小象科技
ChinaHadoop.cn

HDFS 2.0应用场景、原理、基本架构及使用方法

讲师：董西成

议程

1. HDFS概述
2. HDFS基本架构和原理
3. HDFS程序设计
4. HDFS 2.0新特性
5. 总结

HDFS是什么？

小象科技 ChinaHadoop.cn

- 源自于Google的GFS论文
  - ✓ 发表于2003年10月
  - ✓ HDFS是GFS克隆版
- Hadoop Distributed File System
  - ✓ 易于扩展的分布式文件系统
  - ✓ 运行在大量普通廉价机器上，提供容错机制
  - ✓ 为大量用户提供性能不错的文件存取服务

HDFS优点

小象科技 ChinaHadoop.cn

- 高容错性
  - ✓ 数据自动保存多个副本
  - ✓ 副本丢失后，自动恢复
- 适合批处理
  - ✓ 移动计算而非数据
  - ✓ 数据位置暴露给计算框架
- 适合大数据处理
  - ✓ GB、TB、甚至PB级数据
  - ✓ 百万规模以上的文件数量
  - ✓ 10K+节点规模

HDFS优点

小象科技 ChinaHadoop.cn

- 流式文件访问
  - ✓ 一次性写入，多次读取
  - ✓ 保证数据一致性
- 可构建在廉价机器上
  - ✓ 通过多副本提高可靠性
  - ✓ 提供了容错和恢复机制

HDFS缺点

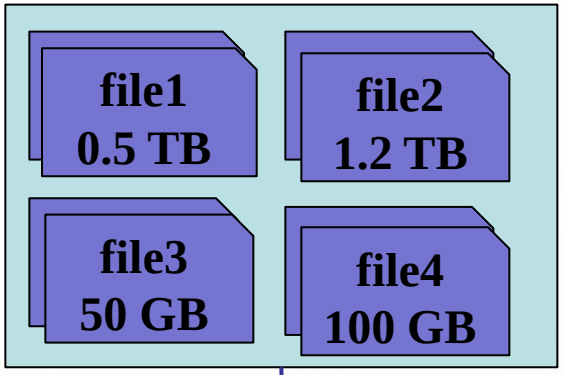
小象科技 ChinaHadoop.cn

- 低延迟数据访问
  - ✓ 比如毫秒级
  - ✓ 低延迟与高吞吐率
- 小文件存取
  - ✓ 占用NameNode大量内存
  - ✓ 寻道时间超过读取时间
- 并发写入、文件随机修改
  - ✓ 一个文件只能有一个写者
  - ✓ 仅支持append

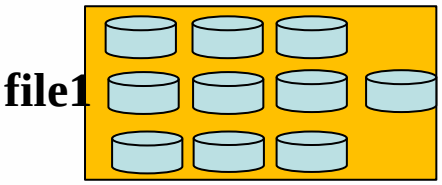
议程

1. HDFS概述
2. HDFS基本架构和原理
3. HDFS程序设计
4. HDFS 2.0新特性
5. 总结

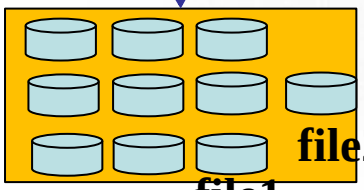
分布式文件系统的一种实现方式



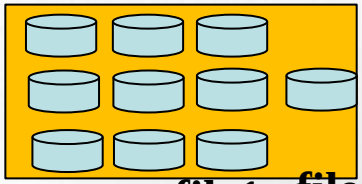
file1:node1,node2,node3
file2:node2,node3,node4
file3:node4,mode5,node6
file4:node5,node6.node7
…….



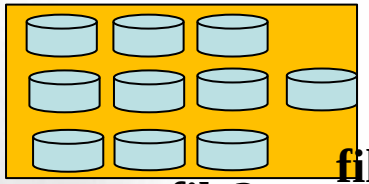
Server
(10 * 1TB)



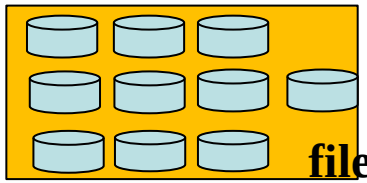
Server
(10 * 1TB)

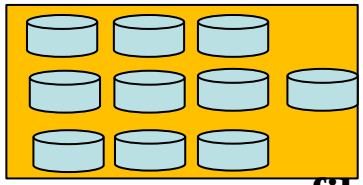


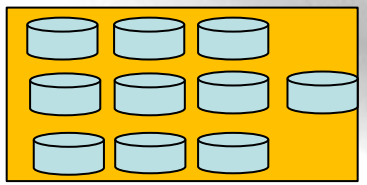
Server
(10 * 1TB)



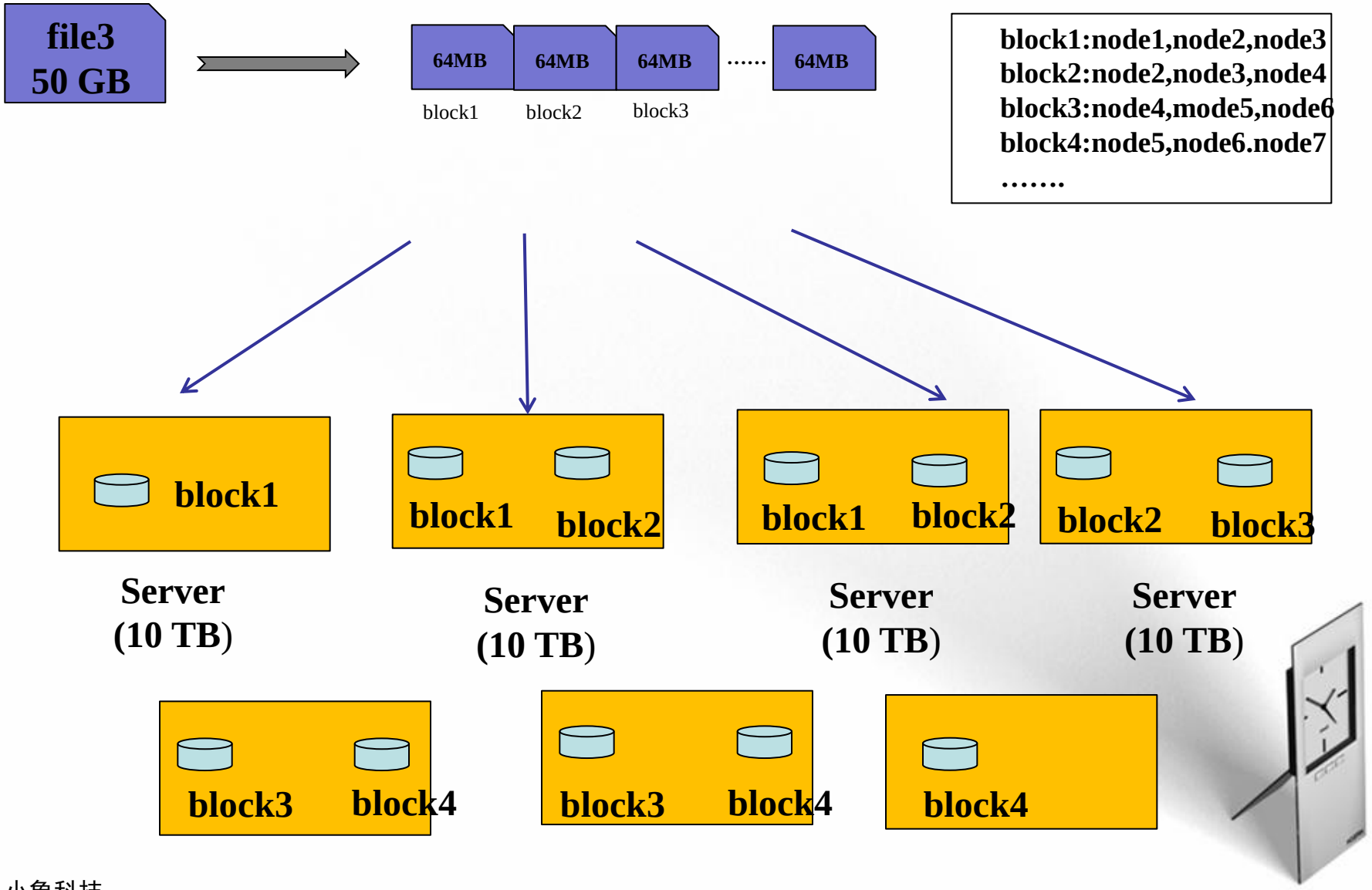
Server
(10 * 1TB)



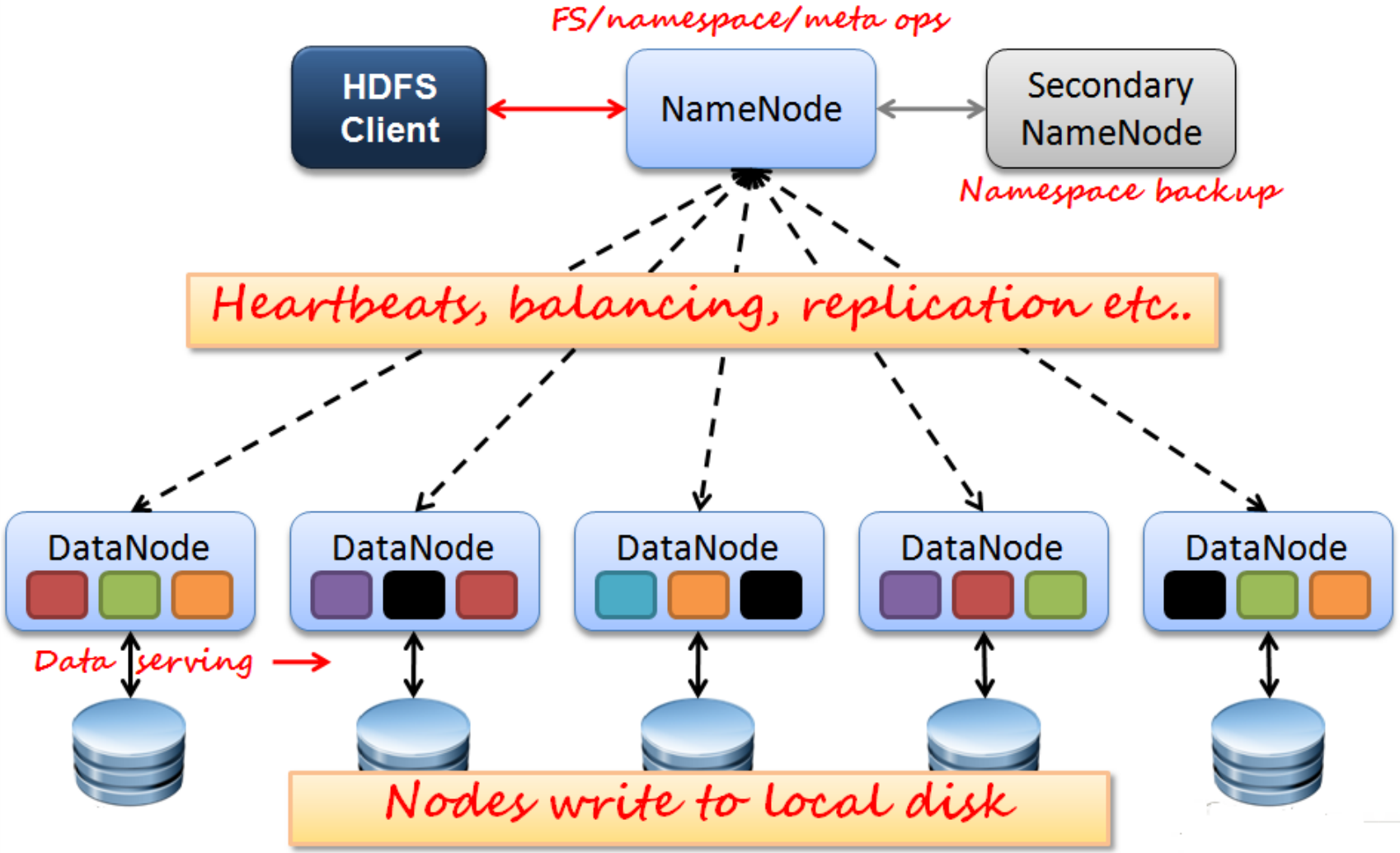




HDFS设计思想



HDFS架构


FS/namespace/meta ops
HDFS Client
NameNode
Secondary NameNode
Namespace backup
Heartbeats, balancing, replication etc..
DataNode
DataNode
DataNode
DataNode
DataNode
Data serving
Nodes write to local disk

HDFS架构

小象科技 ChinaHadoop.cn

Active Namenode

- 主Master（只有一个）
- 管理HDFS的名称空间
- 管理数据块映射信息
- 配置副本策略
- 处理客户端读写请求

Standby NameNode

- NameNode的热备；
- 定期合并fsimage和fsedits，推送给NameNode；
- 当Active NameNode出现故障时，快速切换为新的 Active NameNode。

HDFS架构

小象科技 ChinaHadoop.cn

Datanode

- Slave（有多个）
- 存储实际的数据块
- 执行数据块读/写

Client

- 文件切分
- 与NameNode交互，获取文件位置信息；
- 与DataNode交互，读取或者写入数据；
- 管理HDFS；
- 访问HDFS。

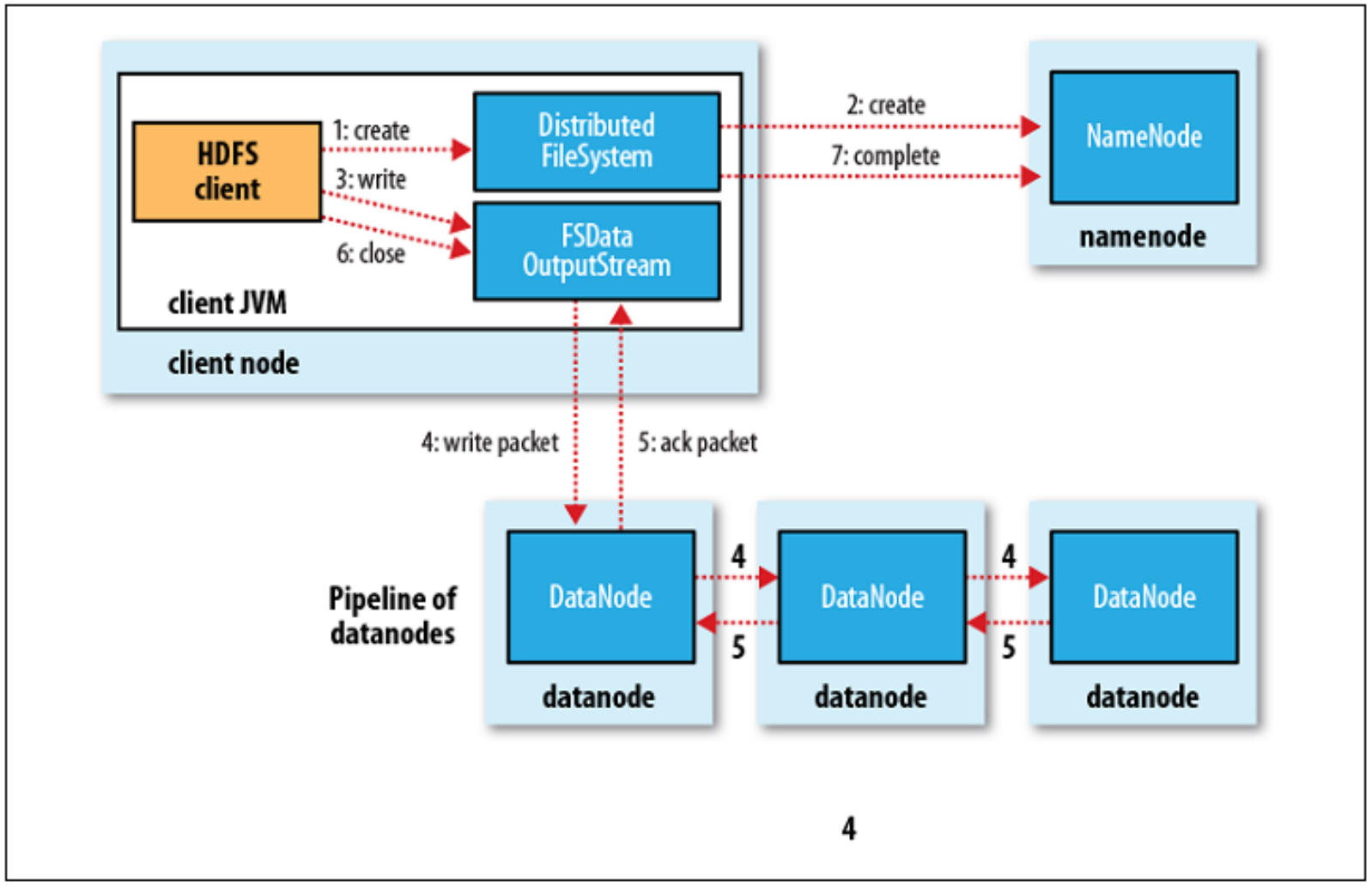
HDFS数据块（block）

小象科技 ChinaHadoop.cn

- 文件被切分成固定大小的数据块
  - ✓ 默认数据块大小为64MB，可配置
  - ✓ 若文件大小不到64MB，则单独存成一个block
- 为何数据块如此之大
  - ✓ 数据传输时间超过寻道时间（高吞吐率）
- 一个文件存储方式
  - ✓按大小被切分成若干个block，存储到不同节点上
  - ✓默认情况下每个block有三个副本

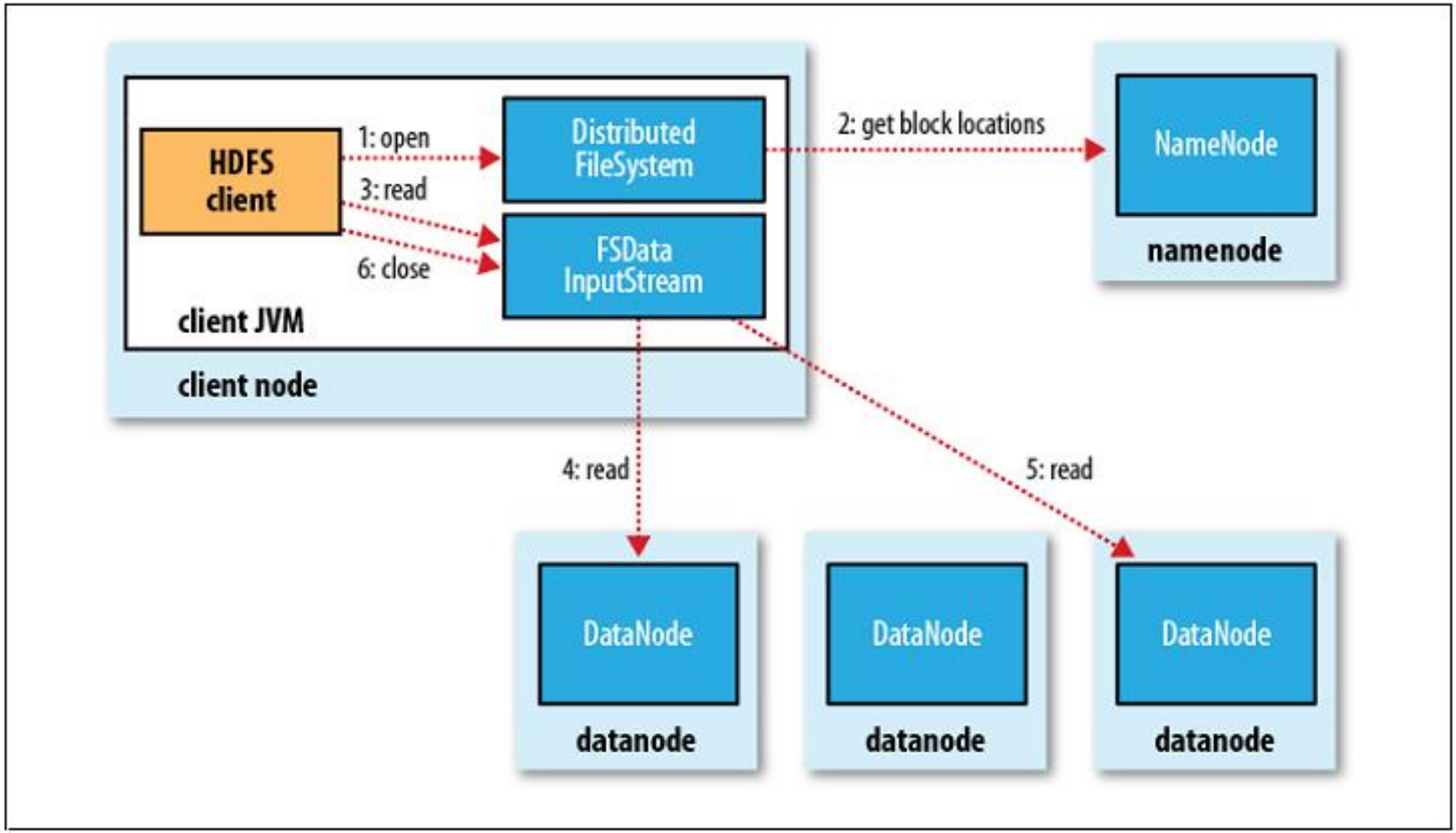
HDFS写流程

小象科技 ChinaHadoop.cn

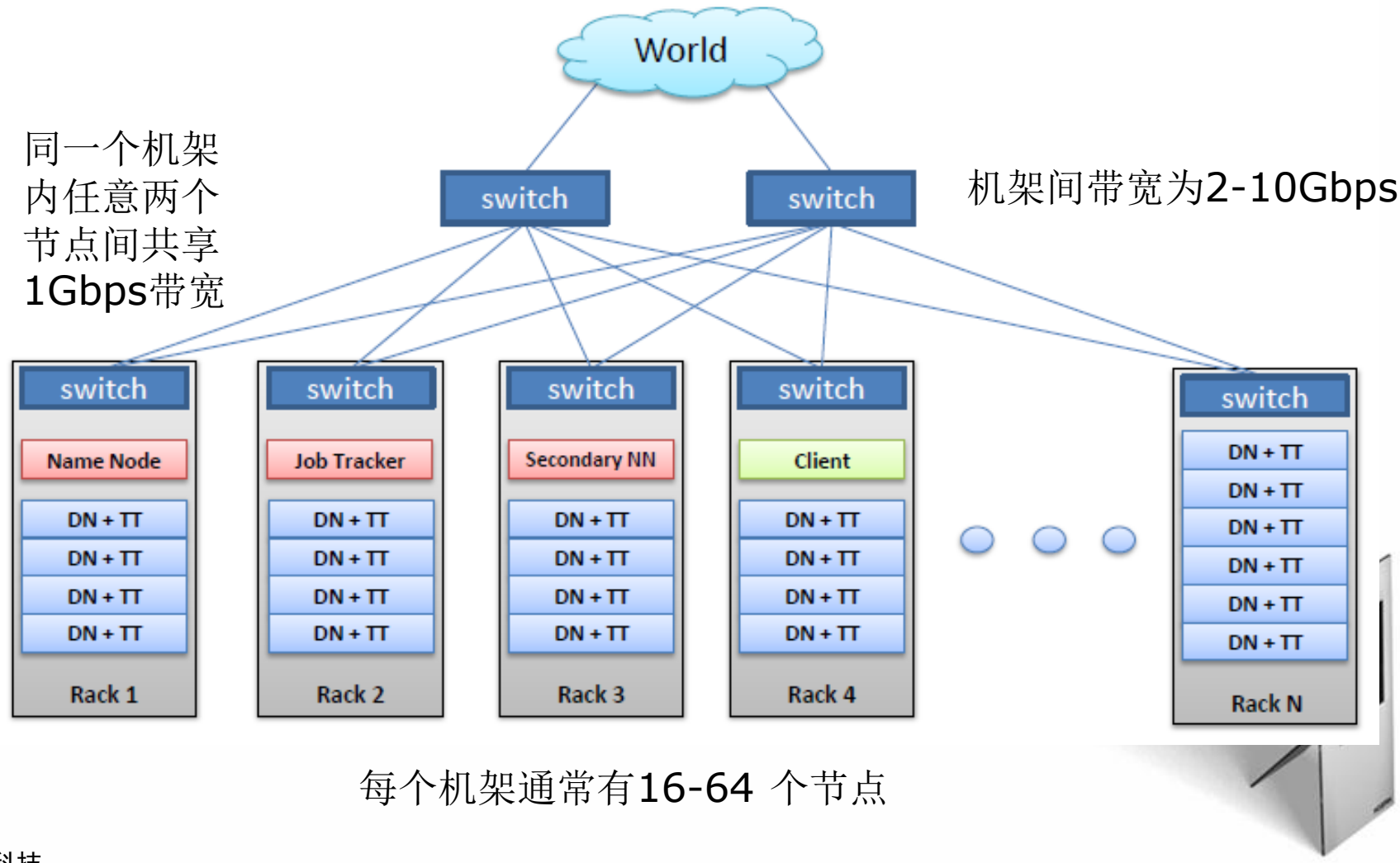


HDFS读流程

小象科技 ChinaHadoop.cn



HDFS典型物理拓扑



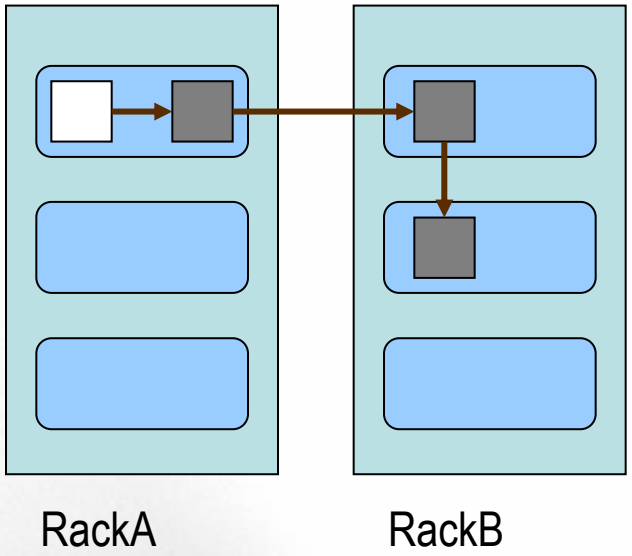
每个机架通常有16-64 个节点

HDFS副本放置策略

小象科技 ChinaHadoop.cn

- 问题：

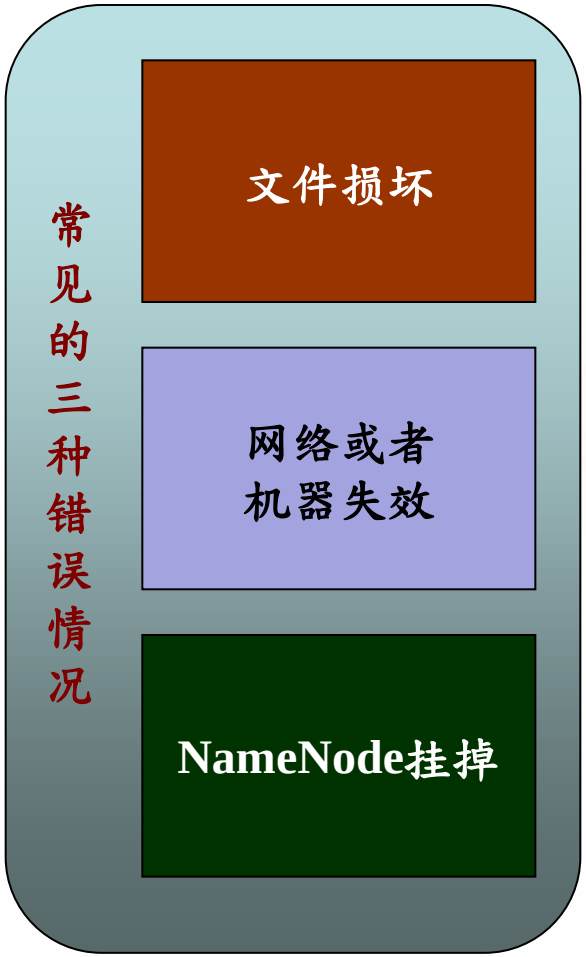
一个文件划分成多个block，每个block存多份，如何为每个block选择节点存储这几份数据？



- Block副本放置策略：
  - ✓ 副本1: 同Client的节点上
  - ✓ 副本2: 不同机架中的节点上
  - ✓ 副本3: 与第二个副本同一机架的另一个节点上
  - ✓ 其他副本:随机挑选

HDFS可靠性策略

小象科技 ChinaHadoop.cn



- 文件完整性
  - CRC32校验
  - 用其他副本取代损坏文件
- Heartbeat
  - Datanode 定期向Namenode发heartbeat
- 元数据信息
  - FSImage（文件系统镜像）、Editlog（操作日志）
  - 多份存储
  - 主备NameNode实时切换

HDFS不适合存储小文件

小象科技 ChinaHadoop.cn

- 元信息存储在NameNode内存中
  - ✓ 一个节点的内存是有限的
- 存取大量小文件消耗大量的寻道时间
  - ✓ 类比拷贝大量小文件与拷贝同等大小的一个大文件
- NameNode存储block数目是有限的
  - ✓一个block元信息消耗大约150 byte内存
  - ✓存储1亿个block，大约需要20GB内存
  - ✓如果一个文件大小为10K，则1亿个文件大小仅为1TB（但要消耗掉NameNode 20GB内存）

议程

1. HDFS概述
2. HDFS基本架构和原理
3. HDFS程序设计
4. HDFS 2.0新特性
5. 总结

HDFS访问方式

小象科技 ChinaHadoop.cn

- HDFS Shell命令
- HDFS Java API
- HDFS REST API
- HDFS Fuse：实现了fuse协议
- HDFS lib hdfs：C/C++访问接口
- HDFS 其他语言编程API
  - ✓ 使用thrift实现
  - ✓ 支持C++、Python、php、C#等语言

HDFS Shell命令—概览

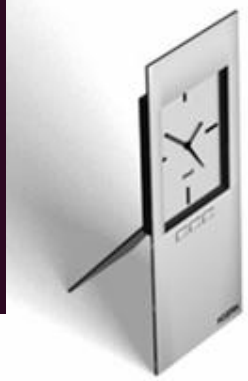
小象科技 ChinaHadoop.cn

```
root@SY-0266:~# /opt/pgs/yarn/bin/hdfs
Usage: hdfs [--config confdir] COMMAND
       where COMMAND is one of:
  dfs                  run a filesystem command on the file systems supported in Hadoop.
  namenode -format     format the DFS filesystem
  secondarynamenode    run the DFS secondary namenode
  namenode             run the DFS namenode
  journalnode          run the DFS journalnode
  zkfc                 run the ZK Failover Controller daemon
  datanode             run a DFS datanode
  dfsadmin             run a DFS admin client
  haadmin              run a DFS HA admin client
  fsck                 run a DFS filesystem checking utility
  balancer             run a cluster balancing utility
  jmxget               get JMX exported values from NameNode or DataNode.
  oiv                  apply the offline fsimage viewer to an fsimage
  oev                  apply the offline edits viewer to an edits file
  fetchdt              fetch a delegation token from the NameNode
  getconf              get config values from configuration
  groups               get the groups which users belong to
  snapshotDiff         diff two snapshots of a directory or diff the
                       current directory contents with a snapshot
  lsSnapshottableDir   list all snapshottable dirs owned by the current user
                                                  Use -help to see options
  portmap              run a portmap service
  nfs3                 run an NFS version 3 gateway
```

HDFS Shell命令—文件操作命令

小象科技 ChinaHadoop.cn

```
dongxicheng@dongxicheng-laptop:~/hadoop/hadoop-0.20.2-cdh3u6$ bin/hadoop fs
Usage: java FsShell
           [-ls <path>]
           [-lsr <path>]
           [-df [<path>]]
           [-du <path>]
           [-dus <path>]
           [-count[-q] <path>]
           [-mv <src> <dst>]
           [-cp <src> <dst>]
           [-rm [-skipTrash] <path>]
           [-rmr [-skipTrash] <path>]
           [-expunge]
           [-put <localsrc> ... <dst>]
           [-copyFromLocal <localsrc> ... <dst>]
           [-moveFromLocal <localsrc> ... <dst>]
           [-get [-ignoreCrc] [-crc] <src> <localdst>]
           [-getmerge <src> <localdst> [addnl]]
           [-cat <src>]
           [-text <src>]
           [-copyToLocal [-ignoreCrc] [-crc] <src> <localdst>]
           [-moveToLocal [-crc] <src> <localdst>]
           [-mkdir <path>]
           [-setrep [-R] [-w] <rep> <path/file>]
           [-touchz <path>]
           [-test -[ezd] <path>]
           [-stat [format] <path>]
           [-tail [-f] <file>]
           [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
           [-chown [-R] [OWNER][:[GROUP]] PATH...]
```



HDFS Shell命令—文件操作命令

小象科技 ChinaHadoop.cn

- 将本地文件上传到HDFS上

bin/hadoop fs -copyFromLocal /local/data /hdfs/data

- 删除文件/目录

bin/hadoop fs -rmr /hdfs/data

- 创建目录

bin/hadoop fs -mkdir /hdfs/data

HDFS Shell命令—管理命令

小象科技 ChinaHadoop.cn

```
dongxicheng@dongxicheng-laptop:~/hadoop/hadoop-0.20.2-cdh3u6$ bin/hadoop dfsadmin
Usage: java DFSAdmin
           [-report]
           [-safemode enter | leave | get | wait]
           [-saveNamespace]
           [-refreshNodes]
           [-finalizeUpgrade]
           [-upgradeProgress status | details | force]
           [-metasave filename]
           [-refreshServiceAcl]
           [-refreshUserToGroupsMappings]
           [-refreshSuperUserGroupsConfiguration]
           [-setQuota <quota> <dirname>...<dirname>]
           [-clrQuota <dirname>...<dirname>]
           [-setSpaceQuota <quota> <dirname>...<dirname>]
           [-clrSpaceQuota <dirname>...<dirname>]
           [-setBalancerBandwidth <bandwidth in bytes per second>]
           [-help [cmd]]
```

HDFS Shell命令—管理脚本

小象科技 ChinaHadoop.cn

- 在sbin目录下
  - ✓ start-all.sh
  - ✓ start-dfs.sh
  - ✓ start-yarn.sh
  - ✓ hadoop-deamon(s).sh
- 单独启动某个服务
  - ✓ hadoop-deamon.sh start namenode
  - ✓ hadoop-deamons.sh start namenode（通过SSH登录到各个节点）

HDFS Shell命令—文件管理命令fsck

小象科技 ChinaHadoop.cn

- 检查hdfs中文件的健康状况
- 查找缺失的块以及过少或过多副本的块
- 查看一个文件的所有数据块位置
- 删除损坏的数据块

```
dongxicheng@dongxicheng-laptop:~/hadoop/hadoop-0.20.2-cdh3u6$ bin/hadoop fsck
Usage: DFSck <path> [-move | -delete | -openforwrite] [-files [-blocks [-locations | -racks]]]
        <path>  start checking from this path
        -move   move corrupted files to /lost+found
        -delete delete corrupted files
        -files  print out files being checked
        -openforwrite   print out files opened for write
        -blocks print out block report
        -locations      print out locations for every block
        -racks  print out network topology for data-node locations
                By default fsck ignores files opened for write, use -openforwrite to report such
lock allocation status
```

HDFS Shell命令—文件管理命令fsck

小象科技 ChinaHadoop.cn

```
dongxicheng@dongxicheng-laptop:~/hadoop/hadoop-0.20.2-cdh3u6$ bin/hadoop fsck /home/dongxicheng/README.txt -files -blocks -locations
FSCK started by dongxicheng (auth:SIMPLE) from /127.0.1.1 for path /home/dongxicheng/README.txt at Sat Sep 21 22:39:52 CST 2013
/home/dongxicheng/README.txt 1366 bytes, 1 block(s):  OK
0. blk_7247523855706538504_1001 len=1366 repl=1 [127.0.0.1:50010]

Status: HEALTHY
 Total size:    1366 B
 Total dirs:    0
 Total files:   1
 Total blocks (validated):      1 (avg. block size 1366 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Sat Sep 21 22:39:52 CST 2013 in 1 milliseconds
```

HDFS Shell命令—数据均衡器balancer

小象科技 ChinaHadoop.cn

- 数据块重分布
  - ✓ bin/start-balancer.sh -threshold <percentage of disk capacity>
- percentage of disk capacity
  - ✓ HDFS达到平衡状态的磁盘使用率偏差值
  - ✓ 值越低各节点越平衡，但消耗时间也更长

HDFS Shell命令—设置目录份额

小象科技 ChinaHadoop.cn

- 限制一个目录最多使用磁盘空间
  - ✓ **bin/hadoop dfsadmin -setSpaceQuota 1t /user/username**
- 限制一个目录包含的最多子目录和文件数目
  - ✓ **bin/hadoop dfsadmin -setQuota 10000 /user/username**

HDFS Shell命令—增加/移除节点

小象科技 ChinaHadoop.cn

- 加入新的datanode
  - ✓ 步骤1：将已存在datanode上的安装包（包括配置文件等）拷贝到新datanode上；
  - ✓ 步骤2：启动新datanode：

```
sbin/hadoop-deamon.sh start datanode
```

- 移除旧datanode
  - ✓ 步骤1：将datanode加入黑名单，并更新黑名单，在NameNode上，将datanode的host或者ip加入配置选项dfs.hosts.exclude指定的文件中
  - ✓ 步骤2：移除datanode

```
bin/hadoop dfsadmin -refreshNodes
```

HDFS Java API介绍

小象科技 ChinaHadoop.cn

- **Configuration类：该类的对象封装了配置信息，这些配置信息来自core-*.xml；**
- **FileSystem类：文件系统类，可使用该类的方法对文件/目录进行操作。一般通过FileSystem的静态方法get获得一个文件系统对象；**
- **FSDataInputStream和FSDataOutputStream类：HDFS中的输入输出流。分别通过FileSystem的open方法和create方法获得。**

以上类均来自java包：org.apache.hadoop.fs

HDFS Java程序举例

小象科技 ChinaHadoop.cn

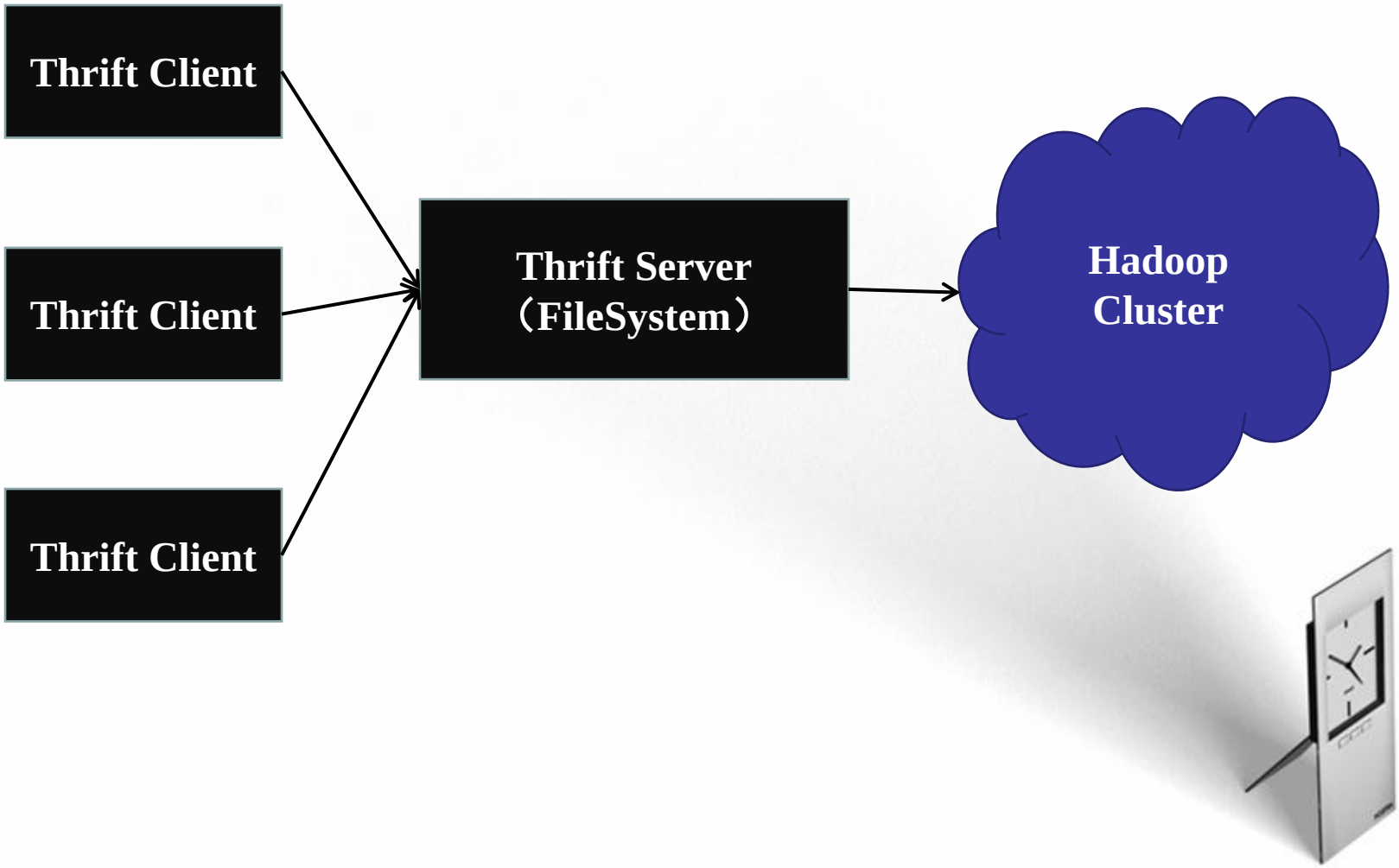
- 将本地文件拷贝到HDFS上；

```
Configuration config = new Configuration();
FileSystem hdfs = FileSystem.get(config);
Path srcPath = new Path(srcFile);
Path dstPath = new Path(dstFile);
hdfs.copyFromLocalFile(srcPath, dstPath);
```

- 创建HDFS文件；

```
//byte[] buff – 文件内容
Configuration config = new Configuration();
FileSystem hdfs = FileSystem.get(config);
Path path = new Path(fileName);
FSDataOutputStream outputStream = hdfs.create(path);
outputStream.write(buff, 0, buff.length);
```

HDFS 多语言API—借助thrift



hadoopfs.thrift接口定义


```
service ThriftHadoopFileSystem
{

  // set inactivity timeout period. The period is specified in seconds.
  // if there are no RPC calls to the HadoopThrift server for this much
  // time, then the server kills itself.
  void setInactivityTimeoutPeriod(1:i64 periodInSeconds),

  // close session
  void shutdown(1:i32 status),

  // create a file and open it for writing
  ThriftHandle create(1:Pathname path) throws (1:ThriftIOException ouch),

  // create a file and open it for writing
  ThriftHandle createFile(1:Pathname path, 2:i16 mode,
                          3:bool overwrite, 4:i32 bufferSize,
                          5:i16 block_replication, 6:i64 blocksize)
                          throws (1:ThriftIOException ouch),

  // returns a handle to an existing file  for reading
  ThriftHandle open(1:Pathname path) throws (1:ThriftIOException ouch),

  // returns a handle to an existing file for appending to it.
  ThriftHandle append(1:Pathname path) throws (1:ThriftIOException ouch),

  // write a string to the open handle for the file
  bool write(1:ThriftHandle handle, string data) throws (1:ThriftIOException ouch),

  // read some bytes from the open handle for the file
  string read(1:ThriftHandle handle, i64 offset, i32 size) throws (1:ThriftIOException ouch),

  // close file
  bool close(1:ThriftHandle out) throws (1:ThriftIOException ouch),
```

PHP语言访问HDFS

小象科技 ChinaHadoop.cn

```php
$transport = new TSocket(HDFS_HOST, HDFS_PORT);
$transport->setRecvTimeout(60000);
$transport->setSendTimeout(60000);
$protocol =new TBinaryProtocol($transport);
$client = new ThriftHadoopFileSystemClient($protocol);
logv("connect hdfs");
$transport->open();
logv("testing existent of `%s'", $remote_uri);
$remote_path = new Pathname(array('pathname' => $remote_uri));
$remote_file = null;
try {
        $remote_file = $client->listStatus($remote_path);
} catch(Exception $e) { }
if (!$remote_file)
    loge("could not open `%s'", $remote_uri);
```

Python语言访问HDFS

小象科技 ChinaHadoop.cn

```
def connect(self):
  try:
    # connect to hdfs thrift server
    self.transport = TSocket.TSocket(self.server_name, self.server_port)
    self.transport = TTransport.TBufferedTransport(self.transport)
    self.protocol = TBinaryProtocol.TBinaryProtocol(self.transport)

    # Create a client to use the protocol encoder
    self.client = ThriftHadoopFileSystem.Client(self.protocol)
    self.transport.open()

    # tell the HadoopThrift server to die after 60 minutes of inactivity
    self.client.setInactivityTimeoutPeriod(60*60)
    return True

  except Thrift.TException, tx:
    print "ERROR in connecting to ", self.server_name, ":", self.server_port
    print '%s' % (tx.message)
    return False
```

```
def do_create(self, name):
  if name == "":
    print "  ERROR usage: create <pathname>"
    print
    return 0

  # Create the file, and immediately closes the handle
  path = Pathname();
  path.pathname = name;
  status = self.client.create(path)
  self.client.close(status)
  return 0
```

议程

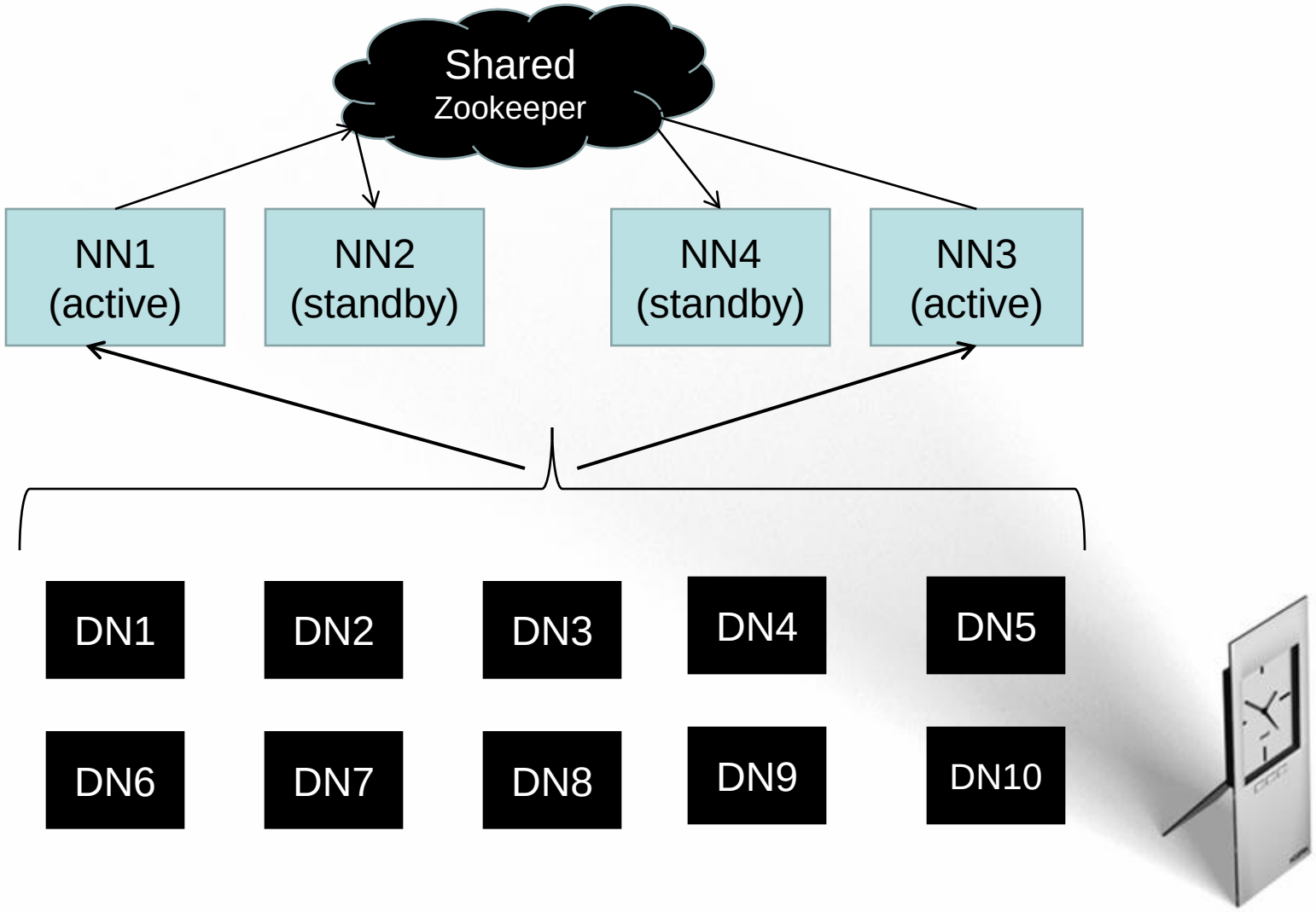
1. HDFS概述
2. HDFS基本架构和原理
3. HDFS程序设计
4. HDFS 2.0新特性
5. 总结

Hadoop 2.0新特性

小象科技 ChinaHadoop.cn

- NameNode HA
- NameNode Federation
- HDFS 快照（snapshot）
- HDFS 缓存（in-memory cache）
- HDFS ACL
- 异构层级存储结构（Heterogeneous Storage hierarchy）

HA与Federation



异构层级存储结构—背景

小象科技 ChinaHadoop.cn

- HDFS将所有存储介质抽象成性能相同的Disk

```
<property>
  <name>dfs.datanode.data.dir</name>
  <value>/dir0,/dir1,/dir2,/dir3</value>
</property>
```

- 存储介质种类繁多，一个集群中存在多种异构介质
  - ✓ **磁盘、SSD、RAM等**
- 多种类型的任务企图同时运行在同一个Hadoop集群中
  - ✓ **批处理，交互式处理，实时处理**
  - ✓ **不同性能要求的数据，最好存储在不同类别的存储介质上**

异构层级存储结构—原理

小象科技 ChinaHadoop.cn

➢每个节点是由多种异构存储介质构成的

```
<property>
  <name>dfs.datanode.data.dir</name>
  <value>[disk]/dir0,[disk]/dir1,[ssd]/dir2,[ssd]/dir3</value>
</property>
```

异构层级存储结构—原理

小象科技 ChinaHadoop.cn

- HDFS仅提供了一种异构存储结构，并不知道存储介质的性能；
- HDFS为用户提供了API，以控制目录/文件写到什么介质上；
- HDFS为管理员提供了管理工具，可限制每个用户对每种介质的可使用份额；
- 目前完成度不高
  - ✓ 阶段1：DataNode支持异构存储介质（HDFS-2832，完成）
  - ✓ 阶段2：为用户提供访问API（HDFS-5682，未完成）

HDFS ACL—背景：现有权限管理的局限性

小象科技 ChinaHadoop.cn

/bank/exchange

-rw-r----- 3 bob team1 0 2014-05-04 21:33 /bank/exchange

-rw-r----- 3 manager team1 0 2014-05-04 21:33 /bank/exchange

manager
- bob
- tom
- lucy

team2和team3也需要读权限，怎么办？？

HDFS ACL—基于POSIX ACL的实现

小象科技 ChinaHadoop.cn

- 对当前基于POSIX文件权限管理的补充(HDFS-4685)；
- 启动该功能；
  - ✓将dfs.namenode.acls.enabled置为true
- 使用方法；
  - ✓hdfs dfs -setfacl -m user:tom:rw- /bank/exchange
  - ✓hdfs dfs -setfacl -m user:lucy:rw- /bank/exchange
  - ✓hdfs dfs -setfacl -m group:team2:r-- /bank/exchange
  - ✓hdfs dfs -setfacl -m group:team3:r-- /bank/exchange

HDFS快照—背景

小象科技 ChinaHadoop.cn

- HDFS上文件和目录是不断变化的，快照可以帮助用户保存某个时刻的数据；
- HDFS快照的作用
  - ✓ 防止用户误操作删除数据
  - ✓ 数据备份

HDFS快照—基本使用方法

小象科技 ChinaHadoop.cn

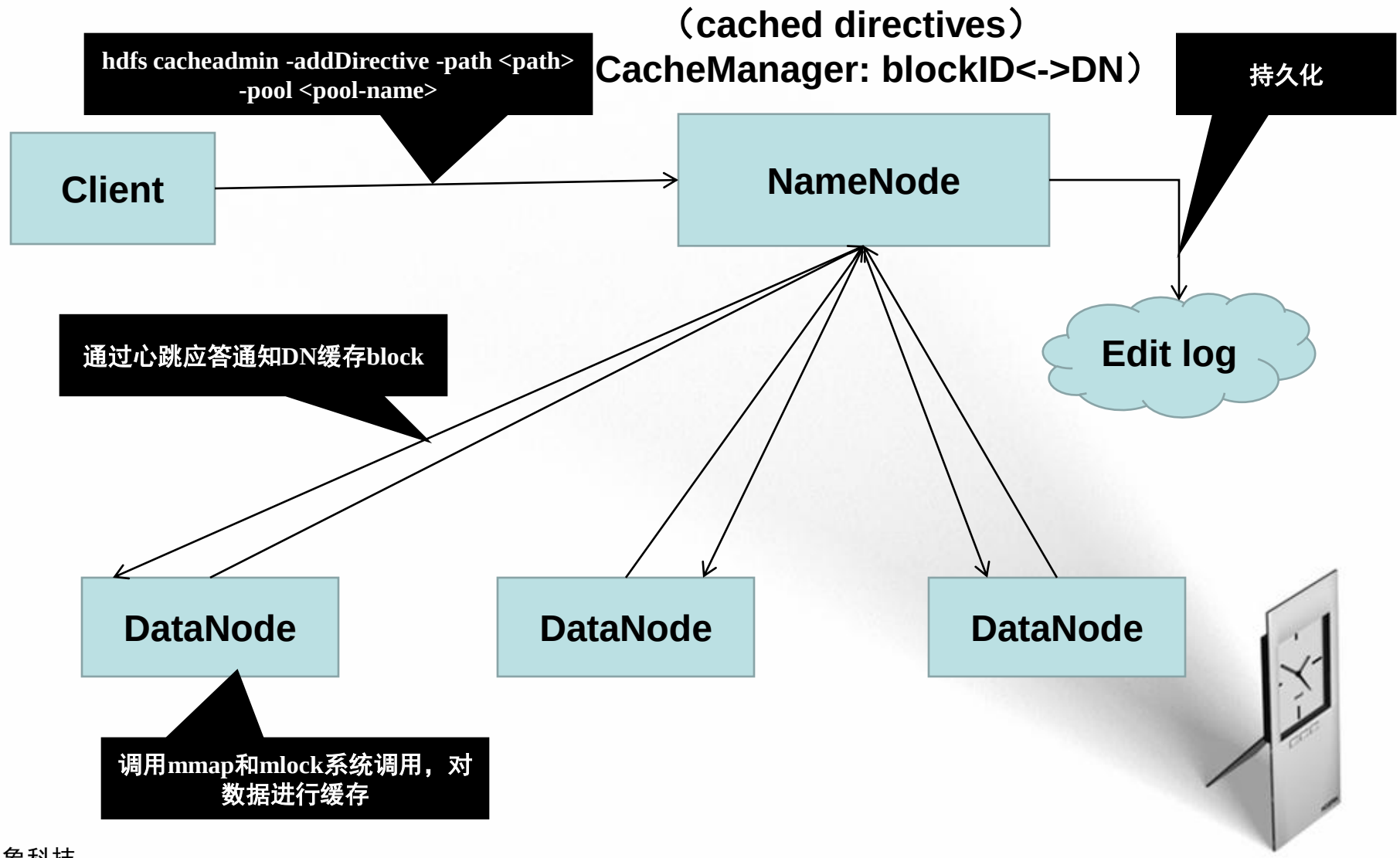
- 一个目录可以产生快照，当且仅当它是Snapshottable；
  - ✓ bin/hdfs dfsadmin allowSnapshot <path>
- 创建/删除快照；
  - ✓ bin/hdfs dfs -createSnapshot <path> [<snapshotName>]
  - ✓ bin/hdfs dfs -deleteSnapshot<path> [<snapshotName>]
- 快照存放位置和特点；
  - ✓ **快照是只读的，不可修改**
  - ✓ **快照位置：**
    - <snapshottable_dir_path>/.snapshot
    - <snapshottable_dir_path>/.snapshot/snap_name

HDFS缓存—背景

小象科技 ChinaHadoop.cn

- HDFS自身不提供数据缓存功能，而是使用OS缓存
  - ✓容易内存浪费，eg.一个block三个副本同时被缓存
- 多种计算框架共存，均将HDFS作为共享存储系统
  - ✓ **MapReduce：离线计算，充分利用磁盘**
  - ✓ **Impala：低延迟计算，充分利用内存**
  - ✓ **Spark：内存计算框架**
- HDFS应让多种混合计算类型共存一个集群中
  - ✓ **合理的使用内存、磁盘等资源**
  - ✓ **比如，高频访问的特点文件应被尽可能长期缓存，防止置换到磁盘上**

HDFS缓存—原理


（cached directives）
CacheManager: blockID<->DN）
hdfs cacheadmin -addDirective -path <path>
-pool <pool-name>
持久化
Client
NameNode
Edit log
通过心跳应答通知DN缓存block
DataNode
DataNode
DataNode
调用mmap和mlock系统调用，对
数据进行缓存

HDFS缓存—实现情况

小象科技 ChinaHadoop.cn

- 用户需通过命令显式的将一个目录或文件加入/移除缓存
  - ✓ 不支持块级别的缓存
  - ✓ 不支持自动化缓存
  - ✓ 可设置缓存失效时间
- 缓存目录：仅对一级文件进行缓存
  - ✓ 不会递归缓存所有文件与目录
- 以pool的形式组织缓存资源
  - ✓ 借助YARN的资源管理方式，将缓存划分到不同pool中
  - ✓ 每个pool有类linux权限管理机制、缓存上限、失效时间等
- 独立管理内存，未与资源管理系统YARN集成
  - ✓ 用户可为每个DN设置缓存大小，该值独立于YARN

议程

1. HDFS概述
2. HDFS基本架构和原理
3. HDFS程序设计
4. HDFS 2.0新特性
5. 总结

总结

- HDFS概述
- HDFS基本架构和原理
- HDFS程序设计
- HDFS 2.0新特性

小象科技
ChinaHadoop.cn

联系我们：

- 新浪微博：ChinaHadoop
- 微信公号：ChinaHadoop

让你的数据产生价值！