Apache Spark Through Email

Markus Dale, medale@asymmetrik.com July 2024

Intro, Slides And Code

- Slides: https://github.com/medale/sparkmail/blob/master/presentation/ApacheSparkThroughEmail.pdf
- · Spark Code Examples: https://github.com/medale/spark-mail/
 - README.md describes how to get and parse Enron email dataset

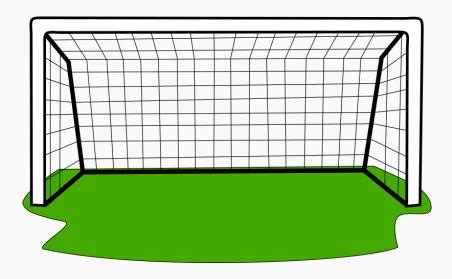


Figure 1: Intro to Apache Spark

Data Science for Small Dataset



Figure 2: Laptop

4

Data Science for Larger Dataset (Vertical Scaling)

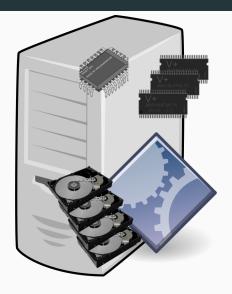
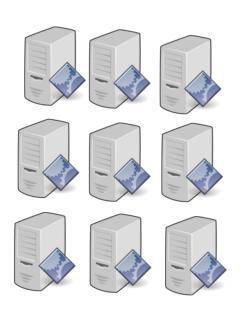


Figure 3: Beefed-up Server

Data Science for Large Datasets (Horizontal Scaling)



Early Big Data Framework - Apache Hadoop



Figure 5: HDFS, MapReduce

Hadoop Ecosystem

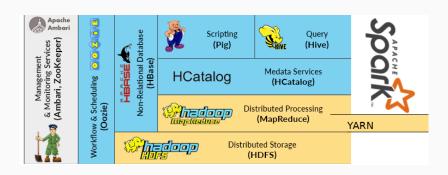


Figure 6: Some Frameworks Around Hadoop

Running Spark

- Local
 - · Download from https://spark.apache.org, untar, add to PATH
 - SDKMAN curl -s "https://get.sdkman.io" | bash
 sdk install spark
 - spark-shell or pyspark
- · Standalone cluster, Hadoop YARN
 - · Need shared file system or common datastore (e.g. AWS S3)
- · Cloud-based managed:
 - · AWS FMR
 - · GCP Dataproc
 - Databricks on Azure, GCP or AWS

Apache Spark API

- Scala
- · Java
- Python
- R
- => Project Tungsten code generation

Apache Spark Components

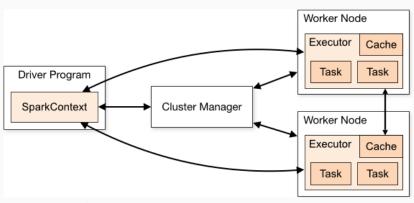
Structured Advanced Libraries & Streaming Ecosystem Analytics Structured APIS Datasets DataFrames SQL Low-level APIS Distributed Variables RDDs

Source: Spark: The Definitive Guide

Hello, Spark Email World!

- · Jupyter Notebook with Apache Toree
- · See Notebook ../notebooks/html/ApacheSparkThroughEmail1.html

Cluster Manager, Driver, Executors, Tasks



Source: Apache Spark website

SparkSession: Entry to cluster

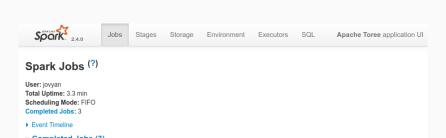
```
• spark: spark.sql.SparkSession
//SparkSession provided by notebook or shell as spark
val homeDir = sys.props("user.home")
val records = spark.read.
   parquet(s"$homeDir/datasets/enron/enron-small.parquet")
//In regular code for spark-submit
//com.uebercomputing.spark.dataset.TopNEmailMessageSenders
val spark = SparkSession.builder().
   appName("TopNEmailMessageSenders").
   master("local[2]").getOrCreate()
```

DataFrameReader: Input for structured data

- spark.read: spark.sql.DataFrameReader
 - jdbc
 - · json
 - · parquet
 - · text...
 - · Also: https://spark-packages.org Avro, Redshift, MongoDB...

Transformations vs. Actions

- Transformation: returns a new RDD (nothing gets executed)
 - · read, cache, filter...
- Actions: trigger execution, catalyst query optimizer, Tungsten code generation
 - · count, write, '
 - Scaling Behind the Scenes



Stages: Pipeline work per stage - shuffle

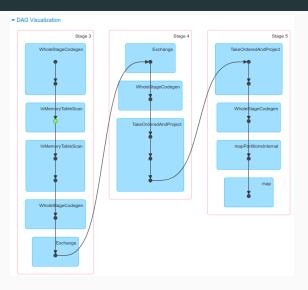
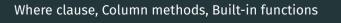


Figure 8: Stages



 $\cdot \ \ See \ Notebook ../notebooks/html/ApacheSparkThroughEmail2.html$

Spark APIs - DataFrameReader, Dataset, Column, functions

>	<pre>def parquet(paths: String*): DataFrame Loads a Parquetfile, returning the result as a DataFrame.</pre>		C organization state and Dataset
	def parquet(path: String): DataFrame		class Dataset[T] extends Serializable
	Loads a Parquet file, returning the result as a DataFrame.		A Dataset is a strongly typed collection of domain-specific objects that can be transformed in parallel using functional or relational operations.
	def schema(schemaString: String): DataFrameReader		Operations available on Datasets are divided into transformations and actions. Transformations are the ones that produce new Datasets, and fitter, select, and aggregate (group@r), Example actions count, show, or writing data out to tile systems.
	Specifies the schema by using the input DDL-formatted string.		Datasets are "szy", i.e. computations are only triggered when an action is invoked. Internally, a Dataset represents a logical plan that describe optimizes the logical plan and generates a physical plan for efficient execution in a parallel and distributed manner. To expire the logical plan
	def schema(schema: <u>StructType</u>): <u>DataFrameReader</u> Specifies the input schema.		To efficiently support domain-specific objects, an <u>finodate</u> is required. The encoder maps the domain specific type T to Span's internal type is used to tell Span's to generate code at nurtine to sentitude the Ptr'son' object into a binary structure. This binary structure often has much lose To understand the internal binary representation for data, use the scheme function.
			 There are typically two ways to create a Dataset. The most common way is by pointing Spark to some files on storage systems, using the received.
	def table(tableName: String): <u>DataFrame</u> Returns the specified table as a DataFrame.		<pre>wal people = spark.read.parquet("").as[Person] // Scala DatasetePerson> people = spark.read().parquet("").as(tncoders.bean(Person.class)); // Java</pre>
	def text(paths: String*): DataFrame		Datasets can also be created through transformations available on existing Datasets. For example, the following creates a new Dataset by ap-
	Loads text files and returns a DataFrame whose schema starts	with a string column named "value	<pre>wal names = people.map(_name) // in Scala; names is a Dataset(String) Dataset<string> names = people.map((Person p) -> p.name, Encoders.STRIMG));</string></pre>
	def text(path: String): DataFrame		Detaset operations can also be untyped, through various domain-specific-language (DSL) functions defined in: Dataset (this class), <u>Column</u> , a in R or Python.
	Loads text files and returns a DaTaFraffe whose schema starts with a string column named "value",		
	<pre>def textFile(paths: String*): Dataset[String]</pre>		<pre>wal ageCol = people('age") // in Scala Column ageCol = people.col('age"); // in Java</pre>
	Loads text files and returns a <u>Dataset</u> of String.		Note that the <u>Column</u> type can also be manipulated through its various functions.
	def textFile(path: String): <u>Dataset</u> [String] Loads text files and returns a <u>Dataset</u> of String.		<pre>// The following creates a new column that increases everybody's age by 10. people("age") + 10 // in Scala people.col("age").psis(19); // in Java</pre>
Expression opera	ators	Date time functions	
>	def %(other: Any): <u>Column</u>		f add months(startDate: Column, numMonths: Int): Column
	Modulo (a.k.a.		
			Returns the date that is numMonths after startDate.
>	def &&(other: Any): <u>Column</u>	→ de	Returns the date that is numMonths after startDate. f current_date(): Column
>	def 66(other: Any): <u>Column</u> Bookean AND.	▶ de	
>	Boolean AND. def *(other: Any): <u>Column</u>		f current_date(): Column
>	Boolean AND.		f current_date(): Column Returns the current date as a date column.
>	Boolean AND. def *(other: Any): <u>Column</u> Multiplication of his expression and another expression. def +(other: Any): <u>Column</u>	▶ de	Corrent_date(): Collum Nourrest date outered date as calab column. Courrest_Classification; Collumn Robum the convent threatmap as a larendamp column. deta_edd(state; Collumn, days; Ent): Collumn
> >	Boolean AND. def *(other: Any): _Column Multiploation of this expression and another expression. def +(other: Any): _Column Sum of this expression and another expression.	> de	General_date(). SQLIBN Number becurred tells a value doctum. Centreal_Lissettep()1: CQLIBN Number becurred tells and sub-doctum. det_edifficient: CQLIBN. depty: R611: CQLIBN Number becurred tells you've dops tells filed.
>	Boolean AND. def *(other: Any): <u>Column</u> Multiplication of his expression and another expression. def +(other: Any): <u>Column</u>	> de	Corrent_Gate(): Collum Rethrom to correct date as a cata column. Corrent_Gate(suppl): Collum Rethrom to correct date as a cata column. Rethrom to correct directange of a threatong column. Rethrom to correct directange of a threatong column. Rethrom to column direct (column, forest): TAT1: Column Rethrom to date bear of days days after Staff date_Format(dateSept: Column, forest): String): Column Art Correct Column forest: String): Column Column direct Column forest: String): Column Column direct Column forest: String): Column Column direct Column forest: String): Column forest: String): Column forest: String): Column forest: Column forest: String): Column forest: Column forest
> > >	Boolean AND. 64 *(otter: Apy) : Calum Multiplication of the expression and arother expression. 64 *(otter: Apy) : Calum Sum of the expression and arother expression. 64 *(otter: Apy) : Calum Sumbertion.	> de > de	Cerrent_date): Collan Nothing the covered date as a radio doutine, Cerrent_datestamp(): Collane Robins the covered date as a radio doutine, Cerrent_datestamp(): Collane Robins the covered datestamp(): Exiting Cellane Ce
> > >	Boolea AND. def *(other: Apy): Column Multiplication of the expression and another expression. def *(other: Apy): Column Our des expression and another expression. def -(other: Apy): Column Debetschon. def /(other: Apy): Column Debetschon. Debetschon column Doministration of the Apy of the A	> de > de > de	Gerrent_deta(): Column Manume becomended as a called column. Cerrent_Listesteep(): Column Manume becomend dates a called column. Getter_deta(start:_Column, days:_test):_Column Manume becomend from days days age set start of deta_det(start:_Column, days:_test):_Column Manume become beat called column. Getter_deta_deta_deta_column. Getter_deta_deta_column. Getter_deta_deta_column. Getter_deta_deta_deta_column. Getter_deta_deta_column. Getter_deta_deta_co
> > > >	Boolean AND. 64 *(otter: Apy) : Calum Multiplication of the expression and arother expression. 64 *(otter: Apy) : Calum Sum of the expression and arother expression. 64 *(otter: Apy) : Calum Sumbertion.	> de > de > de	Cerrent_date): Colline Robbus the covered date as nation column, Cerrent_datestup(): Column Robbus the covered date as nation column, Cerrent_datestup(): Column Robbus the covered datestup(): Extra Column Robbus the covered datestup(): Extra Column Robbus the column April 1987; Column Robbus the column April 1987; Column Covered as adeltonoidampointing to a value of strong in the formal specified by the date formal given by the second argument Certex_datestup(): Column Colum
> > >	Boolean AND. 64 *(steter.Apy)* Calum Multiplication of the expression and arother expression. 64 *(steter.Apy)* Calum Sum of the expression and arother expression. 64 *(steter.Apy)* Calum Sumbertion. 64 *(steter.Apy)* Calum Division the expression by sumber expression. 64 *(steter.Apy)* Calum Division the expression by sumber expression.	> de > de > de > de	Current_Classtan Culture Nutrition Countred (in a radio outside) Culture C
> > > > > > >	Boolean AND. 64 *(steter.Apy)* Callum Multiplication of this expression and another expression. 64 *(steter.Apy)* Callum Sum of this expression and another expression. 64 *(steter.Apy)* Callum Subsection. 64 *(steter.Apy)* Callum Division this expression by subsection and expression. 64 *(steter.Apy)* Callum Lies Bruit. 64 *(steter.Apy)* Callum Lies Bruit.	> de > de > de > de	General (Ast-1): Column Returns to current date a value doctorm. Current, Lisesteep(1): Column Returns to current date as a value doctorm. Gette, add (Start): Column, days; 18c1): Column Returns to doctor the large by day day after start of a feet, add (Start): Column Gette, add (Start): Column, format: String): Column Gette, format(Statt): The format days day and start start Gette, format(Statt): The format days days day after start Gette, format(Statt): The format days days days days after start Gette, format (Start): Column Returns the date of the start of

Parallelism and Partitioning

- · Goldilocks not too many, not too few
- · Initial parallelism number of input "blocks"
- $\cdot \ \, \text{Shuffle-spark.sql.shuffle.partitions} \ \, \text{configuration}$

Explode, Shuffle Partitions, UDF, Parquet partition

 $\cdot \ \ See \ Notebook ../notebooks/html/ApacheSparkThroughEmail3.html$

Pandas on Spark

 $\cdot \ \ \text{See https://spark.apache.org/pandas-on-spark/}$

Apache Parquet/Apache Arrow

- · Avro record-oriented data format
- · Parquet column-oriented data format by page
- Arrow share memory for Python
 (https://spark.apache.org/docs/latest/api/python/user_guide/sql/arrow_page)

Resources

- https://spark.apache.org/
- https://spark-packages.org/ Community 3rd party packages (e.g. data sources)
- https://sparkbyexamples.com/

And now for something completely different: Colon Cancer



- Screening saves lives!
 - Colonoscopy talk to your doc
- · Colorectal Cancer Alliance

Questions?





- · markus.dale@bluehalo.com
- Infrequent blog/past presentations http://uebercomputing.com/
- · Spark Mail repo https://github.com/medale/spark-mail/