Problem 1)

1. The progress from a Hadoop 1's more restricted processing model of batch oriented MapReduce jobs, to more interactive and specialized processing models of Hadoop 2 will only further position the Hadoop ecosystem as the dominant big data analysis platform.
2. Hadoop 1 popularized MapReduce programming for batch jobs and demonstrated the potential value of large scale, distributed processing. MapReduce, as implemented in Hadoop 1, can be I/O intensive, not suitable for interactive analysis, and constrained in support for graph, machine learning and on other memory intensive algorithms. Hadoop developers rewrote major components of the file system to produce Hadoop 2.
3. Hadoop 1.x has only static slots whereas Hadoop 2.x has dynamic slots.
4. Hadoop 1.x supports only Map Reduce processing model wheras Hadoop 2.x supports model other than map reduce.
5. Hadoop 1.x works for only 4000 commodity machines wheras Hadoop 2.x works for even more than 10000 commodity machines.

Problem 2)

**MapReduce 1.0:**

In a typical Hadoop cluster, racks are interconnected via core switches. Core switches should connect to top-of-rack switches Enterprises using Hadoop should consider using 10GbE, bonded Ethernet and redundant top-of-rack switches to mitigate risk in the event of failure. A file is broken into 64MB chunks by default and distributed across Data Nodes. Each chunk has a default replication factor of 3, meaning there will be 3 copies of the data at any given time. Hadoop is “Rack Aware” and HDFS has replicated chunks on nodes on different racks. JobTracker assign tasks to nodes closest to the data depending on the location of nodes and helps the NameNode determine the ‘closest’ chunk to a client during reads. The administrator supplies a script which tells Hadoop which rack the node is in, for example: /enterprisedatacenter/rack2.

Limitations of MapReduce 1.0 – Hadoop can scale up to 4,000 nodes. When it exceeds that limit, it raises unpredictable behavior such as cascading failures and serious deterioration of overall cluster. Another issue being multi-tenancy – it is impossible to run other frameworks than MapReduce 1.0 on a Hadoop cluster.

**MapReduce 2.0**:

MapReduce 2.0 has two components – YARN that has cluster resource management capabilities and MapReduce.

In MapReduce 2.0, the JobTracker is divided into three services:

* ResourceManager, a persistent YARN service that receives and runs applications on the cluster.  A MapReduce job is an application.
* JobHistoryServer, to provide information about completed jobs
* Application Master, to manage each MapReduce job and is terminated when the job completes.

Also, the TaskTracker has been replaced with the NodeManager, a YARN service that manages resources and deployment on a node. NodeManager is responsible for launching containers that could either be a map or reduce task.

This new architecture breaks JobTracker model by allowing a new ResourceManager to manage resource usage across applications, with ApplicationMasters taking the responsibility of managing the execution of jobs. This change removes a bottleneck and let Hadoop clusters scale up to larger configurations than 4000 nodes. This architecture also allows simultaneous execution of a variety of programming models such as graph processing, iterative processing, machine learning, and general cluster computing, including the traditional MapReduce.