

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利说明书

专利号 ZL 200610115770.6

[45] 授权公告日 2009 年 1 月 14 日

[11] 授权公告号 CN 100452052C

[22] 申请日 2006.8.16

[21] 申请号 200610115770.6

[30] 优先权

[32] 2005.9.27 [33] US [31] 11/236,450

[73] 专利权人 国际商业机器公司

地址 美国纽约

[72] 发明人 J·E·阿德金斯

M·A·格拉布斯

[56] 参考文献

CN1184538A 1998.6.10

US5978792A 1999.11.2

US2004/0268068A1 2004.12.30

CN1283029A 2001.2.7

JP2000-47897A 2000.2.18

审查员 李福永

[74] 专利代理机构 北京市中咨律师事务所
代理人 于 静 李 峥

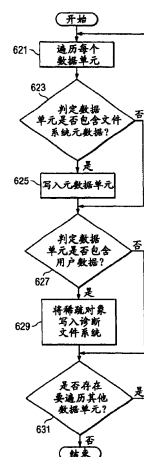
权利要求书 5 页 说明书 13 页 附图 4 页

[54] 发明名称

捕获和传送文件系统的密集诊断数据的方法和装置

[57] 摘要

处理器选择文件系统元数据。处理器标识文件系统中的每个数据单元。所述处理器判定所述数据单元是否包含文件系统元数据。如果所述处理器判定所述数据单元包含文件系统元数据，则所述处理器将元数据的数据单元写入诊断文件。然后所述处理器判定所述数据单元是否包含用户数据。如果所述数据单元包含用户数据，则所述处理器将稀疏对象写入所述诊断文件。



1. 一种计算机实现的用于生成文件系统元数据的方法，所述方法包括：

标识文件系统多个数据单元；

判定所述多个数据单元中的一个数据单元是否包含文件系统元数据；

将所述文件系统元数据从所述数据单元写入诊断文件系统，以响应判定所述数据单元包含所述文件系统元数据；

判定所述多个数据单元中的一个数据单元是否包含用户数据；以及

将稀疏对象写入诊断文件，以响应判定所述数据单元包含用户数据，其中所述诊断文件与所述文件系统元数据关联。

2. 根据权利要求 1 的计算机实现的用于生成文件系统元数据的方法，还包括：

将所述诊断文件与所述诊断文件系统关联；以及

安装所述诊断文件系统。

3. 根据权利要求 1 的计算机实现的用于生成文件系统元数据的方法，还包括：

将所述诊断文件合并到密集诊断数据中。

4. 根据权利要求 3 的计算机实现的用于生成文件系统元数据的方法，其中合并还包括：

将所述诊断文件存储在所述诊断文件系统中；以及

压缩所述诊断文件系统。

5. 根据权利要求 3 的计算机实现的用于生成文件系统元数据的方法，还包括：

传送所述密集诊断数据。

6. 根据权利要求 5 的计算机实现的用于生成文件系统元数据的方法，其中合并还包括：

将所述诊断文件系统存储为文件；以及

压缩所述文件以形成所述密集诊断数据。

7. 根据权利要求 5 的计算机实现的用于生成文件系统元数据的方法，还包括：

创建所述诊断文件系统；以及安装所述诊断文件系统。

8. 根据权利要求 7 的计算机实现的用于生成文件系统元数据的方法，其中合并还包括：

将所述诊断文件系统存储为文件；以及

将所述文件压缩到密集诊断数据。

9. 根据权利要求 8 的计算机实现的用于生成文件系统元数据的方法，其中数据单元包括块。

10. 根据权利要求 1 的计算机实现的用于生成文件系统元数据的方法，其中为所述多个数据单元中的每个数据单元重复执行从所述判定是否包含文件系统元数据的步骤到所述写入所述稀疏对象的步骤的步骤。

11. 一种数据处理系统，所述数据处理系统包括：

总线；

连接到所述总线的存储设备，其中计算机可用代码位于所述存储设备中；

连接到所述总线的通信单元；

连接到所述总线的处理单元，其中所述处理单元执行所述计算机可用代码来生成文件系统元数据，所述处理单元还执行所述计算机可用代码以执行以下操作：标识文件系统多个数据单元；判定数据单元是否包含文件系统元数据；将所述文件系统元数据从所述数据单元写入诊断文件系统，以响应判定所述数据单元包含所述文件系统元数据；判定所述数据单元是否包含用户数据；以及将稀疏对象写入所述诊断文件，以响应判定所述数据单元包含用户数据，其中所述诊断文件与所述文件系统元数据关联。

12. 根据权利要求 11 的数据处理系统，其中所述处理单元还执行所述计算机可用代码以将所述诊断文件与所述诊断文件系统关联；以及安装所述诊断文件系统。

13. 根据权利要求 11 的数据处理系统, 其中所述处理单元还执行所述计算机可用代码以将所述诊断文件合并到密集诊断数据中。

14. 根据权利要求 13 的数据处理系统, 其中在执行所述计算机可用代码以进行合并中, 所述处理单元还执行所述计算机可用代码以将所述诊断文件存储在所述诊断文件系统中; 以及压缩所述诊断文件系统。

15. 根据权利要求 13 的数据处理系统, 其中所述处理单元还执行所述计算机可用代码以传送所述密集诊断数据, 并且其中在执行所述计算机可用代码以进行合并中, 所述处理单元还执行所述计算机可用代码以将所述诊断文件系统存储为文件; 以及压缩所述文件以形成密集诊断数据。

16. 根据权利要求 15 的数据处理系统, 其中所述处理单元还执行所述计算机可用代码以创建诊断文件系统; 以及安装所述诊断文件系统。

17. 根据权利要求 16 的数据处理系统, 其中在执行所述计算机可用代码以进行合并中, 所述处理单元还执行所述计算机可用代码以将所述诊断文件系统存储为文件; 以及将所述文件压缩到密集诊断数据。

18. 根据权利要求 11 的数据处理系统, 其中所述处理单元还执行所述计算机可用代码以重复地执行为所述多个数据单元中的每个数据单元做出判定是否包含文件系统元数据的所述计算机可用代码; 重复地执行从所述数据单元将文件系统元数据写入所述诊断文件系统, 以响应判定所述数据单元包含所述文件系统元数据的所述计算机可用代码; 重复地执行为所述多个数据单元中的每个数据单元做出判定是否包含用户数据的所述计算机可用代码; 以及重复地执行将稀疏对象写入所述诊断文件, 以响应判定所述数据单元包含用户数据的所述计算机可用代码, 其中所述诊断文件与所述文件系统元数据关联。

19. 一种用于生成文件系统元数据的装置, 所述装置包括:

用于标识文件系统多个数据单元的装置;

用于判定所述数据单元是否包含文件系统元数据的装置;

用于将所述文件系统元数据从所述数据单元写入诊断文件系统的装置, 以响应所述用于判定所述数据单元是否包含所述文件系统元数据的装

置的肯定判定;

用于判定所述数据单元是否包含用户数据的装置; 以及

用于将稀疏对象写入诊断文件的装置, 以响应所述用于判定所述数据单元是否包含用户数据的装置的肯定判定, 其中所述诊断文件与所述文件系统元数据关联。

20. 根据权利要求 19 的用于生成文件系统元数据的装置, 还包括:

用于将所述诊断文件与所述诊断文件系统关联的装置; 以及

用于安装所述诊断文件系统的装置。

21. 根据权利要求 19 的用于生成文件系统元数据的装置, 还包括:

用于将所述诊断文件合并到密集诊断数据中的装置。

22. 根据权利要求 21 的用于生成文件系统元数据的装置, 其中用于合并的装置还包括:

用于将所述诊断文件存储在所述诊断文件系统上的装置; 以及

用于将所述诊断文件系统压缩到所述密集诊断数据的装置。

23. 根据权利要求 21 的用于生成文件系统元数据的装置, 还包括:

用于传送所述密集诊断数据的装置。

24. 根据权利要求 23 的用于生成文件系统元数据的装置, 其中用于合并的装置还包括:

用于将所述诊断文件系统存储为文件的装置; 以及

用于压缩所述文件以形成所述密集诊断数据的装置。

25. 根据权利要求 23 的用于生成文件系统元数据的装置, 还包括:

用于创建所述诊断文件系统的装置; 用于安装所述诊断文件系统的装置; 并且其中用于合并的装置还包括:

用于将所述诊断文件系统存储为文件的装置; 以及

用于将所述文件压缩到密集诊断数据的装置。

26. 根据权利要求 19 的用于生成文件系统元数据的装置, 还包括重复地使用所述用于判定是否包含文件系统元数据的装置、所述用于写入所述文件系统元数据的装置、所述用于判定是否包含用户数据的装置、和所

述用于写入所述稀疏对象的装置的装置。

27. 一种用于维护文件系统的方法，所述方法包括：

将用于从所述文件系统生成诊断文件系统的计算机可用程序代码发送给客户，其中所述计算机可用程序代码将文件系统元数据从每个包含所述文件系统元数据的数据单元写入到诊断文件系统；以及对于每个包含用户数据的数据单元，将稀疏对象写入诊断文件，其中所述诊断文件与所述文件系统元数据关联；

从客户接收所述诊断文件系统；

维护所述诊断文件系统以形成已维护的文件系统；以及

将所述已维护的文件系统返回给所述客户。

28. 根据权利要求 27 的方法，其中所述维护步骤包括：

对所述诊断文件系统执行恢复操作以形成已维护的文件系统。

29. 根据权利要求 27 的方法，还包括：

为在维护所述诊断文件系统中所提供的维护对所述客户计费。

捕获和传送文件系统的密集诊断数据的方法和装置

技术领域

本发明一般地涉及维护和改善文件系统完整性。更具体地说，本发明涉及获得密集格式的文件系统元数据以便可以离站传送此类数据以供专家检查和修复的方法、装置和计算机可用代码。

背景技术

在计算中，文件系统是用于存储和组织计算机文件和它们包含的数据以使查找和访问所述文件变得容易的结构。文件系统可以使用诸如硬盘或CD-ROM之类的存储设备并且涉及维护文件的物理位置。替代地，文件系统可以是虚拟文件系统。虚拟文件系统仅作为对虚拟数据或网络上的数据的访问方法存在。

文件系统包括两类数据。通常占绝大多数的第一类数据是用户数据。例如，用户数据的内容类型可以例如是文本、图形、音乐和计算机指令。第二类数据是文件系统元数据。文件系统元数据是用户数据之外的所有数据。元数据通知处理系统包含用户数据的文件的统计信息和结构。

文件系统元数据包括超级块，所述超级块提供文件系统概要并包括指向其他信息的指针。inode是与每个文件关联的文件系统元数据。inode表示文件的长度（以字节为单位）、关联的设备标识符、用户标识符、组标识符、可以是唯一的inode号、文件模式、时间戳以及引用计数。

块是可以分配给文件的磁盘存储的最小单位。例如，处理器可以在特定的文件系统上建立一个1024字节的块。这意味着文件几乎总是填满一个或多个块，且最后一个块只部分地由数据占据。

文件的各部分可以驻留在许多块中，有时分散在磁盘驱动器中。inode

可在 inode 内列出若干块。对于较大的文件，inode 可以包括一个间接块，该间接块可以指向其他块的列表。通常这导致指向块的相继更深层的间接块的多级树状结构。

文件系统通常是非常大的数据结构。当处理器对文件系统做出更改时，处理器通常需要许多单独的写入操作。有时，错误或其他故障中断了写入序列，例如，可能出现电源故障。

当处理器在这种情况下遇到错误时可能会出现竞争条件。竞争条件实质上是当电子设备中的两个事件彼此竞争以查看哪个事件将影响设备的状态或输出时所出现的情况，其中到达的第一个事件或信号会控制设备的状态。在文件系统（如日记文件系统）的上下文中，当更新文件系统时出现两个步骤：1) 删除文件的目录表项；以及 2) 在空闲空间映射中将文件的 inode 标记为空闲空间。

如果电源出现故障，并且步骤 1 在崩溃之前立即发生，则将出现孤立的 inode，并且分配给存储的块将显得比实际分配的块要多。如果步骤 2 在崩溃之前立即发生，则尚未删除的 inode 将被标记为空闲并且可能被其他内容覆盖。

日记文件系统（一种特殊类型的文件系统）具有其他故障模式。除了上面提到的两个步骤之外，日记文件系统具有提交对事务所做更改的第三个步骤。当一切正常运行时，处理器或者向日记日志提交所有事务或者不提交任何事务。在日记文件系统中，处理器可以重放日记日志以在一致的状态下建立元数据。但是，日记文件系统可以在 I/O 错误处理失败时变得不一致，如在写入失败但是处理器将事务的其他部分写入日记日志时所发生的。

涉及远程外包文件系统的维护和恢复的危害是传送的文件系统的详细信息可能被截取。通过互联网传送数据时尤其如此。一旦被截取，即使加密的数据也会泄漏。因此，长期存在的愿望是尽可能避免通过互联网传送敏感信息。

发明内容

本发明的各方面提供了用于生成文件系统元数据的计算机实现的方法、计算机可用代码和装置。处理器标识文件系统中的一个数据单元。所述处理器判定所述数据单元是否包含文件系统元数据。如果所述处理器判定所述数据单元包含文件系统元数据，则所述处理器将元数据的数据单元写入诊断文件。然后所述处理器判定所述数据单元是否包含用户数据。如果所述数据单元包含用户数据，则所述处理器将稀疏对象写入所述诊断文件。

附图说明

在所附权利要求中说明了被认为是本发明特性的新颖特征。但是，当结合附图阅读时，通过参考以下对本发明示例性实施例的详细说明，可以最佳地理解发明本身及其优选使用方式、进一步的目的和优点，这些附图是：

图 1 示出了其中可以实现本发明的实施例的数据处理系统的方块图；

图 2 示出了根据本发明的示例性实施例的客户系统如何与专家系统交互；

图 3 示出了根据本发明的示例性实施例的操作系统调解源文件系统到诊断文件的转换的操作；

图 4A 示出了根据本发明的示例性实施例的传统文件格式的实例；

图 4B 示出了根据本发明的示例性实施例的稀疏文件格式的实例；

图 5A 示出了根据本发明的示例性实施例的获得与文件系统有关的诊断数据的初始步骤；

图 5B 示出了根据本发明的示例性实施例的封装和传送与文件系统有关的诊断数据的步骤；

图 6 示出了根据本发明的示例性实施例的使用增强的文件一致性检查软件来获得诊断数据的详细步骤；以及

图 7 示出了根据本发明的示例性实施例的在 AIX 操作系统的命令行处

输入的命令的实例。

具体实施方式

现在参考附图，具体地说，参考图 1，图 1 示出了其中可以实现本发明的实施例的数据处理系统的方块图。在所示实例中，数据处理系统 100 使用集线器体系结构，所述集线器体系结构包括北桥和存储器控制器集线器 (MCH) 108 以及南桥和输入/输出 (I/O) 控制器集线器 (ICH) 110。处理单元 102、主存储器 104 和图形处理器 118 连接到北桥和存储器控制器集线器 (MCH) 108。图形处理器 118 可以通过加速图形端口 (AGP) 或图形处理器 118 连接到北桥和存储器控制器集线器 108。

在所示实例中，局域网 (LAN) 适配器 112、音频适配器 116、键盘和鼠标适配器 120、调制解调器 122、只读存储器 (ROM) 124、通用串行总线 (USB) 端口和其他通信端口 132 以及 PCI/PCIe 设备 134 通过总线 138 连接到南桥和 I/O 控制器集线器 110。PCI/PCIe 设备可以包括例如以太网适配器、插卡以及用于笔记本电脑的 PC 卡。PCI 使用卡总线控制器，而 PCIe 并不使用卡总线控制器。ROM 124 可以例如是闪速二进制输入/输出系统 (BIOS)。

硬盘驱动器 126 和 CD-ROM 驱动器 130 通过总线 140 连接到南桥和 I/O 控制器集线器 110。硬盘驱动器 126 和 CD-ROM 驱动器 130 可以例如使用集成驱动电子 (IDE) 或串行高级技术附件 (SATA) 接口。超级 I/O (SIO) 设备 136 可被连接到南桥和 I/O 控制器集线器 110。

操作系统在处理器 102 上运行并协调和控制图 1 中的数据处理系统 100 内的各种组件。操作系统可以是可从市场上购买的操作系统，如可从 IBM 公司购买的高级交互执行 (AIX®)。AIX 是 IBM 公司的注册商标。

本发明的各种实施例允许仅传送文件系统不一致的细节而无需传送麻烦的用户数据。用户数据通常包括文件系统数据的主要部分，因此不被发送（除非以保留位置的形式）。文件系统一致性检查程序标识文件系统多个数据单元。标识的一种方法是文件系统一致性检查程序通过从一个

数据单元移到另一个数据单元来标识文件系统的树状结构。某些实施例可以使用像块一样小的数据单元。其他实施例可以使用更大的数据单元（如扇区）。如果数据单元包含一个或多个文件系统元数据，则软件将该元数据单元写入或复制到可以是诊断文件系统的一部分的诊断文件。如果数据单元包含用户数据，则软件将稀疏对象写入或复制到诊断文件。以后，备份或以其他方式合并诊断文件的步骤将稀疏对象看作原始用户数据的精简版本，但完全为空且未被写入。稀疏对象可以是文件的元数据中的空指针。因为该指针未指向任何引用，所以没有存储基础数据—仅表现为空或全部为‘0’的数据单元。因此，此类稀疏对象在某种意义上作为位置保留符（place-keeper）进行操作，代替包括文件系统的盘或其他介质上的数据单元中的少量‘0’。因此，与源文件系统相比，处理系统可以以极大地简化的形式传送每个稀疏对象。

图2示出了根据本发明的示例性实施例的客户系统201如何与专家系统205交互。客户系统201可以使用例如图1的处理系统100来运行。管理员可以判定文件系统行为异常。作为响应，管理员可以运行客户系统201来生成包含密集诊断数据的密集诊断文件。

专家系统205可以从客户系统201接收诊断文件系统。专家系统205维护诊断文件系统以形成已维护的文件系统。专家系统205可以使用例如图1的处理系统100来运行。维护可以包括例如对诊断文件系统进行维护以形成已维护的文件系统。此外，维护可以包括对诊断文件系统执行恢复操作。计费系统207可以监视诊断文件系统上的活动。计费系统207可以监视专家系统205上的活动，以便计费系统207可以为在维护诊断文件系统时提供的维护对客户计费。专家系统205将已维护的文件系统返回客户系统201。

密集文件格式是压缩稀疏文件时产生的结果。管理员可以在命令行编辑器处输入命令，以便执行如下所述的减少用户数据的文件一致性检查。由于处理器可以将用户数据减少到甚至隐藏其中存储的内容的类型，所以生成的一个或多个文件是密集的并具有与文件系统错误的源更直接相关的

元数据。因此，术语密集诊断数据被应用于说明从本发明的各种示例性实施例产生的一个或多个精简的文件。换句话说，密集诊断数据是通过丢弃用户数据或使用位置保留符代替用户数据，然后压缩一个或多个结果文件来创建的。

客户系统 201 通过网络（例如，因特网 203）传送密集诊断数据。密集诊断数据到达专家系统 205。专家系统 205 可以在数据恢复专家的控制下。系统恢复专家可以没有与管理员建立信任关系。此外，因特网 203 被认为是不可靠的传送数据的方法。

图 3 示出了根据本发明的示例性实施例的操作系统调解源文件系统到诊断文件的转换的操作。操作系统 301 可以在处理器（例如，图 1 的处理器 102）上运行。源文件系统 303 可以例如存储在图 1 的硬盘驱动器 126 上。管理员可以使用命令行编辑器 311 将命令分别输入操作系统 301。一个或多个命令可以生成诊断文件 305。此外，实用程序可用于替代命令行编辑器 311，其中实用程序将命令写入操作系统。

图 4A 示出了传统文件格式的实例。分配给文件 411 的每个位被设置为‘0’，尽管文件指针使用查找命令忽略物理盘地址（例如，‘0’块 413）。在此格式中，处理器物理地写入文件的所有已分配的位和块，并且没有在物理级别使用压缩。

图 4B 示出了可以与诊断文件关联的稀疏文件格式（有时称为稀疏对象）的实例。此格式涉及处理器执行对每一位的逻辑写入，但实际上，处理器仅将数据物理地存储在文件指针指示的位置。文件指针是在该处发生下一次物理写入的位置。对于处理器允许在文件写入之间存在间隔的情况，将在元数据中做出记号（例如，块指针表 414）。处理器通过在逻辑地记录了‘0’块处建立空指针 415、416、417 和 418 来在对块指针表 414 做出记号。换句话说，每个空指针都作为否则将被物理写入硬盘驱动器的‘0’块的某种类型的占位符来操作。该格式是稀疏的，因为可以在后续读取时检索到大量‘0’。此外，文件元数据是保存对每个‘0’块的引用的位置，而不是将这些‘0’存储在用户数据中。大量的‘0’（或‘0’块）由稀疏

对象来表示，并且某些文件系统可以在操作系统的普通操作期间（例如，当程序要求操作系统进行读取时）从稀疏对象读取大量的‘0’。

密集文件格式是压缩稀疏文件时出现的结果。处理器可以显著压缩每个‘0’块。同样将压缩包括某种程度上随机顺序的‘1’和‘0’的数据，尽管可能不那么显著。因此，将‘0’块和某种程度上随机的数据一起压缩，以便以压缩形式物理地写入文件的所有部分。一旦完成压缩，作为诊断数据出现的一个或多个文件会成为密集诊断数据。

图 5A 示出了根据本发明的示例性实施例的获得与文件系统有关的诊断数据的初始步骤。管理员可以通过输入命令行以命令操作系统执行每个步骤来指示处理器执行这些步骤。同样适用的是将所有步骤合并为按顺序自动运行的命令脚本。例如，操作系统可以是图 3 的操作系统 301。处理器接收命令以创建诊断文件系统（步骤 501）。诊断文件系统可以与图 3 的诊断文件 305 相关联。在类似 Unix 的系统中，可能需要命令来使操作系统可以访问最新创建的文件系统。在当前的示例性实施例中，该命令为‘安装’。处理器接收安装命令以安装诊断文件系统（步骤 503）。诊断文件系统现在已准备就绪以便收集数据。处理器接收命令以将元数据提取到诊断文件（步骤 505）。此命令可以例如是文件系统一致性检查（fsck）。诊断文件可以包括稀疏对象。诊断文件可以通过成为诊断文件系统的组件来与诊断文件系统关联。

Fsck 生成大量数据，所述数据可能完全占用处理器创建诊断文件系统时分配的存储。因此，处理器进行测试以查看诊断文件系统是否可以容纳提取的元数据（步骤 507）。如果处理器做出否定的判定，则处理器可以接收命令以创建更大的诊断文件系统（步骤 509）。处理在步骤 509 后可以安装诊断文件系统和其他步骤。

如果处理器判定诊断文件系统可以容纳提取的元数据，则处理通过连接符‘A’继续到图 5B。

存在创建文件系统和安装文件系统的替代操作。例如，可以存在用于记录诊断数据的专用文件系统，并且可以在将密集诊断数据写入该专用文

件系统的操作之前将其删除。

恢复数据处理系统的一种方法是在处理器接着安装文件系统时完整地通过或遍历文件系统的数据结构。遍历以及其他步骤可以检测和更正任何不一致性。用于执行此修复的一种工具是文件一致性检查或 fsck 命令。其他文件一致性检查程序包括 Microsoft 'chkdsk' (也称为“检查磁盘”) 程序。计算机管理员在采用具有文件系统的操作系统的计算机中使用 fsck。此类操作系统包括例如 AIX 和 Linux, 它们在操作上与 Unix 类似。Linux 是 Linus Torvalds 的注册商标。UNIX 是 Open Group 的注册商标。其他操作系统类包括 Microsoft™ 公司的各代 Windows™ 操作系统, 以及 Apple™ 公司的 Mac 操作系统。文件系统的实例包括诸如 IBM 的增强的日记文件系统或 JFS2 之类的日记文件系统。

类似 Unix 的操作系统中的文件系统提供了存储冗余数据的便利方法, 其在某些方面表现为似乎存储是为所述冗余对象分配的。所述方法包括创建至少合适大小的块的稀疏对象。当处理系统第一次打开文件时, 处理器可以创建指向文件起始位置的文件指针。文件指针用作位置保留符以指示处理器接着应写入数据的位置, 其中处理器将文件视为线性的。换句话说, 下一次处理器执行写入命令时, 处理器将在文件指针所在的位置进行写入。如果处理器在没有写入命令的情况下前移文件指针, 则存在由文件系统为文件保留的空间, 但是没有相应地将块分配给文件。相反, 文件系统元数据以几个字节的形式将扩展的空头的存在存储到文件, 但是处理器并不将任何数据单元标记为已用。数据单元可以例如是块。如果按此方法将许多块标记为已用, 则会出现实际已用磁盘空间的大量节约, 因为数个字节可以表示许多块。将文件指针向前移动和向其他方向移动的通用方法是使用 lseek 系统调用, 此调用一般存在于类似 Unix 的系统上。其他系统上存在其他的文件指针移动命令。

文件一致性检查要查找的事项之一是数据结构中的不一致。文件一致性检查软件通过导航文件系统的结构来查找不一致。文件系统的结构包括具有根和一个或多个来自所述根的分支的树。例如, 目录结构开

始于根，并且可以在根下拥有一个或多个目录。类似地，每个文件使其组成部分存储在同样可以具有多个分支和层的块中，其中块是在每个分支点处的节点。包括指针或对其他块的引用的块将此类指针存储为元数据。

文件一致性检查软件通过从数据单元遍历到数据单元来导航树，其中每个数据单元都可以是块。遍历意味着文件一致性检查软件检查块下的其他分支的存在，并且如果存在一些分支，则软件依次检查每个分支，直到穷尽所有分支为止。检查意味着许多事项。通常，检查意味着处理器沿其他分支查找其他引用或指针。

在一致的文件系统中，每个数据单元仅具有单个对数据单元的引用，这意味着处理器将仅遍历数据单元一次。再次地，数据单元可以例如是块。但是，在不一致的文件系统中，可以多次引用块。软件可以通过遍历来标识文件的各单元。例如，文件一致性检查软件可以遍历第一个文件并且标记块被分配给第一个文件。所述软件可以遍历第二个文件，并且标记同一个块也被分配给第二个文件。实质上，文件系统偶然将同一个块分配了两次，并且可能使用第二个文件的数据覆盖了第一个文件的数据。通过遍历所有文件，文件一致性检查可以发现许多问题。

文件一致性检查可以修复问题以便以问答的形式响应与用户的交互。但是，此功能可以如此复杂和专业，以致文件系统的管理员可以请求来自专家的外来帮助以在最大程度上恢复数据。遗憾的是，尽管许多专家都可以胜任，但是在文件系统管理员与专家之间可能没有建立信任。换句话说，需要一种机制来保护文件系统用户数据免于泄漏。

软件开发人员可以将代码添加到 fsck 和其他文件一致性检查器中。所述附加的代码在遍历每个文件的已分配的块时采取额外的步骤。除了标记不一致性之外，fsck 可以写入每个文件的某些方面的副本。在以上图 5A 的步骤 505 中执行遍历和写入的过程。所述副本可能完全占用处理器创建诊断文件系统时分配的存储。

图 5B 示出了根据本发明的示例性实施例的封装和传送与文件或文件系统有关的诊断数据的步骤。操作系统（如图 3 的操作系统 301）可以执

行图 5B 中的步骤。替代地，通过实用程序可以实现图 5B 中的步骤，其中实用程序将命令写入操作系统。通常，图 5B 中的步骤完成创建更小并更适于通过网络或其他装置进行传送的数据结构的常规功能。初始地，操作系统备份或以其他方式将诊断文件系统合并到单个文件中（步骤 511）。操作系统然后可以压缩所述文件（步骤 513）。当操作系统进行压缩时，这意味着处理器利用重复和冗余来概括数据并存储已缩短的概要。例如，不同于示出所有 4096 个 ‘0’，描述 4096 个 ‘0’ 的块的另一种方法是只电子地存储 4096x0，这要短得多。此外，由于元数据中可能已存在空指针，所以只存储空指针如同压缩那样工作。甚至可以压缩较大序列的空指针。例如，描述 1000 个空指针的序列（而不是物理地存储每个空指针）的另一种方法是电子地存储 1000 x null。操作系统然后传送已压缩的文件或密集诊断数据（步骤 515）。因此，操作系统将用户数据转换为密集诊断数据。估计具有许多小文件的 1024 千兆字节增强的日记文件系统（JFS2）文件系统可被转换为大约 160 兆字节的密集诊断数据 - 6,000 倍的压缩。

图 6 示出了使用增强的文件一致性检查软件来获得诊断数据的详细步骤。增强的文件一致性检查可以是某一版本的 fsck。Fsck 遍历文件系统中的每个数据单元（步骤 621）。运行 fsck 的处理器可以判定数据单元是否包含文件系统元数据（步骤 623）。“包含”意味着数据单元包括组成文件系统的所有元数据的至少一部分。Fsck 写入元数据单元以响应判定所述数据单元是文件系统元数据（步骤 625）。否则，步骤 623 和 625 继续到步骤 627。Fsck 判定所述数据单元是否包含用户数据（步骤 627）。Fsck 将稀疏对象写入诊断文件以响应判定所述数据单元是用户数据（步骤 629）。写入稀疏对象可以包括处理器在元数据中写入空指针。空指针可以例如是图 4B 的块指针表 414 的空 415。Fsck 判定是否存在要遍历的其他数据单元（步骤 631）。步骤 625 和 629 的每次写入都可以是到诊断文件，其中所述诊断文件可以与图 3 的诊断文件系统 305 关联。稀疏对象与非稀疏对象的不同之处在于到盘的写入并不更改数据，除了标记块为空或者被无效的数据填充以外。Fsck 并不占用整个块，而是只是生成与元数据关联

的所有稀疏对象的列表。替代地，fsck只是使用lseek系统调用来向前移动为写入而打开的文件的文件位置。Lseek系统调用是存在于类似Unix的操作系统上的文件处理系统调用。实质上，lseek命令告知操作系统将文件位置移过文件的当前结尾。如果lseek系统调用将文件位置向前移过整个块，则操作系统记录该块是稀疏块或稀疏对象，而不分配盘驱动器上的存储的物理块。稀疏对象是虚拟的存储块，但是不占用物理存储的完整块的空间。相反，元数据则标识与文件关联的此类块的数量。尽管如此，在列出目录期间确定文件占用的总空间时，处理器将此类稀疏对象添加到总计。

图7示出了根据本发明的示例性实施例的在AIX操作系统的命令行处输入的命令的实例。尖括号<>括起来的文本表示用户指定的串。例如，<new filesystem size>711可以是指定要分配给新文件系统的512字节块数的数目。

第一个语句使用crfs命令701来创建文件系统。标志如下：

-v jfs2 703 指定虚拟文件系统类型；

-g rootvg 505 指定在其上生成文件系统的现有卷组。卷组是一个或多个物理卷的集合；

-m/newfs 707 指定安装点，所述安装点是将在该处使文件系统可用的目录；以及

-a size=<new filesystem size>709 以512字节块为单位指定JFS2的大小，其中，如果指定的大小不能被物理分区大小整除，则处理器将四舍五入到最接近整除的数。

诊断文件系统可以是图3的诊断文件系统305。

Mount/newfs 721 指示处理器安装文件系统。

接着是文件一致性检查并使用fsck 722命令，所述命令包含下列标志：

-n 723 指示处理器以非交互模式执行文件一致性检查，即，不询问用户是否要修复，并且不更改已指定的文件系统；

-o metacapture=/newfs/out1 725 标识指示处理器将元数据的副本和

稀疏对象写入具有稀疏对象的文件的参数，其中文件位于/newfs/out1 或其他用户指定的目录中；

<file system mount point>735 是处理器要执行诊断的文件系统的标识。

处理器基于其余两行进行合并。备份 739 指示处理器将所有在 fsck 722 中创建的文件置于单个文件中。即，将诊断文件系统置于单个文件中。备份 739 包括下列标志：

-f<destination filename>741 标识一个或多个要存储输出的设备；
以及

-o /newfs 743 指定用于备份的源文件系统。

压缩 751 是使用压缩标记替换稀疏对象的步骤。压缩 751 获取的唯一参数是在该处存储密集诊断数据的文件<destination filename>。

可以理解，合并还可以通过获得由文件一致性检查提供的单个文件并压缩每个文件来实现。然后每个文件变成密集诊断数据，并且添加到文件的主体的每个后续文件都将添加到密集诊断数据。

因此，本发明的各方面提供了计算机实现的方法、装置以及计算机可用程序代码，以便获取有关文件系统不一致性的详细信息，其方式为用户数据是安全的，即使对指派修复文件系统不一致性的专家也是如此。此外，本发明的各方面可以允许文件系统开发者诊断正在开发的软件中的问题，其中正在开发的软件可能促成了文件系统不一致性。此外，可以将稀疏对象发送到服务组，所述服务组可以将所述稀疏对象提取到逻辑卷中，或者直接查看所述稀疏对象以调试文件系统问题。

本发明可以采取完全硬件实施例、完全软件实施例或同时包含硬件和软件元素的实施例的形式。在优选实施例中，本发明以软件来实现，所述软件包括但不限于固件、驻留软件、微编码等。

此外，本发明可以采取计算机程序产品的形式，所述计算机程序产品可从提供程序代码以供或结合计算机或任何指令执行系统使用的计算机可用或计算机可读介质来访问。出于此描述的目的，计算机可用或计算机可

读介质可以是任何可以包含、存储、传送、传播或传输程序以供或结合指令执行系统、装置或设备使用的装置。

所述介质可以是电、磁、光、电磁、红外线或半导体系统（或装置或设备）或传播介质。计算机可读介质的实例包括半导体或固态存储器、磁带、可移动计算机盘、随机存取存储器（RAM）、只读存储器（ROM）、硬磁盘和光盘。光盘的当前实例包括光盘-只读存储器（CD-ROM）、光盘-读/写（CD-R/W）和 DVD。

适合于存储和/或执行程序代码的数据处理系统将包括至少一个通过系统总线直接或间接地连接到存储器部件的处理器。所述存储器部件可以包括在程序代码的实际执行期间使用的本地存储器、大容量存储装置以及对至少某些程序代码提供临时存储以便减少在执行期间必须从大容量存储装置检索代码的次数的快速缓冲存储器。

输入/输出或 I/O 设备（包括但不限于键盘、显示装置、指点设备等）可以直接或通过中间 I/O 控制器与所述系统相连。

网络适配器也可以与所述系统相连以便使所述数据处理系统能够通过中间专用或公共网络与其他数据处理系统或远程打印机或存储设备相连。调制解调器、电缆调制解调器以及以太网卡只是少数几种当前可用的网络适配器类型。

出于示例和说明目的给出了对本发明的描述，并且所述描述并非旨在穷举的或是将本发明限于所公开的形式。对于本领域的技术人员来说，许多修改和变化都将是显而易见的。本发明的示例性实施例的选择和描述是为了最佳地解释本发明的原理、实际应用，并且当适合于所构想的特定使用时，使得本领域的其他技术人员能够理解本发明的具有各种修改的各种实施例。

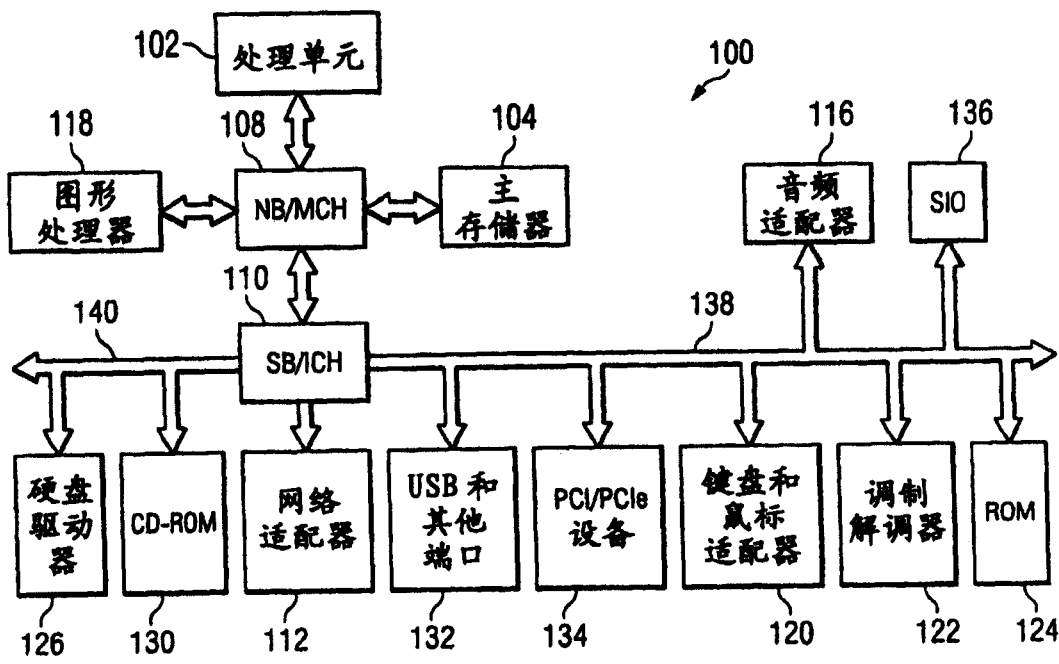


图 1

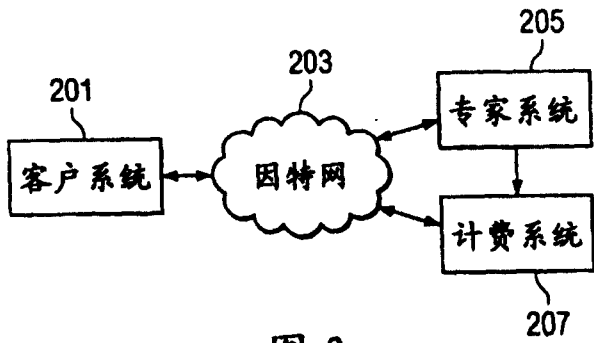


图 2

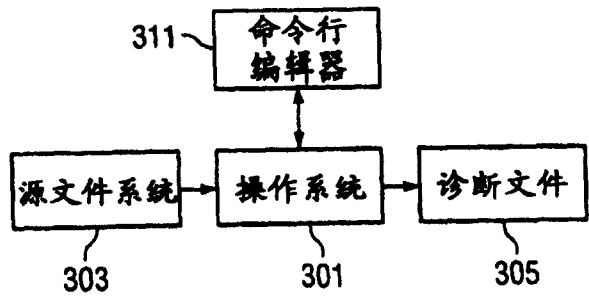


图 3

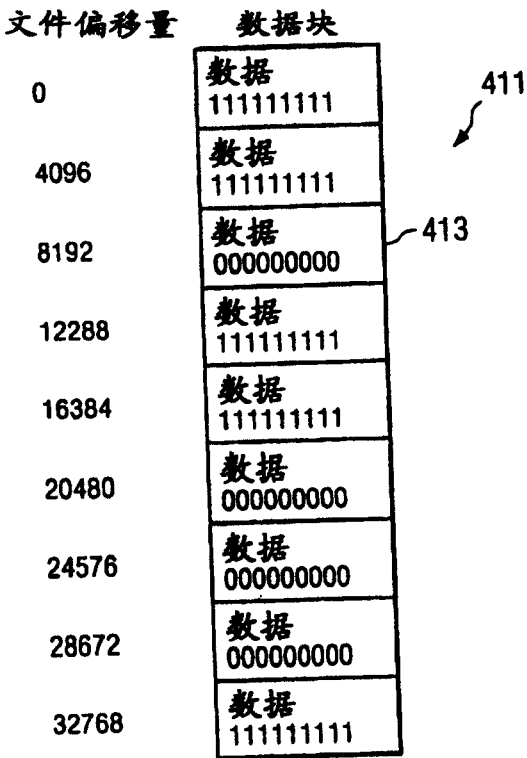


图 4 A

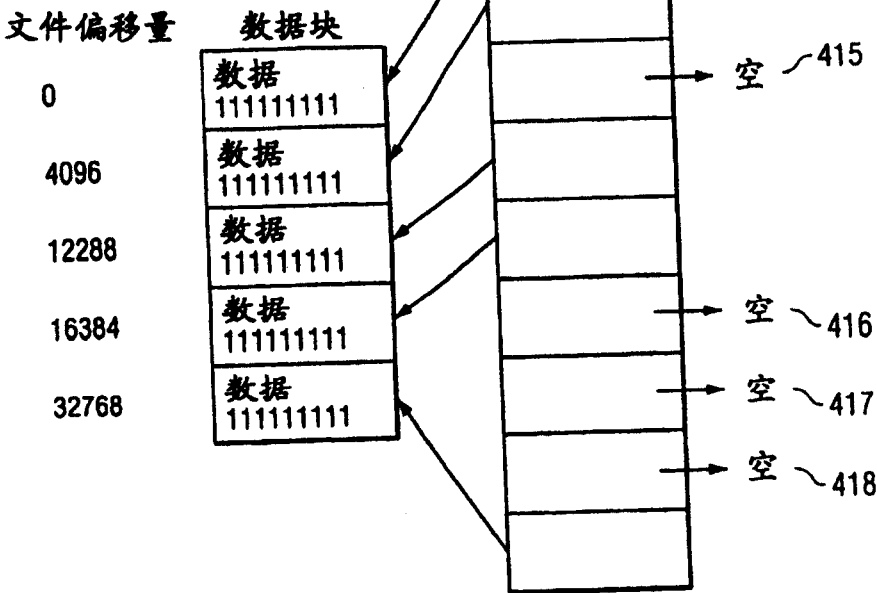


图 4 B

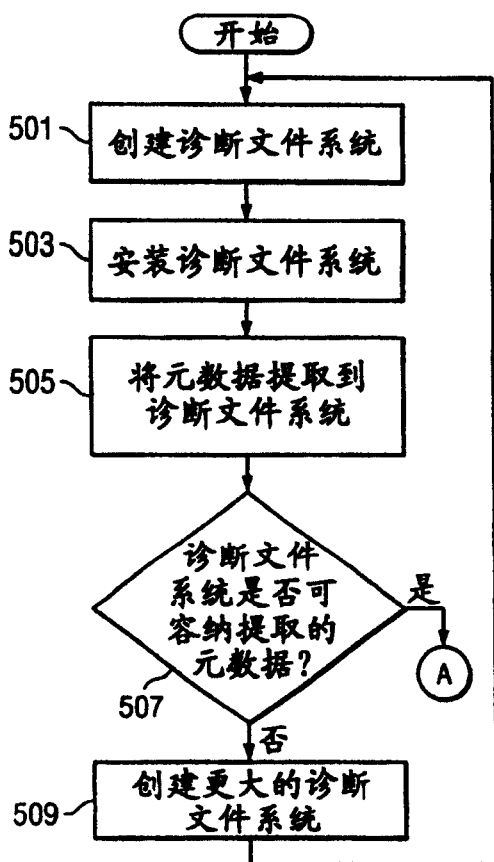


图 5 A

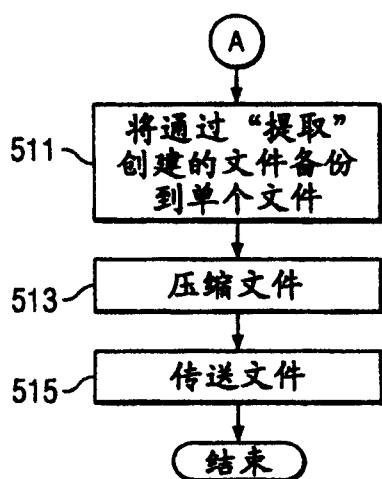


图 5 B

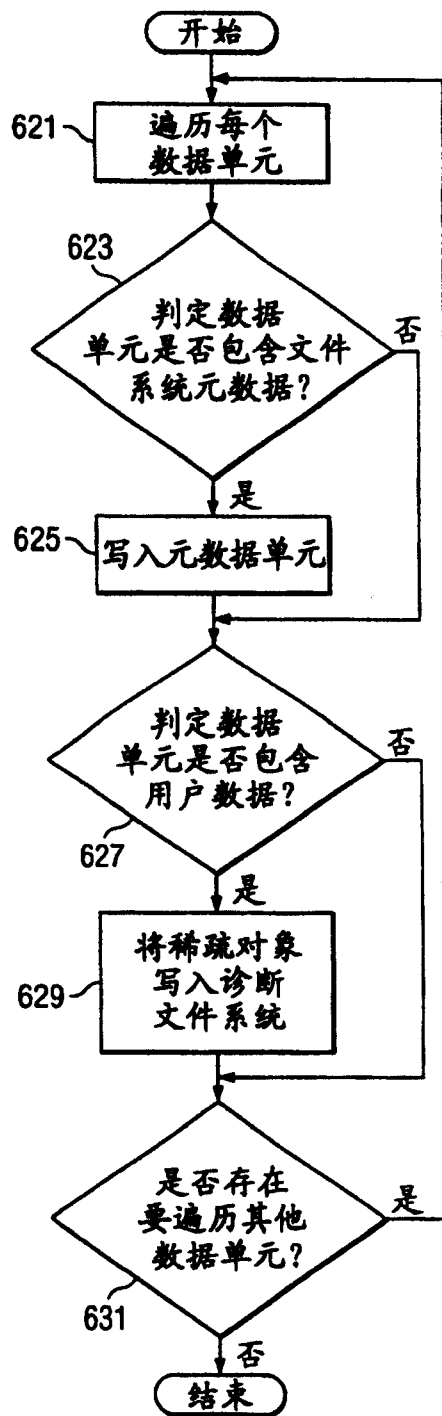


图 6

701 703 705 707 709
\$>crfs -v jfs2 -g rootvg -m /newfs -a size=<new filesystem size>
711

721
\$>mount /newfs

722 725 735
\$>fsck -n -o metacapture=/newfs/out1 <file system mount point>
723

741 743
\$>backup -f <destination filename> -O /newfs
739

751
\$>compress <destination filename>

图 7