**Problem Statement**

1. **Explain the core changes made in Hadoop 2.x**

Hadoop 1.x has several issues which were resolved by Hadoop 2.x. Some major issues faced by Hadoop 1.x are-

1. Name Node memory constraint- Amount of meta data, name node can store is restricted to the memory available to Name node daemon in Hadoop 1.x. Also number of nodes that Hadoop 1.x cluster could manage was around 4000 machines.
2. Name node being single point of failure- Name node was turning out to be the single point of failure as of Hadoop 1.x, if name node goes down, the entire cluster would have come to a stand still. A Hadoop admin intervention is required which could mean a cluster downtime of at least 45 minutes to 1 hour.
3. Job Tracker overloading- While processing the data, sometimes the processing daemon becomes overloaded. Processing daemon does too many things like- Job Allocation, Job Scheduling and Monitoring, Resource Allocation etc. Also the processing part of Hadoop 1.x supported only one processing framework.

These all issues were addressed in Hadoop 2.x by

1. Issue 1 was **Name Node memory constraint** was handled by Hadoop 2.x by **HDFS Federation.**  In HDFS Federation we use multiple name nodes each connected with their 4000 machines containing fsimage and edits. In HDFS Federation, there are multiple name nodes, each storing the meta data and block mapping of files and directories contained in particular sub directories.
2. Issue 2 was **Name node being single point of failure**  which was handled by Hadoop 2.x by introducing **High Availability.** In HDFS High Availability, there is a pair of Name Nodes in an active and standby configuration. In the event of failure of an active name node, the standby takes over its duties without a significant interruption.
3. Issue 3 was **Job Tracker Overloading** which was handled by Hadoop 2.x by introducing **YARN.** YARN is Yet another Resource Negotiator provides support for other processing framework like Message Passing Interface, Graph Processing and many more such frameworks. YARN to address the overloading issue of Job Tracker in Hadoop 1.x also divides the job tracker responsibilities of Hadoop 1.x into same separate components.

**Difference between Hadoop 1.x and Hadoop 2.x**

**Hadoop 1.x-** Job Tracker keeps track of Job monitoring and resource allocation.

* Suited for maximum in 4000 nodes and 40000 Tasks.
* Task Tracker configured witch static slots, a map task cannot run on reduce a slot. So cluster utilization is low.
* Supports map reduce processing models.

**Hadoop 2.x-** Resource manager takes care of resource utilization, application master takes care of job monitoring.

* It can scale up to 10000 nodes and 100000 tasks.
* Resources are dynamic and fine grained leads to better cluster utilization.
* Supports processing models other than map reduce.

1. **Explain the difference between MapReduce 1 and MapReduce 2 / Yarn**

YARN was introduced in Hadoop 2.x version which was brought I picture to overcome the limitation Hadoop 1.x which was Job Tracker overloading. YARN supports for other processing framework like Message Passing Interface, Graph Processing and many more such frameworks. It supports 50 processing frameworks. YARN to address the overloading issue of Job Tracker in Hadoop 1.x also divides the job tracker responsibilities of Hadoop 1.x into same separate components.

YARN is a latest management and processing framework of Hadoop. It was created by dividing the processing engine of Hadoop into smaller, more manageable parts. It also monitors and manages jobs submitted by user in a highly performance efficient manner.

**Difference-**

* In Hadoop 2.x management of Map reduce process were separated from resource management. Due to this separation YARN become a better resource manager than MR1 and also enables versatility-the resource manager can support additional paradigm and not just Map reduce.
* In MR1 Job Tracker was managing all the tasks performed by Map Reduce, but in YARN Application Manager manages all the task in turn application master is there which is responsible for managing separate tasks on a Map Reduce. It has resource manager for each cluster.
* YARN has central resource manager component which manages resources and allocates the resources to the application. Multiple applications can run on Hadoop via YARN and all application could share common resource management.

YARN has five components-

1. Resource manager
2. Node manager
3. Application master
4. Scheduler
5. Conatiner