SparkContext: 是Spark应用程序的Spark入口点

但入口点从2.0版本开始变成了SparkSession

SparkContext实体创建后可以被用来：

1. 创建RDDs
2. 创建累加器
3. 创建广播变量：将数据从一个节点发送到其他节点，且这些数据只能读，无法更新，
4. 进入Spark 服务并开始工作
5. Sparkcontext 本质上来说是Spark执行环境的客户端并且作为Spark应用程序的主文件

一个Spark 程序必须做的第一件事就是创建SparkContext对象，这个对象可以告诉Spark怎么去进入一个集群

Spark shell中，变量名为sc的变量已经被创建，它是一种特殊的可识别的解释器的SparkContext

RDD 持续性：缓存

Spark最重要的功能之一是在内存中跨操作缓存一个数据集

当缓存一个RDD文件时， 每个节点都会储存一部分数据， 你可以再其他action命令中重复使用该文件

每个缓存的RDD文件可以使用不同的保存级别来进行存储。

如：仅保存再内存：

1 将RDD作为反序列化的java对象存储再JVM中

2 如果RDD不能容纳在内存中，这些内存将不会被缓存，并在需要时重新计算

3 这是默认级别

内存和硬盘：

1 如果RDD不能容纳再内存，保存这些分区也无法容纳在硬盘， 并且也无法在他们被需要时能够被阅读

为什么要缓存RDD：

如果要再次执行 errors.count()， 这个文件将会被再次载入和计算

持久化将会告诉Spark去缓存数据进内存， 并且可以为之后对同样数据进行actioin操作而减少数据载入的花费

Error.persist()将不执行任何操作，这事一个懒惰的操作，但是RDD会说：先读取此文件，然后缓存内容。 该动作将触发计算和数据缓存。

Spark Key-Value RDDs

跟map reduce类似，spark 也支持 key-values (感觉跟字典类似)

一组 RDD 的每一个元素都是一个成对的元组（数据无法更改么？）

Reducebykey(返回一个新的分布式数据集对)

SortBykey()： 返回一个新的数据对并且根据key以上升趋势排列

Groupbykey(): 返回一个新的数据对

1挑战：如何从各种不同的数据源中执行ETL操作（提取，转换，加载）

解决办法： 一个dataframe的 API可以在外部数据源和Spark的内置RDD上执行相关操作

2 挑战：执行高级分析（比如机器学习， 图像处理）这些非常难以用相关系统来表达

解决办法： 一个高度扩展的优化器，Catalyst可以使用scala的特征值来添加可组合的规则，控制代码生成，并且定义扩展。

DataFrame-based API for MLlib:

又被称为 “管道” API, 这个API是用于连接构建ML管道的接口

在2.0版本中， 这个dataframe-based API 将会变成对于MLlib的初级API

如：org.apache.spark.ml, pyspark.ml

基于RDD模式的API将会进入维护模式：

仍然会有bug修复，但不会有新功能

如： org.apache.spark.mllib, pyspark.mllib

构建Spark： 数据框架和数据集

Dataframe:（架构， 通用无类型）

：通过索引进入， 被叫做column（像一个table）

Dataset(静态类型，强类型)

Dataframe = dataset[Row]

RDD是低级的（如汇编器）， dataframe和数据集建立在RDD之上

Dataset APIs 好处

静态类型 且 运行时数据类型安全：

SQL的限制最少，直到运行时才出现语法错误

DF/DS:在编译时检测到语法错误

对结构化和半结构化数据的高级抽象和自定义视图

结构简单的API， 丰富的语义和特定于域的操作

Performance and Optimization

SQL Catalyst

DataFrame:

具有相同架构的行的分布式集合

可以从外部数据源或RDD构造成本质上为Row对象的RDD

支持关系运算符（eg where， groupby）以及spark操作

懒惰评估： 未实现的逻辑计划

Spark Dataframes 很快

RDD python最快，其次RDD scala spark python DF和spark scala DF差不多

空间有效性：

RDD远大于datasets

Machine learning library (MLlib)

ML algorithms: 常用的学习算法如分类，回归，聚类，协同过滤

Featurization（特征化）：特征提取， 转换， 维度下降， 选取

Piplines（管线）： 用于构建评估并且调整机器学习管线的工具

Persistence(持久化): 保存和假再算法，模型和管线

Utilities（实用程序）： 线性代数，统计信息，数据处理等