is easy, then what about performance? How does Subversion stack up against CVS? Does it scale to large projects?

To answer these questions, I created virgin Subversion and CVS repositories which were accessed through the local file system (i.e. all network overheads were factored out)[1] I then took a significant body of source I had lying around—the OpenBSD 3.3 kernel, weighing in at about 85 megabytes—and imported it. The observed timings (in seconds) for ths operation were:

System	Real	User	System
CVS	25.586	0.320	0.680
Subversion	114.823	32.240	2.090

It is clear that importing into Subversion is far more expensive than CVS. This is probably due to the overhead of the Subversion transactional system, in which the entire import is a single (large) database transaction. Nevertheless, a couple of minutes to import such a large body of source is not unreasonable.

Interestingly, the CVS repository grew to 81 megabytes,[2] while the Subversion repository inflated to a hefty 176 megabytes. It seems that there are disk space overheads to a full transactional model.

For most people, this is a minor issue due to the low (and falling) cost of this disk. Additionally, a significant chunk of this space is presumably used by transactional logs,[3] and is reused for later transactions.

Checkout Performance

Once the source was imported, the next obvious step is to check it out:

System	Real	User	System
CVS	42.371	2.080	1.490
Subversion	126.609	38.320	4.450

Interestingly enough, Subversion took roughly three times longer for the checkout than CVS. Digging a little deeper explained why... the CVS checkout was around 85 megabytes, but the Subversion checkout was a whopping 222 megabytes. What was all this disk being used for? Apparently, Subversion stores substantially more control information for every directory, presumably to support its richer semantics.

Commit Performance

While large scale imports and checkouts are occasional operations, commits of incremental changes are a developer's bread and butter. How does Subversion stack up when running a commit on the entire tree in which one file has been changed?

System	Real	User	System
CVS	6.920	0.430	0.380
Subversion	9.830	0.670	0.370

Again CVS was the winner, being around 40% faster. However, in practice a few seconds against a commit of a tree this size is likely to have little practical impact. Again, the extra functionality of Subversion seems to come at a price.

Update Performance

Another common operation is an update, to pull in changes from the repository. To simulate this I deleted a file from the working copy, and performed a tree wide update to recover it.

System	Real	User	System
CVS	13.530	0.350	0.260
Subversion	6.913	0.170	0.070

For the first time, Subversion came in ahead of CVS. Updates clearly leverage the underlying database model, leading to better efficiency.

A Release Engineer's Perspective

Release engineers are typically heavy users of a revision control system and, since they tend to deal with source code in large chunks, the scalability of coarse, tree-wide operations are extremely important to them.

In particular the performance of tagging operations are critical, as they form the basis for a release lifecycle. I measured the time require to tag the entire tree:

System	Real	User	System
CVS	18.019	0.890	1.220
Subversion	0.113	0.010	0.000

Subversion was clearly *much* faster than CVS in tagging (and hence branching) operations. This is in spite of the fact that they are a fully versioned

[1] All benchmarks were performed on RedHat 9 Linux running on a 2.4 GHz Pentium 4 CPU with 512 MB of memory and ATA-100 disk. Timings were generated using the *time(1)* command.

[2] This is slightly smaller than the original source due to CVS ignoring some files, by default.

[3] 176 megabytes is suspiciously close to twice the raw source size.

operation, and thus substantally more functional than in CVS.

Release engineers are obviously going to win big from a switch to Subversion, and should probably be one of the key advocates within an organisation for such a change.

A System Administrator's Perspective

Subversion is low maintenance. Once set up and configured, it just works. The only administrative requirement is regular backups[1] Because Subversion is built on a database, there are two distinct ways to backup. The most efficient is to copy the database files in a documented magic order that maintains integrity. This is so cheap, it can be done regularly by *cron*, or even on every commit.

The downside is that the image will not necessarily work with other versions of Subversion, as the underlying schema may have changed. For archival storage, a full database dump (for which tools are provided) is recommended. Again, this is easy to set up as an automated job, and does *not* require repository downtime to execute.

I elected to use a automatically scheduled daily dump as the primary backup mechanism.

Divers Oddments

There is no doubt that the most immediately attractive aspect of Subversion is the solid transactional semantics. Commits exhibit the ACID[2] properties which addresses many of the problems that developers experience with CVS, especially in large environments.

At the same time, Subversion is similar enough to CVS to make a transition low impact and painless enough to succeed. The familiar "copy-modify-merge" model is maintained, and operation is intuitive to existing CVS users.

The merge and conflict mechanism has been slightly changed in Subversion, to prevent the all too common scenario of a developer failing to notice a conflict (or simply ignoring one) and committing improperly merged files. Subversion requires that developers explictly mark conflicts as resolved, via the *resolve* subcommand, prior to a commit.

Many other minor aspects of operation have been improved over CVS in this way, without breaking the behaviour that most developers would expect. The *switch* subcommand, for instance, is a convenient and simple way of switching between branches that has no analog in CVS (although the functionality is available with the right incantation). Another example is the *revert* command for throwing away local changes. Small refinements like this add up to big usability improvements.

Perhaps the most visible change to which developers must adapt is that fact that files no longer have revision numbers. Instead the entire repository's revision number increments after every commit. This takes a little bit of getting used to, but perhaps better reflects dealing with a body of source rather than a bunch of files.

Conclusion

My experiment with Subversion spanned several months, and involved total reliance on it for revision control functionality. In that time, the entire experience was positive. Nothing broke, and after being able to manipulate directories in a properly version controlled fashion, I doubt I'll consider moving back to CVS anytime soon.

During the experiment, I started with an empty repository. Sites with an existing investment in CVS will be encouraged to know that Subversion comes with a tool, *cvs2svn.py* that intelligently imports both the data and the revision history into a Subversion environment.

Although there are GUI interfaces to Subversion available, I found the command line interaction to meet all my needs. This may reflect my preferences and/or prejudices.

The only issue I was dissatisfied with was security. The all-or-nothing access to repositories does not still well with my hyper-paranoid mindset, and I look forward to the promised ACL functionality. That being said, Subversion is no worse than a typical CVS installation in these terms.

As regards performance, Subversion and CVS clearly have different strengths. I was a little disappointed, at first, when I discovered that their overall performance is comparable. I was hoping for a quantum jump in speed which was, in retrospect, a naive expectation. However, Subversion *is* quite fast enough to manipulate large source bases effectively., and people who tag a lot will love it.

[1] Remember: if you don't back up, you will be sorry.

[2] Atomicity, Consistency, Isolation, Durability

On the whole, I would recommend Subversion for new projects. It is a significant improvement over CVS, and I found it stable enough for everyday use[1] Certainly, it is at least worthy of ongoing evaluation, as it seems that it is, indeed, destined to replace CVS in the fullness of time.

This paper is reprinted with the author's permission from the proceedings of the AUUG 2003 conference.

The new AUUG Board of Directors

With the new financial year, AUUG has a new Board of Directors, whom we used to call the Management Committee.

Officially, this is the June 2004 edition of AUUGN, so the board of directors shown on page 2 reflects the previous board. On 1 July, Andrew Cowie and Stephen Rothwell left the board, and Greg Lehey resigned as president. Grant Allen joined as a new member. This leaves two positions vacant. The new board will coopt members in the first meeting of the new year, and they will be presented for ratification at the AGM in Melbourne on 2 September.

It's been a few years since we last introduced our new members, so it's probably time to tell you a little bit about all the members. Here's the first instalment.

Grant Allen

Grant Allen is the newest member of the AUUG board of directors, having joined on 1 July 2004.

At a young age, Grant was introduced to the world of computing with a TRS80. Brief flirtations with Vic 20's, Apple II's and the odd Commodore 64 laid the path to his first acquaintance with UNIX in the 1980s at university. Here he cut his teeth on "state-of-the-art" Apollo Domain machines, Sun SPARCStations and the like. He enjoyed the experience so much (and failed to move quickly enough at the right moment), that he was co-opted as a member of staff for 5 years.

After leaving academia, Grant joined the Australian Public Service, and worked on a number of large projects involving database development and management. It was here that the field of database administration crossed his path, and has dominated his professional life—not to mention spare time and nightmares—ever since. After enjoying the highs and low offered by projects as diverse as superannuation and Navy ship building, he eventually moved to the private sector. For the past decade, he has worked for TOWER Software as their principal database consultant, assisting the company's R&D department, and customers, with the vagaries of database management on platforms including Solaris, AIX, HP-UX, OSF1, Irix, Dynix, Unixware and Tru64.

Linux became an object of fascination for Grant in the late 1990s when leading database vendors such as Oracle and IBM ported their products to it. After performing the ubiquitous dabbling in building custom kernels, forcing X to run on antique hardware, and equally masochistic endeavours, he moved on to more humble pass times. He currently participates in a number of open source development activities, including minor projects for PostgreSQL.

David Purdue

David Purdue is the President of AUUG. He has been on the AUUG Board for just over a decade, and has held every board position except Treasurer. He also edited AUUGN for a while in the 80's.

By day David works for Sun Microsystems looking after support for major global accounts. By night he Morris Dances and plays in the folk band Boadicea's Bad Boys (CD's on sale at the conference ;-)).

Steve Landers

Steve Landers is the Vice President of AUUG. Steve is the Senior Consultant at Digital Smarties. He is active within the Tcl/Tk community and has over twenty years experience in the UNIX and Open Software marketplace, being a founding member of both AUUG and SAGE-AU

Greg Lehey

Greg Lehey is immediate past president of AUUG. His day job keeps him busy with kernel programming. He's done most things in his time, and he's active in the FreeBSD and NetBSD developer communities. He also represents AUUG at the IT Council of South Australia. In what spare time he has, Greg does all sorts of strange things. See *http://www.lemis.com/grog/* for more details.

[1] Naturally, ensure you have an effective backup regime in place before, during and after transition.

Letters to AUUG

This column contains selected messages from the AUUG mailing lists, notably `talk`. *To sign up for this mailing list, visit the mailman pages at http://www.auug.org.au/mailman/listinfo/talk.*

From: **Frank Crawford**
  <frank@crawford.emu.id.au>
To: **auugn@auug.org.au**
Subject: **AUUGN95 Proceedings Wanted**
Date: **Mon, 28 Oct 2002 22:24:39 +1100**

Folks,

As part of the indexing of AUUGN and other AUUG publications (see *http://www.auug.org.au/publications/auugn/search-auugn.html*) I require a copy of the Proceeding from AUUG95. This was one of AUUG's biggest conference and there are no copies of these proceedings left, but it is the only one I am missing from the last 12 years (it has disappeared during various moves). If you have a copy, please contact me and we can discuss details.

Thanks
Frank Crawford
`<frank@crawford.emu.id.au>`

Morris worm

From: **conz@cyber.com.au (Con Zymaris)**
Date: **Thu May 6 14:25:10 2004**
Subject: **[Talk] Media Release: OPEN SOURCE USERS UNAFFECTED BY SASSER WORM - THE INTERNET KEEPS GOING DESPITE FLAWED PROPRIETARY SOFTWARE**

...

The first worm, by Robert Morris Junior, son of a senior NSA computer security expert and Unix pioneer, occurred in 1988. Even though it was not malicious and accidentally escaped from a lab, it brought the Internet to its knees for a few days. It directly caused the creation of a number of agencies, primarily CERT - Computer Emergency and Response Team. What the Morris Worm did clearly demonstrate is that there are substantial advantages for any organisation in using operating systems, middleware and applications from more than one codebase. Organisations who had a variety of platforms were able to keep part of their computing infrastructure going.

Date: **Mon, 10 May 2004 14:44:38 +1000**
From: **David Purdue <david.purdue@auug.org.au>**
To: **Con Zymaris <conz@cyber.com.au>**

This release fails to mention that the Morris Worm propagated by exploiting weaknesses in Sendmail, an open source program.

So it could also be said that what the Morris Worm did is clearly demonstrate that software being open source does not imply that it is immune to virus/worm attack.

If the real lesson is that I should source my applications from multiple code bases, what is the alternate codebase for something that does the same job as Apache?

Date: **Tue, 11 May 2004 08:07:12 +1000 (EST)**
From: **Stephen Jenkin <sjenkin@canb.auug.org.au>**

On Mon, 10 May 2004, David Purdue wrote:

> Just to play devil's advocate... <SNIP>

> This release fails to mention that the Morris
> Worm propagated by exploiting weaknesses in
> Sendmail, an open source program.

rsh, *fingerd* and *sendmail*. There were binaries for VAX & SUN.

http://www.snowplow.org/tom/worm/infect.html

They were all well known & reported security holes the vendors would not patch ;-)

We haven't seen anything as sophisticated in the Windows world - multi-platform, multi-exploit, rate controlled (that didn't work so well).

Date: **Tue, 11 May 2004 14:32:12 +1000 (EST)**
From: **David J N Begley <d.begley@uws.edu.au>**

Earlier today, Stephen Jenkin wrote:

> We haven't seen anything as sophisticated in
> the Windows world - multi-platform, multi-
> exploit, rate controlled (that didn't work
> so well).

Can you imagine how bad things would be if this *did* eventuate? One of my fears is that these two-bit virus/worm authors will be superceded by a *real* programmer who knows what they're doing.

The sad thing is, none of this is new - people were talking about platform diversity, multiple attack vectors and rate-limiting back in the days of the original viruses for MS-DOS.

Even then, after all the best designs and code audits, there's still the age-old "social engineering" trick that never fails to bag a boatload of suckers and render any "secure" system completely vulnerable. :-(

Date: Tue, 11 May 2004 09:50:51 +0930
From: Greg 'groggy' Lehey <Greg.Lehey@auug.org.au>

On Monday, 10 May 2004 at 14:44:38 +1000, David Purdue wrote:
> This release fails to mention that the Morris
> Worm propagated by exploiting weaknesses in
> Sendmail, an open source program.

Well, I don't know if "fails" is the correct word. But it could have made capital of the matter, something along the lines of:

- All software is vulnerable, even UNIX.
- It happened to UNIX first.
- We fixed it. It doesn't happen any more.

> So it could also be said that what the Morris
> Worm did is clearly demonstrate that software
> being open source does not imply that it
> is immune to virus/worm attack.

Well, this was UNIX, not "Open Source" :-)

> If the real lesson is that I should source my
> applications from multiple code bases, what
> is the alternate codebase for something
> that does the same job as Apache?

I don't personally think this is the lesson that people should learn.

Spam on AUUG lists

Date: Fri, 14 May 2004 13:14:25 +1000
From: Greg Black <gjb@auug.org.au>
To: talk@auug.org.au
Subject: Closing AUUG lists to non-subscribers

I belong to several AUUG mailing lists. Of those, the only one that delivers spam to me is the qauug list. I have written more then once to qauug-owner@auug.org.au to request that the list be closed to postings from non-subscribers, but have never had so much as an acknowledgement.

I don't want to unsubscribe, because the few legitimate posts are of interest to me -- but I'm fed up with the regular spam. Surely there is something that can be done about this?

From: "Daniel O'Connor" <darius@dons.net.au>
Date: Fri, 14 May 2004 13:53:19 +0930

You could add Bayes spam filtering to your end and filter mails that way..

From: Greg Black <gjb@auug.org.au>
Date: Fri, 14 May 2004 14:33:11 +1000

Gee, thanks. I don't want my user group to be a vehicle for spam (just as I don't want people to be so ignorant that they ignore a clear Reply-To header on email that I send out).

Just FYI: (1) my spam filters *do* catch most of this crap; but why should AUUG waste its resources sending this out and why should AUUG bother its members with this junk? (2) my duplicate suppression utility does block extra copies of messages from hitting my inbox, but why should I have to process them when I've made it quite clear that I don't want them?

From: "Daniel O'Connor" <darius@dons.net.au>
Date: Fri, 14 May 2004 14:47:01 +0930

> Just FYI: (1) my spam filters *do* catch most
> of this crap; but why should AUUG waste its
> resources sending this out and why should
> AUUG bother its members with this junk? (2)
> my duplicate suppression utility does block
> extra copies of messages from hitting my
> inbox, but why should I have to process them
> when I've made it quite clear that I don't
> want them?

Restricting on sender is very blunt and makes it difficult for people to discuss things freely (especially if cross posting)

A better way wouuld be to add better spam filtering to the AUUG server itself, but that will still let some spam through no matter what you try.

Sender authentication like TMDA would be pretty good though I think.

From: David J N Begley <d.begley@uws.edu.au>
Date: Fri, 14 May 2004 16:37:03 +1000 (EST)

Earlier today, Daniel O'Connor wrote:

> On Fri, 14 May 2004 14:53, Ben Elliston wrote:
>> That's not true. If the list were closed to
>> subscribers, you would never receive the
>> message. Mailman would queue it for
>> moderator approval.
>
> I was speaking in general terms rather than
> this specific case.
>
> My point is that the cure may be worse than
> the disease, especially considering there are
> several mitigating techniques that can be
> applied.

Of course, not every approach or mitigation technique satisfies everyone's concerns - hence the "debate". For example, some people like to play games with crackers by setting up honeypots or other means of wasting crackers' time whilst some people just prefer their systems to utilise "stealth" techniques and appear like a black hole on the network.

Similarly, some people would prefer that spam messages are bounced with an error rather than appear to be happily accepted or worse, trigger some sort of response (the latter two cases leading spammers to think the address is legit and therefore ripe for further spam messages).

For my 5c worth (inflation, y'see), closing the list to non-subscribers is not a major problem since the purpose of the list is to benefit those who are subscribers of the list (and by definition, they will be able to post to the list anyway). The only question should be what to do with messages sent from non-subscribers (bounce with error, silently direct to /dev/null or "flood" list owner's mailbox).

People who are not subscribers casually sending stuff to a closed list have no reason to complain about their messages not getting through - the simple solution is to join the list (and thus, join the discussion). If a non-subscriber is replying to a message with a closed list as one of the recipients, then *surely* we can apply some automatic intelligence to the handling of that reply that satisfies all concerned - after all, it's now 2004 and this is *not* a new problem.

If we as a group of intelligent IT folk cannot solve something as seemingly simple as this for ourselves, what chance do we have of convincing the general public (businesses, governments, whoever) that anything else we advocate is worth believing??

AUUG Corporate Members

As of 1 June 2004

- Apple Computer Australia Pty Ltd
- Australian Bureau of Statistics
- Australian Taxation Office
- BAE Systems
- Cape Grim B.A.P.S
- Corinthian Industries (Holdings) Pty Ltd
- Cray Australia
- CSIRO Manufacturing Science and Technology
- Curtin University of Technology
- Cybersource
- Deakin University
- Department of Land & Water Conservation
- Department of Lands
- Everything Linux & Linux Help
- EWA-Australia Pty Ltd
- IBM
- IBM Linux Technology Centre
- IP Australia
- KAZ Technology Services
- LPINSW
- Macquarie University
- Multibase WebAustralis Pty Limited
- NSW Department of Commerce
- Peter Harding & Associates Pty. Ltd.
- Powerhouse Museum
- Squiz Pty Ltd
- Sydney Water Corporation
- Tellurian Pty. Ltd.
- The University of Western Australia
- Thiess Pty Ltd
- TMD Computing
- University of NSW department of Computer Science & Engineering
- UNiTAB Limited
- University of New England
- University of New South Wales
- University of Sydney
- University of Technology, Sydney
- Workcover Queensland

Book Review: Understanding the Linux Kernel (2nd Edition)

Frank Crawford <frank@crawford.emu.id.au>

Author: Daniel P Bovet & Marco Cesati
Publisher: O'Reilly & Associates
ISBN: 0-596-00213-0
Published: 2003
Pages: 710pp

The documentation of the Unix kernel has been undertaken nearly as long as the writing of the kernel itself. John Lions spent time at Bell Labs in 1976 documenting the source code, in the "Unix Operating System Source Code Level Six" and its companion "A Commentary On the Unix Operating System". Since then every significant kernel family has been similarly documented with Bach's "The Design of the Unix Operating System" for SVR2, Goodheart and Cox's "The Magic Garden Explained: The Internals of the Unix System V Release 4" and McKusic, Karels and Bostic with "The Design and Implementation of the 4.4 BSD Operating System".

Certainly the most active kernel development at present is Linux, and this book is an attempt to document it in a similar fashion. The history of documentation has had a number of different driving forces, usually academical related, but, for example John Lions' commentary was designed to be used as notes for his Operating Systems course, where as a number of the others were more of an exposition of the kernel.

This book is more of a throw-back to John's intentions, as it has primarily come about as an aid to a course taught by the authors at the University of Rome, School of Engineering "Tor Vergata". Originally it started out as a course on Linux 2.0, the first edition of the book was based on Linux 2.2 and this edition is based on Linux 2.4. As you can see by this, it is an attempt to snapshot a fast moving target. To make matters worse, the kernel it is based on is Linux 2.4.18, which was released in February 2002, and since then Linux 2.6 has been released. There are a few references to new features in the then development kernel Linux 2.5, much of which is applicable to the current 2.6 kernel. However, while this is an issue for those who are trying to keep up with the bleeding edge, it is not a major problem for those studying operating system design. In fact as a starting point for learning about the kernel, prior to diving into the latest version, it works well.

The structure of the book itself attempts to be a logical progression from the low level functions through to higher and more specific facilities. In addition, both to limit the scope of the description and as it is the most widely available, the book concentrates on the Intel i386 family for all descriptions.

While the first few chapters give a general overview of what features an operating system provides and some of the history of Linux, it quickly moves into a section on how interrupts and exceptions are handled. Continuing on from there, it describes low-level kernel synchronisation and timing features, taking into account many of the issues with multiprocessor systems and modern processor requirements. From there is moves on the low-level memory management and address space management. While there are few chapters in the sections described, it does occupy nearly half of the book.

Once the description of the low-level functions has been covered, it moves onto higher features such as signal handling, process scheduling, disk and other caches and ultimately file-systems. The one area that the book does not really cover is device drivers, and this is an intentional omission, as it is covered by other books in the O'Reilly range.

In keeping with the intention of being a teaching aid, the book goes into a great deal of depth in its coverage of any section. Each chapter starts with a theoretical overview of the topic, followed by a detailed bottom-up approach. It starts with a description of the data structures needed to support the functionality described, moving on to the lower level functions, then to the higher levels and finally ending up showing how system calls issued by user applications are supported. Of course, not all chapters cover all items, since it depends on the area of the kernel to be covered.

If there is one problem with the book, it sometimes goes into too much detail and too many lists. In some cases there are whole pages just listing data structures, exactly as could be found in any of the '.h' files, with just some simple explanation of what each field does. While this is important in a University course, it does make it very difficult to follow when you are attempting to get an overview of a kernel function. As a result, it takes a long time to get through this book: it took me nearly four months, much of it studying detail and not higher level functions.

Ultimately, I would describe the book as "dense", packed with information but very difficult to get through. It takes a lot of effort to work through, but it does give a good understanding of how the kernel performs its functions, and in the end how Linux works. It is useful for someone who is trying to get a start on the kernel, and those who need to work perform low-level activities, but it isn't aimed at the same audience as such books as "The Magic Garden Explained". In fact it has much more in common with John Lions' original commentary.

Book Review: Automating UNIX and Linux Administration

Grant Allen <Grant.Allen@towersoft.com.au>

Author: Kirk Bauer
Publisher: Apress
ISBN: 1590592123

Before I opened Kirk Bauer's *Automating UNIX and Linux Administration*, I thought I had a rough idea what I was about to read—a guide to automating some of the normal day-to- day administration tasks using faithful tools like cron and shell scripting, spiced with an introduction to a few new tools (well, new to me at least) to broaden the mind. What I hadn't expected was quite how far Bauer would delve into the realm of administration, automating things I would never have thought to automate.

The book kicks off with a thorough introduction to SSH, the secure shell, and its various configuration options. A whole chapter is dedicated to constructing a consistent SSH environment on all machines. This would be a worthwhile exercise in its own right for many IT departments, and for SSH novices, this chapter alone would justify a copy on the bookshelf. After discussing the components of SSH, we get many configuration examples, and finally a technique for using common authorization_keys files on all servers to all shared access to privileged accounts like root, without needing to publicise the root password.

Chapters 3 and 4 deal with harmonising login and shell scripts on different systems, and introduces the first of many building block scripts—in this case the ubiquitous distribution scripts that will be used throughout the book. In this case, they are used to display the ease with which common networking configuration (host files, DNS settings, subnet settings, etc) can be automated and managed centrally for heterogeneous systems.

What really caught me off guard was Bauer's decision to show the art of automation from the very beginning—have the fresh hardware install its own operating system, configure its own DNS, firewall, hosts and other network settings, and distribute whatever packages you've decided are standard for a given machine. There are excellent ideas about handling different "classes" of machines (workstations, dev boxes, serious production machines), and automating the handling of these differences. At the end of these chapters, one is left in the wonderful position of giving the machine a name, deciding its purpose, and then relaxing while the treasure trove of automation goodies does the rest. For those of you who manage large server farms, or regularly rebuild test and development environments, the ideas laid out in these chapters make compelling reading. It may be of less value to those who have moved on to virtual machine technology, but not all of us have this luxury. Chapter 4 was the entrée for this process, but chapters 5 and 6 really deal with the meat of automated system building.

We are introduced to the first of many open source tools—cfengine—used throughout the book, and following chapters deal with sharing data, common mounts, NIS, NIS+ and LDAP, and other necessary plumbing. Rsync, CVS, and package management based on RPM, Debian's apt-get and Solaris pkgadd are run through their paces, and this is perhaps the one area of the book I thought was lacking. Bauer understandably has a preference for open source tools, as he himself has written several (excellent) ones. Unfortunately, admins are often hindered in their ability to deploy gnu-style replacements for apps and utilities, by political, application dependency and even service level constraints. The concepts covered would still be perfectly applicable to a highly controlled AIX, HP-UX or Solaris environment, for instance, but the examples and scripts could need some modification.

All of the tools and scripts covered in the earlier sections are used to show how the usual system administration tasks can benefit from automation and centralised management—system modification, user creation and management, resource monitoring, security patching and lock-down, backups and recovery. There was enough on each of these topics to whet the appetite, and sent me googling for more. The book wraps up with a set of appendices aimed at the novice administrator covering bash shell scripting, grep, awk, and some more interesting coverage of building custom RPM packages.

In all, *Automating UNIX and Linux Administration* was a good introduction to what is possible in the world of automating the drudgery of system administration, and I'd recommend it especially to people just starting in the system administration world.

CDs in this issue

Greg Lehey <Greg.Lehey@auug.org.au>

It hasn't been a good year for AUUGN CDs:

- In September last year, we produced extra CDs for the conference, and due to a comedy of errors they arrived so late that we couldn't put them into the conference bags. Instead, we gave them to all comers at the reception desk—rather too freely, as it turned out, with the result that we didn't have enough for AUUGN.
- Never mind, we thought. We'll give people two CD-Rs in the December edition. And so we did. Unfortunately, they were all unreadable.
- So, we thought, let's give people three CD-Rs in the March edition. Due to problems too embarrassing to describe, that fell through completely, and we ended up with no CD-Rs.
- Instead, we promised *four* CD-Rs in the June edition. And here they are! Well, there are five. Read on for more details.

  We have the ever-popular Knoppix and OpenOffice, the latest edition of FreeBSD (so hot off the press that we wouldn't have been able to include it if we hadn't had delays in producing AUUGN), and Fedora, the new name for "Red Hat".
- As if that wasn't enough, we discovered that we had made a bad misassumption with Fedora. All previous operating CDs that we have distributed can install the base system from the first CD of the set, and this is all we have distributed. Unfortunately, that doesn't work for Fedora: the first disk doesn't even contain the X window system. As a result, we had to include the second disk as well. There are two further disks in the set, available at *http://fedora.redhat.com/download/*, but at least you can do a base install with the first two disks.

As a result, you should have five CDs in this AUUGN. To make up for this, and also because this issue is so late (something we don't intend to repeat), the next issue will not contain a CD-R; the next issue with a CD-R should arrive shortly before Christmas.

Two of last year's CD-Rs are missing in this collection:

- The *Microsoft Survival Kit*, a collection of UNIX-like utilities that run on Microsoft platforms. The name was intended as a joke, but somehow it stuck.
- *Linux from Scratch*, a roll-it-yourself Linux distribution.

Why did we leave them out? We thought that the current collection would interest more members. We still have copies of both of these CD-Rs; if you'd like one, please contact Liz Carroll—see page 2 for details.

Next time?

As always, we try to supply CD-Rs that interest the majority of the membership. Is there a CD that you'd like to see in December? If so, please contact Liz Carroll <busmgr@auug.org.au> and tell her about it.

Knoppix 3.4

Knoppix (the "K" is pronounced) is a standalone Linux distribution designed to run without a hard disk. It was written by Dipl. Ing. Klaus Knopper, and it makes it very obvious that it comes from Germany. It's useful as a rescue disk or as a demonstration, since it will run on just about any computer. I tried the previous CD on my Dell Inspiron 7500 with a 1400x1050 display, which had given me problems with other systems in the past. It also had a Lucent wireless card. Knoppix recognized and configured both the display and the wireless card correctly, which I found quite impressive. On one of my development test machines it came up correctly, but of course it didn't recognize that the monitor was an ancient ICL VGA incapable of more than 640x480. Fortunately I was able to switch back to that resolution and use it anyway. At this point, though, I should point to the following disclaimer, which I have decapitalized to save pain:

> ***Disclaimer: This is experimental software. Use at your own risk. Knopper.net can not be held liable under any circumstances for damage to hardware or software, lost data, or other direct or indirect damage resulting from the use of this software.***

The point here is that Knoppix can't know anything about older monitors. Get things wrong and it *will* burn out the monitor. With a bit of finger trouble you might also find a way to overwrite the hard disk on the machine on which you're running. With a bit of care, though, you should find it a useful tool.

You can mount the CD on another system to look at things, of course. There's some documentation in the directory KNOPPIX, in particular the file *KNOPPIX/KNOPPIX-FAQ-EN.txt*. There's more documentation in the directory Talks, but this most of it is in German.

To start KNOPPIX, just boot from the CD-ROM. It comes up with a functional KDE 3 desktop and doesn't use any local disk. A rather strange quirk used to be that it didn't allow login at all, so you couldn't get a root shell directly. You used to have to use *sudo* without a password instead. This version is a little more sane: *su* now works and doesn't require a password.

FreeBSD 4.10

This issue includes the first CD of the FreeBSD 4.10 set. It contains the complete system, including sources, and it boots and installs in 32 bit Intel platforms. For other platforms, see *http://www.FreeBSD.org/* and contact Liz Carroll if you'd like a boot disk for one of them. There are complete installation instructions in the file *INSTALL.TXT* on the CD. For the experienced, though, the procedure is simple. The following text is reproduced with permission from my book "The Complete FreeBSD":

- If you have another operating system on the machine, for example Microsoft, and you want to keep it,
    1. Make a backup! There's every possibility of erasing your data, and there's absolutely no reason why you should take the risk.
    2. Repartition your disk with FIPS, which is available on the CD at *tools/fips.exe*.
- Insert the CD in the drive before booting.
- Boot the FreeBSD system. The easiest way is to boot directly from the CD.
- Select the Custom installation: it's the only one which allows you to back up a step if you make a mistake.
- If you have repartitioned with FIPS, in the partition editor, delete only the second primary Microsoft slice. The first primary Microsoft partition contains your Microsoft data, and if there is an extended Microsoft partition, it will also contain your Microsoft data. Then create a FreeBSD slice in the space that has been freed.
- Otherwise delete whatever you may find in the partition editor and create new FreeBSD slices.
- On exiting from the partition editor, select the `BootMgr` MBR.
- In the disk label editor, select the FreeBSD slice. If you proceeded as above, it should be empty, but if it contains existing UNIX partitions, delete them. If you're not too worried about the exact size of the partitions, select automatically generated disk labels.
- Alternatively, if you want to specify your file systems yourself, start on the basis of a root file system with 50 MB, a swap partition with 256 MB, and allocate the rest of the space on the disk to the */usr* file system. Note particularly that, if you don't create a */var* file system, you'll need to create a symlink later on.
- Choose the distributions you want. Note that in this menu, you choose the distribution by pressing the space bar, not the **Enter** key.

- Select CD-ROM as the installation medium.
- If you intend to run the X window system, select the installation now. It's much easier than doing it after the system is up and running.
- Confirm installation. The system will be installed.

OpenOffice 1.1

OpenOffice is an open source Microsoft-like office suite derived from StarOffice. As you can see from the label, it includes binaries for FreeBSD, Linux, MacOS X and Solaris. The previous CD-R that we distributed in March 2003 also included binaries for Microsoft. As I said at the time, "we don't intend to continue distributing binaries for Microsoft—after all, we are a UNIX organization—but this time we made an exception so that we could distribute the same CD-Rs at the NOIE seminar described elsewhere in this AUUGN."

We didn't cut out the Microsoft binaries deliberately, though: OpenOffice has evolved considerably since last year, and the current version is too big to include them all on one CD-R. I believe that the FreeBSD version is more important to our members than the Microsoft version.

Installing OpenOffice is relatively simple. There are brief instructions on the CD itself, and more detailed instructions in the top-level directory. Bring much space: it emulates Microsoft in its appetite for disk and memory as well.

Fedora Core 2

Fedora is the new name for what used to be called Red Hat. To quote from the file *RELEASE-NOTES-en*:

> The goal of the Fedora Project is to work with the Linux community to build a complete, general-purpose operating system exclusively from open source software. Development will be done in a public forum. The project will produce time-based releases of Fedora Core about 2-3 times a year, with a public release schedule. The Red Hat engineering team will continue to participate in building Fedora Core and will invite and encourage more outside participation than was possible in the past. By using this more open process, we hope to provide an operating system more in line with the ideals of free software and more appealing to the open source community.

The CD-Rs contain all you need for a basic installation; read the file *README-en* on the first disk for an overview. In principle, you should be able to boot from this CD-R and follow the instructions. It's interesting to note:

> In this release, the XFree86™ X11 implementation has been replaced with the X.org Foundation's new official X11R6.7.0 X Window System release. This release is a merger of the previous official X11R6 release, XFree86 4.4.0rc2, and additionally includes a number of updates to Xrender, Xft, Xcursor, fontconfig libraries, and other significant improvements. Refer to the X.org X11R6.7.0 release notes for more information.

See also Frank Crawford's "My home network" article in this issue, which discusses Fedora in some depth. See page 16 for more details.

Installing

On previous occasions, we only supplied disk 1 of multi-disk distributions. The problem is that, unlike FreeBSD or Debian GNU/Linux, Fedora is not really intended for this purpose, and most installations want to access disks 2 or 3. I haven't seen one that asks for disk 4 yet. Disk 2 is pretty much essential, so we have included it as well. It's possible to install a reasonably complete system with these two disks. If you ask for something that requires disks 3 or 4, though, you're in trouble: there's no way of telling the installer that you don't have the disk, and you have to abort the installation. There also seems to be no way of telling in advance which disks the installer will ask for. The best option seems to be to install a minimal system and then add packages after rebooting.

Apart from the requirement of two CD-Rs, Fedora had other surprises in store for us: it is quite possible to install Fedora from a CD-R and then, after booting, to find that the system can't read the CD-R ***at all***: it reports had errors on every sector. We're at a loss to explain this. It seems to be related to the program that burnt the CD-R, but we haven't established exactly what it is. The same CD-R causes no problems to Red Hat 9 or FreeBSD, so possibly it's related to the Linux 2.6 kernel distributed with Fedora. We're relatively confident in the quality of the CD-R that we finally (after nearly two weeks of messing around with burners, operating systems and the replication house) have supplied. Please let us know if you have any problems.

Sixth Australian Open Source Symposium

Preliminary call for participation

The sixth Australian Open Source Symposium (AOSS 6) will be held in Perth during the week starting 6 September 2004.

This is a preliminary call for participation, pending finalisation of the date and venue.

AOSS is an annual one day symposium, run by Open Source developers, for Open Source developers. Of course, non-developers are welcome and encouraged to attend.

Previous AOSS events were held in Melbourne (1999), Adelaide (2000), Canberra (2001), Sydney (2002) and Brisbane (2003).

The goals of AOSS6 are

- to promote the sharing of information and experience relating to OSS
- to give the OSS community a place to interact,
- to nurture and harness synergies between OSS projects, and
- to raise the awareness and credibility of OSS within the broader ICT community.

AOSS generally involves short (30 minute) presentations by developers of

- existing OSS projects,
- works in progress and
- cool technical stuff

AOSS will involve sessions with specific themes, including

- open source platforms:
  - operating systems (Linux, BSD, etc)
  - web infrastructure (Apache, Squid, etc)
  - desktops (Gnome, KDE, etc)
- open source applications
  - web applications (browsers, content management, etc)
  - office systems (e.g. OpenOffice.org)
  - collaborative applications
- open source languages, such as Lua, Perl, PHP, Python, Ruby, Tcl

This year, we are encouraging a non-technical session as well, covering issues such as:

- legal considerations
- strategies for effective OSS uptake

Timetable

Abstracts (around 100 words) are due Monday 2nd July, 2004.

Presenters will receive free registration

Please e-mail submissions to `<aoss@auug.org.au>`

AOSS 6 is presented by AUUG Inc, with the support of SLPWA—The Society of Linux Professionals, WA.

A Hacker's Diary

Greg Lehey <greg.lehey@auug.org.au>

This is the first of an experimental column which may be of interest. It's the more technically interesting parts of my diary, which you can find at *http://www.lemis.com/grog/diary.html*. I welcome feedback about whether it's interesting or not.

A number of URLs are given as relative to the directory in which the diary is kept. I haven't changed them, because it makes a mess of the format to have long, unbreakable URLs in two-column formats. They're all relative to the URL *http://www.lemis.com/grog/*.

Friday, 2 January 2004

More work on my paper for the "Open Source in Government" conference today. I discovered that there were a number of topics that I had forgotten about, and it's 35 pages already. I'll have to drop some of the other stuff—maybe.

gnuplot drives me crazy. Spent a surprising amount of time trying to convert the graphs into a reasonable format, made all the more frustrating since I solved the problem years ago and then lost the *gnuplot* scripts. It's not clear how much they would have helped: since then, *gnuplot* has been upgraded, and presumably for political reasons *gif* support has been removed. Still, got as far as some reasonable graphs. I don't see myself running out of time.

Saturday, 3 January 2004

Up early for once this morning, which was just as well, as we were planning the fourth annual hackers barbecue (*xmas-bbq-2003.html*), and for once people arrived on the dot of midday. Since Christmas 2000, I've held an annual barbecue for the local open source hackers.

We've always had great fun, though it's surprising how much the atmosphere differs from one year to the next, particularly when so many of the same people came. Originally it was timed to correspond with the presence of exiled SA hackers such as Kris Kennaway and Benno Rice, but this year neither of them were here. Daniel O'Connor has been here every year, and Chris Yeoh and Bernd Wulf were both here for the third year running. A new guest was David Newall.

It was warm again, and though we spent some time outside (no photos), we went inside for lunch, after which the laptops appeared. That's changed a lot, too: three years ago (*bbq.html*) we multiplexed four IRC sessions on a single laptop. This time we had more laptops than people, and we did interesting things with them, though not (as last year) networked games (*xmas-bbq-2002.html*), and also no IRC.

Instead, we rebuilt computers, talked about the upcoming Linux and Open Source in Government conference (*http://www.auug.org.au/events/2004/ocgconf/*), and did lots of other things. Normally people start leaving at about 4:30 pm, but today everybody had so much fun that the first people didn't leave until nearly 10 pm. Tiring, but fun.

Monday, 12 January 2004

Today was the first day of the Linux and Open Source in Government conference, so up early to get there by 8 am, which worked surprisingly well. There we discovered that very little had been arranged. It was a good thing we had our card table, since there was nothing else there. Had a moment of panic with the AV system, which turned out to have been incorrectly wired at a VGA cable level, but managed to get under way without too much trouble. Started with Ian Gilfillan, the first time we've had an MP at one of our conferences.

The quality of the presentations was pleasingly good, and about the only issues we had were with the room (Union Hall at the University of Adelaide): the lighting was dubious, the projection equipment turned off in the middle of a presentation, and for a panel session with 5 people we only had one microphone.

In the evening out to a speakers dinner at The Historian Hotel, where we took a number of photos (*http://www.lemis.com/Photos-20040113.html*). Late to bed.

Tuesday, 13 January 2004

Into town 45 minutes later today, and it made a big difference: what had been a breeze yesterday, from Fullarton Road to the University, took nearly 15 minutes, and we were running a risk of being late as a result, a particular problem since The Hon. Andrew Southcott MP had requested to be met. My daughter Yana got out on the corner of Victoria Drive and went to meet him, but somehow we failed anyway.

Off to a slightly late start, but obviously not too badly—Kate Mackenzie of *The Australian* complimented us on being better organized than most commercial conferences—and heard the politicians speak. First was Kate Lundy, shadow IT minister, who spent a lot of time saying that the Australian government should take a more active role in the computer industry, followed by Andrew Southcott, who gave a short but pithy description of how the Government was taking a more active role in the industry, and then Tony Judge of NOIE, who went into more detail.

After that, a panel session with Kate Lundy, Brenda Aynesley of the ACS, Ian Gilfillan and Dan Shearer. A very successful panel; I think the idea of submitting questions for preparation in advance is very good. Then a number of legal papers. It's interesting how many lawyers are getting involved in open source software. I don't know if this is a positive or negative development, but I suppose it's an inevitable result of free software becoming big business.

Finally the event was over, and I was able to take off tie and shoes. It looks like it's been a really great success.

In the evening had the speaker's reception, this time a dinner. Took a lot of photos (*Photos-20040114.html*), and in general had a good time.

Wednesday, 14 January 2003

Still hadn't finished my Vinum slides *http://www.vinumvm.org/papers/LCA2004/slides.pdf,* so decided not to go to today's tutorial sessions and attended to that instead. There's a ***lot*** of material there: the paper itself is about 40 pages long, and if I had converted the entire paper to slides, I would have had about 90 of them, far too many for what proves to be a 45 minute slot after all (strangely, the slots are 45 minutes with a 15 minute break after every paper), so ended up dropping some of the details of the low-level implementation from the slides. Still ended up with 70 slides, which will provide for a brisk pace.

In the evening to the professional delegates session with Yvonne, my wife, and spoke with many old and new faces. It was in the rotunda of the Adelaide Zoo, quite a pleasant location. Worth keeping for further reference.

Thursday, 15 January 2004

In to town relatively late this morning, even later than intended, and missed the beginning of Jeremy Malcolm's talk about SCO (*sco.html*). After that listened to Chris Yeoh talking about the Linux Standards Base. It sounds very well thought out. It's a pity that the BSDs don't want to know about it.

We had far too little time for lunch, only an hour, including the time it took us to get to Rundle St after people had finally got their act together, so we were quite late getting back, and I missed about half of Peter Chubb's talk about user-mode drivers. Pity, but I suppose I'll get it all on the conference CD. Then Sean Burford on reverse engineering from binaries. A good introduction, but no techniques that I didn't already know. He did point to some useful tools I hadn't heard of, though, and that should be useful.

After that I had invited a number of people out to my place, but timing was tight, and in the end only Peter and Lucy Chubb came out, so we spent most of the time talking about bassoon fingering systems. Pleasant evening.

Friday, 16 January 2004

First up this morning was my Vinum presentation, which went off without much trouble. Was rather flattered that Ian Gilfillan came in to listen to the start, certainly not because of interest in Vinum.

After that to Paul McKenney's (*http://www.rdrop.com/users/paulmck/*) talk about Read-Copy-Update. I had heard the term before, and I thought I had basically understood the concept, but it turns out that I was completely mistaken. The idea sounds a lot better now, and the figures that Paul showed spoke for themselves.

To lunch at Cafe Michael 2, which turned out to be yet another Thai restaurant, not as good as the place across the road yesterday, and with terrible service. As a result missed the first session after lunch, and then to hear Jonathan Corbet talk about the Linux kernel 2.6, which provided a good overview.

Then back home before returning for the conference dinner, at which we took a number of photos (*dinner-20040116.html*).

Saturday, 17 January 2004

Into town late today with an intention to listen to Jeremy Allison's "security soup" talk, but Arjen Lenz waylaid me and got me to attend James Cameron's talk about PIC microcontrollers, which I didn't regret. Amusingly, it ties in with a device I had been thinking of for controlling my temperature-controlled fridge: (*brewing/temperature-control.html; since this diary entry, I implemented something similar*) Paul Sorenson had pointed me to a computer controlled device, and it eventuated that James had designed it. He also spent some time talking about it and demonstrated it, showing that it's probably exactly what I'm looking for.

At lunchtime they had arranged a dunking session, and I was able to keep myself from being involved. To the Red Rock noodle bar, where I had "kimchi noodles" apparently without any kimchi. Back to hear Tridge talking about his junk code directory. It seems that he's finally tidied up his remote mail scripts, which I must investigate.

After that, conference close, and then tried to organize the people who wanted to come to my place. The challenges were many: Martin Pool needed to hire a car, Bdale had to go to the hospital to visit Pia Smith, who had come down with what proved to be flu and not SARS yesterday (she had just got back from China on Tuesday), and others wanted to get back to the hotel. Finally found the last person (Jeremy Allison) talking to Linus, and it proved that Linus didn't have anything else to do, so he came along as well. They went ahead with Rasmus while I took Martin to the airport to pick up his car.

Back home, Tridge was going on about some anomalies between the behaviour of multiple threads and multiple processes on multiple processors on a loop through `readdir()`, while Linus observed that nobody else cared. Well, not quite. I wondered how this would work on FreeBSD, but it turned out that the processes blocked on `Giant`, so that was a non-starter. Score one for Linux.

Apart from that, spent a lot of time talking about all sorts of things. Took still more photos (*barbecue-20040117.html*), and read out the fairings (*fairings.html*) saga, which the Linux people didn't know, but they found it good enough for a spontaneous round of applause. Pleasant evening.

Tuesday, 27 January 2004

We're back into conference mode now: this afternoon we had planned a conference call for AUUG 2004, but we couldn't get through: no such conference ID. Turned out that it had been scheduled for 4:30 am, not 4:30 pm. Another good reason to use a 24 hour clock. Got started a little late and got some things worked out, including no-frills admission for students and similar for only $80. We still need to find a way to define the term to ensure that other groups don't abuse it.

Thursday, 29 January 2004

One of the ever-increasing spams today was a message purportedly from Rasmus Lerdorf asking me to sign up on a web forum called Orkut (*http://www.orkut.com/*). Discussed this on IRC, where I found that it was genuine. It seems that Orkut is a newer Advogato (*http://www.advogato.org*). I never did much with Advogato: it seems like just too much trouble to enter data into web forms when you can use real tools. Advogato has pointed to this diary for some time; possibly that's the way it will end up with Orkut.

Saturday, 31 January 2004

For the last couple of days I've noticed that the most popular page on my web site has been the cats (*cats.html*) page:

```
On 30 January 2004 you had a total of 2875
  HTML hits.
Top 30 hits:
228 /grog/cats.html
136 /grog/diary.html
68 /grog/
```

Today I got an explanation: the University of Georgia used one of the photos in a Student's Clerkship Paper (*http://www.vet.uga.edu/vpp/clerk/baranik/index.htm*), and they linked to that page. That also explains the uneven distribution of the image his:

```
Top 30 images:
250 /grog/Images/Cats/Cats-3.big.jpeg
200 /grog/Images/Cats/choc-on-sofa.jpeg
181 /grog/Images/Cats/Lilac-on-monitor.jpeg
180 /grog/Images/Cats/maddi.jpeg
```

Today they got around to asking if they could use the photo. I wonder if these hits are all from people involved in the paper.

Thursday, 12 February 2004

Got the notification today that AUUG (*http://www.auug.org.au/*) is now a member of the IT Council for South Australia (*http://www.itcouncil.asn.au*), with myself as the representative. That will be interesting: they're still very much oriented towards Microsoft, and I spent 20 minutes trying to use OpenOffice to convert the acceptance mail to PostScript.

Sunday, 15 February 2004

I'm doing so many things for other people that I can't get anything done for myself: FreeBSD core team, AUUG presidency, writing articles for magazines, answering technical questions. Enough is enough! Finished off the article for Daemon News (*http://ezine.daemonnews.org*), which was really due at the beginning of the month, but since they only just published the January edition, it wasn't a problem, and sent off a message to the FreeBSD developers announcing my resignation from the core team. AUUG will have to wait a little longer, but I don't intend to nominate myself for any board position next year. Hopefully that will give me more time to do interesting things.

Monday, 16 February 2004

Gradually people have noticed that I have left the FreeBSD core team, and I've had a surprising number of supportive mail messages. It's a little silly that I can only tell the internal lists, which leaves a lot of people out of the loop, and as one person pointed out, Slashdot will have a field day with the cryptic statement on the FreeBSD web site (*http://www.freebsd.org/news/newsflash.html #event2004215:0*). To make it clear: I resigned because of overwork, not because "FreeBSD is dying".

Things were little less hectic today, though; still needed to prepare for the AUUG board meeting on Thursday, and there's still a lot of mail to get through. Things are looking better, though.

Thursday, 19 February 2004

Martin Schwenke in to pick me up at the hotel today and take me to IBM by what must have been the most adventurous way I've seen in a long time. He was trying to avoid queuing for a roundabout, but I'm not sure he saved any time.

Some AUUG board meetings are relatively peaceful and quiet. Not so today. Admittedly, we didn't throw anything at each other, but we're certainly feeling the winds of change. On the heels of my resignation from the FreeBSD core team, announced that I wouldn't be standing for the position of President this year: I'll be immediate past president instead, which will give me a year to decide whether I want to continue on the board or not. Much discussion about our direction, particularly by Andrew Cowie, who is certainly very engaged. As a result, continued our meeting to 7 pm, 3 hours later than normal. Some prepared to continue another 3, but I made it clear that I wouldn't go along with that, and by 7 pm it was clear to the others that further discussion probably wouldn't get us very far.

To the Lemon Grass Thai restaurant in the evening, then on to the Wig and Pen with David Purdue for a final drink. Somehow I'm not enlarging the subset of Canberra that I know.

Friday, 20 February 2004

Up earlier this morning for the Security Symposium. Though I wasn't "really" organizing it, I had promised Liz that I would help her, and so turned up by 8:30 am.

In the afternoon, Gordon gave me the satellite dish we had used at the AUUG 2003 Conference last September. The suppliers didn't want it back, so I thought I could use it. I was rather surprised, though, to discover it was a 90 cm dish and not the normal 60 cm. Contacted Qantas, who confirmed that the box was larger than allowed, but suggested I went along to the airport with it anyway, and maybe they could do something.

Had an interesting series of talks. This is the first security symposium I have been to, and I had always thought it was a bit off-topic for me, but it was well worth while. In general, people seemed happy with the result. In the evening out with Ben Elliston, who had arranged the seminar, and also Lawrie Brown and Luke Mewburn. Ben was happy enough with the results that he suggested he should do something else—unusual at this point. We're thinking of doing a developers workshop, maybe in Canberra again.

Saturday, 21 February 2004

Up in the middle of the night for an 8 am flight back home. My concerns that I would have difficulty checking in the 90 cm satellite dish proved unfounded: it was checked in with no problem (and the checkin agent was impressed by my hitech piece of tin).

Sunday, 22 February 2004

More work on spam blocking today. I've been seeing a surprising amount of spam sent to rog@lemis.com, obviously a typo on the part of a spammer. Set up a rule to reject anything sent to that address, which made me realize that I've found a partial solution to spam: advertise rog@lemis.com widely, get it onto spam lists, and the rejects will protect other users as well. Unfortunately, *postfix* doesn't quite provide the functionality I need: I can either reject the user, in which case mail to others goes through, or I can discard it, in which case no reject message appears to go back to the originator. I'm doing the latter at the moment, and am discarding about a message an hour:

```
Feb 23 01:37:19 wantadilla postfix/smtpd
[73196]: 8BE565120F: discard: RCPT from
adsl-64-167-111-222.dsl.snfc21.pacbell.net
[64.167.111.222]: <rog@lemis.com>: Recipient
address  See http://www.lemis.com/dontspam.html;
from=<VelmaH@optonline.net> to=<rog@lemis.com>
proto=SMTP
helo=<adsl-64-167-111-222.dsl.snfc21.pacbell.net>
```

As the spammers trawl these web pages, I hope that the number will increase. It might prove interesting to process the mail logs to reject mail from IPs which send this mail.

Tuesday, 24 February 2004

Teleconference for AUUG 2004 in the afternoon. I'm glad I don't have too much to do with the organization this year.

Wednesday, 25 February 2004

Last week I made a minor change to *concontrol* and received a message from Bruce Evans telling me of all the style errors I had made in the commit, a so-called *brucifiction.* He was right, of course, which made it all the more annoying. Today finally got round to looking at the (untested) changes he had suggested. Apart from the certain satisfaction in discovering that he had made both a mistake and a misassumption, spent hours trying to get things right, and finally gave up. I wish I agreed more with the BSD kernel coding style.

Thursday, 26 February 2004

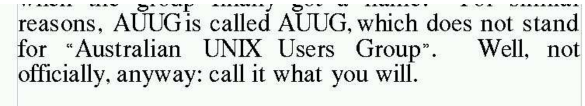
Into town today for the ADUUG lunch, and once again to the Lion Hotel in North Adelaide, where we had to wait so long for lunch a year ago (*diary-feb2003.html#21*). This time we were outside on the terrace, not quite a beer garden, and the food was a little faster, though 25 minutes for a hamburger is still pushing it.

Saturday, 28 February 2004

Came in to the office this morning to find that *echunga* had suffered some hardware failure during the night, and we were off the net. It's not clear exactly what went wrong, except that it was with the disk subsystem. Pressing the reset button enabled it to reboot, but the kernel couldn't find the disks. Power cycling did help. Looks like the controller got itself into a condition from which the driver couldn't recover.

Sunday, 29 February 2004

Spent some time finishing off my articles for this quarter's AUUGN, which is produced with OpenOffice. That always drives me crazy (see the slides "Why I hate OpenOffice" (*/papers/Why-I-hate-OpenOffice-slides.pdf*). There's also a paper at *http://www.lemis.com/papers/Why-I-hate-OpenOffice.pdf*), but today I just couldn't be bothered replacing the `"..."` sequences with "..." one by one with multiple mouse clicks. Instead unpacked the archive and used *Emacs* to change the sequences. In the process, discovered yet another annoying habit of GUI software: without telling me, at least some of the time it added markup which, though visible, was not intelligible on the screen. Specifically, they looked too small:



reasons, AUUG is called AUUG, which does not stand for "Australian UNIX Users Group". Well, not officially, anyway: call it what you will.

That's what I got when I simply followed the instructions and pushed the mouse to `Insert Special Character`. I unpacked the archive and looked at the *content.xml* file (two lines) with *Emacs*:

```
$ mkdir open-office-abortion
$ cd open-office-abortion
$ unzip ../q1-2004.sxw
$ emacs content.html &amp;
```

Emacs showed the following sequence (modified to limit the line length):

```
reasons, AUUG is called AUUG, which does not
stand for <text:span text:style-name="T4">
â\200\234</text:span>Australian UNIX Users
Group<text:span text:style-name="T4">â\200\235
</text:span>.
<text:s/>Well, not officially, anyway: call
it what you will.
```

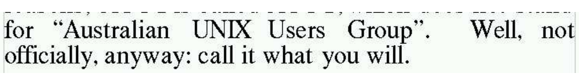
Clearly the sequences `â\200\234` and `â\200\235` are the “ and ” characters. It seems that ä is the leadin character, and the next two bytes (shown here in C octal notation as `\xxx`) are the Unicode character. But what’s that `<text:span text:style-name="T4">` doing there? Making the quotes too small, it seems. It appears to define the style of the enclosed text. I’m not sure how it got there; I certainly didn’t ask for it. Probably it relates to some global default set at the time of insertion, rather than the style of the surrounding text, which is what is used if you simply press a key. The real problem, though, was knowing what was wrong: the *WYSIWYG* paradigm doesn’t tell you things like that. With Emacs, I removed them, leaving the following text:

```
reasons, AUUG is called AUUG, which does not
stand for â\200\234Australian UNIX Users
Groupâ\200\235.  <text:s/>Well, not officially,
anyway: call it what you will.
```

Then repacked the archive:

```
$ zip ../q1-2004.sxw *
```

After that, things looked better:



for “Australian UNIX Users Group”. Well, not officially, anyway: call it what you will.

It’s a real pain dealing with single line documents! In fact, you don’t need to. It seems that OpenOffice has no problem with the wrapped format, though of course it removes the line breaks next time it saves. But what a mess! The whole idea of programs like OpenOffice is that they should be easy to use, and this is worse than pulling teeth. It’s not the way it’s intended to be used, of course, but I find that even this is easier than using OpenOffice the way it was intended.

Thursday, 4 March 2004

Into town to meet with Denis Wall of the IT Council of South Australia in preparation for the monthly meeting on Tuesday. It’s interesting that the IT Council has so far not come up with any viewpoint on the proposed Free Trade Agreement with the USA, which many of us fear will stifle innovation by further restricting use of intellectual property. In particular, reverse engineering will become more difficult, though it does seem that it will still be permitted for purposes of interoperability. The question is, does the ability to play a DVD from a different part of the world fall under the category “interoperability” or not?

Saturday, 6 March 2004

Four years ago today (*diary-mar2000.html#6*) I started working with Linuxcare.. Things have certainly changed since then! The “Open Source” landscape has changed almost beyond recognition, as has the world. What a crazy millennium!

Monday, 8 March 2004

In the evening, playing around with a couple of JPEG utilities in the FreeBSD Ports Collection, *exiftags* and *jhead*. Both dump image information from a JPEG. In the process, it’s clear that *xv* removes significant quantities of such data. Here’s an example with a pretty forgettable photo I took recently. The original, from the camera, is *dscn3050.jpg*, and the processed version is *Photos/20040305/big/airlock.ps*:

```
$ jhead *
File name    : airlock.jpeg
File size    : 245323 bytes
File date    : 2004:03:05 11:07:02
Resolution   : 1536 x 2048
Jpeg process : Baseline
Comment      : CREATOR: XV Version 3.10a Rev:
  12/29/94 (jp-extension 5.3.3 + PNG patch 1.2d)
  Quality = 75, Smoothing = 0
File name    : dscn3050.jpg
File size    : 1058921 bytes
File date    : 2004:03:05 21:15:02
Camera make  : NIKON
Camera model : E880
Date/Time    : 2004:03:05 10:45:03
Resolution   : 2048 x 1536
Flash used   : Yes
Focal length : 13.9mm
Exposure time: 0.017 s  (1/60)
Aperture     : f/9.4
ISO equiv.   : 100
Metering Mode: matrix
Exposure     : program (auto)
Jpeg process : Baseline
Not JPEG: info.txt
$ exiftags *
airlock.jpeg:
exiftags: couldn’t find Exif data
dscn3050.jpg:
Camera-Specific Properties:
Equipment Make: NIKON
Camera Model: E880
Camera Software: E880v1.0
Maximum Lens Aperture: f/3.4
Image-Specific Properties:
Image Orientation: Top, Left-Hand
Horizontal Resolution: 300 dpi
Vertical Resolution: 300 dpi
Image Created: 2004:03:05 10:45:03
Exposure Time: 1/60 sec
F-Number: f/9.4
Exposure Program: Normal Program
ISO Speed Rating: 100
Exposure Bias: 0 EV
Metering Mode: Pattern
Light Source: Unknown
Flash: Flash
Focal Length: 13.90 mm
Color Space Information: sRGB
Image Width: 2048
Image Height: 1536
```

```
Color Mode: COLOR
Image Quality: FINE
White Balance: AUTO
Image Sharpening: AUTO
Focus Mode: AF-C
Flash Setting: NORMAL
ISO Selection: AUTO
Image Adjustment: AUTO
Lens Adapter: OFF
info.txt:
exiftags: doesn't appear to be a JPEG file;
  searching for start of image
exiftags: invalid JPEG format
```

It's interesting to note that *exiftags* didn't find any information at all in the *xv* output.

Friday, 12 March 2004

There's no point getting up early in Singapore: it's a hacker's city, and it doesn't come to life until 10 am. Up round 8 to finish off my talk for tomorrow, involving a significant fight with *grops* and the use of *Emacs* to fix the final PostScript before conversion to PDF, then off to look for breakfast. (*details omitted; see the web page*)

Then the obligatory catastrophe. I think I shouldn't leave home any more. Yvonne had dropped a ball-point pen behind her desk, and while searching for it had disconnected the power to *battunga*, which earlier in the day had claimed:

```
battunga       up 547+01:07,      0 users,
          load 0.07, 0.05, 0.00
```

Not only that: it didn't want to come up. At some time in the last 1½ years, I had upgraded the system, but not rebooted. When Yvonne tried, the kernel didn't want to know the disks, which had changed from *wd* to *ad*. She managed, through much ill-tempered international calls on my part, to get */dev/ad0a* mounted, but we didn't have device nodes for it. Finally booted the old kernel, but the new *login* didn't want to let her log in—or at least that's what I think it is. It's difficult to diagnose these issues from another continent.

Finally, at the airport, tried to connect to the once wonderful Singapore Airport Wireless Network (*diary-nov2000.html#6*). It now costs money, and their credit card authentication doesn't work ("Please contact your bank to solve this problem"; what do they think I am?). While looking for a phone, found a "PC corner" with free Ethernet, so used that instead, and finally managed to start X for Yvonne. By that time, it was time to board the plane for Taipei. Not the way I wanted to spend the day: missed lunch with Harish Pillay, and there was also some exhibition on which I had planned to go to.

In Taipei, met up with Warner Losh, who had just come in from Denver, and off to the Academia Sinica (*http://www.sinica.edu.tw/*), where we arrived round midnight. While checking in, discovered that my plane tickets had been incorrectly issued, and I was booked back on a flight tomorrow at 1405. That will need some attention.

This place is freezing! The temperature in my room was 16°, with no way to warm it.

Saturday, 13 March 2004

I was just planning to go down for breakfast at 9 am when I got a call from Michael Wu saying that there was a planned speaker's session at that time. Managed to get some breakfast anyway, where I bumped into Jeffrey Hsu, who is also speaking at this conference. As the name suggests, he's originally from Taiwan, but hasn't been back here for ten years. Reminds me of myself and Australia.

After the briefing, managed to contact Singapore Airlines, and fortunately I was able to change my flight. Nobody knows if any fees are involved, but that seems to be a good sign in itself.

Did my kernel debugging talk, shortened to 3 hours. after which we had a BoF session, then Jeffrey talking about DragonFlyBSD. They have some really interesting ideas there. I should try it.

In the evening, had a banquet (*Photos-20040313.html#banquet*). Good food, a far cry from what we get at most conferences. Spent some time talking after that, but most people seemed tired. Early to bed.

Sunday, 14 March 2004

There were noticeably fewer people at the conference this morning. Went to Warner Losh's talk about embedded FreeBSD, then a couple of half talks by local people. After that, participated in a panel session on "The future of BSD", which ended up being a talk on the features to be added to FreeBSD in the next couple of years. It's a pity that nobody thinks beyond this framework. That's not a criticism of the conference, more a general issue of viewpoint.

The panel session was originally planned for two hours, but we finished in one, making it possible to listen to Itojun talk about IPv6 via video feed from Tokyo. It worked better than last time at the AUUG conference in Sydney, but it sill left something to be desired.

In the evening went shopping in the local electronics markets, and discovered that I had made a big mistake by buying DVD+Rs in Singapore: they're about a third of the price here, and I got some DVD+RWs (most expensive kind) for half the price of the DVD+Rs I bought on Friday.

Then to an interesting restaurant, apparently a tourist thing: Kenjiro Cho had brought a Japanese guide book with him, and we found it in there. Still, interesting atmosphere.

Took many photos; the raw ones are here (*Photos-20040315.html*).

Monday, 15 March 2004

Met up with most of the overseas guests today: Sam Leffler and family, Robert Watson and wife, and Warner Losh, to go to the National Palace Museum (*http://www.npm.gov.tw/*). They had decided to go by bus and MRT, which meant that we got there just before midday, by which time I had had a series of phone calls asking me to be in Canberra for a meeting on Wednesday. After some deliberation and the unfortunate discovery that yes, indeed, I could be there by then, I decided not to go anyway. There's only so much you can do.

The museum was interesting, though after all people had told me about it, it was an anti-climax: "You haven't seen the Forbidden City until you've seen the National Palace Museum too". In fact, I saw nothing of the treasures of the Forbidden City. Still, a very interesting day.

In the evening, we had planned to go to the night market (not my first choice, but everybody else wanted to go there). Then everybody else was too tired, so went there alone to meet up with Clive Lin and friends, then back to Clive's place to see how a real Taiwanese hacker lives.

More photos; the raw ones are also here (*Photos-20040315.html*).

Tuesday, 16 March 2004

Another day spent with travel, with a little spice added by the meeting in Canberra tomorrow. As a result of the latter, left the Academia Sinica later than planned, and then discovered how bad the traffic here is in week days: it took 90 minutes to get to the airport.

In Singapore found, shown yet another Internet connection (*Photos-20040316.html#changi*), this time with the options of wired, wireless or infrared. I don't do infrared, but was able to confirm that the wired connection was free, and the wireless version required payment. For obvious reasons, didn't check whether the wireless authentication worked.

Thursday, 18 March 2004

Up in the middle of the night to go to a breakfast organized by the IT Council of South Australia, taking Yana with me. I usually avoid Hahndorf on a dirt road to the south and west. Yana queried this, since at this time of the day (6:45 am, not yet light) there was no traffic, and I answered that it was more relaxing, since nothing was liable to come out of a side road. The next thing I knew, three kangaroos jumped out of the bush right in front of me, causing me to brake heavily and throw the contents of the rear seat onto the floor. Once again the kangaroo whistles proved completely useless. One point to Yana.

The meeting was interesting. AUUG has barely joined the IT council, and already it looks like the organization will change completely.

In the afternoon, working on a product requirements spec. How useless Microsoft "Word" is! Converted it to OpenOffice and then tried to handle the XML code, in which I got further than expected. There appear to be a number of tools that will format XML legibly (and more than one line per file). The port */usr/ports/textproc/libxml) includes a program called *xmllint* that will, with some persuasion, reformat and indent the text. *[see the web page at http://www.lemis.com/grog/diary-mar2004.html#18 for more details; it won't fit in this format].*

Friday, 19 March 2004

Finally caught up with the backlog from Taiwan, and did some more thinking about Funnelweb. It's an interesting concept, but at the moment it's just keeping me from doing real work.

At lunchtime, finally received the Digitrex GKX-9000 DVD recorder that I had bought on Ebay nearly two weeks ago. This appears to be the same as the Apex DRX-9000, barely warmed over for Australian conditions (for example, the date format is the wrong way round). Note that the Australian site claims it will record on DVD-R and DVD-RW media. I don't believe this.

The good news: it works, at least some of the time. Like many modern embedded designs, it looks hurried and rushed. The manual and the on-screen displays were obviously written by somebody who doesn't speak English very well:

> 5. In five minutes before it is ready to record the scheduled, it appears a Record Prompt dialog, affirm press OK, abolish press CANCEL, and if without operation for a moment it will record the scheduled automatically.

I'm curious about the background of that text; I'd suspect Eastern European.

Within a few minutes of installation, I managed to shoot myself in the foot: the player supports "progressive scan", a new way to say "non-interlaced video". I didn't know if my TV supported progressive scan, so I tried it. It didn't. Then, of course, I couldn't access the on-screen display, and the non-intuitive method of access (hint: in case of doubt, try `Select` and look for the tiny error message at the right of the screen) made it impossible to get a display again. It took a while to find the button `I/P` on the remote control, which toggles between interlaced and non-interlaced video.

Things weren't exactly plain sailing after that. I'm still trying to work out what's going on, but it seems that

By the end of the evening, I had established:

1. It's rather obtrusive. It has a loud fan, and the power LED is bright blue, about 2cm wide and 5 mm high. I've covered it over.
2. It's very slow. Functions like channel change have a noticeable delay, and some functions (stop recording, power down) seem to require accessing the DVD+RW: the latter takes about 30 seconds. The on-screen text suggests that it's writing to the DVD+RW, even when no write access seems necessary. It doesn't seem to do any harm, tough.
3. Interrupting it while it is recording is a Bad Idea. I managed to make a coaster out of a DVD+R like that. I also managed to hang a recording on a DVD+RW, and the only way to get any reaction was to power cycle it.
4. Power cycling makes it forget the date and time.
5. The user interface is very confusing. To change anything, you need to press `Select`. Do that at the wrong time and you'll erase what was there before (this applies particularly to things like recording dates and times).

Still, it wasn't very expensive, and with a bit of trouble I'll get over the frustration. I really need to get my act together to use computers to handle HiFi and video.

I've started a page (*digitrex.html*) with my experiences.

Saturday, 20 March 2004

Spent some time playing round with the DVD player , which must have some of the most broken firmware I've ever seen. Surely it must be possible to do things better than that. I must start looking for a good open source project to get involved in.

Sunday, 21 March 2004

Talking to people on IRC about DVD playing today. Came up with a number of interesting suggestions: the ports *mplayer* and *mencoder*, rip with *multimedia/dvdrip*, make DVDs from MPEGs with *multimedia/dvdauthor*. There's also an article on the subject at *http://bsdnews.org/01/vcd.php*. All to be followed up on.

The Digitrex DVD recorder itself continues to astound. Tried to explain to Yvonne how to use the thing, and discovered we couldn't enter a start time: the thing has this stupid 12 hour format, requiring you to enter first a (modulo 12 hour) time and then to select AM or PM. The previous start time had been AM, and it was now after midday. The recorder refused to accept the time, stating that it had already passed. It also refused to leave the field to allow me to select PM first. What a load of crap! This is also the first VCR-like device that I have ever seen that doesn't clear the recording slots after the recording is finished. The instructions are also horribly vague, so wrote up how to do it (*digitrex.html#programming*). Hopefully we'll get used to it.

Tuesday, 23 March 2004

In the evening, more fun with the Digitrex DVD player. We had started watching a DVD last night, and wanted to continue. I had noted the time where we had stopped, and I wanted to find a way to continue from there. As with the TiVo, it wasn't possible, but the Digitrex is worse:

- There's a button `GOTO` on the remote control. It appears only to function in the `EDIT` menu, which appears to be accessible only from the `DISC OPER` menu. At any other time, `GOTO` seems to be invalid.

- If you select DISC OPER during normal play, the recorder resets to the beginning of the recording.
- On page 16 of the manual, it suggests that you can use the NAVIGATION menu to enter a specific time. This hasn't worked for me; I can only enter a Chapter or Title.

In the end, found the data the same way I have to on the TiVo: by manual search. It's frightening how little people use the possibilities of new technology. It should be easy enough to store (cache) the position in the recorder and have it remember it next time the disk is inserted.

Saturday, 27 March 2004

Made a recording with the Digitrex recorder on a DVD+R today. The quality was impossibly bad; only the first half displayed at all. This is the second DVD+R I've tried to burn on the machine, and the other one was just as bad. By contrast, the (cheaper) DVD+RWs work fine. Either I have a dud batch of DVD+Rs, or the Digitrex doesn't like them. To check the latter hypothesis, spent some time looking at *mplayer* to see if it can understand the result. *mplayer* is somewhat frustrating, and the first installation didn't install a GUI (though I'm not sure that this is a problem).

Wednesday, 31 March 2004

More work on source code conversion today, taking up most of the day. We've also been thinking of how to check for regressions, and I had the bright idea of comparing the object files. That was a non-starter: they were even different lengths. Looking at the assembler output didn't help much either. It was full of stuff like this:

```
 .LM2:
 pushl   __stderrp
-        pushl   $80
+        pushl   $301
 pushl   $1
 .LCFI5:
```

After some investigation, discovered that the numbers in question were line numbers. Other differences were the file names in various places. Spent some time eliminating these differences, but I also managed to introduce some real differences, so it'll be a while. At any rate the conversion is complicated enough that I don't believe that a program can do it. On the positive side, I don't expect to spend more than 2 or 3 hours per module, and we can do the changeover incrementally.

CACert certificates

Ben Elliston <bje@air.net.au>

CAcert is the community non-profit Certificate Authority. If you are looking to have your web browser trust our certificates, please install our root certificate.

CAcert's goal is to promote awareness and education on computer security through the use of encryption, specifically with the X.509 family of standards. We have compiled a document base that has helpful hints and tips on setting up encryption with common software, and general information about Public Key Infrastructures (PKI).

For the enthusiast looking to dip their toe in the water, we have an easy way of obtaining certificates you can use with your email program. You can use these not only to encrypt, but to prove to your friends and family that your email really does come from you.

For administrators looking to protect the services they offer, we provide host and wild card certificates which you can issue almost immediately. Not only can you use these to protect websites, but also POP3, SMTP and IMAP connections, to list but a few. Unlike other certificate authorities, we don't limit the strength of the certificates, or the use of wild card certificates. Everyone should have the right to security and to protect their privacy, not just those looking to run ecommerce sites.

You can find out more about CAcert at *http://www.cacert.org*.

The certificate

To facilitate the secure distribution of the root certificate, we had intended to distribute a trusted copy of the CAcert root certificate on one of this quarter's CD-Rs, on read-only media via the postal system. Unfortunately (see page 46), problems with the production made this impossible this quarter. We plan to put them on the next CD-R. In the meantime, we're printing the certificate here. You can download the machine-readable version from *http://www.cacert.org/cacert.crt* and compare it with the text below:

```
-----BEGIN CERTIFICATE-----
MIIHPTCCBSWgAwIBAgIBADANBgkqhkiG9w0BAQQFADB5MRAwDgYDVQQKEwdSb290
IENBMR4wHAYDVQQLExVodHRwOi8vd3d3LmNhY2VydC5vcmcxIjAgBgNVBAMTGUNB
IENlcnQgU2lnbmluZyBBdXRob3JpdHkxITAfBgkqhkiG9w0BCQEWEnN1cHBvcnRA
Y2FjZXJ0Lm9yZzAeFw0wMzAzMzAxMjI5NDlaFw0zMzAzMjkxMjI5NDlaMHkxEDAO
BgNVBAoTB1Jvb3QgQ0ExHjAcBgNVBAsTFWh0dHA6Ly93d3cuY2FjZXJ0Lm9yZzEi
MCAGA1UEAxMZQ0EgQ2VydCBTaWduaW5nIEF1dGhvcml0eTEhMB8GCSqGSIb3DQEJ
ARYSc3VwcG9ydEBjYWNlcnQub3JnMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIIC
CgKCAgEAziLA4kZ97DYoB1CW8qAzQIxL8TtmPzHlawI229Z89vGIj053NgVBlfkJ
8BLPRoZzYLdufujAWGSuzbCtRRcMY/pnCujW0r8+55jE8Ez64AO7NV1sId6eINm6
zWYyN3L69wj1x81YyY7nDl7qPv4coRQKFWyGhFtkZip6qUtTefWIonvuLwphK42y
fk1WpRPs6tqSnqxEQR5YYGUFZvjARL3LlPdCfgv3ZWiYUQXw8wWRBB0bF4LsyFe7
w2t6iPGwcswlWyCR7BYCEo8y6RcYSNDHBS4CMEK4JZwFaz+qOqfrU0j36NK2B5jc
G8Y0f3/JHIJ6BVgrCFvzOKKrF11myZjXnhCLotLddJr3cQxyYN/Nb5gznZY0dj4k
epKwDpUeb+agRThHqtdB7Uq3EvbXG4OKDy7YCbZZ16oE/9KTfWgu3YtLq1i6L43q
laegw1SJpfvbi1EinbLDvhG+LJGGi5Z4rSDTii8aP8bQUWWHIbEZAWV/RRyH9XzQ
QUxPKZgh/TMfdQwEUfoZd9vUFBzugcMd9Zi3aQaRIt0AUMyBMawSB3s42mhb5ivU
fslfrejrckzzAeVLIL+aplfKkQABi6F1ITe1Yw1nPkZPcCBnzsXWWdsC4PDSy826
YreQQejdIOQpvGQpQsgi3Hia/0PsmBsJUUtaWsJx8cTLc6nloQsCAwEAAaOCAc4w
ggHKMB0GA1UdDgQWBBQWtTIb1Mfz4OaO873SsDrusjkY0TCBowYDVR0jBIGbMIGY
gBQWtTIb1Mfz4OaO873SsDrusjkY0aF9pHsweTEQMA4GA1UEChMHUm9vdCBDQTEe
MBwGA1UECxMVaHR0cDovL3d3dy5jYWNlcnQub3JnMSIwIAYDVQQDExlDQSBDZXJ0
IFNpZ25pbmcgQXV0aG9yaXR5MSEwHwYJKoZIhvcNAQkBFhJzdXBwb3J0QGNhY2Vy
dC5vcmeCAQAwDwYDVR0TAQH/BAUwAwEB/zAyBgNVHR8EKzApMCegJaAjhiFodHRw
czovL3d3dy5jYWNlcnQub3JnL3Jldm9rZS5jcmwwMAYJYIZIAYb4QgEEBCMWIWh0
dHBzOi8vd3d3LmNhY2VydC5vcmcvcmV2b2tlLmNybDA0BglghkgBhvhCAQgEJxYl
aHR0cDovL3d3dy5jYWNlcnQub3JnL2luZGV4LnBocD9pZD0xMDBWBglghkgBhvhC
AQ0ESRZHVG8gZ2V0IHlvdXIgb3duIGNlcnRpZmljYXRlIGZvciBGUkVFIGhlYWQg
b3ZlciB0byBodHRwOi8vd3d3LmNhY2VydC5vcmcwDQYJKoZIhvcNAQEEBQADggIB
ACjH7pyCArpcgBLKNQodgW+JapnM8mgPf6fhjViVPr3yBsOQWqy1YPaZQwGjiHCc
nWKdpIevZ1gNMDY75q1I08t0AoZxPuIrA2jxNGJARjtT6ij0rPtmlVOKTV39O9lg
18p5aTuxZZKmxoGCXJzN600BiqXfEVWqFcofN8CCmHBh22p8lqOOLlQ+TyGpkO/c
gr/c6EWtTZBzCDyUZbAEmXZ/4rzCahWqlwQ3JNgelE5tDlG+1sSPypZt90Pf6DBl
Jzt7u0NDY8RD97LsaMzhGY4i+5jhe1o+ATc7iwiwovOVThrLm82asduycPAtStvY
sONvRUgzEv/+PDIqVPfE94rwiCPCR/5kenHA0R6mY7AHfqQv0wGP3J8rtsYIqQ+T
SCX8Ev2fQtzzxD72V7DX3WnRBnc0CkvSyqD/HMaMyRa+xMwyN2hzXwj7UfdJUzYF
CpUCTPJ5GhD22Dp1nPMd8aINcGeGG7MW9S/lpOt5hvk9C8JzC6WZrG/8Z7jlLwum
GCSNe9FINSkYQKyTYOGWhlC0elnYjyELn8+CkcY7v2vcB5G5l1YjqrZslMZIBjzk
zk6q5PYvCdxTby78dOs6Y5nCpqyJvKeyRKANihDjbPIky/qbn3BHLt4Ui9SyIAmW
omTxJBzcoTWcFbLUvFUufQb1nA5V9FrWk9p2rSVzTMVD
-----END CERTIFICATE-----
```

Verifying the certificate

The CD was prepared by Ben Elliston, a CAcert assurer. This article was modified by Greg Lehey, an acting AUUGN editor and CD-R mangler. You might know Ben, but most likely not. Despite the fact that he trusts himself, you might not! You are encouraged to go to whatever lengths you feel are necessary to be confident in the authenticity of this root certificate. The authenticity of the root certificate is essential.

Once you have confirmed that the certificate is correct, you can import the certificate into your browser by following the instructions at *http://www.cacert.org/index.php?id=16*.

The following text, from the CAcert web site, provides SHA1 and MD5 checksums of the root certificate which you are encouraged to verify before importing the certificate into your browser. The text was signed with the CAcert high-trust GPG key which you may have a GPG web-of-trust path to.

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

For most software, the fingerprint is reported as:
A6:1B:37:5E:39:0D:9C:36:54:EE:BD:20:31:46:1F:6B

Under MSIE the thumbprint is reported as:
135C EC36 F49C B8E9 3B1A B270 CD80 8846 76CE 8F33
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.2.2 (GNU/Linux)

iD8DBQE/VtRZ0rsNAWXQ/VgRAphfAJ9jh6TKBDexG0NTTUHvdNuf6O9RuQCdE5kD
Mch2LMZhK4h/SBIft5ROzVU=
=R/pJ
-----END PGP SIGNATURE-----
```

AUUG Chapter Meetings and Contact Details

City	Location	Other
Adelaide	Marcellinas Pizza Bar 273 Hindley Street Adelaide	Meetings are held at 7 pm on the second Wednesday of each month.
Brisbane	Inn on the Park 507 Coronation Drive Toowong	For further information, contact the QAUUG Executive Committee via email (`qauug-exec@auug.org.au`). The technologically deprived can contact Rick Stevenson on (07) 5578-8933.
Canberra	Australian National University	For updated information, see *http://www.canb.auug.org.au/cauug/*
Hobart	University of Tasmania	Chapter appears to be dormant. The last known URL for updated information was *http://www.tas.auug.org.au/*
Melbourne	Various. For updated information see *http://www.vic.auug.org.au/*	The meetings alternate between technical presentations in the even numbered months and purely social occasions in the odd numbered months. Some attempt is made to fit other AUUG activities into the schedule with minimum disruption.
Perth	The Victoria League 276 Onslow Road Shenton Park	For updated information, see *http://www.auug.org.au/wauug/waug.html*.
Sydney	Sun Microsystems Ground Floor, 33 Berry Street (cnr Pacific Hwy) North Sydney.	The NSW Chapter of AUUG holds meetings once a quarter in North Sydney in rooms generously provided by Sun Microsystems. More information at *http://www.auug.org.au/nswauug/*.

For up-to-date details on chapters and meetings, including those in all other Australian cities, please check the AUUG website at *http://www.auug.org.au/* or call the AUUG office on 1-800-625655.