Spark

1 使用mvn开发scala

<http://blog.csdn.net/kwu_ganymede/article/details/51832427>

## 二、SparkSQL

# Overview

1 同spark的区别是： 处理有结构的数据，并做了优化

2 执行引擎用的同一个，但是外面使用的语言可以多个。

# Datasets and DataFrames

Dataset概念：RDD

DataFrame 概念：数据有个结构

在scala中：相当于 Dataset[Row]

在java中：相当于 Dataset<Row>

# Getting Started

## Starting Point: SparkSession

先调用静态成员函数builder，生成一个Builder；

在Builder中配置各种属性

调用 getOrCreate方法，返回一个SparkSession对象。

## Creating DataFrames

DataFrame来源：

1 RDD转换而来，引入隐式转换的声明

2 外部文件：读取文件的api都在DataFrameReader中

1 json文件: read.json()

2 text文件:

3 csv文件

## Untyped Dataset Operations

所有的操作都在DataSet中，DataFrame是DataSet[Row]的别名

1 select: select()

2 where: filter()

3 group: groupBy()

4 count(\*) : count

**Running SQL Queries Programmatically**

使用程序中+sql语句的方式，就像查询一个表。需要把df转换成一个临时的视图。

1 createOrReplaceTempView(“”) //创建一个临时表

2 spark.sql("SELECT \* FROM people") //执行sql语句

## Global Temporary View

临时视图是session相关的，以session为生命周期。

要想跨session共享，并直至程序结束，可以创建一个全局的view。

Api：

df.createGlobalTempView("people") //创建一个全局的临时表

## Creating Datasets

DataSet还有一套类似DataFrame的api。

Dataset和Rdd的区别：

1 序列化方式不一样

Rdd是用传统的序列化方式（kryo或其他）将其转换为字节

Dataset虽然也是转换成为字节，但是是有结构的，encoders是动态生成的，支持一系列对数据的操作。

## Interoperating with RDDs

将rdd转换为Dataset有两种方式:

1 使用反射的方式推断schema

定义一个case class

2 程序中动态生成schema

缺点：复杂

优点：不用提前建立，等知道数据后，根据数据来动态生成schema

# Data Sources

### Manually Specifying Options

写入文件：指明写入的格式和名称

write.format("parquet").save("namesAndAges.parquet")

### Save Modes

写并不是原子的也没使用锁，是线程不安全的。

指明写的模式

### Saving to Persistent Tables

可以直接写到hive中，并生成元数据文件。

## Parquet Files

### Loading Data Programmatically

### Partition Discovery

## JSON Datasets

## JDBC To Other Databases

1 直接通过jdbc读取数据库中的内容

# Performance Tuning

## Caching Data In Memory

像rdd一样，把表缓存到内存中。会根据配置自动压缩。

Api:

spark.cacheTable("tableName") or dataFrame.cache()

也可以从内存中删除这个表

## Other Configuration Options

还要其他配置用于调优。

# Distributed SQL Engine

# 其他

# 1 RDD、DataFrame和DataSet的区别

<http://www.jianshu.com/p/c0181667daa0>

<https://databricks.com/blog/2016/07/14/a-tale-of-three-apache-spark-apis-rdds-dataframes-and-datasets.html>

RDD与DataFrame区别：

1 RDD: 中的数据是不可变的。操作后对对象的复制

DataFrame 中的内容是可变的，对原对象进行操作。

2 框架能知道DataFrame的结构，但是不止RDD中每个元素的。

DataFrame和DataSet的区别和联系

1

什么时候使用RDD？

1 使用low-level的api：转换和行为

2 数据没有结构化，即没有特定的列名字

3 想使用rdd的API而不是领域专用语言

4 不使用schema，使用列名访问数据

5 放弃性能优化

RDD被废弃了吗？

没有，当使用底层api事。 三者之间可以无缝切换，DataFrame和DataSet是在RDD之上的。

## DataFrames

1 也是不可变的数据，但是有结构。就像一张表。

因此，可以使用领域专用语言（sql）

2 DataFrame和DataSet api进行了合并。两种api： 有类型的和无类型的。

## Datasets

1 Datasets 是强类型的，

## Benefits of Dataset APIs

2

<https://databricks.com/blog/2015/07/15/introducing-window-functions-in-spark-sql.html>

<http://stackoverflow.com/questions/33655467/get-topn-of-all-groups-after-group-by-using-spark-dataframe>

源码

<https://my.oschina.net/corleone/blog?sort=time&p=3&temp=1490693518474>

# 二、API

<http://www.jianshu.com/p/b3caf825a4b9>

<http://blog.csdn.net/u012102306/article/details/51762294>

## 1、flatMapValues

1首先是对Pari(k,v)的操作-》是对v的操作

2 v原来是一个值，经过操作后，v变成一个数组

3 k只有一个，flat操作相当于 k join (v1, v2, v3...)

4 最终是：（k,v1），（k,v2）,(k,v3)….(k2,v….)