



(12) 发明专利

(10) 授权公告号 CN 107832138 B

(45) 授权公告日 2021.09.14

(21) 申请号 201710860998.6

(22) 申请日 2017.09.21

(65) 同一申请的已公布的文献号  
申请公布号 CN 107832138 A

(43) 申请公布日 2018.03.23

(73) 专利权人 南京邮电大学  
地址 210003 江苏省南京市鼓楼区新模范  
马路66号

(72) 发明人 胡文龙 王少辉 肖甫 王汝传

(74) 专利代理机构 南京知识律师事务所 32207  
代理人 李吉宽

(51) Int. Cl.  
G06F 9/50 (2006.01)

(56) 对比文件

- CN 105069152 A, 2015.11.18
- CN 104731921 A, 2015.06.24
- CN 106407385 A, 2017.02.15
- CN 103152395 A, 2013.06.12
- CN 106789197 A, 2017.05.31
- CN 104008152 A, 2014.08.27
- CN 105512266 A, 2016.04.20
- US 2016275092 A1, 2016.09.22
- US 2016078052 A1, 2016.03.17
- CN 106161495 A, 2016.11.23

马新凡.“DOA下分布式数据注册中心高可用性研究与设计”.《中国优秀硕士学位论文全文数据库 信息科技辑》.2016,(第04期),

审查员 冯世昂

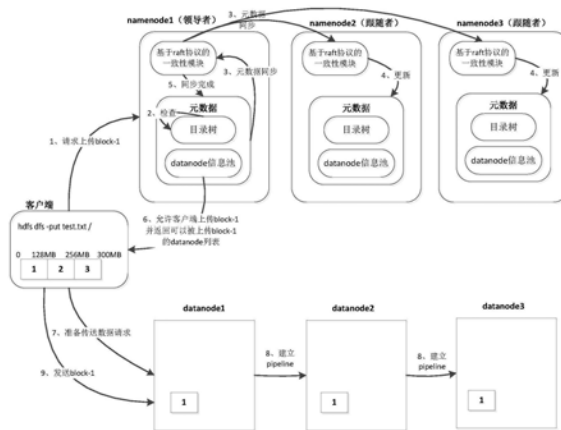
权利要求书2页 说明书5页 附图2页

(54) 发明名称

一种扁平化的高可用namenode模型的实现方法

(57) 摘要

本发明公开了一种扁平化的高可用namenode模型的实现方法,该方法解决了分布式文件系统HDFS潜在单点故障问题,实现了负载均衡。该方法包括一种扁平化的三机namenode模型,该模型中包括领导者节点、候选者节点和跟随者节点三种角色的namenode节点协调工作。一个节点可能充当不止一种角色。相较于当前主/备模式的架构,本发明大大缩短了在主节点宕机后,集群重新选主并恢复服务功能的速度,提升了HDFS文件系统响应客户端读请求时的性能。该模型不仅有效地解决了集群的单点故障问题,还实现了namenode服务器处理客户端读请求时各节点的负载均衡,提升了系统整体性能。



1. 一种扁平化的高可用namenode模型的实现方法,其特征在于,所述方法的领导者namenode的选举包括如下步骤:

步骤1-1:当HDFS刚启动时,所有namenode节点均进入跟随者状态,没有领导者;

步骤1-2:如果在100ms至500ms之间的任意时刻,跟随者namenode没有接收到任何来自领导者namenode的心跳消息,不含数据信息的远程过程调用消息,它就会假定此时集群内没有可达或可用的领导者,那么该跟随者namenode就会发起选举,首先增加自己当前的任期号,创建一个比之前使用过的任何值都要大的新任期号,随即进入候选者角色,并尝试成为整个namenode集群的领导者;

步骤1-3:候选者namenode向其他namenode服务器发送投票请求,同时自己会投给自己一票,在获得集群中超过半数namenode节点反馈的同意响应后,候选者namenode会将自已的状态转换为领导者,并立即向namenode集群中其他服务器发送心跳信息,建立领导者地位;

中断事务包括:

当前候选者namenode如果收到了来自于有效领导者namenode的心跳信息,它就会立即放弃成为领导者的尝试,随即回到跟随者的状态;

候选者经过一个随机的选举超时时间后会再次自增自己的任期号,然后重启新一轮的选举,重复步骤1-3,直至集群最终产生领导者;

当领导者namenode被选举出来后,就能接收来自客户端的请求,请求可以分为读请求和写请求两种类型,包括:

步骤2-1:客户端向领导者提交写一个数据块的请求;

步骤2-2:领导者首先去本机内存中维护的元数据的目录树中检查客户端所请求写入的文件是否已存在于HDFS上,若没有,则会去datanode信息池中挑选副本数量个datanode服务器作为客户端可写入文件的数据节点,并将客户端申请写入HDFS的文件的元信息和挑选出来的datanode节点元信息作为一条日志发送给一致性模块;

步骤2-3:领导者namenode中的一致性模块向超过半数跟随者namenode同步日志,日志同步完成后将之前挑选出来的datanode数据节点列表信息返回给客户端;

步骤2-4:客户端在接收到领导者namenode返回的datanode列表信息后开始往这些datanode上写文件。

2. 根据权利要求1所述的一种扁平化的高可用namenode模型的实现方法,其特征在于,所述的客户端从HDFS上读文件包括:

步骤3-1:客户端向namenode集群中任意一台服务器发送读请求;

步骤3-2:接收到来自客户端读请求的namenode服务器随即去目录树中检查HDFS中是否存在该文件;

步骤3-3:如果HDFS中不存在客户端要读的文件,则namenode服务器返回文件不存在异常,如果存在,则返回该文件对应的block及其副本所在的数据节点的列表信息;

步骤3-4:客户端从返回的block信息列表中挑选一个网络拓扑结构中距离最近的datanode服务器并向其发送读文件请求;

步骤3-5:被请求的datanode服务器向客户端传输文件。

3. 根据权利要求1所述的一种扁平化的高可用namenode模型的实现方法,其特征在于,

当领导者namenode出现崩溃或由于网络原因失去与过半跟随者namenode的联系,为了保证日志在每台服务器节点上的完整性与一致性和整个namenode集群的高可用性,此时namenode集群就会进入崩溃恢复过程,包括:

步骤4-1:某些或某一个跟随者namenode会进入候选者状态,并向其他服务器发起投票请求,请求里会包含自身最后一条日志记录信息的索引(lastIndex)以及任期号(lastTerm);

步骤4-2:当响应投票的服务器接收到请求,它会将候选者的日志信息与自己的日志信息进行比较,如果投票者(跟随者namenode)的日志更完整;

步骤4-3:经过上面的步骤已经选举出了领导者namenode,此时,领导者namenode会不断地向跟随者namenode发送包含自己日志信息的心跳消息;

步骤4-4:跟随者namenode根据接收到的心跳消息,删除所有跟领导者namenode不同的日志记录,并将所有丢失的日志记录依照领导者的日志进行补足。

4. 根据权利要求3所述的一种扁平化的高可用namenode模型的实现方法,其特征在于,所述步骤4-2中当集群中的旧领导者崩溃后,新领导者可以在秒级单位时间内就选举产生,并对外提供服务,从现行的单一namenode节点变成了namenode集群来负责接收所有客户端发来的读、写请求。

## 一种扁平化的高可用namenode模型的实现方法

### 技术领域

[0001] 本发明涉及一种扁平化的高可用namenode模型的实现方法,属于分布式应用技术领域。

### 背景技术

[0002] namenode也称为元数据节点,它的主要功能是管理分布式文件系统中的元数据信息。HDFS中文件的元数据信息包括命名空间、文件到数据块的映射、数据块到数据节点的映射三部分。namenode是否能保持长时间的正常工作,关系到整个分布式文件系统的可用性。

[0003] 行业中针对namenode潜在的单点故障问题而采取的解决方案大致有3类,分别是secondary Namenode机制、Backup Node机制和Avatar机制。

[0004] secondary namenode机制是在运行namenode进程的服务器上,又运行了一个secondary namenode进程。secondary namenode会定期从namenode上下载元数据镜像文件和操作日志,并将其合并为一份准完整的元数据副本,随后回传给namenode并覆盖原来的镜像文件,这一过程称为checkpoint。但checkpoint过程得到的元数据的镜像也只是准完整的,而且随着checkpoint时间变长,数据丢失的风险也会加大。

[0005] backup node机制是令namenode实时地将日志传送给backup node,即当namenode有日志时,不仅会写一份到本地日志文件中,同时还会向backup node中写一份。相较于secondary namenode每隔一段时间从namenode上下载镜像文件和操作日志,backup node可以实时地将得到操作日志合并到镜像文件中。该方案的优点在于实现了低延迟的日志复制,命名空间元数据可以实时同步更新。其缺点是块位置的映射信息未在内存同步,主备节点切换后,需要等待datanode上传自己所含的块信息,造成切换时间较长。

[0006] avatar机制由社交媒体网站FaceBook提出。avatar机制包含两个namenode节点,一个为primary namenode,另一个为standby namenode,primary namenode接替原生的namenode角色,负责响应客户端的请求并在内存中维护一份元数据信息。而standby namenode是一个一直处于safemode(安全模式)的节点,它只维护元数据信息,不接受客户端的请求。在primary namenode宕机时,standby namenode切换为primary namenode的耗时非常短。但缺点是standby namenode在primary namenode正常工作情况下,负责的工作仅仅是同步元数据信息,并不对客户端提供任何服务。

[0007] HDFS作为Hadoop的分布式文件系统,由于设计的原因在架构上只设置了一个namenode节点,而这一个namenode节点既要处理来自客户端的所有读、写请求,又要承担集群中元数据的维护管理任务。这种单一主节点配若干从节点的典型分布式应用架构模型所潜藏的单点故障问题是HDFS高可用性的一大隐患。

### 发明内容

[0008] 本发明目的在于针对上述现有技术的不足,提出了一种扁平化的高可用namenode模型的实现方法,该方法解决了分布式文件系统HDFS潜在单点故障问题,实现了负载均衡。

[0009] 本发明包括一种扁平化的三机namenode模型,该模型中包括领导者节点、候选者节点和跟随者节点三种角色的namenode节点协调工作。一个节点可能充当不止一种角色。相较于当前主/备模式的架构,本发明大大缩短了在主节点宕机后,集群重新选主并恢复服务功能的速度,提升了HDFS文件系统响应客户端读请求时的性能。

[0010] 本发明所涉及的一些关键词定义包括如下:

[0011] 领导者namenode:

[0012] 处理客户端提交的读或写请求,并完成元数据同步。一个任期内只存在一个领导者。

[0013] 候选者namenode:

[0014] 可以通过获得超过半数跟随者namenode的选票成为一个任期内的领导者。

[0015] 跟随者namenode:

[0016] 可以处理客户端提交的读请求。依据领导者的元数据来同步自己服务器上的元数据。

[0017] 任期:

[0018] 从一轮选举开始到下一轮选举开始之间称作一个任期,每一个任期都有一个唯一的编号。

[0019] 方法流程:

[0020] 领导者namenode的选举包括:

[0021] 步骤1-1:当HDFS刚启动时,所有namenode节点均进入跟随者状态,没有领导者;

[0022] 步骤1-2:如果在100ms至500ms之间的任意时刻,跟随者namenode没有接收到任何来自领导者namenode的心跳消息(不含数据信息的远程过程调用消息),它就会假定此时集群内没有可达或可用的领导者,那么该跟随者namenode就会发起选举,首先增加自己当前的任期号,创建一个比之前使用过的任何值都要大的新任期号。随即进入候选者角色,并尝试成为整个namenode集群的领导者;

[0023] 步骤1-3:候选者namenode向其他namenode服务器发送投票请求,同时自己会投给自己一票,在获得集群中超过半数namenode节点反馈的同意响应后,候选者namenode会将自已的状态转换为领导者,并立即向namenode集群中其他服务器发送心跳信息,建立领导者地位;

[0024] 中断事务包括:

[0025] 当前候选者namenode如果收到了来自于有效领导者namenode的心跳信息,它就会立即放弃成为领导者的尝试,随即回到跟随者的状态;

[0026] 候选者经过一个随机的选举超时时间后会再次自增自己的任期号,然后重启新一轮的选举,重复步骤1-3,直至集群最终产生领导者。

[0027] 客户端从HDFS上读文件包括:

[0028] 步骤3-1、客户端向namenode集群中任意一台服务器发送读请求。

[0029] 步骤3-2、接收到来自客户端读请求的namenode服务器随即去目录树中检查HDFS中是否存在该文件。

[0030] 步骤3-3、如果HDFS中不存在客户端要读的文件,则namenode服务器返回文件不存在异常,如果存在,则返回该文件对应的block及其副本所在的数据节点的列表信息。

[0031] 步骤3-4、客户端从返回的block信息列表中挑选一个网络拓扑结构中距离最近的datanode服务器并向其发送读文件请求。

[0032] 步骤3-5、被请求的datanode服务器向客户端传输文件。

[0033] 崩溃恢复包括：

[0034] 领导者namenode可能会出现崩溃或者由于网络原因失去与过半跟随者namenode的联系,为了保证日志在每台服务器节点上的完整性与一致性和整个namenode集群的高可用性,此时namenode集群就会进入崩溃恢复过程。

[0035] 步骤4-1、某些或某一个跟随者namenode会进入候选者状态,并向其他服务器发起投票请求,请求里会包含自身最后一条日志记录信息的索引(lastIndex)以及任期号(lastTerm)。

[0036] 步骤4-2、当响应投票的服务器接收到请求,它会将候选者的日志信息与自己的日志信息进行比较,如果投票者(跟随者namenode)的日志更完整：

[0037]  $(lastTerm_{follower} > lastTerm_{candidate}) ||$

[0038]  $((lastTerm_{follower} == lastTerm_{candidate}) \&\& (lastIndex_{follower} > lastTerm_{candidate}))$

[0039] 它就会拒绝投票,结果是赢得选举的namenode服务器可以保证比大多数投票者有更完整的日志记录。

[0040] 步骤4-3、经过上面的步骤已经选举出了领导者namenode,此时,领导者namenode会不断地向跟随者namenode发送包含自己日志信息的心跳消息。

[0041] 步骤4-4、跟随者namenode根据接收到的心跳消息,删除所有跟领导者namenode不同的日志记录,并将所有丢失的日志记录依照领导者的日志进行补足。

[0042] 进一步地,本发明所述步骤4-2中当集群中的旧领导者崩溃后,新领导者可以在秒级单位时间内就选举产生,并对外提供服务。相较于现行的主一备模式,大大缩短了集群崩溃恢复的时间,并且从现行的单一namenode节点变成了namenode集群来负责接收所有客户端发来的读、写请求,实现了负载均衡,提升了系统整体性能。

[0043] 有益效果：

[0044] 1、本发明的模型不仅有效地解决了集群的单点故障问题,还实现了namenode服务器处理客户端读请求时各节点的负载均衡,提升了系统整体性能。

[0045] 2、本发明提高了HDFS的高可用性,并且提高了namenode节点的高可用性,而且提升了文件系统的整体性能。

[0046] 3、本发明大大缩短了在主节点宕机后,集群重新选主并恢复服务功能的速度,提升了HDFS文件系统响应客户端读请求时的性能。

## 附图说明

[0047] 图1为客户端向namenode集群请求写文件示意图。

[0048] 图2为客户端向namenode集群请求读文件示意图。

[0049] 图3为namenode状态转换示意图。

## 具体实施方式

[0050] 下面结合说明书附图对本发明创造作进一步的详细说明。

[0051] 本发明是一种扁平化的分布式一致性日志模型。如图3所示,模型中需要三种角色的节点来协调工作:领导者(Leader)节点、候选者(Candidate)节点和跟随者(Follower)节点。在具体的实施中,一个进程可能充当不止一种角色。相较于传统的基于paxos协议的日志模型,该日志模型采用了更高效的分布式一致性协议raft,主要提高了主-从结构的分布式应用在主节点崩溃后,集群重新选主并恢复服务功能的速度。

[0052] 本发明领导者的选举实施过程包含在以下具体步骤:

[0053] 领导者namenode的选举包括:

[0054] 步骤1) 当HDFS刚启动时,所有namenode节点均进入跟随者状态,没有领导者。

[0055] 步骤2) 如果在100ms至500ms之间的任意时刻,跟随者namenode没有接收到任何来自领导者namenode的心跳消息(不含数据信息的远程过程调用消息),它就会假定此时集群内没有可达或可用的领导者,那么该跟随者namenode就会发起选举,首先增加自己当前的任期号,创建一个比之前使用过的任何值都要大的新任期号。随即进入候选者角色,尝试成为整个namenode集群的领导者。

[0056] 步骤3) 候选者namenode向其他namenode服务器发送投票请求,同时自己会投给自己一票。在获得集群中超过半数namenode节点反馈的同意响应后,候选者namenode会将自已的状态转换为领导者,并立即向namenode集群中其他服务器发送心跳信息,建立领导者地位。中断事务:

[0057] namenode集群中可能存在着其他候选者试图竞选领导者,并成功获取多数票当选为领导者。此时,当前候选者namenode如果收到了来自于有效领导者namenode的心跳信息,它就会立即放弃成为领导者的尝试,随即回到跟随者的状态。

[0058] 由于namenode集群中存在多个候选者,这些候选者namenode分摊了来自跟随者的选票,造成谁都没有获得多数票,谁都无法当选领导者的情况。解决方案是,候选者经过一个随机的选举超时时间后会再次自增自己的任期号,然后重启新一轮的选举,重复步骤3,直至集群最终产生领导者。

[0059] 客户端向HDFS上写文件包括:

[0060] 当领导者namenode被选举出来后,就可以接收来自客户端的请求,请求可以分为读请求和写请求两种类型。

[0061] 步骤1) 如图1所示,客户端向领导者提交写一个数据块的请求。

[0062] 步骤2) 领导者首先去本机内存中维护的元数据的目录树中检查客户端所请求写入的文件是否已存在于HDFS上,若没有,则会去datanode信息池中挑选副本数量个datanode服务器作为客户端可写入文件的数据节点,并将客户端申请写入HDFS的文件的元信息和挑选出来的datanode节点元信息作为一条日志发送给一致性模块。

[0063] 步骤3) 领导者namenode中的一致性模块向所有跟随者namenode同步日志。日志同步完成后将之前挑选出来的datanode数据节点列表信息返回给客户端。

[0064] 步骤4) 客户端在接收到领导者namenode返回的datanode列表信息后开始往这些datanode上写文件。

[0065] 客户端从HDFS上读文件包括:

[0066] 步骤1) 如附图2所示,客户端向namenode集群中任意一台服务器发送读请求。

[0067] 步骤2) 接收到来自客户端读请求的namenode服务器随即去目录树中检查HDFS中

是否存在该文件。

[0068] 步骤3) 如果HDFS中不存在客户端要读的文件,则namenode服务器返回文件不存在异常,如果存在,则返回该文件对应的block及其副本所在的数据节点的列表信息。

[0069] 步骤4) 客户端从返回的block信息列表中挑选一个网络拓扑结构中距离最近的datanode服务器并向其发送读文件请求。

[0070] 步骤5) 被请求的datanode服务器向客户端传输文件。

[0071] 崩溃恢复包括:

[0072] 领导者namenode可能会出现崩溃或者由于网络原因失去与过半跟随者namenode的联系,为了保证日志在每台服务器节点上的完整性与一致性和整个namenode集群的高可用性,此时namenode集群就会进入崩溃恢复过程。

[0073] 步骤1) 某些或某一个跟随者namenode会进入候选者状态,并向其他服务器发起投票请求,请求里会包含自身最后一条日志记录信息的索引(lastIndex)以及任期号(lastTerm)。

[0074] 步骤2) 当响应投票的服务器接收到请求,它会将候选者的日志信息与自己的日志信息进行比较,如果投票者(跟随者namenode)的日志更完整:

[0075]  $(lastTerm_{follower} > lastTerm_{candidate}) ||$

[0076]  $((lastTerm_{follower} == lastTerm_{candidate}) \&\& (lastIndex_{follower} > lastTerm_{candidate}))$

[0077] 它就会拒绝投票,结果是赢得选举的namenode服务器可以保证比大多数投票者有更完整的日志记录。

[0078] 步骤3) 经过上面的步骤已经选举出了领导者namenode,此时,领导者namenode会不断地向跟随者namenode发送包含自己日志信息的心跳消息。

[0079] 步骤4) 跟随者namenode根据接收到的心跳消息,删除所有跟领导者namenode不同的日志记录,并将所有丢失的日志记录依照领导者的日志进行补足。

[0080] 以上所述仅是本发明的优选实施方式,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。



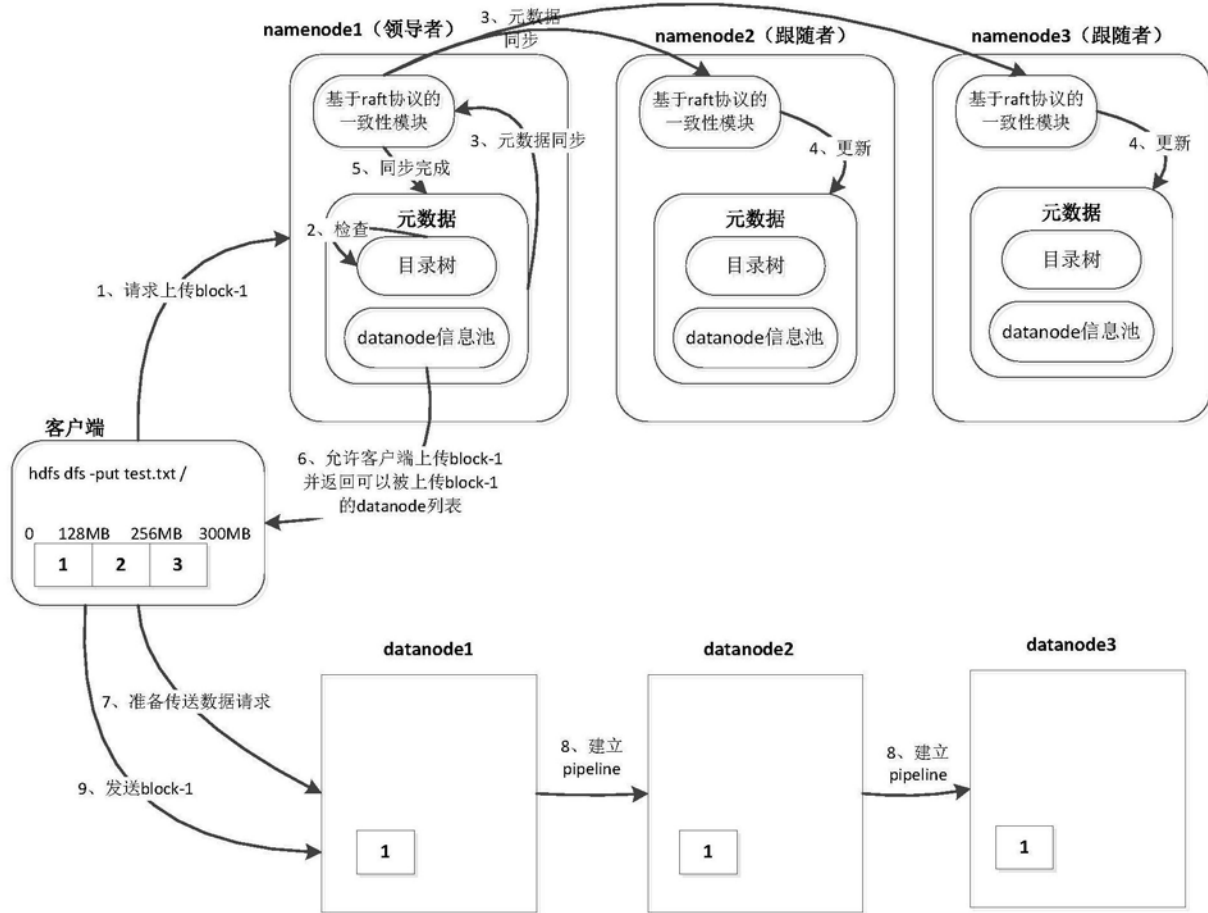


图1

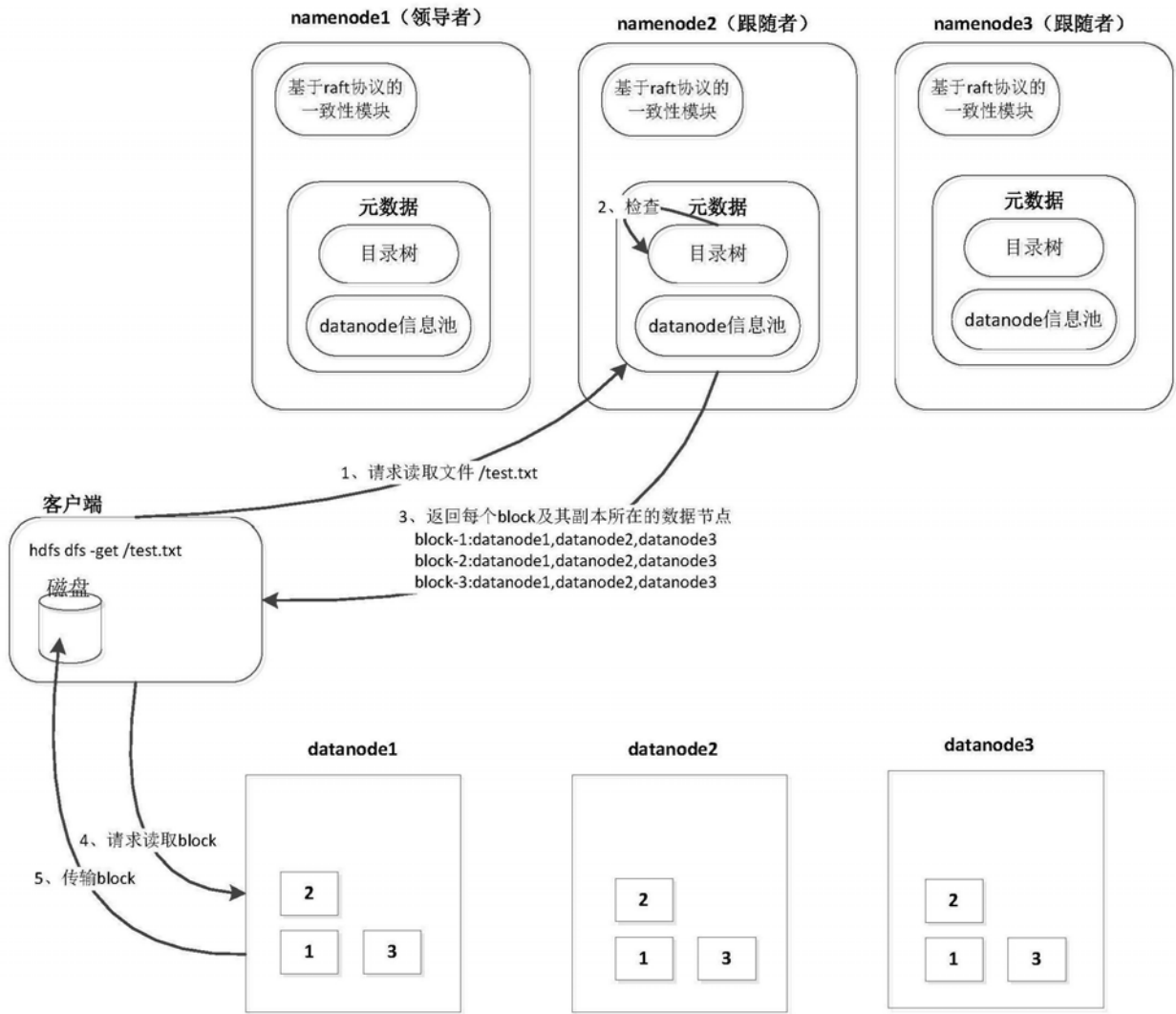


图2

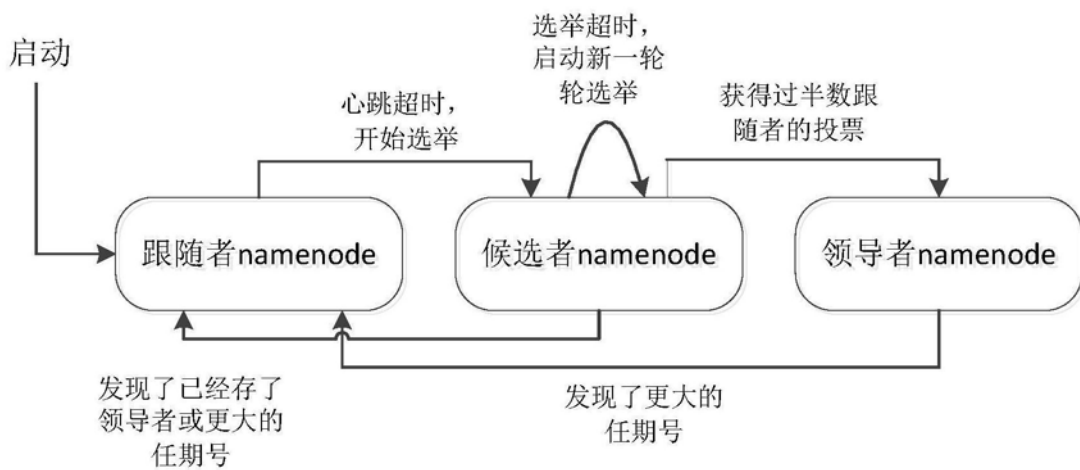


图3