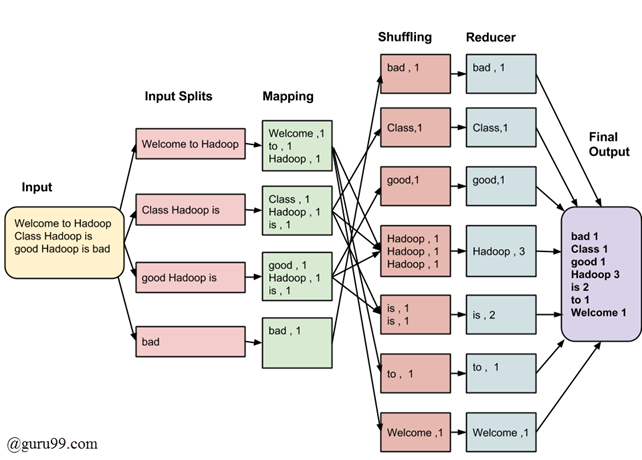
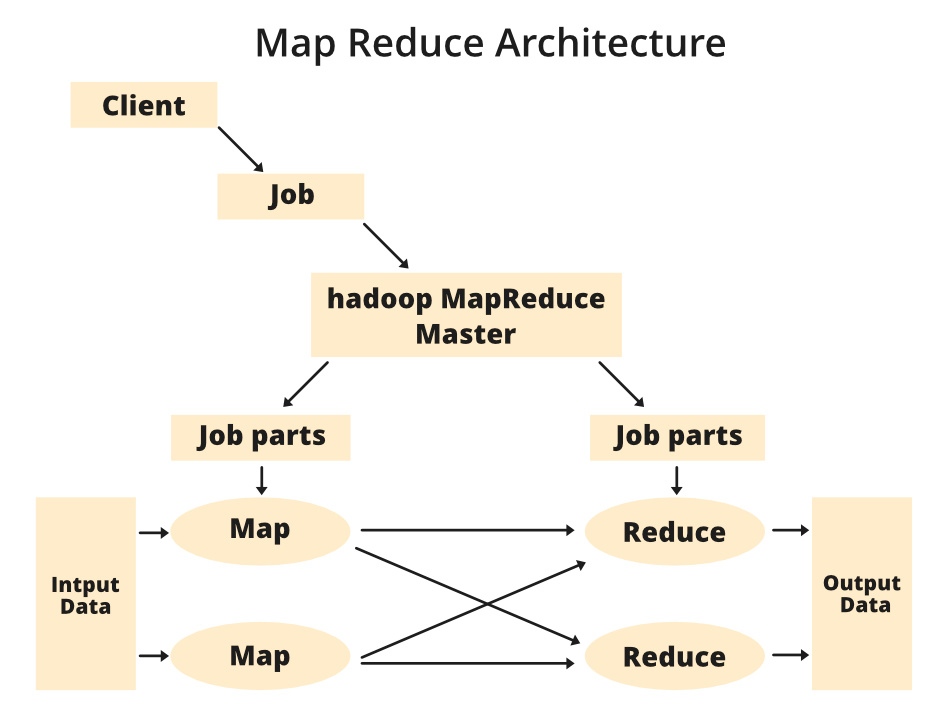
**MapReduce**

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.





There are two primary tasks in MapReduce:

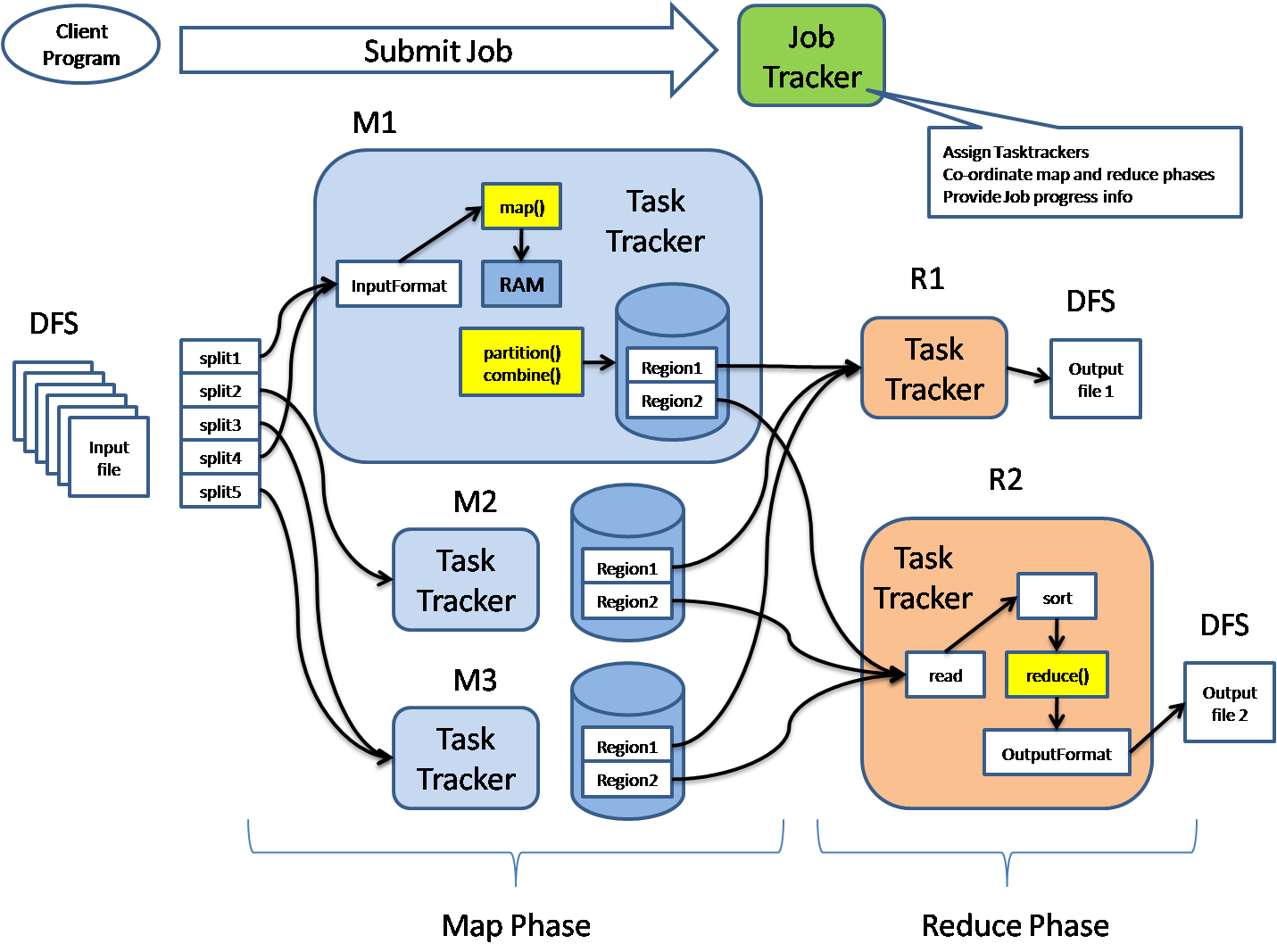
**map and reduce.**

We perform the former task before the latter. In the map job, we split the input dataset into *chunks*. Map task processes these chunks in parallell. The *map* we use outputs as inputs for the *reduce* tasks.

Reducers process the intermediate data from the maps into smaller tuples, that reduces the tasks, leading to the final output of the framework.

The MapReduce framework enhances the scheduling and monitoring of tasks. The failed tasks are re-executed by the framework.

MapReduce can be implemented using various programming languages such as Java, Hive, Pig, Scala, and Python.



**Understanding how MapReduce works**

* **Job: This is the actual work that needs to be executed or processed**
* **Task: This is a piece of the actual work that needs to be executed or processed. A MapReduce job comprises many small tasks that need to be executed.**
* **Job Tracker: This tracker plays the role of scheduling jobs and tracking all jobs assigned to the task tracker.**
* **Task Tracker: This tracker plays the role of tracking tasks and reporting the status of tasks to the job tracker.**
* **Input data: This is the data used to process in the mapping phase.**
* **Output data: This is the result of mapping and reducing.**
* **Client: This is a program or Application Programming Interface (API) that submits jobs to the MapReduce. MapReduce can accept jobs from many clients.**
* **Hadoop MapReduce Master: This plays the role of dividing jobs into job-parts.**
* **Job-parts: These are sub-jobs that result from the division of the main job.**

****

### Benefits of Hadoop MapReduce

* **Speed: MapReduce can process huge unstructured data in a short time.**
* **Fault-tolerance: The MapReduce framework can handle failures.**
* **Cost-effective: Hadoop has a scale-out feature that enables users to process or store data in a cost-effective manner.**
* **Scalability: Hadoop provides a highly scalable framework. MapReduce allows users to run applications from many nodes.**
* **Data availability: Replicas of data are sent to various nodes within the network. This ensures copies of the data are available in the event of failure.**
* **Parallel Processing: In MapReduce, multiple job-parts of the same dataset can be processed in a parallel manner. This reduces the time taken to complete a task.**