(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114546980 B
(45)授权公告日 2022.07.08

(21)申请号 202210436334.8
(22)申请日 2022.04.25
(65)同一申请的已公布的文献号
申请公布号 CN 114546980 A
(43)申请公布日 2022.05.27
(73)专利权人 成都云祺科技有限公司
地址 610041 四川省成都市高新区云华路333号8栋4-5层
(72)发明人 黄传波 谢俊峰 罗睿 谢卓伟 涂磊 钱禹航
(51)Int.Cl.
G06F 16/182(2019.01)
G06F 11/14(2006.01)
(56)对比文件
CN 110442644 A,2019.11.12
CN 105302527 A,2016.02.03
US 9229942 B1,2016.01.05
审查员 王黎

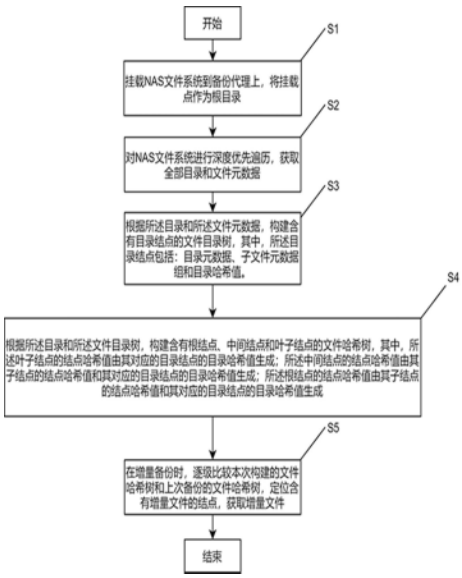
权利要求书3页 说明书10页 附图8页

(54)发明名称

一种NAS文件系统的备份方法、系统及存储介质

(57)摘要

本发明涉及一种NAS文件系统的备份方法、系统及存储介质，属于文件系统备份领域。所述方法包括：挂载步骤；目录及文件元数据获取步骤；文件目录树构建步骤；文件哈希树构建步骤；增量文件获取步骤。所述系统包括：挂载模块；目录及文件元数据获取模块；文件目录树构建模块；文件哈希树构建模块；增量文件获取模块。本发明通过对NAS文件系统进行深度优先遍历，构造文件目录树，并基于文件目录树构造文件哈希树，然后逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树，快速定位含有增量文件的结点，获取两次备份时间点之间的增量文件，实现了NAS文件系统的高效备份。



CN 114546980 B

1.一种NAS文件系统的备份方法,其特征在于,包括以下步骤:

挂载步骤,挂载NAS文件系统到备份代理上,将挂载点作为根目录;

目录及文件元数据获取步骤,对所述根目录进行深度优先遍历,获取全部目录和文件元数据;

文件目录树构建步骤,根据所述目录和所述文件元数据,构建含有目录结点的文件目录树,其中,所述目录结点包括:目录元数据、子文件元数据组和目录哈希值;

文件哈希树构建步骤,根据所述目录和所述文件目录树,构建含有根结点、中间结点和叶子结点的文件哈希树,其中,所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成;所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成;所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成;

其中,所述目录及文件元数据获取步骤、所述文件目录树构建步骤和所述文件哈希树构建步骤同时进行;

增量文件获取步骤,在增量备份时,从根结点开始,逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树,定位含有增量文件的结点,获取增量文件;

其中,所述目录及文件元数据获取步骤,还包括:

根目录的子目录获取步骤,获取根目录下的所有子目录;

线程池创建步骤,创建全局任务队列、工作线程和守护线程,三者并行处理数据;

全局任务获取步骤,将所述根目录的子目录作为遍历任务传入所述全局任务队列;

遍历任务执行步骤,所述工作线程从所述全局任务队列中取得遍历任务,并调用遍历任务函数对所述遍历任务进行处理;

守护线程控制步骤,所述守护线程控制所述工作线程的数量处于预定阈值。

2.根据权利要求1所述NAS文件系统的备份方法,其特征在于,所述文件目录树构建步骤,包括:

目录结点建立步骤,在对所述NAS文件系统进行深度优先遍历的同时,建立目录结点,令所述目录结点与当前目录对应,使所述目录结点包含:目录元数据、子文件元数据组和目录哈希值,其中,目录元数据为对应目录的元数据;子文件元数据组为对应目录的子文件元数据组;目录哈希值由所述目录元数据和所述子文件元数据组生成;

文件目录树建立步骤,重复所述目录结点建立步骤,直至建立文件目录树,其中,所述文件目录树的目录结点与全部目录一一对应。

3.根据权利要求2所述NAS文件系统的备份方法,其特征在于,所述文件哈希树构建步骤,包括:

第一判断步骤,在建立目录结点的同时,判断所述目录结点对应的目录是否包含子目录;

叶子结点建立步骤,若是,则重复所述第一判断步骤;若否,则建立文件哈希树的叶子结点,并使所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成;

第二判断步骤,判断所述叶子结点的所有兄弟结点是否创建完成;

中间结点建立步骤,若是,新建中间结点作为所述叶子结点的父结点,并使所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成;若否,

则重复第一判断步骤；

根结点建立步骤，重复第一判断步骤至中间结点建立步骤，直至创建所述文件哈希树的根结点，并使所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成。

4.根据权利要求1所述的NAS文件系统的备份方法，其特征在于，所述全局任务获取步骤还包括：

遍历任务CPU占用时间记录步骤，备份完成时，记录所述全局任务队列中遍历任务的CPU占用时间；

超时任务判断步骤，读取前次备份记录的全局任务队列中遍历任务的CPU占用时间，并检测是否存在超时任务；

遍历任务细粒度化步骤，若是，则对将所述超时任务进行细粒度化处理，再使所述全局任务队列的队列头指向所述超时任务的下一遍历任务；若否，则执行遍历任务执行步骤。

5.一种NAS文件系统的备份系统，其特征在于，所述系统包括：

挂载模块，用于挂载NAS文件系统到备份代理上，将挂载点作为根目录；

目录及文件元数据获取模块，用于深度优先遍历所述NAS文件系统，获取全部目录和文件元数据；

文件目录树构建模块，用于根据所述目录和所述文件元数据，构建含有目录结点的文件目录树，其中，所述目录结点包括：目录元数据、子文件元数据组和目录哈希值；

文件哈希树构建模块，用于根据所述目录和所述文件目录树，构建含有根结点、中间结点和叶子结点的文件哈希树，其中，所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成；所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；

其中，所述目录及文件元数据获取模块、所述文件目录树构建模块和所述文件哈希树构建模块同时进行；

增量文件获取模块，用于在增量备份时，从根结点开始，逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树，定位含有增量文件的结点，获取增量文件；

其中，所述目录及文件元数据获取模块，还包括：

根目录的子目录获取单元，用于获取根目录下的所有子目录；

线程池创建单元，用于创建全局任务队列、工作线程和守护线程，三者并行处理数据；

全局任务获取单元，用于将根目录的子目录作为遍历任务传入所述全局任务队列；

遍历任务执行单元，用于所述工作线程从所述全局任务队列中取得遍历任务，并调用遍历任务函数对所述遍历任务进行处理；

守护线程控制单元，用于控制所述工作线程的数量处于预定阈值。

6.根据权利要求5所述的NAS文件系统的备份系统，其特征在于，所述文件目录树构建模块，包括：

目录结点建立单元，用于在对所述NAS文件系统进行深度优先遍历的同时，建立目录结点，令所述目录结点与当前目录对应，使所述目录结点包含：目录元数据、子文件元数据组和目录哈希值，其中，目录元数据为对应目录的元数据；子文件元数据组为对应目录的子文

件元数据组;目录哈希值由所述目录元数据和所述子文件元数据组生成;

文件目录树建立单元,用于重复所述目录结点建立单元,直至建立文件目录树,其中,所述文件目录树的目录结点与全部目录一一对应。

7.根据权利要求5所述的NAS文件系统的备份系统,其特征在于,所述文件哈希树构建模块,还包括:

第一判断单元,用于在建立目录结点的同时,判断所述目录结点对应的目录是否包含子目录

叶子结点建立单元,用于若是,则重复所述第一判断单元;若否,则建立文件哈希树的叶子结点,并使所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成;

第二判断单元,用于判断所述叶子结点的所有兄弟结点是否创建完成;

中间结点建立单元,用于若是,新建中间结点作为所述叶子结点的父结点,并使所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成;若否,则重复第一判断单元;

根结点建立单元,用于重复第一判断单元至中间结点建立单元,直至创建所述文件哈希树的根结点,并使所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成。

8.一种计算机可读存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现如权利要求1至4任一项所述的NAS文件系统的 备份方法。

一种NAS文件系统的备份方法、系统及存储介质

技术领域

[0001] 本发明属于文件系统备份领域，涉及一种NAS文件系统的备份方法、系统及存储介质。

背景技术

[0002] 网络附属存储(Network Attached Storage, NAS)是一种通过网络为客户端提供文件共享服务的文件数据服务器。NAS系统部署简单、性价比高、兼容性强，能够为不同操作系统的客户端提供文件数据集中存储和共享服务，因此在政务、医疗、金融、物流等领域有着广泛的应用。为了防止数据损坏、丢失给个人或企业带来无法估量的影响，文件系统数据通常受备份保护。随着社交网络、电子商务等网络业务的飞速发展，各类应用程序的功能越来越复杂，其产生的数据量也呈指数型增长，甚至一个应用就能产生数百万个小文件，文件系统数据备份难度系数直线上升。

[0003] 传统的NAS文件系统备份通过使用网络数据管理协议(Network Data Management Protocol, NDMP)分离控制连接和数据连接，使得数据流量可以通过本地高速通道直接备份到备份存储设备中，避免了IP网络的流量瓶颈，但NDMP协议只支持单数据流会话，备份效率极低，并且NDMP的映像级备份使得在恢复单个文件时，也必须恢复整个文件系统，特别是在面对海量级备份任务时，工作量巨大，处理效率低。而目前，能够实现无需NDMP协议且快速查找增量文件的NAS备份方案几乎没有，这也影响了NAS文件系统备份技术的广泛运用。

[0004] 因此，如何提升NAS文件系统备份中文件系统的遍历速度，快速查找增量文件，成为当前急需解决的技术问题。

发明内容

[0005] 本发明为了解决上述背景技术中的技术问题，本发明实施例提供了一种NAS文件系统的备份方法、系统及存储介质。所述技术方案如下：

[0006] 第一个方面，提供了一种NAS文件系统的备份方法，所述方法包括步骤：

[0007] 挂载步骤，挂载NAS文件系统到备份代理上，将挂载点作为根目录；

[0008] 目录及文件元数据获取步骤，对所述NAS文件系统进行深度优先遍历，获取全部目录和文件元数据；

[0009] 文件目录树构建步骤，根据所述目录和所述文件元数据，构建含有目录结点的文件目录树，其中，所述目录结点包括：目录元数据、子文件元数据组和目录哈希值；

[0010] 文件哈希树构建步骤，根据所述目录和所述文件目录树，构建含有根结点、中间结点和叶子结点的文件哈希树，其中，所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成；所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；

[0011] 其中，所述目录及文件元数据获取步骤、所述文件目录树构建步骤和所述文件哈

希树构建步骤同时进行；

[0012] 增量文件获取步骤，在增量备份时，从根结点开始，逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树，定位含有增量文件的结点，获取增量文件。

[0013] 在其中一个实施例中，所述文件目录树构建步骤，包括：

[0014] 目录结点建立步骤，在对所述NAS文件系统进行深度优先遍历的同时，建立目录结点，令所述目录结点与当前目录对应，使所述目录结点包含：目录元数据、子文件元数据组和目录哈希值，其中，目录元数据为对应目录的元数据；子文件元数据组为对应目录的子文件元数据组；目录哈希值由所述目录元数据和所述子文件元数据组生成；

[0015] 文件目录树建立步骤，重复所述目录结点建立步骤，直至建立文件目录树，其中，所述文件目录树的目录结点与全部目录一一对应。

[0016] 在其中一个实施例中，所述文件哈希树构建步骤，包括：

[0017] 第一判断步骤，在建立目录结点的同时，判断所述目录结点对应的目录是否包含子目录；

[0018] 叶子结点建立步骤，若是，则重复所述目录结点建立步骤；若否，则建立文件哈希树的叶子结点，并使所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成；

[0019] 第二判断步骤，判断所述叶子结点的所有兄弟结点是否创建完成；

[0020] 中间结点建立步骤，若是，新建中间结点作为所述叶子结点的父结点，并使所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；若否，则重复目录结点建立步骤；

[0021] 根结点建立步骤，重复第一判断步骤至中间结点建立步骤，直至创建所述文件哈希树的根结点，并使所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成。

[0022] 在其中一个实施例中，所述目录及文件元数据获取步骤前，还包括步骤：

[0023] 根目录的子目录获取步骤，获取根目录下的所有子目录；

[0024] 线程池创建步骤，创建全局任务队列、工作线程和守护线程，三者并行处理数据；

[0025] 全局任务获取步骤，将根目录的子目录作为遍历任务传入所述全局任务队列；

[0026] 遍历任务执行步骤，所述工作线程从所述全局任务队列中取得遍历任务，并调用遍历任务函数对所述遍历任务进行处理；

[0027] 守护线程控制步骤，所述守护线程控制所述工作线程的数量处于预定阈值。

[0028] 在其中一个实施例中，所述全局任务获取步骤，还包括：

[0029] 遍历任务CPU占用时间记录步骤，备份完成时，记录所述全局任务队列中遍历任务的CPU占用时间；

[0030] 超时任务判断步骤，读取前次备份记录的全局任务队列中遍历任务的CPU占用时间，并检测是否存在超时任务；

[0031] 遍历任务细粒度化步骤，若是，则对将所述超时任务进行细粒度化处理，再使所述全局任务队列的队列头指向所述超时任务的下一遍历任务；若否，则执行遍历任务执行步骤。

[0032] 第二个方面，还提供了一种NAS文件系统的备份系统，所述系统包括：

[0033] 挂载模块，用于挂载NAS文件系统到备份代理上，将挂载点作为根目录；

[0034] 目录及文件元数据获取模块，用于深度优先遍历所述NAS文件系统，获取全部目录和文件元数据；

[0035] 文件目录树构建模块，用于根据所述目录和所述文件元数据，构建含有目录结点的文件目录树，其中，所述目录结点包括：目录元数据、子文件元数据组和目录哈希值；

[0036] 文件哈希树构建模块，用于根据所述目录和所述文件目录树，构建含有根结点、中间结点和叶子结点的文件哈希树，其中，所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成；所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；

[0037] 其中，所述目录及文件元数据获取模块、所述文件目录树构建模块和所述文件哈希树构建模块同时进行；

[0038] 增量文件获取模块，用于在增量备份时，从根结点开始，逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树，定位含有增量文件的结点，获取增量文件。

[0039] 在其中一个实施例中，所述文件目录树构建模块，包括：

[0040] 目录结点建立单元，用于在对所述NAS文件系统进行深度优先遍历的同时，建立目录结点，令所述目录结点与当前目录对应，使所述目录结点包含：目录元数据、子文件元数据组和目录哈希值，其中，目录元数据为对应目录的元数据；子文件元数据组为对应目录的子文件元数据组；目录哈希值由所述目录元数据和所述子文件元数据组生成；

[0041] 文件目录树建立单元，用于重复所述目录结点建立单元，直至建立文件目录树，其中，所述文件目录树的目录结点与全部目录一一对应。

[0042] 在其中一个实施例中，所述文件哈希树构建模块，还包括：

[0043] 第一判断单元，用于在建立目录结点的同时，判断所述目录结点对应的目录是否包含子目录；

[0044] 叶子结点建立单元，用于若是，则重复所述第一判断单元；若否，则建立文件哈希树的叶子结点，并使所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成；

[0045] 第二判断单元，用于判断所述叶子结点的所有兄弟结点是否创建完成；

[0046] 中间结点建立单元，用于若是，新建中间结点作为所述叶子结点的父结点，并使所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；若否，则重复第一判断单元；

[0047] 根结点建立单元，用于重复第一判断单元至中间结点建立单元，直至创建所述文件哈希树的根结点，并使所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成。

[0048] 在其中一个实施例中，所述目录及文件元数据获取模块，还包括：

[0049] 根目录的子目录获取单元，用于获取根目录下的所有子目录；

[0050] 线程池创建单元，用于创建全局任务队列、工作线程和守护线程，三者并行处理数据；

[0051] 全局任务获取单元，用于将根目录的子目录作为遍历任务传入所述全局任务队列；

[0052] 遍历任务执行单元，用于所述工作线程从所述全局任务队列中取得遍历任务，并

调用遍历任务函数对所述遍历任务进行处理；

[0053] 守护线程控制单元，用于控制所述工作线程的数量处于预定阈值。

[0054] 第三个方面，还提供一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现上述NAS文件系统的备份方法。

[0055] 本发明的有益效果：

[0056] 本发明通过对NAS文件系统进行深度优先遍历，构造文件目录树，并基于文件目录树构造文件哈希树，然后逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树，快速定位含有增量文件的结点，进而获取两次备份时间点之间的增量文件，实现了NAS文件系统的高效备份。

附图说明

[0057] 为了更清楚地说明本发明实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明 的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0058] 图1为本发明实施例一中NAS文件系统的备份方法的流程图。

[0059] 图2为本发明实施例一中任务细粒度化方法示意图。

[0060] 图3为本发明实施例一中文件哈希树的比较方法示意图。

[0061] 图4为本发明实施例二中NAS文件系统的备份系统的结构图。

[0062] 图5为本发明实施例二中目录及文件元数据获取模块的结构图。

[0063] 图6为本发明实施例二中文件目录树构建模块的结构图。

[0064] 图7为本发明实施例二中文件哈希树构建模块的结构图。

[0065] 图8为本发明实施例二中NAS文件系统遍历速度图。

[0066] 图9为本发明实施例二中定位增量文件耗费时间图。

具体实施方式

[0067] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0068] 本发明提供的方法，应用范围包括但不限于下述环境：备份代理的操作系统为Ubuntu-16.04，NAS系统的操作系统为OpenVaultMedia5.6.13，解析过程由C语言编写。

[0069] 实施例一

[0070] 如图1所示，提供一种NAS文件系统的备份方法，该方法包括步骤：

[0071] S1.挂载NAS文件系统到备份代理上，将挂载点作为根目录。

[0072] 可以理解的是，NAS文件系统挂载协议可以选择通用文件系统(Common Internet File System，CIFS)协议或网络文件系统(Network File System，NFS)协议。

[0073] S2.对所述NAS文件系统进行深度优先遍历，获取全部目录和文件元数据。

[0074] 可以理解的是，上述目录的元数据具体包括：目录名、inode结点号、目录创建时间。

[0075] 可以理解的是，上述文件元数据具体包括：文件名、inode结点号、文件修改时间、文件创建时间、文件大小、文件权限。

[0076] 可选的，所述S2步骤前，还包括：

[0077] S21.获取所述根目录下的所有子目录。

[0078] S22.创建全局任务队列、工作线程和守护线程，三者并行处理数据。

[0079] S23.将根目录的子目录作为遍历任务传入所述全局任务队列。

[0080] 还可选的，所述步骤S23，还包括：

[0081] S231.备份完成时，记录全局任务队列中遍历任务的CPU占用时间。

[0082] S232.读取前次备份记录的全局任务队列中遍历任务的CPU占用时间，并检测是否存在超时任务。

[0083] 其中，使用箱线图方法检测前次备份记录的全局任务队列中是否存在超时任务。具体而言，将全局任务队列中所有遍历任务的CPU占用时间从大到小排列，记这组数据的下四分位数为Q1，上四分位数为Q3，则四分位距IQR=Q3-Q1。当某遍历任务的CPU占用时间大于上边缘Q3+1.5IQR时，则该遍历任务为超时任务。

[0084] S233.若是，则对将所述超时任务进行细粒度化处理，再使所述全局任务队列的队列头指向所述超时任务的下一遍历任务；若否，则执行步骤S24。

[0085] 为了便于理解，具体的，对步骤S231-S233提供一个操作实例：如表1所示，前次备份时，只将文件系统根目录下的子目录组作为遍历任务，编号分别为1-10。当编号为2-10的任务全部执行完毕后，编号为1的任务还将继续执行很长时间，这导致此时的工作线程变成了单线程运行，无法利用多线程的优势。这是因为遍历的文件系统目录结构不均衡，因此每个遍历任务的CPU占用时间差异巨大。如图2所示，本次备份时，使用箱线图方法判断前次备份中是否存在超时任务并细粒度化超时任务：将前次备份记录的这组数据按CPU占用时间从大到小排序，可得这组数据的下四分位数Q1为23.75秒，上四分位数Q3为34.775秒，则四分位距IQR=11.025秒，上边缘Q3+1.5IQR=51.3125秒，则可判定CPU占用时间为262.26秒的任务1为超时任务。所以，本次备份将任务1对应目录的子目录1/1和1/2作为子任务传入全局任务队列，并将队列头指向任务1的下一遍历任务。与不采用细粒度方法直接遍历相比，将超时任务细粒度化后再遍历的遍历速度提高了100%。

[0086] 表1 前次备份遍历任务CPU占用时间表

任务编号	1	2	3	4	5	6	7	8	9	10
CPU占用时间（秒）	262.26	35	34.7	24.8	26.1	26.3	20.6	25.5	26.7	6.7

[0087] S24.所述工作线程从所述全局任务队列中取得遍历任务，并调用遍历任务函数对所述遍历任务进行处理。

[0088] S25.所述守护线程控制所述工作线程的数量处于预定阈值。

[0089] 可以理解的是，当工作线程数大于任务需求时，所述守护线程消除部分工作线程，减少工作线程数，回收线程资源，避免不必要的切换上下文开销。

[0090] S3.根据所述目录和所述文件元数据，构建含有目录结点的文件目录树，其中，所述目录结点包括：目录元数据、子文件元数据组和目录哈希值。

[0091] 可选的，所述S3步骤，包括：

[0092] S31.在对所述NAS文件系统进行深度优先遍历的同时，建立目录结点，令所述目录结点与当前目录对应，使所述目录结点包含：目录元数据、子文件元数据组和目录哈希值，其中，目录元数据为对应目录的元数据；子文件元数据组为对应目录的子文件元数据组；目录哈希值由所述目录元数据和所述子文件元数据组生成。

[0093] 需要理解的是，使用readdir()函数获取同一目录的子文件组时，其返回结果是随机的，这导致具有相同数据的目录结点经哈希函数计算出的目录哈希值会发生变化。因此，根据文件系统中能够确定唯一文件的特征对子文件组元数据排序，以保证具有相同数据的目录结点经哈希函数计算出的目录哈希值不变，该特征在NFS协议中为inode结点号，在CIFS协议中为文件创建时间。

[0094] 还需要理解的是，由于增量文件可由文件名和文件修改时间确定，为了提高计算效率，目录哈希值由目录元数据中的目录名和有序子文件元数据组中的子文件名和文件修改时间经哈希算法生成。

[0095] 需要注意的是，由于不需要考虑隐私保护，因此哈希算法可以采用非加密哈希算法以提高计算效率。

[0096] S32.重复所述目录结点建立步骤，直至建立文件目录树，其中，所述文件目录树的目录结点与全部目录一一对应。

[0097] S4.根据所述目录和所述文件目录树，构建含有根结点、中间结点和叶子结点的文件哈希树，其中，所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成；所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成。

[0098] 需要理解的是，上述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成，具体是指：所述叶子结点的结点哈希值由所述叶子结点对应的目录结点的目录哈希值生成。

[0099] 还需要理解的是，上述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成，具体是指：所述中间结点的结点哈希值由所述中间结点的子结点的结点哈希值和所述中间结点对应的目录结点的目录哈希值生成。

[0100] 还再需要理解的是，上述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成，具体是指：所述根结点的结点哈希值由所述根结点的子结点的结点哈希值和所述根结点对应的目录结点的目录哈希值生成。

[0101] 其中，步骤S2至步骤S4同时进行。

[0102] 可选的，所述S4步骤，包括：

[0103] S41.在建立目录结点的同时，判断所述目录结点对应的目录是否包含子目录。

[0104] S42.若是，则重复步骤S41；若否，则建立文件哈希树的叶子结点，并使所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成。

[0105] 需要注意的是，为了简化计算，所述叶子结点的结点哈希值等于其对应的目录结

点的目录哈希值。

[0106] S43.判断所述叶子结点的所有兄弟结点是否创建完成。

[0107] S44.若是,新建中间结点作为所述叶子结点的父结点,并使所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成;若否,则重复步骤S41。

[0108] 需要理解的是,使用readdir()函数获取同一目录的子目录组时,其返回结果是随机的,这导致具有相同子结点数据的中间结点经哈希函数计算出的结点哈希值会发生变化。因此,根据文件系统中能够确定唯一目录的特征对子结点排序,以保证具有相同子结点数据的中间结点经哈希函数计算出的结点哈希值不变,该特征在NFS协议中为inode结点号,在CIFS协议中为目录创建时间。

[0109] S45.重复步骤S41至步骤S44,直至创建所述文件哈希树的根结点,并使所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成。

[0110] S5.在增量备份时,从根结点开始,逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树,定位含有增量文件的结点,获取增量文件。

[0111] 为了便于理解,具体的,对步骤S5提供一个操作实例:如图3所示,版本1为上次备份构建的文件哈希树,版本2为本次备份构建的文件哈希树。这两棵文件哈希树的结构类似,都分别有1个根结点f1,2个1级结点f2、f3,4个2级结点f4-f7。从根结点f1开始,逐级比较版本1和版本2的结点的结点哈希值。版本1的f1结点哈希值和版本2的f1结点哈希值不同,则继续比较版本1的f2结点哈希值和版本2的f2结点哈希值,以及版本1的f3结点哈希值和版本2的f3结点哈希值。此时版本1的f2结点哈希值和版本2的f2结点哈希值相同,无需再比较其子结点f4、f5。而版本1的f3结点哈希值和版本2的f3结点哈希值不同,则继续比较版本1的f6结点哈希值和版本2的f6结点哈希值,以及版本1的f7结点哈希值和版本2的f7结点哈希值。此时版本1的f6结点哈希值和版本2的f6结点哈希值相同,并且f6没有子结点,无需再比较。但版本1的f7结点哈希值和版本2的f7结点哈希值不同,f7没有子结点,说明f7是含有增量文件的增量结点。最后,只需遍历比较f7的子文件元数据组中的文件名和文件修改时间就可获取版本1和版本2两个备份时间点之间的增量文件。

[0112] 本实施例的技术方案,通过对NAS文件系统进行深度优先遍历,构造文件目录树,并基于文件目录树构造文件哈希树,然后逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树,快速定位含有增量文件的结点,进而获取两次备份时间点之间的增量文件,实现了NAS文件系统的高效备份。本实例还采用任务细粒度化合理分配NAS文件系统遍历任务,使得多线程调度负载均衡,进一步提高NAS文件系统的遍历速度。

[0113] 实施例二

[0114] 如图4所示,在一个实施例中,提供了一种NAS文件系统的备份系统,该系统包括:

[0115] 挂载模块1001,用于挂载NAS文件系统到备份代理上,将挂载点作为根目录;

[0116] 目录及文件元数据获取模块1002,用于深度优先遍历所述NAS文件系统,获取全部目录和文件元数据;

[0117] 文件目录树构建模块1003,用于根据所述目录和所述文件元数据,构建含有目录结点的文件目录树,其中,所述目录结点包括:目录元数据、子文件元数据组和目录哈希值;

[0118] 文件哈希树构建模块1004,用于根据所述目录和所述文件目录树,构建含有根结

点、中间结点和叶子结点的文件哈希树，其中，所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成；所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；

[0119] 其中，所述目录及文件元数据获取模块1002、所述文件目录树构建模块1003和所述文件哈希树构建模块1004同时进行；

[0120] 增量文件获取模块1005，用于在增量备份时，从根结点开始，逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树，定位含有增量文件的结点，获取增量文件。

[0121] 可选的，如图5所示，在本实施例的基础上，所述目录及文件元数据获取模块1002包括：

[0122] 根目录的子目录获取单元10021，用于获取根目录下的所有子目录；

[0123] 线程池创建单元10022，用于创建全局任务队列、工作线程和守护线程，三者并行处理数据；

[0124] 全局任务获取单元10023，用于将根目录的子目录作为遍历任务传入所述全局任务队列；

[0125] 遍历任务执行单元10024，用于所述工作线程从所述全局任务队列中取得遍历任务，并调用遍历任务函数对所述遍历任务进行处理；

[0126] 守护线程控制单元10025，用于控制所述工作线程的数量处于预定阈值。

[0127] 可选的，如图6所示，在本实施例的基础上，所述文件目录树构建模块1003包括：

[0128] 目录结点建立单元10031，用于在对所述NAS文件系统进行深度优先遍历的同时，建立目录结点，令所述目录结点与当前目录对应，使所述目录结点包含：目录元数据、子文件元数据组和目录哈希值，其中，目录元数据为对应目录的元数据；子文件元数据组为对应目录的子文件元数据组；目录哈希值由所述目录元数据和所述子文件元数据组生成；

[0129] 文件目录树建立单元10032，用于重复所述目录结点建立单元10031，直至建立文件目录树，其中，所述文件目录树的目录结点与全部目录一一对应。

[0130] 可选的，如图7所示，在本实施例的基础上，所述文件哈希树构建模块1004包括：

[0131] 第一判断单元10041，用于在建立目录结点的同时，判断所述目录结点对应的目录是否包含子目录；

[0132] 叶子结点建立单元10042，用于若是，则重复所述第一判断单元10041；若否，则建立文件哈希树的叶子结点，并使所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成；

[0133] 第二判断单元10043，用于判断所述叶子结点的所有兄弟结点是否创建完成；

[0134] 中间结点建立单元10044，用于若是，新建中间结点作为所述叶子结点的父结点，并使所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；若否，则重复第一判断单元10041；

[0135] 根结点建立单元10045，用于重复第一判断单元10041至中间结点建立单元10044，直至创建所述文件哈希树的根结点，并使所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成。

[0136] 下面，我们提供两组实验结果，便于进一步阐述本实施例，实验环境如表2所示：

表2 实验环境

[0137]

	内存	处理器	操作系统
NAS系统	金士顿DDR4 2666MHz 4GB	Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz 两核	OpenVaultMedia5.6.13
备份代理	金士顿DDR4 2666MHz 4GB	Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz 两核	Ubuntu-16.04

[0138] 本实验使用了两个具有不同目录深度和目录结构的文件数据集,以测试本实施例对于不同宽度层次和深度层次的文件数据集的效果。表3列出了每个文件数据集的目录深度,目录结构和总文件数。宽度集的目录深度为3层,目录结构扁平,第一层有10个目录,第二层到第四层分别有100个目录,最底层的目录下分别有100个文件,总共101010个目录,1000万个文件。深度集的目录深度为7层,目录结构细长,每层分别有4到10个目录,最底层的目录下分别有100个文件,总共131560个目录,1000万个文件。

[0139] 表3 文件数据集

[0140]

	目录深度	目录结构	总文件数
宽度集	3	10x100x100	1000万
深度集	7	10x5x5x5x4x5x4	1000万

[0141] 本实验的评价指标为NAS文件系统的遍历速度和定位增量文件耗费时间。经实验,结果如图8和图9所示,分析具体如下:

[0142] 使工作线程的任务对象,即数据集不变,改变线程池工作线程的数量,测量完成任务的时间,结果如图8所示。当线程数小于6个时,随着线程数的增加,遍历速度提高。当线程数超过6个时,由于线程上下文切换频繁导致开销增大,遍历速度提升放缓,甚至降低。因此,最优的遍历工作线程数为3倍CPU数。此时,与单线程相比,多线程遍历文件系统的速度提高了100%。

[0143] 在深度集和宽度集中,分别随机在100个目录中生成100个新文件,100个目录中生成10000个新文件,1000个目录中生成10000个新文件,测量这三种情况下,定位增量文件耗费的时间。结果如图9所示,在深度集中定位增量文件耗费的时间是在宽度集中定位相同量级的增量文件耗费时间的两倍,这表明定位增量文件的时间与文件系统的目录结构有关,目录结构越扁平,定位所消耗的时间越短。结果同样如图9所示,在相同目录结构的文件集中,定位随机在100个目录中生成的100个增量文件或生成的10000个增量文件所耗费的时间差不多,而定位随机在1000个目录中生成的10000个增量文件所耗费的时间是前者的7倍,说明定位增量文件耗费的时间与增量目录结点的数量密切相关,而与增量文件的数量相关性不大。因此,本实施例通过对NAS文件系统进行深度优先遍历,构造文件目录树,并基于文件目录树构造文件哈希树,只需花费定位增量目录结点的时间就可获取增量文件,这大大减少了定位增量文件耗费时间。

[0144] 本实施例的技术方案，挂载模块1001，用于挂载NAS文件系统到备份代理上，将挂载点作为根目录；目录及文件元数据获取模块1002，用于深度优先遍历所述NAS文件系统，获取全部目录和文件元数据；文件目录树构建模块1003，用于根据所述目录和所述文件元数据，构建含有目录结点的文件目录树；文件哈希树构建模块1004，用于根据所述目录和所述文件目录树，构建含有根结点、中间结点和叶子结点的文件哈希树；增量文件获取模块1005，用于在增量备份时，从根结点开始，逐级比较本次备份构建的文件哈希树和上次备份构建的文件哈希树，定位含有增量文件的结点，获取增量文件。本实例使用多线程提高了CPU的利用率，加快了NAS文件系统的遍历速度。使用线程池管理多线程，不但减少了创建、消耗线程的开销，还使得备份元数据文件索引、管理简单。

[0145] 实施例三

[0146] 在一个实施例中，提供了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现实施例一所述NAS文件系统的备份方法。

[0147] 本发明实施例的计算机存储介质，可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0148] 以上所述实施例仅表达了本发明的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对本发明专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本发明构思的前提下，还可以做出若干变形和改进，这些都属于本发明的保护范围。因此，本发明专利的保护范围应以所附权利要求为准。

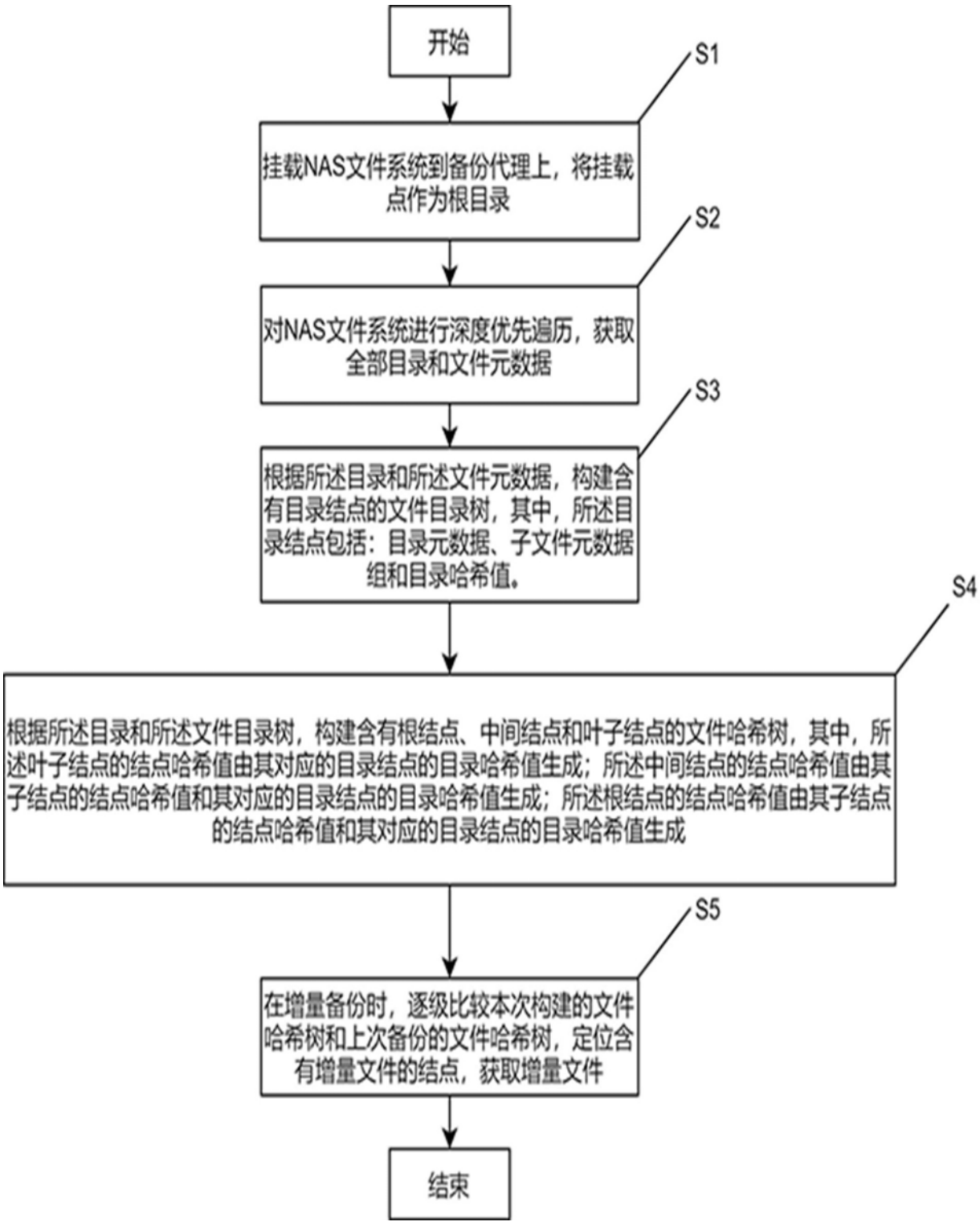

开始
S1
挂载NAS文件系统到备份代理上，将挂载点作为根目录
S2
对NAS文件系统进行深度优先遍历，获取全部目录和文件元数据
S3
根据所述目录和所述文件元数据，构建含有目录结点的文件目录树，其中，所述目录结点包括：目录元数据、子文件元数据组和目录哈希值。
S4
根据所述目录和所述文件目录树，构建含有根结点、中间结点和叶子结点的文件哈希树，其中，所述叶子结点的结点哈希值由其对应的目录结点的目录哈希值生成；所述中间结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成；所述根结点的结点哈希值由其子结点的结点哈希值和其对应的目录结点的目录哈希值生成
S5
在增量备份时，逐级比较本次构建的文件哈希树和上次备份的文件哈希树，定位含有增量文件的结点，获取增量文件
结束

图1

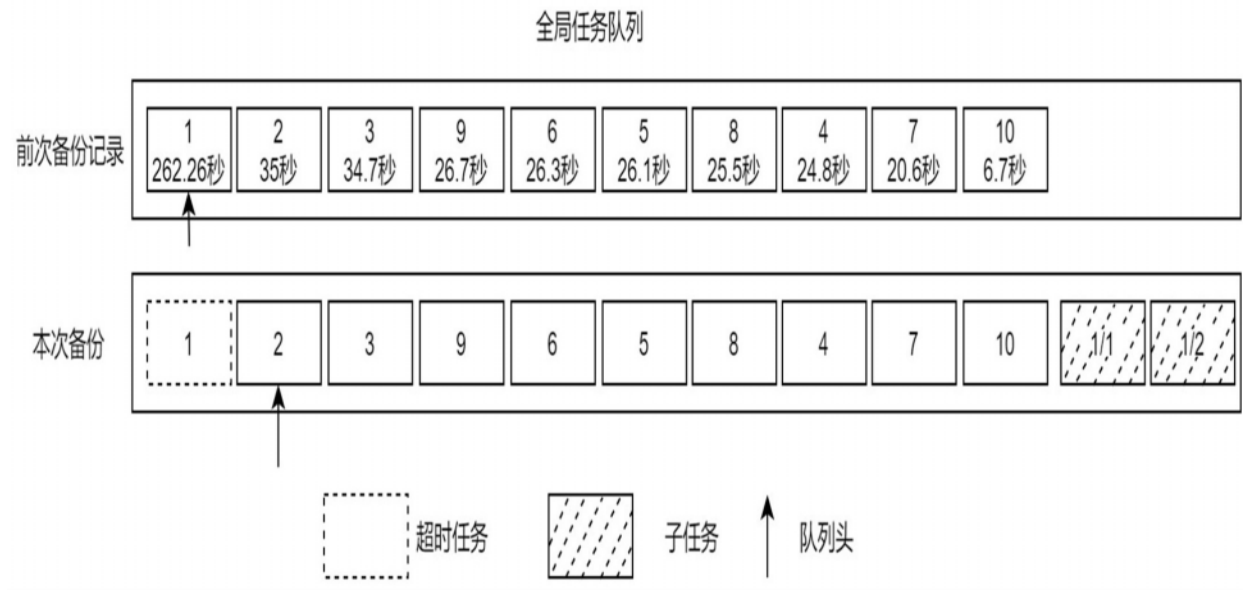

全局任务队列
前次备份记录
1
262.26秒
2
35秒
3
34.7秒
9
26.7秒
6
26.3秒
5
26.1秒
8
25.5秒
4
24.8秒
7
20.6秒
10
6.7秒
本次备份
1
2
3
9
6
5
8
4
7
10
1/1
1/2
超时任务
子任务
队列头

图2

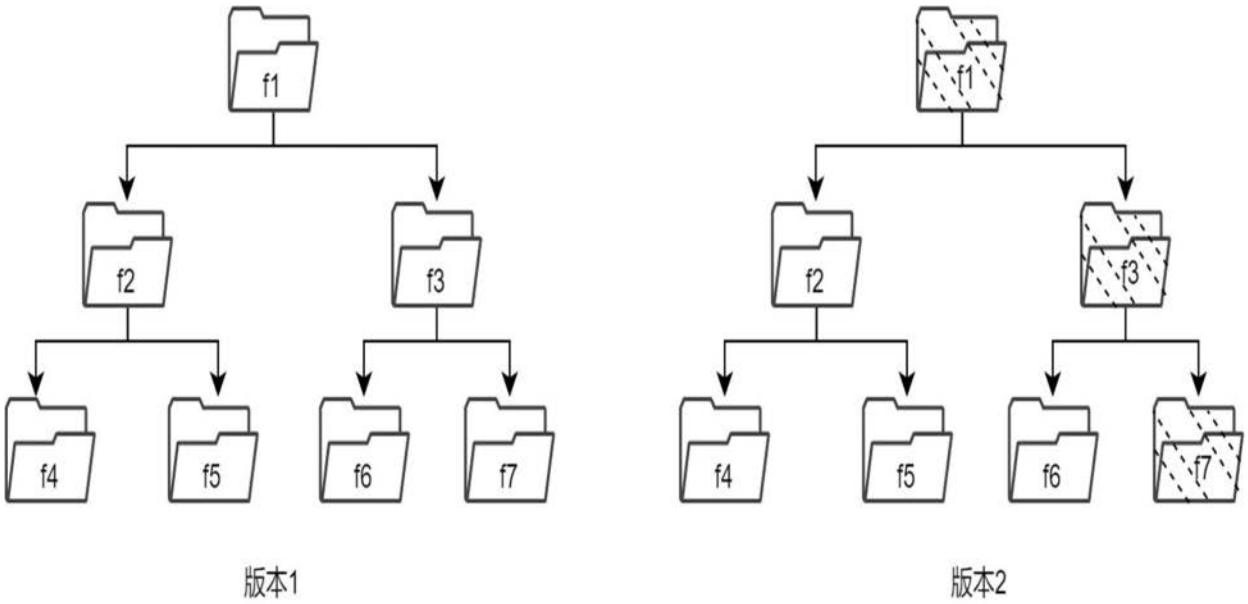

f1
f2
f3
f4
f5
f6
f7
版本1
f1
f2
f3
f4
f5
f6
f7
版本2

图3

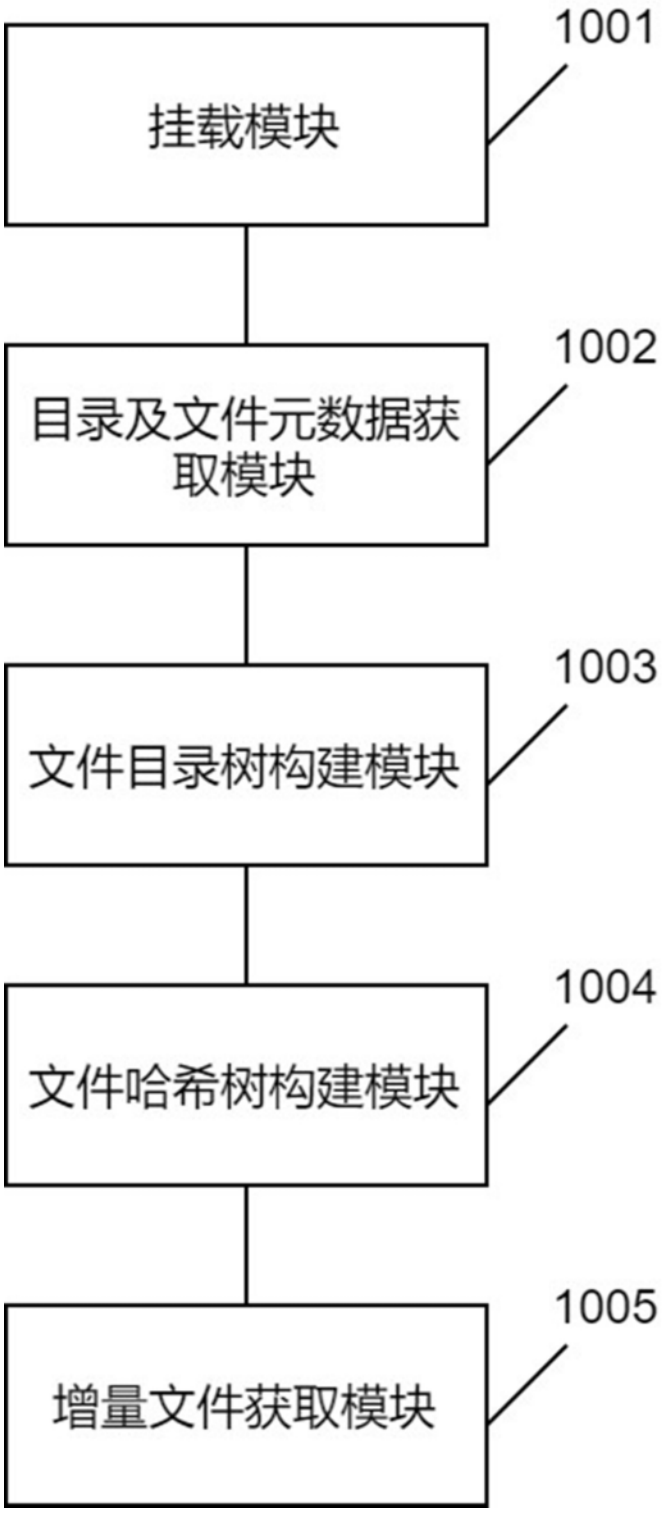

1001
挂载模块
1002
目录及文件元数据获取模块
1003
文件目录树构建模块
1004
文件哈希树构建模块
1005
增量文件获取模块

图4

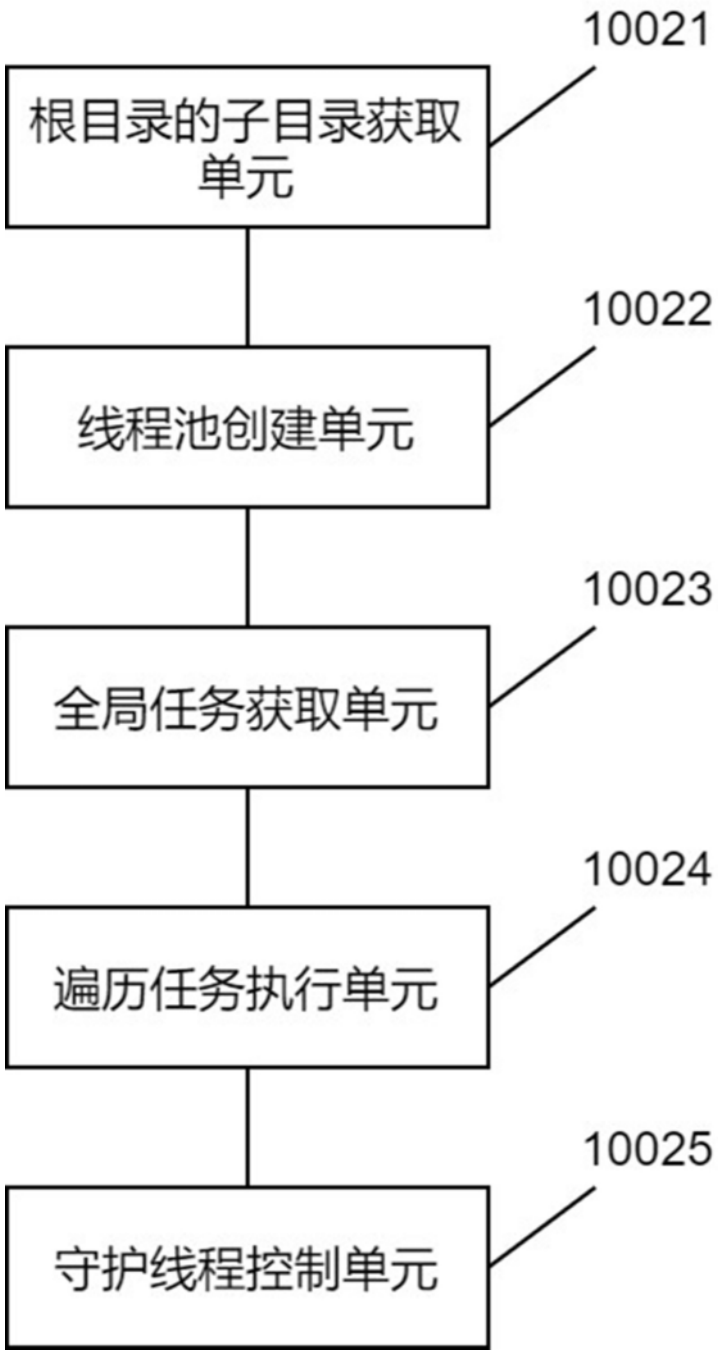

10021
根目录的子目录获取单元
10022
线程池创建单元
10023
全局任务获取单元
10024
遍历任务执行单元
10025
守护线程控制单元

图5

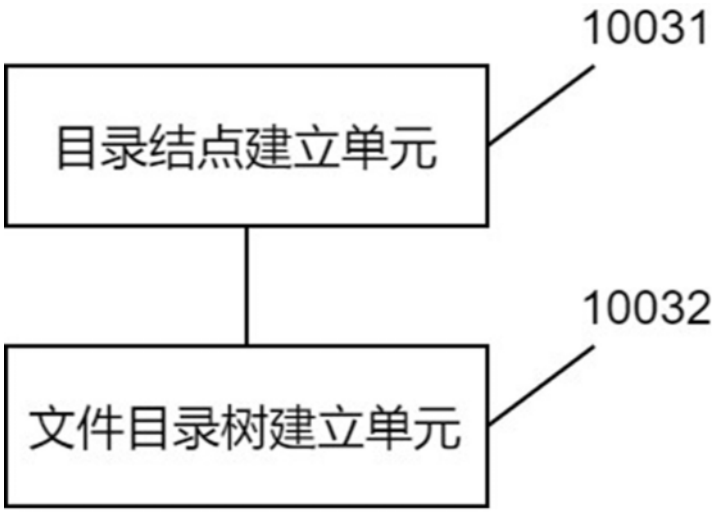


图6

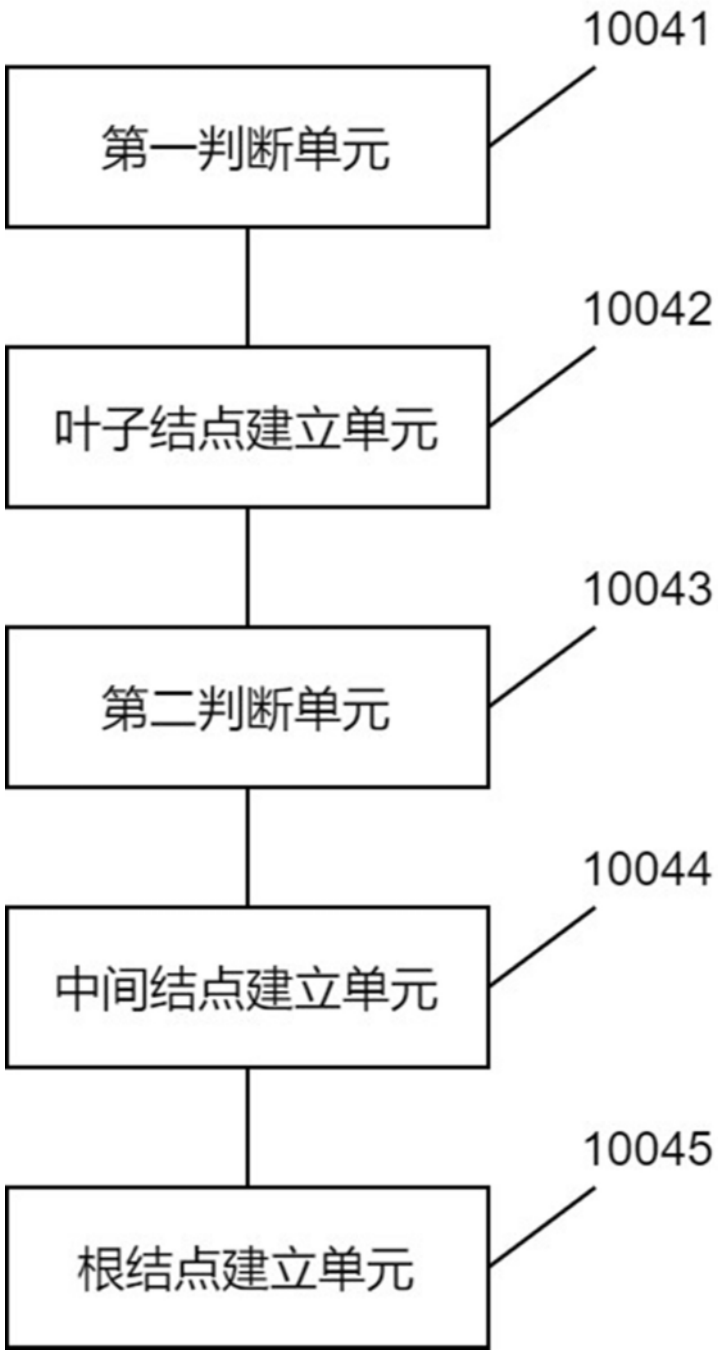

10041
第一判断单元
10042
叶子结点建立单元
10043
第二判断单元
10044
中间结点建立单元
10045
根结点建立单元

图7

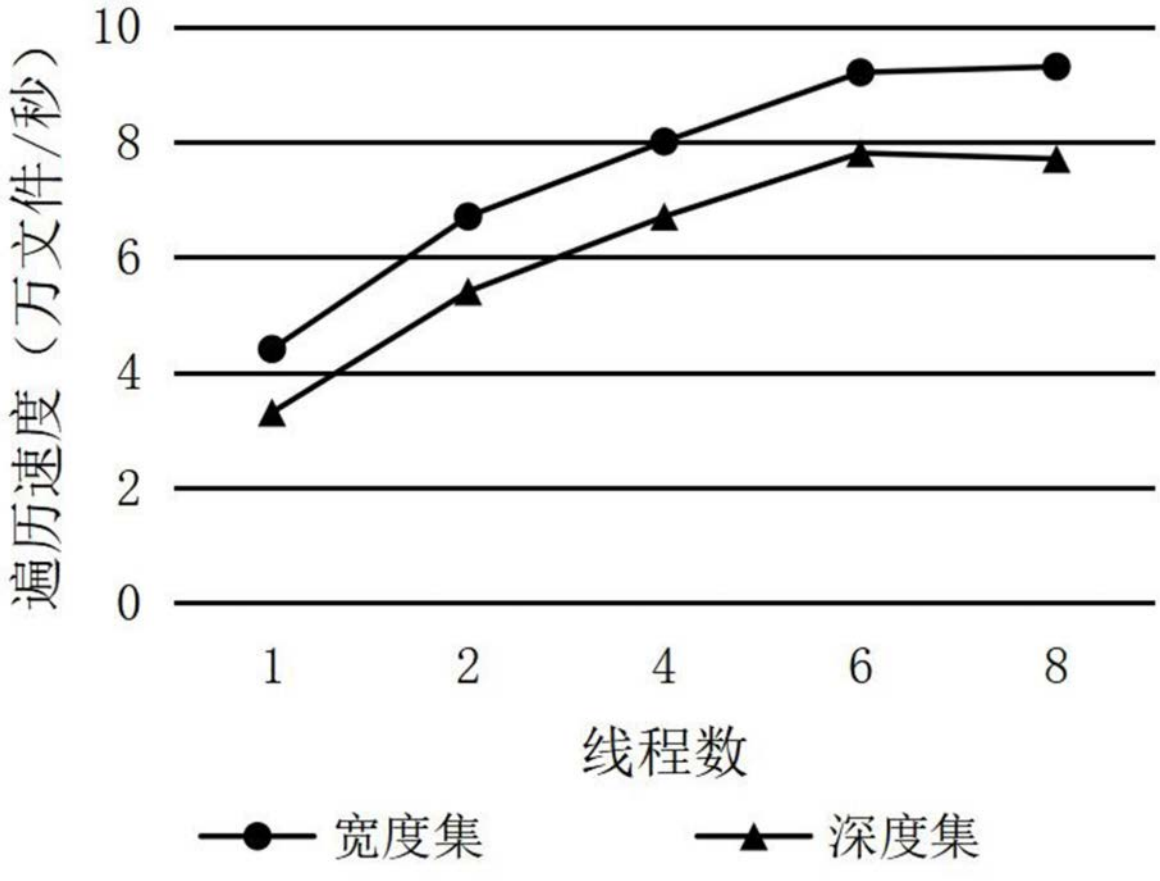


图8

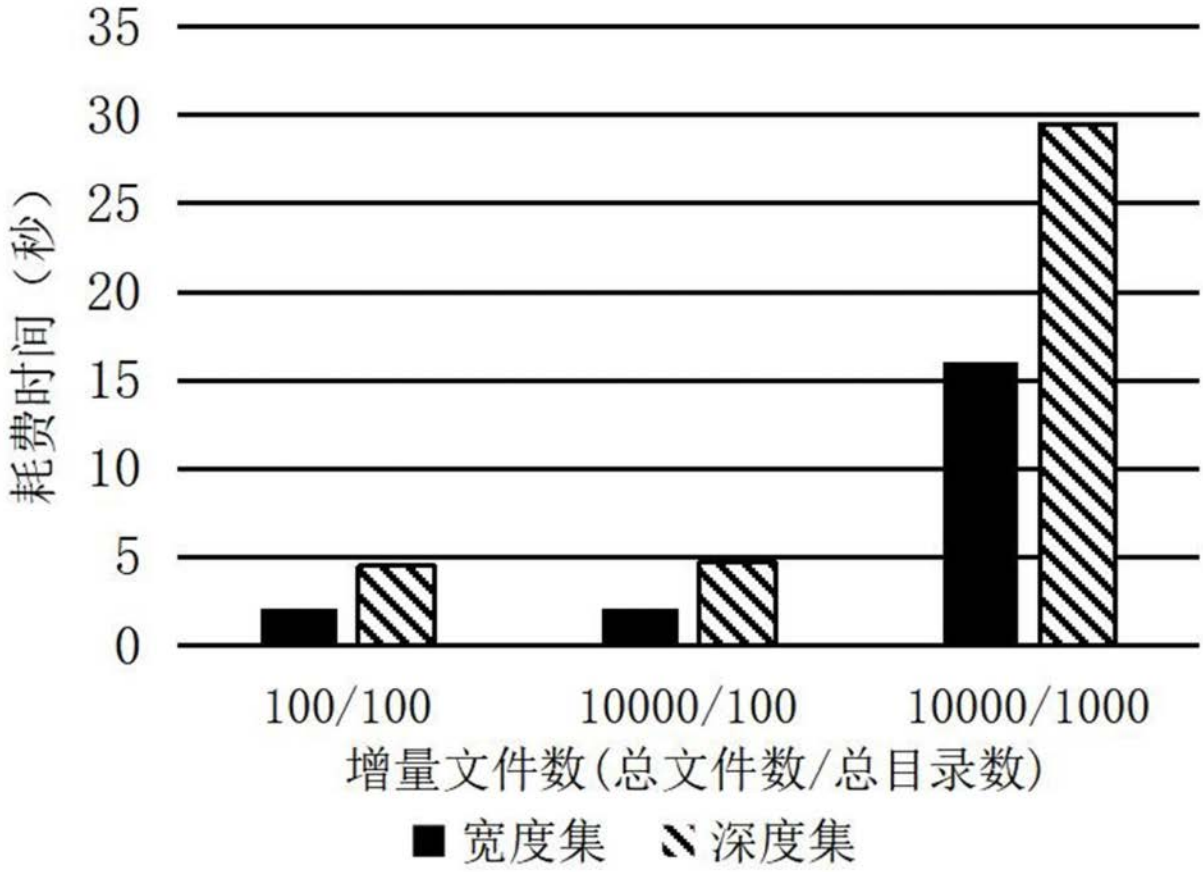

35
30
25
20
15
10
5
0
耗费时间（秒）
100/100
10000/100
10000/1000
增量文件数(总文件数/总目录数)
宽度集
深度集

图9