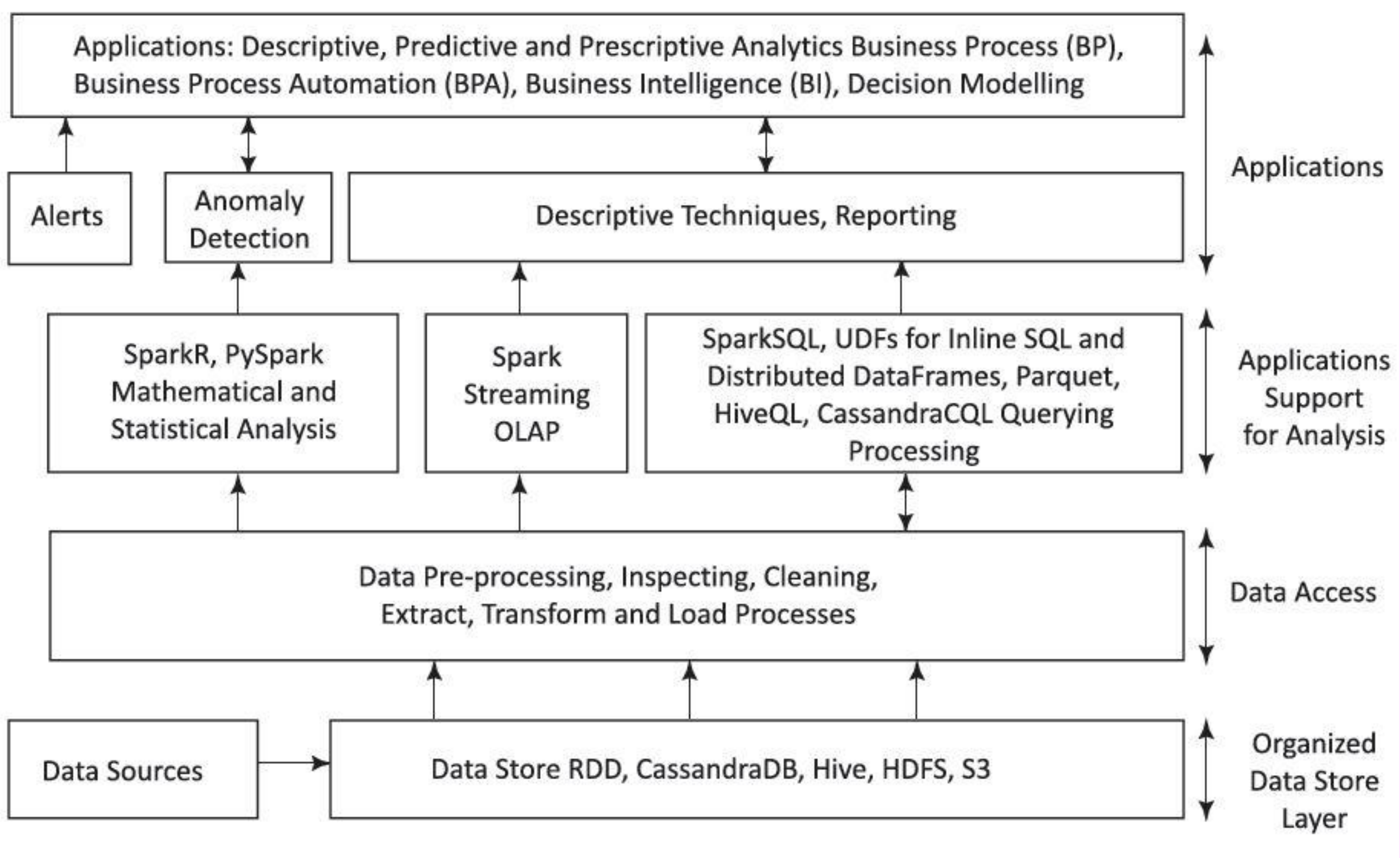
Lesson 3

Data Analytics using Apache® Spark™ Components Spark SQL and DataFrames

Figure 5.4 Steps between acquisition of data from different sources and its applications


Applications: Descriptive, Predictive and Prescriptive Analytics Business Process (BP), Business Process Automation (BPA), Business Intelligence (BI), Decision Modelling

Alerts

Anomaly Detection

Descriptive Techniques, Reporting

SparkR, PySpark Mathematical and Statistical Analysis

Spark Streaming OLAP

SparkSQL, UDFs for Inline SQL and Distributed DataFrames, Parquet, HiveQL, CassandraCQL Querying Processing

Data Pre-processing, Inspecting, Cleaning, Extract, Transform and Load Processes

Data Sources

Data Store RDD, CassandraDB, Hive, HDFS, S3

Steps For Data Analysis

Refer Figure 5.4: Layer 1 Data Storage: Store of data from the multiple sources after acquisition. The Big Data storage may be in HDFS compatible files, Cassandra, Hive, HDFS or S3.

Steps For Data Analysis

Refer Figure 5.4: Layer 2 Data Storage: Store of data from the multiple sources after acquisition. The Big Data storage may be in HDFS compatible files, Cassandra, Hive, HDFS or S3.

Steps For Data Analysis

Refer Figure 5.4: Layer 1 Data Storage: Store of data from the multiple sources after acquisition. The Big Data storage may be in HDFS compatible files, Cassandra, Hive, HDFS or S3.

Steps For Data Analysis

Refer Figure 5.4: Layer 2a Preprocessing:
(a) dropping out of range, inconsistent and outlier values,
(b) filtering unreliable, irrelevant and redundant information,
(c) data cleaning, editing, reduction and/or wrangling,
(d) data-validation, transformation or transcoding.

Steps For Data Analysis

Refer Figure 5.4: Layer 2b ETL
Layer 3: Mathematical and statistical analysis of the data obtained after querying relevant data needing the analysis, Spark Streaming, OLAP, Spark SQL, UDFs for inline SQL, Distributed DataFrames, HiveQL, Parquet, Cassandra QL query processing

Steps For Data Analysis

Refer Figure 5.4: Layer 4 Alerts to Applications, Anomaly detection, Descriptive and Reporting

Steps For Data Analysis

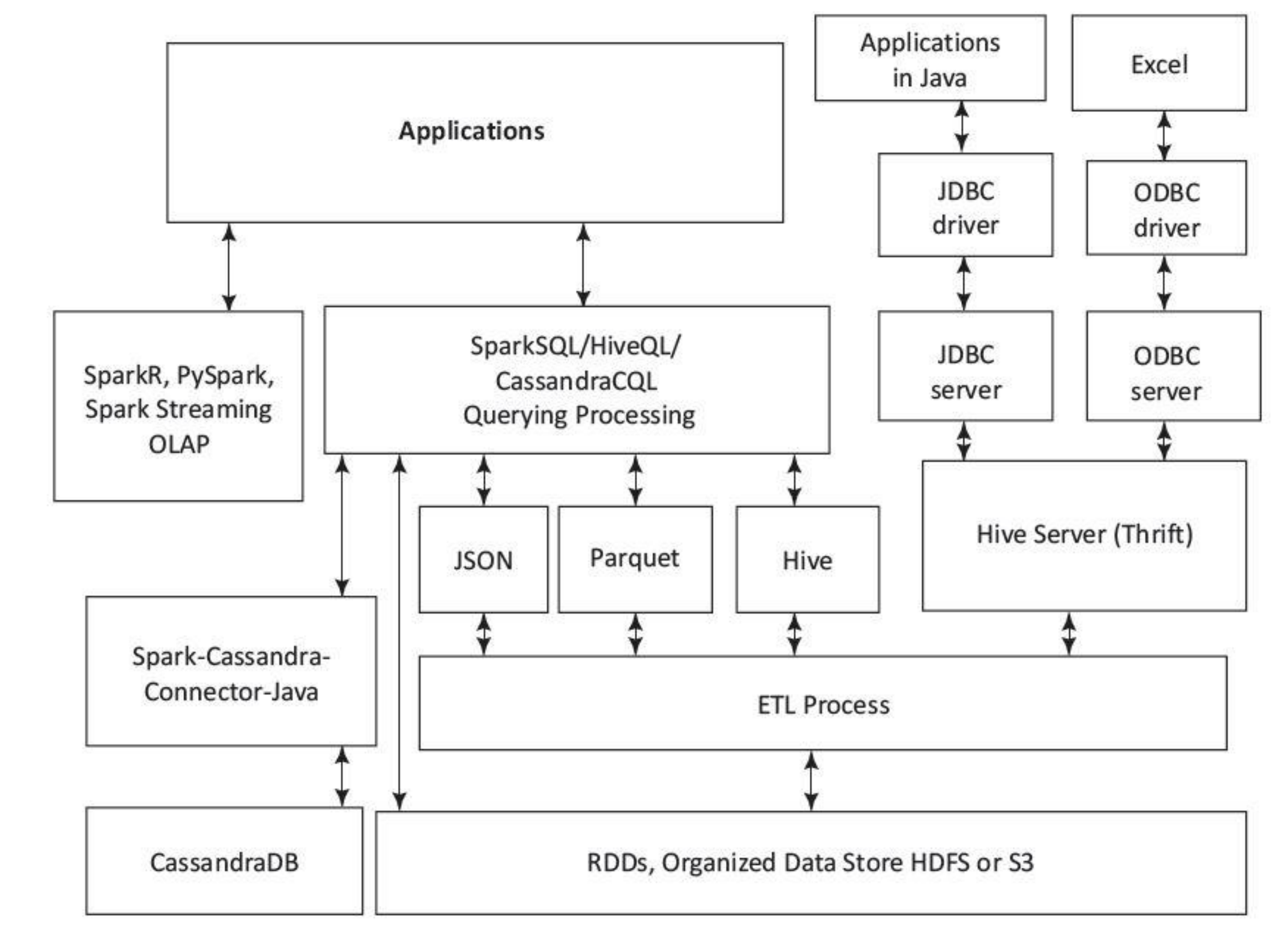
Refer Figure 5.4: Layer 5 Applications for analyzing data, for example, descriptive, predictive and prescriptive analytics, business processes (BPs), business process automation (BPA), business intelligence (BI), decision modelling and knowledge discovery..

Spark SQL Connectivity to Inputs

Refer Figure 5.5 Data Flow

- Cassandra DB, DataFrames, RDDs
- Data into Spark SQL /HiveQL/ CassandraCQL for Querying Processing either through Cassandra-Spark Connector in Java or Data in Parquet, JSON or Hive tables after ETL pipeline

Figure 5.5 Connectivity between the applications and Spark SQL


Applications
Applications in Java
Excel
JDBC driver
ODBC driver
SparkR, PySpark, Spark Streaming OLAP
SparkSQL/HiveQL/ CassandraCQL Querying Processing
JDBC server
ODBC server
JSON
Parquet
Hive
Hive Server (Thrift)
Spark-Cassandra-Connector-Java
ETL Process
CassandraDB
RDDs, Organized Data Store HDFS or S3

Spark SQL/Hive Server (Thrift) Connectivity to outputs

- Spark SQL API JDBC connectivity using JDBC/ODBC drivers
- to the Applications

JDBC Server

- An application reads the data tables in RDBMS using a JDBC client (JDBC API at the application)
- Applications in Java connect to databases using JDBC driver and server

Hive Server (Thrift)

- Enables a remote Hive client or JDBC driver to send a request to Hive and the server sends response to that
- The client requests can be in Scala, Java, Python or R

JSON, Hive, Parquet Objects

- HDFS is highly reliable for very long running queries
- IO operations are slow
- Columnar storage used for faster IOs
- Columnar storage stores the data portion, presently required for the IOs.

JSON, Hive, Parquet Objects

- HDFS is highly reliable for very long running queries. However, IO operations are slow. Columnar storage is a solution for faster IOs. Columnar storage stores the data portion, presently required for the IOs. Load-only columns access during processing. Also, a columnar object Data Store can be compressed or

Columnar object Data Store

- Load-only columns access during processing
- Can be compressed or encoded according to the data type
- Also, executions of different columns or column partitions can be in parallel at the data nodes.

A nested hierarchical columnar storage concept

- Apache Parquet three projects specify the usages of files for query processing or applications
- The projects are (i) parquet-format and Thrift definitions of metadata, (ii) parquet-mr and (iii) parquet-compatibility for compatibly for read-write in multiple languages

Project parquet-mr

- Implements the sub-modules in the core components for reading and writing a nested, column-oriented data stream,

Spark DataFrame (SchemaRDD)

- A distributed collection of data organized into named columns
- Used for transformation using filter, join, or groupby aggregation functions
- Section 10.3 for conversion from CSV format dataset and creating DataFrame from the RDDs.

DataFrames

- Created from different data sources,
- JSON datasets, Hive tables, Parquet row groups, structured data files, external Data Stores and RDDs

Summary

We learnt

- Steps between acquisition of data from different sources and its applications
- Data into Spark SQL /HiveQL/ CassandraCQL for Querying Processing either through Cassandra-Spark Connector in Java or Data in Parquet, JSON or Hive tables after ETL pipeline

Summary

- Connectivity between the applications and Spark SQL
- JDBC Driver
- Parquet, JSON and DataFrames as inputs to Spark SQL or Hive Server

End of Lesson 3 on

Data Analytics using Apache® Spark™ Components Spark SQL and DataFrames