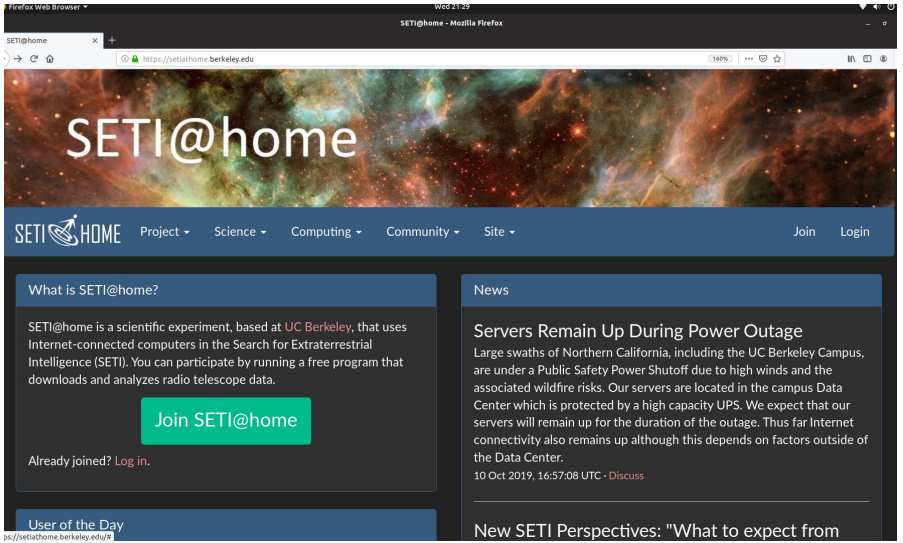
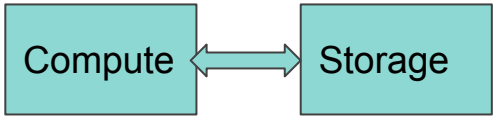
Hadoop & MapReduce

Ms.N.Ravitha Rajalakshmi

Distributed Computing

- Multiple machines communicate and coordinate with each other for accomplishing a task

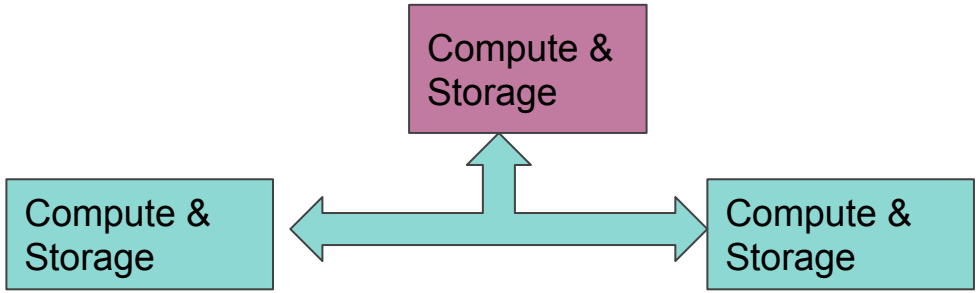




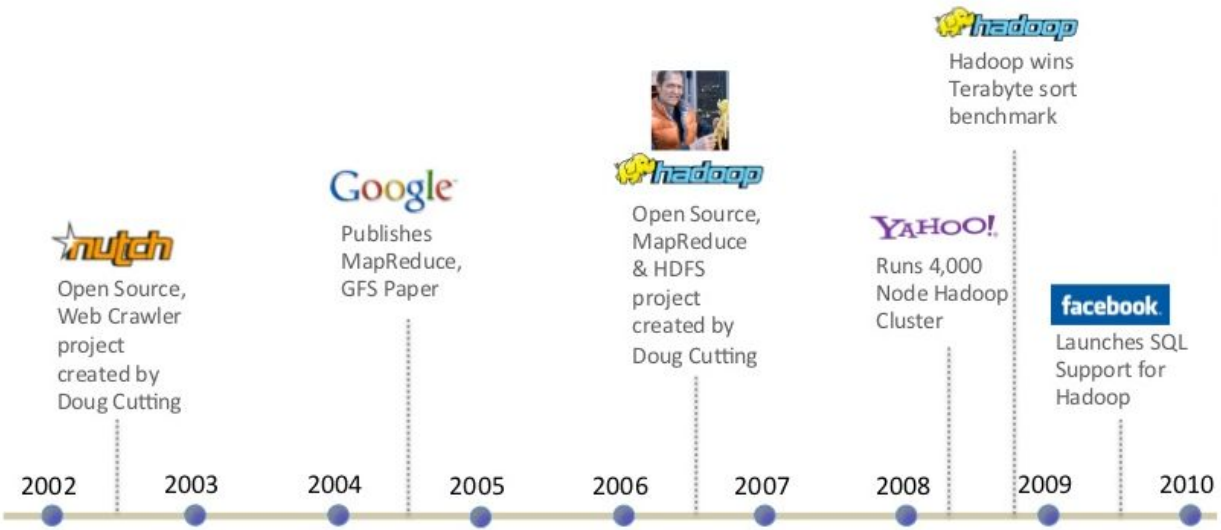
HPC Scenario

Hadoop

- Framework for **distributed processing of large datasets** across **clusters of computers** using **simple programming models**
- **Move Compute to Data**



Hadoop Evolution



Practical Use cases

Uses Hadoop and HBase for :
- Social services
- Structured data storage
- Processing for internal use

Uses Hadoop for :
- Amazon's product search indices They process millions of sessions daily for analytics.

Uses Hadoop for :
- Search optimization
- Research

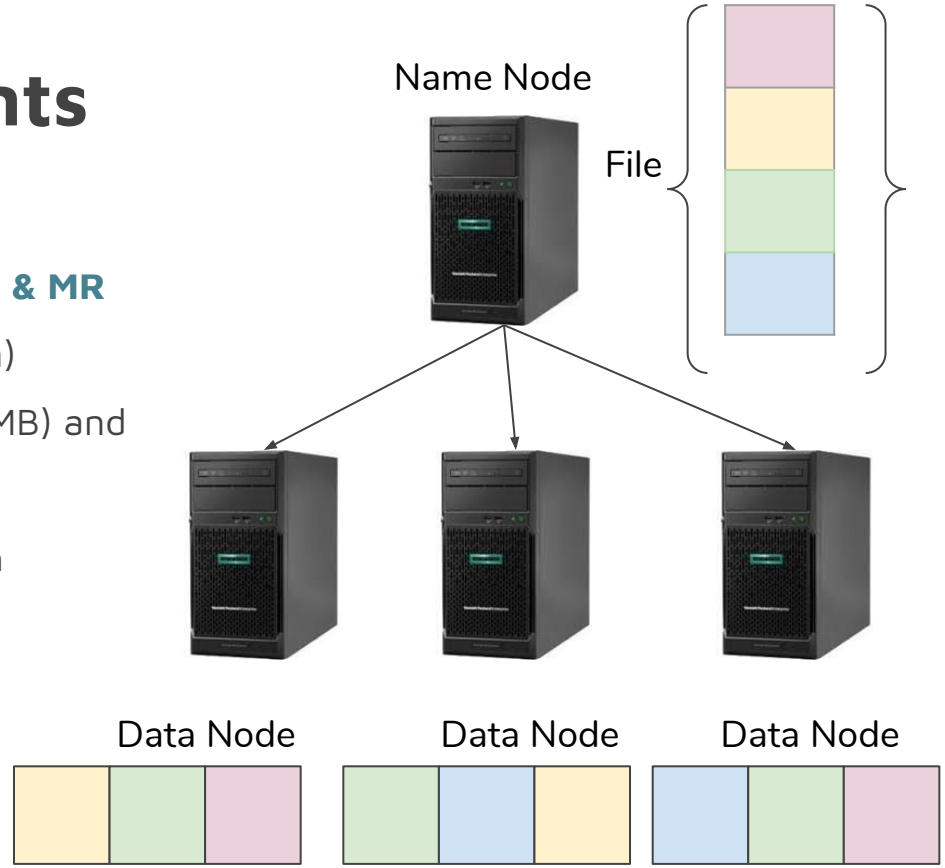
Uses Hadoop :
- As a source for reporting/analytics and machine learning.

Uses Hadoop for :
- Databasing and analyzing Next Generation Sequencing (NGS) data produced for the Cancer Genome Atlas (TCGA) project and other groups

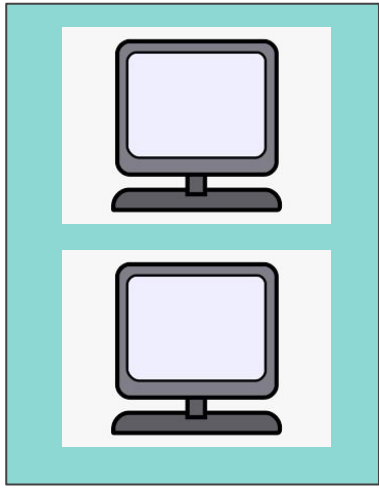
And Many More ....

Hadoop Components

- Hadoop has **two major components: HDFS & MR**
  - **HDFS** (Hadoop Distributed File System)
    - Divides the files into blocks (64 MB) and distributes across the cluster
    - Provides Fault Tolerance through replication



HDFS

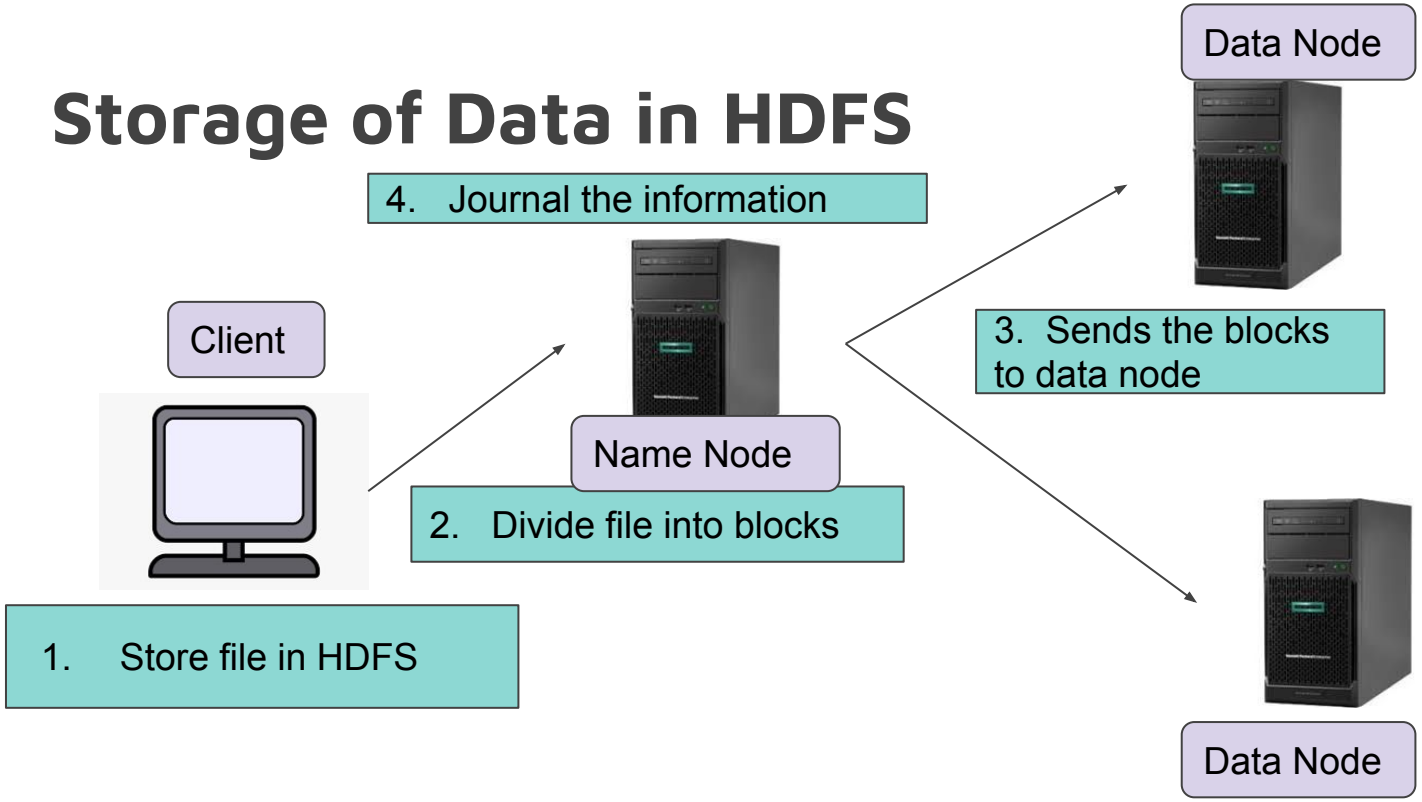


Terminologies:

Name Node: Master node for HDFS
Responsible for file system namespace
File name, Block Information, read/write information, list of replicas
Manage block replication

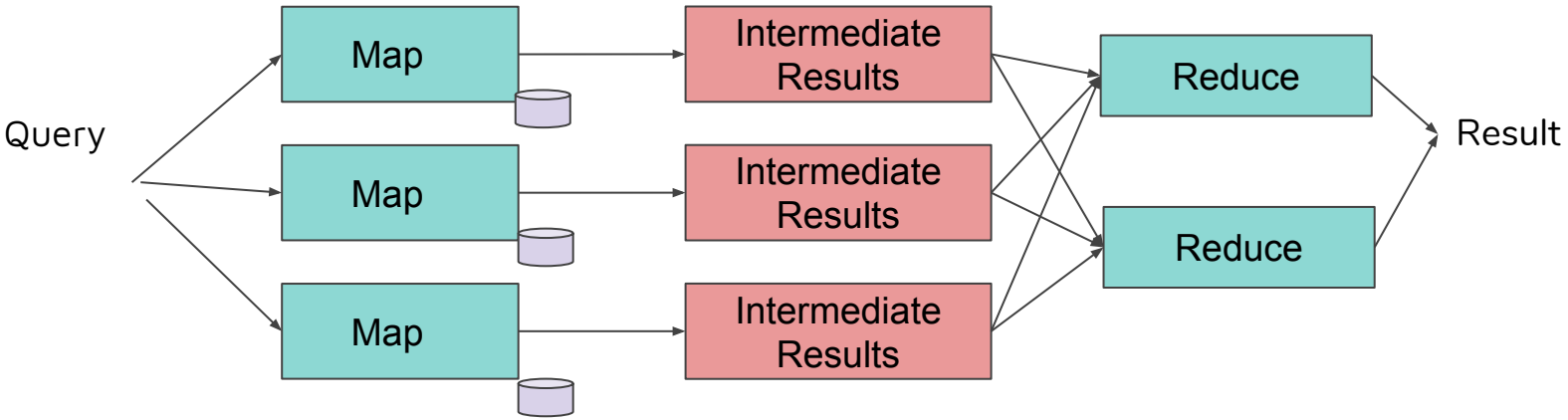
Data Node: Stores the data in the local file system
It also stores checksum
Reports the contents to NameNode
Periodically sends heartbeat to detect node failures
Serves read / write request from Clients directly

Storage of Data in HDFS



Hadoop Components

- **MapReduce Programming Model**
  - It is a **programming model** to express computational tasks



Job ==== {Map} ==========> {Reduce} ======> Solution

Bringing Mapreduce to Hadoop

Advantages :

1. Easily Scalable
2. Managed Workflow
3. Fault Tolerance

MR + 

Hadoop MapReduce

Job Tracker: Assigns map and reduce tasks to task trackers
Schedule resources for user job
Monitor status of job tracker , re-executes upon failure

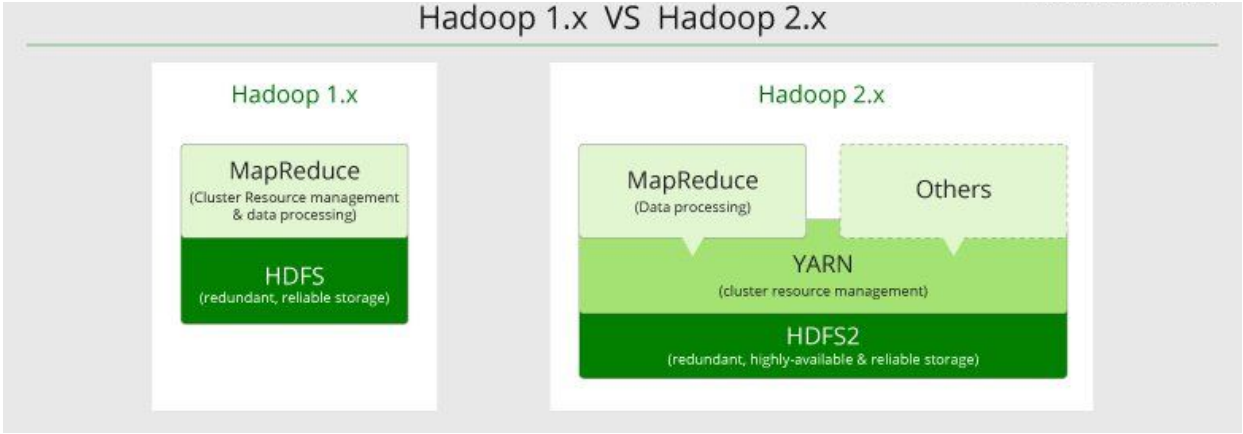
Task Tracker: It runs map and reduce tasks upon instruction from job tracker
Also manages storage and transmission of intermediate results

Hadoop Version 1

Run only MapReduce jobs!!

Hadoop version 2

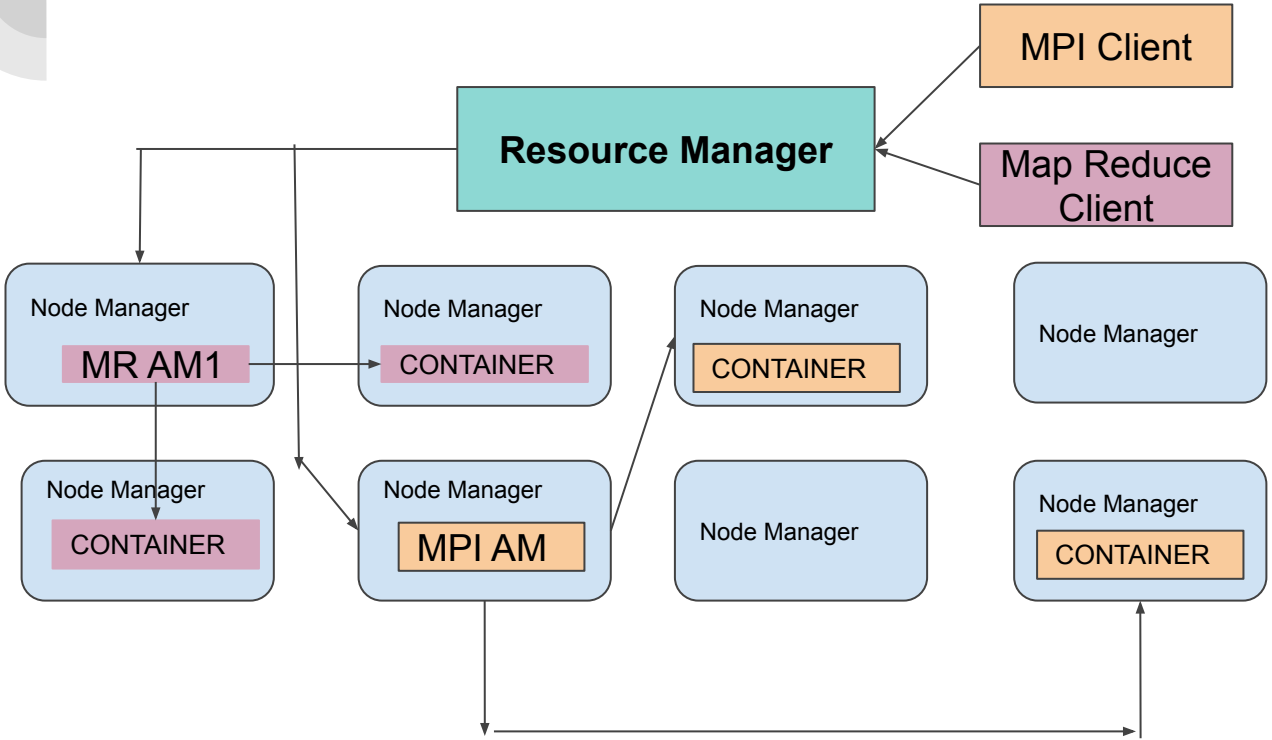
- Separate the Functionalities of resource management and job life cycle management
- Yet Another Resource Negotiator (YARN)
  - Scalability
  - Availability
  - Wider Processing Frameworks (Support for streaming applications)



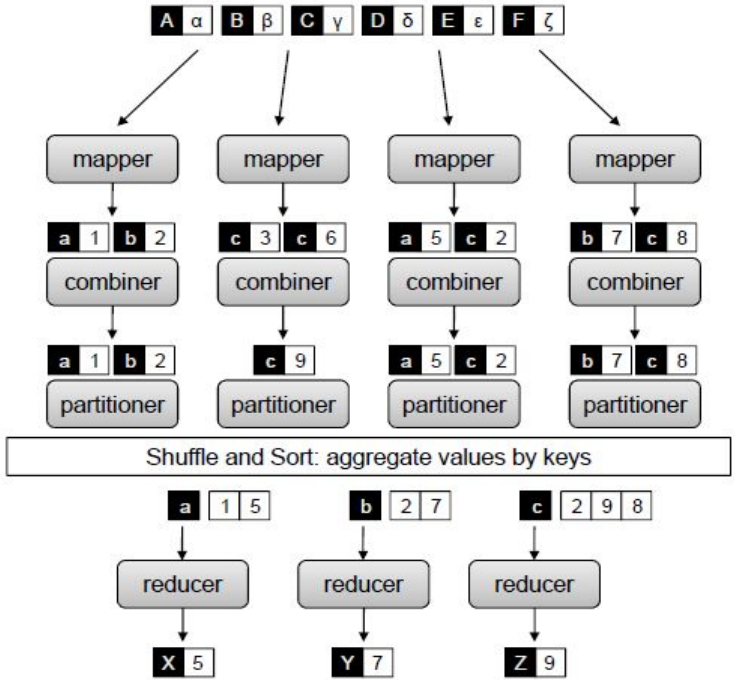
YARN

Resource Manager	Allocating Resources (containers) based on needs of an application
Node Manager	To enforce and track assignments locally
	Monitor resource availability, fault reporting, container life cycle management
Application Master	container which oversees the execution of the application
Job History Server	Client Request and status of client jobs are maintained

YARN


MPI Client
Resource Manager
Map Reduce Client
Node Manager
MR AM1
Node Manager
CONTAINER
Node Manager
CONTAINER
Node Manager
Node Manager
CONTAINER
Node Manager
MPI AM
Node Manager
Node Manager
CONTAINER

MapReduce Programming Model



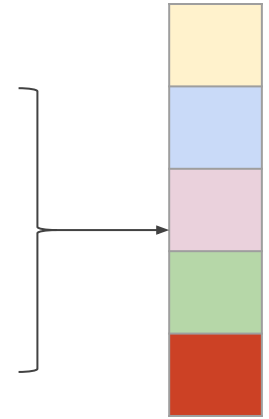
Map : (Key, Value)

Calculate User Session Length

Obtain the user log from the website

Identify how long the user has spent time with the website??

User Id	Date	Length of Time	Last Performed Operation	….



Can it be expressed as mapreduce Job??

For every user , find session length

For user belongs to {U}

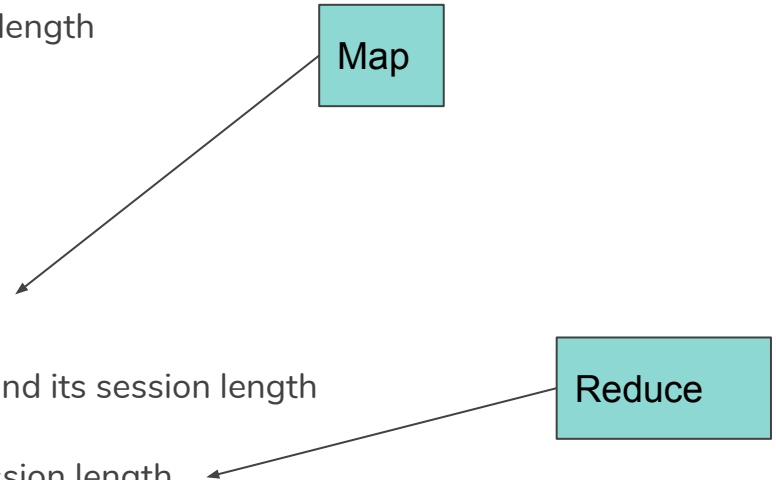
sum = 0

FInd tuples with userid

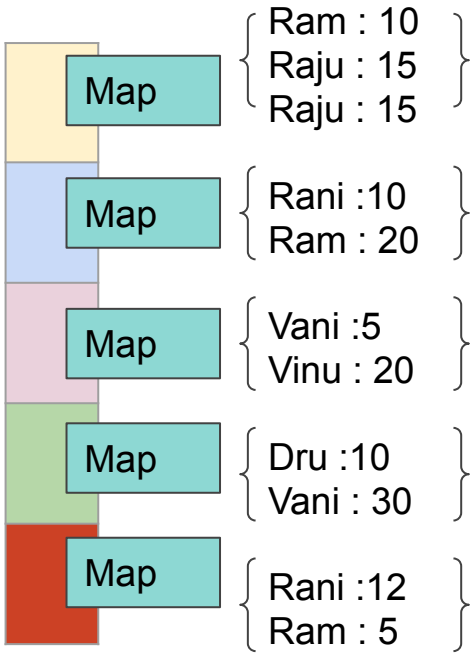
For every tuple find its session length

sum = sum + session length

return sum



MapReduce Model - Map

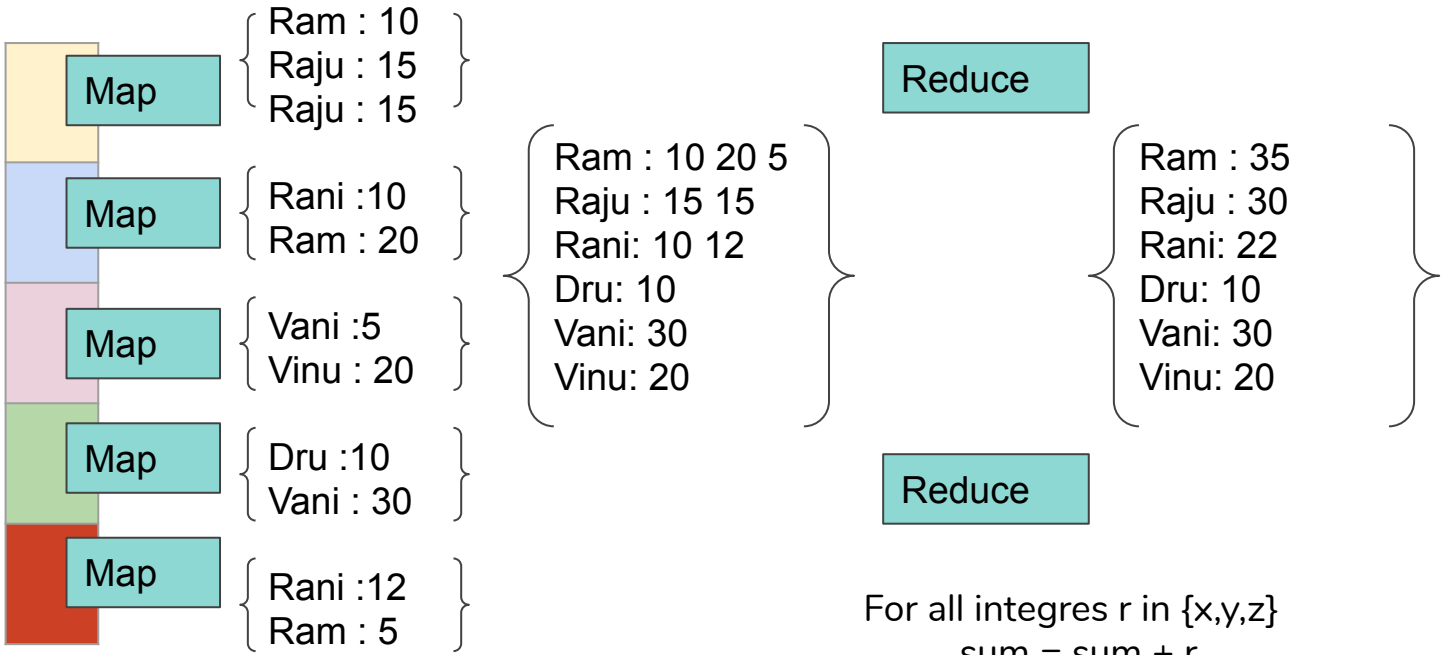


Mapper

Compose key value pairs as required

Emit (User Id , Session Length)

MapReduce Model - Reduce



Thanks

Any Queries (or) Suggestions
Drop in a mail @ nrr.it@psgtech.ac.in