

news@UK

The Newsletter of the UK UNIX Users Group

Published electronically at <http://www.ukuug.org/newsletter/>

Volume 12, Number 4

ISSN 0965-9412

December 2003

Contents

News from the Secretariat	3
UKUUG LISA/Winter Conference 2004	3
OSA2004	4
Announcement: AsiaBSDCon 2004	4
Announcement: NordU2004	4
Announcement: USENIX Annual Technical Conference	5
Announcement: LinuxConf.Au 2004	5
Announcement: UNIX Internationalization Guide	5
OSS Watch	5
UKUUG at the Garden of Eden?	6
Software patents	9
Reinventing Perl	10
SC2003 – SuperComputing 2003 – Igniting Innovation	12
To push desktop Linux, radical shift may be required	14
The IEEE and The Open Group Launch POSIX Certification Program	17
Book review: Essential CVS	18
Book review: BLAST	20
Book review: Amazon Hacks	21
Book review: The Web Programming CD Bookshelf	22
Book review: Learning Perl Objects, References and Modules	23
Book review: LDAP System Administration	24
Book review: Building Wireless Community Networks	25

Book review: JavaScript and DHTML Cookbook	26
Book review: Mac OS X Hacks	26
Book review: Content Syndication with RSS	27
Book review: Learning Web Design 2nd Edition	28
Book review: Learning XML 2nd Edition	28
Book review: C++ in a nutshell	29
Book review: C++ Pocket Reference	29
Book review: eBay Hacks	30
Book review: Optimizing Oracle Performance	31
Book review: Oracle Data Dictionary Pocket Reference	33
Contacts	35

News from the Secretariat

Jane Morrison

Ray Miller (newly co-opted member of Council) has been working extremely hard putting together the programme for the winter 2004 event. The conference, which will be preceded by a half-day tutorial, will be held in Bournemouth on 25th and 26th February and you should find the information booklet and booking form enclosed with this mailing. There is a strong programme this year and you will note that the usual second half-day has been extended to a full day! This will be the first ever UKUUG in Bournemouth.

The venue for the LINUX 2004 conference is almost confirmed and is likely to be Leeds. Alasdair Kergon is visiting the venue on 1st December to make sure it is suitable and within the constraints of our budget. The call for papers for the event is enclosed in this mailing.

Do you need to buy some books? Don't forget that your UKUUG membership subscription allows you to receive a discount of 21.5% on O'Reilly publications - see the UKUUG web site for a list of titles available and prices.

The annual subscription invoices will be sent out in January, please look out for your invoice and as always prompt payment will be gratefully received!

Unbelievably it is that time of year again and I would like to wish you all a very happy Christmas and a peaceful New Year.

The Secretariat will be closed between 22nd December 2003 and January 5th 2004.

Please note the copy date for the next issue Newsletter (March 2004) is: 20th February 2004.

UKUUG Secretariat

PO Box 37

Buntingford

Herts

SG9 9UQ

Tel: 01763 273475

Fax: 01763 273255

office@ukuug.org

<http://www.ukuug.org>

UKUUG LISA/Winter Conference 2004

Ray Miller

Next year's LISA/Winter Conference will take place in Bournemouth on the 25th and 26th February. We are just putting the finishing touches to a very strong programme on the theme of high availability and reliability. Topics include large-scale mail services; high availability Linux; storage replication; MySQL; choosing reliable hardware.

The conference itself will be preceded by a half-day tutorial delivered by Alan Robertson of IBM Linux Technology Centre and Lars Marowsky-Bree of SUSE Labs (you might know them better through their work on the High Availability Linux project). Lars will cover the basics of setting up high availability Linux clusters before moving on to cover some advanced topics, while Alan will discuss using and extending the Heartbeat cluster monitoring software.

Alan and Lars will also deliver a presentation in the main conference, where they are joined by a number of international speakers as well as some old favourites. Tim Chown, who delivered last year's tutorial on IPv6, returns to give us an update on IPv6 implementation issues; and Richard Moore, a regular speaker at our Linux Technical conferences, will talk about preparing Linux for the enterprise.

The tutorial will take place in the three-star Queens Hotel, where we have a block booking for accommodation, with the conference itself just a five-minute walk away in Bournemouth University's Lansdowne Campus.

As well as the formal presentations, the Winter Conference offers members the opportunity to meet together with leading experts in the field. Numbers are strictly limited, so be sure to book your place soon – and take advantage of the early-bird saving.

See the insert for more information and a booking form, and keep an eye on the web site for the latest news and information. We look forward to seeing you in Bournemouth!

<http://www.ukuug.org/events/winter2004/>

OSA2004

Charles Curran

At the beginning of October, we launched OSA2004 (UKUUG's award for free and open source). We have tried to get notice through to each computing science department and computing service, but we should also appreciate help from those of you working in HE/FE to ensure that publicity gets to appropriate people. There are PDF publicity items in

<http://www.ukuug.org/osa/2004/>

and more information in the parent directory, including a submission form.

In the last couple of years, entrants had to be in full time education in the UK. Whilst we are still focusing on that group – indeed one of the main aims is to encourage work and submissions from those in full-time education – nevertheless, we have this year removed that constraint. So, we should be grateful of any publicity you can give this scheme and refer any interested part to the web page

<http://www.ukuug.org/osa/>

Announcement: AsiaBSDCon 2004

We have received details from Michael Wu (Program Coordinator) of the USENIX AsiaBSD-Con 2003 which will take place at Academia Sinica, Taipei, Taiwan between March 13 2004 and March 15 2004.

For further details and the call for papers please see:

<http://www.asiabsdcon.org/>

Announcement: NordU2004

We have received notification of the programme of the NordU2004 conference to be hosted by DKUUG, the Danish UNIX-Systems User Group. The conference takes place on January 31st and February 1st at the Copenhagen Science Park Symbion. The conference is both preceded and followed by tutorials on the 28th to 30th January and the 2nd and 3rd of February.

Prices have been considerably lowered this year in the hope that more people will be able to attend the conference and tutorials. Details are here:

<http://www.nordu.org/NordU2004/>

Announcement: USENIX Annual Technical Conference

We have received details of the annual USENIX Technical Conference to be held between June 27th and July 2nd 2004 at Boston, Massachusetts, USA.

A call for papers is available here:

<http://www.usenix.org/events/usenix04/general.html>

Announcement: LinuxConf.Au 2004

We have received notification of Linux.Conf.Au, Australia's national Linux and open-source conference, which will be held in Adelaide, Australia between January 12th and 17th 2004. Included in the conference will be a "miniconference" jointly organised by AUUG and Linux Australia entitled "Linux and Open Source in Government: The Challenges".

Further details are here:

<http://lca2004.linux.org.au/>

Announcement: UNIX Internationalization Guide

Andrew Josey

The Open Group, in association with Sun Microsystems, is pleased to announce availability of a new Guide Book covering the Internationalization features of Version 3 of the Single UNIX Specification and IEEE Std 1003.1-2001 (POSIX).

The UNIX Internationalization Guide describes the internationalization facilities provided by the Single UNIX Specification, Version 3, which incorporates IEEE Std 1003.1, 2003 Edition (POSIX). Whether you are an experienced developer, a system implementor, a technical manager, or a user of open systems, this book provides the solid base of information you need to exploit the multi-language features of implementations that conform to Version 3 of the Single UNIX Specification.

The book includes the complete Single UNIX Specification, Versions 1, 2 and 3 on the accompanying CD-ROM in HTML and PDF formats. For more information on The UNIX Internationalization Guide including ordering information see:

<http://www.opengroup.org/pubs/catalog/g032.htm>

Andrew Josey is Director, Server Platforms at the Open Group.

OSS Watch

Randy Metcalfe

OSS Watch is the newly formed national free and open source software advisory service for UK higher and further education. Funded by JISC (the Joint Information Systems Committee), OSS Watch is a pilot service providing neutral and authoritative guidance about free and open source software and related standards. It offers a web-based clearing house for up-to-date information, conferences and workshops, focused assistance for institutions and software projects considering open source, and investigative reports.

Stakeholder communities for OSS Watch include IT managers trying to assess the role open source software should play in their institutional IT strategies, software developers concerned

about how to release their work with the appropriate license, and end users trying to decide which software best suits their needs. Providing support for such a wide range of interest is a large task, which OSS Watch can only hope to approach with the input and guidance of its many partners from education, industry and government.

The initial project of the service has been a scoping study on the state of open source software deployment and development in UK HE and FE. The results of this study will help direct the ongoing development of the service itself. The study will be released on 11th December at OSS Watch's inaugural conference, "Open Source Deployment and Development". For further information on this free conference please visit

<http://www.oss-watch.ac.uk/events/2003-12-11/>

OSS Watch is managed by Sebastian Raetz. It based within the Research Technologies Service of Oxford University Computing Services. To stay informed of ongoing developments within the service, please visit

<http://www.jiscmail.ac.uk/lists/oss-watch-announce.html>

to join our JISCmail announcement list. For further information on OSS Watch visit

<http://www.oss-watch.ac.uk/>

or write to info@oss-watch.ac.uk

Randy Metcalfe is communications manager for the Research Technologies Service at the University of Oxford (OUCS) where he divides his time between OSS Watch and the Humbul Humanities Hub. Previously he was communications director of the Institute for Citizenship. Sometime before that he worked on ethics and philosophy of literature.

UKUUG at the Garden of Eden?

Charles Curran

This event – Mac OS X Technology briefing – was different from what UKUUG has been doing in recent years: it was co-organized with Apple and held at their HQ at Stockley Park, near Heathrow. The organizing was done both by Sam Smith, from UKUUG, and Alan Bennett, Apple's Education Development Manager for Strategic Projects. Apple kindly provided the venue, lunch and refreshments, and the costly items in the organization free of charge. Thanks, Alan/Apple. It was well attended with around 100 participants. The speakers' slides should appear on the event's web:

<http://www.ukuug.org/events/apple03/>

There were four talks: Simon Patience, Apple's director of Core OS (= OS X), giving a detailed overview of OS X 10.3/Panther; Stuart McRobert, of Imperial College, speaking on selected bits of history and IC's experience putting together a teaching lab based on G5s; Ken Tabb of the Neural Systems Group, University of Hertfordshire on 'Home-made High Performance Computing', and Simon Cozens who talked about the scripting possibilities under OS X.

Stuart McRobert

started, unusually so for a UKUUG event since we know he is likely to have twice as many slides as minutes allocated to him. He had his usual amusing title, 'Apple Pie, a new Recipe', and somehow persuaded Apple's caterers to turn up with a large pie and a load of tarts (of the edible sort, I assume, but I did not manage to get hold of one). He started with a historical overview, including Apple II with a colour monitor in 1977, the Cray 1 with just 8MB of memory in 1976, the Onyx C8002 as one of the first micros running UNIX (Onix), Amdahl's UTS, the 80s dominated by PDPs and Vaxes. He didn't cover all the UNIX / Apple relationship, or the old story of Seymour Cray having used an Apple to simulate the Cray 3 before Apple/Jobs bought a Cray to help design the next Apple.

Stuart and Imperial's particular interest in OS X now is that they have just installed a G5 lab for the teaching high-performance computing. The theme to his talk was how the G5 is of supercomputing capabilities.

He praised student helpers, saying that they do a wonderful job, if you let them. That they can make and learn from mistakes.

Returning to history, on the network front, he mentioned Ethernet winning over ATM, and the potential of TCP Offload Engines, and NG products with six 10GbE per blade. On the storage front, he said that demand was driven by SPAM, that iSCSI storage was competing with expensive SAN solutions.

He mentioned the Virginia Tech (vt.edu) Terascale Cluster that has been in the news recently (see www.top500.org). It contains 1,100 Apple G5 systems each with two 2GHz PowerPC 970s; each node has 4GB memory and 160GB Serial ATA storage. There is also 176TB secondary storage.

Finally, he turned briefly to the OS, mentioning key differences with other desktop systems and traditional UNIX. He summarized by saying that the G5 was HPC to the masses, and that it came with the very best of up-to-date ingredients.

Ken Tabb

had a more personal approach to HPC. His problem was in Detecting and tracking moving humans which he was doing with Computer Vision and Neural Networks, so the main work is in edge detection. His Mac was supplemented by an AltiVec (a.k.a. VMX and Velocity Engine) co-processor with a 128-bit vector unit; it can process four floats at once. It can be driven by assembler, or an extension library via C, or Fortran. There are also some auto-vectorizing compilers (eg Vast's). He could get a 15-fold speed-up using it. His neural network was using MPI and Rendezvous Distributed between a group of Macintosh.

Ken then mentioned the OS X Performance Analysis tools, explained Neural Networks and finally said a bit about clustering technologies. His talk and enthusiasm reminded of the fun I had with Convex systems the decade before last, and stirred the need to see where compiler technology has gone.

Simon Patience

started things rolling after lunch. He had talked the previous week in France and Germany and was back on his original home ground to talk on OS X 10.3/Panther, Apple's fourth OS X, released in the month before. He gave quite a detailed overview, concentrating on the new and updated features. He said that he had had no experience of Apple systems before moving there last year. He has spent 20 years on UNIX kernel work. He worked for OSF, etc., and was lately director of Linux at SGI. In his talk he covered: the UNIX heritage of OS X, OS X architecture, Panther items for normal UNIX folk, open source and Apple, and the future - well, that what he said. I found his content somewhat lightweight but not uninteresting; I guess that Apple/SJ does not let anyone talk out of turn. Moreover, Simon made it clear whenever he touched on something interesting or was questioned closely that life would not be worth living should he disclose outside his remit. I suppose that's life, nowadays.

He started off saying that many did not know that Mac OS X is UNIX based, some even doubting that it is UNIX. That's the sort of distracting remark that either make you feel glad vendors can at last hide an OS (the way some of us tried 20 years ago), makes you wonder why it has taken so long, makes you think that they are still so pre-occupied by this that they haven't quite managed, etc.

I shall just concentrate on some of the bits of Simon's talk. I suspect that we won't get his slides, though there is a PDF Mac OS X for UNIX users at

http://www.apple.com/macosx/pdf/PantherUnix_TB_10082003.pdf

Apple also list the new features in OS X at

<http://www.apple.com/macosx/newfeatures/>

They chose UNIX because: it is powerful, extensible; it has an active and innovative developer community; it is standards based, although many of the standards are based on implementations in UNIX; it has a mature code base, providing a reliable system; it has multi-tasking; it has security and is multi-user; it has an admin user; it hasn't had a virus yet and UNIX design means that an attack is less likely. Whether that contrasts with the competition or Apple's previous OS was not said.

He mentioned that they had sold 8.5 million systems until October, and thus they would be subject to a very large pressure from those users. They have 6,500 native applications, up from 4,000 a year ago.

He talked on the architecture model: BSDKernel, Mach Kernel, and ioKit, all of which he said where open source except for some drivers over which Apple has no control. 10.3 is based on FreeBSD 4.8 and also has some 5.0 enhancements. He mentioned all the local and networked file system types supported. On Mach he said they used Mach 3 (OSF, rather than CMU) and that it was used for VM, tasks, threads, scheduling, IPC (also exported to user space), He is a proponent of not putting things in the kernel that do not belong there. The ioKit comes from NeXTStep's driver kit. Power management, as in sleeping a PB, is complex.

User space: BSD commands and libraries from FreeBSD, the X11 is based on that in Xfree86 4.3 and implements X11 R6.6, it includes basic X apps, there is direct access to OpenGL, native aqua and X11 applications run side-by-side.

On the mobility front, he mentioned that Steve Jobs said this would be the year of the laptop, but for him (SP) it was the first year of the laptop. He did not explain that but it reminds me to get rid of my ten year old PB165. He covered some of the interesting laptop v. desktop differences: physical v. network v. service changes; reachability and notifications; switching from wired to wireless, leaving and re-entering a wireless network v. going home and connecting from there.

On the command front, they have switched from sendmail to postfix, sar (sample and report OS statistics) has been provided, as have 100 new man pages.

On file systems, he said that HFS+ performance had been improved, and, on the server product, case sensitivity introduced. UFS performance was now close to that of HFS+. They are now supporting and working on many f/s; fat, for instance, is used in many digital cameras. On networked file systems, he mentioned significant performance improvements in NFS (4x-10x). Improved interoperability. NFS locking. OpenAFS 1.2.10.

Networking: significant performance improvements. Full Ipv6 support. L2TP/IPSec. VPN client and server. 802.11x. wireless authentication (TLS, etc). Firewall based on ipfw. Reachability services. Rendezvous Multicast DNS.

On open source, he said that Apple is source orientated; they are using lots of open source and feeding their contributions back.. They were about to release Rendezvous as open source.

On future items, and in general terms but relating specifically for UNIX, he listed for the short/medium term: mobility, dynamic configuration, and Rendezvous; for the medium/long term: information management, information distribution, and network presence.

Simon Cozens

gave a gentle but useful introduction to scripting under OS X. He started by defining scripting as interacting with the application level. He covered the various languages, giving examples and live demonstrations of all those he had had access to: Applescript, command-line (osascript -e blah), Python, Ruby, Perl, Ruby Cocoa, and PyObjC. He favoured OS X as it came with better scripting interfaces than many other systems.

Despite some of my negative comments-mainly because I had been hoping that we could pull off something at a detailed, technical level-the day was useful and there was time to meet with some

of the many interesting people in the audience. In summary, I think the most useful speakers were Ken Tabb and Simon Cozens since they may have given folks something to take away and try out or think about. Stuart's positive remark about students would be my quote of the day. BTW, Those interested UNIX history would do well to consult the likes of

<http://www.levenez.com/unix/>

which has a very interesting, or is it distracting, timeline chart.

Software patents

Alex Macfie

Unless you act soon, US-style software patents may soon be a reality in Europe because of the proposed EU directive “on the patentability of computer-implemented inventions”. Meanwhile, the US patent practice is starting to be seriously questioned within the US itself: the Federal Trade Commission recently released a report which is critical of the use of patents in the software industry [1].

Fortunately, the European Parliament voted in September for exclusion of software from patentability. A series of votes almost completely corrected the original proposal from the European Commission [2], which would have legalised the present, illegal, European Patent Office (EPO) practice of granting patents on software methods in a similar way to the US Patent & Trademark Office (USPTO) [3]. Its supporters claimed that the original directive would limit patentability – virtually no-one claims to be in favour of software patents! Sadly, it contained many loopholes which would allow software patents and its proponents opposed amendments that closed them.

We feared that the European Commission proposal would be accepted by the European Parliament (EP) without repair, as it was supported by leading spokespeople in both major European party groups (including Labour MEP Arlene McCarthy and Conservative MEP Malcolm Harbour). However, many concerned software developers, encouraged by FFII [4], went to the EP and explained to MEPs and assistants the issues at stake for developers, free software and the whole European software sector. It was also criticized by economists, who said that widening the scope of patentability would “lead to an increase in the strategic use of patents, but not to a demonstrable increase in innovation” [5]. As a result, the Parliament voted for a return to a strict interpretation of the original European Patent Convention, which explicitly forbids patents on computer programs.

It is now possible that the EP's good work will be undone. The European Commission opposes most of the EP's good amendments. The next stage in the directive's path to EU law is in the Council of Ministers. The Council's recommendations will be sent to the EP to for voting. It is much more difficult for the Parliament to amend legislation at this late stage.

The Parliament's open decision-making process aims to represent the constituents of the 625 MEPs, but the Council's internal process is designed to find a common position between the EU member state governments [6]. On this issue, the UK government relies on text drafted by the UK Patent Office, which has led the way in encouraging the widest possible patent scope [7]. At the recent meeting of the Council, the UK government was expected to back the original proposal, undoing all the EP's amendments. But instead, it recommended more time be taken to examine and reconcile the two radically different drafts. This was probably because of the large number of letters received by MPs on the subject.

Many other national governments have also taken a similar position [8], and so the proposal will now be subject to further consideration by the Council of Ministers. Therefore there is still the opportunity to influence the government. The most effective way to do this is to write to your local MP [9]. Some sort of EU directive addressing software patentability is likely to eventually

be passed. It is up to you to make sure that it is one which affirms that software should not be patentable, rather than one which simply legitimises existing EPO practice.

[1] summary of FTC report at

<http://www.ffii.org.uk/ftc/ftc.html>

[2] Side-by-side comparison of articles:

<http://www.ffii.org.uk/articles.html>

and recitals

<http://www.ffii.org.uk/recitals.html>

of the two forms of the directive. These also show which amendments are acceptable to the European Commission [3] “Art 52 EPC: Interpretation and Revision”, on European patent rules have been reinterpreted over the past 20 years to allow patentability of software methods

<http://swpat.ffii.org/analysis/epc52/index.en.html>

[4] Foundation for a Free Information Infrastructure,

<http://www.ffii.org/>

<http://www.ffii.org.uk/>

[5] Letter from economists concerned about effect of directive:

<http://www.researchineurope.org/policy/patentdirltr.htm>

Critique:

<http://www.researchineurope.org/policy/critique.htm>

[6] “Lobbying the Council”:

<http://www.ffii.org.uk/council.html>

[7] “The UK Patent Family and Software Patents”:

<http://swpat.ffii.org/players/uk/index.en.html>

[8]

<http://www.ffii.org.uk/latest.html>

[9] Enter your postcode in this form to find out who your MP is:

<http://www.faxyourmp.com/>

Reinventing Perl

Simon Cozens

Yesterday, I was speaking at a UK Unix User’s Group event. I always enjoy these UKUUG things, particularly because they give me the opportunity to have a chat with Josette Garcia of O’Reilly UK sales. It’s always nice to talk to Josette.

During yesterday’s chat, Josette asked me an interesting question: what can we do to reverse the decline in Perl book sales? This led me to have some deep thoughts about the profile of Perl in general. I don’t have as many deep thoughts on technology as I used to, so now that I have done, I’d like to share some of them with you.

A long time ago, Perl was the system administrator’s secret weapon; everyone depended on it, but nobody really admitted to using it. Gradually, in the same way as things like Linux and Samba, it gained acceptance in the market and people were prepared to stand up for it, and to use it for serious development tasks. This period lasted from around the mid nineties to the end of the dot-com bubble.

Towards the beginning of that period, companies such as O’Reilly realised almost a debt of gratitude towards Perl, which had saved their necks on many occasions. As a result, Perl was given some helping hands; again, thinking of O’Reilly, we were happy to set up and run perl.com

almost as a charity case. Others who hadn't had the ability to express their gratitude towards Perl got the opportunity in the first Perl Foundation fund drive - I'm thinking specifically of cases like Blackstar, Pair, and MSDW.

The end of the dot-com boom brought the end of this period of gratitude, and, again like Linux and Samba, Perl became part of the furniture. Those who felt the need to give money to TPF had had the chance to do so, and were relieved of their perceived debt. I believe that the majority of the funds from the other fund drives came from private donors, and that company donations are down.

With no third parties doing its marketing for it, Perl had to compete for management and developer mindshare with the new breed of Java initiatives, Microsoft's .NET, PHP, and so on, on its own merits. And, to be blunt, it's being left behind.

Looking for instance at JobStats, of current job postings, compared to 5% for C, 10% for C++ and 11% for Java. C# comes in at 3%. PHP and the like aren't reflected in job statistics, but seem to be gaining in mindshare at Perl's expense. I wish there was a better way to quantify this.

I'd like to think about what we can do with this. The first thing to think about is, of course, what we actually want. Do we want Perl competing in the same sphere as Java and C++? Do we want a 10% share of the job market, or is 3% actually enough?

Some would say that when Perl 6 comes along, then it will be able to play with the big boys. However, while I believe that when Perl 6 does come along it will be an immensely powerful language, I don't hold with the idea that a new revision of the language is some magic bullet which will simultaneously raise developer interest and give Perl the opportunity to regain lost ground. Perl 6, by the time it finally arrives, may be enough to rekindle interest inside the Perl developer community, but I'd like to do more than just preach to the choir.

Furthermore, surveying the developments in Perl at the moment, I don't see very much innovation. There were many areas in which Perl was doing new and interesting things, but the rest of the world has caught up. Where someone would previously use `mod_perl`, they may now use PHP; Axkit was a great idea, but now people have Cocoon to play with as well; Perl was the darling of the bioinformatics revolution, but there's a good deal of Python in there now as well; and so it goes on.

I'd like to see us try and find some more technology gaps to fit Perl into, such that we can establish Perl as the de facto language for certain niches. I believe this is the way we're going to bring Perl back into the programming mainstream.

Where are these technology gaps going to appear? I think we'll see some in natural language processing - the majority of the NLP tools I deal with at the moment are written in Java, but there are areas where Perl should come in instead - but that's hardly mainstream. Still, establishing some Perl outposts there would certainly not be a bad idea.

As OS X gains in popularity, I'd like to see more noise made about Mac::Glue and the scriptability of Mac applications. (the very thing I was making noise about in my presentation yesterday.) Establishing Perl as the power tool for interacting with OS X should not be difficult, as all the components are there, but it needs some profile-raising.

Nathan Torkington had some ideas a while back about getting Perl into the code analysis space, but I haven't seen any work in that area. It would be nice. I'd also like to see Perl involved in more core network technologies - `qpsmtpd` is a start, but why can't I have a programmable DHCP server or DNS server?

Oh, and finally, there is some good news: a graph on the Jobstats site shows that over the past few months, Perl's share of the job market has been slowly rising, and has crept back up to early-2000 levels. Raising the profile of Perl with new initiatives and new technology areas can only help this trend continue.

This article appears in Simon's blog and is reprinted here with his permission. Simon Cozens does "research programming" for Kasei, and runs perl.com for O'Reilly. He is author of "Beginning Perl", originally published by Wrox (now available in PDF format), and is co-author of "Extending and Embedding Perl". His new book, coming out soon, is a rewrite of O'Reilly's "Advanced Perl Programming".

<http://www.perl.com/>

"Beginning Perl":

http://learn.perl.org/library/beginning_perl/

"Extending and Embedding Perl":

<http://www.manning.com/jenness/>

"Advanced Perl Programming":

<http://www.oreilly.com/catalog/advperl/>

Simon's Blog:

<http://blog.simon-cozens.org/>

Jobstats:

<http://www.jobstats.co.uk/>

<http://www.jobstats.co.uk/jobstats.d/Details.d/Trends.d/SKILL/PERL.d/>

SC2003 – SuperComputing 2003 – Igniting Innovation

Ted Mariner

Reflections on SC2003, Phoenix AZ, 15-21 November

After watching the decline of the Supercomputing world and the SC conferences in the States during the middle of the 90s and then seeing its stirrings again as we moved into the new millennium, I can say, with some certainty and a small portion of relief, that Supercomputing is back. This was the most up-beat conference since the heydays of the Cray, NEC, Fujitsu, Connection Machine, NCube, Intel Paragon, and the rest, when all computers were works of art with reassuringly vibrant banks of LEDs flashing subtle but meaningless messages, as awe-struck attendees jostled past the stands.

It has taken a massive plunge in price from the super expensive days of the performance behemoths to the commodity world of clusters, based on the humble PC, to ignite the minds of the innovators in both academia and the commercial world. AMD's Opteron, announced last April, shows up as the core of several systems that go beyond the traditional Beowulf cluster. Octigabay's 12K High Performance Computer is an interesting example, scheduled for production ship midway through 2004; this box makes use of Infiniband technology to provide a low-latency interconnect that will allow very fast node-to-node communications between its dual processor Opteron based processors. The software development community has met many of the challenges of programming in the clustered world and now we are seeing signs of hardware providers looking to fill in some of the limitations of today's basic clusters.

The more traditional cluster providers, such as Racksaver, had a strong presence. Racksaver have a very business like product using low-cost blade technology to bring Opteron based servers to the market. There was no shortage of such cluster providers with Appro, Rackable Systems, Linux Networx, to mention but a few. Of course the big players, such as IBM, Dell, HP, SUN, SGI, et al, were there with higher end offerings, though these usually provide more high availability features than many HPC users may require and at a higher price too.

Continuing my quest for something different in the number crunching area of Supercomputing I came upon an interesting product from Star Bridge Systems, Inc., an FPGA (Field Programmable Gate Array) with a difference. It looks like it may well have enough resource to

be able to tackle some significant problems. However, as is always the case with FPGA type systems, you must be able to get code into production quickly and you must have enough work to keep the system busy to achieve the price/performance promises. Star Bridge feels they have such problems solved.

Connectivity and communications was my next area of interest, and it is clear that Infiniband is finally sticking its head well above the parapet. Previous Supercomputing shows have, to me, suggested that Infiniband had lost its way but now that has changed, and there are plenty of infrastructure providers, such as Mellanox Technologies, with product at prices that make this a definite candidate for cluster connectivity, even for the current generation of clusters. However all the interoperability claims will have to be demonstrated in the real world, and under pressure, before this technology becomes ubiquitous in the cluster world. I can still recall the many years between the first appearance of Fibre Channel and the time when we could plug together components from different vendors and actually get something to work!

There were interesting switches that provided layering of protocols across various hardware layers. The Topspin 360 switch allowed for an Infiniband connection from a cluster with a Fibre Channel connection to a disk drive and Ethernet connections. The Infiniband connection allows for the Fibre Channel and Ethernet protocols to operate across the Infiniband hardware layer, and so only one i/o connection to the cluster node is needed. Fibre Channel and Ethernet switches were also on show from the main providers but had nothing really new other than increases in maximum numbers of ports and consequent reductions in price per port. An otherwise interesting Plenary session on the future of Networks did not really give me any insights in to what was beyond Ethernet or Infiniband other than faster versions.

Another major area of interest at the show was storage. The emergence of reliable, very low cost IDE/ATA drives has fired the market and caused a buying frenzy of that product recently. There have been many large-scale SAN/NAS type storage solutions emerging over the past few years, fuelled by a venture-capital industry keen to find somewhere where there may be a chance of big returns. Many of these products have fallen by the wayside or been acquired by larger players looking for new drivers to add to their technology portfolio. They can be categorised as full solution (hardware and software), or software-only solutions. The latter has advantages for those wishing to integrate using commodity components, but for some a simple plug-in solution is preferable, especially if a customer does not have a large support staff. It is clear that, for many, the administration of storage is a major expense, and consequently, the sophistication of management tools, and ease of configuration, can be a significant factor in the choice of storage solutions.

Full solutions follow several different approaches, simplest is the traditional NAS approach, which can be very effective for many users. Several suppliers such as BlueArc and Spinnaker Networks have products that are scalable both in terms of capacity and bandwidth. These solutions tend to come in at a high cost per GB unless they have a very high capacity, allowing the low cost of the individual disk drive units to bring down the overall cost of the equipment. Panasas have an interesting object based storage system that has demonstrated very good performance in some benchmarks and, whilst it is not exactly commodity cost, it may be cost effective if the performance claims are sustainable in a real world environment.

There were software solutions available from Sestina, ADIC, Lustre, and Terrascale. They all approach the problem in a slightly different manner. Sestina's GFS solution can achieve good performance but has a significant cost for the full feature set. The Lustre file system solution is making progress but is not yet ready for prime time in a production environment. Quite interesting is the solution from Terrascale, which maintains impressive performance levels but with a very simple approach.

Basic disk solutions based on SATA disk drives were evident from several suppliers, LSI have a good product and have been quality providers in the Fibre Channel storage space. Data Direct

Networks also have good product and now at a more cost effective price.

Throughout the show there was a very strong commitment to provide more than just solutions, I believe all recognise that one of the most valuable additions a vendor can provide is to engage with the customer in the whole solution provision even if they do not provide all the components. No more pointing fingers between suppliers when an undefined problem occurs, or at least we can hope!

As a final note, I recommend visiting the SC2003 website and reviewing some of the papers in the technical program

<http://www.sc-conference.org/sc2003/>

the quality of the presentations was very high this year.

Ted Mariner is Vice President of Advanced Systems for Veritas DGC Inc, one of the leading Global Geophysical Services providers, headquartered in Houston, Texas.

Ted.Mariner@veritasdgc.com

To push desktop Linux, radical shift may be required

Andy Oram

In the server market, Linux and Microsoft were supposed to be mauling each other like jackals by now. Instead they are contentedly polishing off the bison of Solaris, IRIX, and other proprietary Unix server software. Linux and Microsoft Windows have both grown in the server market—Windows faster than Linux.

Linux on the desktop is a similarly confusing story. A conference on desktop Linux

<http://www.butrain.com/itp/desklinux.asp>

the first of its kind, was held in the Boston, Massachusetts area on November 10. The forum allowed the leaders of Linux and free software development to evaluate the progress these have made on the desktop.

Linux as an end-user system is at an early stage, but inroads are impressive. One statistic puts annual growth of Linux on the desktop at 44%. It is already in heavy use as a limited, kiosk type of application (point-of-sale terminals, for instance) and as a technical workstation. More general use is expected to come within the next couple years.

By now, free software office utilities are perfectly satisfactory and largely compatible with Microsoft Office; if they lack certain features of Office, they also lack certain bugs, and compensate with their own features and bugs. A sizeable base of knowledgeable administrators has emerged. And installation shouldn't be such a big issue. Windows installation can be hard too, and people often turn to professionals for installation.

So why hasn't Linux made big inroads yet among ordinary computer users? Let's look at a few theories—two that are relatively commonplace, and one of my own.

The first theory is that Linux's advantages will eventually overcome corporate and government conservatism. A roadmap was even laid out in the Desktop Linux Conference (described in my weblog from the conference at

<http://www.oreillynet.com/cs/weblog/view/wlg/3965>

In fact, the tipping point could be so near that we may all soon be laughing about the time we were worried about Linux's difficulties. Japan and China, combining one of the world's most important established economies with one of its most important emerging ones, are pouring huge amounts of money into Linux. IBM is no slouch either. People are getting it: Linux is a solution to many current computing ills.

A second possibility is that Linux may not catch on at all for Mr. and Ms. Average Schmo, at least not for the foreseeable future. But is that so important? Linux could meanwhile become dominant for servers, embedded systems, and kiosks. It could also reach the Average Schmos on large organizational networks using the Linux Terminal Server Project.

<http://ltsp.org/>

But we should also consider a third theory. Nat Friedman of Ximian (now Novell) explained at the Desktop Linux conference that the highest barrier to Linux adoption is the cost of rewriting applications. This was the conclusion of a consulting firm brought in by the city of Munich to determine whether it should replace Windows with Linux. The consulting firm warned that application migration costs would override the savings in licensing fees, and Microsoft came in with a stunningly low counter-offer. Munich decided to move to Linux anyway, for strategic reasons. But it's a hard decision to make.

Friedman and the Munich consulting firm were not the only ones to point this out. Back in September, the well-known consulting firm Gartner reportedly told companies that it would cost them money to move to Linux – precisely because they'd have to rewrite their applications. For desktop users, "migration costs will be very high because all Windows applications must be replaced or rewritten."

<http://www.silicon.com/software/os/0,39024651,10005952,00.htm>

And this is the same Gartner that had warned companies to get off of Microsoft Windows because of security flaws! (Before Gates and Ballmer started to make grand promises about putting security at the top of their priorities.) Despite Linux's advantages in the areas of licensing, stability, and openness, Gartner believes companies would lose money by switching.

Another article

<http://www.infoworld.com/infoworld/article/03/08/29/34FElinux1.html>

is more hopeful but suggests that it would take five years to see financial benefits after a switch from Windows to Linux.

And that leads me to my theory.

For Linux to reach the ordinary user, it has to offer more than good office suites and The Gimp and other free software implementations of common applications. Most people won't make the move just so they can keep doing what they did before. Security and freedom mean a lot to a few of us, but they are not enough incentive for the vast range of Average Schmos. And we need those Average Schmos; the median is the message.

People will move because they feel forced to – because there is an entirely new way of working that the old system cannot offer, and the new system can. It must be a shift that sweeps up millions of adherents and becomes a perceived necessity.

Historically, graphical user interfaces were just such an innovation (although if you were around when they first came in, you might remember how many ordinary people stubbornly stuck to their old command or full-screen utilities for years). The Internet was another: Microsoft, AOL, and others had to really scramble to avoid being swept into the dustbin by it.

No single new application is enough to cause a switch. Microsoft is perfectly capable of writing applications, so if somebody thinks up a neat utility on Linux, people will soon get something like it on Windows. What we're talking about is a new paradigm (pardon that word); a whole shift as big as GUIs and Internet use. What could it be?

Let me break my chain of reasoning here to point out that Microsoft itself is not afraid of changing the way people use computers. It's forging ahead with initiatives such as Longhorn and SharePoint which, if they live up to the hype, will put people in new relationships with their data and with each other. Microsoft has put tremendous energy into separating data from presentation and creating frictionless chutes that carry the data from database to office application to Web page with minimal user intervention.

As usual, one can get snowed when presented with Microsoft's lists of audacious upgrade features. But what emerges for me, as the basic Microsoft vision for the computing future, is an impressive pervasiveness of data – data that can instantly be viewed and tabulated by anyone who wants it using the most convenient tool at hand, without fussing over conversions and conscious transmission from place to place.

Microsoft is not stuck in the past; they're pulling as hard as they can to move their users to these upcoming innovations – trying to make them seem indispensable to staying competitive – because otherwise the company will have to stand by and watch the hose that gushes license fees gradually diminish to a trickle over the next couple years. No, Microsoft is pushing ahead. If any developers are stuck in the past, it's the free software programmers diligently recreating what's been done before.

But the kink in Microsoft's hose is that its business plan is a plan for business, not for end users. On the whole, Microsoft's initiatives revolve around corporate data use, and depend on adoption by corporations. And corporations are naturally conservative. They're afraid, for instance, that the grand SharePoint achievement of integrating office applications and corporate servers will lead to more bugs and security problems with both. They're not likely to budge.

Individual users, by contrast, are not conservative. History has shown them to be, if anything, quite reckless. Look at what hordes of ordinary people did when they get their hands on Web server software in the early 1990s. Look at the current popularity of instant messaging, and now SMS, both of which started as novelties. Look at the millions who signed up for the original Napster, and then slid over comfortably to current peer-to-peer systems.

So Linux has a natural user base it can appeal to. The very people advocates are trying to reach – individuals at home and in school – are the people likely to drive radical innovation in computing.

The area where Linux excels is services. Apache, Samba, MySQL, and mail transfer agents are practically household words thanks to Linux (although of course they run on many other systems too, and are found on Windows more often than people give credit for). Anything that you need to do that requires running a service benefits from the state-of-the-art network stack and security offered by Linux. This includes peer-to-peer applications, as I explained in a talk I gave back in 2002.

<http://www.openp2p.com/pub/a/p2p/2002/02/28/andyobru.html>

What's the advantage of running an application as a continuous, background service? Many find it hard to remember, because the division between server and client has become so commonplace (and the second-class citizenship of the Average Schmo, exiled to the client side, has been enforced for so long). Advantages include:

You're more in charge of your own data. You don't have to transmit it to some remote system under somebody else's bailiwick or beg for someone to put it up for you before others can access it. Immediacy opens up whole new dimensions, such as the ability to provide dynamic, instantly customized content.

You're more in charge of your own processing. You can choose when to process information in tiny chunks and when to postpone processing and do it in batches. You can choke off access or open up new threads to accommodate more. The simple, synchronous connections clients have may work for small amounts of communication, but when you get busy it's critical to have the flexibility of a server.

You're more likely to be able to support multiple users. Many servers recognize the idea of an account and offer access controls.

But running a service on your computer is socially disruptive. It puts control in your hands rather than in a central professional staff, so it's suspect in large organizations. It also bothers Internet providers because you need potentially more bandwidth, a static IP address, and perhaps

a domain name. But accommodations have been made for activities as diverse as file-sharing, Web servers, and chat. The practice may grow, and that's where the arguments against migration to Linux break down.

This article is reprinted by kind permission of the author. Andy Oram is an editor at O'Reilly & Associates, a book publisher and technology information provider. An employee of the company since 1992, Andy currently specializes in free software and open source technologies. His work for O'Reilly includes the first books ever published commercially in the United States on Linux, and the 2001 title Peer-to-Peer. His modest programming and system administration skills are mostly self-taught. He is also a member of Computer Professionals for Social Responsibility and writes often for the O'Reilly Network (<http://oreillynet.com/>) and other publications on policy issues related to the Internet. His Web site is <http://www.praxagora.com/andyo/> and his email address is andyo@oreilly.com. He works at the O'Reilly office in Cambridge, Massachusetts and lives nearby with his wife, two children, and a six-foot grand piano that can often be heard late at night.

The IEEE and The Open Group Launch POSIX Certification Program

Andrew Josey

The IEEE and The Open Group have launched the POSIX Certification Program for the 2003 Edition of IEEE Std 1003.1.

For the full announcement see:

<http://www.opengroup.org/press/03nov03.htm>

For a summary of web references for the new program see:

<http://www.opengroup.org/certification/idx/posix.html>

Piscataway, NJ and San Francisco, CA, 3 November 2003

The IEEE and The Open Group today launched a new program to certify products to the 2003 edition of IEEE 1003.1-2001, "Standard for Information Technology" Portable Operating System Interface (POSIX), which incorporates the IEEE 1003.1 Corrigendum.

Under the POSIX: Certified by the IEEE and The Open Group Program, suppliers can substantiate claims of conformance to POSIX based on defined test suites so buyers gain assurance that products they specify and procure meet the standard and are warranted by the vendor to do so. POSIX certification complements certifications for other products that draw on the POSIX standard, such as those for the UNIX(R) system, the COE Platform and LSB(TM).

The POSIX certification program is a voluntary program, but is required of suppliers who wish to use the POSIX trademark. POSIX certification is open to any product meeting the conformance requirements and is not restricted to any particular operating system implementation. The program includes a product standard for each type of product that can be certified within the POSIX Certification Program. In this initial iteration of the certification program these are as follows:

1003.1-2003 Base Product Standard: This is a profile product standard that comprises the mandatory functionality from IEEE Std 1003.1, 2003 Edition. It is comprised of two component product standards.

1003.1-2003 System Interfaces Product Standard: This is a component product standard for the mandatory system interfaces and headers related functionality from IEEE Std 1003.1.

1003.1-2003 Shell and Utilities Product Standard: This is a component product standard for the mandatory shell and utilities related functionality from IEEE Std 1003.1.

A product can be certified against one or more product standards and the program allows for two levels of certification: Platform Specific Certification, which applies to a single defined hardware and software environment, and Product Family Certification, which applies to all members of a binary-compatible family.

IEEE-SA will license the POSIX trademark for use with certified products. The Open Group is the certification authority and administers the web-based certification system and the two test suites needed for certification.

The POSIX trademark can be optionally licensed for use in association with certified products meeting the 10003.1-2003 Base Product Standard. The POSIX certification system is a web-based workflow system designed to lead applicants through the process to submit a product for certification. The two POSIX Conformance test suites, VSX-PCTS2003 and VSC-PCTS2003, are freely available to organizations that register to apply for certification.

More information about the program, including all supporting documentation is found at <http://posixcertified.ieee.org/>

Essential CVS

Jennifer Vesperman
O'Reilly and Associates

ISBN 0-596-00459-1

336 pp.

£ 28.50

reviewed by Damian Counsell

For many developers CVS is just part of life. It is fundamental to the management of any coding project. Powerful but discreet, it acts both as a smart catalogue of changes and as a way of combining the efforts of multiple contributors into a coherent working whole. It's more: a back up of your work, a record of your labours, a means of settling arguments and a way of combining multiple imperfect efforts made over time into a more stable product. You only notice CVS when it goes wrong. And the only times CVS version control goes wrong for me are the times when I have made stupid mistakes.

Fear of screwing up a valuable resource may be one reason why most users stick to a few, familiar commands and options and are reluctant to think about the whole system and its potential. When you are afraid you cleave to what you know. If you are similarly inhibited, then this book could encourage you to experiment. Another reason I tend to be conservative with CVS is that I set up new CVS repositories and/or import new projects into CVS rarely. Every time I do, I refer to my own notes or to resources on the Web. Even though I am more or less at ease with the process it's important to get it right. So, like sed or awk (for which there are also O'Reilly books), CVS is a tool that is useful to me, but which I explore infrequently. Even if you don't want to explore the more exotic options of CVS, a book like this can be a valuable prop for general work.

Code examples and commands are usually checked thoroughly in O'Reilly publications. We tend to take this aspect of their creation for granted, so, for this review, I did. The quality of O'Reilly production should also be a given, but I noticed that the paper of this volume was a little poorer (rougher surface, slightly more transparent?) than I've been used to from the the house publishers of the open source community. However, not only is there a useful pull-out reference card in the back cover, but the animals on the front (bobacs) are exceptionally cute by O'Reilly standards.

Essential CVS has an excellent introduction, explaining what CVS is. Version control and the client-server model of computing were two of the more subtle and difficult concepts I had to

get my head around when I first worked in a Web programming team. The opening of the book makes some insightful philosophical points – CVS is not just for the nasty things in life. As the book points out, you can use it for writing prose, graphic art and even to manage a shared household shopping list. At the beginning there is also a solid chapter on “Basic Use” and plus a “Quickstart” chapter. For anyone coming to CVS for the first time, these two chapters would be an excellent place to begin. The installation guide seems handy and impresses by covering multiple installation interfaces and distributions.

What did I learn from the middle of this book? I finally understood the real differences that using the binary file options make to the way CVS handles your work(s). I read about good criteria for project branching (“Tagging and Branching”). In the “Repository Management” chapter I found out where to go to customise the system to set different global behaviour and I discovered some frankly scary things you can do with a CVS repository if you directly modify the files in its root directory. This chapter even lists a script you can use to freeze the whole thing. It’s good that the book has a “Remote Repositories” chapter in which communicating via the various remote protocols are described, but I suspect that the majority of the material there is of academic interest only to most CVS users. They will have a local repository or be fairly restricted in the number of remote repositories they use and the means by which they use them. I skimmed it and I do use both local and remote CVS. In a book like this, though, completeness is important – if only to justify the premium readers must pay for a commercial publication over obtaining their advice online.

I have found, for example, that, when working with CVS gets stickier than I would like, “Google Groups” (<http://groups.google.com/>) has been useful. I suspect that the necessarily limited coverage in the “Troubleshooting” chapter of Essential CVS can’t really compete with the ’Net for finding solutions to specific problems. The “Command Reference” section of the book does a good job of being friendlier than “man”, being more clearly laid out, including more examples and appearing (crisply printed) on low power consumption dead tree. I have recently become an occasional and happy user of “Cervisia” when I want a more graphical overview of my file revisions in a directory and can see now why interfaces like it are useful even to people comfortable with command-line CVS. There is a brief account of Cervisia and various other front-ends to CVS in the “Clients and Operating Systems” appendix. The final chapter “Administrator’s Tools”, introduced me to a whole array of CVS spin-off programs and variants.

Essential CVS does what you might think would be a small job more effectively and more comprehensively than I would have imagined, even allowing for the generally high standard of O’Reilly books. Apart from covering the boring stuff well, it has two big advantages over the Web: it backs up its discussion of various CVS functions with wise advice and policies on why and how to use these facilities – plus it provides readable (and, presumably, tested) examples. Computing science prizes abstraction and generality; computing practice should, like this book, be informed by concrete specifics and sensible rules-of-thumb.

Damian Counsell is a Bioinformatics Specialist at the Medical Research Council’s Rosalind Franklin Institute of Genomic Research (formerly known as the MRC HGMP- RC). His academic homepage is at <http://www.hgmp.mrc.ac.uk/~dcounsel/> and his personal homepage is at <http://www.counsell.com/>. These Websites are not in any way related.

BLAST

Ian Korf, Mark Yandell, Joseph Bedell

O'Reilly and Associates

ISBN 0-596-00299-8

360 pp.

£ 28.50

reviewed by Damian Counsell

Bioinformatics is the term people use to describe the young science of using computers to analyse genetic data, particularly in the form of sequences of genes (DNA) and their products (RNA and protein). I argue that “bioinformatics” describes all sorts of activities on the interface between biology and computing, but it’s like trying to persuade people that computers are not just those three-box devices that sit on their desks, nursing Windows infections.

The most commonly used bioinformatics software belongs to a family of programs called BLAST. BLAST does a simple – but difficult and powerful – thing very fast: it compares one sequence of nucleotides (DNA letters) or amino acid residues (protein letters) with lots of others. Many biologists who would never call themselves bioinformaticians do a lot BLAST jobs; some spend more time running BLAST than they do walking around a lab wearing a white coat.

Of the three O'Reilly bioinformatics books I have looked at so far, I think BLAST is the best. It covers an important topic with a depth and breadth which, to my knowledge, no other book on the market has. I learned more from reading through the practical chapters during a couple of train journeys than in hours of wading through any other BLAST documentation. I am not exaggerating when I say that any biologist planning on doing a large number of BLAST searches – especially with a view to publishing results or drawing significant conclusions – should be made to read the relevant sections of this book first, well before going anywhere near a computer. The “20 Tips To Improve Your BLAST Searches” and “BLAST Protocols” chapters are alone worth the price of the book £28.50 to a practising molecular biologist.

O'Reilly's BLAST book may be the best we have, but it is not perfect. I found the theory chapters both fascinating and confusing: fascinating because they mentioned intriguing experimental results that should change the way people use BLAST in (for example) protein science and because they include concise Perl implementations of important algorithms; confusing because some of the explanations lost me and there were some important statements that I just disagreed with. Examples of the latter: scores are not (as the book says) necessarily metrics; genes that are 80 percent similar do not have an 80 percent chance of a common ancestor; genetic drift is about the “flow” of alleles into and out of a population over time and not about changes in sequence of a gene through time. A further illustration: this book contains a clear description of “dynamic programming” – an approach central to most alignment algorithms – but a poor account of evolution – a process many believe central to most biological sciences.

There is much more of value in BLAST: rich information about installation, optimization, hardware and database issues; useful appendices and, of course, reference chapters on commands and options for the major BLAST versions. Although I believe there are some glitches in the text and I would have constructed the book differently (partly to engage the “casual” reader more completely) this is an important work. BLAST should be a talismanic tome against bad bioinformatic analysis in contemporary biomedical research, but because of the geeky aura of O'Reilly books and because of the priority it gives to theory, it may not be read by as many of the high priests of modern biology as it should; those that do read it may not do so as closely as they should. More fool them. But well done to the authors. I believe this volume will be referred to unambiguously by both computational and biological scientists as “The BLAST Book” for years to come.

Damian Counsell is a Bioinformatics Specialist at the Medical Research Council's Rosalind

Franklin Institute of Genomic Research (formerly known as the MRC HGMP- RC). His academic homepage is at <http://www.hgmp.mrc.ac.uk/~dcounsel/> and his personal homepage is at <http://www.counsell.com/>. These Websites are not in any way related.

Amazon Hacks

Paul Bausch

O'Reilly and Associates

ISBN 0-596-00542-3

302 pp.

£ 17.50

reviewed by Gavin Inglis

The online bookshop Amazon seemed a real novelty when it launched in 1995. Concerns were raised about stolen credit card numbers, and one vocal school of opinion preferred the atmosphere of the high street bookshop: the look, feel and even the smell of its stock.

These days Amazon is more like an institution - or an addiction. In a culture where blogs are a standard means of communication and "you are what you buy", it seems perfectly logical to buy your books (or music, or software, or kitchen gadgets...) from a web site dripping with user features, referral bonuses and integration opportunities for your own site. Throw in a public Web Services interface, and this book was inevitable.

The 100 hacks detailed within are a varied assortment. They range from general concepts (not really hacks) to very specific exploitations of Amazon features. Some seem trivial: shortening URLs or adding a search bar to the address bar of the web browser. If you can't afford a book right now, you can at least use its ISBN number to generate a picture of it with a customised discount label of anything up to 99%. Unfortunately this hack does not actually change Amazon's pricing.

Some other hacks really amount to a tutorial in using Amazon's interface, from writing reviews and managing wish lists to setting birthday reminders. But there are a substantial collection of actual hard code examples, particularly in the section about web services. You may bristle at the early hack which insists on Internet Explorer for Windows, but by the end the technology trail has visited Mozilla, PHP, MySQL, Perl, Javascript, SOAP...this diversity appears to be a deliberate choice on the part of the author. To perform some of the more involved operations it will be necessary to register as a developer and obtain a personal tag.

The hacks are sensibly organised into sections, from simple browsing and searching, to the site's community scheme and becoming a seller in your own right. Some of the hacks are really rather nice, like the one which fetches album cover artwork while you listen to MP3s. And you could hardly get better post-sales support than the author's personal weblog which builds on the book's contents.

Amazon Hacks is effectively a user manual for Amazon pitched at geek level. It serves well as a general introduction to both the shop front and the accessible operations behind the scenes. The book should entertain and educate programmers keen on online shopping and is very easy to get into. But if you buy it, remember it's only a guide to a single, online store. Don't complain if that small independent bookshop on your corner goes out of business...

The Web Programming CD Bookshelf

Stephen Spainhour et al

O'Reilly and Associates

ISBN 0-596-00510-5

576 pp.

£ 92.50 + VAT

reviewed by Gavin Inglis

The web professional inevitably regards electronic texts, or “ebooks”, differently from the casual or recreational reader. Seldom working without multiple browser windows, she is as comfortable locating reference information on the web as on a nearby bookshelf. This observation has motivated an entire series of CD Bookshelves from O'Reilly. These titles collect a number of established paper books on one CD, add a search engine and cut the price by approximately 50%.

Accessing the books is a pleasant surprise – they are formatted in simple but clear HTML and therefore can be accessed in any web browser without the need for decryption or irritating plugins. This is a bold move on O'Reilly's part and is to be applauded.

Half of this particular volume is devoted to PHP. Following the O'Reilly convention, ‘Programming PHP’ is the core manual for the language, written by its original author amongst others. It starts from the basics and passes through the landscape one would expect: language features, databases, objects, XML and security. There are specialised topics too, such as using the GD library to create dynamic graphics and the construction of PDF files on demand.

‘PHP Cookbook’ is the other familiar format here: a library of pre-written examples illustrating likely PHP tasks. These recipes are often useful to solve a problem at short notice; they are also a valuable resource for learning the language. In this volume they range from the humble “Reversing a string by word or character” to the mighty “Displaying weather conditions” – interrogating weather stations around the world using SOAP.

The final entry in the trilogy is ‘Web Database Applications with PHP and MySQL’ – a focus on database-driven web sites with plenty of example code and interesting case studies. Taken together, the three books could bring a PHP newbie up to web application standard, with plenty of support in the form of the cookbook examples. Such a learning process would require a lot of staring at the screen, however.

In the world of paper, these books are dwarfed by David Flanagan's ‘JavaScript, the Definitive Guide’. A title suited primarily for reference, this fourth edition moves away from attempting to catalogue every little browser idiosyncrasy and concentrates on the more recent standards, Javascript 1.5 and level 2 DOM. This type of book suits the electronic format better than purely tutorial texts; our developer gets a lot of reader-friendly definitions on a single CD, which is much more portable than the weighty tome this book would otherwise be.

Speaking of weighty, even the previous volume is cast into shadow by Danny Goodman's ‘Dynamic HTML: The Definitive Reference’. This attempt to incorporate all the vendor and standards information about the devil's stew of technologies that is “DHTML” comprises an immense 1400 pages of dead tree in its second edition. There are seven chapters introducing and illuminating the concepts, but the rest is all reference: HTML/XHTML, DOM, stylesheets, Javascript/ECMAScript... the comprehensive indexes and cross-references are probably worth the price to any serious developer in this area.

Finally there is the third edition of ‘Webmaster in a Nutshell’ which is also included on paper as a “bonus book”. This collection as a whole illustrates that O'Reilly books do significantly overlap, in terms of technology. What differentiates them is their central focus and depth/breadth of coverage. ‘Webmaster’ attempts to cover just enough about a variety of technologies that the average webmaster will reach for it first with simple problems. No doubt this is why it was

chosen as the “hardcopy” book in this package. A significant amount of its content – PHP, Javascript, HTML, CSS, XML – is covered in greater depth in one of the other books. Any developer sufficiently convinced of the CD bookshelf concept would probably prefer to lose the hard copy and a few pounds off the price, but one imagines the paper book gives the package the presence it needs for a bookshop.

A further advantage of purchasing these books on CD is the composite index and search functionality. The index is presented in conventional form but gives references across all six books. There is also a packaged Java applet which offers free text searching across the books without any need for a web server. It works well when it works; unfortunately there appear to be compatibility issues with Mozilla.

The CD Bookshelf format offers a web developer portability and extra functionality across the included titles, along with significant savings; a rational model for the ebook concept. Of course the saving only becomes significant if our web developer would have bought all six books anyway. Despite the hefty price tag, this is a good investment – and friendly to trees.

Learning Perl Objects, References and Modules

Randal Schwartz with Tom Phoenix

O’Reilly and Associates

ISBN 0-596-00478-8

205 pp.

£ 24.95

reviewed by Stephen Quinney

The title of this book does a fairly good job of telling the casual browser in the book store what it is going to cover. To give you more of an idea of what is in this book I will go into slightly more detail. The authors use 5 chapters to give a thorough coverage of the whole topic of references right through from simple usage, creating complex data structures out of nested arrays and hashes, to subroutine references and the associated advanced ideas of callbacks and closures. They devote 4 chapters to the use of objects, which goes a fair way to explaining how to utilise most of the common object-orientated code design principles within your Perl. They spend a further 4 chapters looking at various different aspects of using and writing Perl modules, from simple usage through to how to package and distribute your code on CPAN. It was very good to see that one of these chapters is given over to how to make use of a few of the excellent Perl code testing modules, this is something that I think has been a little overlooked in the past.

This book is aimed directly at those who are fairly new to Perl. It has a secondary title which is “Beyond the Basics of Learning Perl”. The authors have also written the excellent “Learning Perl” book and this new book is designed to be a followup to cover the topics missed out of their previous book. As such the experienced Perl programmer is not going to get a lot out of this book although I did find some sections which really helped to improve my understanding of a topic.

The authors benefit greatly from being involved in teaching Perl programming skills on a regular basis, and are certainly capable of “Making Easy Things Easy & Hard Things Possible”. Having recently taught an introductory Perl programming course based on the Learning Perl book I can say that the authors have a very good style and approach to separating out and explaining individual topics. They are able to break problems down into small enough chunks that the interested reader does not have difficulty in getting to grips with each new topic. This new book certainly lives up to the expected standard they have set for themselves. The examples are clear and well chosen and the explanations help the reader to gain a deeper insight into each topic.

They also have an amusing writing style and a slightly odd theme to each book - this time it is Mr Ed and Gilligan's Island - which helps to make sure things do not get boring.

This book is not intended to be a replacement for "Programming Perl" or "Advanced Perl Programming" but it does provide a much easier way into the advanced topics than either of these two books. It will particularly suit those who are fairly new to Perl but most Perl users will find something of interest. I would definitely recommend this book for those interested in teaching courses as well as those wanting to learn. I cannot really find fault, except maybe to say that it looks a little thin for the price. A mere 200 pages for the same price as the 300 page Learning Perl, but that said there is a lot crammed into this book.

Stephen Quinney is a Unix Systems Programmer, and sometimes Perl tutor, for Oxford University Computing Services. When he's not wrangling Perl code he is often found working hard to get his Debian packages ready for the fabled next stable release.

LDAP System Administration

Gerald Carter

O'Reilly and Associates

ISBN 1-56592-491-6

308 pp.

£ 28.50

reviewed by Raza Rizvi

I have been musing about LDAP for ages, so I jumped at the chance to review this text to see if it cleared up any of the cloudy aspects of what I felt I ought to know. It did the job – though it might be better titled as OpenLDAP System Administration since it deals only with the largest open source, non-commercial implementation (other commercial directory offerings get fleeting mentions except for the Kerberos based MS Active Directory).

This is a clearly written and well structured book with good use of example and figures, in fact at times whole chapters seem to be in the constant width font used to indicate code fragments or user input!

The background is clearly set out with the reasons why one might wish to deploy a directory based system and a distillation of the relevant LDAPv3 terms (LDIF, Schemas, Attributes etc). It is worth paying attention to the basic terminology, as it is used, not unexpectedly, throughout the rest of the book. Those older readers who were familiar with X500 can chuckle quietly to themselves?

Chapter 3 shows how OpenLDAP is installed, configured, and secured, with deployment being covered in chapter 4 using a fictitious company. It was a shame that the company was not also used as the book expanded into how resilience and replication should be implemented?

The second half of the book deals with real-world integration of LDAP into systems applications, starting with the obvious candidate, NIS. This is a full description with good examples, and the book continues to similarly cover email, both popular mail clients accessing white pages of user info akin to that created for the fictitious company and mail servers (send-mail/postfix/exim) using it for mail routing.

The core Internet services (HTTP/FTP/RADIUS/DNS) are given the LDAP integration treatment in satisfactory detail along with Samba and printing.

Although this is an OpenLDAP text, the author is clearly aware that it will often have to live alongside some older database technology or some other pretender to the crown of directory king. Sensibly he chooses to base his example on Microsoft Active Directory (AD) and there is a reasonably detailed example of the creation of a single directory structure using both AD

and OpenLDAP though it's horses for courses as to whether it will be of use to your own organisation. Perhaps more of interest are the latter details on how to have multiple LDAP servers for multiple purposes using multiple vendor solutions, and again Active Directory is the chosen example?

The book rounds off with the PERL Net::LDAP module, and a whole string of useful snippets to search, add, delete, and modify entries. Clearly these will save time and hair-pulling for some people.

Although hard going at times, the book has been immensely useful as an introduction to LDAP at a moderate level. It doesn't cover every aspect of the protocol but there is more than enough to act as a decent grounding.

The OpenLDAP sections are very good given that it is easy to put a test server up to see what and how your company might use LDAP services. The use of Active Directory for examples was wise and well done.

But the best part for me was the integration with real examples of applications, it is clearly illustrated how to configure both LDAP and the application to inter-work.

I thoroughly commend the text to those who are looking to centralise information directories.

Building Wireless Community Networks

Rob Flickenger

O'Reilly and Associates

ISBN 0-596-00502-4

182 pp.

£ 20.95

reviewed by Raza Rizvi

I came to this book somewhat sceptical suspecting that it would be full of waffle about creating cosy environments to get around broadband provider blackspots.

I was wrong, it actually turned out to be both an interesting read, written partly in the first person, and a good source of wireless information written in a relatively accessible style!

It is true that some (but by no means all) of the information is only of use to the United States, but the underlying details of the 802.11b standard will be of use to those who want to either altruistically enable their community, or even to those who want to tinker with setting up stuff in their own homes (or make use of wireless kit in someone else's home without them realising).

The slim book runs through the technical background to the IEEE 802.11 standards family before describing the hardware and software components of the wireless network (cards, DHCP, NAT, routing, VPNs). Access Points themselves are dealt with in a separate chapter using the Apple AirPort as an example (though you need not have a Macintosh to make use of this AP).

For the geeks, chapter 5 covers the building of a Linux PC based access point with plenty of URLs and some code/configuration file snippets. Antennae themselves are covered over the next few chapters, with some US bias, but this is more noticeable when discussion turns to 'cantennas', the use of kitchen items as wireless equipment (ala the ubiquitous Pringles can).

The last 40 pages can be ignored, as they deal with local community projects in the US (except that Consume does get a mention), and with the FCC regulations for wireless spectrum use - obviously not appropriate in the UK.

The author manages to keep your interest through his use of personal stories and tips (like not killing yourself when working on flat roofs late at night!). Despite the low page count, he manages to pack in enough information for it to be useful without it overwhelming the reader.

It's slightly eccentric, it's US biased, but it is worth reading if you have an interest in the area.

Raza Rizvi is Technical Manager at REDNET, a business ISP and Cisco Premier Partner, based in High Wycombe. He recently re-certified as a Cisco Wireless Systems Engineer. He notices (see above) that everything has LDAP support these days ...

JavaScript and DHTML Cookbook

Danny Goodman

O'Reilly and Associates

ISBN 0-596-00467-2

540 pp.

£ 28.50

reviewed by Lindsay Marshall

Having moaned in another review about the "Hacks" series, here is a book that really ought to be in it and it isn't. It dives straight in with no preamble and dishes out page after page of code for doing small, but important, hacky things in JavaScript. It tells you about nasty detail like compatibility with browsers and has deeply meaningless titles for the "recipes" as the author calls them: "Doing something with a property of an Object". What? Nor am I that much wiser after reading the recipe - I can see what's going on but I just don't have enough context to see how or where it would be useful, however, I'm pretty sure that it would be useful to someone somewhere. In fact, it really is a book full of useful code, for all those occasions when your brain dries up and you need a quick fix for a nagging problem. (Always assuming that you do in fact write JavaScript, not something that I personally am in the habit of doing if I can avoid it).

Once you get to the DHTML end of the book, some of the code is getting pretty dense. I assume (and fervently hope) that all the code is downloadable from the net because I shudder to think about having to try to type it in accurately from the book. The Preface talks about downloadable examples, so I am slightly concerned that not everything is available, and there is no CD with the book. Correctly, standards are waved (but not waived) throughout the text, though I think I would have been happier for the author to come out and say "let's just forget about release 4 browsers and earlier", it really is time to move on and use appropriate standards wherever we can and let the non-abiders catch up.

This book reminds me of one of those books you buy in specialist bookshops that tell you how to fix your washing machine yourself - not quite as slick as a Haynes manual but really useful. If you really must soil your fingers by dabbling with things like DHTML then this is a book you need to have next to you.

Mac OS X Hacks

Rael Dornfest and Kevin Hemenway

O'Reilly and Associates

ISBN 0-596-00460-5

430 pp.

£ 17.50

reviewed by Lindsay Marshall

OK, you forced me, but I'll admit it: I don't like the "Hacks" series (at least, what I have seen of it). And in fact I'll go so far as to say that it is a complete waste of time, paper and money. Yes, the books have all the usual O'Reilly production values, dot, dot, dot, and the content is

well written by people in the know just as (nearly) always. But there is something lacking and I suspect that it is a point. In reality, who is interested in Google hacks, or Amazon hacks or Ebay hacks? I use the systems all the time but I have never felt the slightest urge to make them do slightly odd and useless things so why would I want a book on how to do these things? It's a bit like a book of unfunny cartoons (good grief, O'Reilly has that too - User Friendly). And what exactly is a "hack"? I suspect that the authors are using the word in some vague American English sense that means something like "really cool stunt", but what they describe are usually well engineered, if point-free, programs or programming tricks that bear no resemblance to anything I would call a hack.

The book under review here stretches this even further. This is not a book of "hacks". There are descriptions of how to set obscure options for programs and a load of stuff about the UNIX command line and what you can do with it. Lots of useful and good stuff in fact, assuming you want to know about enabling webDAV and such like. (There is weird stuff though like how to get a screenshot of the login screen). But not hacks. Describing them as such is just stupid and possibly insulting. The book's subtitle is "100 Industrial-Strength Tips & Tools" which is a slightly better description of the content, though I would argue with industrial-strength. Yes, it's just another book on Mac OS X but probably because the Hacks series is selling and people want a nice line of matching purple spines, they've done some BoTeX injections and some line-oplasty to reshape it to the desired image. I'd give it a miss.

Content Syndication with RSS

Ben Hammersley

O'Reilly and Associates

ISBN 0-596-00383-8

208 pp.

£ 20.95

reviewed by Lindsay Marshall

I've been generating RSS feeds from a couple of my websites for a quite some time now, and, if I am perfectly honest, I never really understood what was going on. I poked around the web a bit, looked at some examples and cobbled up the PHP to generate something that looked like what I had seen. And nobody complained or said it didn't work, so I left it alone. (Admittedly, the lack of compliant could be entirely due to a readership of zero for the RSS feeds, but I like to think a few people look at them and that they would have moaned about any problems - they usually do if something is wrong). Then I got this book. Great, I thought, now I'll be able to understand what it's all about instead of going round in circles on Dave Winer's website becoming more confused or else reading diatribes about DW being wrong in everything including his use of < and >.

And so, I read this book. And now I am even more confused and I think I know less than I did when I started. If you stand a long way back from RSS (try Alpha Centauri) it looks like a nice straightforward idea, well at least straightforward as anything slightly complex in XML can be. But get in close and find out about versions 0.91, 0.92, 1.0 and 2.0, which don't follow in sequence and which aren't that closely related to each other and which downright conflict with each other, then it gets nasty.

Ben Hammersley struggles manfully to make it all clear but it didn't work for me - I could feel all the politics and history hidden behind those innocuous tags and in the end I just couldn't be bothered with it. I wanted to get hold of the people behind the "standards" and knock their heads together and make them stand in the corner till they promised to play nicely and not do it again.

None of this is the fault of Mr. H who has a very nice looking website which is well worth a visit. In fact if you have to work with this pig's breakfast then I recommend this book to you (it may even be the only game in town!) as it's all there.

Learning Web Design 2nd Edition

Jennifer Niederst

O'Reilly and Associates

ISBN 0-596-00484-2

454 pp.

£ 28.50

reviewed by Lindsay Marshall

I do like this book. I liked the first edition, and I liked the author's other books. They are models of what textbooks should be : clear, concise, nice to look at, useful and correct. This is a larger format book than most O'Reilly texts and the extra space is used to great advantage with lots of sidebar notes, colour illustrations and sensible, helpful information. And best of all it says, in black and white, on page 7, "Writing HTML is not programming" - shout it from the rooftops.

My only quibble with the book is a lack of good coverage of accessibility issues. Advice such as "avoid putting important text...in graphics" is no longer acceptable. "Don't" is the operative word and new designers need to be made aware of the legal requirements that they may have to meet. Usability is referred to, but not accessibility and they are not the same thing. I suppose that they have to have something to put in the third edition! (No mention of validation either. Oh well)

If you know someone who wants to get started with designing websites and wants to progress quickly and reasonably far, this title is definitely for them. Get people who design bad websites to read it to, we can hope.

Learning XML 2nd Edition

Erik T Ray

O'Reilly and Associates

ISBN 0-596-00420-6

400 pp.

£ 28.50

reviewed by Lindsay Marshall

I thought that I had reviewed the first edition of this book, but I can't actually find it on my disc - some other O'Reilly XML books, but not this one. That's one of the troubles with XML: it all starts to blur into a mass of <, > and monospaced type (LucasFont's TheSans Mono Condensed, if you really want to know), especially when they start throwing in namespace references. As I've said before (probably several times) I think XML is OK and that it can be quite useful, and even if it wasn't, you still have to know something about it just so that you can understand what the early adopting, bandwagon jumpers are talking about. This book is not a bad place to start. Good coverage of the basics, through schemata (new material here) ([sigh] another author who doesn't know the correct plural) all the way up to XSL-FO. Tons of examples, lots of explanation - I think that I have a clue about how to use FO now!

Of course, there is a lot of material that is covered in literally dozens of other books that you have on your shelf (for instance CSS) but we've come to expect that. Ultimately there isn't a lot

to say about the useful bits of XML, and the rest is so roccoco that only people who need it will go there.

I lend this book's predecessor to students who want to learn XML and they find it useful. I don't think that you can say much better than that about a textbook, and since this is release 2 it's bound to be better!

BTW my favourite section heading in the book is "Dubious vendor extensions" – how do you parse that?

C++ in a nutshell

Ray Lischner

O'Reilly and Associates

ISBN 0-596-00298-X

808 pp.

£ 28.50

reviewed by John Collins

This is a very comprehensive and carefully written reference manual for C++ for the ISO 14882 standard covering the language and the standard library in the two halves of the book. An appendix covers features of well-known implementations, including Microsoft and GNU. There is a glossary of terms just before the very detailed index.

I thought that the language description section was very clear with some good examples of what is being described and some interesting "tricks" such as recursive templates. It has icons to signify tips and warnings for the user which are helpful, especially in the area of portability.

I really liked the standard library reference section which is far better laid out than Stroustrup's book in my opinion. At last I have everything I might want to do with strings or streams in the one place and in a nice alphabetical order rather than having to put lots of little coloured markers on the edges of pages like I have in my copy of Stroustrup. My particular "gripe" with Stroustrup is that you often have to go to the definition of one container type in a different chapter to work out how to use a particular function with another type.

The library section describes various C library functions properly as well which in "real life" you have to use as well as the C++ style of working. I found myself using it "in anger" in the course of this review for a program I was writing and found it really easy to find what I wanted.

I am sure C++ users will be glad of this book which I would very much recommend. You'll probably want to get the Stroustrup "bible" as well, but this is definitely clearer and easier to follow if you're in a hurry and you'll probably find an example of just what you need.

Definitely a good book, and very comprehensive for one describing itself as being "in a nutshell".

C++ Pocket Reference

Kyle Loudon

O'Reilly and Associates

ISBN 0-596-00496-6

138 pp.

£ 8.95

reviewed by John Collins

This is a pocket reference for C++ describing the "Syntax and Fundamentals". It gives a fairly comprehensive tour of the language syntax and semantics with some very brief examples.

Obviously in a book this size the author has to leave out a fair amount of detail and opinions will vary as to whether the details omitted are less important than those included. I rather felt it rushed too fast through operators and templates and laboured things like examples of literal constants.

The thing that really jumped out at me as missing was all but the most trivial description of the standard library which is an essential part of the C++ language.

I'm not convinced that anyone would be well advised to try to learn C++ syntax from a book like this and even the more experienced programmer who wanted to use a more esoteric feature of the language would want to go to a more expansive book that covered all aspects of the topic in question with lots of worked examples.

I think this book would be of limited use as it stands.

John Collins – Xi Software Ltd – www.xisl.com

eBay Hacks

David A Karp

O'Reilly and Associates

ISBN 0-596-00564-4

360 pp.

£ 17.50

reviewed by Mike Smith

A while ago I reviewed a couple books in the new O'Reilly "Hacks" series. You may want to refer back to them, of course, as they're great. Oh, okay, don't then - but the main point was that I really liked the format: Bite-sized pieces of information. So I was really looking forward to this one.

I'll admit now that I'm not a big eBay fan. I'm probably missing out, but I've never really liked bidding for stuff, or bartering at the market for that matter. I looked at listing something myself once, but didn't want to pay the charges either! However a couple of friends are regular users (one of them has bought two guitars from Canada, even) - so I will run a few tips past them to see what they think.

In this case I was expecting some cool programmatic features for monitoring new listings, or looking at what auctions are about to close, or similar. In actual fact, there is a much wider range of subject matter covered. I was surprised by the first section, which is all about the social aspects of using eBay. Social Engineering of a sort - how to behave because of the importance of the feedback system. Out of the eight chapters, fortunately only the first is dedicated to these non-technical aspects. However it was interesting and actually probably quite useful if you don't realise the importance of the feedback system.

The remaining sections are: Two chapters for Buyers (on searching and Bidding); Five for selling (covering a wide range of techniques - including things you can do with photographs, and running a business), and the final chapter covers the eBay API (we get there in the end!)

So lets walk through the book. The first, non-technical, chapter has 8 'hacks'. As I've mentioned they are actually quite interesting, despite my chastising earlier. For instance I didn't realise that interesting stuff like Feedback extortion goes on! Maybe I have been right not to get involved!

To move onto the chapter on Searching, an interesting technique is not searching for specific items, but using other bidders (who you've been bidding against previously) to locate items you might be interested in. Clever eh? Let them do the hard work. (This is actually also mentioned in the Foreword.)

We all know of the technique for stealing the bid with only seconds to spare of course, referred to as “sniping” – though “Camping” seems a more appropriate description to me:) However I didn’t realise there were services for doing this automatically for you - e.g. eSnipe (www.esnipe.com). There are even more sophisticated features for bidding on groups of auctions. That’s probably for the experts (or the addicts) though.

There is quite a bit of material on how to exploit eBay’s features to improve the quality of your listed item. You can use HTML - tables for instance, and even embedded sound (if that’s appropriate). You can link to your other sales, include dynamic text (just use an iframe, and you can put whatever you want in it), and most importantly add photos which will be just the thing to make a sale. There is therefore a whole chapter (9 hacks) on the subject of pictures. This covers the quality of the photograph, including the composition for instance, editing images to improve them, techniques for stopping other people from stealing them, using thumbnails, and a great little 3D interactive technique. Like it.

We next move into another interesting area - dealing with transactions. Not really technical though, so I’ll skip this.

Its hack number 74 (of 100) before I noticed the first perl script. This is in the chapter about running a business on eBay, and we’re starting to look at automated solutions. There are also some techniques for combining eBay and PayPal to create some powerful features.

Now, at the end, we really do start moving into the technical area. The API is XML based and examples on its use are provided in perl. But there’s a catch. To use the API in the production eBay environment (apparently there is a testing sandbox) you have to be certified. And to be certified you have to PAY! This is not a developer fee - its an application fee and covers the testing requirements for using your application with the API. So if you write a new app, its likely you’ll need to pay another fee. (If you change the way you use the API, you may also be required to pay a fee too.) So this isn’t the same as writing addons for Google, for instance. Disappointing.

In summary, therefore, not as much technology as I would like to have seen, and to use the API there’s a blummin’ fee involved, but the book is very interesting nonetheless. It might just give me the incentive to give eBay another go!

Optimizing Oracle Performance

Cary Millsap

O’Reilly and Associates

ISBN 0-596-00527-X

416 pp.

£ 24.95

reviewed by Mike Smith

There’s a little note across the top of the cover on this book – “A Practitioner’s Guide to Optimizing Response Time”. This is quite key point for me, as this guide specifically addresses one particular performance issue - that of user response time. You might think that this is the only performance issue, or at least the most important one. And you’d probably be right in the latter case, but there are other forms: For instance, I like to get the best out of a system in general - so although I might not get the best response time for a particular transaction, the flip side is that the system might be able to support twice the number of users, or something. It depends what value you apply to the response time (and actually with today’s Service Level Agreements, you can put a value on it) but with the cost of an Oracle license at £6,000 (or whatever) per CPU [and the proportional ongoing maintenance cost too], supporting the user estate with half as many CPUs can often take precedence.

In fact Sun have cottoned-on to this concept – they have been talking about “Throughput” computing for a while, and are developing multi-core CPUs (a bit like Intel’s Hyperthreading). The idea is to get more work done with a CPU, rather than just doing things faster. The primary reason for this approach in the CPU environment is the memory latency – but with the CPU multi-threading stuff and interleaving the memory accesses accordingly, you can get a lot more done in the same timeframe – but each particular piece of work isn’t going to get done any quicker (... and possibly a little slower, in fact). Anyway, back to tuning Oracle ...

I am sure I’ve said in a previous article (most likely another book review) that you can only get a low, 10% maybe, performance improvement at the DBA/System level; 90% of potential improvements are at the application/coding level. The author seems to be in agreement, looking at SQL code amongst other areas. Its worth remembering though that there’s typically a whole host of other technologies a transaction passes through; Not just the database: You need to think about the user interface, wide area network, application servers, LAN and the storage subsystem too. So a holistic view to performance is needed. Tools like Veritas *i*³ (formally from Precise Software) are quite cool for J2EE environments, for instance. And if you’re familiar with EMC’s DB Tuner, this is from the same stable.

The book has three major sections, plus some appendices. The author, Cary Millsap, calls these sections “Method”, “Reference” and “Deployment”. The first section is fairly brief, and provides an overview of where to look in terms of measuring performance. Note that this is not a cookbook of techniques though - its more about teaching you the concepts so that you can use the principles yourself to go further.

The second section goes into a lot of detail - specifics about analysing Oracle trace data, for instance, and many other areas (some theory and modelling too). However there’s a lot of heavy material here, and I haven’t got my head around it all for this review. The point of this section is the detailed analysis of what Oracle is doing. Its not just about EXPLAIN PLAN these days – the main tools are the Oracle wait interface, and the extended SQL trace facility. If you don’t know about these and you’re a DBA, its worth making time to investigate.

The final section is again quite short: Its about deducing what you can do to improve performance. So you might need to tweak SQL*Net parameters, or do the usual things with IO, multiple DB writers, relink in single task mode or whatever. The detail for this discussion doesn’t matter - and indeed there’s precious little technical detail here - its more about the process really. Each chapter also has a set of exercises at the end - maybe Millsap expects the text to support university courses.

So this book is about a methodology for performance analysis and the consequential tuning: How to approach the problem, where to look, and what to measure. Its quite advanced, some new theories (new to me anyway) and not at all what I expected. To reiterate, its not about the quick or easy parameter tweaks (these approaches don’t have any major impact on performance anyway), and its not a list of performance tuning techniques either - its about the approach. As long as you take on board these points and know what you’re getting into, its worth a read.

Millsap setup the System Performance Group in Oracle, and made VP - so knows his stuff. The book has been produced in conjunction with his new venture, www.hotosos.com which is an Oracle performance tuning company (he left Oracle a few years ago to set it up.) I hadn’t heard of them – head in the sand as usual. So I have a slightly uncomfortable feeling that the book is just a way of increasing their exposure and promoting their tools and services. Its worked of course, because I’m writing about it.

Oracle Data Dictionary Pocket Reference

David C Kreines

O'Reilly and Associates

ISBN 0-596-00517-2

144 pp.

£ 8.95

reviewed by Mike Smith

What is to say about a pocket reference? This one has 129 tiny pages, plus an index. Quite often I think a reference as surplus to requirements, as google is often quicker for finding that snippet of information you require. However I'm sure they have their place for flicking through to discover new things. Its always a challenge to know (i.e. be aware of) what you don't know.

This reference is for Oracle 9i Release 2 - so good for a few more weeks! The 10g database is being released in November or December, I think. But don't let that put you off - and to be fair, it was published in April apparently.

I started working with Oracle way back ... Version 6, and the data dictionary structure (which had a major revamp at that time) hasn't changed significantly since - though of course there have been continual enhancements release after release. The book has a brief overview of the structure, and explanation of nomenclature, then we're into the meat.

Where I think the book shines is the categorisation of the information. I mentioned the index. Its good to have a list of objects, but this still takes 10 pages (each with 60 or more items). The problem with an index, as always, is that its alphabetic and if you're looking for, say, that table which has some information on columns its a problem. (ie there are several tables, and they are dotted about all over the index). You'd be better off at the SQLPlus prompt with a

```
select table_name from dba_tables where table_name like '\%COL\%'

```

clause, or a show command or similar. Oh, and the V listing is a bit bloated (that's an in joke for you DBAs - v\$, you know.)

Anyway, I digress; Back to the point. The various views and what-have-you are put into categories, so the book isn't just an alphabetic list of data dictionary objects. There are sections on tables, indexes, jobs, security - and the newer features like replication and partitioning. Then there's OPS (yuk!) and RAC (yum!) too. Not all in that order actually - the sections themselves are placed in an alphabetic fashion to show no favouritism. (There are many more categories, I just haven't listed them.)

The book is split into two major sections - almost exactly down the middle. I've talked about the first half so far, except for my little (and very poor) joke above. The second half covers this area - the dynamic views. For those non-DBAers (though goodness knows why you're reading this, you brave souls) the dynamic views are the ones which commence with a 'v\$'. As if there isn't enough confusion over what v\$

views are (you may or may not know that most are actually views of the x\$ tables), they then go and introduce gv\$ views in Oracle 8 too.

So this latter half has page after page of v\$ views.

I usually only seem to do a `select * from v$database`

- to see where I am (... oh, okay sometimes I look at locks when there's something funny going on, or latches to pretend I know what I'm doing with performance tuning. Hold on, there's another review on that subject somewhere!)

After my dismissiveness at the beginning of this article, I actually quite like this little guide. Its only a listing of data dictionary objects and their structure. But its always when I'm half way through a complex select statement that the mind goes blank - "Is it extent_name

or `segment_name` in `dba_extents` ?” So now I have the option of picking up my Oracle Data Dictionary Pocket Reference and flicking through to page 55 to have a look. (Instead, that is, of the 50ms switch to another window to do a `desc dba_extents` and back again!) And I may notice on page 54 that the `dba_data_files` column for the auto extension flag is `autoextensible`, not `autoextendable`. That would be an easy mistake to make, I’m sure you’ll agree !

Contacts

Charles Curran
Council Chairman; Events; Newsletter
Oxford
Tel: 07973 231 870
Tel: +34 954 371 381
charles.curran@ukuug.org

James Youngman
UKUUG Treasurer
Manchester
james.youngman@ukuug.org

Sam Smith
Website
Manchester
sam.smith@ukuug.org

Alasdair Kergon
Events
Reading
alasdair.kergon@ukuug.org

Alain Williams
Watford
alain.williams@ukuug.org

Roger Whittaker
Schools; Newsletter
Hammersmith
roger.whittaker@ukuug.org

Ray Miller
Events; Newsletter
Oxford
Tel: 01865 273 200
ray.miller@ukuug.org

Jane Morrison
UKUUG Secretariat
PO Box 37
Buntingford
Herts
SG9 9UQ
Tel: 01763 273 475
Fax: 01763 273 255
office@ukuug.org