

[19]中华人民共和国国家知识产权局

[51]Int. Cl⁶

G06F 13/00

G06F 15/16

[12] 发明专利申请公开说明书

[21] 申请号 97197011.4

[43]公开日 1999年9月1日

[11]公开号 CN 1227644A

[22]申请日 97.8.1 [21]申请号 97197011.4

[30]优先权

[32]96.8.2 [33]US [31]60/023,218

[86]国际申请 PCT/US97/12843 97.8.1

[87]国际公布 WO98/06036 英 98.2.12

[85]进入国家阶段日期 99.2.9

[71]申请人 通讯软件有限公司

地址 美国加利福尼亚州

[72]发明人 J·J·沃尔夫

D·莱思罗普

[74]专利代理机构 上海专利商标事务所

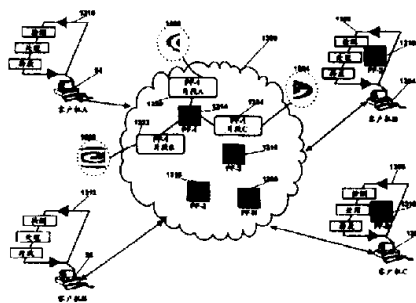
代理人 张政权

权利要求书 10 页 说明书 44 页 附图页数 46 页

[54]发明名称 允许对共享资源分布式控制的方法和装置

[57]摘要

本系统包含一个控制表(206,208)和控制过程。该控制表包括用于指示控制表(206,208)是否可供使用的可用性指示以及数据卷拥有权的指示。控制过程则被在每个客户机(54,56)上执行用于请求和获得关于控制表(206,208)的独占拥有权以对该客户机提供对该数据卷的独占写访问。客户机相互间仲裁对存储卷(64,66)或项目管理文件夹的独占编辑访问。该项目管理文件夹包括一个或多个记录,每个记录包括了源(1458)文件名和目标(1478)文件名以及应用程序代码(1474)。每个客户机执行一个管理过程(1306,1312)以定位项目管理文件夹(1314,1320)并利用它们的驻留应用对处理客户机上发现的既未完成又要求应用的那些记录进行操作。



ISSN 1008-4274



权 利 要 求 书

1. 一种用于管理数据存储卷和对与该数据存储卷相连的多个客户机提供独占写访问的卷管理系统，其特征在于，该卷管理系统包括：

与所述数据存储卷相关联的一张控制表，所述控制表包括：

a)用于指示该控制表是否可供使用的可用性指示；

b)用于指示对该数据存储卷的拥有权的识别指示；以及

多个控制过程，每一个在所述多个客户机之一上执行，所述多个控制过程请求并获得关于所述控制表的独占拥有权以向所述多个客户机的每一个提供对该数据存储卷的独占写访问。

2. 如权利要求 1 所述的卷管理系统，其特征在于，所述控制表进一步包括：

c)用于指示所述多个客户机的每一个对所述数据存储卷的访问特权的安全性指示；以及

所述多个控制过程进一步包含：

用于拒绝该多个客户机之一对该数据存储卷的写访问的过程，该一个客户机在所述控制表中列出的安全性指示具有对该数据存储卷的只读访问特权。

3. 如权利要求 1 所述的卷管理系统，其特征在于，所述多个控制过程进一步包括：

用于更新位于具有只读访问特权的多个客户机的每一个客户机上的与该数据存储卷对应的一个文件目录表的过程。

4. 如权利要求 2 所述的卷管理系统，其特征在于，所述多个控制过程包括：

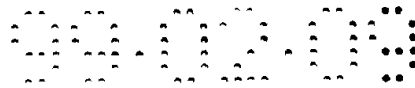
用于更新位于具有只读访问特权的多个客户机的每一个客户机上的与该数据存储卷对应的一个文件目录表的过程。

5. 如权利要求 1 所述的卷管理系统，其特征在于，所述多个控制过程进一步包括：

用于在对所述数据存储卷具有独占写访问的多个客户机之一的监视器上显示一条释放独占写状态请求的过程；以及

用于从该多个客户机之一的输入设备接收对该释放请求的肯定或否定响应的过程。

6. 如权利要求 1 所述的卷管理系统，其特征在于，所述多个控制过程进一步



包括:

用于在客户机的监视器上显示该多个客户机的每一个关于该数据存储卷的状态的过程, 该状态包括: 只读和读/写。

7. 如权利要求 1 所述的卷管理系统, 其特征在于, 所述多个控制过程进一步包括:

用于在该多个客户机的每一个初始加电期间对把该多个客户机的每一个与该数据存储卷相连的多个数据存储卷驱动器的每一个锁定的过程; 以及

用于在启动请求和获取过程时将该多个数据存储卷驱动器的每一个解锁的过程。

8. 如权利要求 1 所述的卷管理系统, 其特征在于, 所述多个控制过程进一步包括:

用于接收所述多个客户机的第一个至所述多个客户机的第二个的消息及把该消息写入所述控制表的过程;

用于确定在所述控制表中一则消息存在以及在所述多个客户机的第二个的监视器上显示该消息的过程。

9. 计算机系统中一种用于与一数据存储卷相连的多个客户机之间实行对该数据存储卷的互斥写访问的方法, 其特征在于, 所述实行互斥写访问的方法包括如下动作:

创建一个与所述多个客户机的每一个相连的控制表, 用于存储控制表可用性指示和对该数据存储卷的拥有权指示;

在所述多个客户机的第一个处检测写请求,

响应于所述检测动作, 在所述多个客户机上的第一个启动 - 请求例程, 其中, 所述请求例程包含如下动作:

a) 基于对该数据存储卷的拥有权指示确定所述多个客户机的另一个对所述数据存储卷有写访问; 以及

b) 响应于所述确定动作, 向其它所述多个客户机发送一释放请求以释放写访问;

c) 对来自所述其它多个客户机的肯定通知作出响应;

i) 用第一客户机的拥有权指示替换该拥有权指示;

ii) 安装所述数据存储卷用于写访问; 以及



在所述第二所述多个客户机处获得所述释放请求;

响应于所述获得动作启动在所述多个客户机的第二个上的释放例程, 其特征在于, 所述释放例程包含下述动作:

a) 释放对所述数据存储卷的写访问; 以及

b) 向所述第一客户机发送关于所述释放动作的肯定通知。

10. 如权利要求 9 所述的方法, 其特征在于, 所述请求例程和所述释放例程可以起源于多个不同的文件系统协议。

11. 如权利要求 9 所述的方法, 其特征在于, 所述创建动作进一步包含下述动作:

提供在所述控制表中存储安全性指示用于为多个客户机中的每一个指示对数据存储卷的访问特权; 以及

所述检测写请求动作进一步包含下述动作:

拒绝在所述控制表中具有安全性指示指明对该数据存储卷为只读访问特权的多个所述客户机之一提出的对该数据存储卷的写请求。

12. 如权利要求 9 所述的方法, 其特征在于, 所述用于实行互斥写访问的方法进一步包括下述动作:

当对所述数据存储卷文件目录作出改变时更新位于对该数据存储卷具有只读访问特权的多个客户机之一上的数据存储文件的副本。

13. 如权利要求 9 所述的方法, 其特征在于, 所述用于实行互斥写访问的方法进一步包含下述动作:

在对所述存储卷具有独占写访问的多个客户机之一的监视器上显示一个释放独占写状态的请求; 以及

从多个客户机之一的输入设备接收对该释放请求的肯定和否定响应。

14. 如权利要求 9 所述的方法, 其特征在于, 所述用于实行互斥写访问的方法进一步包含如下动作:

在该多个客户机之一的显示器上显示多个客户机的每一个关于该数据存储卷的状态, 该状态包括: 只读, 读/写。

15. 如权利要求 9 所述的方法, 其特征在于, 所述用于实行互斥写访问的方法进一步包含下述动作:

在多个客户机的每一个的初始加电期间锁定把该多个客户机中的每一个与



该数据存储卷相连的多个数据存储卷驱动器的每一个；以及

在启动该请求和获得过程时解锁该多个存储卷驱动器的每一个。

16. 如权利要求 9 所述的方法，其特征在于，所述用于实行互斥写访问的方法进一步包括下述动作：

接收从所述多个客户机的第一个到所述多个客户机的第二个的消息以及把该消息写入所述控制表；

确定在所述控制表中存在一则消息；以及

在所述多个客户机的第二个的监视器上显示该消息。

17. 一种计算机可用的媒体，其中包含了计算机可读程序代码装置用于在与该数据存储卷相连的多个客户机之间实行对该数据存储卷的互斥写访问，其特征在于，在所述制造商成品中该计算机可读程序代码装置包括：

计算机可读程序代码装置，用于创建一个与多个所述客户机中每一个相连的控制表用来存储控制表可用性指示和该数据存储卷的拥有权指示；

计算机可读程序代码装置，用于使计算机在所述多个客户机的第一个处检测写请求；

计算机可读程序代码装置，用于对所述写请求检测的响应使计算机启动在所述多个客户机的第一个上的请求例程，其中，所述请求例程包含下述动作：

a) 基于对该数据存储卷的拥有权指示确定所述多个客户机的另一个对所述数据存储卷有写访问；以及

b) 响应于所述确定动作，向所述其它多个客户机发送释放一请求以释放写访问；

c) 对来自所述其它多个客户机的肯定通知作出响应；

i) 用第一客户机的拥有权指示替换该拥有权指示；

ii) 安装所述数据存储卷用于写访问；以及

计算机可读程序代码装置，用于使计算机在所述多个客户机的第二个处获得所述释放请求；

计算机可读程序代码装置，用于使计算机响应于所述释放请求的所述获得启动在所述多个客户机的第二个上的释放例程，其中，所述释放例程包含如下动作：

a) 释放对所述数据存储卷的写访问，以及

b)向所述第一客户机发送关于所述释放动作的肯定通知。

17'. 一种分布式数据库管理系统，用于管理具有多个可访问记录的数据库，以及用于向连接到该数据库的多个用户机提供对每个记录中的每一个字段的独占编辑访问，其特征在于，该数据库管理系统包括：

多个锁定字段，每一个与多个可访问记录的字段中对应的一个相关联；

多个相关性过程，每一个在多个客户机中对应的一个上被执行，所述多个相关性过程用于接收编辑字段的请求，以及用于通过对与该被编辑字段有关的所述多个锁定字段之一发出独占拥有权，在对该字段不存在现有字段锁定时授予该请求对该字段编辑，以及用于在收到退出编辑请求 后对该编辑字段有关的所述多个锁定字段之一解除要求独占拥有权；以及

多个数据库处理过程，每一个在所述多个客户机之一上被执行，所述多个数据库用于接收查询，用于对该查询响应以数据库提供记录，用于接收请求对记录中的字段编辑以及用于将该编辑字段的请求传递到相应的相关性过程以及用于接收退出编辑该字段的请求并把该退出编辑该字段的请求传递到相应的相关性过程。

18. 如权利要求 17 所述的分布式数据库管理系统，其特征在于，所述多个数据库处理过程进一步包括：

用于接收记录删除命令、用于擦除记录字段以及用于把所述记录用“自由”标识符标记的过程。

19. 如权利要求 18 所述的分布式数据库管理系统，其特征在于，所述多个数据库处理过程进一步包括：

用于通过定位具有“自由的”指示符的记录及把所述指示符改为“活动的”而添加一条记录的过程。

20. 如权利要求 17 所述的分布式数据库管理系统，其特征在于，所述多个数据库处理过程进一步包括：

用于保持一个被该多个客户机中每一个观看的该记录列表的过程。

21. 如权利要求 20 所述的分布式数据库管理系统，其特征在于，所述多个数据库处理过程进一步包括：

用于响应退出对第一记录的第一字段编辑的请求的过程，通过向与该列表中第一记录有关的那些客户机广播一则退出编辑通知而完成。



22. 如权利要求 17 所述的分布式数据库管理系统, 其特征在于, 所述多个数据库处理过程进一步包括:

用于通过接收来自退该编辑字段的客户机的广播而确定何时移去现存的字段锁的过程。

23. 如权利要求 17 所述的分布式数据库管理系统, 其特征在于, 所述多个数据库处理过程进一步包括:

用于通过抓取对该数据库的独占写访问, 添加记录以及释放该独占写访问而添加一条记录的过程。

24. 如权利要求 17 所述的分布式数据库管理系统, 其特征在于, 所述多个数据库处理系统进一步包括:

锁定字段, 它们是数据库的一部分。

25. 计算机系统中用于管理具有多个可访问记录的数据库, 以及用于向与该数据库相连的多个客户机提供对每个记录中的每个字段的独占编辑访问的一种方法, 其特征在于, 该用于管理数据库的方法包括下述动作:

在所述多个客户机的第一个处删除编辑一字段请求,

响应于所述删除动作, 启动在该所述多个客户机的第一个上的相关性例程, 其特征在于, 所述相关性例程包含下述动作:

a) 基于与该字段有关的拥有权指示确定所述多个客户机的另一个对该字段有编辑访问; 以及

b) 响应于所述确定动作, 等待与该字段有关的拥有权指示被解除要求;

c) 对与该字段有关的拥有权指示被解除要求作出响应;

i) 把第一客户机拥有权指示附着到要被编辑的字段; 以及

ii) 提供对该字段的编辑;

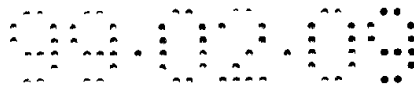
获得退出在第一所述客户机处编辑字段的请求,

响应于所述获得动作, 启动在该第一所述多个客户机上的相关性例程, 其中, 所述相关性例程包含下述动作:

a) 从该字段移去该第一客户机拥有权指示。

26. 如权利要求 25 所述的方法, 其特征在于, 用于管理该数据库的该方法进一步包含下述动作:

接收记录删除命令并对此作出响应:



a)擦除该记录的字段；以及

b)用“自由的”指示符标记该记录。

27. 如权利要求 25 所述的方法，其特征在于，用于管理该数据库的该方法进一步包括下述动作：

接收记录添加命令并对此作出响应；

a)定位具有“自由的”指示符的记录；以及

b)把所述“自由的”指示符改为“活动的”指示符。

28. 如权利要求 25 所述的方法，其特征在于，用于管理该数据库的该方法进一步包括下述动作：

观看记录并对此作出响应；

a) 更新被该多个客户机的每一个观看的记录的列表。

29. 如权利要求 28 所述的方法，其特征在于，用于管理该数据库的方法进一步包括下述动作：

通过向与该列表中第一记录有关的那些客户机广播一则退出编辑通知而对退出对第一记录的第一字段编辑的请求作出响应。

30. 如权利要求 25 所述的方法，其特征在于，用于管理该数据库的方法进一步包括下述动作：

通过接收来自退出该编辑字段的客户机的广播而确定何时移去现存的字段锁。

31. 如权利要求 25 所述的方法，其特征在于，用于管理该数据库的方法进一步包括如下动作：

通过抓取对该数据库的独占写访问，添加记录以及释放该独占写访问而把一个记录添加到该数据库。

32. 如权利要求 25 所述的方法，其特征在于，用于管理该数据的该方法进一步包括下述动作：

在所述数据库中创建所述锁定字段。

33. 一种分布式项目管理系统，用于在连有多个客户机的网络上管理项目，其特征在于，该分布式项目管理系统包含：

至少一个存储在该网络上的项目管理文件夹，且所述项目管理文件夹包含至少一个记录而所述至少一个记录包括：源文件名，源文件位置，第一应用名，第

一应用程序代码，第一应用目标文件名，和第一应用目标文件位置；

多个管理过程，每一个在所述多个客户机的相应的一个上执行，所述多个项目管理过程用于读取所述项目管理文件夹中的至少一条记录和用于使与所述第一应用名对应的驻留应用去读取所述源文件，按照所述第一应用程序代码实行对所述源文件的操作，以及把所得的工作成果用目标文件名以文件形式存储在该目标文件位置上。

34. 如权利要求 33 所述的分布式项目管理系统，其特征在于，在目标文件位置上存储该所得工作成果的所述多个管理过程进一步包括：

用于在该至少一个项目管理文件夹中标记该至少一个记录已完成。

35. 如权利要求 33 所述的分布式项目管理系统，其特征在于，所述多个项目管理过程进一步包括：

用于在该多个客户机之间共享用于该多个客户机中每一个的利用率指示符的过程。

36. 如权利要求 35 所述的分布式项目管理系统，其特征在于，所述多个项目管理过程进一步包括：

根据所述利用率指示符把该项目管理文件夹分配给所述多个客户机之一的过程。

37. 如权利要求 33 所述的分布式项目管理系统，其特征在于，所述多个项目管理过程进一步包括：

用于管理多个项目管理文件夹的过程。

38. 如权利要求 37 所述的分布式项目管理系统，其特征在于，所述用于管理多个项目管理文件夹的过程进一步包括：

用于把处理优先权指定给所述多个项目管理文件夹的过程。

39. 如权利要求 33 所述的分布式项目管理系统，其特征在于，所述用于管理多个项目管理文件夹的过程进一步包括：

用于将项目管理文件夹的管理分派给该多个客户机中被选中的几个的过程。

40. 如权利要求 39 所述的分布式项目管理系统，其特征在于，该多个客户机中被选中的几个以对等关系对所述项目管理文件夹操作。

41. 如权利要求 39 所述的分布式项目管理系统，其特征在于，该多个客户机

中被选中的几个以层次结构关系对所述项目管理文件夹操作。

42. 一种用于在连有多个客户机的网络上管理项目的方法，其特征在于，该用于管理项目的方法包括下述动作：

在计算机网络上创建一个项目管理文件且所述项目管理文件夹包含至少一个记录而所述至少一个记录包括：源文件名、源文件位置、第一应用名，第一应用程序代码、第一应用目标文件名和第一应用目标文件位置；

在所述多个客户机的第一个处检测有关项目管理文件在网络上的位置;

响应于所述检测动作，启动在所述多个客户机的第一个上的管理过程，其中，该管理过程包括如下动作：

a)在所述项目管理文件中读取所述至少一个记录;

b) 确定所述第一应用名与所述多个客户机的所述一个上对应的应用相匹配;

c)将所述第一应用程序代码传递给所述对应的应用，使所述对应的应用按照所述第一应用程序代码对所述源文件进行操作并将所得工作成果用该目标文件名以文件形式存储到该目标文件位置上。

43. 如权利要求 42 所述的用于管理项目的方法, 其特征在于, 该传递动作进一步包含下述动作:

在该至少一个项目管理文件夹中标记该至少一个记录已完成。

44. 如权利要求 42 所述的用于管理项目的方法, 其特征在于, 进一步包括下述动作:

在该多个客户机之间共享用于该多个客户机中每一个的利用率指示符,

其中，该传递动作进一步包括下述动作：

45. 如权利要求 42 所述的用于管理项目的方法, 其特征在于, 进一步包括下述动作:

根据所述利用率指示符分配该项目管理文件夹给所述多个客户机之一。

46. 如权利要求 42 所述的用于管理项目的方法, 其特征在于, 进一步包括如下动作:

管理多个项目管理文件夹。

47. 如权利要求 46 所述的用于管理项目的方法, 其特征在于, 进一步包括如下动作:

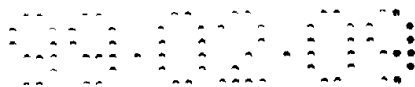
指定一个处理优先权给所述多个项目管理文件夹。

48. 如权利要求 42 所述的用于管理项目的方法，其特征在于，进一步包括如下动作：

把项目管理文件夹的管理分派给该多个客户机中被选中的几个。

49. 如权利要求 48 所述的用于管理项目的方法，其特征在于，该多个客户机中被选中的几个以对等关系对所述项目管理文件操作。

50. 如权利要求 48 所述的用于管理项目的方法，其特征在于，该多个客户机中被选中的几个以层次结构关系对该项目管理文件夹操作。



说明书

允许对共享资源分布式控制的方法和装置

有关临时申请的交叉引用

本申请要求保护 1996 年 8 月 2 日提出的序号为 60/023,218 的美国临时申请的权益, 通过引用包括于此. 本申请关系到美国专利申请序号_____, (代理案号 16598.702), 提出于 1997 年 8 月 1 日, 题为“共享卷的分布式控制方法和装置”, 发明人 James J.Wolff, 以及美国专利申请序号_____, (代理案号 16598.703), 1997 年 8 月 1 日提出, 题为“多处理器的分布式控制方法和装置”, 发明人 James J.Wolff, David Lathrop, 以及美国专利申请号_____, (代理案号 16598.704), 1997 年 8 月 1 日提出, 题为“数据库的分布式控制方法和装置”, 发明人 James J.Wolff. 这些参考被引用而包括于此.

发明背景

版权许可

本专利文档披露的部分包含的资料受到版权保护. 版权拥有人对专利的任何披露部分以影印件出现在美国专利和商标局专利文件和记录中表示无异议, 但专利所有不得移作它用.

发明领域

本发明的领域一般与用于计算机网络中共享资源的分布式控制的系统有关. 本发明领域尤其涉及网络环境中多主机间对资源, 例如存储装置、数据库和项目的分布式控制.

背景

计算机系统中通过各种通信装置共享多个计算机的文件信息是常事. 计算机向其他计算机发送文件信息的一些常用方式是通过使用调制解调器和网络. 与网络相比调制解调器速度极低. 网络要求有文件服务器, 对于高速网络文件服务器

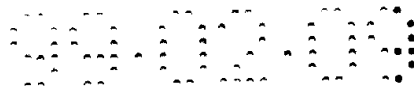
的速度成为瓶颈。这种瓶颈妨碍了需要以高于文件服务器所能达到的速度共享文件信息的应用。文件服务器是在所谓客户/服务器模式下工作。在这种模式下客户向服务器提出请求，服务器随后执行请求。在文件服务器情况下，可以有多个客户向服务器提出请求。文件服务器接收多个请求并将它们排成队列，随后按时对它们服务。因为一个实体，即文件服务器是对控制文件系统的数据库操作，这里文件信息以逻辑和相关方式驻留在存储媒体上，这样文件系统的一致性得以维持。

网络技术的一些进展已允许各种装置把存储器直接连接在网络上，就象人们可将打印机作为网络连接外设(NAP)连到网络上那样。NAP模式可以除去与文件服务器相关的速度瓶颈，但牺牲了灵活性。在NAP模式下无法维持文件系统的一致性。这相当于使多台计算机局限于对文件信息的只读访问，因而防止了对文件信息的修改。如果多台计算机要对文件信息修改则文件系统会变成崩溃，这是因为无法以逻辑和相关的方式在存储媒卷上维持文件系统的数据库结构，因为有多多个输入试图修改文件系统而无法维持高速缓冲器中的文件系统数据库结构的相关性。因此这些NAP型系统表明它们本身是高速、只读、文件信息共享系统。

所需的是一种用于共享文件信息的改进系统和方法。较佳地这样的文件系统将除去文件服务器的瓶颈从而增加访问速度，就象NAP模式所提供的那样，但同时又保持了客户/服务器模式的灵活性。

在计算机系统中密集的计算处理常常需要化费巨大的时间才得以完成。在多处处理系统中将处理分布到计算机中所有的各个处理器可以加速这种计算。因为单机中的处理器共享处理所依赖的以直接方式相连的同一物理资源(例如PCI, 存储器, 磁盘和CPU总线)。这种方法的缺点是它要求有非常昂贵和非通用的硬件设备，而软件编写必须利用与硬件模型有关的这一能力。另外，处理功效严格受可放入在计算机单机系统中的CPU的数量的限制。由于物理资源要在数量不断增加的CPU间分配，这种方法进一步受到由此产生的严重瓶颈的限制。

第二种解决办法是将处理在经网络直接相连的多个处理器间进行分配。由于参预的处理器数量事实上可不受限制，这种方法颇具吸引力，且所用机型可为廉价和通用机。然而一系列缺点使这一方法极其低效而且在许多场合下实际是无用的。网络分布处理的最根本的缺点是要要求在运行该处理的每个机器之间进行信息通信和处理结果的通信可能速度很慢，这是由于它一般须要经低速的网络连接通



过常规的客户/服务器(随之而来的是客户/服务器模型有关的瓶颈效应)而进行。这些局限性产生了极其大量的处理开销,即要求有大量的数据输入和输出。随着更多的机器加入该处理而增加的网络流通量又进一步降低了分布式处理的效率。

所需的是一种通过网络的分布式处理的改进系统和方法。这一系统将除去现行分布式网络有关的瓶颈和缺点,但同时又保留其优点。这一系统将允许分布式处理在交叉平台环境下运行和管理。

发明概述

卷的分布式控制

本发明的第一实施例披露了分布式卷管理系统。在本实施例中网络上的多个客户共享对数据存储卷相关性的控制和响应能力。该系统包括:与数据存储卷相关联的控制表。该控制表包括用于指示控制表是否可用的可用性指示和用于指示控制表拥有权的标识指示。该系统还包括在客户机上运行的各个控制过程。该控制过程请求和获得控制表的独占拥有权以提供客户对数据存储卷的独占与访问。

在本发明的一个实施例中,对于连接到数据存储卷的多个客户机间实行对数据存储卷的互斥写访问的方法包含下述动作:

检测第一客户机处的写请求;

响应检测动作启动第一客户机上的请求例程,这里请求例程包括如下动作:

a)根据控制表中拥有权指示确定第二客户机拥有对数据存储卷的写访问,以及

b)响应于该确定动作,向第二客户机发送释放请求以释放写访问;

c)响应于来自第二客户机的确认通知

i)用第一客户机的拥有权指示替换控制表中的拥有权指示;

ii)安装用于写访问的数据存储卷;以及

在第二客户机获得释放请求;

响应于获得动作启动第二客户机上的释放例程,其中所述释放例程包括下述动作;

a)释放对所述数据体的写访问;以及

b)向第一客户机发送关于释放动作的确认通知。

数据库的分布式控制

本发明的另一实施例中披露了一种分布式数据库管理系统。在本实施例中网络上的多用户共享对具有多重记录的共享数据库的相关性的控制和响应能力。客户机相互间仲裁对每条记录的每一字段的独占编辑。该数据库管理系统包括与多重记录每一字段相关联的锁定字段。每一客户机执行一驻留的相关性处理过程。相关性处理过程对编辑字段请求作出响应。如果没有对该字段作字段锁定则授权该请求。当请求被授权，该相关性处理过程通过对该字段加锁发出对该字段的独占拥有权。相关性处理过程还对退出编辑该字段作出响应。当收到退出编辑请求后相关性处理过程通过移去加锁解除对该字段的独占权。每个客户机还执行一数据库处理过程：接收查询，响应于该查询从数据库提供记录，接受编辑字段的请求，将编辑字段的请求传递给相关性处理过程，数据库处理过程还接受退出编辑字段的请求并将该退出编辑字段请求传递给相关性处理过程。

在本发明的一个实施例中，管理具有多重记录的数据库的方法以及对连接到数据库的客户的每条记录内的每一字段提供独占编辑访问的方法包括下述动作：

在第一客户机处检测编辑字段的请求；

响应于检测动作启动第一客户机上的相关性例程，其中相关性例程包括的动作作为：

a)根据与该字段相关联的拥有权指示确定多客户机中另一个具有对该字段的编辑访问；以及

b)响应于该确定动作，等待与该字段相关联的拥有权指示被解除；

c)对与该字段的拥有权指示的解除作出响应；

i)将第一客户拥有权指示连接到要编辑的字段；以及

ii)提供对该字段的编辑访问；以及

在第一所述多客户处获得退出编辑一字段的请求，

响应于该获得动作启动第一客户机上的相关性例程，其中该相关性例程从该字段上移去第 1 客户拥有权指示。

处理的分布式控制

在本发明的另一个实施例中披露了一种分布式项目管理系统。在该实施例中网络上的多个客户共享对存储在网上并可由客户访问的项目管理文件夹中定义的项目的控制和响应能力。项目管理文件夹包含一个或多个记录。该记录包括源文件名，源文件位置，应用程序名，应用程序代码，目标文件名和目标文件位置。网络上的每个客户执行一管理处理过程用于确定文件夹位置并读出文件夹中的记录。当记录被阅读时，要确定该记录调用的资源，即应用程序是否存在于该客户机上。如果资源不存在，该客户机放弃该文件夹。如果资源存在，该管理处理过程读出记录的内容，包括源文件名，源文件位置，应用程序名，应用程序代码，目标文件名和目标文件位置。管理处理过程随后使客户驻留应用程序/资源打开记录中命名的源文件。随后管理处理过程使客户驻留应用程序在应用程序代码的指引下执行该源文件上的处理，应用程序代码也从记录中获得。当管理处理过程检测到该应用程序已按照应用程序代码完成了任务，该管理处理程序便引导应用程序/资源把完成的任务/工作产品存储在各字对应于记录中目标文件名的文件中，并将该文件存储在记录中找到的相应于目标文件位置的网络位置中。管理处理过程随后把项目管理文件夹中该记录标记为已完成。

在本发明的一个实施例中，将项目分布到连接在计算机网络上所选客户机的方法包括的动作为：

在计算机网络上创建一项目管理文件及至少包含一个记录的项目管理文件夹，该记录包括有源文件名，源文件位置，第一应用程序文件名，第一应用程序代码，第一应用程序目标文件名及第一应用程序目标文件位置；

在第一客户机处检测项目管理文件在网络上的位置；

响应于检测动作启动第一客户机上的管理处理过程，该管理处理过程包括的动作如下：

a) 读出项目管理文件中的记录；

b) 确定第一应用名与第一客户机上的对应的应用相匹配；

c) 把第一应用程序代码传递给相应的应用以使相应的应用按照第一应用程序代码对源文件进行操作并将所得的工作成果以带有目标文件名的文件存储在目标文件位置。

d) 至少使一个记录完成。

附图简述

从下述结合附图的详细描述本发明的这些和其它优点对于本领域内的熟练人员来说将变得更明显:

图 1A 是现有技术的客户服务器网络的硬件方框图。

图 1B 示出了图 1A 所示的每个客户机上存在的软件模块。

图 1C 示出了图 1B 所示模块的功能关系。

图 2A 的硬件方框图示出了多个客户机和共享存储体间的无服务器网络连接。

图 2B 示出了在图 2A 的每个客户机上存在的软件模块。图 2C 示出了图 2A 所示的软件模块间的功能关系。

图 3A 示出了关于图 3A 中所示的共享存储卷的访问控制表。

图 3B 示出了图 2A 所示的共享存储体中的存储卷控制表。

图 4 示出了图 2A 中所示的共享存储卷的文件目录结构的一个示例。

图 5A-E 示出了允许多个客户机对共享存储卷共享读和写访问的处理过程。

图 6 示出了连接到客户网络的现有技术数据库服务器的方框图。

图 7 示出了多个客户机直接连接到共享存储卷上一个数据库的方框图。

图 8A 示出了关于图 7 的共享存储卷的访问控制表。

图 8B 示出了关于图 7 中所示的共享存储卷的存储卷控制表。

图 8C 示出了关于图 7 中所示的共享存储卷的更新列表。

图 9A 和图 9B 分别示出了与图 7 共享存储卷上的任一关系数据库或平直数据库相关联的增强型数据结构。

图 10 的功能框图示出了实行分布式数据库处理的每个客户机上的软件模块。

图 11 是在数据库引擎总体结构中加入了图 10A-F 的每个图后的总体处理流程图。

图 12A-F 示出了为了在共享存储卷上保持一相关数据库, 图 7 所示的每个客户机所实行的处理过程。

图 13A-C 示出了多个客户机协同处理由项目管理器文件指令的任务所需的网络环境。

图 14A-D 是与图 13A-C 所示的项目管理器文件相关联的详细的数据结构。

图 15A-B 示出了可在图 13A-C 所示网络上实现的项目的示例。

图 16A-D 示出了与图 15A 中所示的处理相关联的屏幕接口。

图 17A-F 示出了为了实行分布式处理管理在图 13A-C 的客户机上实施的处理过程。

说明

图 1A 是现有技术客户服务器网络的硬件框图。该网络包含客户机 54-56[分别标为客户机 A-B]，服务器 60 以及存储卷 64 和 66。存储卷 66 是一紧致盘。存储卷 66 是一冗余盘阵列(RAID)。每个存储卷有一与之关联的文件目录。存储卷 64 包含文件目录 62。每个客户机 54-56 分别包含文件目录 62 的高速缓冲副本 50-52。

计算机 54-56 经局域网(LAN)连接到服务器 60。服务器 60 经小型计算机系统接口(SCSI)连接而连接到存储卷 64 和 66。为使客户机 A-B 与卷 64-66 之一相连它们首先必须向服务器 60 登录访问请求。对此响应，为了保持存储卷的相关性，文件服务器 60 运行对应于卷管理写访问和文件管理的过程 58。

图 1B 示出了在开机期间服务器 60 上的硬件和软件模块。服务器 60 包含了非易失性存储器 84，CPU 和本地存储器 86。在非易失性存储器 84 中有一个标记为操作系统(OS)内核的软件模块。存储器 86 含有全套 OS90，相关的驱动器 92 以及启动配置 94。在工作中，开机时，CPU 80 启动操作系统内核 88 的代码序列。CPU 随后访问存储 86 上的主操作系统 90。在操作系统初始化的过程中，各种驱动程序 92 被初始化。这些驱动器把 CPU 与键盘、监视器、打印机和存储器等外围设备相连。操作系统被初始化后便执行启动文件 94，使一个或多个应用程序能运行。

软件模块框 86 所示的软件模块可在加电期间工作。与 OS 有关的模块是访问控制模块 96，命令接收模块 98，调度模块 102 以及逻辑到物理转换和空间分配模块 110。这些模块是屏幕驱动器 104，CD 驱动器 106 以及 RAID 存储驱动器 108。最后，由启动文件 94 使应用模块能工作。

图 1C 的框图显示了图 1B-C 中软件模块间的输入输出 I/O 的功能关系，框图中包括与服务器通信的客户应用程序 150，服务器依次又与存储器 64 或 66 之一

通信，如开始图 1A 所示。文件服务器包含一逻辑 I/O 框 158 和一物理框 160。逻辑框 158 包含命令接收模块 98，访问控制模块 96 以及文件目录 166。物理 I/O 框 160 包含调度模块 102，逻辑到物理转移和空间分配模块 110 以及分别与监视器、CD 和 RAID 存储器有关的驱动器 104-108。RAID 存储器 64 包括的文件目录 64 含有目录树以及对位于存储器 64 上的所有数据的访问权限。

工作中客户应用程序 150 经服务器 60 与存储器 64 和 66 之一通信。从客户应用程序 150 接收的命令包括：创建、删除、打开、关闭、读出和写入。这些命令中的每一条命令是用一个逻辑文件名来实现的。命令从命令接收模块 98 传递到访问控制模块 96。每一个访问控制模块 96 与文件目录 166 的相应一个接口。通常一个物理存储体有一个文件目录。此外，一个存储体可跨越多个物理存储器，如软件带中那样。一个物理存储体可以是单个存储器或是存储器的一个分区。文件目录将结合图 4 详细讨论，一般包括一个逻辑文件名的目录树以及与每个文件相关的访问特权。控制模块 96 根据文件目录 166 中包含的访问特权确定指定的命令是否应被执行。例如，应用命令包含了要对列为只读的文件或存储体进行写请求，则该请求不能被访问控制模块执行。另一方面，若接收的命令将被控制模块执行则与该命令有关的数据便被直接传送到调度模块 102。物理 I/O 框是与计算机系统外环境的主要接口。物理 I/O 处理与磁盘或磁带系统交换的数据块。因此它关心的是那些数据块在辅助存储器上的放置以及那些数据块在主存储器中的缓冲。调度模块关心的是使磁盘或磁带的访问调度到最佳性能。I/O 缓冲器被指定而辅助存储器在这一层次上被分配。一条命令(例从特定逻辑位置读取数据)从调度模块 102 传送到逻辑到物理转换和空间分配模块 110。该模块负责把逻辑位置转换成特定存储器上的实际物理位置。一个文件例如可以不存储在存储器的连续部分上，而是可存储在分散的片段上。该模块然后执行将逻辑文件名转换成一系列的物理段。该模块还负责空间分配。空间分配的意思是当收到写命令后由模块 110 确定数据写入的实际物理位置。服务器 60 和存储器之间的最终接口是经过适当的设备驱动程序实现。每一个设备驱动程序处理一种类型设备。设备驱动程序是操作系统中知道磁盘控制器有多少寄存器以及它们用作何种用途的仅有部分。它独自便知道有关扇区，磁道，柱面，磁头臂移动，交错因子、马达驱动，磁头调整时间以及所有使存储器正常工作的其它机构。一般而言，设备驱动程序的工作是要从设备驱动程序之上与设备独立的软件接收抽象的请求并保

证使请求被执行。

图 2A 是当前发明的较佳实施例的硬件框图，其中客户机可直接连到一个或多个共享存储卷而不损失包含在那些卷上的数据的相关性。客户机 54-56，存储器 64-66 及服务器均被示出。RAID 存储器 64 包括文件目录 204，访问控制表 206 以及卷控制表 208。客户机 54-56 每一个都实现用于分布式管理存储卷 64-66 的处理过程。客户机 54 实现处理过程 214 而客户机 56 实现处理过程 216。客户机 54-56 包含文件目录的驻留拷贝 200-202。

为了对来自客户机 54 或 56 的对存储卷 46-66 的请求服务，不再需要标有 X 记号 212 的服务器 60。代之以将客户机 54-56 直接连到存储卷 64-66。在一个较佳实施例中该连接或者基于光纤信道 ANSI 标准 X3.230 和/或 SCSI-3 ANSI 标准 X3.270，然而也可支持其它媒体连接。本发明的其他实施例可适用于支持连接存储器 802 和主机 804 的其它方法，例如在其它可能方法中实施例使用 Fast-40(Ultra-SCSI)，串联存储器结构(SSA)，“FireWire”IEEE 标准 P1394，异步传输模式(ATM)，或可定标相干接口(SCI)IEEE 标准 1596-1992，或上述的一些组合。光纤信道结构对串联通信和存储器 I/O 两者提供了高速接口。这些链路使用工业标准接口能以当前现存结构速度的 10 到 100 倍速度发送数据。光纤信道是数据通信方面最近革新的一部分，把串联传输线，设计技术和工艺加入到传统上使用基于总线型的并行数据传输的应用和计算机结构中造成了这种革新。传输线用于数据传输具有优良的特性并具有基于总线的设计，使得信号能被无差错地和/或以较高的数据率传输较远的距离。光纤信道支持灵活的拓扑结构，包括点到点链路、分组交换以及共享媒体回路拓扑。在一个较佳实施例中，光纤信道网络把客户机 54-56 与存储卷 64-65 两者链接起来。存储卷 64 配备了分流软件，它可跨越多个磁盘并发地读写数据。这种分流能力使得网络媒体即光纤信道的带宽按存储器的带宽等分。这就可置如下事实于不管，即存储器上的各个磁盘只能以网络连接带宽的一小部分提供数据。把一个文件跨多个磁盘可克服由磁头臂的物理限制所施加的带宽约束，这只要让多个磁盘以平行方式提供数据驱动即可。

为了保持由客户机 A 或客户机 B 写入到存储器 64 的数据的相关性，处理过程 214-216 分别在客户机 54-56 上实行。每个处理过程控制新卷的按装，写访问和文件管理。为了将数据写到存储器 64，客户机必须首先确定在共享存储卷上的访问控制表 206 和卷控制表 208 中的写条件存在。当确定该条件存在，客户机

才可写存储卷。当该条件不存在，写请求被拒绝。在一个较佳实施例中，访问控制表和卷控制表分别驻留在具有读写能力的一个物理存储器的分开的卷上。因此，访问控制表和卷控制表不必且事实上通常也不用驻留在它们要控制访问的同一物理装置上。而是让每个进程 214 和 216 使用共享的访问和卷控制表以确定何时及在何种条件下可以对可能与它们相连的许多卷之一进行写操作。

图 2B 是客户机 54 的硬件方框图。客户机 54 包括 CPU250，非易失性存储器 254 及本地存储器 258。非易失性存储器 254 包括一个 ISO 256，本地存储器 258 包括主 OS 260 设备驱动器 262 和启动文件 264。

在开机时 CPU 250 从引导 ROM 254 加载 OS 内核并从本地存储器 258(例磁盘驱动器)读出操作系统 260 的其余部分。在使操作系统能工作的后期阶段，使设备驱动器 262 和启动文件 264 能工作。软件模块框 252 显示了在开机期间使之能工作的软件模块。在使操作系统 260 能工作期间，命令接收模块 98，访问控制模块 96、调度模块 102、逻辑到物理转换和空间分配模块 110 以及变换模块 268 均被使能工作。在使设备驱动器 262 能工作期间，屏幕驱动器 104、CD 驱动器 106 以及 RAID 驱动器 270 也被使能工作。在启动文件启动期间，卷控制应用程序 266 和驻留应用程序 100 被使能工作。

图 2C 示出了图 2B 中所示的软件模块在加电使能期间它们之间的功能关系。客户机 54 以及物理存储器 64 和 66 也被示出。物理存储器 64 包括文件目录 62、访问控制表 206，卷控制表 208 以及数据块 1 到 N。物理存储器 62 包括它自己的文件目录和数据块 1 到 N。客户机 54 包括卷控制应用程序 266，逻辑 I/O 块 158 及物理 I/O 块 160。逻辑 I/O 块 158 包括命令接收模块 98，访问控制模块 96，文件置换模块 268 及文件目录 166。物理 I/O 块 160 包括调度模块 102，逻辑到物理转换和空间分配模块 110 以及设备驱动器 104-106 和 270。

在工作中，设备驱动器 270 在操作系统初始化期间并未完全能工作。因而物理驱动器 64 尚未显示出可供客户机 54 使用，即被连上。当卷控制 266 能工作后它向命令接收模块送出一系列命令。这些命令通过控制模块 266 造成对锁定存储驱动器 270 的发现。在这一发现处理的过程中卷控制模块也发现了在物理驱动器 64 上的访问和卷控制表 206 和 208。在这个发现过程期间，除了卷控制应用程序外的任何其它应用将不访问物理存储器 64。当卷控制模块 266 已分别从访问控制表 206 和卷控制表 208 读得访问特权和卷特权后，随即忙于一系列所设计的命

令以把这些访问特权写入到文件目录 166 中。例如，已确定卷控制表指示客户机 54 要对物理设备 64 读/写访问，则卷控制模块将发出一条安装 R/W 命令，当被命令接收模块接收后将对文件目录 166 更新并包括这些特权。随后，当文件目录 166 被写回到物理设备 64 时，那时驻留在实际物理设备 64 上的文件目录 62 也将被更新。因而，访问和卷特权是从访问和卷控制表读出并被卷控制模块 266 写入到高速缓冲文件目录 166 和物理文件目录 62。基于在本较佳实施例中访问和卷控制表仅驻留在那些设备之一上，这一过程可以跨越多个物理设备而执行，然而其它位置也可用于这些表。

功能上尚未被讨论的其余模块是协议转换模块 268。协议转换模块 268 访问控制模块 96 到文件目录 166 的接口并对诸如可能在客户机 54 和 56 上存在的不同操作系统间保持一个统一的文件目录结构。客户机 54 例如可以运行 Macintosh System 7®操作系统而客户机 56 可以运行 Windows NT。在每个客户机上的协议转换模块 268 实施单一的文件目录结构从而使得本发明的交叉平台操作能够工作。协议转换模块 268 提供将通信规则和通信格式转换成公共 API 的通信规则和通信格式。因而，目录和文件可以数据格式跨多种操作系统而存储。该格式具有特定的 ASCII 字符，文件名的转换和特定的转换，它们将结合图 4 对文件目录和访问控制作更为详细的讨论。当协议转换单元收到一条命令后它就调用它内部的与命令类型对应的处理例程并由该处理例程处理该命令。

图 3A 示出了上述结合图 2A-C 讨论过的访问控制表 206 的详细的数据结构。在访问控制表的字段 256 上登录表明哪一个用户进程 214-216[见图 2A]已登录到访问和卷控制表上。直到用名登录到访问控制表之前该客户机上的任何应用程序将不能访问任何物理设备，它们的设备驱动器具有类似于设备驱动器 270[见图 2C]的锁定控制。访问和控制表还有一个与每个客户机相关联的刷新通知字段 358。与特定客户机相关联的字段内的逻辑真条件确保系统中其它用户的访问特权改变发生时该客户机将被其驻留的卷控制进程所通知。逻辑段具有相反的作用，即在访问控制表的刷新通知字段中具有逻辑段的客户机将不会被客户机上的卷控制进程提供通知。访问控制表还提供客户机间传递能力的消息。字段 360 包含一则请求消息。字段 362 和 354 分别包含基于文本和基于逻辑的响应消息。最后，访问控制表还包括一信号量字段 352。信号量字段中存在信号量表明客户机 54 或 56 之一已分别掌握了对访问和卷控制表 206-208 的控制。在信号量字段 352 中

已写入标识符的客户进程可以改变与每个卷相关联的特权并能启动写状态的改变。

图 3B 示出了与卷控制表相关的数据结构。具体说，表 208A 和表 208B 已被显示。每一卷有一个卷控制表，它支持对多个写入者的分布式控制。卷控制记录 208A 内的字段均用后缀“ A ”标记。识别为卷控制表 208B 的字段均含有后缀“ B ”。字段 390 标识与特定卷控制表有关的逻辑卷。字段 390A 标识与特定卷控制表有关的逻辑卷。字段 390A 标识 RAID 存储器 64[见图 2A]是与卷控制表 208A 相关联。字段 390B 标识 CD-ROM66[见图 2A]是与卷控制表 208B 相关联。字段 396 包含逻辑真/假以表明特定的卷是否已能写入。字段 392A 表示 RAID 设备 64[见图 2A]是能写入的。字段 392B 表示 CD-ROM66[见图 2A]是不能写入的。字段 394 表示哪一个客户机当前对特定卷有写访问。字段 394-A 表示客户机 54[见图 2A]当前对 RAID 存储器 64 有写访问。字段 394B 表示没有客户机对 CD-ROM66[见图 2A]要写访问。字段 388 表示哪一个客户机对于特定的卷有安装访问特权。逻辑真表示该客户机可安装该卷。逻辑假表示相反。字段 396 表示每个客户关于请求改变它的当前卷设置的能力。逻辑假表示对客户作出的诸如从只读到读写(或相反)的改变请求不予封锁。逻辑真则表示封锁客户作出的改变请求。字段 384 是逻辑真/假以表示当与特定卷有关的其它客户机对该卷作出改变时具有只读特权的客户机是否将被更新。字段 368 是时间戳，指示客户接收更新的文件目录 62[见图 2A]拷贝的最新时间。字段 382 是一个时间戳，指示特定卷被任一客户机更改的最新时间。通过把最新更改时间字段 386 与卷更改时间字段 382 相比较进程 214-216[见图 2A]就能确定具有自动更新特权的客户机何时需要对文件目录刷新。

下述列表描述了在另一实施例中关于卷和访问控制表的数据结构和功能调用。列表描述了用于这些表的一些数据结构元素。访问表包括一个 StudioBOSS-DB-Header 14XX 及一个数组 Partition-Volume-Record's 14XX。

Volume-DB-Header 14XX

在本发明的第一实施例中 Volume-DB-Header 14XX 被存储在 Volume-Table(卷表)的开头，然而其实际位置可改变。对其每个字段的描述如下：

Table-Version-Number 14XX

该表格式的版本号。当前置定为 SB-DB-Verssion 14XX，用与将来的兼用。

Table-Open-Counter 14XX

保持对表曾被打开的次数的跟踪. 当前与 DEMO 副本一起使用以确定 DEMO 版本失效条件之一(见 SB-DB-OPENS -UNTIL-EXPIRE 14XX)

Table -Write-Lockout-Semaphore 14XX

如果表可安全访问设置成 MAXIMUM-USER 14XX . 否则设置成已占据信号量的用户数. 如被占用, 该表不可靠且不应作改变.

Number-of-Users 14XX

在该表中创建的用户总数.

Number-of-Volumes 14XX

表中卷的总数.

User-Names [MAXIMUM-USERS 14XX]14XX

表中以 Pascal 串格式表示的用户名(长度字节 + 字符). 注意: 要求找表中下一名字, 搜索数组中下一非另长串.

USER-Has-Owner[MAXIMUM-USERS 14XX]14XX

如为合法用户设为 TRVE , 否则为 FALSE .

USER-Logged-ON[MAXIMUM-USERS 14XX] 14XX

如该用户当前已被登录在表中设为 TRVE , 否则为 FALSE .

User-Refresh-DB[MAXIMUM-USERS 14XX]14XX

如果需要通知该用户关于表中已作了改变, 要求完全关闭/重启动表文件(如在用户已登录状态下要加入或移去卷等时)设为 TRUE .

User-Receiving-Request 14XX

接收方的用户号, 否则设成 MAXIMUM-USERS 14XX .

User_Sending_Request 14XX

发送方的用户号, 否则设成 MAXIMUM_USERS 14XX .

Volume_With_Request_Position 14XX

从涉及请求的卷记录文件开头算起的索引号, 否则为 0 .

Request_Response 14XX

如果响应是肯定的且 Request_Message 有效则设定为 TRUE .

Sender_Message 14XX

作为请求的一部分从发送者送到接收者的消息的 Pascal 风格串.

Response_Message 14XX

作为请求的一部分从接受者送到发送者的消息的 Pascal 风格串。

DB_Header_Filler [400]

保留备用。应设为 0。

Partition_Volume_Record [MAXMUM VOLUMES 14xx] 14XX

在 Volume_DB_Header 14XX 之后跟有 MAXIMUM_UOLVMES 14XX (200) Partition_Volume_Record's 14XX。对其每一个字段描述如下：

Volume_Info 14XX

该字段是一个 Partition_Volume_Header 14XX 记录且包含有关表中下一卷的信息。参见下述对其字段的完整说明。

Users [MAXIMUM_VSERS 14XX] 14XX

该字段是 Partition_User_Record 144 记录的一个数组并包含该卷直到 MAXIMUM_USERS 14XX(200)用户有关的信息。见下述关于该字段的完整说明。

Partition_Volume_Header 14XX

关于该 Partition_Volume_Header 144 记录的字段的说明如下：

Volume_Name 14XX

卷名的 Pascal 串。

Writer_User_Number 14XX

对该卷有写访问的用户数。若没有则设成 MAXIMUM_USERS 14XX。

Volume_Last_Modified-Time 14XX

包括软件注意到卷被改动的最新时间。

Vol_Header-Filler[100] 14XX

保留备用。应设为 0。

Partition_User_Record 14XX

关于 Partition_User_Record 14XX 记录的字段的说明如下：

User_Number 14XX

这是表中的用户号。它必须等于用户[MAXIMUM_USERS 14XX]数组中它的索引号。

Write_ACCESS 14XX



如果用户有读/写访问, 设为 TRUE, 否则为 FALSE.

Mount_Access 14XX

如果用户想安装该卷以供访问, 设为 TRUE.

Auto_Update 14XX

如果用户希望对卷的变动和更新自动检查, 设为 TRUE. 当为 FALSE 时下述 Auto_Update_Check_Seconds 14XX 字段为无效.

Auto_Update_check_Seconds 14XX

当 Auto_Update 14XX 是 TRUE 时, 这代表了对卷的变化作再次检查前必须经过的秒数.

Last_Modified_Time 14XX

包含了用户注意到卷被改变的最新时间.

Reserved [50]

保留备用. 应设为 0.

由 Preference_Record 14XX 组成的优先权文件位于每一台主计算机上. 这一信息的目的是允许每台主机在卷表内有一个唯一的用户号使得每一主机能登录入该卷表并对每一个文件系统有它自己的配置参数组.

File_Signature 14XX

它被设置成一个唯一值, 该值表示下述信息作为 Preference_Record 14XX 是可信赖的最可能的合法数据组.

Admin_Expired 14XX

这是一个设置为 TRUE 或 FALSE 的值, 它表示这是否是过期的软件的 DEMO 副本.

Total_Tables 14XX

该值指示了已被配置并在上次搜索中找出的卷表总数. 这是一个在纯启动时希望被自动找出的数字.

SB_Process_Ticks 14XX

这是为了检查用户事件或表变化两次调用卷文件控制软件组成之间的时间值.

Monitor_DB_Ticks 14XX

这是在卷文件控制软件组成将检查表改变(例如来自另一用户对写访问的请



求)前必须经过的时间值。

User_Name 14XX

这是一个字符串，表示在卷表登录时将被搜索的用户名，用户号将从用户名生成并用于随后的访问。

Admin_Password 14XX

这是一个字符串，表示为了访问该主机的卷文件控制软件组成的管理功能所需的当前口令字。

Volume_Volume_Names [MAXIMUM_Volumes] 14XX

这是一个字符串数组，表示卷文件控制软件组成应搜索卷表时所处的文件系统名。

Pref_Filler [396] 14XX

这是一个字节数组，这些字节是用作将来扩展用的位置保持器。

Ignore_Bad_Mounts 14XX

这是一个能设置成 TRUE 或 FALSE 的变量，它表示是否应当通知用户关于试图安装文件系统期间出现失败的错误。

Lockout_Access_Changes 14XX

这是一个能设置成 TRUE 或 FALSE 的变量，它表示是否允许用户对卷表作改动，例如请求改变读/写特权。如确定用户应不能改变任何设定该变量可由管理员设定。

Keep_DB_Volume_Locked 14XX

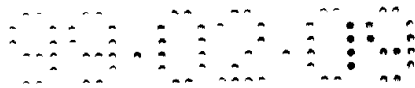
这是一个能设置成 TRUE 或 FALSE 的变量，它表示卷文件控制软件组成是否让卷表文件所在的文件系统保持锁定或不锁定。较佳的是它被设为 TRUE，但这不是一个要求。

Keep_DB_Volume_Unmounted 14XX

这是一个能设置成 TRUE 或 FALSE 的变量并且表示卷文件控制软件组成是否应让卷表所在的文件系统保持不被安装，较佳的是它被置成 TRUE，但这不是一个要求。

Volume_Node_Record 14XX 是被卷文件控制软件组成使用的以链表形式出现的数据结构，每个被监控的文件系统对应一个。

Next_Volume_Node_14XX



这是一个变量，它是链表中指向下一个 Volume_Node_Record 14XX 的指针。
Volume_Reference_14XX

这是一个数据结构，它是对文件系统和它的与操作系统有关的内部结构的一个引用。为了加锁/解锁、安装/卸装以及找到诸如自由空间、最新修改时间等信息，它被用于引用该文件系统。

Volume_Check_Timer 14XX

这个变量是一个计时器，期满时表示此时该检查卷以了解它是否需要刷新。
Free_Megabytes 14XX

这个变量指示了文件系统上的自由空间量。

Volume_Record_Postion 14XX

这个变量是卷表中关于状态的指示，它描述了这个文件系统用于这个主机。
User_Record_Position 14XX

这个变量是卷表中关于状态的，它表示这个用户信息是用于这个主机。

SN_DB_File_Reference 14XX

这个变量是用于访问卷表自己的一种引用数据结构。

Volume_Enabled 14XX

这是一个变量，或者是 TRUE 或者是 FALSE，它表示这个文件系统能或不能工作。不能工作的文件系统不具有与之相关的正常监控功能。这意味不能工作的卷即使需要也将不能被更新和刷新。正常情况下卷是能工作的，因而使文件系统能作为正常的实体参预系统工作。

对卷文件控制软件组成起全局作用的一些数据结构是：

SB_Admin_Prefercences 14XX

该 Prefercnce_File_Record 14XX 用于这个主机。

Head_Volume_Node 14XX

指向 Volume_Node_Records 14XX 的链表的开关，用于从头开始对链表的搜索。

Tail_Volume_Node 14XX

指向 Volume_Node_Records 14XX 的链表的末尾，用于插入新记录。

Current_Volume_Node 14XX

指向当前正被监控的 Volume_Node_Records 14XX 的指针。

Current_DB_Header_Ptr 14XX

指向表的当前 Volume_DB_Header 的指针(该表是当前正被监控的文件系统的主干), 如由 Current_Volume_Node 14XX 所指示。

Current_DB_Volume_Header_Ptr 14XX

指向属于当前正被监控的文件系统的当前 Partition_Volume_Header 14XX 的指针。

Current_DB_User_Record_Ptr 14XX

指向属于当前正被监控的文件系统的当前 Partition_User_Record 14XX 的指针。

SB_Praccess_Timer 14XX

一个定时器, 指示当计前时期满时即检查下一个文件系统。

Monitor_DB_Timer 14XX

一个定时器, 指示当计时期满时即检查是否有任何未处理的表请求。

Administration_Mode 14XX

一个变量, 取值 TRUE 或 FALSE, 指示何时卷文件控制软件组成的操作具有可供用户使用的管理员功能。

Restart_Volume 14XX

一个变量, 取值 TRUE 或 FALSE, 指示此时是否关闭并启动系统, 撤销所有表并重新往回登录。

Suspend_Operations 14XX

一个变量, 取值 TRUE 或 FALSE, 指示所有文件系统和表的监控是否应暂停。

图 4 是能在利用本发明的客户机 A-B 上同时实现的用于统一文件目录格式的数据结构详图。通过如上图 2C 讨论的协议转换模块 268, 该目录结构的维护可不管每个客户机上所用的操作系统如何。图 4 所示是存储器 64 的文件目录 62 和物理地址映射 452。文件目录 62 包括卷标头 454、目录记录 456、文件记录 458 以及文件位置记录(范围)460。这一目录结构通常与提供给 MacIntosh® 计算机的 System8 操作系统相关的 HFS 文件目录格式有关。卷标头 454 包括卷名、它的创建日期, 它的更新日期、软件锁, 属性和特权列表、卷可用性位映射以及概括地定义该物理卷的许多其它参数。其记录 456 被引用的多个目录记录是与卷记录 454

相关联。每个目录记录包括一个指向父目录的指针、名字、建立时间和修改时间。其次就是与其文件记录 458 被引用的每个目录相关联的多个文件记录。每个文件记录包含名字、类型、锁定指示符、建立和修改时间以及其它文件级的信息。与每一文件和目录记录相关联的是其记录 460 被引用的多个文件位置记录。每个文件位置记录包括一指向文件开始的物理地址以及该文件长度的指示。如果文件存储在不相连接的段中，则将有一个上溢指示符指明该文件的下一部分物理地址和该部分的长度。正如在物理地址记录 452 中所指示的那样，文件位置记录地址和地址长度对应于该文件内容的实际物理地址位置。例如在文件位置记录 460 中，文件 ONE 列出为起始于物理地址 1000，具有 44 个扇区长度。在物理地址记录 452 中，文件 ONE 起始于物理内存地址 1000 并延伸到物理地址 1044。总长为 44 个扇区。每个操作系统都有其自己的文件目录结构，在许多方面不同于图 4 中所披露的那样。在上述图 2C 讨论的本发明的较佳实施例中，与每个客户进程 214 - 216 相关的协议转换模块 268 实施统一的文件目录格式，而不管每个客户机使用何种操作系统。这就确保在任一客户机上的任何应用程序间具有交叉平台兼容性而不管每个客户机上可能存在的 OS 如何。因而运行 Macintosh System8® 操作系统的客户机可以读写由另一个运行 Windows NT™ 操作系统的客户机所创建的文件。

图 5A-E 示出了分别与图 2A 所示的客户机 A 和 B 相关联的进程 214 - 216。每一客户机完成图 5A-E 中所示的所有效能。

与加电和登录有关的过程示于图 5A。从过程 500 开始，设备驱动器和卷控制过程被加载到客户机。控制随后传递给过程 502，在该过程中由用户定制的包括了锁定特性的设备驱动器被定位。另外，连接到包含有卷和控制表[见图 2C]的物理卷的特定设备驱动器被标识。控制随后传递到判定过程 504。在判定过程 504，确定卷和访问控制表是否已创建。如果确定是否定的，则将控制传递给过程 506，在这里卷控制过程向管理员提供一个模板，按此模板创建访问和卷控制表。控制随后传递到过程 508，在这里新创建的访问和卷控制表被存储在具有用户定制的即可锁定的驱动器的卷上。随后控制传递给判定过程 512。另一种情况是，在判定过程 504 中如果确定是肯定的，即卷控制表确存在，则将控制直接传递给判定过程 512。

在判定过程 512 中，要确定信号量字段 352[见图 3A]是否为空。倘若确定是

否定的，控制传递到 510 用于引入一延时，在该延时间隔后控制返回到过程 512，当控制过程 512 到达肯定确定时，即访问控制表 206[见图 2A]的信号量字段 352 是空的，则将控制传递给过程 514。在过程 514，客户机标识符被写入该信号量字段。控制随后传递到过程 516，在这里掌握访问和控制表的客户机读取该访问和控制表。控制随后传递到判定过程 518。在判定过程 518，确定字段 352 中的客户机识别符是否是运行该过程的客户机的 I.D.，倘若确定是否定的，客户没有掌握对访问和卷表的控制。为此，控制返回到判定过程 512。如果是肯定的，即判定过程 518 确定客户机已掌握了对访问和控制表的控制，则将控制传递给过程 522。在过程 522 客户机读取该访问和卷控制表。控制随后传递到过程 520。在过程 520，客户机将它们的 I.D.写入到访问控制表 206 中相应的登录字段 356[见图 3A]。控制再传到过程 524。在过程 524，客户机将它们的识别符从信号量字段 352 中移去从而释放访问和控制表。控制再传到过程 526，在这里访问控制模块 266[见图 2C]控制运行此过程的特定客户机的访问和控制表中所示的那些特权对每个物理装置设定访问特权。正如上述图 2C 中所讨论的那样，这些特权是通过卷控制过程向命令接收模块 98 发送安装卷命令而设定，这些命令通过或者读特权或者读/写特权而完成。控制随后传递到连接框 A。

图 5B 示出了在客户机上进行的与请求对文件系统或卷的写特权有关的过程。在连接框 A 处开始，控制传到判定过程 550。在判定过程 550 中，确定卷控制过程 266[见图 2C]是否已收到一个写入请求。该写请求任何时候都可以通过图形用户接口(GUI)被卷控制过程所接收，该 GUI 允许用户选择特定的、该用户对它具有读特权的卷并要求将那些特权更改为写特权。如果得到的确定是否定的，即卷控制过程没有收到或没有生成写入请求，则控制传递到连接框 D。另一方面，如在判定过程 550 得到肯定的确定，则控制传递到判定过程 554。在判定过程 554，确定访问控制表[见图 3A]中字段 352 内是否存在信号量。在确定是否定的情况下，控制传到过程 552 以引入暂停。控制随后返回到判定过程 554。当判定过程 554 得到肯定的确定，控制传到过程 556，在这里发出写命令且运行该过程的客户机的标识符被写入字段 352。控制再传到过程 558。在过程 558 中用于字段 352 的值被读取。控制随即传到判定过程 560。在判定 560 中，确定字段 352 中的值是否与运行该过程的客户机的 I.D.相对应。倘若确定是否定的控制返回到判定过程 554。另一方面，如果判定过程 560 中的确定是肯定的，则控制传递到

判定过程 562。在判定过程 562，根据安装访问字段 388[见图 3B]确定这个特殊的客户机是否能够安装已被提出写入请求的卷。如果这个确定是否定的，则将控制传到过程 570，过程 570 给出写入请求被拒绝的通知。控制随后传到连接框 D。另一方面，如果在判定过程 562 中确定该卷对于运行该过程的这个特殊客户机是可安装的，控制便传到判定过程 564。在判定过程 564，根据字段 396[见图 3B]确定这个客户机是否能获得对特定卷的写访问或是否该客户机的请求访问被封锁了。如果得到的确定是肯定的，即该客户机不能获得写访问则控制传到上述讨论的过程 570。另一方面，如果判定过程 564 中的确定是否定的，即客户机没有被封锁住去获得对卷的访问则控制传到判定过程 566。在判定过程 566，根据字段 394[见图 3B]和字段 356[见图 3A]确定当前的写入者是否事实上已从网络注销。如果那个确定是肯定的，则控制传递到过程 572，发出卸装该卷命令。控制随后传到 574，由卷控制过程 266[见图 2C]向命令接收模块 98 发送安装卷命令。这使卷被安装以反映最近的改变。控制再传到过程 576，用运行该过程的客户机的 I.D.更新卷控制表中的字段 394[见图 3B]。控制再传到过程 578，客户机释放访问和卷控制表，把它们的信号量即 I.D.从卷控制表的信号量字段 352[见图 3A]中移去。控制再传到连接框 C。

另一方面，如果在判定过程 566 中当前写入者没有注销则控制传到过程 568。在过程 568，写请求被写入访问控制表的请求字段 360[见图 3A]。该请求包括了该请求所指向的卷的标识符；提出请求的客户机标识符；以及请求者的文本消息。从卷记录的字段 396 读取请求所指向的卷的标识符。文本消息可在由卷控制过程 266[见图 2]所提供的 GUI 界面中手动输入。控制随后传到判定过程 580。在判定过程 580 中，响应消息和响应逻辑字段 362 与 354[见图 3A]分别被读取并根据这些字段确定写特权是否已被当前写入者放弃。在确定是肯定的情况下，控制传到上述讨论过的过程 574。另一方面，如果该确定是否定的，则控制传到过程 578，当前请求者将它们 I.D.从访问控制表信号量字段 352[见图 3A]中移去，随后控制传到连接框 C。

图 5C 示出对另一个请求写访问特权的客户机的响应相关联的过程。过程部分始于连接框 B 且控制被传递到过程 600。在过程 600 中, 访问控制表, 具体说是请求消息字段 360[见图 3A]被读出。控制再传到判定过程 602。在判定过程 602, 确定与请求有关的客户机 I.D. 是否对应于运行该过程的客户机 I.D., 如果确

定是肯定的，即运行该过程的客户机就是请求所指向的客户机，则控制传到判定过程 604。在判定过程 604 中，确定自动响应的能力是否对该特定客户机能工作。在一个较佳实施例中，可根据访问控制表的另一字段中的输入作出该确定。在该实施例中，有一个字段与每个客户机相对应，该字段中逻辑真状态表示能自动响应而逻辑假状态表示不能自动响应。如果判定过程 604 确定能自动响应，则控制将直接传到判定过程 610。在判定过程 610，根据客户机上的当前进程确定是否在这时放弃控制。例如，如果该客户机进行的大型再现项目正做到一半，否定的判定到达了。在得到这样的确定情况下，控制将传到过程 614。在过程 614，响应的客户机将把逻辑假状态写入访问控制表的响应消息字段 362[见图 3A]。控制将传到连接框 D。

另一方面，如果判定过程 602 确定与请求有关的客户机不是运行该过程的客户机 I.D.，则控制将直接传到连接框 D。

另外，在判定过程 604 中，如果确定请求是由运行过程(过程框 602YES)的客户机提出且进一步确定自动响应特性不能使用，则控制将直接传到过程 606。在过程 606 中卷控制过程 266[见图 2C]的 GUI 部分向该用户提供一消息提示，用户对请求给出两种选择回答：是或否，并且在该响应中包括了一个基于文本的消息。该响应将被分别写入响应消息字段 362 和逻辑响应字段 354 中[见图 3A]。控制随后传到判定过程 608。判定过程 608 确定用户是否放弃了写特权。如果确定是肯定的，则控制传到过程 616。在判定过程 610 中，如果确定写控制当时不能放弃，则控制也传到过程 616。在过程 616，逻辑真状态被写入访问控制表，具体说写入响应字段 354[见图 3A]。这表示收到请求且当前具有写特权的客户正好释放这些特权。控制随后传到 618。在过程 618 中，被请求卷上的任何文件被关闭且该卷有关的读和写访问被卸装并被改装成只读访问模式。控制随后传到连接框 D。

另一方面，判定过程 608 确定用户选择不放弃对被请求卷的写访问特权，则控制传到过程 612。在过程 612 中逻辑假状态被写下到访问控制表，具体为它的逻辑响应字段 354[见图 3A]中。控制随后传到连接框 D。

图 5D 示出了把写特权转换成对文件系统实际写访问过程的有关部分。过程部分始于连接框 C，控制从这里被传递到判定过程 650。判定过程 650 根据卷控制表[见图 3B]中的字段 392 确定写请求所指向的卷是否事实上能使用。如果确定

是否定的，则控制传到过程 666，在这里卷控制过程 266(见图 2C)的 GUI 界面部分向用户传递一则消息，表明该卷不能使用且不能提供写访问。控制再传到连接框 B。另一方面，在判定过程中得到肯定的确定，即写入请求所指向的卷事实上能被使用，则控制传到判定过程 652。过程 652 确定是否要为运行该过程的客户机安装卷。如果确定是否定的，则控制传到 658，卷控制过程 266[见图 2C]向命令接收模块 98[见图 2C]发送具有读写特权的安装命令。其结果是卷被安装并且有读写特权。控制再传到判定框 654。另一方面，如果判定过程得到了肯定的确定，即卷被安装并具有或者读特权或者读/写特权，则控制也传到判定过程 654。判定过程 654 确定文件目录级的软件锁位是否对该特定卷设为能起作用以及是否该软件锁位排除对该卷的访问。若确定是肯定的，则控制传到过程 660。在过程 660 中软件锁和文件目录或者被旁路或者使之不能起作用。在较佳实施例中最好是旁路软件锁。其好处是防止不具有卷控制处理过程的网络上的用户无意中访问该受控的卷。控制随后传到判定过程 656。另一方面，判定过程 654 确定或者在文件目录级上不存在软件锁或者在文件目录级上软件锁可以通过卷控制过程 266 的命令被旁路，则控制直接传到判定过程 656。

判定过程 656 确定当前安装的卷是否具有读和写特权。如果确定是否定的，控制传到过程 662，卷控制过程 266 向命令接收模块 98[见图 2C]发送对所请求的卷卸装卷随后安装卷 R/W。对于该客户机其效果是重新把卷安装为具有读写特权。控制再传到连接框 D。另一方面，如果判定过程 656 作出的确定是肯定的，即卷已被安装且安装为具有读写特权，则控制直接传到连接框 B。由图 5D 过程引起的文件目录中的变化首先体现在与提供了写访问的卷对应的文件目录的高速缓冲器中的副本 166 中[见图 2C]，高速缓冲器中的副本更新随后被倒回或写回到物理装置 64 上的文件目录中，导致体现了新访问特权的文件目录的相关设定。

图 5E 包括了与只读卷相关的过程部分。以只读状态被安装到卷上的客户机对于在该卷可能发生的目录中的变化，即对系统上的其它客户机要对该卷进行写时所发生的变化将不被更新。因而对于这些想要自动更新能力的客户机，图 5E 的过程提供了这种能力。该过程开始于连接块 D，控制从这里传递到判定框 700。在判定框 700，根据卷控制表的字段 384 确定客户机当前以只读状态相连的任何卷是否被标记为自动更新。对于这些卷的每一个卷，这些特殊客户的最新修改字时字段 386[见图 3B]在过程 702 中被读出并与字段 382 相比较[见图 3B]。这个处

理过程的目的如下：

正常情况下，对卷具有只读特权的客户机将不刷新它们的用于该卷的本地的高速缓冲中的文件目录副本。结果，该副本可能变得陈旧，即或者显示不再存在的目录/文件，或者不能显示确实存在的目录/文件。因而，重要的是具有只读访问的客户机在其它用户对卷作改变时能具有可选项使它们高速缓冲的目录副本被定期刷新。这种刷新和更新可选项由字段 384[见图 3B]提供。下一步控制传到过程 704-706，确定是否需要刷新/更新。在过程 704 中，在最新卷改变的时间戳即字段 382 和为了获得文件目录新副本由客户机进行卸装和安装的更新时间即字段 386 之间进行比较。过程 706 确定客户的高速缓存目录副本是否陈旧。如果确定是否定的，客户机包括了驻留在卷上的文件目录的最新副本。由于不必更新，控制从而传到连接框 A。另一方面，如果判定过程 706 作出肯定确定，即高速缓存的文件目录 166[见图 2C]是陈旧的，则控制传递到判定过程 708。判定过程 708 确定访问控制数据库信号量字段 352[见图 3A]是否为空。如果确定是否定的控制则传递到过程 710。过程 710 引入一暂停，此后控制返回到判定过程 708。另一方面，如果判定过程 708 确定信号量字段 352 是可供使用的，则控制传到过程 712，客户机 I.D.被写入控制表 206[见图 3A]被写入到信号量字段。控制随后传到过程 714，在这里访问控制表的信号量字段部分被读出且控制随后传递到判定过程 716。判定过程 716 确定信号量字段中存在的客户机 I.D.，是否与执行该过程的客户机 I.D.一致。如果确定是否定的，则用户机没有掌握访问控制表且过程返回到判定过程 708。另一方面，如果确定是肯定的，即客户机已掌握访问表的控制，则控制传到过程 718。在过程 718，由卷控制过程 266[见图 2C]向命令接收模块 98 发送一卷卸装命令并紧跟一只读访问安装命令。这就造成在过程 720 中文件目录 166 的已更新副本被显示给用户。用户显示可包括关于所有卷以及在这些卷有关的所有客户的状态，即安装/卸装，只读，读/写，锁定/解锁。控制随后传递到连接框 A。

分布式数据库

图 6 所示为现有技术的客户数据库服务器的配置。其硬件组成包括客户机 54-56，服务器 60 和存储器 64。一个具有多个记录并且每个记录含有多字段的数据库 754，被存储在存储器 64 中，该存储器通过一个 SCSI 连接与服务器 60

相连。而服务器 60 又通过一个 LAN 与客户机 54 和 56 相连。无论客户机 54 还是客户机 56 发出查询，以及对该查询作出相应，服务器 60 都将完成数据库功能 750。然后服务器 60 将获得一份存储在物理装置 64 中的数据库拷贝 752，并通过对记录的分类找出那些符合客户机查询的记录。接着，服务器仅将那些符合查询的记录提供给客户机。这种现有技术结构的好处在于能将大量的排序和搜索工作留给服务器，而允许廉价的客户工作站从事查询处理工作。

图 7 所示本发明的另一实例，其中共享资源，即数据库受到任何一个与之直接相连的客户机的分布式管理。这种结构的优点在于不需要服务器介入到那些存有数据库的物理装置和正在访问该数据库的客户机之间。这种结构的缺点是要求每个客户机实现一个完整的数据库引擎。不管这样，在某些数据库应用中，当数据库包括一系列字段，每个字段又含有指针指向立体图象，动画及音频文件时，应用现有发明将显示其优越性。在本例中，任何一客户机上的处理时间与图象传输过程的相关性，比起要寻找被传输的图象声音文件或动画的指针而进行的实际查询及分类过程来说关系更加紧密。在这些情况下，不让服务器成为一个瓶颈是有好处的。还有，对该实施例的一种修改将会让多个传统服务器(带有与 LAN 相连的客户机)全部与共享存储器直接相连，这将允许有更高的数据可用性，因为所有的数据将存储在单一的区域供所有的服务器访问(因而供所有的客户机访问)。

图 7 的组成包括分别标记为客户机 A-B 的客户机 54-56，以及服务器 60 和存储装置 64。数据库 804 驻留在物理存储装置 64 中并包含了适用于实现本发明实例的附加的字段。这些附加字段将在以下的图 9A-B 中作详细描述。存储装置 64 是与每个客户机 A-B 直接相联的。网络连接例如可以是如图 2A 所描述和讨论的光纤信道。这种网络在存储装置 64 和每个客户机之间提供了一条带宽非常高的通信路径。这将允许不妨碍服务器而进行高速传送位映图象，动画或音频片断，就象服务器被打上删除标记 212 消失了一样。客户机 54-56 每个都包含了数据库分类索引的一个高速缓冲副本，分类索引 800 与客户机 54 相连，而分类索引 802 则与客户机 56 相连。这些索引要从存储在实际物理装置 64 中的主索引定期刷新。客户机 54-56 中的任何一个将分别完成增强型数据库处理 808-810。除了正常的数据库功能中的查询，配对及加锁，这些处理还包括记录的加锁，更新和卷控制功能，这将在以后详细讨论。数据库对个别记录和记录中字段的访问的相关性是借助客户驻留过程 808-810 与驻留在物理装置 64 中某一卷上的访问控制

表 206，卷控制表 208 和更新列表 806 之间的交互性而得以保持，卷是可供所有客户机访问读/写的。

图 8A 所示访问控制表 206 的详细数据结构。访问控制表与在图 3A 中描述及讨论的一样，这个访问控制表将在以下讨论的图 12F 所示处理过程中得以运用。

图 8B 所示为与存有数据 804 的可读卷 64 相关的卷控制表的详细数据结构。注意：在封锁字段 396A 中每个客户机 54-56 都被标注为逻辑假，意思是它们相对于该卷都拥有写特权。另外，客户机 A 和 B 在字段 394A 中都被标注为拥有进发写特权。

图 8C 所示为更新列表 806[见图 7]的详细数据结构。更新列表对于每个客户机字段 830，包含了与正在编辑的客户机的字段相关联的相应记录指针 832。增强型数据库过程 808-810[见图 7]将应用这个列表向所有正在阅读该相应记录的客户机广播对一个字段的更新内容。正如字段 832 所指出的，客户机 54 和 56 都正在阅读记录 2 中的字段。

图 9A 所示为关于图 7 讨论的数据库 804 的关系型数据库实施例 804A 相关联的详细数据结构。关系型数据库包括一个姓名表 804A-1，一个工作地址表 804A-2 和一个家庭地址表 804A-3。每个表包括多个字段，并且对于每个字段都有一个相应的锁定字段，在表 804A-1 中，每个记录有一个前缀字段和姓名字段，锁定字段 856 与每个前缀字段相关联，并且锁定字段 856 还与姓名字段相关联。在工作地址表中，每个记录都有街道字段和城市字段。与每个街道字段相关联的是相应的字段锁 862。与城市字段相关联的相应的字段锁 864。家庭地址表 804A-3 的每个记录包括城市字段和街道字段。街道字段有一个相应的锁定字段 868。城市字段有一个相应的锁定字段 870。与每个表中的每条记录相关联的是自由字段 860，866 和 872，它们将表明两种状态中的一种，活动的或是自由的。具有标记为活动状态的自由字段的一条记录包含着有效数据。而拥有标记是自由状态的自由字段的一条记录则没有数据，并且能被用于作为一条新的，增加的记录。一个标记为删除的自由字段则包括一条确定要从数据库中移去的记录，目的是要降低数据库的存储要求。从数据库中删除一条记录，可以把这条记录标记为自由，也可以把它标记为删除。向数据库增加一条记录可以通过把自由字段转变为活动字段达到，也可以采用物理方式向数据库中增加新的记录。与这些动作相关的处

理将在图 12A-12F 中详细讨论。

图 9B 所示为关于数据库 804 的平直数据库实现 804B 相关的详细数据结构。数据库 804B 由行组成，每行对应一个单独的记录。该数据库包含一个前缀字段，一个姓字段，一个家庭和工作地址字段以及一个城市字段。与前缀字段相关联的是一个锁定字段 880B。与姓字段相关联的是相应的锁定字段 882B。与每一个家庭/工作地址字段相关联的是相应锁定字段 884B。与每一个城市字段相关联的是相应的锁定字段 886B。与每一个记录相关联的是相应的自由字段 888B，其状态可能是活动的，自由的或是删除的。这些自由字段状态的作用与在图 9A 中讨论过的一样。

图 10 所示为每个客户机上用于管理分布式数据库的软件模块。客户机 54 和 RAID 存储器 64 被表示出来。RAID 存储器 64 包含一个数据库和锁定字段 804。客户机 54 包含运行时间监督器 908，事务例行程序 902，命令和路径截获 904，缓冲器管理器 908，锁定管理器 906 和锁定重定路由器 910。

在运行中一个查询是由运行时间监督器 900 接收并被送到事务例行程序模块 902。事务例行程序模块提出从存储媒体 64 截获记录的请求。该要求被命令或路径截获模块 904 截获。该命令路径截获模块在将请求继续传送到缓冲器管理器之前启动分布式管理功能，该功能将在图 12A-F 中详细讨论。缓冲器管理器 908 从位于存储器 64 中的数据库 804 中读出及写入数据。当一个事务请求被传送到命令和路径中断模块 904，它会同样被传送到锁定管理器 906。锁定管理器为需要的资源方发出锁定要求，该要求会被锁定重定路由器 910 所截获。锁定重定路由器 910 将负责完成字段锁定过程，这一点将在图 12C-F 中详细讨论。

图 11 所示为与分别位于客户机 54-56 上的分布式数据库过程 808-810 有关的总体处理流程。处理流程从过程 950 开始，在 950 中管理员将在一个特定的存储卷上建立一个数据库，该存储卷具有特定的字段和记录特性并允许多个写入者。数据库中的字段可以包含文本，数字数据，或是指向图象文件、音频文件及动画文件的指针。在较佳实例中，建立的数据库将具有多个用作将来用途的预定大小的多个空的/自由的、可用的记录。这样，新的记录可以被加入到数据库而无需要求改变物理存储器的空间。控制接着被派送到处理 952，在那里，一个 GUI 会被提供给用户去接受和处理该查询。控制接着被传送到过程 954，在那里可看到从数据库检索的记录。控制随后传送到子程序框 956，子程序框 956 对应于更详细

地示于图 12A 中的过程。控制随后传到判定框 958。在判定框 958，确定用户是否打算在正在观看的记录中编辑特定的字段作出。如果判定是否定的，则控制就传到判定框 966。另一方面，如果判定是肯定的，也就是字段需要编辑，控制就传递到子程序框 960。在子程序 960 中的过程将在图 12C 中详细描述。控制接着返回判定过程 962。在判定过程 962 中，确定有关用户何时已退出正被编辑的字段。当决定是肯定的，即用户已退出该字段，控制就传到子程序模块 964。子程序模块 964 对应于在图 12D 中更为详细的进程。执行该子程序之后控制返回判定过程 966。在判定过程 966 中，确定该用户是否已退出该记录。如果确定是否定的，控制就返回到判定过程 958，另一方面，如果判断是肯定的，即用户已退出记录，则控制就传到子程序框 968。子程序框 968 对应于图 12B 中详细描述的过程。从子程序 968 控制传递到判定过程 970。在判定过程 970 中，确定用户是否打算增加一个记录。如果答复是肯定的，控制传到子程序 972，子程序 972 对应于图 12F 中详细描述的过程。从子程序 972 返回，控制传递到删除判定过程 974。另一方面，如果在判定过程 970 中，确定是否定的，即用户不打算增加一条记录，接着控制就直接传到删除判定过程 974。在删除判定过程 974 中，确定用户是否打算删除一个记录，如果确定是否定的，控制返回到查询过程 952。另一方面，如果判定是肯定的，即用户打算删除一条记录，则控制就传到子程序模块 976。子程序模块 976 包含的进程将在以下的图 12E 中详细描述。

图 12A 概述了与图 11 所讨论的观察记录子程序模块 956 相连的过程，过程从连接框 A 开始并传送到截获过程 1000。在截获过程 1000 中，一个对应于被观看的记录的记录指针被命令路径截获模块 904[见图 10]所截获。控制传送到保留扇区过程 1002，在保留扇区过程 1002 中，相应于记录更新表 806[见图 8C]的存储器 64 中扇区将被保留。控制然后传到保留确认判定过程 1004。在保留确认判定过程 1004 中，确定该扇区是否已确实被保留。如果确定是肯定的，控制传到读扇区过程 1006。在读扇区过程 1006 中，包含更新表的扇区被读出并且获得用于表中特定记录的指针，该表具有的 I.D. 与运行该过程的客户机的 I.D. 一致。控制接着传到过程 1008。在过程 1008 中，从过程 1000 中得到的记录 I.D. 和/或者记录指针将写入相应于正在运行该过程的客户机的控制表中的字段。控制然后传到释放扇区过程 1010。在过程 1010 中，正被读着的扇区被释放。控制然后返回到连接框 B。

图 12B 所示为与上述图 11 所描述的子程序模块 968 相关的退出记录的过程。一个子程序从连接部分 C 开始，控制传到过程 1030。在过程 1030 中，一个用户正在退出的记录的记录指针被命令路径截获模块 904[见图 10]得到。控制接着被传送到保留扇区过程 1032。在这个过程中，存储器 64 上的包含更新表[见图 8C]的扇区被保留。控制接着传到保留确认判定过程 1034。在这个判定过程中，确定确认保留是否已被接受和完成，当得到肯定的确定时，控制被传送到读扇区过程 1036。在读扇区过程 1036 中，更新表被读出并且获得用于与执行该过程的客户机相应的那张表中的特定行的一个指针。控制接着被传送到过程 1038，在过程 1038 中，相应于运行该过程的客户机的记录 I.D. 字段被擦去。这表示用户不再观看该记录。控制然后传送到过程 1040。在过程 1040 中，保留扇被释放并且控制通过连接框 D 返回。

图 12C 所示为由子程序模块 960[参照图 11]执行的用于编辑字段的过。子程序从连接框 E 开始，且控制传送到过程 1070。在过程 1070 中，命令路径截获模块 904[见图 10]截获对应于被编辑字段的字段指针。控制接着传到过程 1072，在过程 1072 中，保留扇区请求是为物理存储装置 64[见图 10]中的该字段被存储的扇区提出的。控制然后传送到确认判定过程 1074。在确认判定过程 1074 中，确定被请求的扇区是否已确实被保留。如果确定是否定的，接着控制传到广播判定过程 1076。在广播判定过程 1076 中，确定另一个当前正保留该扇区的客户机是否已广播该扇区已被释放的信息。如果确定是肯定的，控制将返回保留扇区过程 1072。如果在判定过程 1074 中得到肯定的确定，控制接着传到过程 1078。在过程 1078 中，该保留扇区被读出。接着控制传到判定过程 1080。在判定过程 1080 中，确定正被请求有编辑权的有关字段是否确实自由的。这个确定是基于上述图 9A-B 中描述和讨论的锁定字段来作出的。如果确定结果是否定的，控制被传送到释放扇区过程 1082。在过程 1082 中该扇区被释放，并且控制返回到广播判定过程 1076。

另一方面，如果在判定过程 1080 中得到肯定回答，即该被请求的字段没有被锁定，接着控制被传送到过程 1084，在过程 1084 中，锁定重定路由器 910[见图 10]把客户机 I.D. 写入要被编辑的锁定字段。控制接着传到释放扇区过程 1086。在释放扇区过程 1086 中，被保存的扇区被释放，并且控制被传送到过程 1088。在过程 1088 中，用户编辑所选的字段而控制经连接框 F 返回。

图 12D 所示为子程序模块 964[见图 11]为退出编辑字段所完成的过程。结合图 11 中上面描述及讨论的子程序模块 964 对应于这些过程。子程序从连接框 G 开始，控制传到过程 1100。在过程 1100 中，被退出字段的指针和相应的记录的指针由命令路径截获模块 904[见图 10]所得到。控制接着被传送到过程 1102。在过程 1102 中物理存储装置上驻留有记录和字段的扇区被保留。控制接着被传送到保留确认判定过程 1104。在判定过程 1104 中，判定扇区是否确实被保留。如果该确定是否定的，控制被传送到广播判定 1106。在广播判定 1106 中，确定由哪个当前保留该扇区的客户机在什么时候释放该扇区。控制释放并随后转到过程 1102。另一方面，如果在判定过程 1104 中得到肯定确定，即该扇区已经被保留，接着控制就传到过程 1108。在过程 1108 中保留扇区被读出并且控制传到 1110。在过程 1110 中，锁定重定路由器 910[见图 10]擦去与退出字段相关联的块字段，并且控制传送到过程 1112。在过程 1112 中，命令路径截获模块 904[见图 10]把对被编辑字段的更新内容传送到物理存储装置 64。控制然后被传送到过程 1114。在过程 1114 中，更新表 806[见图 8C]被读出并按记录指针字段 832[见图 8C]进行分类。控制再传送到过程 1116，在那里观察与刚被更新的字段有关的记录相同的记录的客户机就被确定。控制接着传到过程 1118。在过程 1118 中，一条广播信息被发送到那些正在观察更新记录的客户机上。该信息包含一个对应于其字段刚被更新的记录的识别符。控制接着传到 1120。在过程 1120 中，保留的扇区被释放，而控制经连接框 H 返回。

图 12E 示出了与删除记录相关的以及与上述图 11 所描述及讨论的子程序模块 976 对应的详细处理过程。该过程从连接框 I 开始，控制被传送到过程 1150。在过程 1150 中，指向要被删除记录的指针由命令路径截获模块 904[见图 10]截获。控制接着传到过程 1152。在过程 1152 中，位于物理存储装置中与要被删除的记录有关的扇区被保留。控制就传到判定过程 1154。在判定过程 1154 中，确定要保留的扇区是否已确实被保留。如果确定是否定的，控制就传到判定过程 1156。在判定过程 1156 中，确定其它已保留该扇区的客户机什么时候释放该扇区。当得到肯定确定时，控制返回过程 1152。

另一方面，如果在判定过程 1154 中得到肯定确定，即该扇区已经被保留，则控制传到过程 1158。在过程 1158 中，锁定重定路由器 910[见图 10]读出物理存储装置上包含该记录的扇区。控制接着传到判定过程 1160，在那里，锁定重

定路由器确定该记录中的所有锁定字段是否都是自由的。如果确定是否定的，即所有字段都不自由，另一个客户机正在编辑其中的一个字段，控制便传到过程 1162。在过程 1162，保留扇区被释放且控制返回到广播判定过程 1156。

另一方面，如果在过程 1160 中得到肯定的确定，即待删除记录中的所有锁定字段是自由的，控制就传到过程 1164，在过程 1164 中，锁定重定路由器 910 把删除指示符写入自由字段[见图 9A-B]。控制接着传到过程 1166。在过程 1166 中，锁定重定路由器 916 擦除被删记录中的所有字段，控制就传到过程 1168。在过程 1168 中，该保留扇区被释放，控制则经连接框 J 返回。

图 12F 所示为与上述结合图 11 中的子程序模块 972 描述讨论的增加记录相关联的详细处理过程。控制从连接框 K 开始，接着传到过程 1180。在过程 1180 中命令路径截获模块 904[见图 10]截获添加事务请求，并把控制传到过程 1182。在过程 1182 中，数据库中所有记录的自由字段被分析，且控制被传到判定过程 1184。在判定过程 1184 中，确定与数据库每个现有记录相关的任何自由字段是否包含一个自由标识符。如果确定是否定的，即没有这样的记录，控制就传到过程 1186。在过程 1186 中，一个用于访问控制表的信号量字段 352[见图 8A]被抓到[见图 5A，过程 510-518]。接着控制传到过程 1188。在过程 118 中，为数据库保留的实际物理存储空间通过增加新记录而增加。那些新记录的自由字段被用自由标识符所标记。控制接着传到过程 1190。在过程 1190 中，控制表 206[见图 8A]的信号量字段 352 被释放。控制就返回到过程 1192。

如果在判定过程 1184 中得到肯定的确定，即数据库中存在自由记录，则控制传到过程 1192。在过程 1192 中，与自由记录有关的物理扇区保留。控制接着传到判定过程 1194，在判定过程 1194 中，要确定扇区是否实际被保留。如果确定是否定的，控制就传到判定过程 1196。在判定过程 1196 中，确定当前已对该扇区作了保留的客户机什么时候已释放了该保留。当这种确定是肯定的时候，控制返回到过程 1192。另一方面，如果在判定过程 1194 中得到肯定的回答，即那个扇区事实上已被保留，接着控制就传到过程 1198。在过程 1198 中包含自由记录的物理扇区被读出。控制接着被传送到过程 1200。在过程 1200 中活动标识符被写入记录的字段[见图 9A-B]。这表示该记录被投入使用。控制接着传到过程 1202，在过程 1202 中，被保留的扇区被释放，并且控制经连接 L 返回。



分布式处理

图 13A-C 所示为在网络环境中对多个处理器的分布式控制的总体格式。网络环境可被配置为 NAP 模式或是客户服务器模式或两者共存。图 13-A-C 包括了与网络 1300 相连的客户机 54-56 和 1302-1304。一系列的项目管理文件夹 1302-1304 可供网络上的所有客户机使用。每个客户机包含应用程序和分布式项目管理过程。客户机 54 包括分布式项目管理过程 1310，客户机 56 包括分布式项目管理过程 1312，客户机 1302 包括分布式项目管理过程 1306，而客户机 1304 则包括分布式项目管理过程 1308。该分布式项目管理过程每一个都具有选择一个项目管理文件夹以启动项目管理所选部分的处理并将所得结果文件存入装配位置的能力。这样，一个复杂的项目，诸如涉及图象编辑、再现和关键格式的转换的多媒体表现，可以通过网络环境中多个客户机的合作完成。

在图 13B 中，客户机 54-56 和 1302 每个都分别获得项目管理文件夹 1314 的相应拷贝 1314A-C。每个客户机选择了项目管理文件[PMF]1314 定义的项目的一部分。项目管理过程 1310-1312 和 1306，在与 PMF1314 中所选部分有关的处理中分别把应用程序引入到它们相应的客户机上。当驻留应用程序完成所要求的任务，其结果被存入文件。

图 13C 所示客户机 54-56 和 1302 已经释放了它们的 PMF1314 的副本，并且，驻留过程 1310-12 和 1306 已经把本地应用程序生成的工作成果存储到网络上。客户机 54 把它的工作成果存入网络上的 1330 文件中，用于示例目的已图示地描述为图像帧的左手部分 1500。客户机 56 把它的工作成果存入网络上的 1332 文件中，图示地描述为图像帧的中间部分 1500。客户机 1302 把它的工作成果存入网络上的 1334 文件中，图示地描述为图像框架的右手部分 1504。这些文件中每一个在网络上的位置都被记录在 PME 1314 中。

用于过程起始的下一个 PMF 是 PMF 1318。客户机 1302-1304 已分别获得 PMF 1318 的相应副本 1318C-D。这些客户机也将签名(sign on)加入这些 PMF 的某些部分的处理。

图 14A-D 所示与项目管理文件有关的数据结构。图 14A 所示为与项目管理文件夹对应的数据结构。所示项目管理者 1314 包括一个项目文件 1450，一个容量表 1350 和处理选项模块 1400A-B。

图 14B 所示为与容量表 1350 有关的数据结构。容量表 1350 包括了用于网络

上的每一个客户机的一行/记录。对于每一个客户机记录应用可用性，字段 1356-1362 记录了在该客户机上的应用程序的可用性。对于每一个这些应用程序，都会生成一个表示由应用程序耗用的处理带宽百分比的指示符。例如，客户机 A[54] 用一个电子图像再现包 1358 表现出消耗了其处理容量的 25 %。一个 CAD 包 1360 则显示在客户机 A[54] 上它消耗了其处理容量的 10 %。再现包 1362 则显示它消耗了客户机 A[54] 的处理带宽的 25 %。最后，其余未指定的应用程序和操作系统将消耗客户机 A 的带宽的又一个 10 %。另外，文字处理包在字段 1356 中列为 N/A 表示该应用程序甚至没有驻留在客户机 A[54] 上。字段 1352 为每一个客户机且在本例中具体说是为客户机 A[54] 在字段 1356-1361 中记录了输入使用的累计值。这样，客户机 A 有一个标明占其总处理带宽 70 % 的使用 1352。而容量表中的其余字段则是以每秒百万条命令计算(IMPS)的处理速度 1354。客户机 A[52] 正如列出的那样有 266 MIPS 的处理能力。网络上的所有客户机都能获得相应的信息。容量表中的每一条记录由分布式项目管理过程 1306-1312 中相应的一个来保持及更新。这些过程使操作系统的调用被生成。这些调用返回的变量对应于根据特定应用的基础分解的利用百分率。该过程然后把这些更新结果写入控制表中与它们的记录相关的对应字段中。这样，在一个公共位置上单个表被用于记录所有网络资源的百分比利用率。

图 14C 所示为对应于处理选项模板 1400A-B 的详细数据结构。典型地，一个处理选项模板与一个特定的应用程序相关联。一个处理选项模板会造成一个应用程序驻留在一个以特定方式执行的客户机上。例如，处理选项模板 1400A 与电子图像(Electric Image®)软件应用程序相关联。字段 1402A-1408A 分别记录着这个应用程序的合适的帧大小、压缩、格式、每秒帧数以及持续期指示符。在图 14C 所示的例子中，示出了大小为 640 × 480 的一帧。标明为 JPEG 压缩技术 1404A。以每秒帧数表达的回放速率 1406A 标示在 15，最后，动画的持续期 1408A 是 2 秒。

处理选项模板 1400B 与计算机辅助设计(CAD)应用程序相关。对于该应用程序，帧大小压缩技术和位深度优先分别标记在字段 1402B-1408B 中。X 轴帧大小是 24。Y 轴帧尺寸是 12，压缩技术是 TIFF，而位分辨率是 8 字节。每个处理选项模板都可以用许多方式中的任何一种方式生成。处理选项模板可以由用户生成，该用户使应用程序工作、选择较佳的运算参数并把这些参数分类到相关的输

出文件。另一方面，处理选项模板也可以管理方式生成并附加到就用于这样目的的外壳程序中的特定文件中。

图 14D 所示为项目文件 1450 的详细数据结构。项目文件包括一个具有签名字段 1454 的有标头和主优先字段 1452。签名字段标明在网络上的众多客户机中间哪些已对项目签名。客户机 A，B，C 都显示在该项目上已有签名，而客户机 D 没有。没有一个客户机已经被要求或是被指定处于主机状态，因为在主机状态字段 1452 中的所有输入都为零。

项目文件 1450 包含一个资源请求部分。这个部分包含着为完成该项目所需应用程序的名称。另外，可执行路径字段 1456 为该应用程序命名了可执行文件。源数据文件字段 1458 指出哪些文件必须被每个应用程序所加载。这个特定的项目需要文字处理应用，电子图像应用，CAD 应用和一个再现应用。电子图像应用需要一个名为“center.vsd”的源数据文件和一个标为 c:\...\center.vsd 的指向该文件的路径。该项目文件的第二部分是片段部分。两个片段部分被显示出。一般，这些部分的数量对应于网络上的客户机数量，这些客户机能同时从事与特定项目相关的处理。一些项目将有一个片段，然而其它能被分割到许多客户机之间的复杂的再现项目将有大量的片段。

每一个片段包含一个片段定义字段 1460，一个程序项字段 1462，一个存储路径字段 1464，一个优先级字段 1466，一个被处理字段 1468 和一个片段状态字段 1470。

片段定义字段 1460 限定了一个特定片段的起点和终点。在第一片段中的电子图像应用的起点和终点列出为帧 1-2。程序控制字段 1462 包含控制脚本，例如，Visual Basic[®]，Java[™] 或是“DDE Script”，它们使应用程序以适合于该特定项目的方式去执行。注意：“application”这个单字包括了运行在计算机上的从 AUTO-CAD[®] 那样的高级程序到包括 OS 在内的低级应用的任何处理过程。存储路径字段 1464 指出结果将存放在哪里[见图 13C 的 1330-1334]。字段 1466 决定了项目中应用程序和片段的处理次序。被处理字段 1468 指示了网络上众多客户机中的哪一个已经使其本地驻留程序去执行 PMF 的一个片段。例如，在片段 1 中，“a”标识符显示在客户机 A 上的项目管理过程已经着手电子图像任务的片段 1 的处理。最后，片段状态字段 1470 指出对于一个项目文件中的每一片段和每一个应用程序，该片段的状态是什么。可能的输入是“0”，“1”或者“2”。



0 输入表示该片段还未被处理，“1”输入表示该片段正在处理，而“2”输入表示片段处理已经完成，并且作为处理结果生成的文件已经被存入网络上的存储路径位置。项目文件的下一部分是装配部分，它包含字段 1472-1478。字段 1472 是一个 I.D. 字段指明众多客户机中的哪一个已经完成最后的装配。字段 1474 包含与执行该任务相关的程序代码或是 DDE 序列。字段 1478 则指出网络上存放结果的位置。项目文件的下一部分是过程选项模板字段 1480。这个字段列出每个应用程序的一个相应过程选项模板或过程选项模板路径。

图 15A-B 所示为分布式项目的两个例子。图 15A 显示了一个二维图像，其中将被再现的图像划为三部分，每个部分在一个独立的客户机上再现。部分 1500 是图像的左边部分，部分 1502 是图像的中间部分，而部分 1504 是图像的右边部分。每一部分都存储在网络上的相应文件 1330-1334 中。

图 15B 是一个含有六帧的动画序列。前两帧 1522 是由一个客户机创建的并存储在网络上的文件 1330 中。下一部分 1524 包含动画片的帧 3 和帧 4，并且存储在网络上的文件 1332。项目的最后部分 1526 是帧 5 和帧 6，则存放在网络上的文件 1334 中。当文件 1330-1334 在最后装配期间被组合后完整的动画顺序 1520 就制作出来了。

图 16A-D 所示为一个客户机驻留应用程序 1550，它在一个分布式项目管理过程的指令下打开一个源数据文件[见图 16A-B]，处理一个文件[图 16C]和保存该文件[图 16D]。与一片段相关的控制脚本 1462[图 14D]使该应用程序以这个方式执行，该代码被分布式管理过程卸载并被这些过程用来使该应用程序按预置方式执行。在图 16A 中显示了用于应用程序的 GUI 1550。叠加在 GUI 上的是一个文件打开 GUI 1554。文件打开 GUI 的启动工作可通过程序代码脚本，去完成在文件菜单 1552 中出现的文件打开命令。在 GUI 操作器 1554 中是一个预览窗口 1556，从中可显示被选中文件 1558 的内容。在图 16B 中，与文件 1558 相关的图像 1560 在 GUI 1550 中被展示。

在图 16C 中项目管理过程发出可执行代码的下一行到 CAD 应用程序, 结果使阴影操作从工具目录 1562 中被选中。作为这个选择的结果, CAD 应用程序对图像 1560 的一部分 1564 打阴影并显示。在图 16D 中下一行代码是由分布式管理过程递给应用程序。收到下一行代码后, 便使应用程序去执行从菜单树 1552 上找到的文件“另存为”功能。作为响应, 一个“另存为”GUI 界面 1566 被启动。

而名为“center.vsd”的文件被放到文件名窗口。接着，与刚被处理的片段对应的文件被存入网络。

图 17A-F 显示了各种分布式项目管理过程的处理流程图表。

图 17A 所示为创建项目过程。从初始框开始，控制传到过程 1652，在过程 1652 中，一个项目文件被创建，在一个较佳实施例中，用户被显示一个 GUI 界面，与图 14D 显示的并没有不同。使用该模板他们为项目文件输入各种参数。控制接着传到过程 1654。在过程 1654 中，过程选项模板被创建。在一个较佳实施例中，这些也可由用户通过类似于图 14C 中所示的 GUI 界面创建。控制接着传到过程 1656。在过程 1656 中，该过程管理文件和相应的过程选项模板被存储在网络上。控制随后传到连接框 A。

图 17B 示出了与维持图 14B 所示利用/容量表 1350 相关的处理过程。如上所描述和讨论的，分布式管理过程 1302，1304，1310 和 1312[见图 13A-C]独立地执行这些处理步骤。从连接框 A 开始，控制传到过程 1680。一个功能调用被发送到 OS 去取得由应用分解的利用信息。控制接着被传送到过程 1682。在过程 1682 中，利用率表被发现并读取。控制接着被传送到过程 1684。在过程 1684 中，利用率表中与运行该过程的客户机对应的行记录被识别。控制接着传到过程 1686。在过程 1686 中，从上面处理 1680 中得到的值被放入利用率表中的适当字段内。控制接着被传送到过程 1688。在过程 1688 中，送入利用率表的客户机处理速度被从表中读出。控制接着传到过程 1690。在过程 1690 中，功能调用被发送到 OS 以取得该客户机的处理速度。控制然后被传送到判定过程 1692。在判定过程 1692 中，确定表中的输入是否对应于正在运行该过程的客户机的真正处理速度。如果两者不匹配，那么控制被传送到过程 1694，在那里表中的输入被更新以反映该客户机的现行处理能力。控制接着传到连接框 B。另一方面，如果在判定过程 1692 中得到肯定的确定，即对该客户机而言该利用率表中的输入是正确的，那么控制就直接传到连接框 B。

图 17C 所示为当主/从关系被启用后执行的分布式管理过程部分。对于分布式处理的管理基本上有两种操作模式。在主/从模式中，无论预置为主机状态或是先取得 PMF 再取得主机状态的客户机都支配着项目文件所有片段的后续处理过程。在另一实施例中，实行对等分布式处理方案。在该对等处理方案中每个客户机要不断寻找一个未完成的 PMF 来处理。

图 17C 中，处理从连接框 B 开始，下一个 PMF 从先进先出(FIFO)队列中被选出，并且对该 PMF 的处理持续到文件中的所有片段完成。以及最后装配被执行。在另一个较佳实施例中，下一个 PMF 将被定义为具有未完成片段的任何文件或尚未最后被装配的任何文件。控制接着传到过程 1722，在该过程中被选定的 PMF 被读出。控制随后传到判定过程 1724。在判定过程 1724 中，确定该特定项目管理文件夹是否能被设置为主/从状态工作。如果确定是否定的，控制传到连接框 D。另一方面，如果确定是肯定的，则控制就传到判定过程 1726。在判定过程 1726 中，根据主机字段 1452[见图 14D]，确定是否有任何主机已对该项目管理文件夹发出开始工作信号(sign on)。如果确定是肯定的，则控制传到连接框 C。另一方面，如果确定是否定的，控制并进到判定过程 1728。在判定过程 1728 中，确定在主机状态字段 1452[图 14D]中被指定为主机状态的客户机是否拥有一个与我方(即运行该过程的客户机)I.D.对应的 I.D.。如果判定是否定的，则控制传到连接框 C。另一方面，如果确定是肯定的，则控制传到处理 1730。在过程 1730 中，客户机发出开始作为接受主机状态的信号，并通过在项目管理文件夹中的适当字段 1454 写入一个签名标识符的方式完成。控制接着传到过程 1732。在过程 1732 中，利用表被读取以确定从机的可用性和处理容量。控制接着传送到过程 1734，在过程 1734 中，取得片段状态字段 1470[图 14D]中尚未标有“2”的且具有最高优先权的片段记录，并确定该片段所要求的应用程序 1456。控制接着传送到过程 1736。

在过程 1736 中该主机从事于从机请求和/或从机分配处理。只有那些拥有本地驻留应用程序并与该片段的应用程序相匹配的从机被选中。另外，在过程 1732 中确定的拥有更多可用处理能力以及较小利用率的从机，将优先于那些具有低处理速率和有限容量的从机。控制接着传到判定过程 1738。在判定过程 1738 中，确定从机是否已接受从机状态。对那些已接受从机状态的从机，控制传到过程 1740，在那里从机 I.D.被确定。控制接着传到过程 1742。在过程 1742 中，基于那个 PMF 片段，该从机被指派去处理的信息由主机发送给从机。该信息包括：从机从事该处理将需要的指向源文件的指针，指示哪个片段要被处理的片段定义，使从机能启用它的驻留应用程序的可执行代码序列以及过程选项模板。最后，主机向从机发送一个从机存储其工作成果的网络位置指针。控制接着传到判定过程 1744。在判定过程 1744 中，确定是否存在任何未完成的其余片段，如果

存在，控制就返回到判定过程 1738 去进行下一个从机的选择。如果没有未完成的其余项目，则控制传到判定过程 1746。在判定过程 1746 中，确定是否所有从机都已完成它们的工作。完成工作是根据项目文件中的从机的输入，具体说是根据片段状态字段 1470[见图 4D]予以确定的。当确定是肯定的，即所有片段都已被处理，则控制传到过程 1748。在过程 1748 中，主机按照包含在项目文件[见图 14D]中的装配部分中的指令完成了项目文件片段的最后装配。控制接着传到过程 1750。在过程 1750 中，主机在 PMF 中，具体说是字段 1472 中写入一个状态标识符，表明最后装配已经完成并且这个项目管理文件夹可以被关闭。控制然后传到连接部分 A。

图 17D 所示为由从机所从事处理过程。处理由连接框 C 开始，接着控制传到判定过程 1780。在判定过程 1780 中，确定主机请求是否被收到。如果确定是否定的，控制传到连接框 B。另一方面，如果确定是肯定的，那么控制传到过程 1782。在过程 1782 中，该从机向主机发出一个肯定的响应，表示它们已接受从机状态。控制接着传到过程 1784，在那里从机从主机那里接收处理其被指派的项目文件片段对它所必须的信息。控制接着传到过程 1786，在过程 1786 中，从机使指定的应用程序运行并用适当的控制脚本和过程选项模板控制该应用程序。这使得该应用程序执行其指定的功能。控制接着传送到过程 1786。在过程 1786 中，从机把状态“1”标识符写入该项目文件[见图 14D]中相应的片段状态字段 1470。这向任何读该项目文件的客户机表示该片段当前正被从机处理。控制接着传到判定过程 1788。在那里，从机确定该驻留应用程序是否已完成它的项目片段。当确定是肯定的时候，控制被传送到过程 1790。在过程 1790 中，使该应用程序把它的工作成果存入处理 1784 指定的位置，并关闭该应用。控制接着传到过程 1800。在过程 1800 中，一个片段完成标识符“2”被写入项目文件中的适当的字段 1470。控制然后传到连接框 B。

图 17E 显示了与客户机之间对等关系有关的而不是与主从关系有关的分布式项目处理部分。处理从连接框 D 开始，控制接着传到过程 1820，在那里下一个 PMF 被读取。控制接着传到过程 1822，在那里利用率表被读取，控制接着传到过程 1824，在那里 PMF 中的片被优先权分类，并确定哪一片段还未完成。该判断是基于上面提及并讨论的优先权 1466 和片段状态 1470 字段[见图 14D]。控制接着传到判定过程 1826，在判定过程 1826 中，确定是否还有任何未完成的其

余片段。如果确定是否定的，则控制传到连接框 E。另一方面，如果得到肯定的确定，即还有其余的片段待处理，则控制就传到过程 1828。在过程 1828 中，确定与下一个优先化片段有关的应用程序是否能在运行该处理的客户机上得到，控制接着到达判定 1830，在那里进行该确定。

在过程 1736 中主机从事于从机请求和/或是从机分配处理。只有那些具有本地驻留应用程序且它们与用于该片段的应用程序 1456 相匹配的从机被选中。另外在一个较佳实施例中，正如由过程 1732 中确定的具有更多可用处理能力和较小利用率的从机将优先于那些具有较低处理速度和有限容量的从机。如果运行该过程的客户机没有被调用的应用程序，控制传到连接框 B。另一方面，如果得到肯定的确定，即客户机拥有必需的驻留应用程序，那么控制传到过程 1832。在过程 1832 中，客户机发出信号对项目文件中的一个片段开始工作。一个客户机通过在该项目文件[见图 14D]中的适当片段状态字段 1470 里写入状态“1”标识符来做到这一点。控制接着传到过程 1834。在过程 1834 中，客户机从处理文件中读取合适的片段信息，包括一个指向源文件的指针，片段定义，应用程序处理代码，过程选项模板和/或是指向过程选项模板的路径说明，以及一个用于存放工作成果的指定位置。控制接着传到过程 1836。在过程 1836 中，客户机利用程序代码序列和从项目文件提供给它的过程选项模板使它们的驻留的应用程序能工作。控制接着传到过程 1838，在那里已完成项目的应用程序存储它的工作成果。该存储位置对应于指定的源路径 1464[见图 14D]。控制接着传到过程 1840。在过程 1840 中，状态字段 1470 中适当的一个被更新为状态“2”，以表示在项目文件中该特定片段已经完成。控制接着传到连接框 B。

图 17F 显示了与所有片段已完成的一个项目文件的再装配有关的以对等关系从事的那些过程。过程从连接框 E 开始，然后进到过程 1860。在过程 1860 中，项目管理文件夹被读取。控制接着传到判定过程 1862。在判定过程 1862 中，根据项目文件[见图 14D]中的最后装配字段 1472 确定 PMF 是否已经被装配。如果确定是肯定的，控制返回到连接框 B。另一方面，如果得到的是否定的确定，即 PMF 还没有受到最后装配，那么控制传到过程 1864。在过程 1864 中，将按照存储在 PMF 中的装配指令进行最后的装配。控制然后传到过程 1866。在过程 1866 中，适当的最后装配字段被标上一个状态“2”，表明最后装配已完成，控制就返回到连接框 B。

应用程序内调用

Distribute_Process(POT_File_Pathname, Project_File_Pathname)

来自	到	说明
MPA	MSB	
MPCA	MSB	
Launch_Process		
来自	到	说明
SSB	SPA	
MSB	SPA	如果需要主机处理功效的话 (尚未打开)

Process_Segment (PMF_File_Pathname, Plugin_ID, Timestamp, Segment_Parameters)

来自	到	说明
SBPI	SPA	
SBPI	MPA	如果需要主机处理功效的话

Processing_Segment (PMF_File_Pathname, Plugin_ID, Timestamp, Segment_Parameters)

来自	到	说明
SPA	SSB	指示被请求段的处理已实际开始
MPA	MSB	指示同上，用于需要主机处理功效的场合下

Segment_Done (PMF_File_Pathname, Plugin_ID, Timestamp, Segment_Parameters)

来自	到	说明
SPA	SSB	指示被请求段的处理完毕
MPA	MSB	指示同上，用于需要主机处理功效的场合

Quit_Process

来自	到	说明
----	---	----

SSB	SPA	
MSB	SPA	如果主机处理功效被用过 (开始时未打开)

Assemble_Process(PMF_File_Pathname, Plugin_ID, Timestamp)

来自	到	说明
SBPI	SPA	当插件本身不能装配结果时的任选装配方法
SBPI	MSB	同上，用于需要主机处理功效的场合

Assemble_Done (PMF_File_Pathname, Plugin_ID, Timestamp)

来自	到	说明
SPA	SSB	如在处理帮助下装配完毕，指示完成
MPA	MSB	同上，用于需要主机处理功效的场合

Process_Complete (POT_File_Pathname, Project_File_Pathname)

来自	到	说明
MSB	MPA	指示整个处理完成
MSB	MPCA	同上

Studio BOSS 到 StudioBOSS 调用

用于过程控制

SB_Processing_Segment (PMF_File_Pathname, PMF_Reference)

来自	到	说明
MSB	SSB	

SB_Process_Segment (PMF_Reference)

来自	到	说明
SSB	MSB	指示被请求段的处理实际已开始

SB_Segment_Done (PMF_Reference)

来自	到	说明
SSB	MSB	指示被请求段的处理完毕

SB_Quit_Process (PMF_Reference)

来自	到	说明
MSB	SSB	过程段完毕后被调用来清除处理

SB_Assemble_Process (PMF_Reference)

来自	到	说明
MSB	SSB	被调用来装配过程结果 (当插件/应用在 MSB 上不能使用)

用于仲裁主/从控制

SB_Become_Slave (PMF_File_Pathname, PMF_Reference)

来自	到	说明
MSB	SSB	被调用成为从机

SB_Slave_responsee (PMF_File_Pathname, PMF_Reference, Response)

来自	到	说明
SSB	MSB	响应(真/假)是否它将是用于该过程的从机

SB_Become_Free

来自	到	说明
MSB	SSB	释放从机

插件调用

PI_Process_Segment (PMF_File_Pathname)

被调用给分布式过程段，造成 Process_Segment IAC 调用。

PI_Assemble_Process (PMF_File_Pathname)

被调用来装配过程结果。该插件或者可以不用任何过程应用交互作用而装配结果，或者它可以作为任选项发出 Assemble_Process IAC 调用去装配每样东西。

符号表

MPA	主进程应用
MPCA	多进程控制器应用
PI	插件
PMF	过程管理文件
SB	Studio BOss
MSB	主 Studio BOSS
SSB	从 Studio BOSS
POT	过程选项模板
SBPI	在 MSB 或 SSB 内存活的 Studio BOSS 插件

过程选项模板(POT)

模板_签名

(应用特定的选项)

POT 示例，内存中装配的结果-用于基于帧的过程

模板_签名

位_深度

帧_大小_X

帧_大小_Y

POT 示例，文件中装配的结果-用于基于时间的过程

模板_签名

帧_大小

帧_每_秒

压缩算子

起始_时间

结束_时间

PMF

项目_文件_路径

其余_段_到_过程

过程_应用_签名

主导主机 ID

装配_方法

数量_关于_从机

(对于每个从机, 如下)

从机_ID

段_被_处理

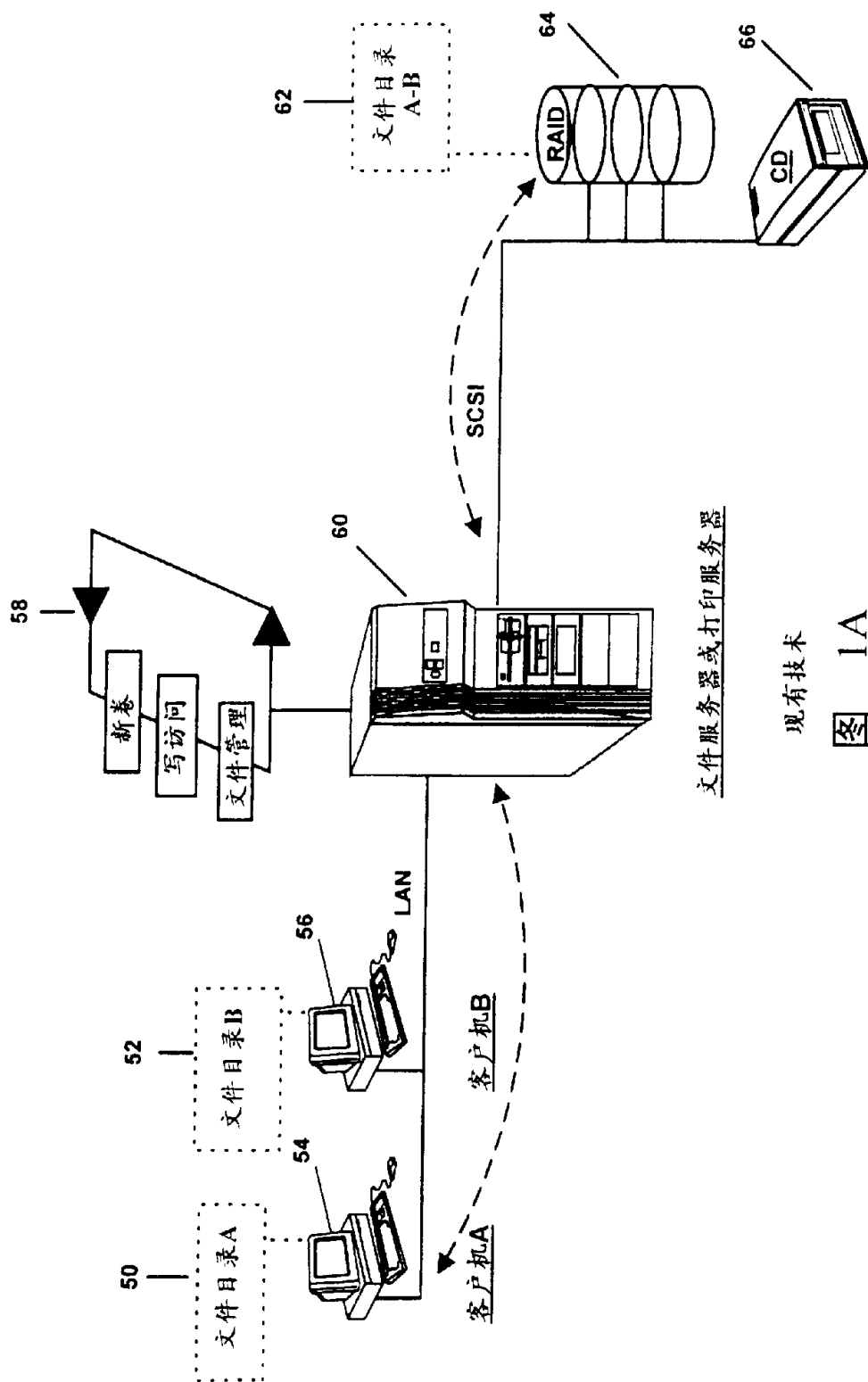
段_状态(不_处理, 处理_正在_进行, 处理_完成)

结果_位置

过程_选项_模板

前面对本发明的一个较佳实施例的描述是为了解释和描述的目的。它并不试图一一列举或把本发明局限于已公布的精确形式。显然许多修改和改变对本领域的熟练实际工写入者来说是明白的。因此本发明的范围将由下述权利要求书及其等效文件所限定。

说明书附图



文件服务器或打印服务器

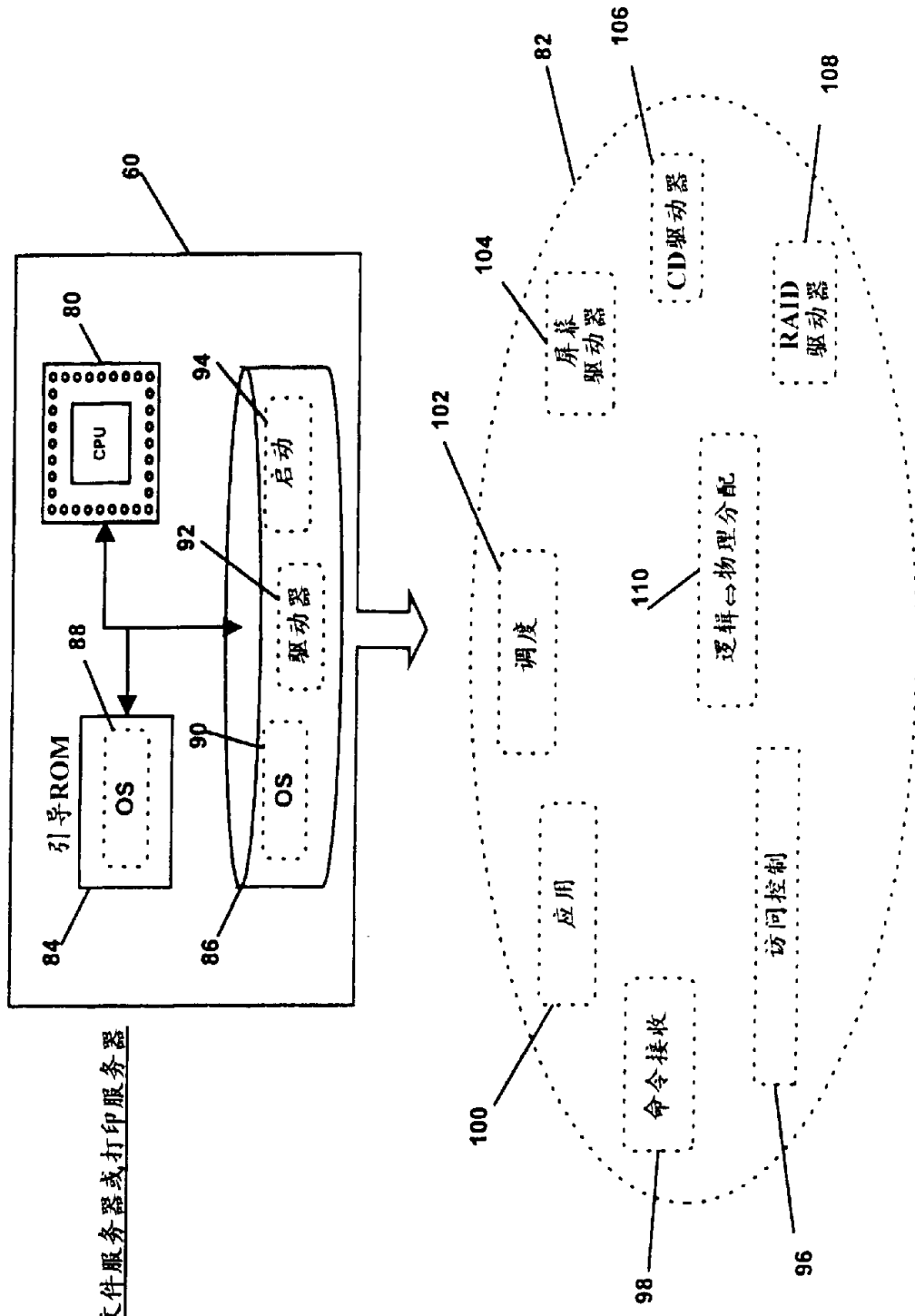


图 1B

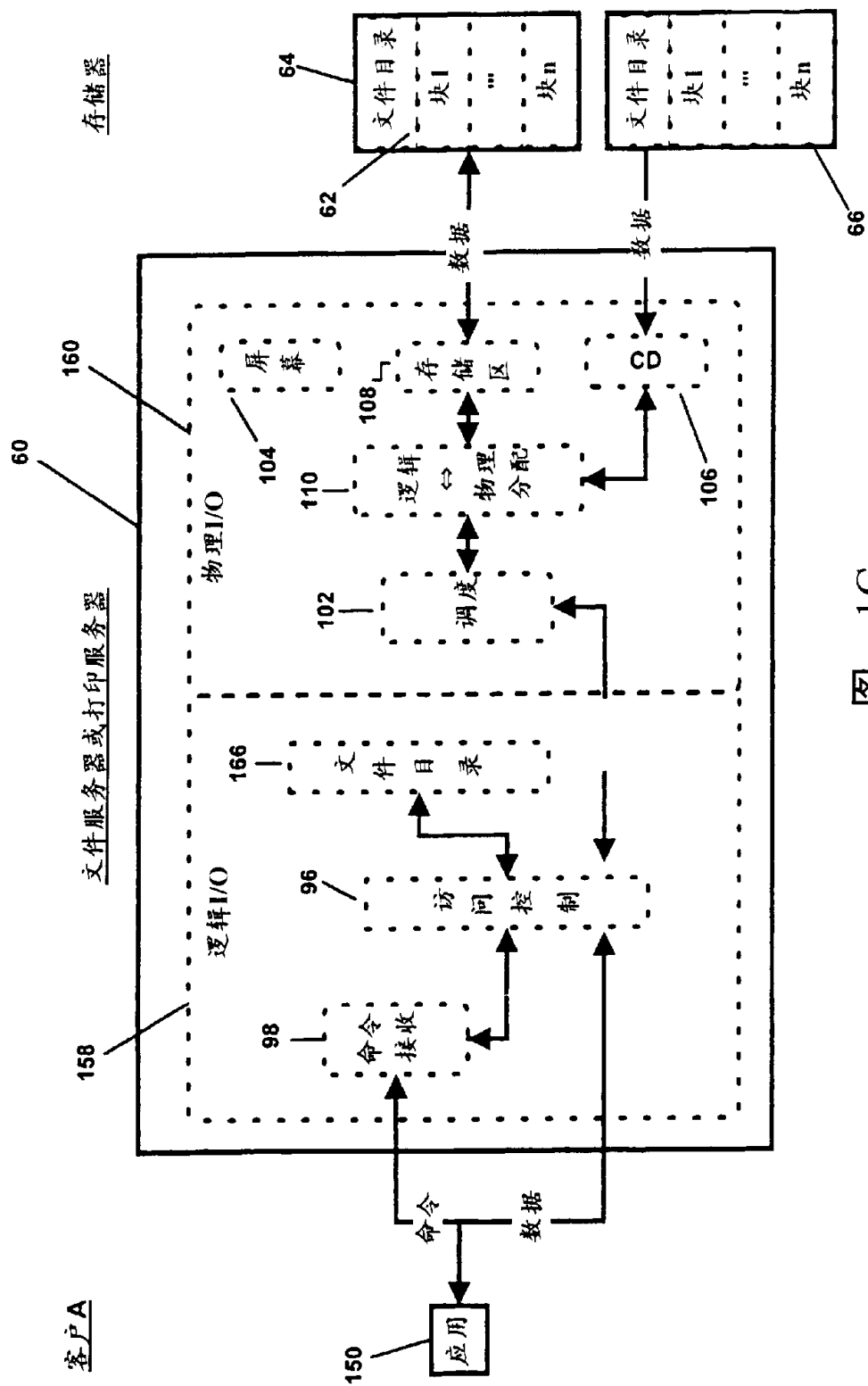


图 1C

文件服务器或打印服务器

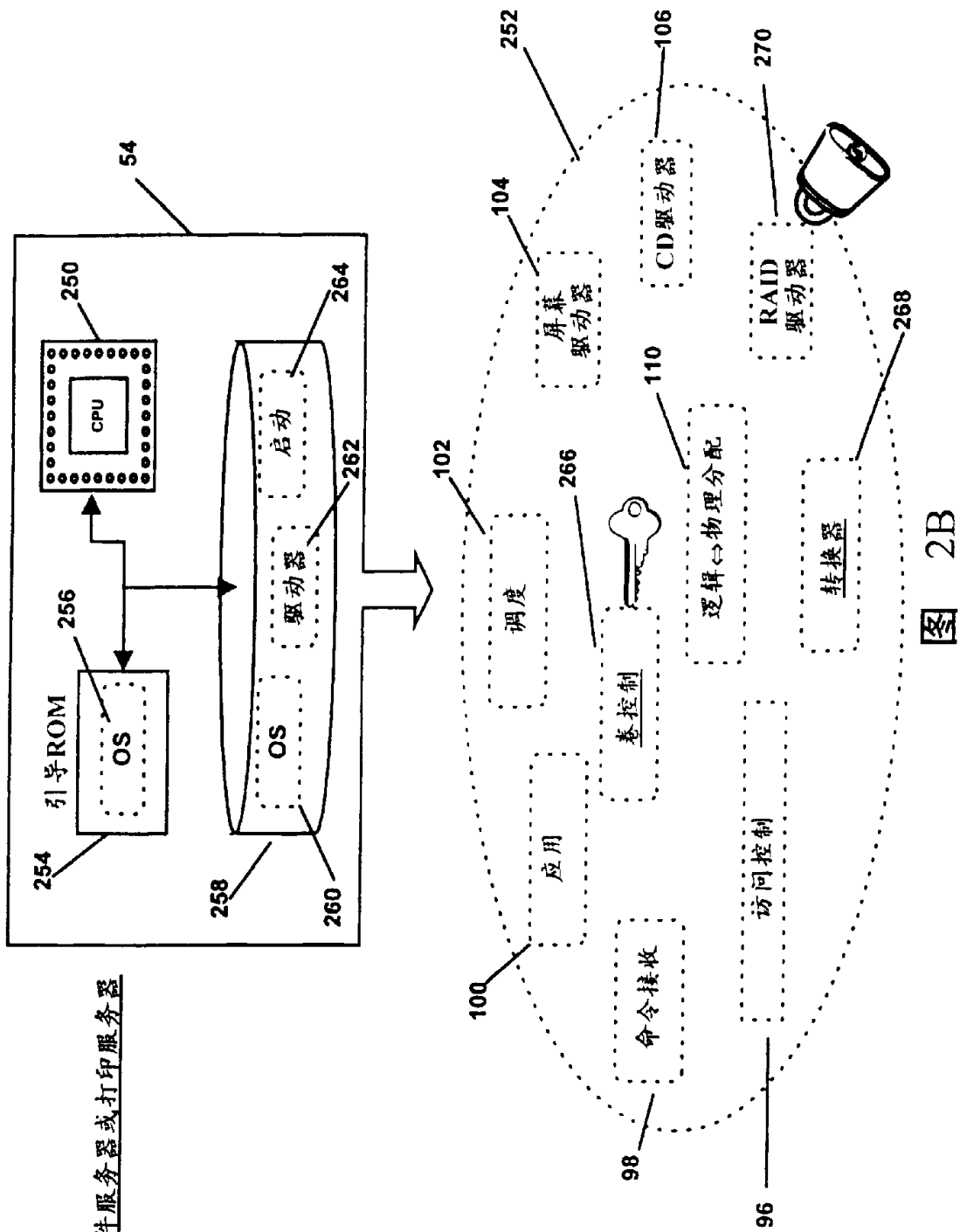


图 2B

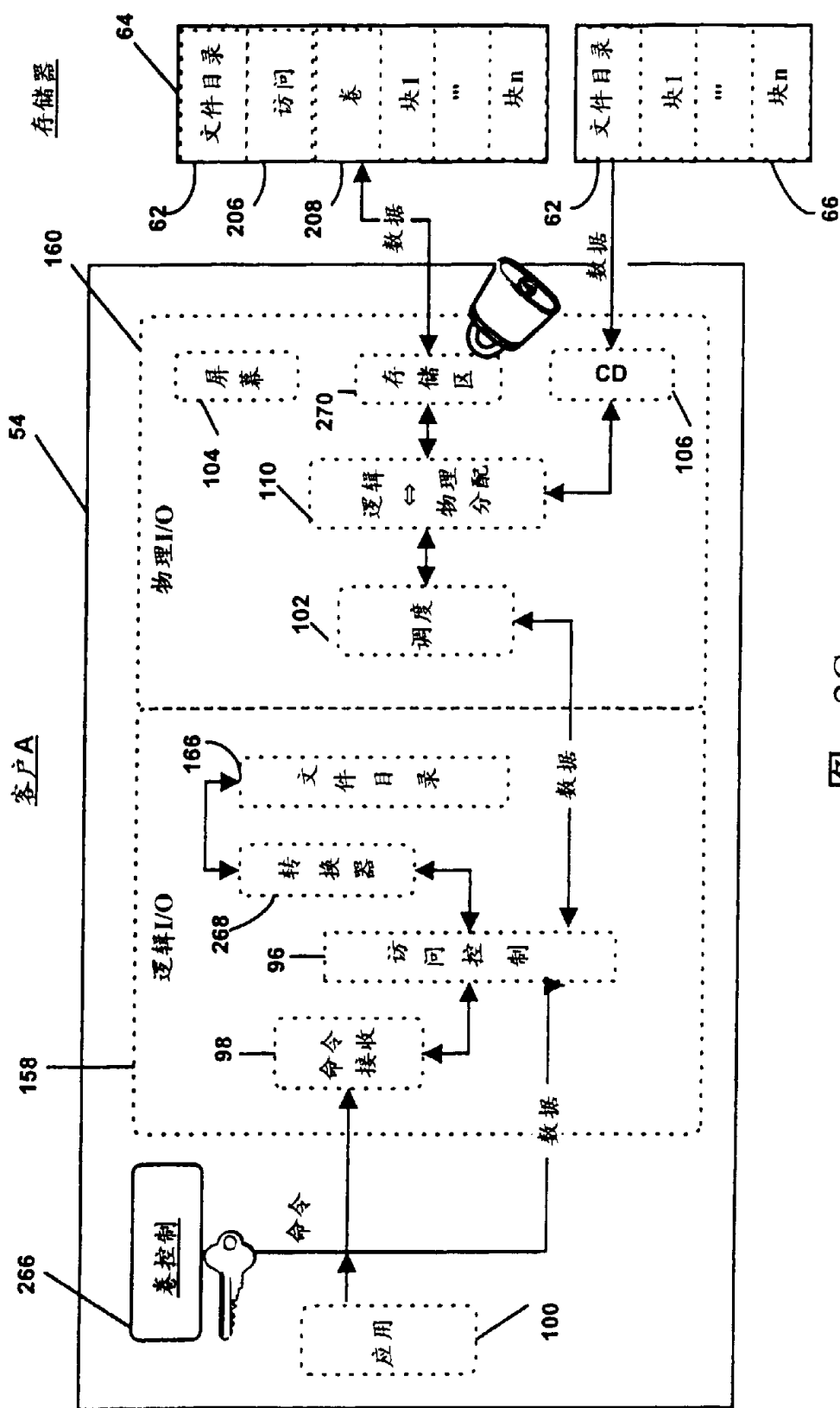


图 2C

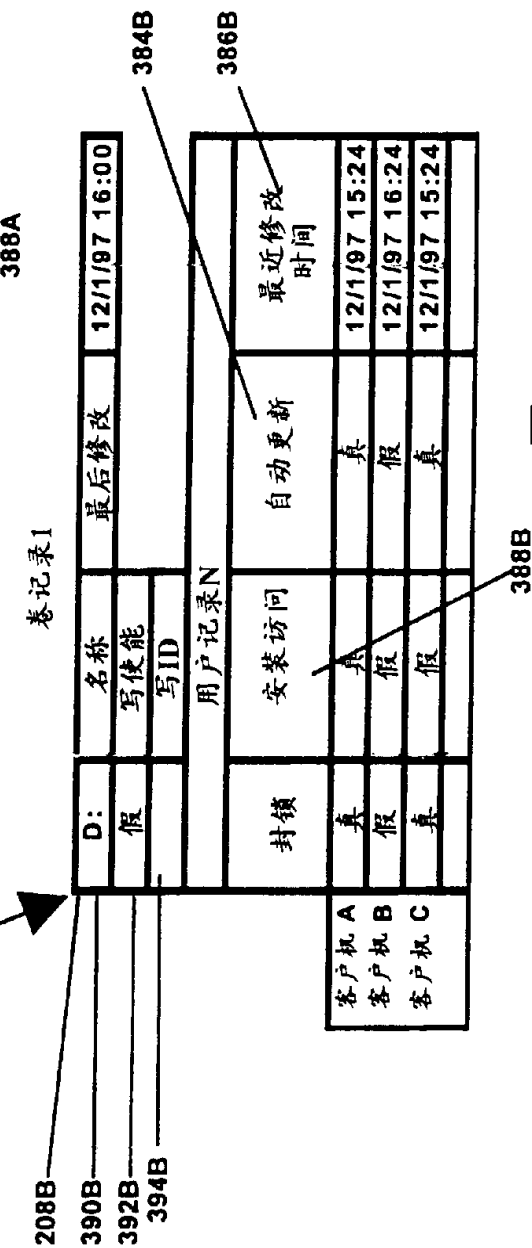
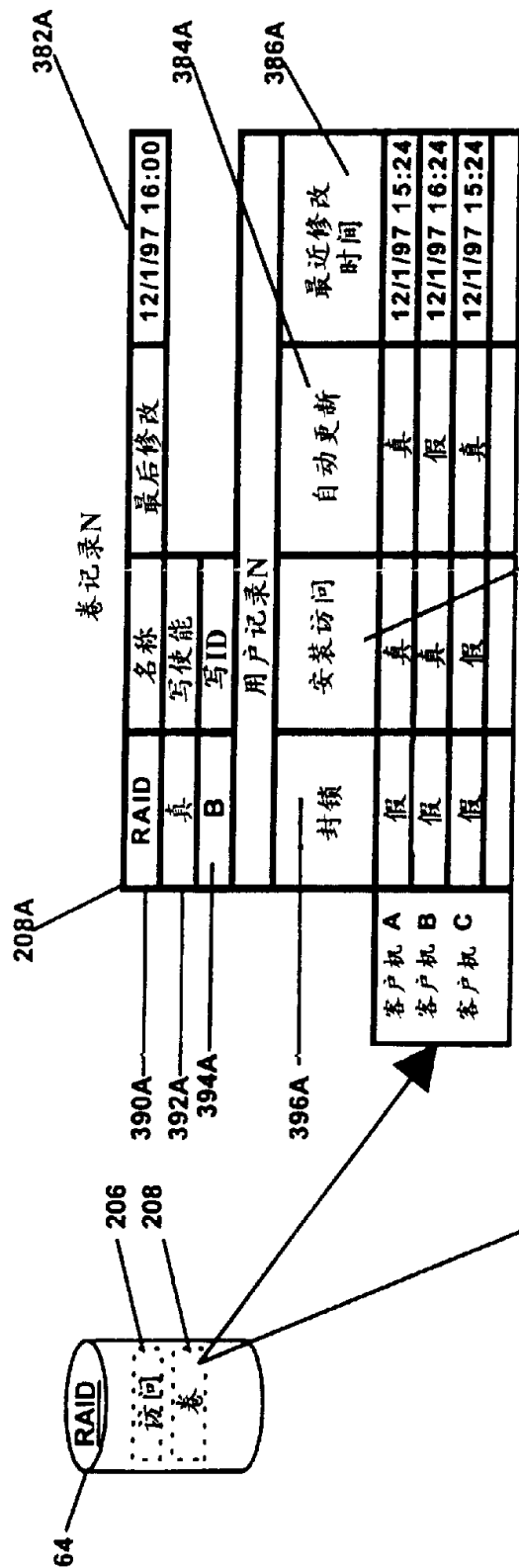


图 3B

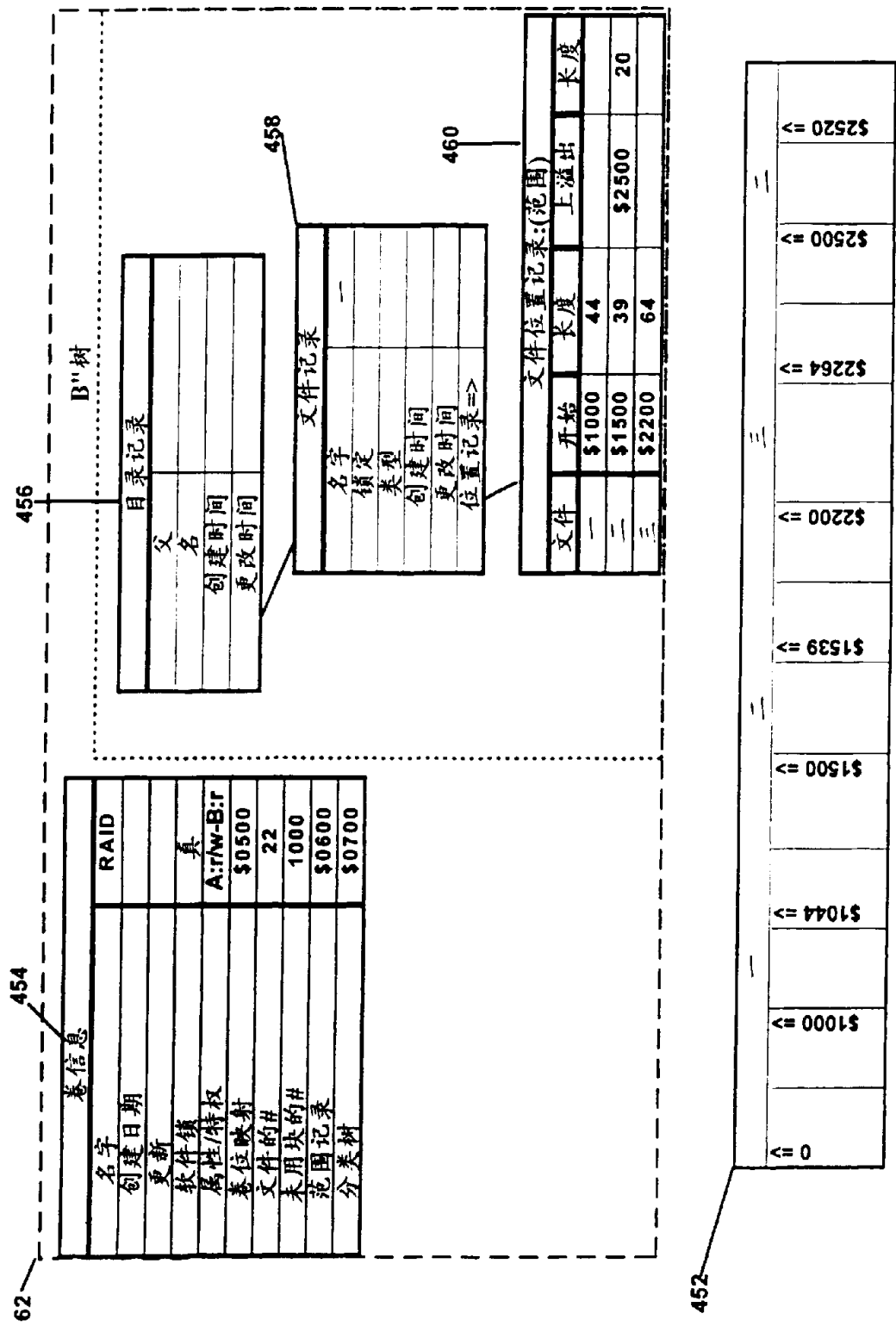


图 4

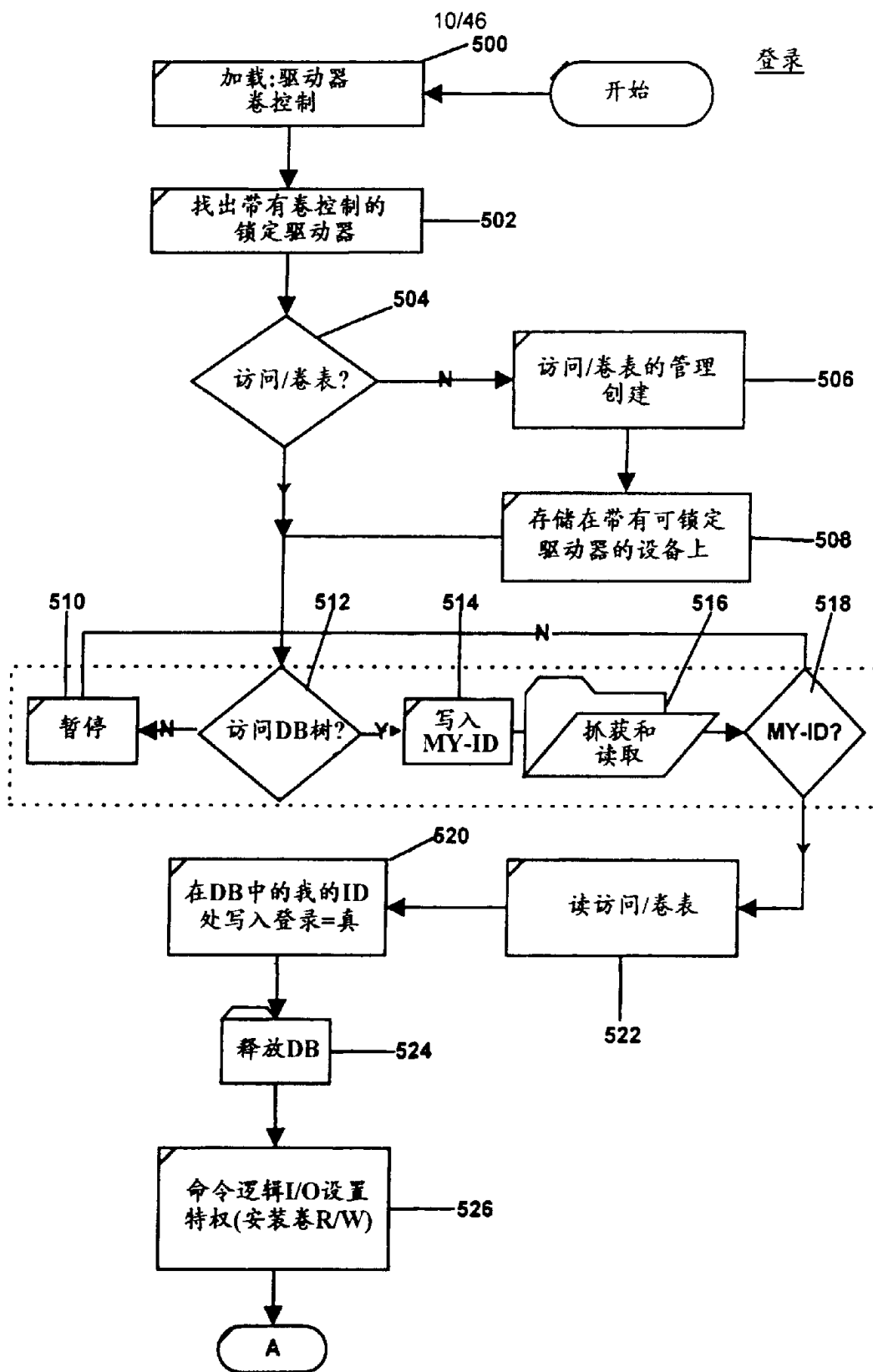


图 5A

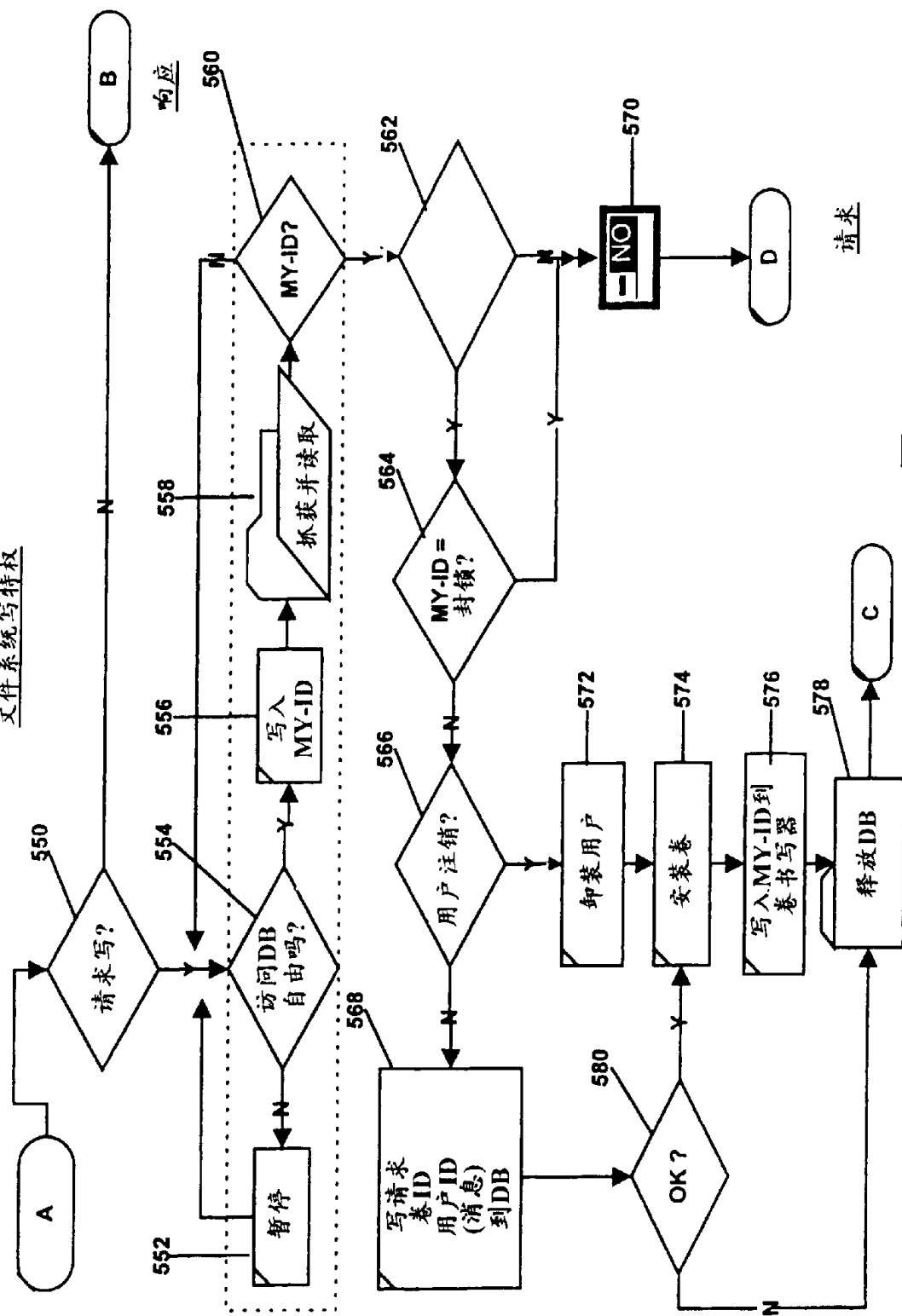
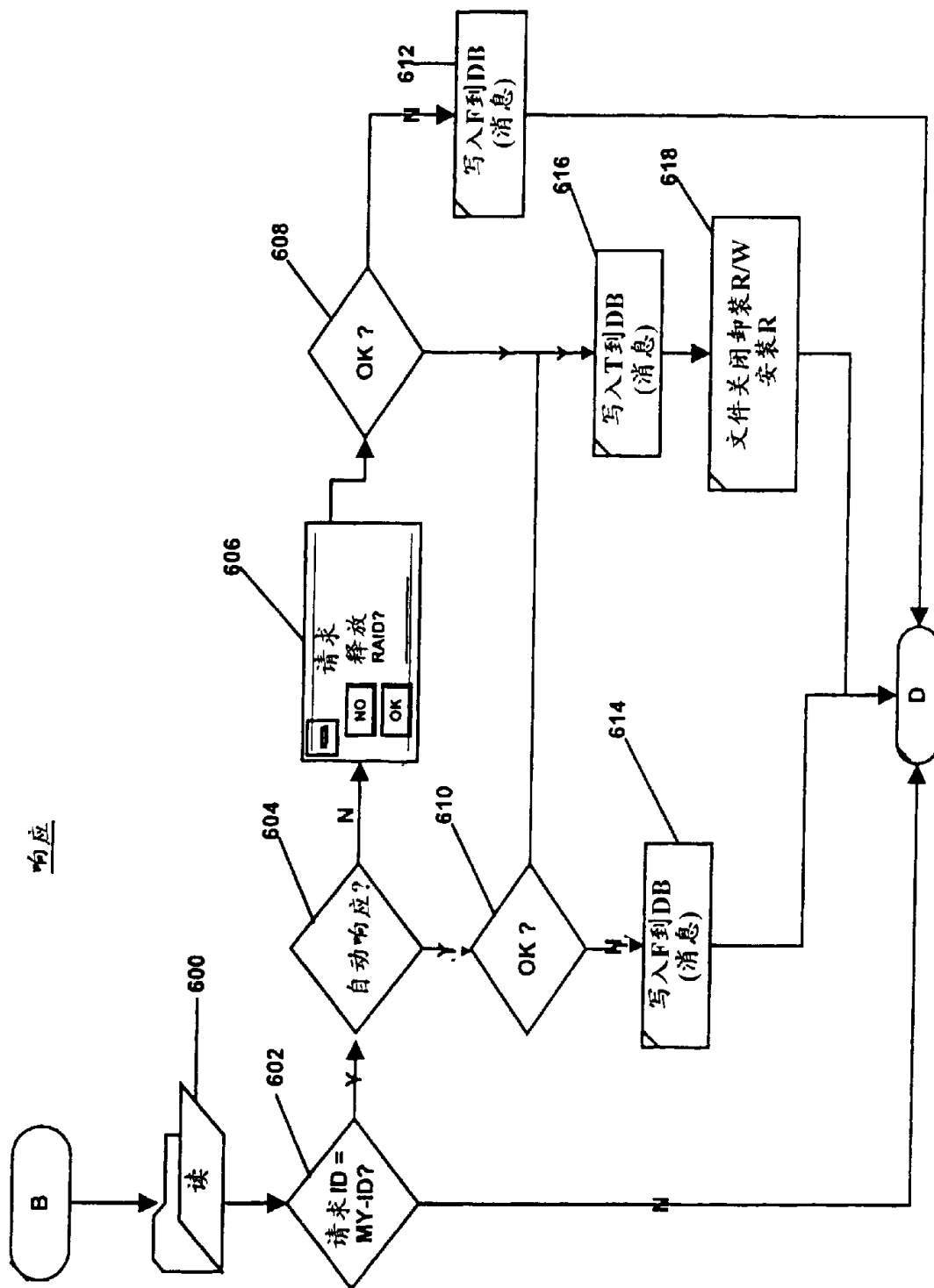


图 5B



对文件系统写访问

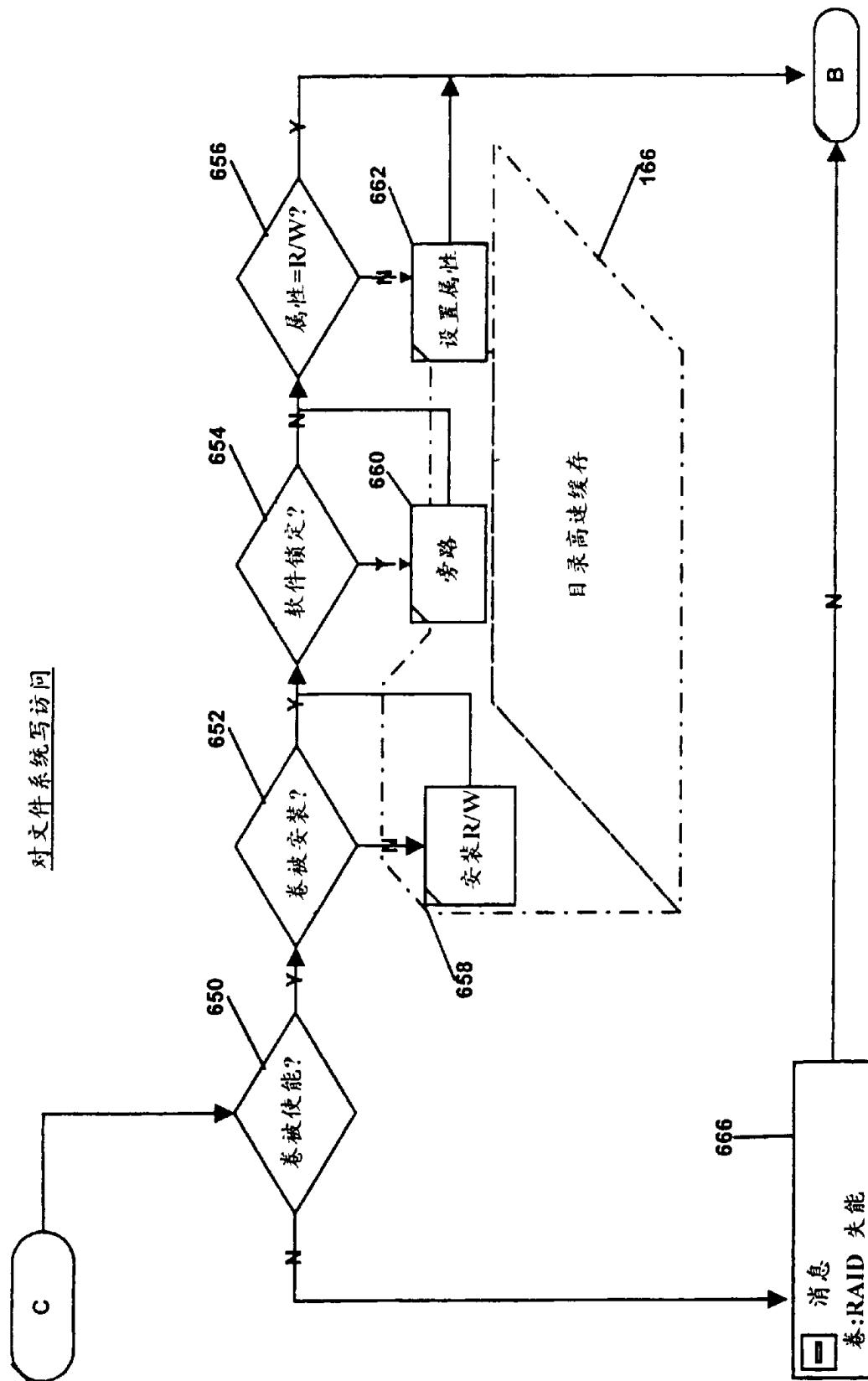


图 5D

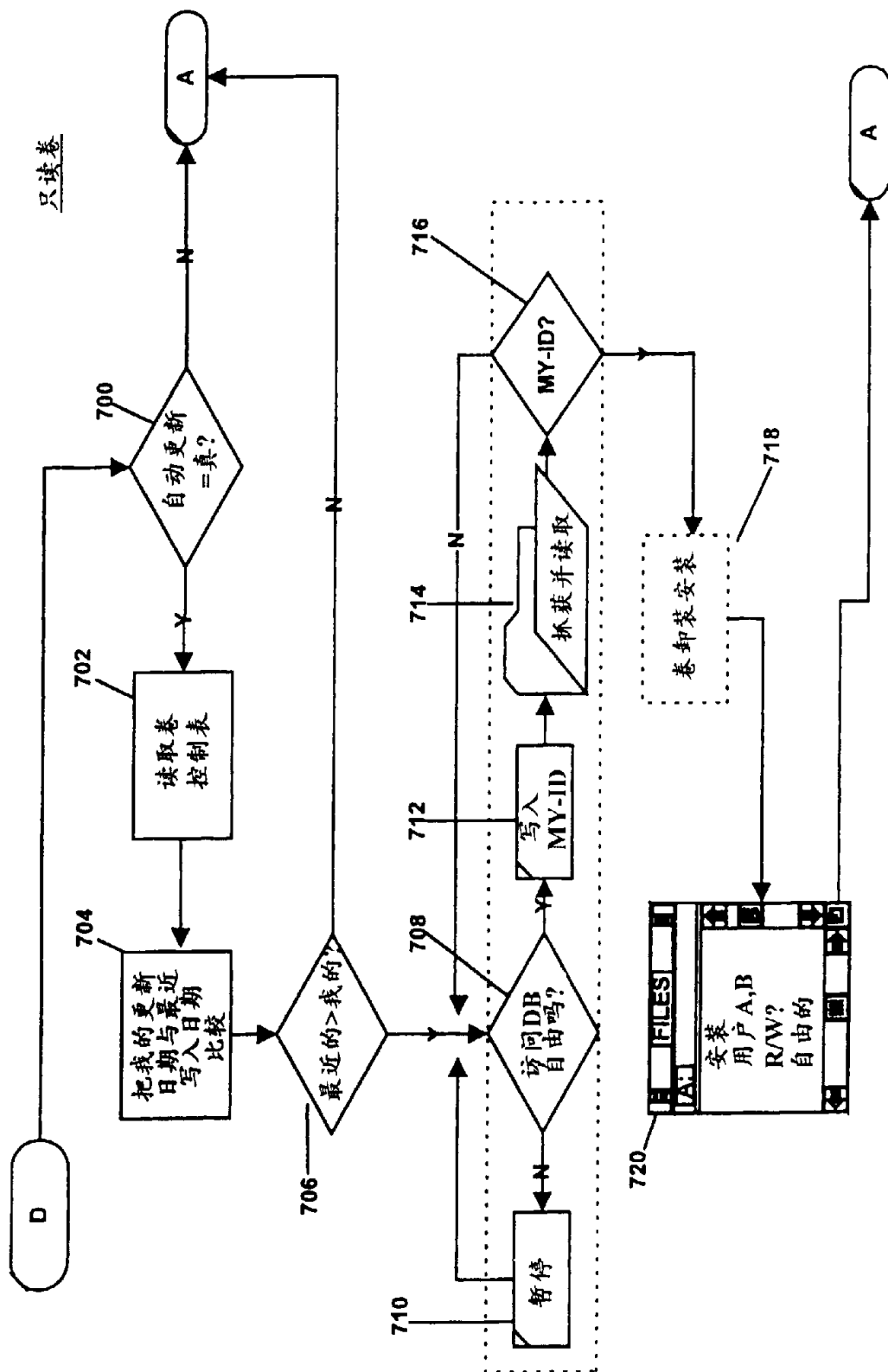


图 5E

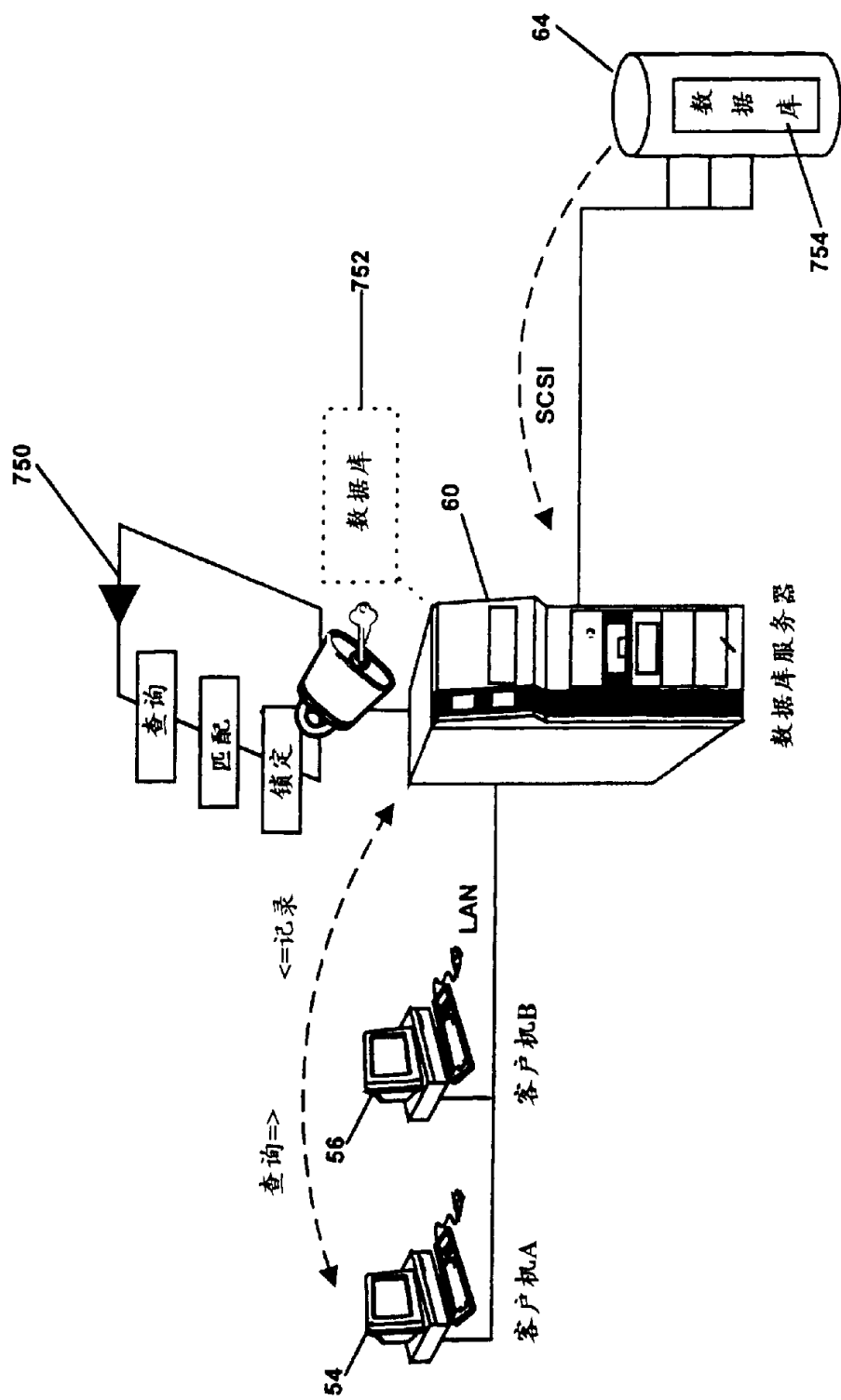


图 6

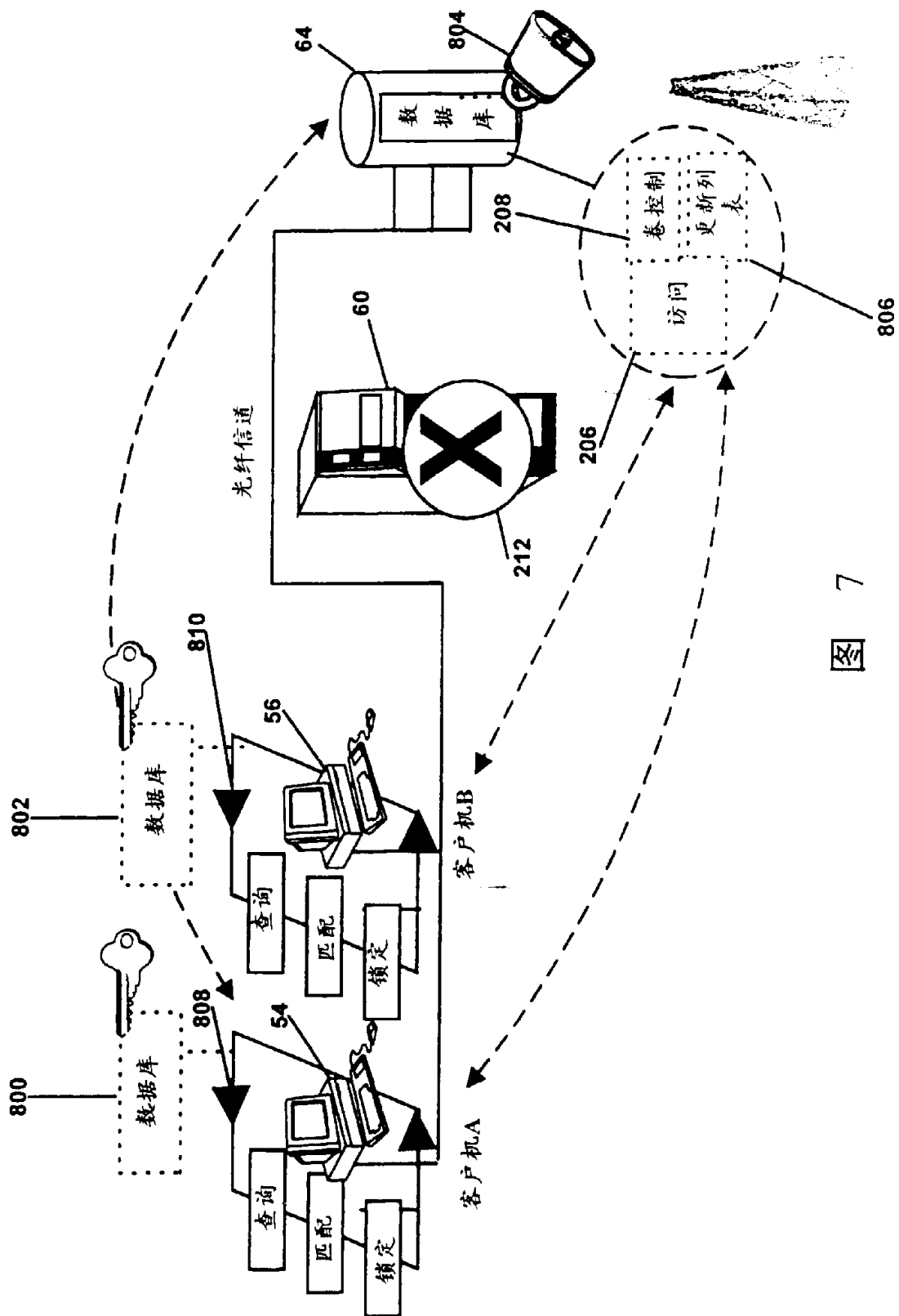


图 7

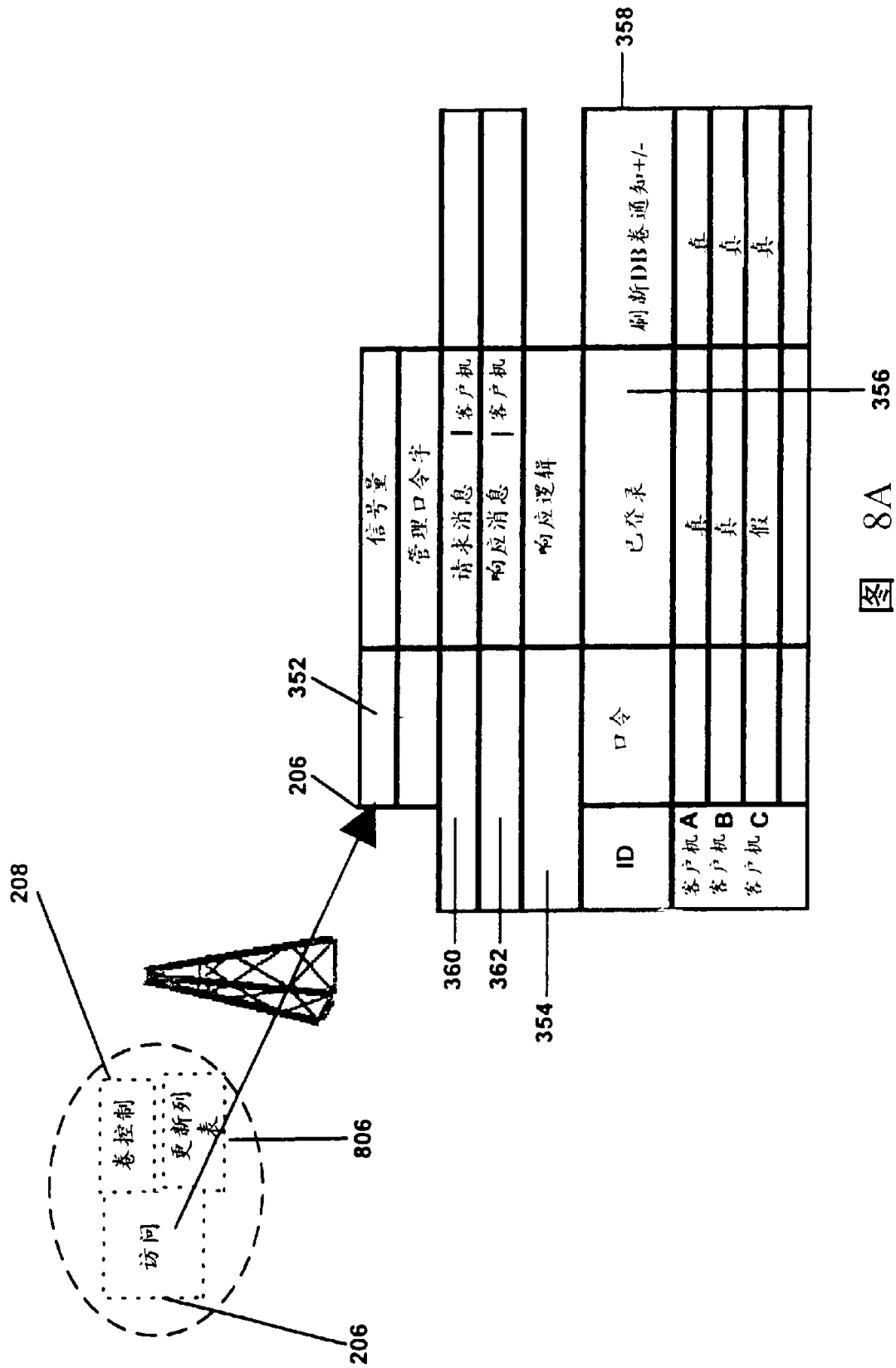


图 8A

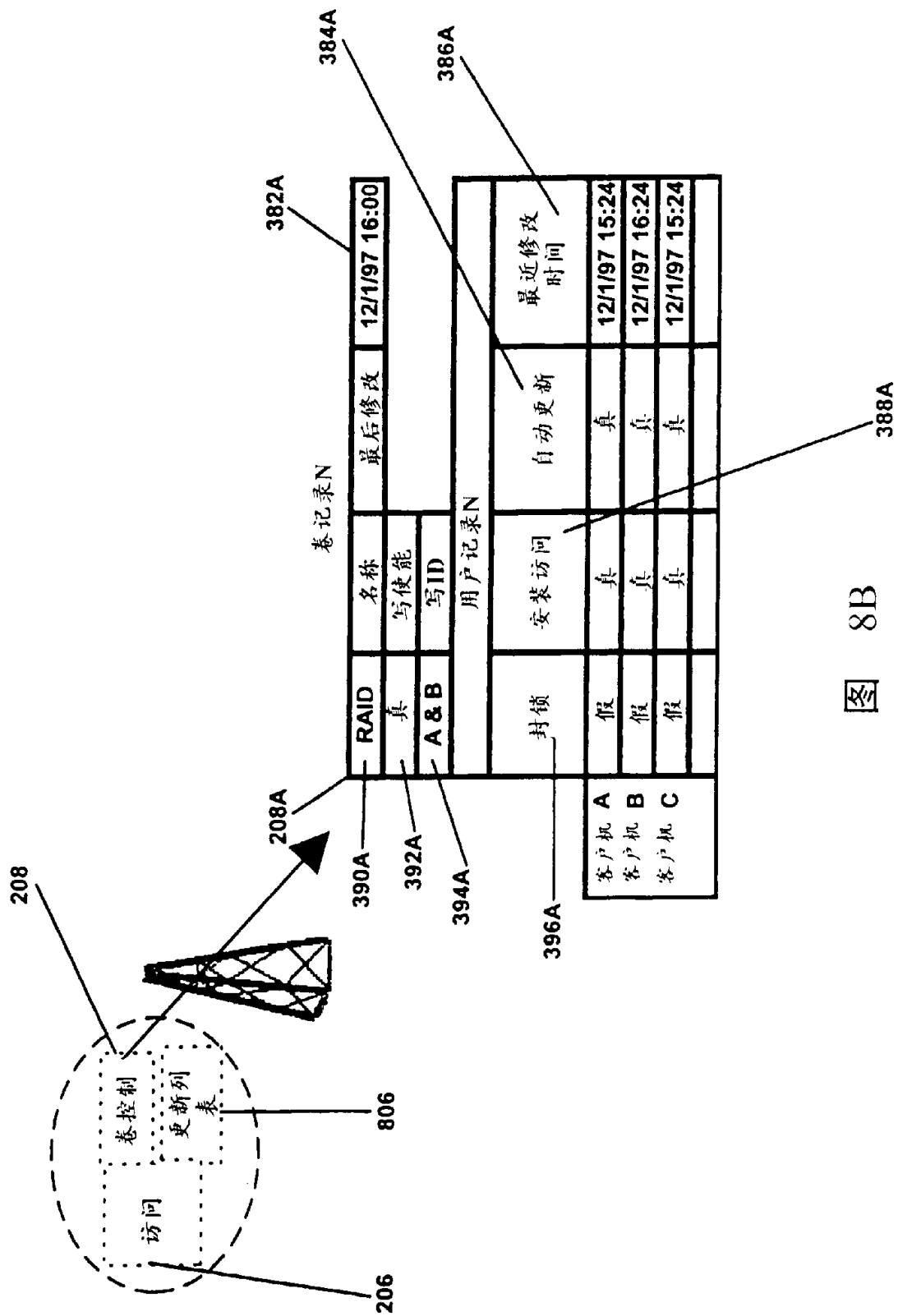


图 8B

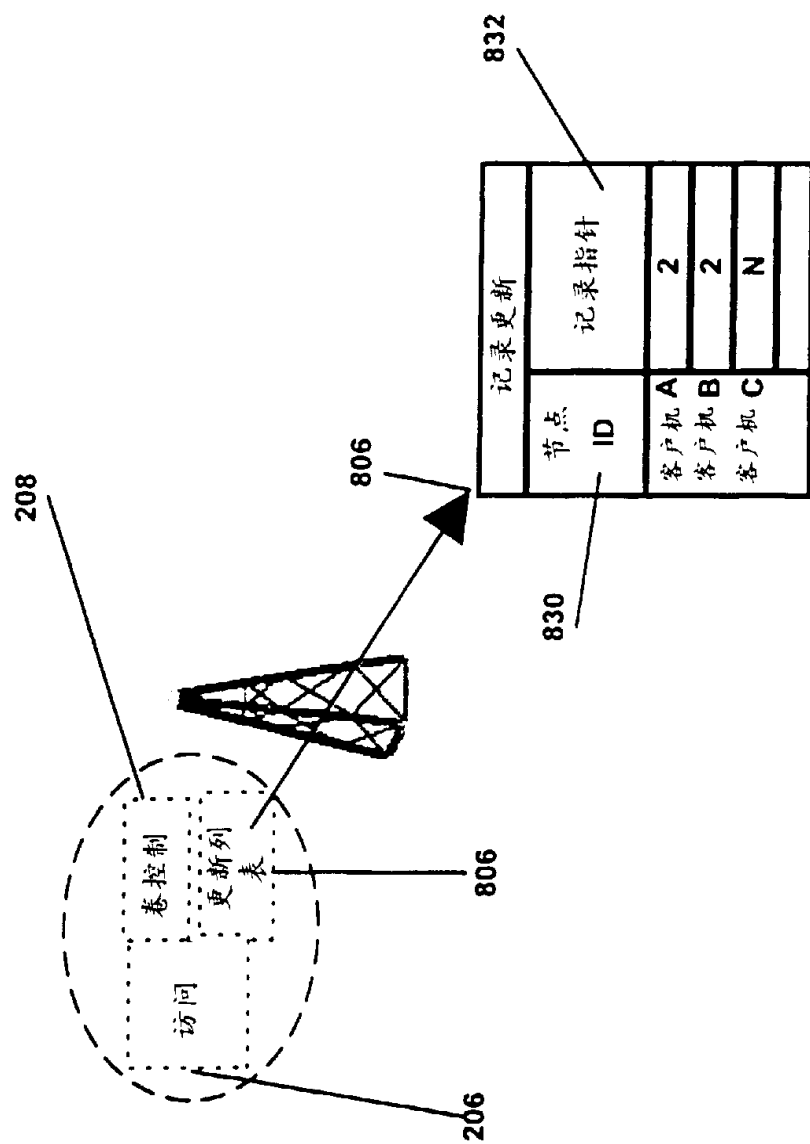
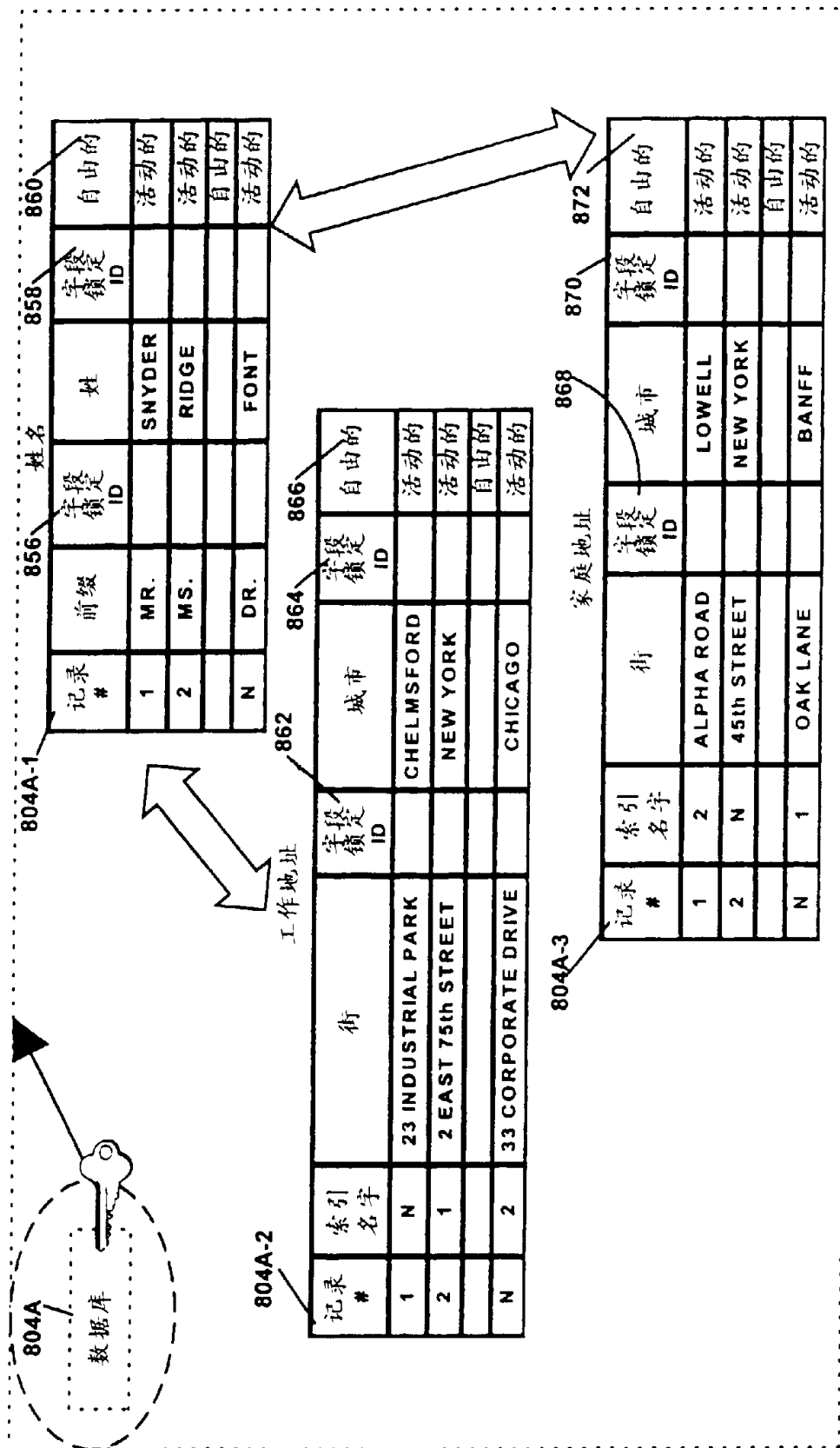
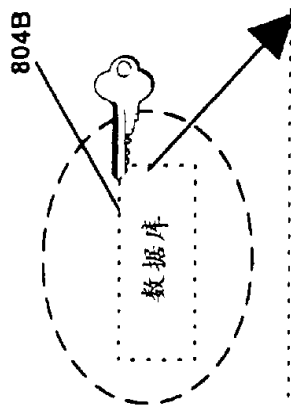


图 8C



关系数据库

图 9A



记录 #	804B 前缀	880B 字模 ID	姓	882B 字模 ID	地址	884B 字模 ID	城市	886B 字模 ID	888B 自由的
1	MR.		SNYDER		2 EAST 75th STREET		NEW YORK		活动的
2	MR.		SNYDER		OAK LANE		BANFF		活动的
3	MS.		RIDGE		33 CORPORATE DRIVE		CHICAGO		活动的
4	MS.		RIDGE		ALPHA ROAD		LOWELL		活动的
									自由的
N	DR.		FONT		23 INDUSTRIAL PARK		CHELMSFORD		活动的
N+1	DR.		FONT		45th STREET		NEW YORK		活动的

平直数据片

图 9B

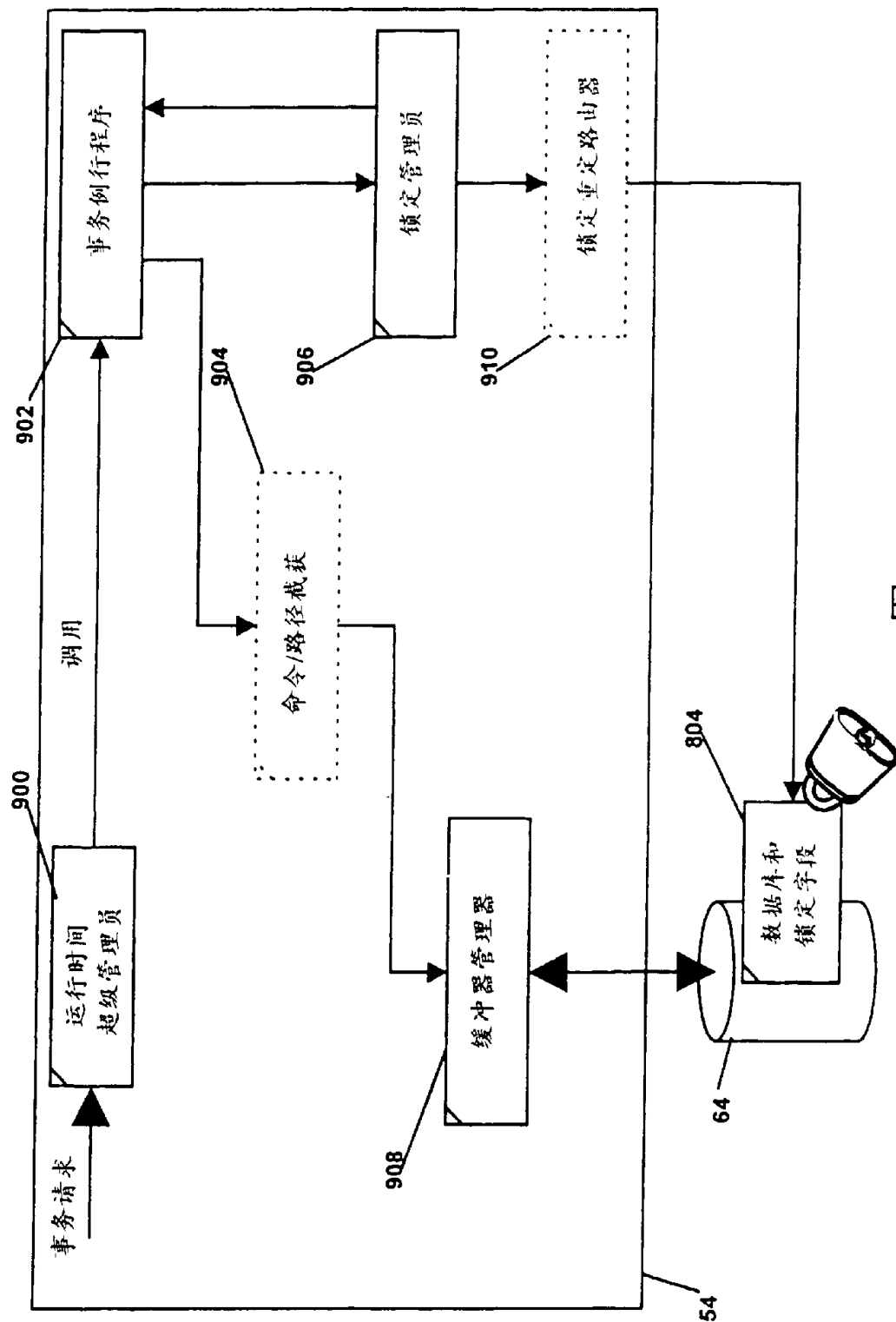


图 10

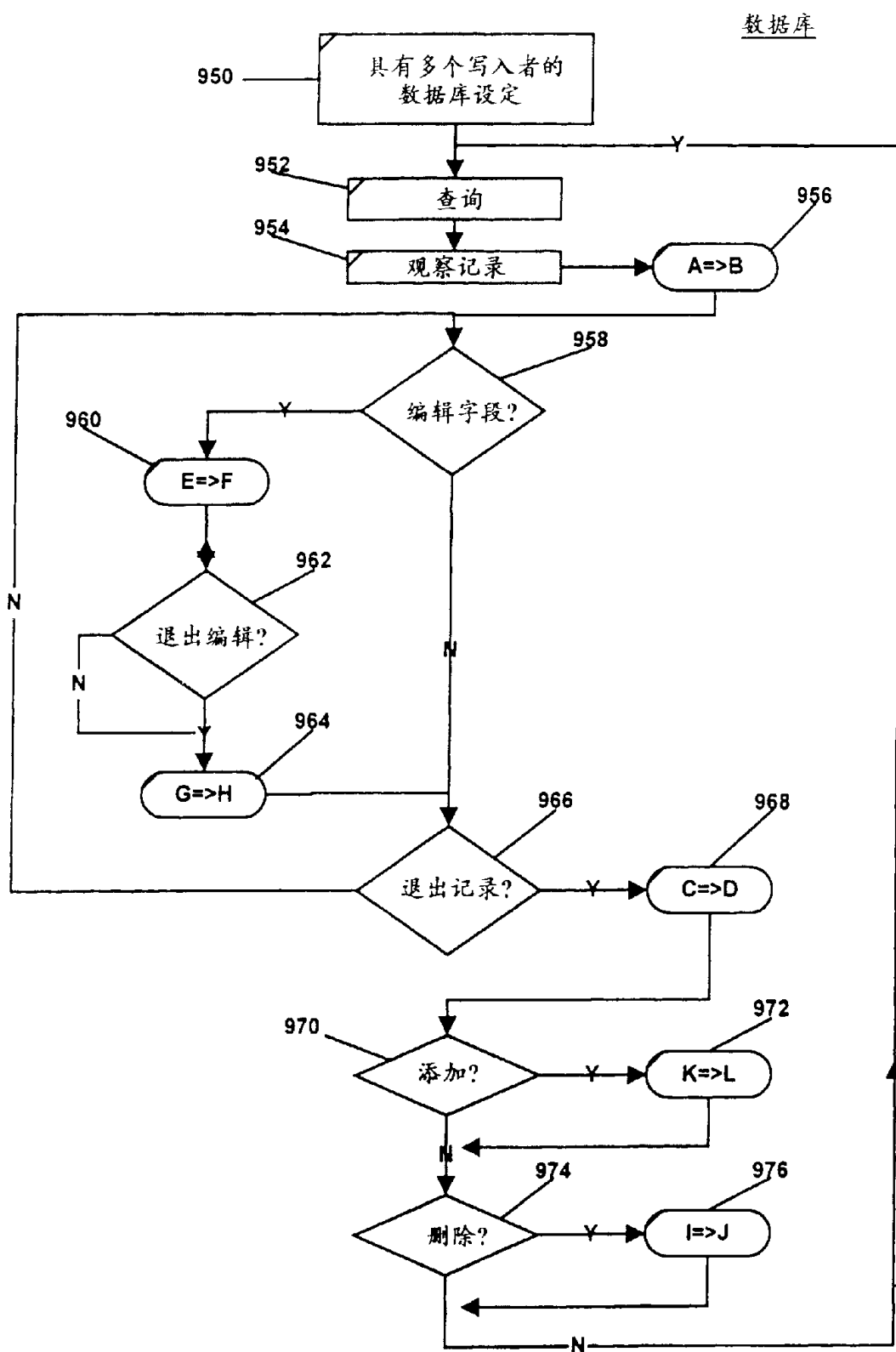


图 11

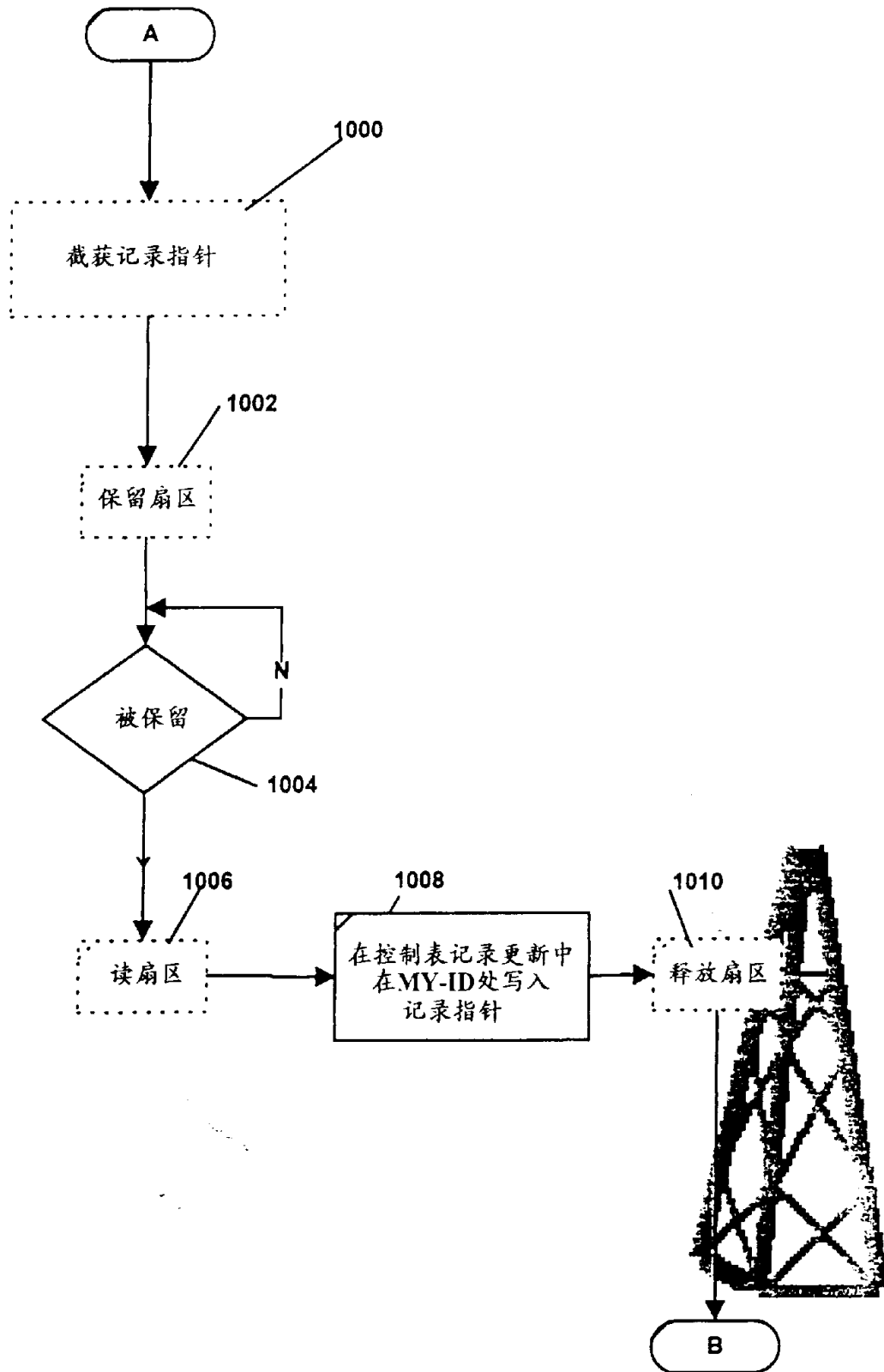


图 12A

退出记录

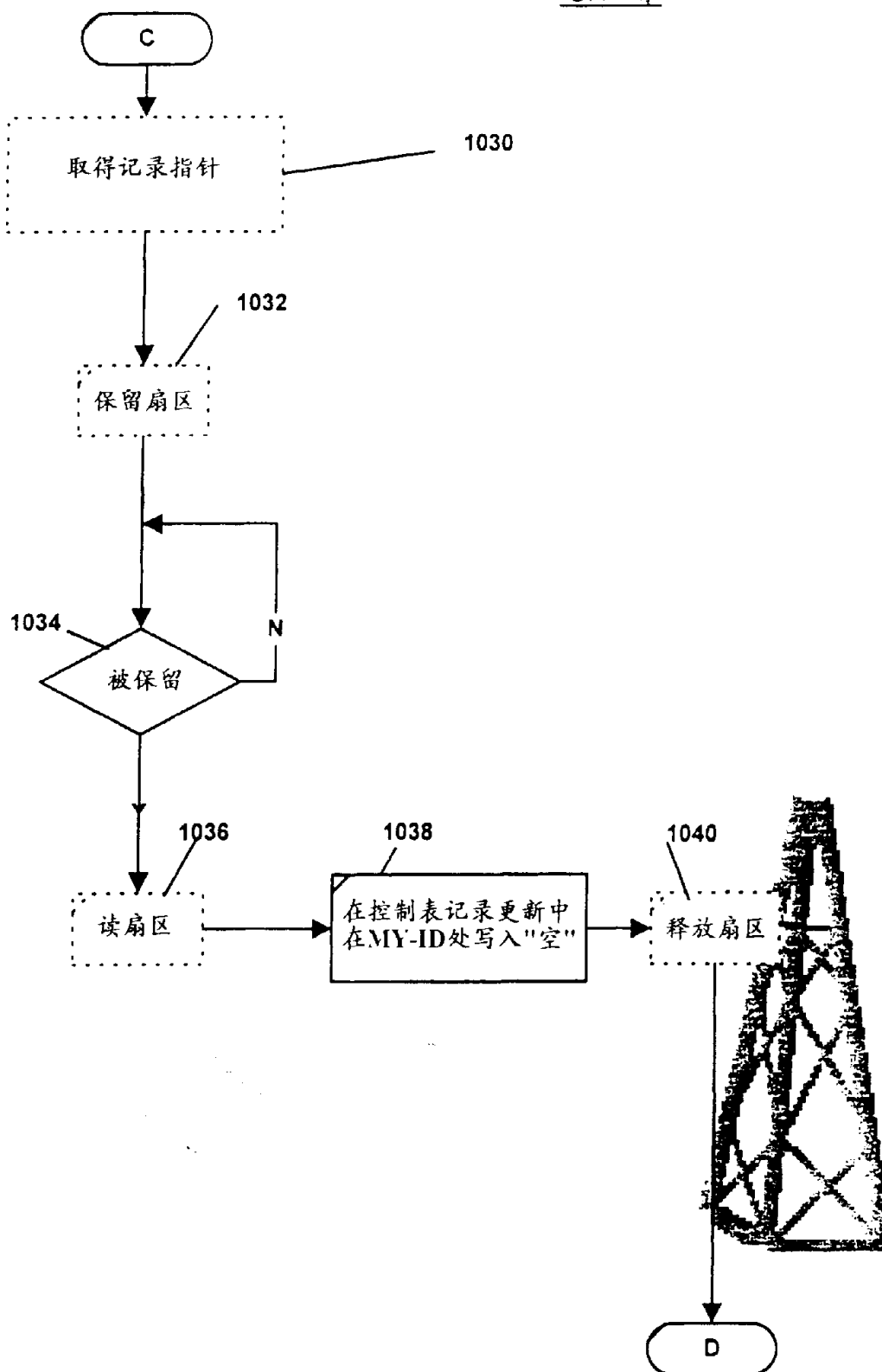


图 12B

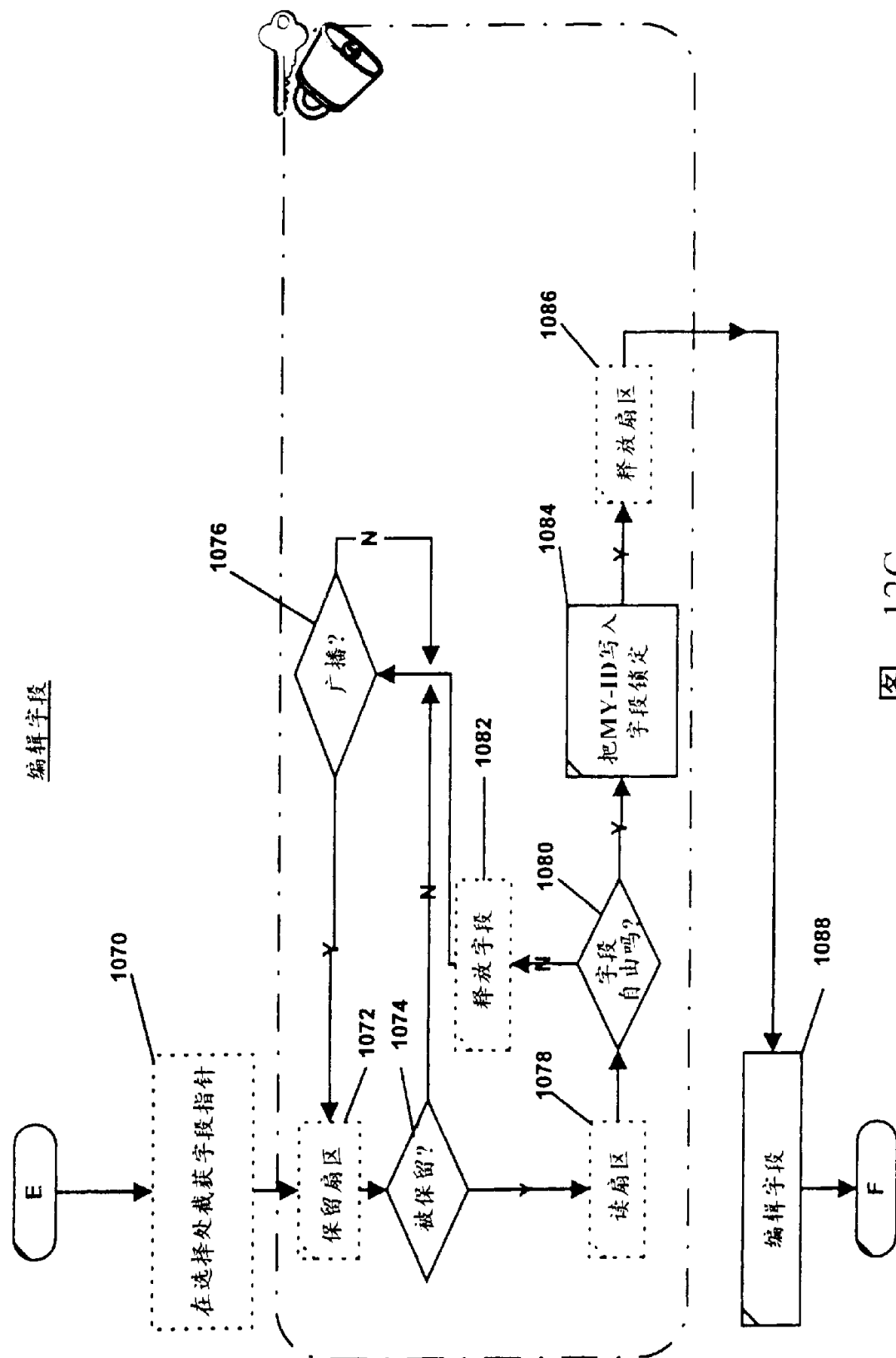


图 12C

退出编辑字段

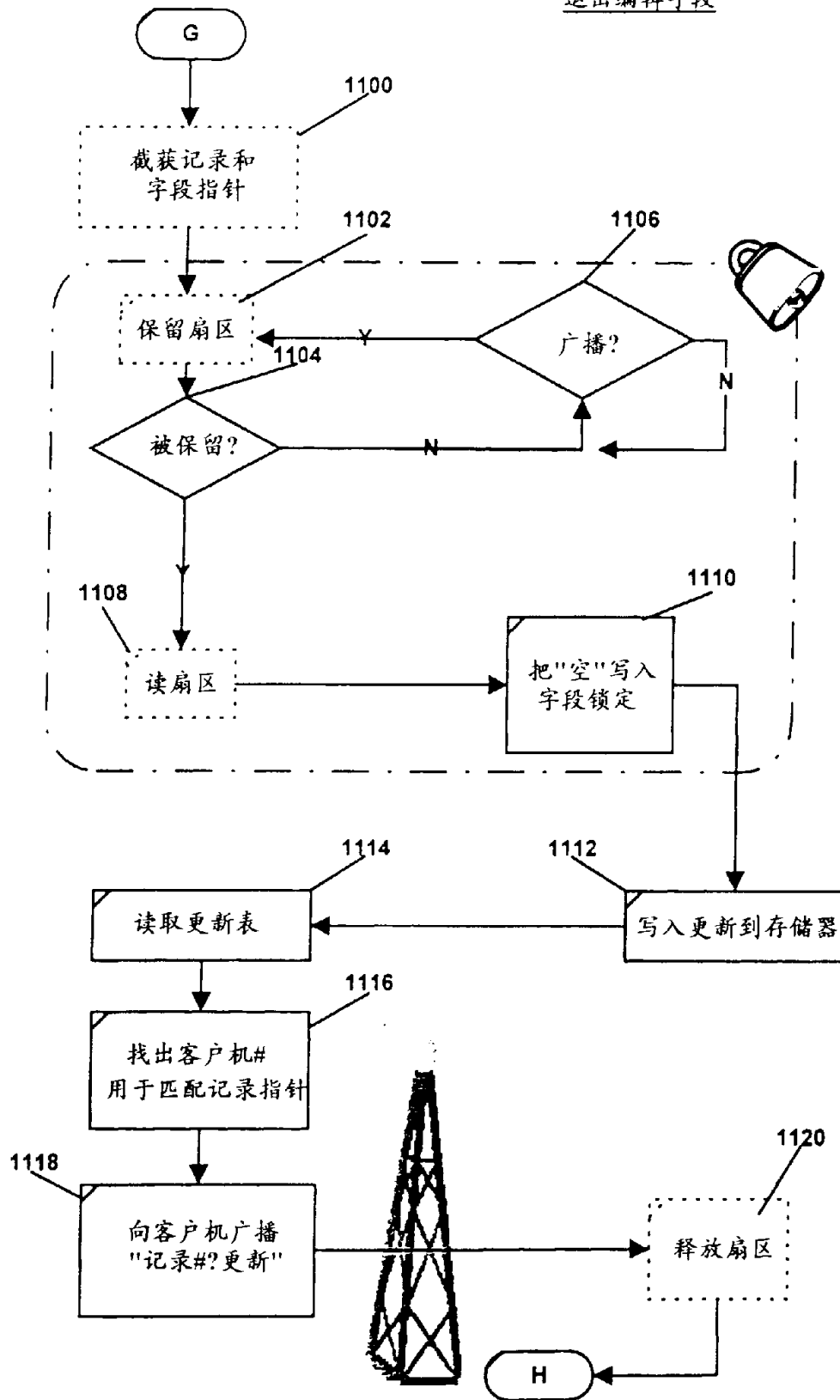


图 12D

删除记录

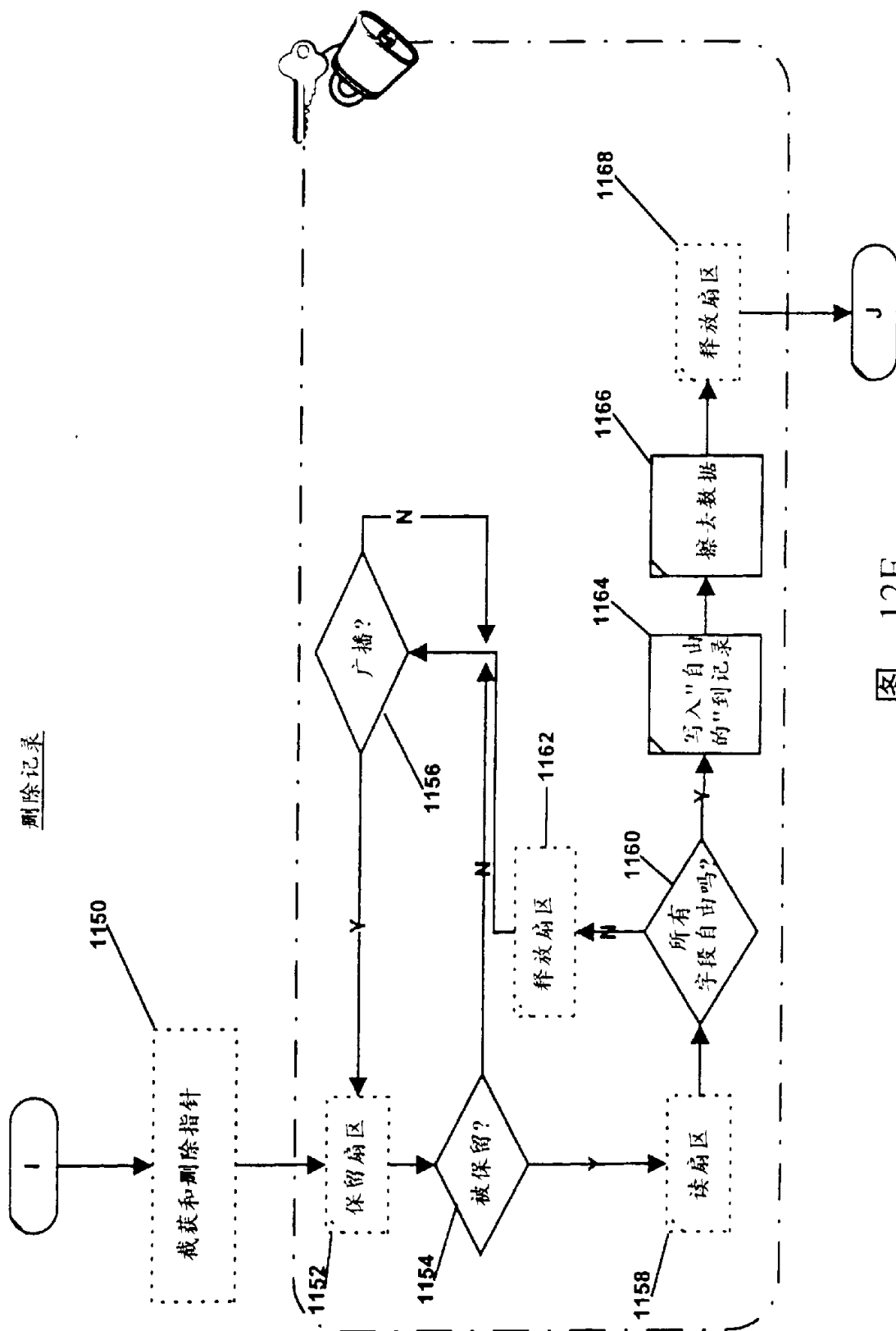


图 12F

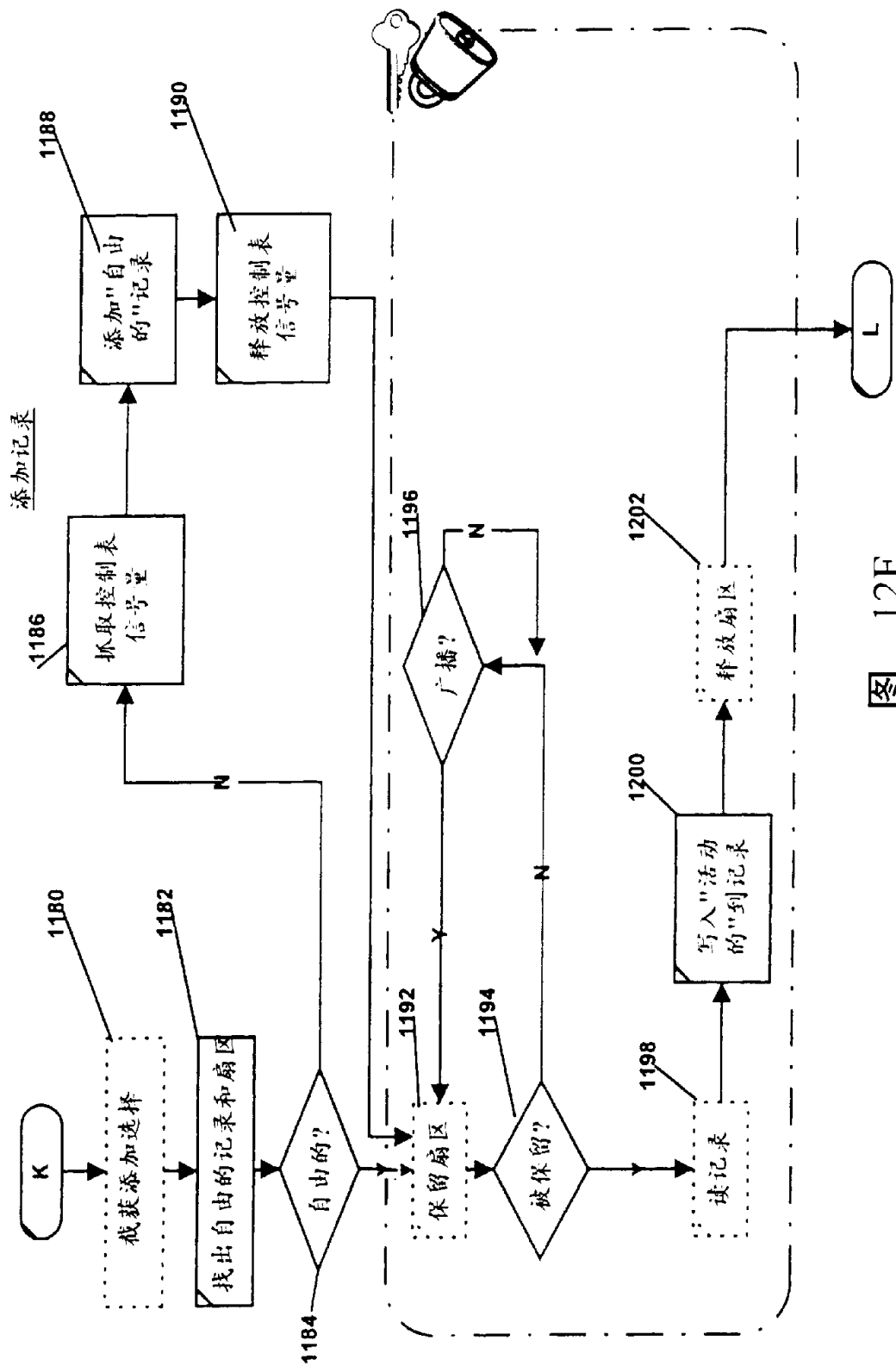


图 12F

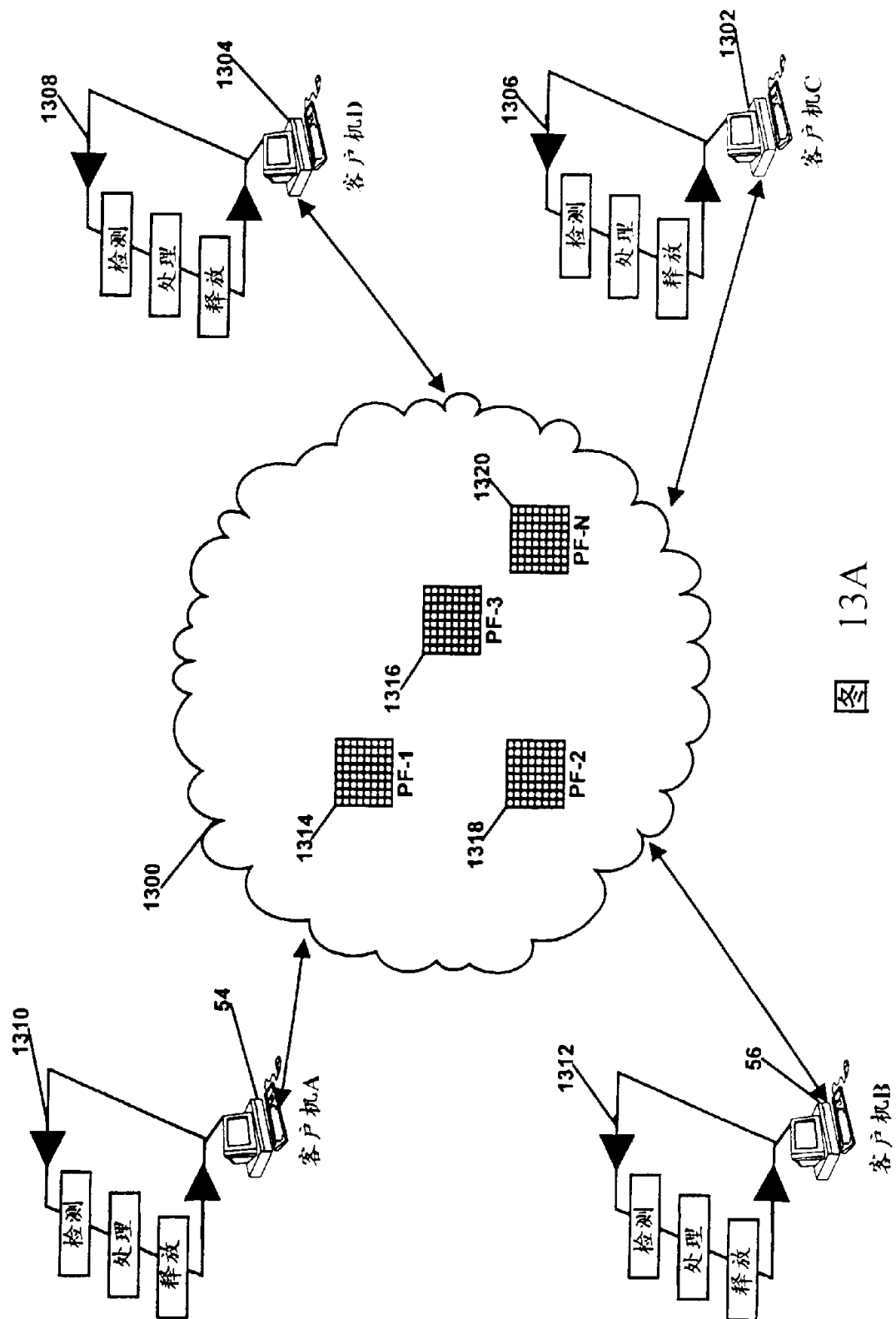


图 13A

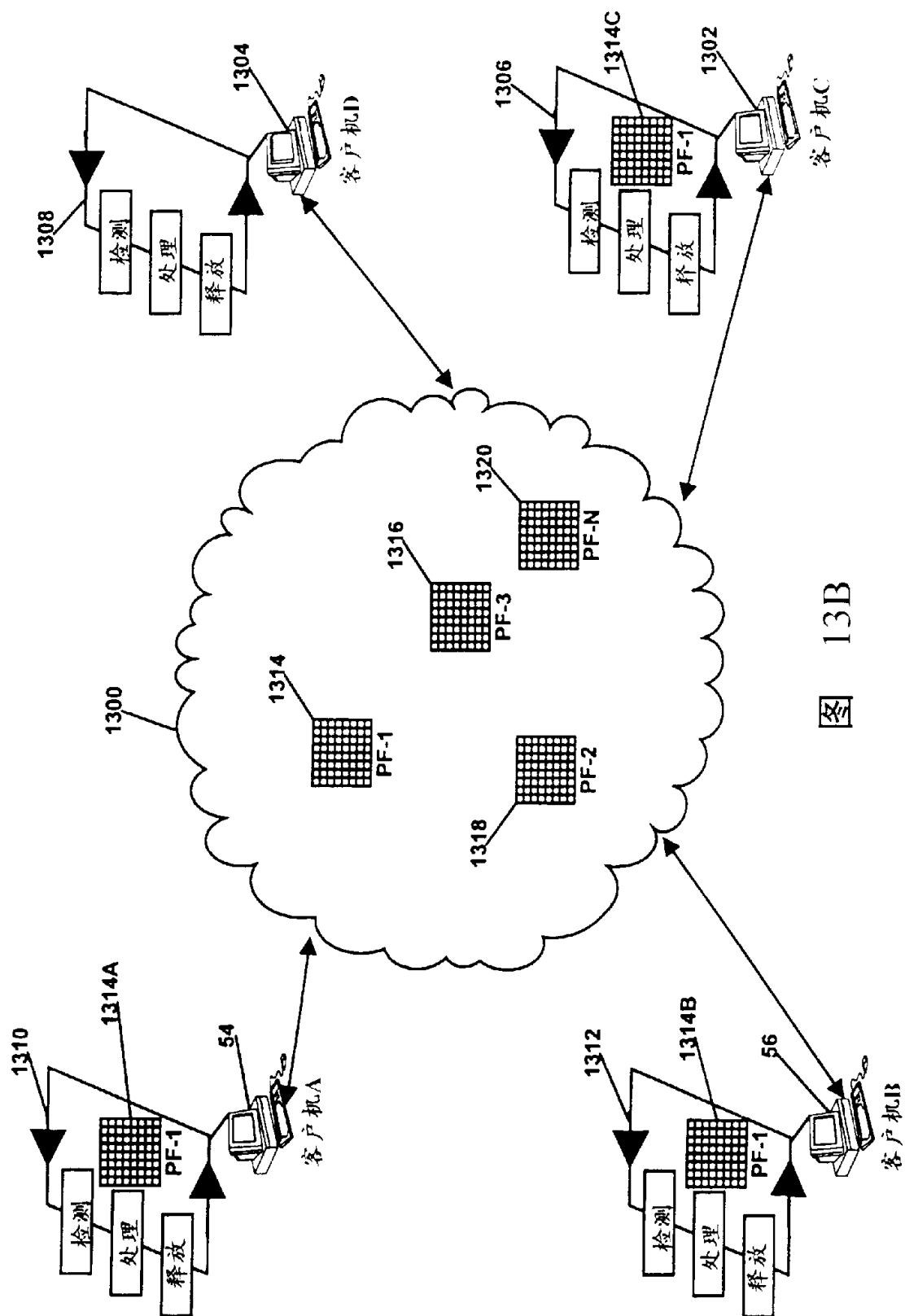


图 13B

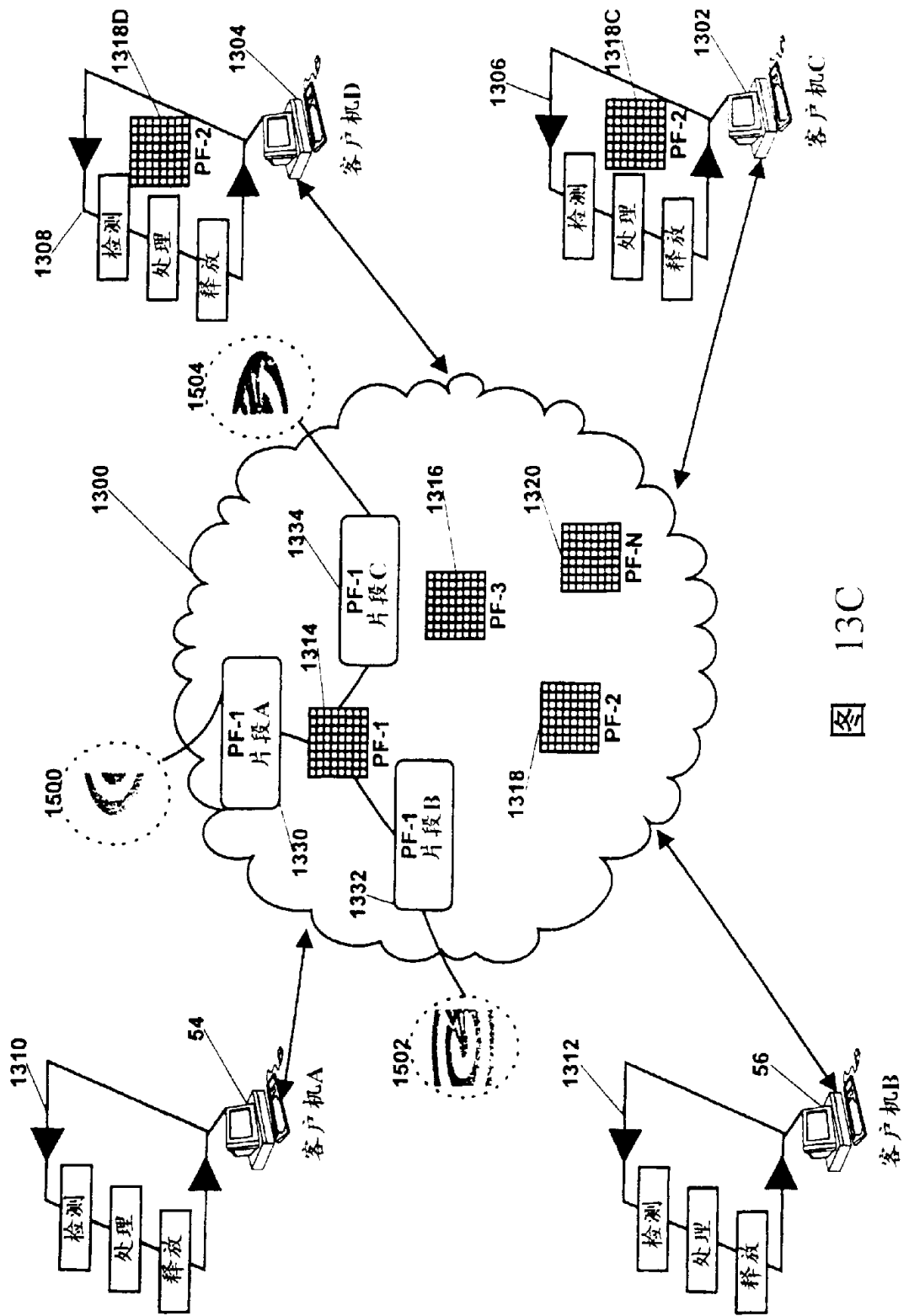


图 13C

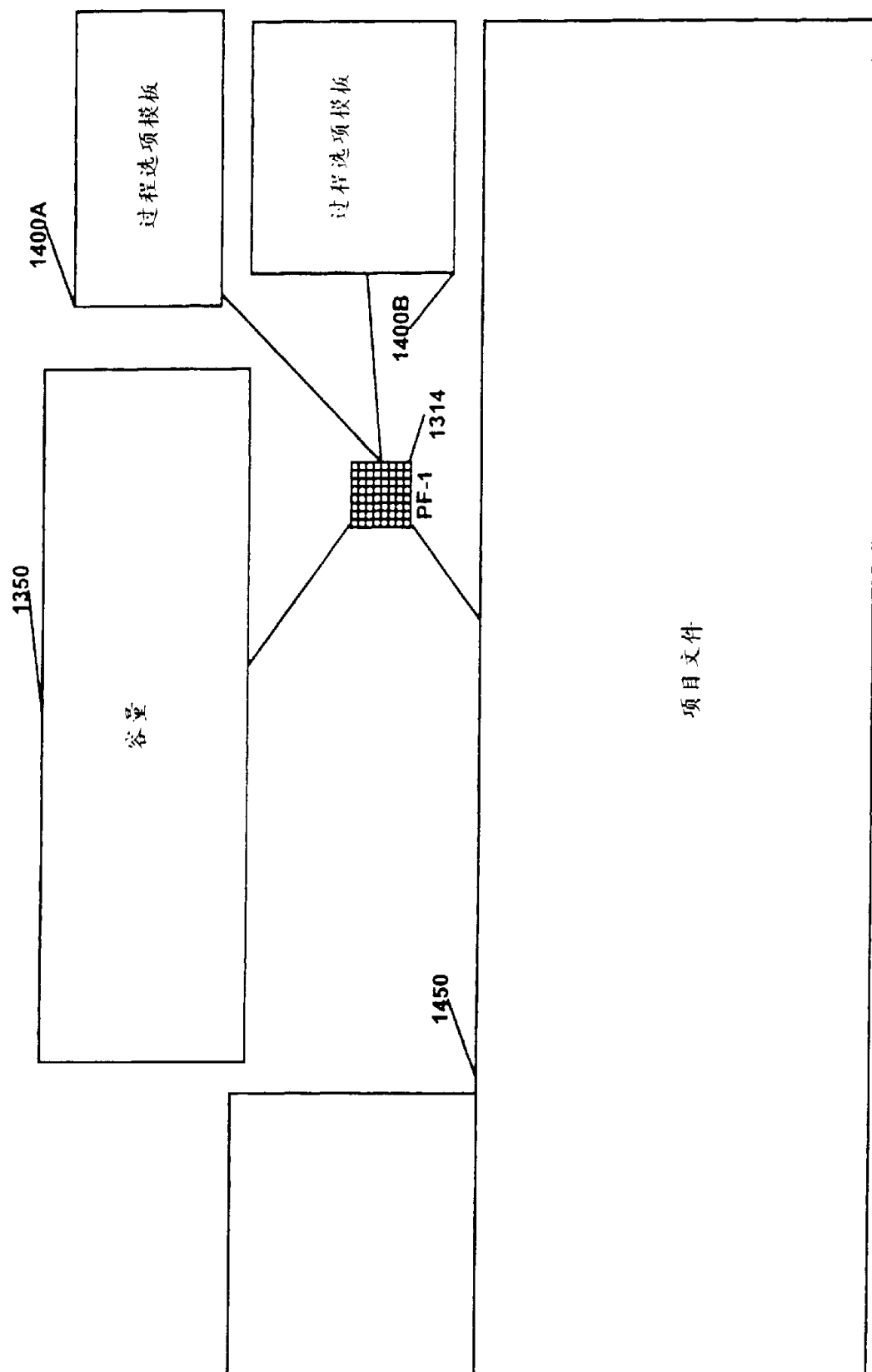


图 14A

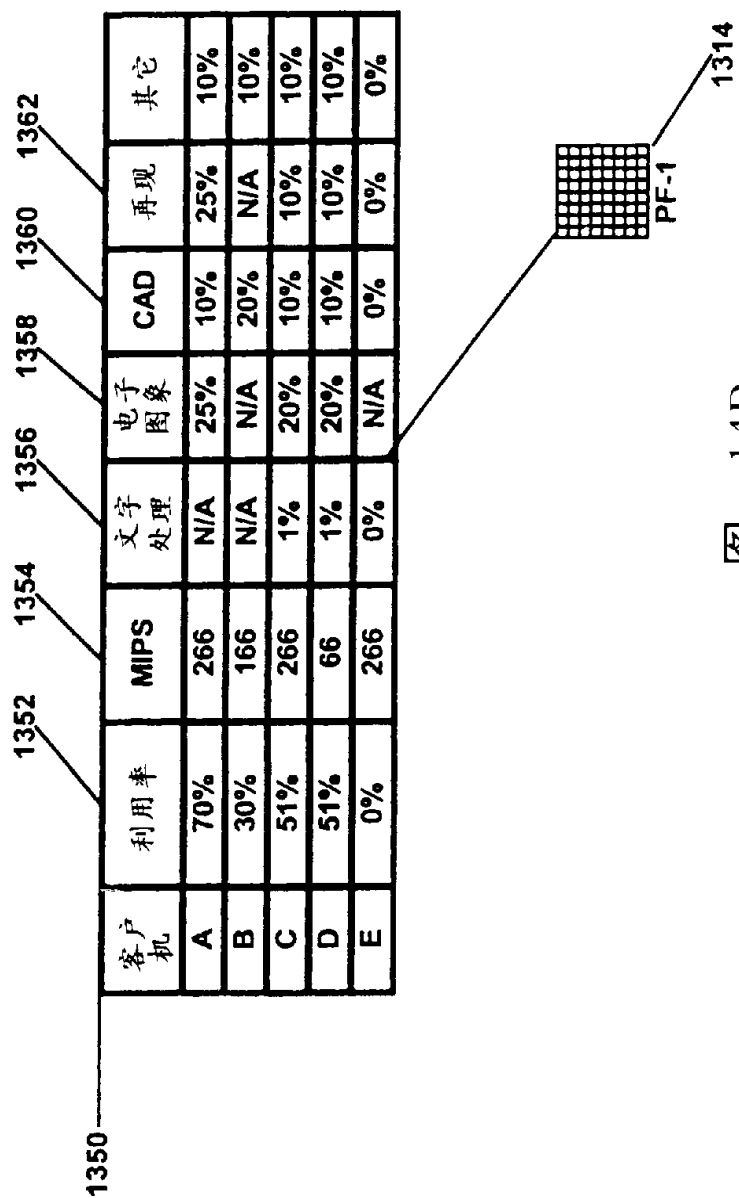


图 14B

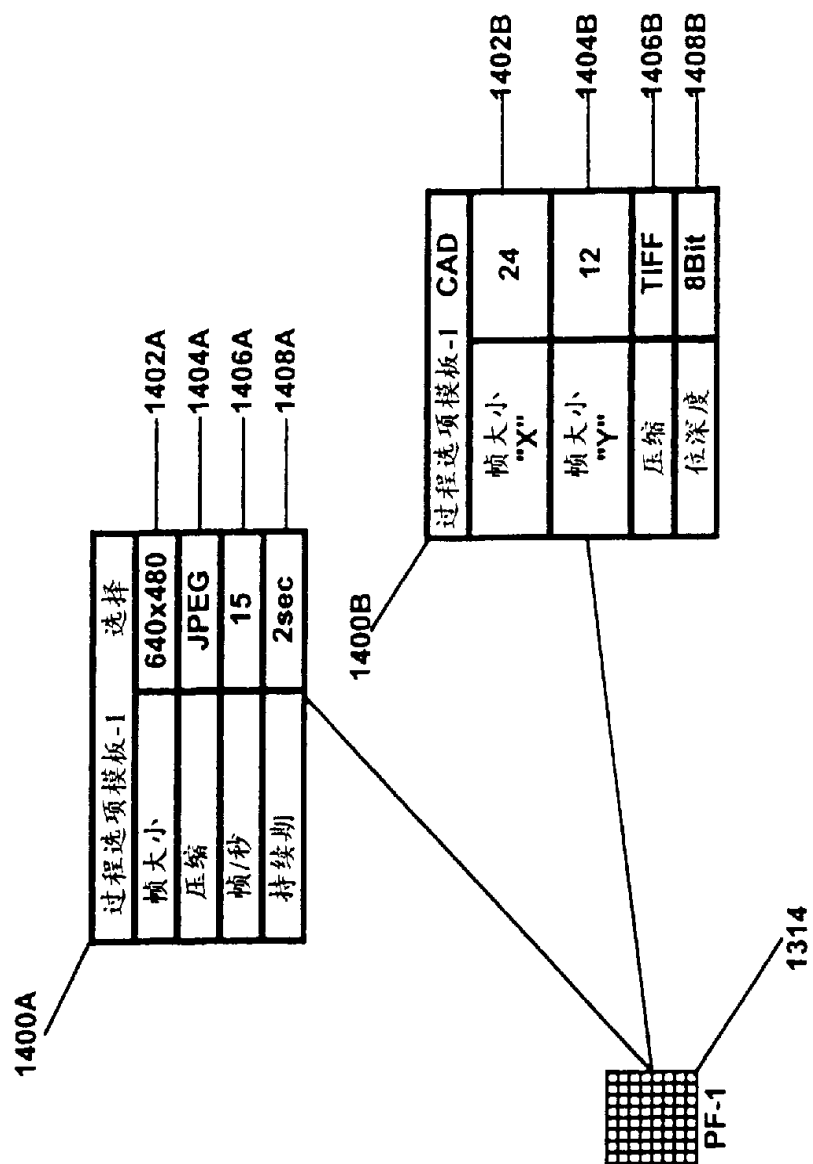
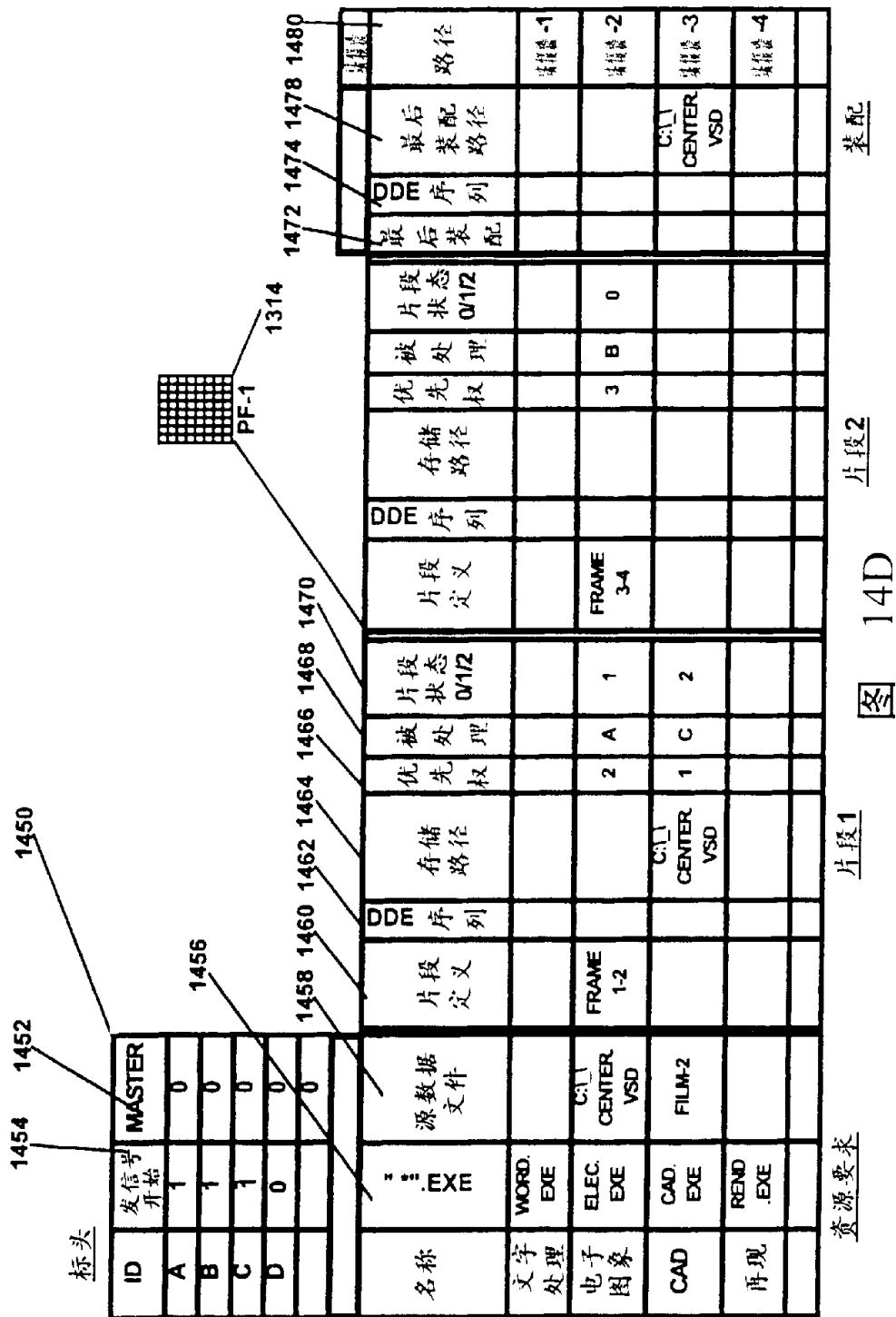


图 14C



资源要求

片段1

图 14D

片段2

装配

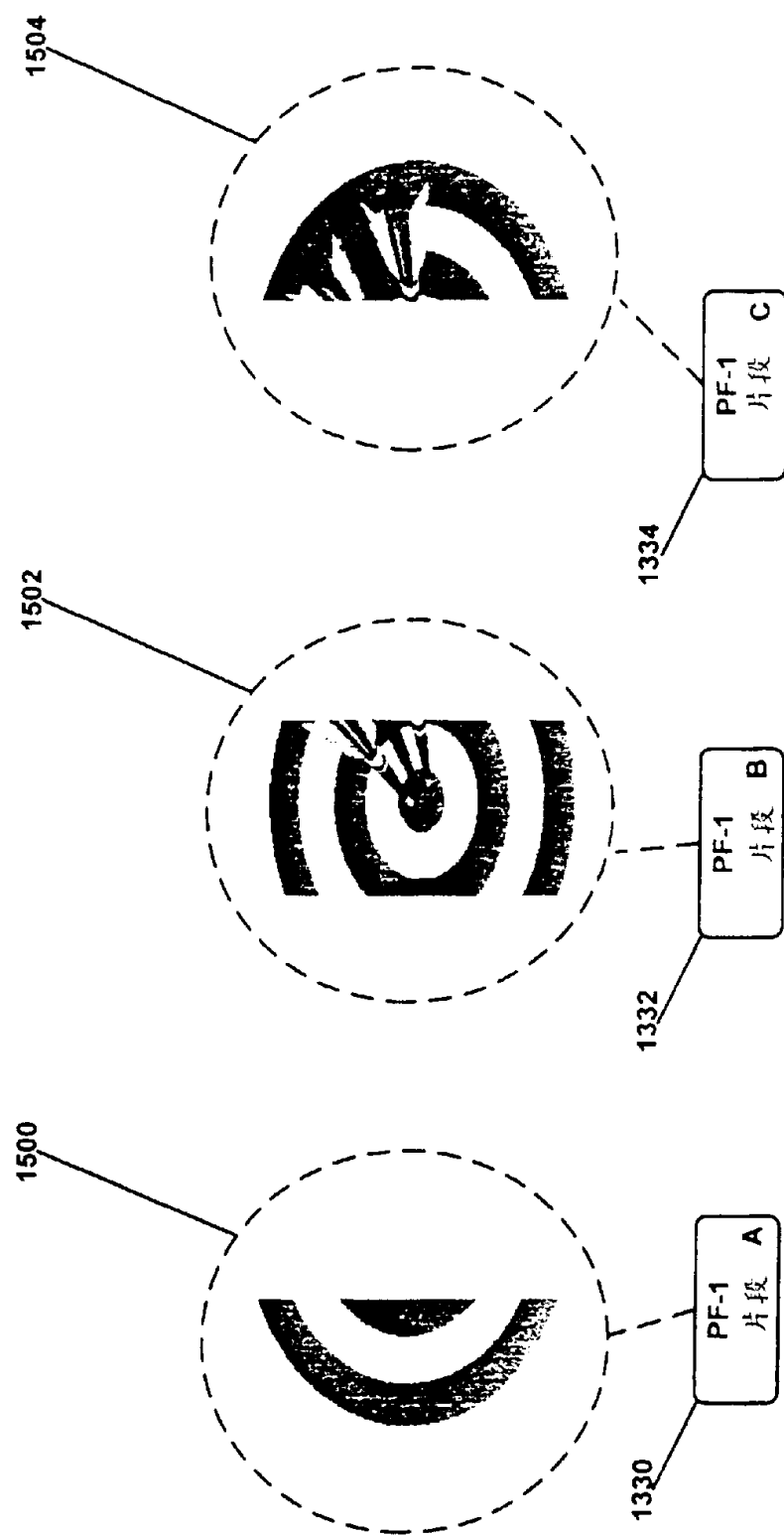


图 15A

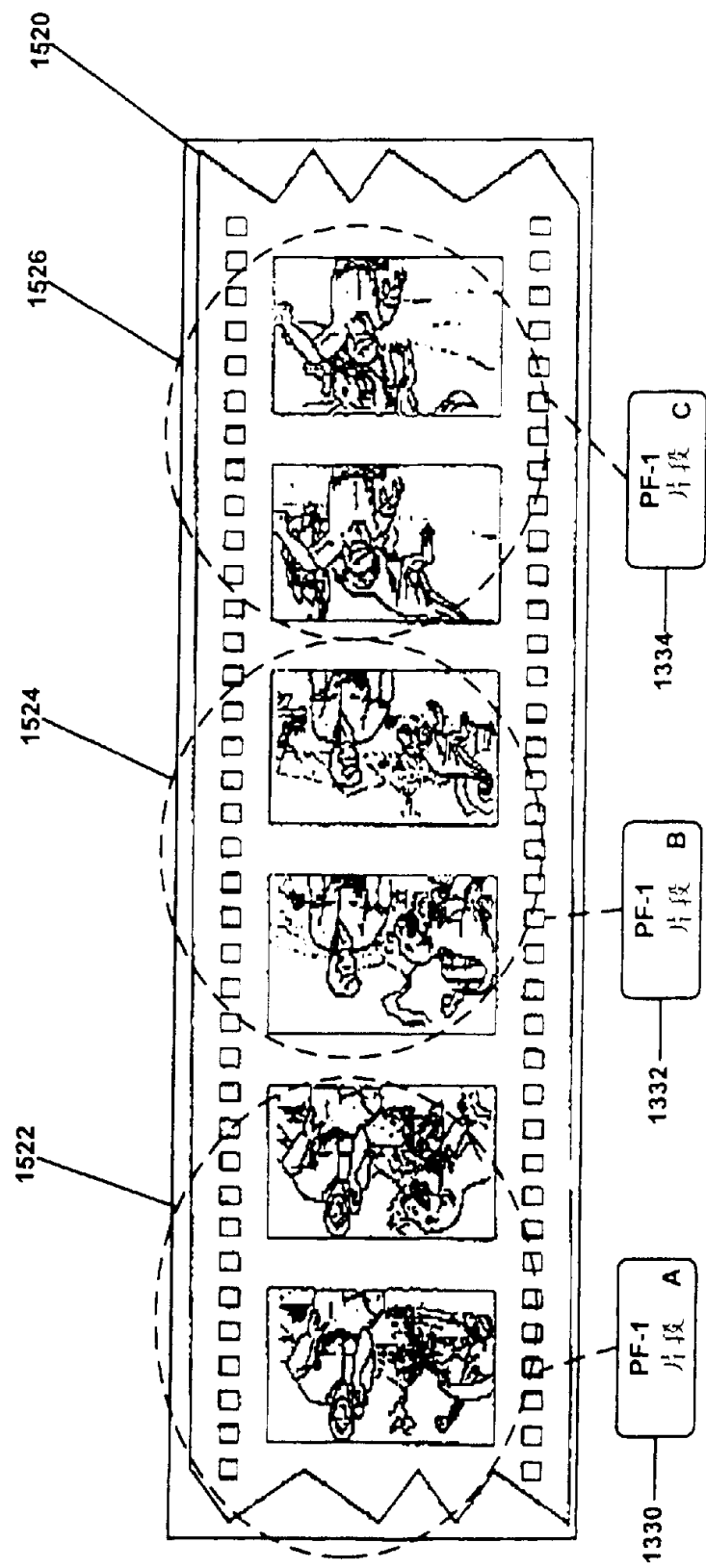


图 15B

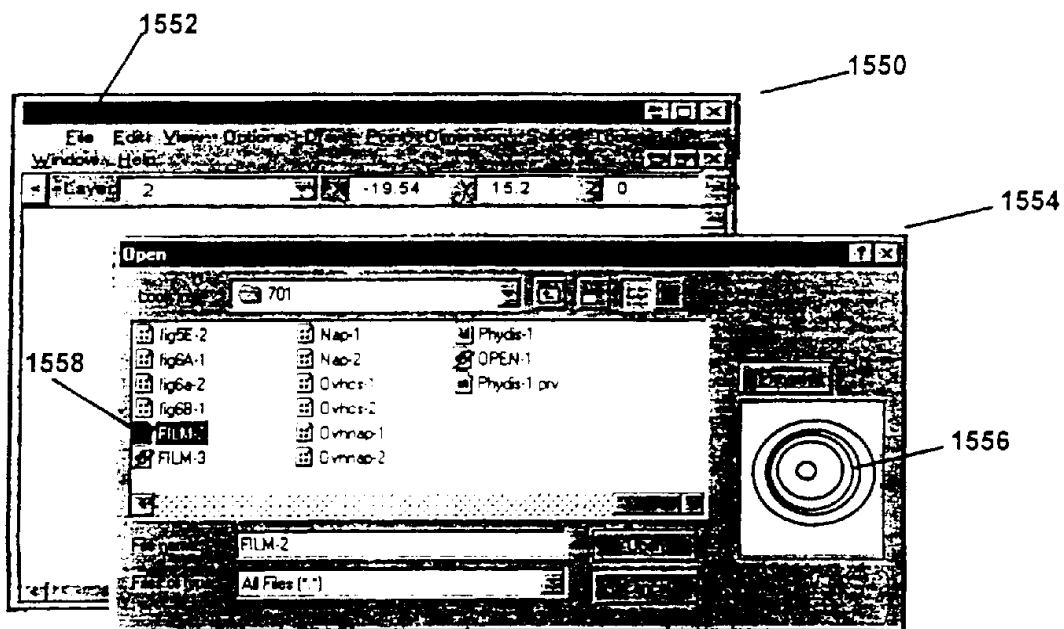


图 16A

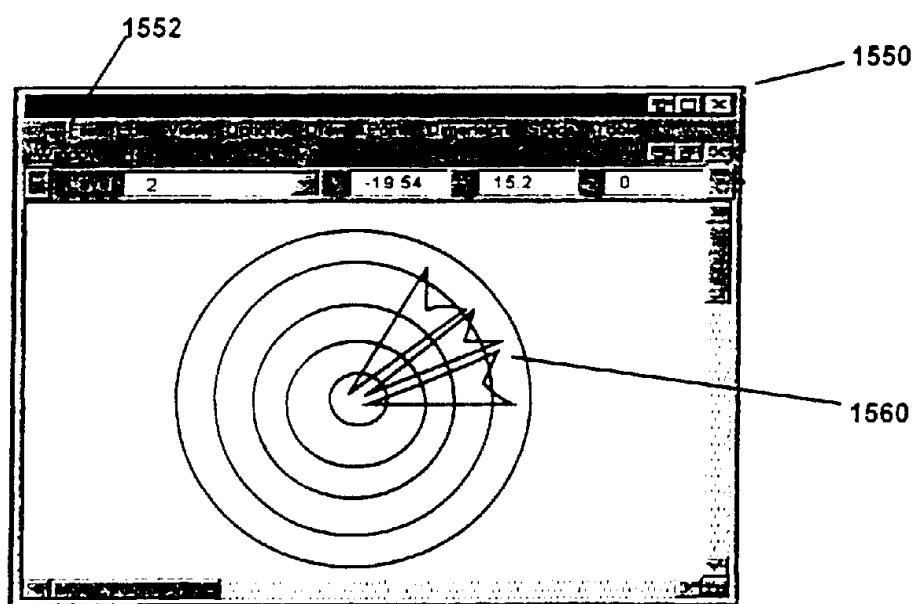


图 16B

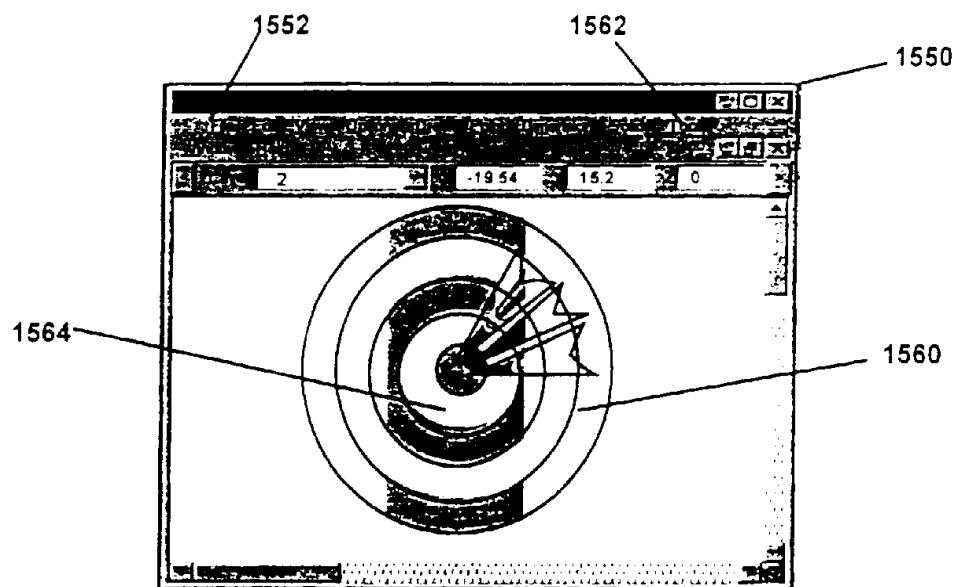


图 16C

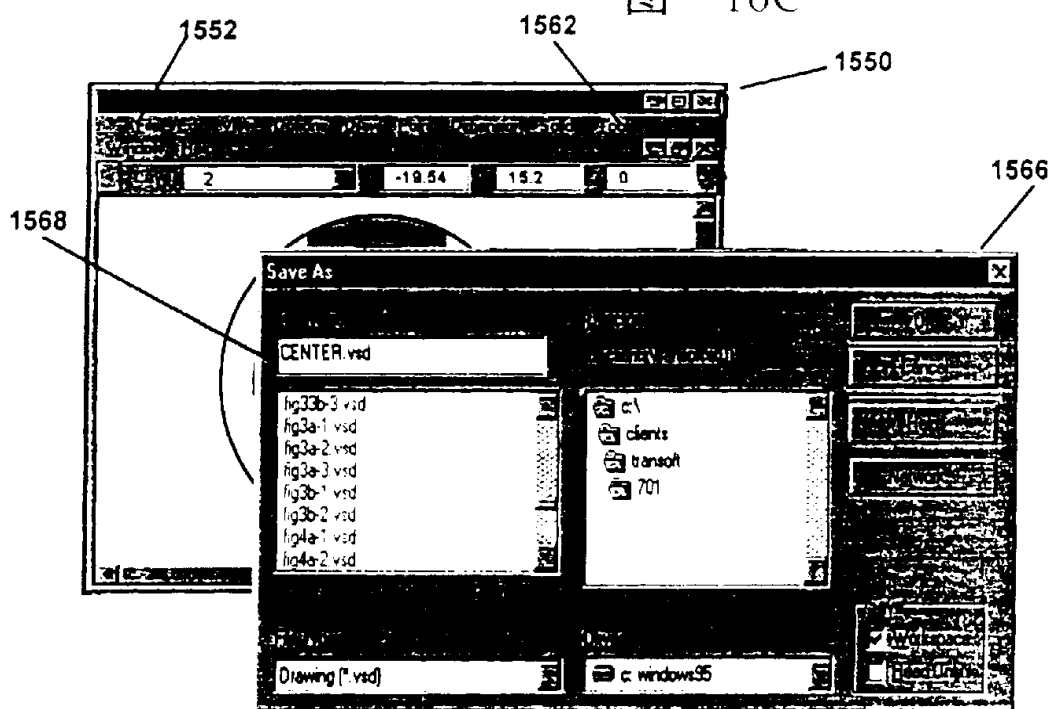


图 16D

创建项目

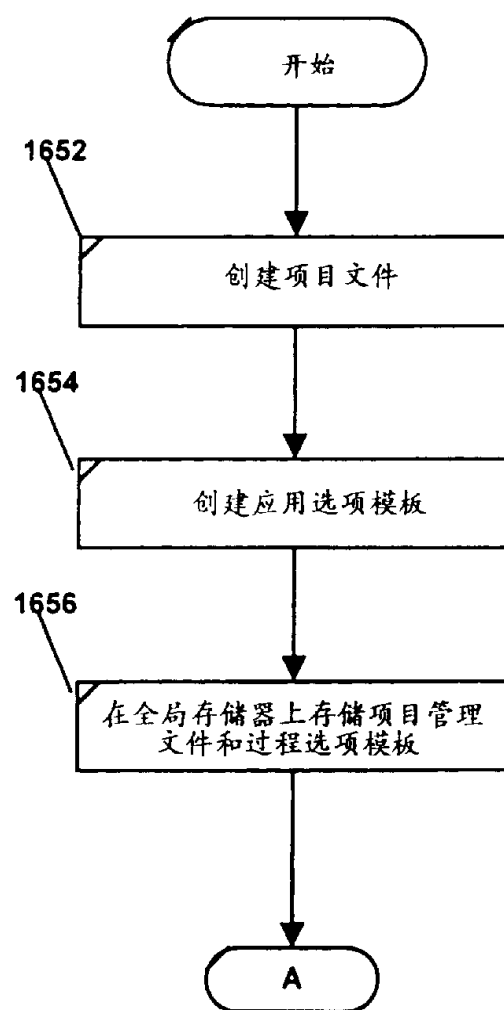


图 17A

更新利用率

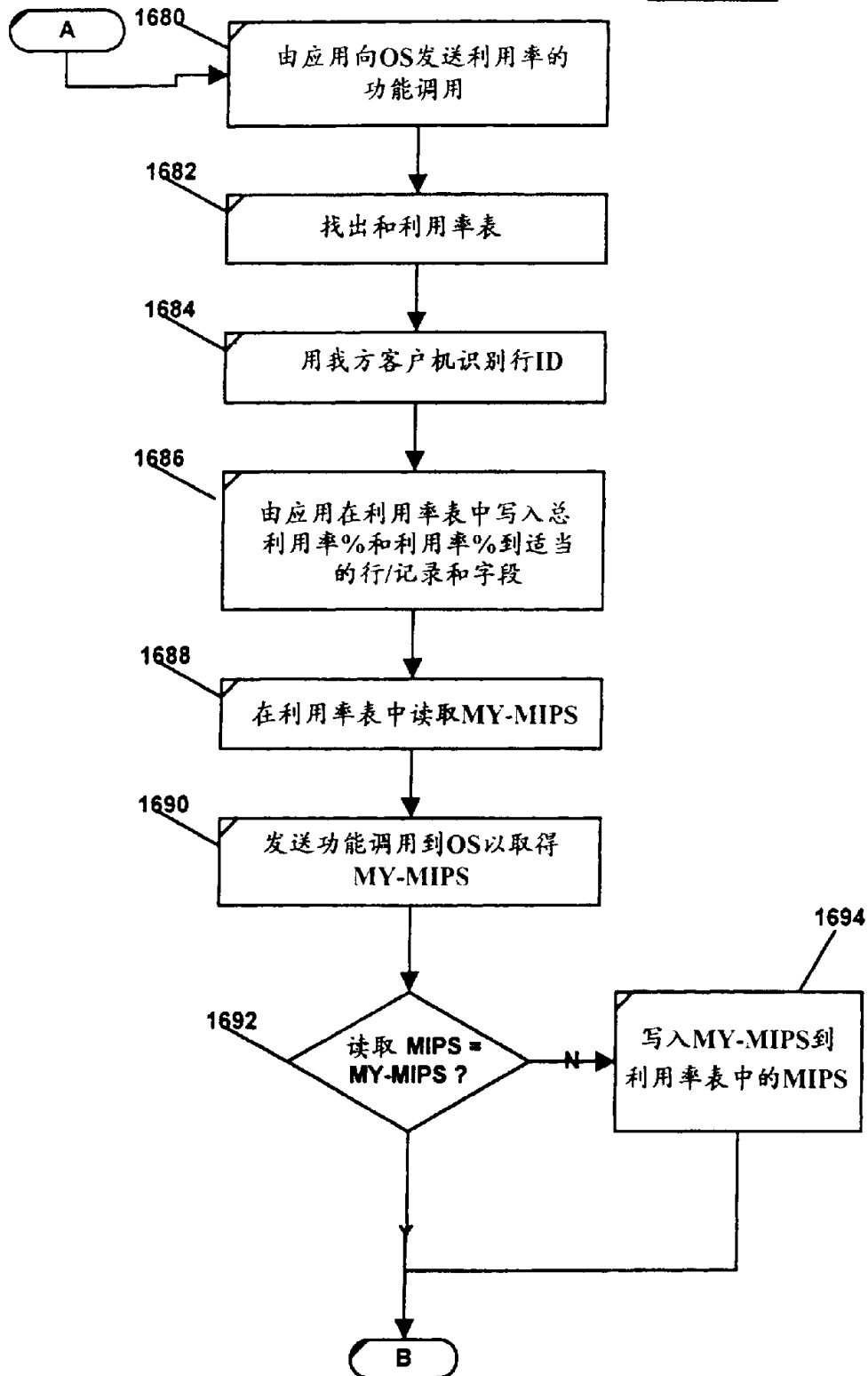


图 17B

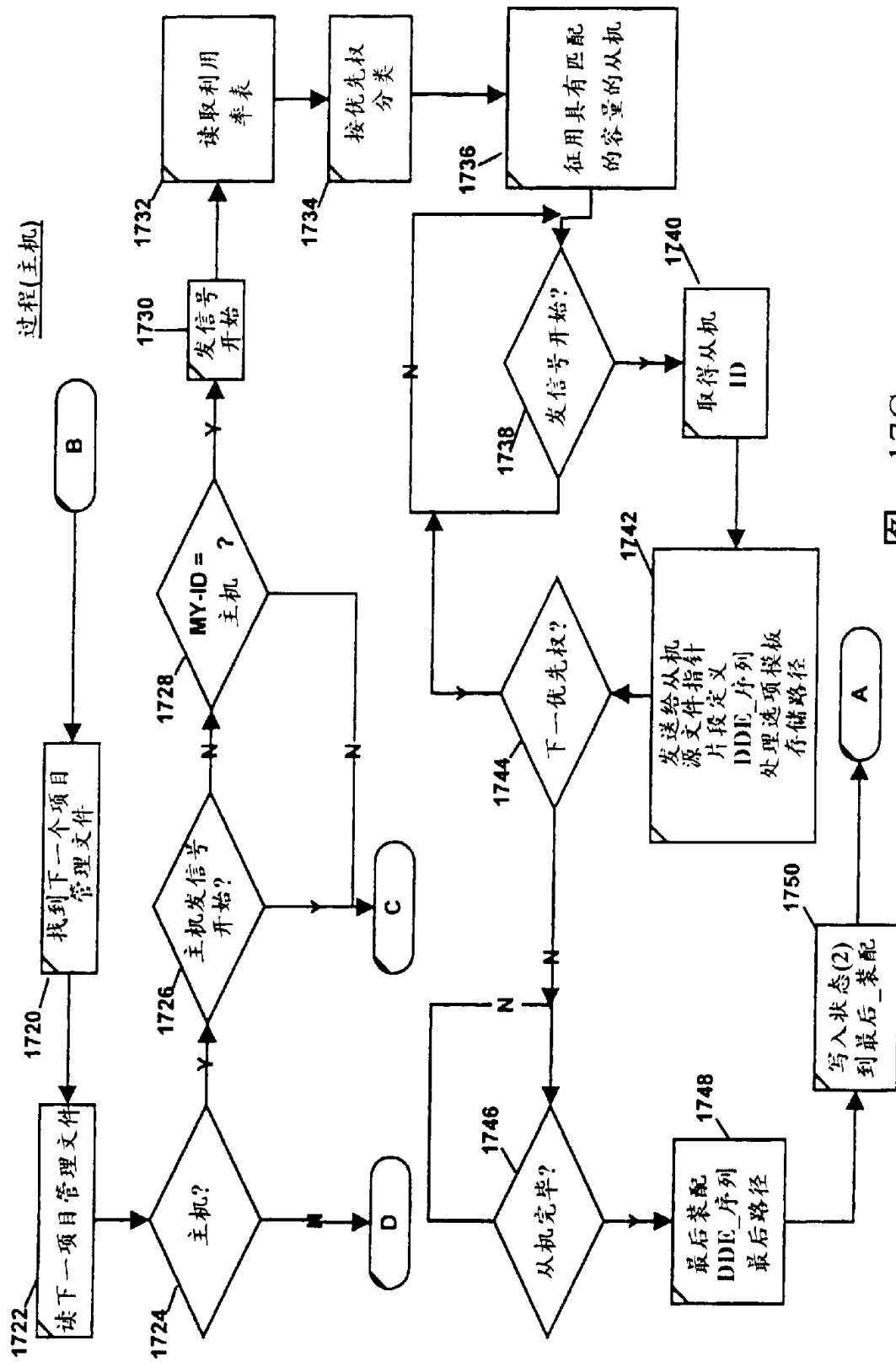


图 17C

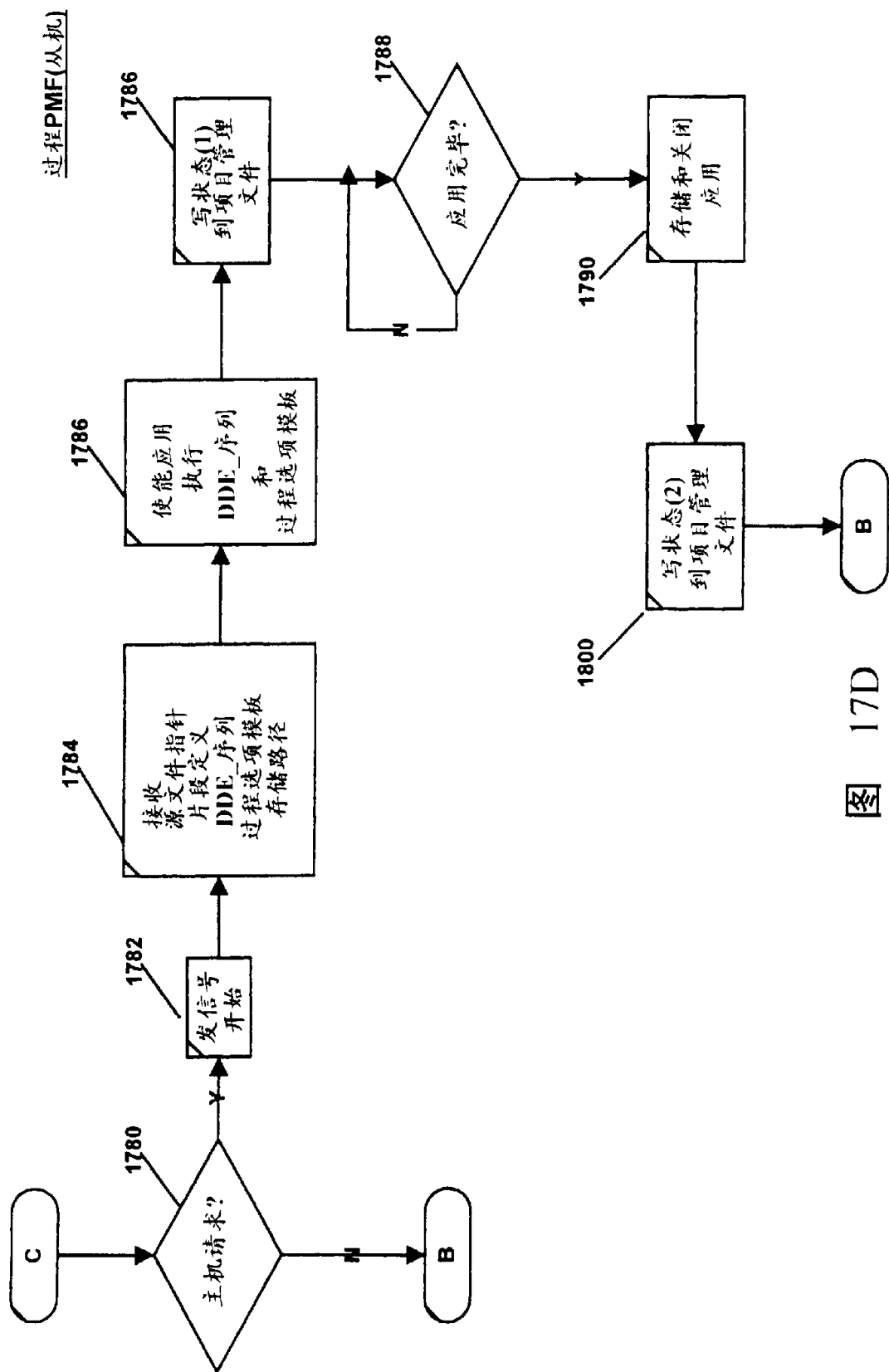


图 17D

过程PMF(PEER)

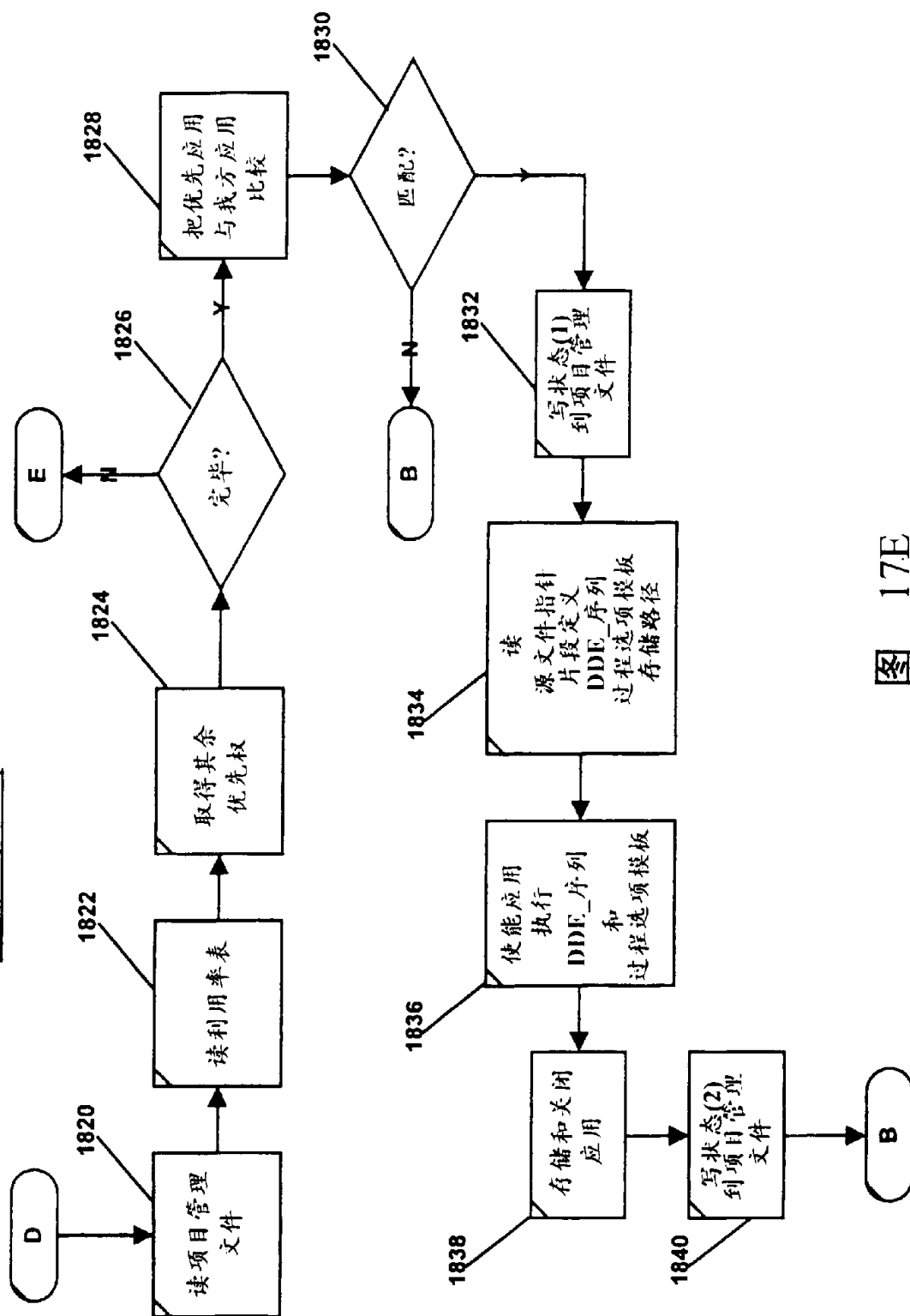


图 17E

过程PMF(同等类似)

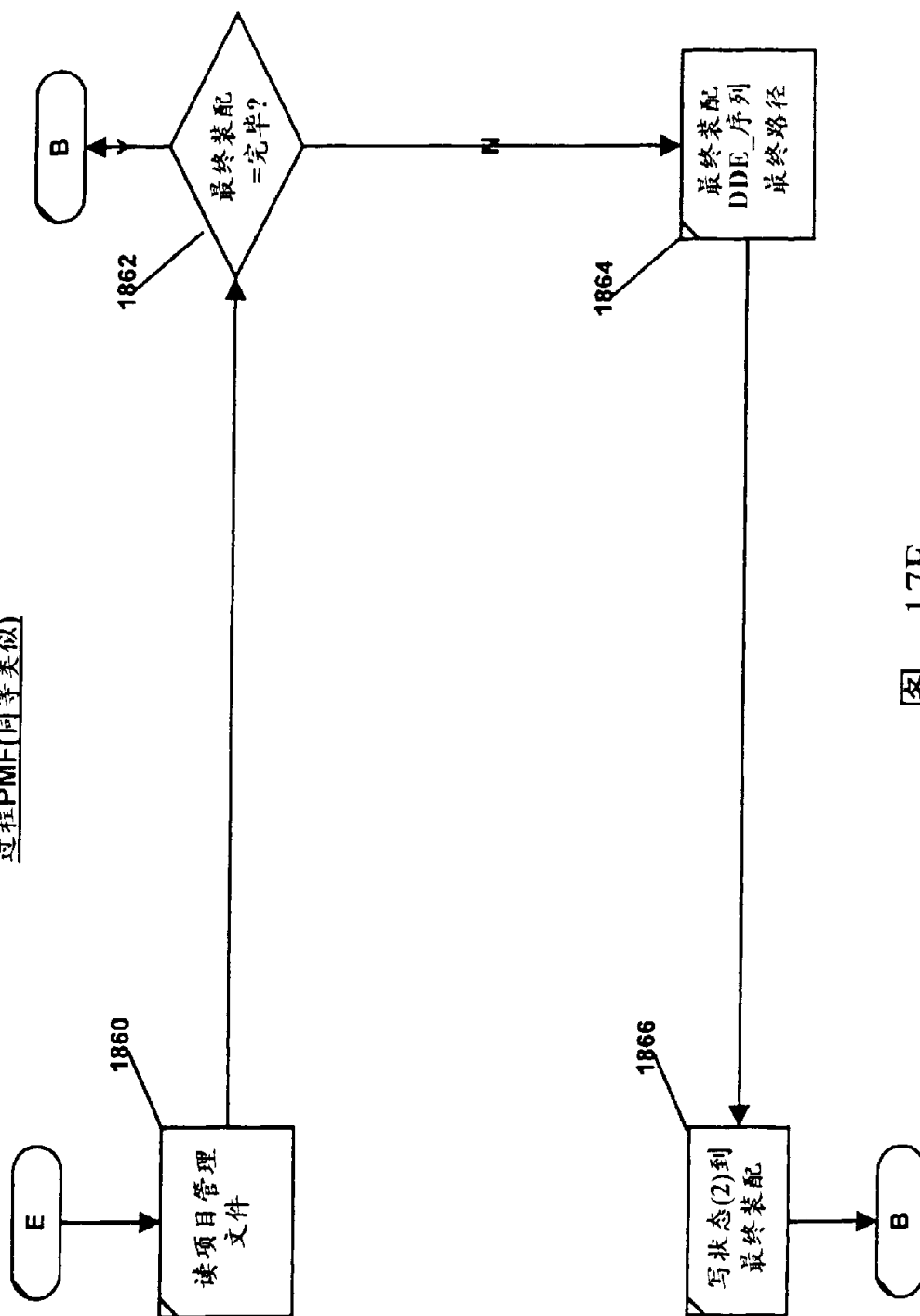


图 17F