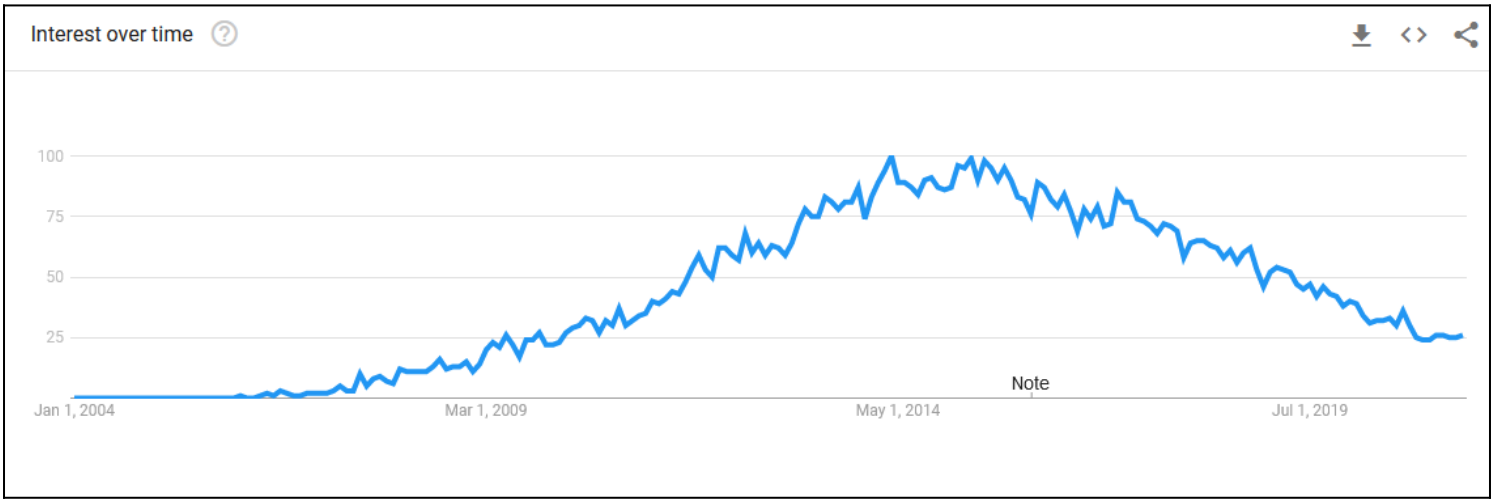
HADOOP VS KUBERNETES: WILL K8S & CLOUD NATIVE END HADOOP?

Apache Hadoop is one of the leading solutions for distributed data analytics and data storage. However, with the introduction of other distributed computing solutions directly aimed at data analytics and general computing needs, Hadoop's usefulness has been called into question.

There are many debates on the internet: is Hadoop still relevant? Or, is it dead altogether?

In reality, Apache Hadoop is not dead, and many organizations are still using it as a robust data analytics solution. One key indicator is that all major cloud providers are actively supporting Apache Hadoop clusters in their respective platforms.

Google Trends shows how interest in Hadoop reached its peak popularity from 2014 to 2017. After that, we see a clear decline in searches for Hadoop. However, this alone is not a good measurement of Hadoop's usage in the current landscape. After all, Hadoop can be integrated into other platforms to form a complete analytics solution.



In this article, we will learn more about Hadoop, its usability, and whether it will be replaced by rapidly evolving technologies like Kubernetes and Cloud-Native development.

(This article is part of our Hadoop Guide. Use the right-hand menu to navigate.)

WHAT IS HADOOP?

Hadoop is an open-source framework that is used to store and process massive datasets efficiently. It is a reliable and scalable distributed computing platform that can be used on commodity hardware.

Hadoop distributes its data storage and analytics workloads across multiple nodes (computers) to handle the work parallelly. This leads to faster, highly efficient, and low-cost data analytics capabilities.

Hadoop modules

Hadoop consists of four main modules that power its functionality:

- **HDFS.** Hadoop Distributed File System is a file system that can run on low-end hardware while providing better throughput than traditional file systems. Additionally, it has built-in fault tolerance and the ability to handle large datasets.
- **YARN.** "Yet Another Resource Negotiator" is used for task management, scheduling jobs, and resource management of the cluster.
- **MapReduce.** MapReduce is a big data processing engine that supports the parallel computation of large data sets. It is the default processing engine available on Hadoop. Currently, however, Hadoop also provides support for other engines such as Apache Tez and Apache Spark.
- **Hadoop Common.** Hadoop Common provides a common set of libraries that can be used across all the other Hadoop modules.

Hadoop benefits

Now, let's look at some top reasons behind the popularity of Apache Hadoop.

- **Processing power.** Hadoop's distributed computing model allows it to handle limitless concurrent tasks.

- **Data safety.** Hadoop automatically creates and manages data backups. So, you can simply recover your data from a backup in case of a failure.
- **Cost.** Hadoop's ability to run on commodity hardware enables organizations to easily deploy a data analytics platform using it. It also eliminates the need for expensive and specialized hardware.
- **Availability.** Hadoop is designed to handle failures at the application layer—which means it provides high availability without relying on hardware.

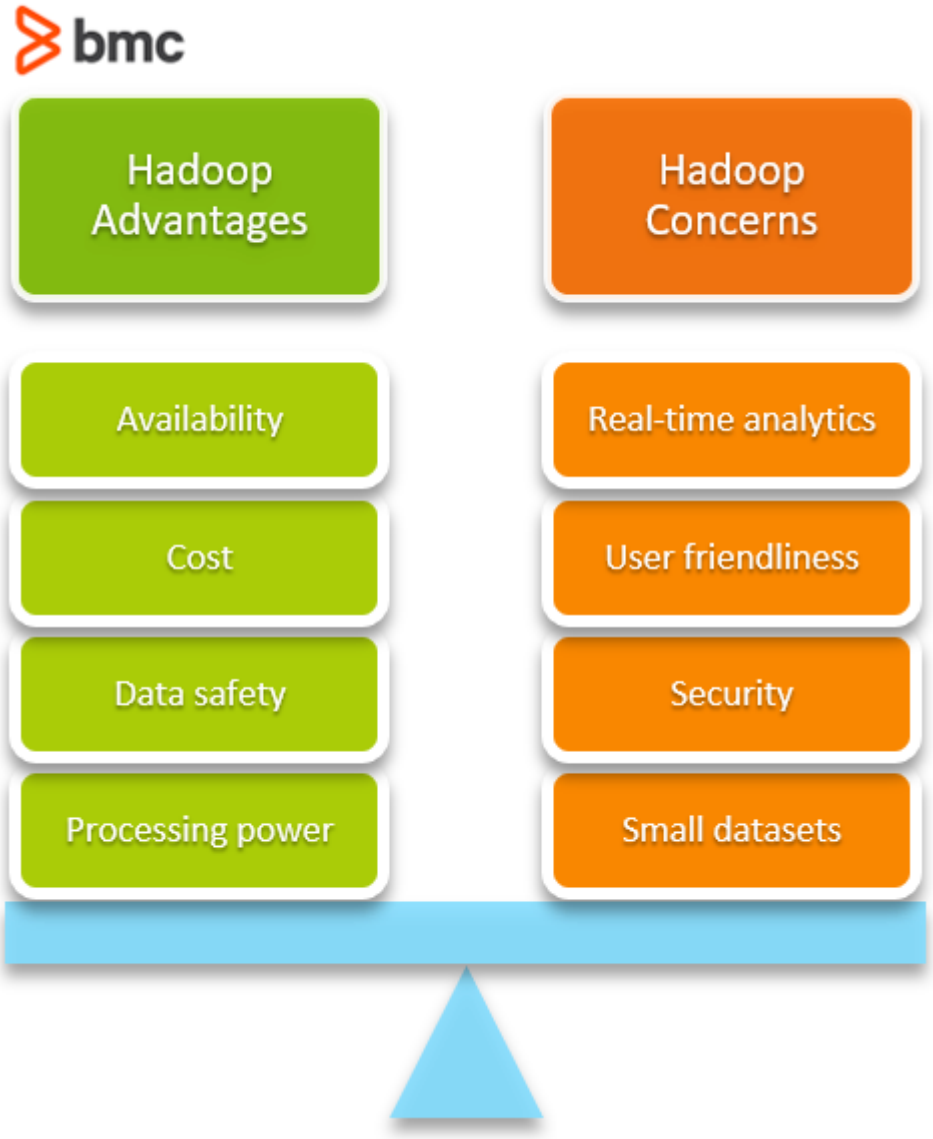
With its flexibility and scalability, Hadoop quickly gained the favor of both individual data engineers/analysts and corporations. This flexibility extends to types of data Hadoop can collect:

- Structured and unstructured
- Through different inputs like social media, streamed data, internal collections, etc.

Then Hadoop checks all these data sets and determines the usefulness of each data set. All this is done *without* having to go through the process of converting data into a single format.

Another feature that elevates Hadoop is its storage capability.

Once a large data set is accumulated, and the required data is extracted, we can simply store the unprocessed data with Hadoop endlessly. This enables users to reference older data easily, and the storage costs are also minimal since Hadoop is running on commodity hardware.



Drawbacks of Hadoop

Apache Hadoop clusters gained prominence thanks to all the above features.

However, as technology advances, new options have emerged, challenging Hadoop and even surpassing it in certain aspects. This, along with the inherent limitations of Hadoop, means it has indeed lost its market lead.

So, what are some drawbacks of Hadoop?

Inefficient for small data sets

Hadoop is designed for processing big data composed of huge data sets. It is very inefficient when processing smaller data sets. Hadoop is not suited and cost-prohibitive when it comes to quick analytics of smaller data sets.

Another reason: Although Hadoop can combine, process, and transform data, it does not provide an easy way to output the necessary data. This limits the available options for business intelligence teams for visualizing and reporting on the processed data sets.

Security concerns

Hadoop includes lax security enforcement by default and does not implement encryption decryption at the storage or network levels. Only Kerberos authentication is officially supported by Hadoop, which is a technology that is difficult to maintain by itself.

In each Hadoop configuration, users need to manually enable security options or use third-party tools to configure secure clusters.

Lack of user friendliness

Hadoop is developed using Java, one of the leading programming languages with a large developer base. However, Java is not the best language for data analytics, and it can be complex for new users.

This can lead to complications in configurations and usage—the user must have thorough knowledge in both Java and Hadoop to properly use and debug the cluster.

Not suitable for real-time analytics

Hadoop is designed with excellent support for batch processing. However, with its limitations in processing smaller data sets *and* not providing native support for real-time analytics, Hadoop is ill-suited for quick real-time analytics.

Hadoop alternatives

So, what other options to Hadoop are available? While there is no single solution to replace Hadoop outright, there are newer technologies that can reduce or eliminate the need for Hadoop.

Apache Spark

Apache Spark is one solution, provided by the Apache team itself, to replace MapReduce, Hadoop's default data processing engine. Spark is the new data processing engine developed to address the limitations of MapReduce.

Apache claims that Spark is nearly 100 times faster than MapReduce and supports in-memory calculations. Moreover, it supports real-time processing by creating micro-batches of data and processing them.

The support of Spark for modern languages enables you to interact using your preferred programming languages. Spark offers excellent support for data analytics using languages such as:

- Scala
- Python
- Spark SQL

(Explore our Apache Spark Guide.)

Apache Flink

Another available solution is Apache Flink. Flink is another processing engine with the same benefits as Spark. Flink offers even higher performance in some workloads as it is designed to handle stateful computation over unbounded and bounded data streams.

Will Kubernetes & cloud-native replace Hadoop?

Even with newer and faster data process engines, Hadoop still limits users to its tools and technologies like HDFS and YARN with Java-based tools. But, what if you need to integrate other tools and platforms to get the best for your specfic data storage and analytics needs?

The solution is using Kubernetes as the orchestration engine to manage your cluster.

With the ever-growing popularity of containerized cloud-native applications, Kubernetes has become the leading orchestration platform to manage any containerized application. It offers features such as:

- Convenient management
- Networking
- Scaling
- High availability

(Explore our comprehensive Kubernetes Guide.)

Consider this scenario: you want to move to cheap cloud storage options like Amazon S3 buckets and managed data warehouses like Amazon Redshift, Google BigQuery, Panoply. This is not possible with Hadoop.

Kubernetes, meanwhile, can easily plug them into Kubernetes clusters to be accessed by the containers. Likewise, Kubernetes clusters have limitless storage with reduced maintenance responsibilities as cloud providers manage all the day-to-day maintenance and availability of data.

Having the storage sorted, Kubernetes can host different services such as:

- Big Data analytics tools (Apache Spark, Presto, Flink)
- Data science tools (BigML, Jupyter, NLTK, TensorFlow, PyTorch, MATLAB)
- Any other tool within a Kubernetes cluster

This gives you the freedom to use any tools, frameworks, or programming languages you're already familiar with or the one that's most suitable for your use case—you're no longer limited to Java.

(See exactly how containers & K8s work together.)

Portability of Kubernetes

Another factor that uplifts Kubernetes is its portability. Kubernetes can be easily configured to be distributed across many locations and run on multiple cloud environments. With containerized applications, users can easily move between development and production environments to facilitate data analytics in any location without major modifications.

By combining Kubernetes with rapid DevOps and CI/CD pipelines, developers can easily create, test, and deploy data analytics, ML, and AI applications virtually anywhere.

Support of Kubernetes for Serverless Computing

Kubernetes has further eliminated the need to manage infrastructure separately with the support for serverless computing. Serverless computing is a rising technology where the cloud platform automatically manages and scales the hardware resources according to the needs of the application.

Some container-native, open-source, and function-as-a-service computing platforms like fn, Apache OpenWhisk, and nuclio can be easily integrated with Kubernetes to run serverless applications—eliminating the need for technologies like Hadoop.

Some frameworks, like nuclio, are specifically aimed at automating data science pipelines with serverless functions.

With all the above-mentioned advantages, Kubernetes is gradually becoming the perfect choice for managing any big data workloads.

Hadoop handles large data sets cheaply

Like any other technology, Hadoop is also designed to address a specific need—handling large datasets efficiently using commodity hardware.

However, evolving technology trends have given rise to new requirements and use cases. Hadoop is not dead, yet other technologies, like Kubernetes and serverless computing, offer much more flexible and efficient options.

So, like any technology, it's up to you to identify and utilize the correct technology stack for your needs.

Related reading

- BMC Machine Learning & Big Data Blog

- Data Architecture Explained: Components, Standards & Changing Architectures
- 3 Keys to Building Resilient Data Pipelines
- Snowflake 101: Intro to the Snowflake Data Cloud
- Big Data: A Big Introduction
- Data Ethics for Companies