Paper 110-2013

Managing and Analyzing Financial Risk on Big Data with SAS® High-Performance Risk and SAS® Visual Analytics

Cary Orange, Stacey Christian and Donald Erdman,

SAS Institute Inc., Cary, NC

ABSTRACT

SAS High-Performance Risk™ is a distributed in-memory analysis engine for financial risk used to compute non-linear risk calculations such as Value at Risk (VaR). Traditional solutions for evaluating risk in a portfolio have suffered from obvious issues of scale. Millions of financial positions valued at thousands of simulated market states quickly yield billions of fine-grained results. Until now, reducing aggregation and reporting flexibility has been accepted as an unfortunate but necessary trade-off to obtain basic results. However, recent advances in both hardware and software for distributed computing make it practical to rethink the accepted architecture of risk solutions. This paper presents the fundamentally new approach to risk analytics used by SAS High-Performance Risk™ and SAS Visual Analytics™.

INTRODUCTION

Financial organizations are exposed to large financial risks due to movements in the markets. It is important for these organizations to analyze this risk in a timely fashion so that they can understand changes in their exposure over time. Unfortunately the analysis process is very computationally intensive and the value of a risk analysis is, in many ways, a perishable item. A retrospective understanding of a loss is clearly inferior to an advance notification.

The use of in-memory distributed architecture brings several benefits to a risk analysis engine. Shorter computation time is realized by distributing the computations to hundreds or thousands of computers. Retention of fine-grain results in distributed memory let users interactively explore results in an ad hoc fashion in near real time. Furthermore, it becomes far easier to scale results as the complexity of risk projects grow.

With the availability of low-cost, parallel processing platforms and the increased demand for 'big data' analytics, users of SAS High-Performance Risk™ can effectively evaluate financial risk in large portfolios with speed and flexibility simply not possible before. This paper presents a few of the architectural features of the SAS High-Performance Risk™ solution and explains how the financial risk community can benefit by using its features.

ANALYZING FINCANCIAL RISK

Most risk analysis projects involve the following steps:

1. Setting up the portfolio
2. Simulating the markets
3. Pricing the portfolio against the simulated markets
4. Aggregating results
5. Producing reports

Generally speaking traditional risk engines handle these steps in a linear fashion and require pre-planning as to how results will be produced. Firm-wide portfolios of up to 100 million positions are not uncommon. Each position is evaluated across thousands of simulated market states at numerous horizons, resulting in numerous statistics on hundreds of output variables. Reporting requirements demand aggregation of these results at varying levels of an organization's hierarchical structure. This result set is so enormous that it cannot possibly be held in memory on a single computer. Some traditional risk engines attempt to mitigate computational expense by distributing calculations of individual positions. While helpful at speeding generation of low-level results, such an approach does not address serious aggregation problems, explained in more detail below.

SAS High-Performance Risk™ addresses these issues by giving analysts the ability to perform steps in a non-linear fashion while keeping intermediate results in memory on a massive gridded network. SAS High-Performance Risk™

minimizes the amount of data by keeping all pricing results in memory on the grid machines where the computations occurred. The aggregation and statistical calculations can be performed in an ad-hoc fashion in near real time.

HISTORICAL RISK ANALYSIS

Risk engines have been evolving over time. Early risk engines used a single machine to price the portfolio and write the results to a single disk. It was recommended that the server machine have abundant available system resources (i.e., CPU, memory and disk). Reports were generated from these pricing results. In the last decade, risk engines started using multiple machines to price the portfolio in parallel. The I/O operations can be very intensive depending on the size of the project settings and data. Some users of traditional risk configure multiple computer servers to divide up the work and perform some parallelization on separable portions of the project. As mentioned above, the results were still written to a file on a shared disk and inevitably, these files had to be combined by a single machine in order to aggregate and produce the final statistical results. It would have required a huge amount of available memory on this central aggregation server to accommodate the work. When the user intended to report on many different combinations of cross classification variables, the server would hit a huge capacity limitation for contention of available disk space to store the results. The following diagram illustrates the bottleneck problem that occurred when attempting to combine pricing results for aggregation and write the final results to disk.

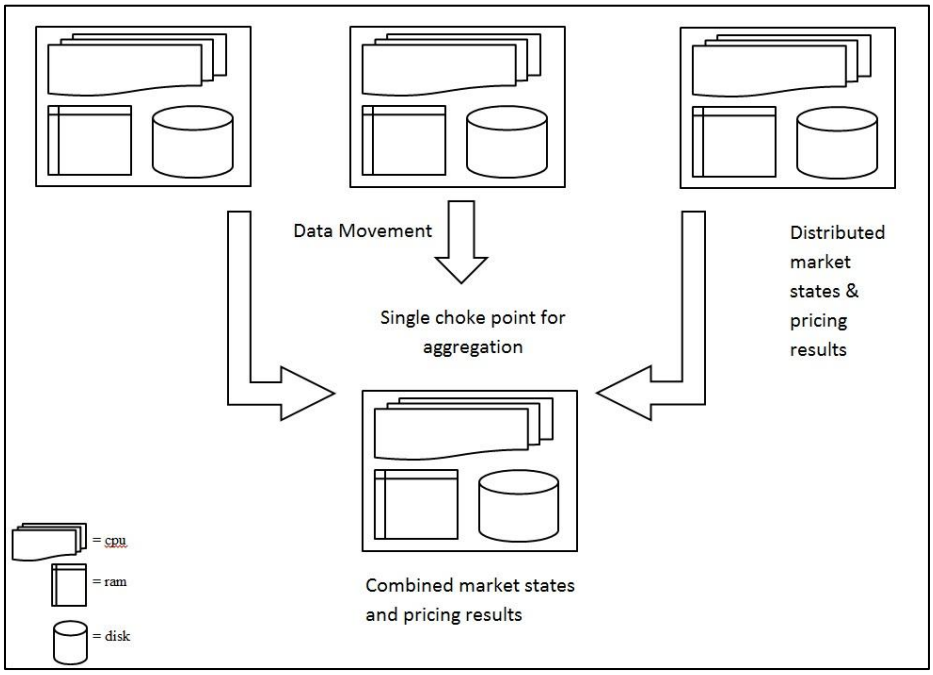


Figure 1. Historical Risk Engine

HIGH PERFORMANCE RISK ANALYSIS

The following diagram illustrates the separation of data between server and the distributed grid machines. Gridded risk analysis introduces an approach where the project parameters are distributed across multiple grid machines and the intermediate results stay in memory on those grid machines. The use of large numbers of independent grid machines not only provides massive computing power to greatly reduce the initial processing time to simulate and price, it also provides enough actual resident memory to hold all of the initial pricing results in memory on the grid.

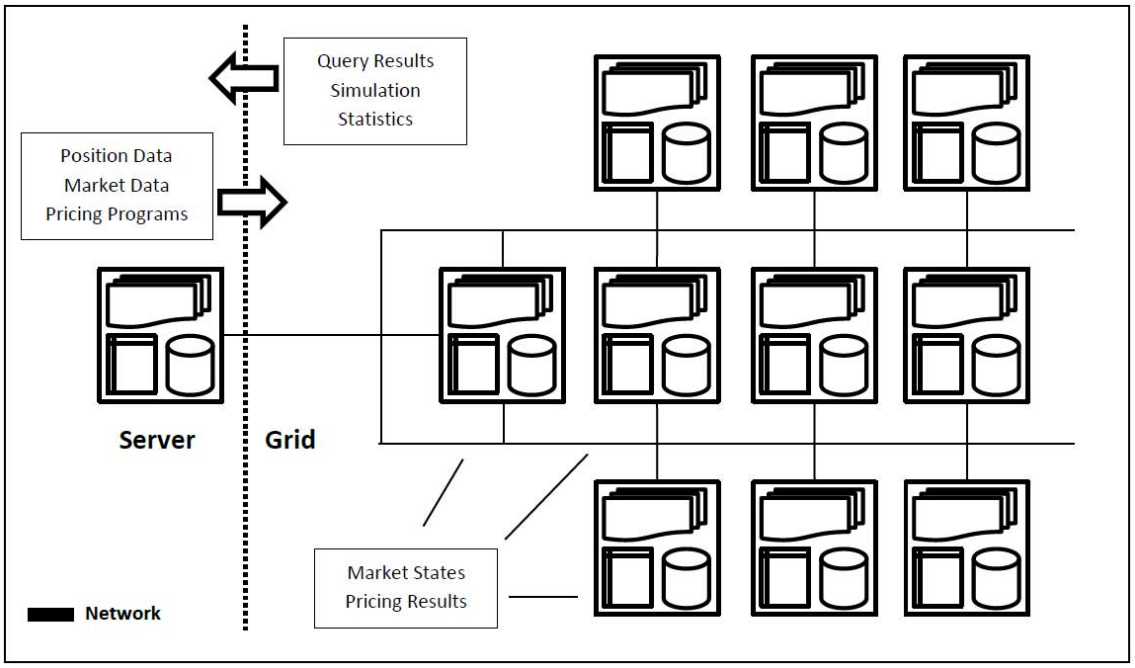


Figure 2. Resource Utilization for Server and Distributed Grid Machine Configuration

GRID COMPUTING

Before discussing how the distributed risk engine works, it is important to understand the platform architecture of the distributed computing network. The grid, as it is called, is viewed as a general purpose distributed computing environment. Typically, the grid computer, or grid node, is a multi-core hyper-threaded 64-bit CPU with at least 32GB of DRAM, 512GB of hard disk space running a standard distribution of Linux. There is nothing special about the grid machine that distinguishes it from other general purpose computing environments. The grid comprises a set of these machines. The collection of machines uses a high-speed mesh network interface for inter-machine communication. There is usually no use of shared disk space on the grid in order to reduce delays due to network traffic. For this reason, each grid machine reads and writes to its own local disk. One machine is dedicated as the head grid node. From a hardware standpoint there is nothing that distinguishes the head grid node from the other grid machines. Software applications, however, might use the head grid node for special operations depending on the application.

Applications that do not use a grid are considered stand-alone in that they run entirely on a single dedicated server machine. Applications that use grid computing set up application extensions to run on one machine referred to as the root node and any number of other machines referred to as worker nodes. The worker nodes are the machines that actually perform the majority of the work. The head node coordinates the worker nodes and serves as a mediator between the grid application running on the server and the worker nodes. At the completion of a job, the processes on each of the machines ends and control is returned to the server application.

Any number of server machines can share the grid. There can be many applications running jobs on the grid at any time independent of one another. There should be no concept of one job having priority over another job as all have equal access to the machine resources (for example, CPU, memory, and disk).

When designed efficiently, grid applications can be viewed as if they are running on one very large machine. By carefully delegating the work and dividing up the results across the machines for storage, the problem of addressing the memory and disk boundaries can be solved.

GRIDDED RISK ANALYSIS

With the knowledge that worker nodes are available to do work and store information in memory, the next thing to understand is the breakdown of work and results across the worker nodes. In the simplest cases, distributed risk analysis involves distributing the simulated market states so that each worker node is responsible for pricing the entire portfolio on its subset of market states. All the intermediate results are kept on the worker nodes.

There are four dimensions used to describe intermediate results: positions, horizons, market states, and prices. The following diagram attempts to illustrate the first three of these dimensions in a 3-D fashion. The X-axis represents the horizons (for example, today, tomorrow, the next day, and so on). The Y-axis represents the individual positions in the portfolio. The Z-axis represents the simulated market states.

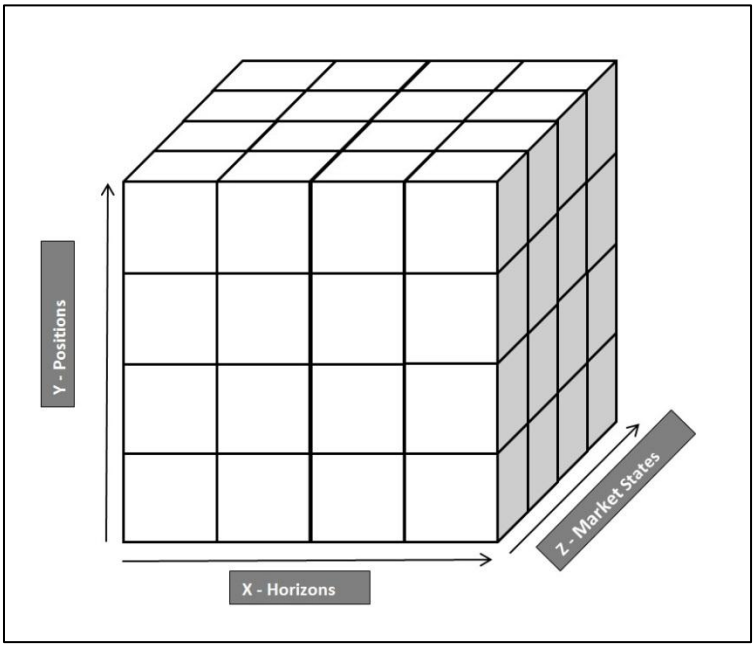


Figure 3. Multi-Dimensional View of Intermediate Pricing Results on the Grid

The fourth dimension, prices, is the various output variables that are evaluated (or priced) and can be represented by a set of these cubes (following figure). Each of the individual 3-D cubes holds a unique output variable (such as price, return, P/L, exposure, etc.). There could be hundreds of different output variables computed for each position across each horizon and each market state.

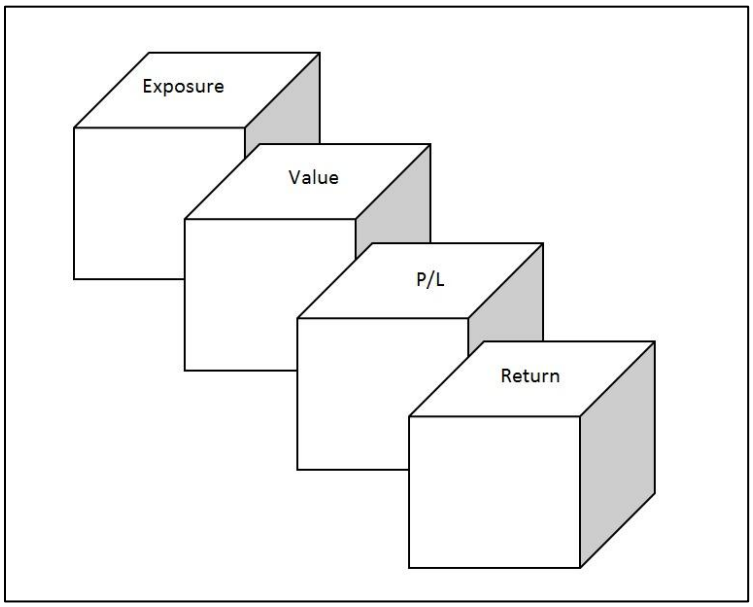


Figure 4: Output Variables: the Fourth Dimension

Market state data can be generated by each of the worker nodes. The following diagram illustrates the distribution of 100,000 market states across four worker nodes. In this example, each worker node prices each position at every horizon for its set of distributed market states (25% of total number of market states).

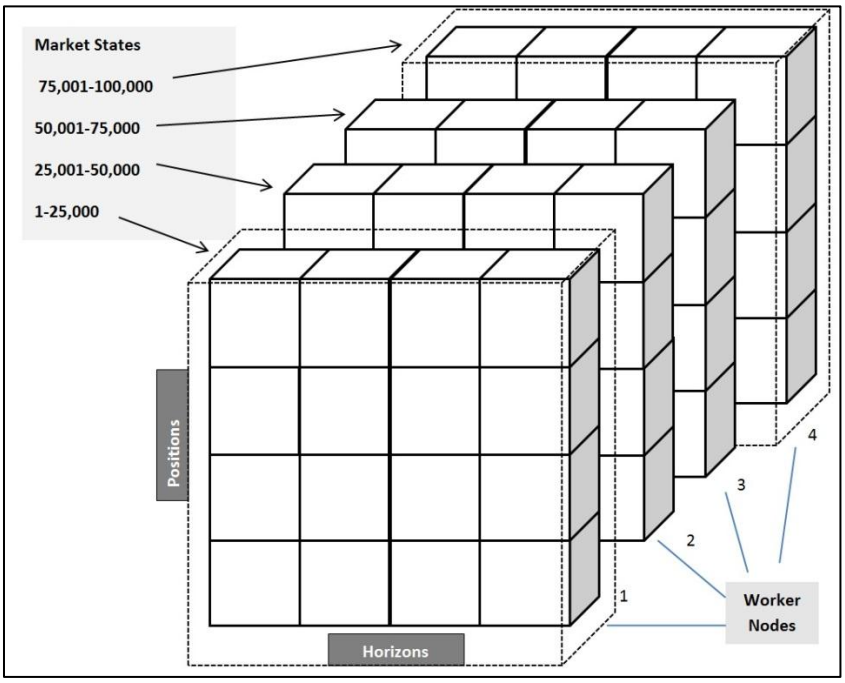


Figure 5. Multi-Dimensional View of Distributed Market States across 4 Grid Nodes

In some cases, the user may want to price a large portfolio over a small number of market states, as when performing mark-to-market or computing a few stress tests (i.e., what-if scenarios). In this case, SAS High-Performance Risk™ will distribute the positions evenly to each worker node instead of market states. All of the worker nodes then price their portion of the distributed positions at all the specified market states. The following diagram illustrates the distribution of positions to four worker nodes. In this example, each worker node receives 25% of the positions.

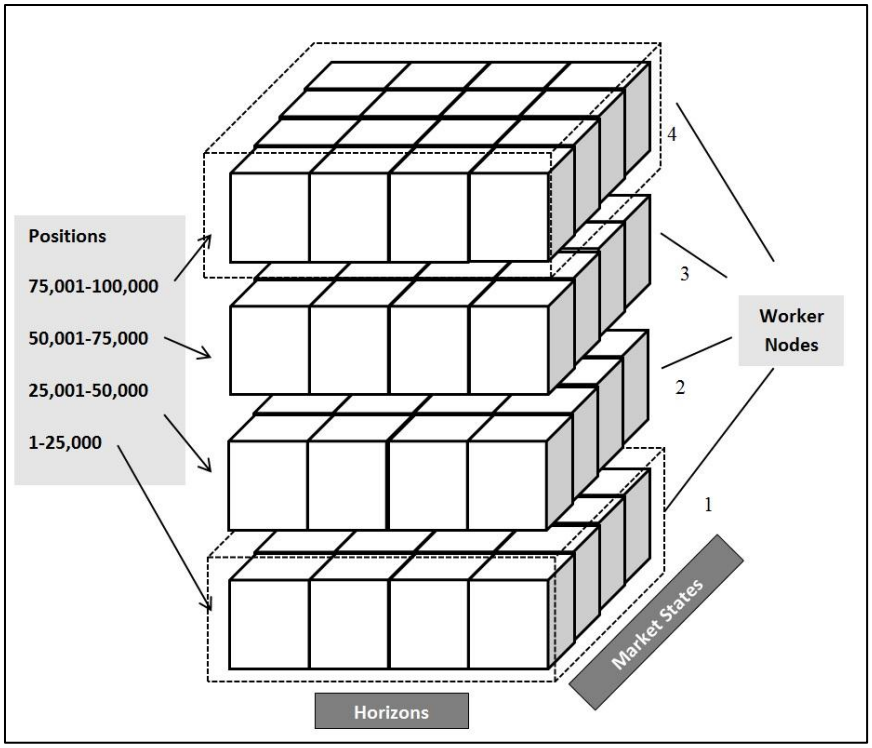


Figure 6. Multi-Dimensional View of Distributed Positions across 4 Grid Nodes

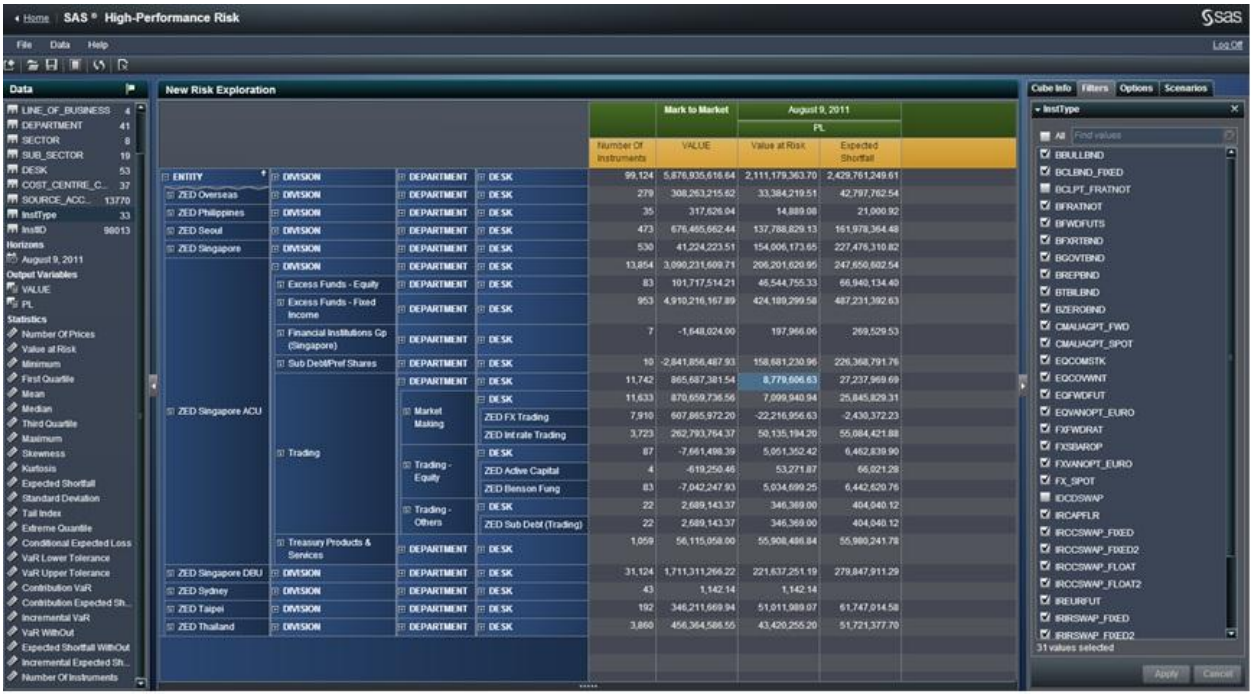
This in-memory distributed pricing methodology has proven extremely powerful in reducing the total run-time to simulate market states and price extremely large financial risk portfolios. Because of this gain in speed, users are better able to evaluate much more complicated instruments as well. SAS High-Performance Risk™ allows the user to define almost any imaginable pricing algorithm in the SAS programming language as well as hook up their own proprietary C-language modules.

EXPLORING RESULTS

The really amazing power of the High Performance Risk Engine comes into play when exploring results. As already stated SAS High-Performance Risk™ works in two stages. The first stage, simulating market states and pricing positions, takes advantage of a highly parallel gridded network to perform extremely fast computations. At this point the risk results (known as Risk Cubes) reside on the gridded network waiting for queries. No queries, aggregations or final statistical calculations are precomputed in the first stage. Instead, whenever a user asks for some result either through a batch SAS query or through the visual explorer, SAS High-Performance Risk™ can use the massively parallel network to compute that query result instantaneously. As the user moves around the 'exploration' they can rearrange the hierarchy in any fashion, apply arbitrary filters and perform post-process calculations at will. And all this is done in real-time.

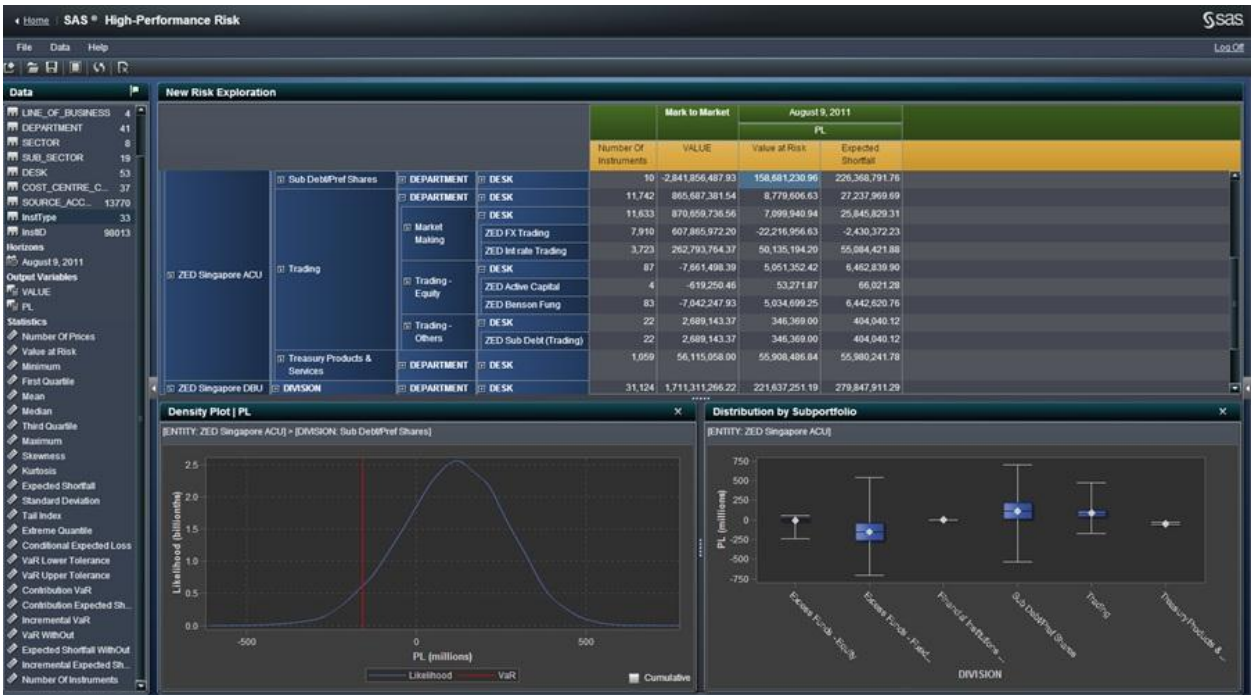
The SAS High-Performance Risk Explorer is an interactive, web-based visual explorer that uses the SAS Visual Analytics™ components. It provides an interface to perform exploratory data analysis on risk cubes in an easy to understand visual form.

In the screenshot below, the user has opened a Risk Cube in the HP Risk Explorer and is presented with cross table at the top-level aggregation for their portfolio. Their portfolio output variable 'Value' is displayed for mark-to-market (current market value) and a single horizon (August 9). Numerous statistics are available (VaR, Median, FirstQuartile, etc.) that the user can add by simply dragging them onto the cross table. Classification variables (Entity, Division, etc.) can also be dragged onto the cross table to drill down into the cube, thus creating sub-totals, and statistics for each of the contained sub-portfolios.



Display 1. Example of Cross Table in SAS High-Performance Risk™ Explorer

In the screenshot below, the user has selected two graphs for a Risk Cube in the HP Risk Explorer from the highlighted 'Value at Risk' statistic on one of the sub-portfolios. The density plot graphs all of the PL values in the simulation analysis with a vertical reference marker drawn at the risk PL value. The distribution plot graphs the PL values at each 'Entity' sub-portfolio level.



Display 2. Example of Graph Data in SAS High-Performance Risk™ Explorer

A major advantage of the exploration tool is that multiple users can independently explore the pricing results for the same cube at the same time by sharing memory resources. When the exploration is opened, a new process is initiated on the grid for each user to explore results. The process runs as long as the user session stays open and the user is actively exploring data. Each user process running on the grid uses both private memory and shared memory. Private memory is dedicated to each user process to handle his specific query requests. Shared memory, which is not drawn from individual process memory, is used to hold the risk cube's intermediate results. This shared memory is annexed, or *mapped*, into local addresses in each process' own address space for fast access. The HP Risk engine utilizes this memory mapping technology to enable multiple users to view data from the same risk cube. Actions performed by each user that operate on the shared data stores calculations in the private memory.

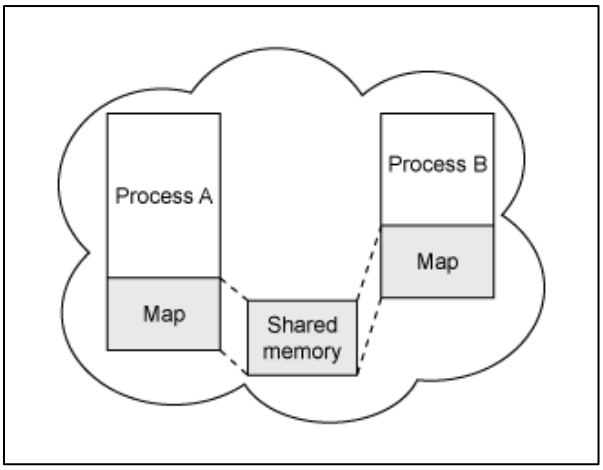


Figure 7. Multiple User Sessions Sharing Data in the same Risk Cube

In this manner the cube can be loaded by hundreds of different clients without actually using 100X the memory on each grid node.

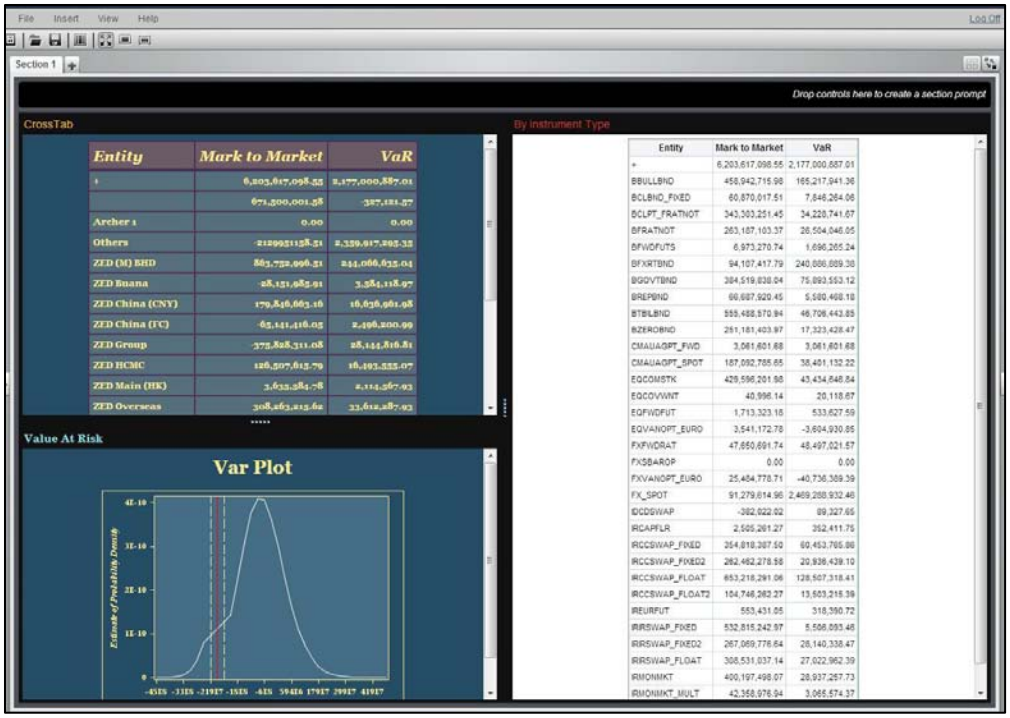
During a risk exploration session, any number of query operations can be performed. In tandem, the explorer, server and worker nodes work together in real time to re-organize position results, re-aggregate sub-portfolio totals, and re-calculate statistics in memory. The session remains open until the user saves and/or closes the application. Once the user has completed a set of query operations in the explorer, the exploration's state can be saved for later use. The exploration can be re-opened by the same user or another user to view the results again with the filters, cross-classifications variables, and any graphs created saved in the exploration environment.

During a risk exploration session, the worker nodes retain information about the last query operation and reuse this cached query to avoid repeating duplicate operations and calculations on the data in memory. This cache also provides an extremely fast paging mechanism for the case when the particular cube being viewed is too large (has too many rows) to display in the window.

SHARING RESULTS

SAS High-Performance Risk™ provides capabilities to write out various types of result sets to the LASR Analytic Server™. One of the output data sets that traditional SAS Risk Dimensions™ users like to see is the ALLPRICE data set. This data set contains output prices for each position for each market state at each horizon. It can be so large that it would fill up the hard disk space on a single-machine server. The time it would take to move the output of this data set to the server would also result in a significant burden on the network. SAS High-Performance Risk™, however, can output these extremely large data sets in fairly reasonable time by writing them in a distributed fashion on the grid. Each worker node writes portion to his local disk space. They can then be explored and/or manipulated with other SAS exploratory applications like the Visual Analytics Explorer™ and the Visual Analytics Designer™.

SAS High-Performance Risk™ data can be combined with LASR Analytic Server™ using the Visual Analytics Designer™ and the stored process server. Here is an example of SAS High-Performance Risk™ output in Visual Analytics Designer™.



Display 3. Example Output of Risk Data with Visual Analytics Designer™

VIRTUAL CUBES

The concept of an in-memory, distributed, analytical risk cube is extended by SAS High-Performance Risk™ to include the concept of a joint 'virtual' cube. A virtual cube is a risk cube made up of any number of sub-risk cubes. To the user, this virtual cube appears to be a single combined cube. To create the virtual cube, SAS High-Performance Risk™ does not copy the sub-cubes into a new, larger cube. Instead the virtual cube is created from metadata describing the location of the sub-cubes as well as classification data that defines the sub-cube relationship to one another. This makes the virtual cube fast to create and requires very little additional resources.

The creation of sub-cubes that comprise a virtual cube can be created independently at different dates or created in parallel. This provides the user additional flexibility to perform the following tasks:

1. **Incremental VaR**: Create one large sub-cube that represents the close of day holdings for a company. During the day, sub-cube for changes to the portfolio can be created and joined with the bigger sub-cube. Because the sub-cubes are small compared to the initial cube, they can be rapidly created. A virtual cube can be rapidly produced that represents the new state of the portfolio incrementally throughout the day.

2. **Combining Analyses from Different Company Divisions**: Companies desire an enterprise view of their risk but typically compute risk along lines of business. A typical example would be the retail, wholesale and trading units of a company. Using the virtual cube capabilities a company could create these two parts separately and join them into a single source, enterprise view of the company.

3. **Monitor Spikes in Risk**: Companies are concerned about big changes in the risks they are exposed to over time. There is a desire to be able to open a single cube be alerted to big day to day, week to week, or month to month changes to VaR. By joining (i.e., side-by-side comparing) a series of cubes created over time the user can then easily drill and monitor risk in all parts of their portfolio.

CONCLUSION

This paper presents numerous advantages for using the SAS High-Performance Risk™ solution. SAS High-

Performance Risk™ offers a scalable, flexible, and attractive approach to meet the need for industries seeking to solve financial risk analysis using big data. As the financial industry grows to take advantage of the availability and uses of grid computing, the user community will realize the growing potential and new applications for distributed analysis and results exploration using SAS High-Performance Risk™. Users will no longer expect to simply view static reports at the end of the day. Instead they want to visually explore their data in real time. This latest release of SAS High-Performance Risk™ lets them explore their results interactively in a way that has never been done before. They can create hierarchical risk aggregations on the fly. They can display big data in all kinds of really cool interactive graphs. They can dynamically join any number of risk cubes together at will. This ad-hoc visualization explorer represents a paradigm shift in how the financial industry will view risk.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Name: Cary Orange, Stacey Christian and Donald Erdman
Enterprise: SAS Institute Inc.
Address: 500 SAS Campus Drive
City, State ZIP: Cary, NC 27513
E-mail: Cary.Orange@sas.com, Stacey.Christian@sas.com, Donald.Erdman@sas.com
Web: http://www.sas.com/industry/financial-services/high-performance-risk.html

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.