news@UK

The Newsletter of UKUUG, the UK's Unix and Open Systems Users Group
Published electronically at `http://www.ukuug.org/newsletter/`

Volume 16, Number 2 **ISSN 0965-9412** **June 2007**

Contents

News from the Secretariat

Jane Morrison

Since the end of March we have been concentrating on the organisation of the Linux 2007 event which this year is being organised as a collaboration between the UK Unix User Group and the German Unix User Group.

Newly named "LinuxConf Europe" the event will take place at the University Arms Hotel in Cambridge from Sunday 2nd September to Tuesday 4th September 2007, immediately preceding the invitation-only Kernel Summit (organised by USENIX) at the same venue.

LinuxConf Europe will consist of two or more streams of talks and tutorials alongside a small exhibition.

We are planning to have the provisional programme details and booking form available by mid-June.

Another event for your diaries is the UKUUG Annual General Meeting which will be held this year on Wednesday 26th September. Further details and the paperwork for the meeting will be sent to you automatically. If you are interested in joining UKUUG's Council, please let me know, or contact any member of Council for an informal discussion.

After consultation with members we are now planning a series of one day seminars. The first of these is provisionally planned for 16th October and the main topic will be databases.

We are planning to hold the next Winter/Spring Conference in March 2008: various venues are under consideration.

We are very pleased that we have been able to obtain approval for UKUUG by the Commissioners for HM Revenue and Customs under Section 344 of the Income Tax (Earnings and Pensions) Act 2003 with effect from 6th April 2006. This means that the Inland Revenue allows UKUUG members to claim their subscription amounts for tax allowance purposes. UKUUG's name will appear in the list of approved bodies, which is due to be updated later this year. Until the list is updated, it will be necessary to contact your local Tax Inspector quoting the Head Office reference SAPP/T1644/10/2007/JEM to obtain this deduction. Copies of the letters from the Inland Revenue can be found on the UKUUG web site at:
http://www.ukuug.org/membership/taxrelief.pdf

I am very pleased to advise that Novell are continuing with their Gold Sponsoring membership for a further year. This gives us the opportunity to keep delegate fees as low as possible at future events.

UKUUG has a 'consultative' email list: `advisory@ukuug.org`. If you feel you would like to be included on discussions on matters of policy etc. please let me know and I will add your name to the list.

The next issue Newsletter may be delayed until the end of September because of LinuxConf Europe – the copy date will be published on the web site in due course. Any interesting articles from members will be very welcome: send submissions to
newsletter@ukuug.org

News about Liaison

Sunil Das

A new initiative is to hold single day events on specific hot topics. The first conference on Tuesday 16 October in London will have talks about databases including their relevance to website development and design. Some speakers have been signed up already but we encourage the membership to help in identifying others and to attend the event. Please note the date in your planner.

There will be a speaker from Oracle at the Database conference in October. In addition, we are pleased that Oracle's Sergio Leunissen agreed to contribute an article to this edition of the Newsletter. Thanks to Peter Salus who continues his Letter from Toronto. In recent months we have been in email and telephone contact with our friends and colleagues within USENIX. We appreciate their permission to

reprint an article which first appeared in ;login and the permission of the authors from XenSource in Cambridge. Our apologies to Clive Darke whose name was misspelled when he contributed to the last Newsletter.

We wish to acknowledge Novell who have renewed their Gold Sponsor Membership. UKUUG and Novell will be pursuing ways of giving visibility to each other via our membership and community.

Please continue to help with the liaison activity by email or telephone. Initial contact can be made using **sunil.das@ukuug.org**

Announcement: European BSD Conference

The sixth European BSD Conference (EuroBSDCon 2007), will be held in Copenhagen, Denmark, between September 12th and 15th 2007.

Tutorials will be held on the two first days, followed by the conference and an optional tour to the original LEGOland in Billund on Sunday the 16th.

The conference features some 20 talks, 5 tutorials, a poster session and "Birds of a Feather" discussion groups. A session on the current status of the three major BSD distributions will also be held.

Don't miss the opportunity to listen to some of the finest developers sharing their long-time knowledge of UNIX and BSD.

Registration will open before the end of May.

Please find more information at our conference website:
http://www.eurobsdcon.org/

Plus ça change

> *On two occasions, I have been asked [by members of Parliament], "Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?" I am not able to rightly apprehend the kind of confusion of ideas that could provoke such a question.*
>
> *– Charles Babbage (1791-1871)*

Clients. Still the same, eh?

The hoarding of IP

Peter H Salus

Patents and copyrights were originally intended to provide for a just reward to creators and inventors.

But just what might be patentable and how patents can be employed have both changed tremendously over the past few decades. And I employ "few decades" most seriously. It was only within the past decade that software has been patentable in the US; and few other countries have followed suit.

Yet, as David Edgerton [*The Shock of the Old* (OUP, 2007), p. 200], points out, "*The rate of patenting has not changed much over time. US patents granted to US residents varied between around 30,000 per annum to 50,000 per annum between 1910 and 1990, despite population growth, and even more significant economic growth.*"

Edgerton goes on to point out that there has been a steady increase since the 1980s, rising to 80,000 per annum at the beginning of this century.

Some of this is attributable to the genuine growth of silicon-based companies. Much more is attributable to what I would call a "filings race".

For example, three years ago, countering a press release stating that IBM's Research Division had filed for several thousand patents in the previous year, Microsoft announced that they would file over 3000 patents in the coming one.

However, it's worth noting that while nearly all patents filed (in the US) over the past decade have been granted, a significant number have been "reconsidered" on challenge.

Among other things, patents have been challenged on the basis of "prior art". That is, there are descriptions (perhaps actual patents) which have been publicly accessible and which describe the process or object which has been patented. The work is thus not original and the patent is revoked.

In 2003 Dan Ravicher founded the Public Patent Foundation. It was in response to the growing concern by many technology professionals over the number of patents granted that were either too trivial to deserve legal protection, or duplicate existing or expired patents. The Foundation usually works by requesting the United States Patent Office to review patents that are suspected of being invalid for some reason, usually by prior art.
`http://www.pubpat.org/`

PUBPAT has filed a number of requests, but the most salient right now are:

- The Pfizer Lipitor patent
- The Columbia Cotransformation patent
- The Microsoft FAT patent
- Forgent Networks JPEG Related patent

Let me quote from the PUBPAT site where the second of these is concerned.

> *In February 2004, PUBPAT filed a request for reexamination of Columbia University's patent on cotransformation, a process for inserting foreign DNA into a host cell to produce certain proteins that is the basis for a wide range of pharmaceutical products, including Epogen for anemia, Activase for heart attacks and stroke, Avonex for multiple sclerosis and Recombinate for hemophilia. PUBPAT's request showed that the patent, issued in 2002, violates the restriction against multiple patenting because Columbia previously received three other patents for the same invention in the 80's and early 90's. The three previous patents expired in 2000; the new patent will not expire until 2019. The Patent Office granted PUBPAT's request in May 2004 and Columbia voluntarily waived any right to assert the patent in December 2004.*

In other words, Columbia University was attempting to increase its income by extending its patent beyond expiry. The University "waived" its right because it feared the bad publicity of attempting to increase its income by "raising the costs" of medications.

Ever since the *courratiers de change* began trading in agricultural debt in France, different bundles of different products have traded on European "bourses". The first of these seems to have been in Bruges in 1309. Tons of wheat; teams of oxen; barrels of wine or oil; cheeses; sheep; etc., were traded. In 1635 tulip bulbs created the first historical bubble. So we should not be surprised at companies buying and trading patents right now.

There's a lot of money that can be made from playing in a "hot" market. The record in 1635 was the equivalent of *over US $30,000* for a single bulb. You can purchase bulbs for under $1.00 via the Web.

But that won't slow speculators.

In 2000 Kevin G. Rivette and David Kline published *Rembrandts in the Attic* (Harvard Business School Press), which became a business best-seller, every would-be CEO or VC had to know just what a company's intellectual property (= IP) was worth. (I worked for a company that had two CEOs between 2000 and 2002 each of whom thought that this and a slim tome called *Marketing Outrageously* were all one needed for financial success. The company no longer exists.)

The television shows that feature experts travelling the world and declaring that Granny's sauce boat is a rare example of something just encourage fantasies: there *are* finds and hidden treasures. But there aren't many. I doubt whether the odds are significantly better than that of the local lottery.

As hope springs eternal, so firms have flourished that buy up filed and awarded patents from bankrupt or no-longer-extant firms and attempt to "leverage" them to profitability.

Thus we have companies like Patriot Scientific, with a mission statement:
`http://www.ptsc.com/`

> *Patriot Scientific's mission is to serve the world through scientific innovation. Patriot Scientific will apply its resources with focused agility and insightful due-diligence – using those resources to invest in discovering and licensing innovative technologies and solutions that create sustainable value to our investors.*

Patriot Scientific

> *"specializes in licensing innovative and proprietary technologies. The Company's intellectual property portfolio includes valuable patents that encompass fundamental microprocessor technology included in products manufactured and marketed by companies around the world."*

"Fundamental microprocessor technology." Hmmm.

In January of this year, PUBPAT filed a formal request that the US Patent and Trademark Office review

> *a patent held by Patriot Scientific (OB: PTSC) that the company, which boasts of "primarily focusing on deriving revenue from licensing patents", is widely asserting against producers of computer microprocessors. In its filing, PUBPAT submitted prior art that the Patent Office was not aware of when reviewing the application that led to the issuance of the patent, described in detail how the prior art invalidates the patent and asked that the patent be revoked. In April 2007, the Patent Office granted PUBPAT's request for reexamination of the patent.*

This is good business: Patriot Scientific has filed several lawsuits and sent over 150 letters threatening litigation. Not only Patriot Scientific but (literally) hundreds of tort lawyers are living on this extortion from an industry. And, ultimately, from passing on extra costs to everyone using a microprocessor – anywhere.

This is not (say) ISI or GM or Ericsson enforcing a patent: Patriot Scientific has no business other that buying up patents and extorting (ahem!) or litigating.

Months ago I cited both the Statute of Anne, 1709, and the US *Constitution*, Article 1, Section 8, 4 March 1789:

> *Congress shall have Power ... To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.*

Of greatest interest to me is that the copyright and patent protections were awarded to "Authors and Inventors", not to rapacious resellers and litigators.

As I noted last year (*news@UKUUG*, December 2006, page 11)

> *"on January 20, 2004, SCOG [The SCO Group] filed suit against Novell for 'making false statements' and 'slandering SCO's title' where SVR5 was concerned. (Novell had stated that they had never sold the title to Unix copyrights to [old] SCO.)"*

Discovery is now long past in the Novell suit. I am certain that I will receive remission of time in Purgatory for my sins, as I have read all the documents in the case that have not been sealed.

I think that it is obvious to anyone reading the APA (= Asset Purchase Agreement) of September 19, 1995, and the two Amendments to it (December 1995 and October 1996), that Novell never conveyed the Unix copyrights to [old] SCO nor did [old] SCO convey such copyrights to Caldera.

To be quite honest, the Unix copyrights are a mess. This has been true for about 25 years: programs and applications from all over the world were incorporated into the BSD versions; many parts of BSD were incorporated into AT&T V7 and after; etc. This was noted during the Novell v BSDI litigation. Even if USL had attempted to convey some version of "everything" to Novell, parts of that "everything" would not have been USL's to convey.

Novell could then have not conveyed "everything" to [old] SCO.

And Caldera/The SCO Group must have realized this.

The attempt at extorting "billions" from IBM has failed. And the parallel attempt at limiting Linux through FUD as failed as well.

If I thought that Dennis Ritchie or Bjarne Stroustrup or James Gosling or Larry Wall got something when I employ C or C++ or Java or Perl, I wouldn't really mind it (though I agree with Richard Stallman that "software should be free" – and so should languages). But the notion that Darl McBride and the SCO Group "own" and "collect royalties" on C++ is absurd.

And we now see that all the Group's suits have the substance of the Emperor's New Clothes.

Hardware virtualization with Xen

Steven Hand, Andrew Warfield, and Keir Fraser

Xen is a virtual machine monitor (VMM) that we've been developing at the University of Cambridge for the past several years. As a VMM, Xen allows a single physical computer to be divided up into a number of smaller virtual computers, each running its own operating system and applications. Xen is free, but is also available as part of a number of commercial offerings.

Xen was designed from day one to get every last ounce of performance out of commodity x86 machines. The past year has seen chip vendors such as Intel and AMD launch next-generation processors that provide hardware assistance for virtualization. In this article, we provide a background to Xen, show how these new hardware features can be used to provide high-performance virtualization even for proprietary or legacy operating systems, and look toward the future of hardware virtualization.

Xen: Virtualization for the masses

System virtualization technology has been around for over four decades. Pioneered by IBM with VM/370, system virtualization allows you to divide a single powerful computer into a number of smaller, less powerful computers called virtual machines. Each virtual machine runs its own operating system and applications and is strongly isolated from other virtual machines. This provides enhanced flexibility, management, and security.

For many years, virtualization was limited to “big iron” machines. However, the increasing power and prevalence of commodity off-the-shelf (COTS) systems have made virtualization an attractive technology for regular x86 boxes. Virtual machine monitors (VMMs) such as Xen and VMware provide system virtualization for COTS systems and are now in use on hundreds of thousands of machines worldwide.

There is, however, a problem: The Intel IA-32 architecture was not designed with virtualization in mind, and so certain instructions which should trap when executed with insufficient privilege simply behave differently, and various privileged states are visible even to user-mode software. This means that traditional system virtualization approaches are insufficient. Instead, new techniques are needed to make x86 VMMs a reality.

The problem with IA-32

In a classic 1974 paper, Popek and Goldberg describe the basic principles for system virtualization. In particular, they identify three requirements for something to be considered a VMM:

- Equivalence: Software running in a virtual machine should behave exactly as it would on a “real” machine (barring timing effects).
- Performance: The vast majority of machine instructions executed when running within a virtual machine should be executed “natively” on the real hardware, and without intervention from the VMM.
- Resource control: The VMM must be in complete control of the hardware resources.

These requirements typically lead to a “trap and emulate” approach in which the VMM runs hosted operating systems in user mode. Most of the time, the software runs exactly as it would on a real machine, but if the operating system (OS) attempts to perform a privileged operation, a hardware trap will occur. Since the VMM executes in supervisor mode, it can catch this hardware exception, inspect the state of the OS that caused it, and emulate the behaviour that would have occurred on real hardware. The VMM can then resume the virtual machine, allowing execution to continue.

This approach will satisfy Popek/Goldberg requirements as long as the processor is guaranteed to trap whenever any privileged operation is attempted in user mode. Unfortunately, the original IA-32 architecture does not guarantee this: Various instructions which should trap don’t, and in some cases they simply have different semantics than they would have on a real machine. In addition, certain kinds of privileged machine state (such as page tables and segment descriptor tables) reside in memory and hence are visible to user-mode software.

Solving the problem

There are two main ways that we can work around these problems with the IA-32 architecture: binary rewriting and paravirtualization. In the former approach, the VMM dynamically scans the memory of the guest OS looking for problematic instructions, and rewrites any it finds with alternative instruction sequences. This approach is costly and fragile, especially as the x86 uses variable-length instructions, but it can be made to work in most cases.

The latter approach, used by Xen, modifies the operating system source code to make it aware that it is running on top of a VMM. The resulting enlightened operating system can run extremely efficiently in a virtual machine environment: Typically an overhead of just 1% is observed. It can also work in cooperation with the VMM to provide advanced features such as CPU, memory, and device hotplug, or even live migration, seamlessly relocating a running virtual machine from one physical node to another.

At the time of writing, there are enlightened versions of modern Linux, BSD, and Solaris operating systems that run efficiently on Xen. Furthermore, Microsoft has announced that it is working on an enlightened version of its forthcoming “Longhorn” operating system.

Nonetheless, there is a large existing base of legacy operating systems that cannot use the paravirtualized technique. To support these, we need to look to the processor vendors and their recently introduced hardware support for virtualization.

Hardware Virtualization

To work around the problems with the original IA-32 architecture, both major processor vendors have

recently introduced hardware extensions. Intel's technology is called VT-x, or VT for short, and ships in most recent processors including the Xeon 51xx series, the Xeon 71xx series, and the Core Duo and Core 2 Duo processors. The equivalent AMD technology, called AMD-V, ships in recent (stepping F2) Opteron and AMD64 processors. (Although there are some important differences between VT-x and AMD-V, this article will avoid them in the interests of simplicity. More technical details on both technologies are available from the references given at the end of the article.)

These hardware virtualization (HV) technologies both operate by making the processor aware of multiple virtual machine contexts (VMCs). A VMC is analogous to a process control block (PCB) in an operating system: a copy of the state required to resume or schedule that virtual machine. The VMC holds a strict superset of the contents of a PCB, however; for example, in addition to the values of general purpose and floating-point registers and flags, the VMC will contain the values of the processor control registers (such as cr0, cr4, and cr8). The VMC will also include the values of certain model-specific registers (MSRs) such as CSTAR and EFER, as well as an expanded version of each segment selector.

Hence with hardware virtualization technology, the VMM acts somewhat like a traditional operating system, but scheduling virtual machines instead of processes. The HV extensions include instructions to launch and/or resume a given VMC, which causes the hardware to load the relevant processor state and continue execution. The new execution environment includes its own privilege levels ("rings" in IA-32 terminology), so the operating system kernel can operate in what it believes is supervisor mode and runs its own applications in what it believes is user mode. The new execution environment can also operate in a completely independent processor mode; for example, the VMM can run in 64-bit mode, one VM in 32-bit paged mode, and another VM in 16-bit real mode.

The act of launching and/or resuming a VMC is sometimes called entering a virtual machine and, as previously mentioned, can be seen as analogous to scheduling a process in an operating system. However, things are different when we consider the opposite case: exiting a virtual machine. Whereas in an operating system a process will usually only be descheduled as a result of an interrupt or system call, a VMM wishes to intercept execution in a much wider range of situations. Examples include instructions that manipulate processor interrupt state, interactions with the TLB, instructions that access or update control registers or MSRs, and attempts to put the processor into a halt state.

To allow maximum flexibility, hardware virtualization allows the VMM to select precisely which events it wants to intercept. The selected events will cause a vmexit, effectively a trap from the running virtual machine into the VMM. Since the set of allowable events includes all privileged x86 instructions, this allows implementation of the classic trap-and-emulate scheme, and hence it enables efficient virtualization of nonparavirtualized operating systems.

Xen: Hardware Virtual Machines

Xen uses the hardware virtualization technologies described above to enable support for legacy or proprietary operating systems. It uses the trap-and-emulate approach to deal with privileged instructions, which enables efficient virtualization without the overhead or fragility of binary rewriting. However, existing hardware virtualization support only provides part of the solution required to enable the execution of hardware virtual machines. In particular we can consider a modern COTS system as comprising three main components:

- The processor
- The memory subsystem
- The I/O subsystem

Current VT-x and AMD-V technologies help with processor virtualization, but they do not deal with memory or I/O. Software support within Xen is required to complete the picture.

Virtualizing Memory

Most operating systems expect a contiguous range of physical memory (RAM) starting from address 0x0. When running on top of a VMM, however, many operating systems are run concurrently, and will be allocated varying amounts of physical memory from the overall pool. One job for the VMM then is to translate between physical addresses as seen by an individual virtual machine ("guest physical addresses"

or just “physical addresses” for short) and the actual physical addresses as seen by the real hardware (“machine addresses”).

There are two interesting cases to consider depending on which mode the virtual machine is executing in: (1) real mode or protected mode or (2) paged mode. In the former case, addresses generated by the virtual machine are physical addresses (albeit modified by segment translation); in the latter case the virtual machine generates virtual addresses, which it expects to be translated via the processor’s paging mechanism. Xen handles both of these cases by the same means: shadow page tables.

The basic idea is simple: The guest creates and manages its own page tables, which translate from virtual to guest physical addresses. When the guest wishes to use an address space for the first time, it will update its cr3 register to point to the root page table. Using hardware virtualization, this causes a vmexit, which allows Xen to create a shadow copy of the root page table. Unlike the guest version, the version used by Xen translates from virtual addresses directly to machine addresses, and so it can be used by the “real” (hardware) MMU.

For space efficiency, it is not necessary to make shadow copies of every part of the current page table; instead, copies can be made on demand as the operating system (or its hosted processes) access various parts of the virtual address space. In addition, Xen must be able to track any updates made by the guest to its page tables and reflect the appropriate changes in the shadow copies. For these reasons, Xen ensures that guest page table pages are always mapped read-only. As a consequence, any page table modification attempted by the guest will result in a fault into the VMM. Xen can then intercept the access and maintain coherence between guest and shadow page tables.

The current implementation (in Xen 3.0.3) has been designed for high per- performance and includes a number of optimizations above and beyond the scheme just described. It also incorporates support for other modes of operation, which may be used for the live migration of virtual machines. Interested readers can learn more from the references given at the end of the article.

Virtualizing I/O

The final part of the picture entails dealing with the I/O subsystem. This includes simple platform devices (such as timers and interrupt controllers), disk drives, video cards, USB controllers, and network interface cards.

COTS systems expect to access such devices either via I/O instructions (direct or memory mapped) or via memory-mapped PCI bus addresses. As with page table updates, Xen intercepts any such accesses and emulates the behaviour of device hardware. For platform devices, this is relatively straightforward since they perform no actual I/O per se. Other devices are more complex and may require the ability to send packets on a real network interface card or read data from a real storage device.

Xen supports these I/O devices by instantiating a device model process for each virtual machine. This emulates the behaviour of the rest of the platform hardware, which can be configured to include the desired number and type of network interface cards, IDE controllers, graphics cards, and USB controllers. These virtual devices handle any accesses made by device drivers running in the virtual machine, mirroring the state transitions that would be made by an equivalent piece of hardware. They also interact with a – potentially virtualized – instance of that hardware: For example, a disk device can be represented as a sparse file.

Providing an emulated platform allows operating systems to run without requiring that they are at all aware of virtualization. However, emulation can be rather slow, particularly for devices such as network interface cards, which can require many (emulated) bus cycles to, for example, transmit a packet.

Hence Xen also provides the ability to load new, virtualization-aware device drivers into the operating system after it has been installed. These paravirtualized drivers understand the underlying VMM, and hence they can more directly interact with the virtual hardware. In the case of networking, this can increase performance by an order of magnitude.

Next Steps in Hardware Virtualization

We’ve seen how Xen uses existing hardware virtualization of the processor to efficiently and robustly run unmodified operating systems, augmenting this with software support for memory virtualization (shadow

page tables) and I/O virtualization (device model).

Looking ahead, we envision a number of further hardware enhancements: hardware support for memory virtualization, platform virtualization, and device virtualization.

Nested/Extended Page Tables

Even though shadow page tables can be implemented efficiently, they still require a number of transitions between the VMM and the guest. These transitions can cost hundreds or even thousands of cycles, and so it is desirable to keep their number to an absolute minimum. To this end, both Intel and AMD have recently announced hardware support for virtualizing the MMU. Intel's scheme is called extended page tables (EPT); AMD's is called nested page tables (NPT).

Both operate by adding an extra level of translation; in essence, a new page table (called the EPT or NPT, respectively) is introduced to translate between (guest) physical and machine addresses. There is one of these per virtual machine, since all address spaces within that virtual machine share the same physical to machine mapping. In addition, the hardware is now explicitly aware of the guest page tables.

Consequently, on a TLB miss, the hardware can walk the guest page tables directly, using the additional EPT or NPT to translate the physical addresses contained within page table entries. On completion of the walk, the TLB is updated with the resulting virtual to machine mapping and execution continues.

Note that even though this extra level of indirection does involve more lookups, it does not require any vmexits, hence improving overall efficiency. The hardware can also cache intermediate translation results to further improve performance.

Virtualizing the Platform

Help is also coming for platform virtualization. First in line are extensions from AMD and Intel that allow enhanced protection from DMA-capable devices.

In today's COTS systems, devices are not subject to any translation or protection checks when they access memory. This means that a malicious or buggy device driver can program a device to read or write any piece of memory in the system, bypassing the VMM and any installed security policy.

Currently shipping AMD-V chips include support for device exclusion vectors (DEVs), which addresses this risk. A DEV is a bitmap with 1 bit for every 4K page of physical (host) memory. Any attempted memory access by a device first causes a lookup (based on the device and bus ids) to a protection domain; this is then used to select an appropriate DEV, and the target address is checked against the appropriate bit. If the bit is set, the access is disallowed. This can be used by a VMM to protect itself and any other key data (such as security policies) from rogue DMA accesses.

Similarly, Intel has announced VT-d, a forthcoming technology aimed at providing enhanced support for platform virtualization. VT-d is also a northbridge-based approach that interposes on device accesses, and it also maps devices to protection domains. However, VT-d takes a more generalized IOMMU approach: In particular, device-issued DMA addresses are no longer "physical" addresses but are instead translated through a hardware table. This allows protection as well as arbitrary remapping of the bus address space. VT-d support is expected to ship in 2007.

Virtualizing Devices

Finally, there is work on making I/O devices themselves virtualization-aware, to allow direct yet safe sharing between multiple virtual machines. This is particularly of interest for high-throughput, low-latency devices such as gigabit network interface cards and next-generation graphics cards.

Some of this work involves proposed extensions to PCIe being developed by the PCI-SIG. These extensions include address translation and the introduction of virtual functions within PCI devices. There is also ongoing development of "smart" I/O devices that provide translation, protection, and multiplexing between multiple clients. Early results indicate that bare- metal performance can be maintained without sacrificing safety.

Conclusion

As virtualization continues to grow as an important technique for managing modern systems, the software

and hardware used to provide it are maturing at a dramatic rate: The original version of Xen stemmed from a research project at the University of Cambridge and allowed a specific handful of modified operating systems to be efficiently virtualized on uncooperative x86 hardware. Nearly four years later, Xen is a mature and robust VMM supporting paravirtualized OSes that are increasingly maintained by the OS developers themselves; Xen has further been incorporated as a core feature in the major Linux distributions, being directly included with their release kernels.

Chip makers have also embraced virtualization and have released hardware features to assist VMMs. Xen now includes support for both Intel's VT and AMD's V processor extensions, allowing unmodified legacy OSes to be efficiently and safely virtualized. As a result, Xen can now host nonparavirtualized OSes, such as Microsoft Windows, on modern hardware. Hardware will continue to evolve in support of virtualization in the immediate future, providing more direct support for both memory and I/O devices. We look forward to incorporating these features into Xen as they become available, as they promise to provide even greater performance and stability for the virtualization of COTS systems.

References

The following resources are useful for finding out more about hardware virtualization and Xen:

"Intel Virtualization Technology", *Intel Technology Journal*:
http://www.intel.com/technology/itj/2006/v10i3/index.htm

AMD64 Architecture Programmers Manual, Volume 2: System Programming:
http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/24593.pdf

Xen downloads:
http://xensource.com/download

Symposium on Operating System Principles (SOSP) 2003 paper on Xen:
http://www.cl.cam.ac.uk/netos/papers/2003-xensosp.pdf

Shadow2 presentation at Fall 2006 Xen Summit:
http://www.xensource.com/files/summit_3/XenSummit_Shadow2.pdf

Xen Source Code Repository:
http://xenbits.xensource.com

This article was originally published in ;login: The Magazine of USENIX, *32, no. 1 (Berkeley, CA: USENIX Association, February 2007): 21-27. We are very grateful for the permission to reprint it here.*

Unbreakable Linux Support

Sergio Leunissen

On October 25th, 2006, during his keynote presentation at Oracle OpenWorld in San Francsico, Larry Ellison announced that Oracle was entering into the Linux operating system support business. Oracle's CEO explained that Linux was an increasingly important operating system for Oracle and its customers but that existing enterprise Linux support offerings didn't live up to the expectations of many Oracle customers. Besides, he said, existing support offerings were too costly. And so, Oracle decided to launch the Unbreakable Linux support program, offering full support for the Linux operating system.

In this article I briefly describe the Unbreakable Linux support program, its goals, and the motivations behind it.

The Program

Oracle Unbreakable Linux is a support program that provides enterprises with world-class, global support for Linux. Recognising the demand for true enterprise-quality Linux support and seeing an opportunity to significantly reduce IT infrastructure costs, Oracle offers Linux operating system support. Oracle is committed to delivering high quality, comprehensive, and integrated support solutions to help ensure enterprise success with the Linux operating system.

The program is designed for customers who run Linux on servers in data centre deployments for enterprise workloads.

For those customers who do run Oracle, the benefit is clear: support for the full stack from the operating system up. When issues arise and you need to call support, there is no finger pointing. “One back to pat”, if you like.

A unique feature of Oracle Unbreakable Linux support’s highest level, called Premier support, is that it allows for individual bug fixes to be backported, on request, to any version of any package released within the previous six months. For users of the traditional Unixes, this is nothing new. However Linux customers have, until the release of Unbreakable Linux support, not had access to these types of backports. Sure, Linux vendors will backport critical fixes to certain packages in previous update levels, but if you happen to be running an older version of that package, you’ll have to upgrade to the newer version and risk introducing many other changes included into your production environment.

Customers of Unbreakable Linux support also enjoy full indemnification – legal protection against claims of intellectual property infringement.

In Oracle’s view, Linux is an operating system that’s ready for production use in mission critical settings. However, lack of backports and lack of adequate indemnification were two key shortcomings in existing Linux support offerings holding back many customers from adopting Linux.

How It Works

There are two ways to make use of the Oracle Unbreakable Linux program. If you already have Red Hat Enterprise Linux installed and you a purchase Unbreakable Linux support, you can simply download an up2date agent from Oracle and connect your server to Unbreakable Linux Network. With this switch complete, you contact Oracle for support rather than Red Hat and your server gets its updates from Unbreakable Linux Network (ULN) rather than Red Hat Network.

If you do not already have Red Hat Enterprise Linux installed, you can download and install Oracle Enterprise Linux for free. Oracle Enterprise Linux is fully binary and source compatible with Red Hat Enterprise Linux as it includes the exact same set of packages at the same version levels with the same source code.

Maintaining Compatibility

You may have noticed that I haven’t talked about how our Linux is better than anyone else’s Linux. Oracle has not forked and has no desire to fork Red Hat Enterprise Linux and maintain its own version. We don’t differentiate on the distribution because we use source code provided by Red Hat to produce Oracle Enterprise Linux and errata. We don’t care whether you run Red Hat Enterprise Linux or Enterprise Linux from Oracle and we’ll support you in either case because the two are fully binary and source compatible. Instead, we focus on the nature and the quality of our support and the way we test Linux using real world test cases and workloads. So, how do we maintain compatibility?

Oracle synchronizes bug fixes at regular intervals with Red Hat Enterprise Linux (RHEL) to maintain full compatibility. Whenever a new version of an individual package (an errata) gets released by Red Hat, the corresponding package for Oracle Enterprise Linux is made available very quickly, in a matter of hours. If a package has no trademarks and no Oracle specific patches, it will simply be recompiled and reissued for Oracle Enterprise Linux immediately after going through testing.

If a package has trademarks or Oracle Enterprise Linux specific changes, Oracle will examine the source code and compare it against the bug fixes that have been already applied and released as part of Oracle Enterprise Linux. If the Oracle patches are still relevant, then they are reapplied, but if the problems have been fixed in the Red Hat version, whether in the same or in a different way, the Oracle specific patches are dropped and the package recompiled (always checking for trademarks and copyrights issues) and released as part of Oracle Enterprise Linux via the Unbreakable Linux Network (ULN).

For official updates of existing major releases, for instance Red Hat Enterprise Linux 4 Update 5, Oracle has re-bundled the Red Hat patches in the update and reissued them as Oracle Enterprise Linux 4 Update 5, including free installable software, almost immediately. As a new major RHEL release is issued, there will need to be some additional testing before Oracle can consider it an official Oracle Enterprise

Linux version. So, for instance, when Red Hat Enterprise Linux 5 was released, Oracle began testing the corresponding Oracle Enterprise Linux product before issuing a version of it, since in the past, it has taken many months for a new major release to stabilize.

Another aspect of compatibility with Red Hat Enterprise Linux is around patches and bug fixes written by Oracle and not appearing in the Red Hat distribution. Oracle has independently been providing bug fixes to customers for problems occurring in Red Hat Enterprise Linux for the last five years, and has maintained full compatibility by ensuring that the resolution of the problems does not break application compatibility.

As an example, the kernel ABI is a very important interface to keep stable, and Oracle has been requiring that none of the patches provided to customers or partners break that stability, by testing the variables and checksums so that the resulting code is fully compliant with that restriction. There are very few modifications to the existing Oracle Enterprise Linux code base, only critical bugfixes and no new features. Patches for any of the bugs that are fixed in Oracle Enterprise Linux will also be contributed back to the Linux community including Red Hat, Novell, and other distributions.

There is just one edition (and just one CD set) for Oracle Enterprise Linux, and it's available for all three levels of support. This corresponds to the Red Hat Enterprise Linux Advanced Server edition. It contains all the packages in Red Hat's high-end offering without limiting the set of packages that are available to customers under support contracts or for free downloads. A new set of installable software will be made available for free download for every new update of Oracle Enterprise Linux. Therefore, users of Oracle Enterprise Linux will be at most one update behind. This is in contrast with the Red Hat Enterprise Linux offering, where just the original initial release is available as a set of installable software while updates are delivered via RHN only to paying subscribers.

History of Linux at Oracle

Oracle has been investing in Linux for nearly a decade, starting in 1998 when we released the first commercial relational database on Linux. The following year Oracle made investments in Linux startups such as Red Hat, VA Linux, and Miracle Linux. Very soon after, Oracle began building a team of Linux kernel developers. These developers worked only on Linux operating system code, following the open source model and contributing their output back to Linux community kernel maintainers and to the distribution vendors. Their work centred mostly around improving asynchronous I/O, large memory support, and building a clustered file system to simplify deployments with Oracle Real Application Clusters (RAC).

As more customers began using Linux in production settings, Oracle began providing fixes to critical bugs directly to the customers in 2002 working with the distribution vendors. This practice became the seed that eventually grew into today's Oracle Unbreakable Linux support program.

While Oracle's customers began adopting Linux, Oracle itself overhauled its IT infrastructure to standardize on commodity x86 and x86-64 hardware based on Linux. As a result, all base development at Oracle is now done on Linux. Also, Oracle's Quality Assurance (QA) farms run Linux, as do more than10,000 servers in our Austin, Texas data centre which support our Global IT needs and our OnDemand business.

Active Community Contributors

Linux distributions are largely a community effort. Sure, vendors like Novell and Red Hat put effort into branding, packaging and installers, but the components that make up a Linux distribution, that is, the kernel itself and the hundreds of packages around it, are produced by the community. Oracle influences Linux by being a trusted community member, as a Linux Foundation board member, an Open Invention Network (OIN) licensee, and perhaps most importantly a contributor to the Linux mainline kernel.

Oracle has had a Linux kernel development team since 2001. The team has grown steadily and now includes well-known community developers such as Jens Axboe, Zach Brown, Olaf Kirch, and Chris Mason. The work that the kernel team is doing is primarily contributed upstream, in the mainline kernel and focuses on improving Linux for server workloads. Some key projects that the team is working on include:

- Rewrite asynch IO (aio) layer
- Cleaning up direct IO layer

- Oracle Clustered File System, OCFS2
- Mainline kernel testing
- Incorporating useful patches from various distributions (e.g. Ubuntu) into mainline
- Xen I/O and performance enhancements

Real World Testing

Customers often look to Oracle as a trusted advisor. When it comes to deploying Linux-based solutions, this is no different. Given the large number of choices available in servers, storage and networking equipment that work with Linux (especially as compared to, say, Sun Solaris or IBM AIX), customers want guidance in selecting a configuration that is known to work well. This is why Oracle launched the Oracle Validated Configuration program.

In the Oracle Validated Configuration program, Oracle works with its hardware partners, such as Sun, Hewlett Packard, Dell, IBM, and EMC to test configurations under real world conditions. Oracle maintains a comprehensive test kit that includes simple installation and pre- requisite tests as well as functional, stress and destructive tests on an Oracle database running in single node and in a clustered configuration. The goal of these tests is to recreate real world situations to see how the entire stack, from storage hardware all the way up to the database holds up. The output of this process is a bill of materials, the bits that make up the configuration, such as:

- product versions
- Linux OS version
- server hardware details
- networking component details
- storage hardware details
- Host Bus Adapter (HBA) details

In addition to a list of components tested, the Validated Configuration includes documentation on what software versions and settings were used, what the best practices are and what issues were encountered during setup and testing.

In running these comprehensive tests, we have uncovered bugs in the Red Hat Enterprise Linux distribution and have included fixes in Oracle Enterprise Linux and via Unbreakable Linux Network (and provided the fixes back to the Linux community). Ultimately, Oracle Validated Configurations give customers a higher level of comfort with the configuration of their choice and it helps them get the configuration up and running faster.

Conclusion

I hope I've made clear that Oracle Unbreakable Linux program is not an attempt to create an "Oracle Linux". Rather, it's an extension of Oracle's long-standing efforts to make Linux better by being an active community member. By ensuring higher quality code through extensive testing and by providing the type of support that customers demand, Oracle is committed to ensuring the success of Linux in the enterprise.

References

Oracle Unbreakable Linux Program:
`http://oracle.com/linux`

Oracle Unbreakable Linux Overview white paper:
`http://www.oracle.com/technologies/linux/ubl-overview.pdf`

Oracle Validated Configurations:
`http://www.oracle.com/technology/tech/linux/validated-configurations/index.html`

Introduction to Neogeography

Andrew Turner

O'Reilly Media (Short Cut)
ISBN 0-596-52995-3
54pp.
$ 7.99
Published: 15th December 2006

reviewed by Mike Smith

This is an O'Reilly "Short Cut", which is a new type of book from O'Reilly (new to me, anyway). Short Cuts take the form of a white paper or essay on a subject – this one is 54 pages long.

Whilst this review was conducted using a paper copy the Short Cuts are generally available for purchase and download as a pdf from the O'Reilly website. This one is about eight dollars. So with the exchange rate as it is at the moment, it won't break the bank.

So onto the review itself. Neogeography (which apparently means "New Geography") covers the subject and tools of mapping and GIS (Geographic Information Systems) that are increasingly being used by the end user on the web to enable them to create their own maps and map-based systems. Google maps and mash ups spring to mind. This is in contrast to some of the more traditional GISs such as products available from the likes of ESRI that I have come across in a few places.

This Short Cut provides an overview of neogeography tools, frameworks and information resources. There is also a companion website on which some examples discussed in the Short Cut are provided.

As you will appreciate given the length of the Short Cut, the information is succinct, getting straight to the point. It's really good for a quick introduction to the subject; what's going on at the moment; tools that are available; some information on APIs and examples.

Just a rapid summary of some of the topics: GPX, GeoRSS, KML, Easygps, Google Earth, Geocoding, Geolocation, Platial, Ning, Mapstraction, OpenLayers, OpenStreetMap, GeoStack, Picasa, Flickr.

From this example, I think these Short Cuts are great ... it's a cheap way of learning about a new subject. However as these are only a soft copy I'm not sure how popular you'll find them as you may be able to find out what you need with a Google search. It's not like owning a physical book, which is part of the O'Reilly attraction for me.

Everyday Scripting with Ruby: For Teams, Testers and You

Brian Marick

Pragmatic Bookshelf
ISBN 0-977-61661-4
310pp.
£ 20.99
Published: 30th January 2007

reviewed by Lindsay Marshall

This is a book in the Pragmatic Programmers series, and, as I vaguely recall saying in another, I have to own up to not being fond of the books in it so far. They have been decidedly underwhelming and not really up to the standards of other O'Reilly books. IMHO: the intertubes are full of rave reviews of the series so I seem to be a bit out on a limb on this one.

The current book does seem to be a little better than others though - lots of good clear, useful examples and even getting into exception handling and more advanced OO programming. But there is still much that I don't find attractive. I find the tone of the book decidedly condescending in places and I wonder who is the intended audience. Experienced programmers will find much of the material redundant, so it must be for keen novices who want to get on the Ruby bandwagon (it's not a juggernaut just yet). There

are exercises at the end of the chapters (something for which I have a deep-seated and entirely irrational dislike) so it may be intended for teaching but it doesn't seem geared quite that way either. There are sort of UML diagrams for some of the code examples which seems to make it a bit more technical. I am at a loss to tell who it's for.

The real weakness of the book though is its presentation. The paper feels coarser than usual and the pages just don't seem clear. In particular example outputs are presented in a typeface that is just seems to be just too small to be comfortably readable, even wearing my glasses. (There is no colophon in this series so I can't tell you what face it is either). Perhaps it is a plot to make Ruby inaccessible to older people by making the books hard to read for aging eyes.

So to summarise: sound technical content, some nice examples, but not nice to read or look at. If you are a member of the target audience then it may well be great, but I have no idea who you are.

Beyond Schemas: Planning Your XML Model

Jennifer Linton
O'Reilly Media (Short Cut)
ISBN 0-596-52770-5
41pp.
$ 9.99
Published: February 12th 2007

reviewed by Lindsay Marshall

OK, this is a weird one. *[No, it's not, it's an O'Reilly "Short Cut" (see Mike Smith's review above) – ed.]* I didn't even realise it was for review: I thought it was some kind of marketing material or something. It was in a cardboard O'Reilly folder and printed single-sided on 41 sheets of A4 paper. It looks for all the world like a bit of student coursework. If students knew anything about layout and could manage to stretch to more than 5 or 6 pages.

As you will no doubt have guessed I was immediately put off by the use of the word "schemas". The plural of schema is schemata. The writing style is a bit undergraduate too, but wait! The author understands about the different categories of users that you can encounter. Things are looking better. But then, one after the other, there are four small pictures of spreadsheets which, though in reality not the same, look pretty much the same apart from the caption. These spreadsheets are supposed to help you set up an information model, but it is not entirely clear to me how.

Suddenly we are doing formal things with documents and even little flurries of XML-ish things. I think I know what is supposed to be going on but I am not confident. I've backtracked a bit because I did think this was going to help me with the perennial problem in XML: what goes in as tags and what goes in as attributes; but reading through again there is no magic bullet just an assertion that the metadata will have been found during my user study.

Now the author is developing some persona and some scenarios. Always an excellent idea but I am less than convinced by the way that she extracts metadata from them. And then suddenly it is about naming schemes and we're at the end.

In reality, I think that there is a really good example of how to create XML representations from existing material buried somewhere in these few pages, but it seems quite well hidden. It all seems much to condensed and there are lots of assumptions and things that are skimmed over. They want 10 bucks for this, and it feels like one of those PDFs that you download from the net that doesn't quite meet your needs.

I'll have to think about this one – I really don't know what to say!

Contributors

Sunil Das is a freelance consultant providing Unix and Linux training via various computing companies.

Keir Fraser is an EPSRC academic fellow and lecturer at the University of Cambridge and a founder of XenSource. He completed his Ph.D. in 2004 and now manages the Xen project.

Sergio Leunissen is Senior Director of Linux Business Solutions in the Linux Engineering division at Oracle.

Steven Hand is a Senior Lecturer at the University of Cambridge and a founder of XenSource, the leading open source virtualization company. His interests span the areas of operating systems, networks, and security.

Lindsay Marshall developed the Newcastle Connection distributed UNIX software and created the first Internet cemetery. He is a Senior Lecturer in the School of Computing Science at the University of Newcastle upon Tyne. He also runs the RISKS digest website and the Bifurcated Rivets weblog.

Jane Morrison is Company Secretary and Administrator for UKUUG, and manages the UKUUG office at the Manor House in Buntingford. She has been involved with UKUUG administration since 1987. In addition to UKUUG, Jane is Company Secretary for a trade association (Fibreoptic Industry Association) that she also runs from the Manor House office.

Peter H Salus has been (inter alia) the Executive Director of the USENIX Association and Vice President of the Free Software Foundation. He is the author of “A Quarter Century of Unix” (1994) and other books.

Mike Smith works in the Chief Technology Office of a major European listed outsourcing company, setting technical strategy and working with hardware and software vendors to bring innovative solutions to its clients. He has over 15 years in the industry, including mid-range technical support roles and has experience with AIX, Dynix/ptx, HP-UX, Irix, Reliant UNIX, Solaris and of course Linux.

Andrew Warfield completed his Ph.D. at the University of Cambridge in May 2006. He now works as the lead storage architect for XenSource, and is also an Adjunct Professor in the Computer Science Department at the University of British Columbia. Andrew currently lives in Vancouver, Canada.

Contacts

Alain Williams
Council Chairman
Watford
Tel: 07876 680256

Sam Smith
UKUUG Treasurer; Website
Manchester

Mike Banahan
Council member
Ely

John M Collins
Council member
Welwyn Garden City

Phil Hands
Council member
London

John Pinner
Council member
Sutton Coldfield

Howard Thomson
Council member
Ashford, Middlesex

Jane Morrison
UKUUG Secretariat
PO Box 37
Buntingford
Herts
SG9 9UQ
Tel: 01763 273475
Fax: 01763 273255
office@ukuug.org

Sunil Das
UKUUG Liaison Officer
Suffolk

Leslie Fletcher
UKUUG Spokesperson
Manchester

Roger Whittaker
Newsletter Editor
London