**MapReduce: Simplified Data Processing on Large Clusters**

MapReduce is a framework used to process large data sets through 2 tasks- Map and Reduce. A good example to understand the process is making sandwiches. Suppose we want to make 3 sandwiches- ham, turkey, and Italian. A list of the ingredients needed is gathered in Map phase. The total amount of each ingredient is determined during the Shuffle and Reduce phase. Generally, the input data in Map is stored in Hadoop File System (HDFS), and the output set after the Reducer process is stored in HDFS. Typically, thousands of machines process terabytes of data with MapReduce. It manages the data-processing details such as issuing tasks, verifying task completion, and copying between nodes and around clusters. The parallelization of the Map tasks in the large distribution ensures that every record (key value pair) is processed only once. The partitioner ensures similarly that every record is passed to only one Reducer. The Shuffle combines intermediate keys and values from Map tasks and sends to targeted Reduce tasks for further processing, which outputs zero or more key value pairs. In a multistage computational workflow, the Reduce output may be input to another Map phase, but the output is final in individual MapReduce application. Since it helps process very large data using thousands of machines, it must tolerate machine failures. In Hadoop, the master process will reschedule after a Map task failure, to a machine that contains the replica of the input data; a task can be rescheduled up to 4 times until it is deemed to have failed. As cleaning in coding is troublesome, debugging problems in MapReduce is also troublesome for the distributed system. Combiner is an advantageous function after Map but before Shuffle phase, since it will reduce the amount of data in Shuffle and the computational load in Reduce phase. It can be used in commutative and associative reduce operations like sum or count (e.g. summing a list of sums), but averaging a list of averages will give an erroneous result. In Hadoop, MapReduce is the core component for processing.

**Terminology**

* **PayLoad** - Applications implement the Map and the Reduce functions, and form the core of the job.
* **Mapper** - Mapper maps the input key/value pairs to a set of intermediate key/value pair.
* **NamedNode** - Node that manages the Hadoop Distributed File System (HDFS).
* **DataNode** - Node where data is presented in advance before any processing takes place.
* **MasterNode** - Node where JobTracker runs and which accepts job requests from clients.
* **SlaveNode** - Node where Map and Reduce program runs.
* **JobTracker** - Schedules jobs and tracks the assign jobs to Task tracker.
* **Task Tracker** - Tracks the task and reports status to JobTracker.
* **Job** - A program is an execution of a Mapper and Reducer across a dataset.
* **Task** - An execution of a Mapper or a Reducer on a slice of data.
* **Task Attempt** - A particular instance of an attempt to execute a task on a SlaveNode.