What is Apache Spark?

CONTENT

Узнаем

- 01 История Spark
- 02 Что такое Spark?
- 03 Особенности Spark
- 04 Компоненты Apache Spark
- 05 Архитектура

History of Apache Spark

Spark был создан в 2009 году в лаборатории AMP университета Беркли и в след. году выпущен с открытым исходным кодом под лицензией BSD. Через 3 года, проект был передан в Apache. В 2014 в компании больших данных Databricks уже широко принят при анализе. Теперь 80% of the Fortune 500, use Apache Spark™. Теперь, 80% компаний из списка Fortune 500 используют Apache Spark.

What is Apache Spark?

Apache Spark — это открытый движок, а даже набор библиотек для обработки больших данных в кластерах компьютеров (использующий простые программные конструкции).

> Движок - 引擎

Для программирования Spark предоставляет API-интерфейсы разработки на языках R, Python, Java и Scala.

> 它提供使用 Java、Scala、Python 和 R 语言的开发 API。

В производственной среде Spark может помочь быстро запрашивать данные, анализировать их и преобразовать данные в больших масштабах.

> 在生产中，Spark可以帮助在大规模范围内开发者快速查询数据、分析数据并且转化数据。

Spark Features

Apache Spark обладает множеством плюсов, которые делают его одним из самых активных проектов в экосистеме Hadoop.

1. Скорость: благодаря использованию RDD (это струкутра данных во фрейворке Spark, который называется устойчивый распределенный набор данных), Spark экономит много времени при записи на диск и, следовательно, может работать в 100 раз быстрее, чем Hadoop.
2. Данные храняются в памяти, поэтому программа может получить данные за короткое время.
3. Поддержка нескольких языков:
4. **Отказоустойчивость**: Механизм кровного родства RDD делает работу узлов отказоустойчивой.
5. Может справиться с различными сценариями работы с большими данными: основанный на движок, Spark расширил функции SQL-запросов, алгоритмов машинного обучения и

граф обработки.

> Отказоустойчивость – 容错 RDD - Resilient Distributed Dataset, устойчивый распределенный набор данных

> Apache Spark 所具有的众多优点使其成为 Hadoop 生态系统中最活跃的项目之一。其中包括： 快速：由于RDD的使用，Spark节省了大量大写磁盘的时间，因此可以比Hadoop快100倍 内存上做计算：通过将数据缓存到内存中来优化数据处理。 支持多种语言： 容错性：RDD的血缘机制让节点的工作具有容错性 可以应对多种大数据应用场景：以 Spark 为基础，Spark 拓展出了 SQL 查询、机器学习算法、复杂分析的功能。

Известно что Spark не просто движок. На основе движока разрабатываются несколько компонентов. Давайте посмотрим.

Components of Apache Spark

Полный фреймворк Spark включает в себя:

> Spark 框架包括： Spark Core 是整个框架的基础，所有其他的 Spark 组件都是建立在 Spark Core 之上的。

- Spark Core — это основа всего фреймворка. Все остальные компоненты Spark построены на основе Spark Core.

> 用于交互式查询的 Spark SQL：Spark SQL 是用于结构化数据处理的组件，可以让你使用 SQL 语句以及 Apache Hive中的SQL方言（HiveQL）来进行数据查询。它也支持多种数据格式，如 Parquet、JSON 等，并且能够与其他数据源例如 HDFS、Hive 或关系型数据库进行交互。内部采用了一个称为 DataFrame 的编程抽象，是对 RDD 的进一步封装，提供了更丰富的优化。

- Spark SQL：Компонент для обработки структурированных данных, который позволяет запрашивать данные с операторов SQL и HiveSQL.

> 用于实时分析的 Spark Streaming：Spark Streaming 提供了处理实时数据流的功能。它可以从多种来源获取数据流，例如 Kafka、Flume等，并将它们转换为 RDDs 或 DataFrames，然后可以使用 Spark 的转换和动作操作进行处理。

- Spark Streaming: Предоставляет возможности для обработки потоков данных в реальном времени.

> 用于机器学习的 Spark Mllib：MLlib 是 Spark 的机器学习（ML）库。它包含了常见的机器学习算法和工具，比如分类、回归、聚类、协同过滤、降维等，以及底层的优化原语和高层的管道API。

- Spark MLlib: это библиотека, в которой реализовано большество алгоритмов машинного обучения.

> 用于图形处理的 Spark GraphX：GraphX 是 Spark 面向图形计算的库。它允许用户以顶点和边的集合来创建一个有向属性图，并提供了一系列的图算法比如 PageRank 和三角形计数。

- Spark GraphX：он позволяет выполнить граф обработки

Давайте узнаем как Spark работает при обработки данных в кластере, то есть его архитектуру.

Архитектура Spark представляет собой очень типичную структуру master-slave. В смысле, на кластер есть 3 роли - Master Node, Cluster Manager и Worker Node.

Master Node - это компьютер, в котором запускает программа Spark(то есть Driver Program). После запуска создается Spark Context, который отвечает за подать заявки на ресурсы, отправить задачи на worker и следить процесс выполнения task и представить результат.

A Cluster Manager отвечает за планирование задач. На основе статусов Worker Cluster Manager решает какие задачи надо отправить куда.

Worker Node - это узла, на которым фактически выполняются задачи. В одном узле может работает несколько процессов Executor, в котором обрабатывают задачи. Задачи представляют собой независимы друг от дурга, поэтому они может параллелино работают. И конечно, все обработки происходят в памяти Worker Node.

> https://cloud.tencent.com/developer/article/2347140?areaId=106001 Master Node & Worker Node：
>
> - Master Node 是一台主机，常驻Master进程，负责分配任务以及监控Worker存活。
> - Worker节点：本质上是多台机器，常驻Worker进程，负责执行任务以及监控任务运行状态。
>
> Spark Application：用户自己写的程序
>
> Spark Driver & Executor Spark Driver：一个进程。负责运行Spark任务中的main()方法，以及创建SparkContext。 Driver在Spark作业时主要负责： 将用户程序转化为任务（job） 在Executor之间调度任务 跟踪Executor的执行情况 通过UI展示查询运行情况
>
> Executor：
>
> - 负责运行组成Spark应用的任务(job)，并将结果返回给Driver进程；
> - 他们通过自身的块管理器（Block Manager）为用户程序中要求缓存的RDD提供内存式存储。RDD是直接缓存在>Executor进程内的，因此任务可以在运行时充分利用缓存数据加速运算。
>
> SparkContext: 用户通往 Spark 集群的唯一入口，可以用来在Spark集群中创建RDD 、累加器和广播变量。>SparkContext 也是整个 Spark 应用程序中至关重要的一个对象， 可以说是整个Application运行调度的核心（不包>括资源调度）。

Что качается Cluster Manager, его может быть по-разному. По умолчанию вы используйте Spark manager, который создает простой FIFO очередь для выполнения задач.

Если хотите даже можно заменить его на MESOS или Yarn, или Kubernetes(кибернете).

> В большинстве случаев, насколько я знаю, китайские компании максимально используют Yarn. это может быть относится к китайским компаниям, где максимально используют Yarn