Assignment - 18
Spark Architecture

HARSH SIDDHAPURA

1230169813

04/09/2024

1. Apache Spark is a powerful open source application engine built on speed, ease of use, and advanced analytics. It has a good design structure with many features such as:
    - **Driver:** This is the heart of the Spark application. Runs the application's main() function and creates a SparkContext. Driver changes and operations are performed here.
    - **SparkContext:** It provides access to all Spark processes. When we run any Spark application, its main role, the driver, is initialized and here your SparkContext is initialized.
    - **Cluster Manager:** This component is responsible for obtaining the resources of the Spark cluster and allocating them to Spark jobs. It can be a single administrator (including Spark) or another administrative group such as Mesos or YARN.
    - **Executors:** When SparkContext connects to a cluster, it retrieves the executors of the nodes in the cluster. Executors are Spark processes that run calculations and store data for your application. They do the work that makes Spark work.
    - **Operation:** A project is a unit of work that can be done on the distribution of information distributed and completed by a manager. SparkContext forwards tasks to executors for execution.
    - **Task:** A task is an instance of multiple tasks (e.g. store, write) created in response to a Spark operation; They are distributed in various packages.
    - **Stages:** The work is divided into several stages. Stages are divided into map or reduce stages (similar to MapReduce - but in Spark operations are done in memory). Stages are separated by changes.
    - **RDD (Resilient Distributed Dataset):** RDD is a simple data model of Spark. It is an immutable, decentralized collection of objects. Each data in the RDD is divided into separate clusters that can be counted in different nodes in the cluster.
    - **DAG planning:** It breaks down worker configuration (mapping and reduction) into phases. Levels contain functions based on input data. The DAG program connects workers to each other. For example, multiple map operators can be programmed in a single stage. This helps develop a successful plan.
    - **Task Scheduler:** It starts running from the controllers group. He knows where objects are and likes to schedule tasks in the same node or on the same shelf as the objects.
    - **Block Manager:** The block manager manages the storage of data in memory or on disk. It deals with local file/disk storage.

- **DataFrame/Dataset API:** Think of this as a smarter, more user-friendly version of RDD (Resilient Distributed Dataset). It is designed to perform data operations similar to working with a library or database using languages such as R or Python. They are especially useful when working with design documents.
- **Spark Core:** This is the backbone of Spark. It provides the core functionality and API (application programming interface) required for distributed computing. You can see the underlying data structure, the schedule of the job, and the current error checking process.
- **Spark SQL:** This module allows you to use SQL (Structured Query Language) queries in Spark. It is designed to work well with existing SQL tools and libraries, so you can execute SQL queries directly against Spark data.
- **Spark Streaming:** This is an extension of the Spark Core API that allows you to stream data in real time. It is designed to make it easy to create and submit submissions for real-time analysis.
- **MLlib (Machine Learning Library):** This is Spark's machine learning library. It is designed to scale and provides a variety of algorithms and tools for machine learning. This includes things like classification, regression, integration, and integration filtering.
- **GraphX:** This is Spark's graph processing library. It allows you to make calculations and analysis on graphs. It is designed to manage image processing at scale and provides a set of APIs for rendering and analyzing image quality.

2. Directed acyclic graph (DAG) is an important concept in mathematics and computer science. It is a directionless map. This means that if you start at a vertex and walk along the edge, you will never return to the starting vertex. The "direction" part of the name comes from the fact that every edge has a direction: it goes from one vertex to another, not the other way around. The "acyclic" part of the name comes from the lack of a cycle, meaning there's no way to start from a vertex, follow the edges, and return to the starting vertex.

   Apache Spark is a distributed computing system that uses DAG to organize and perform calculations. When you send a task to Spark, a DAG is first created that represents the changes your data will undergo. Each part of the DAG represents an RDD (Resilient Distributed Dataset), which is a collection of fault-tolerant elements that can be processed simultaneously. Edges of nodes represent data transformations.

DAGs are "routed" because there is a specific order of transformations: each transformation takes one or more RDDs as input and creates a new RDD as output. DAGs are "acyclic" because each RDD is built from the previous RDD and is not used as input to the previous RDD. When Spark executes a task, it splits the DAG into tasks that can be executed together in a group. This allows Spark to process large amounts of data quickly and efficiently.

SciKit Learn and Spark are powerful tools for data analysis and machine learning, but they are designed for different uses. SciKit Learn is a Python machine learning library that runs in a single piece of memory. This means it does not spread the computation across multiple nodes; This makes it an ideal tool for designing and working with small and medium-sized files. Spark, on the other hand, is designed for distributed computing, which means it can handle larger datasets.

Spark can use multiple nodes to work simultaneously, making it faster than SciKit Learn when processing large datasets. However, Spark's ability to gain speed comes with a trade-off: Spark's work has thrived on distributing data and processing across nodes, so for small data this overhead can make Spark slower than SciKit Learn.

All in all, SciKit Learn is a good choice if you're working with small to medium-sized data and don't need the power of distributed computing. But if you work with big data or need to perform parallel calculations, Spark is your best choice.

3. In machine learning, many models cannot be optimized for performance models. This process is called parameter tuning or hyperparameter optimization. For example, in a random forest classifier, you can set parameters such as the number of trees (n_estimators), maximum tree depth (max_length), and minimum number of samples required, split internal nodes (min_samples_split).

   Parameter tuning can be expensive and time-consuming, especially when dealing with multiple connections or large data sets. This is where decentralized computing comes into play. By utilizing a group of machines, the parameter setting process will be parallelized and the time required to find parameters will be reduced.

   Example: Parameter Tuning for a Random Forest Classifier

   Let’s consider a Random Forest classifier. Suppose we want to try out the following parameter values:

   - n_estimators: 100, 200, 300

- max_depth: 10, 20, 30
- min_samples_split: 2, 5, 10

This results in a total of 27 (3x3x3) combinations of parameters. If we were to do this on a single machine, we would have to train 27 different models sequentially, which could take a significant amount of time.

But if we use a cluster of N nodes, we can distribute these 27 tasks to N nodes. For example, if we have a cluster with 9 nodes (N = 9), we only need to train 3 models per node. This means that assuming perfect parallelism, the time required for testing will be reduced to approximately 1/N of the time required on a single machine.

In practice, it is faster than running parameter tuning on a single machine due to inter-node distribution and processing overhead. The benefits of using groups for change are manifold such as:

- It speeds up computing by distributing the workload across multiple machines. This is especially important when working with large data sets or complex models that require large amounts of data to report.
- It can make better use of resources. By distributing workloads across multiple machines, you can use all the resources of each machine and minimize downtime.
- If a machine fails during the migration process, the tasks assigned to the machine are reassigned to other machines in the cluster, ensuring that the migration process can continue without interruption.

In summary, leveraging group tuning for machine learning tuning can speed up the process by allowing comparison of training models. This is especially important when working with large data sets or complex models that require large amounts of data to report. It also provides fault tolerance to ensure that the transfer process can continue uninterrupted even if a machine fails.