| **Feature** | **Hadoop** | **Spark** |
| --- | --- | --- |
| Data storage | Distributed file system (HDFS) | Distributed file system or in-memory |
| Data processing | Batch processing | Batch or stream processing |
| Data processing engines | MapReduce | In-memory engine, others |
| Resource manager | YARN | Spark Standalone, Mesos, YARN, Kubernetes |
| Scalability | Limited scalability | High scalability |
| Speed | Slower than Spark | Faster than Hadoop |
| Flexibility | Limited flexibility | Greater flexibility |

Hadoop and Apache Spark are open-source frameworks used for distributed computing and analysing large amounts of data. Both frameworks are designed to handle large amounts of data, distributed across multiple machines, and to provide scalable and efficient data processing capabilities. However, there are some key differences between Hadoop and Spark that are worth considering when deciding which framework to use for a particular data pipeline.

**Here is a table diagram comparing the features of of Hadoop and Spark**

One key distinction between Hadoop and Spark is the method they use for data processing. Hadoop utilizes batch processing. Where data is processed in large chunks, typically in the form of MapReduce jobs. This means that Hadoop is well-suited for tasks that involve large amounts of data and can be parallelized easily, such as data transformation and aggregation. However, Hadoop can be slow for tasks that require real-time data processing, such as online analytics and machine learning.

In contrast, Spark uses a streaming processing approach, where data is processed in small batches, typically in the form of DStreams (discretized streams). This allows Spark to process data in real-time, providing fast and efficient data processing capabilities. Spark is also well-suited for tasks that require in-memory data processing, such as machine learning and interactive queries, making it a good choice for tasks that require low-latency access to data.

Another key difference between Hadoop and Spark is their programming models. Hadoop uses the MapReduce programming model, which is based on a two-stage approach where data is first mapped to key-value pairs, and then reduced to a smaller set of aggregated results. This programming model is declarative, meaning that the programmer specifies what should be done, but not how it should be done.

In contrast, Spark uses a declarative programming model, where the programmer specifies the transformations and actions that should be performed on the data. This programming model is more flexible and allows for more complex data processing pipelines. Spark also provides support for a wide range of programming languages, including Java, Scala, Python, and R.

In terms of performance, Spark is generally faster than Hadoop, thanks to its in-memory data processing capabilities and its efficient use of memory and disk space. Spark also provides support for more data storage formats than Hadoop, including CSV, JSON, Parquet, and Avro.

In summary, Hadoop and Spark are both distributed computing frameworks that are used for big data processing and analysis. Hadoop uses a batch processing approach and is well-suited for tasks that involve large amounts of data and can be parallelized easily. Spark uses a streaming processing approach and is well-suited for tasks that require real-time data processing and in-memory data processing. Spark is also generally faster than Hadoop and provides support for a wider range of data storage formats.

**References:**

GeeksforGeeks. (2022, September 20). *Difference Between Hadoop and Spark*. <https://www.geeksforgeeks.org/difference-between-hadoop-and-spark/>

Jevtic, G. (2022, September 8). *Hadoop vs Spark – Detailed Comparison*. Knowledge Base by phoenixNAP. <https://phoenixnap.com/kb/hadoop-vs-spark>

*Hadoop vs. Spark: What's the Difference?* (2021, July 15). <https://www.ibm.com/cloud/blog/hadoop-vs-spark>