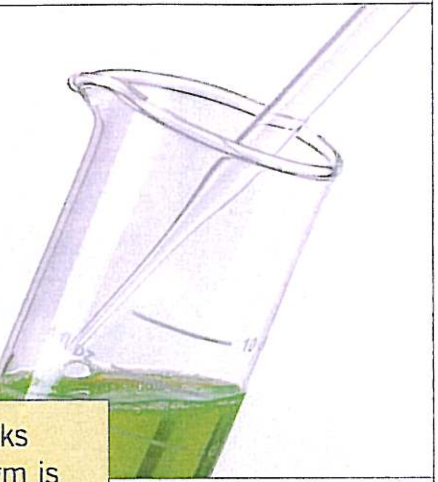
Green computing and performance testing



MICHAEL BLATT asks whether a new paradigm is emerging in the meeting of green computing and performance testing

IN *Focus*

THIS MONTH **Michael Blatt** asks if green computing and software testing can come together to provide a new basis for green software. **Eugene Kaspersky** describes what he terms the cyber crime arms race.

Paul Foley describes the current technical structures for electronic signatures

Nathan Marke describes the interconnectedness of the social and the technical in the next wave of enterprise applications, and **Ian Simpson** argues that virtualisation is not an end in itself, but merely a building block in a wider IT strategy

VALUE *Points*

Test and quality: Software testing can develop to encompass green initiatives, helping to reduce consumption and carbon footprint

Security: Sophisticated techniques, mirroring the legitimate business world, are being used in murky world of cyber crime

Signatures: Specifications are in place for eSignatures, but the appropriate type must be used with each application to be effective

Enterprise 2.0: The next wave of enterprise applications must embrace the best aspects of social networking

Virtualisation: While virtualisation is a transformational technology, it is not a magic bullet

'GREEN COMPUTING' seems to be all the rage these days, and it is easy to see why what with rising fuel and energy costs, the growing threat of climate change and the current financial crisis directly affecting financial institutions and all large users of IT. These point to a clear need to reduce energy consumption; but progress also demands ever more computing power.

Hardware manufacturers have long known the benefits of power-efficient components, and their recent offerings show a strong commitment to this. The traditional TOP500 ranking of the fastest supercomputers in service is now complemented by a GREEN500 ranking of the most energy-efficient supercomputers.

Now software vendors are warming up to the idea as well. Is it not time for software testing to follow that trend? Can we, should we use the skills and methodology of software performance testing to create greener solutions, to optimise not for raw performance, but energy efficiency?

Assessment

Performance testing perhaps more than any other testing discipline, has always needed to assess both the hardware and software aspects of a solution. Energy efficiency depends on both just as much as performance does. It is therefore vital to look at each one of these factors in isolation first, and then feed the information into a more holistic approach.

Computing devices, from smartphones to laptops to mainframes, consume more energy under intensive use. This may be high CPU usage, or large amounts of read/write activity to storage media; or any form of activity that takes place in a computing device. The trend over the last two to three years has been for hardware manufacturers to reduce power consumption, and nowhere is this more apparent than with CPUs. Performance per Watt has become an essential metric in CPU benchmarks, gaining in significance to the extent that one may with confidence state that Performance per watt is slowly becoming the new Gigahertz/MIPS'.

Examples of this include: Transmeta's Crusoe and Efficeon, Sun's UltraSPARC T1, and Intel's Core 2. Intel's latest processor, Atom, is touted not for its performance, but its performance per watt.

Stored

It is likewise for storage media. SSD flash-based disks, despite performance on par with or under that of current hard disks, are advertised for their power efficiency.

We see a general trend of going multi-core, and with Sun Microsystems' concept of a "data centre in a box", it all beats the drum of power efficiency side-by-side with that of performance, not forgetting the associated energy cost savings. Both IBM and Sun now offer a 'Green IT' consulting service, while a major UK retail chain spends around 8% of their IT budget on cooling their data centres. But are the extra costs involved in evaluating, purchasing and servicing greener hardware balanced by the associated energy savings?

US Technical Services vendor CDW Corporation reported earlier this year that just over half of IT decision makers are holding back on employing green IT technologies, citing concerns about cost. But this seems to be a short-term view in light of the energy savings associated with 'green IT'.

Should 'efficiency testing' also consider the manufacturing process as well as more obvious metrics such as power consumption and performance? It seems logical to include the extra effort and cost involved in designing and manufacturing green hardware. We effectively end up assessing the overall carbon footprint of an IT solution, not just its running emissions.

Virtual push

On the software side, virtualisation has been at the forefront of the push for greater energy efficiency. Whether we consider VMware, HP, IBM's mainframes or any of the competing offerings from Microsoft, Citrix, Sun et al, one of their key selling points is 'server consolidation', i.e. having a single power supply unit, powering a single hardware box, running several servers as though they were entirely separate machines. Considering the success of recent virtualisation solutions as pioneered by VMware, this approach seems to be paying off.

Smartphones and laptops are another example of how software solutions can reduce the energy usage of a computing device: laptop users are aware of several methods to improve battery life, such as reducing CPU clock speed, turning the backlight down, so as to commonly double the length of time a laptop can be used without an external power source.

At a lower level, highly efficient code will use less processing power, and therefore less energy; but is the extra development effort (and energy spent doing it) worth it?

The Linux kernel is one example, where the 'ticks', that took place every 10ms, became dynamic and therefore only took place when the CPU was not idle. As a result, several benchmarks indicated CPU power consumption had gone down by 2-5%. That change required a few thousand lines of code, less than 2% of the total size of the project. Such a cost/benefit ratio certainly makes the option worth considering.

Where and how

The question must be asked though, how does this translate to the performance testing methodology? How do we measure power consumption during a performance test, does a technical solution already exist? Where traditional performance testing revolves around the comparison between baseline resource usage (CPU, memory, network, disk I/O...) and peak resource usage, our hypothetical tester would compare baseline energy usage to peak values. Energy usage here would best be measured in Watts-per-hour, which has the advantage of being easily translated to energy cost.

The fundamental questions of performance testing and tuning such as how fast does the system perform with x concurrent users relative to a 1-user baseline and how can we get better response times for our users, or get more users onto the system without degrading performance, now become; how much power does this system use when idle, and how does that power consumption increase as the system gets busier? How can we improve the power efficiency of the system for a given (or all) busy state(s)?

Assessing efficiency

The testing methodology as it stands in performance testing does not need significant changes to use these results to assess the energy efficiency of the system. Energy consumption of the system during the tests could simply become another performance metric that needs to be monitored and taken into account when evaluating results. At a tools level, many data centre vendors monitor power consumption; a simple solution would be to integrate these measurements into a common performance monitoring tool, such as Windows Perfmon or SNMP; alternatively, simply log the monitor samples to a log file which can be analysed subsequently. Intel's LessWatts.org initiative hosts several projects for Linux such as PowerTOP, a dedicated tool to analyse software causes of power consumption, and has been used to great effect on laptops and embedded devices; but it can certainly be used on servers too.

A few other factors may also come into the equation:

- Carbon footprint of designing, manufacturing and assembling the hardware
- Carbon footprint of transporting the hardware to its destination
- Carbon footprint of developing the application that will be deployed on the hardware
- Carbon footprint of any workstations and servers used during development and testing of the system

Once test results are delivered, the evaluation should focus on system usage vs. energy consumption to assess the efficiency of the solution. As with standard performance testing, baselines have to be taken and compared to in-test monitor values. How the key curves evolve (now including energy consumption) will determine just how efficient the system under test is.

Optimisation

Tuning would be similar in nature, although it may differ in implementation. Is software optimisation more important than hardware optimisation when it comes to power efficiency? Recent experience shows that most performance issues are caused by sub-optimal configuration or inefficient code. In such cases, adding hardware does not resolve the issue, but at best pushes it back, delaying the inevitable onset of performance degradation; at worst it can be a costly distraction from the root cause of the performance problem being investigated.

Underlying function

In the context of energy efficiency testing this may well differ, for energy consumption is first and foremost a function of the underlying hardware. Or is it? Efficient code will consume less energy than equivalent un-optimised code running on identical hardware, but what about energy efficient hardware? A wise choice of components may go a long way toward running a solution with a smaller carbon footprint, but how great are the gains that can be achieved via software means, such as optimising code, using RAM rather than storage media when possible (incidentally this would also improve performance), tuning application parameters to reduce logging...?

Using LessWatts.org as a sample, certain configuration changes on a test server have reduced idle power consumption by 8W, and consumption under load by 16W. This may only be about 5% of what a server typically consumes; but the implementation was under 5% of the kernel code, which is itself only part of the total code involved in running the application under test. This can also be replicated to each and every Linux kernel worldwide at very little extra energy cost, thus potentially multiplying the savings by several millions (exact figures are impossible to come by given Linux' status as free software). Software optimisation can be copied ad infinitum - this is something hardware optimisation alone cannot hope to achieve at such small cost.

Four pillars

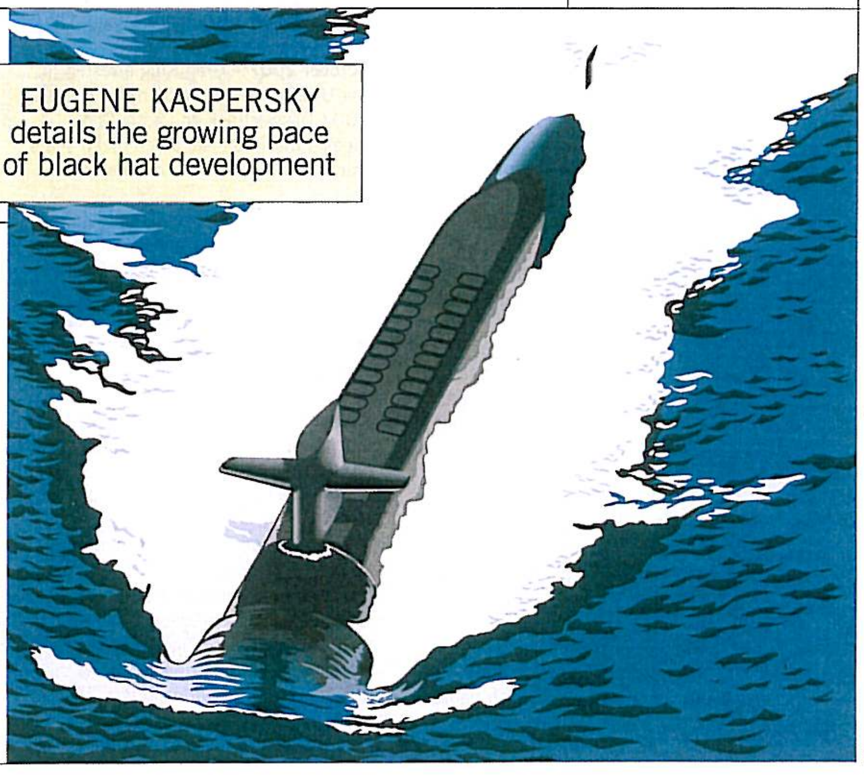
Should project and IT managers go for cost-effective or carbon-efficient solutions when deploying new systems? Are the two mutually exclusive? We now know that carbon-efficient systems are less costly to run in the long term, but it may mean paying a premium at the initial purchase. While the traditional factors have been reliability, performance and cost, it seems a new criterion should now be taken into account. Should this not be tested just like costs (the 'budgetary test') reliability (through soak tests, disaster recovery testing), or performance? Should reliability, performance, cost and carbon-efficiency be the four pillars of IT system design?

:: Michael Blatt is the head of the Performance Testing Innovation Group at SQS UK.

'Smartphones and laptops are another example of how software solutions can reduce the energy usage of a computing device'

The cybercrime arms race

EUGENE KASPERSKY details the growing pace of black hat development



Our society has evolved to the point where many if not most of us spend a significant portion of our lives online. In many ways, this online virtual world mirrors our real world. Criminals, who are an unfortunate but integral part of our social structure, quite naturally have also appeared in the virtual world. The presence of cybercriminals has become more pervasive today because the freely growing online exchange of money and data has created an increasingly tempting target. Today the cybercrime ecosystem is close to maturity – welldefined relationships and business models are already in place. A new class of cybercriminals freely and openly buys and sells malicious code. These cybercriminals range from petty fraudsters who steal small sums in large quantities to individuals who attempt to steal large sums of money at one time.

Criminal activity has always mirrored legitimate business – the image of a Mafia accountant may be the first to come to mind. However, it is worth noting that cybercrime is not currently organised into one or more worldwide

>>