(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103530200 A
(43)申请公布日 2014.01.22

(21)申请号 201210228993.9

(22)申请日 2012.07.04

(71)申请人 腾讯科技(深圳)有限公司
地址 518044 广东省深圳市福田区振兴路赛格科技园2栋东403室

(72)发明人 朱会灿 伍海君 邓大付 杨绍鹏 李锐 邹永强 赵大勇 陈晓东 王磊 阙太富 刘畅 张书鑫 张银锋 董乘宇

(74)专利代理机构 北京德琦知识产权代理有限公司 11018
代理人 张晓峰 宋志强

(51)Int.Cl.
*G06F 11/14*(2006.01)

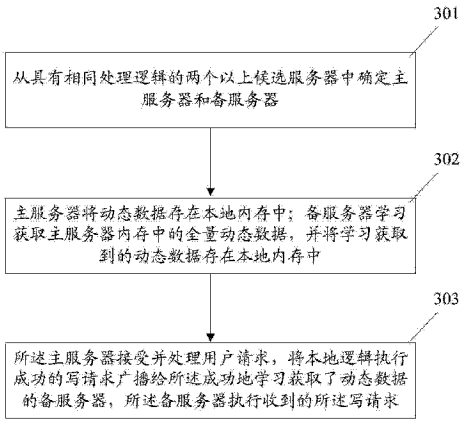
权利要求书3页 说明书7页 附图2页

(54)发明名称

一种服务器热备份系统和方法

(57)摘要

本申请公开了一种服务器热备份系统和方法。包括:选举服务器以及具有相同处理逻辑的两个以上候选服务器。所述选举服务器从所述候选服务器中确定主服务器和备服务器;所述候选服务器被确定为备服务器的情况下,用于学习获取主服务器内存中的全量动态数据,将学习获取到的动态数据存在本地内存中,执行所收到的主服务器广播的写请求;所述候选服务器被确定为主服务器的情况下,用于将动态数据存在本地内存中,接受并处理用户请求,将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器。利用本发明,可以简化热备份的处理逻辑,降低对服务器的性能压力。




CN 103530200 A

1. 一种服务器热备份系统，其特征在于，包括：选举服务器以及具有相同处理逻辑的两个以上候选服务器；

所述选举服务器用于从所述候选服务器中确定主服务器和备服务器；

所述候选服务器被确定为备服务器的情况下，用于学习获取主服务器内存中的全量的动态数据，将学习获取到的动态数据存在本地内存中，执行所收到的主服务器广播的写请求；

所述候选服务器被确定为主服务器的情况下，用于将动态数据存在本地内存中，接受并处理用户请求，将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器。

2. 根据权利要求 1 所述的系统，其特征在于，所述候选服务器具体包括选举客户端，用于在候选服务器启动后向所述选举服务器请求加锁并接收被选举为主服务器还是备服务器的结果，在正常运行时与所述选举服务器保持心跳通信；

所述选举服务器具体包括选举模块和心跳模块；

所述选举模块用于：将第一个请求加锁的候选服务器确定为主服务器，并通知该候选服务器其身份为主服务器；将确定了主服务器后请求加锁的候选服务器确定为备服务器，并通知该候选服务器其身份为备服务器以及主服务器的地址；

所述心跳模块用于：监听与所述主备服务器的心跳通信，若与主服务器的心跳通信超时，则触发选举模块取消该主服务器，从所述备服务器中重新选择一个确定为主服务器，并通知该备服务器其身份变为主服务器，并向其它备服务器通知新选的主服务器的地址；若与备服务器的心跳通信超时，则触发选举模块取消该备服务器。

3. 根据权利要求 2 所述的系统，其特征在于，所述候选服务器由备服务器被新选定为主服务器的情况下，进一步用于查看自身是否有未执行完的广播请求，如果有则先执行完所述广播请求，之后将自身转变为主服务器，接受并处理用户请求。

4. 根据权利要求 1 所述的系统，其特征在于，所述候选服务器具体包括学习模块，用于：

在所述候选服务器被确定为备服务器的情况下，向主服务器发起学习请求，接收主服务器发送的全量动态数据和广播的写请求，将所述写请求放入等候队列，在接收完所述全量动态数据后，依次执行所述等候队列中的写请求；

在所述候选服务器被确定为主服务器的情况下，接收备服务器的学习请求，收到学习请求后向发起学习请求的备服务器发送本地内存中的全量动态数据；将在发送全量动态数据的同时本地逻辑所执行成功的写请求广播给所述备服务器。

5. 根据权利要求 1 所述的系统，其特征在于，所述候选服务器具体包括广播模块，用于：

在所述候选服务器被确定为主服务器的情况下，将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器；如果在发送完广播请求后没有收到某一备服务器的响应，则后续不再向该备服务器广播本地逻辑执行成功的写请求，除非该备服务器重新学习获取了全量动态数据；

在所述候选服务器被确定为备服务器的情况下，接收所述主服务器广播的写请求，在收到写请求后立即向主服务器返回响应，并执行所述写请求。

6. 根据权利要求 1 至 5 任一项所述的系统，其特征在于，所述候选服务器为分布式文件系统中的元数据服务器，所述动态数据为分布式文件系统中的元数据。

7. 一种服务器热备份方法，其特征在于，包括：

从具有相同处理逻辑的两个以上候选服务器中确定主服务器和备服务器；

主服务器将动态数据存在本地内存中；备服务器学习获取主服务器内存中的全量动态数据，并将学习获取到的动态数据存在本地内存中；

所述主服务器接受并处理用户请求，将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器，所述备服务器执行收到的所述写请求。

8. 根据权利要求 7 所述的方法，其特征在于，所述从具有相同处理逻辑的两个以上候选服务器中确定主服务器和备服务器，具体包括：

将两个以上具有相同处理逻辑的服务器作为候选服务器；

所述候选服务器启动后向一选举服务器请求加锁，并在正常运行时与所述选举服务器保持心跳通信；

所述选举服务器将第一个请求加锁的候选服务器确定为主服务器，并通知该候选服务器其身份为主服务器；将确定了主服务器后请求加锁的候选服务器确定为备服务器，并通知该候选服务器其身份为备服务器以及主服务器的地址；

所述选举服务器监听与所述主备服务器的心跳通信，若与主服务器的心跳通信超时，则取消该主服务器，从所述备服务器中重新选择一个确定为主服务器，并通知该备服务器其身份变为主服务器，向其它备服务器通知新选的主服务器的地址；若与备服务器的心跳通信超时，则取消该备服务器。

9. 根据权利要求 8 所述的方法，其特征在于，所述取消该主服务器，从所述备服务器中选择一个作为主服务器之后，进一步包括：

所述被新选定为主服务器的候选服务器查看自身是否有未执行完的广播请求，如果有则先执行完所述广播请求，之后将自身转变为主服务器，接受并处理用户请求。

10. 根据权利要求 8 所述的方法，其特征在于，所述取消主服务器之后，从所述备服务器中重新选择一个确定为主服务器，具体包括：

按照请求加锁的顺序选择下一个备服务器为主服务器；

或者，从所有备服务器中按照预定策略选择一个作为主服务器；

或者，选举服务器通知所述与自身心跳通信正常的备服务器进行抢锁，收到通知的备服务器发出抢锁请求，选举服务器将抢锁成功的备服务器确定为主服务器。

11. 根据权利要求 7 所述的方法，其特征在于，所述备服务器学习获取主服务器内存中的全量动态数据，具体包括：

所述备服务器向主服务器发起学习请求；

主服务器收到学习请求后向发起学习请求的备服务器发送本地内存中的全量动态数据；将在发送全量动态数据的同时本地逻辑所执行成功的写请求广播给所述备服务器；

备服务器接收主服务器发送的全量动态数据和广播的写请求，将所述写请求放入等候队列，在接收完所述全量动态数据后，依次执行所述等候队列中的写请求。

12. 根据权利要求 7 所述的方法，其特征在于，

所述备服务器在收到主服务器广播的写请求后，立即回复响应；

如果主服务器在发送完广播请求后没有收到某一备服务器的响应，则后续不再向该备服务器广播本地逻辑执行成功的写请求，除非该备服务器重新学习获取了全量动态数据。

13. 根据权利要求 7 至 12 任一项所述的方法，其特征在于，所述候选服务器为分布式文件系统中的元数据服务器，所述动态数据为分布式文件系统中的元数据。

# 一种服务器热备份系统和方法

## 技术领域

[0001] 本申请涉及计算机数据处理技术领域，尤其涉及一种服务器热备份系统和方法。

## 背景技术

[0002] 热备份是指在一个数据处理系统的两台或多台服务器中，一台服务器处于激活状态(active)，其它服务器处于备份状态(standby)。用户只能访问激活状态的服务器；且用户的所有更新操作必须同步到其它处于备份状态的服务器，以保证在激活状态的服务器宕机后，任一个备份状态的服务器切换到激活状态都可以提供服务；通常情况下一主一备即可，也可采用一主多备进一步提高服务的可用性。

[0003] 目前的热备份方案都是在有状态服务器中实现热备逻辑。所述有状态服务器是指需要将数据存储在磁盘中(业界称为落地数据)。但是这种处理方案存在的缺点是：由于数据需要在磁盘中持久化，因此会存在写磁盘失败的风险。一旦数据丢失，会造成用户数据损坏或系统状态不一致等问题。因此，这种热备份方案都需要一套复杂的事务机制来保障主备提交数据的原子性，这样一来又会加重服务器的性能负担。

[0004] 例如在目前的分布式文件系统(Distributed File System)中，处理元数据的元数据服务器由于数据量大，需要持久化的信息多，这些已经导致元数据服务器的压力较大，如果按照现有的热备份方案对元数据服务器进行热备份，则会进一步加剧元数据服务器的处理压力。因此目前的分布式文件系统一般均未对元数据服务器节点进行热备份。如GFS(google File System)分布式文件系统中的master服务器节点，以及HDFS(Hadoop Distributed File System)分布式文件系统中的namenode服务器节点，均为系统单点，没有进行热备份。但是，这种没有进行热备份的单点一旦出现故障就会导致整个系统不可服务，安全性太低。

## 发明内容

[0005] 有鉴于此，本发明的主要目的在于提供一种服务器热备份系统和方法，以简化热备份的处理逻辑，降低对服务器的性能压力。

[0006] 本发明的技术方案是这样实现的：

[0007] 一种服务器热备份系统，包括：选举服务器以及具有相同处理逻辑的两个以上候选服务器；

[0008] 所述选举服务器用于从所述候选服务器中确定主服务器和备服务器；

[0009] 所述候选服务器被确定为备服务器的情况下，用于学习获取主服务器内存中的全量动态数据，将学习获取到的动态数据存在本地内存中，执行所收到的主服务器广播的写请求；

[0010] 所述候选服务器被确定为主服务器的情况下，用于将动态数据存在本地内存中，接受并处理用户请求，将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器。

[0011] 一种服务器热备份方法，包括：

[0012] 从具有相同处理逻辑的两个以上候选服务器中确定主服务器和备服务器；

[0013] 主服务器将动态数据存在本地内存中；备服务器学习获取主服务器内存中的全量动态数据，并将学习获取到的动态数据存在本地内存中；

[0014] 所述主服务器接受并处理用户请求，将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器，所述备服务器执行收到的所述写请求。

[0015] 与现有技术相比，本发明将动态数据存在主服务器的本地内存中，备服务器学习获取主服务器内存中的全量动态数据，也保存在本地内存中；主服务器如果接受并执行成功了改变了内存数据的写请求，则将该写请求广播给备服务器，备服务器执行收到的写请求，由于主备服务器具有相同的处理逻辑，因此备服务器执行完写请求后，其内存中的数据与主服务器依然保持同步，因此实现了服务器的热备份，并且在整个热备份过程中没有对磁盘进行操作，避免了写磁盘造成的延时，也不需要一套复杂的事务机制来保障数据的原子性，因此本发明简化了热备份的处理逻辑，降低对服务器的性能压力。

## 附图说明

[0016] 图 1 为本发明所述服务器热备份系统的组成示意图；

[0017] 图 2 为本发明所述服务器热备份系统的又一种具体组成示意图；

[0018] 图 3 为本发明所述服务器热备份方法的一种流程图。

## 具体实施方式

[0019] 下面结合附图及具体实施例对本发明再作进一步详细的说明

[0020] 图 1 为本发明所述服务器热备份系统的组成示意图。参见图 1，所述服务器热备份系统包括：选举服务器以及具有相同处理逻辑的两个以上候选服务器。

[0021] 所述选举服务器用于从所述候选服务器中确定主服务器和备服务器；

[0022] 所述候选服务器被确定为备服务器的情况下，用于学习获取主服务器内存中的全量动态数据，将学习获取到的动态数据存在本地内存中，执行所收到的主服务器广播的写请求；

[0023] 所述候选服务器被确定为主服务器的情况下，用于将动态数据存在本地内存中，接受并处理用户请求，将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器。此处只有写请求会改变内存中的动态数据，因此只需要向备服务器广播执行成功的写请求。

[0024] 本发明中所述的动态数据是指一个处理系统中不必永久存储的数据，或者可以在处理系统中的其它数据服务器中得到的数据，所述候选服务器为用于存储和维护这种动态数据的服务器。

[0025] 例如在分布式文件系统中的元数据就是一种动态数据。所述元数据是指文件名、文件 ID、文件长度、文件的最后修改时间、创建时间、文件块分布等信息。通过元数据，用户可以知道一个文件的基本信息，并通过元数据定位到一个分布式文件实际数据所在的数据服务器。这些元数据可以通过遍历所有数据服务器来获取得到。

[0026] 下面实施例中以分布式文件系统中的元数据服务器 MetaServer 作为热备份对象

为例进行说明。当然，本发明的方案并不限于分布式文件系统，也可以适用于其它用于存储和维护动态数据的服务器。

[0027] 本发明中的候选服务器使用无状态服务器(以下简称 MetaServer)设计，将动态数据(例如分布式文件系统中从数据服务器中收集得到的元数据)存放在 MetaServer 的本地内存中，而不需要持久化存储到 MetaServer 的磁盘。

[0028] 所述无状态服务器是指不需要落地存储(即需要存储在磁盘中)的服务器，也就是说所有的数据仅记录在本地内存中，在重启或宕机后数据可以从其它相关服务器恢复；而现有技术中的热备份方案都采用有状态服务器作为候选服务器进行热备份，所述有状态服务器是指需要落地数据，即需要将数据存储在磁盘中，一旦数据丢失，会造成用户数据损坏或系统状态不一致等问题，因此这种有状态服务器的热备份方案都需要一套复杂的事务机制来保障主备提交数据的原子性，这样一来又会加重服务器的性能负担。

[0029] 本发明将无状态服务器作为候选服务器，所有的动态数据都存储在服务器内存中，因此在整个热备份过程中没有对磁盘进行操作，避免了写磁盘造成的延时，也不需要一套复杂的事务机制来保障数据的原子性，因此本发明简化了热备份的处理逻辑，降低对服务器的性能压力。可以部署多个候选服务器互为备份，并可以通过分布式选举服务进行服务器身份选举(即确定主服务器或者备服务器)，用户始终访问主服务器或者将用户请求始终发送给主服务器。在主服务器发生故障时，可以在秒级切换到备服务器，切换速度快。

[0030] 图 2 为本发明所述服务器热备份系统的又一种具体组成示意图。参见图 2，所述选举服务器例如具体可以为一种应用程序协调服务器，如 zookeeper 服务器，所述候选服务器为分布式文件系统中的元数据服务器MetaServer。在所述候选服务器中具体包括选举客户端 201，如 zookeeper 客户端，还具体包括学习模块 202、和广播模块 203。

[0031] 所述 zookeeper 客户端 201 用于在候选服务器启动后向所述选举服务器请求加锁，所述启动包括初始启动以及故障排除后的重新启动；所述请求加锁的目的就是为了确定本候选服务器的身份是主服务器还是备服务器，因此选举客户端 201 还从选举服务器接收被选举为主服务器还是备服务器的结果(即加锁请求的结果)，并在候选服务器正常运行时与所述选举服务器保持心跳通信。

[0032] 所述选举服务器具体包括选举模块 204 和心跳模块 205。

[0033] 所述选举模块 204 用于：将第一个请求加锁的候选服务器确定为主服务器，并通知该候选服务器为主服务器，将确定了主服务器后请求加锁的候选服务器确定为备服务器，并通知该候选服务器为备服务器。具体的，可以在选举模块 204 中维护一个主备视图，所述主备视图是一个信息列表，其中存储有选举服务器所确定的主服务器 IP 地址和所有备服务器的 IP 地址，当确定某一候选服务器为备服务器后，还需要将主服务器的 IP 地址通知给该备服务器，以供该备服务器向主服务器学习获取全量动态数据。

[0034] 本发明中，所述选举确定的主服务器只有一个，备服务器可以有一个以上。

[0035] 所述心跳模块 205 用于监听与所述主备服务器的心跳通信，以此来判断主备服务器是否发生故障，具体包括：

[0036] 若心跳模块 205 监听到与主服务器的心跳通信超时，则说明主服务器故障，此时需要触发选举模块 204 取消该主服务器，即从所述主备视图中将主服务器的 IP 地址移除，然后从所述备服务器中重新选择一个备服务器确定为主服务器，修改主备视图中的主服务

器 IP 信息，并将主备变化的信息通知给所有的备服务器，具体是通知被新选为主服务器的备服务器其身份已经变为了主服务器，还要将新选的主服务器的 IP 地址通知给其他备服务器。

[0037] 若心跳模块 205 监听到与备服务器的心跳通信超时，则说明该被服务器故障，此时需要触发选举模块 204 取消该备服务器，即从所述主备视图中将该备服务器的 IP 地址移除。

[0038] 这样一旦所述心跳通信超时的原主服务器故障排除并重新启动后，则会向所述选举服务器请求加锁，由于此时已经又选择了一个主服务器，因此此时选举服务器将该候选服务器（即原来的主服务器）确定为备服务器。

[0039] 所述候选服务器由备服务器被新选定为主服务器的情况下，还需要进一步用于首先查看自身是否有未执行完的广播请求，如果有则先执行完所述广播请求，之后将自身转变为主服务器，接受并处理用户请求。

[0040] 在确定了主备服务器之后，用户可以通过主服务器访问所述 MetaServer 所提供的服务。用户访问主服务器的具体方式可以多种：

[0041] 例如可以在系统中专门设置一个代理访问服务器，选举服务器将所述主服务器的 IP 地址通知给该代理访问服务器，在新选主服务器后也要将该新选主服务器的 IP 地址通知给该代理访问服务器，用户客户端只知道该代理访问服务器的 IP 地址，因此用户的请求发送给该代理访问服务器，该代理访问服务器再将所述用户请求发送给主服务器。

[0042] 例如，用户客户端也可以直接访问所述选举服务器请求主服务器的 IP 地址，利用选举服务器返回的 IP 地址直接访问主服务器。当主服务器发生切换后，用户客户端访问主服务器超时，或访问的主服务器明确返回自己身份不是主服务器的时候，用户客户端再到所述选举服务器请求新选的主服务器的 IP 地址，以进行正确的访问。

[0043] 所述被选定为主服务器的 MetaServer 在向备服务器发送全量的元数据之前，需要遍历分布式文件系统中的数据服务器收集元数据，待收集完成后再响应备服务器发送的学习请求。

[0044] 所述候选服务器中的学习模块 202 用于执行主备学习过程，具体用于：

[0045] 在所述候选服务器被确定为备服务器的情况下，该学习模块 202 向主服务器发起学习请求（其中包括备服务器的 IP 地址），接收主服务器发送的全量动态数据和广播的写请求，将所述写请求放入等候队列，在接收完所述全量动态数据后，依次执行所述等候队列中的写请求；

[0046] 在所述候选服务器被确定为主服务器的情况下，该学习模块 202 接收备服务器的学习请求，收到学习请求后向发起学习请求的备服务器发送本地内存中的全量动态数据；由于在发送全量动态数据的同时有可能会收到并执行了新的写请求，导致动态数据变化，因此学习模块 202 还需要将在发送全量动态数据的同时本地逻辑所执行成功的写请求广播给所述备服务器。在发送完全量的动态数据后主服务器认为该备服务器已经成功地学习获取了动态数据，可以记录该备服务器的 IP 地址，例如具体的实现方式是在所述选举服务器中的主备视图中为该备服务器的 IP 地址加一个学习标记，标明该 IP 地址已经成功地学习获取了动态数据，主服务器可以根据该主备视图中的有学习标记的备服务器 IP 地址向这些备服务器广播写请求。当某一备服务器切换为新的主服务器后可以从选举服务器的主

备视图中获取所有有学习标记的备服务器的 IP 地址，并向这些备服务器广播写请求。

[0047] 所述候选服务器的广播模块 203 用于执行主备广播过程，具体用于：

[0048] 在所述候选服务器被确定为主服务器的情况下，所述广播模块 203 将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器；如果在发送完广播请求后，没有收到某一备服务器的响应，则后续不再向该备服务器广播本地逻辑执行成功的写请求，除非该备服务器重新学习获取了全量动态数据。具体的，在超过一预定时间没有收到某一备服务器的响应后需要将选举服务器中的所述主备视图中的该备服务器 IP 地址的学习标记去掉，直到该备服务器重新学习获取了全量的动态数据，再将所述主备视图中该备服务器 IP 地址加上学习标记。从而保证动态数据在主备服务器中的严格同步。

[0049] 在所述候选服务器被确定为备服务器的情况下，所述广播模块 203 接收所述主服务器广播的写请求，在收到写请求后立即向主服务器返回响应，并执行所述写请求。

[0050] 与上述服务器热备份系统对应，本发明还公开了一种服务器热备份方法。图 3 为本发明所述服务器热备份方法的一种流程图。参见图 3，包括主备确定过程 301、主备学习过程 302 和主备广播过程 303；

[0051] 所述主备确定过程 301 包括：从具有相同处理逻辑的两个以上候选服务器中确定主服务器和备服务器；

[0052] 所述主备学习过程 302 包括：主服务器将动态数据存在本地内存中；备服务器学习获取主服务器内存中的全量动态数据，并将学习获取到的动态数据存在本地内存中；

[0053] 所述主备广播过程 303 包括：所述主服务器接受并处理用户请求，将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器，所述备服务器执行收到的所述写请求。

[0054] 下面实施例中以分布式文件系统中的元数据服务器 MetaServer 作为热备份对象为例进行说明，所述动态数据为分布式文件系统中的元数据。当然，本发明的方案并不限于分布式文件系统，也可以适用于其它用于存储和维护动态数据的服务器。

[0055] 在一种具体实施例中，在所述主备确定过程 301 中具体通过抢锁竞争选定主备服务器，具体包括：

[0056] 将两个以上具有相同处理逻辑的 MetaServer 作为候选服务器；所述候选服务器中安装有所述 zookeeper 客户端。

[0057] 所述候选服务器启动后利用所述 zookeeper 客户端向一选举服务器，此处为 zookeeper 服务器，请求加锁，并在正常运行时与所述选举服务器保持心跳通信。

[0058] 所述选举服务器将第一个请求加锁的候选服务器设置为加锁成功，即确定为主服务器，并通知该候选服务器其身份为主服务器；将确定了主服务器后请求加锁的候选服务器设置为加锁失败，即确定为备服务器，并通知该候选服务器其身份为备服务器以及主服务器的 IP 地址。具体的，可以在选举服务器维护一个主备视图，其中存储有选举服务器所确定的主服务器 IP 地址和所有备服务器的 IP 地址，当确定某一候选服务器为备服务器后，需要将主服务器的 IP 地址通知给该备服务器，以供该备服务器向主服务器学习获取全量动态数据。

[0059] 所述选举服务器需要监听与所述主备服务器的心跳通信，以此来判断主备服务器是否发生故障。

[0060] 若选举服务器监听到与主服务器的心跳通信超时，则说明主服务器故障，此时需要取消该主服务器，即从所述主备视图中将主服务器的 IP 地址移除，然后从所述备服务器中重新选择一个备服务器确定为主服务器，修改主备视图中的主服务器 IP 信息，并将主备变化的信息通知给所有的备服务器，具体是通知被新选为主服务器的备服务器其身份已经变为了主服务器，还要将新选的主服务器的 IP 地址通知给其他备服务器。

[0061] 所述与主服务器的心跳通信超时后，从所述备服务器中重新选择一个确定为主服务器的具体确定方法可以是：

[0062] 按照主备视图中的请求加锁的顺序选择下一个备服务器为主服务器；或者从备服务器中按照预定策略选择一个作为主服务器；再或者，选举服务器通知所述与自身心跳通信正常的备服务器进行抢锁，收到通知的备服务器发出抢锁请求，选举服务器将抢锁成功的备服务器确定为主服务器。

[0063] 若选举服务器监听到与备服务器的心跳通信超时，则说明该被服务器故障，此时需要取消该备服务器，即从所述主备视图中将该备服务器的 IP 地址移除。

[0064] 这样一旦所述心跳通信超时的原主服务器故障排除并重新启动后，则会向所述选举服务器请求加锁，由于此时已经又选择了一个主服务器，因此此时选举服务器将该候选服务器(即原来的主服务器)确定为备服务器。

[0065] 所述候选服务器由备服务器被新选定为主服务器的情况下，还需要进一步用于首先查看自身是否有未执行完的广播请求，即当自己身份为备服务器时，主服务器转发给自身的写请求，如果有则先执行完所述广播请求，之后将自身转变为主服务器，接受并处理用户请求。

[0066] 在确定了主备服务器之后，用户可以通过主服务器访问所述 MetaServer 所提供的服务。用户访问主服务器的具体方式可以多种：

[0067] 例如可以在系统中专门设置一个代理访问服务器，选举服务器将所述主服务器的 IP 地址通知给该代理访问服务器，在新选主服务器后也要将该新选主服务器的 IP 地址通知给该代理访问服务器，用户客户端只知道该代理访问服务器的 IP 地址，因此用户的请求发送给该代理访问服务器，该代理访问服务器再将所述用户请求发送给主服务器。

[0068] 例如，用户客户端也可以直接访问所述选举服务器请求主服务器的 IP 地址，利用选举服务器返回的 IP 地址直接访问主服务器。当主服务器发生切换后，用户客户端访问主服务器超时，或访问的主服务器明确返回自己身份不是主服务器的时候，用户客户端再到所述选举服务器请求新选的主服务器的 IP 地址，以进行正确的访问。

[0069] 所述候选服务器在发送加锁请求并接收到选举服务器返回的加锁失败的响应后，确认自身为备服务器，此后该备服务器需要学习获取主服务器内存中的全量动态数据，具体包括以下步骤 121 至步骤 123：

[0070] 121、所述备服务器向主服务器发起学习请求，该学习请求中包括该备服务器的 IP 地址。

[0071] 122、主服务器收到学习请求后向发起学习请求的备服务器发送本地内存中的全量动态数据；由于在发送全量动态数据的同时有可能会收到并执行了新的写请求，导致动态数据变化，因此本步骤还需要将在发送全量动态数据的同时本地逻辑所执行成功的写请求广播给所述备服务器。

[0072] 具体的，所述被选定为主服务器的 MetaServer 在向备服务器发送全量的元数据之前，需要遍历分布式文件系统中的数据服务器收集元数据，待收集完成后再响应备服务器发送的学习请求，即再向发起学习请求的备服务器发送全量的元数据。

[0073] 所述元数据中包括所有的文件 id、文件的长度信息、文件的创建时间、修改时间、以及所有的文件块分布序列等信息，主服务器中的内存中的数据按照文件 id 进行排序，在发送全量元数据时，每次按顺序发送部分文件 id 相关的信息，并附带发送一个标志标明是否还有数据未发送，备服务器根据这个标志来判断全量数据是否发送完成。

[0074] 123、备服务器接收主服务器发送的全量动态数据和广播的写请求，将所述写请求放入等候队列，在接收完所述全量动态数据后，依次执行所述等候队列中的写请求。

[0075] 在发送完全量的动态数据后主服务器认为该备服务器已经成功地学习获取了动态数据，可以记录该备服务器的 IP 地址，例如具体的实现方式是在所述选举服务器中的主备视图中为该备服务器的 IP 地址加一个学习标记，标明该 IP 地址的备服务器已经成功地学习获取了动态数据，主服务器可以根据该主备视图中的有学习标记的备服务器 IP 地址向备服务器广播写请求。当某一备服务器切换为新的主服务器后可以从选举服务器的主备视图中获取所有有学习标记的备服务器的 IP 地址，并向这些备服务器广播写请求。

[0076] 在所述主备广播过程 303 中，具体包括一下步骤 131 至步骤 134 ：

[0077] 131、主服务器接收到用户的写请求，执行该写请求。

[0078] 132、主服务器如果执行所述写请求失败则向用户返回相应的失败信息，执行成功则将该写请求广播给已经成功学习获取了全量数据的备服务器。具体可以从选举服务器的主备视图中查询到哪些备服务器已经成功学习获取了全量动态数据。

[0079] 133、备服务器接收到主服务器广播的写请求，立即返回响应，然后再执行该写请求，所述写请求导致的数据变化就可以同步到备服务器的内存中了。如果备服务器执行该写请求失败了，则说明该服务器上的处理逻辑程序有 bug，因此需要上报告警，并丢弃内存中的所有数据，重新从主服务器学习获取全量数据。

[0080] 134、主服务器如果在发送完广播请求后没有收到某一备服务器的响应，例如在超出某一预定时间后没有收到响应或者在重试预定次数后依然没有收到响应，则后续不再向该备服务器广播本地逻辑执行成功的写请求，除非该备服务器重新学习获取了全量动态数据。具体的，在超过一预定时间没有收到某一备服务器的响应后需要将选举服务器中的所述主备视图中的该备服务器 IP 地址的学习标记去掉，直到该备服务器重新学习获取了全量的动态数据，再将所述主备视图中该备服务器 IP 地址加上学习标记。从而保证动态数据在主备服务器中的严格同步，保证数据的一致性。

[0081] 以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明保护的范围之内。

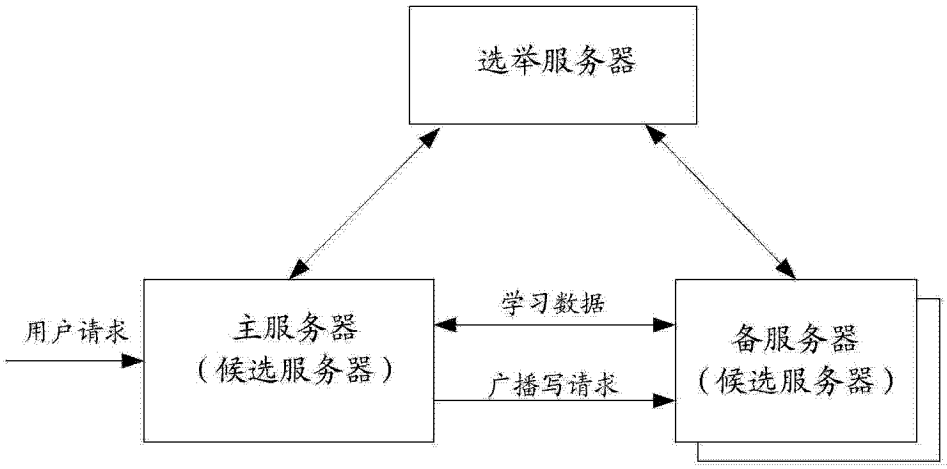

选举服务器
用户请求
主服务器
（候选服务器）
学习数据
广播写请求
备服务器
（候选服务器）

图 1

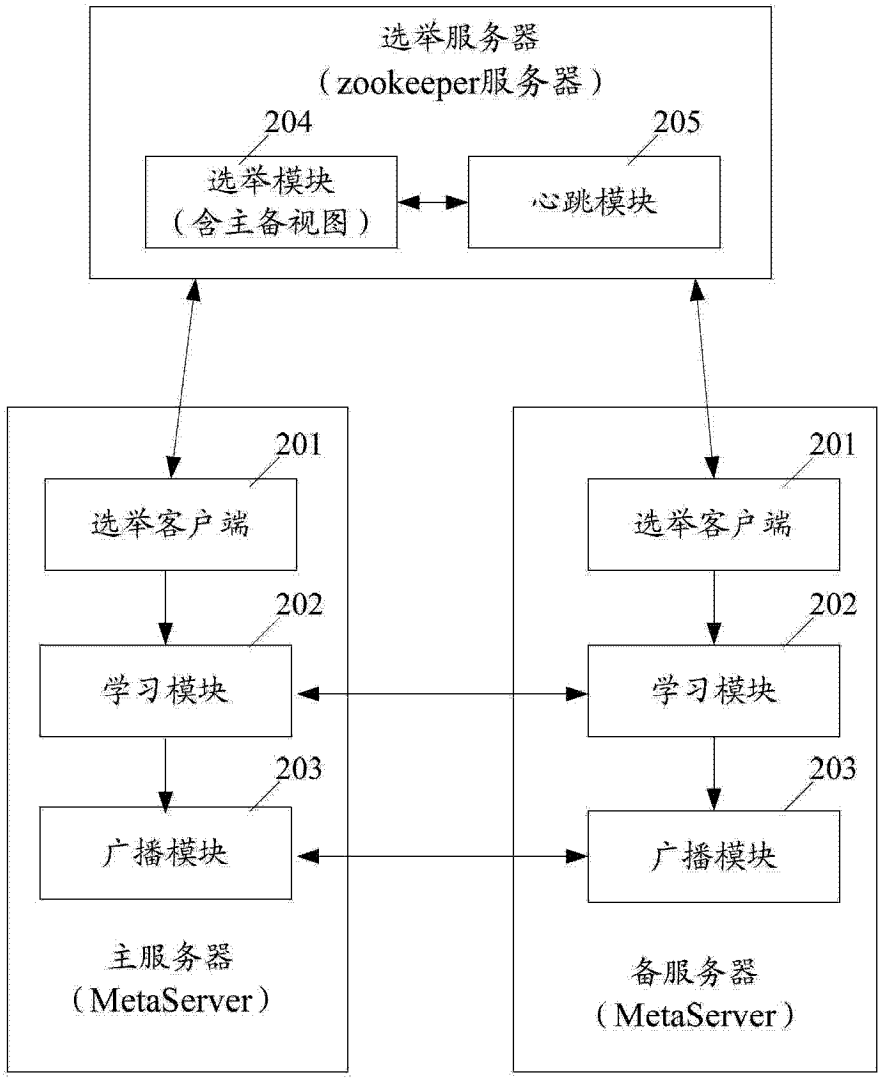

选举服务器
（zookeeper服务器）
204
选举模块
（含主备视图）
205
心跳模块
201
选举客户端
202
学习模块
203
广播模块
主服务器
（MetaServer）
201
选举客户端
202
学习模块
203
广播模块
备服务器
（MetaServer）

图 2

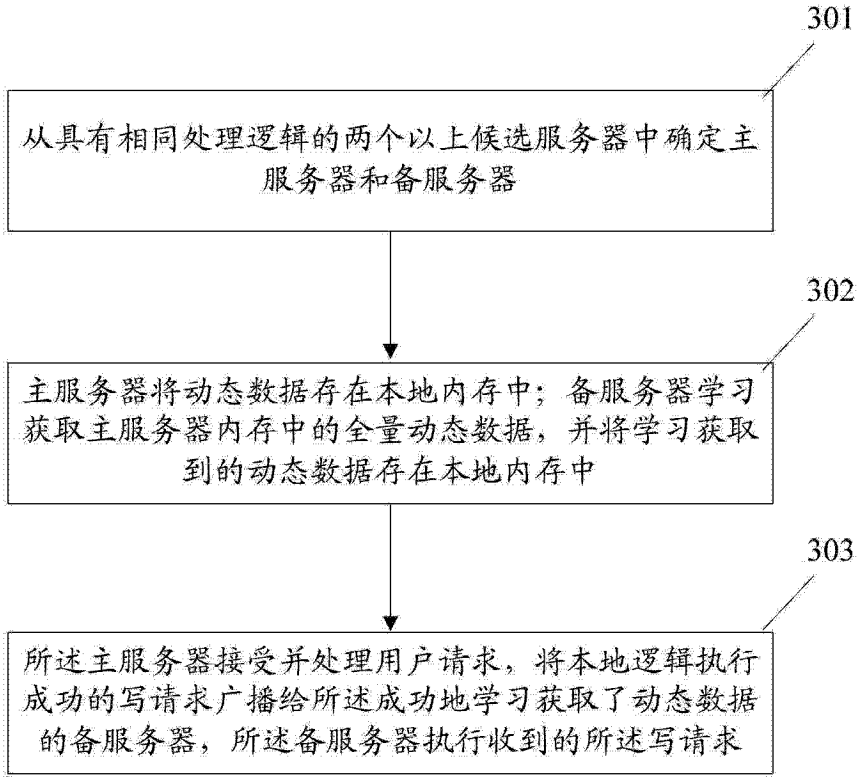

301
从具有相同处理逻辑的两个以上候选服务器中确定主服务器和备服务器
302
主服务器将动态数据存在本地内存中；备服务器学习获取主服务器内存中的全量动态数据，并将学习获取到的动态数据存在本地内存中
303
所述主服务器接受并处理用户请求，将本地逻辑执行成功的写请求广播给所述成功地学习获取了动态数据的备服务器，所述备服务器执行收到的所述写请求

图 3