



(12) 发明专利申请

(10) 申请公布号 CN 102934092 A

(43) 申请公布日 2013. 02. 13

(21) 申请号 201080063617. 8

代理人 康建峰 李春晖

(22) 申请日 2010. 12. 13

(51) Int. Cl.

(30) 优先权数据

G06F 12/00(2006. 01)

61/286, 093 2009. 12. 14 US

(85) PCT申请进入国家阶段日

2012. 08. 10

(86) PCT申请的申请数据

PCT/IL2010/001049 2010. 12. 13

(87) PCT申请的公布数据

W02011/073976 EN 2011. 06. 23

(71) 申请人 达伊阿斯帕纳有限公司

地址 以色列马加尔

(72) 发明人 埃塔马尔·拉龙

(74) 专利代理机构 北京集佳知识产权代理有限公司

11227

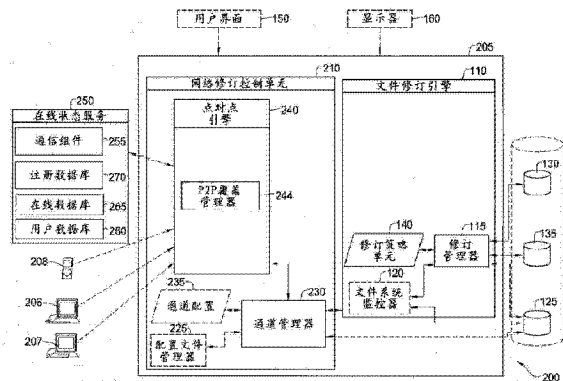
权利要求书 11 页 说明书 46 页 附图 17 页

(54) 发明名称

修订控制系统和方法

(57) 摘要

本发明提供一种文件修订控制系统、方法和设备,其被配置来存储和管理修订历史日志,以及能够追踪和管理、比较并检索修订,所述修订包括存储在指定参考区域上的当前修订、存储在预定义存储区域中的过去修订并且还包含先前在参考区域上的过去修订。根据本发明的特定实施方案,文件的每个修订由修订历史日志中的修订条目表示,所述修订条目可以在指定数据储存库中存储和管理。修订条目存储对应于至少一个修订的数据,而所述修订自身可以存储在指定数据储存库上。进程路径指表示文件随着时间的进程的一个或多个修订条目。如果修订分支成多个修订,那么每个这样的修订将构成不同的进程路径。



1. 一种修订控制系统,其包括:

修订引擎,其与至少一个数据储存库相关并可运行从而能够在实施于所述至少一个数据储存库上的文件系统内定义至少一个参考区域,并且包括:

文件系统监控器,其能够与所述文件系统通信以标识关于所述文件系统内的文件或文件夹的修改事件。

修订管理器,其对所述修改事件作出响应以:

在修订历史日志中创建新修订条目,所述新修订条目对应于所述修改事件;

标识所述修订历史日志中对应于实施所述修改事件所相关的所述文件或文件夹的修订的相关修订条目;和

链接所述新修订条目与所述相关修订条目,从而延伸表示所述文件或文件夹的修订进程的进程路径。

2. 根据权利要求1所述的系统,其中所述修订管理器还对标识沿着与所述相关修订条目相关的进程路径存在对应于与所述相关修订条目相关的修订修改的另一修订条目作出响应,以延伸包括所述相关修订条目和所述新修订条目的新进程路径。

3. 根据权利要求1所述的系统,其中所述修订历史日志中的每个修订条目包括至少一个数据元素并且其中所述修订引擎还可运行以:利用所述至少一个数据元素以在所述修订历史日志中定位具有一个或多个相同数据元素的两个或更多个修订条目;并且基于所述一个或多个数据元素指示分别对应于所述两个或更多个修订条目的两个或更多个修订之间的关联。

4. 根据权利要求3所述的系统,其中所述至少一个数据元素是从对应修订的至少一部分内容中计算出来的内容哈希码,并且所述相关性指示所述两个或更多个修订的所述至少一部分内容相同。

5. 根据权利要求3所述的系统,其中所述至少一个数据元素包括一个或多个元数据元素,所述一个或多个元数据元素的特征在于对应的修订并且其中所述相关性指示所述两个或更多个修订的一个或多个元数据元素相同。

6. 根据权利要求4所述的系统,其中所述修订引擎还可运行以确定所述两个或更多个修订条目是否位于相同进程路径上。

7. 根据权利要求1所述的系统,其中所述修订引擎还可运行以处置针对第一修订执行的恢复命令;所述恢复命令包括:至少一个标识符,其标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征不在于所述修订;所述第二修订条目位于和与所述第一修订相关的第一修订条目相同的进程路径内;

所述修订管理器对所述恢复命令作出响应以:

用所述第二修订中的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素,从而创建新修订;

针对所述新修订创建新修订条目;和

将所述新修订条目存储在所述修订历史日志中。

8. 根据权利要求7所述的系统,其中所述修订管理器被配置来链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而延伸包括所述第二修订条目和所述新修订条目的新进程路径。

9. 根据权利要求 7 所述的系统,其中所述一个或多个数据元素是所述第二修订的内容。

10. 根据权利要求 1 所述的系统,其中所述修订引擎还可运行以处置针对第一修订执行的复制命令;所述复制命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征在于所述修订;所述第二修订条目位于不同于与所述第一修订相关的第一修订条目的进程路径上;

所述修订管理器对所述复制命令作出响应以:

用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的数据元素,从而创建新修订条目;

针对所述新修订创建新修订条目;和链接所述新修订条目到所述修订历史日志。

11. 根据权利要求 10 所述的系统,其中所述修订管理器被配置来链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而延伸包括所述第二修订条目和所述新修订条目的新进程路径。

12. 根据权利要求 1 所述的系统,其中所述修订引擎还可运行以处置针对所述修订历史日志中的两条或更多条进程路径执行的合并命令;所述合并命令至少包括标识所述修订历史日志内的两条或更多条进程路径的信息和标识合并的修订条目的信息;

所述修订管理器对所述合并命令作出响应以:将所述两条或更多条进程路径聚合成所述合并的修订条目并且使所述两条或更多条进程路径以单进程路径继续。

13. 根据权利要求 1 所述的系统,其中所述修订历史日志中的每个修订条目由通用唯一标识符(UUID)唯一标识。

14. 根据权利要求 1 所述的系统,其中所述修订历史日志中的每个修订条目由计算成唯一值的两个或更多个元数据元素的组合唯一标识。

15. 根据权利要求 1 所述的系统,其中对所述修改事件作出响应而在所述参考区域中创建新修订。

16. 根据权利要求 1 所述的系统,其中对在所述参考区域上创建的新文件作出响应触发所述修改事件。

17. 根据权利要求 1 所述的系统,其中所述修改事件通过由下列动作组成的列表中的一个或多个触发:

- a. 删除部分或全部修订内容;
- b. 针对修订改变一个或多个元数据元素;
- c. 重新命名修订;
- d. 将修订移动到新位置;和
- e. 删除修订。

18. 根据权利要求 1 所述的修订控制系统,其中所述修订引擎适于和与第二数据储存库相关的第二修订引擎通信;所述第二数据储存库被定义有第二参考区域;

所述修订引擎可运行以从所述第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示,所述第二修订历史日志与所述第二修订引擎相关;所述第一修订条目表示对所述第二参考区域中的修改事件作出响应而生成的修订;

所述修订管理器对关于所述第一修订条目的所述指示作出响应以:

标识所述第一修订条目是否从与所述修订引擎相关的所述修订历史日志中缺失,并且如果缺失,那么:

创建第二修订条目,所述第二修订条目对应于所述修订;和

链接所述第二修订条目到所述修订历史日志内的第三修订条目,从而延伸表示对所述第一参考区域和所述第二参考区域中的修改事件作出响应而生成的修订进程的进程路径。

19. 根据权利要求 18 所述的系统,其中所述修订管理器还作出响应来标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目,并且其中所述第二修订条目的所述链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

20. 根据权利要求 18 所述的系统,其中所述修订引擎可运行以从所述第二修订引擎请求由所述第一修订条目表示的所述修订。

21. 根据权利要求 20 所述的系统,其中所述修订从所述第二参考区域中检索。

22. 根据权利要求 18 所述的系统,其中所述参考区域在实施于所述第二数据储存库上的文件系统内。

23. 根据权利要求 18 所述的系统,其中所述指示包括标识所述第三修订条目的数据。

24. 根据权利要求 18 所述的系统,其中如果所述修订管理器未能在所述修订历史日志中标识所述第三修订条目,那么所述修订引擎可运行以:

创建请求关于所述第一修订条目和所述第二修订条目的进程路径中的至少一个另外修订条目的缺失信息的询问;

至少将所述询问发送给所述第二修订引擎;和

从所述至少所述第二修订引擎接收指示所述至少一个另外修订条目的新指示。

25. 根据权利要求 18 所述的系统,其还包括网络控制单元,所述网络控制单元可运行以实现与所述第二修订引擎的通信。

26. 根据权利要求 18 所述的系统,其中所述修订引擎和所述第二修订引擎经由直接连接而连接。

27. 根据权利要求 26 所述的系统,其中所述直接连接是 USB 连接。

28. 根据权利要求 18 所述的系统,其中所述修订引擎和所述第二修订引擎经由点对点网络连接。

29. 一种修订控制系统,其包括:

第一修订引擎,其与第一存储区域相关并且适于从与第二存储区域相关的第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示,所述第二修订历史日志与所述第二修订引擎相关;所述第一修订条目表示对所述第二存储区域中的修改事件作出响应而生成的第一修订;

所述第一修订引擎包括:

修订管理器,其适于管理第一修订历史日志以存储关于对所述第一存储区域中的修改事件作出响应而生成的修订的至少一个修订条目;

所述修订管理器对关于所述第一修订条目的所述指示作出响应以:

标识所述第一修订条目是否从与所述第一修订引擎相关的所述第一修订历史日志中缺失,并且如果缺失,那么:

创建第二修订条目,所述第二修订条目对应于所述第一修订;和  
链接所述第二修订条目到所述第一修订历史日志内的第三修订条目,从而延伸表示所述第一存储区域和所述第二存储区域内的修订进程的进程路径;

其中所述修订管理器还作出响应以标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目,并且其中所述第二修订条目的所述链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

30. 根据权利要求 29 所述的系统,其中所述指示包括标识所述第三修订条目的信息。

31. 根据权利要求 29 所述的系统,其中所述第三修订条目表示所述第一修订的先前修订。

32. 根据权利要求 29 所述的系统,其中所述第一修订引擎可运行以从所述第二修订引擎请求由所述第一修订条目表示的所述第一修订,并且将所述修订存储在与所述第一修订引擎相关的数据储存库中。

33. 根据权利要求 32 所述的系统,其中所述第一修订从在所述第二存储区域中的文件系统中定义的参考区域中检索。

34. 根据权利要求 29 所述的系统,其还包括网络修订控制单元,并且其中如果所述修订管理器未能在所述第一修订历史日志中标识所述第三修订条目,那么所述第一修订引擎运行所述网络修订控制单元以创建请求关于所述第二修订历史日志中的所述第一修订条目的进程路径中的至少一个另外修订条目的缺失信息的询问,并且至少将所述询问发送到所述第二修订引擎。

35. 根据权利要求 29 所述的系统,其中所述指示包括关于所述第二修订历史日志中的多个修订条目的信息。

36. 根据权利要求 29 所述的系统,其中对由所述修订管理器生成的拉式更新询问作出响应而接收所述指示并将所述指示传输到所述第二修订引擎。

37. 根据权利要求 29 所述的系统,其中所述指示是由所述第二修订引擎产生的推式更新消息。

38. 根据权利要求 29 所述的系统,其还包括网络控制单元,所述网络控制单元可运行以至少实现与所述第二修订引擎的通信。

39. 根据权利要求 38 所述的系统,其为通道中的成员并可运行以连接到在线状态服务以获得所述通道中的一个或多个其它成员的列表。

40. 根据权利要求 29 所述的系统,其中所述第一修订引擎和所述第二修订引擎经由直接连接而连接。

41. 根据权利要求 29 所述的系统,其中在实施于数据储存库上的文件系统内定义所述第一存储区域。

42. 根据权利要求 38 所述的系统,其中所述第一修订引擎和所述第二修订引擎经由点对点网络连接。

43. 根据权利要求 42 所述的系统,其还包括通道覆盖管理器,所述通道覆盖管理器可运行以管理分布式哈希表架构中的所述点对点的所述点对点网络覆盖。

44. 根据权利要求 29 所述的系统,其中所述第一修订历史日志中的每个修订条目包括至少一个数据元素并且其中所述第一修订引擎还可运行以:利用所述至少一个数据元素以

在所述第一修订历史日志内定位具有一个或多个相同数据元素的两个或更多个修订条目；并且基于所述一个或多个数据元素指示分别对应于所述两个或更多个修订条目的两个或更多个修订之间的相关性。

45. 根据权利要求 44 所述的系统,其中所述至少一个数据元素是从对应修订的至少部分内容中计算出来的内容哈希码并且所述相关性指示所述两个或更多个修订的所述至少部分内容相同。

46. 根据权利要求 44 所述的系统,其中所述至少一个数据元素包括一个或多个元数据元素,所述一个或多个元数据元素的特征在于对应修订并且其中所述相关性指示所述两个或更多个修订的一个或多个元数据元素相同。

47. 根据权利要求 29 所述的系统,其中所述第一修订引擎还可运行以处置针对第一修订执行的恢复命令;所述恢复命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征不在于所述修订;所述第二修订条目位于和与所述第一修订相关的第一修订条目相同的进程路径内;

所述修订管理器对所述恢复命令作出响应以:用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素,从而创建新修订;

针对所述新修订创建新修订条目;和

链接所述新修订条目到所述第一修订历史日志。

48. 根据权利要求 47 所述的系统,其中所述修订管理器被配置来链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而创建包括所述第二修订条目和所述新修订条目的新进程路径。

49. 根据权利要求 47 所述的系统,其中所述一个或多个数据元素为所述第二修订的内容。

50. 根据权利要求 29 所述的系统,其中所述第一修订引擎还可运行以处置针对第一修订执行的复制命令;所述复制命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征不在于所述修订;所述第二修订条目位于不同于与所述第一修订相关的第一修订条目的进程路径上;

所述修订管理器对所述复制命令作出响应以:用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的数据元素,从而创建新修订;针对所述新修订创建新修订条目;和链接所述新修订条目到所述第一修订历史日志。

51. 根据权利要求 50 所述的系统,其中所述修订管理器被配置来链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而创建包括所述第二修订条目和所述新修订条目的新进程路径。

52. 根据权利要求 29 所述的系统,其中所述第一修订引擎还可运行以处置针对所述第一修订历史日志中的两条或更多条进程路径执行的合并命令;所述合并命令至少包括标识所述第一修订历史日志内的两条或更多条进程路径的信息和标识合并的修订条目的信息;

所述修订管理器对所述合并命令作出响应以:将两条或更多条进程路径聚合成所述合并的修订条目并使所述两条或更多条进程路径都以单进程路径继续。

53. 根据权利要求 29 所述的系统,其中所述第一修订历史日志中的每个修订条目由通用唯一标识符 (UUID) 唯一标识。

54. 根据权利要求 29 所述的系统,其中所述第一修订历史日志中的每个修订条目由计算为唯一值的元数据元素的组合唯一标识。

55. 一种运行包括修订引擎的修订控制系统的方法,所述方法包括:

在文件系统内定义至少一个参考区域,所述文件系统实施在数据储存库上;

与所述文件系统通信并标识关于所述文件系统内的文件或文件夹的修改事件;

对所述修改事件作出响应;

在修订历史日志中创建新修订条目,所述新修订条目对应于所述修改事件;

标识所述修订历史日志中对应于实施所述修改事件所相关的所述文件或文件夹的修订的相关修订条目;和

链接所述新修订条目与所述相关修订条目,从而延伸表示所述文件或文件夹的修订进程的进程路径。

56. 根据权利要求 55 所述的方法,其还包括:

标识沿着与所述相关修订条目相关的进程路径存在对应于与所述相关修订条目相关的修订修改的另一修订条目;和

延伸包括所述相关修订条目和所述新修订条目的新进程路径。

57. 根据权利要求 55 所述的方法,其中所述修订历史日志中的每个修订条目包括至少一个数据元素,所述方法还包括:

利用所述至少一个数据元素以在所述修订历史日志内定位具有一个或多个相同数据元素的两个或更多个修订条目;和

基于所述一个或多个数据元素指示分别对应于所述两个或更多个修订条目的两个或更多个修订之间的相关性。

58. 根据权利要求 57 所述的方法,其中所述至少一个数据元素是从对应修订的至少部分内容中计算出来的内容哈希码并且所述相关性指示所述两个或更多个修订的所述至少部分内容相同。

59. 根据权利要求 57 所述的方法,其中所述至少一个数据元素包括一个或多个元数据元素,所述一个或多个元数据元素的特征在于对应修订并且其中所述相关性指示所述两个或更多个修订的一个或多个元数据元素相同。

60. 根据权利要求 57 所述的方法,其还包括:

确定所述两个或更多个修订条目是否位于相同进程路径上。

61. 根据权利要求 55 所述的方法,其还包括:

处置针对第一修订执行的恢复命令;所述恢复命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征不在于所述修订;所述第二修订条目位于和与所述第一修订相关的第一修订条目相同的进程路径内;

所述处置包括:

用所述第二修订的对应数据元素替换对应于所述第一修订中所述类型的数据元素的一个或多个数据元素,从而创建新修订;

针对所述新修订创建新修订条目;和

将所述新修订条目存储在所述修订历史日志中。

62. 根据权利要求 61 所述的方法,其中还包括:

链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而延伸包括所述第二修订条目和所述新修订条目的新进程路径。

63. 根据权利要求 61 所述的方法,其中所述一个或多个数据元件是所述第二修订的内容。

64. 根据权利要求 55 所述的方法,其还包括:

处置针对第一修订执行的复制命令;所述复制命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征在于所述修订;所述第二修订条目位于不同于与所述第一修订相关的第一修订条目的进程路径上;

所述处置包括:

用所述第二修订的对应数据元素替换关于所述第一修订的对应于所述一种或多种类型的数据元素的数据元素,从而创建新修订;

针对所述新修订创建新修订条目;和链接所述新修订条目到所述修订历史日志。

65. 根据权利要求 64 所述的方法,其还包括:

链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而延伸包括所述第二修订条目和所述新修订条目的新进程路径。

66. 根据权利要求 55 所述的方法,其中还包括:

处置针对所述修订历史日志中的两条或更多条进程路径执行的合并命令;所述合并命令至少包括标识所述修订历史日志内的两条或更多条进程路径的信息和标识合并的修订条目的信息;

所述处置包括:

将所述两条或更多条进程路径聚合成所述合并的修订条目并且使所述两条或更多条进程路径以单进程路径继续。

67. 根据权利要求 55 所述的方法,其中所述修订历史日志中的每个修订条目由通用唯一标识符(UUID)唯一标识。

68. 根据权利要求 55 所述的方法,其中所述修订历史日志中的每个修订条目由计算为唯一值的两个或更多个元数据元素的组合唯一标识。

69. 根据权利要求 55 所述的方法,其中对所述修改事件作出响应而在所述参考区域中创建新修订。

70. 根据权利要求 55 所述的方法,其中对在所述参考区域上创建的新文件作出响应触发所述修改事件。

71. 根据权利要求 55 所述的方法,其中通过由下列动作组成的列表中的一个或多个触发所述修改事件:

- a. 删除部分或全部修订内容;
- b. 针对修订改变一个或多个元数据元素;
- c. 重新命名修订;
- d. 将修订移动到新位置;和
- e. 删除修订。



72. 根据权利要求 55 所述的方法,其还包括:

能够在所述修订引擎和与第二数据储存库相关的第二修订引擎之间通信;所述第二数据储存库被定义有第二参考区域;

从所述第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示,所述第二修订历史日志与所述第二修订引擎相关;所述指示包括关于对所述第二参考区域中的修改事件作出响应而生成的修订的数据;

对关于所述第一修订条目的所述指示作出响应:

标识所述第一修订条目是否从与所述修订引擎相关的所述修订历史日志中缺失,并且如果缺失,那么:

创建第二修订条目,所述第二修订条目对应于所述修订;和

链接所述第二修订条目到所述修订历史日志内的第三修订条目,从而延伸表示对所述第一参考区域和所述第二参考区域中的修改事件作出响应而生成的修订进程的进程路径。

73. 根据权利要求 72 所述的方法,其还包括:

标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目,并且其中所述第二修订条目的所述链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

74. 根据权利要求 72 所述的方法,其还包括:

从所述第二修订引擎请求由所述第二修订条目表示的修订。

75. 根据权利要求 72 所述的方法,其中所述参考区域在实施于所述第二数据储存库上的文件系统内。

76. 根据权利要求 72 所述的方法,其中所述指示包括标识所述第三修订条目的数据。

77. 根据权利要求 72 所述的方法,其中如果未在所述修订历史日志中标识所述第三修订条目,那么所述方法还包括:

创建请求关于所述第一修订条目和所述第二修订条目的进程路径中的至少一个另外修订条目的缺失信息的询问;

至少将所述询问发送到所述第二修订引擎;和

至少从所述第二修订引擎接收指示所述至少一个另外修订条目的新指示。

78. 根据权利要求 72 所述的方法,其中所述修订引擎和所述第二修订引擎之间的所述通信是经由直接连接。

79. 根据权利要求 72 所述的方法,其中所述修订引擎与所述第二修订引擎之间的所述通信是经由点对点网络。

80. 一种运行包括与第一存储区域相关的修订引擎的修订控制系统的方法,所述修订引擎包括修订管理器,所述修订管理器适于管理第一修订历史日志以存储关于对所述第一存储区域中的修改事件作出响应而生成的修订的至少一个修订条目,所述方法包括:

从与第二存储区域相关的第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示,所述第二修订历史日志与所述第二修订引擎相关;所述指示包括对所述第二存储区域中的修改事件作出响应而生成的第一修订的数据;和

对关于所述第一修订条目的所述指示作出响应:

标识所述第一修订条目是否从与所述第一修订引擎相关的所述第一修订历史日志中

缺失,并且如果缺失,那么:

创建第二修订条目,所述第二修订条目对应于所述第一修订;和

链接所述第二修订条目到所述第一修订历史日志内的第三修订条目,从而延伸表示所述第一存储区域和所述第二存储区域内的修订进程的进程路径;和

标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目,并且其中所述第二修订条目的所述链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

81. 根据权利要求 80 所述的方法,其中所述指示包括标识所述第三修订条目的信息。

82. 根据权利要求 80 所述的方法,其中所述第三修订条目表示所述第一修订的先前修订。

83. 根据权利要求 80 所述的方法,其还包括:

从所述第二修订引擎请求由所述第二修订条目表示的修订并将所述修订存储在与所述第一修订引擎相关的数据储存库中。

84. 根据权利要求 80 所述的方法,其中如果未在所述第一修订历史日志中标识所述第三修订条目,那么所述方法还包括:

创建请求所述第二修订历史日志中的所述第一修订条目的进程路径中的至少一个另外修订条目的询问,并至少将所述询问发送到所述第二修订引擎。

85. 根据权利要求 80 所述的方法,其中所述指示包括关于所述第二修订历史日志中的多个修订条目的信息。

86. 根据权利要求 80 所述的方法,其中对由所述修订管理器生成的拉式更新询问作出响应而接收所述指示并将所述指示传输到所述第二修订工具。

87. 根据权利要求 80 所述的方法,其中所述指示是由所述第二修订引擎生成的推式更新消息。

88. 根据权利要求 80 所述的方法,其中所述修订引擎和所述第二修订引擎之间的所述通信是经由直接连接。

89. 根据权利要求 80 所述的方法,其中所述修订引擎和所述第二修订引擎之间的所述通信是经由点对点网络。

90. 根据权利要求 89 所述的方法,其还包括:

管理分布式哈希表架构中的所述点对点网络的点对点网络覆盖。

91. 根据权利要求 80 所述的方法,其中所述第一修订历史日志中的每个修订条目包括至少一个数据元素并且其中所述方法还包括:

在所述至少一个数据元素的帮助下,在所述第一修订历史日志内定位具有一个或多个相同数据元素的两个或更多个修订条目;和

基于所述一个或多个数据元素指示分别对应于所述两个或更多个修订条目的两个或更多个修订之间的相关性。

92. 根据权利要求 91 所述的方法,其中所述至少一个数据元素是从对应修订的至少部分内容中计算出来的内容哈希码并且所述相关性指示所述两个或更多个修订的所述至少部分内容相同。

93. 根据权利要求 91 所述的方法,其中所述至少一个数据元素包括一个或多个元数据

元素,所述一个或多个元数据元素的特征在于对应修订并且其中所述相关性指示所述两个或更多个修订的一个或多个元数据元素相同。

94. 根据权利要求 80 所述的方法,其还包括:

处置针对第一修订执行的恢复命令;所述恢复命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征不在于所述修订;所述第二修订条目位于和与所述第一修订相关的第一修订条目相同的进程路径内;

所述处置包括:

用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素;

针对所述新修订创建新修订条目;和

链接所述新修订条目到所述第一修订历史日志。

95. 根据权利要求 94 所述的方法,其还包括:

链接所述新修订条目到对应于所述第二修订条目的所述第二修订条目,从而创建包括所述第二修订条目和所述新修订条目的新进程路径。

96. 根据权利要求 94 所述的方法,其中所述一个或多个数据元素是所述第二修订的内容。

97. 根据权利要求 80 所述的方法,其还包括:

处置针对第一修订执行的复制命令;所述复制命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征不在于所述修订;所述第二修订条目位于不同于与所述第一修订相关的第一修订条目的进程路径上;

所述处置包括:

用所述第二修订的对应数据元件替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素,从而创建新修订;针对所述新修订创建新修订条目;并链接所述新修订条目到所述第一修订历史日志。

98. 根据权利要求 97 所述的方法,其还包括:

链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而创建包括所述第二修订条目和所述新修订条目的新进程路径。

99. 根据权利要求 80 所述的方法,其还包括:

处置针对所述第一修订历史日志中的两条或更多条进程路径执行的合并命令;所述合并命令至少包括标识所述第一修订历史日志内的两条或更多条进程路径的信息和标识合并的修订条目的信息;

所述处置包括:

将所述两条或更多条进程路径聚合成所述合并的修订条目并且使所述两条或更多条进程路径都以单进程路径继续。

100. 根据权利要求 80 所述的方法,其中所述第一修订历史日志中的每个修订条目由通用唯一标识符(UUID)唯一标识。

101. 根据权利要求 80 所述的方法,其中所述第一修订历史日志中的每个修订条目由计算为唯一值的元数据元素的组合唯一标识。

102. 一种可由机器读取的程序存储装置,其有形地具体实施为可由所述机器执行以执

行运行包括修订引擎的修订控制系统的方法步骤的指令程序,所述方法包括:

在文件系统内定义至少一个参考区域,所述文件系统实施于数据储存库上;  
与所述文件系统通信并标识关于所述文件系统内的文件或文件夹的修改事件;  
对所述修改事件作出响应;

在修订历史日志中创建新修订条目,所述新修订条目对应于所述修改事件;

标识所述修订历史日志中对应于实施所述修改事件所相关的所述文件或文件夹的修订的相关修订条目;和

链接所述新修订条目和所述相关修订条目,从而延伸表示所述文件或文件夹的修订进程的进程路径。

103. 一种可由机器读取的程序存储装置,其有形地具体实施为可由所述机器执行以执行运行包括与第一存储区域相关的修订引擎的修订控制系统的方法的方法步骤的指令程序,所述修订引擎包括修订管理器,所述修订管理器适于管理第一修订历史日志以存储关于对所述第一存储区域中的修改事件作出响应而生成的修订的至少一个修订条目,所述方法包括:

从与第二存储区域相关的第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示,所述第二修订历史日志与所述第二修订引擎相关;所述指示包括关于对所述第二存储区域中的修改事件作出响应而生成的第一修订的数据;和

对关于所述第一修订条目的所述指示作出响应;

标识所述第一修订条目是否从与所述第一修订引擎相关的所述第一修订历史日志中缺失,并且如果缺失,那么:

创建第二修订条目,所述第二修订条目对应于所述第一修订;和

链接所述第二修订条目到所述第一修订历史日志内的第三修订条目,从而延伸表示所述第一存储区域和所述第二存储区域内的修订进程的进程路径;和

标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目,并且其中所述第二修订条目的所述链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

## 修订控制系统和方法

### 技术领域

[0001] 本发明涉及文件修订控制领域并且还涉及文件共享环境中文件修订控制和文件更新领域。

### [0002] 发明背景

[0003] 现今存在很多类型的能够在用户之间进行文件共享的系统。通常在这类系统中，存在关于每个文件的单个修订进程。因此，虽然一个用户连接到系统并且正在位于系统的文件共享服务器上的文件上工作，但是可以实施不同方法来防止其它用户对所述文件的同时访问（例如，文件锁定）。如果其它用户在离线时在相同文件的副本上工作并且每个用户创建所述文件的新修订，那么一旦用户重新连接到系统，立即执行冲突解决程序，其中维持一个修订并且丢弃其它修订。

[0004] 下文列出被视为相关而作为本发明背景的现有技术引用。本文中引用的确认不被推断为意味这些引用以任何方式与本文所公开的发明的专利权相关。

[0005] Aboulhosn 等人的美国专利 US6, 938, 042 公开一种用于在一组计算机系统之间共享文件的方法和系统。所述文件共享系统允许定义一组计算机系统。由一个组共享的文件与一个组文件夹相关。组文件夹由所述组的每个成员的文件系统创建的文件夹表示。每个成员处的文件夹包含所述组所共享的每个文件的文件。一个成员的文件夹中的文件可以是存储在成员处、包含共享文件内容的实际文件，或者可以是标识存储在另一成员处的实际文件的虚拟文件。当成员访问虚拟文件时，文件共享系统检测到访问并且依据点对点技术请求文件所有者提供文件副本给访问成员。每当共享文件被修改时，文件所有者发送所述文件的更新元数据给所述组的其它成员。

[0006] Li 等人的美国专利申请公开案第 2008/0005188 号公开文件共享环境中的内容同步。圈子共享中的节点维持表示节点对文件同步事件的了解的版本向量。版本向量数据可以经由呈短通知消息形式的在线状态服务在所述圈子共享的节点之间传输。所述版本向量数据经过比较来标识待同步的一个或多个节点。表示圈子共享的资源感知覆盖被计算出来并且被用于起始通信以确定哪些可用同步节点将提供最多同步值给所述圈子共享中的其它节点。

[0007] Li Jin 的美国专利申请第 2008/0005195 号公开一种“大量文件共享器”（MFS）。所述 MFS 提供大量 P2P 文件共享协议，其具备可选的、跨 P2P 网络的按需文件访问。根据 US2008/0005195，与传统 P2P 文件共享不同，所述 MFS 同时非同步地共享大量文件，同时提供对共享文件的可选按需（同步）访问。根据 US2008/0005195，所述 MFS 结合同级间文件 / 文件夹版本分析和同级间文件 / 文件夹可用性分析使用唯一元数据结构来共享可以包括任何数量的文件夹、子文件夹和文件的复杂目录结构。共享可以是非同步和 / 或同步。具体来说，文件共享通常实现为跨网络的非同步分布。然而，当特定同级想要直接访问特定文件时，使用按需访问模式来实现共享文件的同步递送。这种非同步文件分布和同步文件分布的组合使 MFS 能够支持许多应用，例如随选电影观看、文件 / 文件夹浏览等等。

[0008] Sharon Carmel 的美国专利申请公开案第 2009/0172201 号公开一种在同步网络

成员之间实现点对点同步的方法和系统。提供网络的每个成员上的预定义同步参考区域。提供与同步网络相关的共同标识符给每个成员。在成员上检测到关于数据项的改变。获得同步模块的唯一标识符。获得所述预定义同步参考区域内的数据项的相对路径。基于与所述改变相关的数据项版本的内容来计算唯一值。累加日志编号计数器。创建表示数据项和偶然性的日志。所述日志包括关于改变类型、网络标识符、同步模块的唯一标识符、相对路径、唯一值和日志编号的数据。

[0009] 美国专利第 7,523,146 号公开一种在多个存储器之间同步数据的设备和方法。版本历史与每个存储器中的数据相关。所述版本历史具有一个或多个条目,并且每个条目具有一个标识符和一个值。所述标识符标识已修改所述数据的存储器,并且所述值指示由所述存储器对数据所作的修改数目。当在存储器之间同步数据时,数据的版本历史经过比较来确定一个版本历史是否从属于另一版本历史。所述存储器中具有所述从属版本历史的数据接着用具有主导版本历史的数据替换。当进行比较时,如果版本历史不相同,如果版本历史并非全部具有相同标识符,并且如果一个版本历史不包含具有等于或大于另一版本历史中的值的值的所有标识符,那么需要由用户解决冲突。

## 发明概要

[0010] 根据本发明的第一个方面,提供一种修订控制系统,其包括:修订引擎,其与至少一个数据储存库相关并且可运行从而能够在实施于所述至少一个数据储存库上的文件系统内定义至少一个参考区域,并且包括:文件系统监控器,其能够与文件系统通信以标识关于文件系统内的文件或文件夹的修改事件;修订管理器,其对所述修改事件作出响应以:

[0011] 在修订历史日志中创建新修订条目,所述新修订条目对应于修改事件;标识修订历史日志中对应于实施修改事件所相关的文件或文件夹的修订的相关修订条目;并且链接所述新修订条目与所述相关修订条目,从而延伸表示文件或文件夹的修订进程的进程路径。

[0012] 根据当前公开标的的修订控制系统还对标识沿着与所述相关修订条目相关的进程路径存在对应于与所述相关修订条目相关的修订修改的另一修订条目作出响应,以延伸包括所述相关修订条目和所述新修订条目的新进程路径。

[0013] 根据当前公开标的的修订控制系统,其中修订历史日志中的每个修订条目包括至少一个数据元素并且其中所述修订引擎还可运行以:利用所述至少一个数据元素以在所述修订历史日志中定位具有一个或多个相同数据元素的两个或更多个修订条目;并且基于所述一个或多个数据元素指示分别对应于所述两个或更多个修订条目的两个或更多个修订之间的相关性。

[0014] 根据当前公开标的的修订控制系统,其中所述至少一个数据元素是从对应修订的至少部分内容中计算出来的内容哈希码并且所述相关性指示所述两个或更多个修订的所述至少部分内容相同。

[0015] 根据当前公开标的的修订控制系统,其中所述至少一个数据元素包括一个或多个元数据元素,所述一个或多个元数据元素的特征在于对应修订,并且其中所述相关性指示所述两个或更多个修订的一个或多个元数据元素相同。

[0016] 根据当前公开标的的修订控制系统,其中所述修订引擎还可运行以确定所述两个

或更多个修订条目是否位于相同进程路径上。

[0017] 根据当前公开标的的修订控制系统,其中所述修订引擎还可运行以处置针对第一修订执行的恢复命令;所述恢复命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征在于所述修订;所述第二修订条目位于和与所述第一修订相关的第一修订条目相同的进程路径内;所述修订管理器对所述恢复命令作出响应以:用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素,从而创建新修订;针对所述新修订创建新修订条目;并且将所述新修订条目存储在所述修订历史日志中。

[0018] 根据当前公开标的的修订控制系统,其中所述修订管理器被配置来链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而延伸包括所述第二修订条目和所述新修订条目的新进程路径。

[0019] 根据当前公开标的的修订控制系统,其中所述一个或多个数据元素是所述第二修订的内容。

[0020] 根据当前公开标的的修订控制系统,其中所述修订引擎还可运行以处置针对第一修订执行的复制命令;所述复制命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征在于所述修订;所述第二修订条目位于不同于与所述第一修订相关的第一修订条目的进程路径上;所述修订管理器对所述复制命令作出响应以:用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素,从而创建新修订;针对所述新修订创建新修订条目;并且链接所述新修订条目到所述修订历史日志。

[0021] 根据当前公开标的的修订控制系统,其中所述修订管理器被配置来链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而延伸包括所述第二修订条目和所述新修订条目的新进程路径。

[0022] 根据当前公开标的的修订控制系统,其中所述修订引擎还可运行以处置针对修订历史日志中的两条或更多条进程路径执行的合并命令;所述合并命令至少包括标识所述修订历史日志内的两条或更多条进程路径的信息和标识合并的修订条目的信息;所述修订管理器对合并命令作出响应以:将所述两条或更多条进程路径聚合成合并的修订条目并且使所述两条或更多条进程路径以单进程路径继续。

[0023] 根据当前公开标的的修订控制系统,其中所述修订历史日志中的每个修订条目由通用唯一标识符(UUID)唯一标识。

[0024] 根据当前公开标的的修订控制系统,其中所述修订历史日志中的每个修订条目由计算为唯一值的两个或更多个元数据元素的组合唯一标识。

[0025] 根据当前公开标的的修订控制系统,其中对修改事件作出响应而在参考区域中创建新修订。根据当前公开标的的修订控制系统,其中对在参考区域上创建的新文件作出响应而触发所述修改事件。

[0026] 根据当前公开标的的修订控制系统,其中通过由下列动作组成的列表中的一个或多个触发修改事件:删除部分或全部修订内容;针对修订改变一个或多个元数据元素;重新命名新修订;将修订移动到新位置;和删除修订。

[0027] 根据本发明的另一方面的修订控制系统,其中所述修订引擎适于和与第二数据储

文库相关的第二修订引擎通信；所述第二数据储存库被定义有第二参考区域；所述修订引擎可运行以从所述第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示，所述第二修订历史日志与所述第二修订引擎相关；所述第一修订条目表示对所述第二参考区域中的修改事件作出响应而生成的修订；所述修订管理器对关于所述第一修订条目的指示作出响应以：标识所述第一修订条目是否从与所述修订引擎相关的修订历史日志中缺失，并且如果缺失，那么创建第二修订条目，所述第二修订条目对应于所述修订；并链接所述第二修订条目到所述修订历史日志内的第三修订条目，从而延伸表示对所述第一参考区域和所述第二参考区域中的修改事件作出响应而生成的修订进程的进程路径。

[0028] 根据本发明的另一方面的修订控制系统，其中所述修订管理器还作出响应以标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目，并且其中所述第二修订条目的链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

[0029] 根据本发明的另一方面的修订控制系统，其中所述修订引擎可运行以从所述第二修订引擎请求由所述第一修订条目表示的修订。

[0030] 根据本发明的另一方面的修订控制系统，其中所述修订从所述第二参考区域中检索。

[0031] 根据本发明的另一方面的修订控制系统，其中所述参考区域在实施于所述第二数据储存库上的文件系统内。

[0032] 根据本发明的另一方面的修订控制系统，其中所述指示包括标识所述第三修订条目的数据。

[0033] 根据本发明的另一方面的修订控制系统，其中如果所述修订管理器未能在所述修订历史日志中标识所述第三修订条目，那么所述修订引擎可运行以：创建请求关于所述第一修订条目和所述第二修订条目的进程路径中的至少一个另外修订条目的缺失信息的询问；至少将所述询问发送到所述第二修订引擎；并且至少从所述第二修订引擎接收指示所述至少一个另外修订条目的新指示。

[0034] 根据本发明的另一方面的修订控制系统，其还包括网络控制单元，所述网络控制单元可运行以实现与所述第二修订引擎的通信。

[0035] 根据本发明的另一方面的修订控制系统，其中所述修订引擎和所述第二修订引擎经由直接连接而连接。根据当前公开标的的修订控制系统，其中所述修订引擎和所述第二修订引擎经由点对点网络而连接。

[0036] 根据本发明的第二个方面，提供一种修订控制系统，其包括：第一修订引擎，其与第一存储区域相关并适于从与第二存储区域相关的第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示，所述第二修订历史日志与所述第二修订引擎相关；所述第一修订条目表示对所述第二存储区域中的修改事件作出响应而生成的第一修订；所述第一修订引擎包括：修订管理器，其适于管理第一修订历史日志以存储关于对第一存储区域中的修改事件作出响应而生成的修订的至少一个修订条目；所述修订管理器对关于所述第一修订条目的指示作出响应以：标识所述第一修订条目是否从与所述第一修订引擎相关的所述第一修订历史日志中缺失，并且如果缺失，那么创建第二修订条目，所述第二修订条目对应于所述第一修订；并且链接所述第二修订条目到所述第一修订历史日志内的第三修订条



目,从而延伸表示所述第一存储区域和所述第二存储区域内的修订进程的进程路径;其中所述修订管理器还作出响应以标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目,并且其中所述第二修订条目的链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

[0037] 根据当前公开标的的修订控制系统,其中所述指示包括标识所述第三修订条目的信息。根据当前公开标的的修订控制系统,其中所述第三修订条目表示所述第一修订的先前修订。

[0038] 根据当前公开标的的修订控制系统,其中所述第一修订引擎可运行以从所述第二修订引擎请求由所述第一修订条目表示的所述第一修订,并且将所述修订存储在与所述第一修订引擎相关的数据储存库中。

[0039] 根据当前公开标的的修订控制系统,其中所述第一修订从在所述第二存储区域中的文件系统中定义的参考区域中检索。

[0040] 根据当前公开标的的修订控制系统,其还包括网络修订控制单元,并且其中如果所述修订管理器未能标识所述第一修订历史日志中的所述第三修订条目,那么所述第一修订引擎运行所述网络修订控制单元以创建请求关于所述第二修订历史日志中的所述第一修订条目的进程路径中的至少一个另外修订条目的缺失信息的询问,并且至少将所述询问发送到所述第二修订引擎。

[0041] 根据当前公开标的的修订控制系统,其中所述指示包括关于所述第二修订历史日志中的多个修订条目的信息。

[0042] 根据当前公开标的的修订控制系统,其中所述第一存储区域定义在实施于与所述第一修订引擎相关的数据储存库上的文件系统内。

[0043] 根据当前公开标的的修订控制系统,其中对由所述修订管理器生成的拉式更新询问作出响应而接收所述指示并且将所述指示传输到所述第二修订引擎。

[0044] 根据当前公开标的的修订控制系统,其中所述指示是由所述第二修订引擎生成的推式更新消息。

[0045] 根据当前公开标的的修订控制系统,其还包括网络控制单元,所述网络控制单元可运行以至少实现与所述第二修订引擎的通信。

[0046] 根据当前公开标的的修订控制系统,其为通道成员并且可运行以连接到在线状态服务以获得所述通道中的一个或多个其它成员的列表。

[0047] 根据当前公开标的的修订控制系统,其中所述第一修订引擎和所述第二修订引擎经由直接连接而连接。根据当前公开标的的修订控制系统,其中所述第一修订引擎和所述第二修订引擎经由点对点网络而连接。

[0048] 根据当前公开标的的修订控制系统,其还包括通道覆盖管理器,所述通道覆盖管理器可运行以管理分布式哈希表架构中的点对点网络的点对点网络覆盖。

[0049] 根据当前公开标的的修订控制系统,其中所述第一修订历史日志中的每个修订条目包括至少一个数据元素并且其中所述第一修订引擎还可运行以:利用所述至少一个数据元素以在所述第一修订历史日志中定位具有一个或多个相同数据元素的两个或更多个修订条目;并且基于所述一个或多个数据元素指示分别对应于所述两个或更多个修订条目的两个或更多个修订之间的相关性。

[0050] 根据当前公开标的的修订控制系统,其中所述至少一个数据元素是从对应修订的至少部分内容中计算出来的内容哈希码并且所述相关性指示所述两个或更多个修订的所述至少部分内容相同。

[0051] 根据当前公开标的的修订控制系统,其中所述至少一个数据元素包括一个或多个元数据元素,所述一个或多个元数据元素的特征在于对应修订并且其中所述相关性指示所述两个或更多个修订的一个或多个元数据元素相同。

[0052] 根据当前公开标的的修订控制系统,其中所述第一修订引擎还可运行以处置针对第一修订执行的恢复命令;所述恢复命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征在于所述修订;所述第二修订条目位于和与所述第一修订相关的第一修订条目相同的进程路径内;所述修订管理器对所述恢复命令作出响应以:用所述第二修订的对应数据元素替换关于所述第一修订的对应于所述一种或多种类型的数据元素的一个或多个数据元素,从而创建新修订;针对所述新修订创建新修订条目;和链接所述新修订条目到所述第一修订历史日志。

[0053] 根据当前公开标的的修订控制系统,其中所述修订管理器被配置来链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而创建包括所述第二修订条目和所述新修订条目的新进程路径。

[0054] 根据当前公开标的的修订控制系统,其中所述一个或多个数据元素是所述第二修订的内容。

[0055] 根据当前公开标的的修订控制系统,其中所述第一修订引擎还可运行以处置针对第一修订执行的复制命令;所述复制命令包括:至少一个标识符,其用于标识对应于第二修订的第二修订条目;和一种或多种类型的数据元素,其特征在于所述修订;所述第二修订条目位于不同于与所述第一修订相关的第一修订条目的进程路径上;所述修订管理器对所述复制命令作出响应以:用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素,从而创建新修订;针对所述新修订创建新修订条目;并且链接所述新修订条目到所述第一修订历史日志。

[0056] 根据当前公开标的的修订控制系统,其中所述修订管理器被配置来链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而创建包括所述第二修订条目和所述新修订条目的新进程路径。

[0057] 根据当前公开标的的修订控制系统,其中所述第一修订引擎还可运行以处置针对第一修订历史日志中的两条或更多条进程路径执行的合并命令;所述合并命令至少包括标识所述第一修订历史日志内的两条或更多条进程路径的信息和标识合并的修订条目的信息;所述修订管理器对合并命令作出响应以:将所述两条或更多条进程路径聚合成合并的修订条目并且使所述两条或更多条进程路径都以单进程路径继续。

[0058] 根据当前公开标的的修订控制系统,其中所述第一修订历史日志中的每个修订条目由通用唯一标识符(UUID)唯一标识。

[0059] 根据当前公开标的的修订控制系统,其中所述第一修订历史日志中的每个修订条目由计算为唯一值的元数据元素的组合唯一标识。

[0060] 根据本发明的第三个方面,提供一种运行包括修订引擎的修订控制系统的方法,所述方法包括:在文件系统内定义至少一个参考区域,所述文件系统实施于数据储存库上;

与所述文件系统通信并标识关于所述文件系统内的文件或文件夹的修改事件；对所述修改事件作出响应；在修订历史日志中创建新修订条目，所述新修订条目对应于所述修改事件；标识所述修订历史日志中对应于实施所述修改事件所相关的文件或文件夹的修订的相关修订条目；和链接所述新修订条目与所述相关修订条目，从而延伸表示所述文件或文件夹的修订进程的进程路径。

[0061] 根据当前公开标的方法，其还包括：标识沿着与所述相关修订条目相关的进程路径存在对应于与所述相关修订条目相关的修订修改的另一修订条目；和延伸包括所述相关修订条目和所述新修订条目的新进程路径。

[0062] 根据当前公开标的方法，其中所述修订历史日中的每个修订条目包括至少一个数据元素，所述方法还包括：利用所述至少一个数据元素以在所述修订历史日志内定位具有一个或多个相同数据元素的两个或更多个修订条目；和基于所述一个或多个数据元素，指示分别对应于所述两个或更多个修订条目的两个或更多个修订之间的相关性。

[0063] 根据当前公开标的方法，其中所述至少一个数据元素是从对应修订的至少部分内容中计算出来的内容哈希码并且所述相关性指示所述两个或更多个修订的所述至少部分内容相同。

[0064] 根据当前公开标的方法，其中所述至少一个数据元素包括一个或多个元数据元素，所述一个或多个元数据元素的特征在于对应修订并且其中所述相关性指示所述两个或更多个修订的一个或多个元数据元素相同。

[0065] 根据当前公开标的方法，其还包括：确定所述两个或更多个修订条目是否位于相同进程路径上。

[0066] 根据当前公开标的方法，其还包括：处置针对第一修订执行的恢复命令；所述恢复命令包括：至少一个标识符，其用于标识对应于第二修订的第二修订条目；和一种或多种类型的数据元素，其特征在于所述修订；所述第二修订条目位于和与所述第一修订相关的第一修订条目相同的进程路径内；所述处置包括：用所述第二修订的对应数据元素替换关于所述第一修订的对应于所述一种或多种类型的数据元素的一个或多个数据元素，从而创建新修订；针对所述新修订创建新修订条目；并且将所述新修订条目存储在所述修订历史日志中。

[0067] 根据当前公开标的方法，其中还包括：链接所述新修订条目到对应于所述第二修订的所述第二修订条目，从而延伸包括所述第二修订条目和所述新修订条目的新进程路径。

[0068] 根据当前公开标的方法，其中所述一个或多个数据元素是所述第二修订的内容。

[0069] 根据当前公开标的方法，其还包括：处置针对第一修订执行的复制命令；所述复制命令包括：至少一个标识符，其用于标识对应于第二修订的第二修订条目；和一种或多种类型的数据元素，其特征在于所述修订；所述第二修订条目位于不同于与所述第一修订相关的第一修订条目的进程路径上；所述处置包括：用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素，从而创建新修订；针对所述新修订创建新修订条目；和链接所述新修订条目到所述修订历史日志。

[0070] 根据当前公开标的方法,其还包括链接所述新修订条目到对应于所述第二修订的所述第二修订条目,从而延伸包括所述第二修订条目和所述新修订条目的新进程路径。

[0071] 根据当前公开标的方法,其中还包括:处置针对所述修订历史日志中的两条或更多条进程路径执行的合并命令;所述合并命令至少包括标识所述修订历史日志内的两条或更多条进程路径的信息和标识合并的修订条目的信息;所述处置包括:将所述两条或更多条进程路径聚合为合并的修订条目并使所述两条或更多条进程路径以单进程路径继续。

[0072] 根据当前公开标的方法,其中所述修订历史日志中的每个修订条目由通用唯一标识符(UUID)唯一标识。

[0073] 根据当前公开标的方法,其中所述修订历史日志中的每个修订条目由计算为唯一值的两个或更多个元数据元素的组合唯一标识。

[0074] 根据当前公开标的方法,其中对所述修改事件作出响应而在所述参考区域中创建新修订。

[0075] 根据当前公开标的方法,其中对在所述参考区域上所创建的新文件作出响应而触发所述修改事件。

[0076] 根据当前公开标的方法,其中通过由下列动作组成的列表中的一个或多个来触发所述修改事件:删除部分或全部修订内容;针对修订改变一个或多个元数据元素;重新命名修订;将修订移动到新的位置;和删除修订。

[0077] 根据当前公开标的方法,其还包括:实现所述修订引擎和与第二数据储存库相关的第二修订引擎之间的通信;所述第二数据储存库被定义有第二参考区域;从所述第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示,所述第二修订历史日志与所述第二修订引擎相关;所述指示包括关于对所述第二参考区域中的修改事件作出响应而生成的修订的数据;对关于所述第一修订条目的指示作出响应:标识所述第一修订条目是否从与所述修订引擎相关的所述修订历史日志中缺失,并且如果缺失,那么创建第二修订条目,所述第二修订条目对应于所述修订;并且链接所述第二修订条目到所述修订历史日志内的第三修订条目,从而延伸表示对所述第一参考区域和所述第二参考区域中的修改事件作出响应而生成的修订进程的进程路径。

[0078] 根据当前公开标的方法,其还包括:标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目,并且其中所述第二修订条目的链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

[0079] 根据当前公开标的方法,其还包括:从所述第二修订引擎请求由所述第二修订条目表示的修订。

[0080] 根据当前公开标的方法,其中所述参考区域在实施于所述第二数据储存库上的文件系统内。

[0081] 根据当前公开标的方法,其中所述指示包括标识所述第三修订条目的数据。

[0082] 根据当前公开标的方法,其中如果在所述修订历史日志中未标识所述第三修订条目,那么所述方法还包括:创建请求关于所述第一修订条目和所述第二修订条目的进程路径中的至少一个另外修订条目的缺失信息的询问;至少将所述询问发送到所述第二修订引擎;和至少从所述第二修订引擎接收指示所述至少一个另外修订条目的新指示。

[0083] 根据当前公开标的方法,其中所述修订引擎与所述第二修订引擎之间的通信是

经由直接连接。

[0084] 根据当前公开标的方法,其中所述修订引擎与所述第二修订引擎之间的通信是经由点对点网络。

[0085] 根据本发明的第四个方面,提供一种运行包括与第一存储区域相关的修订引擎的修订控制系统的方法,所述修订引擎包括修订管理器,所述修订管理器适于管理第一修订历史日志以存储关于对所述第一存储区域中的修改事件作出响应而生成的修订的至少一个修订条目,所述方法包括:从与第二存储区域相关的第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示,所述第二修订历史日志与所述第二修订引擎相关;所述指示包括关于对所述第二存储区域中的修改事件作出响应而生成的第一修订的数据;和对关于所述第一修订条目的指示作出响应:标识所述第一修订条目是否从与所述第一修订引擎相关的所述第一修订历史日志中缺失,并且如果缺失,那么创建第二修订条目,所述第二修订条目对应于所述第一修订;并且链接所述第二修订条目到所述第一修订历史日志内的第三修订条目,从而延伸表示所述第一存储区域和所述第二存储区域内的修订进程的进程路径;并标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目,并且其中所述第二修订条目的链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

[0086] 根据当前公开标的方法,其中所述指示包括标识所述第三修订条目的信息。

[0087] 根据当前公开标的方法,其中所述第三修订条目表示所述第一修订的先前修订。

[0088] 根据当前公开标的方法,其还包括:从所述第二修订引擎请求由所述第二修订条目表示的修订并将所述修订存储在与所述第一修订引擎相关的数据储存库中。

[0089] 根据当前公开标的方法,其中如果在所述第一修订历史日志中未标识所述第三修订条目,那么所述方法还包括:创建请求所述第二修订历史日志中的所述第一修订条目的进程路径中的至少一个另外修订条目的询问,和至少将所述询问发送到所述第二修订引擎。

[0090] 根据当前公开标的方法,其中所述指示包括关于所述第二修订历史日志中的多个修订条目的信息。

[0091] 根据当前公开标的方法,其中对由所述修订管理器生成的拉式更新询问作出响应而接收所述指示并且将所述指示传输到所述第二修订引擎。

[0092] 根据当前公开标的方法,其中所述指示是由所述第二修订引擎生成的推式更新消息。

[0093] 根据当前公开标的方法,其中所述修订引擎与所述第二修订引擎之间的通信是经由直接连接。

[0094] 根据当前公开标的方法,其中所述修订引擎与所述第二修订引擎之间的通信是经由点对点网络。

[0095] 根据当前公开标的方法,其还包括:管理分布式哈希表架构中的点对点网络的点对点网络覆盖。

[0096] 根据当前公开标的方法,其中所述第一修订历史日志中的每个修订条目包括至少一个数据元素并且其中所述方法还包括:在所述至少一个数据元素的帮助下,在所述第

一修订历史日志内定位具有一个或多个相同数据元素的两个或更多个修订；和基于所述一个或多个数据元素指示分别对应于所述两个或更多个修订条目的两个或更多个修订之间的相关性。

[0097] 根据当前公开标的方法，其中所述至少一个数据元素是从对应修订的至少部分内容中计算出来的内容哈希码并且所述相关性指示所述两个或更多个修订的所述至少部分内容相同。

[0098] 根据当前公开标的方法，其中所述至少一个数据元素包括一个或多个元数据元素，所述一个或多个元数据元素的特征在于对应修订并且其中所述相关性指示所述两个或更多个修订的一个或多个元数据元素相同。

[0099] 根据当前公开标的方法，其还包括：处置针对第一修订执行的恢复命令；所述恢复命令包括：至少一个标识符，其用于标识对应于第二修订的第二修订条目；和一种或多种类型的数据元素，其特征不在于所述修订；所述第二修订条目位于和与所述第一修订相关的第一修订条目相同的进程路径内；所述处置包括：用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素，从而创建新修订；针对所述新修订创建新修订条目；并且链接所述新修订条目到所述第一修订历史日志。

[0100] 根据当前公开标的方法，其还包括：链接所述新修订条目到对应于所述第二修订的所述第二修订条目，从而创建包括所述第二修订条目和所述新修订条目的新进程路径。

[0101] 根据当前公开标的方法，其中所述一个或多个数据元素是所述第二修订的内容。

[0102] 根据当前公开标的方法，其还包括：处置针对第一修订执行的复制命令；所述复制命令包括：至少一个标识符，其用于标识对应于第二修订的第二修订条目；和一种或多种类型的数据元素，其特征不在于所述修订；所述第二修订条目位于不同于与所述第一修订相关的第一修订条目的进程路径上；所述处置包括：用所述第二修订的对应数据元素替换所述第一修订中对应于所述一种或多种类型的数据元素的一个或多个数据元素，从而创建新修订；针对所述新修订创建新修订条目；并且链接所述新修订条目到所述第一修订历史日志。

[0103] 根据当前公开标的方法，其还包括：链接所述新修订条目到对应于所述第二修订的所述第二修订条目，从而创建包括所述第二修订条目和所述新修订条目的新进程路径。

[0104] 根据当前公开标的方法，其还包括：处置针对所述第一修订历史日志中的两条或更多条进程路径执行的合并命令；所述合并命令至少包括标识所述第一修订历史日志内的两条或更多条进程路径的信息和标识合并的修订条目的信息；所述处置包括：将所述两条或更多条进程路径聚合成所述合并的修订条目并且使所述两条或更多条进程路径都以单进程路径继续。

[0105] 根据当前公开标的方法，其中所述第一修订历史日志中的每个修订条目由通用唯一标识符 (UUID) 唯一标识。

[0106] 根据当前公开标的方法，其中所述第一修订历史日志中的每个修订条目由计算

为唯一值的元数据元素的组合唯一标识。

[0107] 根据本发明的第五个方面,提供一种可由机器读取的程序存储装置,其有形地具体实施可由所述机器执行以执行运行包括修订引擎的修订控制系统的方法步骤的指令程序,所述方法包括:在文件系统内定义至少一个参考区域,所述文件系统实施于数据储存库上;与所述文件系统通信并标识关于所述文件系统内的文件或文件夹的修改事件;对所述修改事件作出响应;在修订历史日志中创建新修订条目,所述新修订条目对应于所述修改事件;标识所述修订历史日志中对应于实施所述修改事件所相关的所述文件或文件夹的修订的相关修订条目;和链接所述新修订条目和所述相关修订条目,从而延伸表示所述文件或文件夹的修订进程的进程路径。

[0108] 根据本发明的第六个方面,提供一种可由机器读取的程序存储装置,所述程序存储装置有形地具体实施可由所述机器执行以执行运行包括与第一存储区域相关的修订引擎的修订控制系统的方法的方法步骤的指令程序,所述修订引擎包括修订管理器,所述修订管理器适于管理第一修订历史日志以存储关于对所述第一存储区域中的修改事件作出响应而生成的修订的至少一个修订条目,所述方法包括:从与第二存储区域相关的第二修订引擎接收关于第二修订历史日志中的第一修订条目的指示,所述第二修订历史日志与所述第二修订引擎相关;所述指示包括关于对所述第二存储区域中的修改事件作出响应而生成的第一修订的数据;和对关于所述第一修订条目的所述指示作出响应:标识所述第一修订条目是否从与所述第一修订引擎相关的所述第一修订历史日志中缺失,并且如果缺失,那么创建第二修订条目,所述第二修订条目对应于所述第一修订;并且链接所述第二修订条目到所述第一修订历史日志内的第三修订条目,从而延伸表示所述第一和所述第二存储区域内的修订进程的进程路径;并且标识沿着与所述第三修订条目相关的进程路径存在对应于与所述第三修订条目相关的修订修改的另一修订条目,并且其中所述第二修订条目的所述链接延伸包括所述第三修订条目和所述第二修订条目的新进程路径。

[0109] 附图简述

[0110] 为了理解本发明并且了解在实践中如何实行本发明,现在将参考附图仅通过非限制性实施例描述优选实施方案,其中:

[0111] 图 1 是根据本发明的一个实施方案的修订控制系统的方框图;

[0112] 图 2 图示了根据本发明的一个实施方案的在网络 200 背景下的修订控制系统 100 的方框图;

[0113] 图 3 图示了根据本发明的一个实施方案的版本历史日志的实施例;

[0114] 图 4 是示出了根据本发明的一个实施方案的修订历史日志更新的流程图;

[0115] 图 5 是示出了根据本发明的一个实施方案的在 P2P 网络背景下更新周期运行的流程图 500;和

[0116] 图 6A 到图 6N 和图 6P 示意地图示了根据本发明的一个实施方案的出于展示修订控制系统运行的目的而示出的参考区域和修订历史日志。

## 具体实施方式

[0117] 除非另有具体说明,否则如从下列论述显而易见,应明白贯穿利用例如“处理”、“计算 (computing/calculating)”、“确定”、“生成”、“标识”、“设置”、“配置”、“适应”的术语

的说明论述指操纵和 / 或变换数据为其它数据的计算机动作和 / 或程序, 所述数据表示为物理量 ( 举例而言诸如电子量 ) 和表示实物。术语“计算机”应广泛地解释为涵盖具有数据处理能力的任何种类电子装置, 包括 ( 作为非限制性实施例 ) 个人计算机、服务器、计算系统、通信装置、处理器 ( 例如, 数字信号处理器 (DSP)、微控制器、现场可编程门阵列 (FPGA)、专用集成电路 (ASIC) 等 ) 和其它电子计算装置和其组合。

[0118] 根据本文中教导的操作可由特别配置用于所需用途的计算机或通过特别配置用于所需用途的通用计算机通过存储在计算机可读的存储媒体中的计算机程序执行。此外, 不参考任何特定编程语言描述本发明的实施方案。应明白多种编程语言可用于实施如本文所述的本发明的教导。

[0119] 如本文所使用, 描述本发明的示例性实施例的用语“例如”、“诸如”和其变型本质上是示例性并且无限制之意。说明书中提及“一个实施方案”、“一实施方案”、“一些实施方案”、“另一实施方案”、“其它实施方案”、“特定实施方案”或其变型意指结合所述实施方案描述的特定特征、结构或特性包括在本发明的至少一个实施方案中。因此, “一个实施方案”、“一实施方案”、“一些实施方案”、“另一实施方案”、“其它实施方案”、“特定实施方案”或其变型的出现不一定指相同实施方案。

[0120] 应了解为清楚起见于单独实施情况中描述的本发明的特定特征也可以在单个实施方案中组合提供。相反地, 为简洁起见在单个实施情况中描述的本发明的各种特征也可以单独或以任何子组合提供。虽然已经针对特定实施方案示出并描述本发明, 但是本发明不因此而受限制。读者现可想出在本发明范畴内的许多修改、改变和改进。

[0121] 在本发明的实施方案中, 可执行少于、多于和 / 或不同于图 4 和图 5 中所示的阶段。在本发明的实施方案中, 图 4 和图 5 中所示的一个或多个阶段可以不同顺序执行和 / 或一组或多组阶段可同时执行。图 1 和图 2 图示了根据本发明的一个实施方案的系统架构的总体示意图。图 1 和图 2 中的每个模块可由执行如本文定义和说明的功能的软件、硬件和 / 或固件的任何组合组成。图 1 和图 2 中的模块可集中于一个位置或分散在多个位置上。在本发明的其它实施方案中, 所述系统可以包括少于、多于和 / 或不同于图 1 和图 2 中所示的模块的模块。

[0122] 记住上文之后, 以下是本公开标的的不同方面的详细描述。存储在计算机文件系统上的计算机文件 ( 诸如文档 ) 的内容和元数据 ( 皆称为数据元素, 其特征在于文件 ) 两者皆时常以各种方式修改。计算机文件 ( 或文件 ) 是存储在文件系统上的二进制数据流。在本发明背景下, 术语“文件系统”指用于存储、组织、操纵和检索数据的专用数据储存库, 或更具体来说, 指计算机文件。逻辑文件系统通常定义在一个或多个存储装置上, 并且被组织为文件夹, 所述文件夹继而可包含另外子文件夹和文件。一些类型的文件系统类型包括例如 NTFS、ufs、HPFS、JFS、FAT (16/32)、ext2、ext3 等等。

[0123] 当用户或软件程序或以另一方式修改这类文件时, 通常在修改事件中捕获这个修改。在本发明背景中, 修改事件描述目的在于修改文件 ( 或文件夹 ) 的内容或元数据的事件。这类事件的实施例包括但不限于以下事件: 创建新文件夹; 从文件系统中删除整个文件; 重写文件中先前存在的数据; 从文件中删除数据; 重新命名文件; 将文件移动到不同位置 ( 例如, 相同文件系统中或另一文件系统中的另一文件夹 ); 修改文件的现有元数据; 及添加元数据到文件。举例来说, 通过下列项目中的一项来管理修改: 存储文件的文件系统、



文件系统驱动程序、负责管理文件的操作系统、文件服务器或网络附接文件系统、或管理文件的硬件装置。

[0124] 一般来说,修改事件描述文件系统上新文件修订的创建。所述新修订通常由新修改的文件表示并且存储在文件系统中,而旧的文件修订通常被丢弃。此类文件系统不将新的文件修改视为文件“修订”,不记录当前修订中关于其先前状态和其先前包括的数据的改变,并且无法使用户与文件的先前修订互动。为了记录修改事件的历史和所得文件修订,本发明利用修订控制。

[0125] 根据本发明的特定实施方案,提供一种文件修订控制系统、方法和设备,其被配置用于存储和管理历史修订日志,并用于实现追踪和管理、比较和检索修订,所述修订包括存储在指定参考区域上的当前修订、存储在预定义存储区域中的过去文件修订(或文件修订)并且还包含先前在参考区域上的过去文件修订。

[0126] 在本发明背景下,术语“参考区域”指至少一个文件系统内的预定义存储空间,所述存储空间可指定用于存储当前修订并且如下文更详细说明可供文件修订控制访问。在本发明背景下,术语“当前修订”指关于最早文件(或最早修订)的最新更新修订,其目前对于给定进程路径而言可用。当前修订当前可存储在经由所述参考区域监控的文件系统中。

[0127] 术语“进程路径”指表示文件随时间的进程(或演变)的一个或多个修订条目,其中每个修订通过修改至少一个过去修订的内容和/或元数据和/或状态而创建。如下文更详细说明,如果修订分支成多个修订,那么每个这种修订将构成一条不同进程路径。

[0128] 术语“过去修订”指除了当前修订以外的修订。过去修订可存储在过去修订的指定数据储存库中。术语“先前修订”(作为相对术语)指与较新修订直接链接的修订。

[0129] 根据特定实施方案,本发明通过计算一组另外数据并使所述组另外数据与实际文件相关而操作。这另外数据尤其提供关于针对文件系统上的文件发生的修改事件的信息,并且在文件进展通过其变化的修订表现时,实现文件进程的存储、追踪、检索和/或显示。

[0130] 根据本发明的特定实施方案,文件的每个修订由修订历史日志中的修订条目(例如,表示为顶点、图节点或记录)表示,所述修订条目在指定数据储存库中存储和管理。所述修订历史日志中的修订条目存储对应于至少一个修订的数据,而修订自身可存储在被监控的文件系统(例如,其经由所述参考区域监控)上和/或过去修订储存库中。

[0131] 每个修订条目是表示文件修改的数据实体(例如,物件、或相关数据库中的记录),并且与修订相关。每个修订条目包含标识并且特征化特定修订的属性(或数据元素)的汇编。例如,修订条目可以包括下列属性列表中的部分或全部:

[0132] 修订标识符;对应于文件版本内容的哈希码;父修订条目的标识符;一个或多个时间戳(例如,当前修订的创建时间);对应于部分或全部修订元数据的哈希码;父元数据的哈希码;对应于这个文件的最早已知修订的哈希码;这个文件的最早已知修订的识别符;文件系统内文件的名称;文件系统内至文件的相对路径、文件系统内至文件的绝对路径;相关文件数据储存库内至当前修订的位置的链接;至父修订条目的链接;至修订条目的后代的链接列表;创建修订的唯一用户标识符。

[0133] 文件系统常见方式是由呈指引到文件系统中的文件的唯一位置的路径形式的文件名标识文件。根据这种方式,文件由其名称和文件夹以及子文件夹的路径的组合唯一标识,从而指向包含所述文件的子文件夹,一起称为文件的“完整路径”。从内部来看,在

文件系统和 / 或文件系统驱动程序, 和 / 或促成它的操作系统内, 文件可由其它标识符标识。这些另外标识符可以指向文件系统和 / 或存储装置上的文件的开始, 指向文件的 ID 号, 指向其第一个 i 节点等等。当出于任何目的访问文件时, 这些另外标识符通常被文件系统、运行在计算机上的软件应用程序和作用在计算机上的操作系统使用。

[0134] 尽管使用文件名和路径标识文件可能是目前最常见的方式, 但是这种方式存在问题并且在就文件历史和其所含数据 (“内容”) 考虑文件时不适用。例如, 两个文件可能包含完全相同的数据, 并且仍被视为给定文件系统上的两个单独文件, 并且所述文件系统可能无法标识这两个文件具有相同内容并且无法使这两个文件相关。此外, 一些文件系统可能无法确定两个文件间是否具有任何历史关系。例如, 文件系统可能无法标识一个特定文件是否表示一个较早文件的修改修订。在另一实施例中, 在一个文件的内容被复制到第二个文件中后, 文件系统可能无法指示第一个文件是第二个文件的来源。

[0135] 因此, 根据特定实施方案, 除了文件标识符 (诸如文件的完整路径) 之外或替代文件标识符 (诸如文件的完整路径) 以外, 本发明的系统、方法和设备针对多种任务和目的还可利用所计算出来的哈希码。

[0136] 每个修订条目可与唯一标识符相关, 所述唯一标识符至少唯一标识修订历史日志域内的修订。这个标识符例如使修订条目能与数据储存库中的对应修订相关。

[0137] 此外, 修订条目可包括哈希码, 所述哈希码使修订能根据修订的内容和 / 或元数据与对应的修订条目相关。所计算出来的哈希码可用于基于相关修订的内容和 / 或元数据定位相关修订条目, 其继而与修订的唯一标识符相关并且 (例如, 通过使文件系统中的文件名与修订条目内的唯一标识符相关, 可选地与哈希码一起) 实现对应修订的检索。

[0138] 哈希码例如可通过针对整个文件内容或其部分执行的计算而生成。这些哈希码在本文中称为内容哈希。从二进制流中生成哈希码的常见运算法则包括例如 MD5 和 SHA1。哈希码可针对给定文件而计算出来并且可用于通过文件的实际内容而非文件名来标识给定文件。因此, 例如, 当所计算出来的哈希码用于标识文件的当前修订、或过去修订时, 所述哈希码可用于在具有相同内容的修订历史日志内定位其它修订。

[0139] 在内容哈希的实施例中, 引入文件系统的、具有陌生文件名的新修订可首先由其内容标识, 并且随后通过所计算出来的内容哈希与记录在修订历史日志中的其它修订相关。一旦标识具有相同内容的修订的修订条目, 特定操作立即可用匹配的修订条目执行。例如, 这些修订的文件名可从对应的修订条目中检索, 或基于使修订条目与其修订自身相关的唯一标识符从文件系统中检索。

[0140] 如上所提及, 修改事件可包括对文件内容的修改和 / 或对关于文件的元数据的修改。在本发明背景下, 术语“元数据”指关于非存储在文件内的实际数据内容的文件的任何信息。文件元数据可包括但不限于下列数据元素: 文件名、不同时间戳 (例如, 最后修改时间或创建时间)、文件大小、文件内容哈希码、文件所有者、最后修改文件的用户的 ID 等等。每个修订条目可存储内容哈希和元数据哈希两者, 并且因此表示关于文件的修订数据。在一些实施方案中, 单独修订条目可构建用于单独存储对文件内容作出的修改和对文件元数据作出的修改。可选地, 指示特定修改事件发生的时间的时间戳用文件的每个修订更新。因此, 取决于具体实施方案, 文件可基于其内容和 / 或其元数据区分和追踪。

[0141] 类似于文件内容, 也可针对包含给定文件的元数据的二进制流计算哈希码。这种

哈希码在本文中称为“元数据哈希码”。元数据哈希码还可用于标识文件的当前修订或过去修订。类似于内容哈希码,元数据哈希码可用于标识特定修订条目(作为密钥),并且可用于访问这些条目以及使修订条目和对应的修订之间相关。

[0142] 在一些实施方案中,元数据哈希码可至少在给定参考区域内用于唯一标识文件的特定修订(例如,出于如下文所述使修订与其对应修订条目相关的目的)。这可以通过从一组具体选定的元数据元素中计算出来的元数据哈希码完成。例如,通过包括提供足够唯一的哈希码的元数据元素(例如,当地日期和时间、随机数量的足够的比特长度等等)的组合,可唯一标识特定修订。例如,在 Microsoft Windows-NT 系统背景下,文件可由从包含元数据的二进制流中计算出来的元数据哈希码唯一标识,所述元数据包括用户 SID(安全标识符)、被追踪文件的相对路径、被追踪文件的文件名、在计算哈希码时生成的 UUID、指定当前时钟的时间戳、处理器的时间戳计数器等等。

[0143] 也可针对文件的一部分和与所述文件相关的元数据的一部分计算哈希码。当针对文件的一部分计算哈希码时,所述哈希码在本文中称为“部分哈希码”。多个部分哈希码可用于指文件的不同部分。例如,如果针对 1 兆字节文件的文件的每 1 千字节区块计算部分哈希码,那么这将产生一千(1000)个这种哈希码。每个哈希码关于文件的一个特定部分。部分哈希码可用于确定文件内的具体区块被修改,而其它区块相对于先前修订或过去修订大体上保持不变。部分哈希码的用途可在若干操作和情况下利用,包括针对给定文件的修改统计的收集;确定在给定文件内是否已发生一个或多个修改;过去先前文件修订中的存储空间的最小化;当在节点之间传达涉及相同文件或更多文件的文件修改时通信量的最小化。

[0144] 修订历史日志可实施为适用于存储并追踪关于文件的一个或多个修订的信息的任何类型的数据结构。例如,修订历史日志可通过将多个顶点(每个顶点表示文件的修订)与顶点之间表示任何单个文件的不同修订之间的关系或进程的边缘(通常也称为链接)链接在一起而构建。例如,修订历史日志可表示为一组树、树、RDBMS 中的一组关系表、NoSQL 数据库中的数据或有向图。

[0145] 进程路径在修订历史日志中由通过边缘在一个方向上链接的修订条目表示。在进程路径中,每个修订条目可最多具有一个先前修订条目并且充当最多一个其它修订条目的先前修订条目。如此,当修订条目(“交汇点”修订条目)分支成多个修订条目时,创建多条进程路径。

[0146] 根据一些实施方案,修订历史日志可实施为针对参考区域内的每个文件创建表示单个文件的修订历史的唯一数据结构。根据其它实施方案,修订历史日志可实施为表示多个文件的修订历史的单个数据结构。根据特定实施方案,修订历史日志实施为表示一个参考区域中的所有文件的进程路径的唯一数据结构(或数据实体),其中存储在参考区域上的每个文件(即,最新修订)由一个“当前修订条目”表示。

[0147] 相同文件的多个副本(例如,每个文件具有不同名称)可与相同内容哈希码相关,并且也可与相同元数据哈希码(例如,哈希是在不考虑差异元素(诸如不同名称)的情况下计算出来的)相关。相同文件的多个副本也可与类似的修订条目相关或可选地甚至与相同修订条目相关。

[0148] 根据本发明的特定实施方案,如果参考区域上的两个或更多个当前修订包括完全

相同的内容,那么针对每个当前修订创建一个或多个修订条目,每个修订条目包括至少一个相同哈希码。所述修订条目添加到相关修订历史日志中的相关进程路径中的每条路径。如此,每个当前修订由不同进程路径表示,并且当当前修订中的一个被进一步修改时,其进程路径可正确保留和追踪。

[0149] 根据其它实施方案,具有相同内容的不同当前修订可与单个修订条目相关,而非针对每个修订创建修订条目。在这种情况下,可存储其它信息以标识哪些文件和修订条目相关,并且能够记录哪些相关的当前修订随后被修改以及以何种方式被修改。

[0150] 图 1 图示了根据本发明的一个实施方案的修订控制系统 100 的方框图。图 1 中的修订控制系统可实施在单个装置或节点上,其中发生对一个或多个文件系统中可由文件修订系统 110 访问的文件的修改。如本文中所使用的术语“节点”可指实施本发明的系统和方法的逻辑的整个硬件装置。或者或另外,节点可指包括耦接到总线和计算机内存的处理器硬件的硬件装置,其中如本文中所述,所述处理器适于实施(或模拟)本发明的修订控制系统和方法的逻辑。节点可为但不限于个人或便携式计算机、PDA、手机或具有适当处理能力和内存并且能够运行并访问适当文件系统的任何其它设备。根据特定实施方案,修订控制系统 100 包括至少一个处理器,或与至少一个处理器相关,所述处理器被配置来管理、执行并控制其相关组件和运行。

[0151] 根据特定实施方案,被配置为修订控制系统 100 的单个节点可适于服务一个或多个用户,其中每个用户与特定用户配置文件相关,所述特定用户配置文件指定所述特定用户的偏好和系统配置。因此,实施系统 100 的每个节点可被多个用户使用,其中每个用户与特定用户配置文件相关,所述特定用户配置文件定义所述用户的偏好和配置并且涉及关于所述用户的特定数据例如:特定参考区域、系统配置(策略)、修订历史日志、通道等等。

[0152] 根据特定实施方案,修订控制系统 100 包括文件修订引擎 110。一般来说,所述文件修订引擎 110 被配置来维持关于一个或多个参考区域的修订历史日志并且对针对相关参考区域内的文件发生的修改事件作出响应而根据预定义策略更新每个修订历史日志。如果每个节点被配置来服务多个用户,那么每个用户配置文件可与对应于所述特定用户的相关参考区域的不同修订历史日志相关。参考区域指文件修订引擎 110 可访问的文件系统中的至少一个特定存储空间。针对参考区域中的文件和/或文件夹发生的文件修改事件由修订引擎 110 监控。

[0153] 文件(例如,当前修订)和文件系统可存储在包括在修订控制系统 100 内(或与修订控制系统 100 相关)的文件数据储存库 125 中。修订控制系统 100 也可与被配置来存储一个或多个修订历史日志的修订历史日志数据储存库 130 相关。在一些实施方案中,修订控制系统 100 也可与过去修订数据储存库 135 相关,所述过去修订数据储存库 135 是指定用于存储正被文件修订引擎 110 监控或过去被文件修订系统引擎 110 监控的文件的过去修订的数据储存库。数据储存库 125、130 和 135 的部分或全部可与修订控制系统 100 直接相关,例如其可与修订控制系统 100 嵌在相同装置上,或直接连接到修订控制系统 100 的装置。或者或此外,数据储存库 125、130 和 135 可位于可由修订控制系统 100 经由任何类型的通信设施或网络(例如,有线或无线网络)访问的远程装置上。在一些实施方案中,数据储存库 125、130 和 135 内的数据可组织为可由修订控制系统 100 经由任何类型的通信设施或网络访问的关系数据库。数据储存库 125、130 和 135 可被配置为可由修订控制系统 100

经由任何类型的通信构件或网络访问的文档管理系统。数据储存库 125、130 和 135 可被配置为许多分布式数据储存库单元或单个数据储存库单元。数据储存库 125、130 和 135 内的数据可组织为数据库（例如关系数据库）、文件系统中的 NoSQL 数据库（例如文件系统上的文件）等等。

[0154] 根据特定实施方案，文件修订引擎 110 包括修订管理器 115、文件系统监控器 120 和修订策略单元 140。根据特定实施方案，修订管理器 115 被配置来创建并管理存储在参考区域内并正被系统 100 监控的文件（和文件夹）的修订历史日志。

[0155] 根据特定实施方案，文件系统监控器 120 连接到一个或多个文件系统（例如，存储在数据储存库 125 上）并且被配置来监控文件系统中关于正被系统 100 监控的文件（例如，位于参考区域内）执行的修改事件。当标识文件修改事件时，文件系统监控器 120 直接或者通过中间机构（举例来说诸如操作系统）警告修订管理器 115。或者，修订管理器 115 可从文件系统监控器 120 或从如上文指定的中间机构检索记录的事件。根据一些实施方案，警告（或事件）包括文件的全名和路径，或或者可转译成完整路径和文件名的某些类型的文件标识符。在一些实施方案中，文件系统和 / 或操作系统也允许文件系统监控器 120 标识所发生的修改事件类型（例如，修改、创建、删除或重新命名事件）。

[0156] 应注意，在一些实施方案中，系统 100 可不包括文件系统监控器 120。在这些实施方案中，系统 100 可通过其它方式（例如，每当修改事件执行时，用户可手动通知系统 100），或可能通过例如由修订管理器 115 在文件系统中执行的针对标识修改事件的扫描操作而更新修改事件。

[0157] 文件系统监控器 120 可执行扫描操作，在扫描操作期间，文件系统监控器 120（例如，对文件修订引擎 110 发出的命令作出响应）扫描一个或多个参考区域。在扫描操作中，参考区域例如从顶级文件夹横穿到其所包含的子文件夹，搜索满足特定预定标准的修改。针对所扫描文件和文件夹的元数据字段和 / 或实际内容定义标准。这类标准可例如指定修改时间戳，所述修改时间戳告知监控器 120 搜索在指定修改时间和日期后修改的文件（即，在 2009 年 11 月 1 日后修改的文件）。如果扫描操作定位到满足预定义标准的文件，那么文件系统监控器 120 可如上所述警告修订管理器 115。或者，文件系统监控器 120 可指示所定位到的文件例如为具有待发警告的文件，而非实际警告修订管理器 115。

[0158] 修订管理器 115 接收指示关于存储在参考区域上的文件的文件修改事件已发生的信息。这种信息可例如以由文件系统监控器 120 发出或通过由文件系统监控器 120 上读取待发警告而检索的通知出现。修订管理器 115 作出响应而更新包含相关文件的修订条目的修订历史日志。修改事件通常导致在文件系统中创建文件的新修订。因此，例如，如果文件内容已改变并且在参考区域上可获取文件的新修订，修订管理器 115 针对新文件修订创建新修订条目（其包含关于修改的相关信息），并且将修订条目链接在修订历史日志中的适当位置（例如，如下文将详细说明，根据这个文件先前已知的修订条目、根据其修改时间戳等等）。

[0159] 根据特定实施方案，修订策略单元 140 连接到修订管理器 115 并且包括修订管理器 115 的操作规则和配置。所述修订管理器操作规则和配置包括例如，参考区域配置、修订历史日志配置和修订历史日志更新策略。

[0160] 根据特定实施方案，修订策略单元 140 被配置成能够在可由修订控制系统 100 访

问的装置的数据储存库内定义参考区域。在一些实施方案中,所述参考区域可例如通过在文件系统内选择特定文件夹或文件而定义。在其它实施方案中,例如,所述参考区域可定义为文件数据储存库 125 的整个存储空间。

[0161] 根据特定实施方案,可针对给定节点定义多个参考区域。如果每个节点被配置来服务多个用户,那么每个用户配置文件可被配置为具有不同参考区域。不同用户可具有带有重叠存储空间的参考区域。如果单个节点上的两个或更多个参考区域被定义有重叠存储空间(例如,不同参考区域与待监控的相同文件夹相关),那么修订管理器 115 可针对所述重叠存储空间采取仅维持一个修订历史日志。在这种情况下,修订管理器 115 也可通过将多个重叠参考区域、文件夹和文件的操作合并为较少操作而使其操作最优化以节省诸如计算时间、带宽和存储空间之类的资源。参考区域配置也可包括例如:参考区域内应被系统 100 监控的文件夹和文件列表(“包括列表”)、不应被系统 100 监控的文件和文件夹的列表(“排除列表”)、是否循环监控包含在参考区域中的子文件夹等等。

[0162] 修订历史日志配置指修订历史日志的不同属性和设置。根据特定实施方案,修订策略单元 140 被配置成能够定义、检索并存储历史日志配置。历史日志配置可包括例如:分配用于存储正被系统 100 监控的文件的过去修订的最大存储空间;可针对正被系统 100 监控的每个文件存储的过去修订的最大数量;在每个修订条目中定义的强制和/或可选属性;在过去修订储存库 135 中维持过去修订的时间间隔(即,3 个月、1 个星期、2 年)等等。修订历史日志配置也可包括例如:修订历史日志的存储位置、修订历史日志成长限制,诸如用于存储的最大修订数量或分配用于日志成长的最大存储空间等等。

[0163] 修订历史日志更新策略指文件修订管理器 115 的更新规则。根据特定实施方案,修订策略单元 140 被配置成能够定义并存储修订管理器 115 更新策略。修订管理器 115 更新策略可包括诸多参数,举例来说诸如:起始修订历史日志更新的事件和操作类型;更新是由系统 100 自动触发还是由用户的明确请求触发。在自动排定更新的情况下,所述更新策略可包括修订历史日志更新之间的最小和最大时间间隔;更新是否应紧接在每个修改事件后发生,或更新是否在预定义数量的修改事件后发生,和/或其它策略参数。例如,修订历史日志更新策略还可包括诸多参数,诸如在修改事件经过处置来创建新修订条目(例如以允许将频繁的修改事件处理为单个修订)之前等待的最小时间、在修订引擎 110 开始扫描修改后的时间间隔等等。

[0164] 根据特定实施方案,系统 100 还包括用户界面 150,所述用户界面 150 被配置成使用户能与系统 100 互动。用户界面 150 可包括特别配置用于访问修订控制系统 100 的用户界面硬件和/或软件组件(例如,鼠标和键盘)。或者或此外,用户界面 150 可为可用于访问修订控制系统 100 的另一计算机或装置的用户界面。根据特定实施方案,系统 100 还包括显示器 160,举例来说诸如用于通过系统 100 显示数据的 LCD 屏幕。显示器 160 可用于例如展示给定文件的修订历史日志的视觉表示、显示在系统 100 上执行的询问的结果、展示错误消息、展示在系统 100 上执行的不同操作的结果等等。

[0165] 根据特定实施方案,在修订策略单元 140 中编码的数据可由系统 100 的用户配置,其中修订策略单元 140 可经由用户界面 150 加以访问和配置。根据特定实施方案,如下文进一步说明,在修订策略单元 140 中编码的数据可通过利用互动元件(诸如用户界面 150 和显示器 160)而配置。根据其它实施方案,在修订策略单元 140 中编码的一些或全部数据

可预先实施和 / 或硬编码和 / 或编程到系统 100 中。

[0166] 根据特定实施方案,被配置来与修订引擎 110 互动的互动元件诸如显示器 160、用户界面 150 和可能的其它硬件和 / 或软件元件可用于与修订引擎 110 互动。修订引擎 110 可被配置成能够在一个或多个互动元件的帮助下发出询问。询问可用于请求关于修订历史日志的信息。这类询问可包括例如修订条目的检索、进程路径的检索、部分进程路径的检索、根据特定标准的修订条目检索(例如,返回包含相同内容哈希码的所有修订条目的询问)、完整修订历史日志的检索。修订引擎 110 可被配置成能够在一个或多个互动元件的帮助下发出命令。显示器 160 可用于显示例如呈表格、图形绘制的进程路径和有向图等等的形式的询问结果。

[0167] 图 2 是根据本发明的一个实施方案在构成网络修订控制系统 205 的网络 200 背景下的修订控制系统 100 的方框图。修订控制系统 205 可实施在运行为文件共享环境中的节点的装置上。

[0168] 文件共享环境的一个实施例是由活动目录和 / 或域控制的微软视窗网络(Microsoft Windows Network(SMB))控制,其中网络文件服务器上的文件可被客户端计算机访问。另外的文件共享环境包括例如以 NETBIOS 为基础的网络、以 NFS 为基础的网络和以 AppleTalk 为基础的网络等等。在这样一个文件共享环境背景下,文件修订引擎 110 可在一个节点上(例如,在文件服务器节点上)被配置来追踪其局部相关的存储空间、或可经由网络(例如,经由 UNC 共享名称)访问的存储空间上的参考区域。被修订引擎 110 追踪和记录的文件修改事件可以包括关于对服务器共享文件系统上的文件所作的修改的信息,并且修订历史日志可(由修订引擎 110)生成和管理来维持表示关于这些事件的信息的对应修订条目。

[0169] 在一些实施方案中,修订引擎 110 可分布在多个节点上。例如,文件系统监控器 120 处于如上所述配置的文件服务器上,而修订引擎 110 的其它组件(例如,修订管理器 115 和修订策略单元 140)位于可运行以与文件服务器通信的不同节点上。文件系统监控器 120 因此可记录由不同用户起始的文件修改事件,并且将这个信息发送到位于不同节点上的修订管理器 115,以实现修订历史日志的管理和更新。

[0170] 在一些实施方案中,容纳修订管理器 115 的节点对文件服务器的访问仅限于针对修订历史日志的管理和更新的修改事件的通信。在其它实施方案中,所述通信可较不受限制并且还包括针对其它目的的通信。

[0171] 文件共享环境的另一实施例是点对点(P2P)网络,其中充当同级的节点通常由点对点覆盖管理并且尝试保留用由相同覆盖的其它同级管理的变化的全部或部分节点文件、修订历史日志以及可能的当前和过去修订。一般来说,同级(P2P)网络是将节点(例如,计算机)链接成通信系统的分布式网络架构的模型,其中每个节点通常被配置成具有类似能力,并且适于运行客户端和服务端两者。根据特定实施方案,本发明的系统和方法能够创建并管理 P2P 网络,P2P 网络能够根据局部节点和节点经由 P2P 网络所连接的其它同级上的修订更新修订历史日志,并且使修订历史日志能允许这样一种网络的同级共享文件并且记录对其它同级中的文件所作的任何改变或更新。

[0172] 图 2 描绘网络 200(构成网络修订控制系统 205)背景下的修订控制系统 100 的实施例,其中修订控制系统 100 包括网络修订控制单元 210,所述网络修订控制单元 210 连接

到文件修订引擎 110 并且一般来说被配置成能够与 P2P 网络通信并在 P2P 网络中在同级之间交换信息。修订控制单元 210 还被配置成能够在关于可用更新的 P2P 网络中在同级之间交换信息。当以这种方式配置时,关于在一个节点上由文件修订引擎 110 追踪的修改事件的信息可经由点对点消息发送到连接到相同覆盖的其它节点。这个配置促进维持由相同网络中的多个同级共享的修订历史日志和文件和文件夹的最新副本。在图 2 所示的实施例中,修订控制系统 100 连接到 3 个其它同级 206、207、208 和在线状态服务 250。

[0173] 根据特定实施方案,网络修订控制单元 210 包括通道管理器 230、通道配置单元 235 和 P2P 引擎 240。通道管理器 230 连接到修订引擎 110 并且被配置用于在 P2P 网络内执行通道背景下的运行。

[0174] 根据特定实施方案,应注意网络修订控制单元 210 可被配置成能够在由例如在如上所述的视窗域文件共享环境背景下配置的局域网 (LAN) 连接的同级之间通信并交换信息。

[0175] 在本发明背景下,“通道”定义一个节点上的一组一个或多个特定参考区域(例如,(诸)文件夹),所述参考区域可由订阅至 P2P 网络中的各自通道的一组同级访问用于共享并更新。对于每个同级,与通道相关的文件和文件夹由同级的对应节点内的文件修订引擎 110 监控,并且针对存储在通道相关的参考区域内的所有相关文件维持修订历史日志。根据本发明的特定实施方案,预定至相同通道的同级被指定为通道的成员并且在本文中称为“成员”。

[0176] 如下文更详细描述,由通道管理器 230 执行的操作可包括例如从通道的其它成员请求更新并且向通道的其它成员告知新的可用更新。在一些实施方案中,通道可由 P2P 网络上希望其文件的部分或全部可被相同网络的其它同级访问的任何同级创建或初始化。

[0177] 点对点覆盖可被描述为网络节点之间的连接的应用层网格,网络节点构成所得网格中的同级。P2P 网络中的覆盖架构通常由同级自身管理,最终消除对通过中央覆盖机构诸如通信服务器协调每个通信的需要。覆盖可根据 P2P 网络内变化的需求和条件动态地重新构建并更新,通过非限制性举例来说,包括当同级加入覆盖并离开覆盖时,和/或当特定同级上的工作量改变时(其出于最优化目的可被传输到其它同级),改变覆盖。覆盖管理运算法则在本领域技术中众所周知并且因此不再进一步论述。已知的覆盖管理运算法则的实施例包括 PASTRY、TAPESTRY、CHORD、KADEMLIA 和 GNUTELLA 等等。其它信息参见 Buford, Yu and Lua, 2009, pp. 12, 139-140, 146-148, 140-146f (“P2P Networking and Applications”, Morgan Kaufmann)。P2P 覆盖管理可例如通过 P2P 覆盖管理器 244 执行。

[0178] 一般来说,在 P2P 网络中,直接或间接(例如,通过其与其它中间同级的连接)连接的节点被视为相同网络或覆盖的同级。通道覆盖是定义相同通道的成员之间的连接的覆盖(连接网格)。在通道覆盖中的成员之间传达的消息仅在预定至相同通道的成员之间路由。

[0179] 根据特定实施方案,通道管理器 230 被配置来从通道配置单元 235 接收关于与修订控制系统 205 相关的通道的信息和关于每个通道的成员的信息。通道管理器 230 还被配置来基于从通道配置单元 235 接收的信息确定建立与通道的其它成员的连接和通信的通道覆盖要求。根据特定实施方案,通道管理器 230 连接到 P2P 引擎 240, P2P 引擎 240 被配置来建立并管理节点与 P2P 网络的网络覆盖。P2P 引擎 240 可被配置来从通道管理器 230



接收关于相关通道成员和所请求操作（例如，如下文所述，针对更新询问的请求）的信息并建立与通道的其它成员的通信和 / 或互动。如上文提及的现有技术参考中所述，针对覆盖管理，P2P 引擎 240 还可被配置来通过应用相关覆盖维持技术和处置覆盖维持通信消息而辅助其它同级维持覆盖。

[0180] 通道配置单元 235 还可包括可用于定义通道的属性的下列实施例的部分或全部：全局唯一通道标识符（关于网络上的所有其它通道）；通道的所有者（例如，创建通道的用户）；通道可访问的磁盘存储空间上的参考区域；特定更新策略，如下文更详细说明，其是定义用于执行相同通道的不同同级之间的文件更新的参数的一组规则；用户列表等等。

[0181] 根据特定实施方案，通道配置单元 235 适于设置和存储通道属性和配置。

[0182] 根据特定实施方案，P2P 网络的同级可通过首先发送请求授予通道访问权的预定请求到通道所有者（或管理员）而变成通道的成员。接着所述请求同级应被通道所有者批准。或者或此外，同级可被通道所有者（或管理员）邀请变成通道的成员，并且接着所述同级应通常批准邀请。一般来说，节点的任何用户可变成多个通道的成员，作为通道的所有者（例如，通过起始通道）或作为现有通道的用户。每个通道可被定义有不同的参考区域，允许不同通道中的每个通道的成员共享并更新特定组的文件和文件夹。一旦同级已连接到至少一个通道覆盖，其立即可与当前连接到所述覆盖的通道的其它成员共享并更新在与通道相关的预定义参考区域内的文件、文件夹以及所述文件和文件夹的修订历史日志。

[0183] 根据特定实施方案，网络修订控制单元 210 连接到在线状态服务 250，在线状态服务 250 给网络同级提供各种服务。例如，这类服务可包括网络上的同级和用户的注册和追踪、同级和用户的认证、当前在线同级和用户的追踪、通道注册等等。在线状态服务 250 可实施在一个或多个服务器计算机上。此外或或者，在线状态服务 250 可在 P2P 网络内的一个或多个节点上（例如，在分布式哈希表 -DHT 中）实施于单个硬件装置或多个硬件装置和 / 或一个或多个虚拟计算机上。在线状态服务 250 的不同组件可实施在一个或多个服务器计算机上，而其它组件可由 P2P 网络内的一个或多个节点例如由分布式哈希表实施。在线状态服务 250 可经由任何类型的通信构件或网络（例如，经由有线或无线网络）连接到网络修订控制单元 210。

[0184] 根据特定实施方案，在线状态服务 250 促进同级身份管理。允许同级登录到在线状态服务 250，例如通过采用用户名和密码或其它标识数据。这类同级的身份可由可辅助将同级之间的通信仅限于所认证的同级的在线状态服务 250 认证。在线状态服务 250 也可验证、提供并交换同级之间的加密密钥，以促进在线状态服务 250 和同级之间的加密数据传达并促进同级自身之间的加密数据传达。例如，试图作为成员连接到特定通道的同级的身份可在允许同级与通道的其它成员通信之前认证。

[0185] 根据特定实施方案，在线状态服务 250 包括通信组件 255，通信组件 255 被配置用于管理系统 205、在线状态服务 250 和 P2P 网络中的其它同级之间的通信。根据特定实施方案，在线状态服务 250 还可包括下列数据储存库：用户数据库 260、在线数据库 265 和注册数据库 270。用户数据库 260 包含对应于注册至 P2P 网络或提供 P2P 网络访问权的用户和 / 或同级的条目。用户数据库 260 可包含注册用户的认证信息和安全信息以提供登录工具。例如，试图认证请求加入通道的另一同级的一个同级（例如，通道所有者或管理器）可联系在线状态服务 250 以询问用户数据库 260 以认证试图加入通道的同级的身份。

[0186] 根据特定实施方案,在线数据库 265 包含当前连接到 P2P 网络(即,在线)的同级列表。这个列表在同级加入并离开网络后更新。每个这类条目可包含同级的地址信息(例如,互联网协议(IP)地址)。所述条目也可包括所连接同级的联系信息,可能需要所述联系信息来与同级通信。例如,在线数据库 265 可在 TCP/IP 通信的情况下保存 IP 地址和端口地址。

[0187] 根据特定实施方案,注册数据库 270 包括 P2P 网络中每个现有通道的条目。注册数据库 270 用关于 P2P 网络中的通道的创建和取消的信息更新。注册数据库 270 中表示的每个通道可包括关于通道成员的信息和可选的其它信息,举例来说诸如指示哪些成员被邀请预定至通道以及哪些成员已请求加入通道的信息。例如,考虑希望更新其与特定通道相关的文件的同级。对请求作出响应,通道管理器 230 可被配置来从在线状态服务 250 接收当前在线的相关通道的成员的联系信息。这种信息可用于创建通道覆盖(如果需要用于与这些成员通信并更新修订历史日志)和存储为与同级相关的文件。这可通过利用存储在例如注册数据库 270 和在线数据库 265 中的信息完成。

[0188] 现在参考图 3,图 3 图示了根据本发明的一个实施方案的修订历史日志的实施例。在下列论述中,图 3 中的修订历史日志实施例用于展示本发明的特征。图 3 包括标注为 T1、T2、T3、T4、T5 和 T6 的 6 个层次树或有向图,各图示了修订历史日志中的方案或情形的可能表示的不同实施例。在一些实施方案中,T1 到 T6(“T1-T6”)可实施为单个统一数据实体,而在其它实施方案中,其可实施为 6 个单独数据实体。T1-T6 中的每个顶点显示针对各自修订而计算出来的内容哈希码。应注意图 3 给定为一个实施例并且不应被解释为限制性。如本领域技术人员将明白,如图 3 中所示的修订条目仅显示存储在实际条目中的部分信息。

[0189] 为清楚和可读性起见,在下列实施例的说明中,哈希码标注为‘h1’、‘h2’等作为内容哈希码。然而,在其它实施方案中,除内容哈希码外或替代内容哈希码,可使用元数据哈希码,以追踪进程路径内的修订条目。

[0190] T1 显示对应于给定的最早已知文件 F1 的修订历史日志的一部分的一个实施例。最初,从原始文件 F1(即,最早已知修订)的内容中计算出内容哈希码 h1。针对这个文件创建修订条目,所述修订条目包含其内容哈希 h1 和关于所述文件的另外元数据(诸如例如修订标识符、文件名、创建时间、历史日志中的最早内容哈希等等)。随后,F1 的文件内容和/或文件元数据因修改事件而被修改。根据特定实施方案,当修改事件发生时,如果文件内容在文件系统上被修改,那么针对新文件版本的内容计算出新哈希码 h2,并且创建新修订条目。类似地,如果使用元数据哈希码,那么当元数据在文件系统上被修改时,可创建新的修订条目,并且可计算出新的元数据哈希码并将其存储在新修订条目内。新修订条目包括新内容和/或元数据的新哈希码 h2,并且链接到包括哈希码 h1 的先前修订条目。文件的后续修订以相同方式记录,如在图 3 中的元素 T1 中所示,元素 T1 表明从 h1 延伸到 h5 的进程路径。应注意,如下文详细说明,即使文件内容保持不变,修改事件也可能引起新修订条目的创建。

[0191] 如本发明中所公开,修订历史日志的配置特性在于文件可分支成许多条进程路径。这可在不同情况下发生,这些情况在文件共享环境背景下尤其常见。例如,相同通道的两个或更多用户各可在离线时改变相同共享文件的给定实例,并且接着变成在线。在另一实施例中,单个用户可在不同场合下登录到不同离线节点并且修改存储在每个节点中相

同文件的实施例（例如，用户可在工作时编辑并修改文件，并且接着在家中修改相同文件的另一副本，而无需在家中的计算机和工作用的计算机之间更新）。根据本发明的特定实施方案，在更新后，修订历史日志将包含由不同节点中的两个用户或相同用户创建的新修订的修订条目。或如上所提及，当用户请求将文件恢复到其过去修订之一，并且接着继续修改所述文件时，分成多条进程路径的情况可能发生，导致新路径从所选定的过去修订中分支出来。

[0192] 图 3 中图示了这个方案；例如参见分支成两条进程路径的修订条目 h5，一条以 h7 继续并且另一条以 h6 继续。

[0193] T1 是显示形成 7 条进程路径的基础的初始进程路径的实施例，所述初始进程路径从一个最早已知修订条目（包含 h1）发展到 7 个修订条目，每个修订条目表示（在这个实施例中）相同文件的不同当前修订（此处表示为包含 h6、h15、h1、h12、h27、h26 和 h29 的修订条目）。这些进程路径中的每条路径表示文件的不同修订，其中当前存在 7 个不同进程。

[0194] 在这个实施例中，每个进程结束于表示替代但为当前的文件版本的当前修订条目。因此，在本发明中，这些版本的全部被视为有效，并且不需要冲突解决方案来确定哪个版本低于或高于另一版本。

[0195] 这与普通同步和修订控制系统的运行相反，所述普通同步和修订控制系统通常被配置来在给定文件存在多个可用当前修订时（例如，在两个不同用户已经用两种不同方式改变文件的情况下）起始冲突解决方案程序。文件共享环境中的普通修订控制系统和以 P2P 为基础的文件同步系统通常需要一个或多个用户来解决这种冲突，或者所述系统可以尝试根据某一冲突解决策略自动解决冲突。不同于本文中所公开的修订控制系统，不管在这些现有技术方法中使用何种特定冲突解决方法，并非文件的每个实例都被维持并被视为文件的进程路径的一分部，这是因为一些实例通常例如由于冲突解决程序而被丢弃。

[0196] 如本发明中所公开的修订历史日志的配置的另一特性在于其可能还能使作为相同副本并且具有不同文件名的文件相关，所述文件通常被文件系统视为两个不相关文件。由于每个文件在修订历史日志中由对应的修订条目表示，并且每个修订条目包含对应文件的内容哈希和 / 或元数据哈希码，所以文件不仅通过其文件名（或不同标识符）追踪，而且还通过其实际内容和 / 或元数据追踪。因此，如果发现两个（或更多个）修订的内容哈希相同，那么可检测到所述两个修订之间的关系。当所述两个修订改变时，其可保持单独进程路径，所述路径记录在修订历史日志中。应注意，在一些实施方案中，具有相同内容的多个文件可与单个修订条目相关，并且所述修订条目可与所有相关文件的文件名列表相关。

[0197] 检测两个修订之间的关系对于许多用途而言是有益的。例如，考虑到第一文件何时被用户复制到参考区域中的另一位置，例如以提供所述第一文件的备份副本，所述第一文件和备份文件之间的关系可用于通过检索第一文件的进程路径显示备份文件的历史。由使具有相同内容的不同修订相关的能力得到的另一益处是减小存储空间的能力。例如，可通过使表示具有相同内容的修订的所有修订条目与存储在存储空间中的内容的单个副本相关，而非维持每个修订的副本而实现这存储空间的减小。根据特定实施方案，修订引擎 110 被配置来标识何时在参考区域中创建新修订、计算其内容哈希码、将其与记录在修订历史日志中的现有内容哈希码进行比较并且确定包括相同内容的其它修订是否已经记录在

历史日志中。

[0198] 图 4 是示出了根据本发明的一个实施方案的修订历史日志更新操作的流程图 400。如上文所说明,文件修订引擎 110 包括修订管理器 115,所述修订管理器 115 被配置来创建和管理关于正被修订控制系统 100 中的文件修订引擎 110 监控的文件和文件夹的修订历史日志。根据特定实施方案,修订管理器 115 被配置来执行本文参考图 4 描述的操作,图 4 图示了对修改事件的标识作出响应而发生的操作。修改事件包括但不限于文件内容的修改、重新命名文件、删除文件和创建新文件。

[0199] 在阶段 410 中,标识修改事件。根据特定实施方案,文件系统监控器 120 或用户将修改事件通知给修订管理器 115。或者或此外,修改事件可通过主动搜索(“扫描”)文件系统中的这类事件而标识。通用文件系统(或操作系统)通常根据所实施操作系统的特性使文件系统监控器能获得已经例如经由合适的系统呼叫、API、系统功能或通过预定至事件分派器而修改的文件的文件系统所使用的文件标识符(例如,完整路径名或文件 ID 号)。通常,也可从文件系统中获得特定类型的修改事件的指示。在较不常见的方案中,其中所述类型的修改事件非直接可用,可基于文件的元数据、可从修订历史日志中获得的当前修订和/或相关修订条目或其组合推断修改事件。例如,当扫描文件修改时,如果文件经过计算产生不同于当前修订条目中表示文件系统上的这个文件的内容哈希码的内容哈希码,那么其指示所述文件已经改变并且可能触发修改事件。在一些实施方案中,修订管理器 115 还被配置来核实修改通知真正表示文件的修改,其不同于例如实际上不改变被访问文件的内容或元数据并且因此在大多数情况可忽略的“访问操作”。

[0200] 如上所提及,修改事件可引起创建修改文件的新修订。关于所修改文件的另外元数据通常还可从文件系统中获得(例如,经由文件系统监控器 120 或直接从文件系统或操作系统中获得),例如,修改事件发生的时间、修改文件的用户的名称、安全信息诸如 NTFS 文件系统的安全描述符、关于文件的哈希码等等。根据特定实施方案,阶段 420 包括一旦已标识修改事件,立刻从文件系统或操作系统接收关于所修改文件的新修订的另外元数据。如果新修订的内容哈希码和元数据哈希码不可从文件系统或操作系统中获得,那么其可从文件内容中计算出来。因此,在阶段 420 完成后,至少可获得关于所修改文件的下列信息:所修改文件的唯一标识符,以及(如果需要)文件的更新内容和所计算出来的哈希码。

[0201] 根据特定实施方案,在下列阶段 430 中,从修订历史日志中获得相关修订条目(例如,通过修订管理器 115)。所述“相关修订条目”是对应于实施修改事件(其引起新修订)的修订的现有修订条目。所述修订条目可基于任何类型的可获得并且用于为修订历史日志内的修订条目做索引(例如,作为密钥)的标识符(或标识符的组合)而定位。

[0202] 修订条目可由多个密钥做索引以允许在不同方案中检索修订条目。一些密钥(至少在如上所提及的修订历史日志域内)可以是唯一的,其它密钥可是基本上唯一的(例如,UUID),而还有其它密钥可以不是唯一的并且对应于若干修订条目。

[0203] 例如,当前修订条目(与当前修订相关的修订条目)还可由对应修订的完整路径和/或内容哈希码和/或元数据哈希码做索引。因此,当仅存在一个关于参考区域中的每个当前修订的当前修订条目时,如果当前修订条目根据对应修订的完整路径做索引,那么当前修订条目可例如通过从修订历史日志中的当前修订条目中搜索与相关完整路径相关的条目而检索。在另一实施例中,参考区域上的每个文件被指派一个唯一标识符。这个唯

一标识符可在阶段 420 期间检索,并且随后可用于通过其唯一密钥检索相关修订条目。

[0204] 如果修改事件是重新命名事件或移动事件,其中现有文件被给定新文件名和 / 或不同路径,那么从文件系统中获得的元数据可包括旧文件名(或另一类型的标识符)和新文件名两个。如果文件在文件夹之间移动,那么这个信息还可包含关于其过去所处的旧文件夹以及其现在所处的新文件夹的详情。所修改文件的相关修订条目可基于其密钥而定位,包括其旧路径或其内容哈希码或如上文说明的任何其它可用标识符。如果修订条目仅通过内容哈希码做索引并且单个内容哈希码与若干当前修订条目相关,那么相关修订条目可通过使用内容哈希码和另外标识符(诸如到所搜索文件名的路径或者如上文所说明的另一可用标识符)而定位。

[0205] 根据特定实施方案,在阶段 440 中,针对新文件修订创建新修订条目,并且关于所述新修订的元数据被计算出来并且被分配给新修订条目,包括新修订条目的可用标识符,诸如举例来说其内容哈希。

[0206] 在删除事件的情况下(即,文件被删除),除如下文阶段 440 和 450 中所述的创建新修订条目并且将其链接到当前修订条目以外或取而代之,还可存在其它指示所述事件的方法。例如,取代创建新修订条目或作为其补充,相关文件已被删除的指示还可被添加到所删除文件的相关修订条目。删除指示可为任何类型的元数据,包括例如特定删除时间戳和删除旗标。或者,整个进程路径可删除,并且可可选地存储在排除列表中(所述列表列出删除的进程路径),例如以记住哪些文件可不再被修订管理器 115 监控。应处置的文件删除的特定方式可由存储在修订策略单元 140 中的策略指示。

[0207] 接下来在阶段 450 中,新修订条目在修订历史日志中链接于适当位置,使得新修订条目变成进程路径中表示当前修订的当前修订条目。根据特定实施方案,新修订条目链接到文件的当前修订条目,从而继续修订历史日志中的进程路径。

[0208] 在成功链接于修订历史日志中后,所述新修订条目变成文件的“当前修订条目”。根据特定实施方案,在修订历史日志用新修订条目更新后,程序可终止或取而代之恢复到阶段 410 以处置新修改事件。

[0209] 在文件重新命名事件的情况下,作为链接新修订条目到当前修订条目的替代,新修订条目可保留为不链接,实际上在修订历史日志中开始新进程路径。更具体来说,重新命名事件也可理解为一系列“删除”和“创建”事件。因此,在一些实施方案中,修订管理器 115 可通过执行删除事件的操作和创建事件的操作而处置重新命名事件。

[0210] 在创建事件中,在参考区域上创建新文件,修订引擎 110 开始监控所述文件,并且针对所述新修订(包括至少一个标识符(例如,元数据哈希))创建新修订条目。在这种情况下,新创建的文件元数据哈希与新修订的文件名相关并且所述新修订条目开始新的单独进程路径。因此,例如,可通过创建与先前修订条目无关联但包含与先前修订(在重新命名事件之前)的修订条目中相同的内容哈希码和其它相关数据的新修订条目而实施重新命名事件。由于在新进程路径和先前进程路径中包含相同内容哈希码,所以可使用相同内容哈希码使所修改文件的两条(或更多条)路径相关并访问所述路径。

[0211] 例如,参见展示重新命名事件的可能结果的图 3 中的元素 T3 和 T4。T3 表示其中最近修订具有内容哈希 h30 的一条进程路径,而 T4 显示与 T3 分开并且仅包含具有相同内容哈希 h30 的一个修订条目的另一进程路径。因此 T4 可表示重新命名事件后 T3 中的当前修

订的新修订。由于重新命名事件（或复制操作）后文件内容与其先前内容相同，所以即使新修订条目不直接链接到先前修订条目，而是开始新进程路径（如由图 3 中的 T3 和 T4 所示），重新命名的文件的修订历史也可基于内容哈希并且可选地基于其它标识符从修订历史日志中获得。例如，考虑搜索由 T4 表示的修订的修订历史。相关文件的内容哈希是 h30，但是相同内容哈希同样适用于 T4 中的当前修订。因此，将可能使 T4 与 T3 相关并且获得在 T3 中可用的 T4 的历史。

[0212] 为指示文件夹重新命名已经发生，处在内部的文件和文件夹还可经过管理来保存其历史。若干符号可用于指定所发生的文件夹重新命名事件。在一个实施例中，重新命名的文件夹内的每个文件和文件夹可表示为具有“重新命名”修改事件，以指示到已经修改的每个文件和文件夹的路径。在另一实施例中，如果文件夹也如文件一样由修订条目表示，那么其可足以通过创建新修订条目并使新修订条目与其进程路径相关而将文件夹指定为“重新命名”。在上文所述的两个实施例中，保存文件夹中所含的文件历史，使得重新命名操作在由新名称从其在修订历史日志中的已知历史中标识出来时不使文件失去关联。

[0213] 根据特定实施方案，除如上文所述处置修改事件外，本发明还涉及针对存储在修订历史日志中的过去事件和其它信息分析修改事件，以及提供对关于所记录文件修订历史的这类操作的解释。例如，修订管理器 115 可被配置来标识恢复事件。在修改事件中，文件可经过修改以包括记录在其进程路径上的过去修订的内容的相同副本。根据特定实施方案，修订管理器 115 被配置来执行下列操作：计算新文件的内容哈希（或使用在阶段 440 中之前计算出来的内容哈希）以及将所计算出来的内容哈希与存储在修订历史日志的修订条目中的内容哈希码进行比较。如果在表示当前修订的相同进程路径中的过去修订的修订条目中发现相同内容哈希，那么修订管理器 115 可通知恢复事件已发生。表示新修订的修订条目可经过更新以包括一个符号，所述符号指示这个修订是可能的恢复事件的结果以及可选地涉及修订历史日志中的先前修订条目。

[0214] 例如，在图 3 中的 T3，其中进程路径（h30）中的最后（当前）修订与相同进程路径（h30）中的第一修订相同，并且因此表示恢复事件。如果在修订历史日志的不同进程路径中找到相同哈希码（即，文件无法通过在修订历史日志上从新修订条目往回撤销而定位），修订管理器 115 可标识已经可能发生复制事件。表示新修订的修订条目可经过更新以包括一个符号，所述符号指示这个修订是可能的复制事件的结果并且可能涉及修订历史日志中的（一个或多个）先前修订条目。

[0215] 修订引擎 110 还可被配置来标识并处置合并事件。考虑从相同分支发起并且与不同修订并列进展的两条进程路径。在某个点，对应于两条进程路径中的修订条目的修订可经过修改以包括相同内容。根据一些实施方案，修订管理器 115 可被配置来标识（例如，对修改事件作出响应或通过扫描修改）两条不同进程路径中的两个当前修订条目的内容哈希相同。根据特定实施方案，修订引擎 110 可被配置来将其进程统一为单条进程路径。修订管理器 115 可被配置来创建将两个单独修订条目统一的新修订条目，其中修订历史日志包括指示这个修订条目是可能的合并事件的结果的符号。例如，这个指示可用新修订条目上的符号，或用新修订条目的两个（或更多个）先前修订条目中的符号实施。例如，参见图 3 中的 T2，其中两条进程路径统一为单个进程路径（h41）。

[0216] 根据特定实施方案，一旦修改事件已发生并且由修订引擎 110 分析为可能表示恢

复事件、复制事件、放弃事件等等,系统 100 立刻被配置来例如经由显示器 160 向用户指示这些事件。因此,用户可被通知关于新修订的历史和新修订与其过去修订之间的历史关系。例如,消息可在显示器上 160 上显示,所述消息指示特定文件已经恢复回先前修订、修订内容被复制到文件等等。

[0217] 通过执行修改事件的前述记录,以及维持修订历史日志,所提出的系统也可使文件系统中的先前不相关的文件相关。根据特定实施方案,修订引擎 110 被配置来使在文件系统中不一定相关的表示文件的修订条目相关并且执行关于所相关的修订的各种操作,例如:显示特定修订的所有副本、将特定修订的冗余副本复制到另一存储位置、从过去修订数据库 135 中移除这类冗余修订、在操作系统 Shell 环境(例如,Windows 操作系统上的 Windows Explorer-explorer.exe)中打开相关修订位置、检索由特定作者修改的修订、检索在特定期限内修改的修订等等。这些操作和相关性中的一些可例如通过使用内容哈希码以在修订历史日志内的不同修订条目内容之间进行比较而实现。根据特定实施方案,在用户界面 150 和显示器 160 的帮助下,用户可管理并执行这些操作。

[0218] 根据特定实施方案,修订引擎 110 被配置来使用修订历史日志中所包含或关于其的数据和信息,对命令作出响应而执行另外操作。这类操作包括但不限于恢复、替换、放弃和合并。先前描述修订引擎可如何分析修改事件并且从而使这些事件与其它操作相关,而此处对特定命令作出响应而执行这些操作。命令可自动发出,或通过与系统 100 互动的另一系统发出,或由用户经由例如用户界面 150 发出。

[0219] 对恢复命令作出响应,修订引擎 110(例如,经由修订管理器 115)利用来自修订历史日志中的可用信息和可能还来自过去修订储存库 135 的可用信息,以用文件的一个过去修订替换文件的内容和/或元数据。恢复命令可指示被请求的特定过去修订以及可能的应被检索的数据类型(例如,内容、元数据等等)。如上所述,过去修订条目可由某种类型的唯一标识符标识并且这些标识符可用于检索特定过去修订条目。在一些实施方案中,可针对特定当前修订条目发出命令。例如,恢复命令可包括数据应被替换的当前修订。在其它实施方案中,命令可在不指定当前修订的情况下发出。修订引擎 110 可从过去修订储存库 135 中获得过去修订(如果可获得),并用过去修订的各自数据替换对应于当前修订的文件(例如,在文件系统中)的相关数据(例如,其部分或全部内容),最终创建包含部分修订的部分或全部数据的新修订。取代替换现有修订中的数据或作为其补充,修订引擎 110 还可配置用于用从过去修订中检索出来的数据创建新修订。

[0220] 根据特定实施方案,对恢复命令作出响应而在文件系统中创建新修订可触发将最终针对新修订创建新修订条目的修改事件,指示这个修订是恢复事件的结果并且可选地涉及修订历史日志中的过去修订条目。根据特定实施方案,如上所说明,新修订条目链接到修订历史日志中的适当位置。

[0221] 对替换命令作出响应,修订引擎 110(例如,通过修订管理器 115)利用可从修订历史日志中获得的信息,并且也可使用过去修订储存库 135 和/或被监控的文件系统 125,以定位所请求的过去修订或当前修订。替换命令类似于恢复命令;但是,其不限于当前修订的过去修订之一,而是也可包括其它修订。

[0222] 对合并命令作出响应,两条或更多条进程路径(“合并的进程路径”)由修订引擎 110(例如,由修订管理器 115)聚合以形成以一条进程路径继续合并的路径的单个修订条

目(“合并的修订条目”)。根据特定实施方案,合并命令可包括将被合并的两条或更多条进程路径的说明和被指定为合并的修订条目的现有修订条目的说明。因此,合并命令可包括用于标识将被合并的不同修订条目(或进程路径)的一个或多个标识符。根据特定实施方案,创建新修订条目以充当合并的进程路径的合并的修订条目。接着合并的进程路径与合并的修订条目链接,以指示其现在聚合成单条合并的进程路径。或者,可在修订历史日志内使用另一符号来指示特定路径现已被合并,前提条件是所述符号充分标识合并的路径。根据特定实施方案,合并命令可包括将被并入合并的修订条目的数据,例如文件内容数据、文件元数据或上述的组合。如果合并的修订条目与存储在过去修订储存库中的修订相关,那么合并命令可触发“替换操作”以从过去修订储存库 135 中检索相关数据并且将其存储在储存库 125 中以确保合并的修订条目与所监控文件系统 125 上的修订相关。

[0223] 对放弃命令作出响应,一旦在一条进程路径(“优选进程路径”)中可获得新修订条目,一条或多条进程路径立刻由修订引擎 110(例如,由修订管理器 115)指定为中断(“放弃的进程路径”)而支持所述优选进程路径。不同于合并命令,在执行放弃命令时,无需指示优选(或获胜)修订。根据特定实施方案,一旦执行放弃命令,修订引擎 110 立即监控优选进程路径并且一旦在所述优选进程路径中可获得新修订,所放弃的进程路径立即全部聚合(例如,链接),其中新修订条目对应于优选进程路径上的新修订。所放弃的进程路径可经过更新以包括这些是放弃的进程路径的指示。

[0224] 在图 3 的 T6 中示例放弃命令,其中包含 h55 的进程路径(用“s”标记)被放弃来支持包含 h53 和 h54 的进程路径。当继续在 h54 到 h56 中(文件的新当前修订条目)可获得新修订时,包含 h55 的所放弃的进程路径也可指向(继续进入)新修订条目 h56。

[0225] 如下文进一步说明,在一个同级上包含文件和文件夹以及其修订历史日志的参考区域可与同样包含相同通道的文件、文件夹和修订历史日志的其它同级的参考区域比较。当这个比较确定存在需由一个同级更新的修订条目时,更新程序可在所述同级和覆盖中的至少一个另外同级之间开始。

[0226] 图 5 是展示根据本发明的一个实施方案的 P2P 网络背景下的更新周期的运行的流程图。如上文参考图 2 所说明,修订控制系统 100 可在 P2P 网络背景下运行,其中存储在 P2P 网络中的一个节点上的文件和文件夹的修订历史日志、当前和过去修订可被更新并与存储在其它节点(其为相同网络中的同级,或更具体来说,为相同通道的成员)上的文件和文件夹的修订历史日志、当前和过去修订同步。

[0227] 在图 5 的背景下,“更新周期”是接收关于通道中的文件或文件部分和文件夹的可用更新修订的信息以及可选地使用这信息以从通道的其它成员获得部分或全部新修订的程序。这个程序允许通道成员包括基本上相同的修订历史日志,并且最终获得存储在与特定通道相关的其它节点的参考区域上的文件和文件夹的可用修订。因此,潜在地,通道的所有成员可具有修订、文件和文件夹的完全相同副本。

[0228] 在第一阶段 510 中,起始更新周期。根据特定实施方案,可对不同事件作出响应而起始更新周期。可对用户的明确手动请求(例如,经由用户界面 150)作出响应而手动起始更新周期。或者或此外,更新周期可根据预定义策略而自动起始。例如,可根据预定义时间表起始更新周期,或者更新周期可在通道成员与通道覆盖断开达预定义时段的情况下起始。根据一些实施方案,用于起始更新周期的条件和策略可存储在通道配置单元 235 中。



[0229] 根据特定实施方案,在阶段 520 中,对起始更新周期作出响应,发出更新询问。更新询问是“拉式”通信,其中通道成员(发端成员)从相同通道的其它成员中搜索更新。更新询问可包括定义询问的不同标准,例如哪些实体(例如,特定文件夹和/或文件)被调查、发端成员所做的最后更新日期、已知的实体内容哈希码、已知的实体元数据哈希码、或发端成员已知的部分或全部进程路径等等。如果更新询问未指定标准,那么可使用默认标准例如以检索自最后更新起的所有更新。根据特定实施方案,网络修订控制系统 210 可被配置来经由 P2P 引擎 240 发出输出的更新询问。可例如对在通道配置单元 235 中指定的预定条件已得到满足作出响应而发出这些询问。根据所用的特定覆盖几何形和技术,更新询问通过通道覆盖可能传播到通道的所有成员或部分成员。在单个成员链接到多个其它成员的情况下,更新询问可分支成不同传播路径。

[0230] 对更新询问作出响应,接收更新询问的通道成员检查在其修订历史日志中发生的任何相关更新,并且如果这些更新发生,那么其将“更新响应”发回给询问成员。更新响应是指示相同通道的一个或多个成员发生的一个或多个更新的消息,并且因此可获得一个或多个新修订。

[0231] 根据特定实施方案,更新询问继续传播通过通道覆盖直到其满足某些类型的预定条件,例如其已达到预定义数量的成员、其已满足覆盖性质(例如,耗尽特定数量的 DHT 查找)或其已失效。例如,更新询问可包括被配置来倒数计秒并当其达到零时指示更新询问的进一步传播的中断。另一个这样的实施例是使用定时器,其中更新询问被给定有限的寿命并且更新询问被配置来一旦其寿命结束便失效。

[0232] 根据特定实施方案,更新响应可包括一组数据元素,所述数据元素使接收成员能标识所发生的修改类型、创建一个或多个新修订条目并使一个或多个新修订条目与在其修订历史日志内的正确位置相关。例如,更新响应可包括下列的部分或全部:任何新文件修订和/或文件夹修订的元数据,包括修改内容的哈希和所修改的元数据、任何新文件修订和/或文件夹修订的相对路径、关于新修订的标识符、关于在一个特定成员或多个特定成员上所修改的全部或部分进程路径的信息,以及可能的整个修订历史日志。

[0233] 根据特定实施方案,更新响应中的传达数据应足以正确地使所接收的新更新修订条目与响应者的修订历史日志内的相关现有修订条目相关。在一些实施方案中,对于单个新修订条目,更新响应中的传达数据可包括例如,标识进程路径中的前任修订条目的唯一父 ID。在其它实施方案中,更新响应中的传达数据可包括标识包含新修订条目的进程路径的唯一 ID、或标识包含新修订条目的进程路径中的“最早已知”修订条目(对应于最早修订)的 ID。

[0234] 因此,例如,为指示特定进程路径具有新修订条目,传达新修订条目的一个或多个标识符、或新修订条目的实际内容,以及指示新修订条目在修订历史日志中应处的位置(例如,通过提供其等相关的父 id 和/或进程路径 id) 足矣。根据特定实施方案,如果在更新响应中接收的信息不足以使新修订与修订历史日志相关(例如,在询问成员的修订历史日志中未找到所提供的父 id),那么可起始新更新询问以检索缺失的信息。

[0235] 取代由询问消息触发或作为其补充,指示可用更新的消息还可以是由通道的另一成员起始的“推式”通信的结果。这种类型的消息在本文中称为“更新消息”。在一些实施方案中,已经更新一个或多个其所监控的文件或文件夹(或出于另一原因)的通道成员可发

送更新消息给相同通道的成员,告知其更新发生并且可获得新修订。因此,在一些实施方案中,更新周期可在更新消息被成员接收后始于阶段 530 中。根据特定实施方案,对修订历史日志的更新作出响应,(例如,在修订引擎 110 已处理修改事件的情况下)通道管理器 230 可经由 P2P 引擎 240 将更新消息发送到通道的其它成员。

[0236] 由于更新询问和更新消息两者通常都由一个成员起始并且被传输给通道中的多个成员,所以两种类型的消息可以类似方式传播通过覆盖。本领域技术中已知在通道覆盖中从一个(发端)成员传送消息到其它(“接收”)成员的各种方法,并且因此将不在下文中详细论述所述方法。在原理上,取决于通道覆盖的类型和几何形,消息可传送到相同通道的一个或多个成员。一个实施例对于“无结构覆盖”而言是典型的,其中对经由通道覆盖直接连接到发送成员的成员执行传送消息。在示例性的在本领域技术中通称“延伸环”方法中,消息可连续转送到相同通道覆盖中的其它连接的成员并且因此其可传播通过覆盖的一部分,或可能通过整个通道覆盖。

[0237] 在其它几何形中,诸如由结构化覆盖和以 DHT 为基础的覆盖示例的几何形,消息可转送到通道的特定成员,例如到用作“超级同级”的成员、到满足特定 P2P 覆盖条件的成员、到被指派一个角色以包含消息中所传送的特定信息的成员、和 / 或满足特定距离函数的成员。在 P2P 网络中,术语“超级同级”普遍指具有提升状态的同级,例如,归因于被连接达较长一段时间,和 / 或具有充足带宽,因此其在与未提升到“超级同级”状态的同级互动时相比需要较高数据持续性和可访问性的情况下可被 P2P 网络中的同级使用。超级同级在 Buford, Yu and Lua (2009, pp. 12, 37-38, 55-56f, 63-64, 129-131, 153) 中加以详细论述。

[0238] 根据特定实施方案,在下一阶段 530 中,更新响应可沿着返回路径再发生,使得发起更新询问的每个成员接收可能具有来自沿着询问的返回路径的每个同级的附加结果的响应。在其通过通道覆盖传播回发端同级期间,更新响应可能经历修改和充实程序以最终包括可由其路径中的成员获得的最新信息。由于更新响应通过覆盖传播回到发端成员,所以其可穿过其它成员。这些成员(“接收成员”)因此接收更新响应,并且可接着(在修订管理器 115 的辅助下)比较其自身的修订历史日志与输入的更新响应内的信息。所述比较可指示接收成员的修订历史日志的当前状态与从更新消息中显而易见的修订历史状态比较是较旧更新、较新更新或同等更新。根据一个实施方案,如果更新响应证明接收成员的历史日志的状态是较旧更新,那么接收成员更新可用更新列表并且可将更新响应发送给其它成员(例如,在无结构化覆盖中,发送给直接连接到其上的同级)。因此,由发出更新询问的一个成员触发的更新响应可用于更新通道中的许多成员。

[0239] 根据另一实施方案,如果更新响应证明接收成员的修订历史日志是较新更新,那么更新响应可用较新更新数据充实,所以其包括可从接收成员中获得的较新更新数据。所修订的更新响应接着可从接收成员发送到其它成员以在网络中进一步传播。

[0240] 根据其它实施方案,如果更新响应中的数据指示更新响应与接收成员的修订历史日志同等更新,那么更新响应可保留而无需任何变化。根据另一实施方案,接收成员的更新响应和修订历史日志都可具有非重叠更新数据。即,当数据从更新响应中出现时,更新响应可指示存在对于接收成员而言是新的数据,并且同时,接收成员可具有对于最近更新而言是新的新数据。根据特定实施方案,在这种情况下,接收成员可根据更新响应中可获得的数据更新其可用更新列表并且还可修改更新响应以除其发端数据以外还包括现可从接收成

员中获得的新数据。这样,实际上创建较新更新的更新响应。应注意更新消息在其传播到通道的不同成员时可能经历相同类型的比较和更新程序,所述程序已在上文中参考响应消息而描述。在比较并可能更新后,更新响应还可基于询问消息的传播返回路径传播到通道的其它成员。

[0241] 为示例修订历史日志和更新响应具有非重叠数据的情况,考虑成员 P10 具有自其最后在线更新起已离线修改若干次的文件 F10。还考虑在这种情况下,另一成员 P20 也以不同方式改变文件 F10,并且传播这个修改到预定至 P10 和 P20 的通道的剩余成员。接着,在随后的时间里,P10 和 P20 变成在线,并且 P10 可参与由 P20 起始的要求针对 P10 更新的更新询问。P10 在参与这个更新的传播路径时可得知其正缺失 P20 的修订,而同时 P10 具有针对 P20 的更新。P10 接着可发出包括 P20 缺失的其修订的更新响应,并且还更新其可用更新列表以包括 P20 所具有的修订。在另一实施例中,如果 P10 需要除更新响应中提供的更新以外的附加更新,那么除发出更新响应外,P10 还可发出其自身的更新询问以得知有关关于 F10 的缺失更新的更多情况。从而 P10 和 P20 两者最终都得以更新。

[0242] 可能发生的是,由不同传播路径传播的更新询问起始许多从每条路径返回的返回更新响应,其中一些更新响应与其他更新响应相比是较新更新或不同更新。根据特定实施方案,这些聚合消息在其聚合点被修改并更新,以从所有输入消息的集合中生成最新的更新消息。例如,如果两个更新响应被一个成员接收,那么各相对于彼此用非重叠数据更新,来自所述两个更新响应的相关数据可被接收成员合并为新更新响应以使现在包括两个消息的相关数据。最终,最新消息到达覆盖中的发端成员。一旦更新响应到达通道中的发端成员,其修订历史日志立刻可被更新(如果需要)。

[0243] 更新响应(或更新消息)积累在发端成员(起始询问的成员)中,并且可用更新列表在发端成员中创建,并且可在沿着返回路径的其它成员中创建。可用更新是从通道成员中缺失的文件和文件夹修订和/或修订条目。所述可用更新列表可存储在存储装置(例如,储存库 125、130 和/或 135)上或存储在成员节点内存中。

[0244] 在一些实施方案中,所接收的一个或多个更新用于更新发端成员(或可选地接收更新的任何其它成员)的修订历史日志。新修订条目现在可创建并添加到修订历史日志,其表示在其它成员上可用并从修订历史日志中缺失的新修订。修订条目可包括对应修订目前无法从所存储的与更新节点相关的修订中获得的指示。

[0245] 可制定用于定义更新询问的一个或多个标准以请求不同类型的数据。例如,更新询问可涉及搜索下列类型的数据中的一种或多种:特定的一个或多个修订条目、特定的一个或多个进程路径、整个修订历史日志、仅对应于当前修订的修订条目等等。更新询问还可指定对应的修订是否应与其条目一起检索。更新询问的制定可取决于系统的不同特性,诸如可用内存、通信带宽、处理能力和当前处理负荷、通道中的成员数量、目前在通道覆盖中连接的成员的数量等等。更新询问的制定还可取决于其它因素,诸如用户(或管理员)偏好以及所实施的方法或运算法则。更新询问的确切格式可例如在通道管理器 230 中配置。

[0246] 针对更新询问生成的更新响应根据更新询问中指定的要求制定。

[0247] 一旦更新响应被询问成员(起始更新询问的成员)接收,所述响应中的信息与已由询问成员获得的信息比较。可使询问成员的修订管理器 115 可运行以从网络控制单元 210 接收关于可用修订条目的信息并且比较所接收条目与其对应修订历史日志中的条目,

以确定所接收的修订条目是否从其修订历史日志中缺失。如果其缺失，那么修订管理器 115 可运行以链接修订历史日志中的缺失修订。为此，修订管理器 115 可使用关于每个所接收修订条目的先前修订条目的信息（例如，使用修订条目唯一 ID、元数据哈希码、或通过具备修订条目数据并标识其先前修订的其它合适标识符）并搜索包含基本上相同信息（例如，相同标识符）的修订条目以标识修订历史日志中的先前修订。如果找到先前修订条目，那么修订管理器 115 可链接所接收的修订到其在修订历史日志中的对应先前修订。

[0248] 但是，可能发生的是，如果所接收修订的先前修订也从修订历史日志中缺失，那么修订管理器 115 无法链接所接收修订到修订历史日志。

[0249] 如果修订管理器 115 未能在修订历史日志中使新修订条目在其进程路径中相关，那么其可通过使用与所接收修订条目一起获得的信息获得的先前修订的标识符而标识缺失的修订条目。另一更新询问接着可发送，其包含基于其标识符检索先前修订条目的标准设置。

[0250] 这可能在例如更新询问限于仅请求对应于当前修订的修订条目更新的情况下发生。在可用网络带宽不足、网络带宽昂贵等等的情况下，这种限制可是必需的，例如以减小网络通信量，尤其在连接到大量用户的拥挤通道覆盖中。

[0251] 另外的更新询问可重复起始直到获得将所有缺失的修订条目链接到修订历史日志所需的所有数据。

[0252] 在其它情况下，请求更新可制定为检索所有进程路径，而非将更新询问限于仅检索特定修订条目（例如，对应于当前修订）。例如，如上所述，如果所接收条目的先前修订条目缺失，那么这种更新询问可发出。一旦接收进程路径，修订管理器 115 可立刻确定每条进程路径中的哪些修订条目缺失，并且将进程路径的缺失部分添加到修订历史日志。

[0253] 但是，传达进程路径（和所有修订历史日志）可招致通信基础设施上的大量负荷并且还可能导致网络上传输的数据的大量冗余。因此，在一些情况下，仅请求进程路径的特定段更有效。

[0254] 例如，第一更新请求可限于检索关于对应于当前修订的修订条目的可用更新。当接收更新响应时，例如，如果其先前修订条目也从修订历史日志中缺失，那么修订管理器 115 可能无法在修订历史日志中使所接收修订条目相关。当这些先前修订条目从修订历史日志中缺失时，可生成请求所接收修订条目的进程路径段的新更新询问。段包括关于多个修订条目的信息，并且因此可减少获得可用更新列表所需的处理量。

[0255] 在本实施例中，所述段被定义为自所接收修订条目直到最近交汇点的进程路径的部分。在这个背景中“交汇点”指前进到多条进程路径的修订条目。例如，参见图 3，在从 h1 延伸到 h29 的 T1 进程路径中，包括三个交汇点，h8、h7 和 h5。如果更新响应中的所接收修订条目包括修订条目 h29，那么最近交汇点将为 h8，并且因此将被检索的进程路径段在另外更新询问标准中将被定义为“以 h29 为结尾的段”。这个更新询问的响应更新将从 h8 延伸到 h29。如果仍需要更多信息来将所接收修订条目（h8 到 h29）链接到修订历史日志，那么可如上所述生成、发送并处理请求下一进程路径段（直到 h7）的新更新询问。这可以重复直到获得所有所需数据。

[0256] 一旦形成可用更新列表，运行立刻转到下个阶段 540，其中更新的文件（即，修订）和文件夹可从通道的其它成员和 / 或 P2P 网络的其它同级中检索。应注意，存在可在这个

阶段期间实施、在 P2P 网络中搜索并检索数据的众所周知的方法,例如参见 Buford, Yu and Lua (2009, pp. 163-181, 183-202) 获得更多详情。

[0257] 通过比较并组合来自所接收更新、更新的修订历史日志和可用更新列表的数据,成员在其修订、文件和文件夹中获得关于缺失修订的信息。根据特定实施方案,在阶段 550 中,确定应检索可用更新列表内的哪些更新。文件、修订、文件部分和文件夹可基于可例如在通道配置单元 235 中配置的预定义策略检索。网络修订控制单元 210 可被配置来例如从通道或仅从新修订的子组中检索所有新修订,所述新修订可被标准仅限于例如最近修订、或在预定义创建日期后修改的修订、或由通道的一个或多个特定成员修改的修订。

[0258] 根据特定实施方案,在下一阶段 560 中,确定从通道中的哪些成员中检索更新。网络修订控制单元 210 可被配置来与通道内的其它成员通信,以在针对更新的修订、文件和文件夹请求应定址的通道内选择成员。成员尤其可取决于覆盖通信偏好和相关文件部分的可用性而选择。成员还可基于请求更新的发端成员与覆盖中的其它成员之间的其它覆盖专用通信而选择,以确定合适成员列表来发送修订、文件、文件夹或文件部分。例如,如果在覆盖中使用 DHT,那么 DHT 专用通信可涉及确定相关修订、文件、文件夹或文件部分的可用性。通道的不同成员可由用于更新不同文件、不同文件夹或不同参考区域的相同单个成员定址。根据特定实施方案,单个文件的不同文件部分可从通道的不同成员中检索。在一些实施方案中,检索更新的修订可仅限于相同通道的成员。在其它实施方案中,非作为与发端成员相同的通道的成员的同级也可用于检索更新修订。

[0259] 在阶段 570 中,从所选成员或同级中检索所请求的更新文件、修订、文件部分和/或文件夹。根据特定实施方案,请求更新文件、修订、文件部分或文件夹的成员可继续并定位具有相关更新的另外成员。例如,当成员离开并加入通道覆盖时,或在实施于覆盖中的 DHT 已从更新询问在覆盖中传输时开始得到更新的情况下,这是有用的。如果修订条目包括例如对应于文件的特定修改部分的部分哈希码,那么这些部分哈希码可用于定位并检索来自内容源(例如,通道中的其它成员)的特定文件部分。

[0260] 根据特定实施方案,更新文件、修订和文件夹的程序可继续直到已被指定用于检索的所有文件、修订和文件夹被成功检索。所述更新程序可对明确用户请求、策略配置(例如,在通道配置单元 235 内配置并由通道管理器 230 控制)作出响应,和/或通过请求其停止的另一程序而在完成前终止。所述更新程序也可由覆盖维持考虑(举例来说诸如覆盖带宽超负荷、包含请求的文件内容部分的成员不足),和/或由节点的局部考虑(诸如局部储存库上的存储空间不足)等等终止。根据其它实施方案,文件的检索可完全由用户控制,其中仅从通道的其它成员中检索由用户明确请求的新修订。根据特定实施方案,除更新的修订历史日志以外,返回的更新响应或更新消息还可包括更新的文件、修订和文件夹自身。因此,修订历史日志更新以及文件和文件夹更新可以单个操作完成。

[0261] 所检索的文件、修订和文件夹存储在成员的数据储存库中。根据特定实施方案,对应于当前修订的文件和文件夹存储在数据储存库 125 中以由文件系统监控器 120 监控。对应于过去修订的其它所检索文件、修订和文件夹可存储在过去修订储存库 135 中。

[0262] 根据特定实施方案,在成功检索修订或文件修订的一部分后,其例如通过计算其内容哈希码以及将其与更新的修订历史日志中的对应修订条目的内容哈希进行比较而检查。当成功检索一个或多个修订时,所接收修订被置于其指定的储存库中。如果过去修订已

被检索,那么其可例如通过与修订引擎 110 互动的网络修订控制单元 210 存储在过去修订储存库内,所述修订引擎 110 被配置来将修订存储在数据储存库 125 或 135 内,并且将关于修订历史日志的修订元数据和 / 或修订位置存储在数据储存库 130 内。如果当前修订(文件)成功被检索,那么其可存储在被监控的文件系统中(例如,在参考区域中),例如通过将文件和文件夹直接置于储存库 125 中或或者通过网络控制单元 210 例如使用替换命令将关于修订历史日志的修订内容和 / 或修订元数据和 / 或修订位置传送到修订引擎 110。

[0263] 根据特定实施方案,一旦一个或多个修订被检索,其对应的修订条目被创建并链接到修订历史日志中的适当位置(如果未在先前阶段中链接)。如果新当前修订存储在参考区域内,那么对应的修订条目被创建并链接到修订历史日志(如果未在先前阶段中链接)。根据特定实施方案,如果在同级上还无法获得新修订的修订条目,那么(例如,通过修订管理器 115)创建修订条目。链接的修订条目包含关于对应修订的相关元数据。

[0264] 在程序 500 完成后,更新的修订历史日志与所接收修订一致。在程序结束时,所更新的修订历史日志包括关于同级内的现有修订的信息以及可能还包括关于存在于其它同级和成员上的修订的信息。换句话说,总之,通过接收消息更新和接收更新响应,系统 205 为成员提供关于发生在其它成员上的修改的信息,以及潜在地还提供修改和更新。这种信息促成系统作为分布式修订控制系统,其中同级可了解发生在其它成员上的修改,因此如上文详细论述,为同级提供其它成员的关于通道中被监控文件和文件夹的活动情况。

[0265] 图 6A 到图 6P 示意图示了根据本发明的一个实施方案的出于展示修订控制系统运行目的而显示的参考区域和修订历史日志。参考图 6A 到图 6P 示例的操作包括:在单个节点(装置)背景下针对修改事件执行的操作;和在文件共享环境背景下执行的操作,其中两个或更多个节点(或装置)通信并且能够共享内容。这些文件共享环境包括例如,两个或更多个直接连接的装置(例如,经由 USB 线直接连接到个人计算机的智能手机)、经由网络连接的两个或更多个节点(例如,在由活动目录定义的相同逻辑网络上)或经由通信网络连接的两个或更多个同级(例如,点对点网络中的同级)。应注意虽然参考下文实施例描述的一些详情本质上可能是特定的,但是下列实施例是出于说明的目的,并且不应以任何方式解释为限制性。

[0266] 针对下列非限制性实施例,修订控制系统 100(或 205)可被图示为实施于装置 P1 上。装置 P1 可为例如个人计算机,其由一个或多个中央处理单元(CPU)、总线、内存(RAM)、硬盘驱动、包括键盘和鼠标的用户界面 150、包括图形 LCD 屏幕的显示器 160、一个或多个通用串行总线(USB)端口,和网络界面卡(NIC)。装置 P1 还运行操作系统,并具有其硬盘驱动,所述硬盘驱动被配置来通过运行其上的文件系统而运行为逻辑文件系统“C:”,所述文件系统包含文件夹“C:\Users\Laron\Patent”L1,包含三个文件“US61-286093.txt”(F1)、“Comments”(F2)。根据当前实施例,最初修订控制系统不运行,例如关闭、关机或另外未启动。

[0267] 操作节点 P1 的用户可指示节点启动修订控制系统 100(或 205)。所述用户在用户界面 150 的帮助下(例如,经由策略单元 140)定义修订控制系统的参考区域 RA1-1。图 6A 是参考区域 RA1-1 的示意图,其显示所述参考区域的所有文件和文件夹。

[0268] 修订控制系统 100(或 205)作出响应而创建修订历史日志 RHL1-1 并且在数据存储器上分配过去修订储存库 135。一旦参考区域被定义,修订引擎 110 立刻开始追踪针对存

储在参考区域 RA1-1 内的文件和文件夹（在这个情况下为文件夹 L1）执行的修改事件。如上所述，关于修改事件的信息可通过修订引擎 110 从操作系统、节点 P1 的文件系统或例如通过用户输入 150 与修订控制系统互动的用户中获得。

[0269] 在修订引擎 110 运行开始时，修订引擎 110 在参考区域 RA1-1 上执行扫描，其中搜索标准设置（例如，在策略单元 140 中定义）来搜索所述参考区域中的所有文件或文件夹，以标识关于参考区域 RA1-1 中的现有修订的修改事件。但是，自这个阶段以后，修订历史日志 RHL1-1 中无表示参考区域 RA1-1 中的文件和文件夹的现有修订条目，在参考区域 RA1-1 中所标识的每个文件或文件夹触发修改事件。由于修订历史日志 RHL1-1 中无关于参考区域 RA1-1 中的文件或文件夹的现有信息，所以被标识的文件或文件夹可以触发“创建”修改事件。

[0270] 对于在扫描期间标识的每个修改事件，系统通过下列程序 400 创建初始修订条目。

[0271] 对每个修改事件作出响应，通过上文参考图 4 描述的下列程序 400 针对对应的文件或文件夹创建第一（或“最早”）修订条目。下列阶段是参考文件 F1 图示的程序 400 的一个实施例：

[0272] 在阶段 410，修订控制系统标识关于文件 F1 的修改事件已发生。

[0273] 在阶段 420，修订引擎 110 获得关于 F1 的元数据和 / 或内容。为获得元数据，修订控制系统从存储文件 F1 的文件系统中提取现有元数据字段。另外，针对 F1 计算出内容哈希码 HC1（例如，SHA1 哈希码）。内容哈希码 HC1 被添加到所获得的元数据，并且接着针对在这个阶段中获得的所有元数据计算出元数据哈希码 MH1。在这个实施例中，唯一标识符 ID1（例如，UUID）也在这个阶段中生成。或者或此外，如上文所说明，文件 F1 的一组适当的元数据字段可用作基本上唯一的标识符。

[0274] 在阶段 430，修订引擎发现在修订历史日志中无相关修订条目。接下来在阶段 440 中，针对 F1 创建新修订条目 RE1。在阶段 430 中获得的元数据被添加到这个修订条目。

[0275] 在阶段 450，修订控制系统已指示不存在先前修订条目，并且因此新修订条目作为关于 F1 的第一修订条目在修订历史日志中相关。

[0276] 在阶段 460，修订引擎重复参考区域 RA1-1 中的每个另外文件（在这个实施例中为文件 F2）的程序，创建新修订条目 RE2，直到未标识其它文件修改事件，并且程序 400 终止。

[0277] 文件夹 L1 自身在修订历史日志中无对应修订条目，因此程序 400 将对仅仅通过在由所监控的参考区域中存在文件夹 L1 所触发的修改事件 ME0 作出响应而实行。当处理文件夹 L1 时，修订控制系统获得元数据，指示这个修改事件涉及文件夹而非文件；创建对应修订条目 RE0（阶段 440）并且用关于文件夹 L1 的信息（元数据）填充修订条目 RE0 并使修订条目 RE0 在修订历史日志 RHL1-1 内相关（阶段 450）。

[0278] 在这个实施例中，修订控制系统计算出包括如从文件系统中获得的描述文件夹自身的元数据以及文件夹所包含的文件 F1 和 F2 的元数据的文件夹 L1 的元数据哈希码 MH0。为促成这种计算，可在包含在 L1 的文件（F1 和 F2）已处理后处理所述计算。

[0279] 接下来，操作节点 P1 的用户在文件夹 L1 内创建名为“Correspondence”的新文件夹。这个动作再次触发由文件系统监控器 120 捕获的修改事件 ME4。修改事件 ME4 经历根据程序 400 的处理。如上文参考文件夹 L1 所述，修订控制系统获得指示这个文件修改事

件涉及文件夹而非文件的元数据、创建对应的修订条目 RE4(阶段 440) 并且用关于文件夹 L2(如上文所述关于文件夹 L2 的处理)的信息(元数据)填充修订条目 RE4 并使修订条目 RE4 在修订历史日志 RHL1-1 内相关(阶段 450)。

[0280] 用户还在文件夹 L2 内创建名为“Comments for Yossi”的新文件 F4。再次,这个动作触发由文件系统监控器 120 捕获的修改事件 ME5。文件 F4 经历如上文针对文件 F1 到 F2 所示例的程序 400 的连续阶段,使得新修订条目 RE5 在关于 F4 的修订历史日志中相关。图 6B 图示了到目前为止,在完成所有程序后修订历史日志 RHL1-1 的状态。在下列描述以及图 6B、6C、6D 和 6F 中,‘CH’指内容哈希码,‘MH’指元数据哈希码,ID 指条目标识符,并且‘元数据’指任何其它元数据和修订条目的信息。

[0281] 在一些实施例中,对于针对上文提及的修改事件处理的每个文件,修订控制系统存储修订,包括文件的相关过去修订储存库 135 中的文件副本;并指示过去修订储存库中的文件在其对应修订条目中的位置。

[0282] 继续本实施例,考虑操作装置 P1 的用户添加新内容到文件 F4。文件系统监控器 120 标识内容添加为修改事件 ME6。修改事件 ME6 的处理根据修订策略单元 140 中针对参考区域 RA1-1 定义的运行规则和配置开始。在这个实施例中,修改事件 ME6 的处理不会在其被标识后马上开始,而是根据预定义时间表执行。一旦其被确定,便是处理文件 F4 之时。修改事件 ME6 根据程序 400 的阶段处理:

[0283] 在阶段 410,修订引擎 110 标识针对文件 F4 的文件修改事件 ME6 已发生。在阶段 420,修订引擎 110 获得关于 F4 的元数据和内容、计算出 F4 的内容哈希码 HC5、添加内容哈希码 HC4 到所获得的元数据,并且接着计算出所有所获得的元数据的元数据哈希码 MHC6。

[0284] 在阶段 430,修订引擎 110(在修订管理器 115 的帮助下)从修订历史日志中检索对应于文件 F4(针对该文件 F4 实施修改事件 ME6)的相关修订条目 RE5。接下来,在阶段 440,针对 F4 创建新修订条目 RE6。在阶段 420 获得的元数据被添加到这个修订条目。唯一标识符 ID6 也被添加到所述条目。

[0285] 在阶段 450,修订引擎 110 链接新修订条目 RE6 到先前修订条目 RE5,从而延伸修订历史日志 RHL1-1 内表示文件 F4 的修订进程的进程路径。图 6C 是示出了在修改事件 ME6 的处理完成后修订历史日志 RHL1-1 的状态的示意图。

[0286] 假设节点 P1 上的用户已继续修改参考区域 RA1-1 上的文件,对这些修改作出响应,其继续出发修改事件。修订控制系统 100 作出响应以类似于上文参考文件 F4 描述的程序的方式创建其它修订条目。为简洁起见,实施例现回到图 6D 中所示的一系列修改事件的所得情况。

[0287] 在上述修改事件后,操作装置 P1 的用户有兴趣查看文件 F4 的历史,可能以与目前为止已经积累在文件的进程路径中的不同修订互动(例如,检查)。

[0288] 用户可通过利用用户界面 150 和显示器 160 与修订引擎 110 互动以命令修订引擎执行下列动作中的一个或多个:选择参考区域 RA1-1;查看在对应修订历史日志 RHL1-1 中具有修订条目的文件和文件夹列表;从所述文件和文件夹列表中选择文件 F4;查看 F4 的历史。对最后命令作出响应,修订控制系统 100 可例如在显示器 160 上显示所请求的信息。所请求的信息可以各种格式显示给用户。例如,所述信息可显示为有向图,诸如在图 6D 中显示的有向图,始于 RE5 并且形成从 RE5 延伸通过 RE6 到 RE9 的进程路径。



[0289] 通过使鼠标指向并点击鼠标或通过使用键盘命令和 / 或快捷方式,操作节点 P1 的用户还能够选择特定修订并且命令修订引擎 110 针对所选修订执行动作。

[0290] 接下来,在当前实施例中,操作装置 P1 的用户可能想要恢复(返回)到文件 F4 的一个过去修订,例如在用户感觉文件的过去修订比当前修订好的情况下,或者在用户想要在显示器 160 上并列查看这个过去修订以进行比较、检查或任何其它目的的情况下。根据当前实施例,为了命令系统恢复到过去修订,用户使用如上文所说明的用户界面装置 150 并且选择他想要恢复的特定修订。在这个实施例中,用户选择在修订历史日志 RHL1-1 中由修订条目 RE5 表示的修订,所述修订为目前记录在修订历史日志 RHL1-1 中的文件 F4 的第一(或最早)修订。用户接着发出指示修订引擎 110 恢复到这个修订的恢复命令。

[0291] 在本文所示的特定实施例中,对恢复命令作出响应,修订管理器 115 标识与 RE5 相关的修订标识为文件 F4 的当前修订并修改存储在参考区域上的文件 F4 的内容和元数据以匹配对应于目前存储在过去修订储存库中的修订条目 RE5 的修订内容,并且匹配对应于修订条目 RE5 的修订的元数据。至少部分所述元数据存储于修订条目 RE5 中。

[0292] 现在假设用户修改文件 F4,并且对新修改事件 ME7 被标识作出响应。在这种情况下,与文件 F4 的当前修订相关的相关修订条目是 RE5。修改事件 ME7 的处理(根据程序 400)开始,期间由修订引擎 110(在修订管理器 115 的帮助下)创建表示文件 F4 的新修订的新修订条目 RE12。在最后修订事件之前,修订条目 RE12 链接到表示文件 F4 的修订条目 RE5,从而针对 F4 在修订历史日志中创建新进程路径,所述新进程路径从 RE5 延伸到 RE12。

[0293] 在这个阶段,在修订历史日志 RHL1-1 中存在针对文件 F4 延伸的两条进程路径,第一条进程路径包含 {RE5、RE6、RE9},并且第二条进程路径包含 {RE5、RE12}。

[0294] 由于文件 F4 被修改,所以其包含的文件夹 L2 的元数据(诸如最后修改日期和时间以及最后修改用户 ID)也已经被修改,并且因此包含文件夹 L2 的文件夹 L1 的元数据也可改变以反映文件夹 L2 的改变。在文件系统上的这些修改分别导致修改事件 ME7-1 和 ME7-2。程序 400 首先开始进行修改事件 ME7-1 以创建对应于文件夹 L2 的修改的修订条目 RE121,并且在完成这个程序后,接下来程序 400 开始进行修改事件 ME7-2 以创建对应于文件夹 L1 的修改的修订条目 RE122。

[0295] 用户接下来将文件 F4 重新命名为新文件名“Comments version2”,发出重新命名事件。作出响应,修订引擎 110 标识修改事件 ME8,其描绘具有相关修订条目 RE12 的文件 F4 的元数据字段‘文件名’已经被修改为新值。再次执行程序 400 以处理修改事件 ME8 并导致创建修订条目 RE13 并使其与 RE12 相关、延伸现在包含 {RE5、RE12、RE13} 的进程路径。图 6E 和图 6F 是显示在这些程序结束时参考区域 RA1-1 和修订历史日志 RHL1-1 的状态的示意图。

[0296] 现在继续参考图 6G 到图 6P,其示例在文件共享环境背景下由修订控制系统 205 执行的一些操作。下列实施例继续先前实施例并且显示存储在单独装置上的不同修订历史日志如何通过装置之间共享信息以及更新每个装置中的修订历史日志还有可能的对应修订而维持有类似内容。本文所示的实施例指文件共享的两种可能方案。在第一个方案中,装置 P1 经由直接通信设施(例如,经由 USB 通信线)与第二装置 P2 通信。在第二个方案中,装置 P1 在点对点网络上与装置 P3 通信。装置 P1 和 P2 都包括尤其能够在装置之间通信的修订控制系统 205。

[0297] 假设操作装置 P1 的相同用户也操作第二装置 P2, 所述第二装置 P2 具有实施于其上的修订控制系统 RCS2。装置 P2 可例如是具有实施于其上的修订控制系统 RCS2 的智能手机、USB 通信组件、至少一个中央处理单元 (CPU)、足够的内存 (RAM)、显示器 160 (诸如内置 LCD 显示器)、和用户界面 150 (诸如 QWERTY 型键盘、和组织为文件系统的可访问存储库)。在这个实施例中, 在修订控制系统 RCS2 上的通道配置单元 235 预先配置有运行规则, 所述运行规则包括每当其连接到另一装置时起始更新询问的规则。

[0298] 用户接着用 USB 线将装置 P2 连接到装置 P1, 并且如上文参考程序 500 详细论述, 在适当运行所述两个装置以经由 USB 建立通信后, 装置 P1 上的修订控制系统 RCS1 和装置 P2 上的修订控制系统 RCS2 可交换信息。

[0299] 用户可使用其用户界面 150 在修订控制系统 RCS1 上定义通道 C1, 通道 C1 包括参考区域 RA1-1。接着装置 P1 的用户与修订控制系统 RCS1 互动以发送邀请到装置 P2 以预定至通道 C1。在一些情况下, 通道 C1 可被自动指派预置默认策略配置。

[0300] 修订控制系统 RCS2 的通道管理器 230 接收邀请并且在其显示器 160 上将所述邀请显示给用户, 使用户能选择接受或拒绝所述邀请。在当前实施例中, 用户接受所述邀请并且作出响应, 修订控制系统 RCS2 指定存储空间作为与通道 C1 相关的装置 P2 上的参考区域 RA1-2、针对参考区域 RA1-2 创建新修订历史日志 RHL1-2, 并定义存储库用于保留过去修订。这时, 装置 P1 和 P2 都是通道 C1 的成员。

[0301] 在成功完成上文提及的操作后, 现在 RCS2 起始上文参考图 5 描述的更新周期。在更新周期程序期间, 修订历史日志 RHL1-2 用修订条目更新, 并且参考区域 RA1-2 和过去修订存储库 135 可填充有从装置 P1 中检索的修订。

[0302] 在阶段 510, 修订控制系统 RCS2 起始更新周期。在这个实施例中, (在阶段 520, 例如在修订控制单元 210 的帮助下) 创建更新周期, 并且定义初始搜索标准。如上文参考图 5 所说明, 所述标准定义更新询问的条件。在这个实施例中, 对‘所有现有实体’设置标准, 所述标准定义在通道 C1 的其它成员上的任何可用修订条目和修订将被检索。接着更新询问经由通道管理器 230 发出并且仅被传播到这个通道的另一成员, 装置 P1。

[0303] 装置 P1 经由其通道管理器 230 接收来自装置 P2 的更新询问、从所述更新询问中读取标准, 并 (根据所接收标准的说明) 检查任何相关更新是否存在于其修订历史日志 RHL1-1 中。装置 P1 接着创建响应更新。由于标准被定义为‘所有现有实体’, 所以装置 P1 发现修订历史日志 RHL1-1 中的所有修订条目符合标准、用关于所有相关修订条目的生成响应更新, 并且将所述响应更新发回给询问成员, 装置 P2。

[0304] 装置 P2 现在经由通道管理器 230 从装置 P1 接收响应更新 (阶段 530)。由于装置 P2 是响应更新的返回路径中的最后成员 (通道 C1 仅包括 2 个成员), 所以装置 P2 不进一步传播这个响应更新给其它装置。装置 P2 上的网络修订控制单元 210 被配置来在文件修订引擎 110 的帮助下比较其自身修订历史日志 RHL1-2 与在所接收响应更新中的修订条目。由于在本实施例的这个阶段修订历史日志 RHL1-2 中不存在其它修订条目, 所以所接收修订条目中的每一个用于更新装置 P2 上的可用更新列表。

[0305] 装置 P2 上的网络修订控制单元 210 现在继续运行其相关文件修订引擎 110 以针对在可用更新列表中所列出的每个修订条目在其修订历史日志 RHL1-2 中创建修订条目, 并且如包含在接收自装置 P1 的修订条目内的信息所指示链接其进程路径中的所创建的修

订条目。

[0306] 在准备好可用更新列表的情况下,装置 P2 现在继续阶段 540,所述阶段 540 包括阶段 550、560 和 570。在阶段 550,确定所需更新列表(选自所列出的可用更新),在当前实施例中,其包括所有更新,因为在修订历史日志 RHL1-2 中不存在修订。

[0307] 在阶段 560,装置 P2 通过咨询响应更新以及装置的通道和预定管理器 230 而确定哪些成员具有所需更新。在当前实施例中,仅具有所需更新的成员是成员 P1。因此,在阶段 570,装置 P2 上的网络修订控制单元 210 与具有所需更新的成员 P1 通信,并且请求检索所需更新。所需更新包括修订、文件和文件夹。

[0308] 对于在阶段 570 从 P1 中检索的每个更新,修订控制系统 RCS2 将检索的文件和文件夹存储在数据储存库 125 上以由成员 P2 的文件系统监控器 120 监控文件,并将所检索的修订存储在过去修订储存库 135 中。

[0309] 图 6G 和图 6H 是分别显示在完成根据图 5 的程序后修订历史日志 RHL1-1 和 RHL1-2 的状态的示意图。如图 6G 和图 6H 中所示,装置 P1 上的 RHL1-1 和装置 P2 上的 RHL1-2 两者基本上相同。此外,两个装置可在其参考区域上以及其数据储存库中包括基本上相同的文件,以及基本上相同的文件夹。

[0310] 继续所述实施例,假设用户修改对应于装置 P2 上的修订条目 RE13 的文件 F4 的修订,而装置 P2 连接到装置 P1。修订控制系统 RCS2 标识文件 F4 的这个修改为文件修改事件 ME9,并且如上所述作出响应在修订历史日志 RHL1-2 上创建新修订条目 RE14。如上文参考图 5 中的阶段 520 和 530 所论述,成员可利用其通道管理器 230 来发送更新消息给通道的其它成员。在这个实施例中,成员 P2 在修订条目 RE14 已创建后创建更新消息,所述更新消息包括来自 RE14 的信息,诸如被修订的修订的 ID 和描述修订的元数据、针对所述修订计算出来的哈希码等等。

[0311] 装置 P1 上的修订控制系统 RCS1 接收更新消息,并且如上所述在阶段 530 开始更新周期,用来自更新消息的信息更新 RCS1 上的可用更新列表,以及在修订历史日志 RHL1-1 上创建修订条目 RE14,所述修订条目 RE14 包括来自更新消息的关于 F4 的信息,以及这个修订条目在参考区域 RA1-1 中和在过去修订储存库 135 中都没有对应修订的指示。此外,对更新消息作出响应,装置 P1 现在可在显示器 160 上显示指示通道 C1 上的被监控的文件已经在装置 P2 上被修改的消息。在这些操作后修订历史日志 RHL1-1 和 RHL1-2 的状态分别图示在图 6I 和图 6J 中。

[0312] 装置 P1 上的修订控制系统 RCS1 接下来可前行到阶段 540,并且继续如我们已在上文所述检索实际可用修订。然而,为了当前实施例,假设这尚未发生,并且对应于 RE14 的实际修订在装置 P1 上从未更新,并且 RCS1 仅知道关于其存在的信息。

[0313] 接下来,用户通过将连接装置 P2 和装置 P1 的 USB 线移除而将装置 P2 从装置 P1 上断开。在将所述装置断开后,用户在装置 P2 上修改由在修订历史日志 RHL1-2 中的修订条目 RE13 表示的文件 F4 的修订。这导致修改事件 ME10 并触发程序 400,所述程序 400 导致由基本上唯一的 ID ID15 标识的新修订条目 RE15。RE15 链接到修订历史日志 RHL1-2 中的 RE14。用户接着进一步修改文件,再次导致修改事件 ME11,其触发程序 400,所述程序 400 再次导致由基本上唯一的 ID(至少在通道背景下)ID16 标识的新修订条目 RE16。RE16 链接到修订历史日志 RHL1-2 中的 RE15。

[0314] 这个实施例的下一部分涉及实施覆盖管理功能的点对点网络背景下的文件共享方案。在这个实施例中,点对点网络的特征在于无结构化通道覆盖几何形,所述网络包括所有可经由互联网链接而彼此访问的上述装置 P1、装置 P3 和 P4 和在线状态服务 PS1。装置 P1、P3 和 P4 都尤其包括修订控制系统 205,所述修订控制系统 205 实现在文件共享环境中的通信。装置 P3 具有实施于其上的修订控制系统 RCS3 并且包括 USB 通信组件、至少一个中央处理单元 (CPU)、足够的内存 (RAM)、显示器 160 (诸如内置 LCD 显示器)、和用户界面 150 (诸如 QWERTY 型键盘、和组织为文件系统的可访问存储库)。修订控制系统 RCS3 上的通道配置单元 235 预先配置有运行规则,所述运行规则包括每当其连接到另一装置时起始更新询问的规则。装置 P4 的配置类似于装置 P3 并且包括修订控制系统 RCS4。

[0315] 在这个实施例开始时,装置 P3 和 P4 在线并且独立地与在线状态服务 PS1 通信。同时,装置 P1 尝试与在线状态服务 PS1 通信。在这个实施例中,在尝试通信之前,装置 P1 上 RCS1 生成将在与通道中的其它成员的未来通信中使用的两个加密密钥 (例如,使用众所周知的 RSA 方法,来生成私人密钥和对应的公开密钥)。

[0316] 用户将装置 P1 连接到互联网,并指示装置 P1 上的修订控制系统 205 经由互联网连接到在线状态服务 PS1,并且试图通过对在线状态服务 PS1 供应用户名和密码以及所生成的公开密钥而登录在线状态服务 PS1。

[0317] 如上文参考图 2 所述,在线状态服务 PS1 促进针对多个同级的同级身份管理。在这个实施例中,在线状态服务 PS1 针对每个成员将用户名和密码对存储在其用户数据库 260 中。由装置 P1 供应的密码和用户名由在线状态服务 PS1 对照存储在用户数据库 260 中的信息进行认证,并且在成功认证密码和用户名之后,在线状态服务 PS1 将标识互联网上的装置 P1 的信息 (诸如 IP 地址、NAT 横越端口等等) 存储在在线数据库 265 中,从而将装置 P1 添加到目前在线同级列表中。

[0318] 装置 P1 上的用户使用用户界面 150 与修订控制系统 205 互动来邀请其它成员到通道 C1。在这个实施例中,用户打出用户名列表、邮件地址、或标识在线状态服务 PS1 上其它成员的其它数据。在其它实施例中,用户可使用其它方法来指示他想要邀请哪些其它成员到通道 C1,例如从以图形显示在显示器 160 上的用户列表中选择用户、从显示在显示器 160 上的地址簿中选择等等。

[0319] 一旦准备好成员列表,用户立刻指示修订控制系统 205 发送邀请给所选的潜在成员。装置 P1 继续将关于通道 C1 的邀请存储在通道配置单元 235 上以用于后续操作。

[0320] 为获得这些潜在成员的联系信息 (其是将邀请传达到正确位置所需的),通道管理器 230 与在线状态服务 PS1 通信以请求追踪目前在线的同级列表的服务。作出响应,在线状态服务 PS1 向其用户数据库 260 和其在线数据库 265 询问有关由来自通道管理器 230 的请求所指示的同级的信息。相关信息从存储服务 PS1 传回到装置 P1 上的通道管理器 230。

[0321] 在这个实施例中,假设检索自存储服务 PS1 的信息涉及下列同级:装置 P3 和装置 P4,以及针对装置 P3、P4 中每个装置传达的信息包括其 IP 地址、其 TCP/IP 端口、其 NAT 地址、其公开加密密钥和可能的其它字段。

[0322] 修订控制系统 RCS1 接着与在线状态服务 PS1 通信以向注册数据库 270 注册通道 C1,以维持对目前离线并且无法立即接收邀请的同级的主动邀请。一旦成员重新连接,在线状态服务 PS1 也立刻维持成员的注册状态 (通道中的成员资格)。

[0323] 修订控制系统 RCS1 使用针对同级（即，潜在成员）而检索自在线状态服务 PS1 的信息来生成对每个潜在成员的邀请，所述邀请包含关于通道的信息；并且通过负责建立与其它同级的通信以及与其它同级互动的 P2P 引擎 240 而将邀请传达给对应的潜在成员中的每个成员。在一些情况下，每个邀请被加密（例如，用针对每个被邀请同级所计算出来的公开密钥）。

[0324] 接下来，在这个实施例中，潜在成员 P3 和 P4 中的每个成员接收邀请，并且执行一组活动以能够交换通道上的更新，并有效地变成通道 C1 的成员。装置 P3 上的修订控制系统 RCS3 在其显示器 160 上显示所接收邀请给用户，使用户能选择接受或拒绝邀请。在当前实施例中，用户接受邀请并且作出响应，修订控制系统 RCS3 指定存储空间为与通道 C1 相关的装置 P3 上的参考区域 RA1-3、针对参考区域 RA1-3 创建新修订历史日志 RHL1-3（如当前实施例中所示，假设对应修订树不存在），并指示用于保存过去修订的存储库。装置 P3 现准备好作为通道 C1 的成员进行通信，并且向在线状态服务 PS1 传达其已接受通道 C1 的邀请。在线状态服务 PS1 验证这一邀请存在于其注册数据库 235 并且将装置 P3 注册为通道 C1 成员。

[0325] 装置 P4 上的修订控制系统 RCS4 也接收邀请，并且如上述段落中所述继续指定存储空间为与通道 C1 相关的参考区域 RA1-4、创建新修订历史日志 RHL1-4 并指示用于保存过去修订的存储空间。接着 RCS4 向在线状态服务 PS1 传达其已经接受邀请，并且在在线状态服务 PS1 验证这一邀请存在后，其也将装置 P4 注册为通道 C1 的成员。此时，装置 P1、P3 和 P4 是通道 C1 的所有成员并且可在其间的通道覆盖中相连，然而，其间仍尚未建立连接或交换的通信。

[0326] 为了使这个实施例清楚起见，假设在所有三个装置 P1、P3 和 P4 上的所有通道管理器 230 已同等确定在传播消息和发送询问时其各自装置应与通道 C1 的其它两个成员连接。应了解在一些实施例中，通道中的每个装置可选择性地连接通道中不同数量的成员。在这种情况下，每个装置上的通道管理器 230 可向其各自的 P2P 引擎 240 咨询以确定最优、最大或另一数量的所需连接及其特性。

[0327] 一旦装置 P1 获得作为通道 C1 的成员的其它装置的联系信息，其立即可继续与已由在线状态服务 PS1 指示为在线的装置 P3 和 P4 建立连接。装置 P1 可试图连接到装置 P3 和 P4，其继而可与在线状态服务 PS1 通信以出于安全原因验证装置 P1 确实是通道 C1 的成员。在成功验证装置 P1 的成员资格后，装置 P3 和 P4 两者都可连接到装置 P1。

[0328] 在这个实施例中，装置 P3 和 P4 中的每个装置已确定其需要两个连接，但目前仅装置 P1 连接到所述两个装置 P3 和 P4，同时装置 P3 和 P4 中的每个装置仍仅连接到 P1。通过与在线状态服务 PS1 通信请求通道 C1 的连接成员列表，装置 P3 可标识装置 P4 是可能的连接，并且装置 P4 可标识装置 P3 是可能的连接。在从在线状态服务 PS1 成功接收响应后，装置 P3 连接到装置 P4。竞争状态可能发生，其中两个装置 P3 和 P4 试图连接到另一装置，并且其中一个成功，而另一个失败。无论哪种方式，都形成连接，并且所述两个装置连接，实现由其各自 P2P 引擎 240 定义的与两个成员连接的请求。

[0329] 每个装置的连接状态基于其能力、其连接带宽容量、P2P 覆盖要求等等确定。在这个实施例中，装置可保持断开直到实际连接请求发出，但在其它 P2P 覆盖几何形诸如 DHT 中，连接可建立，只要装置连接到覆盖以辅助管理 DHT 的活动，例如对实体 ID 和位置的询问

作出响应。在这个实施例中,通道覆盖如下:

[0330] 装置 P1 可经由互联网连接到装置 P3 和 P4,并且可与这些装置通信。此外,装置 P1 也可按需连接在线状态服务 PS1 并与在线状态服务 PS1 通信。

[0331] 装置 P3 可经由互联网连接到装置 P1 和 P4,并且可与这些装置通信。此外,装置 P3 也可按需连接在线状态服务 PS1 并与在线状态服务 PS1 通信。

[0332] 装置 P4 可经由互联网连接到装置 P1 和 P3,并且可与这些装置通信。此外,装置 P4 也可按需连接在线状态服务 PS1 并与在线状态服务 PS1 通信。

[0333] 为清楚起见,在这个实施例中针对由所有参与的装置 P1、P3 和 P4 上的针对通道 C1 通信的各自通道管理器 230 和 P2P 引擎 240 维持的通道覆盖使用无结构点对点几何形。应了解本文中所述的使用“延伸环”运算法则的通信传播可替代地使用 DHT 查找表实施,其中执行所需调整。

[0334] 针对当前实施例,假定每个装置上的通道配置单元 235 被设置为每小时发出拉式更新询问并且每小时还针对可用更新将信息(例如,经由推式更新消息)传达给其它成员。此外,在 P1、P3 和 P4 上的所有通道配置 235 用将到其它成员的更新询问传播限于预定义数量的规则设置。例如,更新询问中的计数器(本文中称为“正向计数器”)可用一个值设置,并且每当更新周期被传播到一个成员时这个值被累加 1。一旦计数器等于 0,更新询问传播立刻终止。

[0335] 一旦 RCS3 已与至少一个其它装置连接,(在这个实施例中,与装置 P1 连接),RCS3 现在可立刻起始上文参考图 5 描述的更新周期,以用通道 C1 中的其它成员的修订条目更新其修订历史日志 RHL1-3。

[0336] 在阶段 510,修订控制系统 RCS3 起始更新周期。在这个实施例中,创建更新询问(在阶段 520),并且更新询问中的初始搜索标准被定义为促进‘延伸环’覆盖搜索运算法则,如下:定义搜索修订条目的条件的标准被设置为‘所有现有实体’,并且正向计数器被赋予一个值 5。接着更新询问由装置 P3(在通道管理器 230 的帮助下)发出并且被传播到所连接的装置 P1。

[0337] 装置 P1 上的 RCS1(经由其通道管理器 230)从装置 P3 接收更新询问、从更新询问中读取标准,并且在修订管理器 115 的帮助下通过定位满足标准的修订条目而检索存在于其修订历史日志 RHL1-1 中的所有相关更新(注意所述标准被定义为‘所有现有实体’)。装置 P1 上的 RCS1 向 P2P 引擎 240 咨询以了解所用几何形(无结构)是否需要转送或重新发送更新询问到其它同级。此时,P2P 引擎 240 指示正向计数器大于 0,但在这个阶段,装置 P3 是唯一一个其它连接的同级,并且因此更新询问不再另外传播。

[0338] RCS1 现准备好创建响应更新。RCS1 中的网络修订控制单元 210 用关于所有相关修订条目的信息填充响应更新,并且接着将其发送回询问成员 P3 上的 RCS3。

[0339] 现在 RCS3 在阶段 530 参与响应更新,并且一旦从 RCS1 接收响应更新,装置 P3 上的网络修订控制单元 210 立即被配置来在其相关修订引擎 110 的帮助下比较其自身修订历史日志 RHL1-3 与关于用响应更新接收的修订条目的信息。由于在本实施例的这个阶段在修订历史日志 RHL1-3 中不存在其它修订条目,所以每个所接收修订条目用于更新装置 P3 上的可用更新列表。接下来,文件修订引擎 110(例如,在修订管理器 115 的帮助下)继续针对可用更新列表中列出的每个修订条目在其修订历史日志 RHL1-3 中创建新修订条目,

并且如由包含在修订条目内的信息所指示链接其进程路径中的新修订条目。

[0340] RCS3 现在前进到阶段 540, 阶段 540 包括阶段 550、560 和 570。在阶段 550, 确定所需更新列表, 其在当前实施例中包括所有更新, 因为由 RHL1-3 中的修订条目表示的所有修订缺失。在阶段 560, RCS3 通过咨询响应更新以及装置的通道和预定管理器 230 而确定哪些成员具有所需更新。在当前实施例中, 具有所需更新的唯一成员是成员 P1。因此, 在阶段 570, 装置 P3 上的网络修订控制单元 210 与具有所需更新的成员 P1 上的 RCS1 通信, 并且请求检索所需更新。所需更新包括修订 (将存储于过去修订数据储存库上)、文件 (将存储于参考区域上的修订) 和文件夹。

[0341] 对于在阶段 570 从 RCS1 中检索的每个更新, 检索控制系统 RCS3 将检索的文件存储在数据储存库 125 上以由 RCS3 的文件系统监控器 120 监控文件并且将所检索修订存储在过去修订储存库 135 中。此时, 成员 P1 和 P3 具有基本上相同的修订历史日志, 并且存储基本上相同的修订条目。

[0342] 在从最后发出的更新周期开始过去一个小时之前, 装置 P1 上的用户修改文件 F2 两次 (由 RE2 表示), 触发修改事件 ME12 和 ME13, 分别在 RHL1-1 中产生修订条目 RE17 和 RE18。在这个实施例中, 这些修改事件还未被转送到其它成员, 这是因为如所述, 从由 RCS1 发出的最后更新周期开始还未过去一个小时。如上所述, 每个修订控制系统上的策略被设置为每满一个小时仅发送一个更新消息。装置 P3 上的用户修改文件 F1 三次 (由 RE1 表示), 触发修改事件 ME14、ME15 和 ME16, 产生修订条目 RE19、RE20 和 RE21。装置 P3 上的用户也修改文件 F2 两次 (由 RE2 表示), 其中所实施的修改不同于用户在装置 P1 上所作的修改。实施在文件 F2 上的两个修改触发修改事件 ME17 和 ME18, 分别在 RHL1-3 中产生修订条目 RE22 和 RE23。根据每个所述装置上的配置单元 235 的设置, 不发送更新, 因为自最后更新起未过去一个小时。图 6K 显示 RHL1-1 的状态, 并且图 6L 显示上述修改发生后 RHL1-3 的状态。

[0343] 考虑这时操作装置 P4 的用户经其用户界面 150 与修订控制系统 RCS4 互动以开始更新周期。修订控制系统 RCS4 现在通过制定更新询问 UQ1 开始更新周期。在这个实施例中, 定义询问的标准被设置为仅检索对应于在最后一个小时内修改的当前修订的修订条目 (即, 对应于关于最早文件 (或最早修订) 的最新更新修订的修订条目, 其目前针对给定进程路径可用)。这些修订还未被发送到 RCS4, 因为其被排定为仅在自最后更新开始过去一个小时后发送。应注意在其它实施例中, 更新询问的标准的可被不同地定义。例如, 其可包括指示哪个是关于 RHL1-4 中可用的给定进程路径的最新更新修订条目的信息和针对较新更新的任何一个或多个修订条目的请求。

[0344] RCS4 连接到 RCS3 和 RCS1, 并且向其传达更新询问 UQ1。修订控制系统 RCS1 和 RCS3 两者都接收并开始处理更新询问 UQ1。

[0345] 在 RCS1 上, (例如, 在修订管理器 115 的帮助下) 检索匹配更新询问 UQ1 中的标准的一个修订条目 {RE18}, 并且创建响应更新 RU1 并用关于所述修订条目的信息填充响应更新。接下来, RCS1 (经由 P2P 引擎 240 或通道管理器 230) 标识其连接的成员是 P3 和 P4。RCS1 创建更新询问 UQ2、填充它以包括与 UQ1 相同的信息、将所包括的正向计数器的值累加 1, 并继续传播更新询问 UQ2 到其它连接的成员, 即装置 P3。

[0346] 在 RCS3 上, (例如, 在修订管理器 115 的帮助下) 检索匹配更新询问中的标准的

修订条目 {RE21、RE23}，并且创建响应更新 RU2 并用关于这些修订条目的信息填充所述响应更新。RCS3 接着（经由 P2P 引擎 240 或通道管理器 230）标识其所连接的成员是 P1 和 P4。然而，由于到现在为止其已接收到由 P1 转送给它的更新询问以及来自 P4 的原始更新询问，修订控制系统 RCS3 可标识所有其连接的成员已经在其更新询问中发送相同标准，并且因此所有连接的成员已接收到这个询问，并且因此确定不存在询问应传播的其它成员。

[0347] 修订控制系统 RCS3 将响应更新 RU2 返回到请求相同更新询问的每个成员，即成员 P1 和 P4。包含 {RE21、RE23} 的响应更新被发送到修订控制系统 RCS1 和修订控制系统 RCS4。

[0348] 当修订控制系统 RCS1 从成员 P3 接收响应更新 RU2 时，其（在修订管理器 115 的帮助下）标识修订条目 RE21 和 RE23 从其修订历史日志中缺失。RCS 1 将修订条目 E21 和 RE23 添加到其响应更新 RU1，所述响应更新 RU1 现在包含条目 {RE18、RE21、RE23}。其接着将响应更新 RU1 返回到发起更新询问 UQ1 的成员 P4。此外，由于这些修订条目在其修订历史日志 RHL1-1 中缺失，所以其将其可用更新列表更新为包括 RE21 和 RE23。

[0349] 修订控制系统 RCS4 现在具有两个响应更新 RU1 和 RU2。其标识其缺失修订条目 RE18、RE21 和 RE23，并且因此用修订条目更新可用更新列表。其还创建将置于其修订历史日志 RHL1-4 中的对应修订条目。修订历史日志 RHL1-4 的状态示意图示在图 6M 中。注意新修订条目目前未与修订历史日志 RHL1-4 中的任何进程路径相关。

[0350] 如果缺失的修订条目的先前修订条目在 RHL1-4 内被标识，那么修订引擎 110 可将接收于响应更新中的所接收修订条目 (RE18、RE21 和 RE23) 链接到 RHL1-4 中对应的所标识的先前修订条目。另一方面，如果未在 RHL1-4 中找到一个或多个先前修订，那么修订引擎 110 可发出请求缺失的修订条目的新更新询问。为确定相关先前修订条目是否存在于 RHL1-4 中，可（例如，在修订管理器 115 的帮助下）使 RCS4 中的修订引擎 110 可运行以利用在所接收修订条目找出的、关于其先前修订条目的信息并通过其 ID、通过元数据哈希码或通过具有修订条目并足以标识缺失的修订条目的其它合适标准而在 RHL1-4 中搜索这些先前修订条目。根据当前实施例，修订引擎 110 未能使新修订条目在其进程路径中相关于修订历史日志中，因为 RE18、RE21 和 RE23 的先前修订条目缺失。

[0351] 在这个实施例中，修订控制系统 RCS4 现在通过用设置的搜索标准制定更新询问 UQ3 而起始另一更新询问以（这次）检索 RE18、RE21 和 RE23 的缺失的先前修订条目。如果诸如在本实施例中，从每个所接收修订条目的‘父’数据中得知缺失的先前修订条目的 ID，那么更新询问 UQ3 中的搜索标准可具体指示其正在搜索分别作为 RE18、RE20 和 RE22 的父（先前）修订条目的修订条目 RE17、RE20 和 RE22。接下来更新询问 UQ3 被传达到 RCS1 和 RCS3。

[0352] 在 RCS1 上，检索匹配更新询问 UQ3 中的标准的一个修订条目 {RE17}，并且创建响应更新 RU3 并用修订条目填充所述响应更新 RU3。接下来，RCS1 标识其连接的成员是 P3 和 P4。由于更新询问已从 P4 到达，所以其将指示 UQ1 中保持正向的数目的计数器值减去 1，本质上创建更新询问 UQ4，并且继续传播更新询问 UQ4 到其其它连接的成员，即装置 P3。

[0353] 在 RCS3 上，检索匹配更新询问 UQ3 中的标准的修订条目 {RE20、RE22}，并且创建响应更新 RU4 并用这些修订条目填充。接下来，RCS3 标识其连接的成员是 P1 和 P4。然而，如前述，目前为止其已接收到由 P1 转送给它的更新询问以及来自 P4 的原始更新询问。修



订控制系统 RCS3 接着标识通道中的所有成员已经在其更新询问中发送相同标准,并且因此确定其不再需要将这消息传播到其它成员。

[0354] 修订控制系统 RCS3 将响应更新 RU4 返回到请求相同更新询问的每个成员,即成员 P1 和 P4。包含 {RE20、RE22} 的响应更新被发送到修订控制系统 RCS1 和修订控制系统 RCS4。

[0355] 当修订控制系统 RCS1 从成员 P3 接收响应更新 RU4 时,其标识其缺失修订条目 RE20 和 RE22。其将所述修订条目添加到其响应更新 RU3,所述响应更新 RU3 现在包含条目 {RE17、RE20、RE22}。RCS1 接着将响应更新 RU3 返回到发起更新询问 UQ1 的成员 P4。此外,由于这些修订条目在其修订历史日志 RHL1-1 中缺失,所以其更新其可用更新列表以包括 RE20 和 RE22。

[0356] 再次,修订控制系统 RCS4 现在具有两个响应更新 RU3 和 RU4。其标识其缺失修订条目 RE17、RE20 和 RE22,并且因此用修订条目更新其可用更新列表。其还在其修订历史日志 RHL1-4 中创建对应修订条目。这次,其可基于每个修订条目 RE17、RE20 和 RE22 的先前修订条目的标识符使新修订条目在其进程路径中相关于修订历史日志中。其还标识其缺失修订条目 RE19、修订条目 RE20 的先前修订,并且因此用修订条目更新可用更新列表。

[0357] 修订管理器 115(在 RCS4 上)继续使对应于 {RE17、RE20、RE22} 的三个新修订条目在修订历史日志 RHL1-4 中相关,并且使接收在响应更新 RU2 {RE18、RE21、RE23} 中的修订条目与针对响应更新 RU4 创建的新修订条目相关,每个修订条目被链接到其各自先前修订。修订历史日志 RHL1-4 的状态示意图示在图 6N 中。注意,目前为止,修订控制系统 RCS4 在其可用更新列表中仅包括一个修订条目 {RE19}。其现在可创建新更新询问 UQ5 来检索从其修订历史日志 RHL1-4 中缺失的修订条目 RE19 以使修订条目 RE20 相关。如上所述,修订控制系统 RCS4 接着可继续获得缺失的修订条目,并且使修订条目 RE19、RE20 和 RE21 在其修订历史日志中相关于正确位置上。

[0358] 在更新其修订历史日志后,修订控制系统 RCS4 现在可继续检索实际修订。修订控制系统 RCS4 必须确定哪些成员具有所需更新(阶段 560)。在以 DHT 为基础的覆盖中,这将意味着针对所需更新的特定 ID 编写查询并且作出响应,接收具有对应于所述 ID 的修订的成员列表。但是,在这个实施例,使用延伸环运算法则来示例成员间通信。还应了解在这个实施例中,单独阶段用于检索修订条目和其对应修订,然而,在其它实施例中,修订可与其对应的修订条目一起以单个阶段检索。

[0359] 修订控制系统 RCS4 与 RCS1 和 RCS3 通信以请求具有对应于修订条目 {RE17、RE18、RE19、RE20、RE21、RE22、RE23} 的修订的成员列表。如上文所说明,修订条目可由包括对应元数据哈希码、UUID 等等的各种类型的标识符标识。

[0360] RCS1 从 RCS4 接收请求并且经由其修订控制引擎 110 确定其是否具有对应于修订条目 RE17、RE18 的元数据哈希码的修订。修订控制系统 RCS1 接着制定至 RCS4 的响应,指示其具有所需修订。

[0361] 修订控制系统 RCS3 还从 RCS4 接收请求并且在其修订控制引擎 110 的帮助下确定其具有对应于修订条目 RE19、RE20、RE21、RE22 和 RE23 的元数据哈希码的修订。修订控制系统 RCS3 接着制定至 RCS4 的响应,指示其具有所需修订。

[0362] 关于更新询问,在所示延伸环运算法则中,如果接收消息的任何成员无法找到对

应于消息中所请求的修订条目的修订,那么其可将消息转送到另一成员。

[0363] 修订控制系统 RCS4 接着接收上述两个响应。根据一个实现,对于对应于可用更新列表中的修订条目的元数据哈希码的每个修订,修订控制系统 RCS4 指示包含所述修订的对应成员 P1 和 P3 发送修订内容、或部分内容,并且接收所述内容。注意在许多情况下,例如如果修订较大,那么可在从对应成员发送内容时使用临时文件来存储所述内容。在接收到所述内容后,RCS4 的修订管理器 115 被配置来将过去修订存储在过去修订储存库 135 中,或者如果对应于当前元数据哈希码的修订条目是当前修订,那么由 RCS4 的修订管理器 115 将修订存储在被监控的文件系统上。

[0364] 图 6P 中图示了在修订控制系统 RCS4 的更新周期结束时其修订历史日志 RHL1-4 的状态。修订控制系统 RCS1 在其可用更新列表中还具有修订条目,并且可独立于修订控制系统 RCS4 的运行前进到阶段 560,因此从成员 P3 中获得更新。

[0365] 修订控制系统 RCS4 可经由显示器 160 向操作装置 P4 的用户指示其已完成更新周期。作出响应,操作装置 P4 的用户可经由用户界面 150 与 RCS4 互动,并且命令 RCS4 显示文件 F4 的历史。作出响应,修订控制系统 RCS4 可检索文件 F4 的所有进程路径,并且将其显示给用户。

[0366] 用户可发出命令给 RCS4 以显示关于记录在其修订历史日志 RHL1-4 中的修订的部分信息或全部信息。例如,用户可发出命令来显示文件 F4 的历史。作出响应,修订控制系统 RCS4 可(在修订管理器 115 的帮助下)检索对应于文件 F4 的所有进程路径。此外,修订管理器可检索位于修订历史日志的其它地方的其它修订条目并且特征在于与对应于文件 F4 中的进程路径中的修订条目的内容哈希码相同的内容哈希码。修订控制系统 RCS4 可在显示器 160 上显示关于所检索的修订条目的信息。

[0367] 应了解根据本发明的系统可为经过适当编程的计算机。同样,本发明涵盖可由计算机读取用于执行本发明方法的计算机程序。本发明还涵盖机器可读内存,所述机器可读内存有形地具体实施为可由机器执行以执行本发明方法的指令程序。

[0368] 虽然不同实施方案已经示出并描述,但是应了解并无将本发明限于这份公开内容之意,而是其意在涵盖落在如随附权利要求书中所定义的本发明范畴内的所有修改和替代构造。

[0369] 应了解下文权利要求中的术语“第一”、“第二”、“第三”等等的使用是为更清楚地区分权利要求内的类似元件并且不意在以任何方式暗示关于这些元件的顺序和重要性的任何事项。

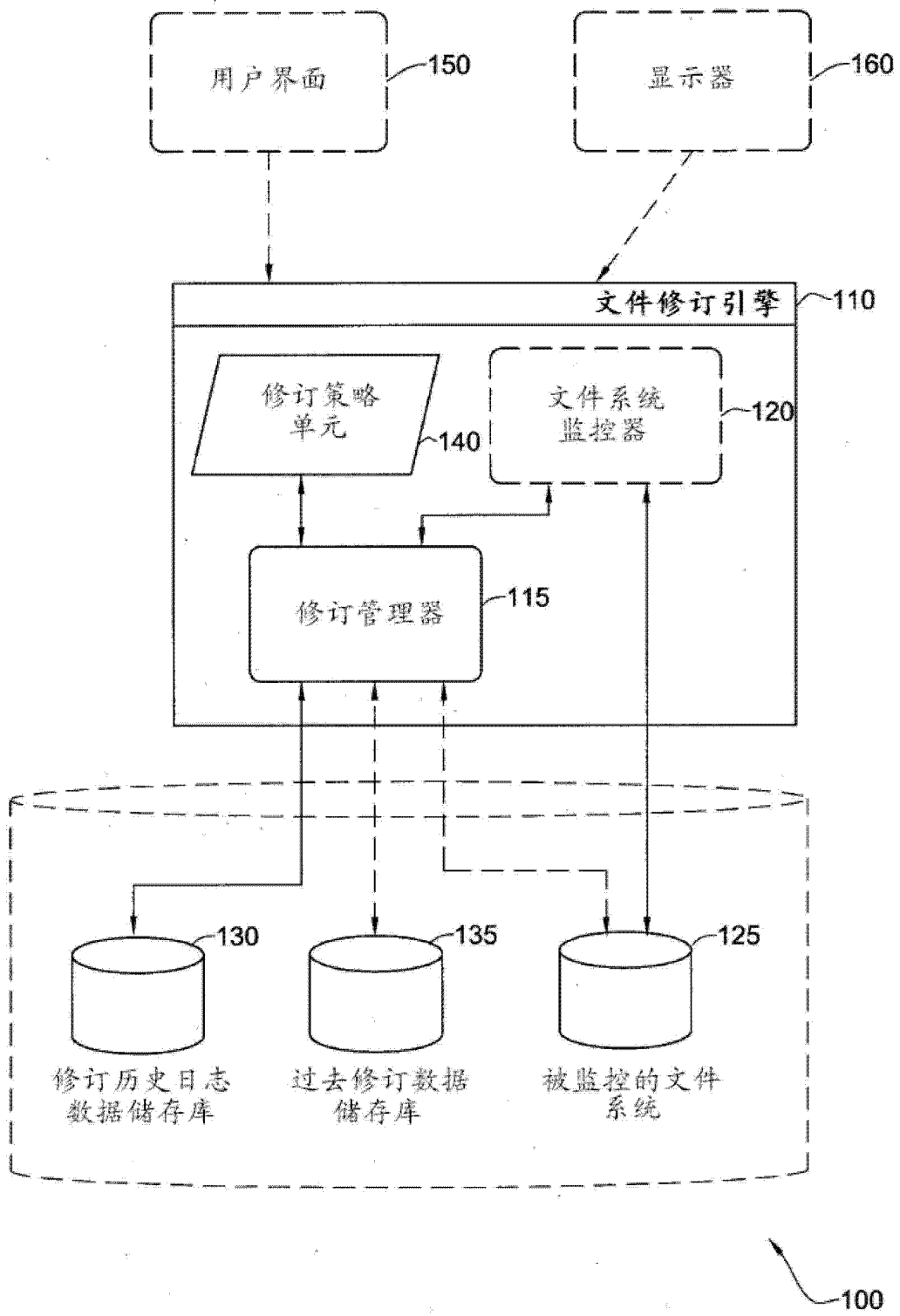


图 1

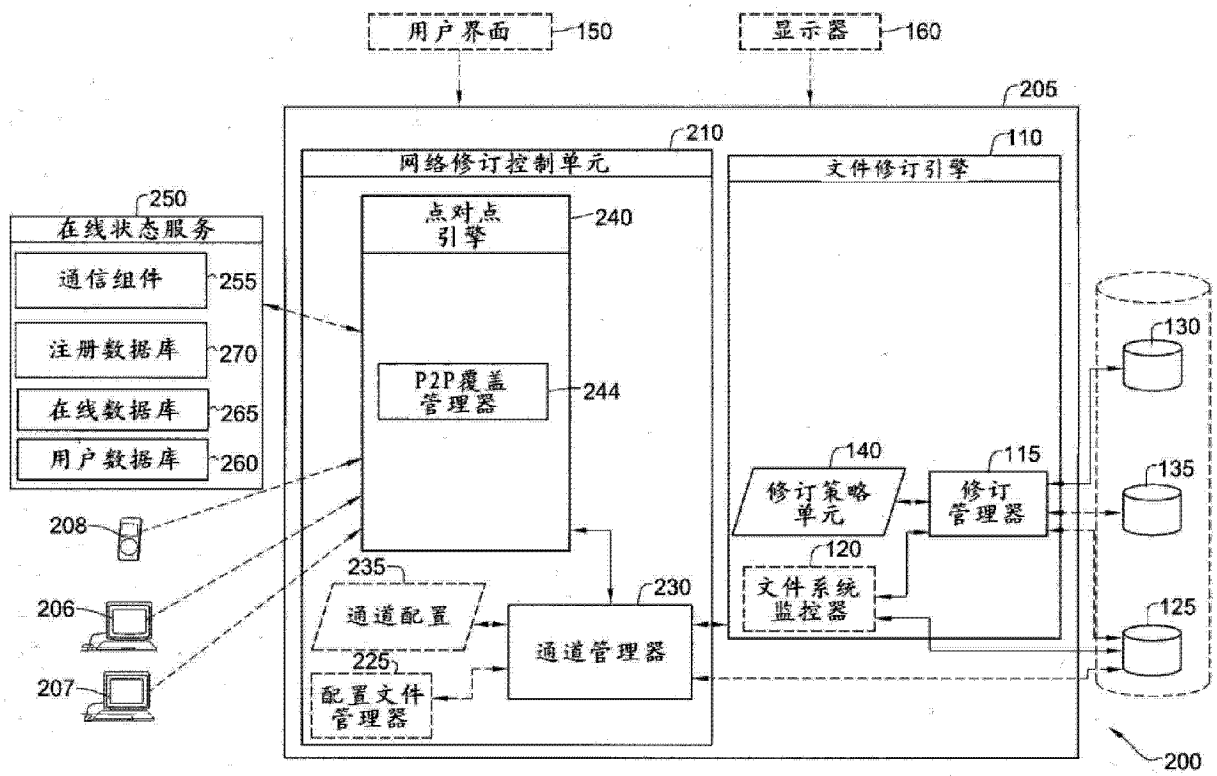


图 2

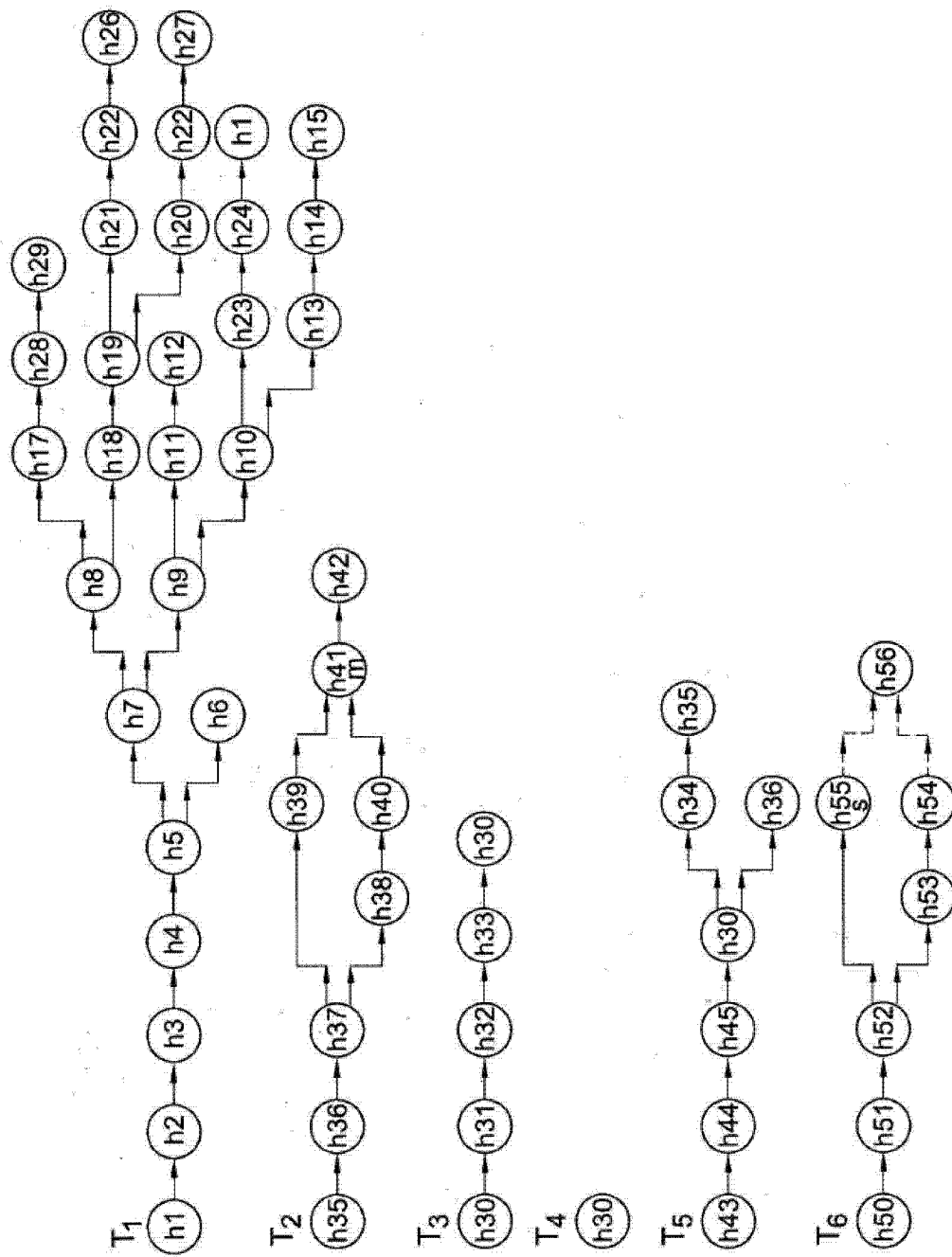


图 3

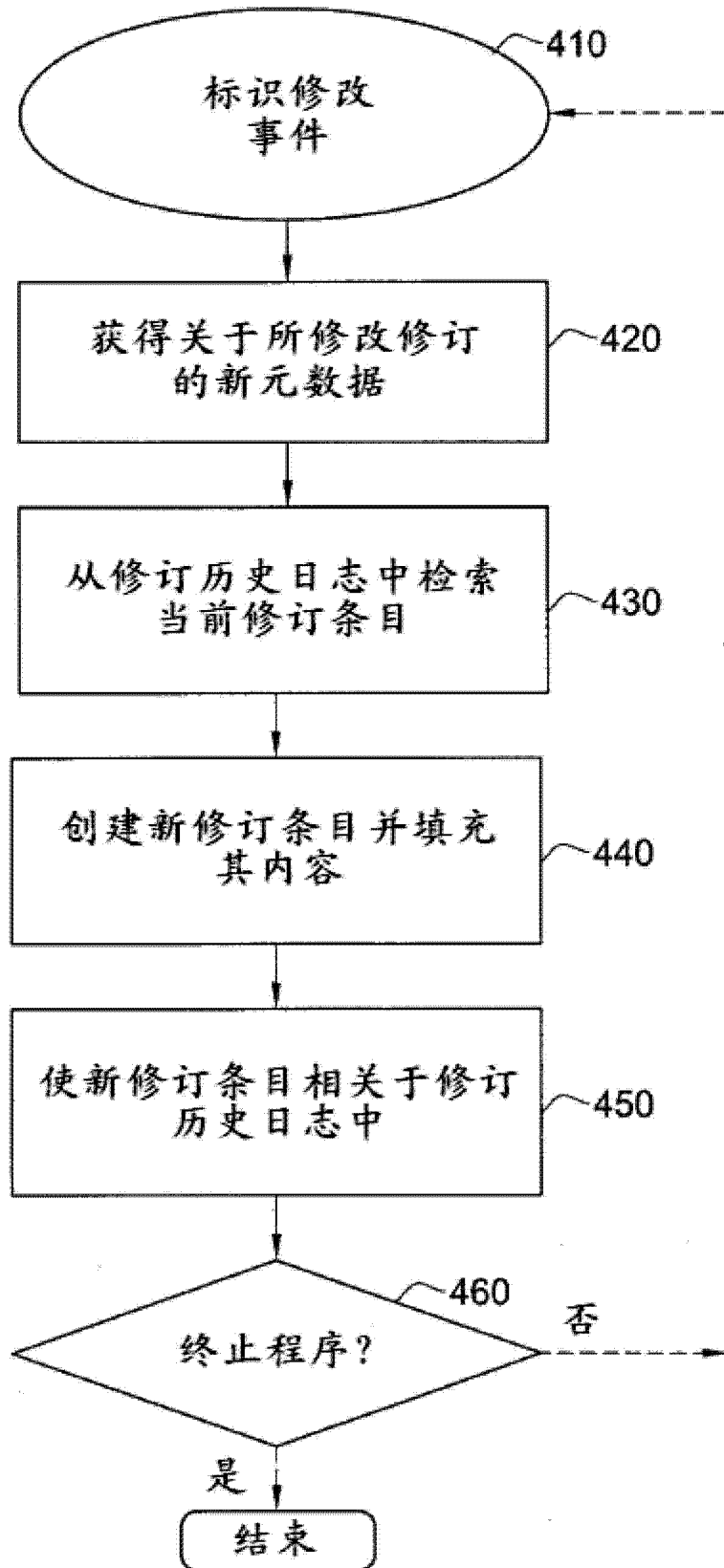


图 4

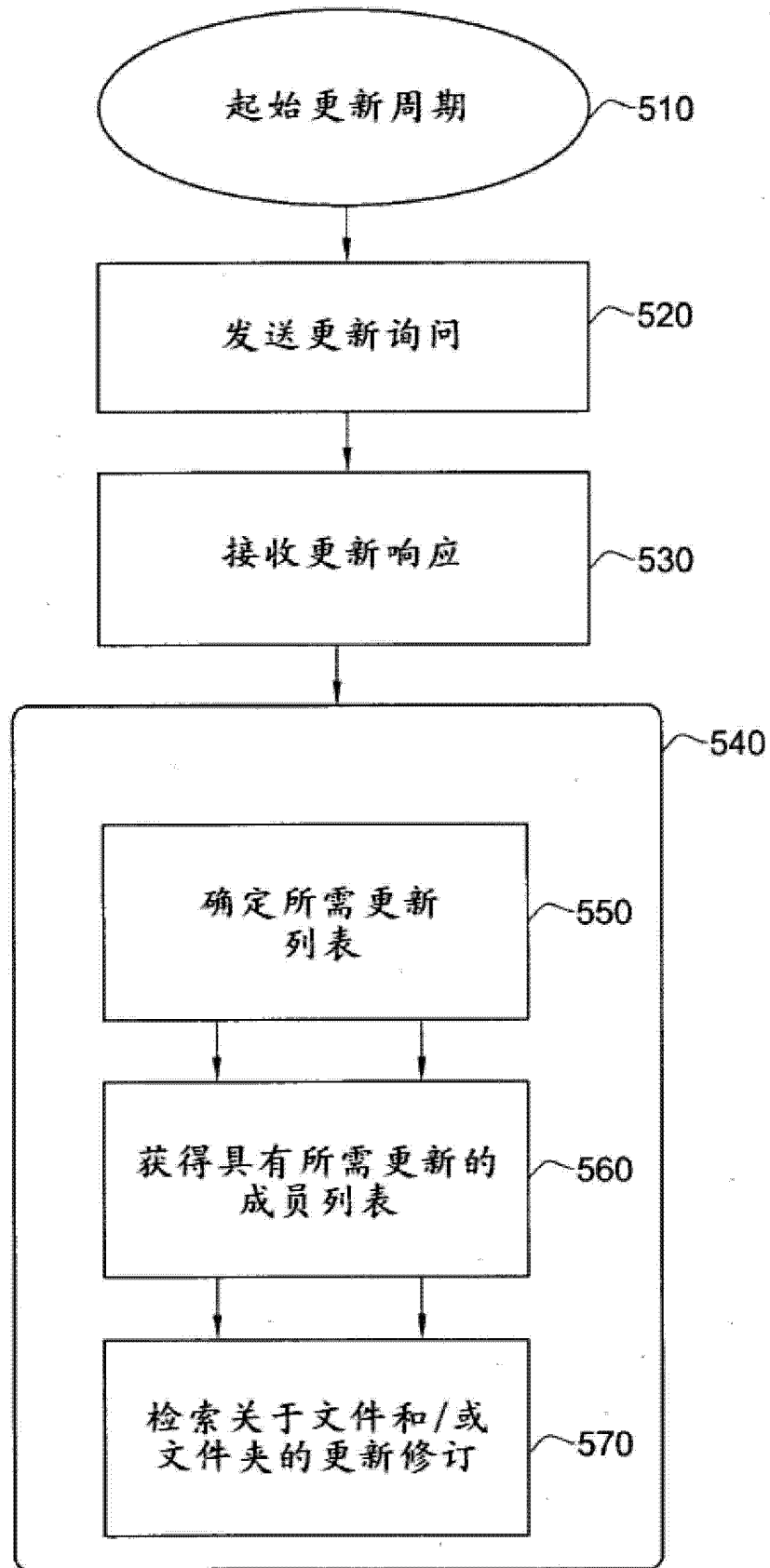


图 5

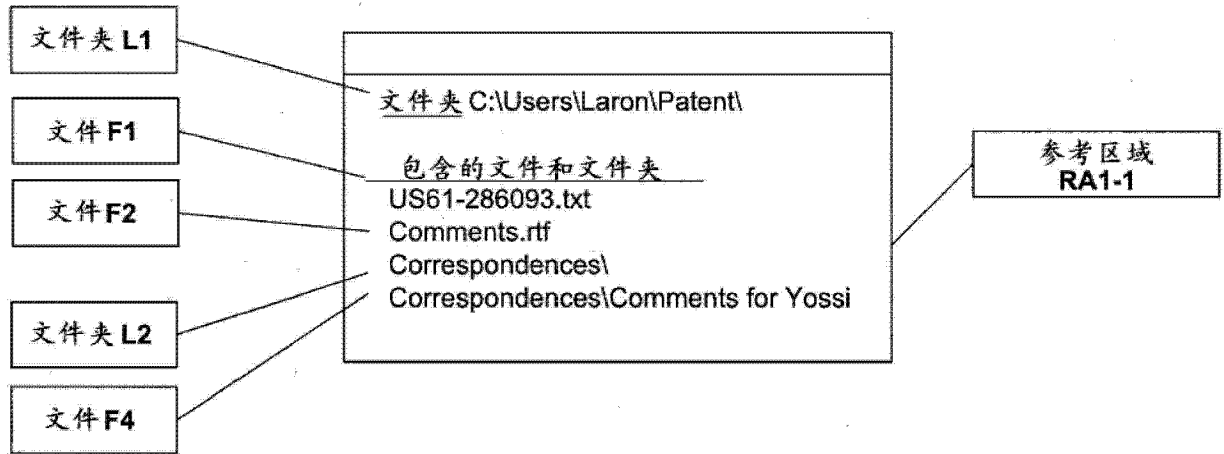


图 6A

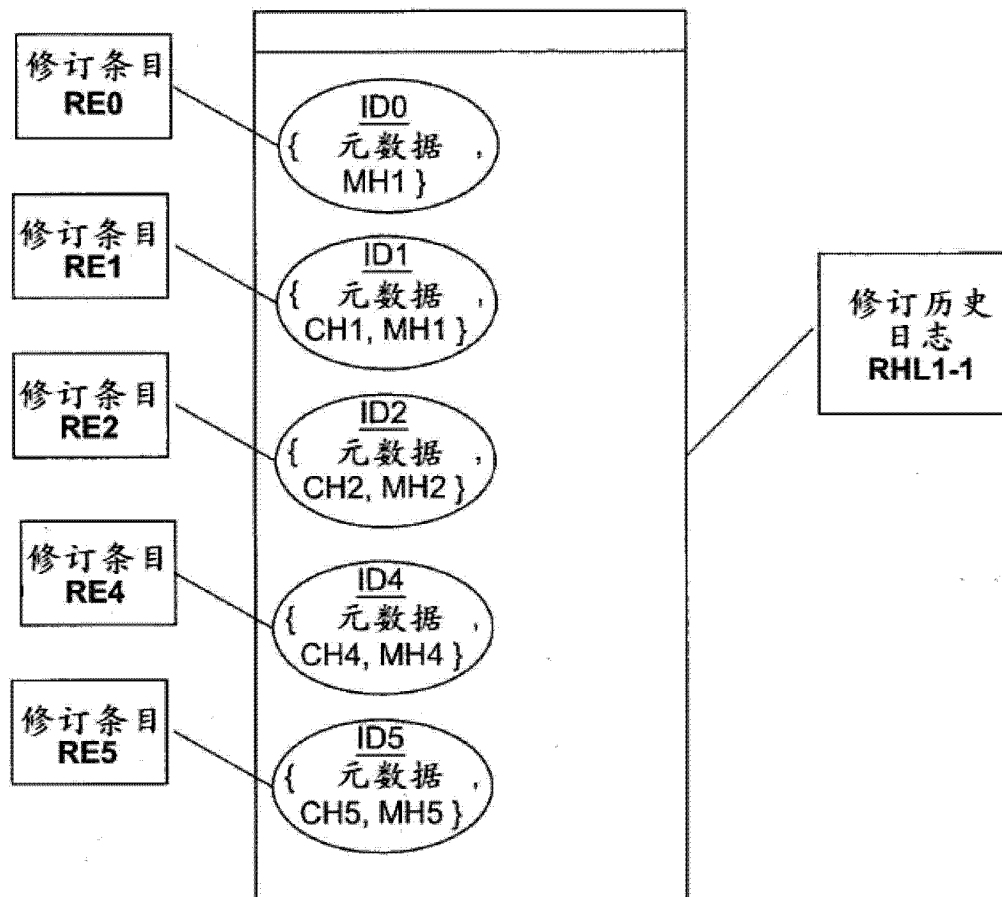


图 6B



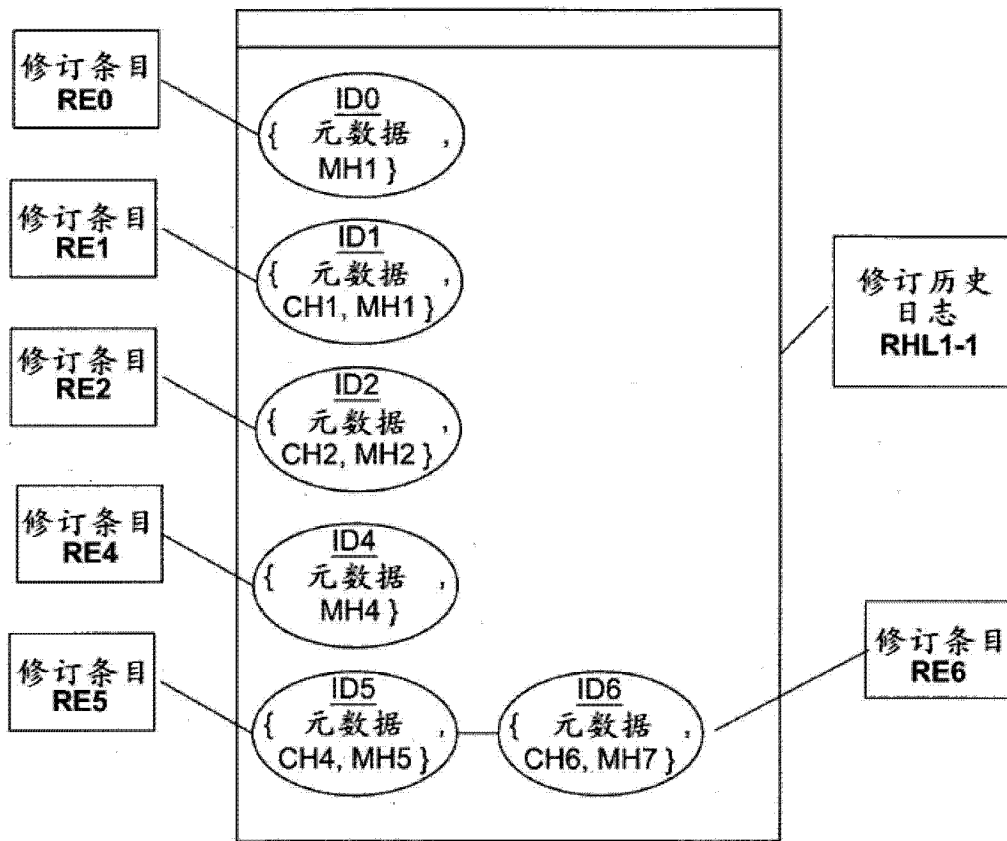


图 6C

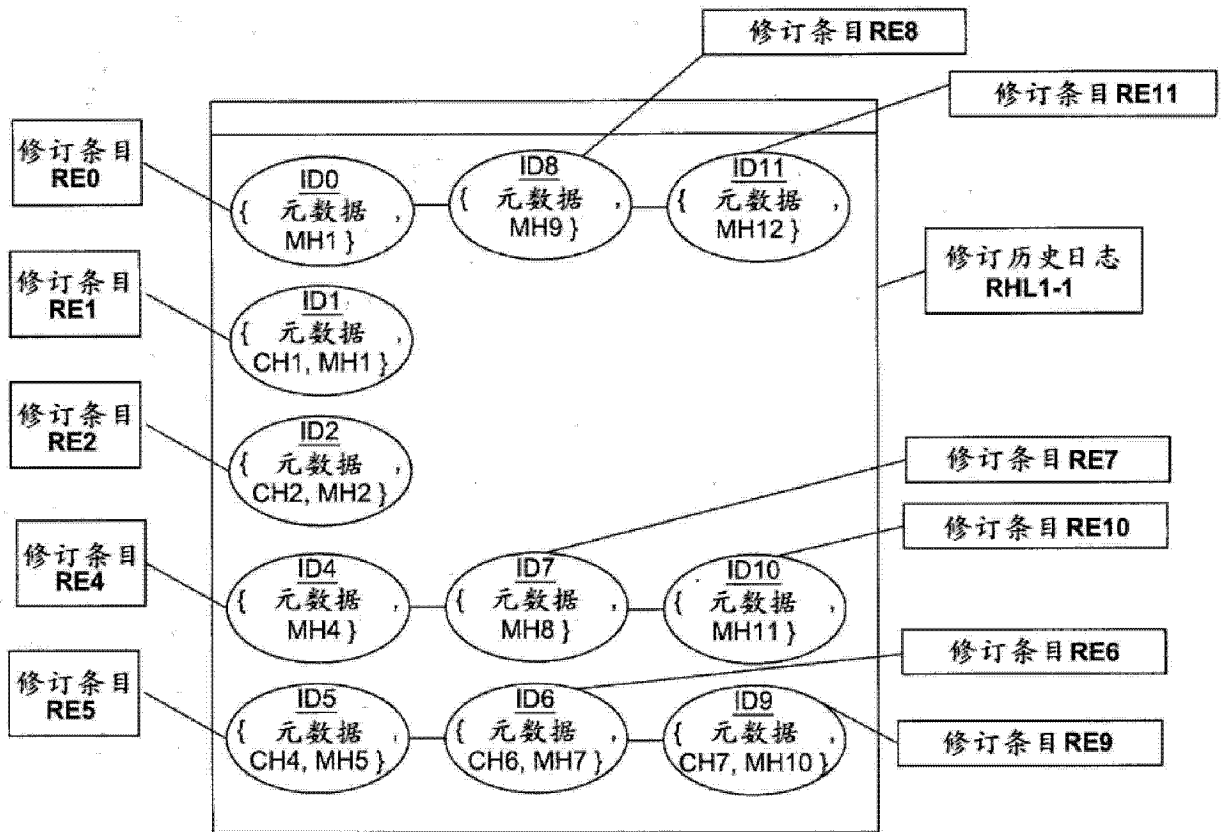


图 6D

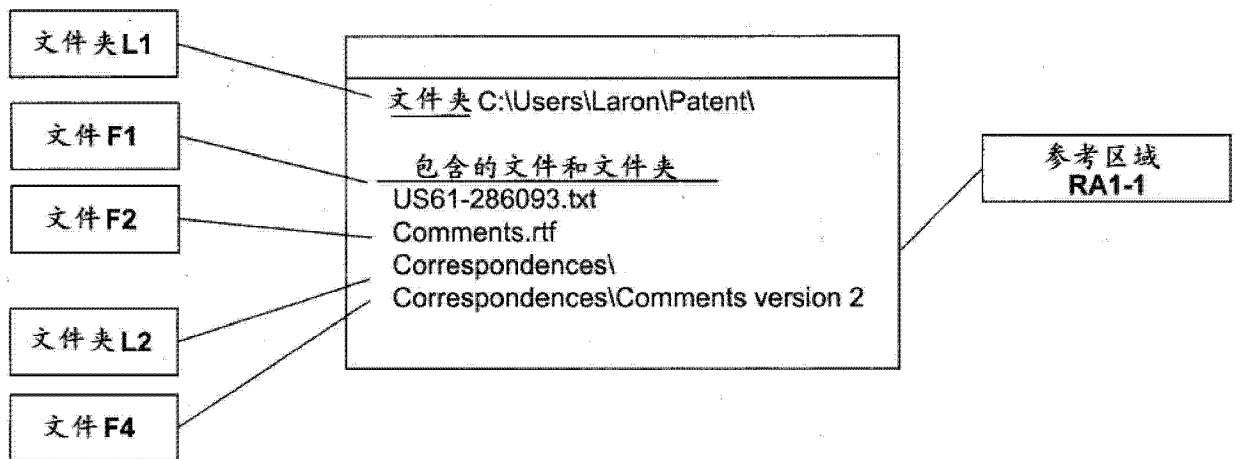


图 6E

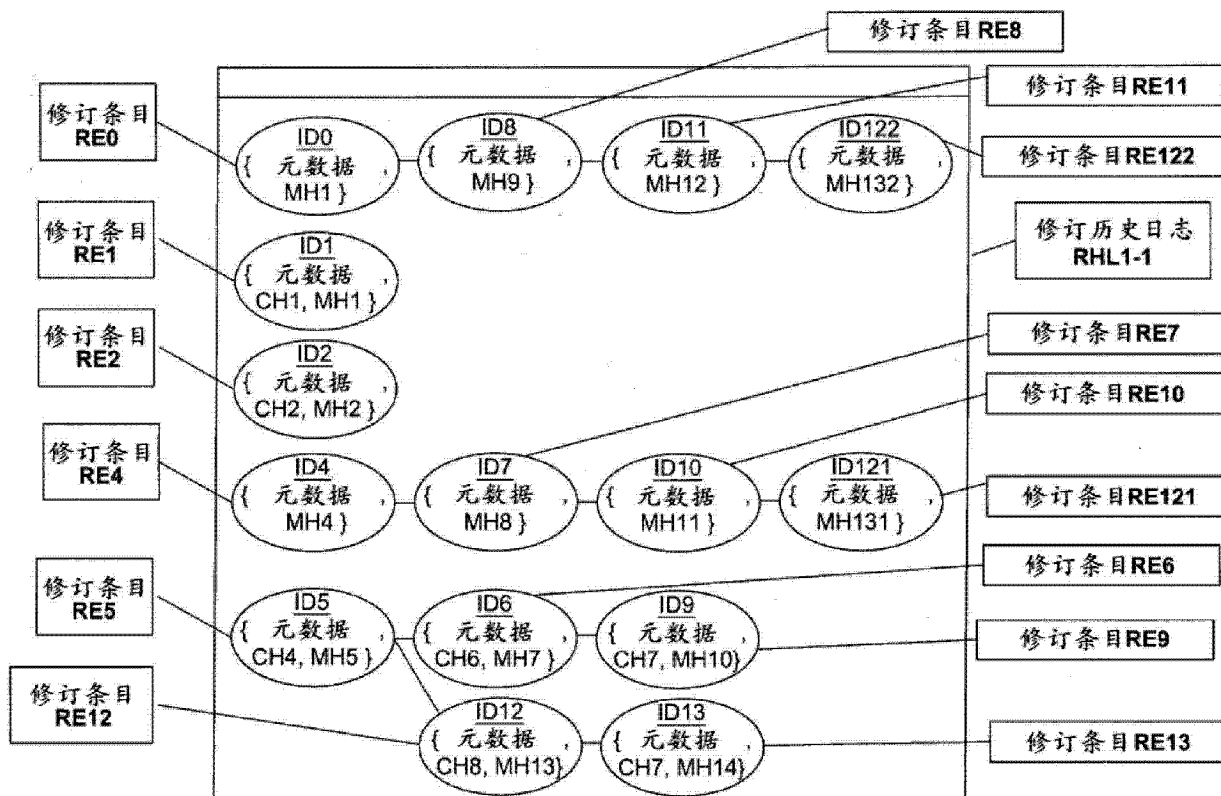


图 6F

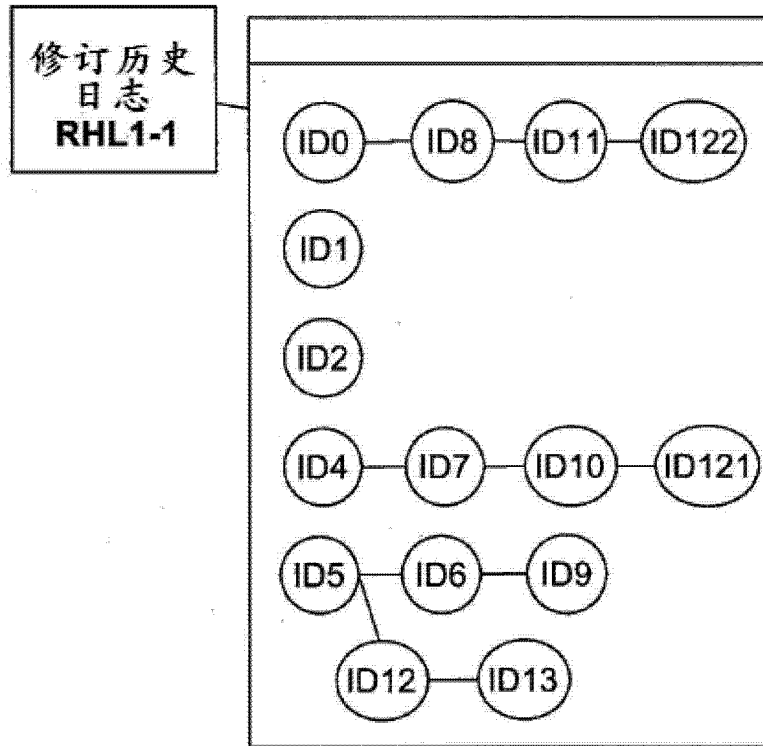


图 6G

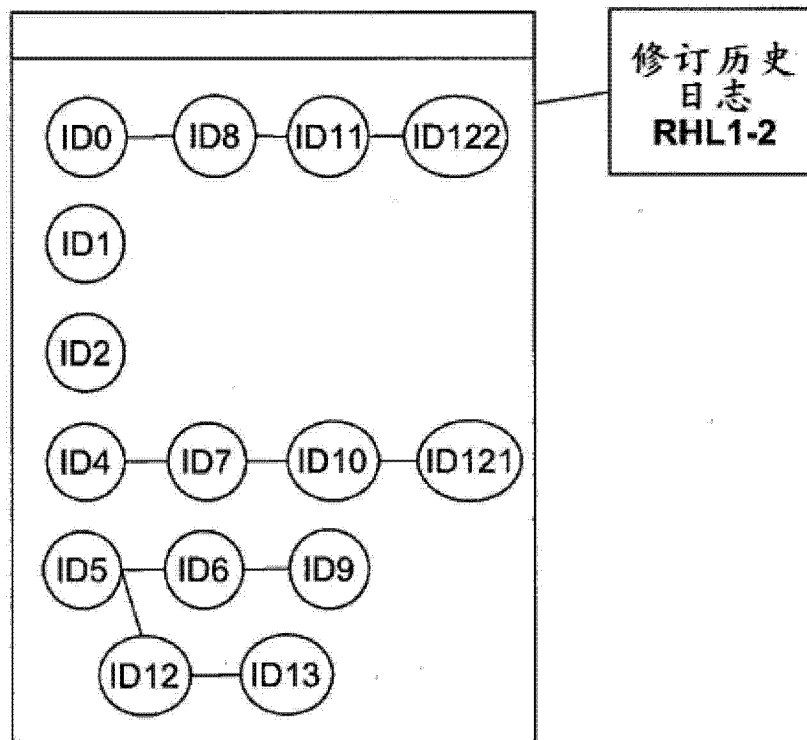


图 6H

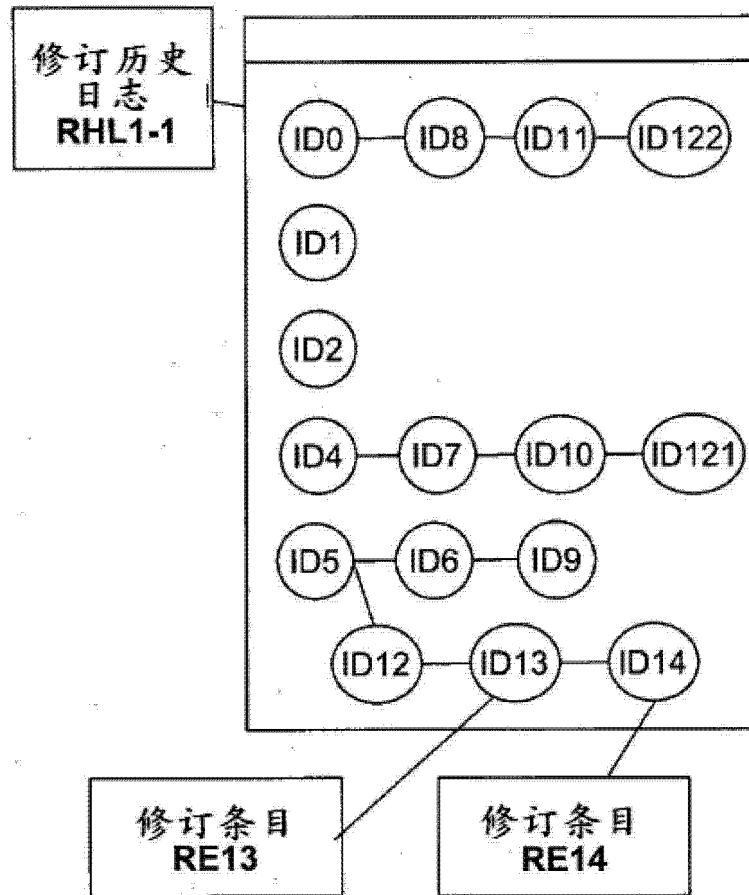


图 6I

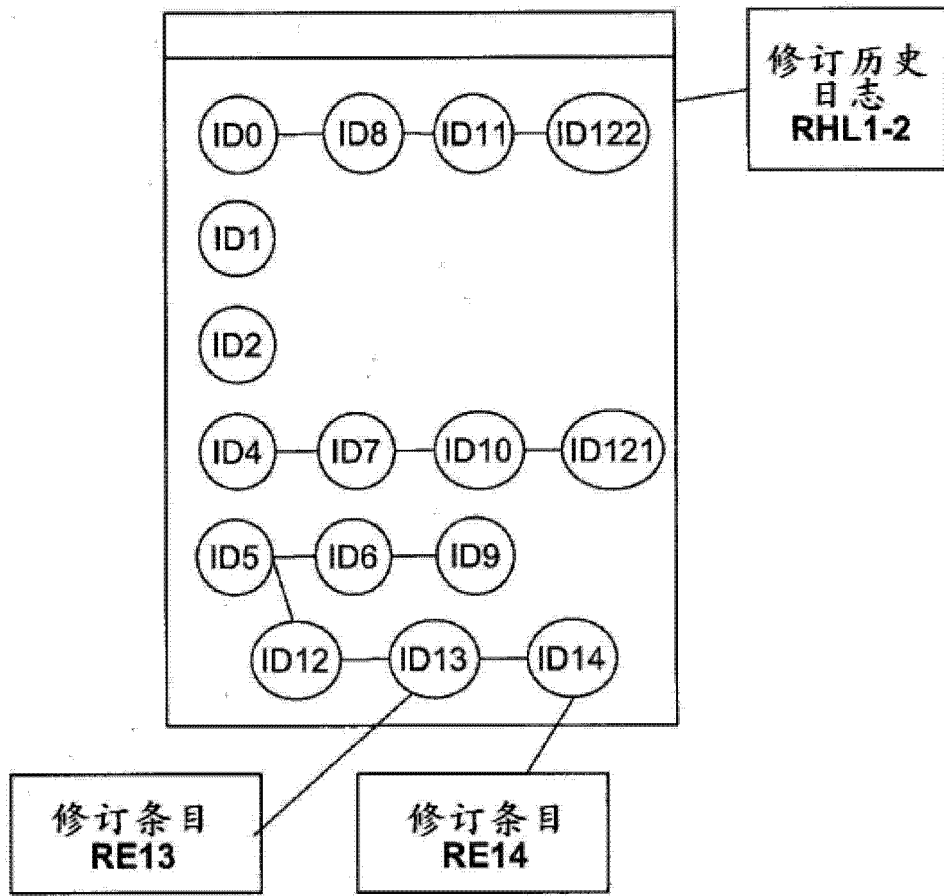


图 6J

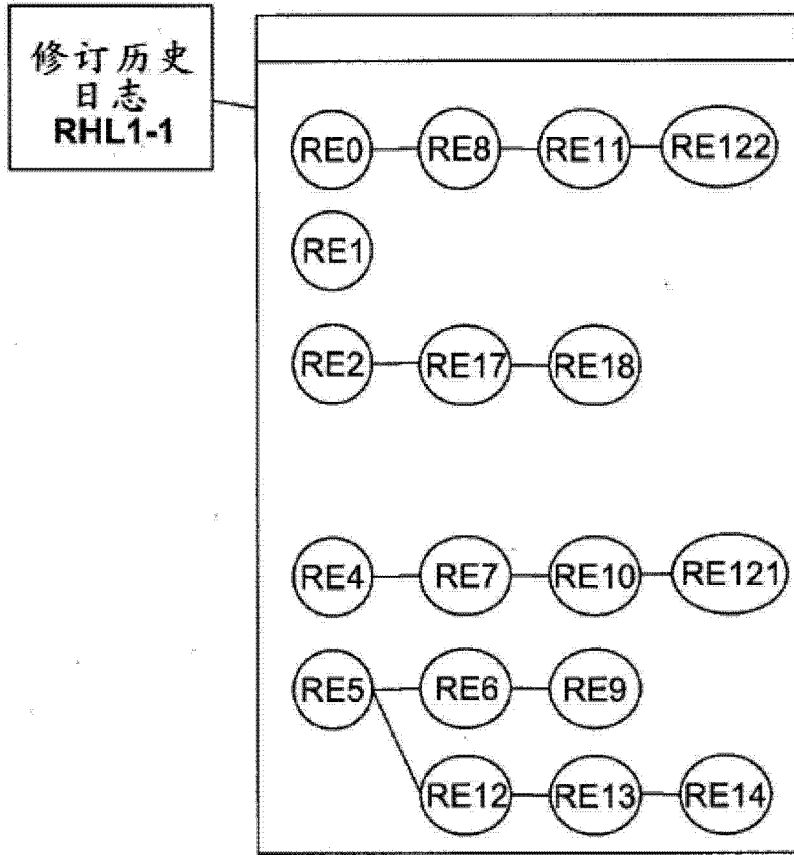


图 6K

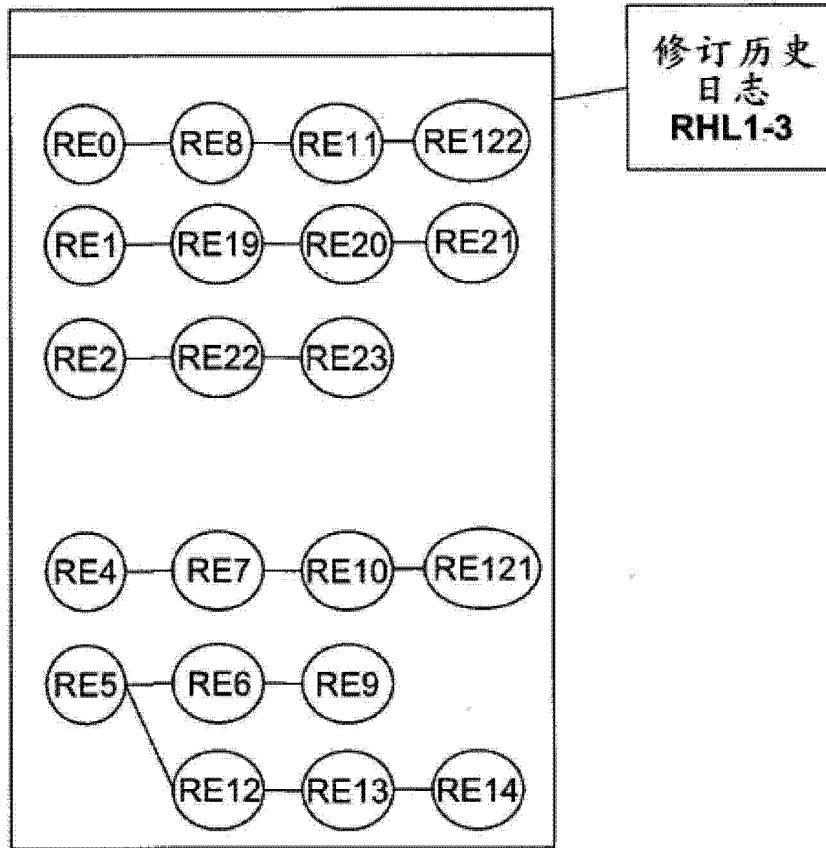


图 6L



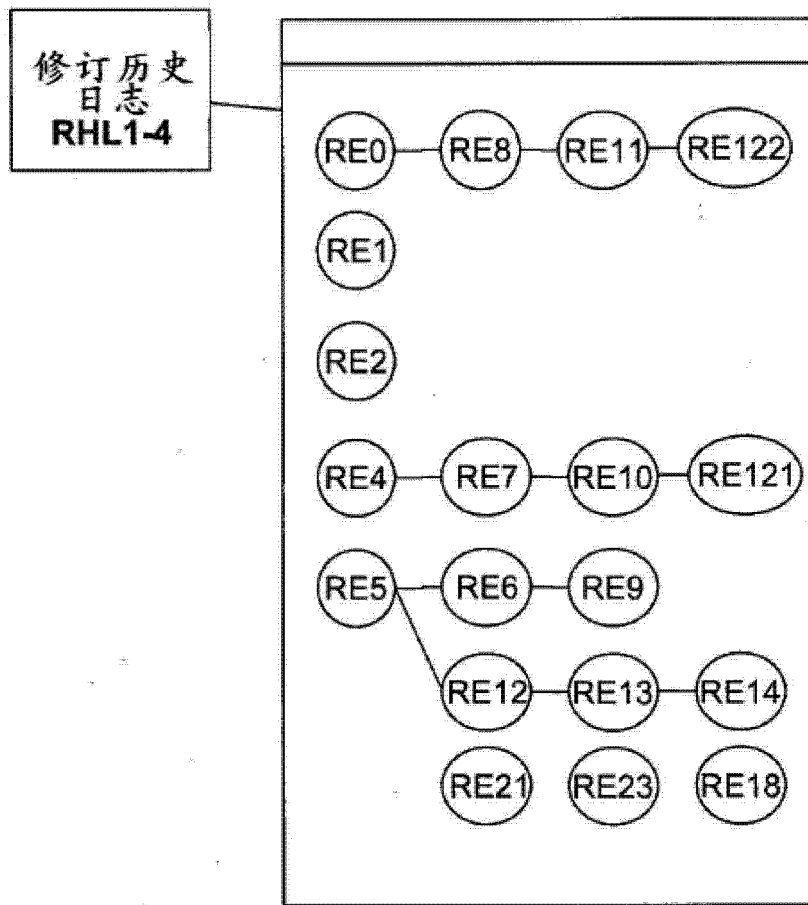


图 6M

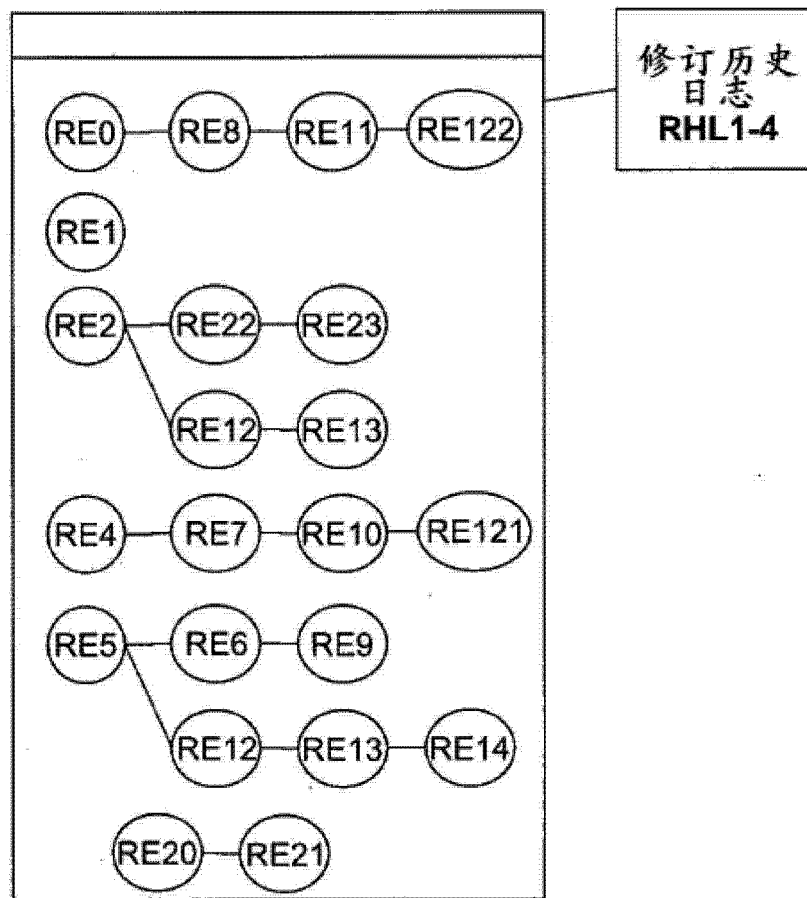


图 6N

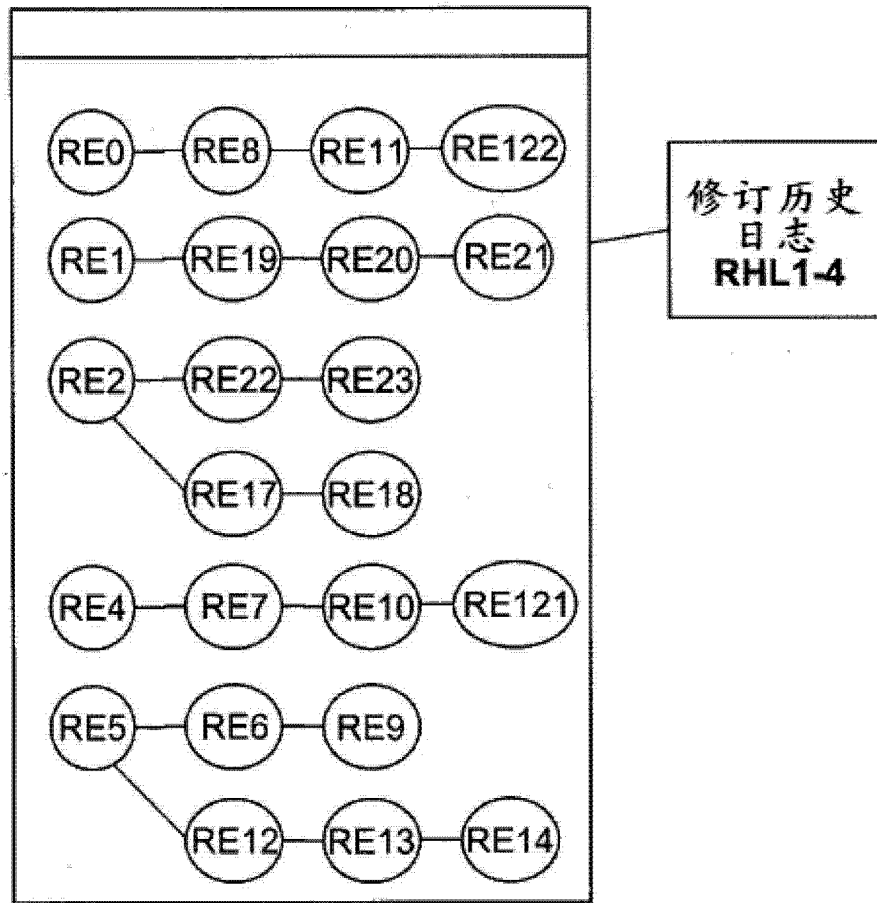


图 6P