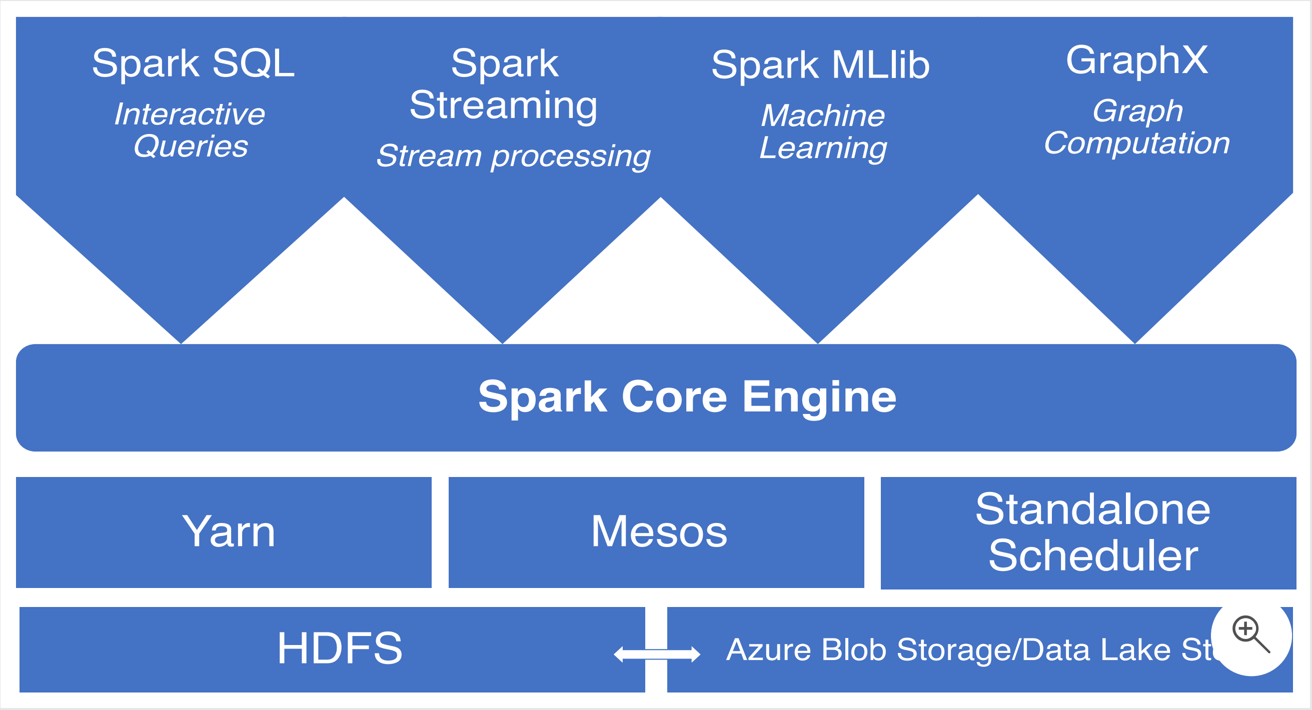
What is Apache Spark in Azure HDInsight

Article • 04/24/2023

Apache Spark is a parallel processing framework that supports in-memory processing to boost the performance of big-data analytic applications. Apache Spark in Azure HDInsight is the Microsoft implementation of Apache Spark in the cloud, and is one of several Spark offerings in Azure.

- Apache Spark in Azure HDInsight makes it easy to create and configure Spark clusters, allowing you to customize and use a full Spark environment within Azure.
- Spark pools in Azure Synapse Analytics use managed Spark pools to allow data to be loaded, modeled, processed, and distributed for analytic insights within Azure.
- Apache Spark on Azure Databricks uses Spark clusters to provide an interactive workspace that enables collaboration between your users to read data from multiple data sources and turn it into breakthrough insights.
- Spark Activities in Azure Data Factory allow you to use Spark analytics in your data pipeline, using on-demand or pre-existing Spark clusters.

With Apache Spark in Azure HDInsight, you can store and process your data all within Azure. Spark clusters in HDInsight are compatible with Azure Blob storage, Azure Data Lake Storage Gen1, or Azure Data Lake Storage Gen2, allowing you to apply Spark processing on your existing data stores.



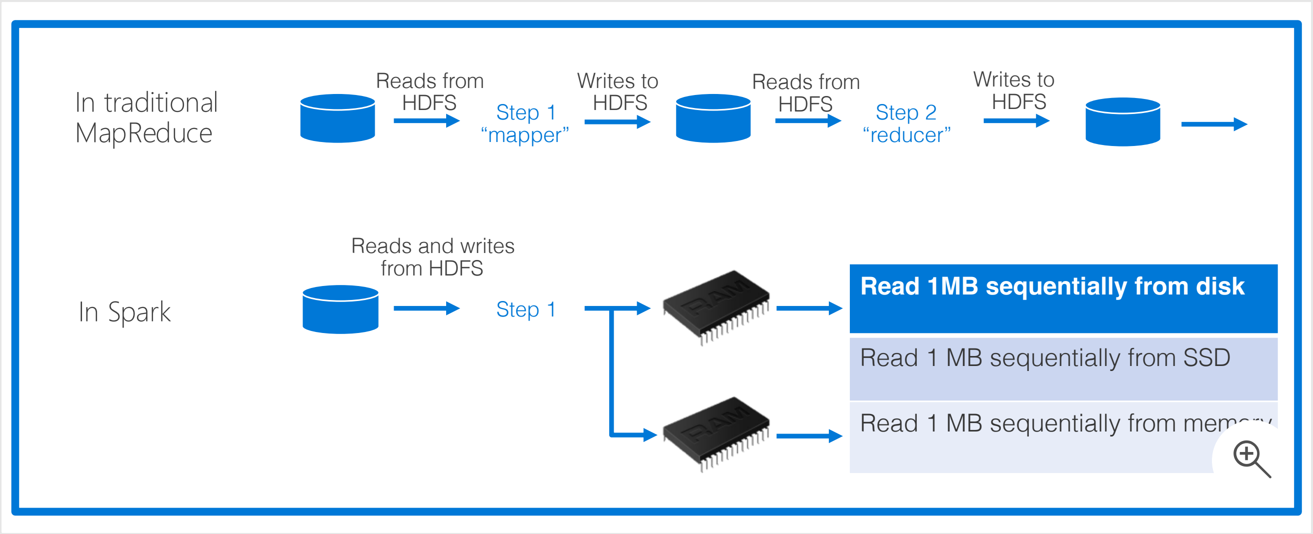
To get started with Apache Spark in Azure HDInsight, follow our tutorial to create HDInsight Spark clusters.

For information about Apache Spark and how it interacts with Azure, continue reading the article below.

For the components and the versioning information, see Apache Hadoop components and versions in Azure HDInsight.

What is Apache Spark?

Spark provides primitives for in-memory cluster computing. A Spark job can load and cache data into memory and query it repeatedly. In-memory computing is much faster than disk-based applications, such as Hadoop, which shares data through Hadoop distributed file system (HDFS). Spark also integrates into the Scala programming language to let you manipulate distributed data sets like local collections. There's no need to structure everything as map and reduce operations.



Spark clusters in HDInsight offer a fully managed Spark service. Benefits of creating a Spark cluster in HDInsight are listed here.

Expand table

Feature	Description
Ease creation	You can create a new Spark cluster in HDInsight in minutes using the Azure portal, Azure PowerShell, or the HDInsight .NET SDK. See Get started with Apache Spark cluster in HDInsight.
Ease of use	Spark cluster in HDInsight include Jupyter Notebooks and Apache Zeppelin Notebooks. You can use these notebooks for interactive data processing and visualization. See Use Apache Zeppelin notebooks with Apache Spark and Load data and run queries on an Apache Spark cluster.
REST APIs	Spark clusters in HDInsight include Apache Livy , a REST API-based Spark job server to remotely submit and monitor jobs. See Use Apache Spark REST API to submit remote jobs to an HDInsight Spark cluster.
Support for	Spark clusters in HDInsight can use Azure Data Lake Storage Gen1/Gen2 as both the

Support for Azure Storage	Spark clusters in HDInsight can use Azure Data Lake Storage Gen1/Gen2 as both the primary storage or additional storage. For more information on Data Lake Storage Gen1, see Azure Data Lake Storage Gen1. For more information on Data Lake Storage Gen2, see Azure Data Lake Storage Gen2.
Integration with Azure services	Spark cluster in HDInsight comes with a connector to Azure Event Hubs. You can build streaming applications using the Event Hubs. Including Apache Kafka, which is already available as part of Spark.
Integration with third-party IDEs	HDInsight provides several IDE plugins that are useful to create and submit applications to an HDInsight Spark cluster. For more information, see Use Azure Toolkit for IntelliJ IDEA, Use Spark & Hive Tools for VSCode, and Use Azure Toolkit for Eclipse.
Concurrent Queries	Spark clusters in HDInsight support concurrent queries. This capability enables multiple queries from one user or multiple queries from various users and applications to share the same cluster resources.
Caching on SSDs	You can choose to cache data either in memory or in SSDs attached to the cluster nodes. Caching in memory provides the best query performance but could be expensive. Caching in SSDs provides a great option for improving query performance without the need to create a cluster of a size that is required to fit the entire dataset in memory. See Improve performance of Apache Spark workloads using Azure HDInsight IO Cache.
Integration with BI Tools	Spark clusters in HDInsight provide connectors for BI tools such as Power BI for data analytics.
Pre-loaded Anaconda libraries	Spark clusters in HDInsight come with Anaconda libraries pre-installed. Anaconda provides close to 200 libraries for machine learning, data analysis, visualization, and so on.
Adaptability	HDInsight allows you to change the number of cluster nodes dynamically with the Autoscale feature. See Automatically scale Azure HDInsight clusters. Also, Spark clusters can be dropped with no loss of data since all the data is stored in Azure Blob storage, Azure Data Lake Storage Gen1, or Azure Data Lake Storage Gen2.
SLA	Spark clusters in HDInsight come with 24/7 support and an SLA of 99.9% up-time.

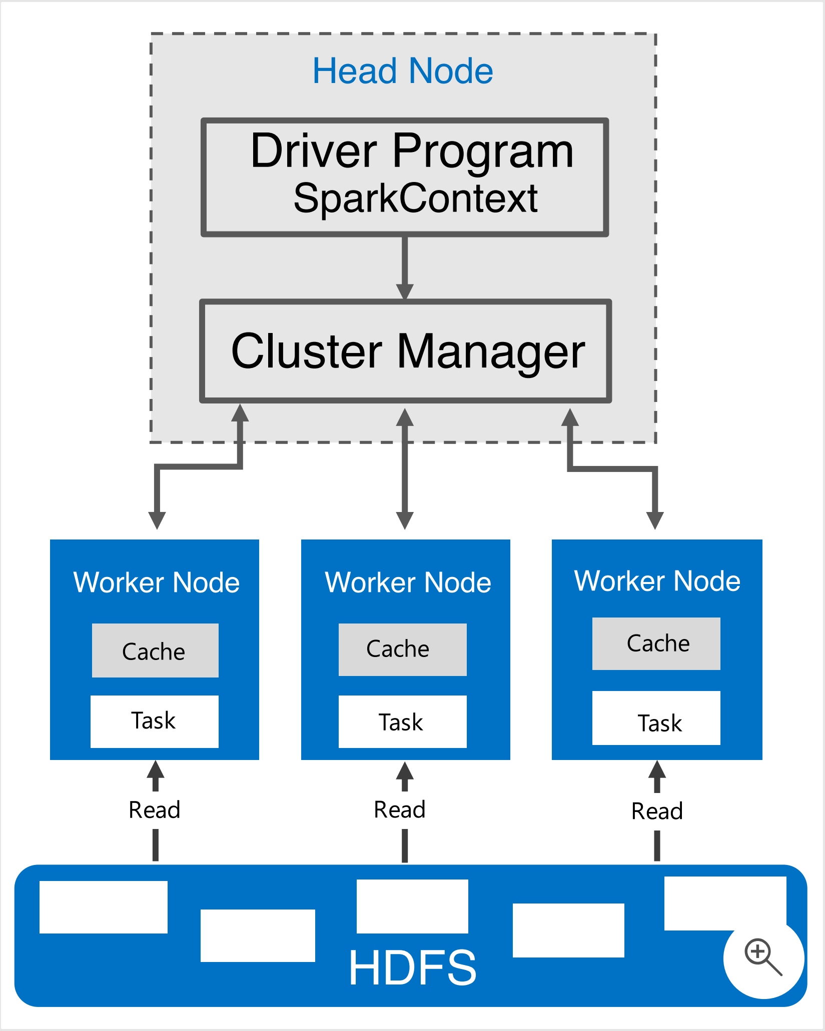
Apache Spark clusters in HDInsight include the following components that are available on the clusters by default.

- Spark Core . Includes Spark Core, Spark SQL, Spark streaming APIs, GraphX, and MLlib.
- Anaconda
- Apache Livy
- Jupyter Notebook
- Apache Zeppelin notebook

HDInsight Spark clusters an ODBC driver for connectivity from BI tools such as Microsoft Power BI.

Spark cluster architecture

Spark cluster architecture



It's easy to understand the components of Spark by understanding how Spark runs on HDInsight clusters.

Spark applications run as independent sets of processes on a cluster. Coordinated by the SparkContext object in your main program (called the driver program).

The SparkContext can connect to several types of cluster managers, which give resources across applications. These cluster managers include Apache Mesos, Apache Hadoop YARN, or the Spark cluster manager. In HDInsight, Spark runs using the YARN cluster manager. Once connected, Spark acquires executors on workers nodes in the cluster, which are processes that run computations and store data for your application. Next, it sends your application code (defined by JAR or Python files passed to SparkContext) to the executors. Finally, SparkContext sends tasks to the executors to run.

The SparkContext runs the user's main function and executes the various parallel operations on the worker nodes. Then, the SparkContext collects the results of the operations. The worker nodes read and write data from and to the Hadoop distributed file system. The worker nodes also cache transformed data in-memory as Resilient Distributed Datasets (RDDs).

The SparkContext connects to the Spark master and is responsible for converting an application to a directed graph (DAG) of individual tasks. Tasks that get executed within an executor process on the worker nodes. Each application gets its own executor processes.

Which stay up during the whole application and run tasks in multiple threads.

Spark in HDInsight use cases

Spark clusters in HDInsight enable the following key scenarios:

Interactive data analysis and BI

Apache Spark in HDInsight stores data in Azure Blob Storage, Azure Data Lake Gen1, or Azure Data Lake Storage Gen2. Business experts and key decision makers can analyze and build reports over that data. And use Microsoft Power BI to build interactive reports from the analyzed data. Analysts can start from unstructured/semi structured data in cluster storage, define a schema for the data using notebooks, and then build data models using Microsoft Power BI. Spark clusters in HDInsight also support many third-party BI tools. Such as Tableau, making it easier for data analysts, business experts, and key decision makers.

- Tutorial: Visualize Spark data using Power BI

Spark Machine Learning

Apache Spark comes with MLlib . MLlib is a machine learning library built on top of Spark that you can use from a Spark cluster in HDInsight. Spark cluster in HDInsight also includes Anaconda, a Python distribution with different kinds of packages for machine learning. And with built-in support for Jupyter and Zeppelin notebooks, you have an environment for creating machine learning applications.

- Tutorial: Predict building temperatures using HVAC data
- Tutorial: Predict food inspection results

Spark streaming and real-time data analysis

Spark clusters in HDInsight offer a rich support for building real-time analytics solutions.

Spark already has connectors to ingest data from many sources like Kafka, Flume, Twitter, ZeroMQ, or TCP sockets. Spark in HDInsight adds first-class support for ingesting data from Azure Event Hubs. Event Hubs is the most widely used queuing service on Azure. Having complete support for Event Hubs makes Spark clusters in HDInsight an ideal platform for building real-time analytics pipeline.

- Overview of Apache Spark Streaming
- Overview of Apache Spark Structured Streaming

Next Steps

In this overview, you've got a basic understanding of Apache Spark in Azure HDInsight. You can use the following articles to learn more about Apache Spark in HDInsight, and you can create an HDInsight Spark cluster and further run some sample Spark queries:

- Quickstart: Create an Apache Spark cluster in HDInsight and run interactive query using Jupyter
- Tutorial: Load data and run queries on an Apache Spark job using Jupyter
- Tutorial: Visualize Spark data using Power BI
- Tutorial: Predict building temperatures using HVAC data
- Optimize Spark jobs for performance