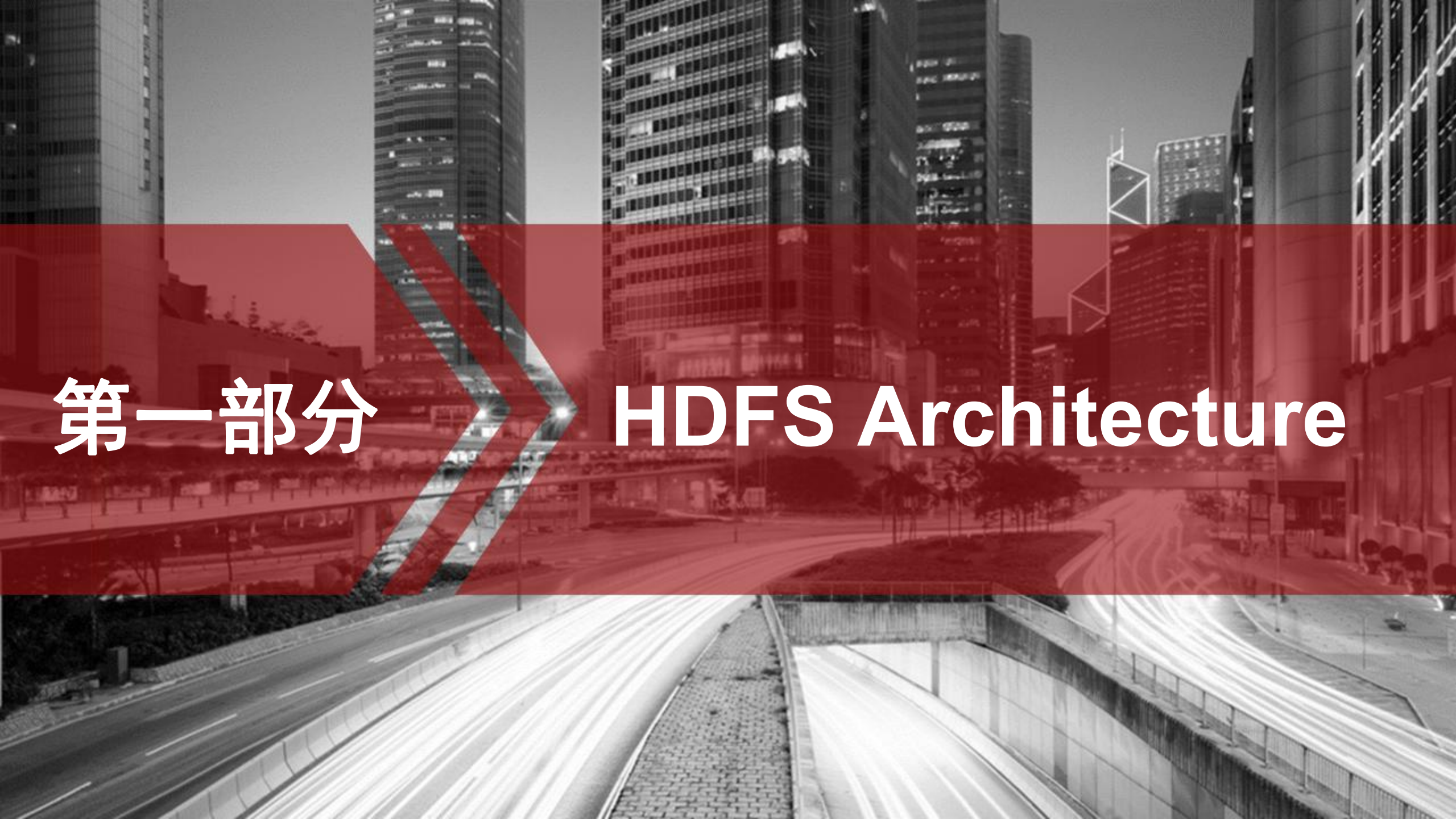
HDFS

云尘

要求及希望达到的目标

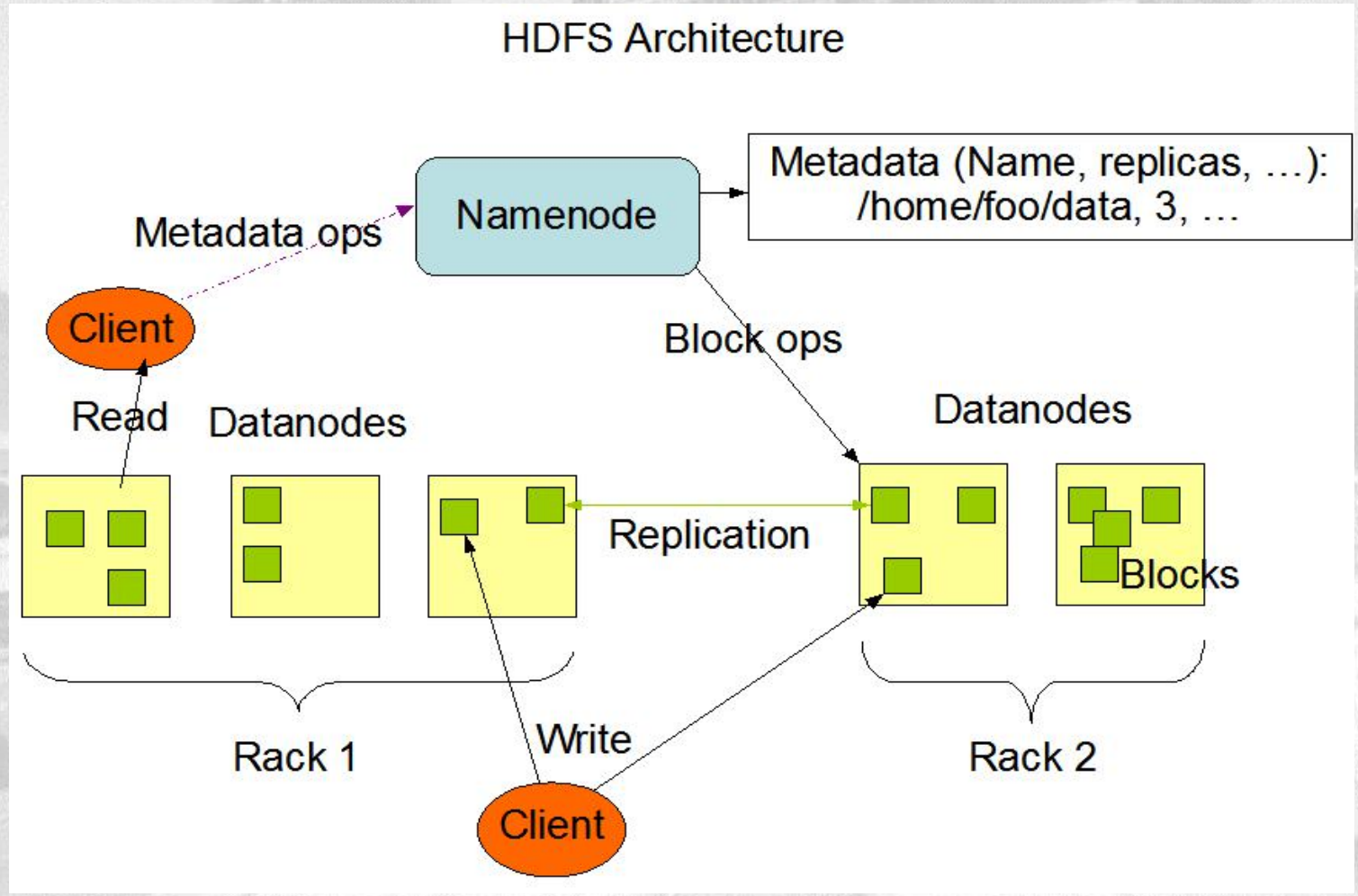
- 要求
  - 对hdfs有一定了解，会基本操作
  - 了解其架构就最好了
  - 有一定的linux基础知识
- 目标
  - 对HDFS有一个全局的认识
  - 理解HDFS架构以及基本功能的使用及适用场景
  - 理解HA、Federation等方案
  - 理解HDFS权限，要知道任务应权限运行失败了该怎么解决

目录 content



第一部分 HDFS Architecture

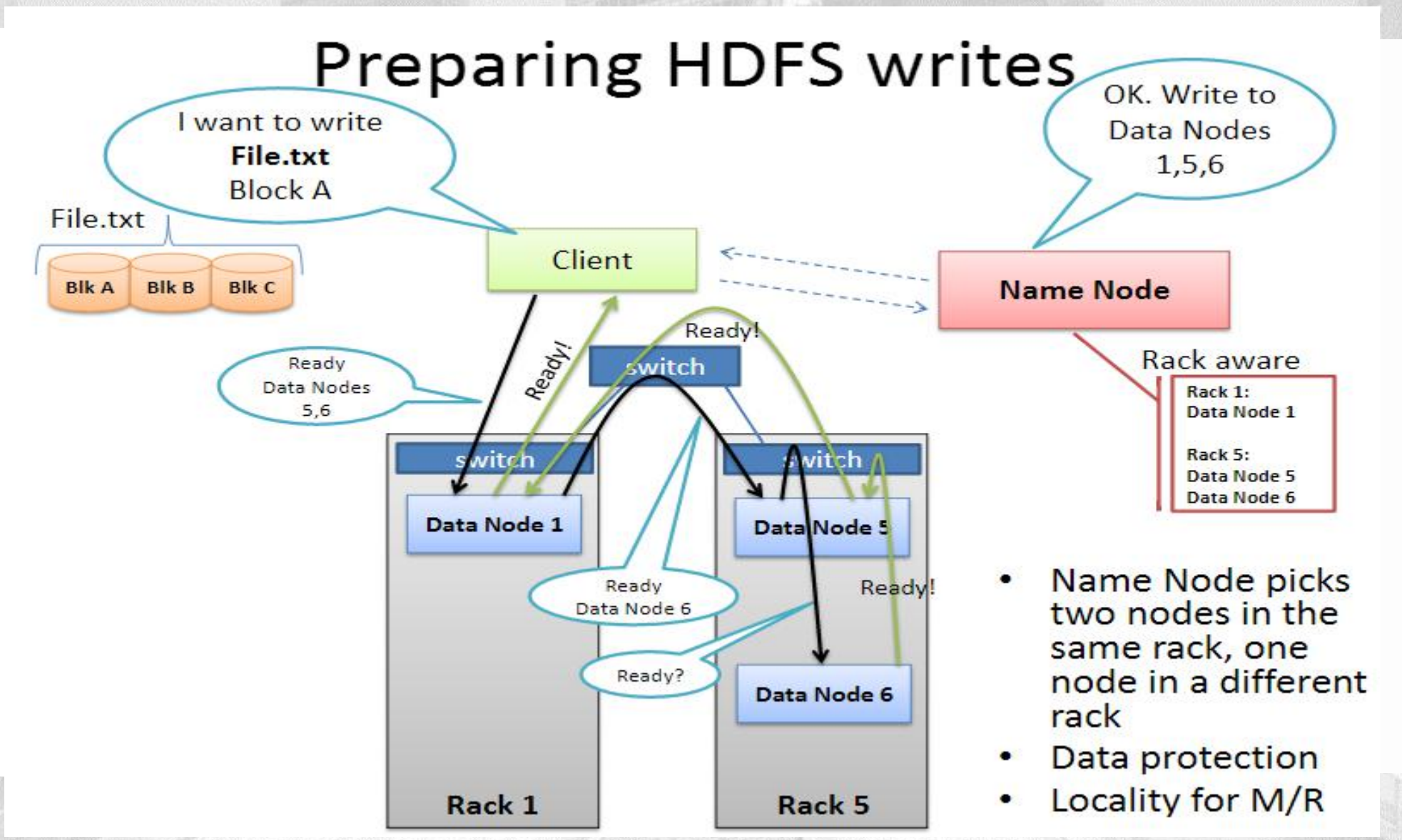
HDFS Architecture



HDFS Architecture

- NameNode:master节点，管理所有的存储在内存中的文件系统元数据。这些元数据包括名字空间、访问控制信息、文件和DataNode的映射信息、以及当前DataNode的位置信息
- DataNode: Slave节点，存储实际的数据；执行数据块的读写；汇报存储信息给NameNode
- Fsimage:元数据存储文件
- Editlog:记录所有更新操作的文件

HDFS Architecture



- Name Node picks two nodes in the same rack, one node in a different rack
- Data protection
- Locality for M/R

第二部分 HDFS Function Outline

第二部分：HDFS Outline

- High Availability With QJM/NFS
- Federation
- ViewFs
- HDFS Snapshots
- HDFS Architecture
- Edits Viewer
- Image Viewer
- Permissions and HDFS
- Quotas and HDFS
- HFTP
- WebHDFS REST API

- HttpFS Gateway
- Short Circuit Local Reads
- Centralized Cache Management
- HDFS NFS Gateway
- Transparent Encryption
- HDFS Support for Multihoming
- Archival Storage, SSD & Memory
- Memory Storage Support

ViewFs

- ViewFs 文件浏览系统
- ViewFs与Unix/Linux系统中client side mount tables类似
- 管理多个Hadoop文件系统命名空间
- 它对于有多个NameNode的联邦集群特别有用

HDFS Snapshots

- HDFS快照是一个只读的基于时间点文件系统拷贝

```
[root@cmagent1 ~]# hdfs dfs -ls /out
Found 1 items
-rwxrwxr-x+  1 root root        273 2016-07-06 17:42 /out/net

[root@cmagent1 ~]# sudo -u hdfs hdfs dfsadmin -allowSnapshot /out
Allowi

[root@
Create

[root@
[root@
16/07/
trash

[root@
drwxr-

[root@
Create

[root@
Differ
M
+    ./
-    ./
```

```
[root@cmagent1 ~]# hdfs dfs -deleteSnapshot /out  s20160708-105229.089

[root@cmagent1 ~]# sudo -u hdfs hdfs dfsadmin -disallowSnapshot /out
disallowSnapshot: The directory /out has snapshot(s). Please redo the operation
after removing all the snapshots.

[root@cmagent1 ~]# hdfs dfs -ls /out/.snapshot
Found 1 items
drwxr-xr-x   - root supergroup          0 2016-07-08 10:55 /out/.snapshot/s20160708-105523.336

[root@cmagent1 ~]# hdfs lsSnapshottableDir
drwxr-xr-x 0 root supergroup 0 2016-07-08 10:55 1 65536 /out

[root@cmagent1 ~]# hdfs dfs -rm -r -f /out/.snapshot/*
16/07/08 11:00:22 WARN fs.TrashPolicyDefault: Can't create trash directory:
 hdfs://jimmy/user/root/.Trash/Current/out/.snapshot

[root@cmagent1 ~]# hdfs dfs -deleteSnapshot /out  s20160708-105523.336
[root@cmagent1 ~]# hdfs dfs -ls /out/.snapshot
[root@cmagent1 ~]# sudo -u hdfs hdfs dfsadmin -disallowSnapshot /out
Disallowing snaphot on /out succeeded
[root@cmagent1 ~]# hdfs lsSnapshottableDir
```

- 比较 - hdfs snapshotDiff <path> <fromSnapshot> <toSnapshot>
- [Java API]

Edits Viewer

- C
-
-

```
<RECORD>
  <OPCODE>OP_ADD</OPCODE>
  <DATA>
    <TXID>2965286</TXID>
    <LENGTH>0</LENGTH>
    <INOD
    <PATH                                                    ATH>
    <REPL
    <MTIM
    <ATIM
    <BLOC
    <CLIE
    <CLIE
    <OVER
    <PERM
      <US
      <GROUPNAME>spark</GROUPNAME>
      <MODE>420</MODE>
    </PERMISSION_STATUS>
    <RPC_CLIENTID>c85f90b9-ca35-4dad-bdd8-4f0d7a7b5a73</RPC_CLIENTID>
    <RPC_CALLID>9441</RPC_CALLID>
  </DATA>
</RECORD>
```

Flag	Description
[-i ; --inputFile] *input file*	必填项，指定edit日志。当扩展名为xml时是xml格式否则为二进制格式。
[**-o ; --outputFile**]*output file*	必填项，指定输出文件如果已存在，那么会被重写。
[**-p ; --processor**]*processor*	指定运行的解释器，目前允许的选项有binary, xml (默认) 和 stats.
[-v ; --verbose]	打印输入输出文件到控制台和指定的文件，如果比较大，会需要很多时间。
[-h ; --help]	显示帮助信息

Image Viewer

- Offline Image Viewer是一个将fsimage的内容转换为方便阅读的工

Flag	Description
-i u007C--inputFile *input file*	必填。输入的文件
-o u007C--outputFile *output file*	必填。输出文件。
-p u007C--processor *processor*	指明处理器类型，现在可以是Ls (default), XML 和Indented..
-skipBlocks	不列举文件中的块。这将节省处理时间和输出的文件大小，Ls处理器读取块确定
-printToScreen	输出到控制台和文件。
-delimiter *arg*	结合使用分隔的处理器时,替换默认选项卡指定的分隔符的字符串参数。
-h u007C--help	help

Quotas

```
[root@cmagent1 ~]# hdfs dfs -mkdir /quotas3
[root@cmagent1 ~]#
[root@cmagent1 ~]# hdfs dfsadmin -setQuota 50 /quotas3
[root@cmagent1 ~]#
[root@cmagent1 ~]# hdfs dfs -put * /quotas3
put: The NameSpace quota (directories and files) of directory /quotas3 is exceeded: quota=50 file count=51
put: The NameSpace quota (directories and files) of directory /quotas3 is exceeded: quota=50 file count=51
put: The NameSpace quota (directories and files) of directory /quotas3 is exceeded: quota=50 file count=51
put: The NameSpace quota (directories and files) of directory /quotas3 is exceeded: quota=50 file count=51
put: The NameSpace quota (directories and files) of directory /quotas3 is exceeded: quota=50 file count=51
[root@cmagent1 ~]#
[root@cmagent1 ~]# hdfs dfs -count -q -v /quotas3
       QUOTA       REM_QUOTA     SPACE_QUOTA REM_SPACE_QUOTA    DIR_COUNT   FILE_COUNT       CONTENT_SIZE PATHNAME
          50               0            none             inf           15           35           60329888 /quotas3
```

- Space Quotas – 文件类的大小

```
[root@cmagent1 ~]# hdfs dfs -mkdir /quotas4
[root@cmagent1 ~]#
[root@cmagent1 ~]# hdfs dfsadmin -setSpaceQuota 50M /quotas4
[root@cmagent1 ~]# hdfs dfs -put * /quotas4
put: The DiskSpace quota of /quotas4 is exceeded: quota = 52428800 B = 50 MB but diskspace consumed = 134217728 B = 128 MB
put: The DiskSpace quota of /quotas4 is exceeded: quota = 52428800 B = 50 MB but diskspace consumed = 134217728 B = 128 MB
put: The DiskSpace quota of /quotas4 is exceeded: quota = 52428800 B = 50 MB but diskspace consumed = 134217728 B = 128 MB
[root@cmagent1 ~]# hdfs dfs -count -q -v /quotas4
       QUOTA       REM_QUOTA     SPACE_QUOTA REM_SPACE_QUOTA    DIR_COUNT   FILE_COUNT       CONTENT_SIZE PATHNAME
        none             inf        52428800        52428800           22            1                  0 /quotas4
```

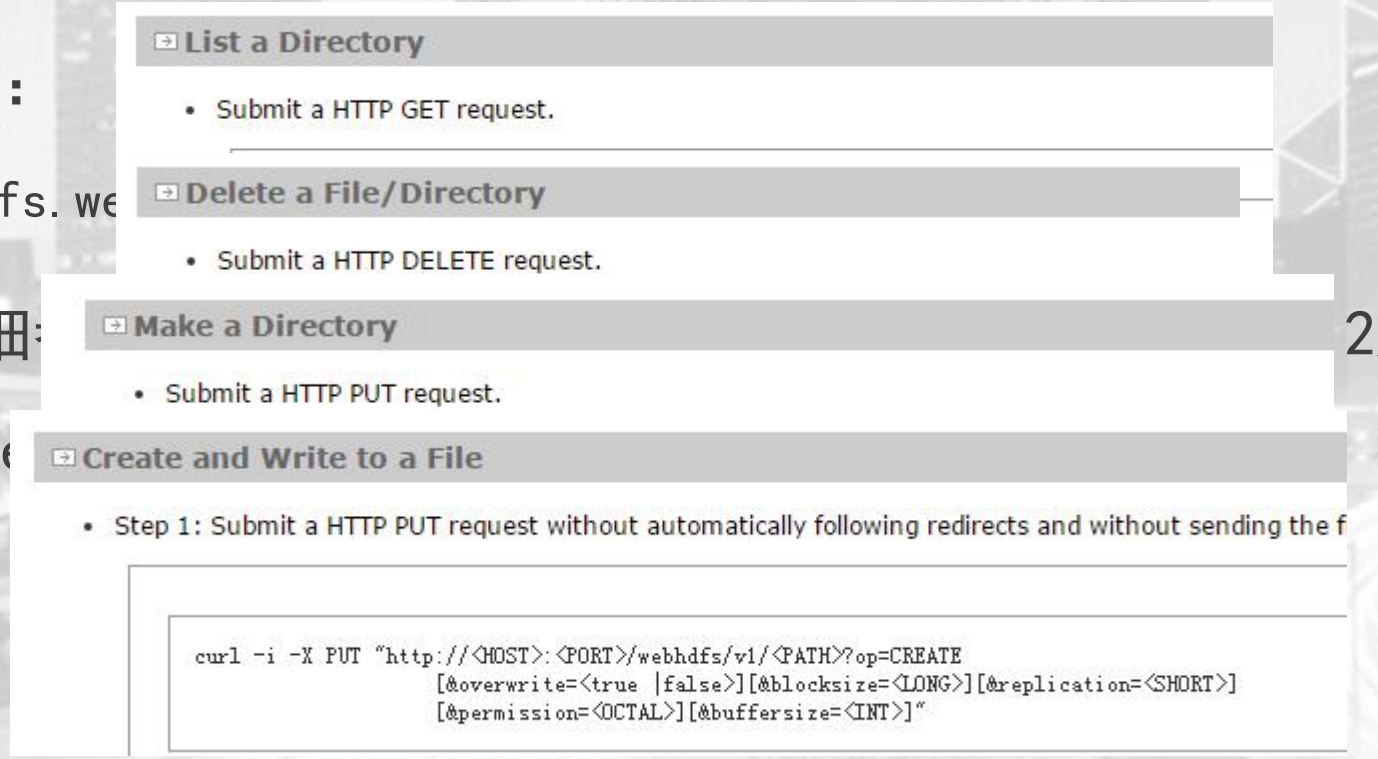
HFTP

- HFTP 是hadoop文件系统用来让你在两个hadoop HDFS集群之间互传数据的组件

```
[root@cmserver ~]# hadoop distcp -i hftp://dser2:50070/all hdfs://cg1:8020/dest
16/06/13 10:17:44 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, deleteMissing=false, ignoreFa
ilures=true, maxMaps=20, sslConfigurationFile='null', copyStrategy='uniformsize', sourceFileListing=null, sourcePaths=[hftp://dser2:5
0070/all], targetPath=hdfs://cg1:8020/dest, targetPathExists=true, preserveRawXattrs=false, filtersFile='null'}
16/06/13 10:17:44 INFO client.RMProxy: Connecting to ResourceManager at cmagent1/192.168.100.101:8032
16/06/13 10:17:46 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 6; dirCnt = 1
16/06/13 10:17:46 INFO tools.SimpleCopyListing: Build file listing completed.
16/06/13 10:17:46 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
16/06/13 10:17:46 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
16/06/13 10:17:46 INFO tools.DistCp: Number of paths in the copy list: 6
16/06/13 10:17:46 INFO tools.DistCp: Number of paths in the copy list: 6
16/06/13 10:17:46 INFO client.RMProxy: Connecting to ResourceManager at cmagent1/192.168.100.101:8032
16/06/13 10:17:46 INFO mapreduce.JobSubmitter: number of splits:6
16/06/13 10:17:47 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1465724197934_0016
16/06/13 10:17:47 INFO impl.YarnClientImpl: Submitted application application_1465724197934_0016
16/06/13 10:17:47 INFO mapreduce.Job: The url to track the job: http://cmagent1:8088/proxy/application_1465724197934_0016/
16/06/13 10:17:47 INFO tools.DistCp: DistCp job-id: job_1465724197934_0016
16/06/13 10:17:47 INFO mapreduce.Job: Running job: job_1465724197934_0016
16/06/13 10:17:54 INFO mapreduce.Job: Job job_1465724197934_0016 running in uber mode : false
16/06/13 10:17:54 INFO mapreduce.Job:  map 0% reduce 0%
```

WebHDFS API

- WEBHDFS是一个基于REST的接口，基于HTTP操作，集成于hadoop中
- 开启：
  - dfs.we
- 详细 2/hadoop-proj
- 举例



HttpFs Gateway

- HttpFS是一个基于REST的接口，基于HTTP操作，是HDFS一个独立的服务

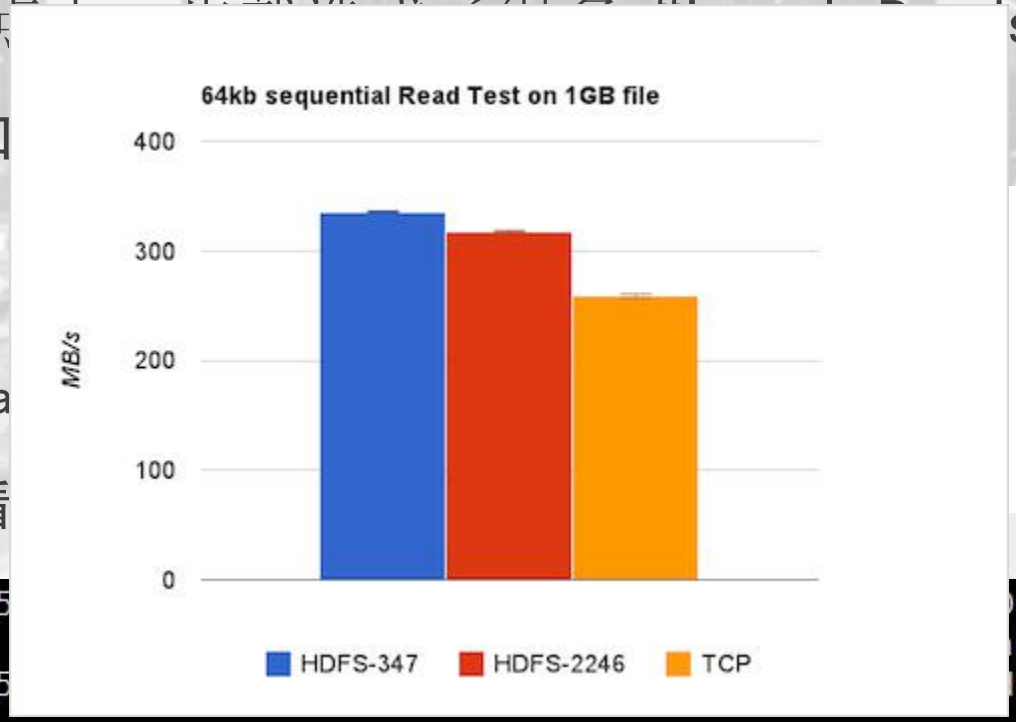
- `$ curl http://httpfs-host:14000/webhdfs/v1/user/foo/README.txt` returns the contents of the HDFS `/user/foo/README.txt` file.
- `$ curl http://httpfs-host:14000/webhdfs/v1/user/foo?op=list` returns the contents of the HDFS `/user/foo` directory in JSON format.
- `$ curl -X POST http://httpfs-host:14000/webhdfs/v1/user/foo/bar?op=mkdirs` creates the HDFS `/user/foo.bar` directory.

HttpFs vs WebHDFS

- 不同点：
  - WebHDFS是HortonWorks开发的，而HttpFS是Cloudera开发的
  - WebHDFS是HDFS内置的组件，已经运行于NameNode和DataNode中。HttpFS是独立于HDFS的一个服务
  - WebHDFS默认端口是50070和50075，HttpFS默认端口14000
- 相同点：
  - HDFS提供了Java Native API，客户端应用程序使用它可以高效的访问HDFS。但是如果客户端应用程序位于HDFS集群之外怎么办？
  - 它们的终极目标完全一致：就是让身处HDFS集群之外的应用程序，不但不用安装Hadoop和Java库，并且可以通过流行的REST风格的接口去访问HDFS集群

HDFS Short-Circuit Local Reads

- Hadoop通常是尽量移动计算到拥有数据的节点上，这就使得Hadoop中读取数据的客户端DFSClient和提供数据的Datanode经常是在一个节点上，此就谓之“Local Reads”，详细参考HDFS-2246和
- 配置：
  - 前提是需要Na
  - 执行命令查看
- 查




```
2016-06-13 14:23:35                                              ataNode.clienttrac
e: src: 127.0.0.1,                                               mId: 265cec3be0f69
1f9f2dabd2132d46465                                              f6699524e, success
: true
```

Centralized Cache Management

- HDFS中的集中式缓存管理机制
- 应用场景：
  - HBase in-memory table：可以直接把某个HBase表的HFile放到centralized cache中，这会显著提高HBase的读性能，降低读请求延迟。
  - 在Hive构建的数据仓库应用中fact表会频繁地与其他表做JOIN，可以让f
- 使用

```
[root@cmagent1 ~]# hdfs cacheadmin -addPool mem
Successfully added cache pool mem.
[root@cmagent1 ~]# hdfs cacheadmin -listPools
Found 1 result.
NAME  OWNER  GROUP  MODE            LIMIT  MAXTTL
mem   root   root   rwxr-xr-x   unlimited   never
[root@cmagent1 ~]# hdfs cacheadmin -addDirective -path /dest/all -pool mem
Added cache directive 1
[root@cmagent1 ~]# hdfs cacheadmin -listDirectives
Found 1 entry
 ID POOL   REPL EXPIRY  PATH
  1 mem       1 never   /dest/all
```

HDFS NFS Gateway

- NFS Gateway支持NFSv3，允许HDFS作为客户端本地文件系统的一部分挂载在本地文件系统
- 启动：hadoop-daemon.sh start nfs3
  - 如果用cdh启动，则需先安装rpcbind服务并启动
- 使

```
[root@cmagent1 ~]# showmount -e cg1
Export list for cg1:
/ *
[root@cmagent1 ~]# mount -t nfs -o vers=3,proto=tcp,nolock cg1:/ /mnt/
[root@cmagent1 ~]#
[root@cmagent1 ~]# ls /mnt/
2629all     flume           flume_2_40_all  quotas   quotas4    test_xc_xc  xxx
benchmarks  flume_1_40_1    flume_3_40_1    quotas1  smallfile  tmp
bigfile     flume_1_40_all  flume_4_40_1    quotas2  system     user
dest        flume_2_40_1    kafka           quotas3  test_xc    v1
[root@cmagent1 ~]#
```

Transparent Encryption

- HDFS实现透明的、端到端的加密，也就是从指定的HDFS读取和写入数据都会透明的进行加密和解密，不需要用户应用程序代码的变更
- 主要应用于静态加密（意思是数据在永久存储上，例如磁盘）以及在传输加密（例如当数据在网络中传输时）

HDFS Support for Multihoming

- 让集群支持混合网络
- 比如大数
- Securi
- Perfor
- Failover/Redundancy

用来进行数据

外部网络用来

千兆网络

```
<property>
  <name>dfs.namenode.rpc-bind-host</name>
  <value>0.0.0.0</value>
</property>

<property>
  <name>dfs.namenode.servicerpc-bind-host</name>
  <value>0.0.0.0</value>
</property>
<property>
  <name>dfs.namenode.http-bind-host</name>
  <value>0.0.0.0</value>
</property>
<property>
  <name>dfs.namenode.https-bind-host</name>
  <value>0.0.0.0</value>
</property>
<property>
  <name>dfs.client.use.datanode.hostname</name>
  <value>true</value>
</property>
<property>
  <name>dfs.datanode.use.datanode.hostname</name>
  <value>true</value>
</property>
```

第三部分 HDFS HA

Archival Storage, SSD & Memory

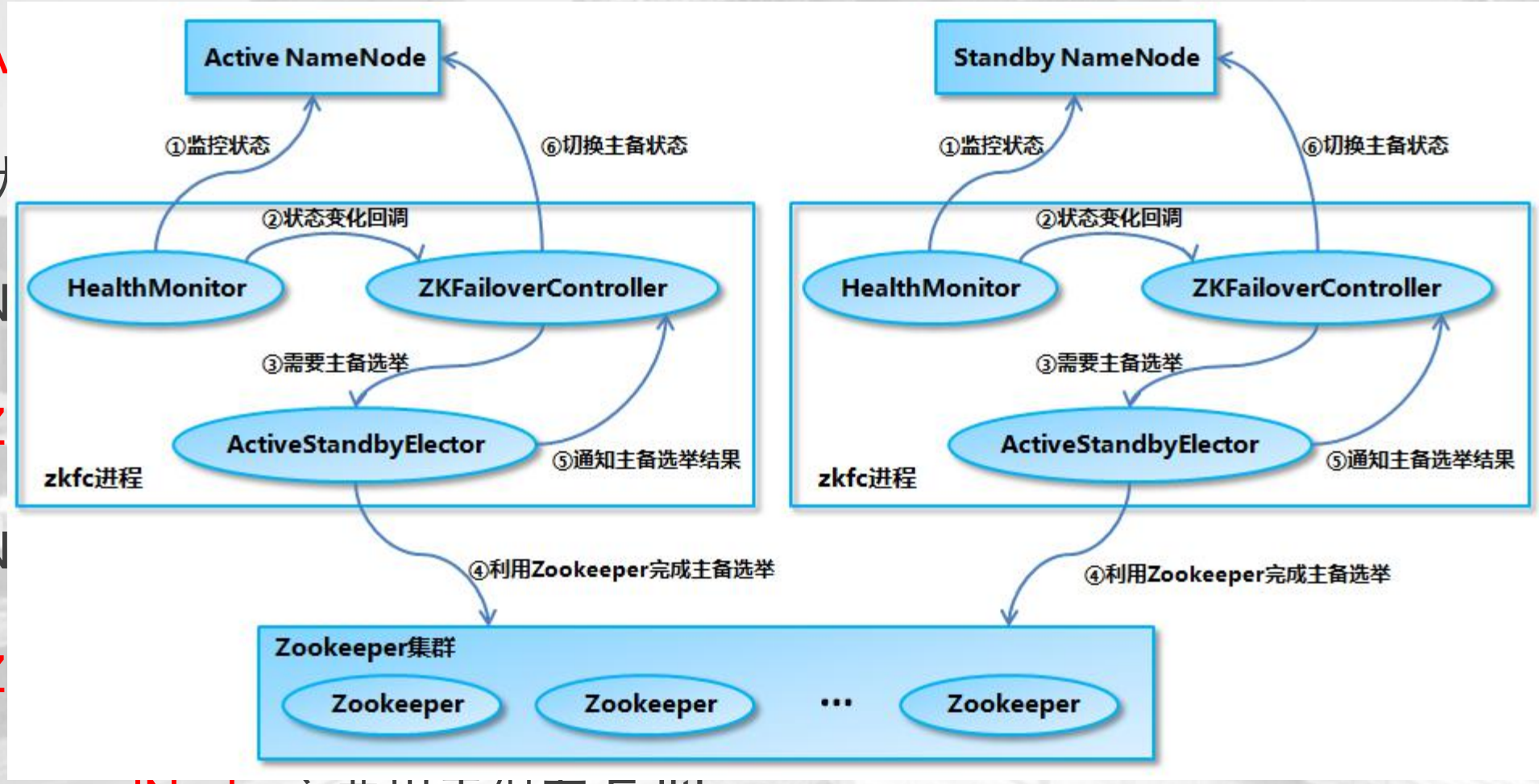
```
# hdfs dfs -mkdir /ramdisk

# hdfs dfs -put hdfs.sh /ramdisk
# hdfs fsck /ramdisk/hdfs.sh -files -blocks -locations
FSCK started by root (auth:SIMPLE) from /192.168.100.101 for path /ramdisk/hdfs.sh ...
/ramdisk/hdfs.sh 115 bytes, 1 block(s):  OK
0. BP-209661133-192.168.100.101-1462326215715:blk_1074177665_437506 len=115 Live_repl=1
[DatanodeInfoWithStorage[192.168.100.101:50010,DS-307815cc-1f96-472e-b49e-68916966e788,DISK]]

# find /dfs /sdd /sdb /mm -name "*1074177665*"
/dfs/dn/current/BP-209661133-192.168.100.101-1462326215715/
current/finalized/subdir6/subdir166/blk_1074177665

# hdfs dfsadmin -setStoragePolicy /ramdisk Lazy_Persist
Set storage policy Lazy_Persist on /ramdisk

# hdfs dfsadmin -getStoragePolicy /ramdisk
The storage policy of /ramdisk:
BlockStoragePolicy{LAZY_PERSIST:15, storageTypes=[RAM_DISK, DISK],
creationFallbacks=[DISK], replicationFallbacks=[DISK]}

# hdfs dfs -put move.sh /ramdisk
# hdfs fsck /ramdisk/move.sh -files -blocks -locations
....
Under replicated BP-1685153915-10.10.10.89-1467739876672:blk_1073741826_1002.
0. BP-1685153915-10.10.10.89-1467739876672:blk_1073741826_1002 len=973 repl=1

....
# find /dfs /mnt -name "*1073741826*" |egrep -v meta
/dfs/current/BP-1685153915-10.10.10.89-1467739876672/current/lazypersist/subdir0/subdir0/blk_1073741826
/mnt/ramdisk/current/BP-1685153915-10.10.10.89-1467739876672/current/finalized/subdir0/subdir0/blk_107374182
```

HDFS HA

- 解决了namenode单点故障问题
- 常用HA方案
  - 基于NFS HA方案
  - **基于QJM** HA方案
  - 基于Bookeeper HA方案
  - 借助DRBD、HeartbeatHA实现主备切换 HA方案

HDFS HA

- A
- Z
- Z
- JournalNode:主要用于保存 EditLog



HDFS HA

core-site.xml

```
<property>
  <name>fs.defaultFS
  <value>hdfs://mycl
</property>
<property>
  <name>dfs.ha.auton
  <value>true</value
</property>
<property>
  <name>ha.zookeeper
  <value>zk1.example
</property>
```

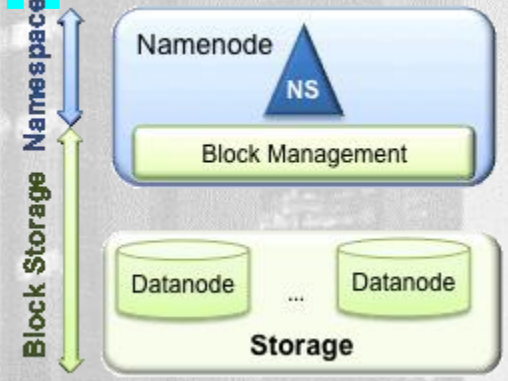
hdfs-site.xml

```
<property>
  <name>dfs.nameservices</name>
  <value>mycluster</value>
</property>
<property>
  <name>dfs.ha.namenodes.mycluster</name>
  <value>nn1,nn2</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.mycluster.nn1</n
  <value>nn1:8020</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.mycluster.nn2</n
  <value>nn2:8020</value>
</property>
<property>
  <name>dfs.namenode.http-address.mycluster.nn1</
  <value>nn1:50070</value>
</property>
<property>
  <name>dfs.namenode.http-address.mycluster.nn2</
  <value>nn2:50070</value>
</property>
<property>
  <name>dfs.namenode.shared.edits.dir</name>
  <value>qjournal://node1:8485;node2:8485;node3:8485/mycluster</value>
</property>
```

```
<property>
    <name>ha.zookeeper.session-timeout.ms</name>
    <value>5000</value>
</property>
<property>
  <name>dfs.ha.fencing.methods</name>
  <value>sshfence</value>
</property>
<property>
  <name>dfs.ha.fencing.ssh.private-key-files</name>
  <value>/home/hdfs/.ssh/id_rsa</value>
</property>
```

第四部分 HDFS Federation

HDFS Federation

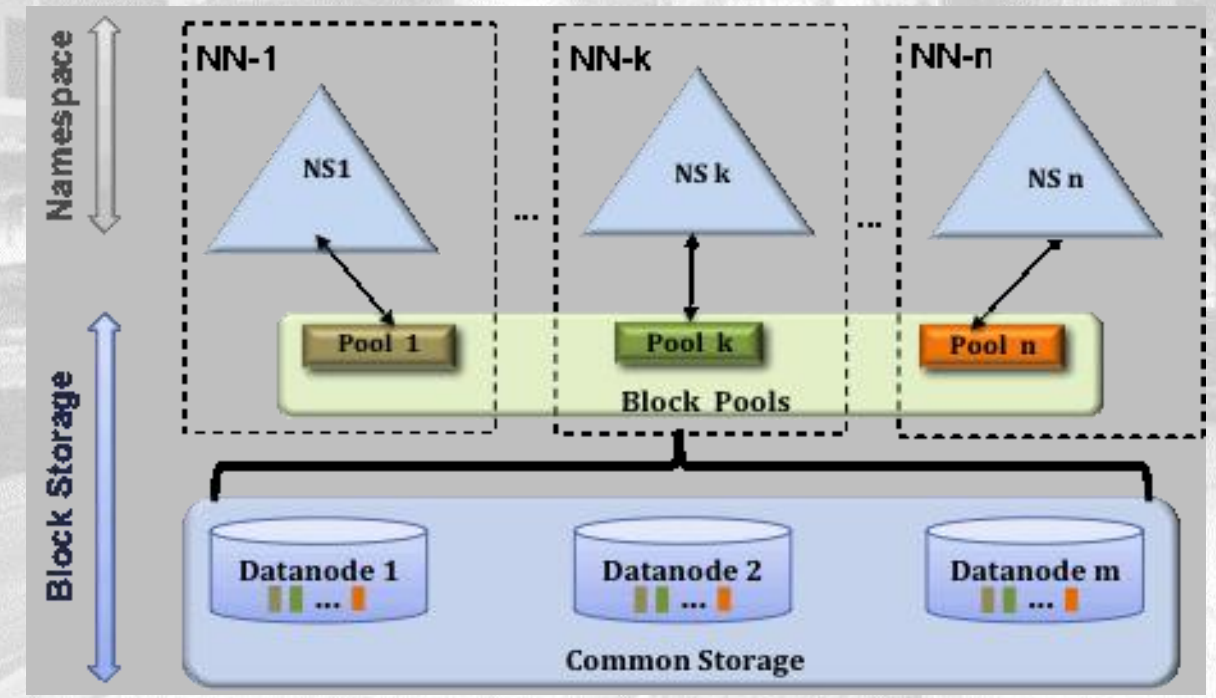


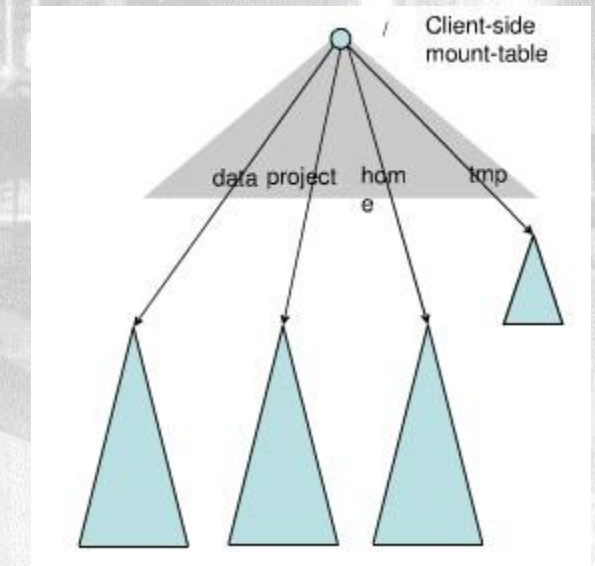
- Namespace
  - 管理目录，文件和数据块
  - 它支持常见的文件系统操作，如创建文件，修改文件，删除文件等

- Block Storage有两部分组成：
  - Block Management
    - 处理Data Node向Name Node注册的请求，处理datanode的成员关系，处理来自Data Node周期性的心跳
    - 处理来自块的报告信息，维护块的位置信息
    - 处理与块相关的操作：块的创建、删除、修改及获取块信息
    - 管理副本放置（replica placement）和块的复制及多余块的删除
  - Physical Storage
    - 存储实际的数据块并提供针对数据块的读写服务

HDFS Federation

- 解决了什么问题？
  - HDFS集群扩展性：多个NameNode分管一部分目录，使得一个集群可以扩展到更多节点，不再像1.0中那样由于内存的限制制约文件存储数目
  - 性能更高效：多个NameNode管理不同的数据，且同时对外提供服务，将为用户提供更高的读写吞吐率
  - 良好的隔离性：用户可根据需要将不同业务数据交由不同NameNode管理，这样不同业务之间影响很小
- 架构





HDFS Federation

core-site.xml

```
<xi:include href="cmt.xml"/>
<property>
    <name>fs.defaultFS</name>
    <value>viewfs://nsX</value>
</property>
```

cmt.xml

```
<configuration>
    <property>
        <name>fs.viewfs.mounttable.nsX.link
        <value>hdfs://ns1/real_share</value
    </property>
    <property>
        <name>fs.viewfs.mounttable.nsX.link
        <value>hdfs://ns2/real_user</value>
    </property>
</configuration>
```

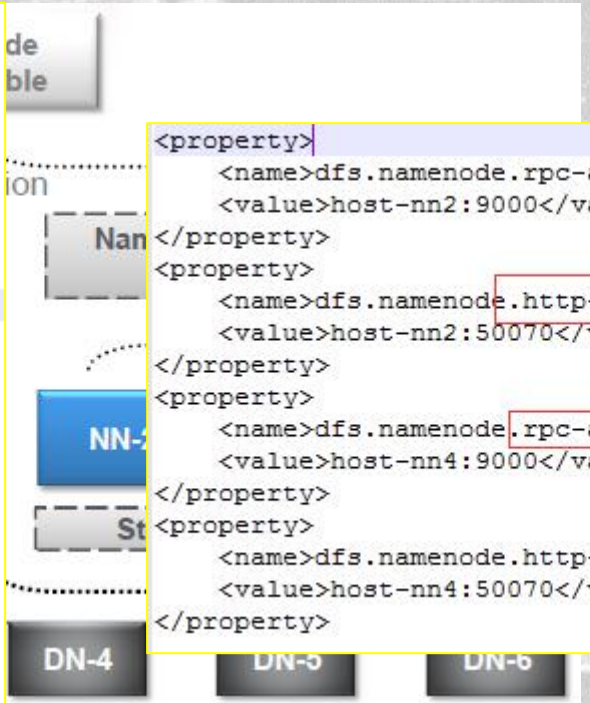
hdfs-site.xml

```
<property>
    <name>dfs.nameservices</name>
    <value>ns1,ns2</value>
</property>
<property>
    <name>dfs.namenode.rpc-address.ns1</name>
    <value>host-nn1:9000</value>
</property>
<property>
    <name>dfs.namenode.http-address.ns1</name>
    <value>host-nn1:50070</value>
</property>
<property>
    <name>dfs.namenode.rpc-address.ns2</name>
    <value>host-nn2:9000</value>
</property>
<property>
    <name>dfs.namenode.http-address.ns2</name>
    <value>host-nn2:50070</value>
</property>
```

HDFS Federation & HA

HA + Federation hdfs-site.xml

```
<property>
    <name>dfs.nameservices</name>
    <value>ns1, ns2</value>
</property>
<property>
    <name>dfs.ha.namenodes.ns1</name>
    <value>nn1,nn3</value>
</property>
<property>
    <name>dfs.ha.namenodes.ns2</name>
    <value>nn2,nn4</value>
</property>
<property>
    <name>dfs.namenode.rpc-address.ns1.nn1</name>
    <value>host-nn1:9000</value>
</property>
<property>
    <name>dfs.namenode.http-address.ns1.nn1</name>
    <value>host-nn1:50070</value>
</property>
<property>
    <name>dfs.namenode.rpc-address.ns1.nn3</name>
    <value>host-nn3:9000</value>
</property>
<property>
    <name>dfs.namenode.http-address.ns1.nn3</name>
    <value>host-nn3:50070</value>
</property>
```




```
<property>
    <name>dfs.namenode.rpc-address.ns2.nn2</name>
    <value>host-nn2:9000</value>
</property>
<property>
    <name>dfs.namenode.http-address.ns2.nn2</name>
    <value>host-nn2:50070</value>
</property>
<property>
    <name>dfs.namenode.rpc-address.ns2.nn4</name>
    <value>host-nn4:9000</value>
</property>
<property>
    <name>dfs.namenode.http-address.ns2.nn4</name>
    <value>host-nn4:50070</value>
</property>
```

第五部分 HDFS Permission

HDFS Permision

- 开启权限配置
  - dfs.permissions.enabled = true
  - dfs.web.ugi = webuser,webgroup
  - dfs.permissions.superusergroup = supergroup
  - dfs.namenode.acls.enabled = true
- 判断用户标识的操作
  - Simple
  - Kerberos
- rwx权限解释
  - 对于文件来说：r
  - 对于目录，r表示
    权限代表可以访
- 基本权限
  - rwxrwxrwx
  - chown：更改文件用户及用户组
  - chmod：更改文件权限

```
[root@cmagent2 flume]# hdfs dfs -put netcat.sh /out/net
[root@cmagent2 flume]# hdfs dfs -ls /out
Found 1 items
-rw-r--r--   1 root supergroup        273 2016-07-06 17:42 /out/net
[root@cmagent2 flume]# hdfs dfs -chmod 775 /out/net
[root@cmagent2 flume]# hdfs dfs -ls /out
Found 1 items
-rwxrwxr-x   1 root supergroup        273 2016-07-06 17:42 /out/net
[root@cmagent2 flume]# hdfs dfs -chown root:ops /out/net
chown: changing ownership of '/out/net': User does not belong to ops
[root@cmagent2 flume]#
[root@cmagent2 flume]# hdfs dfs -chown root:root /out/net
[root@cmagent2 flume]# hdfs dfs -ls /out
Found 1 items
-rwxrwxr-x   1 root root        273 2016-07-06 17:42 /out/net
```

件
除文件或目录，x

HDFS Permision

- 扩展权限ACL
  - +
  - 查看：
  - 更改] |[--set <acl

```
[root@cmagent2 flume]# hdfs dfs -getfacl /out/net
# file: /out/net
```

Options:

- -b: Remove all but the base ACL entries. The entries for user, group and others are retained for compatibility with permission bits.
- -k: Remove the default ACL.
- -R: Apply operations to all files and directories recursively.
- -m: Modify ACL. New entries are added to the ACL, and existing entries are retained.
- -x: Remove specified ACL entries. Other ACL entries are retained.
- `--set`: Fully replace the ACL, discarding all existing entries. The *acl_spec* must include entries for user, group, and others for compatibility with permission bits.
- *acl_spec*: Comma separated list of ACL entries.
- *path*: File or directory to modify.

```
# owner: root
# group: root
user::rwx
user:spark:rw-
group::rwx
group:hdfs:rw-
mask::rwx
other::r-x
```

```
[root@cmagent2 flume]# hdfs dfs -setfacl -b /out/net
[root@cmagent2 flume]# hdfs dfs -getfacl /out/net
# file: /out/net
# owner: root
# group: root
user::rwx
group::rwx
other::r-x
```

第五部分 HDFS Experience

HDFS Experience

- 单台datanode磁盘存储的负载均衡
- namenode意外挂掉之后的恢复过程
- 开启权限后有些map任务无法向/user目录写入文件导致失败
- 有磁盘坏了导致datenote无法启动
- 能够启动datanode，但无法访问
- 怎么让一个datanote节点安全退役
- .....

踊跃发言

分享你在使用hdfs过程中所遇到的问题以及解决方式

总结 end

01 HDFS Architecture

02 HDFS Function Outline

03 HDFS HA

04 HDFS Federation

05 HDFS Permission

06 HDFS Experience

参考网站

- http://hadoop.apache.org/docs/r2.7.2/
- https://www.ibm.com/developerworks/cn/linux/l-cn-sshforward/
- http://ju.outofmemory.cn/entry/47341
- http://blog.cheyo.net/213.html
- https://www.ibm.com/developerworks/cn/opensource/os-cn-hadoop-name-node/
- http://www.infoq.com/cn/articles/hadoop-2-0-namenode-ha-federation-practice-zh
- http://www.cnblogs.com/rilley/archive/2012/02/13/2349858.html
- http://www.aboutyun.com/thread-13138-1-1.html

谢谢大家!