(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 106648442 A
(43)申请公布日 2017.05.10

(21)申请号 201510718506.0

(22)申请日 2015.10.29

(71)申请人 阿里巴巴集团控股有限公司
地址 英属开曼群岛大开曼资本大厦一座四层847号邮箱

(72)发明人 刘俊峰 姚文辉 张海勇 朱家稷

(74)专利代理机构 北京国昊天诚知识产权代理有限公司 11315
代理人 许志勇 刘戈

(51)Int.Cl.
G06F 3/06(2006.01)

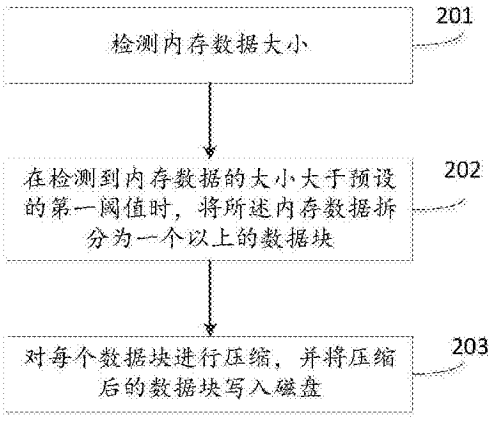
权利要求书2页 说明书8页 附图3页

(54)发明名称

一种元数据节点的内存镜像方法、装置

(57)摘要

本申请公开了一种元数据节点的内存镜像方法、装置，其中，所述方法包括在检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块，每个数据块的大小不超过所述第一阈值；对每个数据块进行压缩，并将压缩后的数据块写入磁盘。可以解决现有的dump内存镜像时磁盘空间压力较大问题。



CN 106648442 A

1.一种元数据节点的内存镜像方法，其特征在于，包括：

在检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块，每个数据块的大小不超过所述第一阈值；

对每个数据块进行压缩，并将压缩后的数据块写入磁盘。

2.根据权利要求1所述的方法，其特征在于，还包括：

在检测到内存数据的大小等于预设的第一阈值时，对所述内存数据进行压缩，并将压缩后的内存数据写入磁盘。

3.根据权利要求1所述的方法，其特征在于，还包括：

在检测到内存数据的大小小于预设的第二阈值时，将小于所述第二阈值的内存数据写入磁盘。

4.根据权利要求1所述的方法，其特征在于，在检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块之后，若存在小于所述第二阈值的数据块时，则所述方法还包括：

将小于所述第二阈值的数据块写入磁盘。

5.根据权利要求3或4所述的方法，其特征在于，将小于所述第二阈值的内存数据或数据块写入磁盘，包括：

将小于所述第二阈值的内存数据或数据块连续写入磁盘，且记录每个小于所述第二阈值的内存数据或数据块的起始位置和数据长度，即小于所述第二阈值的内存数据或数据块之间不存在大于等于所述第二阈值的数据块或内存数据。

6.根据权利要求1或2所述的方法，其特征在于，对每个数据块或内存数据进行压缩，包括：

采用多个线程对每个需要压缩的数据块或内存数据进行压缩。

7.根据权利要求6所述的方法，其特征在于，采用多个线程对每个需要压缩的数据块或内存数据进行压缩，包括：

在对每个需要压缩的数据块或内存数据进行压缩，生成每个需要压缩的数据块或内存数据对应的校验值。

8.一种元数据节点的内存镜像装置，其特征在于，包括：

数据拆分模块，用于在检测模块检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块，每个数据块的大小不超过所述第一阈值；

压缩模块，用于对所述数据拆分模块拆分后的每个数据块进行压缩，并将压缩后的数据块通过写入模块写入磁盘。

9.根据权利要求8所述的装置，其特征在于：

所述压缩模块，还用于在所述检测模块检测到内存数据的大小等于预设的第一阈值时，对所述内存数据进行压缩，并将压缩后的内存数据通过所述写入模块写入磁盘。

10.根据权利要求8所述的装置，其特征在于，还包括：

所述写入模块，还用于所述检测模块在检测到内存数据的大小小于预设的第二阈值时，将小于所述第二阈值的内存数据写入磁盘。

11.根据权利要求8所述的装置，其特征在于：

所述写入模块，还用于在所述检测模块检测到内存数据的大小大于预设的第一阈值

时，将所述内存数据拆分为一个以上的数据块之后，若存在小于所述第二阈值的数据块时，将小于所述第二阈值的数据块写入磁盘。

12. 根据权利要求 10 或 11 所述的装置，其特征在于：

所述写入模块，具体用于将小于所述第二阈值的内存数据或数据块连续写入磁盘，且记录每个小于所述第二阈值的内存数据或数据块的起始位置和数据长度，即小于所述第二阈值的内存数据或数据块之间不存在大于等于所述第二阈值的数据块或内存数据。

13. 根据权利要求 8 或 10 所述的装置，其特征在于：

所述压缩模块，具体用于采用多个线程对每个需要压缩的数据块或内存数据进行压缩。

14. 根据权利要求 13 所述的装置，其特征在于：

所述压缩模块，具体用于在对每个需要压缩的数据块或内存数据进行压缩，生成每个需要压缩的数据块或内存数据对应的校验值。

15. 一种元数据节点，其特征在于，包括：

如权利要求 8-14 中任一项所述的内存镜像装置。

一种元数据节点的内存镜像方法、装置

技术领域

[0001] 本申请涉及分布式存储系统，具体地说，涉及一种元数据节点的内存镜像方法、装置。

背景技术

[0002] 在大规模分布式存储系统中，为了实现集中权限认证、配额控制，大部分采用了集中式元数据管理的方法，即将整个存储系统中所有数据的元数据集中存放在若干个元数据节点 (NameNode) 进行存储。这样的架构中，元数据节点的可用性直接关系到整个存储系统的可用性。当元数据节点出现升级或者是进程重启时，快速恢复元数据节点的内存数据成为主要的需求点，因此，在存储系统中会采用定期将元数据节点中的内存数据写到磁盘中（即 dump 内存镜像），并记录数据操作日志来做到尽快恢复元数据节点中的内存数据，也就是说将磁盘中的保存的内存镜像读出来重新加载到内存中，其中，数据操作日志是对真正存储的数据的修改（如增加或删除等）记录。例如在 Hadoop 分布式文件系统中，元数据节点每次收到数据节点（真正存储数据的节点）写数据操作成功之前，修改元数据节点的数据操作日志并且同步到数据节点的文件系统。

[0003] 发明人在实现本发明的过程中发现：随着元数据的指数级增长，现有的 dump 内存镜像时，磁盘成为内存镜像的瓶颈，大量的元数据节点在 dump 内存镜像时需要占用的磁盘空间大，给磁盘造成较大的空间压力问题。

发明内容

[0004] 有鉴于此，本申请提供一种元数据节点的内存镜像方法、装置，可以解决现有的 dump 内存镜像时磁盘空间压力较大问题。

[0005] 为了解决上述技术问题，本申请第一方面提供一种元数据节点的内存镜像方法，包括：

[0006] 在检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块，每个数据块的大小不超过所述第一阈值；

[0007] 对每个数据块进行压缩，并将压缩后的数据块写入磁盘。

[0008] 可选地，所述的方法还包括：

[0009] 在检测到内存数据的大小等于预设的第一阈值时，对所述内存数据进行压缩，并将压缩后的内存数据写入磁盘。

[0010] 可选地，所述的方法还包括：

[0011] 在检测到内存数据的大小小于预设的第二阈值时，将小于所述第二阈值的内存数据写入磁盘。

[0012] 可选地，在检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块之后，若存在小于所述第二阈值的数据块时，则所述方法还包括：

[0013] 将小于所述第二阈值的数据块写入磁盘。

[0014] 可选地，将小于所述第二阈值的内存数据或数据块写入磁盘，包括：

[0015] 将小于所述第二阈值的内存数据或数据块连续写入磁盘，且记录每个小于所述第二阈值的内存数据或数据块的起始位置和数据长度，即小于所述第二阈值的内存数据或数据块之间不存在大于等于所述第二阈值的数据块或内存数据。

[0016] 可选地，对每个数据块或内存数据进行压缩，包括：

[0017] 采用多个线程对每个需要压缩的数据块或内存数据进行压缩。

[0018] 可选地，采用多个线程对每个需要压缩的数据块或内存数据进行压缩，包括：

[0019] 在对每个需要压缩的数据块或内存数据进行压缩，生成每个需要压缩的数据块或内存数据对应的校验值。

[0020] 本发明还提供一种元数据节点的内存镜像装置，包括：

[0021] 数据拆分模块，用于在检测模块检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块，每个数据块的大小不超过所述第一阈值；

[0022] 压缩模块，用于对所述数据拆分模块拆分后的每个数据块进行压缩，并将压缩后的数据块通过写入模块写入磁盘。

[0023] 可选地，所述压缩模块，还用于在所述检测模块检测到内存数据的大小等于预设的第一阈值时，对所述内存数据进行压缩，并将压缩后的内存数据通过所述写入模块写入磁盘。

[0024] 可选地，所述写入模块，还用于所述检测模块在检测到内存数据的大小小于预设的第二阈值时，将小于所述第二阈值的内存数据写入磁盘。

[0025] 可选地，所述写入模块，还用于在所述检测模块检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块之后，若存在小于所述第二阈值的数据块时，将小于所述第二阈值的数据块写入磁盘。

[0026] 可选地，所述写入模块，具体用于将小于所述第二阈值的内存数据或数据块连续写入磁盘，且记录每个小于所述第二阈值的内存数据或数据块的起始位置和数据长度，即小于所述第二阈值的内存数据或数据块之间不存在大于等于所述第二阈值的数据块或内存数据。

[0027] 可选地，所述压缩模块，具体用于采用多个线程对每个需要压缩的数据块或内存数据进行压缩。

[0028] 可选地，所述压缩模块，具体用于在对每个需要压缩的数据块或内存数据进行压缩，生成每个需要压缩的数据块或内存数据对应的校验值。

[0029] 本发明还提供一种元数据节点，包括：上述的内存镜像装置。

[0030] 本发明实施例通过在 dump 内存镜像时，当内存数据较大时，对内存数据进行拆分为数据块，且对每个拆分后的数据块进行多线程的压缩，不仅可以加快 dump 内存镜像速度，而且可以提高磁盘的空间利用率。

附图说明

[0031] 此处所说明的附图用来提供对本申请的进一步理解，构成本申请的一部分，本申请的示意性实施例及其说明用于解释本申请，并不构成对本申请的不当限定。在附图中：

[0032] 图 1 为一种分布式存储系统的构架图；

[0033] 图 2 为本发明实施例提供的一种元数据节点的内存镜像方法的流程图；

[0034] 图 3 为本发明实施例提供的一种元数据节点的内存镜像方法的流程图；

[0035] 图 4 为本发明实施例提供的一种元数据节点的内存镜像方法的流程图；

[0036] 图 5 为本发明实施例提供的一种元数据节点的内存镜像装置的结构图。

具体实施方式

[0037] 以下将配合附图及实施例来详细说明本申请的实施方式，藉此对本申请如何应用技术手段来解决技术问题并达成技术功效的实现过程能充分理解并据以实施。

[0038] 在一个典型的配置中，计算设备包括一个或多个处理器 (CPU)、输入 / 输出接口、网络接口和内存。

[0039] 内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器 (RAM) 和 / 或非易失性内存等形式，如只读存储器 (ROM) 或闪存 (flash RAM)。内存是计算机可读介质的示例。

[0040] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括非暂存电脑可读媒体 (transitory media)，如调制的数据信号和载波。

[0041] 如在说明书及权利要求当中使用了某些词汇来指称特定组件。本领域技术人员应可理解，硬件制造商可能会用不同名词来称呼同一个组件。本说明书及权利要求并不以名称的差异来作为区分组件的方式，而是以组件在功能上的差异来作为区分的准则。如在通篇说明书及权利要求当中所提及的“包含”为一开放式用语，故应解释成“包含但不限定于”。“大致”是指在可接收的误差范围内，本领域技术人员能够在一定误差范围内解决所述技术问题，基本达到所述技术效果。此外，“耦接”一词在此包含任何直接及间接的电性耦接手段。因此，若文中描述一第一装置耦接于一第二装置，则代表所述第一装置可直接电性耦接于所述第二装置，或通过其他装置或耦接手段间接地电性耦接至所述第二装置。说明书后续描述为实施本申请的较佳实施方式，然所述描述乃以说明本申请的一般原则为目的，并非用以限定本申请的范围。本申请的保护范围当视所附权利要求所界定者为准。

[0042] 还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的商品或者系统不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种商品或者系统所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的商品或者系统中还存在另外的相同要素。

[0043] 图 1 为一种分布式存储系统的构架图，如图 1 所示，其中，datanodes 是真正存储数据的数据节点，本发明所述的数据节点不限于图 1 所示，可以有多个；namenodes 是存储

数据的元数据的节点（简称元数据节点），本发明的元数据节点不限于图 1 所示，可以包括多个。

[0044] 基于图 1 所示的系统架构图，例如，当终端想在数据节点中写数据时，数据节点首先向元数据节点发起写请求（如图 1 中的 DFSclient，一种请求消息），元数据节点接收到写请求之后，元数据节点的内存中保存有该写请求中包括的待写入数据的大小、存储位置、名称或标识等元数据信息，记录本次写数据的日志，之后向数据节点返回写请求的响应消息并携带本次写数据的日志，以使数据节点根据元数据节点返回的响应消息将待写入数据真正保存到数据节点中。

[0045] 又例如，终端向从数据节点中读数据时，数据节点首先向元数据节点发起读请求，元数据节点的内存接收到读请求之后，获取读请求中携带的待读取数据的名称或标识等信息，元数据节点查询元数据信息库，获取到与待读取数据的名称或标识匹配的元数据信息，将获取的元数据信息返回给数据节点，数据节点根据元数据节点返回的元数据信息，其中，元数据信息中有待读取数据的存储位置等信息，数据节点根据元数据信息可以将待读取数据读出展示给终端用户。

[0046] 因此，当数据节点每操作一次数据时，元数据节点都保存了该次操作数据的元数据信息以及操作日志，这样元数据节点中保存有大量的元数据信息。元数据节点的可用性直接关系到整个存储系统的可用性。当元数据节点出现升级或者是进程重启时，快速恢复元数据节点的内存数据成为主要的需求点。

[0047] 发明人在实现本发明的过程中发现：现有技术中，在 dump 内存镜像时，是直接将内存的数据拷贝到磁盘中，随着元数据的指数级增长，磁盘成为内存镜像的瓶颈，大量的元数据节点在 dump 内存镜像时需要占用的磁盘空间大，给磁盘造成较大的空间压力问题。

[0048] 本发明实施例采用的方案是：在 dump 内存镜像时，根据内存数据的大小采取不同的压缩策略的方法，当内存数据小于一定阈值时，不对内存数据压缩，当内存数据大于一定阈值时，对内存数据进行压缩。

[0049] 本发明人在实现本发明的过程中发现：

[0050] 对于小于 4k 的数据，如果每次都组成一个 512k 的缓存 (buffer) 进行压缩，并不能很多的减少数据大小，反而浪费了处理器的处理资源，增加了内存拷贝的时间，造成 dump 内存镜像的速度降低。因此，对于小于 4k 的数据，采用将这部分小于 4k 的数据放置到一个非压缩的 buffer 中，而这个 buffer 中存放的小于 4k 的数据一定是连续的，就是说相邻的小于 4k 的数据块间不能出现大于等于 4k 的数据。

[0051] 例如，一个 buffer 是 512k，那么 buffer 中的数据放置的过程可能是 1k，2k，3k，不可能是 1k，5k，2k，因为出现 5k 时，会换一个 buffer 来写，这个 buffer 就将 1k 的数据写下去。

[0052] 进一步地，Buffer 中会记录这段小于 4k 的数据的起始位置以及长度，通常，每一个 buffer 还有一个附属记录的变量，来记录这个数据的起始位置。

[0053] 对于大于等于 4k 的数据：首先会将大于等于 4k 的数据拆包，一个数据最大为 512k，在将拆包后的数据进行压缩时，会自动识别数据的大小，将小于 4k 数据的 buffer 解锁掉，即直接进行写磁盘，然后将这部分小于 4k 的数据存放到 buffer 中。对于大于等于 4k 的数据采用多个线程进行压缩，每个线程会不断的从需要压缩的队列中取出需要压缩的大

于等于 4k 的数据，然后对大于等于 4k 的数据进行压缩。之后，将压缩后的数据放到需要写的队列中，写线程会不断的从写队列中取出需要写的压缩后的数据，写到写数据的 buffer 中。

[0054] 需要注意的是，在写数据到磁盘时，为减少磁盘的输入和输出操作，写数据时，每 2M 数据写一次磁盘，这样就不会浪费磁盘的输入和输出资源，也减少了对磁盘的操作压力。

[0055] 因此，本发明的技术方案中对内存数据进行压缩后再写磁盘，可以缓解 dump 内存镜像时的磁盘空间压力，提高磁盘空间利用率；而且对内存数据采用多线程进行压缩，提高 dump 内存镜像的速度和效率。

[0056] 图 2 为本发明实施例提供的一种元数据节点的内存镜像方法的流程图，如图 2 所示，包括：

[0057] 201、检测内存数据大小；

[0058] 具体应用场景是，元数据节点每隔一定时间去检查内存数据（如 fsimage 文件，该文件其实是元数据信息的文件），在每个检查时间点（checkpoint）写下内存数据（如 fsimage 文件）。根据本发明的技术方案，在每个检查时间点（checkpoint）写下内存数据（如 fsimage 文件）之前，需要检测内存数据大小，且根据内存数据（如 fsimage 文件）的大小进行不同的压缩策略。

[0059] 202、在检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块；

[0060] 其中，每个数据块的大小不超过所述第一阈值，本发明实施例中，第一阈值例如可以设置为 512k；

[0061] 例如，当检测到 fsimage 文件的大小大于 512k 时，将 fsimage 文件拆分为多个子文件，每个子文件的大小不超过 512k。

[0062] 需要说明的是，为了能够识别子文件，每个拆分后的子文件携带有拆分前的母文件的标识。

[0063] 203、对每个数据块进行压缩，并将压缩后的数据块写入磁盘。

[0064] 具体地，为了减少写磁盘的空间压力，本发明实施例对每个数据块进行压缩，并将压缩后的数据块写入磁盘；

[0065] 进一步地，为了提高写磁盘的速度，节省写磁盘的时间，本发明实施例可以采用多个线程对每个需要压缩的数据块或内存数据进行压缩；

[0066] 进一步地，在对每个需要压缩的数据块进行压缩，生成每个需要压缩的数据块对应的校验值，这里的校验值可以根据每个需要压缩的数据块内容生成的一串字符，用这个来校验值对每个数据块内容做校验，从而可以保证数据正确性，防止错误数据的发生。

[0067] 在发明另一个可选的实施方式中，在检测到内存数据的大小等于预设的第一阈值时（例如，当检测到 fsimage 文件的大小刚好等于 512k 时），对该内存数据采用多线程进行压缩，并且生成该内存数据的校验值，并将压缩后的内存数据写入磁盘。

[0068] 本发明实施例通过在 dump 内存镜像时，当内存数据较大时，对内存数据进行拆分为数据块，且对每个拆分后的数据块进行多线程的压缩，不仅可以加快 dump 内存镜像速度，而且可以提高磁盘的空间利用率。

[0069] 图 3 为本发明实施例提供的一种元数据节点的内存镜像方法的流程图，如图 3 所

示，包括：

[0070] 301、检测内存数据大小；

[0071] 参考图 2 所示实施例的步骤 201；

[0072] 302、在检测到内存数据的大小小于预设的第二阈值时，将小于所述第二阈值的内存数据写入磁盘；

[0073] 其中，发明人在实现本发明的过程中发现：对于小于 4k 的数据，如果每次都组成一个 512k 的缓存进行压缩，并不能减少很多的内存数据的大小，反而浪费了因为压缩需要消耗的处理资源，增加了内存拷贝的时间，造成写 checkpoint 会比较慢，即导致 dump 内存镜像速度的降低，因此，本发明实施例中，将 4k 设置为第二阈值，小于 4k 的内存数据不进行压缩。

[0074] 进一步地，在如图 2 所示实施例的步骤 202 中，当将内存数据拆分为一个以上的数据块，其中存在小于第二阈值的数据块时，将小于第二阈值的数据块写入磁盘。

[0075] 进一步地，本发明实施例中，将小于第二阈值的内存数据或数据块连续写入磁盘，且记录每个小于所述第二阈值的内存数据或数据块的起始位置和数据长度，即小于所述第二阈值的内存数据或数据块之间不存在大于等于所述第二阈值的数据块或内存数据。具体实现时，例如，会将小于 4k 的数据放置到一个非压缩的缓存 (buffer) 中，而这个非压缩的 buffer 中存放的数据一定是连续的，就是说相邻的小于 4k 的数据块间不能出现大于等于 4k 的数据块，这个非压缩的 Buffer 中会记录这段小于 4k 的数据块的起始位置以及长度。

[0076] 本发明实施例通过在 dump 内存镜像时，当内存数据或数据块较小，即小于 4k 时，对小于 4k 的数据块不进行压缩，而是采用将小于 4k 的数据块连续存放非压缩的缓存中，不会消耗额外的处理资源，保证 dump 内存镜像速度，可以实现 dump 内存镜像速度和处理资源的平衡。

[0077] 下面通过具体应用对本发明实施例所述的方法进行说明：

[0078] 图 4 为本发明实施例提供的一种元数据节点的内存镜像方法的流程图，如图 4 所示，包括：

[0079] 401、检测内存数据大小；

[0080] 在内存数据的大小大于 512k 时执行步骤 402，在内存数据小于 4k 时执行步骤 405。

[0081] 402、将内存数据拆分为多个数据块；

[0082] 假设内存数据大小为 10M(即 10000k)，每个数据块的大小最大不超过 512k，这样，可以拆分为 19 个 512k 大小的数据块，1 个 272k 大小的数据块；

[0083] 假设内存数据大小为 1025k，按照每个数据块的大小最大不超过 512k，这样，可以拆分为 2 个 512k 大小的数据块，1 个 1k 的数据块；

[0084] 当数据块的大小超过 4k 时，执行步骤 403，当数据块的大小小于等于 4k 时，执行步骤 404；

[0085] 403、对于大于 4k 的数据块采用多线程进行压缩；

[0086] 例如，本发明实施例中设置一个 512k 的压缩缓存 (buffer)，即给压缩缓存打上需要压缩的标签，对于超过 4k 的数据块，将这些数据放入这个 512k 的压缩缓存，然后放入到压缩队列中等待压缩；为了加快压缩速度，采用多线程进行压缩。

[0087] 进一步地，在压缩每个数据块时，可以根据每个数据块内容产生该压缩数据块的校验值，用于保证该压缩数据块的正确性。

[0088] 404、对压缩后的数据块写入磁盘；

[0089] 将每个压缩后的数据块放入到写磁盘队列中，在写磁盘时，为减少磁盘的输入和输出操作，每2M数据写一次磁盘，也就是说，当压缩后的数据块到2M时才进行一次写磁盘，这样就不会浪费磁盘的输入和输出资源，也减少了对磁盘的操作压力。

[0090] 405、对小于4k的数据块不压缩写入磁盘。

[0091] 例如，对小于4k的数据块，将这些小于4k的数据块放入512k的非压缩缓存中，对于非压缩缓存需要解锁压缩标签，即不放入压缩队列中，而是直接放入写磁盘队列中；

[0092] 需要说明的是，在将小于4k的数据块放入非压缩的buffer中时，存放的小于4k的数据块在非压缩的buffer中一定是连续的，就是说相邻的小于4k的数据块间不能出现大于等于4k的数据块，这个非压缩的Buffer中会记录每一段小于4k的数据块的起始位置以及长度。

[0093] 在写磁盘时，为减少磁盘的输入和输出操作，每2M数据写一次磁盘，也就是说，当小于4k的数据块放入非压缩的buffer中达到2M时才进行一次写磁盘，这样就不会浪费磁盘的输入和输出资源，也减少了对磁盘的操作压力。

[0094] 如果没有数据可以压缩，那么线程会处于等待(condition wait)的状态，当有新的数据需要压缩时写线程会发信号给(signal)压缩线程。

[0095] 本发明的技术方案中对内存数据进行压缩后再写磁盘，可以缓解dump内存镜像时的磁盘空间压力，提高磁盘空间利用率；而且对内存数据采用多线程进行压缩，提高dump内存镜像的速度和效率。

[0096] 图5为本发明实施例提供的一种元数据节点的内存镜像装置的结构图，如图5所示，包括：

[0097] 检测模块51，用于检测到内存数据的大小

[0098] 数据拆分模块52，用于在检测模块检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块，每个数据块的大小不超过所述第一阈值；

[0099] 压缩模块53，用于对所述数据拆分模块拆分后的每个数据块进行压缩，并将压缩后的数据块通过写入模块54写入磁盘。

[0100] 可选地，所述压缩模块53，还用于在所述检测模块51检测到内存数据的大小等于预设的第一阈值时，对所述内存数据进行压缩，并将压缩后的内存数据通过所述写入模块54写入磁盘。

[0101] 可选地，所述写入模块54，还用于所述检测模块51在检测到内存数据的大小小于预设的第二阈值时，将小于所述第二阈值的内存数据写入磁盘。

[0102] 可选地，所述写入模块54，还用于在所述检测模块51检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块之后，若存在小于所述第二阈值的数据块时，将小于所述第二阈值的数据块写入磁盘。

[0103] 可选地，所述写入模块54，具体用于将小于所述第二阈值的内存数据或数据块连续写入磁盘，且记录每个小于所述第二阈值的内存数据或数据块的起始位置和数据长度，即小于所述第二阈值的内存数据或数据块之间不存在大于等于所述第二阈值的数据块或

内存数据。

[0104] 可选地，所述压缩模块 53，具体用于采用多个线程对每个需要压缩的数据块或内存数据进行压缩。

[0105] 可选地，所述压缩模块 53，具体用于在对每个需要压缩的数据块或内存数据进行压缩，生成每个需要压缩的数据块或内存数据对应的校验值。

[0106] 图 5 所示装置可以执行上述图 2- 图 3 任一实施例所述的方法，其实现原理和技术效果不再赘述。

[0107] 本发明实施例还提供一种元数据节点，包括图 3 所示实施例所述的内存镜像装置。

[0108] 上述说明示出并描述了本发明的若干优选实施例，但如前所述，应当理解本发明并非局限于本文所披露的形式，不应看作是对其他实施例的排除，而可用于各种其他组合、修改和环境，并能够在本文所述发明构想范围内，通过上述教导或相关领域的技术或知识进行改动。而本领域人员所进行的改动和变化不脱离本发明的精神和范围，则都应在本发明所附权利要求的保护范围内。

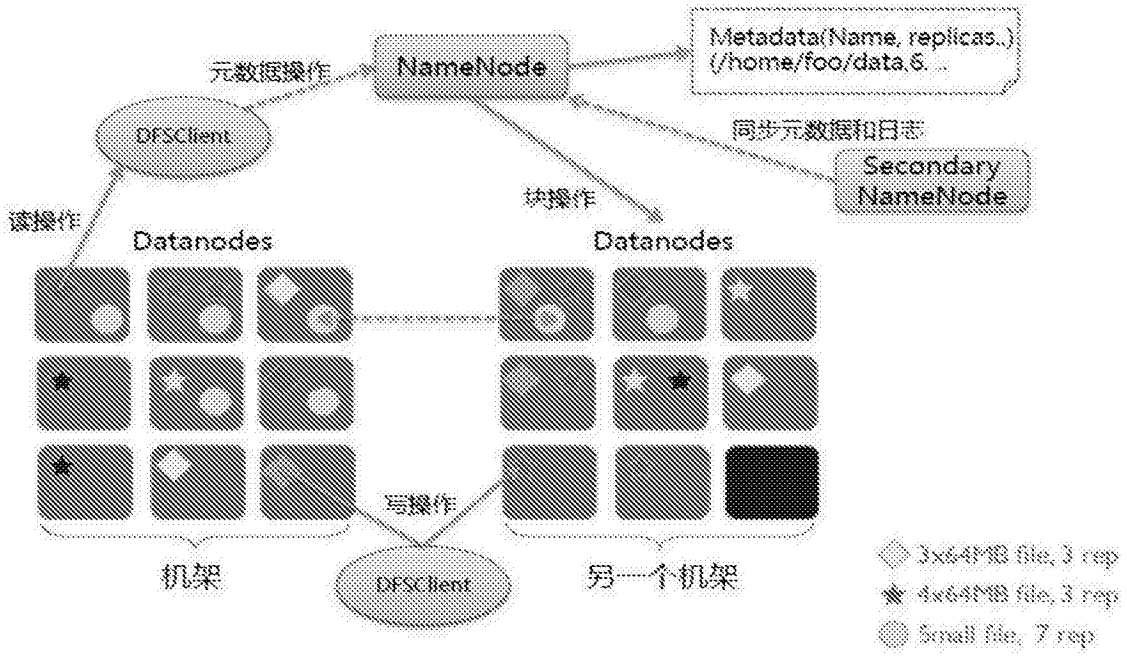

元数据操作
NameNode
Metadata(Name, replicas..)
(/home/foo/data,6. ..
DFSClient
同步元数据和日志
Secondary
NameNode
读操作
块操作
Datanodes
Datanodes
写操作
机架
DFSClient
另一个机架
3x64MB file, 3 rep
4x64MB file, 3 rep
Small file, 7 rep

图 1

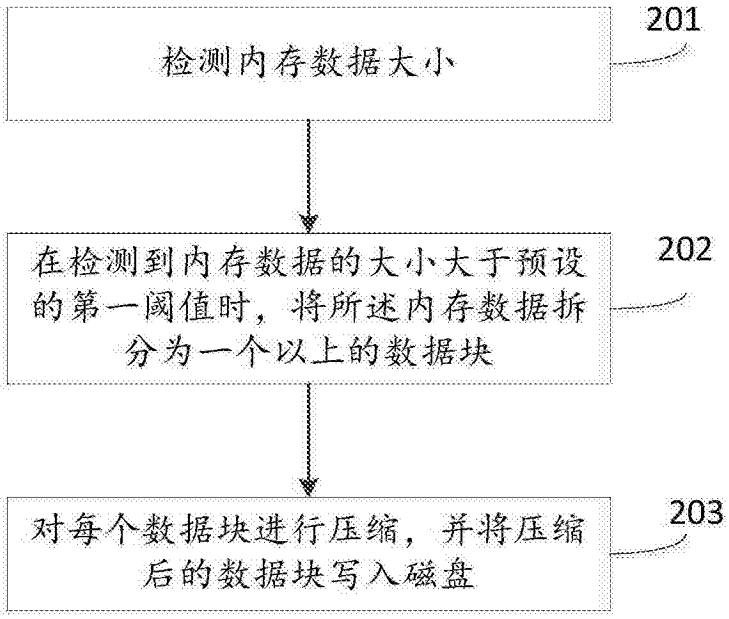

检测内存数据大小
201
在检测到内存数据的大小大于预设的第一阈值时，将所述内存数据拆分为一个以上的数据块
202
对每个数据块进行压缩，并将压缩后的数据块写入磁盘
203

图 2

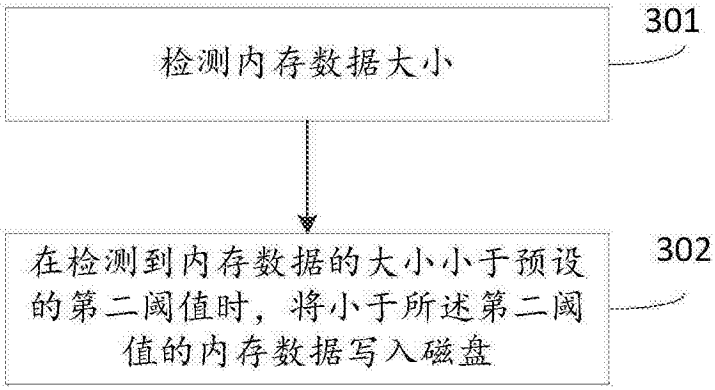

检测内存数据大小
301
在检测到内存数据的大小小于预设的第二阈值时，将小于所述第二阈值的内存数据写入磁盘
302

图 3

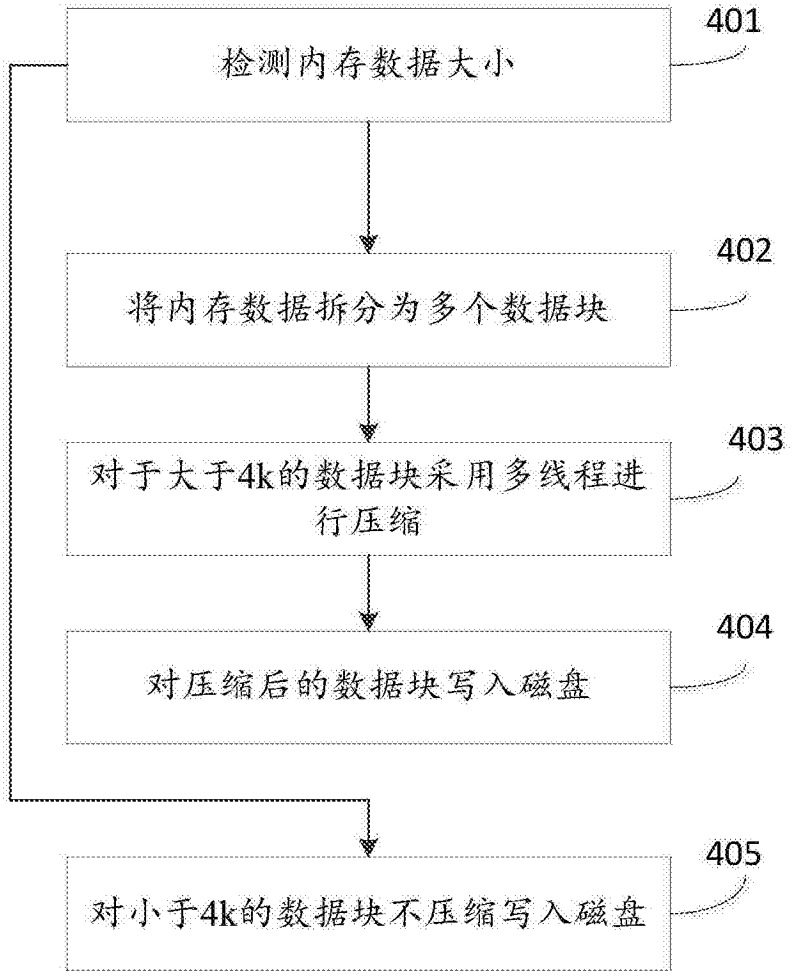

检测内存数据大小
401
将内存数据拆分为多个数据块
402
对于大于4k的数据块采用多线程进行压缩
403
对压缩后的数据块写入磁盘
404
对小于4k的数据块不压缩写入磁盘
405

图 4

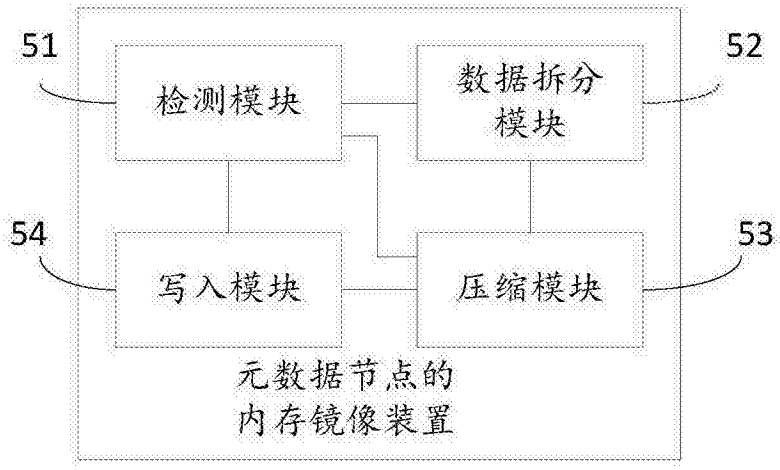


图 5