It’s a shared variable that is used with RDD and DataFrame (distributed data) to perform sum and counter operations like Map\_reduce counters. These variables are shared by all executors to update and add information through aggregation or computative operations. They’re basically shared variables that can be updated by executors and propagate back to driver program for result collections.

The following example comes from the “Accumulators” reference

Broadcast variables are read-only shared variables that are cached and available on all nodes in a cluster in-order to access or use by the tasks. Instead of sending this data along with every task, PySpark distributes broadcast variables to the workers using efficient broadcast algorithms to reduce communication costs. Broadcast variables are used in the same way for RDD and DataFrames and when we run these applications that have the broadcast variables defined and used, PySpark does the following:

* PySpark breaks the job into stages that have distributed shuffling and actions are executed with in the stage.
* Later Stages are also broken into tasks.
* Spark broadcasts the common data (reusable) needed by tasks within each stage.
* The broadcasted data is cache in serialized format and deserialized before executing each task.

Pair RDDs are a specific type of data structure in Spark, organized as key-value pairs. They key is the identifier while the value is known as data. This type of operation is used to perform transformations like reducing, sorting, grouping as well as actions like counting by key. It seems the functionality is similar to Map reduce style operations.