Engineering Unplugged:

A Discussion With Pure Storage's Brian Gold on Big Data Analytics for Apache Spark

Q&A

Apache® Spark™ has become a vital technology for development teams looking to leverage an ultrafast in-memory data engine for big data analytics. Spark is a flexible open-source platform, letting developers write applications in Java, Scala, Python or R. With Spark, development teams can accelerate analytics applications by orders of magnitude.

The rapid growth of Spark has not been without challenges. Most organizations have relied on sprawling deployments of the Hadoop Distributed File System (HDFS), with racks of spinning disks to meet the capacity and performance demands of data-intensive applications. That is about to change, however.

Pure Storage, a pioneer in block-based flash arrays, has developed a technology called FlashBlade™, designed specifically for file and object storage environments. With FlashBlade, IT teams now have a simple-to-manage shared storage solution that delivers the performance and capacity needed to bring Spark deployments on premise.

To help gain a deeper understanding of the storage challenges related to Spark and how FlashBlade addresses them, Brian Gold of Pure Storage sat down with veteran technology journalist Al Perlman of TechTarget for a far-reaching discussion on Spark trends and opportunities. Gold is an R&D director at Pure Storage and one of the founders of the FlashBlade development team. The following are highlights from that conversation.

TechTarget: Brian, what is your take on Apache Spark and what is it about Apache Spark that has made it so popular?

Brian Gold: I would describe Spark as a modern take on a high-performance data processing framework. We think of it as a database engine built from the inside out. It started at UC Berkeley's AMPLab, where they began with really powerful primitives and fairly

PURESTORAGE®

TechTarget | Custom Media

low-level abstractions for data manipulation that scale well across CPU cores and across machines in a data center.

They corrected some of the most problematic issues in earlier Hadoop MapReduce implementations. More recently, Spark has gained higher-level abstractions that move more and more into a declarative analytics platform, much like you'd find in a whole database system.

And, in the recent 2.0 release, Spark has incorporated impressive SQL optimizations that increase its applicability in traditional data warehousing. Moreover, the Spark developers have continued to push the envelope in core data-processing technology, offering industry-leading performance on sorting benchmarks and impressive stream-processing algorithms.

So I'd say our take is that, with the addition of these general-purpose and highly optimized SQL engines and some of the capabilities of its stream-processing extensions, Spark is now going well beyond its roots. Whereas it started as a better system for batch analytics and iterative algorithms, it is now a great platform for building a complete data warehouse—integrating the structured data coming from a transactional store with real-time data streams and extracting pretty deep insights from large-scale analytics.

TT: How do customers benefit from these large-scale analytics and why is Spark a good fit for them?

BG: If you look at every one of our customers, the data and the insights they derive from that data are at the center of their business. The volume of data that you can now collect across an enterprise has grown so fast that existing systems for storing and processing it can't meaningfully keep up.

This isn't a new message, right? It's the same centerpiece of the first big data hype wave from seven to 10 years ago. But increasingly we see companies are able to leverage these insights to derive meaningful competitive advantage; they can then deliver more value to their customers.

We see our customers using Spark for two interrelated classes of work. First, big data is often ugly data. It comes in unstructured, it is poorly formatted, it has errors that need to get cleaned up. Spark has the parallelism and expressiveness that our customers need to perform one of the industry's often unsung but most important tasks—data cleaning and transformation.

Once customers have a more well-formed data set, the second main use case can be broadly classified as machine intelligence or machine

Spark is now going well beyond its roots. Whereas it started as a better system for batch analytics and iterative algorithms, it is now a great platform for building a complete data warehouse—integrating the structured data coming from a transactional store with real-time data streams and extracting pretty deep insights from large-scale analytics.

learning. This can take the form of fairly traditional analytics queries that are running reports on a data warehouse. But increasingly we see our customers identifying anomalies in their data, building advanced prediction models, correlating events across disparate data sets and similar deeper analytics.

As one of the engineers who helped build this product at Pure, I'm particularly excited about this area because Pure Storage, just like any of our own customers, has multi-petabyte data sets that need sophisticated tools to unlock a broader set of insights.

TT: What are some of those broader insights and how do they affect how a modern business might utilize analytics to drive business opportunities?

BG: Let me describe how we, at Pure, use large-scale data and what our analytics infrastructure looks like. With customer approval, each one of our systems can send back internal telemetry data to Pure. We collect a lot of telemetry: There's flash storage health, networking performance, internal metadata management, front-end protocol behavior, statistics about data reduction and space utilization, and much more.

In addition to the in-the-field data, we incorporate test logs from our manufacturing lines to monitor hardware health from the moment we solder chips onto a board. We always want to compare how our internal engineering tests perform, where we're developing software and running tests. How do those behave relative to how customers interact with our products?

As we've ramped up the worldwide deployments of all of our storage arrays, this has turned into a massive data set—multiple petabytes of telemetry data. We use this information to understand how our products are performing in the real world, how our customer experience could be improved, what issues we could resolve proactively.

The running joke internally is that we had to go build our second product, FlashBlade, just to analyze all the telemetry that we collect from our first product, FlashArray. It's a big deal across the entire company—support, engineering, our go-to-market teams.

All this data analysis that Pure has done has allowed us to make our customers extremely happy with the products they get. We see that reflected in things like our Satmetrix-certified Net Promoter Score of 83. We've worked hard to achieve that result, and we're very proud of our customers' satisfaction in our products and service as a company.

TT: We understand the benefits of all this data if it can be used strategically and intelligently. But we also understand that it places tremendous pressure on the storage infrastructure. What are the challenges from a storage standpoint that Spark poses to both users and to the IT teams that support them?

BG: In a nutshell, we built FlashBlade to enable this next generation of analytics workloads, those that need to scale capacity and storage performance with ease. We can't drown our users in needless administration and management complexity and overheads.

We hear this over and over from customers running Apache Spark and other tools in a modern data analytics pipeline. They need big storage capacity. They need fast access to all of their data. And they need simplicity in how you scale and manage all of that data. FlashBlade delivers on all three of these dimensions.

TT: What is FlashBlade and how did Pure go about developing it?

BG: FlashBlade is our scale-out all-flash storage system for file and object storage. To put it in context, Pure started with the product we call FlashArray, which is an all-flash block storage device. By focusing on data reduction and an architecture designed for solid-state media, FlashArray delivers all-flash performance at or below the cost of enterprise disk arrays. Since the company's founding in 2009, FlashArray has gone into some of the most mission-critical enterprise storage use cases.

As FlashArray took off in the market, we observed that there was an adjacent set of workloads and use cases that needed a file-oriented access protocol. These workloads require a large set of machines accessing a shared dataset. So a distributed file system became the next clear target, where we felt an all-flash, built-from-the-ground-up storage system could really deliver transformational value to our customers. This was not something our existing products could serve and, frankly, not something that any competitor's existing product could serve.

FlashBlade is a co-designed hardware and software system. We built everything from first prototypes through a fully functional modular, high-performance file and object system.

TT: What is it specifically about FlashBlade that makes it a good fit for Spark?

BG: Let's look at how Spark is typically deployed today. Spark came out of the Hadoop ecosystem, so the default is typically the Hadoop File

System, or HDFS. HDFS can scale to very large capacities; that's what it was designed for.

But HDFS was also designed in an era when large capacities required spinning disks. Building on a storage platform of spinning disks often leads to performance problems—or massive overprovisioning of spindles to avoid those problems—and an operational nightmare, given the high failure rates of hard drives.

HDFS was built for that environment and has deep inefficiencies when transitioning to an all-flash or SSD-based environment. Running Spark with FlashBlade, on the other hand, gives you three advantages over HDFS and other traditional file and object storage systems as well.

- The first is all-flash performance. More than just data sheet numbers, the performance that FlashBlade unlocks allows consolidation even at the big data scale.

- The second advantage is storage efficiency. We built FlashBlade around an efficient erasure coding scheme that achieves N + 2 redundancy with minimal additional replica overhead. The traditional HDFS deployment has at least three copies of every piece of data—in part because of the extremely high failure rates of hard drives as a whole.

- The third advantage is the design and simplicity it brings. This is a holistic value across all of Pure's product efforts. We wanted a storage system that was extremely simple to set up, configure, operate and scale. If you have a data set and need more capacity, just add a blade to the modular chassis architecture. There's no manual restriping of data. Nothing that requires user intervention. You don't have to plug in additional cables. You don't have to do anything other than just slide in an additional blade that can offer 40, 50 terabytes of effective capacity with each blade.

These advantages come together when a customer runs Spark with their data in FlashBlade. A Spark cluster can now run on compute servers that have no particular storage requirements. And that cluster can be sized appropriately to maximize the achievable parallelism for a given workload.

TT: Can you provide an example of this model in a real-world use case?

BG: One of the things that's particularly exciting about this deployment model is that another cluster can run side by side on the same data set. A lot of times, a customer will have a large-scale Spark cluster running a production data pipeline. That data, accessed by a particular workload, can be streaming data with new log lines coming in from various sensors or, in our case, manufacturing records or customer telemetry.

We ran a recent test comparing an on-premise data warehouse deployment in Spark running with FlashBlade against a similarly sized Spark Cluster with HDFS storage on Amazon. And we found that, on average, queries ran about 3x faster with FlashBlade and up to 6x for particularly data-intensive operations.

One of the novel aspects of FlashBlade is that we built and co-designed all of the software with the hardware, all the way down to the level of the NAND flash components themselves. That allows us to monitor the health and performance of every bit of flash at the individual read and write level. We now have access to whole new forms of data we can use to optimize some of the internal algorithms and build prediction models.

You can also access historical records at the same time. So you've got these complex data pipelines that are running in a production environment, meaning they have to be always on, always up, with fairly high throughput requirements. All of that data can be kept on FlashBlade.

Simultaneously, you can have a data science team that's doing exploratory analysis and you can actually give them their own independent Spark cluster. Because now it's just a set of computers on a high-speed network, they can do their exploration directly on the same data set.

This is a multi-petabyte data set. Perhaps you can't afford to make a copy of it. Or maybe you don't want to make a copy of it and incur the additional overhead of having to run totally separate storage infrastructure for the data science team. So you can run the same class of algorithms that you'd run in production, but do more advanced experiments.

As you see incremental value, you can roll those new algorithms into your production environment. In Pure's case, we have hundreds of people who depend on the quality of the analytics that we run in our own data pipelines. So we're always looking for ways to do more with that. But we can't disrupt it because it's production.

Spark provides innovative new opportunities in how data gets used across the board in companies like ours. FlashBlade can become a centerpiece for how you build out, not just production pipelines, but also these data science pipelines that are enabled by the decoupling of storage and compute, along with the performance and simplicity that FlashBlade offers.

TT: Why not run Spark and similar analytics tools in the cloud? That's kind of been the model so far. Why look to bring Spark on premise?

BG: First we should clarify what we mean by "cloud." Pure has had great traction with software-as-a-service companies and enterprises that are setting up their own private clouds. These are cloud providers of various kinds and flavors, and all of the advantages we've talked about so far are exactly why these companies are increasingly deploying all of Pure's products for various parts of their data infrastructure. Spark and data analytics is a centerpiece of a lot of that.

What's most often meant by the cloud is really public cloud providers—and we think they're great at what they do. The flexible deployment that you get with a very low startup cost has a lot of advantages in a lot of cases. And public clouds are a great way to start experimenting with this

new class of analytics tools. They can have a pretty steep learning curve, these new tools like Spark. And a small ephemeral cluster, sitting in a public cloud infrastructure, can actually be a great starting point to understand how a business could derive value from Spark.

However, when the data set starts to grow larger, when processing demands increase and when we start to find real value, we've found—and our customers have overwhelmingly corroborated—that there's a big advantage to running these workloads on dedicated on-premise infrastructure. By decoupling high-performance storage, in our case FlashBlade, from general-purpose compute, an on-premise or private cloud deployment can have the agility of the public cloud without sacrificing the performance or incurring steep cost overruns.

We ran a recent test comparing an on-premise data warehouse deployment in Spark running with FlashBlade against a similarly sized Spark Cluster with HDFS storage on Amazon. And we found that, on average, queries ran about 3x faster with FlashBlade and up to 6x for particularly data-intensive operations.

Yet, the cost of operating an always-on HDFS cluster in the public cloud environment can quickly become prohibitive at large sizes. The public cloud is particularly good at allowing burst expansion and then spinning down to match demand. While many deployments are now looking at cheaper object interfaces, like S3 in the Amazon ecosystem, the performance just isn't there for iterative analytics.

There isn't a particularly good fit right now in the public cloud. We've seen customers try to introduce tiering and add yet more complexity to work around what ultimately is a storage compute bottleneck. FlashBlade and this decoupled storage-compute architecture gives you the agility and the burst capability that you can often find in the public cloud, but preserves the performance and the cost advantages. Once you have a sizable data set and production demands of running always-on infrastructure, we offer a tremendous value for data-intensive applications.

TT: Have you found a use for machine intelligence techniques in data sets?

BG: At Pure we are a consumer of massive data sets. We're always looking to do more with that. And as we've rolled out FlashBlade, it's taken the company in new dimensions.

Hardware integration is a good example. One of the novel aspects of FlashBlade is that we built and co-designed all of the software with the hardware, all the way down to the level of the NAND flash components themselves. That allows us to monitor the health and performance of every bit of flash at the individual read and write level. Even in a petabyte-scale storage system, we can collect telemetry indicating what the wear looks like, what the error correction rates are.

These are new pieces of data that aren't accessible, even in an SSD-type deployment. By co-designing and deeply integrating our hardware and software, we now have access to whole new forms of data we can use to optimize some of the internal algorithms and build prediction models.We've really just begun exploring that, but the initial results are very positive that we can do more with our own products based on the internal telemetry that we get.

TT: Is there anything that we haven't talked about, in terms of Spark and FlashBlade, that you think we should talk about before we get to the end?

BG: We do see a lot of advantage in the fact that we use the most widely deployed distributed file system interface. One of the challenges of the more restricted file protocols like HDFS, is that they become a silo for data. So you have what may be a petabyte-scale data set sitting in HDFS. If I have Hadoop or Spark or other applications in that ecosystem, I can access and process the data there. But if I want to access it through my normal file system, it's possible but it adds more layers of complexity.

One of the things that makes Spark an ideal fit is they've really decoupled the storage protocol from how they handle all of the compute side. So that allows us to plug in even a very simple, very widely deployed storage distributed file system like NFS. We can deliver 10 to 15 gigabytes per second of read throughput through NFS into a Spark cluster. And simultaneously we can go in through a whole other application and access the same data set through a much more traditional access method, just coming in over NFS through a normal Linux, Unix, Mac OS, Windows environment.

There's a general theme of consolidation that is often lacking in this space. Consolidation for petabyte-scale workloads has not been something that traditional architectures have really permitted. FlashBlade enables a new era of consolidation by delivering unprecedented density, performance and effortless operation.

To learn more, please visit purestorage.com/analytics.