

分布式文件系统 HDFS

孙国库 | 2020年9月

目录

CONTENTS

- 1 HDFS简介
- 2 HDFS原理
- 3 HDFS文件管理
- 4 HDFS系统管理



1 chapter

HDFS简介

- ✓ 什么是HDFS
- ✓ 优缺点

➤ 概念

- Hadoop分布式文件系统（Hadoop Distributed File System）
- 2003年10月Google发表了GFS（Google File System）论文
- HDFS是GFS的开源实现
- HDFS是Apache Hadoop的核心子项目
- 在开源大数据技术体系中，地位无可替代

➤ 设计目标


- 运行在大量廉价商用机器上：硬件错误是常态，提供容错机制
- 简单一致性模型：一次写入多次读取，支持追加，不允许修改，保证数据一致性
- 流式数据访问：批量读而非随机读，关注吞吐量而非时间
- 存储大规模数据集：典型文件大小GB~TB，关注横向线性扩展

➤ 优点

- 高容错、高可用、高扩展
 - 数据冗余，多Block多副本，副本丢失后自动恢复
 - NameNode HA、安全模式
 - 10K节点规模
- 海量数据存储
 - 典型文件大小GB~TB，百万以上文件数量， PB以上数据规模
- 构建成本低、安全可靠
 - 构建在廉价商用服务器上
 - 提供了容错和恢复机制
- 适合大规模离线批处理
 - 流式数据访问
 - 数据位置暴露给计算框架

➤ 缺点

- 不适合低延迟数据访问
- 不适合大量小文件存储
 - 元数据占用NameNode大量内存空间
 - ✓ 每个文件或目录的元数据要占用150Byte
 - ✓ 存储1亿个元素，大约需要20GB内存
 - ✓ 如果一个文件为10KB，1亿个文件大小仅1TB，却要消耗掉20GB内存
 - 磁盘寻道时间超过读取时间
- 不支持并发写入
 - 一个文件同时只能有一个写入者
- 不支持文件随机修改
 - 仅支持追加写入

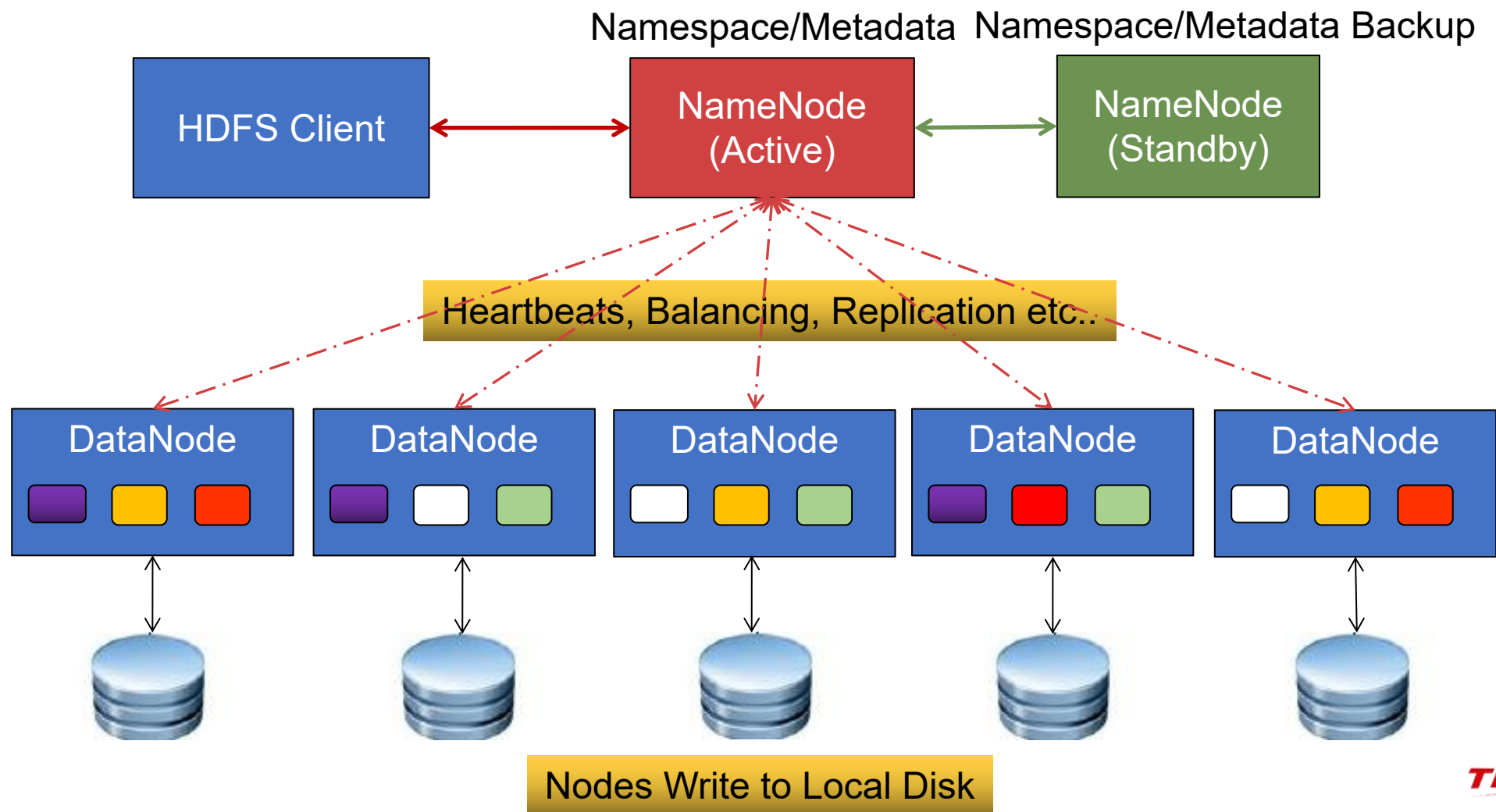


2 chapter

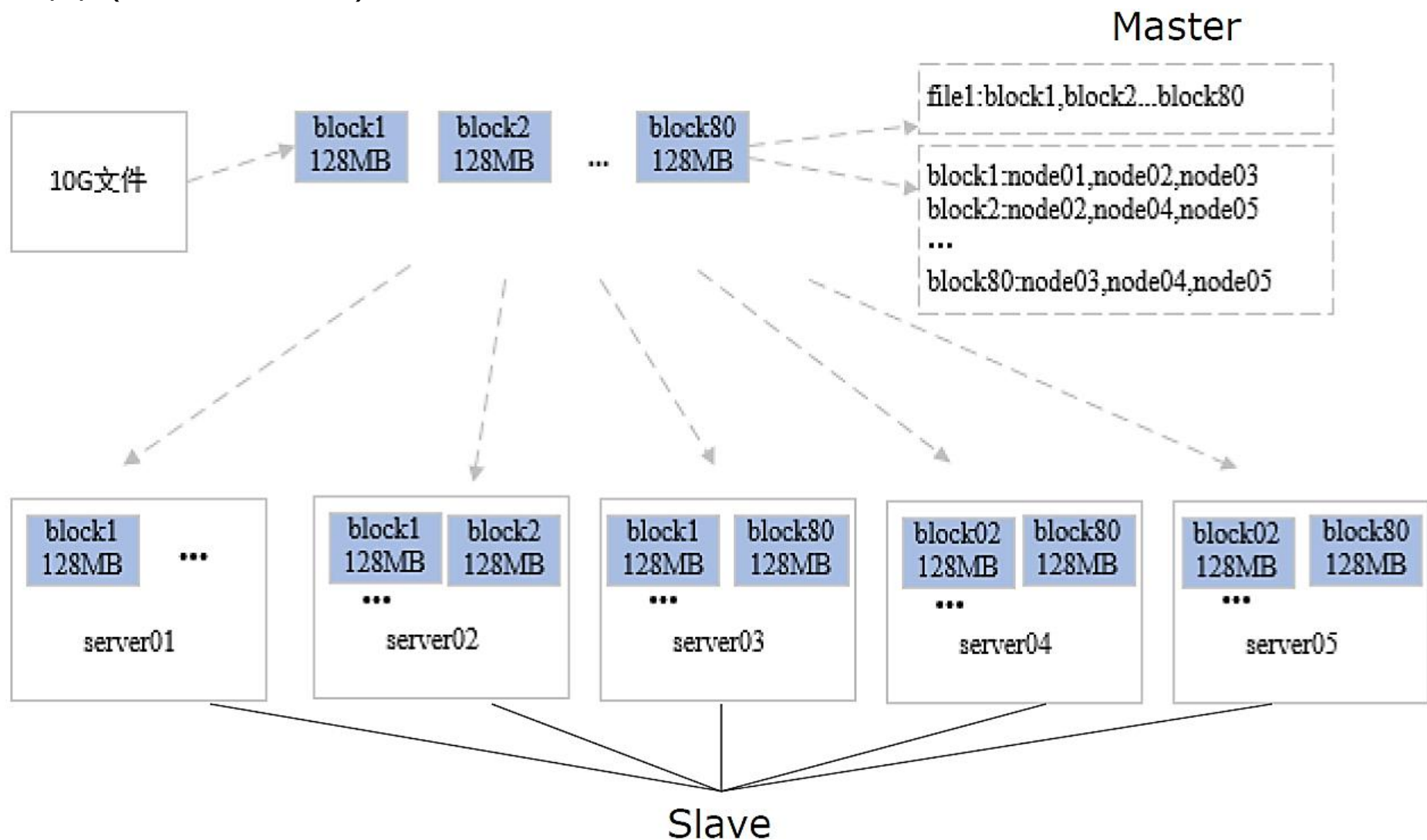
HDFS原理

- ✓ 系统架构
- ✓ 存储机制
- ✓ 读写操作
- ✓ 安全模式
- ✓ 高可用

➤ 系统架构图 (Master/Slave)



➤ 系统架构图 (Master/Slave)

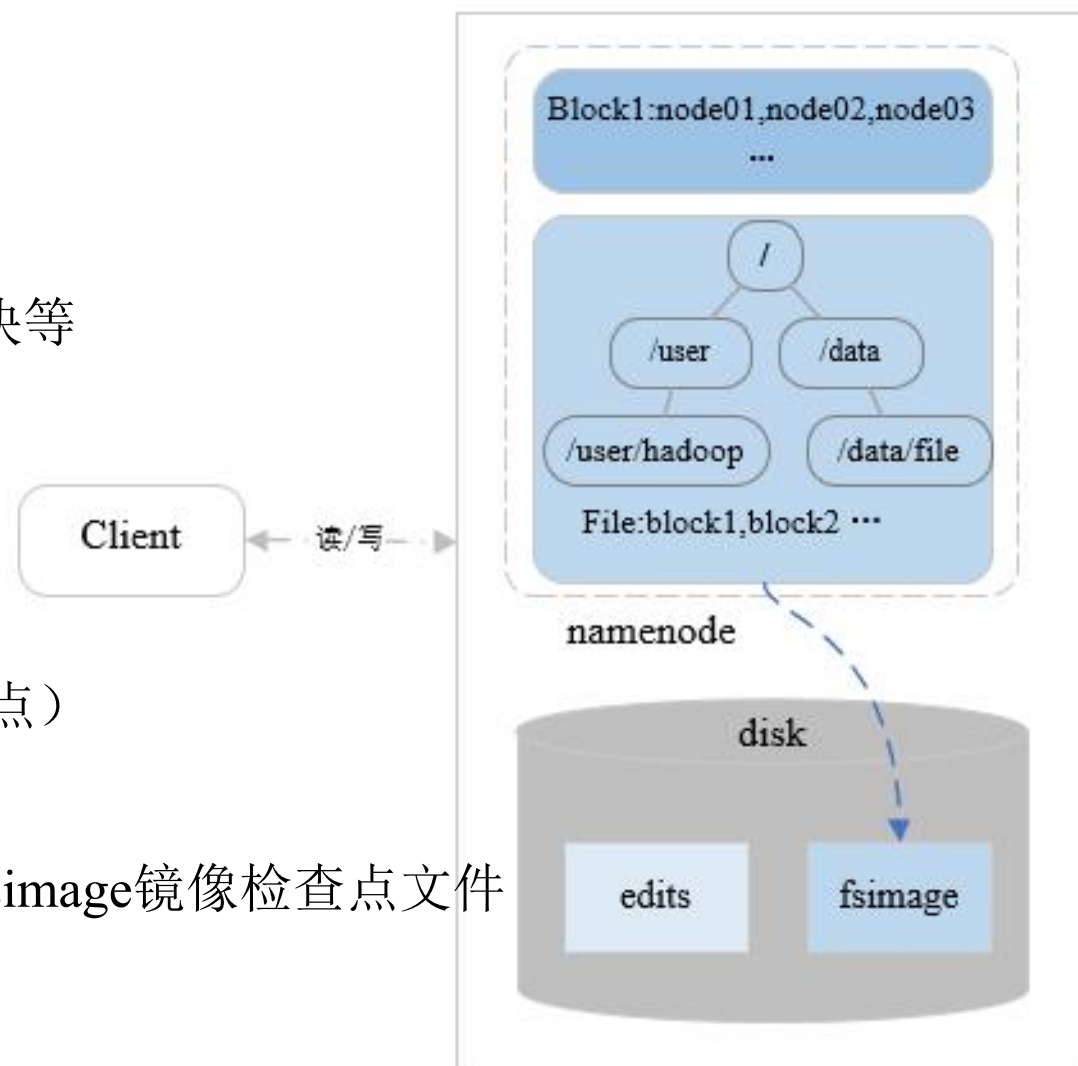


➤ Active NameNode (AN)

- 活动Master管理节点（集群中唯一）
- 管理命名空间
- 管理元数据：文件的位置、所有者、权限、数据块等
- 管理Block副本策略：默认3个副本
- 处理客户端读写请求，为DataNode分配任务

➤ Standby NameNode (SN)

- 热备Master管理节点（Active NameNode的热备节点）
 - Hadoop 3.0允许配置多个Standby NameNode
- 同步元数据，即周期性下载edits编辑日志，生成fsimage镜像检查点文件
- Active NameNode宕机后，快速升级为新的Active



➤ NameNode元数据文件

- edits（编辑日志文件）：保存了自最新检查点（Checkpoint）之后的所有文件更新操作
- fsimage（元数据检查点镜像文件）：保存了文件系统中所有的目录和文件信息，如某个目录下有哪些子目录和文件，以及文件名、文件副本数、文件由哪些Block组成等
- Active NameNode内存中有一份最新的元数据（= fsimage + edits）
- Standby NameNode在检查点定期将内存中的元数据保存到fsimage文件中

➤ DataNode

- Slave工作节点（可大规模扩展）
- 存储Block和数据校验和
- 执行客户端发送的读写操作
- 通过心跳机制定期（默认3秒）向NameNode汇报运行状态和Block列表信息
- 集群启动时，DataNode向NameNode提供Block列表信息

➤ Block数据块

- HDFS的最小存储单元
- 文件写入HDFS会被切分成若干个Block
- Block大小固定，默认为128MB，可自定义
- 若一个Block的大小小于设定值，不会占用整个块空间
- 默认情况下每个Block有3个副本

➤ Client

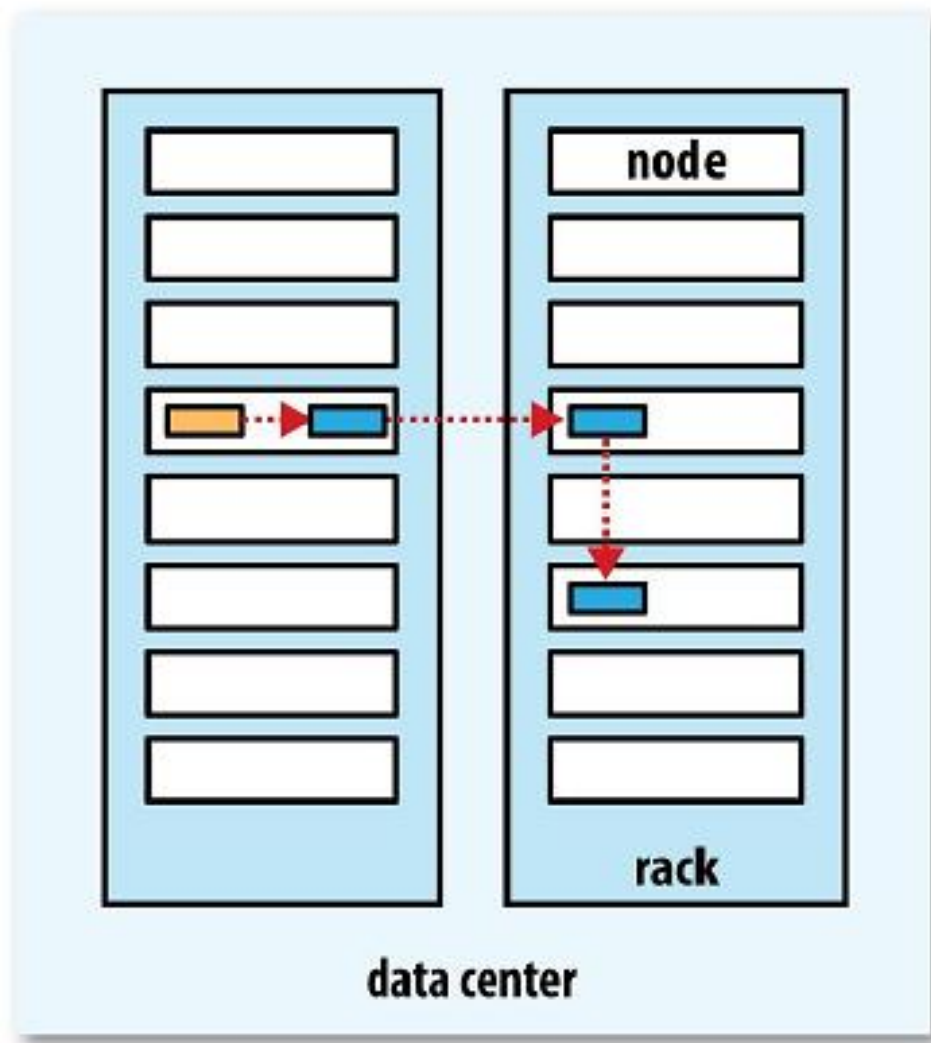
- 将文件切分为Block
- 与NameNode交互，获取文件访问计划和相关元数据
- 与DataNode交互，读取或写入数据
- 管理HDFS

➤ Block存储

- Block是HDFS的最小存储单元
- 如何设置Block大小
 - 目标：最小化寻址开销，降到1%以下
 - 默认大小：128M
 - 块太小：寻址时间占比过高
 - 块太大：Map任务数太少，作业执行速度变慢
- Block和元数据分开存储：Block存储于DataNode，元数据存储于NameNode
- Block多副本
 - 以DataNode节点为备份对象
 - 机架感知：将副本存储到不同的机架上，实现数据的高容错
 - 副本均匀分布：提高访问带宽和读取性能，实现负载均衡

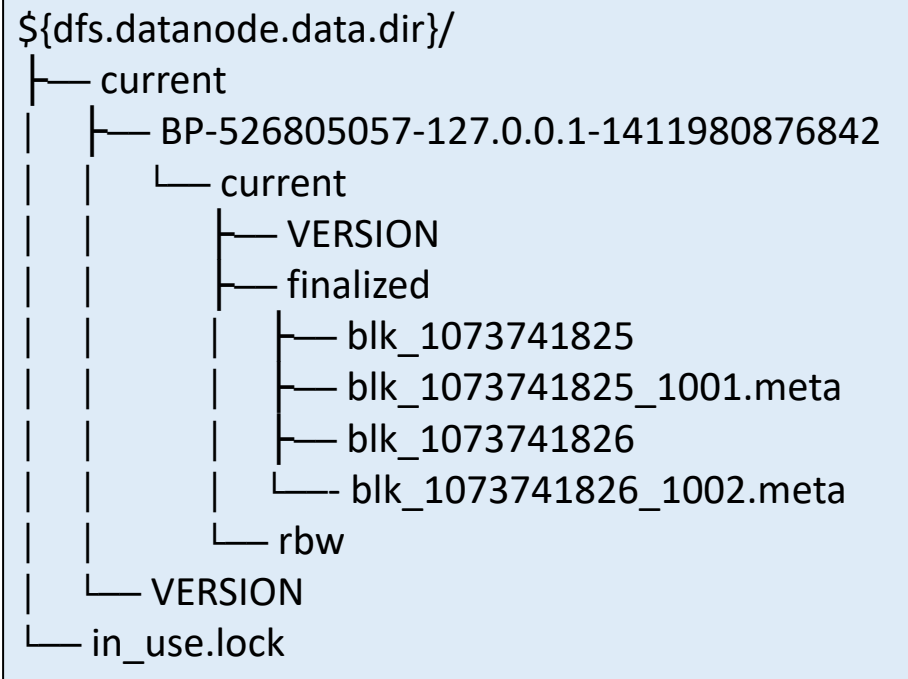
➤ Block副本放置策略

- 副本1：放在Client所在节点
 - 对于远程Client，系统会随机选择节点
- 副本2：放在不同的机架节点上
- 副本3：放在与第二个副本同一机架的不同节点上
- 副本N：随机选择
- 节点选择：同等条件下优先选择空闲节点



➤ Block文件

- Block文件是DataNode本地磁盘中名为“blk_blockId”的Linux文件
 - DataNode在启动时自动创建存储目录，无需格式化
 - DataNode的current目录下的文件名都以“blk_”为前缀
 - Block元数据文件 (*.meta) 由一个包含版本、类型信息的头文件和一系列校验值组成



注：in_use.lock表示DataNode正在对文件夹进行操作

➤ 元数据的两种存储形式

- 内存元数据（NameNode）
- 文件元数据（edits + fsimage）

➤ edits（编辑日志文件）

- Client请求变更操作时，操作首先被写入edits，再写入内存
- edits文件名通过前/后缀记录当前操作的Transaction Id

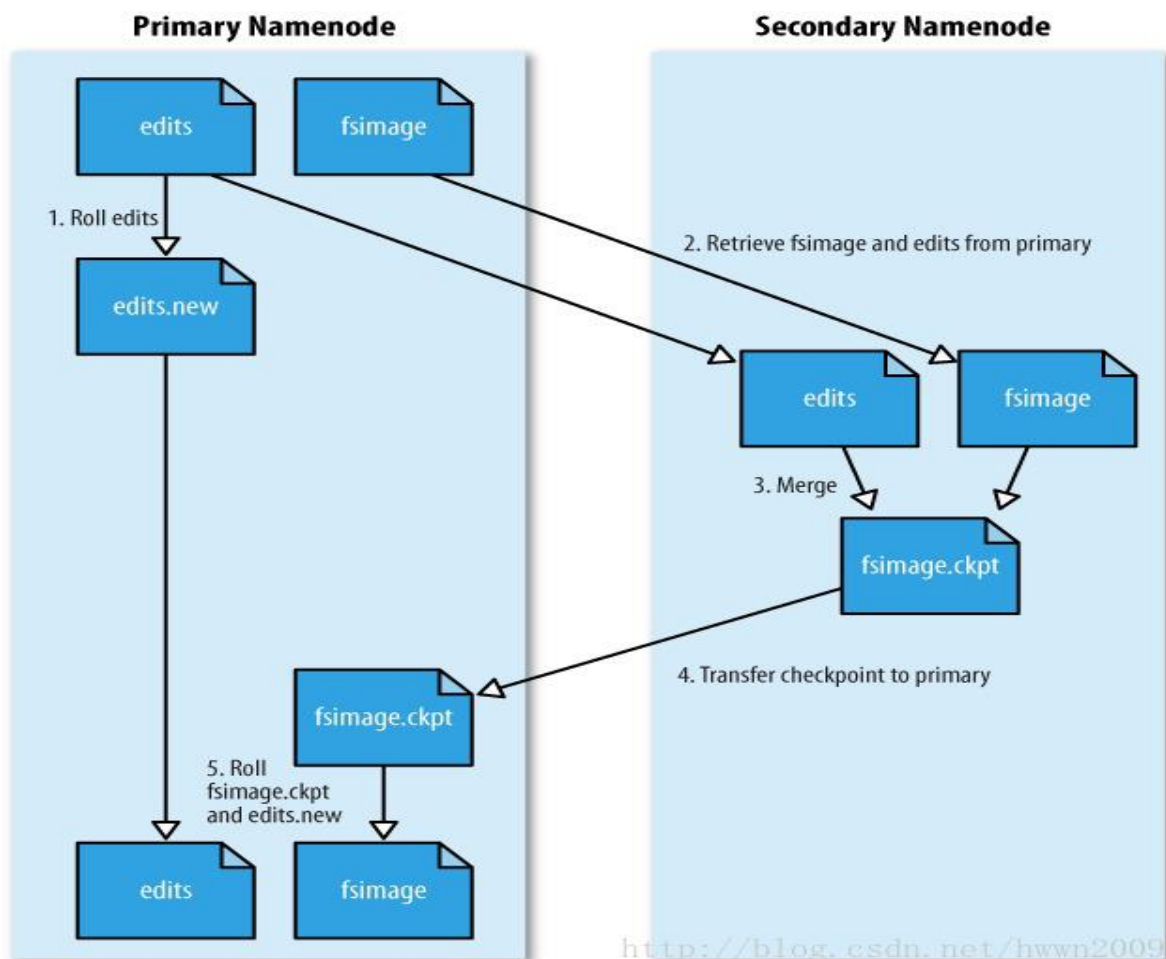
➤ fsimage（元数据镜像检查点文件）

- 不会为文件系统的每个更新操作进行持久化，因为写fsimage的速度非常慢
- fsimage文件名会标记对应的Transaction Id

```
{dfs.namenode.name.dir}/
├── current
│   ├── VERSION
│   ├── edits_00000000000000000001-00000000000000000019
│   ├── edits_inprogress_00000000000000000020
│   ├── fsimage_00000000000000000000
│   ├── fsimage_00000000000000000000.md5
│   ├── fsimage_00000000000000000019
│   ├── fsimage_00000000000000000019.md5
│   └── seen_txid
└── in_use.lock
```

注：in_use.lock表示NameNode正在对文件夹进行操作

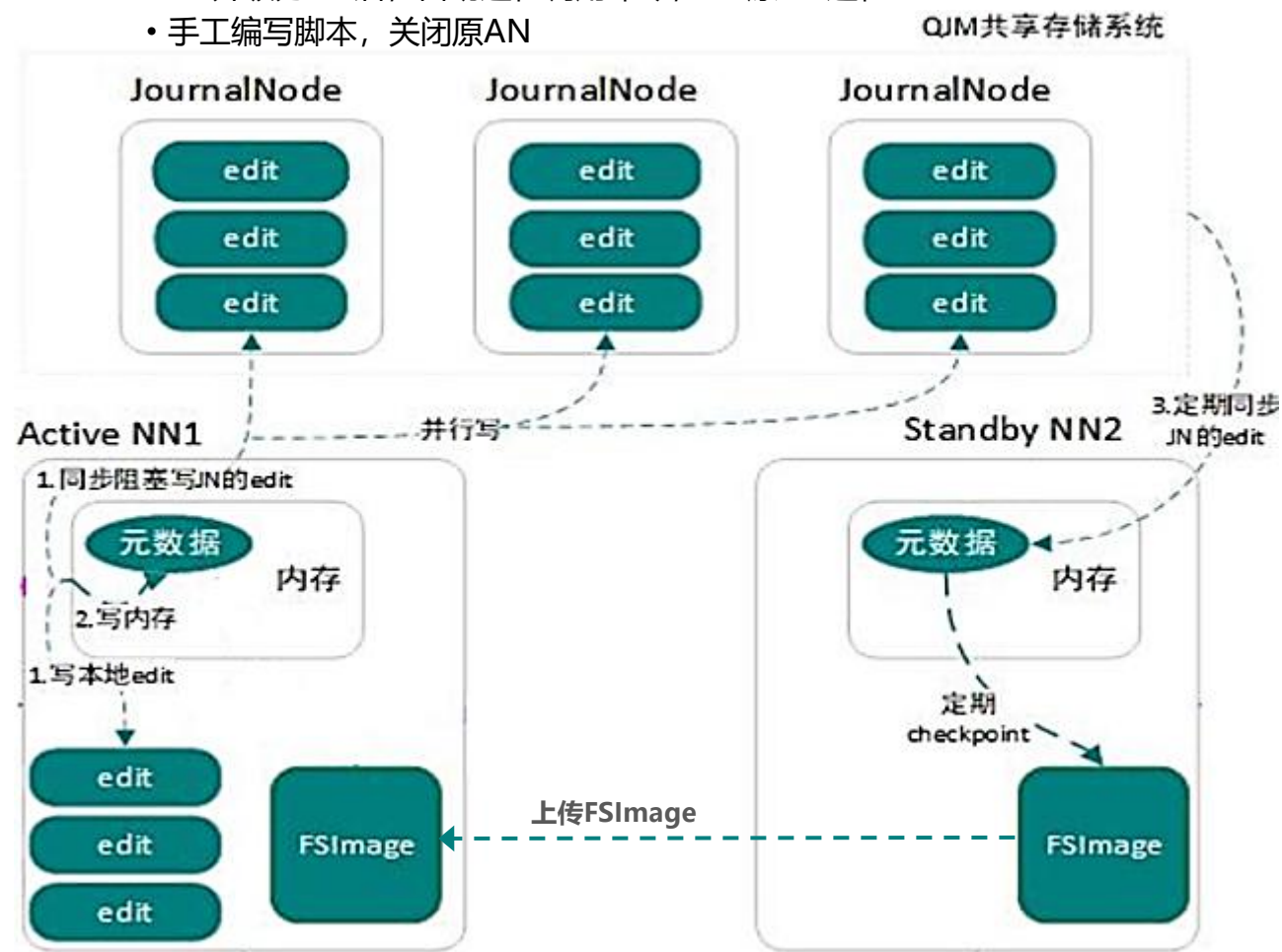
➤ edits与fsimage的合并机制



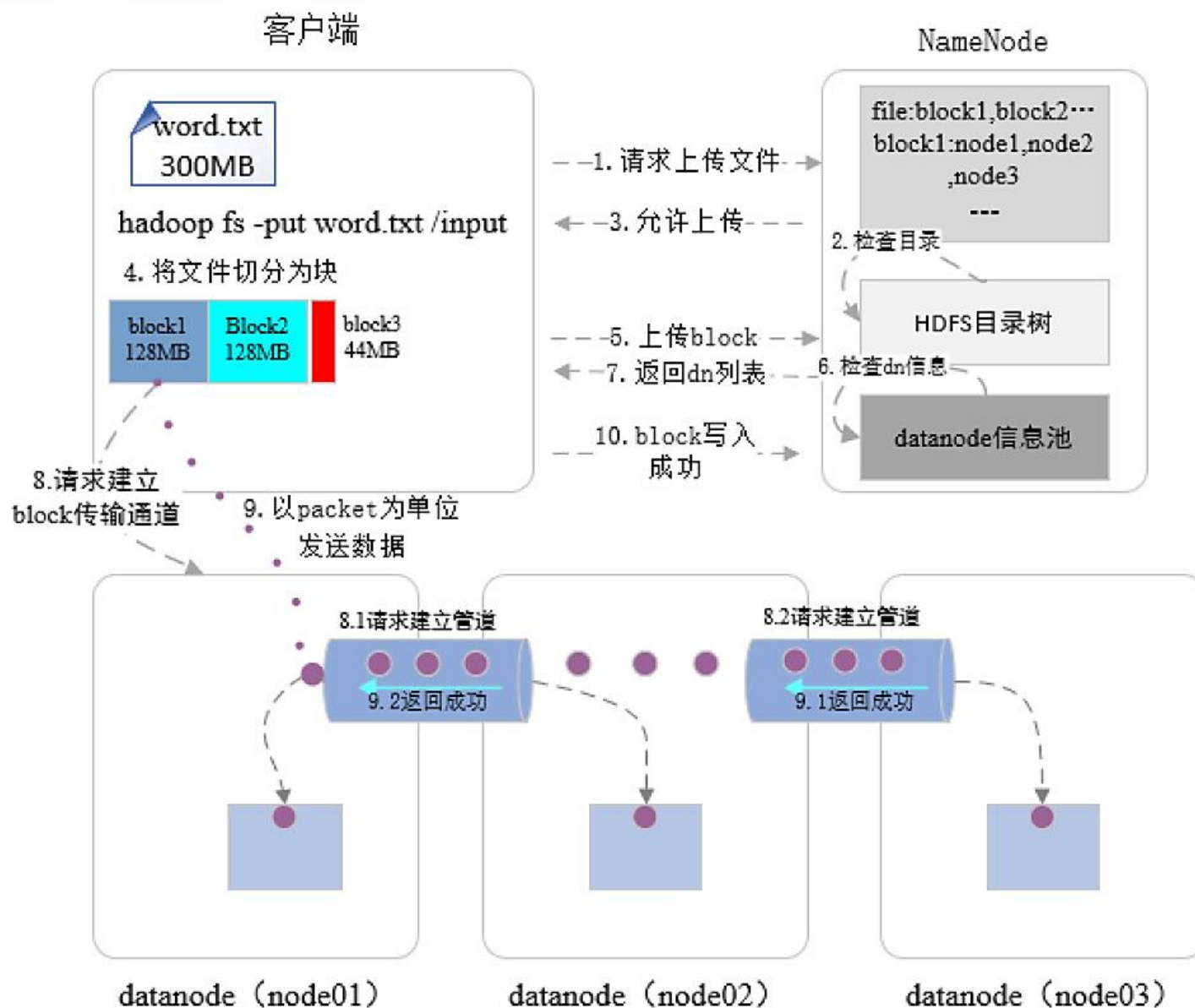
Hadoop 1.x

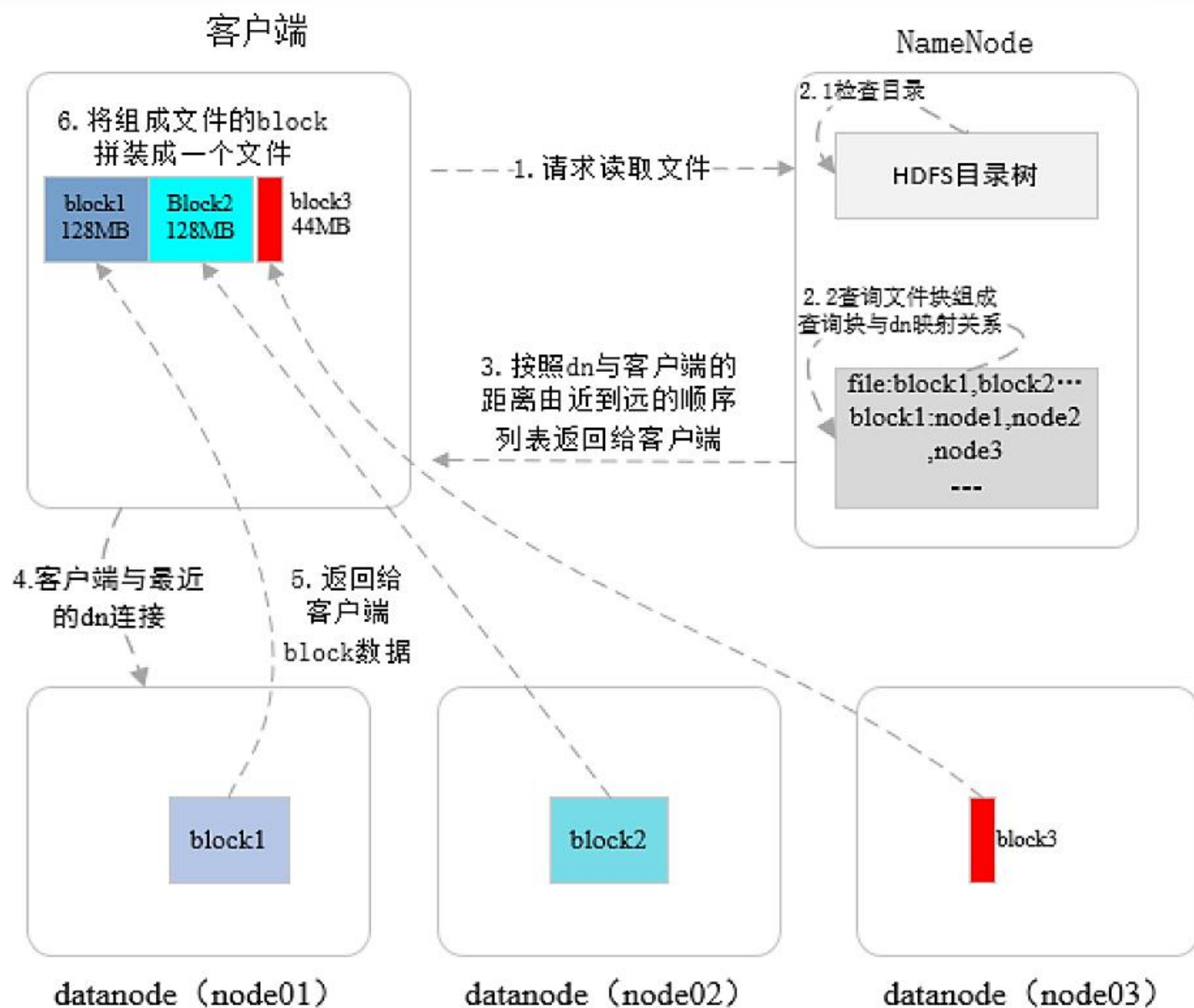
脑裂的处理方法

- SN升级为AN后，自动远程调用命令，将原AN的状态改为Standby
- SN升级为AN后，自动远程调用命令，Kill原AN进程
- 手工编写脚本，关闭原AN



Hadoop 2.x





➤ 什么是安全模式

- 安全模式是HDFS的一种特殊状态，在这种状态下，HDFS只接收读数据请求，而不接收写入、删除、修改等变更请求
- 安全模式是HDFS确保Block数据安全的一种保护机制
- Active NameNode启动时，HDFS会进入安全模式，DataNode主动向NameNode汇报可用Block列表等信息，在系统达到安全标准前，HDFS一直处于“只读”状态

➤ 何时正常离开安全模式

- Block上报率：DataNode上报的可用Block个数 / NameNode元数据记录的Block个数
- 当Block上报率 \geq 阈值时，HDFS才能离开安全模式，默认阈值为0.999
- 不建议手动强制退出安全模式

➤ 触发安全模式的原因

- NameNode重启
- NameNode磁盘空间不足
- Block上报率低于阈值
- DataNode无法正常启动
- 日志中出现严重异常
- 用户操作不当，如：强制关机（**特别注意！**）

➤ 故障排查

- 找到DataNode不能正常启动的原因，重启DataNode
- 清理NameNode磁盘

➤ Active NN与Standby NN的主备切换

➤ 利用QJM实现元数据高可用

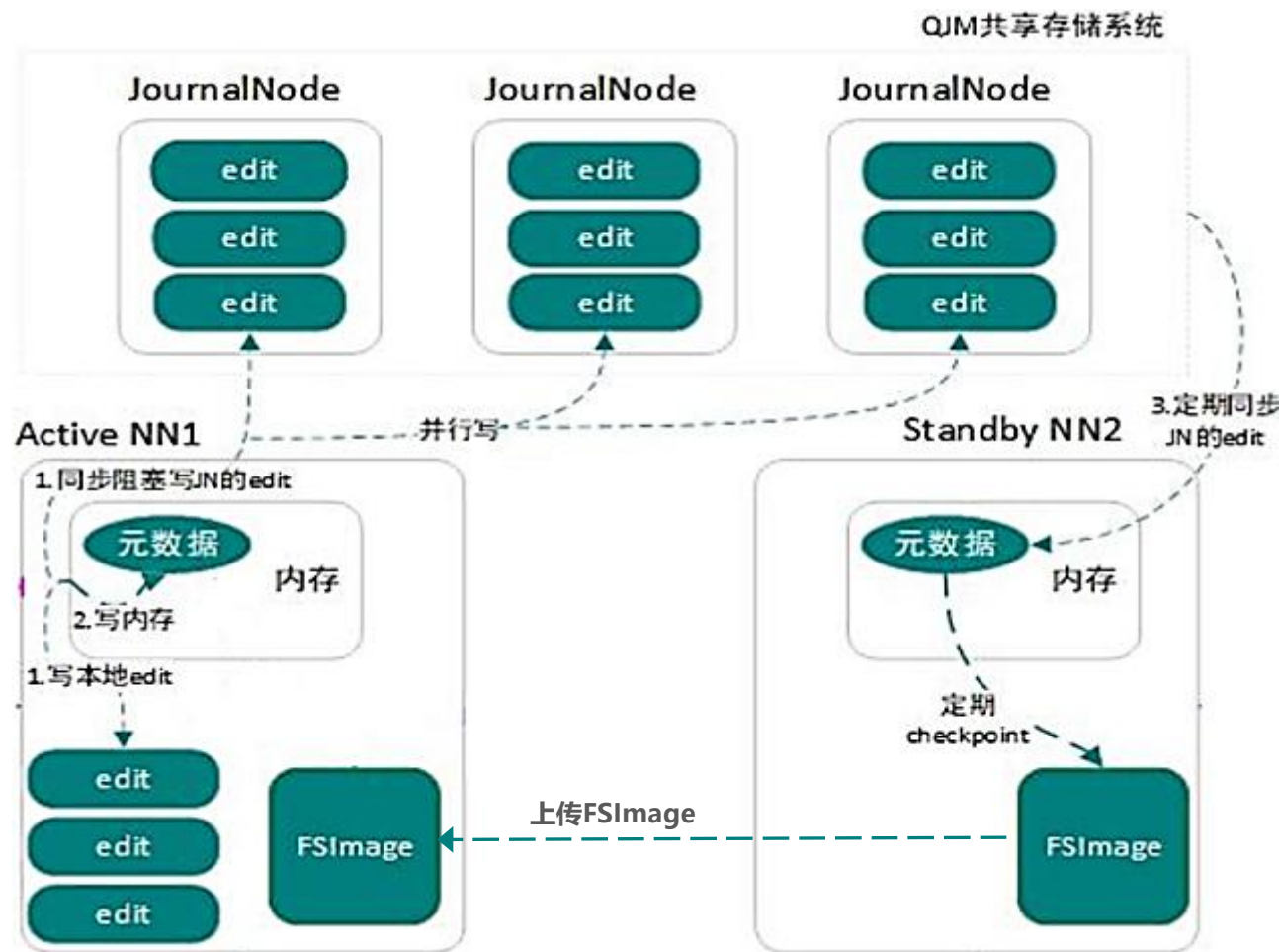
- QJM机制（Quorum Journal Manager）

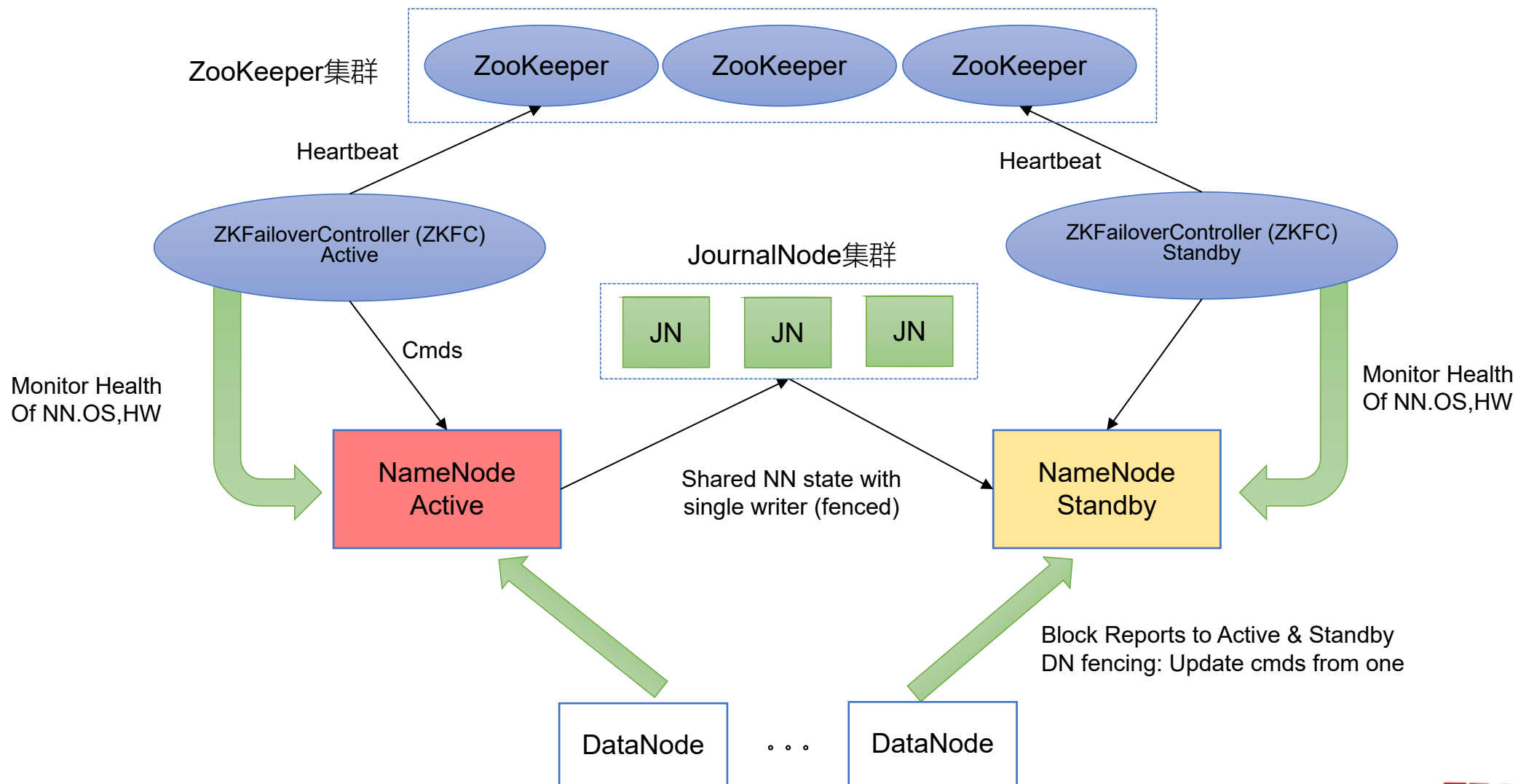
- 只要保证Quorum（法定人数）数量的操作成功，就认为这是一次最终成功的操作


- QJM共享存储系统

- 部署奇数（ $2N+1$ ）个JournalNode
- JournalNode负责存储edits编辑日志
- 写edits的时候，只要超过半数（ $\geq N+1$ ）的JournalNode返回成功，就代表本次写入成功
- 最多可容忍N个JournalNode宕机
- 基于Paxos算法实现

➤ 利用ZooKeeper实现Active节点选举







3 chapter

HDFS文件管理

- ✓ Shell命令
- ✓ REST API

➤ 语法

- `hadoop fs <args>` （使用面最广，可以操作任何文件系统）
- `hdfs dfs <args>` （只能操作HDFS文件系统）
- 大部分用法和Linux Shell类似，可通过`help`查看帮助

➤ HDFS URI

- 格式： `scheme://authority/path`
- 示例： HDFS上的一个文件 `/parent/child`
 - URI全写： `hdfs://nameservice/parent/child` （用`nameservice`替代`namenodehost`）
 - URI简写： `/parent/child`
 - 需在配置文件中定义`hdfs://namenodehost`

Command	Description
<code>hadoop fs -help</code>	Return usage output
<code>hadoop fs -usage command</code>	Return the help for an individual command
<code>hadoop fs -ls [-d] [-h] [-R] <args></code>	Options: -d: Directories are listed as plain files. -h: Format file sizes in a human-readable fashion (eg 64.0m instead of 67108864). -R: Recursively list subdirectories encountered
<code>hadoop fs -get [-ignorecrc] [-crc] <src> <localdst></code>	Copy files to the local file system. Files that fail the CRC check may be copied with the -ignorecrc option. Files and CRCs may be copied using the -crc option. Example: <code>hadoop fs -get /user/hadoop/file localfile</code> <code>hadoop fs -get hdfs://nn.example.com/user/hadoop/file localfile</code>
<code>hadoop fs -put <localsrc> ... <dst></code>	Copy single src, or multiple srcs from local file system to the destination file system. Also reads input from stdin and writes to destination file system.

Command	Description
<code>hadoop fs -cp [-f] [-p -p[topax]] URI [URI ...] <dest></code>	<p>Copy files from source to destination. This command allows multiple sources as well in which case the destination must be a directory.</p> <p>Options:</p> <ul style="list-style-type: none">-f: Overwrite the destination if it already exists.-p: Preserve file attributes [topx] (timestamps, ownership, permission, ACL, XAttr).
<code>hadoop fs -mv URI [URI ...] <dest></code>	<p>Moves files from source to destination. This command allows multiple sources as well in which case the destination needs to be a directory. Moving files across file systems is not permitted.</p>
<code>hadoop fs -rm [-f] [-r -R] [-skipTrash] URI [URI ...]</code>	<p>Delete files specified as args.</p> <p>Options:</p> <ul style="list-style-type: none">-f: the option will not display a diagnostic message or modify the exit status to reflect an error if the file does not exist.-R: the option deletes the directory and any content under it recursively.-r: the option is equivalent to -R.-skipTrash: the option will bypass trash, if enabled, and delete the specified file(s) immediately. This can be useful when it is necessary to delete files from an over-quota directory.

➤ HDFS的所有接口都支持REST API

➤ HDFS URI与HTTP URL

- hdfs://<HOST>:<RPC_PORT>/<PATH>
- http://<HOST>:<HTTP_PORT>/webhdfs/v1/<PATH>?op=...

➤ 写入文件

- Step1: 提交一个HTTP PUT请求, 这个阶段不会传输数据, 只是一个前置条件及一些设定
 - curl -i -X PUT “http://<HOST>:<PORT>/webhdfs/v1/<PATH>?op=CREATE [&overwrite=<true|false>][&blocksize=<LONG>][&replication=<SHORT>] [&permission=<OCTAL>][&bufferize=<INT>]”
- Step2: 提交另一个HTTP PUT请求, 并提供本地的文件路径
 - curl -i -X PUT -T <LOCAL_FILE> “http://<DATANODE>:<PORT>/webhdfs/v1/<PATH>?op=CREATE...”

➤ 获取文件


- 提交HTTP GET请求

- curl -i -L “http://<HOST>:<PORT>/webhdfs/v1/<PATH>?op=OPEN [&offset=<LONG>]
[&length=<LONG>] [&buffersize=<INT>]”

➤ 删除文件

- 提交HTTP DELETE请求

- curl -i -X DELETE “http://<HOST>:<PORT>/webhdfs/v1/<PATH>?op=DELETE [&recursive=<true |false>]”



4

chapter

HDFS系统管理

- ✓ 系统配置
- ✓ Shell命令
- ✓ 系统监控

➤ 核心配置文件

- core-site.xml: Hadoop全局配置
- hdfs-site.xml: HDFS局部配置
- 示例: NameNode URI配置 (core-site.xml)

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://nameservice:9000</value>
  </property>
</configuration>
```

➤ 环境变量文件

- Hadoop-env.sh: 设置了HDFS运行所需的环境变量

➤ `hdfs-site.xml`

Command	Description
<code>dfs.namenode.name.dir</code>	Determines where on the local filesystem the DFS name node should store the name table(fsimage). If this is a comma-delimited list of directories then the name table is replicated in all of the directories, for redundancy.
<code>dfs.datanode.data.dir</code>	Determines where on the local filesystem an DFS data node should store its blocks . If this is a comma-delimited list of directories, then data will be stored in all named directories, typically on different devices. Directories that do not exist are ignored.
<code>dfs.blocksize</code>	The default block size for new files, in bytes. You can use the following suffix (case insensitive): k(kilo), m(mega), g(giga), t(tera), p(peta), e(exa) to specify the size (such as 128k, 512m, 1g, etc.), Or provide complete size in bytes (such as 134217728 for 128 MB).
<code>dfs.datanode.du.reserved</code>	Reserved space in bytes per volume. Always leave this much space free for non hdfs use .
<code>dfs.replication</code>	Default block replication . The actual number of replications can be specified when the file is created. The default is used if replication is not specified in create time.
<code>fs.trash.interval</code>	Number of minutes after which the checkpoint gets deleted . If zero, the trash feature is disabled. This option may be configured both on the server and the client. If trash is disabled server side then the client side configuration is checked. If trash is enabled on the server side then the value configured on the server is used and the client configuration value is ignored.

➤ 服务启动脚本

Service	Script
NameNode	/etc/init.d/hadoop-hdfs-namenode -hdfs1
DataNode	/etc/init.d/hadoop-hdfs-datanode -hdfs1
JournalNode	/etc/init.d/hadoop-hdfs-journalnode -hdfs1
ZKFailoverController	/etc/init.d/hadoop-hdfs-zkfc -hdfs1
ZooKeeper	/etc/init.d/zookeeper-server -zookeeper1

注： 红色字段需根据实际的运行服务而定

➤ NameNode (格式化或恢复)

```
# hdfs namenode [-format [-clustered cid] [-force] [-nonInteractive] ] | [-recover [-force] ]
```

Command Options	Description
-format [-clusterid cid] [-force] [-nonInteractive]	Formats the specified NameNode. It starts the NameNode, formats it and then shut it down. -force option formats if the name directory exists. -nonInteractive option aborts if the name directory exists, unless -force option is specified.
-recover [-force]	Recover lost metadata on a corrupt filesystem.

➤ Report（报告文件系统信息）

```
# hdfs dfsadmin [generic_options] [-report [-live] [-dead] [-decommissioning] ]
```

Command Options	Description
-report [-live] [-dead] [-decommissioning]	Reports basic filesystem information and statistics. Optional flags may be used to filter the list of displayed DataNodes.

```
Configured Capacity: 62396276736 (58.11 GB)
Present Capacity: 62396276736 (58.11 GB)
DFS Remaining: 57935630336 (53.96 GB)
DFS Used: 4460646400 (4.15 GB)
DFS Used%: 7.15%
Under replicated blocks: 36
Blocks with corrupt replicas: 0
Missing blocks: 0

-----
Live datanodes (3):

Name: 172.16.2.84:50010 (t3126poc4)
Hostname: t3126poc4
Rack: /Default
Decommission Status : Normal
Configured Capacity: 20798758912 (19.37 GB)
DFS Used: 1486884864 (1.38 GB)
Non DFS Used: 0 (0 B)
DFS Remaining: 19311874048 (17.99 GB)
DFS Used%: 7.15%
DFS Remaining%: 92.85%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 10
Last contact: Wed Apr 13 12:50:16 CST 2016
```

➤ Fsck (检查文件系统健康状况)

```
# hdfs fsck <path> [-move | -delete] | [-files [-blocks [-locations | -racks] ] ]
```

Command Options	Description
path	Start checking from this path.
-delete	Delete corrupted files.
-files	Print out files being checked.
-files -blocks	Print out the block report
-files -blocks -locations	Print out locations for every block.
-files -blocks -racks	Print out network topology for data-node locations.
-move	Move corrupted files to /lost+found.

➤ Fsck (检查文件系统健康状况)

```
t3126poc4:~ # sudo -u hdfs hdfs fsck /tmp
2016-04-13 12:57:30,365 WARN ssl.FileBasedKeyStoresFactory: The property 'ssl.client.truststore.loc
Connecting to namenode via http://t3126poc4:50070
FSCK started by hdfs (auth:SIMPLE) from /172.16.2.84 for path /tmp at Wed Apr 13 12:57:31 CST 2016
.....Status: HEALTHY
Total size:      496457669 B
Total dirs:      6
Total files:     12
Total symlinks:   0
Total blocks (validated): 13 (avg. block size 38189051 B)
Minimally replicated blocks: 13 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 0 (0.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 3
Average block replication: 3.0
Corrupt blocks: 0
Missing replicas: 0 (0.0 %)
Number of data-nodes: 3
Number of racks: 1
FSCK ended at Wed Apr 13 12:57:31 CST 2016 in 2 milliseconds

The filesystem under path '/tmp' is HEALTHY
```


➤ Safemode (安全模式)

- NameNode启动会自动进入安全模式（也支持手动进入），该模式下只支持读操作
- 检测Block上报率超过阈值，才会离开安全模式
- 在TDH中，为避免用户错误退出安全模式，增加了检查变量，只有设置变量后，命令才可以正确执行
- 慎用 **hdfs dfsadmin leave**

```
# hdfs dfsadmin [generic_options] [-safemode enter | leave | get | wait]
```

Note: Safe mode maintenance command. Safe mode is a Namenode state in which it

1. does not accept changes to the name space (read-only)
2. does not replicate or delete blocks.

Safe mode is entered automatically at Namenode startup, and leaves safe mode automatically when the configured minimum percentage of blocks satisfies the minimum replication condition. Safe mode can also be entered manually, but then it can only be turned off manually as well.

➤ NameNode HA（主备切换）

```
# hdfs haadmin -failover [--forcefence] [--forceactive] <serviceId> <serviceId>
# hdfs haadmin -getServiceState <serviceId>
# hdfs haadmin -transitionToActive <serviceId> [--forceactive]
# hdfs haadmin -transitionToStandby <serviceId>
```

Command Options	Description
-failover	initiate a failover between two NameNodes
-getServiceState	determine whether the given NameNode is Active or Standby
-transitionToActive	transition the state of the given NameNode to Active
-transitionToStandby	transition the state of the given NameNode to Standby

➤ Decommission or Recommission (DataNode退役和服役)

```
# hdfs dfsadmin [generic_options] -refreshNodes
```

Notes: Re-read the hosts and exclude files to update the set of Datanodes that are allowed to connect to the Namenode and those that should be decommissioned or recommissioned.

Command Options	Description
dfs.hosts	Names a file that contains a list of hosts that are permitted to connect to the namenode. The full pathname of the file must be specified. If the value is empty, all hosts are permitted.
dfs.hosts.exclude	Names a file that contains a list of hosts that are not permitted to connect to the namenode. The full pathname of the file must be specified. If the value is empty, no hosts are excluded.

DataNode退役的基本步骤:

- 1. 将计划退役的DataNode列表加入dfs.hosts.exclude文件
- 2. `hadoop dfsadmin -refreshNodes`
- 3. 等待一段时间，这组DataNode的状态由Inservice变为Decommission
- 4. 将这组DataNode从dfs.hosts文件中删除
- 5. `hadoop dfsadmin -refreshNodes`

➤ Decommission or Recommission (DataNode退役和服役)

• 退役和服役 (Web)

<input type="checkbox"/> data node (t3126poc4)	t3126poc4	/Default	Link	● Running	▶ ◻ ▲ ▼ ✕
<input type="checkbox"/> data node (t3126poc5)	t3126poc5	/Default	Link	● Running	▶ ◻ ▲ ▼ ✕
<input type="checkbox"/> data node (t3126poc6)	t3126poc6	/Default	Link	● Running	▶ ◻ ▲ ▼ ✕

• 删除DataNode (先退役再删除)

<input type="checkbox"/> data node (t3126poc4)	t3126poc4	/Default	Link	● Running	▶ ◻ ▲ ▼ ✕
<input type="checkbox"/> data node (t3126poc5)	t3126poc5	/Default	Link	● Running	▶ ◻ ▲ ▼ ✕
<input type="checkbox"/> data node (t3126poc6)	t3126poc6	/Default	Link	● Running	▶ ◻ ▲ ▼ ✕

➤ Balancer (数据重分布)

```
# hdfs balancer [-threshold <threshold>]
                  [-exclude [-f <hosts-file> | <comma-separated list of hosts>] ]
                  [-include [-f <hosts-file> | <comma-separated list of hosts>] ]
```

Command Options	Description
-threshold <threshold>	Percentage of disk capacity. This overwrites the default threshold.
-exclude -f <hosts-file> <comma-separated list of hosts>	Excludes the specified datanodes from being balanced by the balancer.
-include -f <hosts-file> <comma-separated list of hosts>	Includes only the specified datanodes to be balanced by the balancer.

➤ Balancer (数据重分布)

- 集群平衡的标准：每个DataNode的存储使用率和集群总存储使用率的差值均小于阈值
- 默认阈值为10，设置值为0~100

```
t3126poc4:~ # sudo -u hdfs hdfs balancer
2016-04-13 13:39:40,732 INFO balancer.Balancer: namenodes = [hdfs://nameservice1]
2016-04-13 13:39:40,733 INFO balancer.Balancer: p = Balancer.Parameters[BalancingPolicy.Node, threshold=10.0]
Time Stamp      Iteration#  Bytes Already Moved  Bytes Left To Move  Bytes Being Moved
2016-04-13 13:39:41,630 INFO net.NetworkTopology: Adding a new node: /Default/172.16.2.84:50010
2016-04-13 13:39:41,631 INFO net.NetworkTopology: Adding a new node: /Default/172.16.2.86:50010
2016-04-13 13:39:41,631 INFO net.NetworkTopology: Adding a new node: /Default/172.16.2.85:50010
2016-04-13 13:39:41,631 INFO balancer.Balancer: 0 over-utilized: []
2016-04-13 13:39:41,631 INFO balancer.Balancer: 0 underutilized: []
The cluster is balanced. Exiting...
Apr 13, 2016 1:39:41 PM Balancing took 1.355 seconds
```


➤ BalancerBandwidth

- 默认带宽为1M/s，主要为了Balance的同时不影响HDFS操作
- 建议Balance的时候，带宽设为10M/s，并且停止操作HDFS

```
# hdfs dfsadmin [generic_options] [-setBalancerBandwidth <bandwidth in bytes per second>]
```

Command Options	Description
<code>-setBalancerBandwidth</code> <code><bandwidth in bytes per second></code>	Changes the network bandwidth used by each datanode during HDFS block balancing. <code><bandwidth></code> is the maximum number of bytes per second that will be used by each datanode. This value overrides the <code>dfs.balance.bandwidthPerSec</code> parameter. NOTE: The new value is not persistent on the DataNode.

```
t3126poc4:~ # sudo -u hdfs hdfs dfsadmin -setBalancerBandwidth 10
Balancer bandwidth is set to 10 for t3126poc4/172.16.2.84:8020
Balancer bandwidth is set to 10 for t3126poc5/172.16.2.85:8020
```

➤ Distcp (分布式拷贝)

- 大规模集群内部和集群之间拷贝的工具
- 使用MapReduce实现文件分发、错误处理恢复，以及报告生成

```
# hadoop distcp options [source_path...] <target_path>
```

Notes: distcp (distributed copy) is a tool used for large inter/intra-cluster copying. It uses MapReduce to effect its distribution, error handling and recovery, and reporting.

Command Options	Description
-m <num_maps>	Maximum number of simultaneous copies
-overwrite	Overwrite destination
-bandwidth	Specify bandwidth per map, in MB/second.

➤ Quota (配额限制)

- HDFS允许管理员对用户的目录设置Quota，主要从两个维度：文件数量和文件大小
- 限制指定目录及子目录中的文件总数
- 限制指定目录中的所有文件的容量大小，需要考虑副本数

```
# hdfs dfsadmin -setSpaceQuota <N> <directory>...<directory>
```

Notes: Set the space quota to be N bytes for each directory.

```
# hdfs dfsadmin -clrSpaceQuota <directory>...<directory>
```

Notes: Remove any space quota for each directory.

```
# hadoop fs -count -q [-h] [-v] <directory>...<directory>
```

Notes: With the -q option, also report the name quota value set for each directory, the available name quota remaining, the space quota value set, and the available space quota remaining. The -h option shows sizes in human readable format. The -v option displays a header line.

➤ Quota (配额限制)

- 示例: `hadoop fs -count -q`

– 输出: 数量quota | 数量剩余 | 空间quota | 空间剩余 | 目录数量 | 文件数量 | 目录逻辑空间大小 | 路径

```
t3126poc4:~ # sudo -u hdfs hdfs dfs -mkdir /name_quota
t3126poc4:~ # sudo -u hdfs hdfs dfs -mkdir /space_quota
t3126poc4:~ # sudo -u hdfs hdfs dfsadmin -setQuota 100 /name_quota
t3126poc4:~ # sudo -u hdfs hdfs dfsadmin -setSpaceQuota 10g /space_quota
t3126poc4:~ # sudo -u hdfs hdfs dfs -count -q /name_quota
    100          99      none      inf          1          0          0 /name_quota
t3126poc4:~ # sudo -u hdfs hdfs dfs -count -q /space_quota
    none      inf  10737418240  10737418240          1          0          0 /space_quota
```

➤ Snapshot (快照)

- HDFS快照是只读的，记录文件系统在某个时间点的副本
- HDFS快照可应用于根目录或其他子目录

```
# hdfs lsSnapshottableDir
```

Notes: Get all the snapshottable directories where the current user has permission to take snapshots.

```
# hdfs snapshotDiff <path> <fromSnapshot> <toSnapshot>
```

Notes: Get the differences between two snapshots. This operation requires read access privilege for all files/directories in both snapshots.

```
# hdfs dfsadmin -allowSnapshot <path>
```

Notes: Allowing snapshots of a directory to be created.

```
# hdfs dfsadmin -disallowSnapshot <path>
```

Notes: Disallowing snapshots of a directory to be created.

➤ Snapshot (快照)

- 创建好的Snapshot文件夹在源文件夹下，命名为.snapshot/[<snapshotName>]
- 恢复的时候，直接使用cp命令即可

```
# hdfs dfs -createSnapshot <path> [<snapshotName>]
```

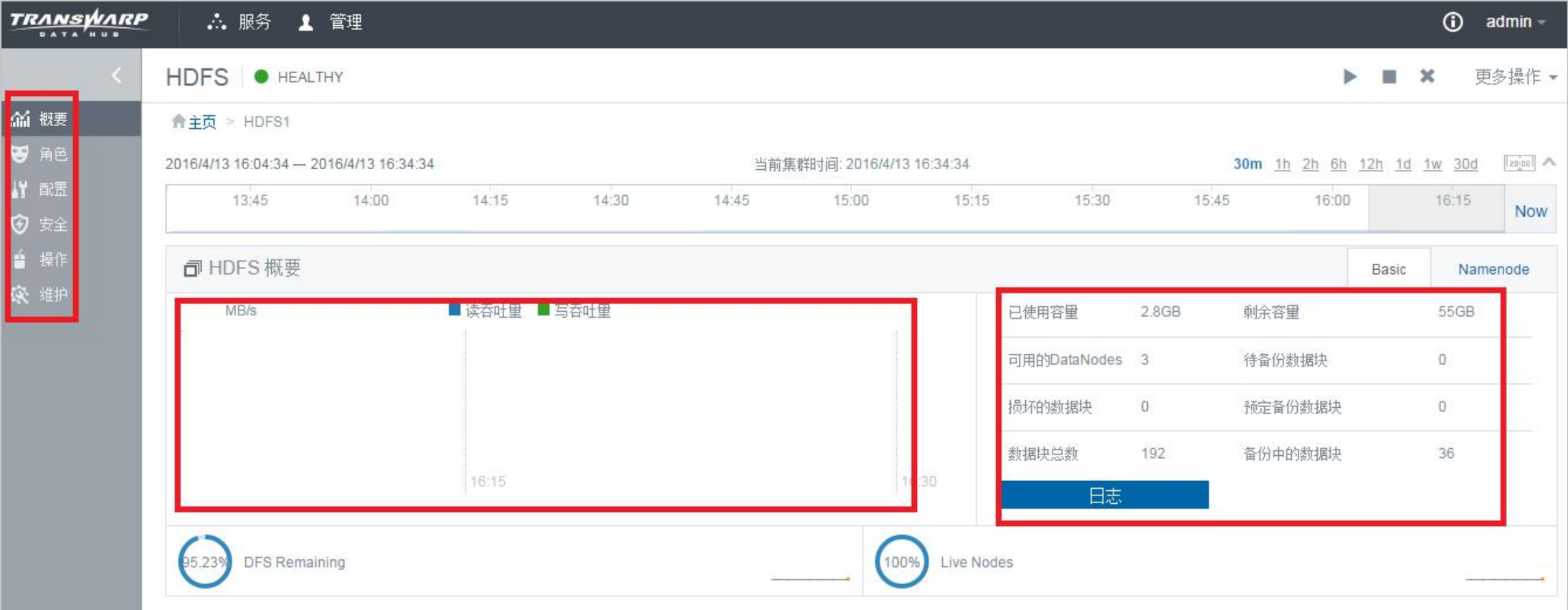
Notes: Create a snapshot of a snapshottable directory. This operation requires owner privilege of the snapshottable directory.

```
# hdfs dfs -deleteSnapshot <path> <snapshotName>
```

Notes: Delete a snapshot of from a snapshottable directory. This operation requires owner privilege of the snapshottable directory.

```
t3126poc4:~ # hdfs dfs -mkdir /tmp/test_snapshot
t3126poc4:~ # hdfs dfs -put /root/wy/koalas/scp.sh /tmp/test_snapshot/
t3126poc4:~ # sudo -u hdfs hdfs dfsadmin -allowSnapshot /tmp/test_snapshot
Allowing snapshot on /tmp/test_snapshot succeeded
t3126poc4:~ # sudo -u hdfs hdfs dfs -createSnapshot /tmp/test_snapshot bak1
Created snapshot /tmp/test_snapshot/.snapshot/bak1
t3126poc4:~ # hdfs dfs -rm /tmp/test_snapshot/scp.sh
2016-04-13 14:09:04,185 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval
Moved: 'hdfs://nameservice1/tmp/test_snapshot/scp.sh' to trash at: hdfs://nameservice1/user/root/.Trash
t3126poc4:~ # hdfs dfs -ls /tmp/test_snapshot/
t3126poc4:~ # sudo -u hdfs hdfs dfs -cp /tmp/test_snapshot/.snapshot/bak1/* /tmp/test_snapshot/
t3126poc4:~ # hdfs dfs -ls /tmp/test_snapshot/
Found 1 items
-rw-r--r--  3 hdfs hadoop      339 2016-04-13 14:10 /tmp/test_snapshot/scp.sh
```


➤ TDH WebUI



➤ Active NameNode WebUI

- <http://activeNameNodeHost:50070>

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Overview 't3126poc4:8020' (active)

Started:	Tue Apr 12 15:45:42 CST 2016
Version:	2.5.2-transwarp, rUnknown
Compiled:	2016-02-04T00:06Z by root from Unknown
Cluster ID:	hdfs1
Block Pool ID:	BP-1057937095-172.16.2.84-1455607990045

Summary

Security is off.

Safemode is off.

464 files and directories, 192 blocks = 656 total filesystem object(s).

Heap Memory used 227.44 MB of 381.5 MB Heap Memory. Max Heap Memory is 3.56 GB.

Non Heap Memory used 43.88 MB of 84.19 MB Committed Non Heap Memory. Max Non Heap Memory is 130 MB.



Q&A

TRANSWARP
星环科技

温故知新

- HDFS架构中包含哪几种角色？各自承担什么功能？
- 为什么HDFS不合适存储大量的小文件？
- Block副本的放置策略是什么？如何理解？
- HDFS离开安全模式的条件是什么？
- HDFS是如何实现高可用的？

