Apache Spark 1.5.0正式发布

Spark 1.5.0是1.x线上的第6个发行版。这个版本共处理了来自230+contributors和80+机构的1400+个patches。Spark 1.5的许多改变都是围绕在提升Spark的性能、可用性以及操作稳定性。Spark 1.5.0焦点在Tungsten项目，它主要是通过对低层次的组建进行优化从而提升Spark的性能。Spark 1.5版本为Streaming增加了operational特性，比如支持backpressure。另外比较重要的更新就是新增加了一些机器学习算法和工具，并扩展了Spark R的相关API。主要做了以下修改：

APIs: RDD, DataFrame和SQL

1、Consistent resolution of column names (see Behavior Changes section)
2、SPARK-3947: New experimental user-defined aggregate function (UDAF) interface
3、SPARK-8300: DataFrame hint for broadcast joins
4、SPARK-8668: expr function for turning a SQL expression into a DataFrame column
5、SPARK-9076: Improved support for NaN values
 5.1、NaN functions: isnan, nanvl
 5.2、dropna/fillna also fill/drop NaN values in addition to NULL values
 5.3、Equality test on NaN = NaN returns true
 5.4、NaN is greater than all other values
 5.5、In aggregation, NaN values go into one group
6、SPARK-8828: Sum function returns null when all input values are nulls
7、Data types
 7.1、SPARK-8943: CalendarIntervalType for time intervals
 7.2、SPARK-7937: Support ordering on StructType
 7.3、SPARK-8866: TimestampType’s precision is reduced to 1 microseconds (1us)
8、SPARK-8159: Added ~100 functions, including date/time, string, math.
9、SPARK-8947: Improved type coercion and error reporting in plan analysis phase (i.e. most errors should be reported in analysis time, rather than execution time)
10、SPARK-1855: Memory and local disk only checkpointing support

Backend Execution: DataFrame and SQL

1、对大多数的DataFrame/SQL函数来说，Code generation默认情况下是开启的。
2、DataFrame/SQL中的aggregation execution有所提升。
 2.1、Cache friendly in-memory hash map layout
 2.2、Fallback to external-sort-based aggregation when memory is exhausted
 2.3、aggregations操作中默认开启Code generation
3、对 DataFrame/SQL中的Join执行有所提升
 3.1、Prefer (external) sort-merge join over hash join in shuffle joins (for left/right outer and inner joins), i.e. join 3.2、data size is now bounded by disk rather than memory
 3.3、Support using (external) sort-merge join method for left/right outer joins

3.4、Support for broadcast outer join
4、对DataFrame/SQL中的排序引擎有所提升
4.1、Cache-friendly in-memory layout for sorting
4.2、Fallback to external sorting when data exceeds memory size
4.3、Code generated comparator for fast comparisons
5、Native memory management & representation
5.1、Compact binary in-memory data representation, leading to lower memory usage
5.2、Execution memory is explicitly accounted for, without relying on JVM GC, leading to less GC and more robust memory management
6、SPARK-8638: Improved performance & memory usage in window functions
7、Metrics instrumentation, reporting, and visualization
7.1、SPARK-8856: Plan visualization for DataFrame/SQL
7.2、SPARK-8735: Expose metrics for runtime memory usage in web UI
7.3、SPARK-4598: Pagination for jobs with large number of tasks in web UI

Integrations: Data Sources, Hive, Hadoop, Mesos and Cluster Management

1、Mesos
1.1、SPARK-6284: Support framework authentication and Mesos roles
1.2、SPARK-6287: Dynamic allocation in Mesos coarse-grained mode
1.3、SPARK-6707: User specified constraints on Mesos slave attributes
2、YARN
2.1、SPARK-4352: Dynamic allocation in YARN works with preferred locations
3、Standalone Cluster Manager
3.1、SPARK-4751: Dynamic resource allocation support
4、SPARK-6906: Improved Hive and metastore support
4.1、SPARK-8131: Improved Hive database support
4.2、Upgraded Hive dependency Hive 1.2
4.3、Support connecting to Hive 0.13, 0.14, 1.0/0.14.1, 1.1, 1.2 metastore
4.4、Support partition pruning pushdown into the metastore (off by default; config flag spark.sql.hive.metastorePartitionPruning)
4.5、Support persisting data in Hive compatible format in metastore
5、SPARK-9381: Support data partitioning for JSON data sources
6、SPARK-5463: Parquet improvements
6.1、将Parquet升级到 1.7
6.2、Speedup metadata discovery and schema merging
6.3、Predicate pushdown on by default
6.4、SPARK-6774: Support for reading non-standard legacy Parquet files generated by various libraries/systems by fully implementing all backwards-compatibility rules defined in parquet-format spec
6.5、SPARK-4176: Support for writing decimal values with precision greater than 18
7、ORC improvements (various bug fixes)
8、SPARK-8890: Faster and more robust dynamic partition insert

9、SPARK-9486: DataSourceRegister interface for external data sources to specify short names

R语言

1、SPARK-6797: Support for YARN cluster mode in R
2、SPARK-6805: GLMs with R formula, binomial/Gaussian families, and elastic-net regularization
3、SPARK-8742: Improved error messages for R
4、SPARK-9315: Aliases to make DataFrame functions more R-like

Machine Learning and Advanced Analytics

1、SPARK-8521: New Feature transformers: CountVectorizer, Discrete Cosine transformation, MinMaxScaler, NGram, PCA, RFormula, StopWordsRemover, and VectorSlicer.
2、New Estimators in Pipeline API: SPARK-8600 naive Bayes, SPARK-7879 k-means, and SPARK-8671 isotonic regression.
3、New Algorithms: SPARK-9471 multilayer perceptron classifier, SPARK-6487 PrefixSpan for sequential pattern mining, SPARK-8559 association rule generation, SPARK-8598 1-sample Kolmogorov-Smirnov test, etc.
4、提升现有的算法
4.1、LDA: online LDA performance, asymmetric doc concentration, perplexity, log-likelihood, top topics/documents, save/load, etc.
4.2、Trees and ensembles: class probabilities, feature importance for random forests, thresholds for classification, checkpointing for GBTs, etc.
4.3、Pregel-API: more efficient Pregel API implementation for GraphX.
4.4、GMM: distribute matrix inversions.
5、Model summary for linear and logistic regression.
6、Python API: distributed matrices, streaming k-means and linear models, LDA, power iteration clustering, etc.
7、Tuning and evaluation: train-validation split and multiclass classification evaluator.
8、Documentation: document the release version of public API methods

Spark Streaming

1、SPARK-7398: Backpressure: Automatic and dynamic rate controlling in Spark Streaming for handling bursty input streams. This allows a streaming pipeline to dynamically adapt to changes in ingestion rates and computation loads. This works with receivers, as well as, the Direct Kafka approach.
2、Python API for streaming sources
2.1、SPARK-8389: Kafka offsets of Direct Kafka streams available through Python API
2.2、SPARK-8564: Kinesis Python API
2.3、SPARK-8378: Flume Python API
2.4、SPARK-5155: MQTT Python API
3、SPARK-3258: Python API for streaming machine learning algorithms: K-Means, linear

regression, and logistic regression

4、SPARK-9215: Improved reliability of Kinesis streams : No need for enabling write ahead logs for saving and recovering received data across driver failures

5、Direct Kafka API graduated: Not experimental any more.

6、SPARK-8701: Input metadata in UI: Kafka offsets, and input files are visible in the batch details UI

7、SPARK-8882: Better load balancing and scheduling of receivers across cluster

8、SPARK-4072: Include streaming storage in web UI

遗弃、移出、Configs和行为变化

Spark Core

1、DAGScheduler中的local task execution模块被移出

2、Driver和executor的默认内存从512m升到1G

3、JVM中MaxPermSize的默认配置从128m升到256m

4、spark-shell的默认日志级别从INFO提升到WARN

5、基于NIO的ConnectionManager已经遗弃，并且将在1.6版本中移出。

Spark SQL & DataFrames

1、Optimized execution using manually managed memory (Tungsten) is now enabled by default, along with code generation for expression evaluation. These features can both be disabled by setting spark.sql.tungsten.enabled to false.

2、Parquet schema merging is no longer enabled by default. It can be re-enabled by setting spark.sql.parquet.mergeSchema to true.

3、Resolution of strings to columns in Python now supports using dots (.) to qualify the column or access nested values. For example df['table.column.nestedField']. However, this means that if your column name contains any dots you must now escape them using backticks (e.g., table.`column.with.dots`.nested).

4、In-memory columnar storage partition pruning is on by default. It can be disabled by setting spark.sql.inMemoryColumnarStorage.partitionPruning to false.

5、Unlimited precision decimal columns are no longer supported, instead Spark SQL enforces a maximum precision of 38. When inferring schema from BigDecimal objects, a precision of (38, 18) is now used. When no precision is specified in DDL then the default remains Decimal(10, 0).

6、Timestamps are now processed at a precision of 1us, rather than 100ns.

7、Sum function returns null when all input values are nulls (null before 1.4, 0 in 1.4).

8、In the sql dialect, floating point numbers are now parsed as decimal. HiveQL parsing remains unchanged.

9、The canonical name of SQL/DataFrame functions are now lower case (e.g. sum vs SUM).

10、It has been determined that using the DirectOutputCommitter when speculation is enabled is unsafe and thus this output committer will not be used by parquet when speculation is on, independent of configuration.

11、JSON data source will not automatically load new files that are created by other applications (i.e. files that are not inserted to the dataset through Spark SQL). For a JSON persistent table (i.e. the metadata of the table is stored in Hive Metastore), users can use REFRESH TABLE SQL command or HiveContext’s refreshTable method to include those new files to the table. For a DataFrame representing a JSON dataset, users need to recreate the DataFrame and the new DataFrame will include new files.

Spark Streaming

1、新的实验性backpressure特性可以通过将spark.streaming.backpressure.enabled设置为true来开启。
2、Write Ahead Log does not need to be abled for Kinesis streams. The updated Kinesis receiver keeps track of Kinesis sequence numbers received in each batch, and uses that information re-read the necessary data while recovering from failures.
3、The number of times the receivers are relaunched on failure are not limited by the max Spark task attempts. The system will always try to relaunch receivers after failures until the StreamingContext is stopped.
4、Improved load balancing of receivers across the executors, even after relaunching.
5、Enabling checkpointing when using queueStream throws exception as queueStream cannot be checkpointed. However, we found this to break certain existing apps. So this change will be reverted in Spark 1.5.1.

MLlib

1、In the spark.mllib package, there are no breaking API changes but some behavior changes:
1.1、SPARK-9005: RegressionMetrics.explainedVariance returns the average regression sum of squares.
1.2、SPARK-8600: NaiveBayesModel.labels become sorted.
1.3、SPARK-3382: GradientDescent has a default convergence tolerance 1e-3, and hence iterations might end earlier than 1.4.
2、In the experimental spark.ml package, there exists one breaking API change and one behavior change:

2.1、SPARK-9268: Java’s varargs support is removed from Params.setDefault due to a Scala compiler bug.
2.2、SPARK-10097: Evaluator.isLargerBetter is added to indicate metric ordering. Metrics like RMSE no longer flip signs as in 1.4.