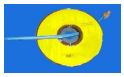

TMurgent Technologies

White Paper:
It's Time to Change the Timers

Catching up with higher clocking CPUs

April 15, 2004

Introduction

The purpose of this paper is to highlight the need for some changes that computer manufacturers and operating system vendors need to make. Enterprises spend considerable money on "Server Based Computing", and they could be getting more for their money than they are.

Those who buy, run, and use server based computing in the enterprise need to become aware and conversant in this topic; become vocal with their vendors; become an advocate for change. It is through you that the vendors will get the message and make the changes necessary so that we may utilize the full power of the systems we buy.

It can't be done alone by the hardware folks, or the operating system folks. We need to get them to work together to make the changes needed.
.

Faster Cycles, Slower Decisions

The number of CPU instructions processed per second have continued to climb in the last 10 years. Not only do we have the processor speeds improving ten-fold, but we also have more processors in our systems.

Compare a single processor system with a 300Mhz Pentium processor against a Dual Processor machine with 3Ghz Xeon Processors. Without enabling Hyperthreading, that is 20 times the number of instructions per second. Depending on the scenario, when you enable HyperThreading you should see 30 to 34 times. While this is a good thing for productivity, it has effects that need to be looked at once in a while.

Some operations within the Operating System are based upon timers. Often, the user of timers solves issues of running the same software on different speed hardware. In the early days of the PC, we used to have a "turbo" button on the PC to slow down the PC to a slower clock speed. This was needed for a variety of game software that didn't use timers and became too fast to play on newer machines.

Timers solve this problem by making operations happen based upon a real-world clock event. So today, the designers of software wait on a timer for maybe 1 second before updating. And it's not just games. Timers show up in all sorts of productivity software, whether it's the 10 minute timer your email client uses to check for new email, or the autosave timer on your word processing software. They also are used within the Operating System.

In many cases, the OS does this because the service provided is human time based and probably should be consistent on different machines. For example, timeouts for any communication protocol need to be fixed to a standard wall clock time. You don't want one device timing out a TCP packet and resending the packet just because it's timer was shorter than the receiving device (which was saving up to acknowledge several packets at once, a standard bandwidth saving technique). However, there are areas of the OS where decisions are made on a timer basis that do not need this consistency.

One example if page trimming. In a virtual memory operating system, the OS likes to keep a portion of the real memory free, so that new pages are immediately available when needed. So the operating system will, once in a while, look for memory pages that have not been accessed for a while and page them our to disk. This is called page trimming, and Microsoft does this once per second.

Another example is CPU utilization monitoring. Internally, the OS looks at what process is running every 10 or 15 milliseconds (depending if uni/multi processor). Whatever process happens to be running at that instant is credited with using the entire period. This might be unfair in the short run, but the laws of probability hold so over a long period of time the results are reasonable. Or at least they were reasonable when the number of instructions processed in 10 milliseconds didn't increase by a factor of 30. But this 10/15 millisecond clock is fixed (today).

The list could go on and on. For example, decisions on how long a thread may run (quantum) also occur on the 10/15 millisecond clock. As does priority enhancement degradation. I could go on and on in detail, however this isn't intended to be a deep technical paper.

The bottom line is that with faster CPU cycles, the OS is making decisions slower and slower.

Not Just Software

It might seem that an easy solution for CPU monitoring would be to just reduce that timer down to 1 millisecond, but it isn't that easy for Microsoft to do.

The operating system depends upon the HAL, the Hardware Abstraction Layer, to provide this clock. And the 10 millisecond clock is the standard. The underlying hardware has much faster clocking built in – it

isn't a hardware issue. The HAL creates an interface to standardize the interface between the hardware and the OS. This interface also provides access to the full clock of the hardware – so we can look at the clocking speeds available. In fact, application software that needs faster clocks poll the real-time clock. An example of this would be an audio player such as Real Audio or Microsoft Media Player. This is not a general purpose solution because polling a clock to determine the time is less efficient than waiting on a timer – but these apps could not work if they had to wait 10 milliseconds.

The HAL provides this 10/15ms clock by generating an interrupt to the processor based upon the HAL timer. The OS uses this interrupt as the basis of the timer. So to lower this value we need not only Microsoft to want to shorten the timer, we need their HAL partners to want so also.

Why Should You Care?

Our systems are becoming less efficient than they could be. The OS is making decisions based on the HAL interrupt which has not changed for a long, long, time. The number of instructions occurring in that time period has expanded – resulting in what is in effect delayed decisions.

In today's terminal server, I believe that these delayed decisions are limiting scalability – the number of users per system – in most cases (The exception being cases where the 4GB memory limitation has been reached).

I am only guessing, but we could conceivably increase the number of users on that 3Ghz box by as much as 10%. 10% without changing hardware!

Just fix the HAL and OS code.

Conclusions

We need to begin a campaign to press both Hardware and Operating System vendors to address this issue. None of them need to "go it alone" and take risk. We just need them to work together and reduce the HAL timer.

While I mentioned a 1 millisecond value in an example in this paper, it was only an example. I am not proposing it as the new value. I trust that Microsoft and the hardware vendors can properly study not only where we are now, but where we will be in another 10 years and select an optimal value for that time period. The optimal value might be less.

And in the process, Microsoft will need to ensure that the OS detects and supports "older hardware" without degradation. It should not be very hard to do if they think about it before-hand.

Those who buy, run, and use server based computing in the enterprise need to become vocal with their vendors; become an advocate for change.

The process is faster if the trade press joins in. I encourage them as well. Feel free to reference this.

And let me know what you think!

