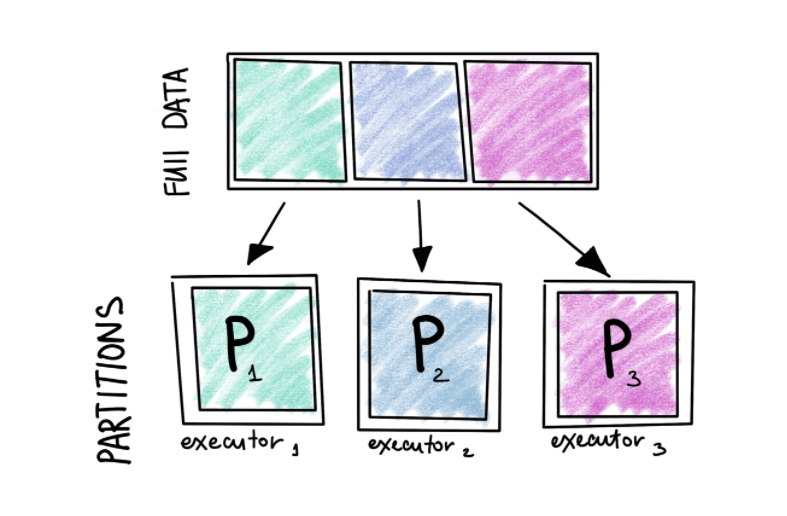
1. Relations between executors and partitions

Ans: As we know that spark engine do parallel processing on a cluster.

It will distribute the work across the cluster, divide the task into the smaller parts, and reduce memory requirements for each node. **Partition** is the main unit of parallelism in Apache Spark. And one the user has submitted his job into the cluster for further processing each partition is sent to a specific **executor**. And it should be noted that only one partition is processed by one executor at a time, so the size and number of partitions transferred to the executor are directly proportional to the time it takes to complete.



1. A typical **data flow sequence for programming Spark** is as follows: Define one or more RDDs, either through accessing **data** stored on disk (e.g., HDFS, Cassandra, HBase, or S3), parallelizing some collection, transforming an existing RDD, or by caching.

In **Spark**, there are supported two **memory** management modes: Static **Memory** Manager and Unified **Memory** Manager.

**Spark** provides a unified interface MemoryManager for the management of Storage **memory** and Execution **memory**.

The tasks in the same Executor call the interface to apply for or release **memory**.