Apache Flink vs Spark – Will one overtake the other?

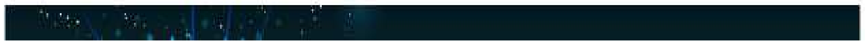
19 Jul 2016

Apache Spark and Apache Flink are both open- sourced, distributed processing framework which was built to reduce the latencies of Hadoop Mapreduce in fast data processing. There is a common misconception that Apache Flink is going to replace Spark or is it possible that both these big data superfluous. But Flink managed to stay ahead in the game because of its stream processing feature, which manages to process rows upon rows of data in real time – which is not possible in Apache Spark's batch processing method. This makes Flink faster than Spark.

According to this IBM study, we are creating about 2.5 quintillion bytes of data

Apache Spark

Spark is an open source, cluster computing framework which has a large global user base. It is written in Scala, Java, R and Python and gives programmers an Application Programming Interface (API) built on a fault tolerant, read only multiset of distributed data items. In a short time of 2 years since its initial release (May 2014), it has seen wide acceptability for real time, in-memory, advanced analytics – owing to its speed, ease of use and the ability to handle sophisticated analytical requirements.



orange "Request info" button on top of this page.

Advantages of Spark

Apache Spark has several advantages over traditional Big Data and MapReduce based technologies. The prominent ones are. It essentially takes MapReduce to the next level with a performance that is several times faster. One of the key differentiators for Spark is its ability to hold intermediate results in-memory itself, rather than writing back to disk and reading from it again, which is critical for iteration based use cases.

- **Speed** – Spark can execute batch processing jobs 10 to 100 times faster than MapReduce. That doesn't mean it lags behind when data has to be written to (and fetched from) disk, as it is the world record holder for large-scale on-disk sorting.

- **Ease of Use** – Apache Sparkhas easy to use APIs, built for operating on large datasets.

- **Unified Engine** – Spark can run on top of Hadoop, making use of its cluster manager (YARN) and underlying storage (HDFS, HBase, etc.). However, it can also run independent of Hadoop, joining hands with other cluster managers and storage platforms (the likes of Cassandra and Amazon S3). It also comes with higher – level libraries that support SQL queries data streaming, machine learning and graph processing.

- **Choose from Java, Scala or Python** – Spark doesn't tie you down to a particular language and lets you choose from the popular ones such as Java, Scala, Python, R and even Clojure.
- **In-memory data sharing** – Different jobs can share data within the memory, which makes it an ideal choice for iterative, interactive and event stream processing tasks.
- **Active, expanding user community** – An active user community has led to a stable release of Spark (in June, 2016) within 2 years of its initial release.

Apache Flink

German for 'quick' or 'nimble', Apache Flink is the latest entrant to the list of open-source frameworks focused on Big Data Analytics that are trying to replace Hadoop's aging MapReduce, just like Spark. Flink got its first API-stable version released in March 2016 and is built for in-memory processing of batch data, just like Spark. This model comes in really handy when repeated passes need to be made on the same data. This makes it an ideal candidate for machine learning and other use cases that require adaptive learning, self-learning networks, etc. With the inevitable boom of Internet of Things (IoT) space, Flink user community has some exciting challenges to look forward to.

Advantages of Flink

- **Actual stream processing engine** that can approximate batch processing, rather than being the other way around.

- **Better memory management** – Explicit memory management gets rid of the occasional spikes found in Spark framework.

- **Speed** – It manages faster speeds by allowing iterative processing to take place on the same node rather than having the cluster run them independently. Its performance can be further tuned by tweaking it to re-process only that part of data that has changed rather than the entire set. It offers up to five-fold boost in speed when compared to the standard processing algorithm.

- **Less configuration**

Apache Flink vs Spark

why would one require another data processing engine while the jury was still out on the existing one? One has to dig deeper into the capabilities of Flink to observe what sets it apart, though a number of analysts have billed it up as the "4G of Data Analytics".

Deeply embedded inside Spark's settings is a little weakness that Flink has targeted and is trying to capitalize upon. Though it stands true for the purpose of casual discussions, Spark is not purely a stream-processing engine. As observed by Ian Pointer in the InfoWorld article 'Apache Flink: New Hadoop contender squares off against Spark', Spark is essentially a fast-batch operation which works on only a small part of incoming data during a time unit. Spark refers to this as "micro batching" in its official documentation. This issue is unlikely to have any practical significance on operations unless the use case requires low latency (financial systems) where delay of the order of milliseconds can cause significant impact. That being said, Flink is pretty much a work in progress and cannot stake claim to replace Spark yet.

Flink is a stream processing framework that can run the chores requiring batch processing, giving you the option to use the same algorithm in both the modes, without having to turn to a technology like Apache Storm that requires low latency response.

Both Spark and Flink support in-memory processing that gives them distinct advantage of speed over other frameworks. When it comes to real time processing of incoming data, Flink does not stand up against Spark, though it has the capability to carry out real time processing tasks.

Spark and Flink both can handle iterative, in memory processing. When it comes to speed, Flink gets the upper hand as it can be programmed to process only the data that has changed, which is where it comes out on top of Spark.

Growth stories – Spark and Flink