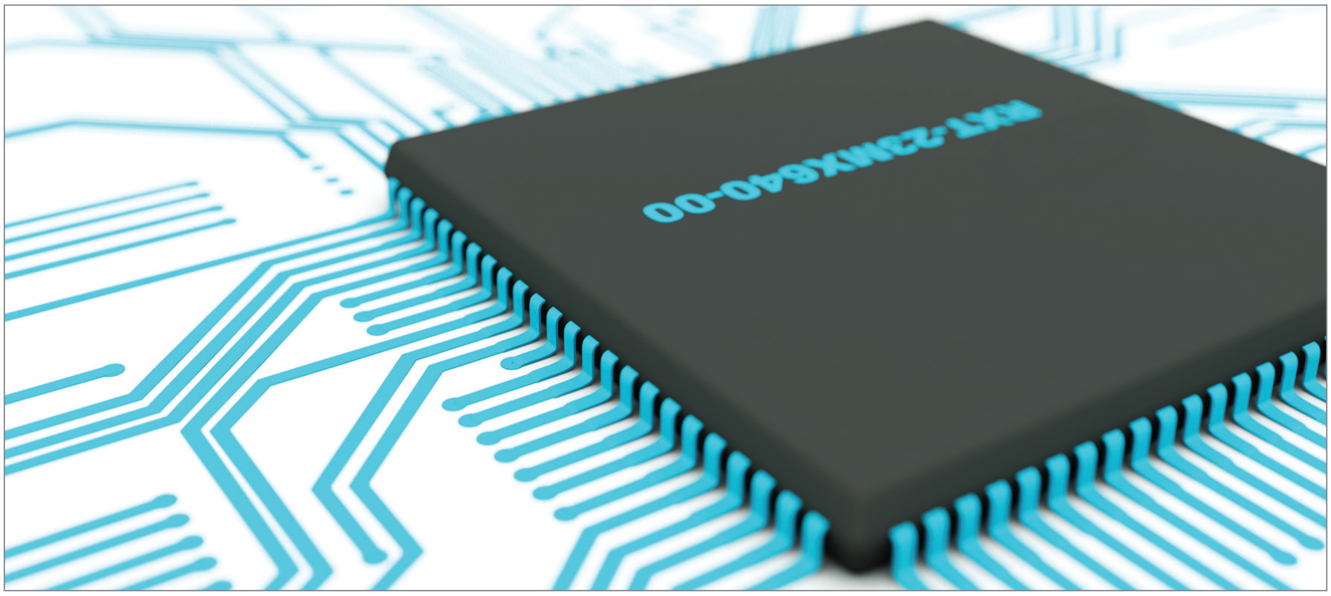
REPRINTED FROM
Risk
RISK MANAGEMENT | DERIVATIVES | REGULATION
RISK.NET APRIL 2013

Chip and win

As computational demands on banks have increased, some have turned to powerful graphics processing units, but these were initially applied at the transaction pricing level. Now, they are starting to cover portfolio valuations and other enterprise-level tasks. By ***Clive Davidson***



The first recognised land speed record was set in 1898 by the Jeantaud Duc, a car that looked like a mechanical whale set on four big, spoked wheels – it clocked in at 63.15 kilometres an hour. The current record holder – a 16-metre-long monster that was essentially a jet fighter without wings – recorded a speed of 1,228 kilometres an hour. It took 99 years, the deaths of an estimated 35 drivers and a huge amount of ingenuity to achieve that 19-fold increase in speed.

Faced with the need to run bigger, ever-more-complex calculations, banks are engaged in a similar contest, and some are abandoning their old vehicles – the central processing units (CPUs) used in conventional computing – for the graphics processing units (GPUs) primarily designed to satisfy the demands of modern computer games. The increase in speed can be dramatic, as ING discovered when it first tested GPUs on historical value-at-risk calculations in 2009. The run time dropped from seven hours and 26 minutes on a CPU to 8.5 minutes – a 52-fold increase. And no-one died in the process.

The Dutch bank is just one of the institutions turning to GPUs as a computational short-cut. In part, that's because of the growing importance of numbers such as credit valuation adjustment (CVA) – which can require hundreds of billions of individual calculations, says Tim Wood, an Amsterdam-based quant at ING Bank – and in part it's because GPUs are becoming cheaper and more powerful.

"The availability of affordable raw computing power is no longer an issue," he says – but new issues have taken its place instead.

When GPUs first rose to prominence a few years ago, they were primarily used to price individual trades (*Risk* November 2010, pages 65–67, *www.risk.net/1741590*). Now, they are being applied to more demanding, multi-step processes. But while GPUs might be tailor-made for operations that require raw computing power – such as Monte Carlo simulations, in which huge numbers of calculations can be carried out at the same time – conventional CPUs are better at performing sequential tasks. As a result, banks have to examine the problems they want to solve, identify the parts that are best tackled with GPUs, and design their applications accordingly. GPUs also require new software tools – programming languages and development toolkits that need highly specialised skills and different ways of thinking.

The starting point, in many cases, is the raw material – data. Put simply, there is no point having a processor that can execute massive numbers of parallel instructions if the data can't keep up. This has become a bigger issue as banks move from deploying GPUs for front-office pricing, to enterprise risk analysis. "Calculating CVA at the portfolio level involves large, complex input and output data, including trades, market data to price the trades, counterparty information, and netting and collateral information," says Wood of ING.

This data has to be marshalled and delivered to the processor to match its work rate. Conventional relational databases running on hard disks can't keep pace, so banks are turning to in-memory databases – such as VMware's GemFire, Oracle's Exalytics and SAP's Hana – that can store information alongside the GPU, shooting data across in sync with the processor's clock cycles.

"Efficient data handling is key to efficient GPU implementation," says Vladimir Piterbarg, head of quantitative analytics at Barclays. The bank went live with a GPU-based system for its rates business in December (see box, *No limits*).

The next challenge is to work out which bits of a complex process should be handed over to GPUs – something Barclays also had to confront. "In a Libor market model (LMM), for example, there is a calibration step that has some associated computational overhead. You don't gain as much from putting that step on a GPU as you do when running Monte Carlo simulations," says Thomas Roos, head of quantitative analytics for fixed-income rates at Barclays.

So, how did Barclays approach the problem? "We started from our existing production LMM model, looked specifically at the pieces that would gain the most from executing on a GPU, then wrote GPU versions of those routines," says Roos.

That sounds simple enough, but this delegation of tasks to different technologies has to be done intelligently, he says. Code for things such as Monte Carlo path generation – required for both CPU and GPU elements of the application – tends to be stable and is rarely touched once written. Other elements of the application require ongoing maintenance – those describing payouts, for example.

"You don't want to be in a situation where you have to write two versions of the payout for every new product you introduce, building a large maintenance burden," says Roos. Barclays will not say how it solved this particular conundrum, but one possibility would be to use a tool like Xcelerit, which allows quants to program in their familiar C++ language and then translates this into code GPUs can execute.

Banks also need to decide what tools they will use to implement GPUs. Although there are a number of chip makers out there, most focus on the video gaming industry, making their products less suitable for financial modelling. As a result, banks and financial technology vendors tend to use California-based Nvidia's GPU boards. One reason is that Nvidia also offers the proprietary high-level Cuda development environment

No limits

While many of the biggest banks have in-house graphics processing unit (GPU) enterprise risk management projects – BNP Paribas, JP Morgan and Société Générale are among those known to be using GPUs already – system vendors are starting to make the technology available to smaller institutions, too.

In February, Stockholm-based TriOptima introduced a counterparty credit risk analytics service, including potential future exposure (PFE), credit valuation adjustment (CVA) and funding valuation adjustment calculations, which institutions can use to validate and benchmark their own internal systems or use as an outsourced risk service.

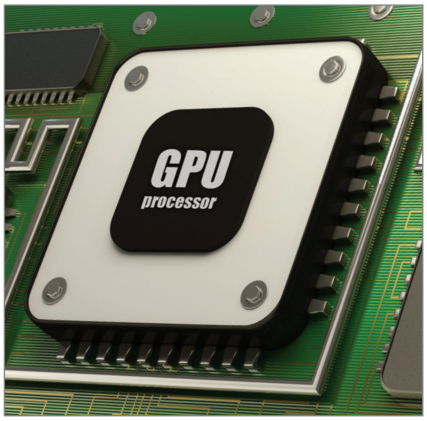
London-based Misys has also introduced a version of its Global Risk system with GPU-based calculation of PFE and CVA. The company turned to GPUs when it realised it would not be able to deliver a system with the desired performance or costs using CPUs. Nor could it rely on a bank's front-office systems to price portfolios fast enough to feed the PFE/CVA calculations in its risk system, says Thomas Moser, product manager for Misys Global Risk.

"A GPU card costing around $3,000 can have 4,000–5,000 cores, which means you can carry out 5,000 operations in parallel, whereas a single CPU has only up to 64 cores," says Moser. The company has banks in Portugal, Austria and Russia currently implementing its GPU-based system, he says.

Paris-based Murex was one of the first pricing and risk system vendors to commit its entire pricing and risk analytics libraries to GPUs. It claims to have a number of banks now in production with GPU-based versions of its software, although it would not identify them. Pierre Spatz, head of the quantitative analysis team at Murex, says clients are typically seeing 30 times acceleration of performance compared with conventional processors. One bank that has implemented GPUs to support intra-day portfolio management of complex equity derivatives exotics has also been able to triple volumes, as well as increase the number and accuracy of risk measures it computes, he claims. "This technology is not only about being faster, but also about handling more volumes and computing more outputs with better accuracy," says Spatz.

Pierre Spatz, Murex

But technology specialists say it is impossible to generalise about the performance gains GPUs offer. There is a big difference between using a GPU to speed up a single-factor pricing model for an individual trade – where enormous performance gains are achievable – and using GPU engines to solve specific problems in an enterprise market or counterparty risk management system. Barclays, for instance, had already optimised its CPU code, roughly doubling performance on modern server architecture. Rewriting the system for GPUs has given a significant advance on that, although the bank would not give an exact figure. "People claim a theoretical improvement factor in performance with GPUs in the hundreds, but that is rarely achievable in a real-world setting, although we have gone quite a long way towards that," says Thomas Roos, head of quantitative analytics for fixed-income rates at Barclays.



> “We started from our existing production LMM model, looked specifically at the pieces that would gain the most from executing on a GPU, then wrote GPU versions of those routines”
>
> Thomas Roos, Barclays

– an extension of the programming language familiar to most quants and tailored for optimal performance on its chips.

The alternative would be to use the Nvidia GPU boards in conjunction with an open-source software standard, such as OpenCL. Companies including Misys and Murex have done exactly this, in part because they want to avoid being tied too closely to a single vendor. OpenCL also runs on CPUs, field-programmable gate arrays (FPGAs) and now Intel’s new Xeon Phi parallel processing chip (see box, *Fast forward*). But many banks that develop their own software are happy with Cuda, claiming it is richer in features and associated tools than OpenCL. “And no GPU vendor wants to implement OpenCL optimally because they have their own proprietary languages,” says one bank developer.

Whatever the choice of programming language, banks may find themselves short of the necessary expertise. Both Cuda and OpenCL are just one step up from machine code, which instructs the chip about the operations it must perform. Most quants – and indeed most developers in bank IT departments, and even at pricing and risk system vendors – have no experience of such low-level programming, and typically use higher-level languages such as C, C++, C# or Java that are not only more abstract, but also assume a sequential – rather than parallel – program design. The lack of experienced parallel design programmers with financial knowledge is one of the major barriers to GPU adoption, banks say.

One workaround is to use conversion tools that can take quants’ sequential code and turn it into parallel instructions. Companies such as Texas-based SciComp and Dublin-based Xcelerit offer versions of these tools specifically tailored for financial services. There will always be some performance penalty when using automated methods when compared with a skilled coder working in the underlying language, but Xcelerit claims 98–99% optimisation of code using its tools. Nevertheless, some banks prefer to build in-house GPU skills and do the optimisation themselves, saying it costs roughly the same as buying the tools.

But there are some obstacles that can’t simply be sidestepped – because some programme designs and languages are more suited to a parallel format than others, running them on GPUs might mean rebuilding the application from the ground up. “GPUs might be able to solve the underlying problem more efficiently, but to use them you would have to throw away the existing application and there might be stakeholders who don’t want it thrown away,” says ING’s Wood.

What this adds up to is a profound change in pricing and risk technology. Traditionally, a bank could have a fairly standardised development environment, with its various trading and risk applications running on essentially the same hardware. The main differentiator was size – the bigger the task, the bigger the hardware box, or more recently, the computing grid.

But the drive to real-time pricing and enterprise risk management – combined with evolving technology – means banks now need to carefully analyse computational tasks before matching them with appropriate hardware and development tools. This could mean assigning an entire task, such as derivatives pricing, to a GPU environment, or mixing and matching elements of enterprise risk analysis to GPUs and CPUs, as Barclays has done. Equally, it could mean employing alternatives such as FPGAs or Intel’s Xeon Phi chip.

It could even mean switching from one technology to another for a single type of task, says Hicham Lahlou, chief executive and co-founder of Xcelerit. As an example, he says it often takes more time to assemble the relevant market data for vanilla derivatives pricing than it does to run the computation, so smaller portfolios may as well remain on a CPU. However, as the portfolio grows, it can make sense to transfer the processing to a GPU, so the developer might want to set a threshold beyond which the task switches from one chip to the other – as long as the application is coded in a generic language that operates on both CPUs and GPUs.

“The future is hybrid,” says Lahlou – banks need a toolbox of varied technologies in order to optimise the speed, accuracy and cost of each application. Wood of ING agrees: “It’s horses for courses – it’s a matter of looking at each problem and choosing the appropriate hardware, and coding for that,” he says.

That is an increase in complexity when compared with the traditional way of doing things, but the extra effort may be worth it – with the judicious deployment of GPUs, new CPUs and other technologies, even the most demanding risk calculations institutions face are becoming tractable, vendors and banks claim. ■

Fast forward

The need for speed does not just pit central processing units (CPUs) against graphics processing units (GPUs). Other alternatives are also available – for example, field-programmable gate arrays (FPGAs) are microprocessors that allow the user to configure the logic gates and memory so the chip’s architecture is optimised for specific tasks. This enables algorithms to run far more rapidly than on other chips, but also requires specialised machine coding expertise.

JP Morgan already uses FPGAs for counterparty credit risk and algorithmic trading, and they are widely used in high-frequency trading platforms (*Risk* February 2012, pages 62–64, *www.risk.net/2140629*).

But the newest kid on the block is the Intel Xeon Phi – a 62-core microprocessor designed for parallel processing but which will also run standard CPU code. Initial versions released in January cost up to $2,600, making them competitive with GPU boards. ING, Misys and others are already benchmarking the technology against available GPUs and assessing what part they could play in the industry’s expanding toolbox.