Choosing a batch processing technology in Azure

11/03/2018 • 3 minutes to read • Contributors

In this article

Technology choices for batch processing
Key selection criteria
Capability matrix

Big data solutions often use long-running batch jobs to filter, aggregate, and otherwise prepare the data for analysis. Usually these jobs involve reading source files from scalable storage (like HDFS, Azure Data Lake Store, and Azure Storage), processing them, and writing the output to new files in scalable storage.

The key requirement of such batch processing engines is the ability to scale out computations, in order to handle a large volume of data. Unlike real-time processing, however, batch processing is expected to have latencies (the time between data ingestion and computing a result) that measure in minutes to hours.

Technology choices for batch processing

Azure SQL Data Warehouse

SQL Data Warehouse is a distributed system designed to perform analytics on large data. It supports massive parallel processing (MPP), which makes it suitable for running high-performance analytics. Consider SQL Data Warehouse when you have large amounts of data (more than 1 TB) and are running an analytics workload that will benefit from parallelism.

Azure Data Lake Analytics

Data Lake Analytics is an on-demand analytics job service. It is optimized for distributed processing of very large data sets stored in Azure Data Lake Store.

- Languages: U-SQL (including Python, R, and C# extensions).
- Integrates with Azure Data Lake Store, Azure Storage blobs, Azure SQL Database, and SQL Data Warehouse.
- Pricing model is per-job.

HDInsight

HDInsight is a managed Hadoop service. Use it deploy and manage Hadoop clusters in Azure. For batch processing, you can use Spark, Hive, Hive LLAP, MapReduce.

- Languages: R, Python, Java, Scala, SQL
- Kerberos authentication with Active Directory, Apache Ranger based access control
- Gives you full control of the Hadoop cluster

Azure Databricks

Azure Databricks is an Apache Spark-based analytics platform. You can think of it as "Spark as a service." It's the easiest way to use Spark on the Azure platform.

- Languages: R, Python, Java, Scala, Spark SQL
- Fast cluster start times, autotermination, autoscaling.
- Manages the Spark cluster for you.
- Built-in integration with Azure Blob Storage, Azure Data Lake Storage (ADLS), Azure SQL Data Warehouse (SQL DW), and other services. See Data Sources.
- User authentication with Azure Active Directory.
- Web-based notebooks for collaboration and data exploration.
- Supports GPU-enabled clusters

Azure Distributed Data Engineering Toolkit

The Distributed Data Engineering Toolkit (AZTK) is a tool for provisioning on-demand Spark on Docker clusters in Azure.

AZTK is not an Azure service. Rather, it's a client-side tool with a CLI and Python SDK interface, that's built on Azure Batch. This option gives you the most control over the infrastructure when deploying a Spark cluster.

- Bring your own Docker image.
- Use low-priority VMs for an 80% discount.
- Mixed mode clusters that use both low-priority and dedicated VMs.
- Built in support for Azure Blob Storage and Azure Data Lake connection.

Key selection criteria

To narrow the choices, start by answering these questions:

- Do you want a managed service rather than managing your own servers?
- Do you want to author batch processing logic declaratively or imperatively?
- Will you perform batch processing in bursts? If yes, consider options that let you auto-terminate the cluster or whose pricing model is per batch job.
- Do you need to query relational data stores along with your batch processing, for example to look up reference data? If yes, consider the options that enable querying of external relational stores.

Capability matrix

The following tables summarize the key differences in capabilities.

General capabilities

Capability	Azure Data Lake Analytics	Azure SQL Data Warehouse	HDInsight	Azure Databricks
Is managed service	Yes	Yes	Yes [1]	Yes
Relational data store	Yes	Yes	No	No
Pricing model	Per batch job	By cluster hour	By cluster hour	Databricks Unit[2] + cluster hour

[1] With manual configuration and scaling.

[2] A Databricks Unit (DBU) is a unit of processing capability per hour.

Capabilities

Capability	Azure Data Lake Analytics	SQL Data Warehouse	HDInsight with Spark	HDInsight with Hive	HDInsight with Hive LLAP	Azure Databricks
Autoscaling	No	No	No	No	No	Yes
Scale-out granularity	Per job	Per cluster	Per cluster	Per cluster	Per cluster	Per cluster
In-memory caching of data	No	Yes	Yes	No	Yes	Yes
Query from external relational stores	Yes	No	Yes	No	No	Yes
Authentication	Azure AD	SQL / Azure AD	No	Azure AD[1]	Azure AD[1]	Azure AD
Auditing	Yes	Yes	No	Yes [1]	Yes [1]	Yes
Row-level security	No	Yes[2]	No	Yes [1]	Yes [1]	No
Supports firewalls	Yes	Yes	Yes	Yes [3]	Yes [3]	No
Dynamic data masking	No	No	No	Yes [1]	Yes [1]	No

[1] Requires using a domain-joined HDInsight cluster.

[2] Filter predicates only. See Row-Level Security

[3] Supported when used within an Azure Virtual Network.