



TMurgent Technologies

Supersized x64 Servers:

**Do you want 64 or 32 bit apps
with that server?**

Introduction

New servers based on what is generally called the “x64” architecture CPUs from AMD and Intel are now available on the market. These new processors allow for a 64-bit operating system to run both 64 and 32-bit applications side by side without modification.

This advance is of great interest to the Terminal Server and Citrix communities due to the unique scalability limitations of the 32-bit Terminal Server, which is limited to 4GB of usable RAM and suffers from a shortage of “Page Table Entries” (an internal housekeeping function of the Microsoft OS)¹.

With the necessary software finally coming together, it will soon become viable to run x64 based Terminal Servers. In addition to scalability due to x64, enterprises may see form factor (dual core) and price incentives to buy x64 on new servers in the near future.

This short paper addresses the question of which applications need be, or should be, 64-bit executables when run on an x64 based server.

The conclusions of this paper should be considered an opinion at this point in time (early 2006). Additional developments may yet affect the conclusions, however well researched they are. These conclusions are based upon the experience of porting and testing dozens of programs involving over 250,000 lines of code.

The Hardware

Intel (and HP) developed the Itanium 64-bit processor which was first released in 2001². Use of Itanium has been limited because in that chip architecture, both the operating system and applications must be 64-bit. Because there is a limited, although important, need for applications to be 64-bit the ISV community did not port many applications to Itanium. While software based emulation of 32bit applications is theoretically possible (much as 16bit applications are emulated on 32bit processors by Microsoft’s “ntvdm”) performance concerns prevented that from happening. The Itanium based servers have thus held a small niche market handling a few very large applications.

The x64 hardware has been available for some time now. With x64, most 32-bit applications may run unmodified with emulation provided by the hardware.

¹ See “*Terminal Services Scaling and Performance on x64-Based Versions of Windows Server 2003*” <http://www.microsoft.com/downloads/details.aspx?familyid=9B1A8518-D693-4BBB-9AF8-B91BBC0D2D55&displaylang=en>

² See “*Itanium, A Cautionary Tale*” http://news.zdnet.com/2100-9584_22-5984747.html?tag=st.num

The distinction between hardware and software emulation is very important from a performance point of view. Software emulation requires multiple hardware instructions to execute for each emulated instruction. With x64 hardware emulation, this emulation is performed on a one-to-one basis – the 32-bit instructions are executed directly by the CPU. This means that 32-bit applications run in an x64 system will run with relatively the same performance as if they were run on an identical x86 processor.

The hardware industry is also shifting to multi-core technology in new server designs. This is due to a combination of footprint, heat, and scalability. To keep their costs down, if the software market for x64 based machines exists, they may well place pressure on system builders to push x64. It would not be surprising to find it harder and harder to purchase an x86 server in the future.

The Software

The adoption of the x64 hardware platform will have a minor impact on most of the ISV³ community. They can (generally) support the x64 platform without a costly port or maintain two versions; existing executables will mostly run. However there is some software that must be ported to a full 64-bit.

The first software that must be 64-bit is the Operating System. Microsoft was slow in releasing the x64 Operating Systems, which was in Beta for over a year but was finally released in 2005. The Windows Server 2003 release for x64 parallels that of the 32-bit version, and it appears that future OS releases from Microsoft will be released in parallel also. We have been running this OS with Terminal Services enabled in our lab at TMurgent since release and can report no problems in stability⁴.

I should point out that in the releases of Microsoft's x64 OS, all components, except those to support 32-bit applications, are 64-bit. This includes the "free" applications included in the OS such as mspaint. This should not necessarily be interpreted as an indication that all applications should be ported. One suspects that Microsoft wanted to make a point that all applications could be ported and someone internally made an edict that they would be.

Second, any software that runs inside the OS kernel must also be 64-bit⁵. This means all kernel mode drivers. Until recently, this has been holding back the x64 market. Hardware vendors are now mostly ready with their drivers. Any good

³ ISV: Independent Software Vendors. This is a generic industry term for vendors that make application software.

⁴ It has only been rebooted twice in 6 months, once when the lab was moved, once for a security update.

⁵ Note this quote: "Microsoft has announced that the next major release of Microsoft Exchange Server, code-named "E12," will ship for production deployments as a 64-bit-only application" see <http://www.microsoft.com/presspass/features/2005/apr05/04-11Itanium.msp>

system builder can ensure that you are buying components for which there are available drivers, but some care is still cautioned. Most important are external components that connect up and need drivers. Printer drivers are obvious, but also consider items like security keys.

Citrix is of course a key component in this market. In late 2005 Citrix made the 64-bit version of Presentation Server available. As the Presentation Server includes quite a number of drivers and services, a 64-bit version of many of the components of Presentation Server 4 was needed in order to run on the x64 hardware.

Many other third party vendors in this market either have, or will soon have, versions of their product that either are 32-bit products tested on the x64 OS or a 64-bit version. But what about the run-of-the-mill software applications that office workers depend upon? The vast majority of these applications will work unchanged, but some of these vendors will product 64-bit versions.

The Question

So Hardware, OS, and Citrix are ready for x64. Let's say that you are ready too. Let's also say that a vendor of a software package you need has both a 32-bit and 64-bit version available and either will run in x64. Finally, add in that there is no added functionality in the 64-bit version of this application.

Which one do you want on your x64 server, 32 or 64 bit applications?

Some additional information is necessary to answer this question.

If we look at something like mspaint (Microsoft Paint) in the operating system, functionally we can quickly determine that there is no effective functional difference between the 32 and 64 bit versions. Nobody in their right mind is going to paint a picture using that tool which requires a larger address space. You can copy the 32-bit version of mspaint.exe to an x64 server and it runs just fine also.

The size of the x64 executable is larger than the x86 (32-bit) version⁶. If you run the two side by side on an x64 server you will see that the memory footprint is also larger. While the OS is no longer constrained to 4GB or ram⁷, with 100 users on a server you might have 1000 processes – if each process consumes a few extra MB of memory it starts adding up real fast. Someone still has to pay for all that ram!

⁶ File size 557kb versus 336kb, a 66% increase.

⁷ Theoretical limit for x64 systems is about 20 Exabytes (actual is lower). An Exabyte equals one billion Gigabytes.

In our lab we took several software programs and ported them to x64 to look at examples of increased memory usage where we could be sure we were looking at apples versus apples.

To build x64 applications, we first needed to port them from Visual Studio 2003 to the new Visual Studio 2005. The 2005 version was needed so that we could produce both 32 and 64 bit versions using the same Microsoft library APIs.

Microsoft
 Visual Studio
 2005

Applications build with Visual Studio 2005 may be slightly larger than with 2003 individually. This is because Microsoft has re-written the run-time libraries. These new libraries will help applications avoid stability and security threatening buffer overruns, and are well worth the porting effort, even ignoring 64 bit. I have seen that the new run-time also appears to share better, so that even though an individual application may be slightly larger, concurrent application memory use may actually be lower in some cases.

So what is actually bigger in the software when built for 64 bits? In the 16 to 32 bit port we had both the memory size increase and the basic way we count things, the integer, double in size. In 32 to 64 bit porting, we don't change the basic integer size. If 32-bits was enough for a counter before, it probably still is big enough now⁸. Pointers (and things that act like pointers, such as handles and sizeof variables) double in size⁹. Thus one should expect an identically built piece of software to use more memory on an x64 server when built as a 64-bit exe than when built as a 32-bit exe, but not to double in size.

I ran a small test program that shows the following difference in memory size and handle use:

Example	Exe Size	Working Set	Page	Handles
VS2003 built 32-bit	264KB	3660KB	1244KB	27
VS2005 built 32-bit	320KB	3612KB	1228KB	27
VS2005 built 64-bit	478KB	4976KB	2628KB	20

These numbers were all taken from running the software on the same x64 server. Different software would have different results, but a 30 to 70% increase in memory between the identically built 32-bit and 64-bit program should be expected. If you receive no benefit from the 64-bit rebuild, and you have to provide that extra memory, you might prefer to run the 32-bit version.

There are additional advantages to running the 64-bit version to be considered. The first is shown in the final column of the previous table. Notice how the

⁸ We already have 64-bit integers available in 32-bit processors, if needed.

⁹ Time also doubles in size. However, this change is related to the new 2005 runtime and not 32/64 bits. It will keep us from having another Y2K in around 30 years.

number of handles used by the software reduced in the 64-bit build? The table shows that this is not related to the new run-time libraries. The extra 7 handles appear to be related to how the WOW does its work. Ultimately 32-bit code does have to interface with a 64-bit OS and those extra handles are a part of the small penalty that must be paid. Not that I have ever heard of a system “running out of handles”, but clearly there must be some performance overhead here.

The second advantage to a 64-bit build is that the compiler may make use of additional registers that become available. This makes the processing more efficient, as the processor does not need to spend cycles (and memory bus bandwidth) moving data in and out of registers. This probably more than offsets the fact that some of that data (pointers and handles and the like) are now bigger to move, on applications that make complex, lengthy, calculations¹⁰.

How much memory can you save by using 32-bit apps on the x64 server? Try this experiment. Check the memory (page file) usage of one of your 32-bit terminal servers before any users log on. This is the Base OS (and Citrix, if installed) usage. Now check the usage after everyone is logged in. Again, you want to look at the total memory being used, not just RAM. Subtract the Base OS from this to get the difference. Now take 50% of that difference (an estimate of how much more memory those applications would use if rebuilt as 64 bit). Cost out how much money it takes to add that amount of memory to each of your servers. Ignore the 50% due to the base OS because you will have to pay for that anyway.

The Answer

There is no one answer to the question posed at the beginning of the last section, but a general statement can be made¹¹.

For most run-of-the-mill office applications, to achieve the best performance at the least cost on a terminal server, you should try to stay with 32-bit applications.

There are exceptions to that statement.

Exceptions include situations where there is additional functionality which only becomes available in the 64-bit version of the application. It also includes applications that consume vast amounts of CPU performing complex calculations.

It may also include cases where the 32-bit software just won't work, although in my experience more things are broken in rebuilding to 64-bit.

¹⁰ Computing a large factorial number may be lengthy and consume a lot of CPU, but it would not gain much advantage with additional registers. Weather forecasting, on the other hand, would.

¹¹ This White Paper is free; remember that you get what you pay for.

Other Thoughts

One outstanding question is whether software vendors will give us a choice in this. Some may produce an installer that detects the OS and installs the 64-bit version automatically. This is not desirable, and ISVs would do better to offer a choice in the installer.

The community will need to be vocal with these vendors to make them understand that you want that choice.

Note: Various terms and company names have been used in this white paper which are trademarks of those companies, including Microsoft, IBM, AMD, Intel, Hewlett-Packard, and Citrix.