(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 103944958 A
(43)申请公布日 2014.07.23

(21)申请号 201410095627.X

(22)申请日 2014.03.14

(71)申请人 中国科学院计算技术研究所
地址 100190 北京市海淀区中关村科学院南路 6 号
申请人 天津中科蓝鲸信息技术有限公司

(72)发明人 马留英 刘浏 许鲁 刘振军 闫鹏飞 蔡杰明 何文婷

(74)专利代理机构 北京律诚同业知识产权代理有限公司 11006
代理人 祁建国 梁挥

(51)Int.Cl.
H04L 29/08(2006.01)
G06F 17/30(2006.01)

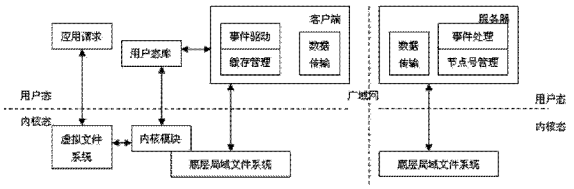
权利要求书2页 说明书11页 附图4页

(54)发明名称

一种广域文件系统及实现方法

(57)摘要

本发明公开了一种广域文件系统及实现方法，涉及文件系统技术领域，是在广域上适用的文件系统的一种实现方法。该系统包括：通过事件驱动模块接收操作请求，则根据所请求文件的节点号，通过缓存管理模块查询本地缓存中是否存在该文件，若存在，则从缓存中访问该文件，若不存在，则根据该节点号通过节点管理模块查询该文件在服务器端数据库中的路径名，并根据操作请求信息通过事件处理模块获得该文件和该文件元信息，将该文件和该文件元信息发送到缓存管理模块，缓存管理模块将该文件和该文件的元信息存入本地缓存。本发明不依赖于底层文件系统，仅要求底层文件系统可共享即可，并提供缓存的空间管理功能，提高了系统的资源利用率。



CN 103944958 A

1. 一种广域文件系统，其特征在于，服务器端包括节点号管理模块、数据传输模块、事件处理模块；客户端包括事件驱动模块、数据传输模块、缓存管理模块，

其中，通过该事件驱动模块接收操作请求，如果该请求为读取请求，则根据所请求广域文件的节点号和区间信息，通过该缓存管理模块查询本地缓存中是否存在该广域文件，若存在，则从该本地缓存中读取该广域文件，若不存在，则根据该节点号通过该节点号管理模块查询该广域文件在该服务器端数据库中的路径名，并根据该区间信息通过该事件处理模块获得该广域文件的数据信息，将该数据信息发送到该缓存管理模块，该缓存管理模块根据缓存更新策略将该数据信息及与其相对应的区间信息存入该本地缓存，以方便读取，该服务器端与该客户端通过各自的数据传输模块进行信息传递。

2. 如权利要求 1 所述的广域文件系统，其特征在于，还包括该事件驱动模块接收创建广域文件请求，将新广域文件在该服务器端数据库中的父目录节点号和该新广域文件的名称发送给该节点号管理模块，该节点号管理模块根据该父目录节点号查询该服务器端数据库获取该父目录的路径名，根据该路径名定位该父目录在该服务器端底层局域文件系统中的位置，并在该位置下根据该名称创建该新广域文件并获取该新创建文件的元信息，同时该节点号管理模块为该新创建广域文件分配节点号，该事件处理模块将该分配的节点号和该元信息发送到该缓存管理模块存入该本地缓存。

3. 如权利要求 1 所述的广域文件系统，其特征在于，还包括该事件驱动模块接收向广域文件写入请求，根据该广域文件的节点号定位该广域文件在该本地缓存中的位置，将要写入数据与该写入请求的区间信息写入该本地缓存，并标记该本地缓存中该区间信息为“1”，以表示该要写入数据仅存在于该本地缓存中，与该服务器端的该广域文件中的数据有差别。

4. 如权利要求 3 所述的广域文件系统，其特征在于，还包括数据回刷，具体步骤为：当启动数据回刷后，根据该节点号定位该广域文件在服务器端底层局域文件系统中的位置，将该要写入数据写入该服务器端该广域文件，并标记该本地缓存中该区间信息为“2”，以表示该本地缓存中该要写入数据与该服务器端的该广域文件中的数据一致。

5. 如权利要求 1 所述的广域文件系统，其特征在于，还包括该事件驱动模块接收删除广域文件请求，将待删除广域文件在该服务器端数据库中的父目录节点号和该待删除广域文件的名称发送给该节点号管理模块，该节点管理模块根据该父目录节点号定位该父目录在服务器端局域文件系统中的位置，通过事件处理模块删除该位置下具有该名称的广域文件，同时通过该缓存管理模块查询该待删除广域文件是否被引用，如果是，则将该本地缓存中该待删除广域文件隐藏，当引用结束后删除该本地缓存中的该待删除广域文件，如果否，则删除该本地缓存中该待删除广域文件。

6. 如权利要求 1 所述的广域文件系统，其特征在于，该缓存更新策略包括：记录该本地缓存中广域文件元信息的引用计数，当该引用计数变为 0 时，将该元信息从该本地缓存中替换出去，并存入数据库。

7. 如权利要求 6 所述的广域文件系统，其特征在于，该缓存更新策略还包括：记录该客户端从该服务器端获取元信息时的系统时间，通过该系统时间与访问该本地缓存中该元信息时的系统时间进行比较，若差值小于超时时间，则该本地缓存中的该元信息有效且与该元信息相对应的数据有效，并从该本地缓存中获取该元信息与该数据，否则从该服务器

端获取新元信息，比较该本地缓存中该元信息的修改时间与该新元信息的修改时间是否相同，如果相同，则该数据有效，否则该数据失效。

8. 如权利要求 6 所述的广域文件系统，其特征在于，该缓存更新策略还包括：检测已使用的该本地缓存空间是否已经达到设定的上限值，当达到该上限值时，则唤醒异步清理线程，该线程删除元信息已经被替换到数据库中的广域文件所占用的数据缓存，如果剩余该本地缓存空间达到设定的下限值，则该线程进入睡眠状态，否则根据广域文件的访问时间，删除该本地缓存中最久未被访问广域文件的数据缓存，直到剩余缓存空间达到该下限值。

9. 一种采用权利要求 1-9 任意一项的实现方法。

一种广域文件系统及实现方法

技术领域

[0001] 本发明属于文件系统(file system)技术领域,是在广域上适用的文件系统的一种实现方法。

背景技术

[0002] 随着信息技术深入社会生活的方方面面,分布于全球的数据总量正以几何速度飞速增长。在全球数据量爆炸增长的背景下,下一代数据中心、全球化的企业、云存储系统都要求能够有效并可靠地对广域分布的数据进行共享访问,因此,高效可靠的广域存储系统显得尤为重要。

[0003] 虽然有一些大规模集群文件系统例如GPFS,Lustre等,还有一些在互联网领域广泛应用的文件系统例如GFS,HDFS等,都能够在访问带宽和容量方面实现扩展,对大量客户端提供PB级数据的访问支持,但上述文件系统在设计时均为面向局域的网络环境,而在广域的网络环境下,文件系统在设计实现时必须考虑两个关键约束:相对于局域环境,广域环境的低带宽、高延迟、网络抖动等特点都会对广域文件的访问性能和可用性带来挑战,并且广域数据中心往往已经处于使用状态并可能使用异构的局域文件系统,这就需要广域文件系统的设计具有良好的异构兼容性和较小的实施成本。

[0004] 目前,面向广域环境设计的广域存储系统,比如Amazon Dynamo设计和实现为一个高度可用的key-value存储系统,其提供了一个简单的主键作为唯一的接口。Dynamo在Amazon中得到了成功的应用,并能够跨数据中心部署于上万个节点上提供服务。Yahoo的数据存储平台PNUTS主要设计为在线存储服务,主要解决在线的大量单记录或者小范围记录集合的读和写访问,支持简单的关系型数据模型,对数据对象的访问是以key为id并且读写操作对应整条记录,并允许应用程序对广域环境下的一致性与性能之间做出选择,包括读取记录的最新版,某个版本,或者不低于某个指定版本的任意版本。微软的云存储平台Windows Azure Platform提供面向对象的存储模式,主要提供了三种数据存储方式(Blob、Table和Queue)以满足应用程序的不同需求,并支持用户使用因特网登录系统来使用存储。上述三种存储系统均是开发一个供内部多种业务逻辑进行数据访问的存储服务,并不能够提供在多地异构数据中心之间进行文件共享访问服务,并且不支持POSIX文件访问接口标准。

发明内容

[0005] 本发明的目的在于提供一种支持跨地域数据访问的广域文件系统及实现方法。本发明是典型的C/S(客户端/服务器端)模式,包含一个客户端和一个服务器端。在服务器端仅需要设置导出目录后,在客户端就可以看到与广域上的服务器端导出目录一致的目录视图。客户端通过使用标准的FUSE KERNEL(FUSE内核模块)和FUSE LIB(FUSE用户态库)来兼容标准的POSIX API接口,支持对远程(即服务器端)文件及目录使用标准的系统调用(例如:open/creat/mkdir/unlink/rmdir/read/write/opendir/readdir,etc.)进行存

取访问。

[0006] 具体地来讲，本发明公开了一种广域文件系统，服务器端包括节点号管理模块、数据传输模块、事件处理模块；客户端包括事件驱动模块、数据传输模块、缓存管理模块，

[0007] 其中，通过该事件驱动模块接收操作请求，如果该请求为读取请求，则根据所请求广域文件的节点号和区间信息，通过该缓存管理模块查询本地缓存中是否存在该广域文件，若存在，则从该本地缓存中读取该广域文件，若不存在，则根据该节点号通过该节点号管理模块查询该广域文件在该服务器端数据库中的路径名，并根据该区间信息通过该事件处理模块获得该广域文件的数据信息，将该数据信息发送到该缓存管理模块，该缓存管理模块根据缓存更新策略将该数据信息及与其相对应的区间信息存入该本地缓存，以方便读取，该服务器端与该客户端通过各自的数据传输模块进行信息传递。

[0008] 所述的广域文件系统，还包括该事件驱动模块接收创建广域文件请求，将新广域文件在该服务器端数据库中的父目录节点号和该新广域文件的名称发送给该节点号管理模块，该节点号管理模块根据该父目录节点号查询该服务器端数据库获取该父目录的路径名，根据该路径名定位该父目录在该服务器端底层局域文件系统中的位置，并在该位置下根据该名称创建该新广域文件并获取该新创建文件的元信息，同时该节点号管理模块为该新创建广域文件分配节点号，该事件处理模块将该分配的节点号和该元信息发送到该缓存管理模块存入该本地缓存。

[0009] 所述的广域文件系统，还包括该事件驱动模块接收向广域文件写入请求，根据该广域文件的节点号定位该广域文件在该本地缓存中的位置，将要写入数据与该写入请求的区间信息写入该本地缓存，并标记该本地缓存中该区间信息为“1”，以表示该要写入数据仅存在于该本地缓存中，与该服务器端的该广域文件中的数据有差别。

[0010] 所述的广域文件系统，还包括数据回刷，具体步骤为：当启动数据回刷后，根据该节点号定位该广域文件在服务器端底层局域文件系统中的位置，将该要写入数据写入该服务器端该广域文件，并标记该本地缓存中该区间信息为“2”，以表示该本地缓存中该要写入数据与该服务器端的该广域文件中的数据一致。

[0011] 所述的广域文件系统，还包括该事件驱动模块接收删除广域文件请求，将待删除广域文件在该服务器端数据库中的父目录节点号和该待删除广域文件的名称发送给该节点号管理模块，该节点管理模块根据该父目录节点号定位该父目录在服务器端局域文件系统中的位置，通过事件处理模块删除该位置下具有该名称的广域文件，同时通过该缓存管理模块查询该待删除广域文件是否被引用，如果是，则将该本地缓存中该待删除广域文件隐藏，当引用结束后删除该本地缓存中的该待删除广域文件，如果否，则删除该本地缓存中该待删除广域文件。

[0012] 所述的广域文件系统，该缓存更新策略包括：记录该本地缓存中广域文件元信息的引用计数，当该引用计数变为 0 时，将该元信息从该本地缓存中替换出去，并存入数据库。

[0013] 所述的广域文件系统，该缓存更新策略还包括：记录该客户端从该服务器端获取元信息时的系统时间，通过该系统时间与访问该本地缓存中该元信息时的系统时间进行比较，若差值小于超时时间，则该本地缓存中的该元信息有效且与该元信息相对应的数据有效，并从该本地缓存中获取该元信息与该数据，否则从该服务器端获取新元信息，比较该本

地缓存中该元信息的修改时间与该新元信息的修改时间是否相同，如果相同，则该数据有效，否则该数据失效。

[0014] 所述的广域文件系统，该缓存更新策略还包括：检测已使用的该本地缓存空间是否已经达到设定的上限值，当达到该上限值时，则唤醒异步清理线程，该线程删除元信息已经被替换到数据库中的广域文件所占用的数据缓存，如果剩余该本地缓存空间达到设定的下限值，则该线程进入睡眠状态，否则根据广域文件的访问时间，删除该本地缓存中最久未被访问广域文件的数据缓存，直到剩余缓存空间达到该下限值。

[0015] 由以上方案可知，本发明的优点在于：

[0016] 系统的架构具有良好的通用性：支持基于标准的可兼容的 POSIX 接口；不依赖于底层文件系统，可兼容不同厂商的异构局域文件系统，仅要求底层文件系统可共享即可。不需要繁琐的导入，仅需在服务器端设置导出目录，在客户端就能够看到与服务器端相同的目录视图。同时采用了持久化的局域化缓存，分别缓存服务器端对象的元信息和数据信息，提高远程热点数据的访问性能，降低广域网络上的交互流量。通过基于时效的缓存的有效性判定，更适合广域网络环境及应用需求，能够支持在广域网络的不稳定性的情况下，已缓存的文件能够提供持续的访问。并提供缓存的空间管理功能，提高了系统的资源利用率。

附图说明

[0017] 图 1 为系统结构图；

[0018] 图 2 为系统整体流程图；

[0019] 图 3 为读取目录流程图；

[0020] 图 4 为创建文件流程图；

[0021] 图 5 为删除对象流程图；

[0022] 图 6 为读数据流程图；

[0023] 图 7 为写入数据流程图。

[0024] 其中，附图标记为：

[0025] 模块 1 为客户端，包括模块 11/12/13；

[0026] 模块 2 为服务器端，包括模块 22/23/11；

[0027] 步骤 100 为读取目录步骤，包括步骤

[0028] 101/102/103/104/105/106/107/108/109/110/111/112/113/114/115/116/117/118/119/120

[0029] 步骤 200 为创建文件步骤，包括步骤

[0030] 201/202/203/204/205/206/207/208/209/210/211/212/213/214/215/216/217/218

[0031] 步骤 300 为删除对象步骤，包括步骤

[0032] 301/302/303/304/305/306/307/308/309/310/311/312/313/314/315/316/317/318/319/320

[0033] 步骤 400 为读数据步骤，包括步骤

[0034] 401/402/403/404/405/406/407/408/409/410/411/412/413/414/415/416/417/418/419/420

[0035] 步骤 500 为写入数据步骤，包括步骤

[0036] 501/502/503/504/505/506/507/508/509/510/511/512/513/514/515/516/517/518/519

具体实施方式

[0037] 如图 1 所示，服务器端 2 包含节点号管理模块 23、数据传输模块 11、事件处理模块 22。所述客户端 1 包含事件驱动模块 12、数据传输模块 11、缓存管理模块 13。以下为各模块的具体用途：

[0038] 节点号管理模块 23，用于能快速定位到操作对象，且为了便于管理对象，该模块 23 为对象（object，指文件或目录）产生一个全局唯一的标识，在整个对象的生命周期内保持不变，且是与客户端 1 进行通信的对象的唯一标识。该模块 23 提供对节点号的增加、查询、删除的操作。该模块 23 在创建对象时，为创建对象分配唯一的节点号，产生对应于该节点号的一条记录，记录该节点号与实际对象在服务器端 2 数据库中路径名的映射关系，并将该条记录存储在数据库中；在删除对象时，在数据库中将待删除对象的节点号所对应的记录删除。当事件处理模块 22 接收到客户端 1 的访问请求时，可以通过使用节点号查询到对应的记录，获取该对象在服务器端 2 数据库中的路径名，然后执行相应的操作。

[0039] 事件处理模块 22，该模块 22 用于处理客户端 1 的访问请求。在接收到客户端 1 的请求后，会依据请求中的节点号经节点号管理模块 23 定位到请求对象在服务器端 2 数据库中的位置，然后根据请求的类型做出相应的处理。

[0040] 数据传输模块 11，该模块 11 存在于服务器端 2 和客户端 1，主要用于客户端 1 与服务器端 2 的通讯（数据和协议），并支持 zlib 等压缩算法进行协议数据包的压缩，以提高广域网络上的传输效率。

[0041] 事件驱动模块 11，该模块 11 用于支撑相关的 POSIX 接口（便携式的操作系统接口）调用，能够接收文件系统创建、删除、读、写等系统调用请求。

[0042] 缓存管理模块 13，该模块 13 用于通过缓存更新策略管理元信息及数据。具体如下：

[0043] 将系统正在引用的对象的元信息缓存在内存中，并且在内存中维护元信息的引用计数，当引用计数变为 0 时，就会将对象的元信息从内存中替换出去，并持久化到数据库中，这样做不仅满足当前元信息的存储可靠性和查询、检索能力，还可在未来对容量扩展性等方面铺下良好的基础。而对于数据信息，支持以区间（extent）为粒度的缓存方式，其区间信息作为元信息的一部分进行缓存，而区间信息所对应的实际数据内容则缓存在本地数据库中。同时对缓存有效性的判断采取简单的超时策略，在访问元信息时，缓存中的元信息一旦超过了设置的超时时间限制，则认为元信息失效，需要与服务器端通信来更新元信息。为了提高系统资源利用率，该模块 13 还提供了对缓存空间的阈值管理，在系统初始化时设定缓存空间的上限值和下限值并设计了异步清理 cleaner 线程，在系统运行过程中，一旦缓存容量达到上限值，cleaner 线程会被唤醒并将部分数据缓存清除，以保证缓存空间在设定的阈值内。

[0044] 缓存更新策略还包括：

[0045] 对元信息有效性的判定采取的超时策略。在系统初始时设置元信息的超时时间，

在每次从远程获得元数据信息时，记录当时的系统时间。当访问对象元信息时，通过系统当前时间与从远程获得元数据时记录的系统时间进行比较，如果两者的差值小于设定的超时时间，则元信息有效，可直接从缓存中获取元信息；否则认为元信息失效，此时访问元信息就需要从远程获得。

[0046] 对数据有效性的判定采取如下策略：如果其元信息有效，则认为其缓存中的数据有效；如果其元信息失效，待重新从远程获得元数据后，通过比较缓存中元信息的修改时间与新获得元数据中的修改时间是否相同进行判定，如果相同，则认为缓存中数据有效；否则认为缓存中的数据已经失效。

[0047] 以下为本发明的基本处理流程：

[0048] 如图 2 所示，应用请求到达虚拟文件系统层，虚拟文件系统层会继续将请求经过内核模块发送给用户态库，然后用户态库触发客户端 1 的事件驱动模块 12，之后是本发明内部具体的协议处理流程，经过缓存管理模块 13、数据传输模块 11、事件处理模块 22、节点号管理模块 23 的处理后，处理结果返回给用户态库，然后经内核模块将请求结果返回给虚拟文件系统层，最后虚拟文件系统层将请求结果返回给应用请求。

[0049] 下面将会依次介绍本发明中典型协议（读取目录 / 创建 / 删除 / 获取属性 / 读取 / 写入）的处理流程。

[0050] 读取目录协议过程：读取对应对象的请求数据区间内的目录数据。虚拟文件系统层接收到读取目录请求后，会将请求转化为 <ino，offset，size>（其中 ino：读目录对象的节点号 ino；offset：读目录请求的偏移位置；size：读目录请求的读取字节数）这样的一个三元组，并传递给客户端的事件驱动模块。[读目录请求的偏移位置，读目录请求的读取字节数] 为读取目录请求的区间信息。

[0051] 如图 3 所示，读取目录协议过程具体流程如下：

[0052] 客户端 1 流程：步骤 101 事件驱动模块 12 接收到来自用户态库的读取目录请求。通过步骤 102 缓存管理模块 13 接收到事件驱动模块 12 传递过来的读取目录请求。步骤 103 缓存管理模块 13 根据读取目录请求的读目录对象的节点号查询对应对象的元信息，并判定其元信息中对应请求的区间缓存信息是否存在，如果存在，执行步骤 104 直接从本地数据库缓存中读取对应区间的目录数据，否则读目录请求需要与服务器端通信并获得对应区间的目录数据，步骤 105 事件驱动模块 12 接收缓存管理模块 13 的处理结果，如果已经读取到对应的区间的目录数据，则通过步骤 120 事件驱动模块 12 将对应的区间的目录数据返回给用户态库，读取目录流程结束；否则执行步骤 106 事件驱动模块 12 将请求传递给数据传输模块 11，数据传输模块 11 将读目录请求发送给服务器端 2。

[0053] 服务器端 2 流程：步骤 107 服务器端 2 的数据传输模块 11 接收到客户端 1 发送读目录请求，解析得到对应的 < 读目录对象的节点号、读目录请求的偏移位置、读目录请求的读取字节数 > 三元组，步骤 108 数据传输模块 11 将读取目录请求传递给事件处理模块 22，通过步骤 109 事件处理模块 22 将读目录请求传递给节点号管理模块 23，步骤 110 节点号管理模块 23 根据读目录请求的节点号查询数据库，得到对应记录，并获得其在数据库中的路径名，步骤 111 是将对应的路径名信息返回给节点号管理模块 23，执行步骤 112 节点号管理模块 23 将路径名信息传递给事件处理模块 22，步骤 113 根据路径名访问底层局域文件系统对应的目录，根据区间信息读出对应的目录数据，并获得对应各个目录项的属性信息。此

处一并获得各个目录项的属性信息的原因是为了加速后续的获取属性请求，在超时时间内避免客户端 1 与服务器端 2 的多次交互。执行步骤 114 将获得的信息返回给事件处理模块 22，步骤 115 事件处理模块 22 将获得的信息传递给数据传输模块 11，并由数据传输模块 11 将获得的信息发送给客户端 1。

[0054] 客户端 1 流程：步骤 116 数据传输模块 11 接收到服务器端 2 返回的信息，步骤 117 事件驱动模块 12 接收数据传输模块 11 的信息，执行步骤 118 事件驱动模块 12 将接收的信息传递给缓存管理模块 13，步骤 119 缓存管理模块 13 根据接收到的信息将读目录请求的区间内的目录数据信息、区间信息以及每个目录项对应的属性信息缓存到本地数据库中，之后执行步骤 120 事件驱动模块 12 向用户态库返回请求区间信息内的目录数据内容，读目录流程结束。

[0055] 创建文件协议过程：虚拟文件系统层接收创建文件请求后，会将请求转化为 <pino，name，mode>（其中 pino：创建对象父目录的节点号 ino；name：要创建对象的名字；mode：创建模式）这样的一个三元组，并传递给客户端 1 的事件驱动模块 12。

[0056] 如图 4 所示，创建文件协议过程具体流程如下：

[0057] 客户端 1 流程：步骤 201 客户端 1 的事件驱动模块 12 接收到来自用户态库的创建请求，步骤 202 事件驱动模块 12 将对应的请求传递给数据传输模块 11。数据传输模块 11 将创建请求打包成发送到服务器端 2。

[0058] 服务器端 2 流程：步骤 203 服务器端 2 的数据传输模块 11 接收到客户端 1 发送的创建文件请求，解析该请求得到对应创建请求，步骤 204 服务器端 2 将创建请求传递给事件处理模块 22，事件处理模块 22 根据创建请求中的要创建文件父目录的节点号查询数据库获得其对应的路径名，定位到要创建文件的父目录在底层局域文件系统中的位置，步骤 205 在父目录下执行创建操作，根据要创建文件的名称创建文件，并获得该新创建文件的属性信息。步骤 206 将创建成功的信息返回给事件处理模块 22，步骤 207 事件处理模块 22 将创建成功的信息传递给节点号管理模块 23，执行步骤 208 节点号管理模块 23 为新创建文件分配其唯一标识节点号，并生成对创建文件的一条记录，将该记录插入到数据库中，执行步骤 209 将插入记录成功的信息返回给节点号管理模块 23，执行步骤 210 节点号管理模块 23 将新创建文件的节点号发送给事件处理模块 22，步骤 211 事件处理模块 22 将创建文件的节点号及属性信息传递给数据传输模块 11，并由数据传输模块 11 将信息打包发送给客户端 1。

[0059] 客户端流程：步骤 212 客户端 1 的数据传输模块 11 接收到服务器端 2 发送的打包的信息，执行步骤 213 数据传输模块 11 将打包的信息发送给事件驱动模块 12，执行步骤 214 事件驱动模块 12 将打包的信息发送给缓存管理模块 13，执行步骤 215 缓存管理模块根据打包的信息，将新创建对象的元信息进行缓存，执行步骤 216 将缓存成功的信息返回给缓存管理模块 13，执行步骤 217 通过缓存管理模块 13 将缓存成功的信息返回给事件驱动模块 12，执行步骤 218 通过事件驱动模块 12 向用户态库返回创建成功的信息，创建流程结束。

[0060] 删除协议过程：删除对应的对象。虚拟文件系统层接收删除请求后，会将请求转化为 <pino，name>（其中 pino：删除对象父目录的节点号 ino；name：要删除对象的名字）这样的一个二元组，并传递给客户端 1 的事件驱动模块 12。

[0061] 如图 5 所示，删除协议过程具体流程如下：

[0062] 客户端 1 流程 :步骤 301 客户端 1 的事件驱动模块 12 接收到来自用户态库的删除请求,执行步骤 302 事件驱动模块 12 将删除请求发送给缓存管理模块 13,执行步骤 303 缓存管理模块 13 根据删除请求定位到要删除的对象在本地缓存中的位置,并判断该对象是否正在被引用,如果是,则将该删除请求改为重命名请求(将该对象重命名为根目录下以“.”开头的隐藏文件),否则继续,执行步骤 304 将请求的类型返回给缓存管理模块,步骤 305 事件驱动模块 12 接收缓存管理模块 13 的处理结果,步骤 306 事件驱动模块 12 将对应的请求传递给数据传输模块 11,数据传输模块 11 将请求打包发送到服务器端 2。

[0063] 服务器端流程 :步骤 307 服务器端 2 的数据传输模块 11 接收到客户端 1 发送的请求,解析该请求得到对应信息,执行步骤 308 数据传输模块 11 将请求发送给事件处理模块 22,步骤 309 事件处理模块 22 将信息发送给节点号管理模块 23,执行步骤 310 节点号管理模块 23 根据请求的节点号查询数据库,得到对应记录获取父目录对应的路径名,并与要删除对象的名称拼接,获得操作对象在数据库中的路径名,执行步骤 311 将对应的路径名信息返回给节点号管理模块 23,步骤 312 节点号管理模块 23 将对应的操作对象的位置信息传递给事件处理模块 22,步骤 313 事件处理模块 23 根据请求的类型执行相应的操作,操作成功后,将请求对象在数据库中的记录修改成删除状态,执行步骤 314 将操作成功的信息返回给事件处理模块 22,执行步骤 315 事件处理模块 22 将操作成功的信息传递给数据传输模块 11。并由数据传输模块 11 将信息打包发送给客户端 1。

[0064] 客户端 1 流程 :步骤 316 客户端 1 的数据传输模块 11 接收到服务器端 2 发送的打包的信息,步骤 317 数据传输模块 11 将该信息发送给事件驱动模块 12,步骤 318 事件驱动模块 12 将该信息发送给缓存管理模块 13,执行步骤 319 通过缓存管理模块 13 分析该信息,如果是重命名操作的返回,则在缓存管理模块 13 中将待删除对象设置为隐藏状态,表明该对象以后不可被访问,当该对象的引用计数减为 0 后,该对象才被真正的删除,否则将删除对象的缓存信息直接删除,步骤 320 事件驱动模块 12 向用户态库返回操作成功的信息,删除流程结束。

[0065] 获取属性协议过程 :获取对象的属性信息。过程描述如下 :

[0066] 客户端 1 处理流程 :客户端 1 接收获取属性请求后,判定对象的属性信息在是否在本地缓存中,若存在,则判定其缓存的属性信息是否有效,判断的原则是 :上次属性信息从服务器端 2 获得后记录的系统时间与当前的系统时间之间的差值是否小于系统初始时设定的超时时间,若小于,则说明缓存的属性信息有效,可直接使用缓存中的属性信息向上层的获取属性请求返回,至此,获取属性请求处理结束,否则,说明缓存的属性信息已经失效,此时需要重新从服务器端 2 获得属性信息,若不存在,说明之前没有发生过对该对象的获取属性请求,此时需要从服务器端 2 获得属性信息。

[0067] 以下为从服务器端 2 获取对象的属性信息 :客户端 1 将获取属性请求发送给服务器端 2。

[0068] 服务器端 2 处理流程 :服务器端 2 接收到客户端 1 发过来的获取属性请求后,在其导出目录所在的底层局域文件系统中定位到请求对象,并获得其属性信息,并将属性信息发送给客户端 1。

[0069] 客户端 1 处理流程 :客户端 1 接收到请求的返回后,如果是第一次对该对象执行获取属性请求,则将属性信息进行缓存,并记录当前的系统时间,用于对属性信息是否有效的

判断，并向上层的获取属性请求返回，至此，获取属性请求过程结束。如果缓存中存有已经失效的属性信息，首先判断新获得的属性信息与失效的属性信息中的修改时间是否相同，若相同，则说明该对象的数据缓存仍然有效，否则，说明该对象的数据缓存已经失效，则将其数据缓存清除。这样便保证了在获得属性之后，其缓存的信息与服务器端 2 的状态是一致的。然后将新获得的属性信息更新到缓存中，并记录当前的系统时间，最后向上层的获取属性请求返回，至此，获取属性请求过程结束。

[0070] 读取数据协议过程：读取对应对象在数据区间内的数据。虚拟文件系统层接收读取数据请求后，将请求转化为〈ino，offset，size〉（其中 ino：被读对象的节点号 ino；offset：被读数据的偏移位置；size：读取字节数）这样的一个三元组，并传递给客户端 1 的事件驱动模块 12，此处将［被读数据的偏移位置，读取字节数］定义为读请求的区间信息。

[0071] 如图 6 所示，读数据协议过程具体流程如下：

[0072] 客户端 1 流程：步骤 401 事件驱动模块 12 接收用户态库的读数据请求，步骤 402 缓存管理模块 13 接收事件驱动模块 12 发送的读数据请求，执行步骤 403 事件驱动模块 12 根据读数据请求的节点号查询读对象的元信息，并判定其元信息中对应读请求的区间缓存信息是否存在，如果存在，执行步骤 404 直接从本地数据库的数据缓存中读取对应的区间数据，否则读数据请求需要与服务器 2 端通信并获得对应区间数据，步骤 405 事件驱动模块 12 接收缓存管理模块 13 的处理结果，如果已经读取到对应的区间数据，则事件驱动模块 12 将对应的区间数据返回给用户态库，读流程结束，否则，执行步骤 406 事件驱动模块 12 将请求发送给数据传输模块 11。数据传输模块 11 将上述读数据请求打包发送给服务器端 2。

[0073] 服务器端 2 流程：步骤 407 服务器端 2 的数据传输模块 11 接收到客户端 1 发送读数据请求，解析得到对应信息，步骤 408 数据传输模块 11 将读数据请求发送给事件处理模块 22，步骤 409 事件处理模块 22 将读数据请求发送给节点号管理模块 23，执行步骤 410 节点号管理模块 23 根据读数据请求的节点号查询数据库，得到对应记录，并获得其在数据库的路径名，步骤 411 将对应的路径名信息返回给节点号管理模块 23，步骤 412 节点号管理模块 23 将该路径名信息发送给事件处理模块 22，步骤 413 根据该路径名访问底层局域文件系统中对应的文件，并根据读数据请求的区间信息读出对应的数据，步骤 414 将读到的区间数据返回给事件处理模块 22，执行步骤 415 事件处理模块 22 将读到的数据传递给数据传输模块 11，并由数据传输模块 11 打包发送给客户端 1。

[0074] 客户端 1 流程：步骤 416 数据传输模块 11 接收服务器端 2 返回的信息，步骤 417 事件驱动模块 12 接收数据传输模块 11 发送的信息，步骤 418 事件驱动模块 12 将接收的信息发送给缓存管理模块 13，执行步骤 419 缓存管理模块 13 根据该信息将读数据请求的数据内容以及区间信息更新到本地缓存中。在将数据信息写入到数据库时，会检测已使用的缓存空间是否已经达到系统设定上限值，如前文所述系统设计了异步清理线程，此线程一般处于睡眠状态，当检测到缓存空间超过了设定的上限值时，就会唤醒该线程，该线程会首先删除那些已经被替换到数据库中的对象所占用的数据缓存，如果此时剩余缓存空间达到设定的下限值，则该线程继续睡眠，否则就会继续根据对象的访问时间，删除那些驻留在内存中的对象对应的数据缓存，直到剩余缓存空间达到设定的下限值，然后线程转入睡眠状态。为支持缓存空间管理，在缓存空间不够时，将最久未被访问的对象缓存删除，需要记录文件的访问时间，为加快查找文件的速度，已经替换到数据库中的对象按照访问时间从小到大

排序，在内存中对象则会根据访问时间进行 LRU（近期最少使用算法）排序，删除时将访问时间小的对象的数据缓存删除。以上步骤执行完毕后执行步骤 420 事件驱动模块 12 向用户态库返回请求的数据内容，读流程结束。

[0075] 写入协议过程：将区间内的数据写入到对应的对象。虚拟文件系统层接收写入请求后，会将请求转化为 <ino，buf，offset，size>（其中 ino：被写入数据的对象的节点号 ino；buf：要写入数据所存放的地址；offset：该数据在被写入对象中的偏移位置；size：写入的字节数）这样的一个四元组，并传递给客户端 1 的事件驱动模块 12，其中［该数据在被写入对象中的偏移位置，写入的字节数］为写请求的区间信息。传统的 linux 写请求过程分两步，首先把写的内容写到内存中，其次再将写的内容同步到磁盘上。借鉴这样的思想，并使写请求的处理过程更适合于广域网络，本发明的写过程也分为两步，首先将数据写入到本地缓存中并记录写入数据所对应的区间信息，此时就可以向应用请求返回写操作已经结束，该过程称之为写入；其次只有当触发回刷操作才会将修改真正写到服务器端 2，这时一个写请求才真正的完成。

[0076] 如图 7 所示，写入协议过程具体流程如下：

[0077] 客户端 1 流程：步骤 501 事件驱动模块 12 接收用户态库的写请求，步骤 502 缓存管理模块 13 接收事件驱动模块 12 发送的写请求，执行步骤 503 缓存管理模块 13 根据写请求的节点号查询到写对象的元信息，并定位到其在底层局域文件系统的路径名，客户端 1 根据写请求在缓存管理模块 13 中定位到对应的写对象，将写请求的数据内容以及对应的区间信息写入对应的缓存中并标记该区间数据为“1”，并将对应的区间信息作为元信息的一部分进行缓存，步骤 504 将该写请求的处理结果返回给缓存管理模块，至此，广域文件系统的写流程结束。步骤 505 事件驱动模块 12 接收缓存管理模块 13 的处理结果，步骤 519 将写入成功的信息返回给用户态库，写入过程结束。

[0078] 下面将描述回刷过程：一旦回刷过程被触发，执行步骤 506 事件驱动模块 12 将请求传递给数据传输模块 11，数据传输模块 11 将上述写请求打包发送给服务器端 2。

[0079] 服务器端 2 流程：步骤 507 服务器端 2 的数据传输模块 11 接收到客户端 1 发送的写请求，解析得到对应信息，步骤 508 数据传输模块 11 将写请求发送给事件处理模块 22，执行步骤 509 事件处理模块 22 将写请求发送给节点号管理模块 23，执行步骤 510 节点管理模块 23 根据写请求的节点号查询数据库，得到对应记录，并获得其在数据库中的路径名，步骤 511 将对应的路径名信息返回给节点号管理模块 23，步骤 512 节点号管理模块 23 将该路径名信息发送给事件处理模块 22，执行步骤 513 根据该路径名访问底层局域文件系统对应的文件，并根据写请求的信息将对应的数据写入，步骤 514 将写入成功的信息返回给事件处理模块 22，执行步骤 515 事件处理模块 22 将信息发送给数据传输模块 11，并由数据传输模块打包发送给客户端 1。

[0080] 客户端 1 流程：步骤 516 数据传输模块 11 接收到服务器端 2 返回的信息，步骤 517 事件驱动模块 12 接收数据传输模块 11 的信息，步骤 518 事件驱动模块 12 将接收的信息发送给缓存管理模块 13，缓存管理模块 13 将对应写对象的区间信息标记为“2”，表示该区间数据与服务器端 2 的数据是一致的，执行步骤 519 事件驱动模块 12 向用户态库返回成功的信息，回刷流程结束。

[0081] 以下为本发明整体运行流程：

[0082] 一种广域文件系统，服务器端包括节点号管理模块、数据传输模块、事件处理模块；客户端包括事件驱动模块、数据传输模块、缓存管理模块，

[0083] 其中，通过该事件驱动模块接收操作请求，如果该请求为读取请求，则根据所请求广域文件的节点号和区间信息，通过该缓存管理模块查询本地缓存中是否存在该广域文件，若存在，则从该本地缓存中读取该广域文件，若不存在，则根据该节点号通过该节点号管理模块查询该广域文件在该服务器端数据库中的路径名，并根据该区间信息通过该事件处理模块获得该广域文件的数据信息，将该数据信息发送到该缓存管理模块，该缓存管理模块根据缓存更新策略将该数据信息及与其相对应的区间信息存入该本地缓存，以方便读取，该服务器端与该客户端通过各自的数据传输模块进行信息传递。

[0084] 所述的广域文件系统，还包括该事件驱动模块接收创建广域文件请求，将新广域文件在该服务器端数据库中的父目录节点号和该新广域文件的名称发送给该节点号管理模块，该节点号管理模块根据该父目录节点号查询该服务器端数据库获取该父目录的路径名，根据该路径名定位该父目录在该服务器端底层局域文件系统中的位置，并在该位置下根据该名称创建该新广域文件并获取该新创建文件的元信息，同时该节点号管理模块为该新创建广域文件分配节点号，该事件处理模块将该分配的节点号和该元信息发送到该缓存管理模块存入该本地缓存。

[0085] 所述的广域文件系统，还包括该事件驱动模块接收向广域文件写入请求，根据该广域文件的节点号定位该广域文件在该本地缓存中的位置，将要写入数据与该写入请求的区间信息写入该本地缓存，并标记该本地缓存中该区间信息为“1”，以表示该要写入数据仅存在于该本地缓存中，与该服务器端的该广域文件中的数据有差别。

[0086] 所述的广域文件系统，还包括数据回刷，具体步骤为：当启动数据回刷后，根据该节点号定位该广域文件在服务器端底层局域文件系统中的位置，将该要写入数据写入该服务器端该广域文件，并标记该本地缓存中该区间信息为“2”，以表示该本地缓存中该要写入数据与该服务器端的该广域文件中的数据一致。

[0087] 所述的广域文件系统，还包括该事件驱动模块接收删除广域文件请求，将待删除广域文件在该服务器端数据库中的父目录节点号和该待删除广域文件的名称发送给该节点号管理模块，该节点管理模块根据该父目录节点号定位该父目录在服务器端局域文件系统中的位置，通过事件处理模块删除该位置下具有该名称的广域文件，同时通过该缓存管理模块查询该待删除广域文件是否被引用，如果是，则将该本地缓存中该待删除广域文件隐藏，当引用结束后删除该本地缓存中的该待删除广域文件，如果否，则删除该本地缓存中该待删除广域文件。

[0088] 所述的广域文件系统，该缓存更新策略包括：记录该本地缓存中广域文件元信息的引用计数，当该引用计数变为 0 时，将该元信息从该本地缓存中替换出去，并存入数据库。

[0089] 所述的广域文件系统，该缓存更新策略还包括：记录该客户端从该服务器端获取元信息时的系统时间，通过该系统时间与访问该本地缓存中该元信息时的系统时间进行比较，若差值小于超时时间，则该本地缓存中的该元信息有效且与该元信息相对应的数据有效，并从该本地缓存中获取该元信息与该数据，否则从该服务器端获取新元信息，比较该本地缓存中该元信息的修改时间与该新元信息的修改时间是否相同，如果相同，则该数据有

效，否则该数据失效。

[0090] 所述的广域文件系统，该缓存更新策略还包括：检测已使用的该本地缓存空间是否已经达到设定的上限值，当达到该上限值时，则唤醒异步清理线程，该线程删除元信息已经被替换到数据库中的广域文件所占用的数据缓存，如果剩余该本地缓存空间达到设定的下限值，则该线程进入睡眠状态，否则根据广域文件的访问时间，删除该本地缓存中最久未被访问广域文件的数据缓存，直到剩余缓存空间达到该下限值。

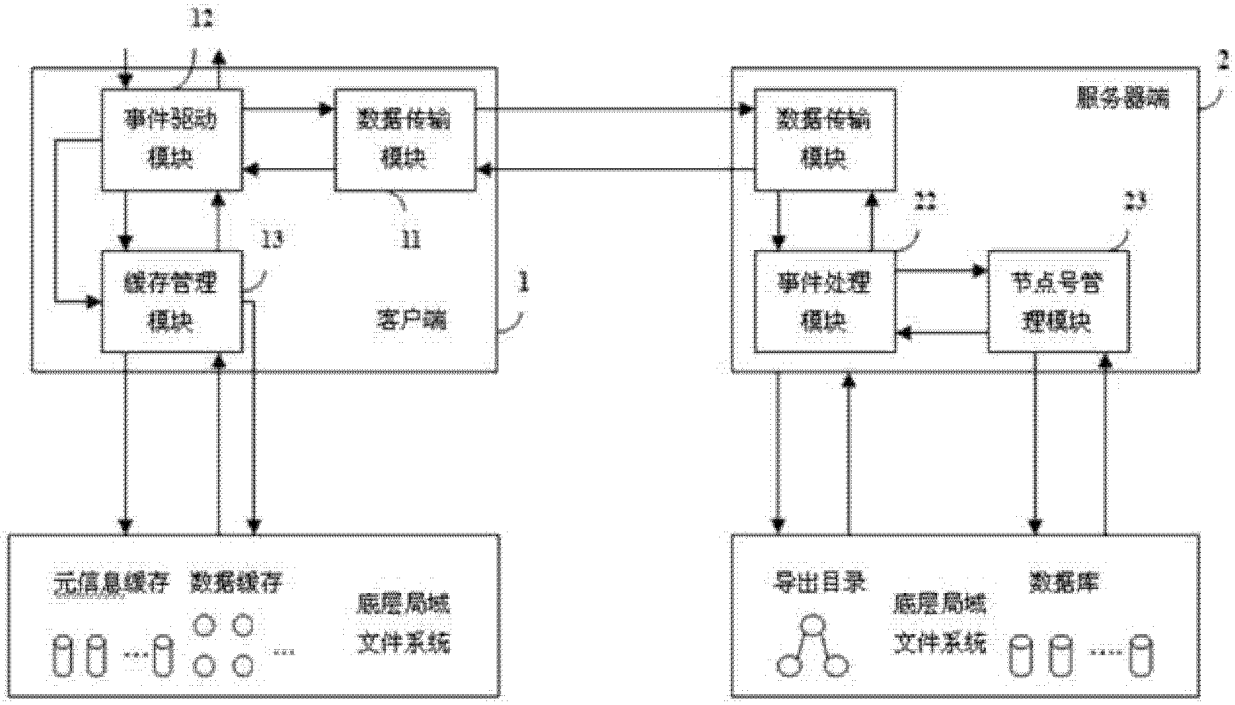

12
事件驱动模块
数据传输模块
数据传输模块
服务器端
2
22
23
13
11
缓存管理模块
客户端
1
事件处理模块
节点号管理模块
元信息缓存
数据缓存
底层局域文件系统
导出目录
底层局域文件系统
数据库

图 1

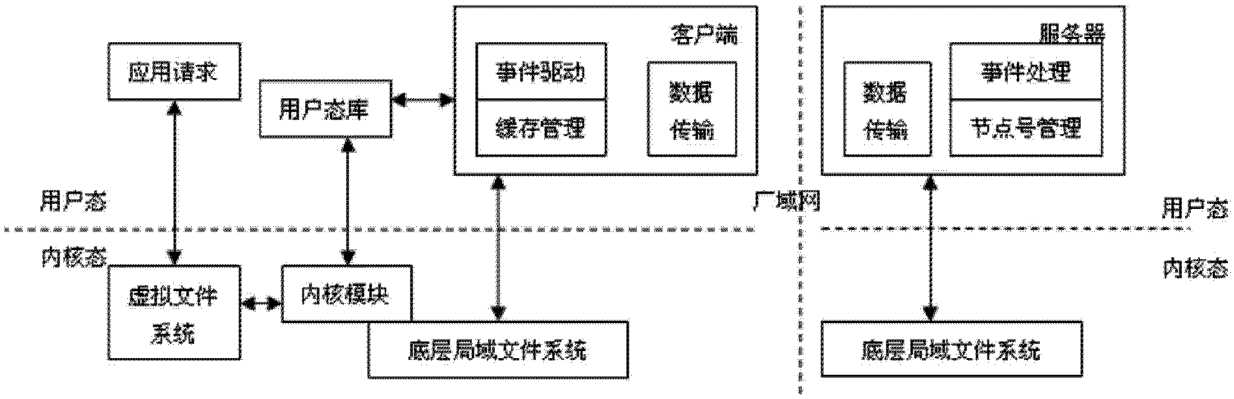

应用请求
用户态库
客户端
事件驱动
缓存管理
数据传输
服务器
数据传输
事件处理
节点号管理
用户态
广域网
用户态
内核态
内核态
虚拟文件系统
内核模块
底层局域文件系统
底层局域文件系统

图 2

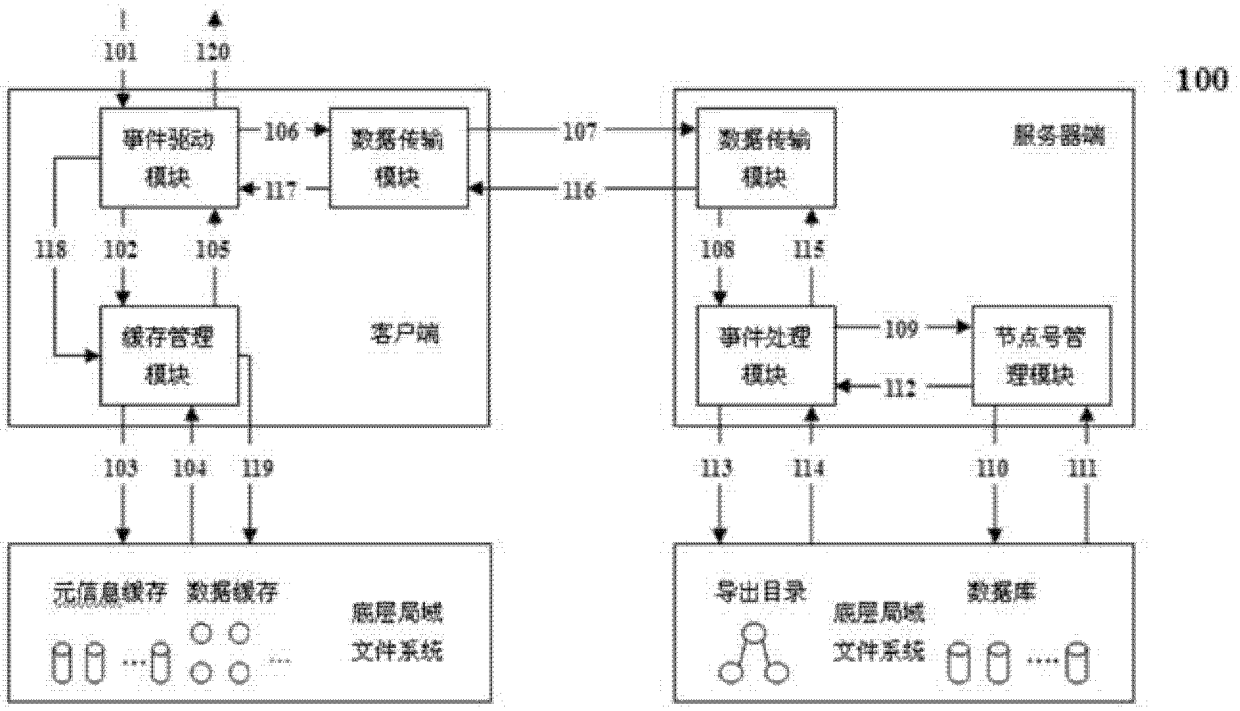

100
101
120
事件驱动模块
106
数据传输模块
117
107
116
数据传输模块
服务器端
118
102
105
108
115
缓存管理模块
客户端
事件处理模块
109
112
节点号管理模块
103
104
119
113
114
110
111
元信息缓存
数据缓存
底层局域文件系统
导出目录
底层局域文件系统
数据库

图 3

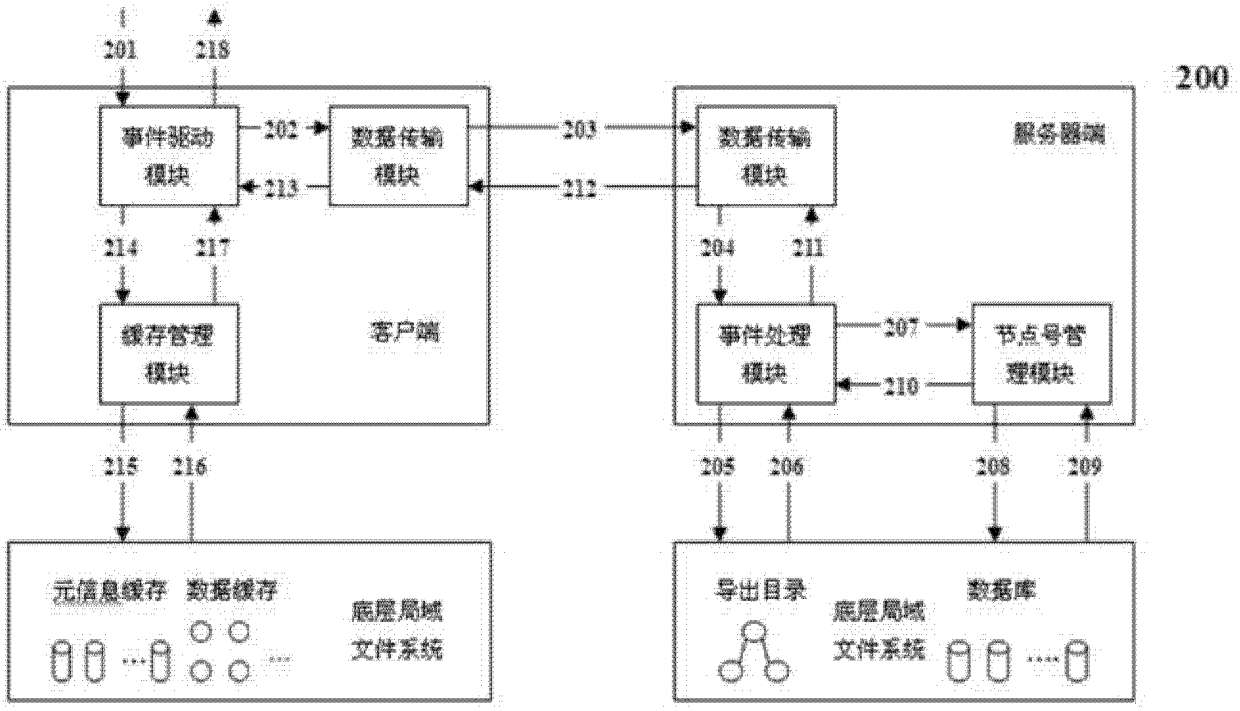

200
201
218
事件驱动模块
202
数据传输模块
213
203
212
数据传输模块
服务器端
214
217
204
211
缓存管理模块
客户端
事件处理模块
207
210
节点号管理模块
215
216
205
206
208
209
元信息缓存
数据缓存
底层局域文件系统
导出目录
底层局域文件系统
数据库

图 4

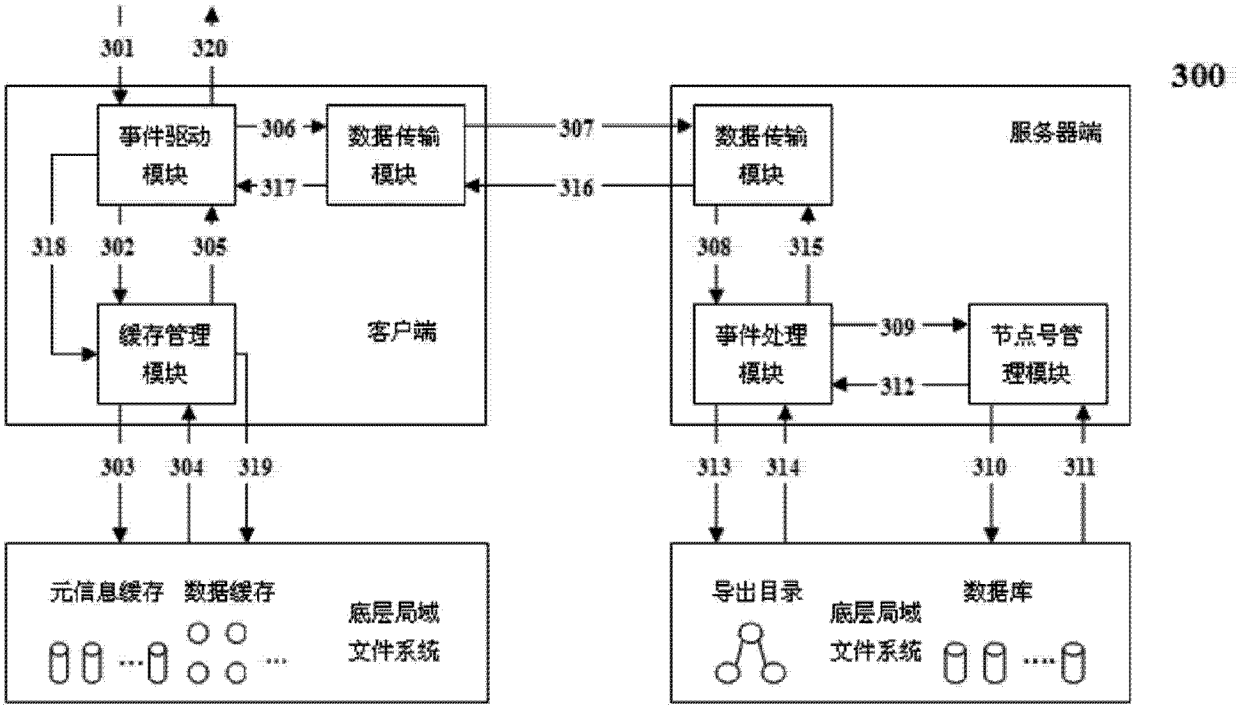

300
301
320
事件驱动模块
306
数据传输模块
307
317
316
数据传输模块
服务器端
318
302
305
308
315
缓存管理模块
客户端
事件处理模块
309
312
节点号管理模块
303
304
319
313
314
310
311
元信息缓存
数据缓存
底层局域文件系统
导出目录
底层局域文件系统
数据库

图 5

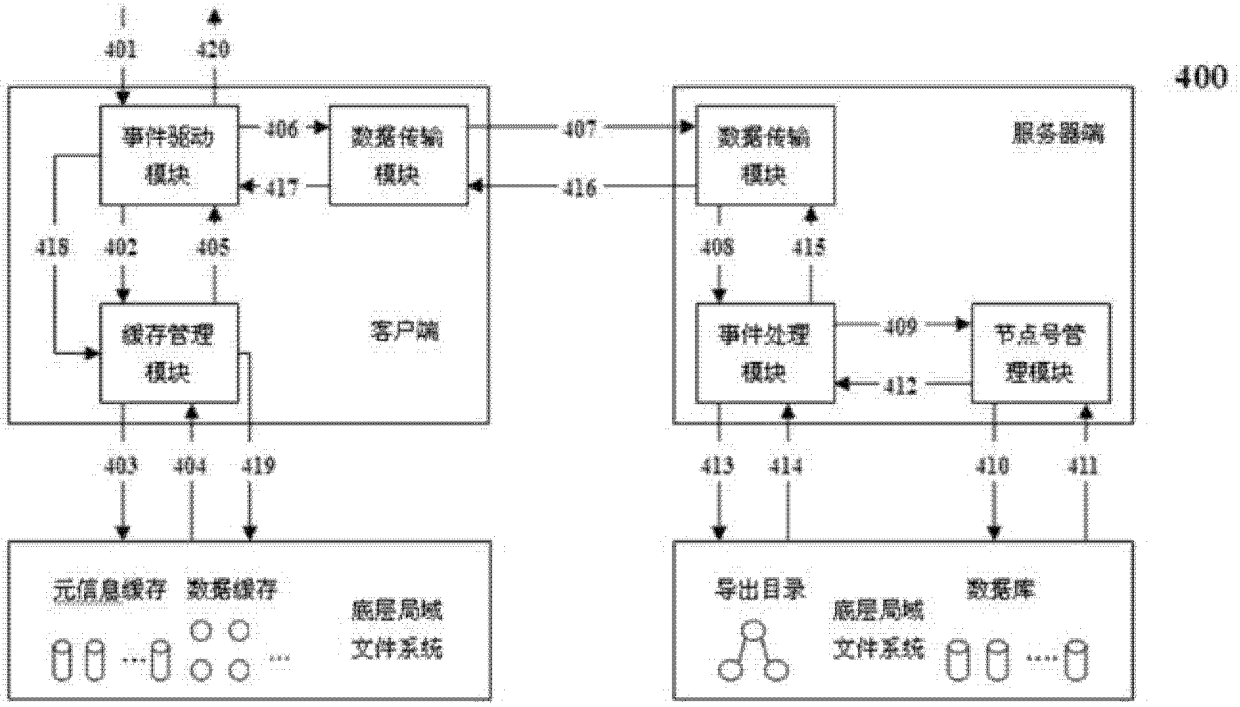

400
401
420
事件驱动模块
406
数据传输模块
407
417
416
数据传输模块
服务器端
418
402
405
408
415
缓存管理模块
客户端
事件处理模块
409
412
节点号管理模块
403
404
419
413
414
410
411
元信息缓存
数据缓存
底层局域文件系统
导出目录
底层局域文件系统
数据库

图 6

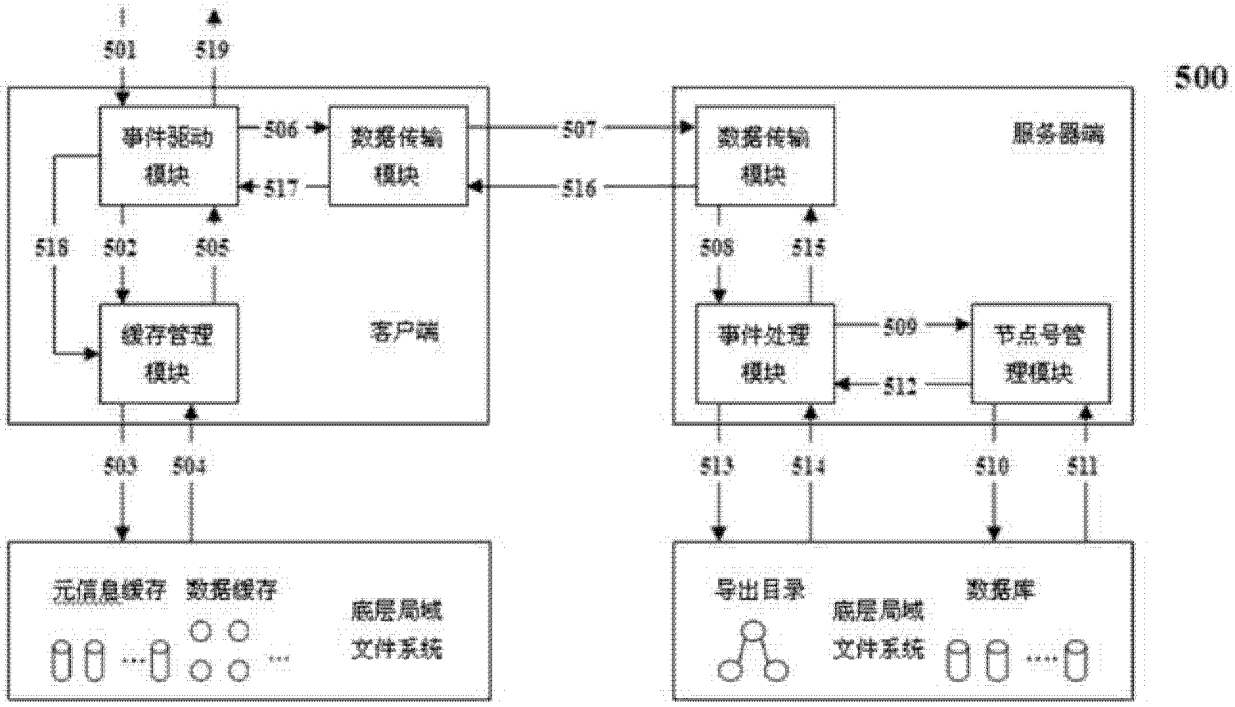

500
501
519
事件驱动模块
506
517
数据传输模块
507
516
数据传输模块
服务器端
518
502
505
508
515
缓存管理模块
客户端
事件处理模块
509
512
节点号管理模块
503
504
513
514
510
511
元信息缓存
数据缓存
底层局域文件系统
导出目录
底层局域文件系统
数据库

图 7