# hdfs总结

**NN的启动过程**

1. Namenode如果是正常启动（非format），则会调用initialize方法，在这个方法里完成初始化和启动的全部工作:
   1. Initialize方法首先进行一些权限方面的配置，然后初始化**this**.namesystem = **new** FSNamesystem(**this**, conf);大部分的启动工作都在FSNamesystem内部完成。
   2. FSNamesystem内部也有一个initialize方法，在这个initialize方法中，会初始化整个系统的大部分模块：
      1. 初始化**this**.dir = **new** FSDirectory(**this**, conf);
      2. 加载保存的文件系统镜像**this**.dir.loadFSImage
      3. 初始化安全模式**this**.safeMode = **new** SafeModeInfo(conf);
      4. 初始化几个线程，用于监控系统状态，包括：
         1. 心跳监控：HeartbeatMonitor
         2. 租约监控：LeaseManager.Monitor
         3. 副本状态监控：ReplicationMonitor
         4. Datanode监控：DecommissionManager.Monitor
         5. 正在复制的副本监控：PendingReplicationMonitor
      5. 另外还有几个数据结构，用于维护系统状态：
         1. 整个文件系统的逻辑层次：FSDirectory dir
         2. Block->BlcokInfo的映射：BlocksMap blocksMap（BlcokInfo包含了INode, datanodes, self ref等信息）
         3. 冲突的block信息Block -> TreeSet<DatanodeDescriptor>：CorruptReplicasMap corruptReplicas
         4. datanode到其上block的映射StorageID -> DatanodeDescriptor：NavigableMap<String, DatanodeDescriptor> datanodeMap（DatanodeDescriptor中存储了BlcokInfo，而通过BlcokInfo可以索引到datanode上所有的block）
         5. 废弃的block信息，StorageID -> ArrayList<Block>：Map<String, Collection<Block>> recentInvalidateSets
         6. 多余的block信息，StorageID -> TreeSet<Block>：Map<String, Collection<Block>> excessReplicateMap
         7. 心跳信息：ArrayList<DatanodeDescriptor> heartbeats
         8. 需要更多副本的block集合：UnderReplicatedBlocks neededReplications
         9. 正在执行副本备份的block集合：PendingReplicationBlocks pendingReplications
         10. 租约集合：LeaseManager leaseManager
         11. host到datanode的映射信息：Host2NodesMap host2DataNodeMap（因为一个host可能被映射到多个datanode）
   3. FSNamesystem初始化结束后，Namenode初始化rpc server：**this**.server = RPC.*getServer*。然后启动http和rpc服务器startHttpServer(conf);和**this**.server.start();
   4. 最后，会启动垃圾回收器：startTrashEmptier(conf);
2. FSDirectory，存储filesystem directory state，实时维护了filename->blockset的映射，并记录日志到硬盘。
   1. 启动NN时，在FSNamesystem的初始化方法中会调用loadFSImage来加载整个文件系统的镜像，实际是先后调用FSImage的loadFSImage和loadFSEdits两个方法
   2. FSImage的loadFSImage方法从Image文件中加载整个文件系统的镜像，镜像文件存储格式为：



共有numFiles个文件，循环遍历每个文件，把读取的文件添加到系统的FSDirectory数据结构中。（上图中的block只包含了blockid，而没有包含block的位置信息，block到底存储在哪个DN上，要等待DN的报告）

* 1. FSImage的loadFSEdits方法实际是调用FSEditLog的*loadFSEdits*方法，这个方法循环读取edits文件，每次先读取操作符，直到读取到了*OP\_INVALID*，表示文件的结束。对于不同的操作符，执行不同的操作，最终都把日志中的变更应用到之前加载到内存中的文件系统上。具体参考FSEditlog.doc。



