THE EVOLUTION OF BIG DATA: NOSQL, HADOOP, SPARK & BEYOND
As the big data ecosystem continues to expand, new technologies are addressing the requirements for managing, processing, analyzing, and storing data to help companies gain the most benefit from the rich sources of information flowing into their organizations.

From NoSQL databases to open source projects such as Spark, Hive, Drill, Kafka, Arrow, and Storm to commercial products offered on-premises and in the cloud, the future of big data is being driven by innovative new approaches across the data management lifecycle. The most pressing areas include real-time data processing, interactive analysis, data integration, data governance, and security.

The big data revolution is diminishing the sharp delineations between data types, handling all data types equally. This provides a unique opportunity to tap into all of this data to provide information to decision makers.

NEW PROBLEMS CALL FOR NEW SOLUTIONS

Addressing the need to store and manage data that does not fit neatly in rows and columns, NoSQL technologies are at the forefront, representing a broader world that connects to the internet at large, the Internet of Things, and clouds.

NoSQL databases can run on commodity hardware, support the unstructured, non-relational data flowing into organizations from the proliferation of new sources, and are available in a variety of structures that open up new types of data sources, providing ways to tap into the institutional knowledge locked in PCs and departmental silos. For example, the emerging blockchain technology is designed to store data, general-ledger style, in a highly distributed approach across the wider internet.

The four key database types that fall under the NoSQL category are key-value stores, which allow the storage of schema-less data, with a key and actual data; column family databases, which store data within columns; graph databases, which employ structures with nodes, edges, and properties to store data; and document databases, which enable simple storage of document aggregates.

And, despite what the name might imply, NoSQL database vendors are increasingly addressing their customers’ need to use SQL as a primary language for querying data. While the NoSQL landscape is still relatively new, it is evolving quickly with new features for greater accessibility, interoperability, security, and governance, and showing signs of its future potential for the enterprise.
THE GROWING HADOOP ECOSYSTEM

Central to the big data technology landscape, Hadoop, which this year marked its 10-year anniversary, has expanded well beyond a platform for storage and batch processing of large quantities of data on commodity servers. The Apache Hadoop framework, consisting of Hadoop Common, the Hadoop Distributed File System (HDFS), Hadoop YARN, and Hadoop MapReduce, is a core component to most big data projects and to the creation of data lakes. In addition, there are now well over 100 open source projects in the greater Hadoop stack, adding the security, flexibility, and accessibility of Hadoop with additional enterprise features.

Prior to Hadoop, capturing and analyzing data of any kind meant using proprietary tools, which was an expensive and resource-intensive undertaking. Hadoop, as well as the open source ecosystem that surrounds it, provides a more cost-effective option for big data analytics within the reach of more users. However, the challenge to data managers will be acquiring the skills needed to build out these open source environments.

THE EMERGENCE OF SPARK

Spark, a data analytics framework, is one of the newest technologies in the Hadoop ecosystem. It is a data analytics framework rooted in the Apache Hadoop world. Some vendors even refer to Spark as an “analytics operating system,” suggesting that it can form the foundation of a broad array of data analytics functions and applications that can be built on top of it.

Hadoop and Spark are part of the open source wave that continues to offer useful capabilities to enterprises of all sizes, providing a cost-effective and highly scalable means to package and deploy large and varied datasets.

Some of Spark’s proponents point out that the open source framework picks up where Hadoop leaves off. For starters, Spark inherently supports real-time requirements and offers faster processing and a more robust analytics engine powered by in-memory parallel-processing capabilities. Among other benefits, the framework includes resident libraries that enable faster development of applications that target and process structured, semi-structured, and unstructured datasets. The Spark framework helps manage a variety of jobs, from streaming to traditional ETL to data lakes and the latest real-time streaming applications.

The major components of Spark include SparkSQL, which is its query access language that accesses data sources via SQL queries; Spark MLlib, which provides predictive analytics capabilities; SparkR, which provides access to Spark data; and GraphX, an API for graph computations with a built-in library of common algorithms.

COMPLEMENTING HADOOP

Spark is not intended to replace Hadoop but complement it. Spark is built on the Hadoop Distributed File System and fits well within environments that already have Hadoop skillsets or tools.

The main distinction is that Spark is purely an analytics engine, while Hadoop is a data management and storage system. Spark does not require Hadoop for data management and storage, however. The benefit is that data managers can keep their current configurations—including Hadoop or other data environments—in place, without the need to undertake migrations.

Organizations often find themselves with silos of data and data frameworks. Even Hadoop projects tend to end up in their own silos. The Spark framework is extremely versatile and can be deployed in many ways for many analytic functions. Spark capabilities can also be integrated into existing applications, such as those built on Java. Another key feature is Spark’s support for Resilient Distributed Datasets (RDDs), which enables objects to be managed and stored anywhere across the infrastructure, both on disks or in memory. This also helps ensure greater high availability.

Data analytics has long been confined to specialized teams of analysts who tended to work remotely from other business teams. Spark may represent the biggest step yet toward the holy grail of analytics—ubiquitous enterprise access by all levels of employees. Spark is accessible to many players from across the enterprise—especially those who are concerned with managing, analyzing, and bringing data to the fore of a business strategy. Spark is also an optimum framework for scientists and analysts, as many complex capabilities are already built into its “analytics operating system.” The framework supports the well-known programming languages, and applications can be quickly built on top of Spark’s foundational capabilities, which include machine learning, real-time processing, and graph processing.

Spark is still a maturing technology platform, meaning that there are still some aspects of the framework that need to be learned by data managers, particularly with the open source community and supporting vendors. For example, some users report that aspects of the solution are not user-friendly and require some noodling to get around. This is a natural phenomenon with all emerging technologies, and as Spark’s base of developers and vendors continue to enhance its functionality, it will have a more central place in the enterprise.

THE PRESSURE TO COMPETE ON ANALYTICS

This is an era in which organizations are under pressure to compete on analytics in a hyper-competitive global economy. Many are building big data stores, but lack effective ways to convert that data into actionable insights, when and where they are needed.

The combination of Hadoop big data processing and the larger Hadoop ecosystem, including Spark-accelerated analytics and the wealth of NoSQL databases, provides the capabilities targeted to business problems and opportunities.

—Joe McKendrick
Data Engineering with Apache Hadoop

WHAT IS DATA ENGINEERING?

Data engineering is the process of building analytic data infrastructure or internal data products to support the collection, cleansing, storage, and processing (in batch or real time) of data, for answering business questions—usually, by a data scientist, a statistician, or someone in a related role.

Examples of data engineering include, but are not limited to:
- Building data pipelines that aggregate data from multiple sources
- The productionization, at scale, of machine-learning models
- The creation of pre-built tools that assist in the query process (e.g., UDFs)

Data engineers rely on the Apache Hadoop ecosystem, including components, such as Apache Spark, Apache Kafka, and Apache Flume, as the foundation for this infrastructure. Regardless of use cases and components involved, this infrastructure should be compliance-ready with respect to security, data lineage, and metadata management.

This Data Engineering eBook walks through technical concepts pertaining to building and maintaining analytic data infrastructure on a Hadoop-powered enterprise data hub. We introduce some of these concepts at a high level here, but dive deeper in the eBook.

ARCHITECTURAL PATTERNS

For Near Real-Time Data Processing

Evaluating which streaming architectural pattern is the best match to your use case is a precondition for a successful production deployment. In this eBook, we discuss four major streaming patterns, and how to implement those patterns architecturally.

FRAUD-DETECTION

To design effective fraud detection architecture, look no further than the human brain (with some help from Spark Streaming and Kafka). At its core, fraud detection circles around the detection of anomalies and reactions to those anomalies.

Effective fraud detection architecture requires that three subsystems work cohesively to detect anomalies in streams of events: operationalizing for real-time, stream-processing systems, and offline-processing systems.

NEAR-REAL TIME SESSIONIZATION

With Spark and Hadoop

In this section, we demonstrate and walk through common and advanced Spark Streaming functionality via the use case of doing near-real-time sessionization of website events, then loading stats about that activity into Apache HBase, and finally populating graphs in your preferred BI tool for analysis.

APACHE KAFKA FOR BEGINNERS

Apache Kafka is creating a lot of buzz these days. While LinkedIn, where Kafka was founded, is the most well-known user, there are actually many companies successfully deploying the technology. Regardless of use cases and components involved, this infrastructure should be compliance-ready with respect to security, data lineage, and metadata management.

TRANSLATE MAPREDUCE TO SPARK

Hadoop was originally designed for large-scale log processing and batch-oriented ETL operations. With broadening Hadoop usage today, alternative architectures like Impala and Spark have been created to accommodate new operations. Spark has grown so much that it is poised to succeed MapReduce as Hadoop’s general-purpose computation paradigm.

Fortunately, this section of the eBook explains how it’s entirely possible to re-implement MapReduce-like computations in Spark.

TUNE APACHE SPARK JOBS

When writing Apache Spark code and paging through the public APIs, one often comes across words such as “transformation, action, and RDD.”

Similarly, if things start to fail, or the application takes an inordinate amount of time, a new vocabulary of words like job, stage, and task get thrown around. To reiterate, understanding Spark at this level is essential in executing good Spark programs—and by good, we mean FAST!

In this section, we cover the basics of how Spark programs are actually executed on a cluster, followed by practical recommendations on the capacity that Spark’s execution model carries for writing efficient programs.

SUMMARY

If modern strategies in data engineering are interesting to you, download this eBook and feel free to reach out to Cloudera with any questions you may have.

ABOUT CLOUDERA

Cloudera delivers the world’s fastest, easiest, and most secure platform for data management and analytics, built on Apache Hadoop and the latest open source technologies.

CLOUDERA

www.cloudera.com
For any business that wants to successfully compete in today’s digital economy, it is not a question of if but rather how they need to evolve their business to survive. Today’s customers won’t give you a second chance when it comes to digital customer experiences. If you fail to deliver on their expectations, they will leave and not come back.

Providing an amazing customer experience is a must. To do this for your cloud applications that provide real-time value at epic scale to your customers, you need a database platform that is distributed, highly responsive, and intelligent.

Why NoSQL over RDBMS?

According to The Forrester Wave™: Big Data NoSQL, Q3 2016 report, a survey of more than 2,000 data and analytics technology decision makers found that more than 60% of enterprises already have implemented, plan on implementing, or are expanding/upgrading their implementation of NoSQL solutions within the next 12 months. “NoSQL is not an option—it has become a necessity to support next-generation applications,” wrote Noel Yuhanna, Principal Analyst at Forrester Research.

What do cloud applications need?

A cloud application is defined as an application with many endpoints, including browsers, mobile devices, and/or machines, that are geographically distributed, intensely transactional, always available, and instantaneously responsive no matter the number of users or machines using the application. Customers expect personalized information at their fingertips and the ability to take action when and where they want.

While each cloud application is unique, these are the foundational set of database requirements that must be met for the business to successfully compete in the market.

- Distributed to ensure 100% uptime
- Responsive to minimize latency and linearly scale up or down as the business demands
- Intelligent to accommodate different types of data models and workloads, while managing issues as they arise

And, all of this has to be in a single, secure, enterprise-ready platform that can be intelligently managed and monitored with ease.

RDBMS technologies fail to deliver on these expectations and businesses are turning to NoSQL databases to handle these requirements.

Harness the true power of NoSQL with DataStax Enterprise (DSE)

Only DataStax Enterprise lets you harness the true power of a multi-model NoSQL database to deliver an always-on customer experience, faster performance, and powerful contextual recommendations. All this while supporting agile DevOps with management tools that make it easy for the Ops team to monitor and operate your production environments.

While high availability may be achieved with legacy RDBMS failover procedures, true continuous availability for a cloud application requires an architecture that was designed to never allow single points of failure. DataStax Enterprise builds on the core architecture of Apache Cassandra™, which sports a masterless design where every node in a database cluster operates independently with respect to database operations. This results in 100% uptime. Not 99.99% uptime, 100% uptime.

DataStax Enterprise—Solve the Toughest Digital Customer Experience Challenges

“DataStax keeps us in business,” says Christos Kalantzis, Cloud Database Engineering Manager at Netflix. He recounted a horror story about how Netflix went down for more than 48 hours when its Oracle database failed. “We couldn’t risk that happening again. Oracle wasn’t built for the cloud and it doesn’t work in the cloud at the level we need it to.”

Digital natives and 100-year-old enterprises alike are already starting to deliver value derived from connected, not collected data. This ability to understand a customer’s interactions across business silos and digital channels, and to personalize and offer highly relevant and contextual experiences is fast becoming the norm.

Customers won’t give your company a second chance to deliver a great digital customer experience—and with DataStax, you won’t need one. Build and manage cloud applications that exceed your customers’ expectations for speed, access, and relevant experiences, by harnessing the true power of a multi-model NoSQL database to deliver real-time value at epic scale.
Future-Proofing Your Big Data Solutions

Big Data turned 10 years old this year and much has changed since the invention of Apache Hadoop. Early on, we saw new projects in the ecosystem develop quickly that filled in major functional areas necessary to effectively process the volume, velocity, and variety of data never before seen. In the years following, additional projects have been developed to round out much of the major features required to enable varied use cases and developer access, and more recently with the rise of Apache Spark.

Developers and organizations don’t want to build integrations—they want to build solutions

As Hadoop adoption started to take off, vendors began delivering specialized integration tools—for ETL, data integration, security, governance, etc.—to simplify the development of big data solutions. However, in most cases today, creating a big data application requires integrating between a large and growing number of these individual tools, forcing big data developers to shift their attention from application and data logic—where most of the IP and value resides—to figuring out integration logic. As a result, developers and organizations end up spending valuable time on infrastructure and integration tasks while dealing with a multitude of vendors and the intricacies of their tools, rather than focusing their energy on applications and insights.

It’s challenging to move from prototype to production

Just like traditional 3-tier applications, distributed data applications don’t just magically appear; they are being built, tested, optimized and staged on their way from a prototype to a production environment. But because of the significant differences between developing code on a laptop or workstation versus the highly distributed multi-node production environment, a significant amount of recoding and configuration changes are often required to accommodate these variances. This dramatically slows time to market. Some of the challenges might not even lie within the skillset of the original developers, especially as production-specific requirements (packaging, versioning, etc.) come into the picture. As big data solutions evolve, organizations need to clear this hurdle, efficiently operationalize data apps from prototype to production, and enable a self-service environment for business users before they can start to derive value from their data.

It’s challenging to move from prototype to production

The solution is unified integration for big data with CDAP

The Cask Data Application Platform (CDAP) has been designed to help solve the issues that arise when moving from prototype to production, whether on-premise or in the cloud, on a laptop or on a 1,000-node server. CDAP is a truly unified integration platform that combines application management and data integration capabilities with a code-free self-service environment and enterprise-grade governance. CDAP ensures data and process consistency between applications and underlying infrastructure technologies across multiple environments and between different parts of the IT organization.

To future-proof the Hadoop applications built on CDAP, the 100% open source and extensible platform acts as an abstraction layer, separating integration logic from application and data logic. Future changes to the data or the business logic of the application become much easier, streamlining ongoing IT operations for big data solutions.

CDAP also includes Cask Hydrator, a drag-and-drop extension to CDAP that enables users to quickly and easily create code-free data pipelines to ingest and transform data in Hadoop, accelerating the process of building enterprise data lakes.

A second CDAP extension called Cask Tracker provides the ability to discover data, audit data access and trace data lineage, all necessary for enterprise governance.

CDAP enables rapid development and deployment of big data applications, including the ability to easily move them into production, while meeting stringent governance requirements. CDAP is the only platform you need not only to ensure your Hadoop environment is on an efficient and safe path to production, but also to easily evolve along with the ecosystem while future-proofing the big data solutions you are building today. For more information about these products, please go to the Cask website, and to stay updated on product and company news, follow us @caskdata.

CASK
www.cask.com