INDUSTRIAL TRAINING

Apache Spark

Quick Start

- Interactive Analysis with the Spark Shell
  - More on Dataset Operations
  - Caching
- Self-Contained Applications
- Where to Go from Here

RDD and Shared Variables

- Linking with Spark
- Initializing Spark
  - Using the Shell
- Resilient Distributed Datasets (RDDs)
  - Parallelized Collections
  - External Datasets
  - RDD Operations
    - Passing Functions to Spark
    - Understanding closures
      - Example
      - Local vs. cluster modes
      - Printing elements of an RDD
    - Working with Key-Value Pairs
    - Transformations
    - Actions
    - Shuffle operations
      - Background
      - Performance Impact
  - RDD Persistence
    - Which Storage Level to Choose?
    - Removing Data
- Shared Variables
  - Broadcast Variables
  - Accumulators
- Deploying to a Cluster
- Launching Spark jobs from Java / Scala
- Unit Testing
- Where to Go from Here

  - SQL
  - Datasets and DataFrames
- Getting Started
  - Starting Point: SparkSession
  - Creating DataFrames
  - Untyped Dataset Operations (aka DataFrame Operations)
  - Running SQL Queries Programmatically
  - Global Temporary View
  - Creating Datasets
  - Interoperating with RDDs
    - Inferring the Schema Using Reflection
    - Programmatically Specifying the Schema
  - Aggregations
    - Untyped User-Defined Aggregate Functions
    - Type-Safe User-Defined Aggregate Functions
- Data Sources
  - Generic Load/Save Functions
    - Manually Specifying Options
    - Run SQL on files directly
    - Save Modes
    - Saving to Persistent Tables
    - Bucketing, Sorting and Partitioning
  - Parquet Files
    - Loading Data Programmatically
    - Partition Discovery
    - Schema Merging
    - Hive metastore Parquet table conversion
      - Hive/Parquet Schema Reconciliation
      - Metadata Refreshing
    - Configuration
  - ORC Files
  - JSON Datasets
  - Hive Tables
    - Specifying storage format for Hive tables
    - Interacting with Different Versions of Hive Metastore
  - JDBC To Other Databases
  - Troubleshooting
- Performance Tuning
  - Caching Data In Memory
  - Other Configuration Options
  - Broadcast Hint for SQL Queries
- Distributed SQL Engine
  - Running the Thrift JDBC/ODBC server
  - Running the Spark SQL CLI
- PySpark Usage Guide for Pandas with Apache Arrow
  - Apache Arrow in Spark
    - Ensure PyArrow Installed

  - Enabling for Conversion to/from Pandas
  - Pandas UDFs (a.k.a. Vectorized UDFs)
    - Scalar
    - Grouped Map
  - Usage Notes
    - Supported SQL Types
    - Setting Arrow Batch Size
    - Timestamp with Time Zone Semantics
- Migration Guide
  - Upgrading From Spark SQL 2.2 to 2.3
  - Upgrading From Spark SQL 2.1 to 2.2
  - Upgrading From Spark SQL 2.0 to 2.1
  - Upgrading From Spark SQL 1.6 to 2.0
  - Upgrading From Spark SQL 1.5 to 1.6
  - Upgrading From Spark SQL 1.4 to 1.5
  - Upgrading from Spark SQL 1.3 to 1.4
    - DataFrame data reader/writer interface
    - DataFrame.groupBy retains grouping columns
    - Behavior change on DataFrame.withColumn
  - Upgrading from Spark SQL 1.0-1.2 to 1.3
    - Rename of SchemaRDD to DataFrame
    - Unification of the Java and Scala APIs
    - Isolation of Implicit Conversions and Removal of dsl Package (Scala-only)
    - Removal of the type aliases in org.apache.spark.sql for DataType (Scala-only)
    - UDF Registration Moved to `sqlContext.udf` (Java & Scala)
    - Python DataTypes No Longer Singletons
  - Compatibility with Apache Hive
    - Deploying in Existing Hive Warehouses
    - Supported Hive Features
    - Unsupported Hive Functionality
    - Incompatible Hive UDF
- Reference
  - Data Types
  - NaN Semantics

Structured Streaming & Programming

- Quick Example
- Programming Model
  - Handling Event-time and Late Data
  - Fault Tolerance Semantics
- API using Datasets and DataFrames
  - Creating streaming DataFrames and streaming Datasets
    - Input Sources
    - Schema inference and partition of streaming DataFrames/Datasets
  - Operations on streaming DataFrames/Datasets
    - Operations - Selection, Projection, Aggregation
    - Window Operations on Event Time
      - Handling Late Data and Watermarking
    - Join Operations
      - Stream-static Joins
      - Stream-stream Joins
        - Inner Joins with optional Watermarking
        - Outer Joins with Watermarking
        - Support matrix for joins in streaming queries
    - Streaming Deduplication
    - Arbitrary Stateful Operations
    - Unsupported Operations
  - Starting Streaming Queries
    - Output Modes
    - Output Sinks
      - Using Foreach
    - Triggers
  - Managing Streaming Queries
  - Monitoring Streaming Queries
    - Reading Metrics Interactively
    - Reporting Metrics programmatically using Asynchronous APIs
    - Reporting Metrics using Dropwizard
  - Recovering from Failures with Checkpointing
- Continuous Processing
- Additional Information

Spark Streaming Programming

- A Quick Example
    - Linking
    - Initializing StreamingContext
    - Discretized Streams (DStreams)
    - Input DStreams and Receivers
    - Transformations on DStreams
    - Output Operations on DStreams
    - DataFrame and SQL Operations
    - MLlib Operations
    - Caching / Persistence
    - Checkpointing
    - Accumulators, Broadcast Variables, and Checkpoints
    - Deploying Applications
    - Monitoring Applications
- Performance Tuning
    - Reducing the Batch Processing Times
    - Setting the Right Batch Interval
    - Memory Tuning
- Fault-tolerance Semantics
- Where to Go from Here