

Paper 303-2011

The Architecture of SAS® High-Performance Markdown Optimization

Henry Bequet, SAS Institute Inc., Cary, NC

ABSTRACT

This paper presents an overview of the architecture of the SAS® High-Performance Markdown Optimization solution. We begin with a brief description of the many-core chip architecture and explain why it is relevant to software development. We then describe SAS High-Performance Markdown Optimization and present the main design pattern of its parallel architecture. We also discuss a couple of deployment scenarios and show that this architecture has the flexibility of handling a broad spectrum of situations. We finally review the performance that we have measured in those deployment scenarios.

INTRODUCTION

In the coming decade, the focus of the processor industry will be in the development of chips with many cores (the many-core architecture). For example, Intel is talking about tens or even hundreds of cores per chip (Garver & Crepps, 2009). This is a paradigm shift for software developers. Concretely, it means that software developers must now architect applications to take advantage of the many-core era. As we show in the remainder of this paper, SAS High-Performance Markdown Optimization takes advantage of what the many-core chips have to offer today and is architected to leverage what they will offer in the coming decade. But before discussing where we are going, we must first acknowledge where we are coming from.

THE FREE LUNCH IS OVER

As early as 2005 (Sutter, 2005), software developers realized that their “free lunch” was about to end. For decades, software developers had the luxury of relying on the performance improvements provided by the hardware industry to get their programs running faster. In the last few years, the CPU designers were mostly relying on increasing the clock speed (doing the work faster) and the chip’s complexity (doing the work smarter). As you have probably noticed, the clock of the laptop or the desktop that you can buy today does not run much faster than the one you could buy three years ago: the clock speed has plateaued south of 4 GHz. The increase in chip complexity has not reached the same limit because it relies on Moore’s Law (Moore, 1965) that is forecasted to last at least until 2020. In practice, this increase in chip complexity has meant an increase in on-die caches and core counts. Increases in cache sizes are generally good news, as they will typically help the performance of your applications. Increases in the number of cores will not bring sustainable performance benefits to single-threaded applications. It is worth pointing out that even if your application is single-threaded, it might still benefit from multiple cores for several reasons. This is simply due to the fact that you typically do not run your application in a vacuum: the CPU must be shared by several programs like the OS, your e-mail, your anti-virus software, and your application. But improvements that rely solely on no longer having to share the CPU with other applications are not sustainable; once a single-threaded application is running exclusively on one core, adding more cores will not help. To see real benefits on a machine with many cores, your application must run many threads. And as most software developers (including myself) learn, that is very hard indeed.

The important point to take away from this section is that software developers can no longer simply rely on hardware enhancements for their programs to run faster. We must have a hard look at our applications to leverage many cores. Without further delay, let’s have a closer look at the application that is our focus of attention in this paper: Markdown Optimization.

SAS MARKDOWN OPTIMIZATION

So what does SAS Markdown Optimization do? It provides retailers the ability to determine which items should be marked down, by how much they should be marked down, when, and in which markets or stores. Let’s have a closer look at the elements of this definition.

Which Items?

Why do retailers mark down items? Essentially to get rid of them (faster) so they can make room for new merchandise that (they hope) will sell better and bring in higher profit margins.

By How Much?

Without going into a discussion about price elasticity, you can see that a deeper discount is likely to lead to a faster sell through. However, you don't want to give too deep of a discount or you will leave money on the table (presumably, giving away the product for free should make room for newer products fairly quickly). By the same token, you don't want to give too little of a discount because the product won't clear the store.

Which Markets or Stores?

Products will not sell uniformly in all stores. For example, a beach store is likely to sell more bathing suits than a mountain store (which might still sell a few to those lucky enough to go on a cruise). Retailers know this in advance, so they stock differently based on the anticipated sales. When the time comes to make room for new products, it is better to analyze the situation of each store (or market) individually: stores with poor sales and high inventory will typically need to give a deeper discount than stores with good sales and low inventory.

When?

When to take the markdowns is not a discrete event. Retailers usually work with a markdown horizon, such as 16 weeks, during which they will take deeper and deeper discounts. The actual markdowns can be fine-tuned every week.

For large retailers, these multiple variables and their cardinality mean that SAS Markdown Optimization must solve a very large problem. For example, for the performance numbers that we discuss in this paper, we worked with 273 million active product-locations.

How does SAS Markdown Optimization solve this large problem? A detailed description of the implementation (as opposed to the architecture) of SAS Markdown Optimization is outside the scope of this paper. However, for the purpose of our discussion, it is enough to state that SAS Markdown Optimization decomposes the problem by groups of products (for example, a division of the retailer). Those groups of products are called partitions, and partitions can be processed independently of each other.

That reality is a key driver of the design of the high-performance version of SAS Markdown Optimization. As we will see in the next section, the high performance comes from the parallelization of SAS Markdown Optimization across many cores using a data decomposition pattern.

THE ARCHITECTURE OF SAS MARKDOWN OPTIMIZATION

Markdown Optimization is a typical SAS 9.2 solution as you can see in the following diagram:

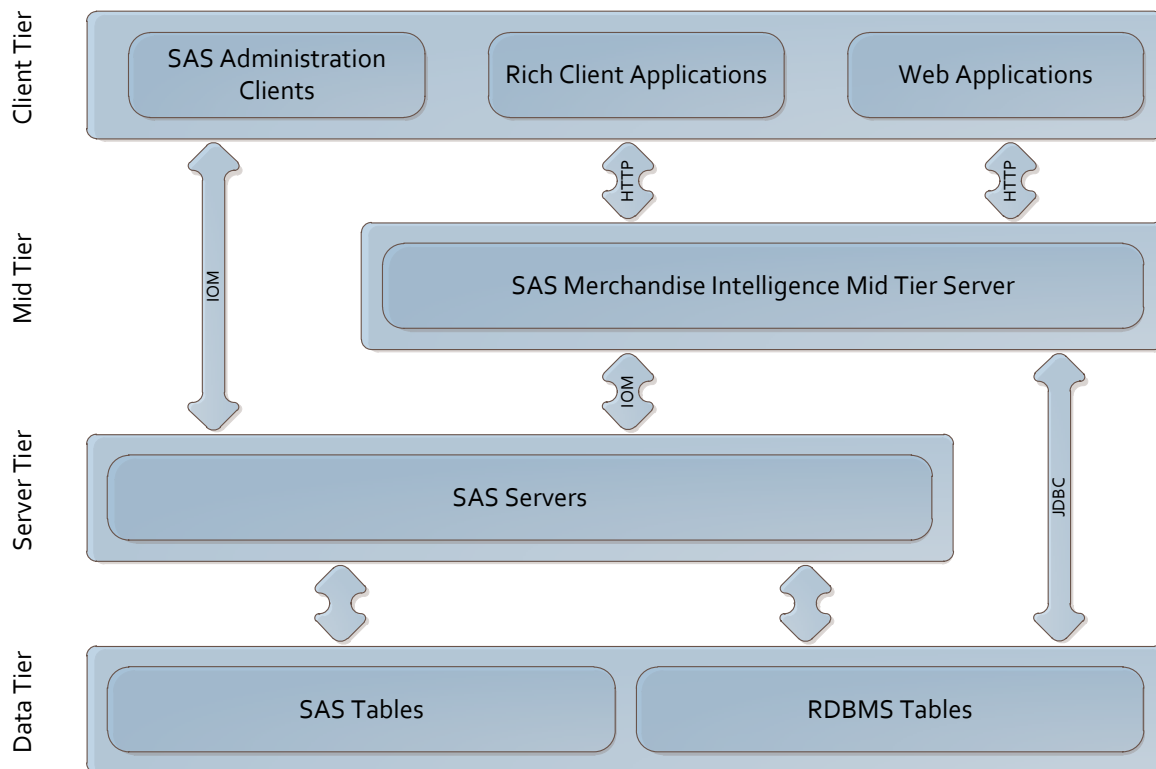


Figure 1: The Architecture of Markdown Optimization.

Like most SAS 9.2 enterprise solutions, SAS Markdown Optimization contains four tiers:

- The client tier is made of rich client and web applications.
- The mid tier is deployed in an application server (for example, JBoss). The mid tier has very little load to speak of since it is primarily responsible for repackaging the client requests and sending them to the back-end tiers.
- The server tier contains the SAS programs: some are part of the SAS 9.2 platform, and some are specifically written for SAS Markdown Optimization.
- The data tier consists of SAS tables and transacted tables. The transacted tables are solely there to guarantee ACID transactional integrity to the end users during online operations.

All SAS Markdown Optimization customers follow some variations of this workflow:

- **Batch Operations**
Batch operations (or the “batch”) typically take place over the weekend, when new data is available (for example, new sales figures). The new data is uploaded to SAS Markdown Optimization via ETL. Once the data load is complete, the batch updates the statistical models and re-optimizes the markdown plans. During the batch, only the server tier and the data tier are operational. Because the batch typically takes hours of processing, it is the determining factor in the choice and cost of the hardware.
- **Online Operations**
During the week, interactive users work with the GUI to create new markdown plans or update existing ones.

Since the duration of the batch is the main performance pain point, we need to focus primarily on the batch to significantly improve performance. We can also narrow our attention to the server tier and the data tier since they are the only operational tiers during the batch operation.

A fine-tuning of the storage is critical to reduce the duration of the batch, but a discussion on this topic is outside of the scope of this paper because it has little to do with the architecture of the application.

The other angle that we can attack is the computation throughput. As we have discussed earlier, to increase the computation throughput, we can no longer sit, relax, and wait for the clock rates to go up. We must take a careful look at the architecture and the implementation of SAS Markdown Optimization to distribute the processing to many cores.

Let's have a closer look at the strategy that we used to add the high-performance label to SAS Markdown Optimization.

THE ARCHITECTURE OF SAS HIGH-PERFORMANCE MARKDOWN OPTIMIZATION

Software developers have known the importance of design patterns for quite some time (Garlan & Shaw, 1994) (Gamma, Helm, Johnson, & Vlissides, 1994). Design patterns for parallel computing are still evolving rapidly, but one design pattern that we have found useful in the context of SAS Markdown Optimization is the Data Decomposition Pattern, which Mattson, Sanders, and Massingill (2004) define as a pattern where:

“we need to look (...) at the central data structures defining the problem and consider whether they can be broken down into chunks that can be operated on concurrently”

As we discussed previously, the Markdown Optimization problem space can be divided into groups of products. That natural data decomposition gave us a natural partitioning for the solution.

The following diagram illustrates the architecture of SAS High-Performance Markdown Optimization. The new components and tiers are in **bold** to emphasize the differences between SAS Markdown Optimization and SAS High-Performance Markdown Optimization:

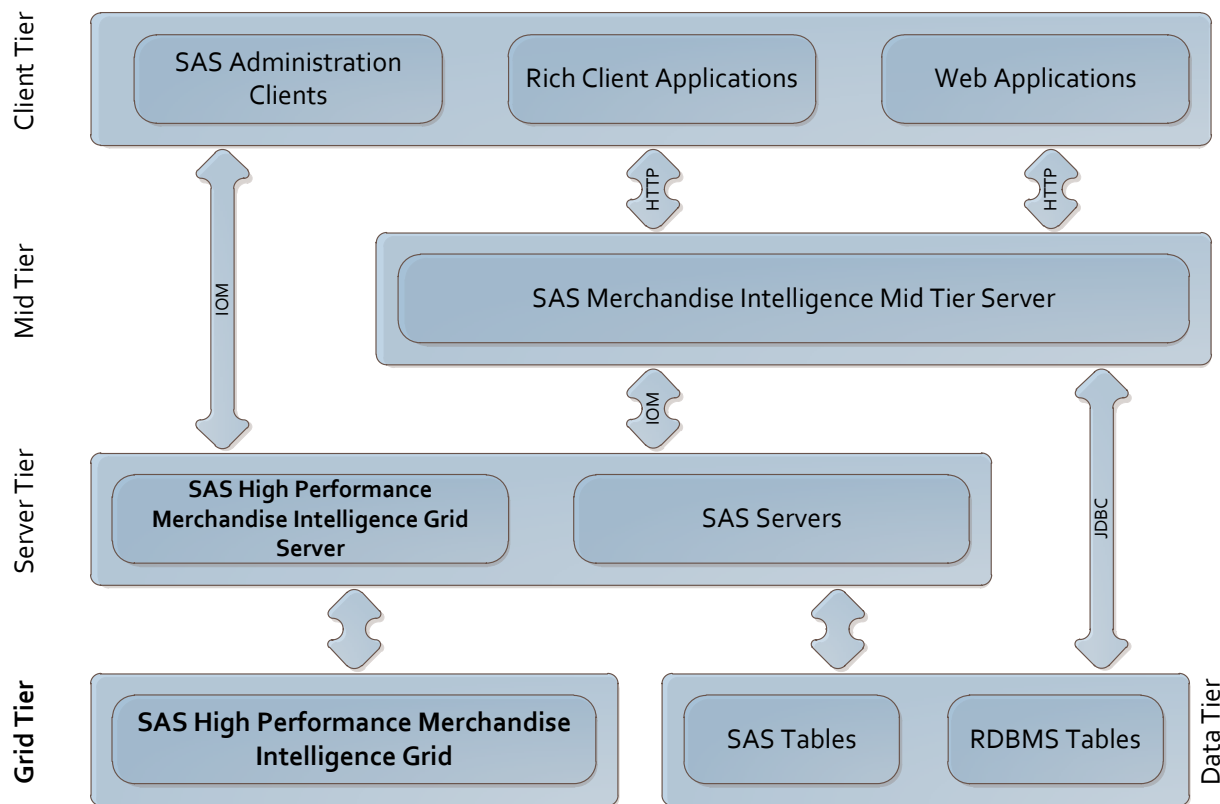


Figure 2: The Architecture of SAS High-Performance Markdown Optimization.

If you compare this diagram to Figure 1, the architectural differences are few:

- The SAS High-Performance Merchandise Intelligence Grid Server is added to the server tier. From the mid-tier standpoint, that additional server fulfills the role of the Stored Process Server. However, instead of queuing and running stored processes, it queues and runs jobs for the grid. Another key difference that has more to do with robustness than performance is that the Grid Server provides transactional queues.
- The grid tier is added to the SAS Markdown Optimization tiers. The grid tier runs the computations that in SAS Markdown Optimization are confined to a stored process. Using a grid allows a high degree of parallelism, which provides significant performance improvements without increasing the clock rate of the cores. The implementation of the SAS Markdown Optimization computations no longer relies on the MVA technology, but on the Threaded Kernel technology (What is the threaded kernel (TK)?).

The name “grid” should be understood as a grid of (computational) cores. There is no requirement that the cores be on separate machines; they could all be on one Symmetric Multiprocessing (SMP) machine or they could be distributed over hundreds of blades. A blade refers to a low-cost server (Hewlett Packard). As we will discuss further below, the choice between a cluster of machines and one machine is a deployment decision often based on economic considerations. As of this writing, hundreds of cores on one machine is a very expensive proposition, but as Moore’s law runs its course, it will cease to be true. The fact that SAS High-Performance Markdown Optimization has an architecture that works equally well on today’s and tomorrow’s hardware is a key advantage. In the next section, we will review a couple of deployment scenarios.

DEPLOYMENT SCENARIOS

Grid of Blades Deployment

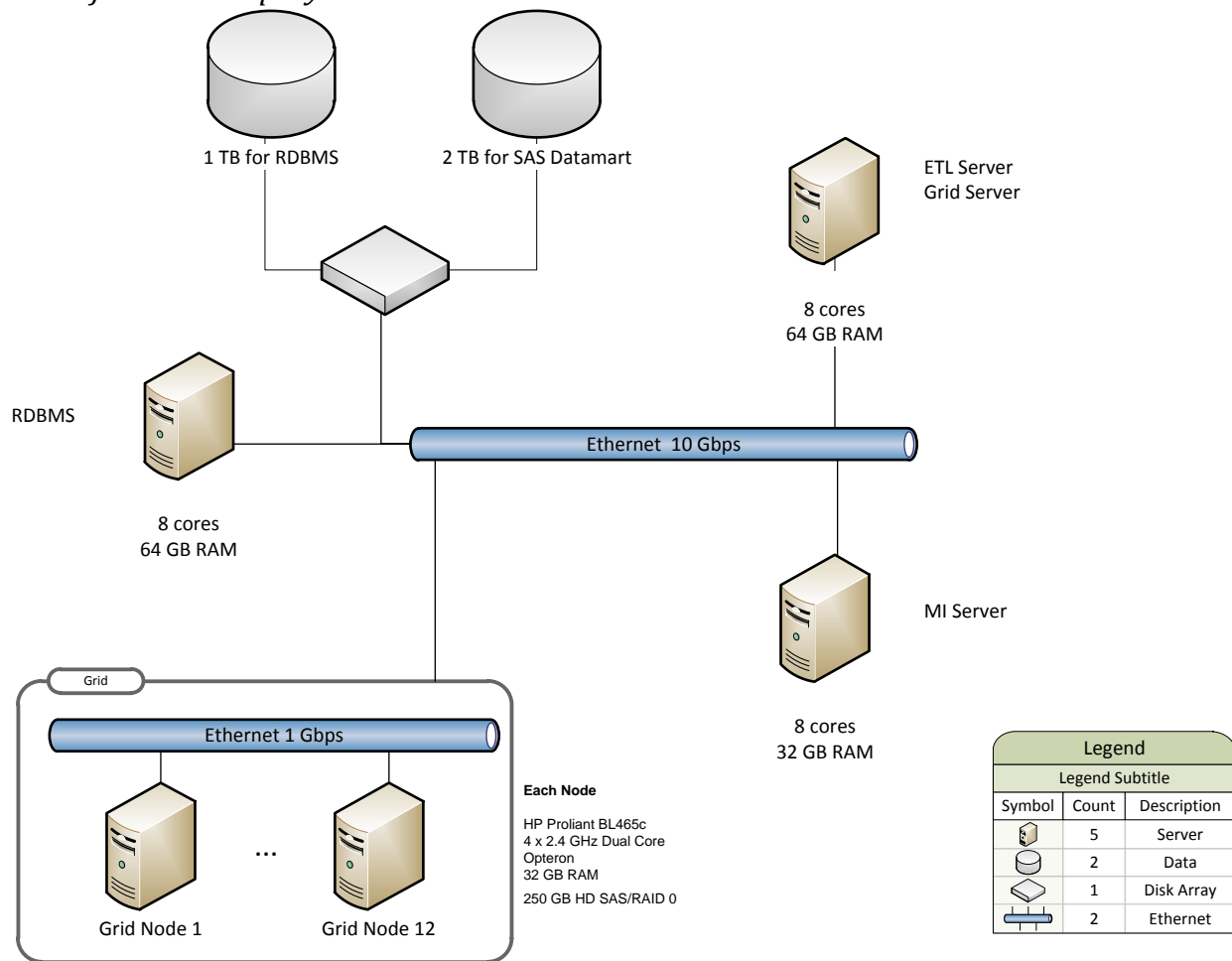


Figure 3: A Sample Grid of Blades Deployment for SAS High-Performance Markdown Optimization.

When SAS High-Performance Markdown Optimization deploys over a grid of blades, the responsibility of the SAS High-Performance Grid Server is as follows:

1. Copy the data to the blades.
The SAS High-Performance Grid Server first goes through some analysis of the partitions and then assigns them to specific blades. Once that step is completed, the SAS High-Performance Grid Server can proceed with the copy of the data from the ETL Server to the blades.
2. Launch the computations on the blades.
In addition to starting the work on the blades, this step also involves queuing the units of work so the blades are not overloaded.
3. Manage the uploads of the results to a relational database management system (RDBMS).
Some results of the SAS Markdown Optimization computations must be uploaded into an RDBMS because they will be reviewed and updated by interactive users. For these potentially concurrent operations to be safe, the robustness of ACID transactions is needed.

In this type of configuration, once the data is copied to the blade and before the uploading to the RDBMS takes place, each blade can work completely independently of each other. This type of computation is called a shared-nothing or an embarrassingly parallel problem (Foster, 1995): the tasks can execute more or less independently, without communication. This lack of communication is actually a very good thing; it allows each blade, and consequently the entire grid, to perform as fast as possible.

The network configuration in this example relies on 10 Gbps between the servers and the chassis (the blade enclosure) and 1 Gbps inside the chassis. This is typical, but not universal. Another popular configuration is to have 10 Gbps down the individual blades. In the next section, we will see yet another network configuration for SAS High-Performance Markdown Optimization.

Server Deployment

The following diagram shows a deployment with a couple of SMP servers, or very large blades:

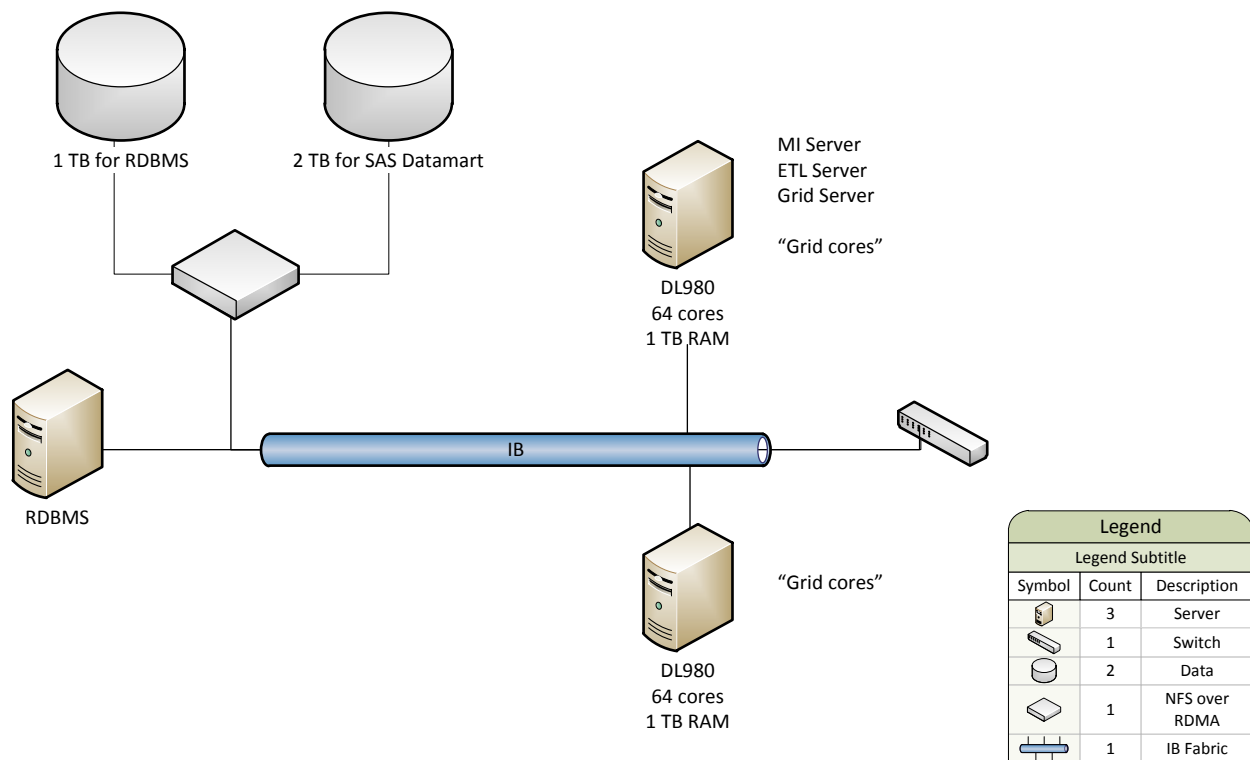


Figure 4: A Sample SMP Deployment for SAS High-Performance Markdown Optimization.

In this type of configuration, the data is typically not copied to the blades but accessed on demand via the network. So when SAS High-Performance Markdown Optimization deploys over one or more SMP servers, the responsibility of the SAS High-Performance Grid Server is as follows:

1. Launch the computations on the blades.
2. Manage the uploads of the results to an RDBMS.

The overall architecture of the solution does not change. As you might imagine, this type of deployment requires very good network connectivity: high bandwidth and low latency. In the previous example, this high throughput and low latency are provided by InfiniBand (IB) (Pentakalos, 2002). In a nutshell, IB provides potentially an infinite bandwidth by aggregating multiple links. In practice, the availability of switches limits the bandwidth. As of this writing, the fastest IB switches support 120 Gbps.

So when will you choose a deployment of blades versus a deployment of servers? We have SAS High-Performance Markdown Optimization customers deployed with both. The main driving factor is the number of cores that you need to meet your batch window requirements: as you increase the size of problem by increasing the number of stores, the number of products, and the duration of your markdown plan, you will require more cores. The rule of thumb is that the more cores you need, the more cost effective the blades are going to be. As Moore's law continues to run its course, the demarcation between a grid of blades and SMP servers will move closer to SMP servers. The important takeaway from this section is that the architecture of SAS High-Performance Markdown Optimization is designed to handle that transition.

In the next section, we briefly compare the performance of SAS Markdown Optimization and SAS High-Performance Markdown Optimization.

PERFORMANCE MEASUREMENTS

A detailed discussion on the SAS Markdown Optimization benchmarks quickly becomes complex and lengthy because of the data volumes and variety of deployment scenarios. It is therefore outside of the scope of this paper, but it is available upon request.

To give you an idea of the performance gains that you will see between SAS Markdown Optimization and SAS High-Performance Markdown Optimization, we present you with one example for the typical operations that make up the weekly batch processing.

In the table below, we list performance numbers that we have measured in SAS Markdown Optimization 4.2 with a single large SMP server (46 cores and 256 GB RAM) versus SAS High-Performance Markdown Optimization with a 48-node grid deployment (576 cores and 1.5 TB RAM). Both measurements were done on the same data: 273 million active product-locations and 82 million product-locations in markdown plans.

Tasks	SAS Markdown Optimization 4.2 Benchmark Elapsed Time (h)	SAS High-Performance Markdown Optimization Benchmark Elapsed Time (h)
Copy data to blades		00:20
Optimization	05:07	00:30
Upload to RDBMS		00:35
Estimation and Forecasting	19:30	00:33
Total Time:	24:37	01:58

Table 1: Batch Operations and Times.

As you can see, the performance improvements achieved with SAS High-Performance Markdown Optimization were quite significant. In our analysis, we found that the performance gain was due to its ability to use many cores. The SAS High-Performance Markdown Optimization numbers contain the two items that we mentioned earlier: copying the data to the blades and uploading the results to RDBMS. In a monolithic deployment like SAS Markdown Optimization, there was no need to copy the data to the blades; so that number does not apply. Uploading the results

to an RDBMS applies to SAS Markdown Optimization and SAS High-Performance Markdown Optimization, but that number is bundled in the optimization numbers for SAS Markdown Optimization because it is not a separate operation.

Another important point is the reduction in hardware cost: grid deployments typically cost a fraction of the cost of a very large SMP machine running a clustered file system (3 to 10 times).

CONCLUSION

In the coming decade, significant performance improvements in computing are not going to be free. Software developers need to take a close look at the architecture of their applications so that they can scale on today's and tomorrow's hardware. In this paper, we have described how the architecture of SAS Markdown Optimization was refined to take advantage of the current and future advances in parallel machines.

Now that we have a robust parallel architecture for SAS High-Performance Markdown Optimization that has proven its worth in several deployments, we will bring other retail applications like SAS Regular Price Optimization and SAS Promotion Optimization under the same high-performance umbrella. In addition, we will continuously monitor advances in hardware development to provide our customers with the best possible performance at the lowest possible cost. Finally, we plan to investigate more streamlined deployment scenarios based on an appliance model.

REFERENCES

- Amarasinghe, S. *The Looming Software Crisis due to the Multicore Menace*. Available at <http://groups.csail.mit.edu/commit/papers/06/MulticoreMenace.pdf>. Accessed on February 21, 2011.
- Arif, M. "The rise of the computing appliance". *ComputerWeekly.com*. Available at <http://www.computerweekly.com/Articles/2007/06/05/224539/The-rise-of-the-computing-appliance.htm>. Accessed on February 21, 2011.
- Foster, I. 1995. *Designing and Building Parallel Programs (Section 1.4.4)*. Boston, MA: Addison-Wesley.
- Gamma, E., et al. 1994.. *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston, MA: Addison-Wesley.
- Garlan, D., and M. Shaw. January 1994. *An Introduction to Software Architecture*. School of Computer Science. Carnegie Mellon University. Available at <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.1359&rep=rep1&type=pdf>.
- Garver, Shu-ling and B. Crepps, "The New Era of Tera-scale Computing". Intel. January 16, 2009. Available at <http://software.intel.com/en-us/articles/the-new-era-of-tera-scale-computing/>.
- Hewlett Packard. "What is a Blade-BladeSystem?" Available at <http://h18004.www1.hp.com/products/blades/info/what-is-a-blade.html>. Accessed on February 21, 2011.
- Krste Asanovic, R. B. October 2009. "A View of Parallel Computing". *Communications of the ACM, Volume 52 Issue 10*, Available at <http://portal.acm.org/citation.cfm?id=1562764.1562783>.
- Mattson, T. G., B.A. Sanders, and B.L. Massingill. 2004. *Patterns for Parallel Programming*. Addison-Wesley Professional.
- Moore, G. E. 1965. "Cramming more components onto integrated circuits". Available at ftp://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf
- Moore's Law*. Intel. Available at <http://www.intel.com/technology/mooreslaw/>. Accessed on February 21, 2011.
- Pentakalos, O. 2002. "An Introduction to the InfiniBand Architecture". O'Reilly Media, Inc. Available at <http://www.oreillynet.com/pub/a/network/2002/02/04/windows.html>.

SAS Institute Inc. 2007. SAS Note 24349, "What is the threaded kernel (TK)? " Available at <http://support.sas.com/kb/24/349.html>.

Sutter, H. March 2005. "The Free Lunch Is Over, A Fundamental Turn Toward Concurrency in Software". *Dr. Dobbs's Journal*. Available at <http://www.drdobbs.com/index.jhtml>.

ACKNOWLEDGEMENTS

None of this work would have been possible without the hard work of Gül Ege's Merchandise Intelligence Solutions Department. I would like to explicitly thank Gül Ege for her trust and continuous encouragement along with Thomas Burkhardt and John Pechacek for their patience and support.

CONTACT INFORMATION

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

Henry Bequet
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
919-531-0428
E-mail: henry.bequet@sas.com
Web: <http://www.sas.com/industry/retail/merchandise/>

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.