

Apache Spark 3.1.1 版本发布，众多新特性

Apache Spark 3.1.1 版本于美国当地时间2021年3月2日正式发布，这个版本继续保持使得 Spark 更快，更容易和更智能的目标，Spark 3.1 的主要目标如下：

- 提升了 Python 的可用性；
- 加强了 ANSI SQL 兼容性；
- 加强了查询优化；
- Shuffle hash join 性能提升；
- History Server 支持 structured streaming

注意，由于技术上的原因，Apache Spark 没有发布 3.1.0 版本，虽然你可以在 Maven 仓库看到 Apache Spark 3.1.0 版本，但千万别使用。Apache Spark 3.1 版本线的第一个版本是 Apache Spark 3.1.1，它不是一个稳定版。

本文将介绍 Apache Spark 3.1.1 版本的比较重要的特性及改进，限于篇幅仅仅是简单介绍，更加深入的原理可以关注数砖近期发布的博客。另外更加完整的 ISSUE 可以参见 [Apache Spark 3.1.1 Release Note](#)。

ANSI Compliance



Runtime Error



Create Table Syntax



Explicit Cast



Char/Varchar

Execution



Shuffle Hash Join



Node Decommissioning



Partition Pruning



Stage-level Scheduling

Python



Installation Option for PyPi



New Doc for PySpark



Python Type Support



Dependency Management

Streaming



History Server Support for SS



Streaming Table APIs



Stream-stream join



State Schema Validation

Performance



Shuffle Removal



Nested Field Pruning



Sub-expression Elimination



Predicate Pushdown

More



Search Function in Spark Doc



Spark on K8S GA



Catalog APIs for JDBC



Ignore Hints

如果想及时了解Spark、Hadoop或者HBase相关的文章，欢迎关注微信公众号：iteblog_hadoop

Project Zen

在这个版本中，Zen 项目的启动是为了从以下三个方面提高 PySpark 的可用性：

- 更加 Python 化 (Being Pythonic) ；
- PySpark 中更好更容易的可用性；
- 与其他 Python 库更好的互操作性

作为这个项目的一部分，这个版本包括 PySpark 的许多改进——从利用 Python 类型提示到重新设计的 PySpark 文档，比较重要的改进如下。

支持 Python 类型 (Python typing support)

PySpark 中的 Python 类型支持最初是作为第三方库 `pyspark-stubs` 来创建的，现在已经成为一个成熟和稳定的库。在这个版本中，PySpark 正式包含了带有许多功能 (参见 [SPARK-32681](#)) 的 Python 类型提示 (Python type hints)。Python 类型提示在 IDE 和 notebooks 中最有用，它使用户和开发人员能够利用无缝的自动完成功能，包括最近在 Databricks notebooks 中添加的自动完成支持。此外，IDE 开发人员可以通过 Python 类型提示中的静态类型和错误检测来提高生产力。

支持依赖关系管理

PySpark 中的依赖管理支持已经完成，对应的文档也添加完成，以帮助 PySpark 用户和开发人员，参见 [SPARK-33824](#)。以前，PySpark 对依赖管理的支持是不完整的，它只在 YARN 中有效，并且没有相关的操作文档。在这个版本中，通过利用 `-archive` 选项 (参见 [SPARK-33530](#), [SPARK-33615](#))，Conda、virtualenv 和 PEX 等包管理系统可以在任何类型的集群中工作。关于这个可以看下数砖的这篇文章 [How to Manage Python Dependencies in PySpark](#) 以及对应的文档。

为 PyPI 用户提供新的安装选项

这个版本为 PyPI 用户提供了新的安装选项 (参见 [SPARK-32017](#))。pip 是安装 PySpark 最常用的方法之一。然而，上一个版本只允许在 PyPI 中使用 Hadoop 2，但允许使用 Apache Spark 的其他发行渠道中的其他选项，如 Hadoop 2 和 3。在这个版本中，作为 Project Zen 的一部分，PyPI 用户也可以使用所有选项。这使他们能够从 PyPI 安装并在现有的任何类型的 Spark 集群中运行他们的应用程序。

PySpark 相关的文档完善

这个版本 ([SPARK-31851](#)) 中引入了 PySpark 的新文档。之前 PySpark 的文档很难导航，只包含 API 引用。在这个版本中，文档被完全重新设计，具有细粒度的分类和易于导航的层次结构 (参见 [SPARK-32188](#))。docstrings 具有 numpdoc 风格的更好的人类可读的文本格式 ([SPARK-32085](#))，并且有许多有用的页面，如如何调试 ([SPARK-32186](#))、如何贡献和测试 ([SPARK-32190](#)、[SPARK-31851](#)) 和使用 live notebook 的快速入门 ([SPARK-32182](#))。

ANSI SQL 兼容性

这个版本为 ANSI SQL 的兼容性增加了额外的改进，这有助于简化从传统数据仓库系统到 Spark 的工作负载的迁移，主要如下。

自 Spark 3.0 发布以来，ANSI 方言模式已经被引入并得到了增强。在 ANSI 模式中，如果不是严格地遵守 ANSI SQL 行为，那么 Spark 会把它弄成与 ANSI SQL 风格是一致的。在这个版本中，当输入无效时（SPARK-33275），更多的操作符/函数抛出运行时错误，而不是返回 NULL。对显式类型转换进行更严格的检查也是这次发布的一部分。当查询包含非法的类型转换时（例如，日期/时间戳类型被转换为数字类型），就会抛出编译时错误，通知用户这是无效的转换。ANSI 方言模式仍然处于活跃的开发中，因此它在默认情况下是禁用的，但可以通过设置 `spark.sql.ansi=true` 来启用，我们希望它在即将发布的版本中保持稳定。

这个版本中添加了各种新的 SQL 特性。添加了广泛使用的标准 CHAR/VARCHAR 数据类型，这个数据类型是作为变体 String 类型。增加了更多的内置函数（例如 `width_bucket`（SPARK-21117）和 `regexp_extract_all`（SPARK-24884））。目前内置操作符/函数的数量已经达到 350 个。更多的 DDL/DML/utility 命令得到了增强，包括 INSERT（SPARK-32976）、MERGE（SPARK-32030）和 EXPLAIN（SPARK-32337）。从这个版本开始，在 Spark WebUI 中，SQL 计划将以一种更简单、更结构化的格式呈现，比如使用 EXPLAIN FORMATTED 展示。

统一 CREATE TABLE SQL 语法已经在这次发布中完成。目前 Spark 维护两组 CREATE TABLE 语法。当语句中不包含 USING 和 STORED AS 子句时，Spark 使用默认的 Hive 文件格式。当 `spark.sql.legacy.createHiveTableByDefault` 被设置为 false（Spark 3.1 版默认为 true，Databricks Runtime 8.0 版默认为 false），默认的表格式依赖于 `spark.sql.sources.default` 的设置（Spark 3.1 版默认为 parquet，Databricks Runtime 8.0 版默认为 delta）。这意味着在 Databricks Runtime 8.0 中 Delta Lake 现在是默认格式，将提供更好的性能和可靠性。下面是一个演示了当用户没有显式指定 USING 或 STORED AS 子句时使用 CREATE TABLE SQL 例子。

```
CREATE TABLE table1 (col1 int);
```

```
CREATE TABLE table2 (col1 int) PARTITIONED BY (partCol int);
```

下表展示了上面两个语句在不同环境的变化：

	Spark 3.0 (DBR 7) or before	Spark 3.1 *	DBR 8.0
Default Format	Hive Text Serde	Parquet	Delta

注意，我们需要显示的将 `spark.sql.legacy.createHiveTableByDefault` 设置为 false，否则 Apache Spark 将使用 Hive Text Serde。

性能提升

Catalyst 是对大多数 Spark 应用程序进行优化的查询编译器。在 Databricks，每天有数十亿个查询被优化和执行，这个版本增强了查询优化并加速了查询处理。

Predicate pushdown

谓词下推是最有效的性能特性之一，因为它可以显著减少扫描和处理的数据量。Spark 3.1 中完成了各种增强：

- JSON 和 Avro 数据源（参见 SPARK-32346）支持谓词下推，ORC 数据源支持嵌套字段的谓词下推。
- Filters 也可以通过 EXPAND 算子进行下推（参见 SPARK-33302）。

其他改进可以参见 SPARK-32858 和 SPARK-24994。

Shuffle 消除，子表达式消除和嵌套字段修剪

Shuffle 消除（Shuffle removal），子表达式消除（subexpression elimination）和嵌套字段修剪（nested field pruning）是另外三个主要的优化特性。Shuffle 是最昂贵的操作之一，在某些情况下可以避免 Shuffle（参见 SPARK-31869、SPARK-32282、SPARK-33399），但在消除 Shuffle 后，自适应查询规划可能不适用。此外，可以删除重复或不必要的表达式求值（参见 SPARK-33092、SPARK-33337、SPARK-33427、SPARK-33540）以减少计算量。列修剪可以应用于各种操作符（参见 SPARK-29721、SPARK-27217、SPARK-31736、SPARK-32163、SPARK-32059）中的嵌套字段，以减少 I/O 资源的使用，便于后续的优化。

Shuffle-Hash Join (SHJ) 支持所有的 join 类型

从这个版本开始 Shuffle-Hash Join (SHJ) 支持所有的 join 类型（SPARK-32399），同时支持相应的 codegen execution（SPARK-32421）。与 Shuffle-Sort-Merge Join (SMJ) 不同的是，SHJ 不需要排序，因此当 join 一个大表和一个小时表时，SHJ 的 CPU 和 IO 效率比 SMJ 更高。注意，当构建端（build side）很大时，SHJ 可能会导致 OOM，因为构建 hashmap 是内存密集型的。

Streaming 的改进

Spark 是构建分布式流处理应用程序的最佳平台。Databricks 每天有超过10万亿的 records 通过 structured streaming 处理。这个版本增强了 Structured Streaming 的监控、可用性和功能。

为了更好地调试和监控 Structured Streaming 应用程序，添加了历史服务器（History Server）支持（参见 SPARK-31953）。在 Live UI 中，添加了更多的 metrics（SPARK-33223）、水印间隔（watermark gap）（参见 SPARK-33224）和更多的状态自定义度量（state custom metrics）（参见 SPARK-33287）。

添加了新的 Streaming table APIs，用于读取和写 streaming DataFrame 到表中，就像 DataFrameReader 和 DataFrameWriter 中的 table API 一样。在 Databricks Runtime

中，推荐使用 Delta table 表格式，以实现精确一次（exactly-once）的语义和更好的性能。

Stream-stream Join 增加了两种新的 join 类型，这个版本中包括了 full outer (SPARK-32862) 和 left semi (SPARK-32863) join。在 Apache Spark 3.1 之前，已经支持了 inner, left outer 以及 right outer stream-stream joins。

其他 Spark 3.1 的改进

除了这些新特性，这个版本还关注可用性、稳定性；改进和解决了大约1500个问题。总共超过200位贡献者，包括个人和公司，如

Databricks，谷歌，苹果，Linkedin，微软，英特尔，IBM，阿里巴巴，Facebook, Nvidia, Netflix, Adobe 等。在这篇博文中，我们重点介绍了 Spark 中一些关键的 SQL、Python 和 streaming 改进，限于篇幅 Apache Spark 3.1 中还有许多其他功能这里并没有涉及，比如 Spark on Kubernetes GAed, node decommissioning, state schema validation 等，可以到 [Apache Spark 3.1.1 Release Notes](#) 和 Spark 的官方文档查找。

本文主要翻译自：[Introducing Apache Spark™ 3.1](#)

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: 【】（）