What's the Difference Between Hadoop and Spark?

Apache Hadoop and Apache Spark are two open-source frameworks you can use to manage and process large volumes of data for analytics. Organizations must process data at scale and speed to gain real-time insights for business intelligence. Apache Hadoop allows you to cluster multiple computers to analyze massive datasets in parallel more quickly. Apache Spark uses in-memory caching and optimized query execution for fast analytic queries against data of any size. Spark is a more advanced technology than Hadoop, as Spark uses artificial intelligence and machine learning (AI/ML) in data processing. However, many companies use Spark and Hadoop together to meet their data analytics goals.

What are the similarities between Hadoop and Spark?

Both Hadoop and Spark are distributed systems that let you process data at scale. They can recover from failure if data processing is interrupted for any reason.

Distributed big data processing

Big data is collected frequently, continuously, and at scale in various formats.

To store, manage, and process big data, Apache Hadoop separates datasets into smaller subsets or partitions. It then stores the partitions over a distributed network of servers. Likewise, Apache Spark processes and analyzes big data over distributed nodes to provide business insights.

Depending on the use cases, you might need to integrate both Hadoop and Spark with different software for optimum functionality.

Fault tolerance

Apache Hadoop continues to run even if one or several data processing nodes fail. It makes multiple copies of the same data block and stores them across several nodes. When a node fails, Hadoop retrieves the information from another node and prepares it for data processing.

Meanwhile, Apache Spark relies on a special data processing technology called *Resilient Distributed Dataset* (RDD). With RDD, Apache Spark remembers how it retrieves specific information from storage and can reconstruct the data if the underlying storage fails.

Key components: Hadoop vs. Spark frameworks

Both Hadoop and Spark are made of several software modules that interact and collaborate to make the system work.

Hadoop components

Apache Hadoop has four main components:

- Hadoop Distributed File System (HDFS) is a special file system that stores large datasets across multiple computers. These computers are called *Hadoop clusters*.
- Yet Another Resource Negotiator (YARN) schedules tasks and allocates resources to applications running on Hadoop.
- Hadoop MapReduce allows programs to break large data processing tasks into smaller ones and runs them in parallel on multiple servers.
- Hadoop Common, or Hadoop Core, provides the necessary software libraries for other Hadoop components.

Spark components

Apache Spark runs with the following components:

- Spark Core coordinates the basic functions of Apache Spark. These functions include memory management, data storage, task scheduling, and data processing.
- Spark SQL allows you to process data in Spark's distributed storage.
- Spark Streaming and Structured Streaming allow Spark to stream data efficiently in real time by separating data into tiny continuous blocks.
- Machine Learning Library (MLlib) provides several machine learning algorithms that you can apply to big data.
- GraphX allows you to visualize and analyze data with graphs.

Key differences: Hadoop vs. Spark

Both Hadoop and Spark allow you to process big data in different ways.

Apache Hadoop was created to delegate data processing to several servers instead of running the workload on a single machine.

Meanwhile, Apache Spark is a newer data processing system that overcomes key limitations of Hadoop. Despite its ability to process large datasets, Hadoop only does so in batches and with substantial delay.

Architecture

Hadoop has a native file system called Hadoop Distributed File System (HDFS). HDFS lets Hadoop divide large data blocks into multiple smaller uniform ones. Then, it stores the small data blocks in server groups.

Meanwhile, Apache Spark does not have its own native file system. Many organizations run Spark on Hadoop’s file system to store, manage, and retrieve data.

Alternatively, you can also use Amazon Redshift or Amazon Simple Storage Service (Amazon S3) as data storage options for Spark.

Performance

Hadoop can process large datasets in batches but may be slower. To process data, Hadoop reads the information from external storage and then analyzes and inputs the data to software algorithms.

For each data processing step, Hadoop writes the data back to the external storage, which increases latency. Hence, it is unsuitable for real-time processing tasks but ideal for workloads with tolerable time delays. For example, Hadoop is suited to analyze monthly sales records. But it may not be the best choice to determine real-time brand sentiment from social media feeds.

Apache Spark, on the other hand, is designed to process enormous amounts of data in real time.

Instead of accessing data from external storage, Spark copies the data to RAM before processing it. It only writes the data back to external storage after completing a specific task. Writing and reading from RAM are exponentially faster than doing the same with an external drive. Moreover, Spark reuses the retrieved data for numerous operations.

Therefore, Spark performs better than Hadoop in varying degrees for both simple and complex data processing.

Machine learning

Apache Spark provides a machine learning library called MLlib. Data scientists use MLlib to run regression analysis, classification, and other machine learning tasks. You can also train machine learning models with unstructured and structured data and deploy them for business applications.

In contrast, Apache Hadoop does not have built-in machine learning libraries. Instead, you can integrate Spark with other software like Apache Mahout to build machine learning systems. The choice of software depends on the specific requirements of the workload. You can consider things like the size and complexity of data, the type of machine learning models you want to use, and the performance and scalability requirements of your application.

Security

Apache Hadoop is designed with robust security features to safeguard data. For example, Hadoop uses encryption and access control to prevent unauthorized parties from accessing and manipulating data storage.

Apache Spark, however, has limited security protections on its own. According to Apache Software Foundation, you must enable Spark’s security feature and ensure that the environment it runs on is secure.

Spark supports several deployment types, with some more secure than others. For example, deploying Spark on Hadoop improves overall security due to Hadoop's encrypted distributed storage.

Scalability

It takes less effort to scale with Hadoop than Spark. If you need more processing power, you can add additional nodes or computers on Hadoop at a reasonable cost.

In contrast, scaling the Spark deployments typically requires investing in more RAM. Costs can add up quickly for on-premises infrastructure.

Cost

Apache Hadoop is more affordable to set up and run because it uses hard disks for storing and processing data. You can set up Hadoop on standard or low-end computers.

Meanwhile, it costs more to process big data with Spark as it uses RAM for in-memory processing. RAM is generally more expensive than a hard disk with equal storage size.

When to use Hadoop vs. Spark

Apache Spark was introduced to overcome the limitations of Hadoop's external storage-access architecture. Apache Spark replaces Hadoop's original data analytics library, MapReduce, with faster machine learning processing capabilities.

However, Spark is not mutually exclusive with Hadoop. While Apache Spark can run as an independent framework, many organizations use both Hadoop and Spark for big data analytics.

Depending on specific business requirements, you can use Hadoop, Spark, or both for data processing. Here are some things you might consider in your decision.

Cost-effective scaling

Apache Hadoop is the better option for building and scaling a cost-effective data processing pipeline. Adding more computers to an existing Hadoop cluster can increase Hadoop's processing capacity. This is more affordable than purchasing additional RAM to scale the Apache Spark framework.

Batch processing

Batch processing refers to when you process large numbers of data without being confined to a stipulated timeline. When batch processing is preferred, organizations use Apache Hadoop because it supports parallel processing across multiple nodes. For example, you can use Hadoop to generate non-time-sensitive inventory reports from tens of thousands of records.

Real-time analytics

Use Apache Spark if you're dealing with fast-moving data. A data stream is information or data transmitted continuously by software. Apache Spark can process live data streams and provide insightful analysis in real time. For example, financial institutions use Apache Spark to detect fraud in ongoing transactions and alert bank officers.

Read about streaming data »

Machine learning capability

Machine learning involves training software functions or models with large numbers of datasets. Apache Spark is more suitable for such tasks because of its built-in machine learning library. This means Spark can train machine learning models in real time without additional integrations.

Security, speed, and interactive analytics

You can use Hadoop and Spark to benefit from the strengths of both frameworks. Hadoop provides secure and affordable distributed processing. If you run Spark on Hadoop, you can shift time-sensitive workloads, such as graph analytics tasks, to Spark's in-memory data processors. You get performance and secure external storage processing for your analytics.

Summary of differences: Hadoop vs. Spark

	Hadoop	**Spark**
Architecture	Hadoop stores and processes data on external storage.	Spark stores and process data on internal memory.
Performance	Hadoop processes data in batches.	Spark processes data in real time.
Cost	Hadoop is affordable.	Spark is comparatively more expensive.
Scalability	Hadoop is easily scalable by adding more nodes.	Spark is comparatively more challenging.
Machine learning	Hadoop integrates with external libraries to provide machine learning capabilities.	Spark has built-in machine learning libraries.
Security	Hadoop has strong security features, storage encryption, and access control.	Spark has basic security. IT relies on you setting up a secure operating environment for the Spark deployment.

How can AWS support your big data workloads?

Amazon EMR is an online platform that helps you build, deploy, and scale big data solutions affordably. It supports various open-source big data frameworks, including Apache Hadoop and Spark. Organizations use Amazon EMR for petabyte (PB)-scale data processing, interactive analytics, and machine learning applications.

Here are other ways you can benefit from using Amazon EMR:

- Amazon EMR automatically scales the compute resources your big data application needs
- Running big data applications on Amazon EMR costs less than half of on-premises infrastructure
- Amazon EMR allows you to store big datasets on data stores besides Hadoop Distributed File System (HDFS). For example, you can store on Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.