Summary: HDFS

Here is a quick summary of HDFS for you!

We'll cover the following

- Summary
- System design patterns
- References and further reading

Summary#

- HDFS is a scalable distributed file system for large, distributed data-intensive applications.
- HDFS uses commodity hardware to reduce infrastructure costs.
- HDFS provides APIs for usual file operations like `create`, `delete`, `open`, `close`, `read`, and `write`.
- Random writes are not possible; writes are always made at the end of the file in an append-only fashion.
- HDFS does not support multiple concurrent writers.
- An HDFS cluster consists of a **single NameNode** and **multiple DataNodes** and is accessed by multiple clients.
- **Block**: Files are broken into fixed-size blocks (default 128MB), and blocks are replicated across a number of DataNodes to ensure fault-tolerance. The block size and the replication factor are configurable.
- DataNodes store blocks on local disk as Linux files.
- NameNode server is the coordinator of an HDFS cluster and is responsible for keeping track of all filesystem metadata.

responsible for keeping track of all filesystem metadata.

- NameNode keeps all metadata in memory for faster operations. For fault-tolerance and in the event of NameNode crash, all metadata changes are written to the disk onto an **EditLog**. This EditLog can also be replicated on a remote filesystem (e.g., NFS) or a secondary NameNode.
- The NameNode does not keep a persistent record of which DataNodes have a replica of a given block. Instead, the NameNode asks each DataNode about what blocks it holds at NameNode startup and whenever a DataNode joins the cluster.
- **FsImage**: The NameNode state is periodically serialized to disk and then replicated, so that on recovery, a NameNode may load the checkpoint into memory, replay any subsequent operations from the edit log, and be available again very quickly.
- **HeartBeat**: The NameNode communicates with each DataNode through Heartbeat messages to pass instructions and collect its state.
- **Client**: User applications interact with HDFS through its client. HDFS Client interacts with NameNode for metadata, but all data transfers happen directly between the client and DataNodes.
- **Data Integrity**: Each DataNode uses checksumming to detect the corruption of stored data.
- **Garbage Collection**: Any deleted file is renamed to a hidden name to be garbage collected later.
- **Consistency**: HDFS is a strongly consistent file system. Each data block is replicated to multiple nodes, and a write is declared to be successful only after all the replicas have been written successfully.
- **Cache**: For frequently accessed files, the blocks may be explicitly cached in the DataNode's memory, in an off-heap block cache.
- **Erasure coding**: HDFS uses erasure coding to reduce replication overhead.

System design patterns#

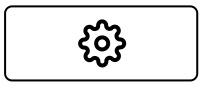
System design patterns#

Here is a summary of system design patterns used in HDFS.

- **Write-Ahead Log**: For fault tolerance and in the event of NameNode crash, all metadata changes are written to the disk onto an EditLog which is a write-ahead log.
- **HeartBeat**: The HDFS NameNode periodically communicates with each DataNode in HeartBeat messages to give it instructions and collect its state.
- **Split-Brain**: ZooKeeper is used to ensure that only one NameNode is active at any time. Fencing is used to put a fence around a previously active NameNode so that it cannot access cluster resources and hence stop serving any read/write request.
- **Checksum:** Each DataNode uses checksumming to detect the corruption of stored data.

References and further reading#

- HDFS paper (https://storageconference.us/2010/Papers/MSST/Shvachko.pdf)
- HDFS High Availability (HA)architecture (https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HDFSHighAvailabilityWithNFS.html)
- Apache HDFS Architecture (https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html)
- Distributed File Systems: A Survey (http://ijcsit.com/docs/Volume%205/vol5issue03/ijcsit20140503234.pdf)



← Back

HDFS Characteristics

Next →

Quiz: HDFS

Mark as Completed

Report an Issue