|  |  |  |
| --- | --- | --- |
| **LsSinkRDD**: Writer RDD Requirement | **Satisfied?** | **Comments** |
| Accepts data from a parent RDD | Y | Tested with LsSourceRDD |
| Allows various input data types from the parent RDD | Y | Theoretically yes – only tested with strings |
| Allows various output data types | Y | Theoretically yes – only tested with strings |
| Performs computations lazily | Y | “new LsSinkRDD” performs no operations. Waits until “saveToTextFile” invoked |
| Overrides RDD.saveAsTextFile() and performs operations within that method | Y |  |
| Provides proper info and error messages | Y | TODO: more messages will be added to cover important cases |
| Creates correct output partitioning structure in the target datasource (local file system) | Y | Invokes filesystem.mkdirs |
| Translates parent partitioning structure into correct output partitioning (via custom partitioner) | Y |  |
| Allows explicit selection of output partitions at a row-level basis – via the row key. When not specified gracefully handles via default HashPartitioner | Y | Custom partitioner is used for general case. First token in each input (and output) record must be partition path. |
| Reuses parent SparkContext | Y | Reuses parent SparkContext |
| Cleans up after itself | Y |  |
| Efficient with resource utilization | N | TODO: Local ThreadPool not yet implemented. ETA: mid June |
| Supports saving data to remote partitions | N | TODO: support remote Writes via an scp mechanism. ETA: TBD |
| Support getPreferredLocations() for host and rack level configurations | Host: Y  Rack: N | TODO: Partially implemented via rackToHost(). Still need to support rack\_to\_host.yml config file |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **LsSourceRDD**: Reader RDD Requirement | **Satisfied?** | **Comments** |
| Accepts an existing set of directories with data | Y | User provides list of host:rack:directory paths |
| Allows various input data types from the source directories | Y | Theoretically yes – only tested with strings |
| Allows various output data types | Y | Theoretically yes – only tested with strings |
| Performs computations lazily | Y | “new LsSourceRDD” performs no operations. Waits until “compute()” invoked by Spark DAG Scheduler |
| Translates parent partitioning structure into correct output partitioning (via custom partitioner) | Y | Partitioning performed according to |
| Overrides RDD.saveAsTextFile() and performs operations within that method | Y |  |
| Provides proper info and error messages | Y | TODO: more messages will be added to cover important cases |
| Translates on-disk partitioning structure into correct output partitioning | Y |  |
|  |  |  |
| Cleans up after itself | Y |  |
| Efficient with resource utilization | Y | Reads are optionally parallelized via local threadpool. TODO: provide config hooks to determine threadpool size |
| Supports reading data to remote partitions | N | TODO: support remote Reads via an scp mechanism. ETA: TBD |
| Support getPreferredLocations() for host and rack level configurations | Host: Y  Rack: N | TODO: Partially implemented via rackToHost(). Still need to support rack\_to\_host.yml config file |

|  |  |  |
| --- | --- | --- |
| **P2pRDD**: Transformer RDD Requirement | **Satisfied?** | **Comments** |
| Works as one stage within the standard RDD transformation workflows | Y | Use “new P2pRDD(parentRdd)” and then any other standard RDD operations may be invoked on the result. |
| Allows various input data types from the parent RDD | Y | Theoretically yes – only tested with strings |
| Allows various output data types | Y | Theoretically yes – only tested with strings |
| Performs computations lazily | Y | “new LsP2pRDD” performs no operations. Waits until “compute()” invoked by the Spark DAG scheduler |
| Provides proper info and error messages | Y | TODO: more messages will be added to cover important cases |
| Retains parent partitioning scheme | Y |  |
| Reuses parent SparkContext | Y | Reuses parent SparkContext |
| Cleans up after itself | Y |  |
| Efficient with resource utilization | Y | Reuses native Libs via the local Linear Algebra support of Breeze. TODO: Document algorithmic resource requirements |
| Support getPreferredLocations() for host and rack level configurations | Host: Y  Rack: N | TODO: Partially implemented via rackToHost(). Still need to support rack\_to\_host.yml config file |