Apache Spark 2.0.0正式发布及其更新介绍

原创 2016-07-28 iteblog Hadoop技术博文

Apache Spark 2.0.0正式发布。它是2.x版本线上的第一个版本。主要的更新是API可用性，SQL 2003的支持，性能提升，structured streaming，R中支持UDF以及操作性的提升。此外，本版本一共包括来自300位contributors的超过2500个patches。关于Spark 2.0的相关文章可以参见：Spark 2.0分类目录：http://www.iteblog.com/archives/tag/spark-2-0

本文将列出比较重要的更新。

API Stability

Apache Spark 2.0.0是2.x主线的第一个版本。Spark将保证所有2.x版本的非实验性的API是稳定的。虽然这些API和1.x版本的很类似，但是Spark2.0在API方面还是有很大的改变。这些将在下面的Removals, Behavior Changes and Deprecations章节介绍。

Core and Spark SQL

Programming APIs

Spark 2.0最大的改变之一就是新的API：

1、DataFrame和Dataset统一（可以参见《Spark 2.0技术预览：更容易、更快速、更智能》：http://www.iteblog.com/archives/1668）：在Scala和Java语言中，DataFrame和Dataset得到统一，比如DataFrame仅仅是Dataset[Row]的别名。在Python和R语言中，因为缺乏类型安全特性，DataFrame仍然是主程序的接入点。

2、SparkSession：一个新的切入点。将要替换旧的SQLContext和HiveContext，但是为了向后兼容，

SQLContext 和HiveContext仍然保留着；

3、一个为配置SparkSession新的简化版API；

4、简单以及性能更好的accumulator API；

5、在Dataset中为typed aggregation引入一个新的升级版的Aggregator。

SQL

Spark 2.0大幅提升了SQL功能，并支持SQL2003。Spark SQL现在可以运行所有的99 TPC-DS查询。此外，下面的特性也是比较重要的：

1、支持ANSI-SQL和Hive SQL的内置SQL解析器；

2、内置实现了DDL命令；

3、支持子查询，包括（1）、不相关的标量子查询(Uncorrelated Scalar Subqueries)；（2）、相关的标量子查询(Correlated Scalar Subqueries)；（3）、NOT IN谓词子查询（在WHERE/HAVING语句中）；（4）、IN谓词子查询中（在WHERE/HAVING语句中）；（5）、(NOT) EXISTS谓词子查询中（在WHERE/HAVING语句中）。

4、支持视图规范化。

除此之外，当编译的时候没有加入Hive的支持(也就是没加入-Phive)，Spark SQL将支持几乎所有Hive支持的功能，除了Hive连接，Hive UDF以及脚本转换。

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

New Features

1、内置的CSV数据源，基于Databricks的spark-csv模块（之前版本的Spark这个一直都是作为第三方数据源）；

2、缓存和运行时执行都支持堆外内存管理。

3、支持Hive风格的bucketing；

4、使用sketches进行粗略的总结统计(Approximate summary statistics)，包括approximate quantile,

Bloom filter以及count-min sketch。

Performance and Runtime

1、常见的SQL操作和DataFrame通过一个称为whole stage code generation技术之后有了实质性的性能提升（大约有2-10x）；

2、通过vectorization技术提升了Parquet文件扫描的吞吐量；

3、提升了ORC的性能；

4、在Catalyst query optimizer中为常见的工作流(common workloads)进行了优化；

5、通过内置实现所有的窗口函数来提升Windows的性能(Spark 2.0 Window使用可以参见《Spark 2.0介绍：Spark SQL中的Time Window使用》：http://www.iteblog.com/archives/1705)；

6、为内置的数据源进行自动地文件合并。

MLlib

现在基于DataFrame的API是主要的API了。而基于RDD的API已经进入到维护阶段。详细细节请参考MLlib用户指南。

New features

1、ML persistence: 基于DataFrame的API现在为Scala、Java、Python以及R语言提供了几乎完全的保存和加载ML模型和Pipelines 的支持。(SPARK-6725, SPARK-11939, SPARK-14311)

2、MLlib in R: SparkR now offers MLlib APIs for generalized linear models, naive Bayes, k-means clustering, and survival regression. See this talk to learn more.

3、Python: PySpark now offers many more MLlib algorithms, including LDA, Gaussian Mixture Model, Generalized Linear Regression, and more.

4、Algorithms added to DataFrames-based API: Bisecting K-Means clustering, Gaussian Mixture

Model, MaxAbsScaler feature transformer.

Speed/scaling

存放在DataFrame中的Vectors和Matrices现在使用了比较高效的序列化，这样可以在调用MLlib算法的时候减少开销。(SPARK-14850)

SparkR

The largest improvement to SparkR in Spark 2.0 is user-defined functions. There are three user-defined functions: dapply, gapply, and lapply. The first two can be used to do partition-based UDFs using dapply and gapply, e.g. partitioned model learning. The latter can be used to do hyper-parameter tuning.

In addition, there are a number of new features:

1、Improved algorithm coverage for machine learning in R, including naive Bayes, k-means clustering, and survival regression.

2、Generalized linear models support more families and link functions.

3、Save and load for all ML models.

4、更多的DataFrame功能: Window functions API, reader, writer support for JDBC, CSV, SparkSession

Streaming

Spark 2.0开始引入了实验性的Structured Streaming，它是构建在Spark SQL和Catalyst optimizer之上的高级streaming API。Structured Streaming使得用户可以在流数据的sources和sinks使用静态的数据源一样的DataFrame/Dataset API，并使用Catalyst optimizer自动生成查询计划。

在DStream API方面，最大的更新是支持Kafka 0.10。

Dependency and Packaging Improvements

Spark的操作和打包过程有很多的改变：

1、Spark 2.0在生产部署的时候不再需要fat assembly jar；

2、Akka的依赖已经被全部移除了。所以用户的程序可以引入任何版本的Akka；

3、Kryo的版本升级到3.0；

4、编译时默认使用Scala 2.11而不是2.10。

Removals, Behavior Changes and Deprecations

Removals

下面的特性在Spark 2.0已经被移除了：

1、Bagel

2、支持Hadoop 2.1及其之前版本；

3、配置closure serializer能力；

4、HTTPBroadcast；

5、基于TTL的元数据清理；

6、半私有的类org.apache.spark.Logging，建议直接使用slf4j；

7、SparkContext.metricsSystem；

8、面向块的和Tachyon进行整合；

9、Spark 1.x中所有被标记遗弃的方法；

10、Python语言中所有DataFrame返回RDD的方法（map, flatMap, mapPartitions等等），不过这些方法仍然可以通过dataframe.rdd访问，比如dataframe.rdd.map；

11、不常用的流连接器，包括Twitter, Akka, MQTT, ZeroMQ；

12、Hash-based shuffle manager

13、独立模式的Master历史服务器功能；

14、对Java和Scala语言，DataFrame不再作为一个类存在。所以数据源可能需要升级；For Java and Scala, DataFrame no longer exists as a class. As a result, data sources would need to be updated.

15、Spark EC2脚本已经被完全移到external repository hosted by the UC Berkeley AMPLab。

Behavior Changes

下面的改变可能需要更新现有的应用系统：

1、编译时默认使用Scala 2.11而不是2.10；

2、在SQL中，浮点数字现在解析成decimal数据类型，而不再是double数据类型；

3、Kryo的版本升级到3.0；

4、Java RDD的flatMap和mapPartitions函数之前要求传进来的函数返回Java Iterable，现在需要返回Java iterator，所以这个函数不需要materialize所有的数据；

5、Java RDD的countByKey和countAprroxDistinctByKey函数现在将K类型的数据返回成 java.lang.Long而不是java.lang.Object；

6、当写Parquet文件的时候，默认已经不写summary files了，如果需要开启它，用户必须将parquet.enable.summary-metadata设置为true；

7、基于DataFrame的API(spark.ml)现在取决于spark.ml.linalg中的本地线性代数，而不是spark.mllib.linalg。现在所有的spark.mllib.*都被替换成spark.ml.*了。(SPARK-13944)。

更详细的改变可以参见SPARK-11806。

Deprecations

下面的特性在Spark 2.0已经被标记为遗弃，可能在未来的Spark 2.x版本中被移除：

1、Apache Mesos的细粒度模式；

2、Java 7的支持；

3、Python 2.6的支持。

Known Issues

1、Lead and Lag’ s behaviors have been changed to ignoring nulls from respecting nulls (1.6’ s behaviors). In 2.0.1, the behavioral changes will be fixed in 2.0.1 (SPARK-16721).

2、Lead and Lag functions using constant input values does not return the default value when the offset row does not exist (SPARK-16633).

由于个人技术水平有限，上面翻译如有问题欢迎留言指正。

点击阅读原文，成为ITGeGe首批在线自品牌讲师。

阅读原文