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

a) Single point of failure is rectified

b) High availability is available

c) Allows other applications also to integrate with HDFS

d) Map reduce slots are changed static to dynamic

e) Nodes limitation (4000 to unlimited) is rectified

f) Support both interactive, graph iterative algorithms but in 1.x its not supported

g) Job tracker bottleneck is rectified

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

In mapreduce 1, hadoop centralized all tasks to the jobtracker. It allocate resources and scheduling the jobs across the cluster. In yarn, de-centralized this to ease the work pressure on the jobtracker. Resource manager responsibility allocate resources to the particular nodes and node manager schedule the job on the application master. Yarn allows parallel execution and application master managing and execute the job. This approach can ease many job tracker problems and improves to scale up ability and optimize the job performance. Additionally Yarn can allow to create multiple applications to scale up on the distributed environment.

Hadoop 2: YARN

HDFS federation brings important measures of scalability and reliability to Hadoop. YARN, the other major advance in Hadoop 2, brings significant performance improvements for some applications, supports additional processing models, and implements a more flexible execution engine.

YARN is a resource manager that was created by separating the processing engine and resource management capabilities of MapReduce as it was implemented in Hadoop 1. YARN is often called the operating system of Hadoop because it is responsible for managing and monitoring workloads, maintaining a multi-tenant environment, implementing security controls, and managing high availability features of Hadoop.

Like an operating system on a server, YARN is designed to allow multiple, diverse user applications to run on a multi-tenant platform. In Hadoop 1, users had the option of writing MapReduce programs in Java, in Python, Ruby or other scripting languages using streaming, or using Pig, a data transformation language. Regardless of which method was used, all fundamentally relied on the MapReduce processing model to run.

YARN supports multiple processing models in addition to MapReduce. One of the most significant benefits of this is that we are no longer limited to working the often I/O intensive, high latency MapReduce framework. This advance means Hadoop users should be familiar with the pros and cons of the new processing models and understand when to apply them to particular use cases.