PROCEEDINGS OF SCIENCE



# Financial Services GRiD Virtualisation for Increased Business Performance and Lower TCO

**Labro Dimitriou**
*Platform Computing*
*Industry Manager*
*FS GlobalOperations*
*245 Park Avenue - 39 Floor*
*New York, NY*
*10167 USA*
*E-mail:* labro@platform.com

**Antonio Zurlo**
*Platform Computing*
*Director of Services*
*FS GlobalOperations*
*245 Park Avenue - 39 Floor*
*New York, NY*
*10167 USA*
*E-mail:* azurlo@platform.com

*Grid Technology for Financial Modeling and Simulation*
*February 3/4, 2006 – Palermo, (Italy)*

**Abstract**

Financial Services companies are increasingly faced with lower margins, competitive pressures, and new regulatory requirements. More so than ever, IT decisions are scrutinized for the business value they add, the potential ROI to the enterprise, and the Total Cost of Ownership (TCO). To be competitive in a global world, FS are creating ever more complex products for managing currency, interest rates, credit, and other exposures. This presents a real challenge.

Structured products with complex payoff structures, interest rate derivatives requiring stochastic Monte Carlo, and VaR recalculations challenge the current computing model and put tension on already stretched budgets. Traditional silo-ed computing infrastructures result in a “cause-effectum” problem. Scalability and peak performance can be addressed only with over provisioning. This results in a dual problem: idling of expensive IT infrastructure when not in use and suboptimal utilization of IT capital.

Through this case study, the paper illustrates benefits central to a GRiD virtualization implementation. Solutions to the following business and technical dilemmas are provide:

- How to provide mission critical reliability at commodity prices
- How business flexibility can be supported without costly support services
- How an enterprise can share computing resources among lines of business while guaranteeing service level agreements
- How new application architectures and legacy systems can simultaneously be supported on the shared infrastructure.
- How lines of business can avoid provisioning systems for peak usage and use compute resource on demand
- How disaster recovery and business continuity can be a working asset rather than an idle expense

**Virtualization**

Central to this case study is the notion of virtualization. Virtualization is hardly a new term; it is arguably the holy grail of computing. The term goes back to the early days of computing when scientists realized that people couldn’t possibly afford to have one program at a time running on those expensive machines called computers. Furthermore, the physical memory couldn’t be large enough for one-to-one mapping of a program. This realization facilitated the genesis of Multiple Virtual Storage (MVS) operating systems and Time Sharing Option

(TSO) interfaces. This started the first era of virtualization. In the 60's and 70's, more operating systems and noticeable UNIX continued the effort of decoupling the hardware architecture from the operating environment.

The virtualization wave continued with the proliferation of emulators – a complete operating environment executing on top of host operating systems. In many cases, emulating non-physical/existing environments or very different hosts, like x86 emulators, were hosted on Mac OS. In 1972 a rather strange book came out of XEROX, Palo Alto: Smalltalk-72. The book signaled the rise of the virtual machines. Smalltalk introduced virtualization at two distinct levels: macro and micro.

At the macro level, a program did not execute directly on the operating system but ran on a so-called virtual machine (VM). The VM provided an abstraction to the OS and the underlying machine architecture. In essence the VM was the virtual operating system. The benefits were clear: develop programs once and run them everywhere. At the micro level, Smalltalk provided a new programming paradigm using just a few simple and yet powerful constructs: object and messages that indirectly operated or queried the object. The programmer no longer had to operate directly on data and develop graphics by turning bits on and off and poking at memory registers. Object oriented programming decoupled the physical/model layer from the programming layer. The new programming construct allowed the programmers to create virtual environments of real-world situations like accounting systems, war game simulations, and microchip designs not yet physically developed. The objective was clear: deliver to the user the experience of owning a virtual computer, no matter how many simultaneous users were using the computer at a time. In other words, partition one computer unit into many virtual units.

As the proliferation of distributed computing continued with Java and .net, in-expensive farms of computers and GRiDs started replacing large monolithic mainframes or CPU bound SMP machines. A new virtualization requirement emerged: ability to unite a number of computers as a one virtual larger computer. Today, this requirement is essential for distributing compute intense applications to grids.

**Platform's GRiD Virtualization Software**

Platform's software is truly a distributed operating system that presents all compute resources as a one large compute fabric but also creates logical virtual partitions that can be dedicated to processes, users, or Lines Of Business, (LOB). The same way that MVS and TSO allowed simultaneous users to share the mainframe, Platform's GRiD virtualization software allows LOBs to share the enterprise compute fabric. Key to the offering is the following feature: logical

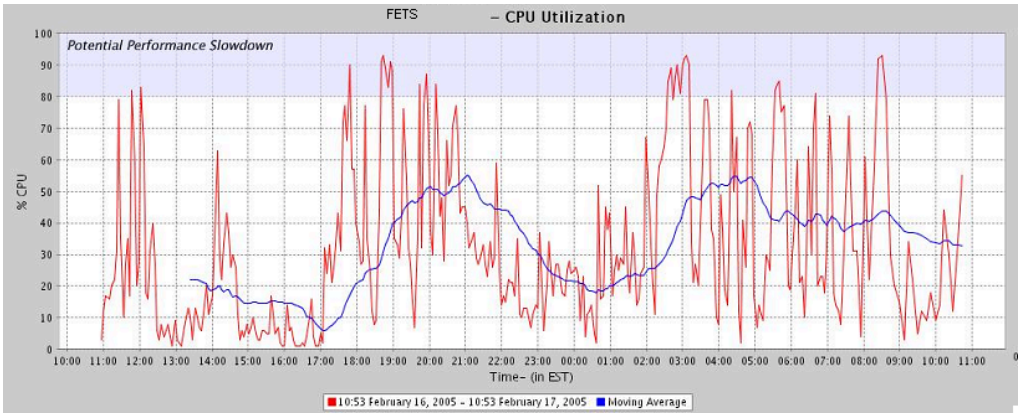
partitions that can shrink and grow dynamically subject to policy and work load characteristics. This unique ability to "lend and borrow" amongst partitions ensures that critical workloads have the compute power to complete on time while IT achieve optimal compute utilization.

**The challenge of staying profitable**

Decimalization and product maturity made profits and margins practically disappear. Feeling the pressure, a major North American Investment Bank had to substantially increase trading volumes to stay profitable. This presented a major operational challenge for the IT group supporting the LOB. Two key applications had to provide the scalability need while improving reliability and minimizing risk of failure. At the same time, the total cost of ownership had to be such to ensure a positive ROI.

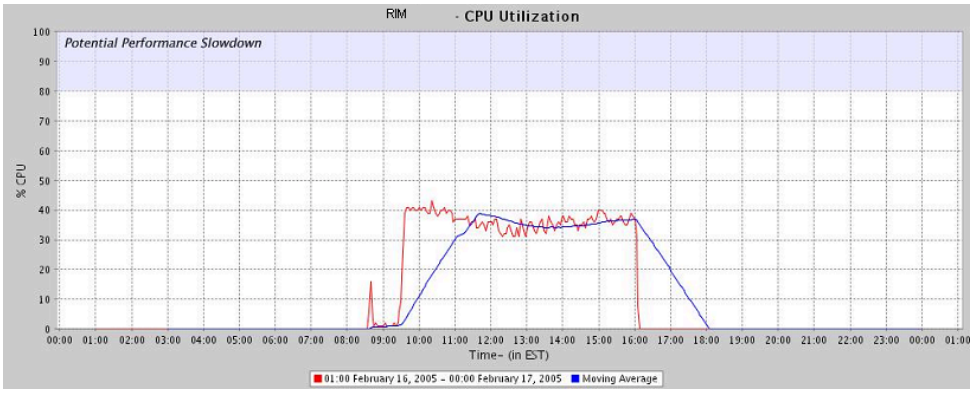
The first two trading applications were: a Foreign Exchange Trading System (FETS) running on leased Cray Computers and Risk Management (RIM) running on infrastructure owned by the LOB. Each of the applications presented a different set of operational challenges.

Challenge #1: Different run-time characteristics. FETS had bursts of computing nearly 24x7 with a small window of down time between NY closing time and Japan's trading desk starting.




[Figure 1]

Challenge #2: RIM was a batch application requiring sustained compute power during scheduled time [figure 2].




[Figure 2]

Challenge #3: FETS was already distributed while RIM was not. FETS was critical during trading operations while RIM had to complete execution within in a small window that left no time for errors and recovery. About 300 FETS clients would run 3000 to 4000 jobs daily using two leased CRAYS. The business unit was looking to a 30% growth the first year and doubling the business in year two.

**Meeting the challenge with GRiD Virtualization software**

Keeping status quo was not an option. To meet the challenge, the development group looked into a GRiD solution provided by Platform computing. Because FETS was distributed already, it was chosen to be the first application to migrate to the new architecture. A farm of 84 LINUX boxes provided the initial compute power at an initial cost of 30% less than the Crays.

Since fault tolerance was inherent to Platform's software, business continuity planning requirements were easily met. Platform's architecture facilitates the master/slave design pattern, common to distributed computing. Master nodes, called management nodes, keep track of state, distribute tasks, and manage failover of slave nodes, which execute computations or work loads. Since all the state related information was kept at the management nodes running on site, work was distributed to disaster recovery sites for extra capacity.

Within a very short time, IT was comfortable with FETS providing mission-critical performance and meeting enterprise IT robustness requirements. Most importantly, users got their services level agreements while lowering total cost of ownership. It was now time to tackle the RIM migration.

RIM was running on antiquated dual boxes with proprietary technology and PVM middleware for distribution. RIM was retrofitted to take advantage of

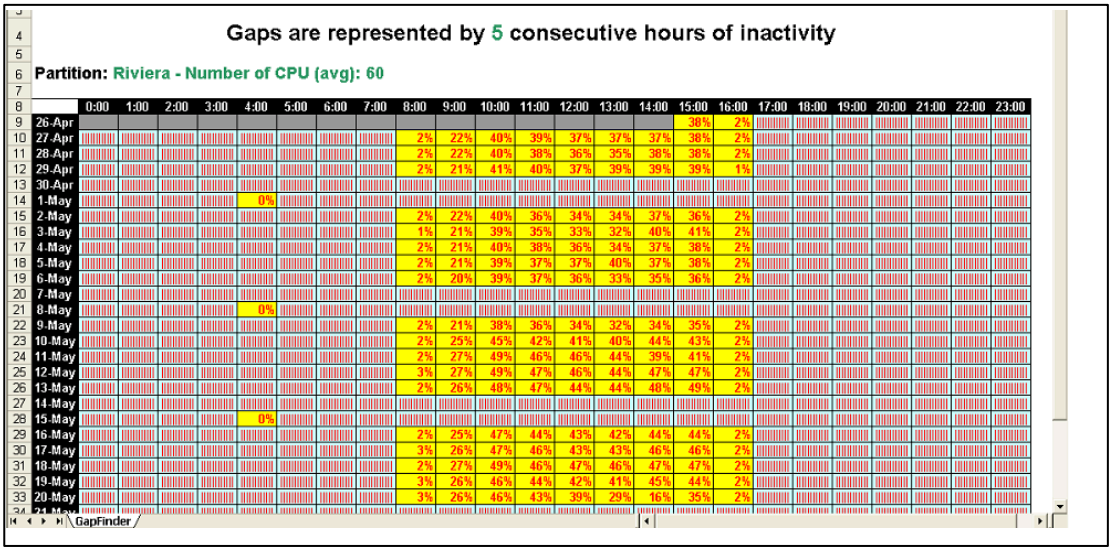
Platform's Service Oriented Architecture (SOA) framework. A farm of 64 LINUX boxes provided the initial compute power at an initial 300% speed-up.

The following year, the business grew twice as much as initially projected, to a stunning 200%! Overgrowth is typical to GRiD architectures. Having the compute power available makes end users strive for higher accuracy and faster results.

**Sharing the power of GRiD with Virtualization Software**

Within a short time it became evident that GRiD software was able to meet the original requirements and provided and invaluable quality of service - horizontal scalability. It could now add on-demand white-label commodity hardware at the best price/performance offering. The group was now ready to conquer the next frontier - sharing the compute fabric amongst multiple applications based on SLAs.

Platform's chargeback accounting service provided the supporting framework. The two GRiDs were merged together to form one virtual compute GRiD. Using the utilization logs from the two applications, IT determined the optimal 24x7 compute requirements. It was then used to virtually divide the compute fabric into two partitions. Each partition was mapped back to the two applications. Finally, IT turned the compute requirements into a SLA using a powerful rules base input. This completed the set-up and configuration.




[Figure 3]

Equipped with two successful GRiD implementations and leveraging Platform's utilization map [figure 3] together with the chargeback accounting, IT was able to build business cases for bringing another 3 applications in the second year and grow the GRiD to 1000 CPUs.

## Conclusions

This paper, describes how a financial services organization implemented Platform's GRiD Virtualization software to address the growing challenges of a Foreign Exchange and Equities LOB for a large North American Investment Bank.

The implementation started with one application and 84 boxes and grew to an enterprise GRID with over 1000 CPUs hosting 5 applications the second year. Today, the enterprise hosts over 12 applications with thousands of CPUs. Some of the applications achieved over 300% speedup and initial TCO was slashed to nearly one third of the previous cost.

Using Platform's chargeback accounting system, the organization was able to identify optimum execution loads for hosting multiple applications in virtual partitions. Lending-and-borrowing of the virtual partitions increased CPU utilization by 60%. By adding five applications, the cost for an application to operate per hour per CPU came down from 1 dollar to 60 cents.
At the same time, enterprise IT was able to seamlessly achieve horizontal scalability in a fault tolerant environment and leverage and comply with business continuity initiatives while harnessing extra computational power.

## References

[1] TBA