1. HDFS Architecture: HDFS is designed to work in large clusters of servers, with each server hosting storage and executing user application tasks. The storage and computation are distributed across many servers, allowing the system to scale with demand while remaining cost-effective.
2. Components: Hadoop consists of various components, including HDFS (the focus of this paper), MapReduce (a computation framework), HBase (a column-oriented table service), Pig (a dataflow language and execution framework), Hive (a data warehouse infrastructure), ZooKeeper (a distributed coordination service), Chukwa (for collecting management data), and Avro (a data serialization system).
3. HDFS as Part of Hadoop: HDFS is a crucial part of the Hadoop project. Yahoo! has played a significant role in its development and contributed to 80% of the core components of HDFS and MapReduce.
4. HDFS Architecture: HDFS stores file system metadata and application data separately. The NameNode is responsible for managing metadata, while DataNodes store the actual data. Replication is used for data durability, increasing data transfer bandwidth and optimizing data locality.
5. Namespace Implementation: HDFS uses a single NameNode to manage the file system's namespace and keeps the entire namespace in RAM. The DataNodes store data blocks and report block locations to the NameNode.
6. File System Operations: HDFS provides standard file system operations like reading, writing, deleting files, and creating directories. An HDFS client communicates with the NameNode to determine the locations of data blocks and reads/writes data from/to the nearest DataNodes.
7. Image and Journal: HDFS stores the namespace image, which represents the file system's organization, and the journal, a commit log for changes. These are used for recovery and consistency.
8. CheckpointNode: In addition to the primary NameNode, HDFS can have a CheckpointNode that periodically combines checkpoint and journal files, creating new checkpoints. This process is essential for system reliability and maintenance.
9. BackupNode: The BackupNode maintains an in-memory image of the file system namespace and accepts journal transactions from the active NameNode. It provides an up-to-date backup in case of a NameNode failure.
10. Upgrades and Snapshots: HDFS allows creating snapshots to protect data during software upgrades. Snapshots capture the file system's state at a specific point, enabling rollback if the upgrade results in data loss or corruption.
11. Layout Version: HDFS uses layout versions to manage changes in data representation formats. During software upgrades, older formats are automatically converted to newer ones.
12. Handshake: The handshake is a communication process that occurs when a DataNode connects to the NameNode in HDFS. It is a crucial step to establish trust and verify the integrity of the DataNode. The primary purposes of the handshake include:
    1. Verification: During the handshake, the NameNode verifies the identity of the DataNode by checking whether it has the correct namespace ID and software version. The namespace ID is a unique identifier for the file system instance. This verification ensures that the DataNode is compatible with the cluster and is not a rogue or unauthorized node.
    2. Compatibility: Ensuring that the DataNode's software version matches that of the NameNode is essential to prevent data corruption and maintain the consistency of the file system. Incompatible software versions can lead to errors or data loss.
    3. Cluster Integrity: The handshake helps maintain the overall integrity of the HDFS cluster by preventing nodes with different namespace IDs or incompatible software versions from joining the cluster.