Overactive software is wasting energy

Data centers have ignored the energy wasted by software, says *Ed Ansett*. To save the planet, idle software should start going to sleep

The reduction of energy consumption in data centers has been a focus for almost everyone concerned with engineering and technology for nearly a decade. Better cooling systems, switched mode power supply efficiency and virtualization have yielded impressive energy benefits. There is more to be done, but solid progress in these areas has been made globally.

But the same cannot be said for fixed-platform software, for several reasons. Back in 2013, when we (i3 Solutions Group) conducted research for the Singapore Green Data Centre Technology Roadmap, we wanted to look at each aspect of engineering and technology that consumes energy. What we found was totally baffling at first.

You might think the hypervisor and application developer community would be pretty hot on energy-reduction techniques since software is often used on mobile platforms powered by batteries. Conserving battery charge through low energy use could extend platform usage time.

But in a data center, battery power is not a concern, so the impact of energy-aware language structures, low energy coding techniques and the use of hardware sleep states have largely been overlooked by software developers. That seemed pretty odd to us, but the research supported the assertion that fixed-platform developers generally aren't cognizant of energy usage in the context of software.

How significant is software energy inefficiency? It can be more important than the inefficient cooling systems affecting data centers a decade ago.

A key factor causing the problem is that energy-proportional computing is not yet a reality. To understand why this is, we need to look at typical enterprise active components, particularly processors and their idle energy consumption. At zero percent utilization, a typical processor averages 50 percent of its maximum energy consumption.

So the question is, how much time do processors spend idling? The fact is, we don't really know, because utilization varies so much among users. What we do know is that processors generally spend most of their time idling. Even if we assume the idle period is 50 percent, these devices are burning 50 percent of their maximum energy while doing nothing half of the time.

It gets worse! Now consider the upstream impact of being idle 50 percent of the time. We still have to power the IT devices, so there's energy wasted in each section of the power chain – from the distribution transformer all the way through to the final DC to DC converters. Furthermore, heat from this idle energy has to be removed, so there's even more energy wasted. We call this the 'energy cascade effect.'

The need for action by the software community to deal with this issue is obvious. We initially thought not much could be done about this. Surely you can't just go around switching IT hardware off, can you? No, you can't if the software is required to react instantly to a time-independent event. But business processes fall into three categories: 1) an immediate response to a time-independent event is required (an instantaneous response); 2) a less time-sensitive response to a time-independent event (a small delay response can be tolerated); and 3) a time-dependent event (process is periodic).

The drawback to using sleep states is a latency penalty due to the additional time required to wake in response to an event. This depends on the level of sleep used – the deeper the sleep the less energy used, but it takes longer to wake.

The argument for routinely using sleep states stands up well in the second two categories. We don't need an instantaneous response to many organizational processes. Pretty much anything in category two that is time-independent, with an adequate interval between event, could benefit from using sleep states. An added complexity is the nature of IT workloads. In many instances, this technique cannot be justified, but there are many more where it can be employed.

We can't go on ignoring the issue. The energy-saving opportunity is immense, and the financial savings are too big to ignore. In a situation such as this, where most, if not all, organizations stand to significantly benefit financially, it surely is only a matter of time before the issue is addressed.

Sleep states are one important way of conserving energy; others include energy-cognizant coding of applications, application services and operating systems, also rate adaptation, energy-aware mapping of virtual machines and energy-optimized compilers.

There is much work to do, so now over to you hypervisor and application developers. Go save the planet! ●

Ed Ansett is group chairman of i3 Solutions Group

Illustration: studionippoldt.com