Introduction to Apache Spark

An Overview of Features

08-24-2016
Introduction to Apache Spark

Agenda

- What is Apache Spark
- Major Vendors and Users
- Key Features
- Hadoop Vs Spark
- Spark Architecture
- Spark Streaming
- Spark Processing
- Examples and Use Cases

Disclaimer: Apache Hadoop and Spark is a registered trademark of the Apache Software Foundation (ASF). Hadoop Express and Net Serpents is not affiliated in any way to ASF. All educational material is created and owned by Net Serpents (dba Hadoop Express) and is intended only to provide training. Net Serpents does not own any of the products on which it provides training, many of which are owned by Apache while others are owned companies such as SAS, Python and Oracle. Net Serpents LLC is committed to education and online learning. All recognizable terms, names of software, tools, programming languages that appear on this site belong to the respective copyright and/or trademark owners.
What is Apache Spark

Apache Spark is an open source big data processing framework built around speed, ease of use, and sophisticated analytics.

- General data processing engine compatible with Hadoop data
- Used to query, analyze and transform data
- Developed in 2009 at AMPLab at University of California, Berkeley
- Became an Apache open source project in 2010
- Became top level project of Apache in 2014
- First discussed in the Mesos Whitepaper created in AMPLab
- Optimized to run in memory
  - 100 times faster than MapReduce when run in memory
  - 10 times faster than MapReduce when writing data to disk
What is Apache Spark

- A general-purpose data processing engine, suitable for use in a wide range of circumstances
- Interactive queries across large data sets, processing of streaming data from sensors or financial systems, and machine learning tasks
- Supports other data processing tasks with developer libraries and APIs
- Support of languages like Java, Python, R and Scala
- Often used alongside Hadoop’s HDFS
- Can also integrate equally well with other popular data storage subsystems such as HBase, Cassandra, MapR-DB, MongoDB and Amazon’s S3
What is Apache Spark

- Often used alongside Hadoop’s HDFS
- Can also integrate equally well with other popular data storage subsystems such as HBase, Cassandra, MapR-DB, MongoDB and Amazon’s S3.
Major Vendors and Users

Major Vendors

- Data Bricks – founded by founders of Spark at Berkeley
- Cloudera
- Hortonworks
- MapR

Major Users

- More than 1000 organizations are using Spark in production
- IBM, Huawei, Baidu, Aliba Taobao (eCommerce web site)
- Tencent (social networking site with 800 million users; 8000 compute nodes)
- Amazon, Ebay, Yahoo! And many others....
Key Features

Simplicity / Ease of Use

- Rich set of APIs
  - to interact with large datasets
  - Well documented
  - Structured
Key Features

Speed

In Memory / On Disk

- Spark is designed for speed, operating both in memory and on disk.
- In 2014, won the Daytona Gray Sort benchmarking challenge
- Processed 100 terabytes of data on solid-state drives in 23 minutes. The previous winner used Hadoop that took 72 minutes.
Key Features

Stream processing

Process “streams” of data from multiple sources simultaneously

Machine learning

- Well suited to training machine learning algorithms. Running broadly similar queries again and again, at scale, significantly reduces the time required to iterate through a set of possible solutions in order to find the most efficient algorithms.

Interactive analytics

- Explore data interactively by viewing query results and then either altering the initial query slightly or drilling deeper into results

Data integration

- Spark (and Hadoop) are increasingly being used to reduce the cost and time required for ETL process.
Key Features

Development Language Support

- SCALA
- Python
- Java
- SQL
- R
Hadoop Versus Spark

- Hadoop has **cluster management** features provided by YARN while Spark requires a cluster manager.

- Spark can run on top of Hadoop and utilize its cluster manager (YARN) or run separately utilizing other cluster managers such as Mesos.

- Spark is not designed for **data management** and cluster management. Hadoop handles these well.

- Hadoop provides advanced **data security** which is missing in Spark.

- Hadoop provides **Disaster Recovery** capabilities to Spark.

- Spark provides for fast **in-memory data processing** of large data volumes which Hadoop does not.

- Spark provides enterprise-class **streaming, graph processing** and **machine learning** capabilities which can be utilized by Hadoop.

Spark is not a replacement of Hadoop. Spark and Hadoop complement each other.
Architecture

Integrations

Spark can run in following modes:

- Standalone cluster mode
- On Hadoop YARN
- On Apache Mesos

Spark can access data in:

- HDFS
- Cassandra
- Hive
- Hbase
- Tachyon
- Any Hadoop data source
SPARK Technology Stack

- SPARK SQL
- SPARK Streaming (Streaming)
- MLlib (Machine Learning)
- GraphX (Graph Computation)
- Spark R (R on Spark)

SPARK Core Engine

- Standalone Scheduler
- YARN
- MESOS
Architecture

SPARK Technology Stack

SPARK Core Engine

- Basic functionality of Spark
- Uses RDDs (Resilient Distributed Datasets)
- Contains APIs for manipulating RDDs

Spark RDDs are a collection of items distributed across compute nodes. Spark core APIs allows manipulation of these RDDs in parallel.
Architecture

SPARK Technology Stack

SPARK SQL

- Used for working with structured data
- Allows querying with SQL and HQL (Hive QL)
- Data sources can be Hive tables, Parquet, JSON, others..
- Allows intermixing SQL with programmatic manipulation of RDDs in Python, Scala, Java

Note: Shark is an older version of SPARK SQL developed by UC, Berkeley
Architecture

SPARK Technology Stack

SPARK Streaming

- Used for processing live streams of data
- Eg., log files / message queues
- Can manipulate data stored on disk or in-memory as it arrives in real time

Streaming offers high throughput and is fault tolerant and scalable
MLib

- Provides machine learning (ML) algorithms
- Eg., clustering, regression analysis, classification, filtering, model evaluation, data import
- Includes lower level ML primitives like gradient descent

MLib is a library with methods that have the capability to scale out across a cluster.
GraphX

- Library for manipulating graphs
- Allows viewing data as graphs called **property graphs**
- Pregel API is an API to create custom iterative graph algorithms

Property graphs are immutable, fault tolerant and distributed (just like RDDs)
Architecture

Spark R

- Support for R in Spark is more recent (with release 1.4)
- Allows data scientists working in R to utilize Spark capabilities
Spark Streaming

- Allows ingestion of data from a wide range of data sources
- Data processed by Spark can be stored in external systems or presented in dashboards
Spark Streaming

Input stream of data is divided into discreet chunks

Each chunk represents data collected during a brief period and is processed individually
SPARK Processing

Source: https://spark.apache.org/docs/latest/cluster-overview.html
Driver program accesses Spark through a SparkContext object.

Source: https://spark.apache.org/docs/latest/cluster-overview.html
SPARK Processing

Spark Context represents a connection to a computing cluster. Once created, it can be used to build RDDs.

Source: https://spark.apache.org/docs/latest/cluster-overview.html
SPARK Processing

**Cluster Manager** is an external service

- A default built-in cluster manager called Standalone Cluster manager is pre-packaged with Spark
- Hadoop YARN and Apache Mesos are two popular cluster managers
- Driver requests cluster manager to provide resources for launching executors
- Cluster manager launches executors which are then used by driver to run tasks

Source: https://spark.apache.org/docs/latest/cluster-overview.html
SPARK Processing

Tasks are the smallest unit of physical execution

- The driver program implicitly creates a DAG (Direct Acyclic Graph) of operations
- This DAG is converted to a physical execution plan
- The execution plan is used by the driver to execute tasks using executors on the worker nodes

Source: https://spark.apache.org/docs/latest/cluster-overview.html
SPARK Processing

**Executors** are processes that execute tasks

- Executors run the tasks and return results to the driver
- Also provide in-memory storage for RDDs

Source: https://spark.apache.org/docs/latest/cluster-overview.html
SPARK Use Cases

Spark Streaming Use Cases

ETL (Extract Transform Load)

- With Spark streaming it is possible to run ETL on streaming data that is continually cleaned and aggregated before moving it to data stores
- This is different from tradition approach of ETL based on batch processing
- IoT data collected via sensors on devices can be continually collected, cleaned and stored in datastores for analytics

Online Data Enrichment

- With Spark Streaming it is possible to combine historical data of online customers with changes in their buying behavior and preferences to present targeted advertisements in real time
SPARK Use Cases

Spark Streaming Use Cases

Trigger Event Detection

• Spark streaming is being utilized to detect events and respond quickly to them by raising alerts. Eg., fraudulent transaction detection by banking systems and detecting changes in a patient’s vital signs such as heartbeat and blood pressure in a hospital.

Session Analysis on the web

• Spark Streaming can be used to analyze a user’s online activity on a web site and provide real-time recommendations. Eg., suggesting movies to a user on Netflix.
Machine Learning Use Cases

MLib is used for common big data functions like customer segmentation and sentiment analysis.

**Network Security**: Predictive Intelligence can be used to inspect and detect threats on data packets arriving over the network before passing them to the storage platform.
SPARK Use Cases

Business examples

• Uber uses Kafka, Spark Streaming and HDFS to analyze and terabytes of user data by collecting and converting it from unstructured event data into structured data

• Pinterest uses an ETL pipeline to gain insights into how users are engaging all over the world with Pins to help them select products to buy or plan trips to destinations.

• Conviva uses Spark to optimize video streams and manage live video traffic of over 4 million video feeds per month
Special thanks to references

Special thanks to the following authors and contributors for providing valuable material used in this presentation:

**Apache website:** spark.apache.org

**Learning Spark (Lightning fast data analytics)** by Holden Karau, Andy Konwinski and Matei Zaharia

**Getting started** on Apache Spark by James A Scott

**Top Apache Use Cases** : https://www.qubole.com/blog/big-data/apache-spark-use-cases/

**Introduction to Apache Spark** by Databricks.com (download slides: http://cdn.liber118.com/workshop/itas_workshop.pdf)
Thank You!

For queries / suggestions/ feedback please send an email to info@hadoopexpress.com or shashi@netserpents.com