SPARK 2.0新特性与展望

董西成 @ HULU

自我介绍

- 董西成
- 就职于HULU北京研发中心
- 《Hadoop技术内幕》书籍作者
- 专注于Hadoop与Spark等大数据技术

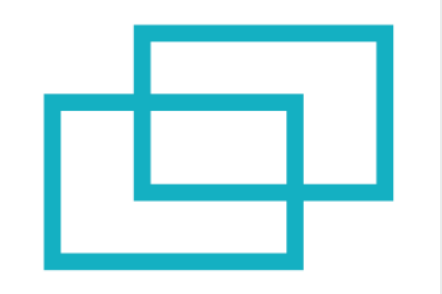
AGENDA

- Spark 2.0主要改进与优化
- Spark Core与Spark SQL
- Spark Streaming
- MLlib
- Spark 2.0现状与展望
- Q & A

SPARK 2.0主要改进与优化：功能方面

Spark Core/SQL:
Tungsten Phase 2
优化CPU与Memory方面

Structured Streaming



Spark SQL 2003 &
统一化DataFrame与Dataset

SPARK 2.0主要改进与优化：性能方面

- 利用“whole stage code generation”，使得SQL和DataFrame中算子性能优化2-10x
- 通过“向量化计算”提升Parquet扫描吞吐率
- 提升ORC读写性能
- 提升Catalyst查询优化器性能
- 提升窗口函数性能

HULU 2017校园招聘开始了！

扫描二维码！

关注hulu官方微信，获取最新校招信息

面试流程

在线申请 > 在线笔试 > Onsite面试 > Offer

网申地址

http://www.hulu.com/jobs/positions/location/beijing

AGENDA

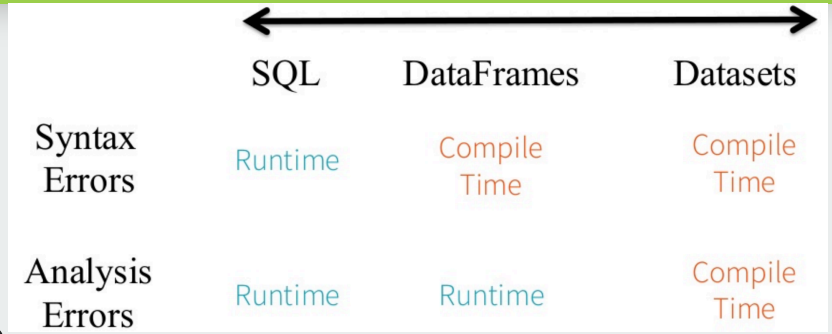
- Spark 2.0主要改进与优化
- Spark Core与Spark SQL
- Spark Streaming
- MLlib
- Spark 2.0现状与展望
- Q & A

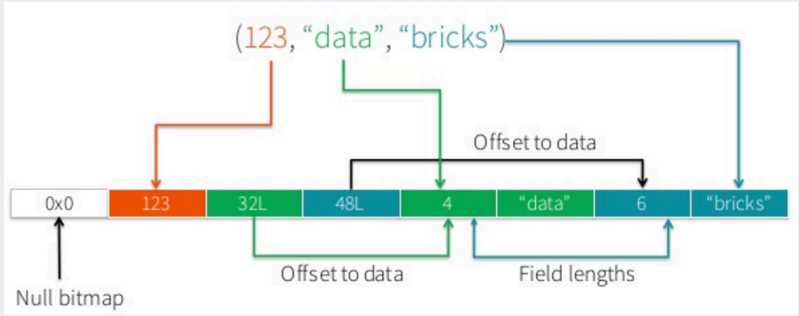
SPARK CORE：ACCUMULATOR

- 新的accumulator API（旧的被标注为“deprecate”，SPARK-14626）
  - 设计动机：克服已有accumulator的问题
    - 类型层级过于复杂：Accumulator, Accumulable, AccumulatorParam, AccumulableParam
    - 要求输入输出类型必须匹配，导致一些累加器难以实现，比如average
    - 没有一个统一的API同时适用于Java和Scala

SPARK SQL：API（1）

- 统一DataFrame与Dataset
  - DataFrame存在问题
    - 不是类型安全的(not type-safe)
    - 缺乏函数式编程能力(not object-oriented)
  - Dataset
    - 类型安全，面向对象编程方式
    - 支持非结构化数据(json)
    - java与scala统一接口
    - 性能极好的序列化框架





SPARK SQL：EXAMPLES

Using RDDs

```
data = sc.textFile(...).split("\t")
data.map(lambda x: (x[0], [int(x[1]), 1])) \
    .reduceByKey(lambda x, y: [x[0] + y[0], x[1] + y[1]]) \
    .map(lambda x: [x[0], x[1][0] / x[1][1]]) \
    .collect()
```

Using DataFrames

```
sqlCtx.table("people") \
    .groupBy("name") \
    .agg("name", avg("age")) \
    .collect()
```

Dataset示例

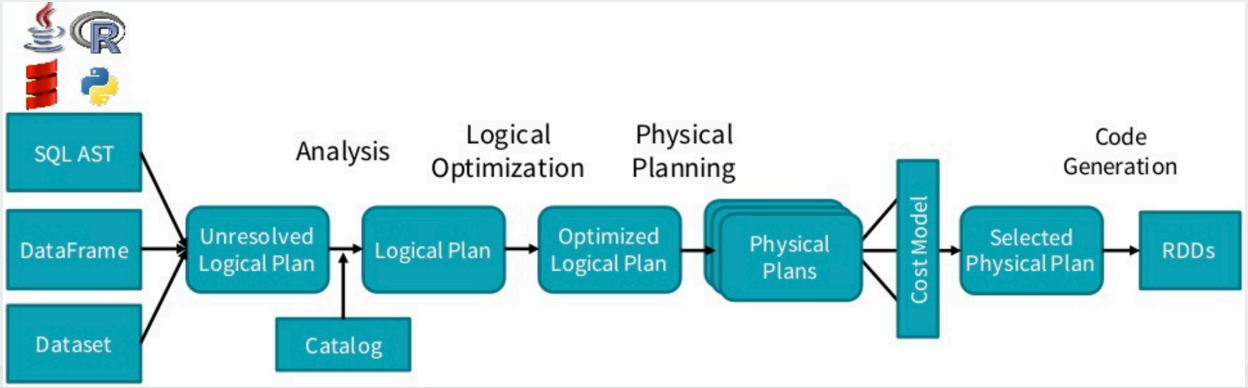
```
val df = ctx.read.json("people.json")

// Convert data to domain objects.
case class Person(name: String, age: Int)
val ds: Dataset[Person] = df.as[Person]
ds.filter(_.age > 30)

// Compute histogram of age by name.
val hist = ds.groupBy(_.name).mapGroups {
  case (name, people: Iter[Person]) =>
    val buckets = new Array[Int](10)
    people.map(_.age).foreach { a =>
      buckets(a / 10) += 1
    }
    (name, buckets)
}
```

SPARK SQL：API（2）

- 统一各种Context
  - SparkSession = SQLContext + HiveContext + StreamingContext
    - 其他Context仍将保留以保证向后兼容性
  - SparkSession与SparkContext
    - SparkSession：面向Dataset的编程（high level）
    - SparkContext：面向RDD的编程（low level）



SPARK SQL: TUNGSTEN PHASE 2

- 专注优化CPU和memory方面
  - 性能提升5~20X
- 优化点(具体参考这篇文章)
  - Memory Management and Binary Processing
    - Java GC严重，且java对象内存开销大，可采用类似C语言机制，直接操纵binary data(sun.misc.Unsafe)
  - Cache-aware Computation
    - 合理使用CPU的L1/L2/L3 cache，设计对cache友好的算法
  - Code Generation
    - 可去除条件检查，减少虚函数调度等

SPARK SQL: 支持SQL 2003

- 可跑通所有(99个)TPC-DS查询
- 改进
  - 解析器可同时支持ANSI-SQL 和Hive QL
  - 实现了DDL
  - 支持大部分子查询
  - 支持View

HULU招聘：大数据高级工程师

- 地点：北京
- 方向：Big Data，Pipeline，BI，OLAP，Batch, realtime
- 邮箱：kunliang.zhao#hulu.com (#换为@)

WHAT YOU'LL DO

- Develop, scale, and improve the data pipeline / analytics / BI / User persona system
- Contribute to the design, architecture and implementation of a data-engineering infrastructure
- Handle architectural and design considerations such as performance, scalability, reusability and flexibility issues
- Advocate engineering best practices, including use of design patterns, code review and automated unit/functional testing
- Collaborate with program managers, and product managers, researchers in an open, creative environment
- Review the technical design and perform code review of other developers' work
- Coach and provide guidance to junior team members

WHAT TO BRING

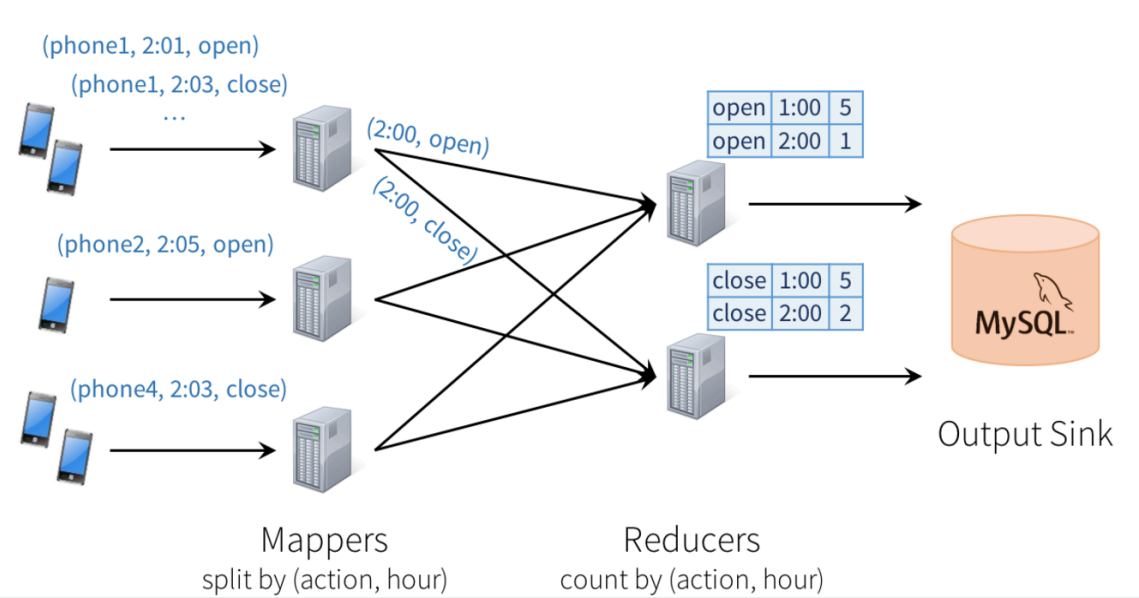
- BS or MS degree in CS related major
- 5+ years of professional programming and design experience in Scala/Java/Python and Linux
- Strong knowledge of Systems / Application Design & Architecture
- Experience with processing large amounts of data at petabyte level
- Knowledge of Hadoop, Hive, Hbase, Spark, Pig, Kafka, Druid
- Excellent leadership and communication skills.
- Experiences on CRM / DMP / user persona related work is a plus

AGENDA

- Spark 2.0主要改进与优化
- Spark Core与Spark SQL
- Spark Streaming
- MLlib
- Spark 2.0现状与展望
- Q & A

STRUCTURED STREAMING : BACKGROUND

```
SELECT action, WINDOW(time, "1 hour"), COUNT(*)
FROM events
GROUP BY action, WINDOW(time, "1 hour")
```



- 问题
  - 一致性
  - 容错
  - 乱序

[Structured Streaming详细介绍]

STRUCTURED STREAMING

- 基于Spark SQL构建了high-level API
  - DataFrame / Dataset
  - Event time, windowing, sessions, sources & sink
- Joining Streams with Static Data
  - 连接流式数据与静态数据集
- 交互式查询结果
  - 通过JDBC server将RDD结果暴露出去，以便于交互式查询

STRUCTURED STREAMING与其他系统比较

Property	Structured Streaming	Spark Streaming	Apache Storm	Apache Flink	Kafka Streams	Google Dataflow
Streaming API	incrementalize batch queries	integrates with batch	separate from batch	separate from batch	separate from batch	integrates with batch
Prefix Integrity Guarantee	✔	✔	✘	✘	✘	✘
Internal Processing	exactly once	exactly once	at least once	exactly once	at least once	exactly once
Transactional Sources/Sinks	✔	some	some	some	✘	✘
Interactive Queries	✔	✔	✘	✘	✘	✘
Joins with Static Data	✔	✔	✘	✘	✘	✘

CONTINUOUS APPLICATION

Batch Aggregation

```
logs  = ctx.read.format("json").open("s3://logs")

logs.groupBy(logs.user_id).agg(sum(logs.time))
    .write.format("jdbc")
    .save("jdbc:mysql//...")
```

Continuous Aggregation

```
logs  = ctx.read.format("json").stream("s3://logs")

logs.groupBy(logs.user_id).agg(sum(logs.time))
    .write.format("jdbc")
    .startStream("jdbc:mysql//...")
```

AGENDA

- Spark 2.0主要改进与优化
- Spark Core与Spark SQL
- Spark Streaming
- MLlib
- Spark 2.0现状与展望
- Q & A

MLLIB 2.0

- ML Pipeline
  - 可以持久化到磁盘(saving & loading)
- 基于DataFrame的API
  - 新包:org.apache.spark.ml, pyspark.ml
  - 基于RDD的API不再更新
    - org.apache.spark.mllib, pyspark.mllib

- MLlib 2.0新特性
  - 广义线性模型
  - Python & R API
  - 模型持久化能力
  - Pipieline定制化

MLLIB 2.0：GLM

- GLM(广义线性模型)
  - 统一了线性回归和逻辑回归等模型，ML中最重要的model
- 实现的模型
  - Logistic regression
  - Linear regression

  } Spark 1.6或更早
  - Many other types of models
  - Model summary statistics

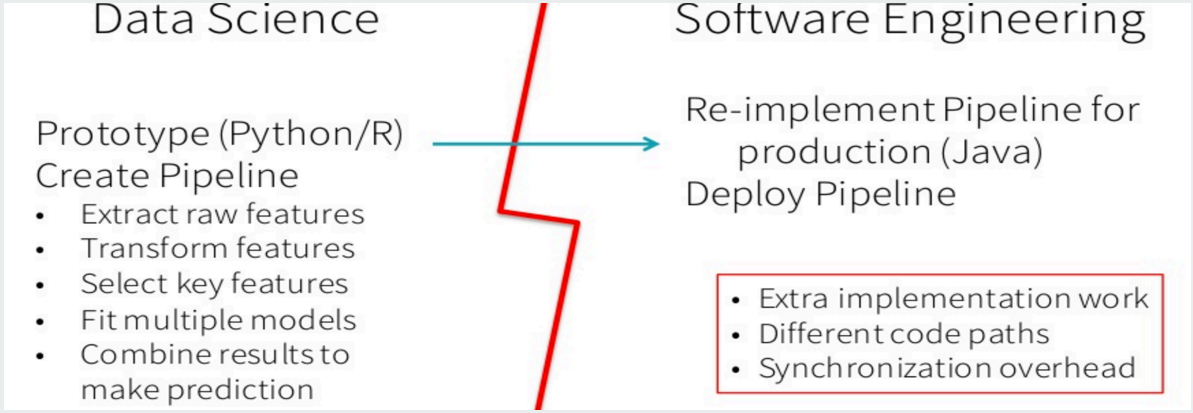
Model family	Supported link functions
Gaussian	Identity, Log, Inverse
Binomial	Logit, Probit, CLogLog
Poisson	Log, Identity, Sqrt
Gamma	Inverse, Identity, Log

MLLIB 2.0：PYTHON & R API

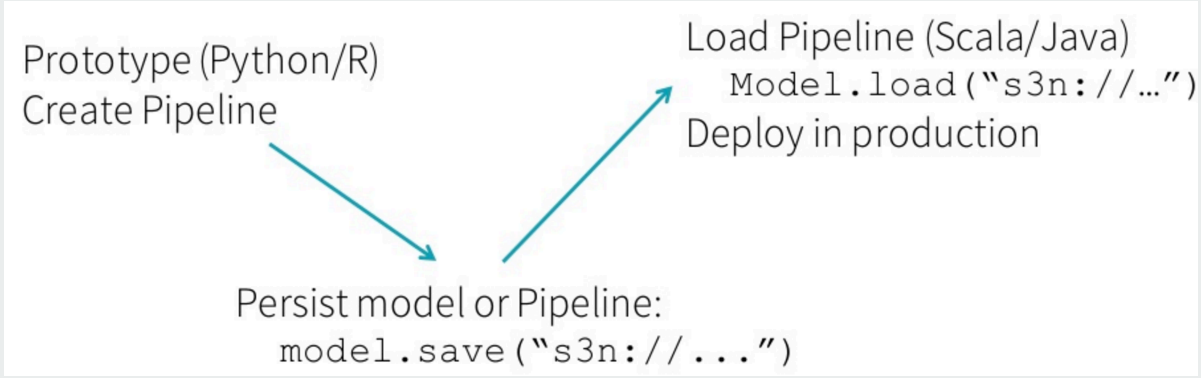
- 动机
  - 扩展ML API，支持更广泛的语言，为数据科学家提供便利
- Python算法
  - Clustering algorithms: Bisecting K-Means, Gaussian Mixtures, LDA
  - Meta-algorithms: OneVsRest, TrainValidationSplit
  - GeneralizedLinearRegression
  - Feature transformers: ChiSqSelector, MaxAbsScaler, QuantileDiscretizer
  - Model inspection: summaries for Logistic Regression, Linear Regression, GLMs
- R算法
  - Regression & classification: Generalized Linear Regression, AFT survival regression
- Clustering: K-Means

MLLIB 2.0：模型持久化

传统方式



优化后



- 现状
  - Java/Scala: ✓
  - Python/R: going
- 统一的底层实现
- 数据交换格式
  - metadata: json
  - Model: parquet

MLLIB 2.0：模型持久化

- 动机
  - 扩展ML API，支持更广泛的语言，为数据科学家提供便利
- Python算法
  - Clustering algorithms: Bisecting K-Means, Gaussian Mixtures, LDA
  - Meta-algorithms: OneVsRest, TrainValidationSplit
  - GeneralizedLinearRegression
  - Feature transformers: ChiSqSelector, MaxAbsScaler, QuantileDiscretizer
  - Model inspection: summaries for Logistic Regression, Linear Regression, GLMs
- R算法
  - Regression & classification: Generalized Linear Regression, AFT survival regression
- Clustering: K-Means

MLLIB 2.0：PIPELINE定制化

- 可定制模块
  - Transformer
  - Estimator & Model
  - Evaluator
- 已完成模块
  - 29个feature transformers (Tokenizer, Word2Vec, ...)
  - 21个模型（聚类、分类、回归等）
  - Model调优与评价

AGENDA

- Spark 2.0主要改进与优化
- Spark Core与Spark SQL
- Spark Streaming
- MLlib
- Spark 2.0现状与展望
- Q & A

SPARK 2.0现状与展望

Spark 2.0已经发布
但远不成熟

我的微信公众号

Q & A