E-Guide

BIG DATA: MOVING APPS INTO PRODUCTION & OVERCOMING DATA MODELING CHALLENGES

TechTarget | SearchData Management

BIG DATA POSES big challenges, but for most organizations, pushing forward with big data initiatives is worth the effort in the long-run. How to do so efficiently, however, isn't exactly obvious. This two-part expert e-guide provides best practices for moving big data apps into production, as well as tips for handling new technologies like NoSQL.

BEST PRACTICES FOR MOVING BIG DATA APPS INTO PRODUCTION

George Lawton

At Spark Summit, distinguished Amazon engineer Marvin Theimer weighed in on some of the best practices they have identified to get big data apps built on Apache Spark production-ready. “Most of this conference is about getting to success, which means going from prototype to production,” he said. “Once the enterprise has designed its big data algorithms, it needs to put them into production. In the car world, it is one thing to design a new car. But then, the hard work comes of stamping out a million of them.”

UNDERSTAND NEW REQUIREMENTS

Enterprise architects need to consider a variety of what Theimer called ilities to get the apps ready for production, including scalability, high availability, maintainability, evolvability, auditability and reproducibility. Many of these requirements have always been important to enterprise architects. Some, like maintainability, need to consider the faster pace of change in new data

analytics tools. Others, like auditabilty and reproducibility, grow in importance, as these apps drive business decisions and interact with the system of record in new ways.

With maintainability, enterprises need to plan for the rapid pace of upgrades to core data infrastructure underlying the Spark platform. It's like changing the tires on a car as it is driving, so organizations have to be prepared to quickly deploy new security patches.

It is also important to consider the applications are developed using test data. The production application needs to manage personally identifiable information, like credit card data, so it does not get hacked and users don't accidentally see each other's data.

The enterprise needs to be able to meter usage of customers and send out an appropriate bill. When customers dispute a bill, there needs to be an ability to go back with an audit log of metering records and reproducibly show how the bill was calculated in order to come to a resolution. All of these capabilities have to be built into the cloud infrastructure behind the big data apps.

AIM FOR MUNDANE EFFICIENCY

Developers need to think about going from algorithmic efficiency to mundane efficiency. It's fine to use the most performant services in development, but deciding to use a slower storage service like Glacier, rather than Simple Storage Service, can have a big effect on the bill. It's also important to have a procedure in place for getting rid of data that never gets used. In the beginning, it may make sense to store everything to enable new use cases. But enterprises should consider a regular analysis to identify and toss out data that is never used, nor required to be stored. "You have to remember which to throw away," Theimer said.

This also applies to managing millions of jobs. Some of these will fail in ways the enterprise will not be able to predict. The enterprise needs to have a process in place for identifying and killing these zombie services.

Additionally, the company should consider supporting a different kind of user. Early adopters may want fine -grained control, while the majority in the enterprises will want ease of use. As big data apps go into production, they will be adopted by people with greater skill in other areas, like business, biology or astronomy, rather than data science.

PLAN FOR RECKLESS USE

It is also important to keep track of how the applications are being used. As enterprises begin to open this big data infrastructure to others in the organization, many of these apps will be used in highly inefficient and costly ways. “Developers will craft Ferraris, but then users will take them into the mud. This could be disastrous when the application is running on a multi-tenant system,” Theimer explained. “Consequently, enterprises need to think about how to get users other applications, like a Jeep, or make it easier to take the Ferraris off-road and still get good results. You need to design systems that will tolerate usages you never expected and still give a good user experience.”

Another good practice is to identify and plan for the effect of large-scale events, such as an entire data center going offline. At this point, everyone in that data center will be trying to move their applications to different data centers, overwhelming the network. It’s akin to everyone clogging up the freeways to escape a tornado. It is important to architect conservative recovery mechanisms so recovery processes don’t overwhelm the network.

CREATE ROBUST STOPPING POINTS

Implement stopping points to make it easier to roll back parts of a process when something goes wrong. In cases when data is corrupted, it is more efficient to reprocess the point in the processing pipeline after the corruption than rerun the whole process.

One approach is to create a bread crumb infrastructure showing the path of data as it is transformed through the data pipeline. If the system spits out the wrong number, this makes it easier to go back and find the place where a problem occurred.

Organizations should consider adapting the kind of accounting mechanisms used in the financial industries to track the representations of money as it flows through different systems. Data will inevitably go wrong. This kind of an approach will make it easier to find the input sources, so solutions are easier to identify.

CONSIDER THE ECONOMICS OF THE IMPLEMENTATION

Mundane efficiency involves being proactive in working with usage reports, billing alerts, cost explorer and lifecycle management. As these applications start to go into products, business imperatives are likely to drive many of these

cloud applications toward multi-tenant services to lower costs. Enterprise architects also need to think through tracking the machine learning resource usage in order to manage costs.

"To be production-ready, you need to build this stuff from the beginning. After you have gotten to the 'eureka' point of getting data in, then you have to automate everything," Theimer noted. "It is also important to make this stuff secure by default. It is also important to think about pushing through in the direction of serverless event-driven computing. The good news is that the cloud can help a lot with all of this."

BIG DATA CHALLENGES TRADITIONAL DATA MODELING TECHNIQUES

Jack Vaughan

The onslaught of big data, with its high data volumes and diverse data structures, has given rise to new technologies in the form of NoSQL, Hadoop, Spark and the like. NoSQL, particularly, calls for changes in established data modeling techniques.

Some basic learning is order, too, when it comes to NoSQL databases, such as MongoDB, Cassandra and Redis, advised at least one data veteran during the recent Enterprise Data World (EDW) 2016 conference in San Diego.

"Nobody is born knowing NoSQL," said Ted Hills, an enterprise data architect at information provider LexisNexis, based in New York. Data modelers should realize everything they know about logical modeling is still true, he continued, but they should also realize that "NoSQL gives a richer tool box," with which developers can work.

Data pros should be ready to the accept change, and to embrace the new capabilities of big data tools, Hills said, even though the tools lead to changes in existing modeling methods.

Hills, author of the recently released NoSQL and SQL Data Modeling, suggested a need for new modeling notations that embrace NoSQL functionality.

DATA MODELING TECHNIQUES MEET SCHEMA-ON-WRITE

One effect of the NoSQL side of big data development has been to delay schema creation. The early definition of the data schema was once a lynchpin of data quality practices, and a prerequisite for just getting a project going.

Schema creation may be moving to a different stage in the development cycle, according to Karen Lopez, data architect and principal consultant at InfoAdvisors.

"It's not that we don't care about quality. It's that we are not caring about the schema upfront," she said. This doesn't mean designs become "schema-less." Instead, they come to support something akin to "schema-on-read" model, she said.

Hills concurred, saying people's enthusiasm for NoSQL becomes tempered as they wonder "what is all that stuff I slammed into my [database management system]?" He expects data modeling to move from a solely prescriptive mode to

one that includes some descriptive modeling, where data schemas are created after initial data development.

EMBRACE THE AGILE

The term descriptive is apt for data architecture today, according to Lakshmi Randall, an independent analyst and consultant on hand at EDW 2016.

"Now, as far as data modeling is concerned, things are more descriptive. Instead of trying to plan everything out ahead of time, you see use cases developed sort of on the fly," she said.

One type of NoSQL database may be particularly related to this style of data design, Randall said. That is the graph database.

The NoSQL graph database has the ability to capture information on the many interactions that occur in, for example, Web and customer relationship systems, she said. In this way, it can be helpful in creating a descriptive model of wide-ranging application, she added.

For his part, LexisNexis' Hills emphasized the challenge of NoSQL design is part of a larger trend to make both businesses and software development more flexible.

The trend bears the name of Agile methodology, and many of its principles -- for example, projects organized around early delivery, short but frequent iterations and moderate use of upfront schema -- are alien to traditional modeling.

Agile means teams work in smaller project "chunks" than before, he said, and the business side of the organization is involved at every step.

"Data modelers should learn from Agile development, working in small sprints. Our traditional data modeling world has been more of a Waterfall process," Hills said, referring to a project style that has become associated with multiyear development.

Hills said there is nothing wrong with beginning to store data in NoSQL prior to schema creation. He urged data modelers to be open-minded to the value of the new technology.

"Don't view that as an enemy -- view it as an opportunity," he said. "You should be on that Agile development team, and talking to the business as much as your development colleagues are."

FREE RESOURCES FOR TECHNOLOGY PROFESSIONALS

TechTarget publishes targeted technology media that address your need for information and resources for researching products, developing strategy and making cost-effective purchase decisions. Our network of technology-specific Web sites gives you access to industry experts, independent content and analysis and the Web's largest library of vendor-provided white papers, webcasts, podcasts, videos, virtual trade shows, research reports and more —drawing on the rich R&D resources of technology providers to address market trends, challenges and solutions. Our live events and virtual seminars give you access to vendor neutral, expert commentary and advice on the issues and challenges you face daily. Our social community IT Knowledge Exchange allows you to share real world information in real time with peers and experts.

WHAT MAKES TECHTARGET UNIQUE?

TechTarget is squarely focused on the enterprise IT space. Our team of editors and network of industry experts provide the richest, most relevant content to IT professionals and management. We leverage the immediacy of the Web, the networking and face-to-face opportunities of events and virtual events, and the ability to interact with peers—all to create compelling and actionable information for enterprise IT professionals across all industries and markets.