Cloud Computing

Introduction

Camilo Valdes
cvalde03@fiu.edu
Agenda

• Overview
• Amazon Web Services (AWS)
  - AWS Console
  - AWS Educate
• Spark
  - MapReduce
  - Clusters
• Tensorflow
  - Introduction
  - Examples
Overview

• Big Data & Computing
• Sometimes, a single computer cannot process all the data, or it would take too long
• Rather than use a single powerful machine, we could use many commodity ones
• Process data in parallel, in small chunks, and aggregate the results
MapReduce

Single Machine

Master Node

Worker Nodes
Building Serverless Applications

Build and run your applications and services without thinking about servers.

Learn more »

- Lightsail: Everything you need to get started on AWS—for a low, predictable price
- AWS Fargate: Run containers without managing servers
- Batch Processing at Any Scale: Run hundreds of thousands of jobs on EC2, fully managed by AWS Batch
- 90,000+ Databases Migrated to AWS: Save time & cost—migrate to fully managed databases

https://aws.amazon.com
Teach Tomorrow's Cloud Workforce Today

With the increasing demand for cloud professionals, AWS Educate provides an academic gateway for the next generation of IT and cloud professionals. AWS Educate is Amazon's global initiative to provide students and educators with the resources needed to accelerate cloud-related learning.

Join AWS Educate

Login to AWS Educate

Join Hundreds of Thousands of Students, over 10,000 Educators, and More Than 2,400 Institutions Who Use AWS Educate
AWS Educate

• Students
  - Get $100 AWS credit for signing up
    ▶ FIU email
  - Resources for Learning
    ▶ Tutorials, videos, etc.
  - Specialization Paths
  - AWS Certifications
  - Job Board

https://aws.amazon.com/education/awseducate/
Explore Cloud Career Pathways

Explore AWS Educate's Cloud Career Pathways to start building the key cloud skills you'll need to be successful in leading technology careers. Earn a completion credential for each pathway and share with prospective employers to show what you've learned.

Check out the roles below to learn more about each pathway and get started!

- **Cloud Computing 101**
  Take a crash course on the cloud, its history, solutions, and why companies across the globe are looking for employees with AWS cloud expertise.

- **Application Developer**
  Curious how App Developers design, test, and improve engaging web and mobile applications in the cloud? Learn more about the skills you'll need.

- **Cloud Support Associate**
  If you're excited by the future of cloud computing and enjoy working directly with customers, learn more about becoming a Cloud Support Associate.

- **Cloud Support Engineer**
  Interested in multiple technologies and working with companies to support AWS cloud solutions? Learn more about becoming a Cloud Support Engineer.

- **Cybersecurity Specialist**
  Cybersecurity Specialists use expertise in networking, programming, and coding to protect customer data every day. Learn more about the skills they use.

- **Data Integration Specialist**
  Excited about bringing data sources together to tell the story of a product's performance? Discover ways to build and improve products through data.

- **Data Scientist**
  Curious how discovering patterns in large data sets can translate into new business strategies? Learn more about how Data Scientists do this every day.

- **DevOps Engineer**
  If you like working behind the scenes to tackle challenges and are curious about skills like scripting and coding, learn more about becoming a DevOps Engineer.
Apache Spark

MLlib
- Machine Learning

Streaming
- Real-time analytics

SQL
- Interactive Queries

GraphX
- Graph processing

Core

R
Python
Scala
Java
Apache Spark

• Spark is a Big Data Processing Engine — a Fast, General-Purpose, Cluster-computing Platform.

• Handles the Scheduling, Distribution, and Monitoring of applications spanning many worker machines.

• Has a Rich API to distribute data across the cluster, and process it in parallel.

• Supports a variety of workloads such as Machine Learning (MLlib), Streaming, interactive queries, graph programming and SQL.

• Execution Frameworks have language support for Python, R, Java, and Scala.
Spark — Unified Stack

• The Spark project contains multiple high-level specialized components (MLlib, Streaming, etc.).

• Spark’s main programming abstraction are Resilient Distributed Datasets (RDDs), a data structure distributed across nodes that can be worked on in parallel.

• Spark’s multiple components operate on RDDs, which allows for close interoperability and tight integration.

• Applications that use multiple processing models can be written without high maintenance and development costs.
Spark — Main Benefits

Solve problems faster, and on a much larger scale

- **Ease of Use** — Rich, high level APIs
- **Speed** — Fast parallel execution
- **General Engine** — Combine processing models
- **Open Source** — Freely Available

- Makes developing General Purpose Distributed programs easier, less painful.
- Reduces the **management burden** of maintaining separate tools.
- Allows the **close Interoperability** of high-level components
Spark Core

- Spark Core contains the basic functionality of Spark, including components for task scheduling, memory management, fault recovery, interacting with storage systems, and more.

- Spark Core is also home to the API that defines resilient distributed datasets (RDDs), which are Spark’s main programming abstraction.

- RDDs represent a collection of items distributed across many compute nodes that can be manipulated in parallel.
Spark — Data Processing

• Spark provides a simple way to parallelize applications across clusters, and hides the complexity of distributed systems programming, network communication, and fault tolerance.

• The system gives control to monitor, inspect, and tune applications while allowing implementation of common tasks quickly.

• The modular nature of the API (based on passing distributed collections of objects) makes it easy to factor work into reusable libraries and test it locally.
Storage Layers for Spark

• Spark can create resilient distributed datasets, RDDs, from any file stored in the Hadoop distributed filesystem (HDFS).

• Spark also support other storage systems supported by the Hadoop APIs (including your local filesystem, Amazon S3, Cassandra, Hive, HBase, etc.).

• It’s important to remember that Spark does not require Hadoop.

• It simply has support for storage systems implementing the Hadoop APIs.
Spark REPL

• Spark can be used from Python, R, Java, or Scala.

• **Spark itself is written in Scala**, and runs on the Java Virtual Machine (JVM).

• To run Spark on either your laptop or a cluster, all you need is an installation of Java 6 or newer.

• If you wish to use the Python API you will also need a Python interpreter (version 2.6 or newer).

• You don’t need to have Hadoop.

• Spark comes with interactive shells that enable ad hoc data analysis.

• Spark’s shells will feel familiar if you have used other shells such as those in R, Python, and Scala,
Installing Spark

• Spark is a framework
• Language bindings for
  - Python, Scala, Java
• Install with a package manager
  - Homebrew in macOS
  - Pycharm Repository for Windows
pyspark

• Python version of the Spark Shell.
• In Spark, we express our computation through operations on distributed collections that are automatically parallelized across the cluster.

• These collections are called resilient distributed datasets, or RDDs.

• RDDs are Spark’s fundamental abstraction for distributed data and computation.
RDDs

• An RDD is simply a distributed collection of elements.

• In Spark all work is expressed as either creating new RDDs, transforming existing RDDs, or calling operations on RDDs to compute a result.

• Spark automatically distributes the data contained in RDDs across your cluster and parallelizes the operations you perform on them.

• An RDD in Spark is simply an immutable distributed collection of objects.

• Each RDD is split into multiple partitions, which may be computed on different nodes of the cluster.

• RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.

• Once created, RDDs offer two types of operations: transformations and actions.
RDDs

- **Transformations** construct a new RDD from a previous one.

- **Actions** compute a result based on an RDD, and either return it to the driver program or save it to an external storage system.

- Although you can define new RDDs any time, Spark computes them only in a lazy fashion — that is, the first time they are used in an action.

- Spark provides two ways to create RDDs
  - loading an external dataset.
  - Parallelizing a collection in your driver program.
Spark Cluster

• Every Spark application consists of a **driver** program that launches various parallel operations on a cluster.

• The driver program contains your application’s **main function** and defines distributed datasets on the cluster, then applies operations to them.

• The driver communicates with a potentially large number of **distributed workers** called executors.

• A driver and its executors are together termed a **Spark application**.
Building Serverless Applications

Build and run your applications and services without thinking about servers

Learn more ➤

Lightsail
Everything you need to get started on AWS—for a low, predictable price

AWS Fargate
Run containers without managing servers

Batch Processing at Any Scale
Run hundreds of thousands of jobs on EC2, fully managed by AWS Batch

90,000+ Databases Migrated to AWS
Save time & cost—migrate to fully managed databases

https://aws.amazon.com
Demo
TensorFlow

• Open source platform for Machine Learning
• Developed by Google
• Very popular in Deep Learning
Deep Representation

image credit: François Chollet
Democratization of Deep Learning

• Back in the early days, to do any deep learning, you needed a lot of experience with C++ and CUDA (NVIDIA’s driver API)

• Developing deep learning models was cumbersome, and there were no tools for easy debugging

• Tensorflow was created to develop ML applications at scale and in production
  - Simplicity
  - Scalability
  - Versatility and Reusability

Tensorflow’s first released to Open Source was on November 8, 2015 (version 0.5.0)
TensorFlow

• Name comes from its basic data structure, a “tensor”

• A tensor in TensorFlow is a multidimensional data structure
  - Scalar - 0D tensor
  - Vector - 1D tensor
  - Matrix - 2D tensor
  - 3D tensor, etc.

• Tensors have attributes such as shape, data type, and the number of axes (rank)
Anatomy of a Neural Network

• Training a Neural Network involves
  - Layers (combined into a network)
  - Input Data and its Targets (labels)
  - Loss Function (feedback signal)
  - Optimizer

image credit: François Chollet
Demo

MNIST Dataset
60,000 Training Images
10,000 Test Images

http://yann.lecun.com/exdb/mnist/