Mid-term IT Server trends – A Geek's Eye View

Greg Baker – gregb@ifost.org.au

December 9, 2001

Contents

1 Introduction **1**

1.1 What won't matter 1

2 Virtualisation **3**

2.1 Where we are today 3

2.2 Why is virtualisation so important? 3

2.3 Clusters, Load Balancing and High Availability 4

2.4 Putting it all together 4

2.5 Summary 5

3 Componentisation **5**

3.1 Appliances 5

3.2 Internet Storage 5

3.3 Putting it together 7

3.4 Appliance Developers 7

3.5 Who would? 8

4 Free versus Proprietary Software **8**

4.1 Polarisation 8

4.2 A word on Microsoft 9

4.3 What will survive? 10

4.4 Implications 11

4.5 What is there to do? 11

5 Summary **12**

1 Introduction

The purpose of this document is to explore what are the interesting new technologies that will shape server technologies over the next five years. It does not attempt to address changes in desktop systems, handhelds or mobile infrastructure.

Mostly this document is the fault of Greg Baker of the Institute for Open Systems Technologies, although he has sought input from several other people, who remain nameless for fear of being associated with him.

1.1 What won't matter

As always, there are many trends going on in the IT industry and more buzzwords being floated now than ever before – hype sells. Here is a short list of trends that are probably irrelevant to the near future of server technologies:

Wireless and 3G 802.11b is offering excellent bandwidth and will change the way we layout office spaces. Likewise, a pervasive city-wide moderate-speed network will change the way we handle mobile users. But servers do not move around

much, and wired networks will always have better bandwidth and reliability than non-wired ones. Servers will still end up being racked (or similar) and will be wired (possibly with fibre-optics rather than ethernet) into the network as they are now for the forseeable future.

ASPs and application leasing At least in Australia, bandwidth is so expensive that any cost benefit from using a third party application service provider is utterly swamped by the cost of the bandwidth to do it. For example, even to run a single Citrix session for one user at 9.6kb/s during business hours across the wider internet (where download bandwidth costs 19c/Mb) would cost over $11,000 annually. The bandwidth cost is slowly dropping, but it would have to drop by at least an order of magnitude for it to be economic for most organisations. Leasing applications while running them in-house is essentially the same business model as charging annual licensing fees (as many high-end applications already do) and only offers some cash-flow benefits rather than any noticeable revolution in technology infrastructure.

Security problems Currently almost all applications that are released need post-install patching to fix security problems and other bugs. There is no sign of this situation improving, but other than causing a lot of work and pain for system administrators the overall equation stays the same – it is still possible to have a hacker-free environment, but a larger portion of the IT budget has to be allocated to do this. Of course, many companies will fail to act upon the threat – or rely on unreliable vendors – and suffer loss, but it is merely an organisational learning process to cope with it.

Single sign-on It still seems unlikely to eventuate – there are so many competing technologies (Kerberos+LDAP/ActiveDirectory, SSH keys, SSL certificates, SmartCard, etc.) that few organisations will be able to deploy it. Secondly, it poses a bit of a risk – if someone walks away from their desk for a moment, an eavesdropper can do anything they want as that user – which may slow corporate adoption of it. Finally, any solution that does present itself will have to integrate in backwardly-compatibly with vast amounts of existing infrastructure; it is not something that it would be possible to plan for yet.

Migration to IPv6 The internet crunch is significant, urgent and unavoidable. It may sound hopeless that after nearly ten years, IPv6 penetration has been so minimal, but it will replace the existing internet simply because there is no way for the internet to continue to expand. It is only recently that Cisco have started making IPv6 a high priority, and until most vendors are working on it, deployment cannot begin. The next five years will see nearly every vendor of internet infrastructure support IPv6, leaving the way open for companies to use it more effectively. Understanding the new internet protocol and redesigning infrastructure to use it will take a lot of IT staff time and effort. But ultimately, it does not make signficant changes to that way we would structure a network or deploy servers, it just optimises, simplifies and improves it.

This document does not attempt to cover changes in how desktop systems will be. For example, the current slow rise of Linux-based

desktops will probably make thin-client systems connecting to servers to run their sessions remotely a standard practice in most organisations. While this will change where applications run, and increase the load on servers, it does not in itself cause much of a change in the way servers are managed, run or deployed.

2 Virtualisation

2.1 Where we are today

The most significant changes will come in the way we relate operating systems to hardware. At the moment, the general trend is that each individual computer runs a copy of an operating system.

For several years now IBM zSeries systems have had supported running several instances of an operating system on a single system. This has mostly been used to run isolated MVS domains, or the occasional AIX system. It was only around two years ago that it became possible to do this with Linux, which suddenly launched greater interest in doing such things.

At the end of 2001, we have several options for doing this now:

- On Linux and MS-Windows, VMWare (a commercial product) can run multiple copies of almost any x86-based operating system, with a moderate performance penalty. Resource requirements are high – for example, there is no scope for sharing disk space between multiple identical copies of the same operating system.

- On Linux, FreeBSD and NetBSD, Plex86 (a free product) does much the same thing as VMWare (again, still constrained to the x86 architecture).

- There is a port of the 2.4 Linux kernel to run as a process under Linux itself. *User-mode Linux* then lets you run many copies of Linux on the same hardware. There is only a very slight performance penalty for running inside a Linux process rather than a Linux operating system. Disk and memory requirements are very light – a cleverly-planned installation would allow 10 instances of the operating system to appear to have independant disk images, but perhaps only use up (total) twice (or less) the amount of disk space that any individual operating system requires.

- There are various efforts underway to make use of the PowerPC chip's capabilities of doing in hardware what VMWare and Plex86 do in software.

- Hewlett-Packard and Sun have made some steps towards running multiple independant operating systems on the one system. Typically this is done in hardware, with a fixed number of CPUs being given to each operating system instance. This gives the least performance impact, but at the expense of flexibility. For example, it is not possible to run 33 different operating system images at the same time on a 32 processor HP Superdome.

2.2 Why is virtualisation so important?

Most systems today have under-utilised CPUs. It is a rare computer in which the CPU does not spend at least 50% of its time idling, waiting for disk activity, network activity, or most likely, waiting for a user to do something. So, on the one hand, there is scope for running at least two independant operating systems on almost any computer, and usually scope for far more.

On the other hand, reducing the amount of computer hardware in a company is a worthwhile goal:

- Fewer pieces of hardware mean that there are fewer things to go wrong. Even if the impact of a failure is greater, less resources are needed to maintain the infrastructure.
- The costs of acquisition, floor space, power and cabling can be reduced.
- There are dis-economies of scale in large computer networks, as the need for communication between administrators scales exponentially with the size of the network.

2.3 Clusters, Load Balancing and High Availability

In sections 2.1 and 2.2, the virtualisation discussed is about the relatively obscure practice of running several operating systems on one piece of hardware.

For several years now many large organisations needing high availability have done something that is almost the opposite. The standard way of clustering is to have a process (or several processes) monitored – if something goes wrong, it gets restarted on another computer. In a sense, the idea is to have an "operating system-like environment" spread across several computers.

Where virtualisation focuses on one-system-many-OS, clusters try to do many-systems-one-OS. Digital (now Compaq) have probably pushed this further than any other vendor, with their global filesystem. The one disk image is shared (live) between many computers. (For comparison, HP, Veritas, IBM and CA have clustering products in which the disk images are *available* to many computers, but only live on one at a time.)

The MOSIX project takes this further (URL...?). In a MOSIX cluster, if processes are making heavy CPU demands on one system while another is idle, some of the processes will automatically be migrated so that the load is as evenly balanced as possible across all systems in the group.

2.4 Putting it all together

Compaq are funding (with some assistance from VA Linux) a project to bring virtualisation (with user-mode Linux), clustering (with Sistena GFS), load-balancing (with MOSIX) and high availability (from the Linux-HA project). It is open source and free, and will probably be picked up by almost every other vendor in the near future – the possibilities are eye-opening. (URL...?)

Brave early adopters using the SSI software will run a separate operating system image for each task they have in mind. For example, each member of staff might get their own operating system image to work with, plus the company email servers, web servers, firewalls and file-and-print servers. They may conclude that this entails having 150 operating system images. They would then work out the CPU load involved, and perhaps conclude that 8 servers would be enough to manage it. (This is quite a normal figure, by the way.)

So they buy 8 servers, cluster them and run 150 copies of user-mode Linux on them. If they guessed wrong, they will just buy another server, include it in the cluster and watch as the load balances itself evenly across the (now) 9 servers.

This makes performance and capacity planning entirely trivial. If users complain of poor performance, add some more servers. If you need to save some money, remove some servers and sell them. If systems are getting old, remove

them one-by-one and replace them with newer ones. None of this activity needs downtime or interrupts users' work in any way. As far as the users are concerned, all 150 operating systems are up and running and working 24 hours per day, gradually running faster and faster as new machines are installed.

If one of the system administrators has an idea that they want to try out, they could simulate the 150 images on their laptop (somewhat slowly) and see if it works. If it does, they just copy the disk images on to the server cluster – or even, merge their laptop into the server cluster and let the new versions migrate themselves off.

2.5 Summary

Downtime, CPU performance problems and weekend work are probably soon going to be eliminated. In return we will have weird I/O bottleneck problems and far more complex environments that require vastly more experienced system administrators to understand.

Currently the technology to do this only exists on Linux (x86), and to a lesser extent on the IBM zSeries and iSeries. It should be relatively straightforward for this all to work on other hardware platforms that Linux supports.

At this stage, no other major operating system platform supports the idea of clustered, load-balanced, high-availability virtualisation. However, it offers enough of a revolutionary step forward that it is likely the every other vendor will play catch-up to implement the necessary functionality. The fact that the source code for the Linux versions of these pieces of software is available will make their job easier and faster.

3 Componentisation

3.1 Appliances

At the opposite end of the spectrum to section 2, there is a continuing trend towards device simplification. Some of the most interesting devices are single use ones. Very few devices or pieces of software have picked up a reputation of being "set and forget"; even fewer have the reputation for being usable by novice users. But with the growing cost of system administrators, such devices are becoming more cost-effective.

For example, the "standard" now for home broadband connections appears to be Netgear. Small and quite cheap, they miss some fundamental functionality necessary in a large organisation (such as SNMP). An organisation large enough to employ network administrators would be complex enough that the missing functionality would rule the Netgear routers out of use – but an organisation *not* large enough to employ a network administrator needs something very, very simple. (More so for home users.) And that is what they provide.

March Networks SME Server (formerly known as E-smith) is another good example. Their Linux distribution misses some crucial components – no graphical user interface, no development tools, support only for a maximum of two hard disks, etc. But it is astoundingly popular – because it is simple enough that a secretary can maintain it. It attempts to do one job (be a file and print server and gateway for a small business) and to do that well.

3.2 Internet Storage

iSCSI is an attempt to implement the SCSI (small computer systems interface) over

TCP/IP (the internet protocol) – a yawn-inducing set of acronyms for what will fundamentally change the way computers handle storage. (The idea has existed before in a few different ways; for example, the Linux kernel includes a "network block device" so that a couple of computers can be chained together to form one big storage facility.)

At the moment, computers access their hard disks through one of three mechanisms – FibreChannel (fast and extremely expensive), IDE (cheap, not wonderfully reliable) and SCSI (a compromise between the two). Most desktop computers and low-end servers are equipped with IDE, which co-incidentally, is the only technology that can't be shared between two computers. With FibreChannel and SCSI, it is possible to have a large tower of disks sitting together, managed together, which is available as common space to several computers. EMC's Symmetrix, Hitachi's XP, and Raidzone's OpenNAS range of disk arrays are the premier players here.

Storage management is a fine art in large organisations. Disk arrays are expensive (often in the millions of dollars) and only have a fixed number of slots for putting in hard disks. Breaking this capacity up to give sufficent space to a large number of computers is hard work – and getting it wrong is expensive and time-consuming to fix.

But with iSCSI, this is about to change. Disks will be stand-alone things that exist on a network. Implementing the protocol and adding an ethernet card to a disk drive is not going to substantially increase the price of a disk – perhaps only by $AUS10-$AUS20 – so we will be seeing this sort of technology rapidly spread down to the cheapest of devices.

Instead of one big central array holding all the storage, storage could be scattered all over an organisation. This is of course, an IT management nightmare, but it does provide some interesting opportunities. For example, up until now, having a "live" off-site back-up site has been extremely expensive, involving proprietary protocols and dedicated links. Soon, it will be just be a matter of putting a few spare disks into another building, and making sure that every server duplicates its data to the other site.

There is one big problem with this model, and that is that the vast majority of internal company networks simply would not be able to cope with the load of disk-data transfers going across them. Many database applications are written with the expectation of having 100MB/s bandwidth from the computer to its disks; a "standard" ethernet network can reasonably deliver 4Mb/s (= 0.5MB/s), and an ordinary phone line can handle less than $\frac{1}{100}$ of that.

There will probably be several solutions to this. The first is that bandwidth costs will drop; the current crop of designer Apple Macintosh systems come with gigabit ethernet (capable of a respectable 400Mb/s (= 50MB/s)); wireless ethernet is getting faster and can sometimes be used for wide area networking.

The second factor is that many low-end programs simply don't need particularly high bandwidth, and can cope with low speed access to disks. For example, it is a very large organisation whose email system would cause more than 1Mb/s of bandwidth usage to disk.

The process of backups will begin to change as well. Firstly, keeping backups will not be quite as necessary if the data is spread over many disks in many locations. (It removes the need for backups for hardware disaster recovery, leaving backups just for recovery from operator errors.) iSCSI will also work (it appears) just as easily with tape drives, autochangers and jukeboxes. The software intelligence re-

quired to do low level backups of disk is relatively simple, and it makes sense to actually embed this into the tape drive units. "Serverless" backups for simple applications may become more common. Larger disk arrays will continue to support mirror-splitting (either in reality or by faking it in the manner of Veritas' snapshot mounts); there are a few issues of timing and synchronisation that would have to get resolved, but these are not difficult problems to solve. Will this fundamentally change how we back data up? Probably not; slowly the idea of taking a server off-line to back it will fade out even on lower-end sites, and slowly it won't be necessary to waste any CPU power of a production server to run a backup.

3.3 Putting it together

Most of the appliance devices developed so far have done intensely network-based tasks. Routers, gateways, DHCP servers, firewalls, video editors, print servers and music jukeboxes have all succumbed to the idea of a video-recorder-sized box targeted at that one role. iSCSI has now opened up a number of other options.

1. Data storage appliances – they would be used in conjunction with other appliances. Presumably, the other appliances would broadcast on a local network, identify what storage options there were and save any permanent data to them. In the forseeable future, 30GB data stores could easily be built for under $AUS300.

2. Email *server* appliances – it is easy to implement a full internet email gateway and postoffice box in a few megabytes of stable storage, using entirely free software, and low-end processors. If it weren't for the fact that people want to save and send big emails, such a device would be implementable for a few hundred dollars – it's much simpler than a router. With a data storage appliance available, the data storage problem is solved, and it all becomes relatively easy to implement.

3. Office suite appliances – this one requires some justification. The X-Windows system has been part of the Unix environment for nearly twenty years now. It lets a graphical program run on one computer, and display to another. There are beginning to be a number of office suites available (usually for free) that display using X-Windows technology (such as Abiword, StarOffice, KOffice and Siag). So the software has been written; the resulting programs can run on the kind of hardware that appliances typically use. Being completely isolated from changes on the user's desktop, it is much harder for anything the user does to cause problems. Again, the user would need to store their data on a storage appliance somewhere. As the upgrade cycle for desktop suites slows down (it is now not quite so essential to have the latest version of a piece of software to read files generated by other users), the idea of a fixed-in-firmware office suite is not quite as silly as it sounds.

There are probably other ideas that could be made to work if sufficient standardisation for interactions becomes possible. For example, a database appliance might become possible if a reliable, portable method of querying it becomes available. (Or custom ODBC drivers might be used). The performance as it read data off disk storage appliances would of course be terrible by any comparative standard – but it may be enough to run a general ledger or stock tracking system for a small organisation.

3.4 Appliance Developers

For the developer of an appliance, there are a number of advantages.

- They have complete control over hardware and software.
- They do not have to cope with changes made by the vendor of the underlying operating system.
- The complexity is vastly reduced because there will be very few interactions with other programs – an appliance may have to find (or be told about) disk storage appliances, but it will probably require not other configuration to speak of. Software development and debugging should be cheaper and faster. Technical support may be easier as users may find it easier to identify where the problem actually lies.
- There may be no other way of actually selling software except to bundle it with hardware and charge a modest premium for the hardware. For a discussion on the squeeze affecting software developers, see section 4.1. It may be the only effective way of keeping hold of intellectual copyright and preventing piracy.

3.5 Who would?

Most of these appliances would be extremely silly to implement in a big organisation. Big organisations will want to centralise all their infrastructure into big virtualised clusters (as discussed in section 2) to make best use of their system administrators, and to give them maximum flexibility.

That said, big organisations will probably find iSCSI gradually become a viable option. Network bandwidth will become cheaper and become faster until it overtakes all but the most expensive of dedicated technologies. iSCSI will probably end up in consumer devices, priced at consumer-product levels and will gradually edge out the non-TCP/IP alternatives. Perhaps in five years, large organisations may have racks of iSCSI disk storage appliances.

It is the smaller organisations that will not be able to afford system administrators who will look more widely to fixed un-upgradable, easy-to-manage appliances. Flexibility is unimportant as they would neither have the technical know-how to implement it, nor any particular need to make use of it; long-term stability and low administrative costs are the crucial requirements.

It does not seem likely that small companies will shift away from PCs (or their equivalent) as desktop systems. They will still be just as hard to manage (if not worse so), and will have their functionality being supplemented by the facilities offered by numerous appliance devices.

4 Free versus Proprietary Software

This section (section 4) covers an extremely depressing topic. You will notice that it contains almost no technology references – but the state of the IT industry is such that the technology comes a pale second to the politico-economic aspects of the changes ahead...

4.1 Polarisation

Across a wide range of business software markets, a fundamental polarisation is taking place. It can be summarised in a few rules about the popularity of software products:

- Microsoft will have a solution in the top three.

- There will be several open source solutions; put the number of users of each of the open source solutions together, and they will be in the top three. (This is quite reasonable because of the much lower "friction" in moving between different open source solutions. For example, moving to FreeBSD from Linux is a straightforward activity with all the same programs working pretty much the same.)

- There will be a third commercial party in the top three who is not Microsoft, and not open source. Mostly they will be losing market share, losing money on the product and in danger of abandoning it.

Here are a few examples to demonstrate. The ordering may be wrong in a few places, and most haven't been checked by reputable sources. They are fairly easy to verify though.

Platform	**1st**	**2nd**	**3rd**
Desktop Operating Systems	MS-Windows	Macintosh	*Linux*
File and Print Servers	MS-Windows	*Linux*	Novell
Databases	*PostgreSQL*	Oracle	MS-SQL
Office suites	MS-Office	*StarOffice*	Claris
Web servers	*Apache*	MS-IIS	iPlanet
Mail servers	*Sendmail*	MS-Exchange	Notes
Programming Languages (other than C/C++)	*Perl*	Java	MS-VB

Italics show open source solutions; Microsoft products begin with "MS-".

The trend shows now sign of abating. We face an IT future consisting of solutions provided by Microsoft, or available for free as open source. There are only a handful of products that can be expected to be still existing over the next five years, and even then ...

- Oracle may still have a strong presence with their database products

- Sun's Java may still be a viable programming platform if Microsoft are unable to kill it.

- IBM will probably still have a very large number of legacy Lotus Notes users.

- Apple may continue to exist with MacOS – but it won't count as a commercial player with the underlying operating system becoming progressively more open source.

4.2 A word on Microsoft

In the midst of technological article, it seems odd to digress into the financial health of Microsoft. However, they are the only non-open source developer with any expectation of continued dominance in any market. So the question is – will Microsoft be around long enough to continue to direct technology for the next five years? It seems likely that they will be, although perhaps not as strongly as they have directed the IT industry for the last five.

Microsoft achieved better than 50% annual revenue growth after Windows 95 was released. This is a remarkable amount of growth for what was then quite a large company. Each year since then their revenue growth has dropped by around 10 percentage points. If this holds steady (and there is no compelling reason not to think that it will) 2001, or perhaps 2002 will mark their last year with positive revenue growth. Of course, they may still be extremely profitable, but they will be gradually shrinking over the next five years.

There is still uncertainty over the major court cases pending against them in Europe and the USA. So far they have lost over

$USD250million n other civil cases for damages caused.

Just before the dot-com crash, they were earning more from their investments than they were from their software. It has been asserted ([find the URL Greg]) that the way they have been reporting their their employee share options in their financial returns gives an inaccurate picture of their true financial health. (The profit figure they present to shareholders does not have to include some of the major costs of providing share options to their employees.) Since Microsoft have been generous with share options to keep employees, it has been reasonably asserted that they have not been profitable since 1998. It should be noted that Microsoft did not pay any corporate tax in 2000 because the deductions they claimed for share options (unpresented to share holders) more than wiped out their presented profit.

MS-Windows is now the largest software development project in human history, and is now probably twice as long (in lines of code) as any other completed project. It may be the most complex engineering project ever undertaken. To put it into perspective, the source code to MS-Windows is now at least 50% bigger than the estimated size of the software components of the ill-fated Star Wars defence project, which at the time, was not believed to be possible to complete. This is an enormously difficult body of code to work with, and they are faced with keeping backward compatibility with several generations of it.

Despite all this, Microsoft has vast amounts of cash to continue to fund development of its software; probably enough for it to continue R&D activities at current levels for another five years. Towards the end of that time it may be beginning squeezed by falling revenue, but it suffers no immediate dangers. source base it is now working on, this will of course give them less actual productive change than in the past five years. Moreover, with no particular commercial competition and no obvious significant goals, they will have no particular force driving them to innovate.

Microsoft's biggest challenge is convincing buyers to migrate from their existing Win95/98 desktop systems to newer versions of their operating systems. Their current plan appears simply to be attrition – XP is not designed to run on even moderately old systems.

In summary, Microsoft will be very slow moving over the next five years. They will be developing `.NET` and positioning themselves more strongly as an ASP but will not offer any significant improvemnts in server technology beyond what is available in XP. The only obvious exception would be something like the virtualisation technology discussed in section 2 which might be a noteworthy enough technology gap to raise Microsoft's attention.

4.3 What will survive?

Clearly, commercial software vendors are not in particularly good shape. They must somehow walk a line between competiting against the global `.NET` aspirations of the richest software company in the world (suicide) or competiting against free software (also suicide). There does not appear to be much of an option for anything in between.

So how will software be developed? Who will do it? Will there be software development companies?

Many large organisations have their own internal programming staff. For them, it will be business as usual. Instead of using proprietary compilers and development tools, they will probably be using some `gcc` front-end and a free development environment. Their oper-

ating costs will go down, but the end-users expectation will continue to rise as they always have done. Some of the software they generate will be fed back into the body of open source software, and most will be far too specific to the one company to be of any use to anyone.

Most consulting companies, large and small, have some programming expertise. From the armies of programmers that IBM (particuarly IBM GSA) will put on to a legacy integration job through to the sole contractor spending an afternoon automating a task for a small client, people will get paid hourly rates (or fixed price contracts) to perform programming. Some of this work will form the basis of work for the next client and some will end up as open source. Customers have been getting more and more suspicious of closed-source software (particulary from small vendors), so there will be steadily less scope for selling the packaged intellectual property. But essentially, there is no big change here; the bulk of the money for these kinds of consulting projects has always been the services to implement it and ultimately it makes little difference whether it is based on a Microsoft framework or free software.

The most heavily hit will be the commercial off-the-shelf software vendors. Even the software development giants of two years ago seem to have run out of steam. SAP typifies this process with hundreds of millions of dollars of software sold per year in recent memory – they are now trying to reposition themselves out of the software sales business.

Some commercial software developers will move into appliances – see section 3.4 – if the sort of software they develop makes sense in that context.

4.4 Implications

Simply put, if your current company infrastructure is based around commercial software, you are in trouble and can expect churn over the next five years. There is a very good chance you will be migrating off it – regardless of whatever it is.

- If you are Microsoft-based you will be price-gouged as Microsoft attempts to keep up revenue growth. You will want to migrate off it to get your costs down – the longer you wait, the more expensive the transition will be.

- If you are not Microsoft-based, your vendor may not be in business for long. If you rely on more than a few software vendors, you can be pretty sure that one of the important ones will fail in the near future. Check out their latest annual report and see if you are still confident.

After this depressing little pair of statements, it gets worse. The only logical and prepared course of action to survive is to build an entirely open-sourced infrastructure base, or to have your own software developed in-house. Unfortunately, most business will find that there are too many missing components to fill in, components without which the business cannot run. For example, customers of one of the unlikely-to-survive database vendors would probably consider migrating to PostgreSQL (the premier open source database)... only to discover that it is too weak in its high availability support, or its hot backup mode isn't sufficient, or that you have been using proprietary SQL extensions which it does not support.

4.5 What is there to do?

So what is there to do? It is extremely difficult to judge. Essentially it comes down to this – in the next five years, you will either:

1. Pay bucket loads of money to Microsoft to get a complete integrated end-to-end suite of products that interoperate with nothing other than Microsoft, are filled with security holes that Microsoft may or may not get around to fixing and run up obscure bugs that no human being will ever be able to trace;

2. Or, you could switch to open source early, lose functionality, have an incomplete IT infrastructure and limp along and hope that the money you saved on IT is less than the money you lose at the front end of your business as a result of the switch;

3. Or, you could switch from other commercial vendors to open source late (e.g. when the vendor fails) and run up massive integration bills in very short time frames as you race to complete the switch before support stops. You will probably do this several times.

All three options have the potential to destroy a company completely if done incorrectly. All three will pretty much guarantee that the CIO gets fired for making a bad decision even if it was the best option.

5 Summary

There are no short or mid-term limitations on the growth of hardware performance. Larger organisations will be pushing more applications on to few machines, and relying on software partitioning and process migration to keep applications isolated. Smaller organisations and home workers will look more to appliances for their computing infrastructure. Everyone is going to be adversely affected by the shakeups happening in the software industry.