FUNDAMENTALS OF BIG DATA

Data analysis & ML

Marcelo Horacio Fortino
MBA PM | PSM I | ITIL & ISO 20000
www.fortinux.com | @HoracioGRC

Copyright

Based on the following works:

SKIENA, Steven S. The Data Science Design Manual. (2017). Switzerland: Springer.

KUBAT, Miroslav. An Introduction to Machine Learning. Second Edition (2017). Switzerland: Springer.

TURKINGTON, Garry. Hadoop Beginner's Guide. (2013). UK: Packt Publishing.

O'REILLY RADAR TEAM. Planning for Big Data. (2012). USA: O'Reilly Media.

Objectives

- **Know Big Data solutions and technologies such as Apache Hadoop.**
- **Acquire knowledge to design business intelligence strategies integrating large data sets and data warehouses.**
- **Develop Machine Learning in-house using Spark MLlib and TensorFlow.**

Contents

- Introduction to Big Data and Analytics.
- Big Data market and trends.
- Big Data definition and history.
- Types of Big Data: structured, unstructured, semi-structured.
- Big Data use cases.
- Best practices for Big Data analytics.
- Hadoop: HDFS & MapReduce, YARN.

Contents

- Big Data processes: ingest, store, process/query, visualize.
- Tools and technologies: Hadoop, Sqoop, Kafka, Mesos, Redis, CouchDB.
- Document stores: MongoDB.
- Column stores: HBase + Cassandra.
- Big Data analytics: Spark, Storm.
- Elastic Stack: Logstash, ElasticSearch and Kibana.
- Machine learning techniques:
- Spark (MLlib, Streaming).
- TensorFlow.

Frameworks and tools for Big Data

- Big Data Processes

Big Data Processes

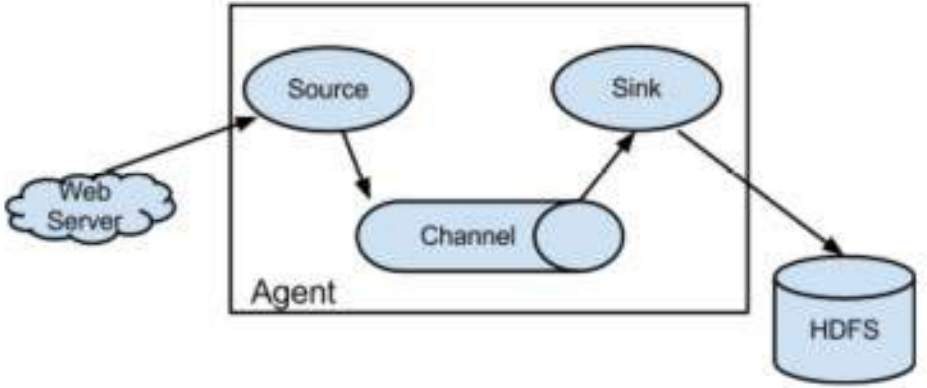
- **Data Ingestion.**
- **Data Storage.**
- **Data Processing / Data Query.**
- **Data Visualization.**

Big Data Ingestion

- **It's the first step for the data coming from variable sources to a medium storage where it can be accessed, used, and analyzed by the organization.**
- **The data here is prioritized and categorized.**
- **The destination is typically a data warehouse, data mart, database, or a document store.**

Apache Flume

- **A distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.**



Extracted from: https://flume.apache.org/

Apache Nifi

- **It’s one of the best data ingestion tools that provide an easy to use, powerful, and reliable system to process and distribute data.**

Extracted from: https://nifi.apache.org/

Font image: https://nifi.apache.org/

Elastic Logstash

- **Open-source, server-side data processing pipeline that ingests data from a multitude of sources, simultaneously transforms it, and then sends it to your “stash, ” i.e., Elasticsearch.**

elastic

Extracted from: https://www.elastic.co/

Apache Flink

- **Framework in data ingestion pipeline for distributed stream processing that provides results that are accurate, even in the case of out-of-order or late-arriving data or Distributed Data Processing.**

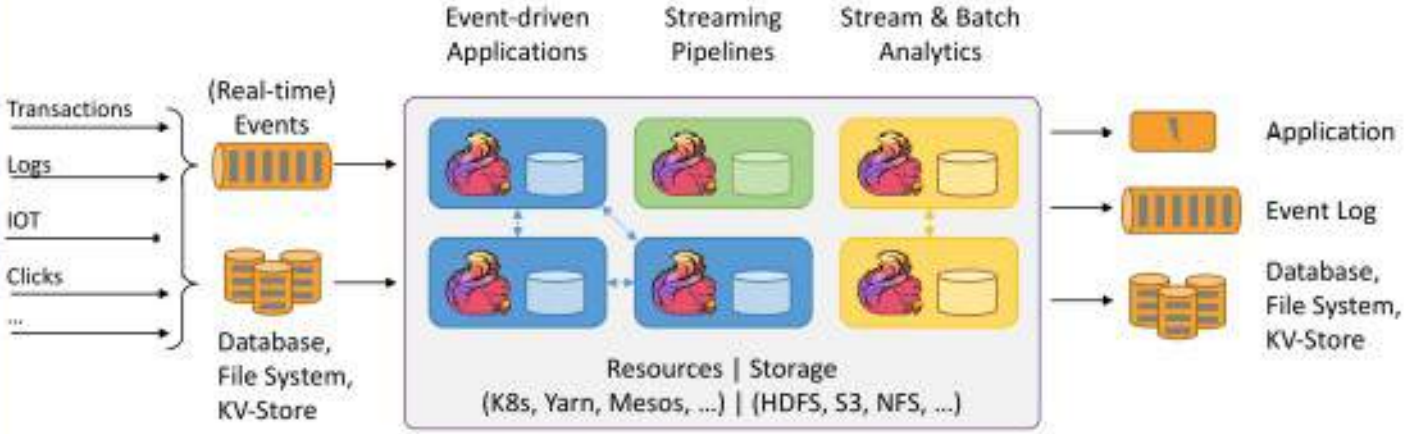


Image and text extracted from: https://flink.apache.org/

Big Data Storage

- **Storage becomes a challenge when the size of the data you are dealing with becomes large.**
- **Several possible solutions like Data Ingestion Patterns can rescue from such problems.**
- **Finding the most efficient storage solution is the scope of this step.**

Big Data Storage Tools

- **HDFS : Hadoop Distributed File System.**
- **Ozone: An object store for Hadoop, the HDFS next generation.**
- **GlusterFS: Dependable Distributed File System.**
- **Ceph**
- **Cloud storage: Amazon S3 Storage Service, IBM Cloud Object Storage, Azure Blob Storage, Google Cloud Storage.**

HDFS

- **HDFS is a Java-based file system that provides scalable and reliable data storage, and it was designed to span large clusters of commodity servers.**
- **HDFS holds a huge amount of data and provides easier access.**

Extracted from:
https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

HDFS

- **To store such massive data, the files are stored on multiple machines.**
- **These files are stored redundantly to rescue the system from possible data losses in case of failure.**

Extracted from:
https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

HDFS

- **HDFS also makes applications available for parallel processing in Data ingestion.**
- **HDFS is built to support applications with large data sets, including individual files that reach into the terabytes.**

Extracted from:
https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

Apache Ozone

- **Ozone is a scalable, redundant, and distributed object store for Hadoop.**
- **Apart from scaling to billions of objects of varying sizes, Ozone can function effectively in containerized environments such as Kubernetes and YARN.**

Extracted from: https://ozone.apache.org/

Apache Ozone

- **Applications using frameworks like Apache Spark, YARN and Hive work natively without any modifications.**
- **Ozone is built on a highly available, replicated block storage layer called Hadoop Distributed Data Store (HDDS).**

Extracted from: https://ozone.apache.org/

GlusterFS

- **Gluster is a free and open source software scalable network filesystem.**

Extracted from: https://docs.gluster.org/en/latest/ | https://www.gluster.org/

GlusterFS

- **Scale-out storage systems based on GlusterFS are suitable for unstructured data such as documents, images, audio and video files, and log files.**
- **Using this, we can create large, distributed storage solutions for media streaming, data analysis, data ingestion, and other data- and bandwidth-intensive tasks.**

Extracted from: https://docs.gluster.org/en/latest/ | https://www.gluster.org/

Ceph

- **Ceph's foundation is the Reliable Autonomic Distributed Object Store (RADOS), which provides your applications with object, block, and file system storage in a single unified storage cluster—making Ceph flexible, highly reliable and easy for you to manage.**

Extracted from: https://ceph.io/

Ceph

- **Ceph's CRUSH algorithm liberates storage clusters from the scalability and performance limitations imposed by centralized data table mapping.**

Extracted from: https://ceph.io/

Ceph

- **It replicates and rebalances data within the cluster dynamically—eliminating this tedious task for administrators, while delivering high-performance and infinite scalability.**

Extracted from: https://ceph.io/

Amazon S3 Storage Service

- **Amazon Simple Storage Service (Amazon S3) is object storage with a simple web service interface to store and retrieve any amount of data from anywhere on the internet.**
- **It is designed to deliver 99.999% durability, and scale past trillions of objects worldwide.**

Extracted from: https://aws.amazon.com/free/storage/s3/

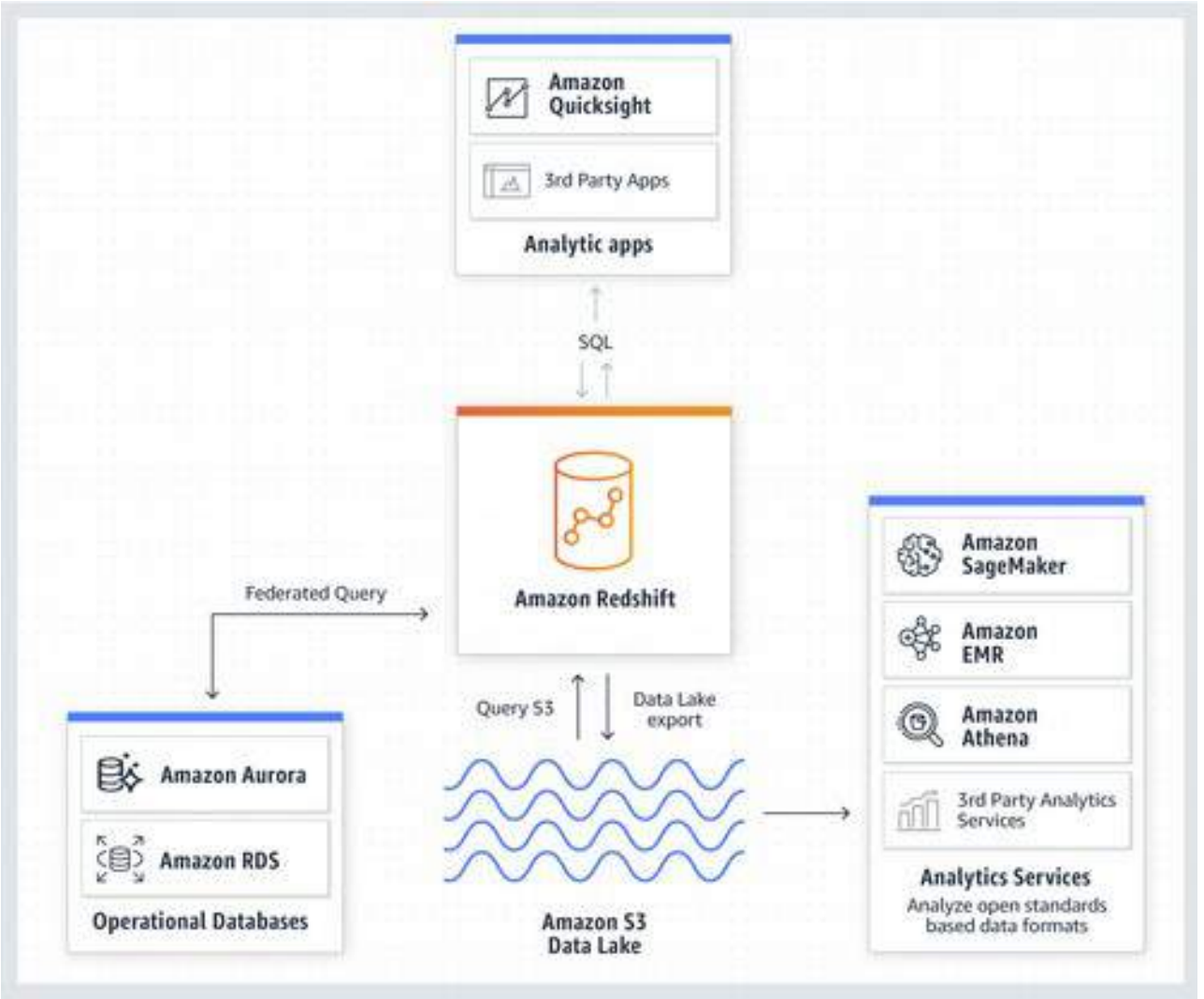
Amazon Redshift

- **Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud.**
- **Redshift lets you easily save the results of your queries back to your S3 data lake using open formats like Apache Parquet to further analyze from other analytics services like Amazon EMR, Amazon Athena, and Amazon SageMaker.**

Extracted from: Amazon Redshift https://aws.amazon.com/redshift/

- **Amazon Redshift.**

Image: Amazon Redshift
https://aws.amazon.com/redshift/



Big Data Processing

- **The data we have collected in the previous layer is to be processed in this layer.**
- **Here the data pipeline processing system route the data to a different destination, classify the data flow and it's the first point where the analytic may take place.**
- **This is the layer where queries and active analytic processing takes place.**

Batch Processing System

- **For offline analytics, it's a simple batch processing system.**
- **Apache Sqoop is the tool for doing this.**
- **It efficiently transfers bulk data between Apache Hadoop and structured datastores such as relational databases.**

Extracted from: https://sqoop.apache.org/

Apache Sqoop

- **Apache Sqoop can also be used to extract data from Hadoop and export it into external structured data stores.**
- **Apache Sqoop works with relational databases such as Teradata, Netezza, Oracle, MySQL, Postgres, and HSQLDB.**

Extracted from: https://sqoop.apache.org/

Apache Sqoop

- **Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.**

Extracted from: https://sqoop.apache.org/

Apache Sqoop

- **You can use Sqoop to import data from a relational database management system (RDBMS) such as MySQL or Oracle or a mainframe into the Hadoop Distributed File System (HDFS), transform the data in Hadoop MapReduce, and then export the data back into an RDBMS.**

Extracted from: https://sqoop.apache.org/

Apache Sqoop

- **Sqoop automates most of this process, relying on the database to describe the schema for the data to be imported.**
- **Sqoop uses MapReduce to import and export the data, which provides parallel operation as well as fault tolerance.**

Extracted from: https://sqoop.apache.org/

Apache Storm

- **It is a system for processing streaming data in real-time during Data ingestion.**
- **It is scalable, fault-tolerant, and is easy to set up and operate.**

Extracted from: https://storm.apache.org/

Apache Storm

- **Apache Storm is a free and open source distributed realtime computation system.**
- **Apache Storm makes it easy to reliably process unbounded streams of data, doing for realtime processing what Hadoop did for batch processing.**
- **Apache Storm is simple and can be used with any programming language.**

Extracted from: Apache Storm Docs & website: https://storm.apache.org/

Apache Storm

- **There's no hack that will turn Hadoop into a realtime system; realtime data processing has a fundamentally different set of requirements than batch processing.**

Extracted from: Apache Storm Docs & website: https://storm.apache.org/

Apache Storm

- **It adds reliable real-time data processing capabilities to Enterprise Hadoop.**
- **Storm on YARN is powerful for scenarios requiring real-time analytics, machine learning and continuous monitoring of operations.**

Extracted from: https://storm.apache.org/

Apache Storm Use Cases

- **Realtime analytics.**
- **Online machine learning.**
- **Continuous computation.**
- **Distributed RPC, ETL, and more.**

Extracted from: Apache Storm Docs & website: https://storm.apache.org/

Apache Spark

- **Open source and parallel processing framework for running large-scale data analytics applications across clustered systems.**

Extracted from Spark website: https://spark.apache.org/

Apache Spark

- **It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Structured Streaming for incremental computation and stream processing.**

Extracted from Spark website: https://spark.apache.org/

Apache Spark

- **Spark stack**

Apache Spark

- **Used in conjunction with heavy compute jobs and Apache Kafka technologies.**
- **Developed at the University of California, Berkeley.**

Apache Spark

- **With Spark running on Apache Hadoop YARN, developers can create applications to exploit Spark's power, derive insights, and enrich their data science workloads within a single, shared data set in Hadoop.**

Storm Vs. Spark

Situation	Spark	Storm
Stream processing	Batch processing	Micro-batch processing
Latency	Latency of a few seconds	Latency of milliseconds
Multi-language support	Lesser language support	Multiple language support
Languages	Java – Scala	Java – Scala – Clojure
Stream sources	HDFS	Spout
Resource management	Yarn, Mesos	Yarn, Mesos
Provisioning	Basic using Ganglia	Apache Ambari
Messaging	Netty, Akka	ZeroMQ, Netty

Apache Impala

- **Impala raises the bar for SQL query performance on Apache Hadoop while retaining a familiar user experience.**
- **With Impala, you can query data, whether stored in HDFS or Apache HBase – including SELECT, JOIN, and aggregate functions – in real time.**

Extracted from: https://impala.apache.org/

Apache Impala

- **Impala uses the same metadata, SQL syntax (Hive SQL), ODBC driver, and user interface (Hue Beeswax) as Apache Hive, providing a familiar and unified platform for batch-oriented or real-time queries.**
- **Hive users can utilize Impala with little setup overhead.**

Extracted from: https://impala.apache.org/

Presto

- **SQL engine developed by Facebook for ad-hoc analytics and quick reporting.**
- **Open-source distributed SQL query engine for running interactive analytic queries against data sources of all sizes ranging from gigabytes to petabytes.**

https://prestodb.io/

Big Data Visualization

- **The visualization or presentation tier is where the data pipeline users may feel the VALUE of DATA.**
- **Visualization of findings helps us to make better business decisions.**

Elasticsearch

- **Elasticsearch is a distributed, open source search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured.**

elastic

Extracted from Elastic.co website: https://www.elastic.co/

Elasticsearch

- **Elasticsearch is built on Apache Lucene and was first released in 2010 by Elasticsearch N.V. (now known as Elastic).**

elastic

Extracted from Elastic.co website: https://www.elastic.co/

Apache Lucene

- **Lucene Core is a Java library providing powerful indexing and search features, as well as spellchecking, hit highlighting and advanced analysis/tokenization capabilities.**
- **The PyLucene sub project provides Python bindings for Lucene Core.**

Extracted from Apache Lucene website: https://lucene.apache.org/

Apache Solr

- **Solr is a high performance search server built using Lucene Core.**
- **Solr is highly scalable, providing fully fault tolerant distributed indexing, search and analytics.**
- **It exposes Lucene's features through easy to use JSON/HTTP interfaces or native clients for Java and other languages.**

Extracted from Apache Lucene website: https://lucene.apache.org/

Elasticsearch

- **Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage, analysis, and visualization.**
- **Commonly referred to as the ELK Stack: Elasticsearch, Logstash, and Kibana.**
- **Nowadays changes its name to Elastic Stack.**

Extracted from: https://www.elastic.co/what-is/elasticsearch

Elastick Stack

- **Beats 7.10**
- **APM Server 7.10**
- **Elasticsearch 7.10**
- **Elasticsearch Hadoop 7.10**
- **Kibana 7.10**
- **Logstash 7.10**

Font: https://www.elastic.co/what-is/elasticsearch

Elasticsearch Use Cases

- **Application search.**
- **Website search.**
- **Enterprise search.**
- **Logging and log analytics.**
- **Infrastructure metrics and container monitoring.**
- **Application performance monitoring.**
- **Geospatial data analysis and visualization.**
- **Security analytics.**
- **Business analytics.**

Font: https://www.elastic.co/what-is/elasticsearch

Kibana

- **A Kibana dashboard displays a collection of saved visualizations.**
- **You can arrange and resize the visualizations as needed and save dashboards, so they are reloaded and shared.**

Kibana

- **Kibana acts as an analytics and visualization platform that builds on Elasticsearch to give you a better understanding of your Data ingestion framework.**

Monitoring Data

- **Continuous monitoring of data is an important part of the governance mechanisms.**
- **Apache Flume is useful for processing log data.**
- **Apache Storm is desirable for operations monitoring and Apache Spark for streaming data, graph processing, and machine learning.**
- **Monitoring can happen in the data storage step.**

Frameworks and tools for Big Data

- Big Data tools

Apache HBase

- **It's an open-source, distributed, versioned, non-relational database modeled after Google's Bigtable: A Distributed Storage System for Structured Data by Chang et al.**

Extracted from: https://hbase.apache.org/

Apache HBase

- **Just as Bigtable leverages the distributed data storage provided by the Google File System, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.**

Extracted from: https://hbase.apache.org/

Apache Mesos

- **A distributed systems kernel.**
- **Mesos is built using the same principles as the Linux kernel, only at a different level of abstraction.**
- **The Mesos kernel runs on every machine and provides applications (e.g., Hadoop, Spark, Kafka, Elasticsearch) with API's for resource management and scheduling across entire datacenter and cloud environments.**

Extracted from: https://mesos.apache.org/

NoSQL databases

- **NoSQL databases, (not-only SQL) or non relational, are mostly used for the collection and analysis of big data.**

NoSQL databases

- **NoSQL database allows for dynamic organization of unstructured data.**
- **Relational databases on the other hand, has structured and tabular design.**

Big Data: Redis

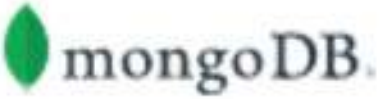
- **Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker.**

redislabs
HOME OF REDIS

https://redislabs.com/

MongoDB

- **MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era.**
- **MongoDB is a document database, which means it stores data in JSON-like documents.**



https://www.mongodb.com/

Cassandra

- **Linear scalability and proven fault-tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data.**
- **Cassandra's support for replicating across multiple datacenters is best-in-class, providing lower latency for your users.**

https://cassandra.apache.org/

Apache CouchDB

- **Apache CouchDB is an open-source document-oriented NoSQL database, implemented in Erlang.**
- **CouchDB uses multiple formats and protocols to store, transfer, and process its data, it uses JSON to store data, JavaScript as its query language using MapReduce, and HTTP for an API.**

https://couchdb.apache.org/

Couchbase

- **Couchbase is an award-winning distributed NoSQL cloud database.**
- **It delivers unmatched versatility, performance, scalability, and financial value across cloud, on-premises, hybrid, distributed cloud, and edge computing deployments.**

Couchbase

NoEQUAL

https://www.couchbase.com/

Apache Kafka

- **Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.**

Extracted from: Apache Kafka Docs & website: https://kafka.apache.org/

Apache Kafka

- **Scalable messaging system that lets users publish and consume large numbers of messages in real time by subscription.**

https://kafka.apache.org/

Event Streaming

- **Event streaming is the digital equivalent of the human body's central nervous system.**
- **It is the technological foundation for the 'always-on' world where businesses are increasingly software-defined and automated, and where the user of software is more software.**

Extracted from: Apache Kafka Docs & website: https://kafka.apache.org/

Event Streaming

- **Technically speaking, event streaming is the practice of capturing data in real-time from event sources like databases, sensors, mobile devices, cloud services, and software applications in the form of streams of events; storing these event streams durably for later retrieval; manipulating, processing, and reacting to the event streams in real-time as well as retrospectively; and routing the event streams to different destination technologies as needed.**

Extracted from: Apache Kafka Docs & website: https://kafka.apache.org/

Event Streaming

- **Event streaming thus ensures a continuous flow and interpretation of data so that the right information is at the right place, at the right time.**

Extracted from: Apache Kafka Docs & website: https://kafka.apache.org/

Event Streaming

- **Kafka combines three key capabilities so you can implement your use cases for event streaming end-to-end with a single battle-tested solution.**

Extracted from: Apache Kafka Docs & website: https://kafka.apache.org/

Apache Hive

- **Open source data warehouse system for analyzing data sets in Hadoop files.**

https://hive.apache.org/

Apache Hive

- **The Apache Hive ™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL.**
- **Structure can be projected onto data already in storage.**
- **A command line tool and JDBC driver are provided to connect users to Hive.**

Extracted from: https://hive.apache.org/

Databricks

- **Founded in 2013 by the original creators of Apache Spark™, Delta Lake and MLflow, Databricks brings together data engineering, science and analytics on an open, unified platform so data teams can collaborate and innovate faster.**

https://databricks.com/

Apache Zookeeper

- **ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.**

Extracted from Apache Zookeeper website: https://zookeeper.apache.org/

Apache Ambari

- **The Apache Ambari project is aimed at making Hadoop management simpler by developing software for provisioning, managing, and monitoring Apache Hadoop clusters.**

Extracted from Apache Ambari website: https://ambari.apache.org/

Apache Ambari

- **Ambari provides an intuitive, easy-to-use Hadoop management web UI backed by its RESTful APIs.**

Apache Ambari

Extracted from Apache Ambari website: https://ambari.apache.org/

Apache Ranger

- **Apache Ranger™ is a framework to enable, monitor and manage comprehensive data security across the Hadoop platform.**

Extracted from Apache Ranger website: https://ranger.apache.org/

Apache Sentry

- **Apache Sentry™ is a system for enforcing fine grained role based authorization to data and metadata stored on a Hadoop cluster.**

Extracted from Apache Sentry website: https://sentry.apache.org/

Big Data Tools

- ML: TensorFlow & Spark MLlib

Machine Learning

- **Machine learning is the practice of helping software perform a task without explicit programming or rules.**
- **With traditional computer programming, a programmer specifies rules that the computer should use.**

Extracted from TensorFlow: https://www.tensorflow.org/about

Machine Learning

- **ML requires a different mindset, though. Real-world ML focuses far more on data analysis than coding.**
- **Programmers provide a set of examples and the computer learns patterns from the data.**
- **You can think of machine learning as “programming with data”.**

Extracted from TensorFlow: https://www.tensorflow.org/about

Steps ML

- **Step 1: Gather Data.**
- **Step 2: Explore Your Data.**
- **Step 2.5: Choose a Model.**
- **Step 3: Prepare Your Data.**
- **Step 4: Build, Train, and Evaluate Your Model.**
- **Step 5: Tune Hyperparameters.**
- **Step 6: Deploy Your Model.**

Extracted from Maching Learning:
https://developers.google.com/machine-learning/guides/text-classification/

ML: Neural Network

- **is a type of model that can be trained to recognize patterns.**
- **It is composed of layers, including input and output layers, and at least one hidden layer.**
- **Neurons in each layer learn increasingly abstract representations of the data.**

Extracted from TensorFlow: https://www.tensorflow.org/about

Training a Neural Network

- **Neural networks are trained by gradient descent.**
- **The weights in each layer begin with random values, and these are iteratively improved over time to make the network more accurate.**

Extracted from TensorFlow: https://www.tensorflow.org/about

Training a Neural Network

- **A loss function is used to quantify how inaccurate the network is, and a procedure called backpropagation is used to determine whether each weight should be increased, or decreased, to reduce the loss.**

Extracted from TensorFlow: https://www.tensorflow.org/about

Apache MLlib

- **MLlib is Apache Spark's scalable machine learning (ML) library.**
- **Its goal is to make practical machine learning scalable and easy.**

Font: https://spark.apache.org/mllib/

Apache Mllib Tools

- **ML Algorithms: common learning algorithms such as classification, regression, clustering, and collaborative filtering.**
- **Featurization: feature extraction, transformation, dimensionality reduction, and selection.**

Font: https://spark.apache.org/docs/latest/ml-guide.html

Questions?

Bibliography

SKIENA, Steven S. The Data Science Design Manual. (2017). Switzerland: Springer.

KUBAT, Miroslav. An Introduction to Machine Learning. Second Edition (2017). Switzerland: Springer.

TURKINGTON, Garry. Hadoop Beginner's Guide. (2013). UK: Packt Publishing.

O'REILLY RADAR TEAM. Planning for Big Data. (2012). USA: O'Reilly Media.

Copyright

Based on the following works:

SKIENA, Steven S. The Data Science Design Manual. (2017). Switzerland: Springer.

KUBAT, Miroslav. An Introduction to Machine Learning. Second Edition (2017). Switzerland: Springer.

TURKINGTON, Garry. Hadoop Beginner's Guide. (2013). UK: Packt Publishing.

O'REILLY RADAR TEAM. Planning for Big Data. (2012). USA: O'Reilly Media.

Copyright

FUNDAMENTALS OF BIG DATA

THANK YOU

FUNDAMENTALS OF BIG DATA

Data analysis & ML

Marcelo Horacio Fortino
MBA PM | PSM I | ITIL & ISO 20000
www.fortinux.com | @HoracioGRC