



Brian Sentance

Cluster Wars - Scaling-Up Derivatives

Xenomorph's CEO investigates the technologies and companies involved in High Performance Computing (HPC) within financial markets, and how the recent entry of Microsoft to this market will be an interesting development for both the incumbent players involved and the quant community as a whole

Structured products, credit derivatives and the need for intra-day risk management are driving many investment banks, hedge funds and fund managers to look towards parallelized, distributed computing as the fastest and most cost-effective method of performing Monte Carlo simulations, portfolio optimization and the re-pricing of computationally-intense derivative products. This article looks at some of the technology involved in High Performance Computing, the main players in the market and how things may develop now that Microsoft is making a play in this area.

Super computing vs. distributed computing

Firstly, I think it best to start with some definitions because there are more than enough abbreviations and acronyms used in this area. High Performance Computing (henceforth: HPC) seems to be the current industry acronym for modern-day "Super Computing" - which I will forever associate with the Cray supercomputers (very much alive and kicking, see www.cray.com) and classic 1970s science fiction movies such as *The Forbin Project*.

Anyway, it seems that in recent times Super Computing has become distributed. Cornell

University define distributed computing as "computing systems in which services to users are provided by teams of computers collaborating over a network". Back in 2000, Massively Parallel Processing (MPP) solutions (of which Cray would be one) held around 50 per cent of the top 500 Super Computing benchmarks, with distributed computing (more specifically, clustering) barely registering in this esteemed company. Moving forward to the present day, then distributed solutions now hold around 70 per cent of the top 500 benchmarks (take a look at www.top500.org and leave MySpace and YouTube behind you ...).

Time to go parallel

For those of you that enjoy the "Laws" of mathematics and computing, then it is probably worth looking up Amdahl's Law, published in 1967, which defines theoretically how much faster a given task would run if it were parallelized as opposed to being run sequentially:

$$\text{Max}(\text{Speedup}) = \frac{1}{s + \frac{1-s}{p}}$$

where:

s is the % of the calculation that is serial

p is the number of processors

For example, taking a problem with 16 processors where 20 per cent of the problem has to be done serially, we end up with a speed-up factor of only 4. With $s=4\%$, the speed-up factor gets to 10. In the limit as s tends to zero, the calculation can be speeded up by the number of processors/processor cores applied to the problem, in this example: 16. I am stating the obvious for many of you, but the most direct application of Amdahl's Law within computational mathematics is to Monte Carlo simulations, where the majority of the computational effort can be parallelized and hence the overall calculation time theoretically reduced by the number of processors applied.

So now we understand the theoretical speed benefits of distributed computing, what exactly are "Cluster Computing" and "Grid Computing"? Whilst Grid Computing has been far the more fashionable term over recent years, the definition and differences between Grid and Cluster are understandably blurred in my opinion - so let's try to clarify this next.

A dedicated cluster

One approach to HPC is Cluster Computing, as illustrated in Figure (1). This involves a dedicated and homogeneous group of computers, in that they are running the same or very similar configu-

ration of hardware, OS and application software.

A typical usage pattern would be that an end-user request to perform a set of calculations (a “Job”) is made to the cluster and gets routed to the “Head Node” of the cluster over the corporate IT network. The Head Node then takes the components (known as “Tasks”) of the Job and orchestrates the distribution of them across the members of the cluster (the “Compute Nodes”). The Head Node and the Compute Nodes may be linked by one or more private networks, in order to maximize speed and minimize latency of data flow, both around the cluster, and between the cluster and data contained within databases and file systems.

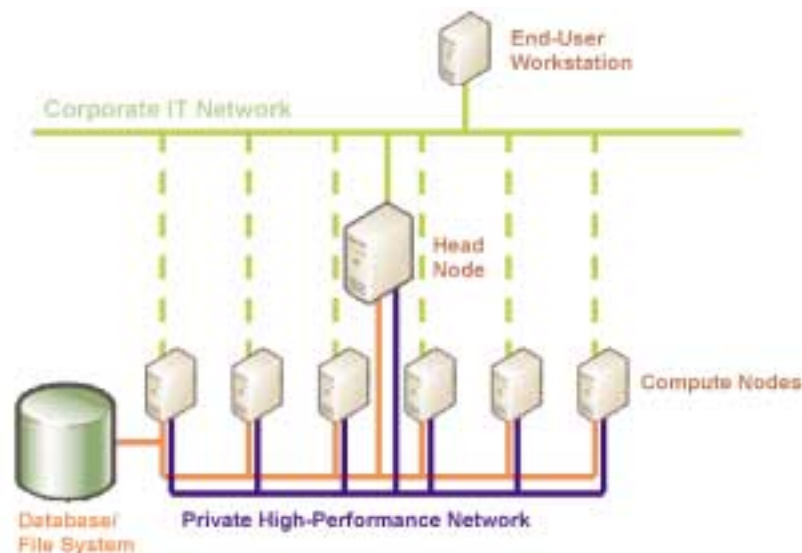


Figure (1) - An Example Cluster Configuration

A fluid grid

A more loosely-coupled approach to HPC is Grid Computing. Grid Computing originated from the excellent idea that there is enormous computer power that is going unutilized each and every day as most computers are not fully occupied most of the time. Think of all those trader workstations sitting idle every evening on the trading floor for example. Figure (2) illustrates an example Grid architecture, involving many different kinds of computers, located on many different network domains and locations, communicating with each other using technologies such as Web Services.

The number of computers present in the Grid may also change dynamically, even whilst a group of tasks are being undertaken as part of a single Job. Given the heterogeneous nature of the computers in a Grid, and that the membership of the Grid can fluctuate up and down, Grid computing solutions are necessarily based on a loosely-coupled architecture that can cope with such variability.

The grid vs. the cluster

As I said, the definition between the two is blurred and here are some of the understandable reasons why. Both Grid and Cluster Computing are forms of

distributed computing, where a Job is broken up into smaller parallel Tasks to get the Job done quicker. Both offer fault-tolerance in retrying and redistributing a Task should one or more members of the Cluster/Grid break down during the calculation Job. Furthermore, a cluster can operate as a node within a grid solution - there is a cluster of five servers illustrated as operating as a Compute Node in Figure (2), for example. Finally, a grid can be very successfully configured as a cluster - think of the Web Services communication of Figure (2) being located on the dedicated high performance networks illustrated in Figure (1).

Given that Grid solutions rely on non-dedicated, heterogeneous resources, they tend to be more suited to academia and research, where the computational load is very high but the time-frame for completion is not so critical. For example, take a look at <http://setiathome.berkeley.edu/> or <http://folding.stanford.edu/>. The clustering approach, with dedicated resources applied to each compute Job, is more suit-

ed to problems that need to be completed within reasonably predictable timeframes. As a result of this, and sometimes due to internal politics over resource sharing, dedicated clusters (and grids configured as clusters) currently dominate the usage of distributed computing within financial markets.

The pull vs. the push

Focusing on another matter of architecture, then Grid solutions are mostly based around a “pull” architecture where members of the Grid pull a new task for calculation down from the “management” layer of Grid software. This approach lets the members of the Grid themselves effectively decide what workload

they are capable of. This loosely-coupled approach is highly scalable and can respond dynamically to decreases and increases in members of the Grid.

In contrast, many cluster solutions are based around “push” architectures where the cluster management software (located in what is called a “head” node) schedules up workloads for the compute nodes of the cluster to perform. This

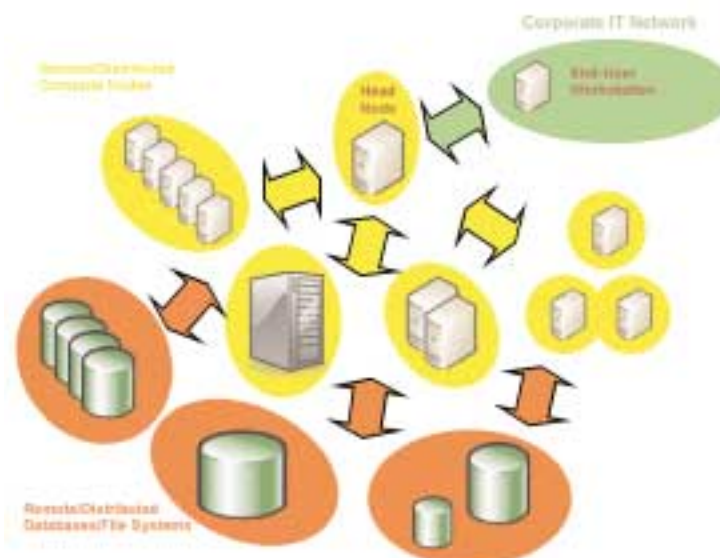


Figure (2) - Loosely-Coupled Grid Architectures

"push" approach can prove more efficient than "pull" in situations where the calculation tasks being undertaken are homogenous and behaviorally well-understood, enabling more efficient scheduling and distribution of the tasks comprising the job across the cluster.

The grid vs. the data

One issue for large clusters is that access to data needed by the Compute Nodes can become the bottleneck that effectively chokes the ability of the cluster to parallelize Tasks. In terms of Amdahl's Law, then limits on database access (probably caused by disk I/O) effectively increase s, the proportion of the Job that has to be done serially as each cluster node has to wait in line for data.

As a result, a number of companies such as Gemstone (www.gemstone.com), Gigaspaces (www.gigaspaces.com), Tangosol (www.tangosol.com) have developed high-performance, in-memory, object-caching solutions that sit in-between the compute nodes and the sources of data such as databases. This kind of architecture is illustrated in Figure (3) below - once again we are introduced to yet another acronym with this kind of system being known as a "Data Grid" or "Enterprise Data Fabric" (EDF).

A classic example within financial markets for the kind of object caching that would be done in this intermediate layer would be the building of a yield curve object to price a large portfolio of derivatives. A single yield curve is likely to be needed for pricing many different derivatives on many different nodes. Rather than rebuild the curve through accessing the database each time the yield curve is needed, the object cache takes an image of the yield curve object from the node it was first created on and puts it in the cache, making it available to other nodes as and when they need it and avoiding heavy database access and file I/O.

Linux OS vs. windows OS

Over 90 per cent of the HPC market as a whole

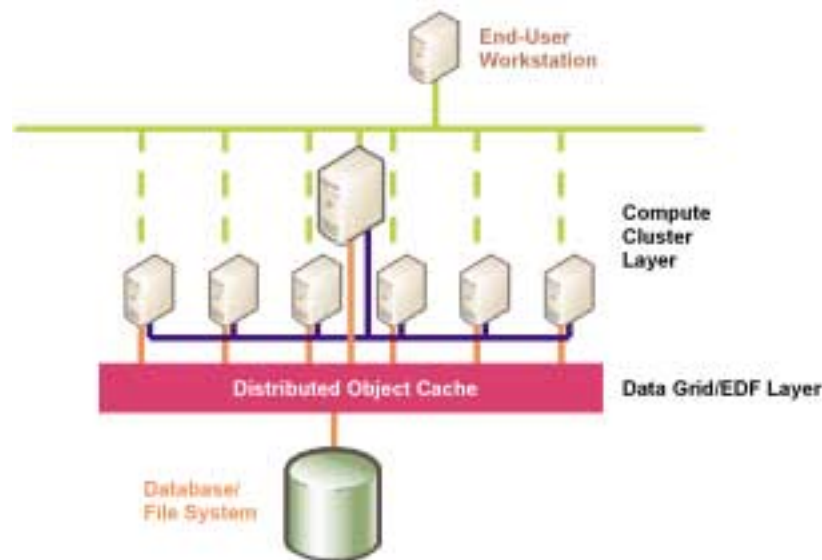


Figure 3 - Excel web access

runs its clustering software on the Linux Operating System (OS), with over a third of all Linux servers being used within HPC solutions. There are a number of factors behind these impressive statistics: HPC has a long association with academia; academia has long favored the Unix OS; Linux as the dominant Unix OS; and the academic and commercial growth of Open Source computing on Linux. For some addition background on Linux-based clustering, it is probably worth taking a look at the "Beowulf" site at www.beowulf.org.

Continuing on the Operating Systems theme, then the story for Microsoft Windows within HPC is historically very bleak; however, when focusing on capital markets there seems to be some better news for Microsoft and a bit of a geographic divide between United States and Europe. It would seem that the Windows OS is used and at least accepted within clustering solutions at many major UK and European investment banks, whereas the tendency in the US is to go for Linux.

In order to attack the dominant position of Linux as the operating system for clustering, then Microsoft has created Windows Server 2003, Compute Cluster Edition (CCE). This is a cut-down version of the Windows OS (set those compiler flags!) targeted specifically at HPC solu-

tions, and in this regard is aggressively priced by Microsoft to at least match, if not undercut, the cost of having Linux as the OS for clustering applications/infrastructure. Despite this aggressive pricing, then it will be interesting to see if the Linux vendors respond, and indeed whether even lower pricing is a key factor for success in this already low-cost market.

The blade vs. the pizza-pan

Hardware manufacturers like HP, Sun and IBM have been pushing high-density "Blade Servers" as the appropriate server platform for HPC Clustering. In a Blade Server, the Blade enclosure houses power, network connectivity and other services

in the interests of keeping each computer "Blade" small (for a specialist example, see www.egen-era.com). Blade Servers are a great idea, however some clients are moving back to rack-mounted servers (the type that look like pizza-pan boxes) networked together using technologies such as Gigabit Ethernet, Infiniband or Myrinet. This movement away from Blades has been due to Blade hardware being much too proprietary and effectively "locking-in" the client to a particular hardware manufacturer and being too expensive relative to more generic rack-mounted hardware.

AMD vs. Intel

AMD (www.amd.com) has been extremely successful in financial markets and clustering over the past few years with both its 32 bit and 64 bit processors. This has certainly grabbed the attention of Intel (www.intel.com). Intel has now pushed back with a new set of 64 bit chips that currently seem to put them back ahead in this continuing battle of processor performance. Additionally, both companies are pushing multi-core processor architectures, which effectively have the potential to put a "Grid on a Chip" (see www.intel.com/pressroom/archive/releases/20070204comp.htm).

Despite this fight back on performance by Intel, differences in register sizes and the paral-

lization of floating point calculations on multi-core architectures are causing slight numerical differences for the same calculations run on AMD and Intel 64-bit. Whilst Intel argues that these differences are both numerically and materially insignificant, it seems that some of the investment banks are still reluctant to switch to faster processors if it means reporting different valuation and risk numbers to the regulators! Maybe the underlying issue here is that we should all try to understand more about the numerical accuracy and stability of converting mathematics into floating point arithmetic?

Data synapse vs. platform computing

So far we have discussed the architectures, operating systems and hardware used in clustering, but what about the clustering software itself? The two dominant providers of clustering software to the investment banking community are Data Synapse (www.datasynapse.com) and Platform Computing (www.platform.com). Data Synapse has around two thirds of the current investment banking market with its Java-coded, pull-architecture GridServer product, with comparatively few clients outside of financial markets.

Symphony from Platform Computing is a C/C++ coded, push-architecture based solution that has around a quarter of the investment banking market, but with a much larger client base in the wider market given its historic origins in HPC for engineering applications. Both products are cross-platform compatible, and both use a Service Oriented Architecture (SOA - an overused term currently if ever there was one) approach to the serialization and distribution of Tasks around the cluster.

The Microsoft cluster

Microsoft's offering to the HPC market is Microsoft Compute Cluster Server 2003 (CCS, see www.microsoft.com/windowsserver2003/ccs). I have already mentioned the first part of CCS, which is the Compute Cluster Edition of the Windows OS. The second part of Microsoft's clustering solution is Microsoft Compute Cluster Pack (CPP), which runs on CCE and other variants of 64-bit Windows. This is the cluster management

software, the piece that competes with Data Synapse and Platform Computing's offerings.

Compute Cluster Pack has a push-architecture and like Symphony from Platform Computing this version is designed specifically for clustering and not grid computing. Unlike the dominant offerings above, CPP is batch-based, meaning that the Tasks run in the cluster have to be command-line executables. Additionally, whilst the Compute Nodes of CCP are fault tolerant, the Head Node of CPP is not clustered itself and so represents a single point of failure for each cluster. In this regard and others the initial offering from Microsoft is unsurprisingly less sophisticated than those from Platform or Data Synapse. For more sophisticated SOA-like remote invocation of objects/methods etc within the cluster then this will have to wait for version 2 or 3 of CPP, although one solution that already offers this approach specifically on the Windows OS is the grid solution from Digipede (www.digipede.com).

Microsoft's marketing plan for clustering is to use server-side Excel, Excel Services, as the obvious "Trojan-Horse" end-user application to get financial markets to begin to use Microsoft CCS. This approach also fits for what Microsoft sees as the fastest growing part of the HPC market, that of the "deskside" or "departmental" cluster of under say 64 nodes. Also, partner software companies will be encouraged to use CSS in their solutions. From here growth to the enterprise clustering market can be developed over the next few years (example, recent Top 500 Supercomputing benchmark using Microsoft CCS), and Microsoft will continue to emphasize the out-of-the box nature of their clustering offering, leveraging existing Windows security and system management technology.

Easier clustering for all?

The main aim for Microsoft is to initially make clusters easier to set up, easier to use and hence more widespread. Okay, putting aside the fact that Microsoft do not tend to do things just for the greater good of society (which companies do?), the entry of Microsoft to this market sounds like a good thing, in that competitive threat should drive all clustering software vendors to

deliver solutions that are easier for quants to manage and apply to ever-larger pricing and data management problems.

To finish - the grid and the virtual dream

So my version of High Performance Computing in financial markets is done and I think the market is an interesting one. Taking a more philosophical and blue skies approach for a moment, the dream for "the Grid" is much wider than that which I have described above, with almost infinite computing and storage power available on demand whenever and wherever it needs to be applied (see <http://gridcafe.web.cern.ch/gridcafe> for example). This computing dream of global computing and storage virtualization is not going to be a reality any time just yet, but the pieces required are coming together. Maybe "Grid Capacity Derivatives" are not too long away from becoming a mainstream financial product...

For Xenomorph's take on clustering see: <http://www.xenomorph.com/wilmott>

My thanks to:

Irina Taran, *Senior Developer, Xenomorph*
John Barr, *Grid Computing Architect, Intel*
Antonio Zurlo, *Technology Specialist HPC, Microsoft*