



[12] 发明专利申请公布说明书

[21] 申请号 200680045934.0

[43] 公开日 2008 年 12 月 17 日

[11] 公开号 CN 101326491A

[22] 申请日 2006.9.28

[21] 申请号 200680045934.0

[30] 优先权

[32] 2005.10.7 [33] US [31] 11/246,921

[32] 2005.10.7 [33] US [31] 11/246,019

[32] 2005.10.7 [33] US [31] 11/246,018

[32] 2005.10.7 [33] US [31] 11/246,021

[86] 国际申请 PCT/US2006/037602 2006.9.28

[87] 国际公布 WO2007/044230 英 2007.4.19

[85] 进入国家阶段日期 2008.6.6

[71] 申请人 茨特里克斯系统公司

地址 美国佛罗里达州

[72] 发明人 J·H·诺尔德 D·R·霍伊

B·J·彼得森 T·N·特雷德

A·S·古亚拉蒂

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 张雪梅 张志醒

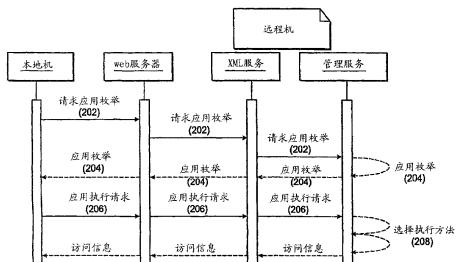
权利要求书 10 页 说明书 98 页 附图 35 页

[54] 发明名称

用于在应用程序的预定数量的执行方法之间
选择的方法

[57] 摘要

一种用于在预定数量的应用程序执行方法之间进行选择的方法，所述方法包括：响应于所接收的与本地机相关的信任状提供本地机可用的多个应用的枚举。响应于策略选择执行被枚举的应用的预定数量的方法之一，所述预定数量的方法包括用于所述被枚举的应用的应用流传输的方法。无须从远程位置下载文件即可满足对文件元数据的请求。可以访问与应用程序相关联的目录结构中的文件。本地机可以访问包括应用程序的多个文件。



1. 一种用于由远程机选择应用程序的执行方法的方法，所述方法包括：

(a) 接收信任状；

(b) 响应于所接收的信任状提供本地机可用的多个应用的枚举；

(c) 接收执行被枚举的应用的请求；以及

(d) 响应于策略选择执行被枚举的应用的预定数量的方法之一，所述预定数量的方法包括用于所述被枚举的应用的应用流的方法。

2. 根据权利要求 1 所述的方法，还包括提供与所述被枚举的应用相关联的访问信息的步骤，所述访问信息是响应于所选择的方法产生的。

3. 根据权利要求 1 所述的方法，其中步骤 (c) 还包括接收执行被枚举的应用的请求，所述被枚举的应用包括多个应用文件。

4. 根据权利要求 3 所述的方法，其中步骤 (c) 还包括接收持有多个应用文件的服务器的标识。

5. 根据权利要求 3 所述的方法，其中步骤 (c) 还包括接收多个应用文件的位置的标识，所述标识符合通用命名标准 (UNC)。

6. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括选择到本地机的所述被枚举的应用的应用流的方法。

7. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括选择用于在远程机上执行所述被枚举的应用的方法。

8. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括选择用于在本地机上执行所述被枚举的应用的方法。

9. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括选择用于下述的方法：

(d-1) 向远程机流传输所述被枚举的应用；

(d-2) 在所述远程机上执行所述被枚举的应用；以及

(d-3) 向所述本地机提供执行所述被枚举的应用产生的应用输出数据。

10. 根据权利要求 9 所述的方法，其中步骤 (d) 还包括在步骤 (d-1) 之前评估所述本地机。

11. 根据权利要求 10 所述的方法，其中步骤 (d) 还包括响应于

所述评估决定对所述被枚举的应用进行流传输。

12. 根据权利要求 9 所述的方法，其中步骤 (d-1) 包括对包括所述被枚举的应用的多个文件之一进行流传输。

13. 根据权利要求 9 所述的方法，其中步骤 (d) 还包括响应于对所接收的信任状的评估选择所述方法。

14. 根据权利要求 9 所述的方法，其中步骤 (d-3) 还包括通过表示层协议提供所述应用输出数据。

15. 根据权利要求 14 所述的方法，其中步骤 (d-3) 还包括通过独立计算结构表示层协议或远程桌面 Windows 表示层协议或 X-Windows 表示层协议提供应用输出数据。

16. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括标识要执行的所述被枚举的应用的版本。

17. 根据权利要求 16 所述的方法，其中步骤 (d) 还包括响应于所述本地机的特征的标识来标识所述版本。

18. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括决定执行与所述本地机的特征兼容的所述被枚举的应用的版本。

19. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括决定执行与所述本地机的操作系统兼容的被枚举的应用的版本。

20. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括决定执行与所述本地机的操作系统的修订级兼容的被枚举的应用的版本。

21. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括决定执行与由所述本地机上的操作系统指定的语言兼容的被枚举的应用的版本。

22. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括通过向所接收的信任状应用策略来做出访问控制决定。

23. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括响应于对所接收的信任状的评估选择所述方法。

24. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括提供响应于对所接收的信任状的评估而产生的启动权证，所述启动权证包括认证信息。

25. 根据权利要求 1 所述的方法，其中步骤 (d) 还包括提供与所述可执行程序相关联的访问信息，所述访问信息包括所述被枚举的应

用的位置的标识，所述标识符合通用命名标准（UNC）。

26. 根据权利要求 1 所述的方法，其中步骤 (a) 还包括如下步骤：

- (a-1) 从本地机接收访问资源的请求；
- (a-2) 由收集代理收集信任状；以及
- (a-3) 接收所收集的信任状。

27. 根据权利要求 26 所述的方法，其中步骤 (a-2) 还包括通过在所述本地机上执行至少一个脚本来收集信任状。

28. 根据权利要求 26 所述的方法，其中步骤 (a-3) 还包括响应于所接收的信任状评估所述本地机。

29. 根据权利要求 26 所述的方法，其中步骤 (a-3) 还包括响应于所收集的信任状决定对所述被枚举的应用进行流传输。

30. 根据权利要求 26 所述的方法，其中步骤 (a-3) 包括响应于所收集的信任状对包括所述被枚举的应用的多个文件之一进行流传输。

31. 一种用于在网络化计算机系统中在本地对文件元数据的请求做出响应而无需从远程位置下载文件的方法，所述方法包括：

- (a) 从远程机接收 (i) 表示所述远程机存储的应用程序的目录结构以及 (ii) 与包括所存储应用程序的每个文件相关联的元数据；
- (b) 存储所述目录结构和所述元数据；
- (c) 接收访问与所述目录结构中的特定文件相关联的元数据的至少一个请求；以及
- (d) 利用所存储的元数据对所述至少一个请求做出响应。

32. 根据权利要求 31 所述的方法，还包括如下步骤：

- (e) 接收枚举与本地应用程序相关联的目录结构的请求；以及
- (f) 利用所存储的目录结构对所述请求做出响应。

33. 根据权利要求 31 所述的方法，其中步骤 (a) 还包括从远程机接收与包括所存储的应用程序的每个文件相关联的元数据，所述元数据包括所述至少一个文件的别名。

34. 根据权利要求 31 所述的方法，其中步骤 (a) 还包括从远程机接收与包括所存储的应用程序的每个文件相关联的元数据，所述元数据包括所述至少一个文件的名称，所述名称具有八个字符的长度、点和三个字符的扩展名。

35. 根据权利要求 31 所述的方法，其中步骤（b）还包括在隔离环境中存储所述目录结构和所述元数据。

36. 根据权利要求 31 所述的方法，其中步骤（c）还包括接收确定本地是否有包括所存储的应用程序的文件拷贝的请求。

37. 根据权利要求 36 所述的方法，其中步骤（d）还包括访问所存储的元数据并伪装地表示本地有所述文件的拷贝。

38. 根据权利要求 31 所述的方法，其中步骤（d）还包括响应于所接收的元数据满足 Windows 操作系统 FindFirst 操作。

39. 根据权利要求 31 所述的方法，其中步骤（d）还包括响应于所接收的元数据满足 Windows 操作系统 Findnext 操作。

40. 根据权利要求 31 所述的方法，其中步骤（d）还包括响应于所接收的元数据满足标识目录结构中的根节点的操作。

41. 根据权利要求 31 所述的方法，其中步骤（d）还包括响应于所接收的元数据满足标识目录结构中的节点的操作。

42. 根据权利要求 31 所述的方法，其中步骤（d）还包括响应于所接收的元数据满足标识与目录结构中的节点相关的访问时间的操作。

43. 根据权利要求 31 所述的方法，其中步骤（d）还包括响应于所接收的元数据满足标识与目录结构中的节点相关的修改时间的操作。

44. 根据权利要求 31 所述的方法，其中步骤（d）还包括响应于所接收的元数据满足标识目录结构中的被修改节点的操作。

45. 根据权利要求 31 所述的方法，其中步骤（d）还包括生成枚举所接收的目录结构的本地目录结构的枚举。

46. 一种用于在网络化计算机系统中在本地对文件元数据的请求做出响应而无需从远程位置下载文件的系统，所述系统包括：

 目录结构，其标识与至少一个应用程序相关联的多个文件并包括与所述多个文件中的至少一个相关联的文件元数据，所述多个文件中的至少一个驻留在远程机上；

 存储所述目录结构的高速缓冲存储器元件；以及

 文件系统过滤器驱动程序，其拦截访问与该至少一个远程存储的文件相关联的元数据的请求，访问所述高速缓冲存储器元件并利用所

存储的目录结构对所述至少一个请求做出响应。

47. 根据权利要求 46 所述的系统，其中所述目录结构还包括与至少一个应用程序相关联的多个文件，所述至少一个应用程序驻留在远程机上。

48. 根据权利要求 46 所述的系统，其中所述目录结构还包括与至少一个应用程序相关联的多个文件，所述多个文件驻留在远程机上。

49. 根据权利要求 46 所述的系统，其中所述目录结构还包括与所述多个文件中的至少一个文件相关联的文件元数据，所述文件元数据包括所述至少一个文件的别名。

50. 根据权利要求 46 所述的系统，其中所述目录结构还包括文件元数据，所述文件元数据包括至少一个文件的名称，所述名称具有八个字符的长度、点以及三字符的扩展名。

51. 根据权利要求 46 所述的系统，其中所述目录结构还包括与所述多个文件中的至少一个文件相关联的文件元数据，所述文件元数据包括用于对 Windows 操作系统 Find First 请求做出响应的信息。

52. 根据权利要求 46 所述的系统，其中所述目录结构还包括与所述多个文件中的至少一个文件相关联的文件元数据，所述文件元数据包括用于对 Windows 操作系统 FindNext 请求做出响应的信息。

53. 根据权利要求 46 所述的系统，其中所述目录结构还包括与所述多个文件中的至少一个文件相关联的文件元数据，所述文件元数据包括用于对标识目录结构中的根节点的请求做出响应的信息。

54. 根据权利要求 46 所述的系统，其中所述目录结构还包括与所述多个文件中的至少一个文件相关联的文件元数据，所述文件元数据包括用于对标识目录结构中的节点的请求做出响应的信息。

55. 根据权利要求 46 所述的系统，其中所述目录结构还包括与所述多个文件中的至少一个文件相关联的文件元数据，所述文件元数据包括用于对标识目录结构中的被修改节点的请求做出响应的信息。

56. 根据权利要求 46 所述的系统，其中所述目录结构还包括与所述多个文件中的至少一个文件相关联的文件元数据，所述文件元数据包括用于对标识与目录结构中的节点相关联的修改时间的请求做出响应的信息。

57. 根据权利要求 46 所述的系统，其中所述目录结构还包括与所

述多个文件中的至少一个文件相关联的文件元数据，所述文件元数据包括用于对标识与目录结构中的节点相关联的访问时间的请求做出响应的信息。

58. 根据权利要求 46 所述的系统，其中所述目录结构还包括树形结构，所述树中的至少一个节点表示与应用程序相关联的文件。

59. 根据权利要求 46 所述的系统，其中所述目录结构还包括嵌入于至少一个文件中的文件元数据。

60. 一种用于访问与应用相关联的目录结构中的文件的方法，所述方法包括：

- (a) 拦截应用访问文件的请求；
- (b) 将所述请求重定向到第一隔离环境；
- (c) 确定所述所请求的文件不存在于所述第一隔离环境中；
- (d) 响应于在包括驻留在远程机上的多个应用文件的目录结构枚举中标识该文件的确定，将所述请求重定向到第二隔离环境；以及
- (e) 响应于所述第二隔离环境不包含所述文件且在枚举中标识该文件的确定，从所述远程机检索所请求的文件。

61. 根据权利要求 60 所述的方法，其中步骤 (a) 包括拦截应用访问可执行文件的请求。

62. 根据权利要求 60 所述的方法，其中步骤 (a) 包括拦截应用访问文件的请求，在本地机上执行所述应用的一部分。

63. 根据权利要求 60 所述的方法，其中步骤 (a) 还包括拦截能够接收应用流的应用访问文件的请求。

64. 根据权利要求 60 所述的方法，其中步骤 (b) 还包括拦截所述应用访问所述文件的请求，所述应用的至少一部分驻留在所述本地机上的第一隔离环境中。

65. 根据权利要求 60 所述的方法，其中步骤 (e) 还包括从第二远程机检索所述所请求的文件。

66. 根据权利要求 60 所述的方法，其中步骤 (e) 还包括通过所述远程机将所请求的文件流传输到所述本地机。

67. 根据权利要求 60 所述的方法，其中步骤 (e) 还包括在所述第二隔离环境中存储所检索的文件。

68. 根据权利要求 60 所述的方法，其中步骤 (e) 还包括对所检

索的文件加密。

69. 根据权利要求 60 所述的方法，其中步骤 (e) 还包括以加密形式存储所检索的文件。

70. 根据权利要求 60 所述的方法，还包括如下步骤：响应于在所述枚举中未标识所述文件的确定，将对所述文件的请求重定向到所述第一隔离环境之外和所述第二隔离环境之外的环境。

71. 根据权利要求 60 所述的方法，还包括如下步骤：

(f) 拦截在第三隔离环境中执行的另一应用提出的访问文件的请求；

(g) 响应于在所述枚举中枚举该文件且所述第二隔离环境不包含所述文件的确定，将所述请求重定向到所述第二隔离环境；以及

(h) 通过所述第二隔离环境向所述另一应用提供对所述文件的访问。

72. 根据权利要求 71 所述的方法，其中步骤 (h) 还包括在所述第三隔离环境中存储所述文件。

73. 一种用于访问与应用相关联的目录结构中的文件的系统，包括：

第一隔离环境，应用从所述第一隔离环境执行并请求访问文件；

第二隔离环境，其存储与安装在远程机上的多个应用文件相关联的目录结构的枚举；

过滤器驱动程序：

拦截访问所述文件的请求；

将所述请求重定向到所述第一隔离环境；

确定所述所请求的文件不存在于所述第一隔离环境中；

响应于在所述目录结构枚举中标识该文件的确定将该请求重定向到所述第二隔离环境；以及

与所述过滤器驱动程序通信的收发器，所述收发器响应于所述过滤器驱动程序做出的所述第二隔离环境不包含所述文件且在所述枚举中标识该文件的确定，向远程机发送所述重定向的请求并响应于所述请求接收包括所请求的文件的流。

74. 根据权利要求 73 所述的系统，其中所述收发器还包括在所述第二隔离环境中存储流传输的文件。

75. 根据权利要求 74 所述的系统，还包括所述过滤器驱动程序拦截另一应用访问所述文件的请求，所述过滤器驱动程序将所述请求重定向到所述第二隔离环境。

76. 根据权利要求 73 所述的系统，还包括响应于所述收发器在所述第二隔离环境中存储流传输的文件的确定，所述过滤器驱动程序拦截另一应用访问所述文件的请求，所述过滤器驱动程序将所述请求重定向到所述第二隔离环境。

77. 根据权利要求 73 所述的系统，其中所述第二隔离环境还包括高速缓冲存储器元件。

78. 根据权利要求 73 所述的系统，其中所述第二隔离环境还包括在高速缓冲存储器元件中存储所述目录结构的枚举。

79. 根据权利要求 73 所述的系统，其中所述收发器从所述远程机接收包括所请求的文件的流。

80. 根据权利要求 73 所述的系统，其中所述收发器从第二远程机接收包括所请求的文件的流。

81. 一种用于通过本地机访问包括应用程序的多个文件的方法，所述方法包括：

(a) 通过本地机接收包括访问信息的文件，所述访问信息用于访问多个应用文件并执行能够接收应用流的第一客户端；

(b) 响应于所述文件检索所述多个应用文件的标识；

(c) 响应于所述文件检索执行所述多个应用文件所需的至少一个特征；

(d) 确定所述本地机是否包括所述至少一个特征；以及

(e) 响应于所述本地机缺少所述至少一个特征的确定执行第二客户端，所述第二客户端请求在远程机上执行所述多个应用文件。

82. 根据权利要求 81 所述的系统，其中步骤 (a) 还包括接收访问信息，所述访问信息包括包含应用程序的多个应用文件的位置的标识。

83. 根据权利要求 81 所述的系统，其中步骤 (b) 还包括检索多个应用文件的标识，所述多个应用文件包括一个或多个应用程序。

84. 根据权利要求 81 所述的系统，其中步骤 (b) 还包括接收多个可用应用程序的枚举，所述枚举是响应于所述访问信息而产生的。

85. 根据权利要求 81 所述的系统，其中步骤 (d) 还包括评估所述本地机上的操作系统。

86. 根据权利要求 81 所述的方法，其中步骤 (d) 还包括标识所述本地机上的操作系统使用的语言。

87. 根据权利要求 81 所述的系统，其中步骤 (d) 还包括标识所述本地机上的操作系统的修订级。

88. 根据权利要求 81 所述的系统，其中步骤 (d) 还包括标识驻留在所述本地机上的应用程序的应用版本。

89. 根据权利要求 81 所述的系统，其中步骤 (d) 还包括确定所述本地机是否包括设备驱动程序。

90. 根据权利要求 81 所述的方法，其中步骤 (d) 还包括确定所述本地机是否包括执行所述多个应用文件的许可。

91. 根据权利要求 81 所述的方法，其中步骤 (e) 还包括通过所述第二客户端接收在所述远程机上执行所述多个应用文件而产生的应用输出数据。

92. 根据权利要求 91 所述的方法，其中步骤 (e) 还包括通过所述第二客户端在所述本地机上显示所述应用输出。

93. 根据权利要求 81 所述的方法，其中步骤 (e) 还包括通过所述远程机执行所述多个应用文件。

94. 根据权利要求 81 所述的系统，其中步骤 (e) 还包括通过所述远程机接收包括所述多个应用文件的应用流以加以执行。

95. 根据权利要求 81 所述的系统，其中步骤 (e) 包括响应于所述本地机包括所述至少一个特征的确定执行所述第一客户端，所述第一客户端从远程机接收包括所述多个应用文件的应用流以在所述本地机上执行。

96. 根据权利要求 81 所述的方法，其中步骤 (e) 还包括通过表示层协议接收在所述远程机上执行所述多个应用文件而产生的应用输出数据。

97. 根据权利要求 96 所述的方法，其中步骤 (e) 还包括通过独立计算结构表示层协议或远程桌面 Windows 表示层协议或 X-Windows 表示层协议接收应用输出数据。

98. 一种用于访问包括应用程序的多个文件的系统，所述系统包

括：

包括访问信息的文件，所述访问信息用于访问多个应用文件；

能够接收应用流的第一客户端，所述第一客户端接收所述文件、响应于所述文件检索多个应用文件的标识和执行所述多个应用文件所需的至少一个特征，并确定所述本地机是否包括所述至少一个特征；以及

第二客户端，其响应于所述第一客户端做出的所述本地机缺少所述至少一个特征的确定请求在远程机上执行所述多个应用文件。

99. 根据权利要求 98 所述的系统，其中所述文件还包括访问信息，所述访问信息包括包含应用程序的多个应用文件的位置的标识。

用于在应用程序的预定数量的执行方法之间
选择的方法

技术领域

本发明涉及用于执行应用程序的方法，具体而言，涉及用于在应用程序的预定数量的执行方法之间进行选择的方法。

背景技术

现代企业环境的管理员在提供对应用程序的访问时可能面对很多挑战。其中的一项挑战涉及向最终执行应用的环境提供并维护（即更新）应用的问题，因为大量的机器具有不同的执行环境，所述的不同执行环境又具有对多个公司网的不同类型的访问。第二项挑战涉及在目标机上提供某种环境，从而在不干扰其他可能存在抵触要求的应用程序的情况下实现应用程序的执行，或者使应用程序能够在并非被设计为运行所述应用程序的环境下得到执行（即，使单用户应用在多用户操作系统中隔离运行）。

大量应用程序的提供和更新给信息技术（IT）部门带来了很多困扰。一种困扰包括怎样为不同的执行环境提供应用，所述执行环境包括个人计算机（执行很多种不同类型的操作系统）、服务器、刀片机或虚拟机。另一种困扰包括如何更新已经交付的应用。对于更新频繁的企业应用而言，在不同类型的目标机上安装应用的升级版本可能即昂贵又耗时。另一项挑战是针对访问请求进行集中开发和应用的企业策略。

典型的企业环境中的另一项困扰是在采用了大量的具有不同要求的应用的环境中应用之间的兼容性。IT 部门在向支持不同执行环境的目标机部署应用程序时，以及在每一应用程序具有不同的、可能存在抵触的安装要求的情况下执行可用应用程序的不同子集时可能面临很多挑战。

确保与各种企业策略的顺应性又给典型企业环境的管理员带来了更多的困扰。针对各个应用，对目标机上的具体应用程序进行评估和授权可能是不切实际的。我们希望能够获得一种集中式系统，其能够

在安装或执行具体的应用程序之前实现目标机的自动评估，以判断与（例如）安全策略或专利使用许可的顺应性。

还希望获得这样一种方法，其能将应用一次性安装到代表性目标机上，并且够响应于具体目标机的应用访问请求将应用交付给各种目标环境。

我们还希望使策略的集中应用能够提供对应用的水平不一的访问，包括响应于应用程序的要求或本地机的特征来选择执行应用的不同方法。

发明内容

本发明涉及一种用于在应用程序的预定数量执行方法之间进行选择的方法。在一个方面中，本发明涉及一种通过远程机进行选择的方法，一种执行应用程序的方法。接收与本地机关联的信任状（credential）。响应于所接收的信任状提供本地机可用的多个应用的枚举。接收请求以执行被枚举的应用。响应于策略选择执行被枚举的应用的预定数量的方法之一，所述预定数量的方法包括用于所述被枚举的应用的应用流传输的方法。在一个实施例中，所述方法包括选择将所述被枚举的应用流传输到所述本地机的方法。在另一个实施例中，所述方法包括选择用于将被枚举的应用流传输到远程机的方法，在远程机上执行被枚举的应用以及向本地机提供在远程机上执行被枚举的应用而产生的应用输出数据。

在另一个方面中，用于在本地对文件元数据请求做出响应而无须从远程位置下载文件的方法包括：从远程机接收表示所述远程机存储的应用程序的目录结构以及与包括所存储应用程序的每个文件相关联的元数据。存储所述目录和元数据。接收至少一个访问与目录结构中特定文件相关联的元数据的请求。使用所存储的元数据以对至少一个请求做出响应。在一个实施例中，接收枚举与本地应用程序相关联的目录结构的请求。用所存储的目录结构对枚举请求做出响应。在另一个实施例中，元数据包括文件的文件名和文件的别名（alternate name）之间的映射。在另一个实施例中，利用所接收的元数据满足标识目录结构中的根节点的请求。在又一个实施例中，接收确定本地是否有包括所存储的应用程序的文件拷贝的请求。访问所存储的元数据并给出

伪装指示，表示本地没有文件拷贝。

在又一个方面中，用于在本地对文件元数据请求做出响应而无须从远程位置下载文件的系统包括目录结构、高速缓冲存储器元件和文件系统过滤器驱动程序。目录结构标识与至少一个应用程序相关联的多个文件并包括与所述多个文件中的至少一个相关联的文件元数据，所述多个文件中的至少一个驻留在远程机上。高速缓冲存储器元件存储目录结构。文件系统过滤器驱动程序拦截访问与至少一个远程存储的文件相关联的元数据的请求，访问高速缓冲存储器元件并利用所存储的目录结构对至少一个请求做出响应。

在一个实施例中，所述目录结构包括与至少一个应用程序相关联的多个文件，所述多个文件驻留在远程机上。在另一个实施例中，文件元数据包括用于对标识目录结构中的根节点的请求做出响应的信息。在又一个方面中，用于访问与应用相关联的目录结构中的文件的方法包括拦截应用访问文件的请求的步骤。将该请求重定向到第一隔离环境。做出第一隔离环境中不存在所请求文件的判断。响应于在包括驻留在远程机上的多个应用文件的目录结构枚举中标识该文件的判断，将所述请求重定向到第二隔离环境。响应于所述第二隔离环境不含所述文件且在枚举中标识该文件的判断，从文件服务器检索所请求的文件。

在一个实施例中，请求访问文件的应用是能够接收流传输的应用文件的应用。在另一个实施例中，请求访问文件的应用是本地机的用户所请求的应用。在又一个实施例中，在第一隔离环境中存储所请求的文件。在又一个实施例中，在第二隔离环境中存储所请求的文件。在一些实施例中，拦截应用访问可执行文件的请求。在其他实施例中，拦截应用访问文件的请求，在本地机上执行所述应用的一部分。

在一个实施例中，拦截由第三隔离环境中执行的另一应用提出的请求。响应于在所述枚举中枚举该文件且所述第二隔离环境不含所述文件的判断，将所述请求重定向到所述第二隔离环境。第二隔离环境为应用提供对文件的访问。

在另一个实施例中，该文件为可执行文件。在一些实施例中，多个应用文件包括第二应用。在其他实施例中，该应用是应用流传输客户端。在其他实施例中，应用流传输客户端请求并执行可执行文件以

在本地机上执行第二应用的实例。

在一个方面中，一种用于访问与应用相关联的目录结构中的文件的系统包括第一隔离环境、第二隔离环境、过滤器驱动程序和收发器。在第一隔离环境中，应用执行并请求访问文件。第二隔离环境存储与安装在远程机上的多个应用文件相关联的目录结构的枚举。过滤器驱动程序响应于在目录结构枚举中标识该文件的判断，拦截访问文件的请求，将该请求重定向到第一隔离环境，判定所请求的文件不存在于第一隔离环境中并将该请求重定向到第二隔离环境。与所述过滤器驱动程序通信的收发器，所述收发器响应于所述过滤器驱动程序做出的所述第二隔离环境不含所述文件且在所述枚举中标识该文件的判断，向文件服务器发送所述重定向的请求并响应于所述请求接收包括所请求的文件的流。

在一个方面中，一种用于由本地机访问包括应用程序的多个文件的方法包括本地机接收包括访问信息的文件的步骤，该访问信息用于访问多个应用文件并执行能够接收应用流的第一客户端。响应于所述文件检索所述多个应用文件的标识。响应于所述文件检索执行所述多个应用所需的至少一个特征。做出本地机是否包括至少一个特征的判断。响应于所述本地机缺少所述至少一个特征的判断执行第二客户端，所述第二客户端请求在远程机上执行所述多个应用文件。

在一个实施例中，响应于所检索的多个应用文件的标识检索至少一个特征。在另一个实施例中，响应于对本地机的评估做出本地机是否包括至少一个特征的判断。在又一个实施例中，本地机评估本地机。在又一个实施例中，响应于被评估的本地机缺少接收多个应用文件以在本地执行的授权这一判断，做出决定，执行第二客户端而非第一客户端。

在另一个方面中，一种用于访问包括应用程序的多个文件的系统包括文件、第一客户端和第二客户端。该文件包括用于访问多个应用文件的访问信息。能够接收应用流的第一客户端接收所述文件、响应于所述文件检索多个应用文件的标识和执行所述多个应用文件所需的至少一个特征，并判断所述本地机是否包括所述至少一个特征。第二客户端响应于所述第一客户端做出的所述本地机缺少所述至少一个特征的判断从第一客户端接收该文件并请求在远程机上执行所述多个应

用文件。

在一个实施例中，第一客户端评估本地机以判断本地机是否包括至少一个特征。在另一个实施例中，第一客户端向本地机采用策略以做出判断。在又一个实施例中，第二客户端接收在远程机上执行多个应用文件所产生的应用输出数据。在又一个实施例中，第二客户端包括在本地机上显示从远程机接收的应用输出数据的功能。

附图说明

通过下文的详细说明和附图，本发明的这些和其他方面将变得显而易见，其中，所述附图的作用在于对本发明进行举例说明，而不是限制本发明，其中：

图 1A 是示出了适于实现本发明的示范性实施例的环境的方框图；

图 1B 和 1C 是示出了可以结合本发明使用的计算机的实施例的方框图；

图 1D 是示出了利用本发明的服务器农场（server farm）的实施例的方框图；

图 1E 是示出了用于通过 web 服务目录中公布 GUI 来提供可以为本地机所使用的多个应用程序的系统的一个实施例的方框图；

图 2 是示出了选择应用程序的执行方法所采用的步骤的一个实施例的流程图；

图 3A 是示出了本地机通过万维网（World Wide Web）启动执行程序邻域应用的一个实施例的方框图；

图 3B 是示出了本地机利用 web 服务目录访问枚举的应用程序所采取的步骤的一个实施例的流程图；

图 4A 是为本地机提供对应用程序的基于策略的访问的网络实施例的方框图；

图 4B 是示出了策略引擎更详细实施例的方框图；

图 4C 是示出了策略引擎基于所接收到的关于本地机的信息做出访问控制决定所采取的步骤的一个实施例的流程图；

图 4D 是示出了其中提供对多个应用会话的授权远程访问的计算机网络实施例的方框图；

图 4E 是示出了会话服务器将本地机与其相关应用会话连接起来所

采取的步骤的一个实施例的流程图；

图 5 是示出了会话服务器将客户端节点与其相关应用会话连接起来所采取的步骤的一个实施例的流程图；

图 6 是示出了包括管理服务、提供应用枚举的远程机的一个实施例的方框图；

图 7 是示出了访问包括应用程序的多个文件所采取的步骤的一个实施例的流程图；

图 8A 是示出了在操作系统控制下运行的计算机的一个实施例的方框图，该操作系统具有应用兼容性和应用交际性降低的问题；

图 8B 是示出了具有应用兼容性和应用交际性降低的问题的多用户计算机的方框图；

图 8C 是示出了用于将过程与隔离范围相关联的方法中所采取的步骤的一个实施例的流程图；

图 9 是示出了在执行应用程序的方法中采取的步骤的一个实施例的流程图；

图 10 是示出了驻留在远程机上的多个应用文件的一个实施例的流程图；

图 11 是示出了对与远程存储的文件相关的文件元数据的请求在本地做出响应的方法中采取的步骤的一个实施例的流程图；

图 12 是示出了对与远程存储的文件相关的文件元数据的请求在本地做出响应的系统的一个实施例的方框图；

图 13 是示出了用于访问目录结构中与本地执行的应用程序相关的远程文件的方法中采取的步骤的一个实施例的流程图；

图 14 是示出了用于访问与应用相关的目录结构中的文件的系统的一个实施例的方框图；

图 15 是包括许可管理子系统的远程机的一个实施例的方框图；

图 16 是示出了远程机上的管理服务中的部件的一个实施例的方框图；

图 17 是示出了从远程机请求和维持许可所采取的步骤的一个实施例的流程图；

图 18 是示出了可能与管理服务监测的会话相关的一个实施例的状态的方框图；

图 19 是示出了包括两个目标的程序包 (package) 的方框图，每个目标包括多个包括应用的应用文件；

图 20 是示出了用于无须重新引导操作系统来安装应用程序的基于策略的方法中采取的步骤的一个实施例的流程图；

图 21 是示出了用于无须重新引导操作系统来安装应用程序的基于策略的方法中采取的步骤的一个实施例的流程图；

图 22 是示出了枚举脚本以在本地机上执行的一个实施例的屏幕快照；

图 23 是示出了包括打包机构 (packaging mechanism) 的系统实施例的方框图，该打包机构将安装程序执行到隔离环境中；

图 24 是示出了其中执行安装程序需要重新引导操作系统的环境中采取的步骤的一个实施例的流程图；

图 25 是示出了打包机构向其上安装应用程序的远程机的一个实施例的方框图；以及

图 26 是示出了在应用隔离环境中安装应用所采取的步骤的一个实施例的流程图。

具体实施方式

本发明的示范性实施例适用于分布式连网环境，其中，本地机的用户请求访问存储在远程机上的应用。在讨论本发明的细节之前，首先来讨论一下可以采用本发明的示范性实施例的一些网络环境。

图 1A 是示出了适于实现本发明的示范性实施例的环境的方框图。本地机 10 或 20 的用户能够连接至远程机，例如远程机 30、30'、30'' 或 30'''（下文统称为远程机 30）。尽管图 1A 所示的实施例中仅示出了两个本地机 10 和 20，以及四个远程机 30，但是应当理解，对于这些部件中的任何一种或每一种，所述系统都可以提供多个。例如，在一个实施例中，所述系统可以包括多个逻辑分组的远程机 30，可以采用其中的一个或多个代表本地机 10、20 执行应用。在这些实施例中，可以将远程机的逻辑组称为“服务器农场 (server farm)”，在图 1A 中将其表示为农场 38。在一些实施例中，远程机 30 在地理上可以是分散的。可以将农场 38 作为一个整体进行管理。

每一农场 38 中的远程机 30 可以是不同种类的。也就是说，远程

机 30 中的一个或多个可以按照某种类型的操作系统平台（例如，由华盛顿雷德蒙的微软公司制造的 WINDOWS NT）工作，而其他远程机 30 中的一个或多个可以根据另一种类型的操作系统平台（例如， Unix 或 Linux）工作。构成每一农场 38 的远程机 30 没有必要在其农场 38 内彼此实际相邻。因而，可以利用广域网（WAN）连接或中等区域网（MAN）连接使在逻辑上组合成农场 38 的由远程机 30 构成的组互连。例如，农场 38 可以包括实际上处于不同的州、城市、学校或房间的远程机 30。如果利用局域网（LAN）连接或某种形式的直接连接来连接远程机 30，那么可以提高农场 38 内的远程机 30 之间的数据传输速度。

可以将远程机 30 称为服务器、文件服务器、应用服务器或远程机。在一些实施例中，远程机 30 可以具有起着应用服务器或主应用服务器的作用的能力。在一个实施例中，远程机 30 可以包括现用目录（Active Directory）。也可以将本地机 10、20 称为客户端节点或端点。在一些实施例中，本地机 10、20 所具有的能力既能够使其充当寻求对应用的访问的客户端节点，又能够使其充当为其他本地机提供对其持有的（hosted）应用的访问的应用服务器。

在一个实施例中，本地机 10 与农场 38 中的远程机 30 之一直接通信。在另一实施例中，本地机 10 通过执行程序邻域（neighborhood）应用与农场 38 中的远程机 30 通信。在另一实施例中，远程机 30 提供主节点的功能性。在一些实施例中，本地机 10 通过通信链路与农场 38 中的远程机 30 通信。本地机 10 可以通过通信线路 150（例如）请求执行农场 38 中的远程机 30、30'、30" 和 30'" 所持有的各种应用，并接收所述应用的执行结果的输出，以供显示。通信链路 150 可以是同步或非同步的，并且其可以是 LAN 连接、MAN（中等区域网）连接或 WAN 连接。此外，通信链路 150 可以是无线链路，例如红外信道或卫星频带。在一些实施例中，只有主节点提供识别和提供与持有所请求的应用的远程机 30' 相关的地址信息所需的功能。

在一些实施例中，本地机 10 与远程机 30' "通信。在这些实施例之一中，远程机 30' "提供 web 服务器的功能。在这些实施例的另一个当中，远程机 30' "接收来自本地机 10 的请求，将所述请求发送至远程机 30，并采用来自远程机 30 的对所述请求的响应来回应所述的本地机 10 的请求。在这些实施例的另一个当中，远程机 30 获取本地机 10 可用

的应用的枚举和与持有所述应用枚举所标识的应用的远程机 30' 相关的地址信息。在这些实施例的另一个当中，远程机 30' "利用 web 接口向本地机 10 提供对所述请求的响应。在一个实施例中，本地机 10 与远程机 30' 直接通信，以访问所标识的应用。在另一实施例中，本地机 10 接收来自远程机 30' "的应用输出数据，所述应用输出数据是通过执行远程机 30' 上的所标识的应用而得到的。

在多个实施例中，将远程机 30 以及本地机 10 和 20 设为个人计算机或计算机服务器，其可以具有加利福尼亚库佩蒂诺的苹果计算机公司、纽约白原市的 IBM 公司、加利福尼亚帕洛阿尔托的惠普公司或者得克萨斯州 Round Rock 的戴尔公司制造的类型。在一些实施例中，远程机 30 可以是运行于诸如刀片服务器的服务器上的虚拟机。在这些实施例中，单个实际的服务器可以提供两个或更多的应用服务器。

图 1B 和 1C 示出了在这些实施例中可以被用作远程机 30 或本地机 10、20 的典型计算机 100 的方框图。如图 1B 和 1C 所示，每一计算机 100 包括中央处理单元 102 和主存储器 104。每一计算机 100 还可以包括其他任选元件，例如，一个或多个输入/输出装置 130a-130n (统一采用附图标记 130 表示) 和与中央处理单元 102 通信的高速缓冲存储器 140。

中央处理单元 102 是对从主存储器 104 中取出的指令作出响应并处理所述指令的任何逻辑电路。在多个实施例中，通过微处理器单元提供所述中央处理单元，其中，所述微处理器单元可以是加利福尼亚芒廷维尤的 Intel 公司、伊利诺斯州绍姆堡的摩托罗拉公司、纽约白原市的 IBM 公司或者加利福尼亚桑尼维尔的 AMD 公司制造的。

主存储器 104 可以是能够存储数据并且允许通过微处理器 102 直接访问任何存储位置的一个或多个存储芯片，例如，其可以是静态随机存取存储器(SRAM)、脉冲串(burst)SRAM 或同步脉冲串 SRAM(BSRAM)、动态随机存取存储器(DRAM)、快速页面式 DRAM(FPM DRAM)、增强型 DRAM(EDRAM)、扩展数据输出 RAM(EDO RAM)、扩展数据输出 DRAM(EDO DRAM)、脉冲串扩展数据输出 DRAM(BEDO DRAM)、扩展 DRAM(EDRAM)、同步 DRAM(SDRAM)、JEDEC SRAM、PC100 SDRAM、双数据率 SDRAM(DDR SDRAM)、增强型 SDRAM(ESDRAM)、同步链接 DRAM(SLDRAM)、直接存储器总线 DRAM(DRDRAM) 或铁电 RAM(FRAM)。

在图 1B 所示的实施例中，处理器 102 通过系统总线 120（在下文中将对其给出更为详细的说明）与主存储器 104 通信。图 1C 示出了计算机系统 100 的实施例，其中，处理器通过存储器端口与主存储器 104 直接通信。例如，在图 1C 中，主存储器 104 可以是 DRDRAM。

图 1B 和图 1C 示出了主处理器 102 通过次级总线与高速缓冲存储器 140 直接通信，其中有时将次级总线称为“后侧”总线。在其他实施例中，主处理器 102 利用系统总线 120 与高速缓冲存储器 140 通信。高速缓冲存储器 140 通常具有比主存储器 104 更快的响应时间，并且通常通过 SRAM、BSRAM 或 EDRAM 实现。

在图 1B 所示的实施例中，处理器 102 通过局部系统总线 120 与各种 I/O 装置 130 通信。可以采用各种总线将中央处理单元 102 连接至所述 I/O 装置 130，所述总线包括 VESA VL 总线、ISA 总线、EISA 总线、徽信道架构 (MCA) 总线、PCI 总线、PCI-X 总线、PCI-Express 总线或 NuBus。对于所述 I/O 装置为视频显示器的实施例而言，处理器 102 可以采用加速图形接口 (AGP) 与显示器通信。图 1C 示出了计算机系统 100 的实施例，其中，主处理器 102 通过 HyperTransport、Rapid I/O 或 InfiniBand 与 I/O 装置 130b 直接通信。图 1C 还示出了混合了局域总线和直接通信的实施例：处理器 102 在与 I/O 装置 130b 直接通信的同时利用局域互连总线与 I/O 装置 130a 通信。

在计算机系统 100 中可以存在各种各样的 I/O 装置 130。输入装置包括键盘、鼠标、跟踪板、跟踪球、传声器和图形输入板。输出装置包括视频显示器、扬声器、喷墨打印机、激光打印机和染料升华打印机。所述 I/O 装置还可以为计算机系统 100 提供大容量存储器，例如硬盘驱动器、用于接收诸如 3.5 寸盘、5.25 寸盘或 ZIP 盘的软盘的软盘驱动器、CD-ROM 驱动器、CD-R/RW 驱动器、DVD-ROM 驱动器、各种格式的磁带驱动器和 USB 存储装置，例如，所述 USB 存储装置可以是加利福尼亚 Los Alamitos 的 Twintech Industry 公司制造的装置的 USB Flash Drive 线或加利福尼亚库佩蒂诺的苹果计算机公司制造的装置的 iPod Shuffle 线。

在其他实施例中，I/O 装置 130 可以是系统总线 120 和外部通信总线之间的桥梁，例如，所述外部通信总线可以是 USB 总线、Apple Desktop 总线、RS-232 串联、SCSI 总线、FireWire 总线、FireWire 800

总线、以太网总线、AppleTalk 总线、吉比特以太网总线、异步传输模式总线、HIPPI 总线、超级 HIPPI 总线、SerialPlus 总线、SCI/LAMP 总线、FibreChannel 总线或者串行附加小型计算机系统接口总线。

具有图 1B 和图 1C 所示的类型的通用台式计算机通常在操作系统的控制下工作，其中，操作系统控制着任务进程和对系统资源的访问。典型的操作系统包括：华盛顿雷德蒙的微软公司制造的 MICROSOFT WINDOWS、加利福尼亚库佩蒂诺的苹果计算机公司制造的 MacOS、纽约 Armonk 的 IBM 公司制造的 OS/2 以及犹他州盐湖城的 Caldera 公司免费发布的操作系统 Linux，等等。

本地机 10 和 20 可以是任何个人计算机（例如 Macintosh 计算机或者基于处理器的计算机，例如，所述处理器可以是 286、386、486、Pentium、Pentium II、Pentium III、Pentium IV、Pentium M、Celeron 或 Xeon 处理器，所有的这些处理器都是由加利福尼亚芒廷维尤的 Intel 公司制造的）、基于 Windows 的终端、网络计算机、无线装置、信息设备、RISC、Power PC、X 装置、工作站、小型计算机、主机计算机、个人数字助理或者其他具有基于 Windows 的桌面和足够的用于执行小的直观显示程序的持久存储器（persistent store）的计算装置。直观显示程序利用通过通信信道发送给它的命令和数据实现图形显示。本地机 10 和 20 所支持的面向 Windows 的平台可以包括但不限于：Windows 3.x、Windows 95、Windows 98、Windows NT 3.51、Windows NT 4.0、Windows 2000、Windows 2003、Windows CE、Windows XP、Windows vista、MAC/OS、Java、Linux 和 UNIX。本地机 10 和 20 可以包括视觉显示装置（例如，计算机监视器）、数据输入装置（例如，键盘）、用于存储下载的应用程序的持久或易失存储器（例如，计算机存储器）、处理器和鼠标。小的直观显示程序的执行允许本地机 10 和 20 参与分布式计算机系统模型（即，基于服务器的计算模型）。

对于本地机 10 或 20 是移动装置的实施例而言，所述装置可以是能够实现 JAVA 的蜂窝电话，例如，由伊利诺斯州绍姆堡的摩托罗拉公司、日本京都的 Kyocera、韩国汉城的三星电子公司制造的蜂窝电话。在本地机 10 或 20 可移动的其他实施例中，其可以是在 PalmOS 操作系统的控制下工作的个人数字助理（PDA），例如，其可以是由加利福尼亚米尔皮塔斯的 palmOne 公司制造的装置。在其他实施例中，本地机 10

或 20 可以是在 PocketPC 操作系统的控制下工作的个人数字助理 (PDA)，例如，由加利福尼亚帕洛阿尔托的惠普公司制造的 iPAQ 装置、加利福尼亚 Walnut 的 ViewSonic 制造的装置或者纽约的美国东芝公司制造的装置。在另一些实施例中，客户端节点是 PDA/电话装置的组合，例如，由加利福尼亚米尔皮塔斯的 palmOne 公司制造的 Treo 装置。在另一些实施例中，本地机 10 或 20 是在 PocketPC 操作系统的控制下工作的蜂窝电话，例如，由摩托罗拉公司制造的蜂窝电话。

在一个实施例中，本地机 10 与农场 38 中的远程机 30 之一直接通信。在一些实施例中，本地机 10 通过通信链路 150 与农场 38 中的远程机 30 通信。本地机 10 可以通过通信线路 150 (例如) 请求执行农场 38 中的远程机 30、30'、30" 和 30'" 所持有的各种应用，并接收所述应用的执行结果的输出，以供显示。通信链路 150 可以是同步或非同步的，并且其可以是 LAN 连接、MAN (中等区域网) 连接或 WAN 连接。此外，通信链路 150 可以是无线链路，例如红外信道或卫星频带。

在一些实施例中，本地机 10 与远程机 30 通信。在这些实施例的一个当中，远程机 30 为本地机 10 提供本地机 10 能够执行的应用的枚举。在这些实施例的另一个当中，远程机 30 为本地机 10 提供与持有所述应用枚举所标识的应用的远程机 30' 相关的地址信息。在这些实施例的另一个当中，本地机 10 与远程机 30' 通信，以访问所标识的应用。在一个实施例中，本地机 10 通过执行程序邻域应用与远程机 30 和 30' 通信。在一些实施例中，每一远程机 30 提供识别和提供与持有所请求的应用的远程机 30' 相关的地址信息所需的功能。

在一些实施例中，本地机 10 与远程机 30'" 通信。在这些实施例中的一个当中，远程机 30'" 提供 web 服务器或文件服务器的功能。在所述的实施例的另一个当中，远程机 30'" 接收来自本地机 10 的请求，将所述请求发送至远程机 30， 并采用来自远程机 30 的对所述请求的响应来回应所述的本地机 10 的请求。在这些实施例的另一个当中，远程机 30 获取本地机 10 可用的应用的枚举和与提供对所述应用枚举所标识的应用程序的访问的远程机 30' 相关的地址信息。在这些实施例的另一个当中，远程机 30'" 利用 web 接口向本地机 10 提供对所述请求的响应。在一个实施例中，本地机 10 与远程机 30' 直接通信，以访问所标识的应用。在另一实施例中，本地机 10 接收来自远程机 30'" 的应用输出数

据，所述应用输出数据是通过执行远程机 30' 上的所标识的应用而得到的。

现在参考图 1D，构成农场 38 的远程机 30 均包括网络端接口 202 和农场端接口 204。远程机 30 的网络端接口可以与一个或多个本地机 10、20 或与网络 210 通信。网络 210 可以是 WAN、LAN 或者诸如因特网或万维网的国际网。本地机 10、20 可以利用 210 与远程机 30 建立连接。

远程机 30 的农场端接口 204 通过通信链路相互连接，从而使远程机 30 能够相互通信。在每一远程机 30 上，农场端接口 204 与网络端接口 202 通信。农场端接口 204 还与持久存储器 230，以及一些实施例中的动态存储器 240 通信（由箭头 220 表示）。将远程机 30、持久存储器 230 和动态存储器 240（在提供时）的组合统称为农场 38。在一些实施例中，远程机 30 与持久存储器 230 通信，另一远程机 30' 与远程机 30 通信，以访问存储在所述持久存储器内的信息。

实际上，可以将所述持久存储器 230 实现到磁盘、特大容量磁盘、廉价磁盘冗余阵列、可写光盘或者任何其他允许数据读写并且在存储装置断电后能够保持写入的数据的装置上。单个实际装置可以提供实现多个持久存储器的存储器，即，可以采用单个实际装置为一个以上的农场 38 提供持久存储器 230。持久存储器 230 保持与农场 38 中的每一远程机 30 相关的静态数据和农场 38 内的所有远程机 30 采用的全局数据。在一个实施例中，持久存储器 230 可以按照低权（Lightweight）目录访问协议（LDAP）数据模型保持远程机数据。在其他实施例中，持久存储器 230 将远程机数据存储到顺应 ODBC 的数据库中。出于本说明书的目的，“静态数据”是指不发生频繁改变的数据，即，仅在小时、天或周的基础上发生变化的数据或者根本不发生变化的数据。每一远程机采用持久存储器子系统从持久存储器 230 读取数据或向其内写入数据。

出于在物理或逻辑上的可靠性的目的，可以复制持久存储器 230 所存储的数据。例如，可以采用一组冗余的镜像盘提供物理冗余，其中，每一冗余镜像盘提供一个数据副本。在其他实施例中，可以采用标准数据库技术复制数据库自身，从而提供对所述数据库的多重复制。在其他实施例中，可以同时采用物理和逻辑复制。

可以通过各种方式实现动态存储器 240（即，所有记录表格的集合）。在一个实施例中，动态存储器 240 是集中式的；也就是说，将所有的运行时间数据存储到农场 38 中的一个远程机 30 的存储器内。所述远程机起着主网络节点的作用，农场 38 中的所有其他远程机 30 在寻求对所述运行时间数据的访问时都要与所述主网络节点通信。在另一实施例中，农场 38 中的每一远程机 30 保持动态存储器 240 的整个副本。这里，每一远程机 30 与每一其他远程机 30 通信，从而使它的动态存储器 240 的副本保持更新。

在另一实施例中，每一远程机 30 保持其自身的运行时间数据，并在试图获得来自每一其他远程机 30 的运行时间数据时与它们通信。因而，例如，试图找到本地机 10 所请求的应用程序的远程机 30 可以直接与农场 38 中的每一其他远程机 30 通信，从而找到持有所请求的应用的一个或多个远程机。对于具有大量远程机 30 的农场 38 而言，由这些实施例产生的网络流量可能变得非常沉重。其中的一个实施例通过将农场 38 中的远程机 30 的子集（通常具有两个或更多的远程机）指定为“收集点”来缓解沉重的网络流量。通常，收集点是指收集运行时间数据的远程机。每一收集点存储从农场 38 中的某一其他远程机 30 收集的运行时间数据。农场 38 中的每一远程机 30 均能够作为收集点工作，因而均能够被指定为收集点。在一个实施例中，每一收集点存储整个动态存储器 240 的副本。在另一实施例中，每一收集点存储动态存储器 240 的一部分，即，其保持具有特定数据类型的运行时间数据。可以根据一种或多种标准预先确定远程机 30 存储的数据类型。例如，远程机 30 可以基于其引导程序顺序存储不同类型的数据。或者，可以由管理员利用管理工具 140 设置远程机 30 存储的数据的类型。在这些实施例中，动态存储器 240 分布在农场 38 中的两个或更多远程机 30 当中。

未被指定为收集点的远程机 30 知晓农场 38 中被指定为收集点的远程机 30。未被指定为收集点的远程机 180 在发出和请求运行时间数据时可以与特定收集点通信。因而，收集点减轻了网络流量，因为农场 38 中的每一远程机 30 在寻求对运行时间数据的访问时都与单个收集点远程机 30 通信，而不是与每一其他远程机 30 通信。

每一远程机 30 可以作为针对一种以上的数据类型的收集点工作。

例如，远程机 30"可以作为针对许可信息和针对加载信息的收集点工作。在这些实施例中，每一收集点可以收集不同类型的运行时间数据。例如，为了对这种情况举例说明，远程机 30' "可以收集许可信息，而远程机 30"可以收集加载信息。

在一些实施例中，每一收集点存储在农场 38 中的所有远程机 30 之间共享的数据。在这些实施例中，每一特定类型的数据的收集点与农场 38 中的针对该种类型的数据的每一其他收集点交换该收集点收集的数据。因而，在完成了所述数据的交换的同时，每一收集点 30" 和 30 都拥有了相同的数据。而且，在这些实施例中，每一收集点 30 和 30" 还能够使每一其他收集点始终跟上对所述运行时间数据的更新。

浏览 (browsing) 能够使本地机 10 查看农场 38、远程机 30 和农场 38 中的应用，并访问可用信息，例如整个农场 38 中的会话。每一远程机 30 包括 ICA 浏览器子系统 260，从而为本地机 10 提供浏览功能。在本地机 10 建立与远程机 30 中的任何一个的 ICA 浏览器子系统 260 的连接之后，所述浏览器子系统将支持各种本地机请求。所述本地机请求包括：(1)枚举农场中的远程机的名称，(2)枚举农场中公布的应用的名称，(3)将远程机名称和/或应用名称解析成本地机 10 可用的远程机地址。ICA 浏览器子系统 260 还支持由运行程序邻域应用的本地机 10 所发出的请求，其根据请求为所述本地机 10 提供对该用户得到了授权的农场 38 中的应用的查看。ICA 浏览器子系统 260 将所有的上述本地机请求发送给远程机 30 中的适当的子系统。

在一个实施例中，农场 38 中的每一具有程序邻域子系统的远程机 30 可以为本地机 10 的用户提供对农场 38 中的应用的查看。所述程序邻域子系统 270 可以限制对那些所述本地机 10 的用户得到了访问授权的应用的查看。典型地，这一程序邻域服务将应用作为列表或图符组提供给用户。

程序邻域子系统 270 所提供的功能可以为两种类型的本地机所使用：(1)能够直接从本地机桌面访问功能的能够启用程序邻域的本地机，(2)能够通过运行远程机上的能够启用程序邻域的桌面来访问所述功能的不能启用程序邻域的本地机(例如，遗留本地机(legacy local machine))。

可以在建立于 ICA 虚拟信道的顶部的专用虚拟信道上进行能够启

用程序邻域的本地机和程序邻域子系统 270 之间的通信。在其他实施例中，可以利用 XML 服务实施所述通信。在这些实施例的一个当中，能够启用程序邻域的本地机与 XML 子系统通信，例如，所述 XML 子系统可以是下文结合图 6 说明的 XML 服务 516，其提供了远程机 30 上的程序邻域功能。

在一个实施例中，能够启用程序邻域的本地机不具有与带有程序邻域子系统 270 的远程机的连接。对于这一实施例而言，本地机 10 向 ICA 浏览器子系统 260 发送请求，以建立与远程机 30 的 ICA 连接，从而识别出本地机 10 可用的应用。之后，本地机 10 运行获取用户的信任状的客户端侧对话。所述信任状被 ICA 浏览器子系统 260 接收，并被发送至程序邻域子系统 270。在一个实施例中，程序邻域子系统 270 向用户管理子系统发送所述信任状，以实现对其的鉴定。用户管理子系统可以返回一组存在区分的名称，其表示所述用户所属的帐户的列表。在鉴定的同时，程序邻域子系统 270 建立程序邻域虚拟信道。这一信道一直保持打开状态，直到完成应用过滤为止。

之后，程序邻域子系统 270 从与这些帐户相关的公共应用子系统 524 请求程序邻域信息。公共应用子系统 524 从持久存储器 230 获得程序邻域信息。在接收到所述程序邻域信息的同时，程序邻域子系统 270 为所述程序邻域信息设置格式并将其通过所述程序邻域虚拟信道返回给所述本地机。之后关闭部分 ICA 连接。

对于能够启用程序邻域的本地机建立了与远程机的局部 ICA 连接的另一个例子而言，考虑选择了农场 38 的本地机 10 的用户。对农场 38 的选择将来自本地机 10 的请求发送至 ICA 浏览器子系统 260，从而建立起与所选的农场 38 的远程机 30 中的一个的 ICA 连接。ICA 浏览器子系统 260 向程序邻域子系统 270 发送选择农场 38 中的远程机 30 的请求。通过 ICA 浏览器子系统 260 识别与远程机 30 相关的地址信息，并将其返回给本地机 10。接下来，可以将本地机 10 连接至对应于所接收到的地址信息的远程机 30。

在另一实施例中，能够启用程序邻域的本地机 10 建立 ICA 连接，在所述 ICA 连接上建立程序邻域虚拟信道，只要所述 ICA 连接存在就使所述程序虚拟信道保持畅通。通过这一程序邻域虚拟信道，程序邻域子系统 270 将程序邻域信息更新发送给本地机 10。为了获得更新，

程序邻域子系统 270 预订来自公共应用子系统 524 的事件，从而允许程序邻域子系统 270 检测到所公布的应用的变化。

参考图 1E，其示出了用于通过在 web 服务目录中公布 GUI 来提供可以为本地机所使用的多个应用程序的系统架构的另一个实施例的方框图。所述系统包括本地机 10 和多个远程机 30。一个远程机 30 起着内容服务器的作用。远程机 30' 提供 web 服务器功能。远程机 30" 提供用于提供对应用文件的访问的功能，并且起着应用服务器或文件服务器的作用。本地机 10 能够通过网络 155 从内容服务器 30、web 服务器 30' 和应用服务器 30" 上下载内容。在一个实施例中，本地机 10 能够通过客户端 - 应用服务器通信信道 150 从应用服务器 30" 下载内容（例如，应用）。

在一个实施例中，本地机 10 上的 web 浏览器 11 采用加密套接字协议层（SSL）支持来实现与内容服务器 30 和/或 web 服务器 30' 的通信。SSL 是由加利福尼亚芒廷维尤的 Netscape 通信公司开发的安全协议，现在已经成为了因特网工程任务组颁布的标准。或者，可以利用其他安全协议将网浏览器 11 连接至内容服务器 30 和/或 web 服务器 30'，例如，所述安全协议可以是但不限于由加利福尼亚州的洛斯阿尔托斯的 Terisa Systems 开发的加密超级文本传输协议（SHTTP）、HTTP over SSL（HTTPS）、由华盛顿雷德蒙的微软公司开发的专用通信技术（PCT）和 IETF 颁布的传送级安全（TLS）标准。在其他实施例中，web 浏览器 11 利用不存在加密的通信协议，例如超级文本传输协议（HTTP）与服务器 30 通信。

此外，本地机 10 包括用于通过客户端 - 应用服务器通信信道 150 与应用服务器 30" 建立和交换通信的应用客户端 13。在一个实施例中，应用客户端 13 为 GUI 应用。在一些实施例中，应用客户端 13 是由佛罗里达州 Fort Lauderdale 的 Citrix Systems 公司开发的独立计算架构（ICA）客户端，在下文中又将其称为 ICA 客户端 13。应用客户端 13 的其他实施例包括由华盛顿雷德蒙的微软公司开发的远程显示协议（RDP）客户端、X-Windows 客户端 13、能够执行多媒体应用、email、Java 或 .NET 代码的客户端侧播放器、解释器或模拟器。此外，在一个实施例中，可以通过 ICA 客户端 13 在本地机 10 上显示在应用服务器 30" 上执行的应用的输出。在一些实施例中，应用客户端 13 是诸如应

用流动客户端 552 的应用客户端，在下文中将参考图 5 更详细地说明应用流动客户端 552。本地机 10 通过搜索 web 服务目录 160 来寻找 web 服务。在一个实施例中，所述搜索是人工搜索。或者，所述搜索是自动搜索。所述 web 服务目录 160 还可以提供基于服务的视图，例如，白页或黄页，从而在 web 服务目录中搜索 web 服务。在另一实施例中，web 服务目录 160 支持基于 GUI 应用的结构化服务名和服务种类的分级浏览。在一个实施例中，web 服务目录 160 在独立于内容服务器 30 的远程机上，例如，在目录服务器上运行。在其他实施例中，所述 web 服务目录 160 在多个服务器上运行。

在一些实施例中，内容服务器 30 能够通过在 web 服务目录 160 中提供信息或分析使本地机 10 能够基于所述的额外的分析或信息选择 web 服务。所述 web 服务目录 160 所能列举的服务信息的例子包括但不限于提供服务的商家的名称、服务类型、服务的文字描述、一个或多个访问点 (SAP)、网络类型、使用路径（例如，TCP 或 HTTPS）和服务质量 (QoS) 信息。此外，所述服务信息可以是客户端装置类型或用户（例如职责）的详细说明。因而，服务选择可以以一个或多个上述属性为基础。

在一个实施例中，服务类型表示本地机 10 必须用来访问所述 web 服务的编程接口。例如，所述服务类型可以表明通过接口描述语言对所述服务进行了编码，例如所述接口描述语言可以是 web 服务描述语言 (WSDL)。

所述服务访问点或 SAP 是应用的唯一地址。所述 SAP 能够使计算机系统在本地机 10 和每一远程机 30 上支持多个应用。例如，应用服务器 30" 可以支持电子邮件（即，e-mail）应用、文件传输应用和/或 GUI 应用。在一个实施例中，这些应用均具有 SAP，其在应用服务器 30" 内是唯一的。在一个实施例中，所述 SAP 是 web 或因特网地址（例如，域名系统 (DNS) 名、IP/端口或资源定位码 (URL)）。因而，在一个实施例中，SAP 将 web 服务器 30' 的地址识别成存储在 web 服务器 30' 上的应用的地址的一部分。在一些实施例中，SAP 将公布服务器插件程序 165 的地址识别成存储在 web 服务器 30' 上的应用的地址的一部分，如下文所述。在一个实施例中，所述 SAP 是来自 UDDI 注册的“接入点 (accessPoint)”。

为了准备用于在 web 服务目录 160 中公布的项目，内容服务器 30 包括 web 公布工具 170。在一个实施例中，web 公布工具 170 是一种软件模块。或者，web 发布工具 170 是另一种服务器，其可以位于内容服务器 30 的外部或内部。

在一个实施例中，网络服务器 30' 向本地机 10 提交 web 页。Web 服务器 30' 可以是能够向本地机 105 提供 web 页的任何远程机 30。在另一实施例中，web 服务器 30' 是企业信息门户（例如，企业内联网或加密的企业对企业外联网）。企业门户是公司的网站，其汇集了各种应用、数据和内容，使其个性化并将其提供给用户，同时提供用于更为有效地组织和使用信息的管理工具。在一些公司内，其门户已经采用对虚拟工作位置的基于浏览器的访问替代了常规的桌面软件。

Web 服务器 30' 还包括用于实现图形用户界面（GUI）应用的公布的公布服务器插件程序 165。更准确地说，所述公布服务器插件程序 165 将新的 web 服务项 URL 转换为 GUI 应用服务，从而能够通过 web 服务目录 160 访问 GUI。在一个实施例中，公布服务器插件程序 165 是公共网关接口（CGI）脚本，这是一种被设计为接受和返回符合 CGI 规约的数据的程序。可以通过任何程序设计语言，例如 C、Perl、Java 或 Visual Basic 编写所述程序。在另一实施例中，所述公布服务器插件程序 165 是 Java 服务器页（JSP）。利用公布服务器插件程序 165 促进远程 GUI 应用的公布，本地机 10 能够由此通过整个 GUI 接口，例如，Citrix 的 ICA 或 Microsoft 的 RDP 而并非通过编程接口或 web 页访问 web 服务。

应用服务器 30" 持有一个或多个本地机 10 可用的应用。所述应用的例子包括由雷德蒙的微软公司制造的诸如 MICROSOFT WORD 的字处理程序和诸如 MICROSOFT EXCEL 的电子表格程序、财务报表程序、客户注册程序、提供技术支持信息的程序、客户数据库程序或应用集管理程序。

在一些实施例中，在不同的网络上建立一个或多个通信链路 150。例如，客户端 - 内容服务器通信信道 150' 可以属于第一网络（例如，万维网），客户端 - web 服务器通信信道 150" 可以属于第二网络（例如，加密外联网或虚拟专用网（VPN））。

在一个实施例中，web 公布工具 170 存储与 web 公布工具 170 在持

久大容量存储器 225 中的 web 服务目录 160 内公布的应用有关的信息。在一个实施例中，所述信息是动态公布服务器插件程序 165 的 URL。持久大容量存储器 225 可以是磁盘或磁光驱动器。在一个实施例中，所述持久大容量存储器 225 为数据库服务器，其将与所公布的应用相关的数据存储到一个或多个局部服务数据库内。所述持久大容量存储器 225 可以是处于任何或所有远程机 30 之内或之外的部件。

在其他实施例中，内容服务器 30 或 web 服务器 30' 与农场 38 中的远程机通信，以检索应用列表。在这些实施例之一中，内容服务器 30 或 web 服务器 30' 与农场 38 通信，而不是和持久大容量存储器 225 通信。

现在参考图 2，流程图示出了选择应用程序的执行方法所采取的步骤的一个实施例。简而言之，接收与本地机相关联或与本地机的用户相关联的信任状，请求枚举可以由本地机执行的应用（步骤 202）。响应于所接收的信任状提供对本地机可用的多个应用程序的枚举（步骤 204）。接收请求以执行被枚举的应用（步骤 206）。响应于策略选择执行被枚举的应用的预定数量的方法之一，所述预定数量的方法包括用于被枚举的应用的应用流传输的方法（步骤 208）。

接收与本地机相关联或与本地机的用户相关联的信任状，请求枚举可以由本地机执行的应用（步骤 202）。在一个实施例中，远程机利用信任状从本地机 10 接收枚举可用应用的请求。在另一个实施例中，远程机 30 上的 XML 服务接收请求和信任状并将请求和信任状传输到远程机 30 上的管理服务。

在一些实施例中，起到 web 服务器功能的远程机 30 从本地机 10 接收通信并将通信转发到远程机 30'。在这些实施例之一中，Web 服务器将通信转发到远程机 30' 上的 XML 服务。在这些实施例的另一个中，Web 服务器位于本地机上。在由 web 服务器将来自本地机 10 的通信路由到远程机 30' 的其他实施例中，可以响应于本地机 10 的网际协议（IP）地址选择远程机 30。

在一些实施例中，本地机 10 请求访问远程机 30 上驻留的应用。在这些实施例之一中，本地机 10 请求由远程机 30 执行驻留在远程机 30 上的应用。在这些实施例的另一个中，本地机 10 请求检索包括应用的多个应用文件。

在一些实施例中，用户经由远程机 30 提供给本地机 10 的图形用户界面向远程机 30 提供信任状。在其他实施例中，具有 web 服务器功能的远程机 30'' 向本地机 10 提供图形用户界面。在其他实施例中，由远程机 30 发送给本地机 10 的收集代理收集来自本地机 10 的信任状。在一个实施例中，信任状是指用户名和口令。在另一个实施例中，信任状不限于用户名和口令，而是包括，但不限于本地机 10 的机器 ID、操作系统类型、操作系统补丁的存在、已安装网卡的 MAC 地址、客户端装置上的数字水印、活动目录中的成员身份、病毒扫描器的存在、个人防火墙的存在、HTTP 报头、浏览器类型、装置类型、诸如网际协议地址或地址范围的网络连接信息、远程机 30 的机器 ID、包括改变时区的调节的访问请求的日期或时间以及授权信任状。

在一些实施例中，与本地机相关联的信任状是与本地机的用户相关联的。在这些实施例之一中，信任状是用户所有的信息。在这些实施例的另一个中，信任状是用户身份验证信息。在其他实施例中，与本地机相关联的信任状是与网络相关联的。在这些实施例之一中，信任状是与本地机可以连接到的网络相关联的信息。在这些实施例的另一个中，信任状是与收集本地机信息的网络相关的信息。在其他实施例中，与本地机相关联的信任状是本地机的特征。

响应于所接收的信任状提供对本地机可用的多个应用程序的枚举（步骤 204）。在一个实施例中，本地机 10 的用户可能知道网络 40 中的远程机 30 所持有的应用程序的可用性，但不知道在哪里找到这种应用，并没有链接到这种应用所需的技术信息。这些可用的应用程序构成用户的“程序邻域”。用于为本地机判断程序邻域的系统包括应用程序（下文称为“程序邻域”应用）、用于存储应用程序部件的存储器以及用于执行应用程序的处理器。如下所述，可以在本地机 10 和/或远程机 30 的存储器中安装程序邻域（PN）应用。

根据程序邻域应用而运行的远程机 30 从农场 38 中的每台远程机 30 收集与应用相关的信息。用于每个所持有的应用的与应用相关的信息可以是各种信息，例如包括持有应用的远程机地址、应用名称、被授权使用应用的用户或用户组以及在建立连接之前为了运行应用本地机 10 所需的最低能力。例如，应用可能对视频数据进行流传输，因此所需的最低能力可以是本地机支持视频数据。其他实例为本地机支持

音频数据或具有处理加密数据的能力。与应用相关的信息可以存储在数据库中。

当本地机 10 连接到网络 40 时，本地机 10 的用户提供用户信任状。用户信任状可以包括本地机 10 的用户的用户名、用户的口令以及用户被授权的域名。或者，可以从智能卡、时基令牌、社会保障号码、用户口令、个人身份 (PIN) 号码、基于对称密钥或椭圆曲线密码术的数字证书、用户的计量生物学特征或任何通过其能够获得本地机 10 的用户身份并提交进行验证的其他方式获得用户信任状。对本地机 10 响应的远程机 30 可以基于用户信任状对用户进行认证。可以在执行程序邻域应用的任何地方存储用户信任状。对于本地机 10 执行程序邻域应用的实施例而言，可以在本地机 10 上存储用户信任状。对于远程机 30 执行程序邻域的实施例而言，可以在该远程机 30 上存储用户信任状。

远程机 30 从用户信任状和与应用相关的信息还可以判断出远程机 30 所持有的哪些应用程序是可以由本地机 10 的用户使用的。远程机 30 向本地机 10 发送表明可用应用程序的信息。该过程消除了本地机 10 的用户建立应用连接的必要。此外，远程机 30 的管理员可以在本地机 10 的多个用户之间控制对应用的访问。

在一些实施例中，远程机 30 执行的用户认证可以足以授权使用提供给本地机 10 的每个所持有应用程序，尽管这样的应用可能驻留在另一远程机 30 上。因此，当本地机 10 启动（即开始执行）所持有应用之一时，可以不必认证本地机 10 再次输入的用户信任状来认证应用的使用。于是，单独一项用户信任状可以用于判断可用应用并授权启动这种应用，而无须用户进行额外的人工登录认证过程。

本地机 10 或远程机 30 的任一个都可以启动程序邻域应用。在本地机 10、20 的显示屏 12、22 上显示结果。在基于图形窗口的实现中，可以在程序邻域图形窗口中显示结果，并可以在该窗口中用图形符号表示每个授权的应用程序。

在一个实施例中，程序邻域应用滤除掉未授权本地机 10 执行的应用程序，仅显示授权的（即可用的）程序。在其他实施例中，程序邻域应用可以显示授权和未授权的应用。在不滤除未授权应用而加以显示的时候，可以提供通知，表明这种应用是不可用的。或者，程序邻域应用可以向本地机 10、20 的用户报告远程机 30 所持有的所有应用，

而不标明授权或未授权本地机 10、20 执行哪些应用。随后可以在本地机 10、20 试图运行这些应用之一时决定授权情况。

本地机 10 可以请求从远程机 30 进行应用枚举。应用枚举使本地机 10 的用户能够看到每个发布的应用的名称。在一个实施例中，无论用户是否有执行应用的授权，本地机 10 的用户都可以看到应用名称。在另一个实施例中，用户仅看到授权用户执行的那些应用的名称。

根据本地机 10 运行的具体过程，应用枚举请求传送到 ICA 浏览器子系统 260、到程序邻域子系统 270 或到公共应用子系统 524。例如，当本地机 10 在运行程序邻域应用时，将应用枚举请求发送到远程机 30 上的程序邻域子系统 270。当本地机 10 通过网页提出枚举请求时，该请求传送到公共访问点子系统 524。对这些实施例而言，在本地机 10 希望枚举应用时，公共应用子系统 524 充当着程序邻域子系统 270、ICA 浏览器子系统 260 和公共应用子系统的初始访问点。在一些实施例中，当本地机 10 通过网页提出枚举请求时，托管 web 服务器的中间远程机 30 接收该请求并将请求转发到远程机 30'。

在接收到枚举请求时，公共应用子系统 524 向持久存储器 230 查询所有应用的列表。对于从程序邻域子系统 270 和公共访问点 645 子系统接收的请求，根据本地机 10 的用户的信任状对这一应用列表进行过滤（即用户仅看到用户被授权的那些应用）。

本地机 10 还可以请求远程机枚举。远程机枚举使本地机 10 的用户能够看到农场 38 中的远程机列表。在一个实施例中，可以根据远程机类型对远程机列表进行过滤，这是由该远程机上的专用远程机子系统决定的。

根据本地机 120 运行的具体过程，将远程机枚举请求传送到 ICA 浏览器子系统 260 或公共访问点子系统 645。例如，当本地机 120 通过网页提出远程机枚举请求时，该请求传送到公共访问点子系统 645。对这些实施例而言，公共远程机子系统 300 充当着 ICA 浏览器子系统 260 和公共访问点 645 子系统的初始访问点。在接收到远程机枚举请求时，公共远程机子系统从持久存储器 230 查询所有远程机列表。任选地，根据远程机类型过滤该远程机列表。

图 3A 是示出了本地机 10 启动执行程序邻域应用的过程的另一实施例的方框图，在该实例中是通过万维网启动的。本地机 10 执行网络

浏览器应用 80，例如由加利福尼亚 Mountain View 的 NETSCAPE Communications, Inc. 制造的 NETSCAPE NAVIGATOR，或者由华盛顿 Redmond 的 Microsoft Corporation 制造的 MICROSOFT INTERNET EXPLORER，或者由加利福尼亚 Mountain View 的 Mozilla Foundation 制造的 FIREFOX，或者由挪威奥斯陆的 OPERA Software ASA 制造的 OPERA，或者由加利福尼亚 Cupertino 的 Apple Computer, Inc. 制造的 SAFARI。

本地机 10 通过网络浏览器 80 发出请求 82，希望访问对应于远程机 30 上驻留的 HTML 页面的统一资源定位符 (URL) 地址。在一些实施例中，由远程机 30 向本地机 10 返回的第一个 HTML 页面 84 是视图识别本地机 10 的认证页面。

仍然参考图 3A，一旦本地机 10 获得远程机 30 的认证，远程机 30 准备并向本地机 10 发送 HTML 页面 88，该页面包括程序邻域窗口 58，该窗口中给出了表明本地机 10 已经访问过的应用程序的图形符号 57、57'。本地机 10 的用户通过点击该图标 57 来启用图标 57 表示的应用的执行。

在一些实施例中，远程机 30 代表本地机 10 的用户执行程序邻域应用。在这些实施例之一中，远程机 30 是位于本地机 10 和远程机 30' 之间的中间远程机。

参考图 3B，流程图示出了通过在 web 服务目录中的 GUI 发布提供本地机可用的多个应用程序所采取的步骤的一个实施例。Web 发布工具 170 接收要发布的应用(例如 GUI 应用)的 web 服务描述和访问信息(步骤 300)。在一个实施例中，Web 服务描述包括上述服务信息(例如，提供 web 服务的单位名称、服务类型、服务的文本描述和 SAP)。访问信息例如可以包括发布的应用名称、传输控制协议 (TCP) 浏览服务器农场地址和 MetaFrame 服务器 IP 地址。在一些实施例中，访问信息指定穿过网络或安全网关或网桥装置所用的地址和权证。

然后 web 发布工具 170 构建服务发布请求以请求发布 web 服务(例如 GUI 应用) (步骤 305)。在一个实施例中，服务发布请求包括 SAP。在一些实施例中，SAP 是一种包括 web 服务器 30' 的 web 地址和发布服务器插件 165 的 URL。此外，web 地址可以是通用资源标识符 (URI)，这是引用 web 上目标的名称和地址类型的通用术语。URL 是一种 URI。

URI 的实例为 web 服务器 30' 的名称（例如“web-server”）和用于发布服务器插件 165 的 CGI 脚本名称（例如“dynamic-component”）。

Web 发布工具 170 在持久大容量存储器 225 中存储与 SAP 相关联的 SAP 项（步骤 310）。在一些实施例中，Web 发布工具 170 还将所发布的应用信息（例如 ICA-published-app-info）与 GUI 应用关联起来。在其他实施例中，web 发布工具 170 还包括服务发布请求中的密钥，以标识内容服务器 30 在持久大容量存储器 225 中存储的 SAP 项。例如，该密钥可以具有值“123456677”。标识 web 服务器 30' 的 SAP、发布服务器插件 165 的 CGI 脚本名称和上述密钥的实例为
“http://web-server/dynamic-component/?app=123456677.”

与上述 SAP 相关联的 SAP 项的实例为“密钥=123456677，值=ICA-published-app-info”。该密钥可以是任意长度（例如 56 比特密钥，128 比特密钥）。在一个实施例中，该密钥为密码随机数。该密钥还可以提供对密钥持有者的访问权。虽然用密钥进行示例，但可以使用任何手段来为持久大容量存储器 225 中存储的 SAP 项提供一种形式的安全措施。

Web 发布工具 170 向内容服务器 30 提供服务发布请求，希望在 web 服务目录 160 中进行发布（步骤 315）。此外，在一个实施例中，内容服务器 30 向本地机 10 发送 SAP 的密钥，请求特定 web 服务，以便在随后定位 SAP 项时使用。在一个实施例中，发布服务发布请求使本地机 10 的用户能够访问该服务。在一个实施例中，利用弗罗里达 Fort Lauderdale 的 Citrix Systems, Inc. 开发的 NFUSE 在 web 服务目录 160 上发布 GUI 应用。在一些实施例中，GUI 应用的发布者利用 Application Launching And Embedding (ALE) 定制在 web 服务目录 160 上对 GUI 应用的发布，ALE 也是由 Citrix Systems, Inc. 开发的。ALE 使得能够从 HTML 页面启动 GUI 应用，或者能够将应用嵌入到 HTML 页面中。

本地机 10 然后从 web 服务目录 160 查询服务名称（步骤 320）。内容服务器 30 从本地机 10 接收查询（步骤 325）并在 web 服务目录 160 中找到所请求的服务名称。在另一个实施例中，本地机 10 的用户在 web 服务目录 160 中浏览，直到定位出本地机 10 的用户正在寻找的特定服务名称为止。虽然以本地机 10 进行了示例，其实任何 web 服务

目录客户端（例如 UDDI 客户端或 LDAP 浏览器）都可以查询或浏览 web 服务目录 160 以发现所发布的 web 服务。

在定位出与所接收的查询相关联的 SAP 时，内容服务器 30 向本地机 10 发送 SAP（步骤 330）。本地机 10 接收 SAP（步骤 335）并从该 SAP 判定发布服务器插件 165 的地址。本地机 10 随后向 web 服务器 30' 发送对 GUI 应用的请求（步骤 340）。在一些实施例中，来自本地机 10 的请求是从网络浏览器 11 传输到 web 服务器 30' 的 HTTP 请求。在其他实施例中，在本地机 10 上执行的应用（例如通常的目录浏览器或 HTML U1）从内容服务器 30 接收 SAP 并将 SAP 作为要旨提供给网络浏览器 11。网络浏览器 11 然后可以自动地向 web 服务器 30' 发送（对 GUI 应用的）HTTP 请求。根据前述实例，发给 web 服务器 30' 的应用请求的具体实例为“http://web-server/dynamic-component/?app=123456677.”。

Web 服务器 30'，尤其是发布服务器插件 165 接收与 SAP 相关的应用请求（步骤 345）并确定与该请求相关的 SAP 项（步骤 350）。在一个实施例中，发布服务器插件 165 从本地机 10 接收请求并检索与已经（作为 SAP 项的一部分）存储在持久大容量存储器 225 中的请求相关联的所发布的应用信息。在一些实施例中，发布服务器插件 165 使用本地机 10 从内容服务器 30 接收的 SAP（或 SAP 的一部分）作为密钥来访问存储在持久大容量存储器 225 中的适当服务项（例如已发布的应用信息）。

发布服务器插件 165 然后构建具有已发布的应用信息（例如应用服务器 30" 的 HTTP 地址）的文件或文档（步骤 352）并将该文档发送到本地机 10（步骤 355）。发布服务器插件 165 构建该文件，使得该文件具有与应用客户端 13 兼容的格式。在一个实施例中，该文档为多用网络邮件扩展（MIME）或安全 MIME（S/MIME）文档。在另一个实施例中，该文档为含有 ICA web 客户端嵌入对象 HTML 标记的 HTML 文档。在又一个实施例中，该文档为含有应用流传输客户端嵌入对象 HTML 标记的 HTML 文档。

网络浏览器 11 随后接收该文档并试图打开该文档。在一个实施例中，如果应用客户端 13 未安装在本地机 10 上，则本地机 10 与应用服务器 30" 通信以下载并安装该应用客户端 13。在安装应用客户端 13 时，

或者，如果应用客户端 13 已经安装在本地机 10 上了，本地机 10 启动应用客户端 13 以查看从 web 服务器 30' 接收的文档（步骤 360）。

一旦在本地机 10 上安装并执行应用客户端 13，应用服务器 30" 则执行该应用并在应用客户端 13 上显示该应用（步骤 365）。在替换实施例中，应用服务器 30" 向应用客户端 13 发送包括该应用的多个应用文件以在本地机 10 上执行，如下文参考图 7 要进一步详述的。在另一个实施例中，本地机 10 查看该文档（即使在启动应用客户端 13 之前）并使用文档中的信息来从应用服务器 30" 获得 GUI 应用。在本实施例中，GUI 应用的显示包括应用客户端 30" 的安装和执行情况。此外，文档的查看可以是对本地机 10 的用户透明的。例如，本地机 10 可以从 web 服务器 30' 接收文档并在自动从应用服务器 30" 请求 GUI 应用之前解释该文档。

于是，应用客户端 13 向已发布的应用、桌面、桌面文档和应用客户端 13 支持的任何其他应用提供基于服务的访问。应用客户端 13 能够向其提供访问的应用实例包括但不限于 WINDOWS 桌面、均由华盛顿 Redmond 的 Microsoft Corporation 开发的诸如 MICROSOFT EXCEL、WORD 和 POWERPOINT 的 WINDOWS 文档、诸如由加利福尼亚 Palo Alto 的 SUN Microsystems 开发的 SUN SOLARIS 的 Unix 桌面以及由北卡罗来纳 Durham 的 Red Hat, Inc. 销售的 GNU/Linux 等。

在一些实施例中，响应于策略引擎对本地机是否可以以及如何访问应用的判断，提供本地机 10 可用的多个应用程序的枚举（步骤 204）。策略引擎可以在做出决定之前收集有关本地机的信息。现在参考图 4A，示出了根据本发明构建的计算机网络的一个实施例，其包括本地机 10、收集代理 404、策略引擎 406、策略数据库 408、农场 38 和应用服务器 30'。在一个实施例中，策略引擎 406 为远程机 30。在另一个实施例中，应用服务器 30' 为远程机 30'。尽管在图 4A 所示的实施例中仅示出了一个本地机 10、收集代理 404、策略引擎 406、农场 38 和应用服务器 30'，但应当理解，该系统可以对这些部件中任一种或每种提供多个。

简而言之，当本地机 10 向策略引擎 406 发送访问应用的请求 410 时，收集代理 404 与本地机 10 通信，检索关于本地机 10 的信息，并将本地机信息 412 发送到策略引擎 406。策略引擎 406 通过将来自策略数据库 408 的策略施加到所接收的信息 412 上来做出访问控制决定。

更详细地讲，本地机 10 向策略引擎 406 发送对资源的请求 410。在一个实施例中，策略引擎 406 驻留在应用服务器 30' 上。在另一个实施例中，策略引擎 406 为远程机 30。在又一个实施例中，应用服务器 30' 从本地机 10 接收请求 410 并将请求 410 发送到策略引擎 406。在又一个实施例中，本地机向远程机 30''' 发送对资源的请求 410，远程机 30''' 将请求 410 发送到策略引擎 406。

在一些实施例中，本地机 10 通过网络连接发送请求 410。该网络可以是局域网（LAN）、城域网（MAN）或诸如因特网的广域网（WAN）。本地机 10 和策略引擎 406 可以通过各种连接连接到网络，各种连接包括标准电话线、LAN 或 WAN 链接（例如 T1、T3、56kb、X.25）、宽带连接（ISDN、帧中继、ATM）和无线连接。本地机 10 和策略引擎 10 之间的连接可以使用各种数据链路层通信协议（例如 TCP/IP、IPX、SPX、NetBIOS、NetBEUI、SMB、以太网、ARCNET、光纤分布式数据接口（FDDI）、RS232、IEEE 802.11、IEEE 802.11a、IEEE 802.11b、IEEE 802.11g 和直接异步连接）。该连接也可以是上述通信链路 150。

在接收到请求时，策略引擎 406 启动由收集代理 404 进行的信息收集。收集代理 404 收集有关本地机 10 的信息并将信息 412 发送到策略引擎 406。

在一些实施例中，收集代理 404 通过网络连接收集并发送信息 412。在一些实施例中，收集代理 404 包括字节码，例如以字节码程序设计语言 JAVA 编写的应用。在一些实施例中，收集代理 404 包括至少一个脚本。在这些实施例中，收集代理 404 通过在本地机 10 上运行至少一个脚本来收集信息。在一些实施例中，收集代理包括本地机 10 上的活动 X 控件。活动 X 控件是一种实现一组界面的专用部件对象模型（COM）的对象，所述界面使其看起来和运行起来像控件。

在一个实施例中，策略引擎 406 向本地机 10 发送收集代理 404。在一个实施例中，策略引擎 406 在收集代理 404 已经向策略引擎 406 发送过信息 412 之后要求第二次执行收集代理 404。在本实施例中，策略引擎 406 可能没有充分的信息 412 来判断本地机 10 是否满足特定的条件。在其他实施例中，策略引擎 406 响应于所接收的信息 412 要求多次执行收集代理 404。

在一些实施例中，策略引擎 406 向收集代理 404 发送指令，确定

收集代理 404 收集的信息类型。在这些实施例中，系统管理员可以配置从策略引擎 406 发送到收集代理 404 的指令。这对所收集的信息类型提供了更大控制。由于对所收集的信息类型更大的控制，这也扩展了策略引擎 406 能够做出的访问控制决定的类型。收集代理 404 所收集的信息 412 包括但不限于本地机 10 的机器 ID、操作系统类型、操作系统补丁的存在、已安装网卡的 MAC 地址、客户端装置上的数字水印、活动目录中的成员身份、病毒扫描器的存在、个人防火墙的存在、HTTP 报头、浏览器类型、装置类型、诸如网际协议地址或地址范围的网络连接信息、远程机 30 的机器 ID、包括改变时区的调节的访问请求的日期或时间以及授权信任状。

在一些实施例中，装置类型为个人数字助理。在其他实施例中，装置类型为蜂窝电话。在其他实施例中，装置类型为膝上电脑。在其他实施例中，装置类型为台式计算机。在其他实施例中，装置类型为因特网信息站。

在一些实施例中，数字水印包括数据嵌入。在一些实施例中，水印包括插入到文件中的图案以提供有关文件的源信息。在其他实施例中，水印包括数据散列文件以提供窜改检测。在其他实施例中，水印提供有关文件的版权信息。

在一些实施例中，网络连接信息涉及到带宽能力。在其他实施例中，网络连接信息涉及到网际协议地址。在其他实施例中，网络连接信息由网际协议地址构成。在一个实施例中，网络连接信息包括标识本地机向其提供认证信任状的登录代理的网络分区。

在一些实施例中，授权信任状包括多个类型的认证信息，包括但不限于用户名、客户端名称、客户端地址、口令、PIN、语音示例、一次性通过代码、计量生物学数据、数字证书、权证等以及其组合。在接收到所收集的信息 412 之后，策略引擎 406 基于所接收的信息 412 做出访问控制决定。

现在参考图 4B，方框图示出了策略引擎 406 的一个实施例，包括第一部件 420，第一部件包括条件数据库 422 和登录代理 424，还包括第二部件 430，第二部件包括策略数据库 432。第一部件 420 将来自条件数据库 422 的条件施加到所接收的关于本地机 10 的信息并判断所接收的信息是否满足该条件。

在一些实施例中，条件可以要求本地机 10 执行特定的操作系统以满足该条件。在一些实施例中，条件可以要求本地机 10 执行特定的操作系统补丁以满足该条件。在其他实施例中，条件可以要求本地机 10 为每个安装的网卡提供 MAC 地址以满足该条件。在一些实施例中，条件可以要求本地机 10 表明在特定活动目录中的成员身份以满足该条件。在另一个实施例中，条件可以要求本地机 10 执行病毒扫描器以满足该条件。在其他实施例中，条件可以要求本地机 10 执行个人防火墙以满足该条件。在一些实施例中，条件可以要求本地机 10 包括特定的装置类型以满足该条件。在其他实施例中，条件可以要求本地机 10 建立特定类型的网络连接以满足该条件。

如果所接收的信息满足条件，第一部件 420 在数据集 426 中存储该条件的标识符。在一个实施例中，如果该信息使条件为真，则所接收的信息满足条件。例如，条件可以要求安装特定的操作系统。如果本地机 10 具有该操作系统，则条件为真且得到满足。在另一个实施例中，如果该信息使条件为假，则所接收的信息满足条件。例如，条件可以查询本地机 10 上是否存在间谍软件。如果本地机 10 不含有间谍软件，该条件为假且得到满足。

在一些实施例中，登录代理 424 驻留在策略引擎 406 之外。在其他实施例中，登录代理 424 驻留在策略引擎 406 上。在一个实施例中，第一部件 420 包括登录代理 424，登录代理 424 启动对有关本地机 10 的信息收集。在一些实施例中，登录代理 424 还包括数据存储器。在这些实施例中，数据存储器包括收集代理可以搜集信息的条件。该数据存储器与条件数据库 422 不同。

在一些实施例中，登录代理 424 通过执行收集代理 404 启动信息搜集。在其他实施例中，登录代理 424 通过将收集代理 404 发送到本地机 10 以在本地机 10 上执行来启动信息搜集。在其他实施例中，登录代理 424 在接收到信息 412 之后启动额外的信息收集。在一个实施例中，登录代理 424 还接收信息 412。在本实施例中，登录代理 424 基于所接收的信息 412 生成数据集 426。在一些实施例中，登录代理 424 通过将来自数据库 422 的条件施加到从收集代理 404 接收的信息来生成数据集 426。

在另一个实施例中，第一部件 420 包括多个登录代理 424。在本实

施例中，多个登录代理 424 中的至少一个位于本地机 10 可以从其发送资源请求的每个网域上。在本实施例中，本地机 10 向特定的登录代理 424 发送资源请求。在一些实施例中，登录代理 424 向策略引擎 406 发送本地机 10 从其访问登录代理 424 的网域。在一个实施例中，本地机 10 从其访问登录代理 424 的网域被称为本地机 10 的网络分区。

条件数据库 422 存储第一部件 420 施加到所接收的信息上的条件。策略数据库 432 存储第二部件 430 施加到所接收的数据集 426 上的策略。在一些实施例中，条件数据库 422 和策略数据库 432 在与 ODBC 兼容的数据库中存储数据。例如，条件数据库 422 和策略数据库 432 可以被提供为由加利福尼亚 Redwood Shores 的 ORACLE Corporation 制造的 ORACLE 数据库。在其他实施例中，条件数据库 422 和策略数据库 432 可以是华盛顿 Redmond 的 Microsoft Corporation 制造的 Microsoft ACCESS 数据库或 Microsoft SQL server 数据库。

在第一部件 420 向条件数据库 422 中的每个条件施加所接收的信息之后，第一部件向第二部件 430 发送数据集 426。在一个实施例中，第一部件 420 向第二部件 430 仅发送数据集 426。因此，在本实施例中，第二部件 430 不接收信息 412，仅接收所满足的条件的标识符。第二部件 430 接收数据集 426 并通过基于数据集 426 内标识的条件施加来自策略数据库 432 的策略来做出访问控制决定。

在一个实施例中，策略数据库 432 存储被施加到所接收的信息 412 的策略。在一个实施例中，由系统管理员至少部分地指定策略数据库 432 中存储的策略。在另一个实施例中，用户执行策略数据库 432 中存储的策略中的至少一些。将用户指定的策略存储为优先项。策略数据库 432 可以存储在易失性或非易失性存储器中，或者，例如通过多个服务器分布。

在一个实施例中，仅当满足一个或多个条件时策略才允许访问资源。在另一个实施例中，策略允许访问资源但禁止将资源传输到本地机 10。另一种策略可以根据请求访问的本地机 10 位于安全网络之内而建立连接。在一些实施例中，该资源为应用程序，且本地机 10 已请求执行应用程序。在这些实施例之一中，策略可以允许在本地机 10 上执行应用程序。在这些实施例的另一个中，策略可以使本地机 10 能够接收包括应用程序的文件流。在本实施例中，可以在隔离环境中存储并

执行文件流。在这些实施例中的又一个中，策略可以仅允许在诸如应用服务器的远程机上执行应用程序，并要求远程机向本地机 10 发送应用输出数据。

现在参考图 4C，流程图示出了策略引擎 406 基于所接收的有关本地机 10 的信息做出访问控制决定所采取的步骤的一个实施例。在接收所收集的有关本地机 10 的信息时（步骤 450），策略引擎 406 基于信息生成数据集（步骤 452）。数据集 426 包含所接收的信息 412 所满足的每个条件的标识符。策略引擎 406 将策略施加到数据集 426 中的每个标识的条件上。该应用产生出本地机 10 可以访问的资源枚举（步骤 454）。策略引擎 406 然后向本地机 10 提供枚举。在一些实施例中，策略引擎 406 生成用于向本地机提供枚举的超级文本标志语言（HTML）文档。

参考图 4D，示出了根据本发明构建的网络的一个实施例，其包括本地机 10、收集代理 404、策略引擎 406、策略数据库 408、条件数据库 410、本地机 20、会话服务器 420、所存储应用数据库 422、远程机 30'、第一数据库 428、远程机 30" 和第二数据库 432。简而言之，当本地机 10 向访问控制服务器 406 发送访问应用程序的请求 412 时，收集代理 404 与本地机 10 通信，检索关于本地机 10 的信息，并将本地机信息 414 发送到策略引擎 406。如以上在图 4A 和 4B 中所述，策略引擎 406 做出访问控制决定。本地机 10 接收对与本地机 10 相关联的可用应用的枚举。

在一些实施例中，会话服务器 420 在本地机 10 和与本地机 10 相关联的多个应用会话之间建立连接。在其他实施例中，策略引擎 406 判定本地机 10 具有检索包括应用的多个应用文件并在本地执行应用程序的授权。在这些实施例之一中，远程机 30' 存储应用会话数据和包括应用程序的多个应用文件。在这些实施例的另一个中，本地机 10 与存储应用会话数据和包括应用程序的多个应用文件的远程机 30' 建立应用流传输会话。

现在参考图 4E，流程图示出了会话服务器 420 为本地机 10 提供对其相关联应用会话的访问所采取的步骤的一个实施例。会话服务器 420 从含有策略引擎 406 所做出的访问控制决定的策略引擎 406 接收有关本地机 10 的信息（步骤 480）。会话服务器 420 生成相关应用枚举（步

骤 482)。会话服务器 420 可以将本地机 10 连接到相关应用(步骤 484)。在一个实施例中，该信息还包括本地机信息 414。在另一个实施例中，该信息包括对在本地执行应用程序的授权。

会话服务器 420 生成相关应用枚举(步骤 482)。在一些实施例中，策略引擎 406 标识已经与本地机 10 相关联的多个应用会话。在其他实施例中，会话服务器 420 标识与本地机 10 相关联的已存储应用会话。在这些实施例中的一些中，会话服务器 420 在从策略引擎 406 接收信息时自动标识所存储的应用会话。在一个实施例中，所存储应用数据库 422 驻留在会话服务器 420 上。在另一个实施例中，所存储应用数据库 422 驻留在策略引擎 406 上。

所存储应用数据库 422 含有与农场 38 中执行应用会话或提供对应用会话数据和包括应用程序的应用文件的访问的多个远程机相关的数据。在一些实施例中，识别与本地机 10 相关联的应用会话需要查询与一个或多个远程机相关联的所存储数据。在这些实施例中的一些中，会话存储器 420 查询与一个或多个远程机相关联的所存储数据。在这些实施例中的其他实施例中，策略引擎 406 查询与一个或多个远程机相关联的所存储数据。在一些实施例中，第一应用会话在远程机 30' 上运行，第二应用会话在远程机 30" 上运行。在其他实施例中，所有的应用会话都在农场 38 之内的单个远程机 30 上运行。

会话服务器 420 包括与用户发起的应用会话相关的信息。会话服务器可以存储在易失性或非易失性存储器中，或者，例如通过多个服务器分布。表 1 示出了在示例性会话服务器 420 的一部分中包括的数据：

应用会话	应用会话 1	应用会话 2	应用会话 3
用户 ID	用户 1	用户 2	用户 1
客户端 ID	第一客户端		第一客户端
客户端地址	172.16.0.50		172.16.0.50
状态	活动	断开连接	活动
应用	文字处理器	数据库	电子表格
过程数	1	3	2
服务器	服务器 A	服务器 A	服务器 B
服务器地址	172.16.2.55	172.16.2.55	172.16.2.55

表 1

表 1 中的示例性会话服务器 420 包括将每个应用会话与发起该应用会话的用户相关联的数据、当前用户从其连接到远程机 30' 的客户端计算机 10 或 20 的身份（如果有的话）、以及客户端计算机 10 或 20 的 IP 地址。示例性会话服务器 420 还包括每个应用会话的状态。例如，应用会话状态可以是“活动”（表示用户连接到应用会话）或“断开连接”（表示用户未连接到应用会话）。在替换实施例中，也可以将应用会话状态设置成“执行中-断开连接”（表示用户已经从应用会话断开连接，但应用会话中的应用仍在执行）或“停机-断开连接”（表示用户断开连接且应用会话中的应用不在执行中，但它们的运行状态恰在存储断开连接之前）。会话服务器 420 还存储表示在每个应用会话中执行的应用 116 的信息以及表示服务器上每个应用的过程的数据。在远程机 30' 为农场 38 的部分的实施例中，会话服务器 420 是动态存储器的至少一部分，并且还包括表 1 中最后两行中的数据，该数据表明每个应用正在/曾经在农场 38 中哪个远程机 30 执行以及远程机 30 的 IP 地址。在替换实施例中，会话服务器 420 包括用于每个应用会话中的每个应用的状态指示符。

例如，在表 1 的实例中，存在三个应用会话，即应用会话 1、应用会话 2 和应用会话 3。应用会话 1 与当前正使用终端 1 的用户 1 相关。终端 1 的 IP 地址为 152.16.2.50。应用会话 1 的状态为活动，在应用会话 1 中正在执行文字处理程序。在服务器 A 上正在执行作为过程 1 的文字处理程序。服务器 A 的 IP 地址为 152.16.2.55。表 1 中的应用会话 2 是断开连接的应用会话 118 的实例。应用会话 2 与用户 2 相关，但应用会话 2 未连接到本地机 10 或 20。应用会话 2 包括在 IP 地址 152.16.2.55 处在服务器 A 上作为过程 3 执行的数据库程序。应用会话 3 是用户如何能够与运行在不同远程机 30 上的应用会话交互的实例。如应用会话 1 那样，应用会话 3 与用户 1 相关。应用会话 3 包括在 IP 地址 152.16.2.56 处在服务器 B 上作为过程 2 执行的电子表格程序，而应用会话 1 中包括的应用会话是在服务器 A 上执行的。

在另一个实例中，用户可以通过在诸如服务器 A 的远程机 30' 上执行的应用会话访问第一应用程序，同时跨过应用流传输会话与诸如服务器 B 的第二远程机 30" 通信，以从第二远程机 30" 检索第二应用程序

以在本地执行。本地机 10 的用户在无法满足第一应用程序的执行条件的同时，可能已经获取了执行第二应用程序的授权。

在一个实施例中，将会话服务器 420 配置成接收断开请求，以断开与本地机 10 相关联的应用会话并响应于该请求断开该应用会话。在从应用会话断开本地机 10 之后，会话服务器 420 继续执行应用会话。在本实施例中，会话服务器 420 访问所存储应用数据库 422 并更新与每个断开连接的应用会话相关联的数据记录，使得该记录表明与本地机 10 相关联的应用会话已断开。

在接受到与连接到网络的本地机相关联的认证信息之后，会话服务器 420 查询所存储应用数据库 422 以查明是否有任何如下的活动应用会话，该活动应用会话与本地机的用户相关联，但是如果该认证信息例如与本地机 20 相关联，该活动应用会话与不同的本地机，例如本地机 10 相连接。在一个实施例中，如果会话服务器 420 发现了任何这种活动应用会话，会话服务器 420 自动从本地机 10 断开该应用会话并将应用会话连接到当前本地机 20。在一些实施例中，所接收的认证信息将会限制本地机 10 可以重新连接到的应用会话。在其他实施例中，在可能以将对本地机 10 否决授权的情况下，所接收的认证信息授权在本地机 20 上执行应用程序。在这些实施例之一中，会话服务器 420 可以提供本地机访问信息以检索应用程序从而在本地执行。

接收请求以执行被枚举的应用（步骤 206）。在一个实施例中，本地机 10 的用户从所接收的可用应用枚举中选择要执行的应用。在另一个实施例中，用户独立于所接收的枚举选择要执行的应用。在一些实施例中，用户通过选择客户端代理在本地机 10 上提供的应用图形表示来选择要执行的应用。在其他实施例中，用户通过选择 web 服务器或其他远程机 30'’ 上提供给用户的应用图解表示来选择要执行的应用。

在其他实施例中，用户请求访问文件。在这些实施例之一中，需要执行应用以为用户提供对文件的访问。在这些实施例的另一个中，在选择要访问的文件时自动选择要执行的应用。在这些实施例中的又一个中，在请求访问文件之前，将应用与一种文件类型相关联，从而能够在识别与所请求文件相关联的文件类型时自动选择应用。

在一个实施例中，被枚举的应用包括多个应用文件。在一些实施例中，多个应用文件驻留在远程机 30' 上。在其他实施例中，多个应用

文件驻留在独立的文件服务器或远程机 30" 上。在其他实施例中，可以将多个应用文件传输到本地机 10。

在其他实施例中，可以在向本地机 10 传输多个应用文件中的第二文件之前执行多个应用文件中的文件。在一些实施例中，远程机 30 从远程机 30' 检索与被枚举的应用相关的信息。在这些实施例之一中，远程机 30 接收持有多个应用文件的远程机 30" 的标识。在这些实施例的另一个中，远程机 30 接收多个应用文件的位置的标识，该标识符合通用命名标准 (UNC)。在这些实施例中的又一个中，该标识包括网络位置和应用流传输协议的套接字。

在一个实施例中，远程机 30 检索含有有关被枚举的应用的信息的文件。该文件可以包括持有被枚举的应用的服务器位置的标识。该文件可以包括持有被枚举的应用的多个版本的标识。该文件可以包括多个应用文件的枚举，该枚举包括被枚举的应用。该文件可以包括压缩文件的标识，该压缩文件包括多个应用文件，该多个应用文件包括被枚举的应用。该文件可以包括执行被枚举的应用的机器要满足的前提条件的标识。该文件可以包括与被枚举的应用相关联的数据文件的枚举。该文件可以包括要在执行被枚举的应用的机器上执行的脚本的枚举。该文件可以包括与被枚举的应用相关联的注册数据的枚举。该文件可以包括在隔离环境之中执行被枚举的应用的实施例中使用的规则的枚举。在一个实施例中，该文件可以被称为“清单”文件。结合以下图 21 更详细地描述该文件可以含有的信息。

在一些实施例中，远程机 30 向本地机 10 的被标识特征应用策略。在这些实施例之一中，远程机 30 响应于被标识的特征标识要执行的被枚举的应用版本。在这些实施例的另一个中，远程机 30 做出决定，执行与本地机 10 的特征兼容的被枚举的应用版本。在这些实施例中的又一个中，远程机 30 做出决定，执行与本地机 10 上执行的操作系统兼容的被枚举的应用版本。在又一个实施例中，远程机 30 做出决定，执行与本地机 10 上的操作系统的修订级 (revision level) 兼容的被枚举的应用版本。在这些实施例之一中，远程机 30 做出决定，执行与本地机 10 上的操作系统指定的语言兼容的被枚举的应用版本。

响应于策略选择执行被枚举的应用的预定数量的方法之一，所述预定数量的方法包括用于被枚举的应用的应用流传输的方法（步骤

208)。在一个实施例中，响应于向所接收的与本地机 10 相关的信任状应用策略来做出选择。在一些实施例中，通过如以上在图 4A、图 4B 和图 4C 所述的策略引擎 406 的策略引擎做出选择。在其他实施例中，接收信任状和执行被枚举的应用的请求的远程机 30 还包括这样的策略引擎 406。

在一个实施例中，预定数量的方法包括在远程机 30' 上执行被枚举的应用的方法。在另一个实施例中，预定数量的方法包括在本地机 10 上执行被枚举的应用的方法。在又一个实施例中，预定数量的方法包括在第二远程机 30' 上执行被枚举的应用的方法。

在一些实施例中，预定数量的方法包括跨越应用流传输会话向本地机 10 提供被枚举的应用的方法。在这些实施例之一中，本地机 10 包括流传输服务代理，该流传输服务代理能够发起与远程机 30' 的连接并从远程机 30' 接收所发送的数据包流。

数据包流可以包括包括被枚举的应用的应用文件。在一些实施例中，应用文件包括与应用程序相关联的数据文件。在其他实施例中，应用文件包括执行应用程序所需的可执行文件。在其他实施例中，应用文件包括元数据，元数据包括与文件相关的信息，例如位置、兼容性要求、配置数据、注册数据、在隔离环境中使用的执行脚本规则标识或授权要求。

在一些实施例中，经流传输的应用在传输包括流传输应用的多个应用文件中的每个应用文件之前执行。在这些实施例之一中，在本地机 10 接收到多个应用中的一个应用文件时开始执行经流传输的应用。在这些实施例的另一个中，在本地机 10 接收到多个应用文件中的可执行应用文件时开始执行经流传输的应用。在这些实施例中的又一个中，本地机 10 执行多个应用文件中第一个接收的应用文件，该第一个接收的应用文件请求访问多个应用文件中的第二应用文件。

在一个实施例中，流传输的应用在本地机 10 上执行，而不永久驻留在本地机 10 上。在本实施例中，流传输的应用可以在本地机 10 上执行并在流传输的应用终止时从本地机 10 被删除。在另一个实施例中，在将每个应用文件的预展开拷贝存储在本地机 10 上之后在本地机 10 上执行流传输的应用。在一实施例中，在将每个应用文件的拷贝存储在本地机上的隔离环境中之后，在本地机 10 上执行流传输的应用。

在又一个实施例中，在将每个应用文件的拷贝存储在本地机 10 上的高速缓存中之后在本地机 10 上执行流传输的应用。

在一个实施例中，响应于本地机 10 可能接收到流传输的应用文件的判决，从预定数量的方法中选择将应用流传输到本地机 10 的方法。在另一个实施例中，响应于本地机 10 已经具有在本地执行流传输的应用文件授权的判决，从预定数量的方法中选择将应用流传输到本地机 10 的方法。

在其他实施例中，预定数量的方法包括向本地机 10 提供应用输出数据的方法，所述应用输出数据是在远程机 30 上执行被枚举的应用而生成的。在这些实施例之一中，远程机 30 是接收执行被枚举的应用的请求的远程机 30。在这些实施例中的另一个中，远程机 30 为诸如文件服务器或应用服务器的第二远程机 30'。在一些实施例中，被枚举的应用驻留在执行被枚举的应用的远程机 30' 上。在其他实施例中，执行被枚举的应用的远程机 30' 首先跨越应用流传输会话从第二远程机 30' 接收被枚举的应用。在这些实施例之一中，远程机 30' 包括流传输服务代理，该流传输服务代理能够发起与第二远程机 30' 的连接并从第二远程机 30' 接收所发送的数据流。在这些实施例的另一个中，可以利用负载均衡技术标识第二远程机 30'。在这些实施例的又一个中，可以基于与远程机 30' 的接近情况标识第二远程机 30'。将结合以下的图 9 更详细地描述这些实施例。

在一些实施例中，远程机 30 从执行被枚举的应用的预定数量的方法中选择用于将被枚举的应用流传输到远程机 30、在远程机 30 上执行被枚举的应用并向本地机 10 提供执行被枚举的应用而产生的应用输出数据的方法。在这些实施例之一中，远程机 30 响应于本地机 10 的评估而选择方法。在这些实施例的另一个中，响应于将策略应用于评估本地机 10 来做出判决。在这些实施例的又一个中，响应于对所接收的信任状的评估做出判决。在一个实施例中，远程机 30 接收包括被枚举的应用的多个应用文件。在另一个实施例中，远程机 30 通过诸如 ICA 表示层协议或远程桌面窗口表示层协议或 X-窗口表示层协议的表示层协议提供应用输出数据。

在一些实施例中，远程机 30 还提供与被枚举的应用相关联的访问信息，该访问信息是响应于选定的方法而产生的。在这些实施例之一

中，访问信息向本地机 10 提供所选定的执行被枚举的应用程序的方法的表示。在这些实施例的另一个中，访问信息包括被枚举的应用的位置标识，该标识符合通用命名标准 (UNC)。在这些实施例的又一个中，访问信息包括会话管理服务器的标识。

在一些实施例中，访问信息包括启动权证 (launch ticket)，启动权证包括认证信息。在这些实施例之一中，本地机 10 可以使用该启动权证来认证从远程机 30 接收的访问信息。在这些实施例的另一个中，本地机 10 可以使用启动权证来向持有被枚举的应用的第二远程机 30 认证它自身。在这些实施例的又一个中，远程机 30 响应于本地机 10 对启动权证的请求在访问信息中包括启动权证。

现在参考图 5，方框图示出了本发明的一个实施例，其中，本地机 10 请求执行应用程序，而远程机 30 选择执行应用程序的方法。在一个实施例中，远程机 30 从本地机 10 接收信任状。在另一个实施例中，远程机 30 从本地机 10 接收对可用应用枚举的请求。

在一些实施例中，提供了多个冗余的远程机 30、30'、30''、30''' 和 30''''。在这些实施例之一中，例如，可以有多个文件服务器、多个会话管理服务器、多个分级机、多个 web 接口或多个访问套件控制台。在这些实施例的另一个中，如果远程机出现故障，则选择冗余的远程机 30 来提供出故障机器的功能。在其他实施例中，尽管将远程机 30、30'、30''、30''' 和 30'''' 以及 web 接口 558 和访问套件控制台 520 描述为具有管理服务器、会话管理服务器、分级机、文件服务器、web 服务器和访问套件控制台的独立功能，但可以提供具有所有这些机器的功能的单个远程机 30。在其他实施例中，远程机 30 可以提供其他远程机之一或多个的功能和服务。

现在更详细地参考图 5，方框图示出了提供对应用程序的访问的远程机 30 的一个实施例。除了上文参考图 1D 所述的接口和子系统之外，远程机 30 还可以包括管理通信服务 514、XML 服务 516 和管理服务 504。管理服务 504 可以包括应用管理子系统 506、服务器管理子系统 508、会话管理子系统 510 和许可管理子系统 512。远程机 30 可以与访问套件控制台 520 通信。

在一个实施例中，管理服务 504 还包括专用远程过程调用子系统、MetaFrame 远程过程调用 (MFRPC) 子系统 522。在一些实施例中，MFRPC

子系统 522 在远程机 30 上诸如 XML 服务 516 和管理服务 504 的子系统之间对通信进行路由。在其他实施例中，MFRPC 子系统 522 提供用于调用管理功能的远程过程调用（RPC）接口，提供对管理服务 504 的 RPC 调用并向发起调用的子系统返回结果。

在一些实施例中，远程机 30 与诸如以上图 4B 所述的协议机 406 的协议机通信。在这些实施例之一中，远程机 30 与驻留在远程机 30' 上的协议机 406 通信。在其他实施例中，远程机 30 还包括协议机 406。

远程机 30 可以与访问套件控制台 520 通信。访问套件控制台 520 可以持有管理工具，供远程机 30 或农场 38 的管理员使用。在一些实施例中，远程机 30 利用 XML 与访问套件控制台 520 通信。在其他实施例中，远程机 30 利用简单对象访问协议（SOAP）与访问套件控制台 520 通信。

对于诸如图 1D 和图 5 中所述的那些远程机 30 包括子系统子集的实施例，管理服务可以包括多个子系统。在一个实施例中，每个子系统或者是单线程的或者是多线程的子系统。线程是多任务环境中运行的独立执行流。单线程子系统能够一次仅执行一个线程。多线程子系统能够支持多个并行执行的线程，即，多线程子系统可以同时执行多个任务。

应用管理子系统 506 关于与能够被流传输的多个应用相关的信息。在一个实施例中，应用管理子系统 506 处理来自其他部件的请求，例如存储、删除、更新、枚举或解析应用的请求。在另一个实施例中，应用管理子系统 506 处理由与能够被流传输的应用相关的部件发送的请求。可以将这些事件分成三种事件：应用发布、应用枚举和应用启动，以下将详细描述每种。在其他实施例中，应用管理子系统 506 还包括对应用解析、应用公开和应用发布的支持。在其他实施例中，应用管理子系统 506 使用数据暂存器存储应用性质和策略。

服务器管理子系统 508 处理对于在服务器农场配置中的应用流传输特定的配置。在一些实施例中，服务器管理子系统 508 还处理需要检索与农场 38 的配置相关的信息的事件。在其他实施例中，服务器管理子系统 508 处理由与跨应用流传输提供对应用的访问的远程机相关的其他部件发出的事件和这些远程机的性质。在一个实施例中，服务器管理子系统 508 存储远程机性质和农场性质。

在一些实施例中，远程机 30 还包括为一个或多个专用应用子系统提供服务的一个或多个公共应用子系统 524。这些远程机 30 也可以具有一个或多个为一个或多个专用远程机子系统提供服务的公共远程机子系统。在其他实施例中，不提供公共应用子系统 524，每个专用应用和远程机子系统都实施所有所需的功能。

在远程机 30 包括公共应用子系统 524 的一个实施例中，公共应用子系统 524 管理所发布应用的公共性质。在一些实施例中，公共应用子系统 524 处理需要检索与所发布应用或公共性质相关的信息的事件。在其他实施例中，公共应用子系统 524 处理由与公共应用或它们的性质相关的其他部件发送的所有事件。

公共应用子系统 524 可以向农场 38 “发布” 应用，这使得本地机 10 可以枚举和启动每个应用。通常，在希望能使用该应用的每个远程机 30 上安装该应用。在一个实施例中，为了发布应用，管理员运行管理工具，执行信息，例如持有该应用的远程机 30、每个远程机上的可执行文件名称、要执行应用的本地机所需的能力（例如音频、视频、加密等）以及能使用该应用的用户列表。将这种执行的信息分成应用特定信息和公用信息。应用特定信息的实例为：访问该应用的路径名以及用于运行应用的可执行文件的名称。公用信息（即公共应用数据）例如包括应用的用户友好的名称（例如“Microsoft WORD 2000”）、应用的唯一标识和应用的用户。

可以将应用特定信息和公用信息发送到控制持有应用的每个远程机 30 上的应用的专用应用子系统。专用应用子系统可以将应用特定信息和公用信息写入持久存储器 240 中。

在提供公共应用子系统 524 时，公共应用子系统 524 也提供了用于管理农场 38 中所发布应用的设施。通过公共应用子系统 524，管理员能够利用诸如访问套件控制台 520 的管理工具来管理农场 38 的应用，以配置应用组并产生这些应用组的应用树形体系。每个应用组可以表示成应用树形体系中的文件夹。应用树形体系中的每个应用文件夹可以包括一个或多个其他应用文件夹和远程机的特定实例。公共应用子系统 524 提供了生成、移动、重命名、删除和枚举应用文件夹的功能。

在一个实施例中，公共应用子系统 524 在处理应用枚举和应用解

析请求时支持应用管理子系统 506。在一些实施例中，公共应用子系统 524 提供了响应于数据文件类型和处理该数据文件类型的应用之间的映射而识别要执行的应用的功能。在其他实施例中，第二应用子系统提供了文件类型关联的功能。

在一些实施例中，远程机 30 还可以包括策略子系统。策略子系统包括用于判断是否可以依据本地机 10 要执行应用的请求而将应用流传输到本地机 10 的策略规则。在一些实施例中，策略子系统标识出与访问套件控制台 520 中发布的流传输的应用相关联的服务器访问选项。在这些实施例之一中，策略子系统将服务器访问选项用作代替策略规则的策略。

会话监测子系统 510 维持和更新与本地机 10 相关联的应用流传输会话的会话状态并执行应用流传输会话的许可要求。在一个实施例中，会话管理子系统 510 监测会话并记录事件日志，所述事件例如是应用的启动或应用流传输会话的终止。在另一个实施例中，会话监测子系统 510 接收从本地机 10 向远程机 30 发送的通信，例如心跳消息。在又一个实施例中，会话管理子系统 510 对来自管理工具，例如访问套件控制台 520 中的工具的有关会话的查询做出响应。在一些实施例中，管理服务 504 还包括与会话管理子系统通信的许可管理子系统，以提供并维持对本地机执行应用的许可。

在一个实施例中，管理服务 504 提供用于应用枚举和应用解析的功能。在一些实施例中，管理服务 504 还提供用于应用启动、会话监测和跟踪、应用发布和许可执行的功能。

现在参考图 6，方框图示出了包括提供应用枚举的管理服务的远程机 30 的一个实施例。管理服务 504 可以利用与 XML 服务 516 交互的 web 接口提供应用枚举。在一个实施例中，XML 服务 516 为本地机 10 的用户枚举应用。在另一个实施例中，XML 服务 516 实施上述 ICA 浏览器子系统和程序邻域子系统的功能。XML 服务 516 可以与管理通信服务 514 交互。在一个实施例中，XML 服务 516 利用管理通信服务 514 生成应用枚举请求。应用枚举请求可以包括表明在执行被枚举的应用时使用的执行方法的客户端类型。应用枚举请求被发送到公共应用子系统 524。在一个实施例中，公共应用子系统 524 返回与应用枚举请求的客户端类型相关联的应用的枚举。在另一个实施例中，公共应用子系统 524

返回本地机 10 的用户可用的应用枚举，该枚举是响应于向与本地机 10 相关的信任状应用策略选择的。在本实施例中，策略引擎 406 可以向收集代理 404 收集的信任状应用策略，如以上参考图 4 所述。在又一个实施例中，返回应用枚举并推迟向本地机 10 应用策略，直到请求执行被枚举的应用为止。

管理服务 504 可以提供应用解析服务，用于识别持有应用的第二远程机 30'。在一个实施例中，第二远程机 30' 是文件服务器或应用服务器。在一些实施例中，管理服务 504 从包括标识符的文件查询持有应用的多个远程机 30。在一个实施例中，管理服务 504 响应于本地机 10 执行应用的请求提供应用解析服务。在另一个实施例中，管理服务 504 识别出能够实施与第一远程机 30 不同的执行应用的方法的第二远程机 30'。在一些实施例中，管理服务 504 识别出能够将应用程序流传输到本地机 10 的第一远程机 30' 以及能够执行应用程序并向本地机 10 提供执行应用程序所产生的应用输出数据的第二远程机 30'。

在一个实施例中，Web 接口向 XML 服务 516 发送应用解析请求。在另一个实施例中，XML 服务 516 接收应用解析请求并将请求发送到 MFRPC 子系统 522。

在一个实施例中，MFRPC 子系统 522 识别出所接收的应用解析请求中包括的客户端类型。在另一个实施例中，MFRPC 子系统向客户端类型应用策略并决定向本地机 10 “流传输” 该应用。在本实施例中，MFRPC 子系统 522 可以向应用管理子系统 506 转发应用解析请求。在一个实施例中，在从 MFRPC 子系统 522 接收到应用解析请求时，应用管理子系统 506 可以识别出作为本地机 10 的会话管理服务器 562 工作的远程机 30''''。

在一些实施例中，本地机向会话管理服务器 562 发送心跳消息。在另一个实施例中，应用管理子系统 506 可以识别出持有包括要流传输到本地机 10 的应用的多个应用文件的远程机 30'。在一些实施例中，应用管理子系统 506 使用枚举了多个持有多个应用文件的远程机的文件来识别远程机 30'。在其他实施例中，应用管理子系统 506 识别出 IP 地址与本地机 10 的 IP 地址类似的远程机 30'。

在其他实施例中，应用管理子系统 506 识别出 IP 地址在本地机 10 可访问的 IP 地址范围内的远程机 30'。在又一个实施例中，MFRPC 子

系统 522 向客户端类型应用策略并判定可以在远程机 30' 上执行该应用，该远程机 30' 向本地机 10 发送执行应用产生的应用输出数据。在本实施例中，MFRPC 子系统 522 可以向公共应用子系统 524 转发应用解析请求以检索远程机 30' 的宿主机地址的标识符。在一个实施例中，被识别的远程机 30' 可以利用诸如 ICA 或 RDP 或 X 窗口的表示层协议向本地机发送应用输出数据。在一些实施例中，远程机 30' 跨越应用流传输会话从第二远程机 30' 接收应用。

在一个实施例中，在完成应用枚举和应用解析时，将访问信息发送到本地机 10，该访问信息包括用于执行被枚举的应用的方法的标识和持有被枚举的应用的远程机 30' 的标识符。在管理服务 504 确定将要在本地机 10 上执行被枚举的应用的一个实施例中，web 接口创建含有有关被枚举的应用的解析域名信息的文件并将其发送到本地机 10。在一些实施例中，可以利用 “.rad” 扩展名标识该文件。本地机 10 可以响应于所接收文件的内容执行被枚举的应用。表 2 示出了该文件中所含信息的一个实施例：

字段	描述	源
UNC 路径	指向文件服务器上的容器主清单文件	XML 服务
初始程序	从容器启动的程序	XML 服务
命令行	利用 FTA 启动文档	XML 服务
Web 服务器 URL	用于从 RADE 客户端到 WI 的消息	WI 配置
农场 ID	应用所属的农场 - 心跳消息所需	WI 配置
启动权证	应用流传输客户端使用启动权证获取执行程序的许可授权	XML / IMA
ICA 低效启动信息	如果允许低效率运行，为用于低效率运行的嵌入式 ICA 文件	XML 服务

表 2

该文件还含有本地机执行应用时所用的启动权证，如表 2 所述。在一些实施例中，启动权证在预定一段时间之后期满。在一个实施例中，本地机向持有要执行的被枚举应用的远程机提供启动权证。使用

启动权证授权本地机的用户访问被枚举的应用有助于防止用户重复使用该文件或者生成该文件未授权版本以不适当当地访问应用。在一个实施例中，该启动权证包括大的、随机生成的数字。

如以上参考图 2 所述，在接收到与本地机 10 或本地机 10 的用户相关的信任状时开始选择执行应用程序的方法的方法（步骤 202），并响应于所接收的信任状提供本地机 10 可用的多个应用程序的枚举（步骤 204）。接收请求来执行被枚举的应用（步骤 206）并响应于策略选择执行被枚举的应用的预定数量的方法之一，所述预定数量的方法包括用于被枚举的应用的应用流传输的方法（步骤 208）。

现在参考图 7，流程图示出了访问包括应用程序多个文件所采取的步骤的一个实施例。本地机执行本地机的启动前分析（步骤 210）。在一个实施例中，本地机 10 在检索和执行包括应用程序的多个应用文件之前执行启动前分析。在另一个实施例中，本地机 10 响应于所接收的指示执行启动前分析，该指示表明为了授权访问包括应用程序的多个应用文件需要进行启动前分析。

在一些实施例中，本地机 10 从远程机 30 接收与多个应用文件相关联的访问信息。在这些实施例之一中，访问信息包括持有多个应用文件的远程机 30' 的位置的标识。在这些实施例的另一个中，本地机 10 接收包括应用程序的一个或多个版本的多个应用的标识。在这些实施例的又一个中，本地机 10 接收包括一个或多个应用程序的多个应用文件的标识。在其他实施例中，本地机 10 接收本地机 10 可用的应用程序枚举，以进行检索和执行。在这些实施例之一中，枚举来源于对本地机 10 的评估。在其他实施例中，本地机 10 响应于所检索的包括应用程序的多个应用文件的标识，检索至少一个特征。

在一些实施例中，访问信息包括能够授权本地机访问多个应用文件的启动权证。在这些实施例之一中，响应于对本地机 10 的评估向本地机 10 提供启动权证。在这些实施例的另一个中，在本地机 10 进行本地机 10 的启动前分析之后向本地机 10 提供启动权证。

在其他实施例中，本地机 10 检索用于执行多个应用文件所需的至少一个特征。在这些实施例之一中，访问信息包括至少一个特征。在这些实施例的另一个中，访问信息表明了本地机 10 检索的文件位置，该文件枚举了至少一个特征。在这些实施例的又一个中，枚举至少一

个特征的该文件还包括多个应用文件的枚举和持有多个应用文件的远程机 30 的标识。

本地机 10 判断本地机上至少一个特征的存在。在一个实施例中，本地机 10 作为启动前分析的一部分执行该判断。在另一个实施例中，本地机 10 判断本地机 10 是否具有至少一个特征。

在一个实施例中，判断本地机 10 上是否存在至少一个特征包括判断是否在本地机上安装了设备驱动程序。在另一个实施例中，判断本地机 10 上是否存在至少一个特征包括判断是否在本地机 10 上安装了操作系统。在一实施例中，判断本地机 10 上是否存在至少一个特征包括判断是否在本地机 10 上安装了特定的操作系统。在又一个实施例中，判断本地机 10 上是否存在至少一个特征包括判断是否在本地机 10 上安装了特定版本水平的操作系统。

在一些实施例中，判断本地机 10 上是否存在至少一个特征包括判断本地机 10 是否已经获得了执行被枚举的应用的授权。在这些实施例之一中，由本地机 10 判断本地机 10 是否已接收到执行被枚举的应用的许可。在这些实施例的另一个中，由本地机 10 判断本地机 10 是否已接收到跨越应用流传输会话接收包括被枚举的应用的多个应用文件的许可。在其他实施例中，判断本地机 10 上是否存在至少一个特征包括判断本地机 10 是否具有检索和执行被枚举应用的足够大的可用带宽。

在一些实施例中，判断本地机 10 上是否存在至少一个特征包括在本地机 10 上执行脚本。在其他实施例中，判断本地机 10 上是否存在至少一个特征包括在本地机 10 上安装软件。在其他实施例中，判断本地机 10 上是否存在至少一个特征包括在本地机 10 上修改注册表。在一实施例中，判断本地机 10 上是否存在至少一个特征包括向本地机 10 传输收集代理 404 以在本地机 10 上执行，用来收集与本地机 10 相关的信任状。

本地机 10 从远程机 30 请求执行多个应用文件的授权，该请求包括启动权证（步骤 212）。在一些实施例中，本地机 10 响应于本地机 10 上是否存在至少一个特征的判断做出请求。在这些实施例之一中，本地机 10 判定本地机 10 上存在多个特征，该多个特征与被枚举的应用相关并是响应于执行被枚举应用的请求而接收的。在这些实施例的

另一个中，本地机 10 是否接收到执行被枚举的应用文件的授权的指示取决于本地机 10 上是否存在至少一个特征。

在一个实施例中，本地机 10 接收应用程序的枚举，请求执行被枚举的应用并接收访问信息，该访问信息包括至少一个特征和启动权证，该启动权证授权在判定本地机 10 上存在至少一个特征时执行被枚举的应用。在一个实施例中，本地机 10 从远程机 30 接收授权执行多个应用文件的许可。在一些实施例中，该许可授权在预定时间期间内执行。在这些实施例之一中，该许可要求传输心跳消息，以维持执行多个应用文件的授权。

在另一个实施例中，本地机 10 从远程机 30 接收许可以及与监测多个应用文件的执行的远程机 30 相关的标识符。在一些实施例中，如以上图 5 所示，远程机是会话管理服务器 562。在这些实施例之一中，会话管理服务器 562 包括监测与本地机 10 相关的会话的会话管理子系统 510。在其他实施例中，独立的远程机 30'/'为会话管理服务器 562。

本地机 10 接收并执行多个应用文件（步骤 214）。在一个实施例中，本地机 10 跨越应用流传输会话接收多个应用文件。在另一个实施例中，本地机 10 在本地机 10 上的隔离环境中存储多个应用文件。在又一个实施例中，本地机 10 在接收多个应用文件中的第二个之前执行多个应用文件之一。在一些实施例中，远程机向多个本地机发送多个应用文件，多个本地机中的每个本地机已经与远程机建立了独立的应用流传输会话。

在一些实施例中，本地机 10 在高速缓存中存储多个应用文件并延迟执行应用文件。在这些实施例之一中，本地机 10 接收授权以在预定时间段期间执行应用文件。在这些实施例的另一个中，当本地机 10 没有对网络访问时，本地机 10 接收授权以在预定时间段内执行应用文件。在其他实施例中，本地机在高速缓存中存储多个应用文件。在这些实施例之一中，应用流传输客户端 552 建立内部应用流传输会话以从高速缓存检索多个应用文件。在这些实施例的另一个中，当本地机 10 没有对网络访问时，本地机 10 接收授权以在预定时间段内执行应用文件。

本地机 10 向远程机发送至少一个心跳消息（步骤 216）。在一些实施例中，本地机 10 发送至少一个心跳消息以维持授权，以执行包括被枚举的应用的多个应用文件。在其他实施例中，本地机 10 发送至少

一个心跳消息以维持授权，以检索多个应用文件中的应用文件。在其他实施例中，本地机 10 接收许可，该许可授权在预定时间段中执行多个应用文件。

在一些实施例中，本地机 10 向第二远程机 30'/'可以发送心跳消息。在这些实施例之一中，第二远程机 30'/'可以包括监测多个应用文件的检索和执行的会话管理服务器 562。在这些实施例的另一个中，第二远程机 30'/'可以响应于所发送的心跳消息更新授权执行多个应用文件的许可。在这些实施例的又一个中，第二远程机 30'/'可以响应于所发送的心跳消息向本地机 10 发送命令。

返回到图 5，本地机 10 可以包括应用流传输客户端 552、流传输服务 554 和隔离环境 556。

应用流传输客户端 552 可以是可执行程序。在一些实施例中，应用流传输客户端 552 可以能启动另一可执行程序。在其他实施例中，应用流传输客户端 552 可以发起流传输服务 554。在这些实施例之一中，应用流传输客户端 552 可以为流传输服务 554 提供与执行应用程序相关的参数。在这些实施例的另一个中，应用流传输客户端 552 可以利用远程过程调用发起流传输服务 554。

在一个实施例中，本地机 10 请求执行应用程序并从远程机 30 接收关于执行的访问信息。在另一个实施例中，应用流传输客户端 552 接收该访问信息。在又一个实施例中，应用流传输客户端 552 向流传输服务 554 提供访问信息。在又一个实施例中，访问信息包括与包括应用程序的多个应用文件相关的文件位置的标识。

在一个实施例中，流传输服务 554 检索与多个应用文件相关的文件。在一些实施例中，被检索的文件包括多个应用文件的位置标识。在这些实施例之一中，流传输服务 554 检索多个应用文件。在这些实施例的另一个中，流传输服务 554 在本地机 10 上执行所检索的多个应用文件。在其他实施例中，流传输服务 554 向远程机发送心跳消息以保持对检索和执行多个应用文件的授权。

在一些实施例中，所检索的文件包括超过一个多个应用文件的位置标识，每一多个应用文件包括不同的应用程序。在这些实施例之一中，流传输服务 554 检索包括与本地机 10 兼容的应用程序的多个应用文件。在这些实施例的另一个中，流传输服务 554 响应于对本地机 10

的评估接收对检索特定多个应用文件的授权。

在一些实施例中，在诸如 CAB、ZIP、SIT、TAR、JAR 的档案文件或其他档案文件中压缩多个应用文件并将其存在文件服务器上。在一个实施例中，存储在档案文件中的多个应用文件包括应用程序。在另一个实施例中，存储在档案文件中的多个应用文件的每个包括应用程序的不同版本。在又一个实施例中，存储在档案文件中的多个应用文件的每个包括不同的应用程序。在一些实施例中，档案文件包括与多个应用文件中的每个文件相关联的元数据。在这些实施例之一中，流传输服务 554 响应于所包括的元数据产生目录结构。如下文将要参考图 12 更详细描述的，可以用元数据来满足应用程序对目录枚举的请求。

在一个实施例中，流传输服务 554 对档案文件解压以获取多个应用文件。在另一个实施例中，流传输服务 554 在从多个应用文件检索文件之前判断本地机 10 上的高速缓冲存储器中是否存在多个应用文件之内的文件的本地拷贝。在又一个实施例中，文件系统过滤器驱动程序 564 判断高速缓冲存储器中是否存在本地拷贝。在一些实施例中，流传输服务 554 在检索多个应用文件之内的文件之前修改注册项。

在一些实施例中，流传输服务 554 在本地机 10 上的高速缓冲存储器中存储多个应用文件。在这些实施例之一中，流传输服务 554 可以提供在接收到高速缓存多个应用文件的请求时高速缓存多个应用文件的功能。在这些实施例的另一个中，流传输服务 554 可以提供在本地机 10 上固定高速缓冲存储器的功能。在这些实施例的另一个中，流传输服务 554 可以使用算法来调节高速缓冲存储器的大小和位置。

在一些实施例中，流传输服务 554 在本地机 10 上生成隔离环境 556。在这些实施例之一中，流传输服务 554 使用隔离环境应用编程接口生成隔离环境 556。在这些实施例的另一个中，流传输服务 554 在隔离环境 556 中存储多个应用文件。在这些实施例的又一个中，流传输服务 554 在隔离环境之内执行多个应用文件中的文件。在这些实施例中的又一个中，流传输服务 554 在隔离环境中执行应用程序。

对于接收到在本地机 10 上执行应用的授权的实施例而言，可以在隔离环境 556 中发生应用的执行。在一些实施例中，在执行应用之前在本地机 10 上存储包括应用的多个应用文件。在其他实施例中，在执

行应用之前在本地机 10 上存储多个应用文件的子集。在其他实施例中，多个应用文件不驻留在隔离环境 556 中。在其他实施例中，多个应用的子集不驻留在本地机 10 上。无论多个应用文件的子集或多个应用文件中的每个应用文件是否驻留在本地机 10 上或隔离环境 556 中，在一些实施例中，都可以在隔离环境 556 中执行多个应用文件中的应用文件。

隔离环境 556 可以由核心系统构成，该核心系统能够提供文件系统虚拟化、注册系统虚拟化和命名对象虚拟化，以减少应用兼容性问题，而无须向应用源代码的任何变化。隔离环境 556 在用于注册和命名对象虚拟化的用户模式中以及在使用文件系统过滤器驱动程序进行文件系统虚拟化的内核程序中都可以利用转播来重定向应用资源请求。下文是对隔离环境 556 的一些实施例的描述。

现在参考图 8A，示出了在具有较少应用兼容性和应用交际性问题的操作系统 8100 的控制下运行的计算机的一个实施例。操作系统 8100 通过其系统层 8108 使应用程序 8112、8114 能够使用各种本地资源。将把系统层 8108 收录的资源视图称为“系统范围”。为了避免应用程序 8112、8114 对本地资源 8102、8104、8106、8107 的冲突访问，提供隔离环境 8200。如图 8A 所示，隔离环境 8200 包括应用隔离层 8220 和用户隔离层 8240。从概念上讲，隔离环境 8200 通过应用隔离层 8220 为应用程序 8112、8114 提供诸如文件系统 8102、注册表 8104、对象 8106 和窗口名称 8107 的本地资源的独特视图。每个隔离层修改提供给应用的本地资源视图。将把层提供的修改后的本地资源视图称为该层的“隔离范围”。如图 8A 所示，应用隔离层包括两个应用隔离范围 8222、8224。范围 8222 表示提供给应用 8112 的本地资源视图，范围 8224 表示提供给应用 8114 的本地资源视图。于是，在图 8A 所示的实施例中，为 APP1 8112 提供文件系统 8102' 的特定视图，而为 APP2 8114 提供对其特有的文件系统 8102" 的另一视图。在一些实施例中，应用隔离层 8220 向在操作系统 8100 上执行的每一个体应用程序提供本地资源 8102、8104、8106、8107 的特定视图。在其他实施例中，可以将应用程序 8112、8114 分成组，在这些实施例中，应用隔离层 8220 提供每组应用程序的本地资源的特定视图。可以将冲突的应用程序放入分开的组中以提高应用的兼容性和交际性。在其他实施例中，可以由管理

员配置属于一组的应用。在一些实施例中，可以将“穿过”隔离范围定义为精确对应于系统范围。换言之，在穿过隔离范围之内执行的应用直接在系统范围上运行。

在一些实施例中，进一步将应用隔离范围分成分层子范围。主子范围包含基本应用隔离范围，额外的子范围包含可以由应用的多个执行实例看得见的对该范围的各种修改。例如，子范围可以包含对范围的修改，该修改体现出应用补丁水平的改变或额外特征的安装或移除。在一些实施例中，被使得对执行应用的实例可见的额外子范围的组是可以配置的。在一些实施例中，可见子范围的组对于执行应用的所有实例而言都是相同的，不论是代表哪个用户执行应用的。在其他实施例中，该组可见子范围可以相对于执行应用的不同用户而变化。在其他实施例中，可以定义多组子范围，用户对于使用哪组可以具有选择机会。在一些实施例中，在不再需要时可以抛弃子范围。在一些实施例中，可以将一组子范围中包含的修改合并到一起来形成单个子范围。

现在参考图 8B，示出了具有减少的应用兼容性和应用交际性问题的多用户计算机。多用户计算机包括系统层 8108 中的本地资源 8102、8104、8106、8107 以及上文刚刚讨论过的隔离环境 8200。应用隔离层 8220 如上所述运行，为应用或应用组提供本地资源的修改视图。从概念上讲，用户隔离层 8240 为应用程序 8112、8114 提供本地资源视图，基于代表其执行应用的用户身份进一步改变该本地资源视图。如图 8B 所示，用户隔离层 8240 可以被认为包括若干用户隔离范围 8242'、8242''、8242'''、8242''''、8242'''''（统称为 8242）。用户隔离范围 8242 提供本地资源的应用特异性视图的用户特异性视图。例如，为代表用户“a”在用户会话 8110 中执行的 APP1 8112 提供由用户隔离范围 8242' 和应用隔离范围 8222 二者改变或修改的文件系统视图 8102' (a)。

换言之，用户隔离层 8240 通过在应用隔离范围 8222 提供的应用特异性视图修改上“层叠”用户隔离范围 8242''' 提供的用户特异性视图修改（该应用特异性视图修改又被层叠在系统层提供的本地资源的系统范围视图之上）来为每个个体用户改变本地资源视图。例如，当 APP1 8112 的第一实例访问注册表数据库 8104 中的条目时，查询对第一用户会话和应用 8104' (a) 特定的注册表数据库视图。如果在注册表

8104' (a) 的用户特异性视图中找到了所请求的注册表项，则将该注册表项返回到 APP1 8112。如果没有找到，则查询对应用 8104' 特定的注册表数据库视图。如果在注册表 8104' 的应用特异性视图中找到了所请求的注册表项，则将该注册表项返回到 APP1 8112。如果未找到，则将系统层 8108 中的注册表数据库 8104 中存储的注册表项（即本地注册表项）返回到 APP1 8112。

在一些实施例中，用户隔离层 8240 为每一个体用户提供隔离范围。在其他实施例中，用户隔离层 8240 为一组用户提供隔离范围，该组用户可以由组织内的分工界定或可以由管理员预先确定。在其他实施例中，不提供用户隔离层 8240。在这些实施例中，应用程序看到的本地资源视图是由应用隔离层 8220 提供的视图。虽然针对支持多个用户同时执行应用程序的多用户计算机描述了隔离环境 8200，但隔离环境 8200 也可以用在单用户计算机上，以解决因不同用户在同一计算机系统上顺次执行应用程序而造成的应用兼容性和交际性问题以及因同一用户安装和执行不兼容程序而造成的那些问题。

在一些实施例中，进一步将用户隔离范围分成子范围。用户隔离范围指向在该范围内执行的应用提供的视图所做的修改是该范围内每个子范围之内所含的修改汇集。将子范围彼此层叠，在汇集视图中，对较高子范围中的资源所做的修改优先于对较低层中同一资源的修改。

在这些实施例中的一些中，这些子范围中的一个或多个可以包含对相对于用户特定的视图的修改。在这些实施例中的一些中，一个或多个子范围可以包含对相对于用户组特定的视图的修改，用户组可以由系统管理员定义或被定义为操作系统中的用户组。在这些实施例中的一些中，这些子范围之一可以包含对相对于特定登录会话特定并因此在会话结束时被抛弃的视图的修改。在这些实施例中的一些中，与用户隔离范围相关联的应用实例对本地资源所做的改变总是影响这些子范围之一，在其他实施例中，根据所改变的特定资源，这些改变可能会影响不同的子范围。

上述概念性体系允许为代表用户执行的应用提供本地资源的汇集、或统一的虚拟化视图，这取决于应用和用户的组合。可以将这种汇集视图称为“虚拟范围”。为代表用户执行的应用实例提供反应本

地资源的所有有效虚拟化实例的单个本地资源视图。从概念上讲，这种汇集视图首先由系统范围中的操作系统提供的本地资源组构成，该本地资源组被适用于执行应用的应用隔离范围中所体现的修改覆盖，进一步被适用于代表用户执行应用的用户隔离范围中体现的修改覆盖。系统范围中的本地资源的特征在于由系统上的所有用户和应用共有，除非在操作系统许可拒绝对特定用户或应用访问的情况下。对应用隔离范围中体现的资源视图的修改的特征在于由和该应用隔离范围相关联的所有应用实例共有。对用户隔离范围中体现的资源视图的修改的特征在于，由和适用的应用隔离范围相关联的、代表与该用户隔离范围相关联的用户执行的所有应用共有。

可以将该概念扩展到子范围；对用户子范围内体现的资源视图的修改由与适用隔离子范围相关联的、代表与用户隔离子范围相关联的用户或用户组执行的所有应用共有。在整个说明书中，应当理解，无论何时总括性提到“范围”，也意在是指子范围（如果它们存在的话）。

当应用请求枚举诸如文件系统或注册表数据库一部分的本地资源时，首先枚举本地资源”系统范围的”实例，即在系统层中发现的实例（如果说有的话），由此构建虚拟化枚举。接下来，枚举所请求资源“应用范围的”实例，即在适当应用隔离范围内发现的实例（如果说有的话）。将在应用隔离范围内遇到的任何被枚举资源添加到该视图。如果在视图中已经存在所枚举的资源（因为它还存在于系统范围内），则用在应用隔离范围内遇到的资源实例替换其。类似地，枚举所请求资源“用户范围的”实例，即在适当用户隔离范围内发现的实例（如果说有的话）。同样，将在用户隔离范围内遇到的任何被枚举资源添加到该视图。如果在视图中已经存在该本地资源（因为它存在于系统范围或适当的应用隔离范围内），则用在用户隔离范围内遇到的资源实例替换其。通过这种方式，本地资源的任何枚举都将适当地反应所枚举本地资源的虚拟化。从概念上讲，同样的方法也适用于枚举包括多个子范围的隔离范围。枚举个体子范围，汇集视图中来自较高子范围的资源替代来自较低子范围的匹配实例。

在其他实施例中，可以从用户隔离范围层向下到系统层来进行枚举，而不是相反的情况。在这些实施例中，枚举用户隔离范围。然后枚举应用隔离范围，并将出现在用户隔离范围内未枚举的应用隔离范

围中的任何资源实例添加到构建中的汇集视图。可以对仅出现在系统范围中的资源重复类似过程。

在其他实施例中，可以同时枚举所有隔离范围并组合各枚举。

如果应用试图打开本地资源的现存实例而无意修改该资源，被返回到该应用的特定实例是在虚拟范围中找到的那个，或者相当地，是出现在所请求资源双亲的虚拟化枚举中的实例。从隔离环境的角度来看，将该应用说成请求打开“虚拟资源”，将用于满足该请求的本地资源的特定实例说成是对应于所请求资源的“字面资源”。

如果代表用户执行的应用试图打开资源并表明它正在这么做且有意修改该资源，一般为该应用实例赋予要修改资源的私用拷贝，因为应用隔离范围和系统范围中的资源是由代表其他用户执行的应用所共有的。除非用户范围的实例已经存在，通常制作资源的用户范围的拷贝。由虚拟范围提供的汇集视图的定义表示向用户隔离范围拷贝应用范围的或系统范围的资源这一动作不会改变虚拟范围为相关用户和应用提供的汇集视图，也不会改变为任何其他用户提供的汇集视图，也不会改变为任何其他应用实例提供的汇集视图。代表用户执行的应用实例随后对所拷贝的资源所做的修改不会影响不共享相同用户隔离范围的任何其他应用实例的汇集视图。换言之，这些修改不会改变用于其他用户，或用于与同一应用隔离范围不相关的应用实例的本地资源的汇集视图。

可以将应用安装到特定隔离范围之内（下文更详细描述）。被安装到隔离范围内的应用总是与该范围相关。或者，可以将应用启动到特定隔离范围中，或者到若干隔离范围中。实际上，应用被启动到一个或多个隔离范围中且与一个或多个隔离范围相关联。相关联的隔离范围为该过程提供特定的本地资源视图。也可以将应用启动到系统范围中，即它们可以不与隔离范围相关联。这允许在隔离环境中优先选择地执行诸如 Internet Explorer 的操作系统应用以及第三方应用。

这种不论应用安装在哪里都可以将应用启动到隔离范围中的能力减轻了应用兼容性和交际性问题，而无需在隔离范围之中分开安装应用。在不同隔离范围中有选择地启动所安装的应用的能力提供了如下能力，即，使需要帮助应用的应用（例如 Word、Notepad 等）以相同的规则集启动这些帮助应用。

此外，在多个隔离的环境中启动应用的能力允许在隔离的应用和公共应用之间更好地进行集成。

现在参考图 8C，简而言之，将过程与隔离范围相关联的方法包括启动挂起状态下的过程的步骤（步骤 882）。检索与期望的隔离范围相关联的规则（步骤 884），在存储元件中存储过程标识符和所检索的规则（步骤 886）并继续进行挂起的过程（步骤 888）。截取或转播该过程做出的访问本地资源的后续调用（步骤 890），并用与过程标识符相关联的规则（如果有的话）虚拟化对所请求资源的访问（步骤 892）。

仍然参考图 8C，更详细地讲，在挂起状态下启动过程（步骤 882）。在一些实施例中，用自定义启动程序来完成该任务。在这些实施例中的一些中，启动程序被特定设计成将过程启动到选定的隔离范围中。在其他实施例中，启动程序例如通过命令行选项接收期望的隔离范围的规格说明来作为输入。

检索与期望的隔离范围相关联的规则（步骤 884）。在一些实施例中，从持久性存储元件，例如硬盘驱动器或其他固态存储元件中检索规则。可以将规则存储成关系型数据库、平面文件数据库、树形结构化数据库、二叉树结构或其他不变数据结构。在其他实施例中，可以将规则存储在特别为存储它们而配置的数据结构中。

在存储元件中存储过程标识符，例如过程 ID (PID) 和所检索的规则（步骤 886）。在一些实施例中，提供内核模式驱动程序，其接收涉及到新过程创建的操作系统消息。在这些实施例中，可以在驱动程序的语境中存储 PID 和所检索的规则。在其他实施例中，提供文件系统过滤器驱动程序或小型过滤器来拦截本地资源请求。在这些实施例中，可以在过滤器中存储 PID 和所检索的规则。在又一些其他实施例中，通过用户模式转播进行所有拦截而且不存储任何 PID。在过程初始化期间由用户模式转播设备加载规则，没有任何其他部件需要知道应用于 PID 的规则，因为是在整个过程中进行规则关联的。

继续进行挂起的过程（步骤 888），拦截或转播该过程做出的访问本地资源的后续调用（步骤 890），并用与过程标识符相关联的规则（如果有的话）虚拟化对所请求资源的访问（步骤 892）。在一些实施例中，文件系统过滤器驱动程序或小型过滤器或文件系统驱动程序拦截访问本地资源的请求并判断与被拦截请求相关联的过程标识符是否已经与

一组规则相关联。如果是这样的话，用与所存储的过程标识符相关联的规则来虚拟化访问本地资源的请求。否则，不加修改通过该访问本地资源的请求。在其他实施例中，将动态链接库加载到新创建的过程中，该库加载隔离规则。在其他实施例中，将内核模式技术（转播、过滤器驱动程序、小型过滤器）和用户模式技术都用于拦截访问本地资源的调用。对于文件系统过滤器驱动程序存储规则的实施例而言，该库可以从文件系统过滤器驱动程序加载规则。

作为与隔离范围相关的过程的“子女”的过程与它们“双亲”过程的隔离范围相关。在一些实施例中，这是由创建子过程时内核模式驱动程序通知文件系统过滤器驱动程序而实现的。在这些实施例中，文件系统过滤器驱动程序判断父过程的过程标识符是否与隔离范围相关联。如果是这样的话，文件系统过滤器驱动程序存储新建子过程的过程标识符与父过程的隔离范围之间的关联。在其他实施例中，可以直接从系统调用文件系统过滤器驱动程序，而无须使用内核模式驱动程序。在其他实施例中，在与隔离范围相关联的过程中，转播或拦截创建新过程的操作系统功能。当从这种过程接收到创建新过程的请求时，存储新子过程和双亲隔离范围之间的关联。

在一些实施例中，可以将范围或子范围与单个线程而不是整个过程相关联，这允许在每个线程基础上执行隔离。在一些实施例中，每个线程的隔离可以用于服务和 COM+服务器。

在一些实施例中，用隔离环境向应用流传输客户端 552 提供额外的功能。在这些实施例之一中，在隔离环境之内执行应用程序。在这些实施例的另一个中，检索到的多个应用文件驻留在隔离环境中。在这些实施例的又一个中，在隔离环境中对本地机 10 上的注册表做出更改。

在一个实施例中，应用流传输客户端 552 包括隔离环境 556。在一些实施例中，应用流传输客户端 552 包括拦截对文件的应用请求的文件系统过滤器驱动程序 564。在这些实施例之一中，文件系统过滤器驱动程序 564 拦截打开现有文件的应用请求并判定该文件不在该隔离环境 556 中。在这些实施例的另一个中，文件系统过滤器驱动程序 564 响应于该文件不在隔离环境 556 中的判断向流传输服务 554 重定向该请求。流传输服务 554 可以从多个应用文件提取文件并将文件存储在

隔离环境 556 中。文件系统过滤器驱动程序 564 然后可以利用所存储的文件拷贝向对该文件的请求做出响应。在一些实施例中，该文件系统过滤器驱动程序 564 可以响应于流传输服务 554 未检索到该文件或多个应用文件的指示以及该文件不在隔离环境 556 中的判断将对文件的请求重定向到文件服务器 540。

在一些实施例中，文件系统过滤器驱动程序 564 使用严格的隔离规则来防止隔离环境 556 中出现冲突或不一致的数据。在这些实施例之一中，在用户隔离环境中拦截对资源的请求的文件系统过滤器驱动程序 564 可以将请求重定向到应用隔离环境。在这些实施例的另一个中，文件系统过滤器驱动程序 564 不将请求重定向到系统范围。

在一个实施例中，流传输服务 554 使用 IOCTL 命令与过滤器驱动程序通信。在另一个实施例中，利用 Microsoft SMB 流传输协议接收发往文件服务器 540 的通信。

在一些实施例中，打包机构 530 在清单文件中存储被发布成可用应用的文件类型列表并使应用发布软件可以使用该信息。在这些实施例之一中，打包机构 530 通过监测在分级机上的隔离环境中安装应用程序来接收该信息。在这些实施例的另一个中，打包机构 530 的用户向打包机构 530 提供该信息。在其他实施例中，访问套件控制台 520 中的应用发布软件查询该清单文件以向访问套件控制台 520 的用户提供可以与所请求的被发布应用相关联的可能的文件类型。用户选择文件类型以与特定的被发布应用相关联。在应用枚举的时候向本地机 10 提供该文件类型。

本地机 10 可以包括客户端代理 560。客户端代理 560 提供如下功能：将文件类型与应用程序相关联并响应于该关联选择执行应用程序的方法。在一个实施例中，客户端代理 560 为程序邻域应用。

在选择应用程序以执行时，本地机 10 对与应用程序文件类型相关联的执行方法做出判断。在一个实施例中，本地机 10 判定该文件类型与需要检索应用文件的应用流传输会话并需要在隔离环境之内执行的执行方法相关联。在本实施例中，本地机 10 可以将该请求重定向到应用流传输客户端 552，而不是启动应用程序的本地版本。在另一个实施例中，客户端代理 560 做出判断。在又一个实施例中，客户端代理 560 将该请求重定向到应用流传输客户端 552。

在一个实施例中，应用流传输客户端 552 从远程机 30 请求与应用程序相关联的访问信息。在一些实施例中，应用流传输客户端 552 接收含有访问信息的可执行程序。在这些实施例之一中，应用流传输客户端 552 接收能够在本地机 10 上显示因在远程机上执行应用程序而生成的应用输出数据的可执行程序。在这些实施例的另一个中，应用流传输客户端 552 接收能够通过应用流传输会话接收应用程序并在本地机 10 上的隔离环境中执行应用程序的可执行程序。在本实施例中，应用流传输客户端 552 可以执行所接收的可执行程序。在这些实施例的又一个中，远程机 30 响应于如上所述执行应用解析来选择可执行程序以提供到本地机 10。

现在参考图 9，流程图示出了在执行应用的方法中采取的步骤的一个实施例。如以上在图 7 中所述，对步骤 214 而言，本地机 10 接收并执行多个应用文件。简而言之，本地机 10 接收包括访问信息的文件，该访问信息用于访问多个应用文件并执行能够接收应用流的第一客户端（步骤 902）。本地机 10 响应于该文件检索多个应用文件的标识（步骤 904）。本地机 10 响应于该文件检索执行多个应用文件所需的至少一个特征（步骤 906）。本地机 10 判断本地机 10 是否包括至少一个特征（步骤 908）。本地机 10 响应于本地机 10 缺少至少一个特征的判断执行第二客户端，第二客户端请求在远程机上执行多个应用文件（步骤 910）。

参考图 9，并且更详细地讲，本地机 10 接收包括访问信息的文件，该访问信息用于访问多个应用文件并执行能够接收应用流的第一客户端（步骤 902）。在一个实施例中，本地机 10 接收访问信息，该访问信息包括包含应用程序的多个应用文件位置的标识。在另一个实施例中，本地机 10 响应于请求执行应用程序而接收该文件。在又一个实施例中，访问信息包括表明多个应用文件驻留在诸如应用服务器或文件服务器的远程机 30' 上的指示。在又一个实施例中，访问信息表明本地机 10 可以通过应用流传输会话从远程机 30 检索多个应用文件。

本地机 10 响应于该文件检索多个应用文件的标识（步骤 904）。在一个实施例中，本地机 10 响应于包括访问信息的文件识别多个应用文件所驻留的远程机。在另一个实施例中，本地机 10 从远程机 30 检索标识多个应用文件的文件。在一些实施例中，多个应用文件包括

T/US2006/037602 应用程序。在其他实施例中，多个应用文件包括多个应用程序。在其他实施例中，多个应用文件包括单个应用程序的多个版本。

向前参考图 10，流程图示出了驻留在诸如文件服务器 540 的远程机 30' 上的多个应用文件的一个实施例。在图 10 中，被称为程序包的多个应用文件包括包括一个或多个应用程序的三个不同版本的应用文件。

在一个实施例中，讲包括一个或多个应用程序版本并被存储在程序包的应用文件的每个子集称为目标。例如，目标 1 包括字处理应用程序和数据表程序版本，该版本与 Microsoft Windows 2000 操作系统的英文版兼容。目标 2 包括字处理应用程序和数据表程序版本，该版本与 Microsoft XP 操作系统的英文版兼容。目标 3，字处理应用程序和数据表程序版本，该版本与具有服务套件 3 的 Microsoft Windows 2000 操作系统的日文版兼容。

现在参考图 9，在一些实施例中，从持有多个应用文件的远程机 30 检索的文件包括对程序包的描述以及多个应用文件中包括的目标。在其他实施例中，从远程机 30 检索的文件识别多个应用文件，该多个应用文件包括本地机 10 请求执行的应用程序。

本地机 10 响应于该文件检索执行多个应用文件所需的至少一个特征（步骤 906）。在一些实施例中，本地机 10 可以不执行应用程序，除非本地机包括特定的特征。在这些实施例之一中，不同的应用程序需要本地机 10 包括与其他应用程序所需的特征不同的特征。在这些实施例的另一个中，本地机 10 接收执行包括本地机 10 所请求的应用程序的多个应用文件所需的至少一个特征的标识。

本地机判断本地机 10 是否包括至少一个特征（步骤 908）。在一个实施例中，本地机 10 评估本地机 10 上的操作系统以判断本地机 10 是否包括至少一个特征。在另一个实施例中，本地机 10 识别本地机 10 上的操作系统所用的语言，以判断本地机 10 是否包括至少一个特征。在又一个实施例中，本地机 10 识别本地机 10 上的操作系统的版本水平以判断本地机 10 上是否包括至少一个特征。在又一个实施例中，本地机 10 识别驻留在本地机 10 上的应用程序的应用版本以判断本地机 10 上是否包括至少一个特征。在一些实施例中，本地机 10 判断本地机

10 是否包括设备驱动程序以判断本地机 10 是否包括至少一个特征。在其他实施例中，本地机 10 判断本地机 10 是否包括操作系统以判断本地机 10 是否包括至少一个特征。在其他实施例中，本地机 10 判断本地机 10 是否包括执行多个应用文件的许可以判断本地机 10 是否包括至少一个特征。

本地机 10 响应于本地机 10 缺少至少一个特征的判断执行第二客户端，第二客户端请求在远程机 30 上执行多个应用文件（步骤 910）。在一个实施例中，当本地机 10 判定本地机 10 缺少至少一个特征时，本地机 10 不执行能够接收应用流的第一客户端。在另一个实施例中，当本地机 10 缺少至少一个特征时，策略禁止本地机 10 通过应用流接收多个应用文件。在一些实施例中，本地机 10 判定本地机 10 不包括至少一个特征。在这些实施例之一中，本地机 10 执行第一客户端，第一客户端从远程机 30 接收包括多个应用文件的应用流以在本地机上执行。

在一些实施例中，本地机 10 在判定本地机 10 没有至少一个特征时执行第二客户端，第二客户端请求在远程机上执行多个应用文件。在这些实施例之一中，第二客户端向持有多个应用文件的远程机 30 发送请求。在这些实施例的另一个中，远程机 30 执行包括应用程序的多个应用文件并生成应用输出数据。在这些实施例的又一个中，第二客户端接收在远程机上执行多个应用文件产生的应用输出数据。在一些实施例中，第二客户端通过独立计算结构（Independent Computing Architecture）表示层协议或远程桌面 Windows（Remote Desktop Windows）表示层协议或 X-Windows 表示层协议接收应用输出数据。在这些实施例的又一个中，第二客户端在本地机 10 上显示应用输出。

在一些实施例中，第二客户端向未持有多个应用文件的远程机 30 发送请求。在这些实施例之一中，远程机 30 可以从持有多个应用文件的第二远程机 30 请求多个应用文件。在这些实施例的另一个中，远程机 30 可以通过应用流传输会话从第二远程机 30 接收多个应用文件。在这些实施例的又一个中，远程机 30 在隔离环境中存储所接收的多个应用文件并在隔离环境中执行应用程序。在这些实施例的又一个中，远程机向本地机上的第二客户端发送所产生的应用输出数据。

返回到图 5，在一个实施例中，能够接收应用流的第一客户端为应

用流传输客户端 552。应用流传输客户端 552 接收文件，响应于该文件检索多个应用文件的标识以及执行多个应用文件所需的至少一个特征，并判断本地机 10 是否包括至少一个特征。在另一个实施例中，第二客户端为客户端代理 560。在一些实施例中，客户端代理 560 响应于应用流传输客户端 552 做出的本地机 10 没有至少一个特征的判断，从应用流传输客户端 552 接收文件。

在一些实施例中，在本地机 10 上执行的应用 566 利用 Win32 FindFirstFile() 和 FindNextFile() API 调用枚举与应用 566 相关联的文件。在这些实施例之一中，多个应用文件包括应用 566。在这些实施例的另一个中，并非多个应用文件中的所有文件都驻留在本地机 10 上。在这些实施例的又一个中，流传输服务 554 检索档案文件中的多个应用文件，但仅提取多个应用文件的子集。在这些实施例的又一个中，即使在所请求的文件不在本地机 10 上时，流传输服务 554 和文件系统过滤器驱动程序 564 也提供用于满足枚举请求的功能。

在一个实施例中，通过如同多个应用文件中的所有文件都驻留在本地机 10 上一样来拦截枚举请求并提供数据，由此提供该功能。在另一个实施例中，通过由文件系统过滤器驱动程序 564 拦截被作为 IOCTL 命令，例如 IRP_MJ_DIRECTORY_CONTROL IOCTL 发送的枚举请求来提供该功能。当文件系统过滤器驱动程序 564 拦截调用时，文件系统过滤器驱动程序 564 将该请求重定向到流传输服务 554。在一个实施例中，文件系统过滤器驱动程序 564 在将请求重定向到流传输服务 554 之前判定所请求的枚举在本地机 10 上的隔离环境中。在另一个实施例中，流传输服务 554 利用多个应用文件中的文件实现该请求，该文件包括与多个应用文件相关联的目录结构的枚举。在又一个实施例中，流传输服务 554 向文件系统过滤器驱动程序 564 提供对该请求的响应以满足枚举请求。

现在参考图 11，流程图示出了在本地响应对与远程存储的文件相关联的文件元数据的请求的方法中所采取的步骤的一个实施例。简而言之，从远程机接收 (i) 表示远程机存储的应用程序的目录结构以及 (ii) 与包括所存储应用程序的每个文件相关联的元数据(步骤 1102)。存储目录结构和元数据 (步骤 1104)。接收至少一个访问与目录结构中的特定文件相关联的元数据的请求 (步骤 1106)。利用所存储的元

数据对至少一个请求做出响应（步骤 1108）。

更详细地参考图 11，从远程机接收表示远程机存储的应用程序的目录结构以及与包括所存储应用程序的每个文件相关联的元数据（步骤 1102）。在一个实施例中，流传输服务 554 接收目录结构和元数据。在另一个实施例中，在流传输服务 554 检索包括所存储的应用程序的多个应用文件时，流传输服务 554 接收目录结构和元数据。在又一个实施例中，在多个应用文件中的文件中存储目录结构和元数据。

在一个实施例中，与每个文件相关联的元数据包括至少一个文件的别名。在另一个实施例中，与每个文件相关联的元数据包括至少一个文件的短名、具有八个字符长度的名称、点以及三字符的扩展名。在一实施例中，与每个文件相关联的元数据包括至少一个文件的别名和至少一个文件的短名之间的映射。在一些实施例中，多个应用文件中的文件具有备选文件名。在这些实施例之一中，当由指向本地机的流传输服务 554 检索文件时，响应于该文件的别名和至少一个文件的短名之间的映射，将该文件与短名相关联。

存储目录结构和元数据（步骤 1104）。在一个实施例中，在隔离环境 556 中存储目录结构和元数据。在另一个实施例中，在高速缓冲存储器元件中存储目录结构和元数据。在又一个实施例中，利用表示远程机存储的应用程序的目录结构来生成表示在本地机上执行的应用程序的目录结构枚举。

接收至少一个访问与目录结构中的特定文件相关联的元数据的请求（步骤 1106）。在一个实施例中，该请求为枚举文件的请求。在另一个实施例中，该请求为确定本地是否有包括所存储的应用程序的文件拷贝的请求。

在一个实施例中，由在本地机上的隔离环境中执行的应用 566 提出该请求。在另一个实施例中，由应用流传输客户端 552 提出该请求。在又一个实施例中，代表应用 566 提出该请求。

在一个实施例中，由文件系统过滤器驱动程序 564 拦截请求。在另一个实施例中，由文件系统过滤器驱动程序 564 将请求转发到应用流传输客户端 552。在又一个实施例中，由文件系统过滤器驱动程序 564 将请求转发到流传输服务 554。

在一些实施例中，由代替操作系统功能或枚举目录的功能的功能

转播请求。在另一个实施例中，用转播动态链接库来拦截请求。可以在用户模式或内核模式下执行转播功能。在以用户模式执行转播功能的实施例中，可以在生成过程时将转播功能加载到过程的地址空间中。对于在内核模式下执行转播功能的实施例，转播功能可以与调度对文件操作的请求时使用的操作系统资源相关联。对于为每种文件操作提供独立的操作系统功能的实施例，可以独立转播每种功能。或者，可以提供单个转播功能，其拦截针对若干类型的文件操作的生成或打开调用。

利用所存储的元数据对至少一个请求做出响应（步骤 1108）。在一个实施例中，文件系统过滤器驱动程序 564 对请求做出响应。在另一个实施例中，应用流传输客户端 552 对请求做出响应。在又一个实施例中，流传输服务 554 对请求做出响应。在一个实施例中，访问所存储的元数据以对至少一个请求做出响应。在另一个实施例中，利用文件远程拷贝在本地的误指示对请求做出响应。

在一个实施例中，响应于所接收的元数据，满足 Windows 操作系统 FindFirst 操作。在另一个实施例中，响应于所接收的元数据，满足 Windows 操作系统 FindNext 操作。在又一个实施例中，响应于所接收的元数据满足在目录结构中识别根节点的操作。在一些实施例中，使用诸如 WIN32_FIND_DATA API 的应用层 API 对操作做出响应。在其他实施例中，使用诸如 FILE_BOTH_DIR_INFORMATION 的内核层 API 对操作做出响应。

在一个实施例中，元数据满足用于识别与目录结构中的节点相关联的一次访问的操作。在另一个实施例中，元数据满足用于识别与目录结构中的节点相关联的一次修改的操作。在又一个实施例中，元数据满足用于识别目录结构中的修改节点的操作。

现在参考图 12，方框图示出了用于在本地对与远程存储的文件相关联的文件元数据的请求做出响应的系统的一个实施例，该系统包括流传输服务 554、文件系统过滤器驱动程序 564、目录结构 570、多个应用文件 572、元数据 574 和高速缓冲存储器元件 576。简而言之，该目录结构 570 标识与至少一个应用程序相关联的多个文件。元数据 574 与多个文件中的至少一个相关，该多个文件中的至少一个驻留在远程机上。在一个实施例中，目录结构 570 包括元数据 574。高速缓冲存储

器元件 576 存储目录结构 570。文件系统过滤器驱动程序 564 拦截访问与至少一个远程存储的文件相关联的元数据的请求，访问高速缓冲存储器元件并利用所存储的目录结构对至少一个请求做出响应。

在一些实施例中，流传输服务 554 接收目录结构 570 和元数据 574。在这些实施例之一中，目录结构 570 代表多个与应用程序相关联的应用文件 572，该多个应用文件 572 驻留在诸如远程机 30 的远程机上。在这些实施例的另一个中，元数据 574 包括用于对 Windows 操作系统 FindFirst 请求做出响应的信息。在这些实施例的又一个中，元数据 574 包括用于对 Windows 操作系统 FindNext 请求做出响应的信息。在这些实施例的又一个中，元数据 574 包括用于对识别目录结构中的根节点的请求做出响应的信息。在这些实施例的另一个中，元数据 574 包括用于对识别目录结构中的节点的请求做出响应的信息。在一些实施例中，使用诸如 WIN32_FIND_DATA 的应用层 API 对操作做出响应。在其他实施例中，使用诸如 FILE_BOTH_DIR_INFORMATION 的内核层 API 对操作做出响应。

在一些实施例中，可以将关于文件的少量元数据 574 直接存储在文字文件名中，例如将元数据指示符负在虚拟名后面，其中元数据指示符是与特定元数据状态唯一相关的字符串。元数据指示符可以表示或编码一个或几个元数据比特。希望通过虚拟文件名访问文件的请求检查是否有因为存在元数据指示符而导致的文字文件名的可能变化，转播或拦截希望检索文件自身名称的请求以便利用文字名称做出响应。在其他实施例中，可以用虚拟文件名和元数据指示符形成文件的一个或多个备选名称，且可以利用文件系统提供的硬链接或软链接设施来生成。如果给出利用链接名称访问文件的请求，则通过表示未找到该文件来通过隔离环境使这些链接的存在相对于应用隐藏起来。特定链接的存在或不存在可以为每个元数据指示符表示元数据的一比特，或者可能有能采用多个状态表示元数据若干比特的与元数据指示符的链接。在文件系统支持备选文件流的其他实施例中，可以制作备选文件流来体现元数据，流的大小表示元数据的若干比特。在其他实施例中，文件系统可以直接提供为文件系统中每个文件存储一些第三方元数据的能力。在其他实施例中，可以用独立的子范围来记录被删除的文件，用该子范围内存在文件（未标识为占位符）表示该文件被

删除。

在一个实施例中，组合用户隔离环境、应用隔离环境和系统范围中的数据以形成表示应用的目录结构的本地枚举。在另一个实施例中，流传输服务 554 访问元数据 574 和目录结构 570 以填充应用隔离环境。在又一个实施例中，文件系统过滤器驱动程序 564 生成目录结构的本地枚举。在又一个实施例中，目录结构的本地枚举标识出多个应用文件 572 中的至少一个文件，该至少一个文件驻留在远程机上而不是本地机上。在一些实施例中，在高速缓冲存储器元件 576 上存储目录结构的本地枚举。在其他实施例中，流传输服务 554 生成应用隔离环境和目录结构的本地枚举。

在一个实施例中，文件系统过滤器驱动程序 564 拦截被发送到系统范围希望访问目录结构的本地枚举的请求。在另一个实施例中，文件系统过滤器驱动程序 564 在拦截该请求之后生成本地枚举。在又一个实施例中，文件系统过滤器驱动程序 564 将对本地枚举的请求重定向到用户隔离环境。在又一个实施例中，文件系统过滤器驱动程序 564 将对本地枚举的请求重定向到应用隔离环境。

在一些实施例中，文件系统过滤器驱动程序 564 拦截对目录本地枚举中的文件标识的访问请求，该文件驻留在远程机上。在这些实施例之一中，如下文参考图 3 要更详细描述的，文件系统过滤器驱动程序 564 请求由流传输服务 554 检索该文件。

在隔离环境中运行的应用提出对文件的请求时，过滤器驱动程序拦截这些请求。如果该请求是要打开文件，过滤器驱动程序将首先将请求重定向到隔离环境，以判断隔离环境是否可以满足该请求。如果调用成功，过滤器驱动程序将利用位于隔离环境中的文件实例对请求做出响应。

然而，如果所请求的文件不在隔离环境中，过滤器驱动程序则向流传输服务 554 发送请求以从多个应用文件检索该文件，一直中断到完成请求，然后重试初始的文件打开。在一些实施例中，将流传输服务 554 在接收到来自过滤器驱动程序的请求时从多个应用文件检索文件的功能称为“按需高速缓存”。

现在参考图 13，流程图示出了在访问与本地执行的应用程序相关联的目录结构中的远程文件的方法中采取的步骤的一个实施例。简而

言之，拦截应用访问文件的请求（步骤 1302）。将该请求重定向到第一隔离环境（步骤 1304）。做出第一隔离环境中不存在所请求文件的判断（步骤 1306）。响应于在与驻留在远程机上的多个应用文件相关联的目录结构枚举中识别出该文件的判断，将该请求重定向到第二隔离环境（步骤 1308）。响应于第二隔离环境不含该文件且在枚举中识别出该文件的判断，从远程机检索所请求的文件（步骤 1310）。

参考图 13，更详细地讲，拦截应用访问文件的请求（步骤 1302）。在一个实施例中，由文件系统过滤器驱动程序拦截请求。在另一个实施例中，文件系统过滤器驱动程序拦截访问文件的所有请求。在又一个实施例中，应用流传输客户端 552 拦截该请求。在一些实施例中，拦截应用访问可执行文件的请求。在其他实施例中，拦截应用访问文件的请求、在本地机 10 上执行的应用的一部分。

将该请求重定向到第一隔离环境（步骤 1304）。在一个实施例中，在第一隔离环境之内执行应用。在一个实施例中，该应用是诸如文字处理程序或数据表程序的应用程序。在另一个实施例中，该应用是应用流传输客户端。在又一个实施例中，该应用是应用流传输客户端 552 中试图代表本地机 10 的用户启动应用程序的部件。在另一个实施例中，文件系统过滤器驱动程序将该请求重定向到第一隔离环境。

做出第一隔离环境中不存在所请求文件的判断（步骤 1306）。在一个实施例中，文件系统过滤器驱动程序接收第一隔离环境中不存在所请求的文件的指示。

响应于在与驻留在远程机上的多个应用文件相关联的目录结构枚举中识别出该文件的判断，将该请求重定向到第二隔离环境（步骤 1308）。在一个实施例中，利用关于执行第一应用的访问信息接收目录结构枚举。在另一个实施例中，该枚举标识包括第二应用的多个应用文件。在本实施例中，第一应用是第二应用的本地拷贝。

响应于第二隔离环境不含该文件且在枚举中识别出该文件的判断，从远程机检索所请求的文件（步骤 1310）。在一个实施例中，从第二远程机检索所请求的文件。在另一个实施例中，从文件服务器检索所请求的文件。在一些实施例中，目录结构枚举识别驻留在本地机上的多个应用文件。在其他实施例中，目录结构枚举指出多个应用文件在本地机上。在这些实施例之一中，当应用请求访问目录结构枚举

已指出驻留在本地机上的多个应用文件中的文件时，在拦截访问请求时从文件服务器获取该文件。在这些实施例的另一个中，文件服务器将所请求的文件流传输到本地机。在这些实施例的又一个中，在接收到所请求的文件时，在第二隔离环境中存储所请求的文件。在其他实施例中，当应用请求访问目录结构枚举已指出驻留在本地机上的多个应用文件中的文件时，从本地高速缓冲存储器向应用提供该文件的拷贝。

在一些实施例中，所请求的文件是加密的。在其他实施例中，所请求的文件被存储成加密的形式。在其他实施例中，如果请求该文件的应用没有访问所请求的文件的授权，可以防止该应用对所请求的文件解密。

在一个实施例中，做出目录结构枚举未识别出该文件的判断。在本实施例中，可以将访问文件的请求重定向到第一隔离环境之外和第二隔离环境之外的环境。

在一些实施例中，拦截访问文件的第二请求。在这些实施例之一中，由第二应用提出访问文件的请求。在这些实施例的另一个中，第二应用在第三隔离环境中执行。在这些实施例的又一个中，响应于在该枚举中枚举了该文件且第二隔离环境不含该文件的判断，将请求重定向到第二隔离环境。可以做出在从文件服务器接收到该文件时本地机将文件存储在第二隔离环境中的决定。在又一个实施例中，将该文件存储在第三隔离环境中。

现在参考图 14，方框图示出了用于访问与应用相关的目录结构中的文件的系统的一个实施例。简而言之，本地机 10 包括应用流传输客户端 552、流传输服务 554、隔离环境 556、文件系统过滤器驱动程序 564 和第一应用 566。本地机 10 可以与文件服务器 540、远程机 30、web 接口 558 和第二应用 566' 交互。

本地机 10 对应用流传输客户端 552 初始化以执行第一应用 566。在一个实施例中，应用流传输客户端 552 对流传输服务 554 初始化以检索和执行第一应用 566。在一些实施例中，多个应用文件包括第一应用 566。在这些实施例之一中，流传输服务 554 检索多个应用文件并将它们存储在隔离环境 556 中。在这些实施例的另一个中，流传输服务 554 识别多个应用文件所在的远程机位置，但不检索多个应用文件。在

这些实施例的又一个中，流传输服务 554 检索多个应用文件的文件子集。在这些实施例的又一个中，流传输服务 554 检索含有多个应用文件的档案文件。

在一个实施例中，第一应用 566 包括驻留在远程机 30 上的第二应用 566' 的本地拷贝。在另一个实施例中，多个应用文件驻留在远程机 30 上且包括驻留在远程机 30 上的第二应用 566'。在又一实施例中，为了执行第二应用 566'，本地机 10 检索多个应用文件，在本地机上生成第一应用 566，并执行第一应用 566。在一些实施例中，应用 566 和 566' 为用户应用，例如文字处理应用或电子表格应用或演示文稿应用。

在一些实施例中，多个应用文件包括标识与远程机 30 上的多个应用文件相关联的目录结构的文件。在这些实施例之一中，该文件包括与多个应用文件中的每个文件有关的元数据。在这些实施例的另一个中，流传输服务 554 从该文件检索元数据以产生与多个应用文件相关联的目录结构枚举，如上文参考图 12 所述。在这些实施例的又一个中，流传输服务 554 存储与包括第二应用 566' 的多个应用文件相关联的目录结构枚举。在一些实施例中，流传输服务 554 在第二隔离环境中存储该枚举。

在一个实施例中，流传输服务 554 检索与第一应用 566 相关联的初始可执行文件。在另一个实施例中，流传输服务 554 在检索初始可执行文件时在本地机 10 上执行第一应用 566。在又一个实施例中，第一应用 566 请求访问多个应用文件中的其他文件，因为继续执行第一应用 566 需要这些文件。在一些实施例中，在隔离环境 556 中执行第一应用 566。

文件系统过滤器驱动程序 564 拦截在隔离环境 556 中执行的第一应用 566 访问多个应用文件中的文件的请求。文件系统过滤器驱动程序 564 将该请求重定向到隔离环境 556。

如果所请求的文件在隔离环境 556 中，则向第一应用 566 提供对所请求的文件的访问。如果所请求的文件不在隔离环境 556 中，文件系统过滤器驱动程序 564 则将该请求重定向到第二隔离环境。在一个实施例中，第二隔离环境包括由流传输服务 554 产生并与包括第二应用 566' 的多个应用文件相关联的目录结构枚举。在另一个实施例中，做出在目录结构枚举中识别所请求的文件的决定。

在一些实施例中，流传输服务 554 向隔离环境 556 提供信号量。在这些实施例之一中，文件系统过滤器驱动程序 564 利用信号量向流传输服务 554 表示需要访问多个应用文件中的文件。在其他实施例中，文件系统过滤器驱动程序 564 使用线程向流传输服务 554 表示需要访问该文件。

在从文件系统过滤器驱动程序 564 接收到通知时，流传输服务 554 从多个应用文件检索所请求的文件。在这些实施例的又一个中，流传输服务 554 在第二应用隔离环境中存储所请求的文件。在一个实施例中，利用从多个应用文件中检索且存储在第二隔离环境中的文件实例满足访问文件的请求。在另一个实施例中，还在第一隔离环境中存储所请求的文件。

在一些实施例中，做出第二隔离环境不含该文件且在枚举中识别该文件的决定。在这些实施例之一中，在与包括第二应用 566' 的多个应用文件相关联的目录结构枚举中识别该文件，且该文件是多个应用文件中的文件。在这些实施例的另一个中，流传输服务 554 不从远程机检索该文件。在这些实施例的又一个中，流传输服务 554 不检索包括所请求的文件的多个应用文件。在这些实施例的又一个中，流传输服务 554 在档案文件中检索多个应用文件，但不从该档案文件检索所请求的文件。

在一个实施例中，流传输服务 554 包括与文件系统过滤器驱动程序通信的收发器。在另一个实施例中，收发器从文件系统过滤器驱动程序接收重定向的请求。在一实施例中，收发器将对该文件的请求转发到持有所请求文件的远程机。在一个实施例中，远程机为文件服务器 540。在另一个实施例中，将该请求转发到远程机 30，远程机 30 将该请求路由到文件服务器 540。在一些实施例中，文件服务器 540 将所请求的文件流传输到本地机 10 上的收发器。在其他实施例中，远程机 30 将所请求的文件流传输到本地机 10 上的收发器。在其他实施例中，在从文件服务器 540 接收到所请求的文件时，收发器将所接收的文件存储在第二隔离环境中。

在一个实施例中，文件系统过滤器驱动程序 564 在第三隔离环境中拦截由在本地机 10 上执行的第三应用 566"提出的访问文件的第二请求。在另一个实施例中，文件系统过滤器驱动程序 564 将访问文件

的请求重定向到第二隔离环境。在又一个实施例中，文件系统过滤器驱动程序 564 决定在拦截第三应用”的访问请求之前流传输服务 554 将所接收的文件存储在第二隔离环境中。

在一些实施例中，在初始化时，流传输服务 554 可以在执行应用程序之前填充隔离环境中的高速缓冲存储器。在这些实施例之一中，流传输服务 554 向隔离环境中安装注册表文件。在这些实施例的另一个中，流传输服务 554 存储文件的长名称和短文件名之间的映射。

在一个实施例中，为了节省本地机上的空间，可以限制高速缓冲存储器的大小。在一些实施例中，当高速缓冲存储器接近其大小极限时，将自动清除高速缓冲存储器中最老的文件以为新文件腾出空间。在这些实施例之一中，通过操作系统保持的表示“上次访问”时间戳的时间戳确定文件的年龄。除了文件年龄之外，还可以考虑文件类型-可以将二进制可执行文件 (.EXE, .LL 等) 保持比类似年龄的其他类型文件更长的时间。

在初始化时，流传输服务 554 可以枚举当前在高速缓冲存储器中的文件并判断高速缓冲存储器的总大小。在由隔离环境 556 或流传输服务 554 向高速缓冲存储器添加文件之后，流传输服务 554 调用功能，以向高速缓冲存储器系统通知新文件、其位置和其大小。将每个新缓存文件的大小添加到高速缓冲存储器当前正运行的总大小。然后将这一新的总大小与高速缓冲存储器大小极限进行比较，如果已经超过该极限，代码发出线程以增加高速缓冲存储器的年龄。在任何给定时间只能有该线程的一个实例在运行。

该线程生成高速缓冲存储器中当前所有文件的列表，按照上次访问时间戳对该列表排序，然后开始沿该列表向下删除文件，直到我们具有足够大的磁盘空间来满足该线程的退出标准为止。退出标准基于高速缓冲存储器大小下降到极限以下水平，该水平是作为极限百分比确定的（缺省值为 10%）。删除比所需更多的内容，以防止超过极限，防止高速缓冲存储器在每次增加新文件时都不稳定。

在一些实施例中，流传输服务 554 提供了以压缩文件格式将包括应用程序的多个应用文件中的每个文件拷贝到本地机 10 的能力。可以将这种能力称为“预高速缓存”。在这些实施例之一中，在随后执行应用程序时，所有的打包请求都指向本地拷贝而不是通过网络传播。

这些实施例可以使本地机 10 的用户能够在用户无法访问网络的时候执行应用程序。

远程机 30 包括监测本地机 10 使用应用的情况的功能。远程机 30 可以监测本地机 10 使用的每个应用的状态，例如在执行或终止应用时进行监测。在一个实施例中，远程机 30 要求本地机 10 发送有关本地机 10 执行的应用状态的消息。在另一个实施例中，当本地机 10 连接到远程机 30 所在的网络时，本地机 10 发送消息，表示本地机 10 已连接到该网络。

在一个实施例中，当本地机 10 与远程机 30 交互并执行一个或多个应用时，说本地机 10 具有会话。在另一个实施例中，远程机 30 要求本地机 10 在会话期间维持从远程机接收的授权执行应用的许可。在又一个实施例中，会话具有由远程机分配的唯一的会话标识符。

在一个实施例中，本地机 10 向与其交互的远程机 30 发送消息，以接收和执行应用程序。在另一个实施例中，本地机 10 从远程机 30 接收诸如会话管理服务器 562 的第二远程机的标识符，第二远程机接收并存储所发送的与本地机 10 上的会话相关的所有消息。

在一些实施例中，会话管理服务器 562 是提供许可管理和会话监测服务的远程机 30。在这些实施例之一中，会话管理服务器 562 包括提供这些服务的服务器管理子系统 508。

在一个实施例中，本地机 10 直接向会话管理服务器 562 发送消息。在另一个实施例中，本地机 10 向远程机 30 发送消息，远程机 30 将该消息转发到具有本地机 10 的标识的会话管理服务器 562。

本地机 10 可以向远程机 30 发送心跳消息。在一个实施例中，心跳消息包括对许可的请求。在本实施例中，本地机 10 可以在接收到与本地机 10 请求授权执行的应用程序相关联的访问信息之后，发送心跳消息。本地机 10 可以在执行应用之前发送心跳消息。在一个实施例中，本地机 10 在心跳消息中包括利用访问信息接收的启动权证。在本实施例中，远程机 30 可以在成功验证启动权证时为本地机 552 授予许可。

在另一个实施例中，心跳消息包括表明本地机已经开始执行应用的指示。在一实施例中，心跳消息包括表明本地机已经终止执行应用的指示。在又一个实施例中，心跳消息包括表明无法执行应用的指示。

在一个实施例中，心跳消息包括对第二会话管理服务器，例如会话管理服务器 562 的标识的请求。在另一个实施例中，心跳消息包括表明本地机 10 已经连接到远程机 30 所在网络的指示。

在一些实施例中，心跳消息包括复位应用流传输会话的请求。在这些实施例之一中，当发生错误时本地机 10 发送该心跳消息，并终止远程机 30 所在网络和本地机 10 之间的连接。在这些实施例的另一个中，本地机 10 与心跳消息一起发送与会话相关的信息。在这些实施例的又一个中，如果会话未到期，远程机 30 可以向本地机 10 发送与会话相关的数据。

在这些实施例的另一个中，如果远程机 30 从其所在网络断开连接，本地机 10 可能无法接收到对发送到远程机 30 的心跳消息的回复。在一个实施例中，本地机 10 可以通过向远程机 30 发送请求会话复位的消息来重新建立会话。在另一个实施例中，本地机 10 可以通过向第二远程机 30 发送请求会话复位的消息来重新建立会话。在一些实施例中，当远程机 30 重新连接到网络时，它将会针对在远程机 30 断开连接时接收的每个会话复位请求建立新的会话。在这些实施例之一中，新会话将与重新连接且未许可状态相关联。在这些实施例的另一个中，将不会为新会话获取任何新许可。在这些实施例的又一个中，在本地机 10 执行应用时，将获取新的许可，并且与本地机 10 相关的所有会话将与活动且许可状态相关联。

在一些实施例中，本地机 10 上的应用流传输客户端 552 产生心跳消息。在这些实施例之一中，应用流传输客户端 552 将心跳消息转发到 web 接口 558，以发送到本地机 10，以发送到远程机 30。在其他实施例中，远程机 30 上的管理服务 504 通过 web 接口 558 从本地机 10 接收心跳消息。在其他实施例中，包括收集点 240（上文参考图 1D 所述）的远程机 30 接收并存储心跳消息。

在一些实施例中，应用流传输客户端 552 从远程机 30 请求许可。在这些实施例之一中，该许可授权在本地机 552 上执行应用程序。在这些实施例的另一个中，远程机 30 可以访问第二远程机以提供许可。在这些实施例的又一个中，远程机 30 可以向本地机提供该许可。在这些实施例的又一个中，远程机 30 可以向第二远程机提供授权可以接受的许可。在一些实施例中，在终止执行应用程序的时候收回该许可。

在一些实施例中，农场 38 中的远程机 30 包括许可管理子系统，该许可管理子系统用于为需要操作许可的那些子系统配置和维持许可并控制通往这种子系统的连接数量。在其他实施例中，远程机 30 将许可管理子系统的功能并入诸如应用管理子系统和会话管理子系统的其他子系统中。在一个实施例中，每个远程机 30 包括许可管理子系统或与许可管理子系统相关的功能。许可管理子系统管理两种许可：(1) 特征许可和(2) 连接许可。简而言之，许可管理子系统使用特征许可来控制对许可软件产品的“特征”的访问，例如负载管理，使用连接许可来控制这些许可软件产品允许的用户连接数量。特征可以是软件产品的一些方面或特定功能，或者特征可以是没有特征许可就无法工作的整个产品。

图 15 示出了农场 38 中的远程机 30 的一个实施例，其中远程机 30 包括许可管理子系统 1510、组子系统 1520、持久存储器系统服务模块 1570、动态存储器系统服务模块 1580、关系子系统 1530、专用远程机子系统 1540 和公共访问点子系统 524。图 15 中所示的这些子系统是为了描述许可管理子系统 1510 的行为。远程机 30 可以包括其他类型的子系统。

许可管理子系统 1510 通过事件总线与组子系统 1520 通信，以形成和维持许可的逻辑分组（下文称“许可组”），以促成许可池、许可分配和许可组。许可组包括如下所述的许多许可串和/或其他许可组。许可组收集类似特征的许可并从而实现许可的积蓄。积蓄的许可是农场 38 中的任何远程机 30 都可以使用的许可。每个许可组都保存该许可组和其他许可子组（即许可组之内的其他许可组）中的许可的集体能力。在一个实施例中，与许可池相关的信息是保持在动态存储器 240 中的。在本实施例中，每个许可管理子系统 1610 都在本地存储许可的总数以及分配给农场 38 中的远程机 30 的许可数量。在授予积蓄的许可时，授予许可的许可管理子系统 1510 在动态存储器 240 中建立条目，表示该积蓄的许可在“使用中”。其他每个许可管理子系统 1510 发现不能授予这样的积蓄许可。在一个特定实施例中，动态存储器 240 存储与每个许可组相关联的远程机 ID/客户端 ID 对，以标识使用中的积蓄许可。

关系子系统 1530 维护着许可和远程机 30 之间以及许可组与远程

机 30 之间的关联。该关联界定了每个许可和许可组中仅关联的远程机 30 可以获得的许可数量（即“本地许可”）。本地许可是被分配给农场 38 中一个远程机且不和其他远程机 30 共享的许可。许可管理子系统 1510 与关系子系统 1530 通信以生成、删除、查询和更新这种关联。公共访问点子系统 524 提供远程过程调用（RPC），供远程机 30 上的软件产品使用。这些 RPC 界面使这种软件产品能够通过公共访问子系统 524 通信，以访问许可信息。

仍然参考图 15，专用远程机子系统 1540 与许可管理子系统 1510 通信，以针对需要许可的专用远程机子系统 1540 的每种能力获得特征许可。这发生在专用远程机子系统 1540 初始化时以及任何许可事件之后。如果不能获得特征许可，专用远程机子系统 1540 会限制应为其提供许可的子系统的功能。而且，只要在与远程机 30 发起客户端会话的时候，专用远程机子系统 1540 就使用许可管理子系统 1510 来获得客户端连接许可。

许可管理子系统 1510 与持久存储器系统服务模块 352 通信，以便将特征和连接许可作为根据命名约定形成的许可串存储在许可储存库 1550 中。许可储存库 1550 位于持久存储器 230 中。循环冗余校验（CRC）防止在将这种许可存储在许可储存库 1550 中时该许可被篡改。许可管理子系统 1510 还在许可储存库 1550 中存储与许可串相关的信息。例如，该信息可以表示哪个许可被分配给农场 38 的哪个远程机 30，在一些实施例中，可以表示每个许可的激活状态。在一个实施例中，连接许可表 1560 存储已经获得连接许可的那些本地机的标识符。

在一个实施例中，许可管理子系统 1510 支持来自请求使用被许可能力的子系统的事件，例如对可用积蓄许可的请求。该事件包括请求许可的子系统的 UID 以及该子系统所在远程机 30 的 UID。该事件还包含以许可组 ID 形式请求的许可类型（即特征或连接许可）。持久存储器 230 中存储的实际许可组 ID 是任意的，但遵守命名约定会为今后向远程机 30 增加新软件产品（即子系统）提供了灵活性。

由寻求许可的请求子系统发送的事件包括（1）许可组类型的指示、请求许可的本地机和远程机的身份以及“强迫获取”标志。许可组类型的指示可以包括诸如负载管理的特征许可或诸如软件应用产品的连接类型许可的标识。标识本地机和寻求许可的远程机的字段可以包括

与远程机和本地机相关联的唯一标识符。例如，可以使用强迫获取标志以在许可更改事件之后获取连接许可。许可更改事件表示持久存储器 230 中的许可信息被更改；例如，删除、添加或转让了许可。在许可更改事件时，每个远程机 30 试图再次获取在许可更改事件之前它所拥有的所有连接许可，因为对于该远程机而言许可更改事件的具体原因是未知的。如果设置了该标记，该标记表示必需要获取连接许可，即使这样做使通往远程机 30 的连接数超过允许连接的预定最大数。随后不授予任何新的连接许可，直到使用中的连接许可数量降到该预定最大数之下。通过这种方式，不会在会话中因为许可更改事件而终止本地机连接。

现在参考图 16，方框图示出了强迫许可中涉及的部件的一个实施例。远程机 30 包括服务器管理子系统 508 和许可管理子系统 512。在一些实施例中，服务器管理子系统 508 和许可管理子系统 512 提供上述许可管理子系统 1510 的功能。在其他实施例中，应用管理子系统 506 和会话管理子系统 510 提供上述许可管理子系统 1510 的功能。在其他实施例中，其他子系统提供上述许可管理子系统 1510 的功能。

在一个实施例中，服务器管理子系统 508 可以包括用于请求发出和收回许可的许可部件。在另一个实施例中，许可管理子系统 512 可以向从服务器管理子系统 508 接收的发出或收回许可的请求采用策略。在又一个实施例中，许可管理子系统 512 可以向提供强迫许可功能的远程机 30 发送请求。

在一些实施例中，管理服务 504 可以维持与提供强迫许可功能的第二远程机 30 的连接。在其他实施例中，远程机 30 提供强迫许可功能。在一些实施例中，在本地机 10 未能向远程机发送预定数量心跳消息的时候，许可期满并停止有效状态。在这些实施例之一中，许可期满收回本地机 10 执行应用程序的授权。

在其他实施例中，在预定时间期满时会话超时。在一个实施例中，管理服务 504 在许可期满后维持与会话相关的数据，直到会话期满为止。在一些实施例中，与会话相关的数据可以包括诸如会话名称、会话 ID、客户端 ID、客户端名称、会话开始时间、服务器名（文件服务器的 UNC 路径）、应用名称（本地机基于浏览器名称产生的唯一名称）、别名、会话状态（活动/许可、活动/未许可、重新连接/未许可）的信

息。在另一个实施例中，本地机 10 停止发送心跳消息并在稍后时间点重新开始发送心跳消息。在又一个实施例中，如果本地机 10 在会话期满之前重新开始发送心跳消息，管理服务 504 可以重新发放许可并使本地机 10 可以使用所维持的与会话相关的数据。

现在参考图 17，流程图示出了在本地计算机 10 的会话期间从远程机 30 请求并维持许可所采取的步骤的一个实施例。简而言之，应用流传输客户端请求许可（步骤 1702）。远程机 30 接收对许可的请求，验证与请求相关联的权证并生成许可（步骤 1704）。远程机 30 向本地机 10 提供许可以及与许可相关联的信息（步骤 1706）。本地机 10 执行以上参考图 7 的步骤 214 所述的应用。本地机发送表示本地机已经执行应用的心跳消息（步骤 1708）。远程机 30 接收该心跳消息并验证随该心跳消息发送的识别信息（步骤 1708）。远程机 30 生成与所执行应用和本地机 10 相关联的会话（步骤 1710）。将生成会话的结果发送到本地机 10（步骤 1712）。如以上参考图 7 的步骤 216 所述，在执行应用的整个期间本地机发送心跳消息。本地机接收对所发送的心跳消息的响应（步骤 1714）。本地机发送表示终止执行应用的心跳消息（步骤 1716）。远程机 30 接收该心跳消息并判断是否删除与会话相关的数据以及是否释放与本地机 10 和所终止的应用相关联的许可（步骤 1718）。将远程机 30 做出的判断结果发送到本地机 10（步骤 1720）。

现在参考图 17，更详细地讲，本地机 10 上的应用流传输客户端请求许可（步骤 1702）。在一些实施例中，本地机 10 在接收到与应用程序相关联的访问信息时请求该许可。在这些实施例之一中，本地机从发出由本地机 10 执行应用程序的授权的远程机 30 请求许可。在一些实施例中，对许可的请求包括从远程机 30 随访问信息接收的启动权证。在其他实施例中，本地机 10 上的应用流传输客户端 552 向 web 接口 558 发送请求，web 接口 558 向远程机 30 发送请求。在其他实施例中，远程机上的会话管理子系统 510 接收并处理对许可的请求。

远程机 30 接收对许可的请求，验证与请求相关联的权证并生成许可（步骤 1704）。在一个实施例中，远程机 30 验证是否授权本地机 10 执行应用。在另一个实施例中，远程机 30 判断本地机 10 是否已经与现有许可相关联。在又一个实施例中，远程机 30 判定本地机 10 与现有许可相关联，并为本地机 10 提供用于会话管理服务器 562 管理现

有许可的标识符。在又一个实施例中，远程机 30 生成新许可、会话标识符和管理新许可的会话管理服务器 562 的标识并向本地机 10 提供。

在一些实施例中，在许可管理子系统 1510 接收许可请求的实施例中，远程机 30 使用许可管理子系统 1510 对许可请求做出响应。该请求可以是为了特征许可或连接许可。许可管理子系统 1510 判断是否已经授予许可，即，是否已经开始该特征或是否已经存在本地机的连接。如果已经授予了该许可，许可管理子系统 1510 向许可请求者发送“授予”事件。如果以前没有授予许可，许可管理子系统 1510 判断本地许可是否可用，本地许可即已经永久分配给远程机 30 的许可。在一些实施例中，许可管理子系统 1510 通过检查本地存储器进行该判断。如果本地许可可用，即，远程机 30 具有比当前授予的更多的永久分配许可，许可管理子系统 1510 向许可请求者发送“授予”事件。

远程机 30 向本地机 10 提供许可以及与许可相关联的信息（步骤 1706）。在一个实施例中，在从远程机 30 接收到许可、会话标识符和会话管理服务器 562 的标识时，本地机 10 执行应用。本地机 10 可以如以上参考图 7 的步骤 214 所述执行该应用。本地机发送表示本地机已经执行应用的心跳消息（步骤 1708）。在一个实施例中，本地机向远程机 30 发送心跳消息，以将心跳消息发送到会话管理服务器 562。在另一个实施例中，本地机 10 响应于从远程机 30 接收的会话管理服务器 562 的标识符直接向会话管理服务器 562 发送心跳消息。

远程机 30 接收该心跳消息并验证随该心跳消息发送的识别信息（步骤 1708）。在一个实施例中，远程机 30' 为会话管理服务器 562。在另一个实施例中，会话管理服务器 562 验证由本地机 10 与心跳消息一起提供的服务器标识符。在又一个实施例中，服务器标识符是由远程机 30 向本地机 10 提供的标识符。

远程机 30 生成与所执行应用和本地机 10 相关联的会话（步骤 1710）。在一个实施例中，会话管理服务器 562 在接收到心跳消息时生成与执行应用相关的新会话。在另一个实施例中，第三远程机 30 生成新会话。在一些实施例中，会话管理服务器 562 在生成新会话时存储与会话相关的信息。

将生成会话的结果发送到本地机 10（步骤 1712）。在一些实施例中，该结果确认会话的生成。在其他实施例中，该结果标识与会话相

关的一个或多个应用。如以上参考图 7 的步骤 216 所述，在执行应用的整个期间本地机发送心跳消息。在一个实施例中，在执行应用程序的整个期间本地机 10 以周期性间隔定期向会话管理服务器 562 发送心跳消息。本地机接收对所发送的心跳消息的响应（步骤 1714）。在一个实施例中，本地机 10 从会话管理服务器 562 接收心跳消息的接收确认。在另一个实施例中，本地机 10 响应于会话管理服务器 562 接收到心跳消息，从会话管理服务器 562 接收执行命令。

本地机发送表示终止执行应用的心跳消息（步骤 1716）。远程机 30 接收该心跳消息并判断是否删除与会话相关的数据以及是否释放与本地机 10 和所终止的应用相关联的许可（步骤 1718）。将远程机 30 做出的判断结果发送到本地机 10（步骤 1720）。

现在参考图 18，方框图示出了可能与管理服务 504 监测的会话相关联的状态的一个实施例。在一个实施例中，管理服务 504 上的会话维护子系统 510 监测本地机 10 的会话并向会话分配状态。在另一个实施例中，会话维护子系统 510 维持与许可相关的数据列表，该列表可以包括与本地机相关的标识符、与会话相关的标识符、会话状态和表示远程机 30 从本地机 10 上次接收消息的时间戳。在一些实施例中，会话维护子系统 510 包括会话监测线程。在这些实施例之一中，会话监测线程在周期性许可超时间隔醒来，扫描与许可相关的数据列表并更新会话的会话状态。

会话可以处于的第一状态为活动且许可状态。在一个实施例中，当处于该状态下时，本地机 10 维持有效许可，该许可授权执行应用。在另一个实施例中，会话管理服务器 562 保持与会话相关的数据。在一些实施例中，会话管理服务器 562 在第二远程机上存储与会话相关的数据。在一个实施例中，当本地机 10 一开始执行应用时，用于本地机的会话处于活动且许可状态。

会话可以处于的第二状态为活动且未许可状态。在一个实施例中，当本地机 10 未能发送心跳消息且对本地机 10 的许可期满时，会话处于该状态。在另一个实施例中，如果会话处于该状态，那么尽管许可已经期满，但距会话期满还未过去足够时间，则认为会话是活动的。在一些实施例中，当会话处于这种状态时，远程机 30 或会话管理服务器 562 可以代表本地机 10 存储与会话相关的数据。在其他实施例中，

如果本地机 10 在会话期满之前发送心跳消息，则向本地机 10 发送带有新许可的与会话相关的数据，且会话返回到活动且授权状态。在一个实施例中，远程机 30 使用会话标识符以及与本地机相关联的标识符验证会话未期满并向本地机提供适当的与会话相关的数据。

会话可能所处的第三状态是断开连接且不存在状态。当会话期满时，删除与会话相关的数据。

会话可以处于的第四状态为再次连接且未许可状态。在一个实施例中，当本地机 10 上的会话期满时，删除与会话相关的数据。在另一个实施例中，当本地机 10 发送新的心跳消息时，为本地机 10 生成新会话标识符和本地机标识符。在一些实施例中，本地机 10 再次向远程机 30 进行认证，接收新许可并进入活动且授权状态。

表 3 总结了可能与会话相关的状态。

会话状态	描述
活动\许可	正常运行模式
活动\未许可	丢失心跳的期间>许可超时 且 丢失心跳的期间<会话超时
重新连接\未许可	丢失心跳的期间>会话超时 或 持有会话的 CPS/RADE 下线后再次上线

表 3

在一些实施例中，打包机构使得能生成与应用程序相关联的多个应用文件。在这些实施例之一中，打包机构实现多个应用文件的标识。在这些实施例的另一个中，打包机构使得能将个体应用文件分组成多个应用文件。在这些实施例的又一个中，打包机构实现在诸如文件服务器或应用服务器的远程机上持有很多个应用文件。

在一个实施例中，打包机构在被称为“分级机”的远程机上执行。在另一个实施例中，打包机构在“清洁机”上执行。清洁机可以是其上仅安装了操作系统的远程机，而没有额外的软件、驱动程序、注册表项或其他文件。在又一个实施例中，打包机在远程机上执行，该远程机模仿可以在其上执行应用程序的本地机。在一些实施例中，打包机构在其上执行的远程机包括提供清洁机环境的隔离环境，即使在远

程机自身不是清洁机的情况下也可以向该清洁机环境中安装应用。

在一个实施例中，将多个应用文件称为“程序包”。在另一个实施例中，该程序包可以是存储多个应用文件的档案文件。在又一个实施例中，该程序包可以是存储多个应用文件以及包括与多个应用文件中的至少一个文件相关联的元数据的文件的档案文件。在一些实施例中，程序包包括多个应用文件，该多个应用文件包括应用程序。在其他实施例中，程序包包括多个应用文件，该多个应用文件包括应用程序套件。在其他实施例中，程序包包括多个应用文件，该多个应用文件包括应用程序和执行应用程序所需的先决条件。

在一个实施例中，打包机构启动在隔离环境中安装程序的执行。在另一个实施例中，打包机构监测由安装程序对隔离环境产生的更改。在又一个实施例中，该打包机构监测安装程序在隔离环境中生成文件的情况。在又一个实施例中，该打包机构监测安装程序在隔离环境中修改文件的情况。在一些实施例中，多个应用文件包括由安装程序生成或修改的文件。在其他实施例中，打包机构实现文件系统过滤器驱动程序 564 以监测该隔离环境。

在一些实施例中，打包机构可以生成多份多个应用文件，每一个应用文件包括不同版本的应用程序，这些应用程序被配置成在不同目标环境下执行。在这些实施例之一中，将多个应用文件配置成在具有特定操作系统、修订级、语言配置和主驱动器的本地机上执行（例如，可以将一份多个应用文件配置成在具有 Windows XP Professional 操作系统、修订级 SP2 和以上，使用英语且主驱动器为 C:\ 的本地机上执行）。在这些实施例的另一个中，可以将超过一份多个应用文件组合成单个档案文件。在这些实施例的又一个中，可以将每份多个应用文件称为“目标”。在这些实施例的又一个中，可以将含有一份或多份多个应用文件的档案文件称为“程序包”。

现在参考图 19，方框图示出了包括两个目标的程序包，每个目标包括包括应用的多个应用文件。在图 19 中，将应用程序“Foo”打包装成两个目标。两个目标之间的差异为“目标语言”。具体而言，目标 1 支持“英语”，目标 2 支持“德语”。在一个实施例中，枚举可用应用程序可以列出应用程序“Foo”。在另一个实施例中，将适当的多个文件发送到请求访问应用程序的本地机。在又一实施例中，响应于对

本地机的评估做出决定，以向本地机发送特定目标。在又一个实施例中，与程序包相关联的文件标识与程序包中的目标相关联且在本地机上执行所需的至少一个特征。

在一些实施例中，打包机构 530 通过执行与应用程序相关联的安装程序来准备要进行流传输的应用程序。在这些实施例之一中，打包机构在执行打包机构的远程机 30 上产生隔离环境。在这些实施例的另一个中，打包机构在隔离环境中执行应用程序。在这些实施例的另一个中，打包机构标识由安装程序生成或修改的多个应用文件。在这些实施例的另一个中，打包机构生成包括多个应用文件的档案文件。在这些实施例之一中，打包机构生成包括多个应用文件的.CAB 文件。在这些实施例的另一个中，打包机构生成目录并在目录中存储多个应用文件。在一些实施例中，打包机构在文件服务器或其他远程机 30 上存储多个应用文件。在其他实施例中，打包机构在多个远程机上存储多个应用文件。

现在参考图 20，流程图示出了在用于无须重新引导操作系统而有效安装应用程序的基于策略的方法中采取的步骤的一个实施例。简而言之，打包机构在隔离环境之内执行安装程序，安装程序将与第二应用相关联的至少一个应用文件安装到隔离环境中（步骤 2002）。拦截安装程序对至少一个应用编程接口（API）的调用，该调用需要在重新引导操作系统之后执行动作（步骤 2004）。不重新引导操作系统而执行至少一个被拦截调用的动作（步骤 2006）。接收至少一个应用文件的文件类型标识（步骤 2008）。响应于被标识的文件类型将至少一种执行方法与至少一个已安装应用文件相关联（步骤 2010）。在至少一个服务器上存储至少一个已安装应用文件（步骤 2012）。生成第二应用、至少一个已安装应用文件、至少一个服务器位置和至少一个执行方法的枚举（步骤 2014）。

现在参考图 20，更详细地讲，打包机构在隔离环境之内执行安装程序，安装程序将与第二应用相关联的至少一个应用文件安装到隔离环境中（步骤 2002）。在一个实施例中，在隔离环境之内执行安装程序使打包机构能够隔离安装程序对本地机上的文件或注册表所做的更改。在另一个实施例中，打包机构拦截安装程序请求的更改并将更改重定向到隔离环境以防止本地机上发生更改。在又一实施例中，打包

机构在隔离环境之内执行第二安装程序，第二应用将与第三应用相关联的至少一个应用文件安装到隔离环境中。

在一些实施例中，打包机构在隔离环境之内执行安装程序，安装程序在隔离环境内部执行与应用相关联的至少一个可执行应用。在安装程序执行应用的一个实施例中，执行应用使得能够安装第二应用。

在这些实施例的另一个中，除了执行安装程序之外，安装应用还需要执行至少一个可执行应用。在这些实施例的又一个中，除了执行安装程序之外，安装应用需要执行因特网（Internet）浏览器应用。在一些实施例中，执行安装程序以安装程序，执行安装程序包括执行安装程序所需的第二程序。在这些实施例之一中，该程序为插件。在这些实施例的另一个中，该程序为活动 X 部件。在这些实施例的又一个中，该程序为 Flash 部件。在这些实施例的又一个中，该程序为定制工具条，例如 Yahoo! 或 Google 工具条。在其他实施例中，该程序为安装到第二程序中的部件，不可独立于第二程序执行。

拦截安装程序对至少一个应用编程接口（API）的调用，该调用需要在重新引导操作系统之后执行动作（步骤 2004）。不重新引导操作系统而执行至少一个被拦截调用的动作（步骤 2006）。在一些实施例中，执行动作包括执行在安装期间修改的注册表项的动作。参考以下的图 25 提供了关于不重新引导操作系统而执行至少一个被拦截调用的更多细节。

接收至少一个应用文件的文件类型标识（步骤 2008）。响应于被标识的文件类型将至少一种执行方法与至少一个已安装应用文件相关联（步骤 2010）。在一个实施例中，至少一种执行方法使得能将至少一个应用文件流传输到客户端。在另一个实施例中，该至少一种执行方法使得能在客户端上执行至少一个已安装应用文件。在又一个实施例中，至少一种执行方法使得能在服务器上执行至少一个已安装应用文件。在又一个实施例中，该至少一种执行方法使得能将至少一个应用文件流传输到服务器。

在至少一个服务器上存储至少一个已安装应用文件（步骤 2012）。在一些实施例中，在将至少一个已安装应用文件存储在至少一个服务器上之前在隔离环境内执行已安装应用程序。在这些实施例之一中，响应于执行已安装应用程序生成额外的应用文件。在这些实施例的另

一个中，生成数据文件。在这些实施例的又一个中，已安装应用程序需要信息来完成安装，在初始安装过程之后需要该信息。在这些实施例的又一个中，需要诸如软件产品标识符、许可标识符或其他信任状的信息。

在一些实施例中，提供标识符来标识至少一个已安装应用文件在至少一个服务器上的位置。在这些实施例之一中，标识符符合通用命名标准（UNC）。在其他实施例中，将至少一个已安装应用文件置于诸如.CAB文件的档案文件中。在这些实施例之一中，在档案文件中存储多个应用文件，在至少一个服务器上存储档案文件。在这些实施例的又一个中，在多个服务器上存储至少一个已安装应用文件。在其他实施例中，将至少一个应用文件置于存储应用文件的目录中。

生成第二应用、至少一个已安装应用文件、至少一个服务器位置和至少一个执行方法的枚举（步骤 2014）。在一些实施例中，在文件中存储枚举。在其他实施例中，在清单文件中存储枚举。在其他实施例中，在 XML 文件中存储枚举。

在一个实施例中，生成多个应用、与多个应用的每个相关联的多个已安装应用文件以及存储多个已安装应用文件的至少一个服务器的位置的枚举。在另一个实施例中，生成包括第二应用与多个已安装应用文件之间的关联的枚举。在一实施例中，生成包括第二应用与含有至少一个已安装应用文件的压缩文件之间的关联的枚举。

现在参考图 21，流程图示出了在用于无须重新引导操作系统而安装应用程序的基于策略的方法中采取的步骤的一个实施例。简而言之，打包机构在隔离环境之内执行安装程序，安装程序将与第二应用相关联的至少一个应用文件安装到隔离环境中（步骤 2102）。拦截安装程序对至少一个应用编程接口（API）的调用，该调用需要在重新引导操作系统之后执行动作（步骤 2104）。不重新引导操作系统而执行至少一个被拦截调用的动作（步骤 2106）。接收至少一个应用文件的特征标识（步骤 2108）。响应于被标识的特征将至少一种执行先决条件与至少一个已安装应用文件相关联（步骤 2110）。在至少一个服务器上存储至少一个已安装应用文件（步骤 2112）。生成第二应用、至少一个已安装应用文件、至少一个服务器位置和至少一个执行先决条件的枚举（步骤 2114）。

现在参考图 21，更详细地讲，打包机构在隔离环境之内执行安装程序，安装程序将与第二应用相关联的至少一个应用文件安装到隔离环境中（步骤 2102）。在一个实施例中，在隔离环境之内执行安装程序使打包机构能够隔离安装程序对本地机上的文件或注册表所做的更改。在另一个实施例中，打包机构拦截安装程序请求的更改并将更改重定向到隔离环境以防止本地机上发生更改。在又一实施例中，打包机构在隔离环境之内执行第二安装程序，第二应用将与第三应用相关联的至少一个应用文件安装到隔离环境中。

在一些实施例中，打包机构在隔离环境之内执行安装程序，安装程序在隔离环境内部执行与应用相关联的至少一个可执行应用。在安装程序执行应用的一个实施例中，执行应用使得能够安装第二应用。在这些实施例的另一个中，除了执行安装程序之外，安装应用还需要执行至少一个可执行应用。在这些实施例的又一个中，除了执行安装程序之外，安装应用需要执行因特网浏览器应用。

向前参考图 23，方框图示出了包括打包机构 530 和文件系统过滤器驱动程序 534 的系统的一个实施例，打包机构 530 将安装程序 2350 执行到隔离环境 532 中，文件系统过滤器驱动程序 534 与打包机构 530 和隔离环境 532 通信。

在一个实施例中，打包机构 530 通过将应用程序安装到隔离环境 532 中产生程序包（如以上结合图 21 所述）。在另一个实施例中，打包机构 530 通过执行安装程序 2350 来将应用程序安装到隔离环境 532 中。在一些实施例中，打包机构 530 包括图形用户界面。在这些实施例之一中，图形用户界面使打包机构 530 的用户能够定制由打包机构 530 产生程序包的过程。在这些实施例的另一个中，打包机构 530 与访问控制套件 520 上的图形用户界面通信，使得访问控制套件 520 的用户能够定制打包机构 530 产生程序包的过程。

在一些实施例中，文件系统过滤器驱动程序 532 使得能在隔离环境 532 中安装应用程序。在这些实施例之一中，文件系统过滤器驱动程序 532 拦截安装程序 2350 的请求。在这些实施例的另一个中，文件系统过滤器驱动程序 532 将安装程序 2350 的请求重定向到隔离环境 532。在这些实施例之一中，文件系统过滤器驱动程序 532 存储安装程序 2350 所提请求的记录。在这些实施例又一个中，文件系统过滤器驱

动程序 532 存储安装程序 2350 生成或修改的文件拷贝。在一些实施例中，将所存储的由文件系统过滤器驱动程序 532 生成的记录作为包括应用程序的多个应用文件而存储在一起。在其他实施例中，在文件服务器 540 上存储多个应用文件。

返回到图 21，拦截安装程序对至少一个应用编程接口（API）的调用，该调用需要在重新引导操作系统之后执行动作（步骤 2104）。不重新引导操作系统而执行至少一个被拦截调用的动作（步骤 2106）。在一些实施例中，执行该动作包括安装驱动程序，该驱动程序被配置成在引导计算机系统时启动。在其他实施例中，执行动作包括执行在安装期间修改的注册表项的动作。

接收至少一个应用文件的特征标识（步骤 2108）。在一些实施例中，接收操作系统类型的标识。在其他实施例中，接收操作系统所用语言的标识。在其他实施例中，接收第二应用版本的标识。

响应于被标识的特征将至少一种执行先决条件与至少一个已安装应用文件相关联（步骤 2110）。在一个实施例中，响应于向特征应用策略将至少一种执行先决条件与至少一个已安装应用文件相关联。在另一个实施例中，脚本与至少一个已安装应用文件相关，该脚本包括可执行程序，该可执行程序判断客户端上是否存在至少一个执行先决条件。向前参考图 22，屏幕快照示出了要在本地机上执行的脚本枚举的一个实施例。脚本 2202 的类型表示应当何时执行该脚本，例如，在执行应用之前，或在终止执行应用之后。隔离指示符 24 表示是否应当在本地机 10 上的隔离环境中执行脚本。如图 22 所示，在一些实施例中，在将多个应用文件打包到一起并存储在持有多个应用文件的远程机 30' 上时，将脚本与应用程序相关联。

在一些实施例中，至少一个执行先决条件要求在执行至少一个已安装应用文件的系统上安装操作系统的版本。在其他实施例中，至少一个执行先决条件要求在执行至少一个已安装应用文件的系统上安装第二应用的版本。在其他实施例中，将一指令与至少一个已安装应用文件相关联，该指令表明将由客户端使用的第二已安装应用文件未能满足至少一个执行先决条件。在其他实施例中，将一指令与至少一个已安装应用文件相关联，该指令表明用于在客户端上执行至少一个已安装应用文件的第二执行方法未能满足至少一个执行先决条件。在这

些实施例之一中，将一执行方法与至少一个已安装应用文件相关联，该执行方法授权将包括第二应用的多个应用文件流传输到本地机以在本地机上执行。在这些实施例的另一个中，本地机的评估标识与本地机上未包括的至少一个已安装应用文件相关联的至少一个特征。在这些实施例的又一个中，收回对执行多个应用文件的授权。在这些实施例的又一个中，提供第二执行方法用于执行多个应用文件，第二执行方法使得能在远程机上执行多个应用文件并从远程机向本地机传输应用输出数据。

在至少一个服务器上存储至少一个已安装应用文件（步骤 2112）。在一些实施例中，在将至少一个已安装应用文件存储在至少一个服务器上之前在隔离环境内执行已安装应用程序。在这些实施例之一中，响应于执行已安装应用程序生成额外的应用文件。在这些实施例的另一个中，生成数据文件。在这些实施例的又一个中，已安装应用程序需要信息来完成安装，在初始安装过程之后需要该信息。在这些实施例的又一个中，需要诸如软件产品标识符、许可标识符或其他信任状的信息。

在一些实施例中，提供标识符来标识至少一个已安装应用文件在至少一个服务器上的位置。在这些实施例之一中，标识符符合通用命名标准（UNC）。在其他实施例中，将至少一个已安装应用文件置于诸如.CAB 文件的档案文件中。在这些实施例之一中，在档案文件中存储多个应用文件，在至少一个服务器上存储档案文件。在这些实施例的又一个中，在多个服务器上存储至少一个已安装应用文件。在其他实施例中，将至少一个已安装应用文件置于存储应用文件的目录中。

生成第二应用、至少一个已安装应用文件、至少一个服务器位置和至少一个执行先决条件的枚举（步骤 2114）。在一些实施例中，在文件中存储枚举。在其他实施例中，在清单文件中存储枚举。在其他实施例中，将枚举存储在 XML 文件中。

在一个实施例中，生成多个应用、与多个应用的每个相关联的多个已安装应用文件以及存储多个已安装应用文件的至少一个服务器的位置的枚举。在另一个实施例中，生成包括第二应用与多个已安装应用文件之间的关联的枚举。在一实施例中，生成包括第二应用与含有至少一个已安装应用文件的压缩文件之间的关联的枚举。

返回步骤 2106，在不重新引导操作系统而执行至少一个被拦截调用的动作时，在一些实施例中，提供虚拟化安装和执行环境，其消除了在执行已安装应用之前重新引导系统的需要。

现在参考图 24，流程图示出了执行安装程序需要重新引导执行安装程序的本地机上的操作系统的实施例。常规的应用安装程序将文件拷贝到被安装应用的远程机上（步骤 2402）。在一些实施例中，拷贝文件可能导致重新引导远程机。应用安装程序试图将至少一个文件拷贝到锁定文件（步骤 2404）。在一个实施例中，仅在执行操作系统（或“重新引导”）时可以对锁定文件写入。在 MoveFileEx() Win32 API 中设置 MOVE_FILE_DELAY_UNTIL_REBOOT 选项（步骤 2406），应用安装程序调用系统关闭/重新引导功能（步骤 2408）。重新引导之后，则在重新引导时安装一开始锁定的文件（步骤 2410）。

现在参考图 25，方框图示出了打包机构在其上安装应用程序的远程机 30 的一个实施例。远程机 30 包括系统资源 2502、系统 API 2504 和用于安装应用的应用安装程序 2506。远程机 30 还包括功能转播机构 2508、安装后处理器模块 2510 和应用隔离环境 2512。在一些实施例中，将应用程序安装到隔离环境 2512 中使得能不重新引导远程机 30 而进行安装。在这些实施例之一中，对隔离环境 2512 中虚拟化的系统资源 2502 所做的更改不改变远程机 30 上的相应系统资源 2502。由于未更改远程机 30 上的系统资源，因此不需要重新引导机器以保护系统资源不被进行不当更改。

现在参考图 25，更详细地讲，系统资源 2502 可以包括注册表项、系统 DLL 及其他操作系统防止在远程机 30 执行时写入的锁定文件。系统 API 2504 包括用于重新引导被应用安装程序 2506 调用且被功能转播机构 2508 转播的系统的 API，以防止重新引导远程机 30。

应用隔离环境 2512 向应用安装程序 2506 提供了具有操作系统资源的视图的环境。在一个实施例中，应用隔离环境 2512 为隔离环境 556。在一些实施例中，应用隔离环境 2512 提供了诸如文件系统、注册表和命名对象的虚拟化。在一个实施例中，应用安装程序 2506 在应用隔离环境 2512 之内执行。在另一个实施例中，应用安装程序 2506 将应用程序安装到应用隔离环境 2512 中。在又一个实施例中，应用安装程序 2506 在应用隔离环境 2512 之外执行并在应用隔离环境 2512 内

安装应用程序。

在一些实施例中，在应用安装程序 2506 向应用隔离环境 2512 中安装应用时，应用隔离环境 2512 避开了重新引导远程机 30 的需要。在一个实施例中，应用隔离环境 2512 拦截将应用文件拷贝到锁定文件的请求。在另一个实施例中，应用隔离环境 2512 将拷贝应用文件的请求重定向到未锁定文件。在又一个实施例中，应用隔离环境 2512 将拷贝应用文件的请求重定向到虚拟化文件。在又一个实施例中，对拷贝应用文件的请求重定向使得能够在无须重新引导远程机 30 的情况下安装应用文件。例如，如果应用安装程序 2506 试图向锁定文件写入，例如向 c:\windows\system32\mfcc40.d11 写入，则应用隔离环境 2512 拦截该请求并将文件重定向到另一未锁定位置。这种避免锁定文件的能力 表 示 可 以 无 须 利 用 MoveFileEx0 API 和 MOVE_FILE_DELAY_UNTIL_REBOOT 标记来安装文件。这种能力消除了对重新引导远程机 30 的需要。

在一个实施例中，功能转播机构 2508 是文件系统过滤器驱动程序 564。在另一个实施例中，文件系统过滤器驱动程序 564 包括功能转播机构 2508。在又一个实施例中，功能转播机构 2508 拦截来自应用安装程序 2506 的请求以重新启动远程机 30。在一些实施例中，应用隔离环境 2512 提供了将应用文件拷贝到未锁定文件。然而，应用隔离环境 2512 不解决应用安装程序 2506 要求重新引导远程机 30 的请求。功能转播机构 2508 拦截重新引导请求并对应用安装程序 2506 做出响应。

应用隔离环境 2512 使得能将应用文件拷贝到未锁定文件。然而，在一些实施例中，安装应用需要其他动作，这些动作可以在重新引导时发生。防止重新引导并不阻止在安装过程中完成这些动作的需要。功能转播机构 2508 可以提供执行与安装应用相关的动作的功能。

例如，在安装应用期间，可以写入注册表项，例如 HKLM\SYSTEM\CurrentControlSet\Control\Session-Manager\Pending-FileRenameOperations。其他应用可以安装需要在引导机器时启动的服务或驱动程序。安装后处理器模块 2510 识别在安装期间被修改的应用文件，并执行与该应用文件相关联的动作。

现在参考图 26，流程图示出了在应用隔离环境 2512 中安装应用所采取的步骤的一个实施例。应用隔离环境 2512 向应用安装程序提供服

务器操作系统的虚拟化视图（步骤 2602）。对涉及系统重新引导和停机的服务器上的 API 进行转播（步骤 2604）以防止应用安装程序 2506 导致重新引导。应用安装程序 2506 请求到锁定文件的文件拷贝操作，该请求被拦截并重定向到不冲突的位置（步骤 2606）。当应用安装程序 2506 试图通过调用系统 API 重新引导时，拦截该请求并防止重新引导（步骤 2608）。安装后处理器模块 2510 执行在重新引导之后通常发生动作（步骤 2610），然后可以不用重新引导远程机 30 而在应用隔离环境 2512 中执行应用（步骤 2612）。

在一些实施例中，将应用程序安装到应用隔离环境 2512 中之后，打包机构标识在安装应用程序期间生成或修改的多个应用文件。在这些实施例之一中，在远程机上存储多个应用文件。在这些实施例的另一个中，检索多个应用文件的本地机可以执行该应用程序。

在一些实施例中，打包机构 530 在包括隔离环境 532 和文件系统过滤器驱动程序 534 的远程机上执行并将应用程序安装到隔离环境 532 中。在这些实施例之一中，该远程机被称为“清洁机（clean machine）”或“分级机”。在这些实施例的另一个中，隔离环境 532 包括应用隔离范围，该应用隔离范围提供由清洁机上的操作系统提供的本地资源的可修改、虚拟化实例。在这些实施例的又一个中，隔离环境 532 包括提供本地资源的只读视图的系统隔离范围。在这些实施例的又一个中，本地资源的只读视图包括文件系统的快照和清洁机上驻留的注册表。

在一个实施例中，重定向器拦截要求更改本地资源的请求。在一些实施例中，重定向器为文件系统过滤器驱动程序 534。在另一个实施例中，由打包机构 530 执行的安装程序提出更改请求。在又一个实施例中，将应用程序安装到清洁机上需要对本地资源进行更改。在又一个实施例中，重定向器将该请求重定向到隔离环境 532。

在一些实施例中，将更改本地资源的请求重定向到隔离环境 532 造成与安装应用程序相关联的更改被隔离。在其他实施例中，将更改本地资源的请求记录并存储在存储元件中。在这些实施例之一中，与安装应用程序相关联的所有更改都驻留在存储元件中。在这些实施例的另一个中，本地机 552 检索存储元件的内容并对驻留在本地机 552 上的隔离环境 556 中的本地资源实施更改，这导致在本地机 552 上安

装应用程序。

在一些实施例中，可能需要对本地机 10 进行启动前分析。在这些实施例之一中，本地机 10 验证至少一个特征包括在本地机 10 中。在这些实施例的另一个中，在启动前分析判定本地机 10 缺少至少一个特征之后将该至少一个特征添加到本地机 10。在这些实施例的又一个中，持有应用程序的远程机包括该至少一个特征且本地机未能包括该至少一个特征将阻止执行应用程序。在又一个实施例中，该应用程序要求要执行的本地机上存在至少一个特征。

在一些实施例中，打包机构使得能识别至少一个特征，以在对本地机进行启动前分析时使用。在其他实施例中，打包机构使得能将至少一个特征与可以在本地机上执行的应用程序相关联。在其他实施例中，打包机构使得能将可执行脚本与应用程序相关联，本地机执行该可执行脚本以完成启动前分析。在其他实施例中，在执行应用程序之后需要本地机上存在该至少一个特征。

打包机构可以提供签署多个应用文件的功能。在一个实施例中，签署多个应用文件使得本地机能够验证多个应用文件的完整性。在另一个实施例中，签署多个应用文件防止本地机执行被破坏的应用程序。在一些实施例中，计算多个应用文件中文件的密码校验和，例如 MD4 散列码、MD5 散列码或 SHA-1 散列码。

在其他实施例中，计算多个应用文件中每个文件的密码校验和。在这些实施例之一中，在第二文件中存储密码校验和。在这些实施例的另一个中，将第二文件与多个应用文件相关联。在一些实施例中，将第二文件添加到多个应用文件。在其他实施例中，利用诸如 X.509 证书的证书签署第二文件。在其他实施例中，检索多个应用文件的本地机利用证书的公开部分验证签名。在其他实施例中，本地机接收证书的公开部分和证书委托列表的标识以验证签名。在这些实施例之一中，本地机接收包含证书委托列表的标识的注册表项。

在一个实施例中，打包机构提供了用于定制隔离环境的功能。在另一个实施例中，打包机构提供了用于生成存储隔离环境定义的文件的功能。在又一个实施例中，打包机构包括具有多个应用文件的文件，该多个应用文件包括应用程序。在又一个实施例中，本地机从远程机接收具有访问信息的文件。

在一些实施例中，将多个应用文件存储在档案文件中。在这些实施例之一中，该档案文件为 CAB 文件格式。在这些实施例的另一个中，该档案文件格式不支持由应用程序对文件的短文件名进行规格说明。在这些实施例的又一个中，诸如 WINDOWS 2000 的操作系统可能不支持由应用程序对文件的短文件名做规格说明。在其他实施例中，诸如 WINDOWS XP 的操作系统支持由应用程序对文件的短文件名做规格说明。在这些实施例之一中，执行文件的请求必需包括文件的正确文件名。

在一个实施例中，可以生成映射以将多个应用文件中文件的长文件名与文件的短名相关联。在另一个实施例中，在多个应用文件中的文件中存储映射。在又一个实施例中，仅当文件的长文件名长于 12 个字符时文件才具有短文件名。在一些实施例中，短文件名是与该文件相关联的虚拟文件名。在这些实施例之一中，在文件以长文件名存储的情况下，将该文件发送到本地机 10 来执行。在这些实施例的另一个中，本地机 10 上的应用文件利用短文件名请求执行文件。在这些实施例的又一个中，尽管执行文件的请求未使用本地机上的文件名称（长文件名），但映射使得能够执行该文件。

在一些实施例中，打包机构 530 生成该映射。在这些实施例之一中，打包机构 530 为具有长文件名的文件选择短文件名。在这些实施例的另一个中，执行打包机构 530 的远程机 30' 上的操作系统为具有长文件名的文件选择短文件名。在这些实施例的又一个中，选择不与远程机 30' 上的第二个短文件名冲突的唯一短文件名。在这些实施例的又一个中，由打包机构 530 执行的安装程序生成包括长文件名和短文件名之间映射的文件。在其他实施例中，将该映射发送到检索该文件的本地机 10。在这些实施例之一中，本地机 10 在执行文件时引用该文件。

以下示例性实示例出了可以如何利用上述方法和系统来选择包括应用程序的多个文件，将其流传输到本地机并在本地机上执行。这些实例是为了示例而非限制本发明。

实例 1

在一个实施例中，本地机 10 的用户请求访问应用程序枚举中标识的应用程序，例如文字处理程序、web 浏览应用或数据表程序。在本实施例的一个实例中，本地机 10 执行程序邻域应用，该程序邻域应用从

远程机 30 接收本地机 10 可用的应用枚举。在本实施例的另一个实例中，本地机 10 与诸如远程机 30' 的 web 服务器通信，以接收应用枚举。本地机 10 的用户可以通过选择代表被枚举的应用程序的图形表示来请求访问被枚举的应用程序。本地机 10 的用户可以请求访问此前未安装在本地机 10 上的应用程序。

本地机 10 将访问应用程序的请求发送到远程机 30。本地机 10 接收远程机 30" 的标识，该远程机 30" 提供对包括应用程序的多个应用文件的访问。本地机 10 标识执行应用程序所需的至少一个特征。在本实施例的一个实例中，本地机 10 接收由远程机 30 发送到本地机 10 的具有远程机 30" 的标识的至少一个特征。在本实施例的另一个实例中，本地机 10 在接收到远程机 30" 的标识之后从远程机 30" 检索至少一个特征。可以要求本地机 10 在接收到检索多个应用文件的授权之前包括至少一个特征。或者，可以要求本地机 10 在执行多个应用文件之前包括至少一个特征。在本实施例的一个实例中，可以要求本地机 10 在执行多个应用文件的整个期间包括至少一个特征。

在本地机 10 验证本地机 10 包括至少一个特征时，本地机 10 检索多个应用文件中的至少一个应用文件并执行所检索的应用文件以执行应用程序。

实例 2

远程机 30 从本地机 10 接收访问应用程序的请求。远程机 30 对本地机 10 进行认证。在本实施例的一个实例中，远程机 30 从本地机 10 请求信任状，例如用户名和口令。在本实施例的另一个实例中，远程机 30 向本地机 10 发送收集代理 404。收集代理 404 收集有关本地机 10 的信息并将该信息发送到远程机 30 以便在认证本地机 10 时使用。在本实施例的又一个实例中，远程机 30 向策略引擎 406 提供有关本地机 10 的信息，用于对本地机 10 进行认证。远程机 30 可以包括策略引擎 406。或者，远程机 30 可以与包括策略引擎 406 的远程机 30' 通信。

远程机 30 选择应用程序的执行方法。远程机 30 可以响应于对本地机 10 的认证做出选择。在本实施例的一个实例中，远程机 30 向所收集的本地机 10 的信息采用策略。在本实施例的另一个实例中，远程机 30 响应于用于应用程序的策略做出选择。在本实施例的又一个实例中，远程机 30 响应于用于与应用程序相关的文件类型的策略做出选择。

远程机 30 可以查询文件以选择应用程序的执行方法。远程机 30 可以选择使本地机 10 能够接收在远程机 30' 上执行应用程序所产生的应用输出数据的应用程序执行方法。远程机 30 可以选择使本地机 10 能够在检索包括应用程序的多个应用文件之后在本地执行应用程序的应用程序执行方法。

在一个实施例中，远程机 30 选择使本地机 10 能够在通过应用流传输会话检索包括应用程序的多个应用文件之后在本地执行应用程序的应用程序执行方法。在本实施例的一个实例中，本地机 10 与持有多个应用文件的远程机建立应用流传输会话，本地机 10 跨过应用流传输会话启动多个应用文件的检索，且本地机 10 在检索多个应用文件中的第二应用文件时执行所检索到的多个应用文件中的第一应用文件。在本实施例的另一个实例中，本地机 10 在从第一应用接收到访问第二应用文件的请求时执行多个应用文件中的第一应用文件并检索多个应用中的第二应用文件。

对于所选择的执行方法使本地机 10 能够在包括应用程序的多个应用文件中检索至少一个应用文件的实施例而言，远程机 30 标识持有可由本地机 10 访问的应用程序的远程机 30"。远程机 30"持有包括应用程序的多个应用文件。

远程机 30"可以持有多份包括各种应用程序的多个应用文件。在本实施例的一个实例中，远程机 30"持有针对应用程序若干不同版本的每种的多个应用文件。远程机 30"持有将包括特定应用程序的多个应用文件与应用程序的描述相关联的文件。该文件还可以标识在向机器发送多个应用文件之前需要在机器上标识的一个或多个执行先决条件。该文件还可以包括远程机 30"在网络上位置的标识。在本实施例的一个实例中，远程机 30 查询文件以标识远程机 30"在网络上位置。

远程机 30 选择远程机 30"。远程机 30 可以选择其在网络上的位置可以被本地机 10 访问到的远程机 30"。远程机 30 可以选择持有与本地机 10 兼容的应用程序版本的远程机 30"。远程机 30 响应于接收到访问应用程序的请求，向本地机 10 发送所选择的应用程序执行方法的标识以及远程机 30"的标识。远程机 30 还可以向本地机 10 发送文件。

实例 3

在一个实施例中，本地机 10 接收所选择的应用程序执行方法的标

识以及提供对包括应用程序的多个应用文件的访问的远程机 30" 的标识。本地机 10 验证对访问应用程序的授权。在本实施例的一个实例中，本地机 10 执行自身的启动前分析。本地机 10 标识至少一个特征并验证本地机 10 上是否存在至少一个特征。该至少一个特征可以是维持访问和执行应用程序的授权的先决条件。验证本地机 10 上存在至少一个特征可以确保本地机 10 的特征和应用程序的系统需求之间的兼容性，并可以额外确保与安全政策或许可协议的遵守。

在成功完成启动前分析时，本地机 10 与提供对多个应用文件的访问的远程机 30" 建立应用流传输会话。应用流传输会话可以是本地机 10 可以请求和接收多个应用文件中的文件的任何连接。建立应用流传输会话可以使本地机 10 能够在检索多个应用文件中的所有文件之前执行多个应用文件中的第一应用文件。本地机 10 可以启动执行应用程序，同时继续检索多个应用文件中的其他应用文件。或者，本地机 10 可以在档案文件中检索多个应用文件并在从档案文件中提取第二应用文件的同时执行第一个提取的应用文件。

实例 4

在一个实施例中，本地机 10 上的应用流传输客户端 552 从远程机 30 检索多个应用文件。应用流传输客户端包括流传输服务 554、隔离环境 556 和文件系统过滤器驱动程序 564。流传输服务 554 与远程机 30 建立应用流传输会话，用于请求和检索多个应用文件。流传输服务 554 在隔离环境 556 之内执行应用文件。文件系统过滤器驱动程序 564 通过拦截来自执行应用文件的请求并将该请求重定向到隔离环境 556 使得能够在隔离环境 556 之内执行应用文件。

在本实施例的一个实例中，流传输服务 554 检索包括多个应用文件的档案文件，该多个应用文件包括应用程序。流传输服务 554 从档案文件中提取来自多个应用文件的第一应用文件。第一应用文件可以是可执行文件。流传输服务 554 可以在隔离环境 556 之内执行第一应用文件。执行第一应用文件可以启动应用程序的执行。

在另一个实施例中，在隔离环境 556 中执行的第一应用文件从本地机 10 请求对多个应用文件的枚举。文件系统过滤器驱动程序 564 拦截枚举请求并将请求重定向到流传输服务 554。在流传输服务 554 检索多个应用文件的实施例中，流传输服务 554 可以生成多个应用文件的

枚举。在流传输服务 554 检索包括多个应用文件的档案文件的实施例中，流传输服务 554 可以响应于所检索的档案文件中包括的枚举生成多个应用文件的枚举。在其他实施例中，在多个应用文件中的至少一个应用文件驻留在远程机 30 上且还没有被流传输服务 554 检索到本地机 10 上的时候，流传输服务 554 仅检索多个应用文件的枚举。在这些实施例中，流传输服务 554 可以响应于所检索的枚举产生多个应用文件的枚举。在这些实施例的一个实例中，虽然本地机 10 仅有枚举，但流传输服务 554 向第一应用文件表明多个应用文件驻留在本地机 10 上。

实例 5

在一个实施例中，在隔离环境 556 中执行的第一应用文件从本地机 10 请求对多个应用文件的枚举标识的文件进行访问。如果所请求的文件在第一应用文件可访问的隔离环境 556 之内的用户范围内，第一应用文件访问所请求的文件。

如果所请求的文件不在用户范围或隔离环境 556 中，文件系统过滤器驱动程序 564 拦截该请求并将该请求重定向到流传输服务 554。如果所请求的文件是含有多个应用文件的档案文件中的文件，流传输服务 554 提取所请求的文件并将所请求的文件存储在本地机 10 上。流传输服务 554 可以在隔离环境 556 之内存储该文件。当文件被存储在隔离环境 556 中时满足对该文件的请求。

如果所请求的文件不在隔离环境 556 中或包括多个应用文件的档案文件中，流传输服务 554 从远程机 30 请求该文件。流传输服务 554 可以通过应用流传输会话从远程机 30 接收文件。流传输服务 554 将所接收的文件存储在隔离环境 556 中。当文件被存储在隔离环境 556 中时满足对该文件的请求。

在本实施例的一个实例中，在隔离环境 556 中的第二用户范围中执行第二应用文件。第二应用文件请求访问最初由第一应用文件请求的文件。如果所请求的文件的拷贝不在第二用户范围内，使用隔离环境 556 中存储的所请求的文件拷贝满足对应用文件的请求。

实例 6

在一个实施例中，本地机 10 从远程机 30 接收所选择的应用程序执行方法的标识以及提供对包括应用程序的多个应用文件的访问的远

程机 30' 的标识。本地机 10 成功完成本地机 10 的启动前分析。本地机 10 从远程机 30 接收授权执行应用程序的许可。在本实施例的一个实例中，该许可要求本地机 10 向会话管理服务器 562 发送心跳消息以维持执行应用程序的授权。心跳消息可以包括表示启动执行应用程序、终止执行应用程序的消息以及在执行应用程序的整个期间周期性发送的消息。心跳消息还可以包括关于本地机 10 的状态的消息，比如本地机 10 何时连接到网络，或本地机 10 何时终止与网络的连接。在本实施例的另一个实例中，该许可执行预定时间段，在该预定时间段内本地机 10 具有执行应用程序的授权。

本地机 10 与远程机 30' 建立应用流传输会话并在多个应用文件中检索至少一个应用文件。在执行至少一个应用文件期间，在所接收的许可要求发送心跳消息的实施例中，本地机 10 向会话管理服务器 562 发送心跳消息以维持执行至少一个应用文件的授权。

实例 7

在一个实施例中，本地机 10 接收所选择的应用程序执行方法的标识以及提供对包括应用程序的多个应用文件的访问的远程机 30' 的标识。本地机 10 成功完成本地机 10 的启动前分析。本地机 10 接收指定预定时间段的许可，在该预定时间段内本地机 10 具有执行应用程序的授权。

本地机 10 与远程机 30' 建立应用流传输会话并在多个应用文件中检索至少一个应用文件。在本实施例的一个实例中，本地机 10 检索多个应用文件的子集，该子集包括在本地机 10 未连接到网络时执行应用程序必需的每个文件。本地机 10 将该子集存储在本地机 10 上的高速缓冲存储器中。

在该预定时间段内的时间点，从网络断开本地机 10，本地机 10 从本地机 10 的用户接收访问应用程序的请求。在本实施例的一个实例中，本地机 10 是诸如膝上计算机的装置，本地机 10 的用户处于禁止连接到网络的环境中，例如处在飞机中。在从用户接收到该请求时，本地机 10 可以从高速缓冲存储器检索来自多个应用文件的应用文件并执行该应用程序。

实例 8

在另一个实施例中，本地机 10 接收所选择的应用程序执行方法的

标识以及提供对包括应用程序的多个应用文件的访问的远程机 30' 的标识。本地机 10 可以接收驻留在本地机 10 上以检索多个应用文件的第一客户端代理的标识，该第一客户端代理例如为应用流传输客户端。

在本实施例的一个实例中，本地机 10 未能成功完成自身的启动前分析。本地机 10 可能缺少与应用程序要求兼容所需的特征，例如特定的设备驱动程序或操作系统。本地机 10 可能缺少遵守安全策略所需的特征，该安全策略例如是特定活动目录中的成员身份或对访问专用网络的授权。

本地机 10 可以是一种与应用程序的要求兼容的机器，例如试图访问计算强化的应用程序的个人数字助理，或试图执行由专用网络上的远程机持有的安全应用的信息站处的公共机器。本地机 10 响应于本地机 10 缺少访问应用程序所需的至少一个特征的判断做出不通过应用流传输会话检索多个应用文件的决定。本地机 10 执行驻留在本地机 10 上的第二客户端代理而不是执行所标识的第一客户端代理。在本实施例的一个实例中，本地机 10 在无法成功完成启动前分析的情况下接收要执行的第二客户端代理的标识。本地机 10 请求在远程机 30" 上执行应用程序。第二客户端代理接收在远程机 30" 上执行应用程序产生的应用输出数据。第二客户端代理在本地机 10 上显示应用输出数据。

实例 9

在一个实施例中，网络管理员为本地机 10 的用户提供对应用程序的访问。管理员在远程机 30' 上执行应用以产生包括应用程序的多个应用文件。该应用可以包括图形用户界面。管理员可以使用图形用户界面来标识应用程序以及与该应用程序相关联的安装程序，定义要在授权访问应用程序时所采用的策略，并指定有关于所提供的访问类型的特征，包括试图访问或执行应用程序的本地机 10 要满足的要求。管理员可以标识安装整个应用程序的安装程序或应用程序的一部分，例如升级或补丁。

在本实施例的一个实例中，远程机 30 包括打包机构 530。打包机构 530 在远程机 30 上的隔离环境 532 中执行安装程序。执行安装程序造成向隔离环境 532 中安装与应用程序相关联的至少一个应用文件。远程机 30 可以包括文件系统过滤器驱动程序 534，其通过拦截安装程序在本地机 10 上安装应用文件的请求并将该请求重定向到隔离环境

532 来确保将应用文件安装到隔离环境 532 中。打包机构 530 可以使用文件系统过滤器驱动程序 534 来维持安装到隔离环境 532 中的每个应用文件的记录。

安装程序可以向隔离环境 532 中安装多个应用文件。打包机构 530 生成包括多个应用文件中的应用文件枚举的文件。该文件可以包括与多个应用文件相关的信息以及策略要求，与多个应用文件相关的信息例如为多个应用文件包括的应用程序的类型、应用程序的版本、与应用程序相关联的执行先决条件，策略要求例如为特定应用程序所需的执行方法。打包机构 530 在远程机 30' 上存储多个应用文件和该文件。

在一个实施例中，网络管理员标识包括现有应用程序更新版本的应用程序或包括应用程序的多个应用文件中的应用文件。

可以将本发明提供为实现在一种或多种制品上或中的一种或多种计算机可读程序。该制品可以是软盘、硬盘、光盘、数字多用盘、闪速存储卡、PROM、RAM、ROM 或磁带。通常，可以用任何程序设计语言实现计算机可读程序。可以使用的一些语言实例包括 C、C++、C# 或 JAVA。可以将软件程序作为目标代码存储在一种或多种制品上或中。

尽管已经参考特定优选实施示例出和描述了本发明，本领域的技术人员应当理解，在不脱离如以下权利要求所界定的本发明的精神和范围的情况下可以在其中做出各种形式和细节的变化。

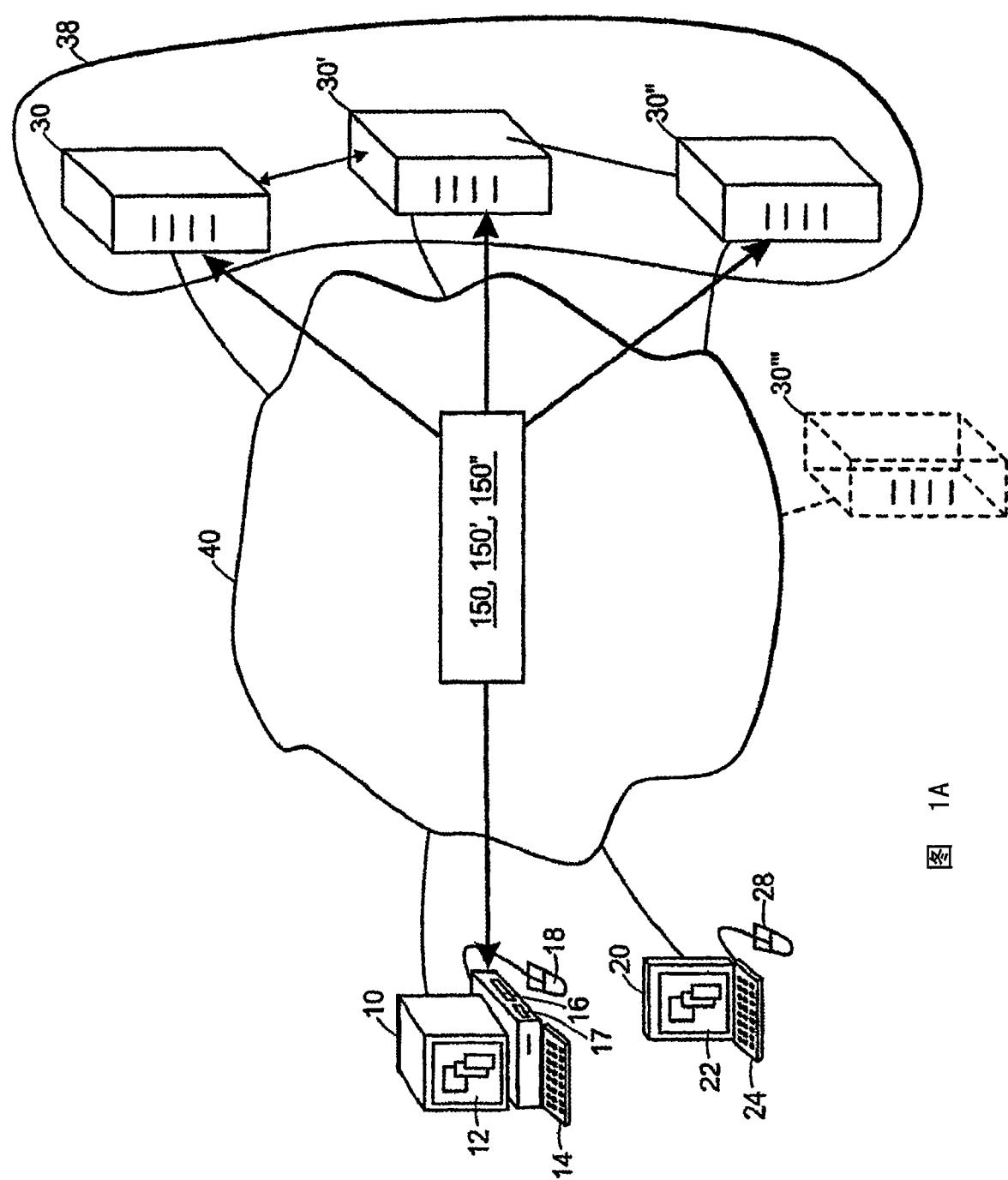


图 1A

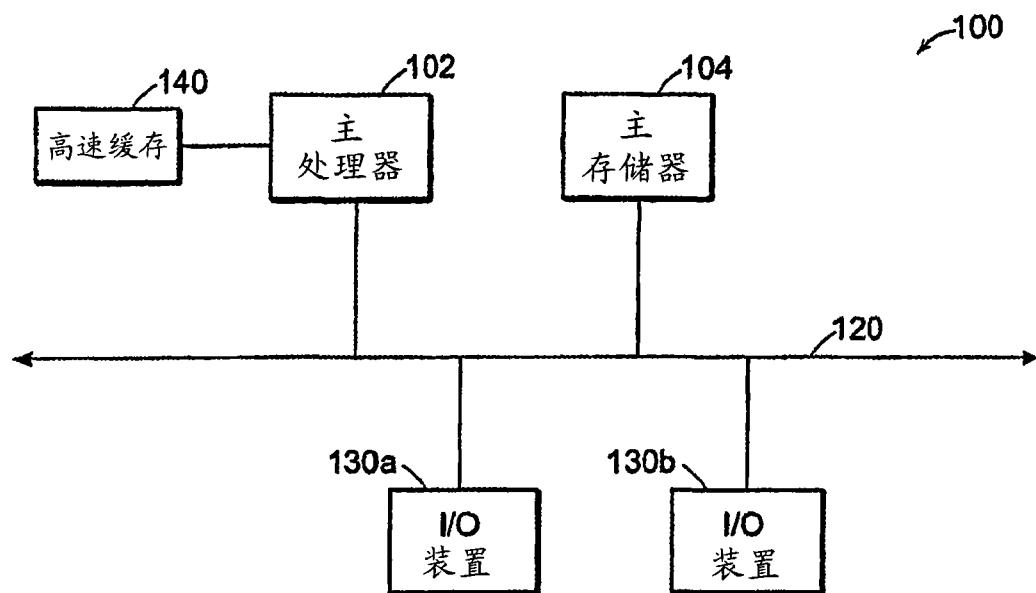


图 1B

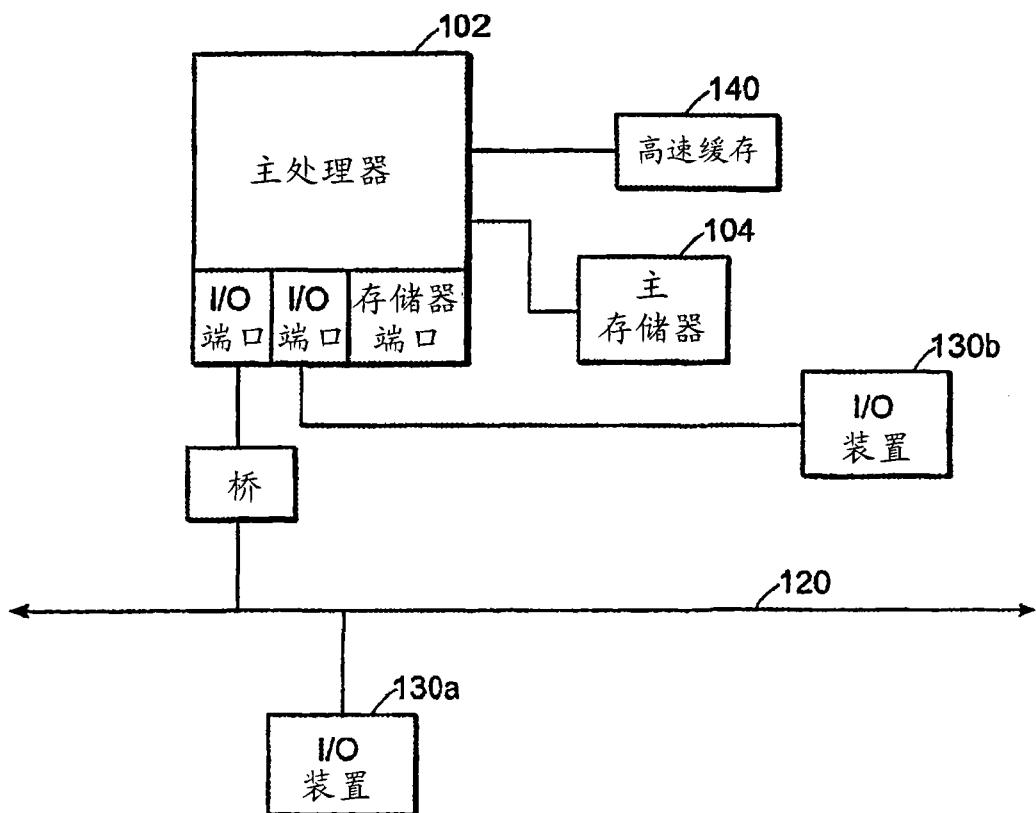


图 1C

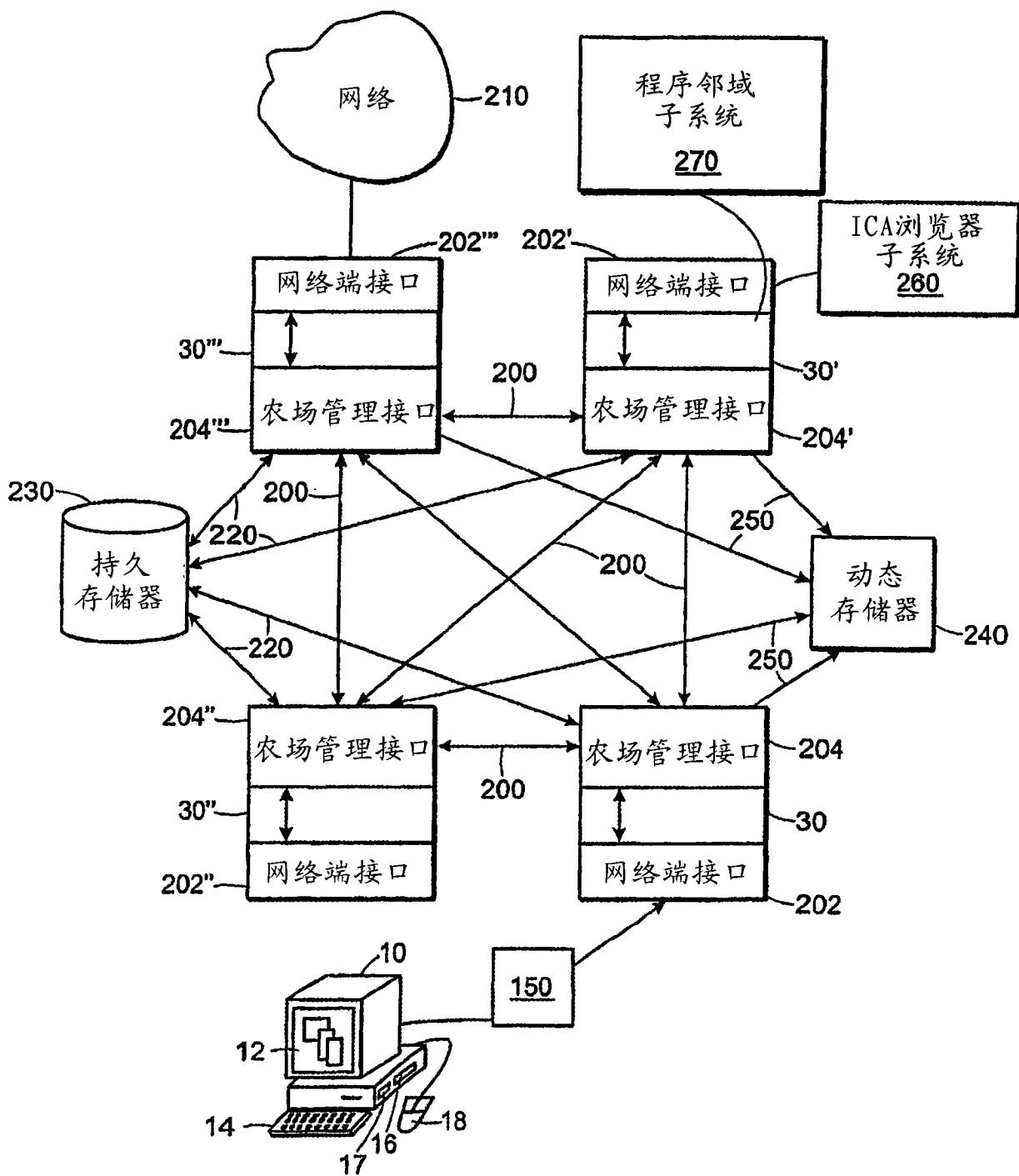


图 1D

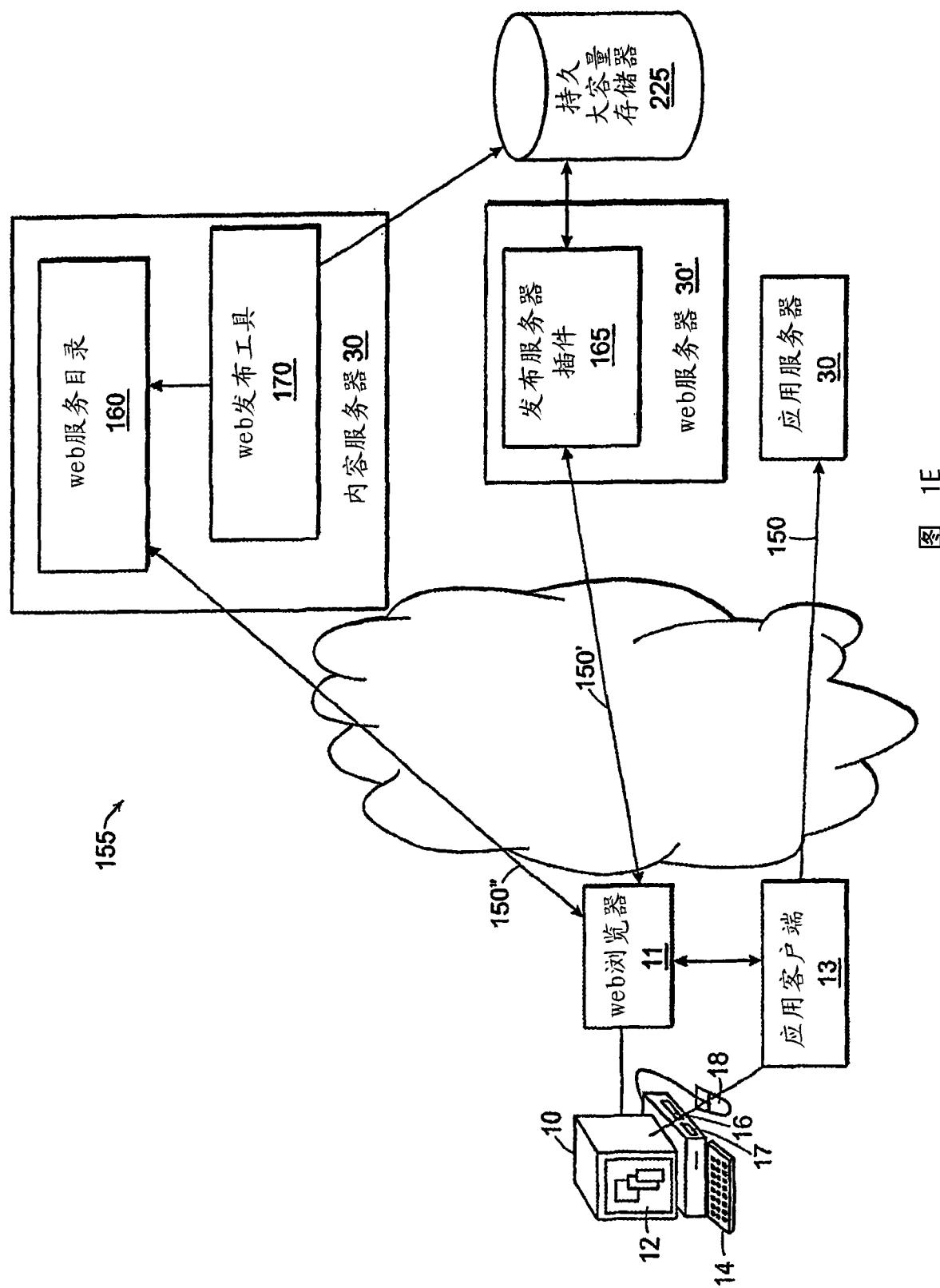


图 1E

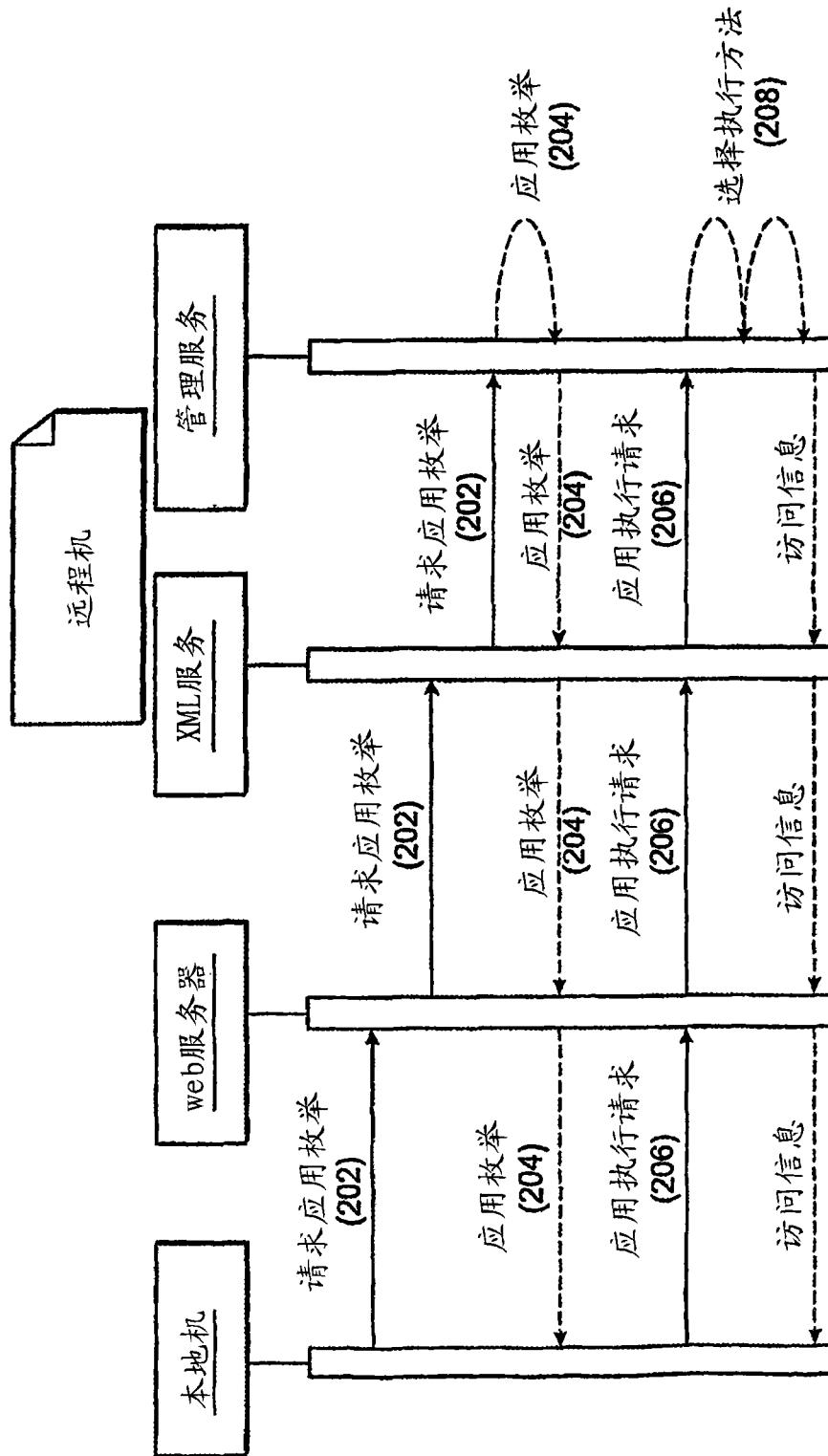


图 2

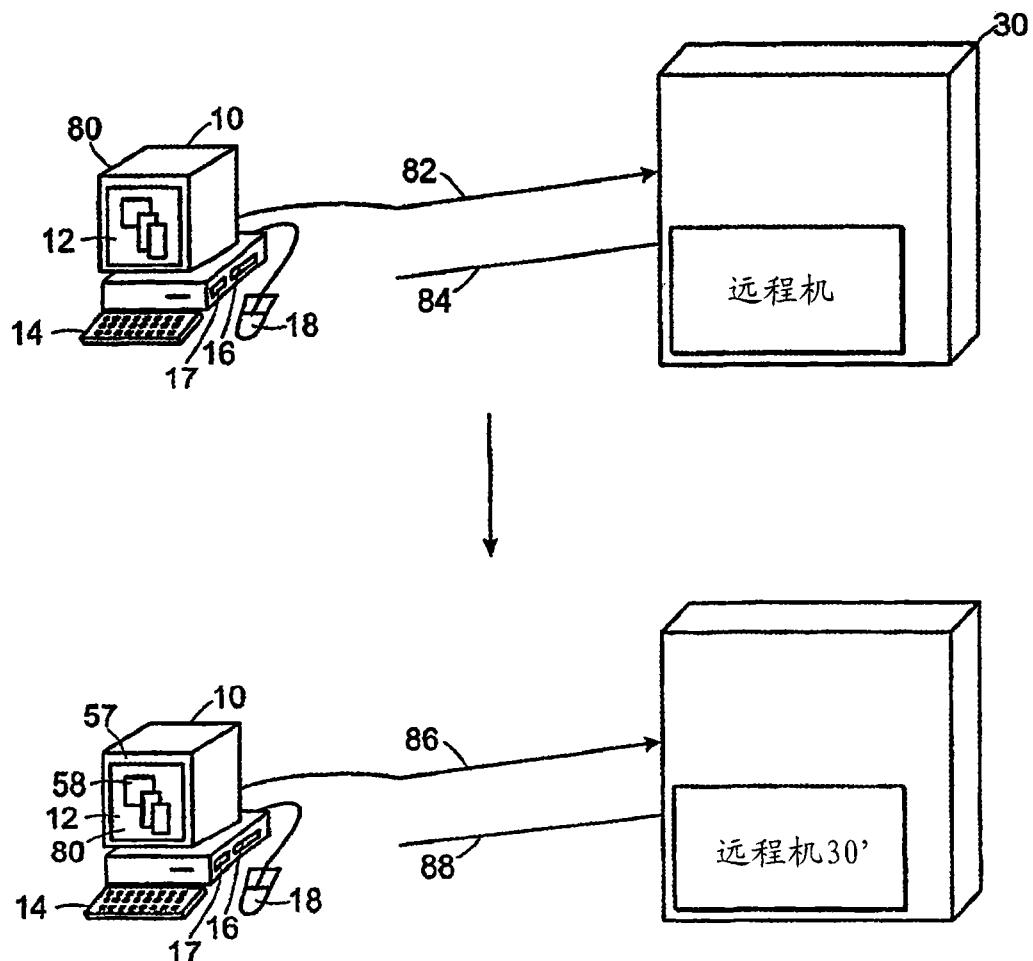


图 3A

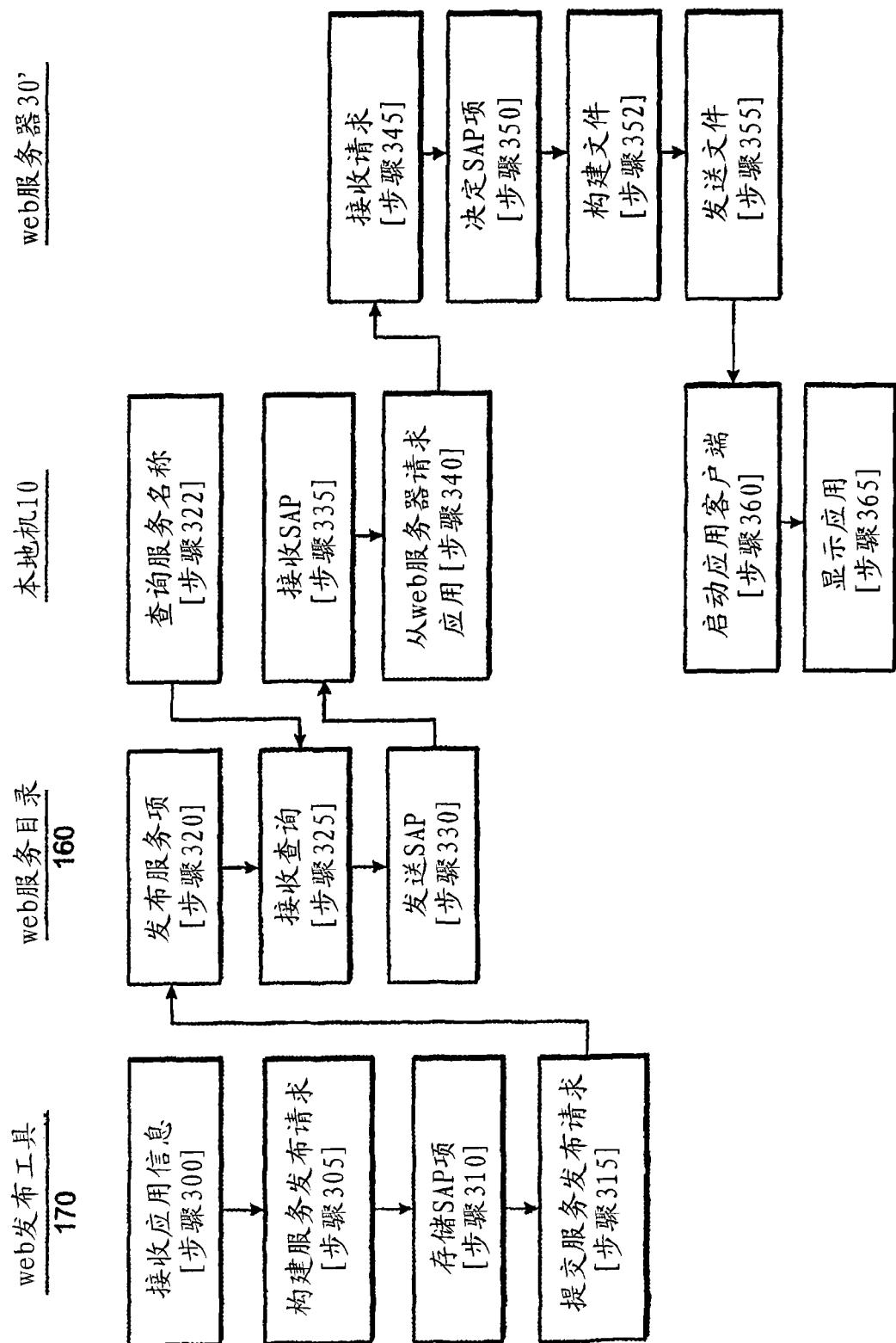


图 3B

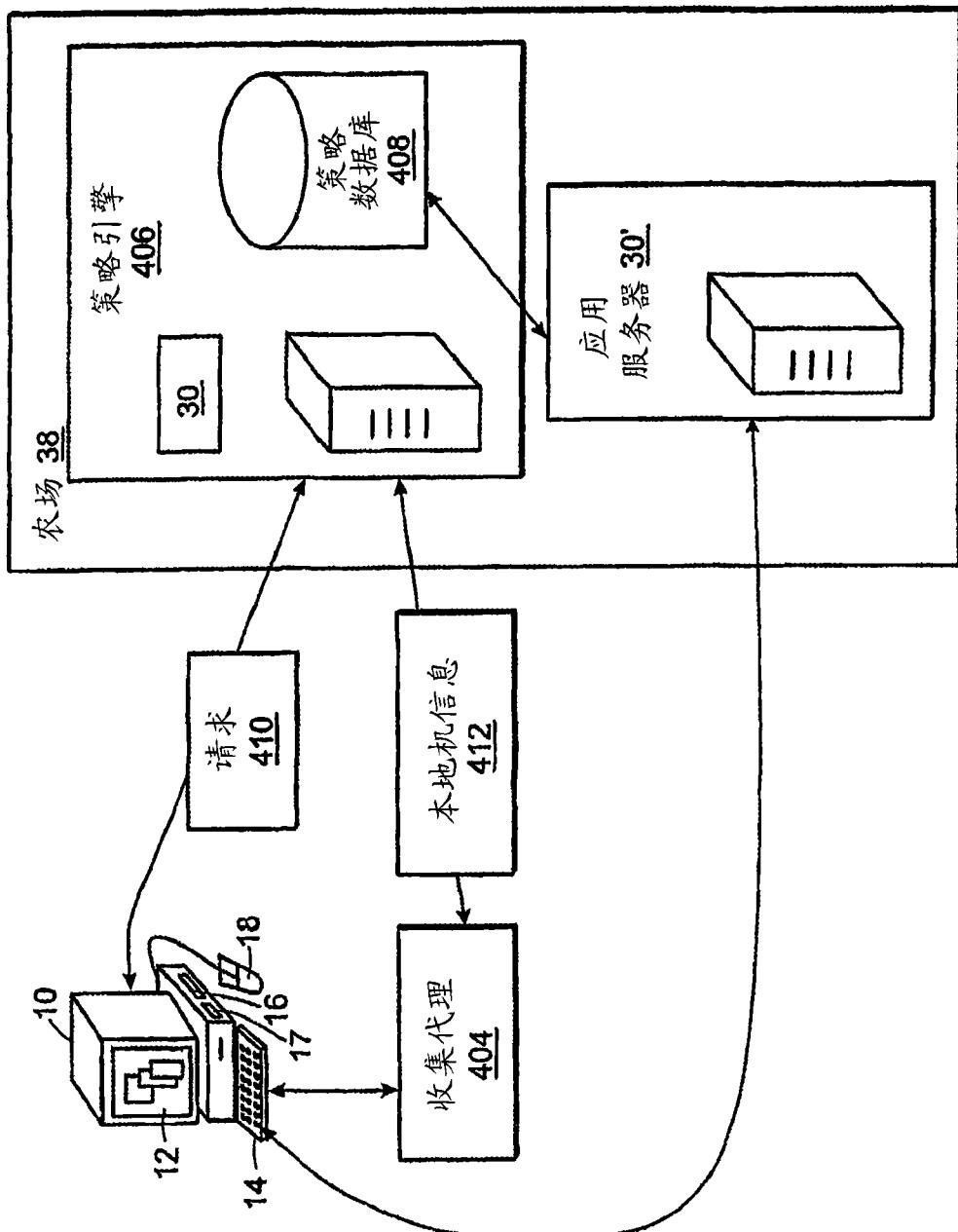


图 4A

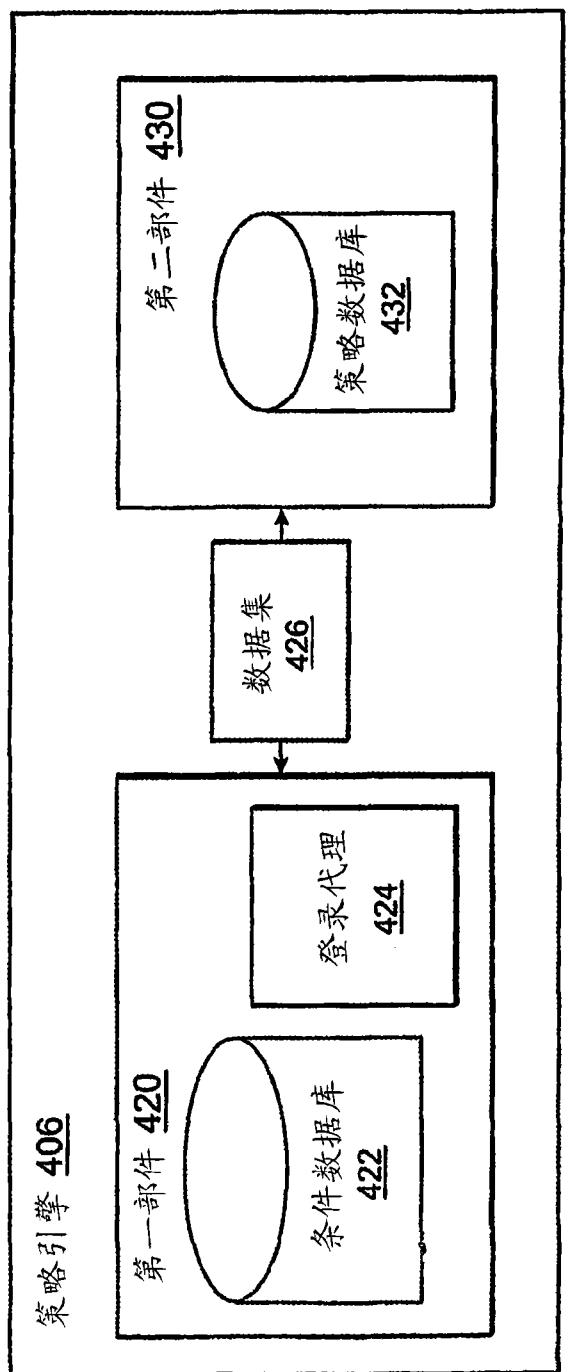


图 4B

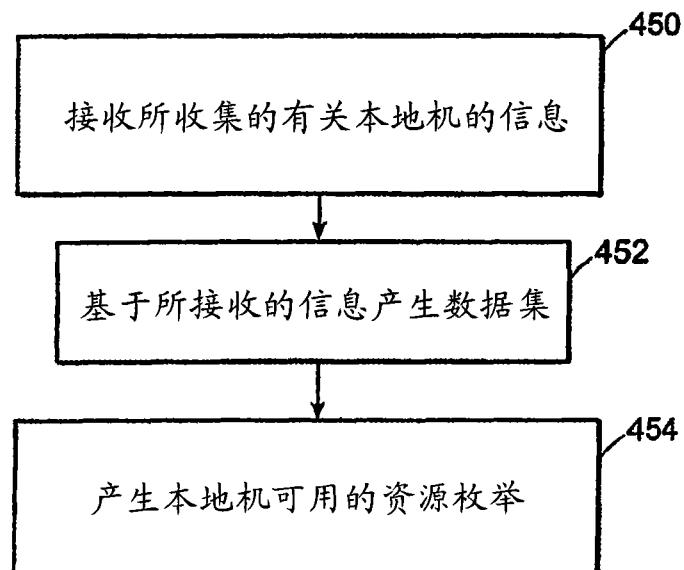


图 4C

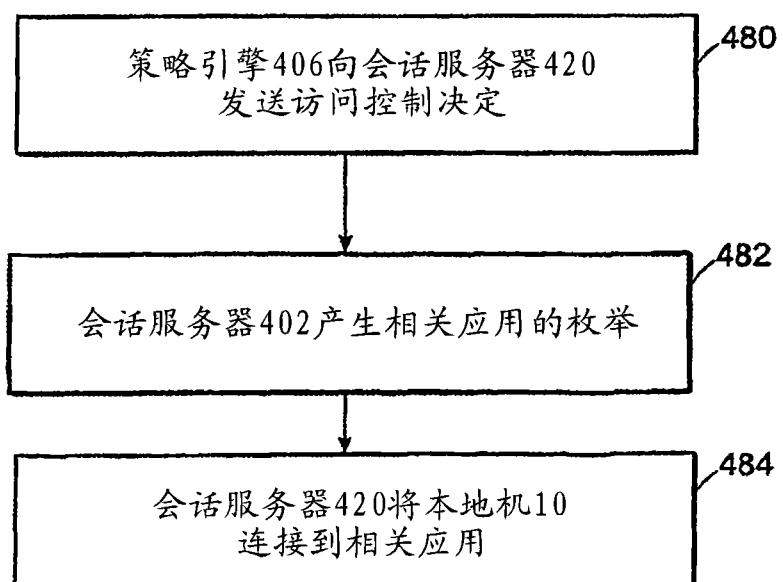


图 4E

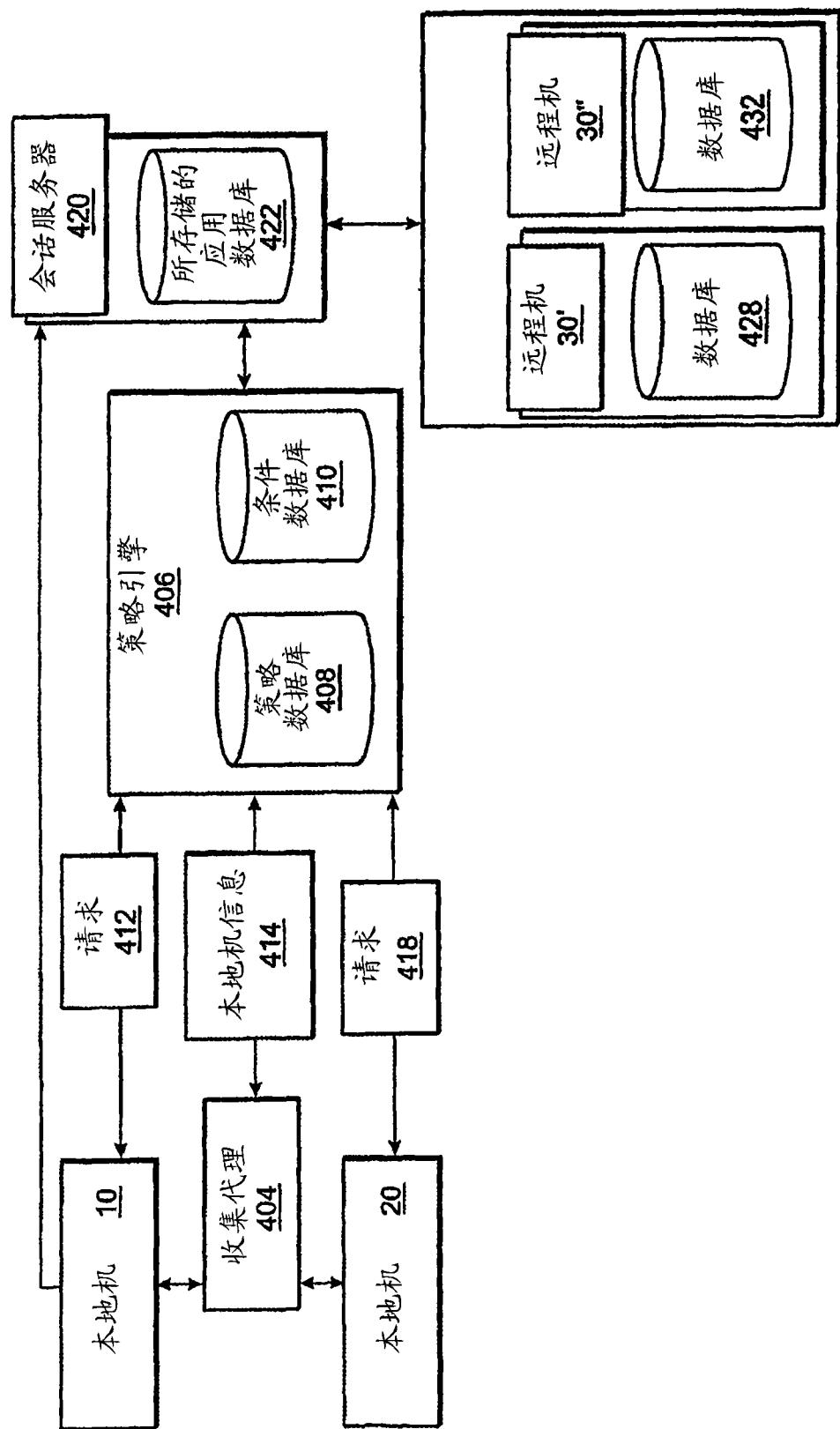


图 4D

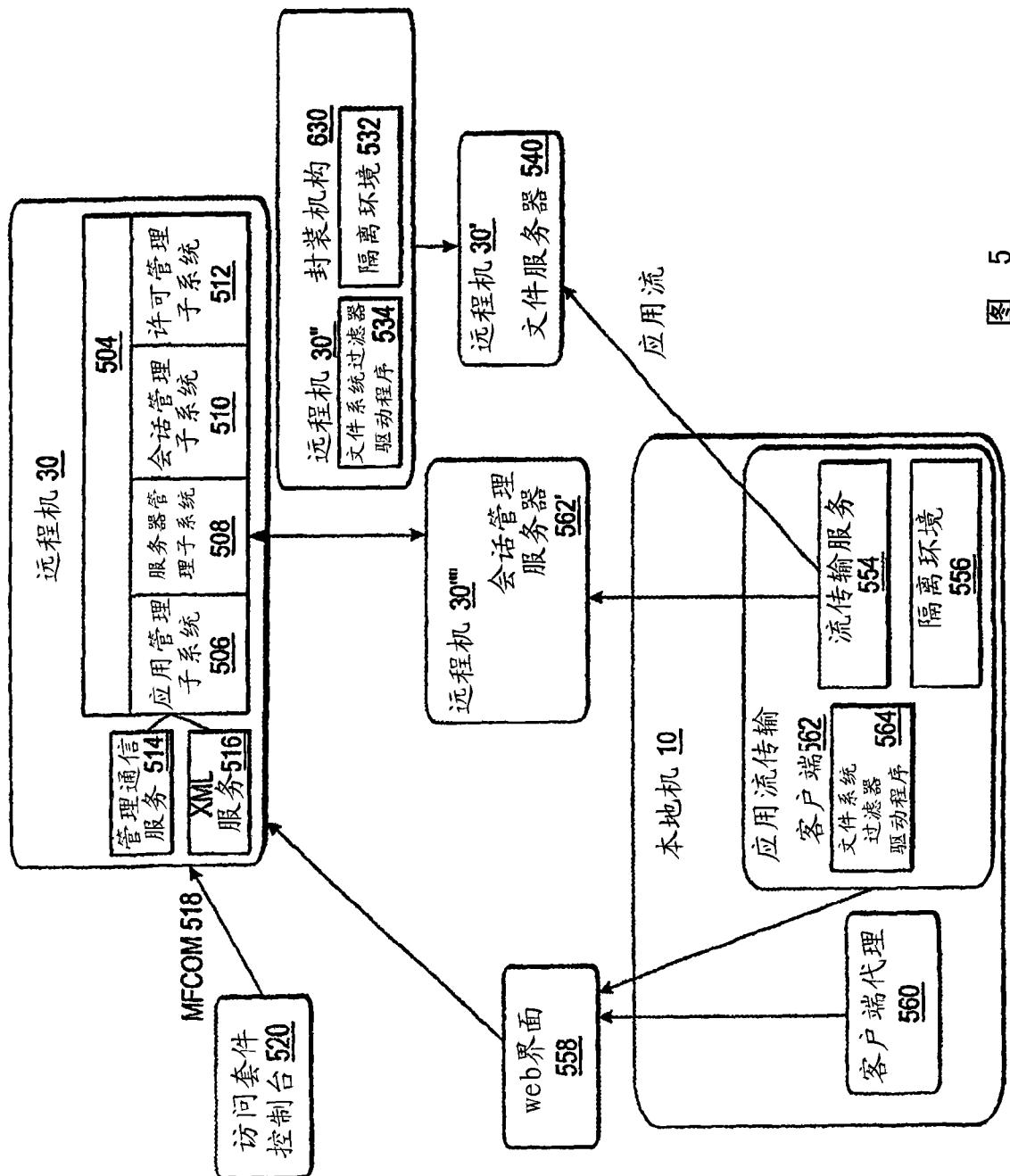


图 5

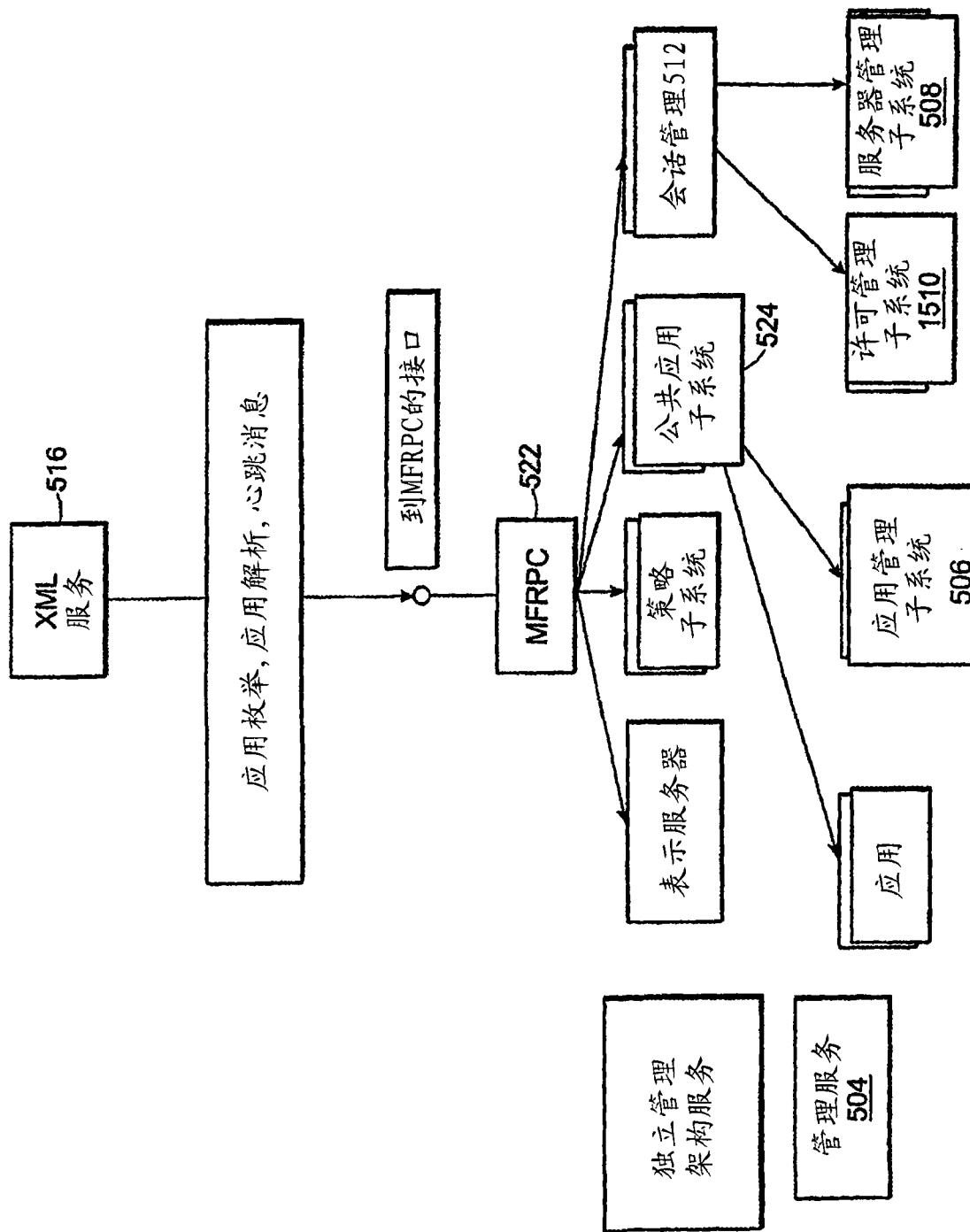


图 6

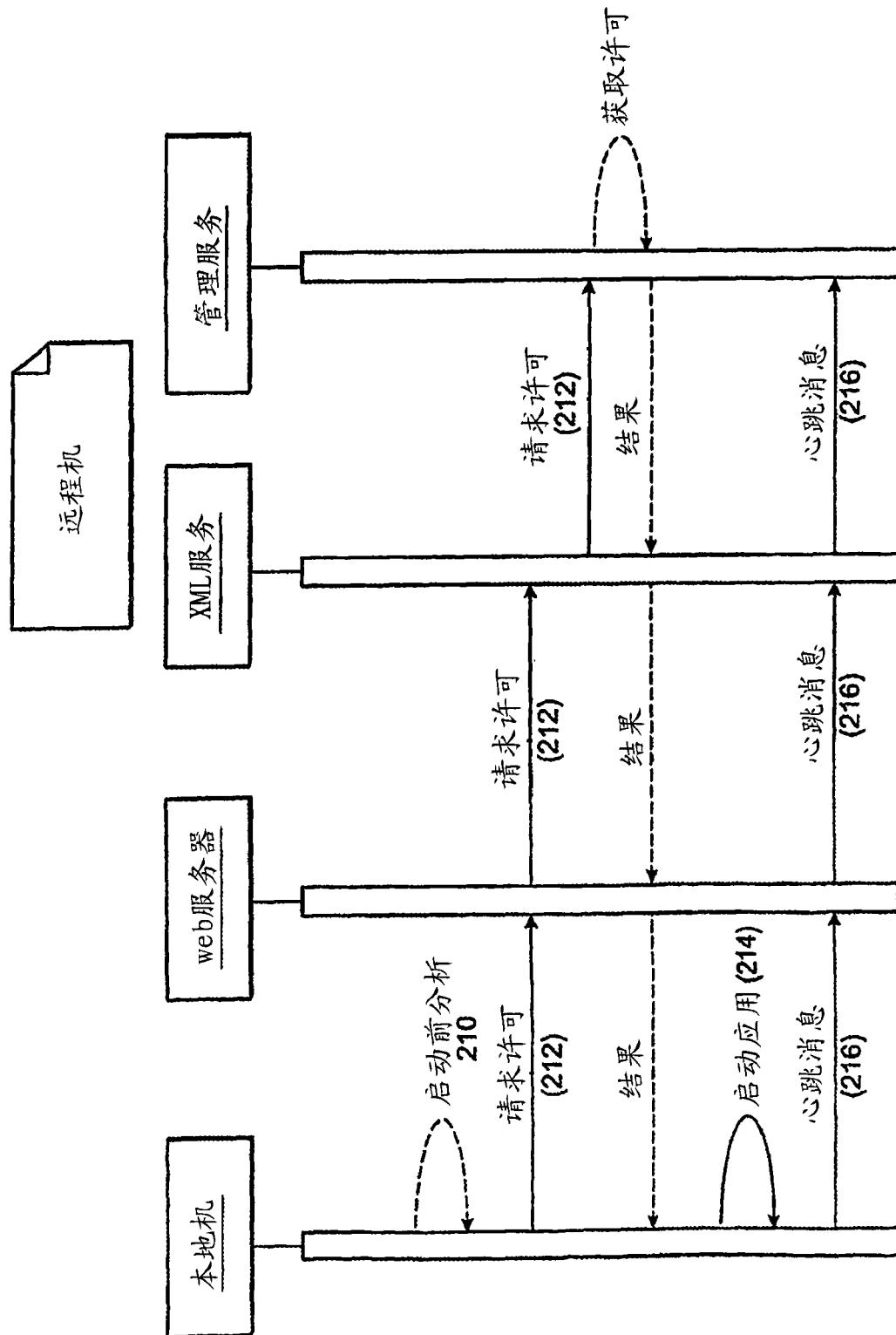


图 7

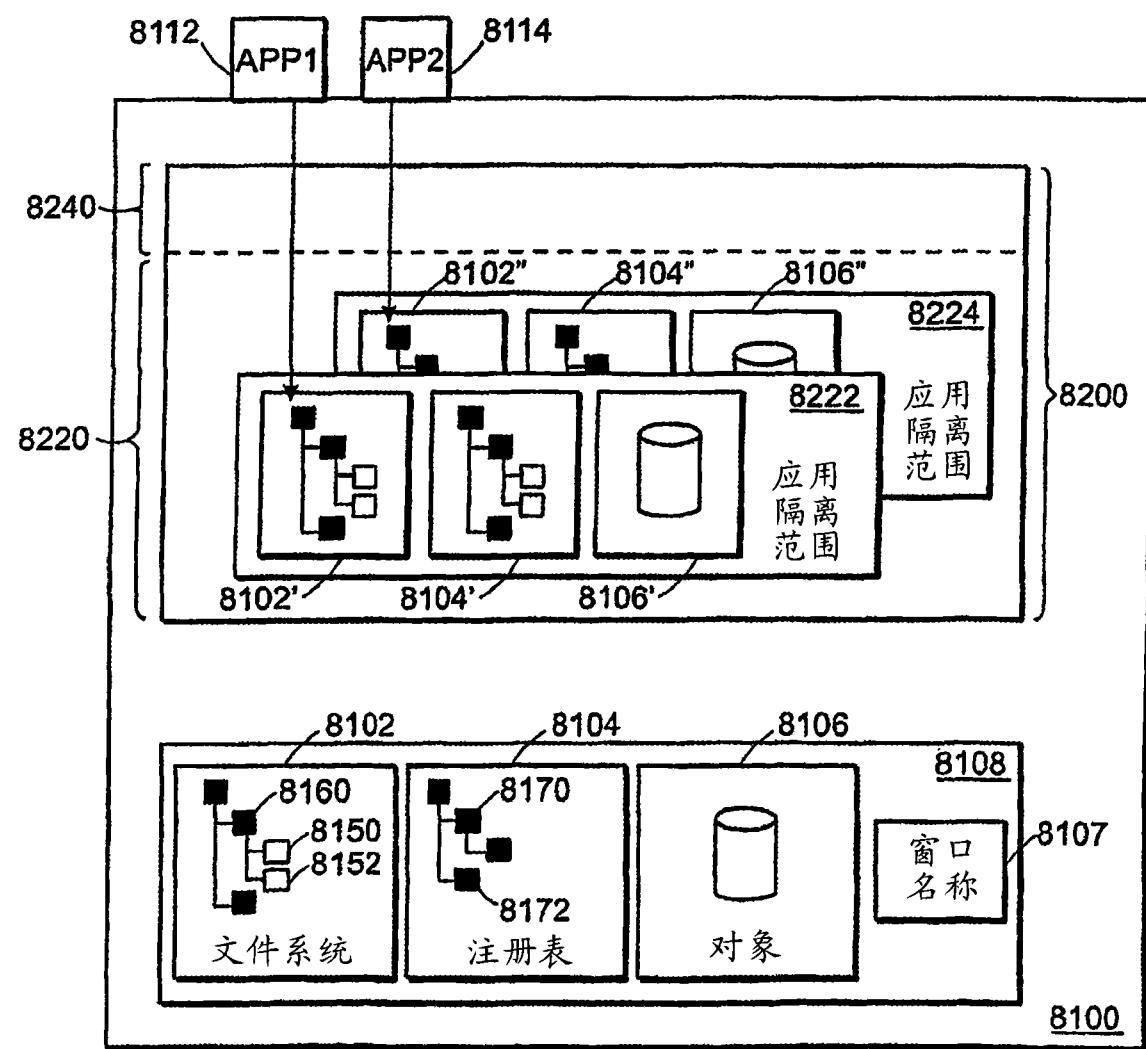


图 8A

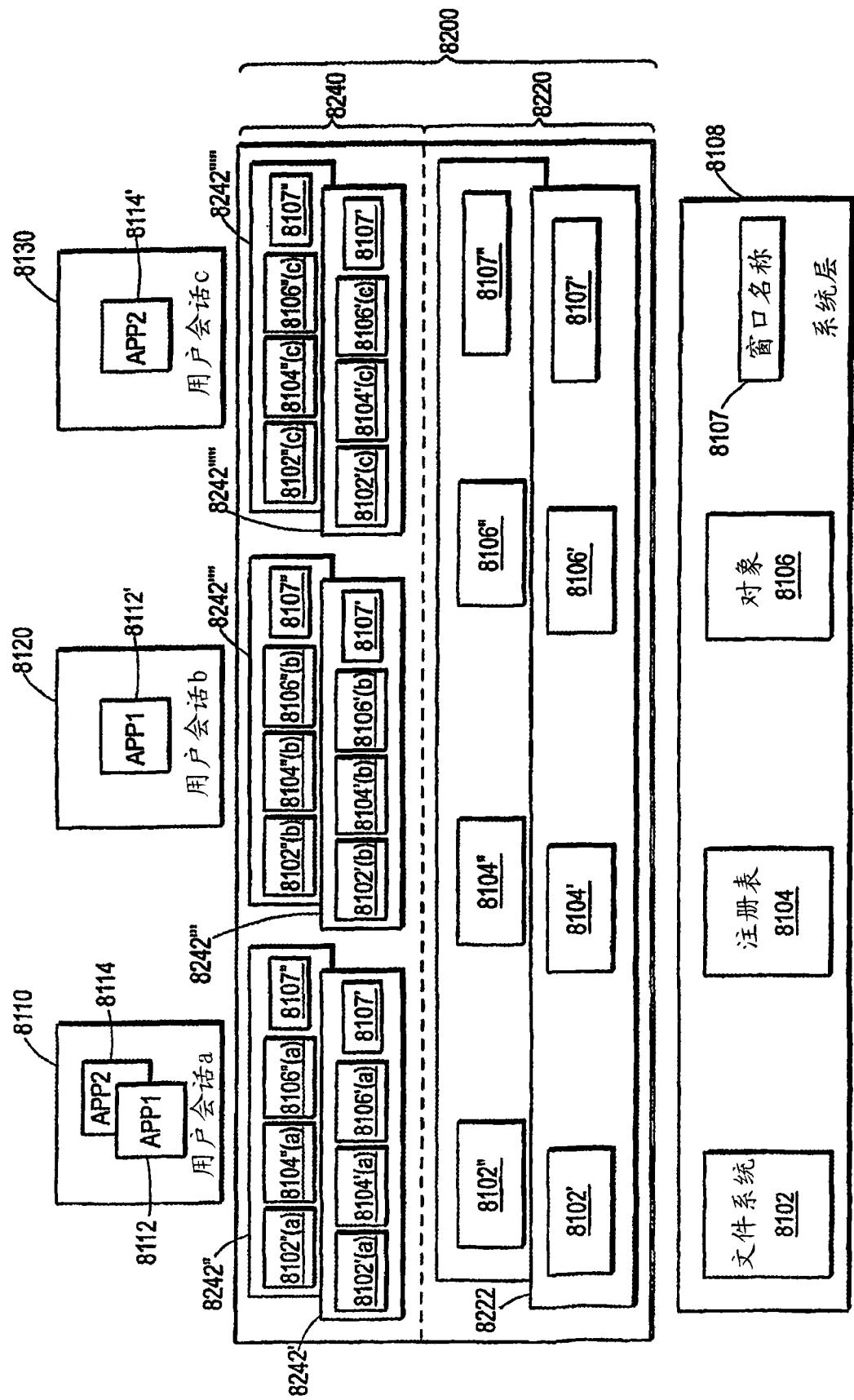


图 8B

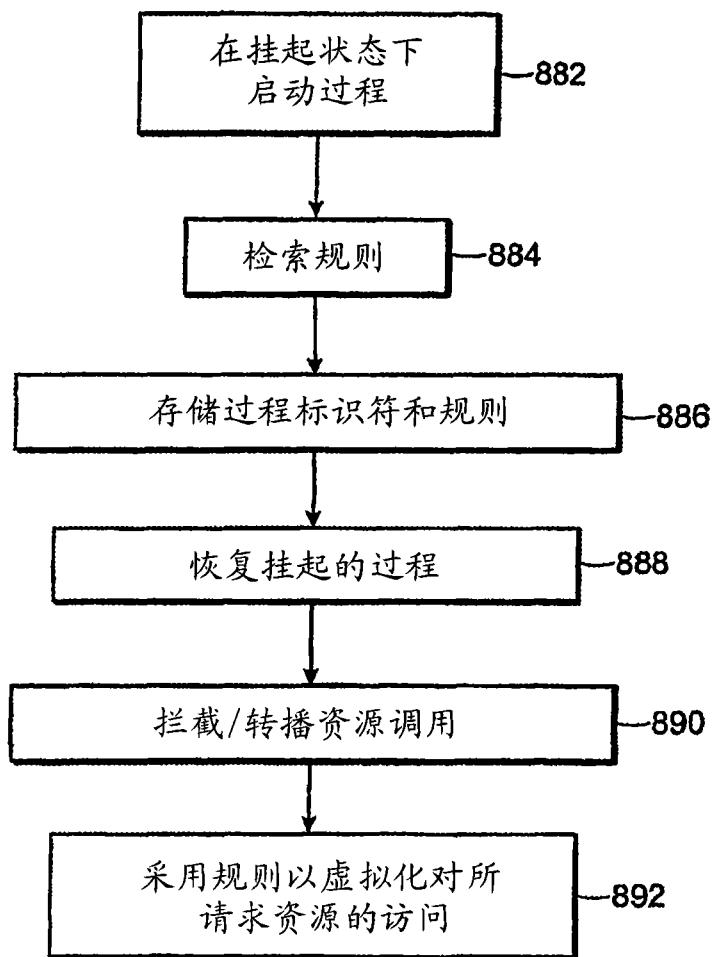


图 80

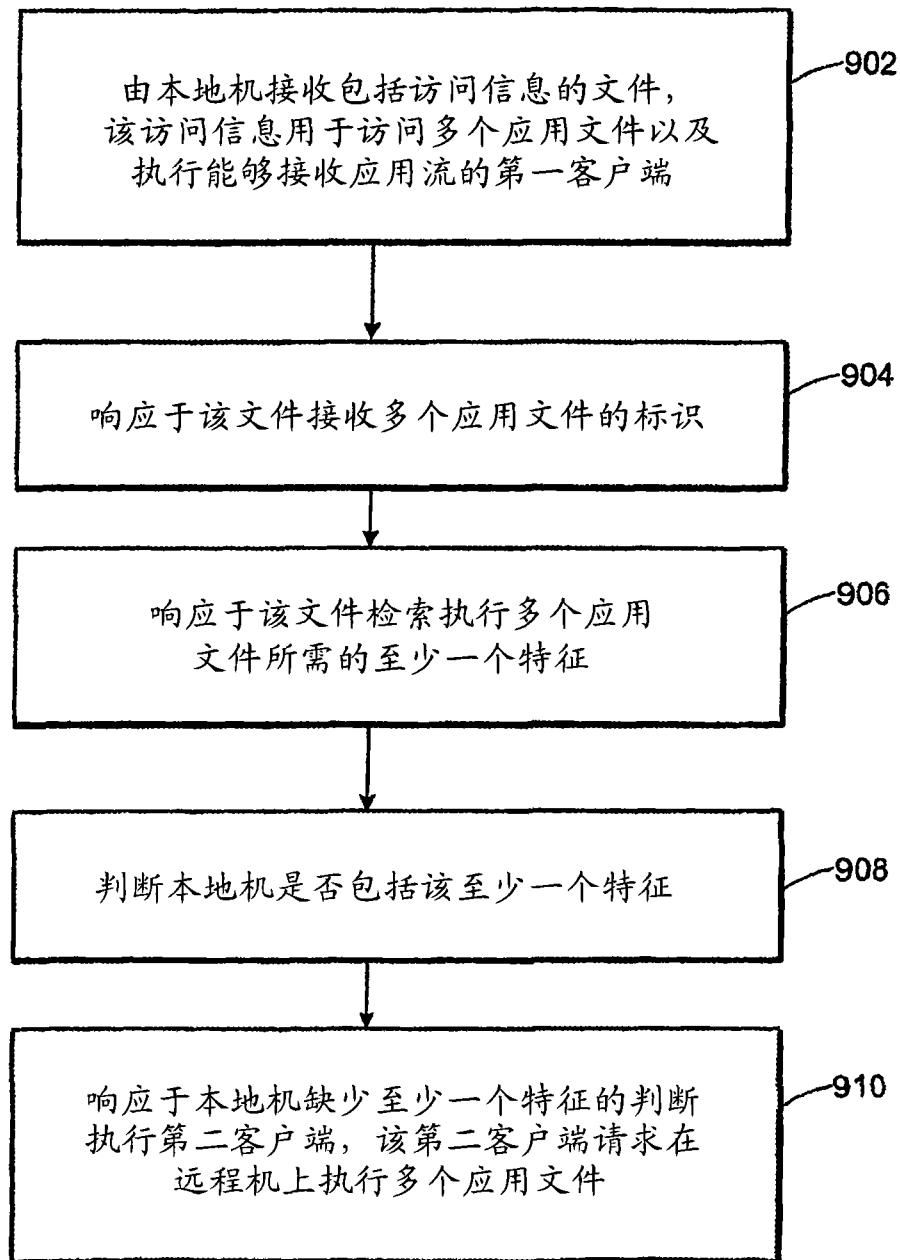


图 9

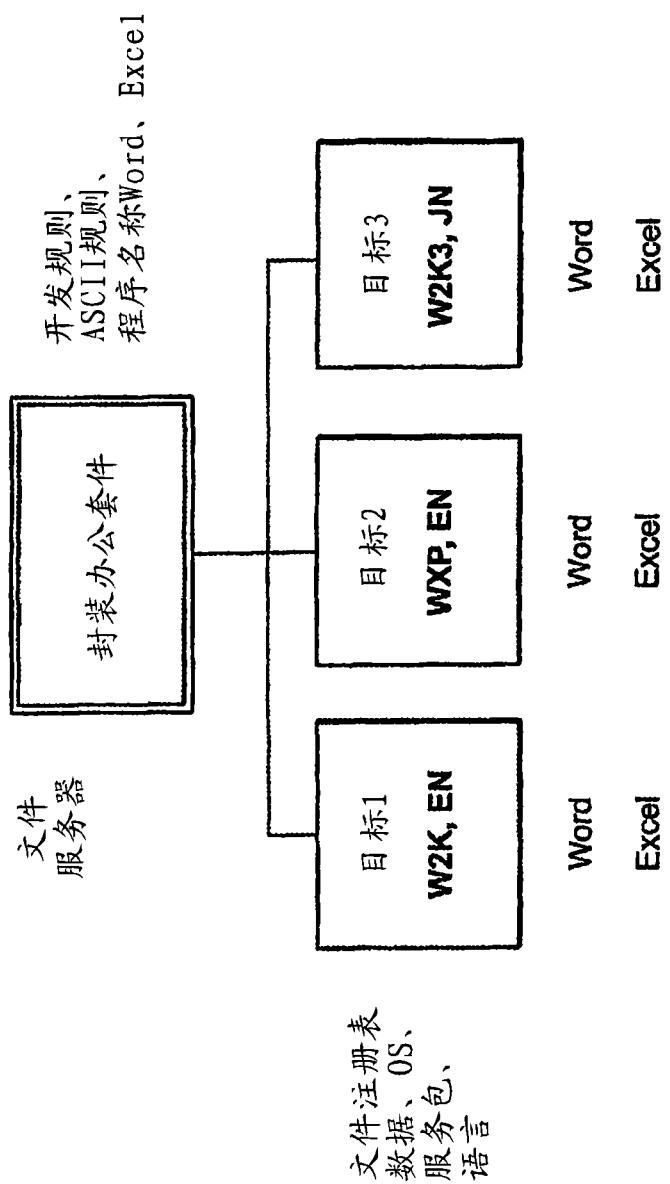


图 10

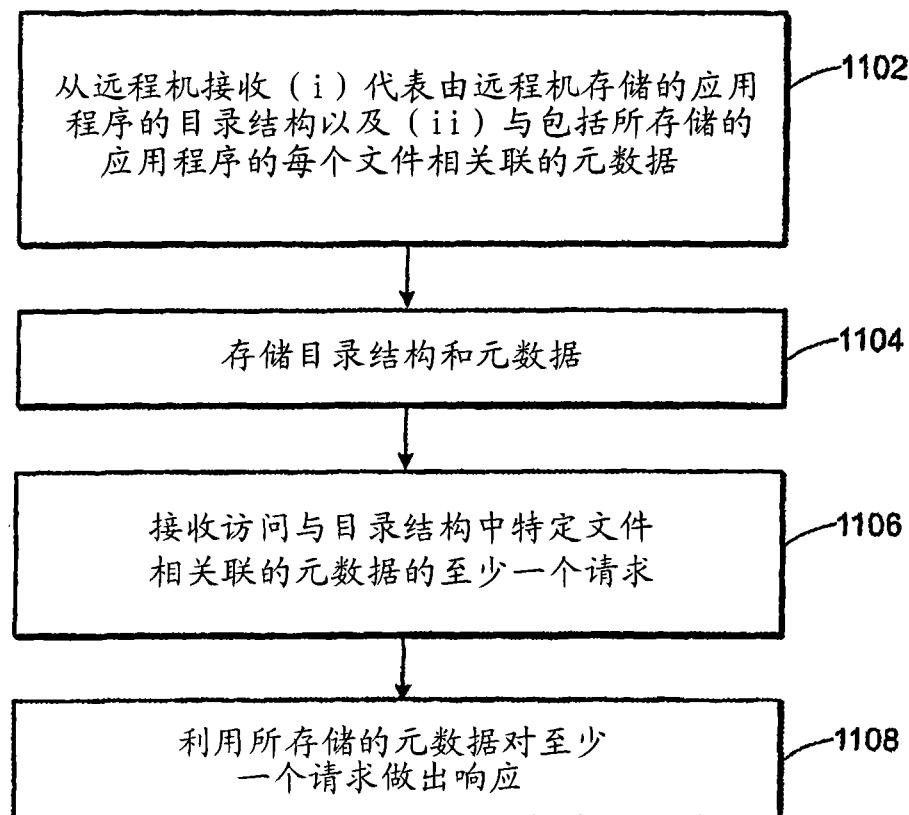


图 11

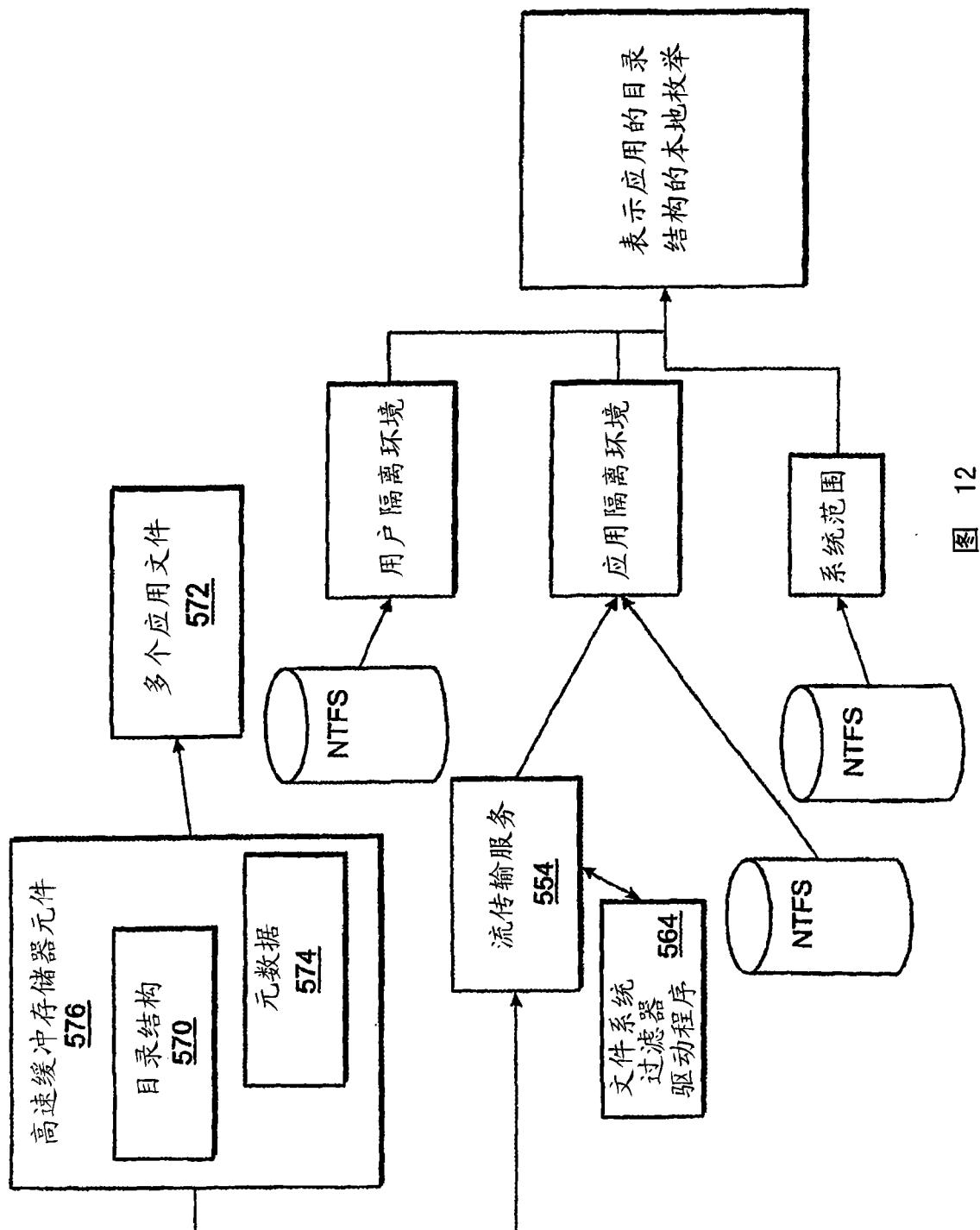


图 12

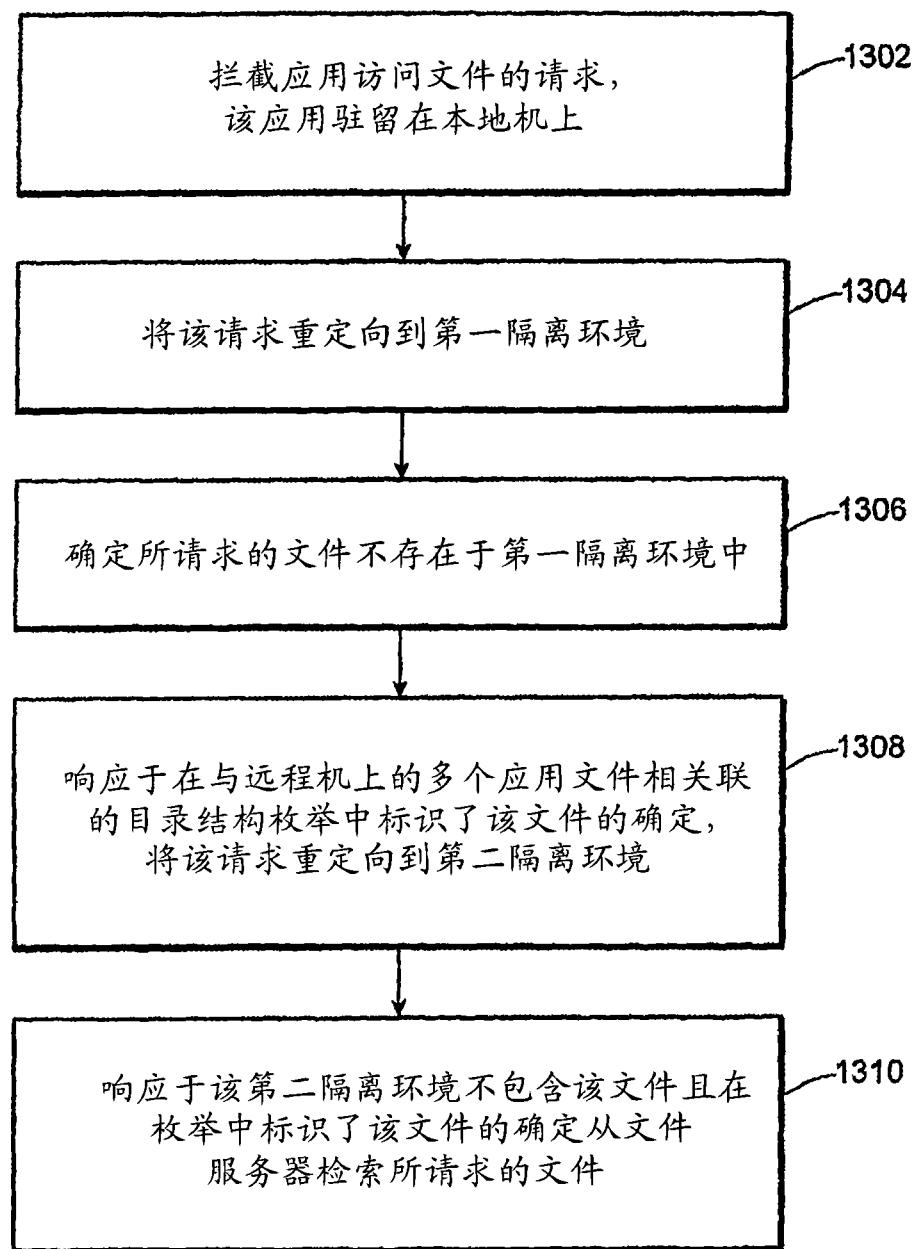


图 13

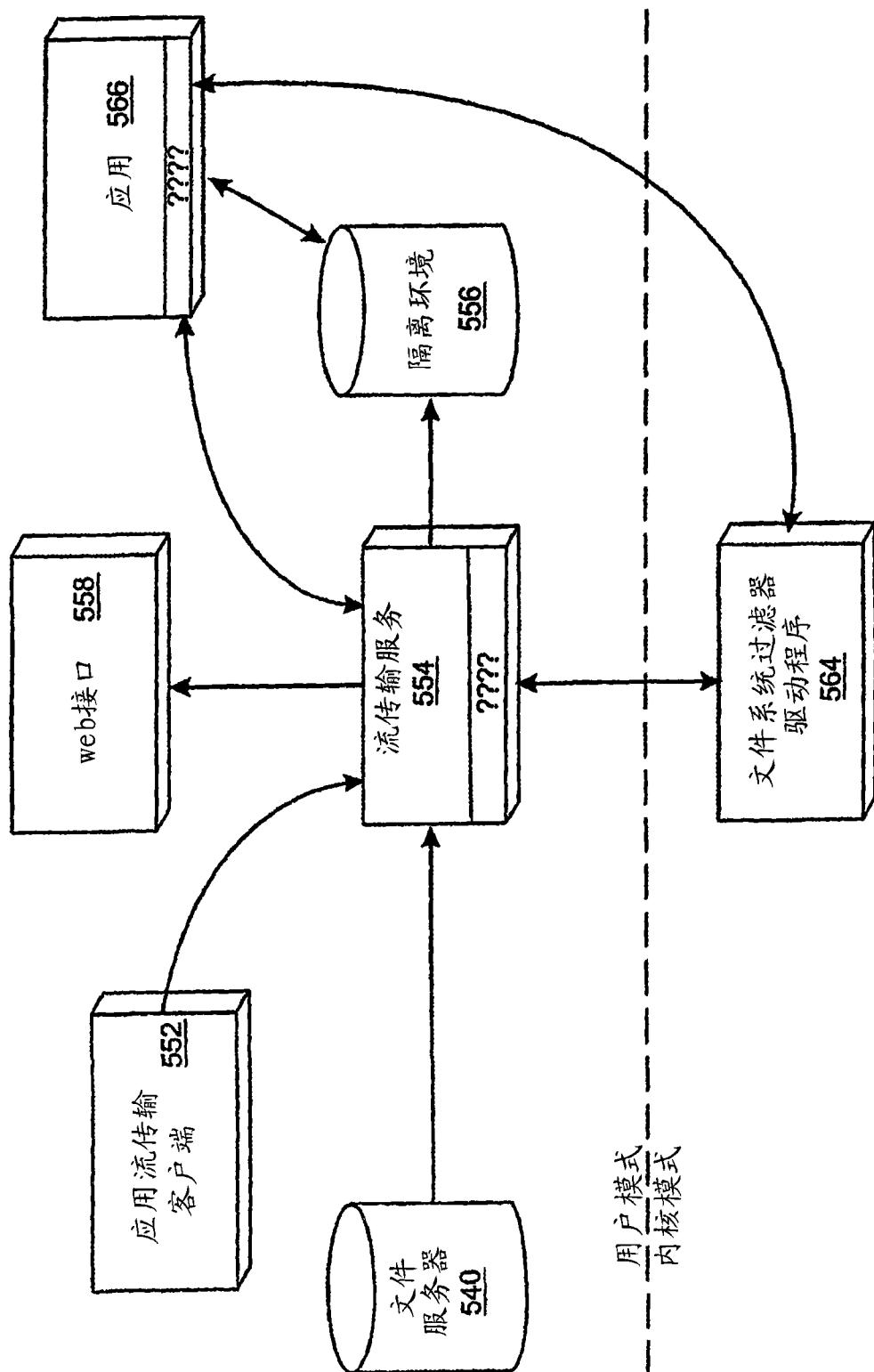


图 14

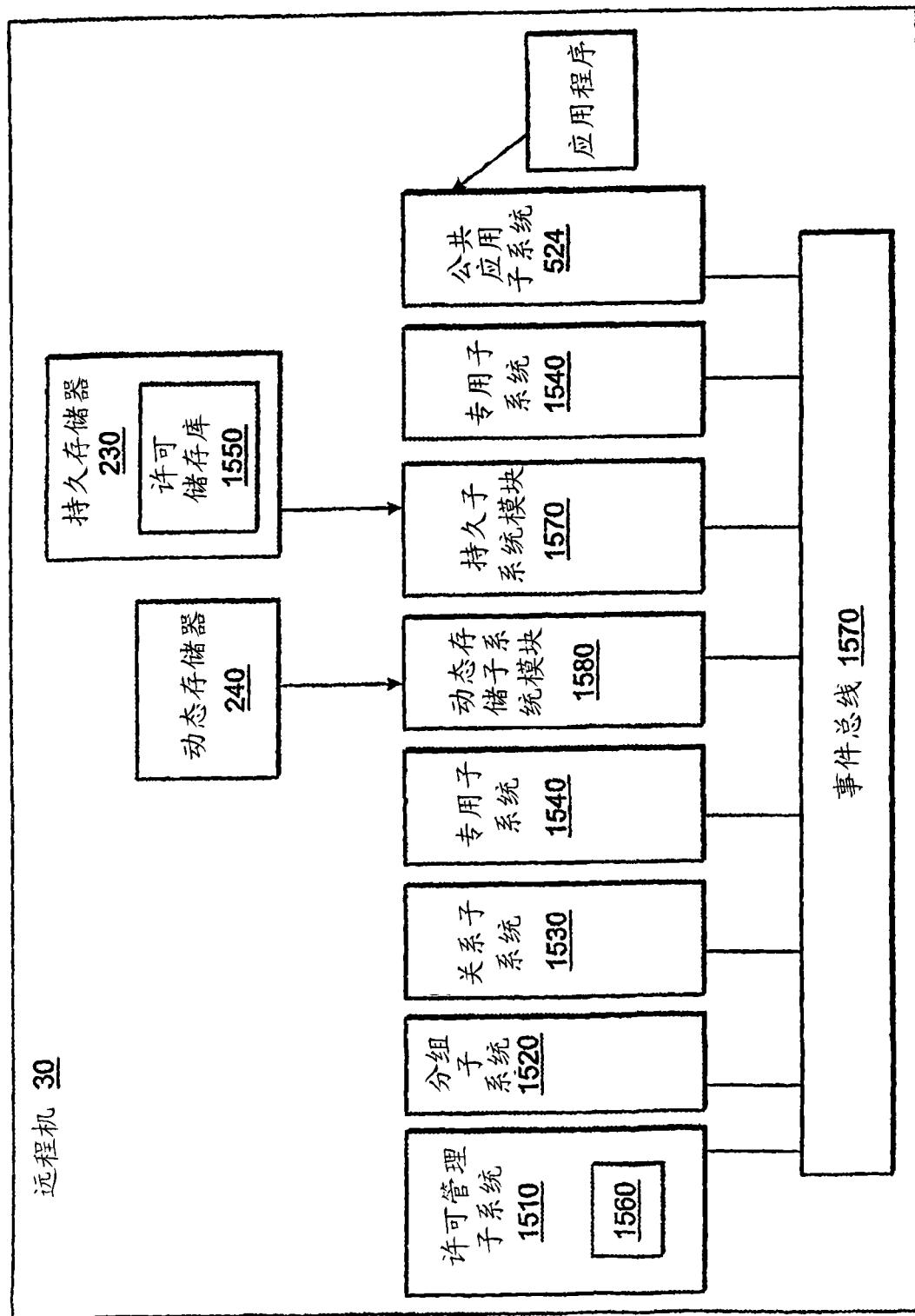
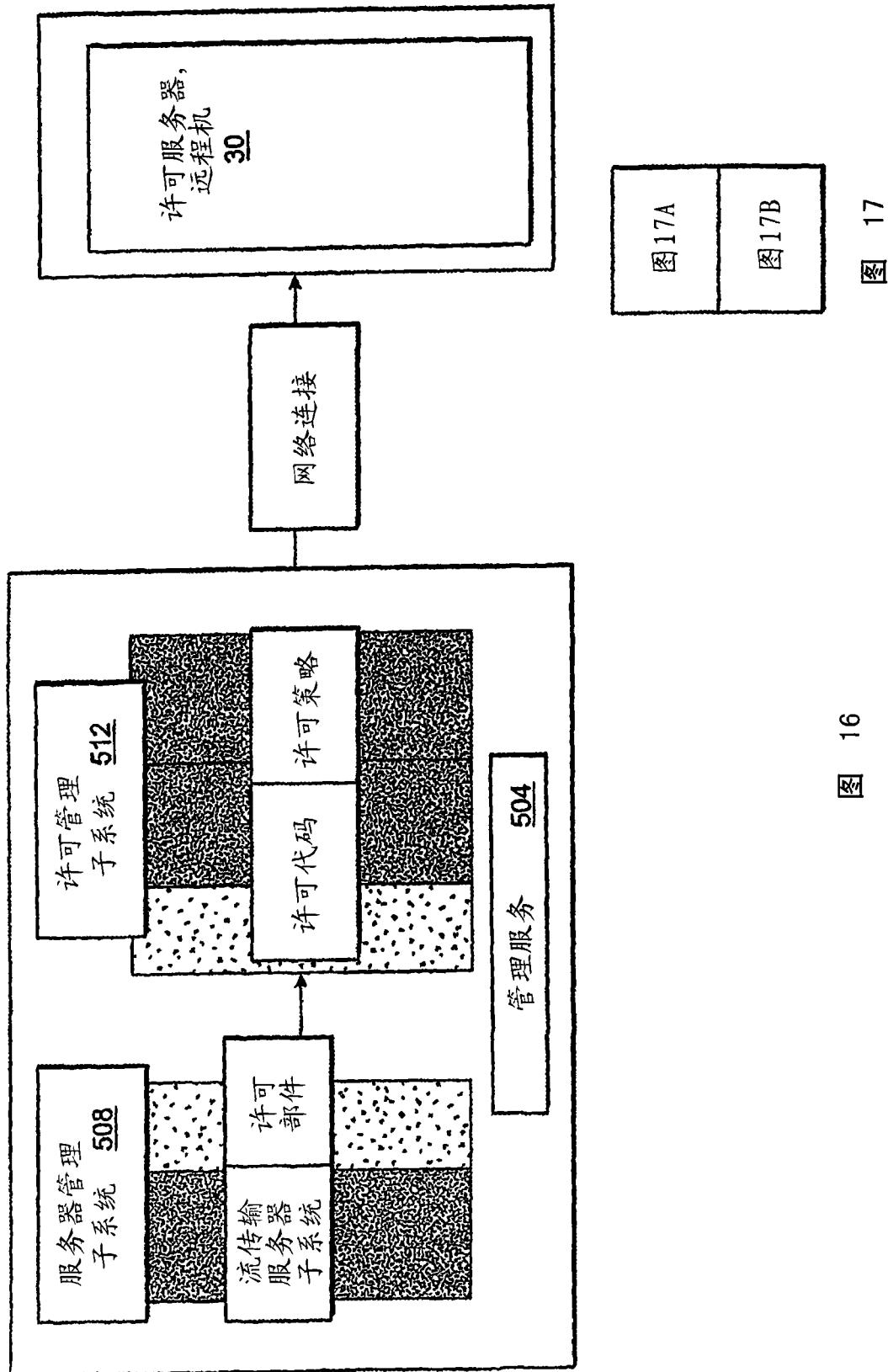


图 15



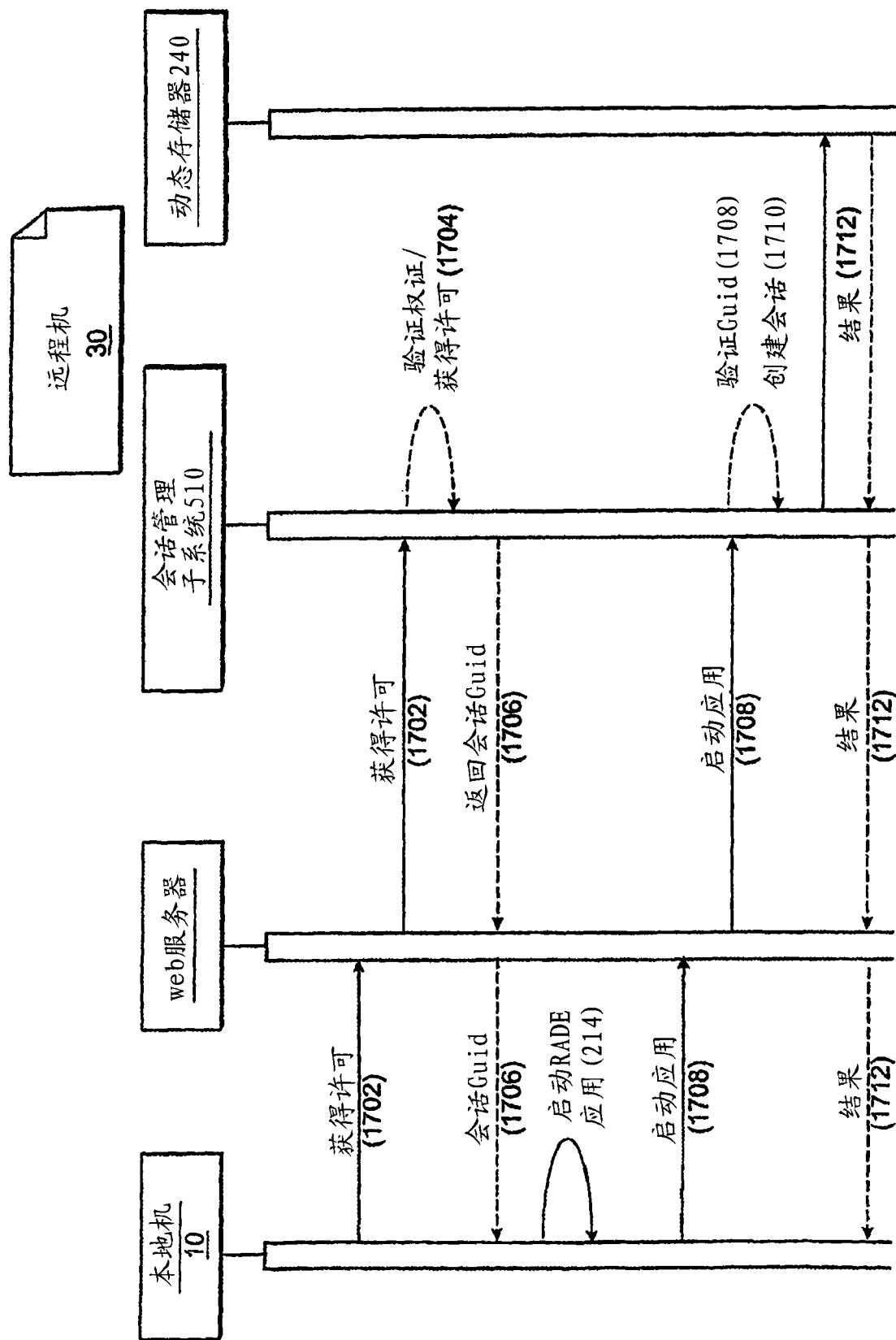


图 17A

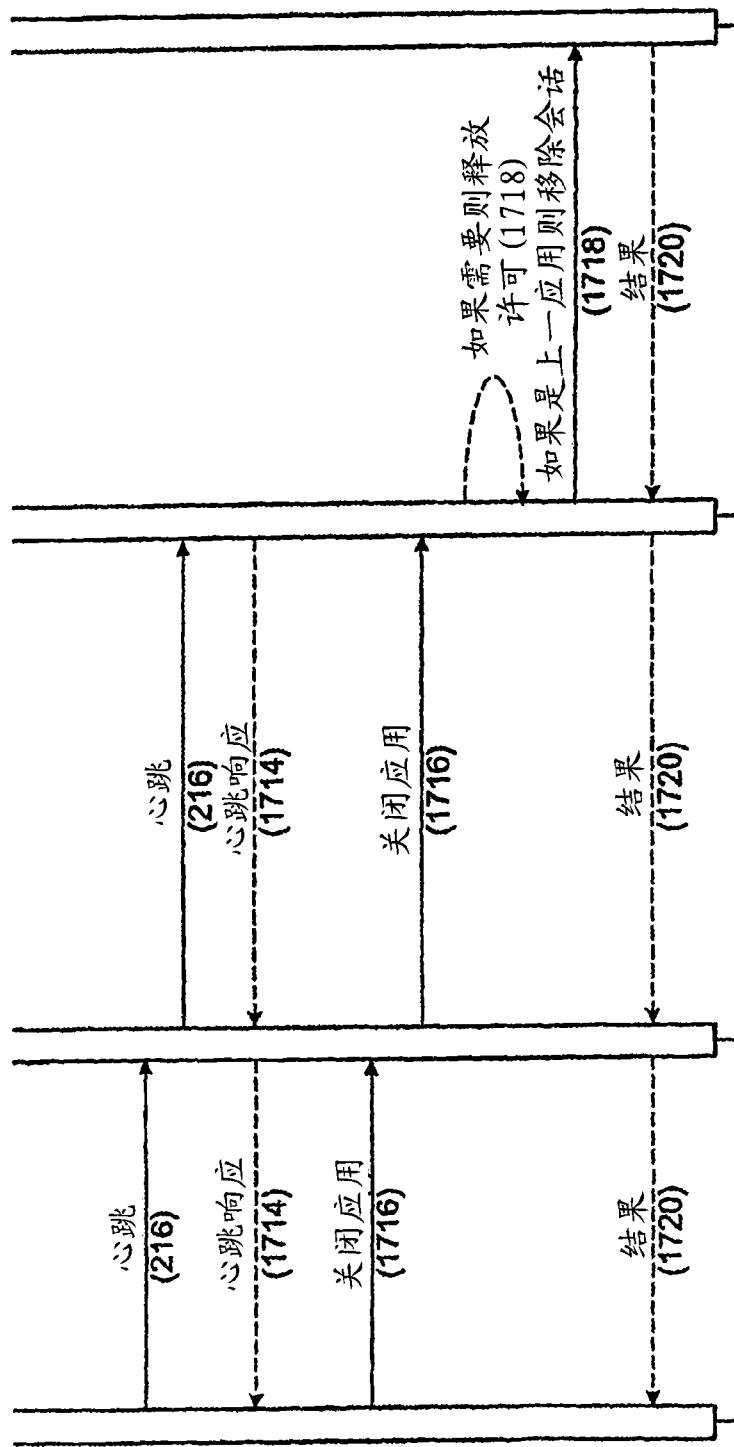


图 17B

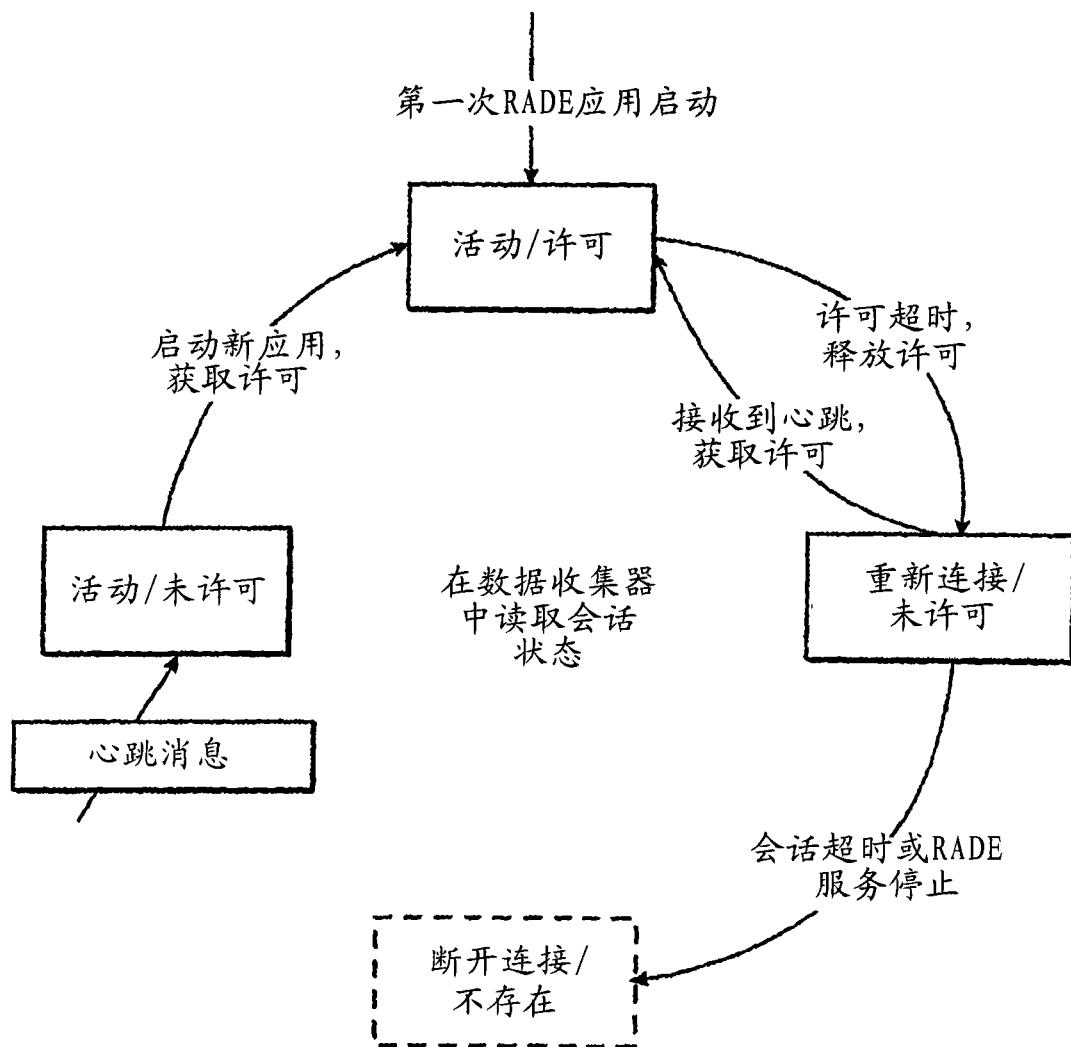


图 18

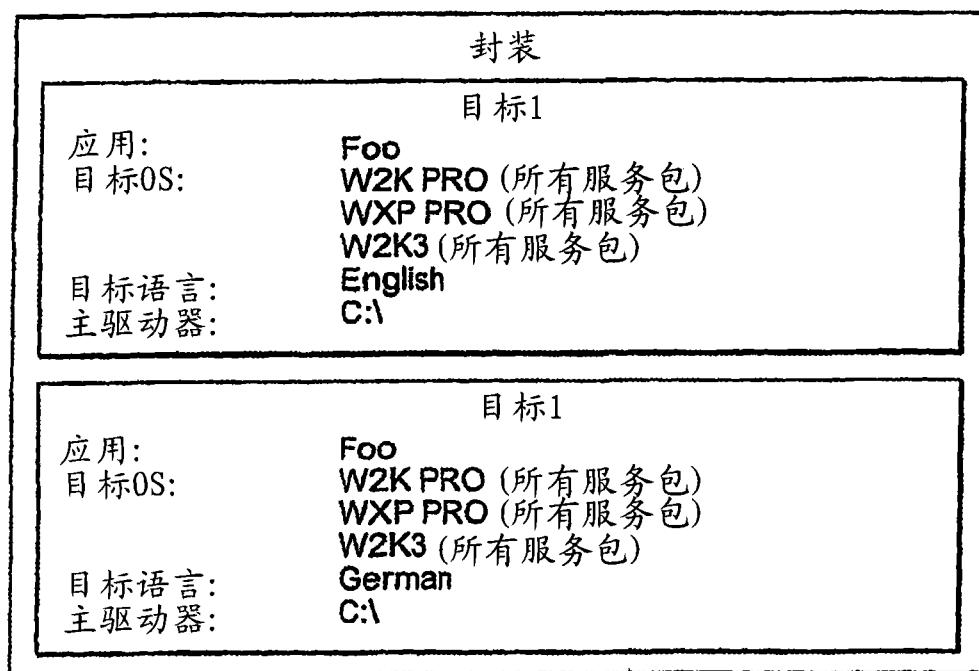


图 19

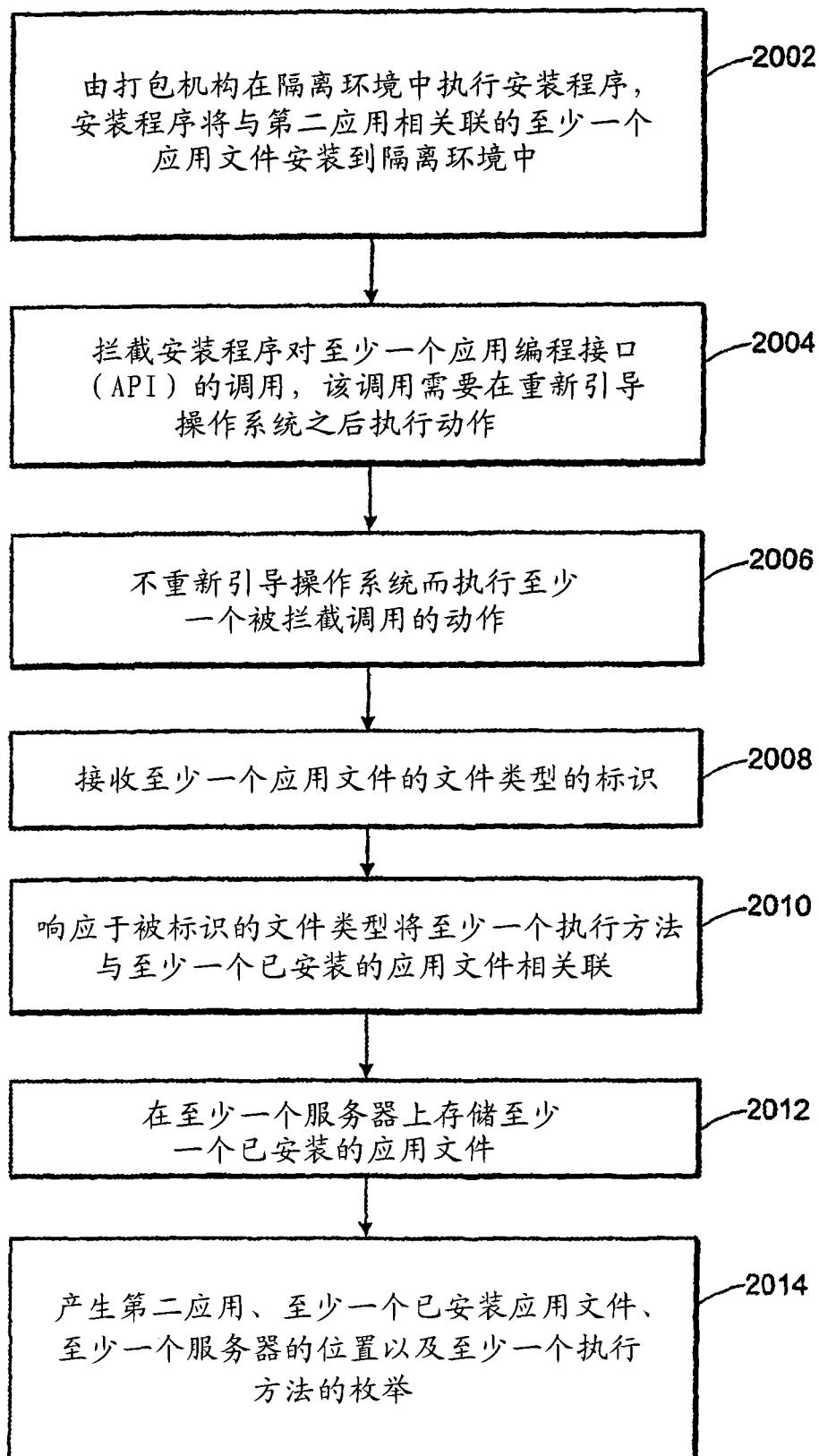


图 20

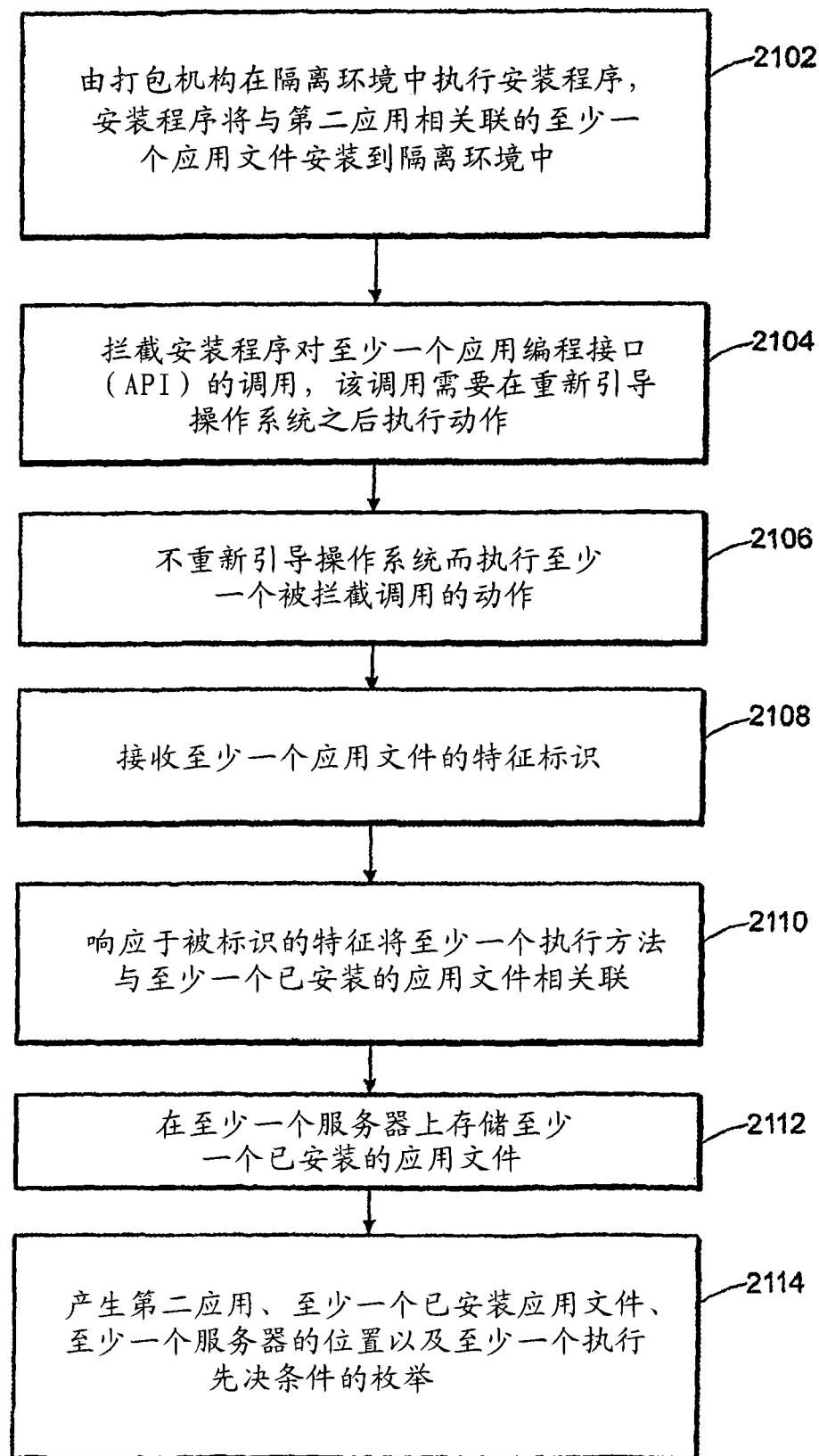


图 21

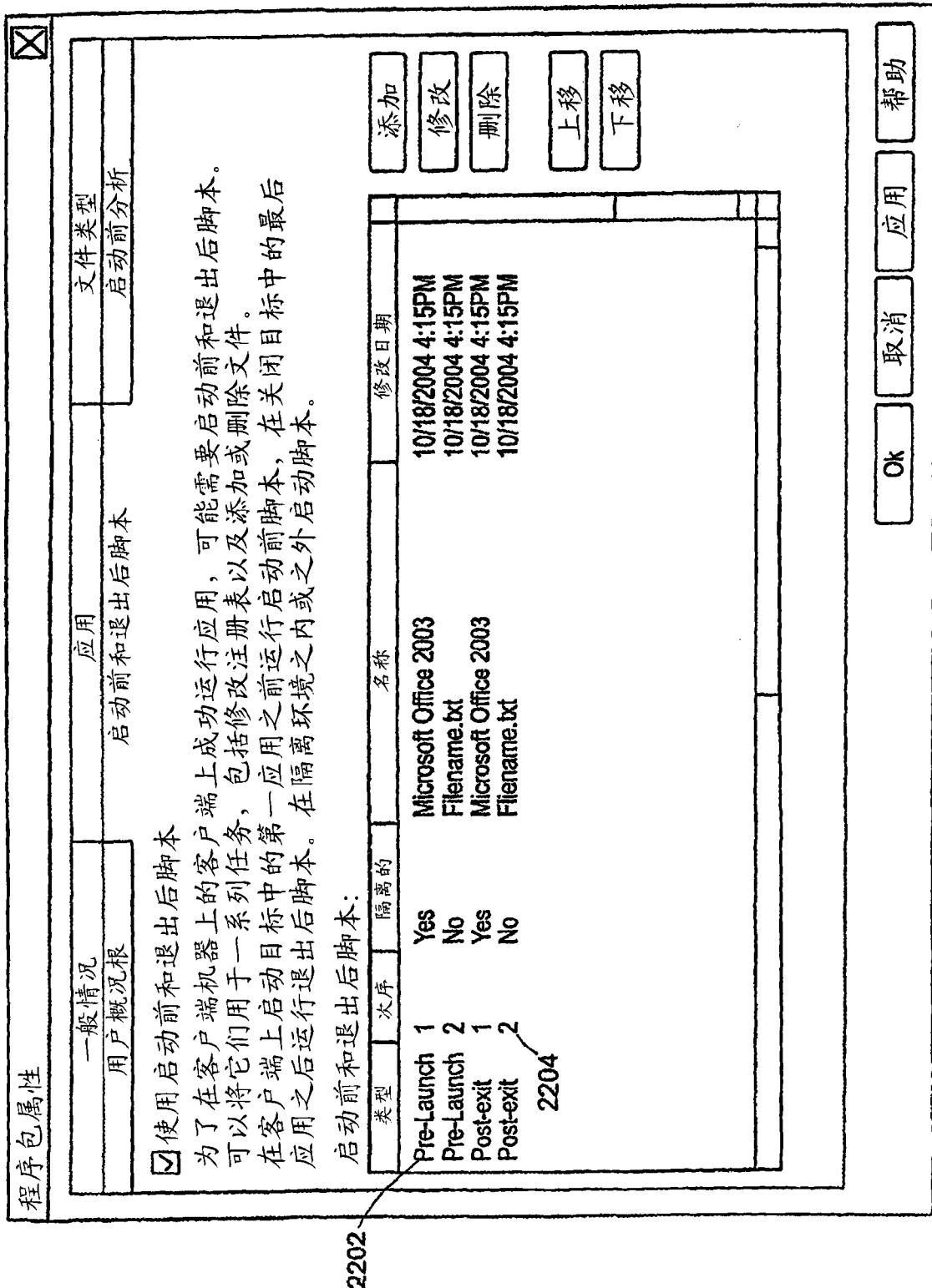


图 22

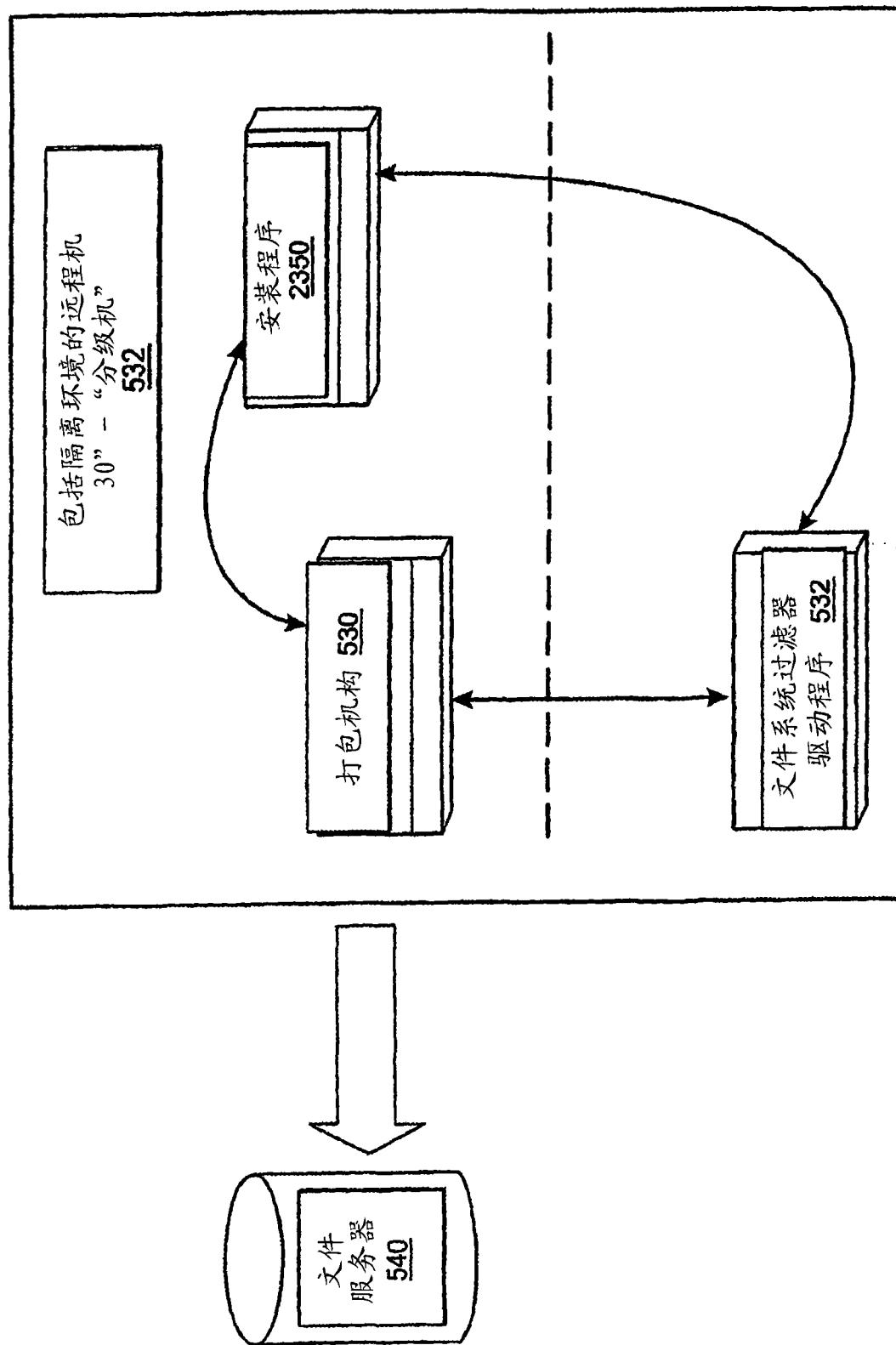


图 23

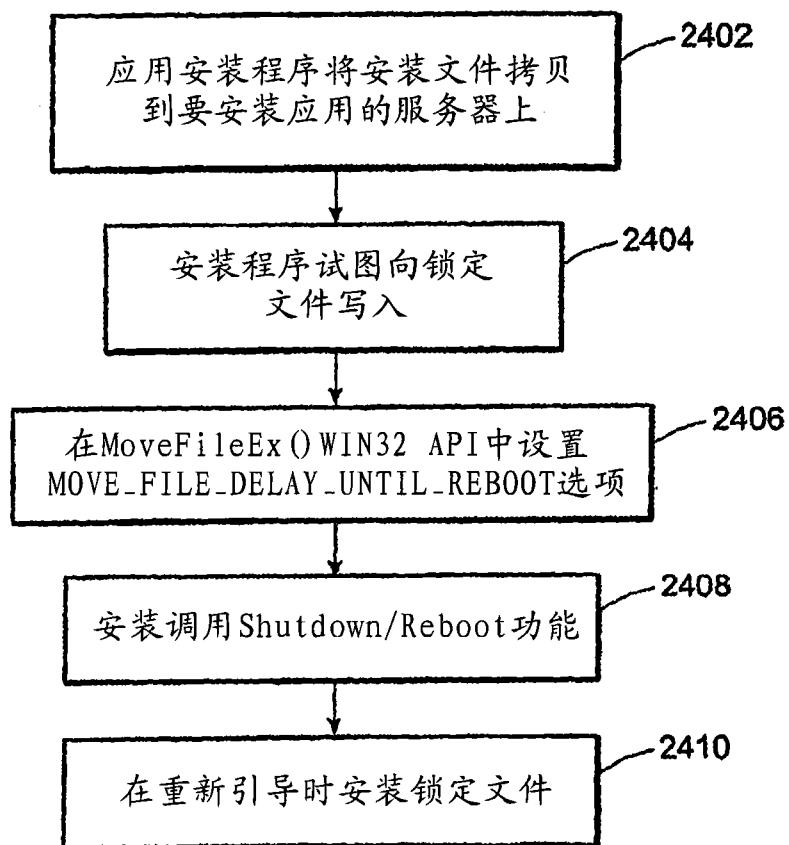


图 24

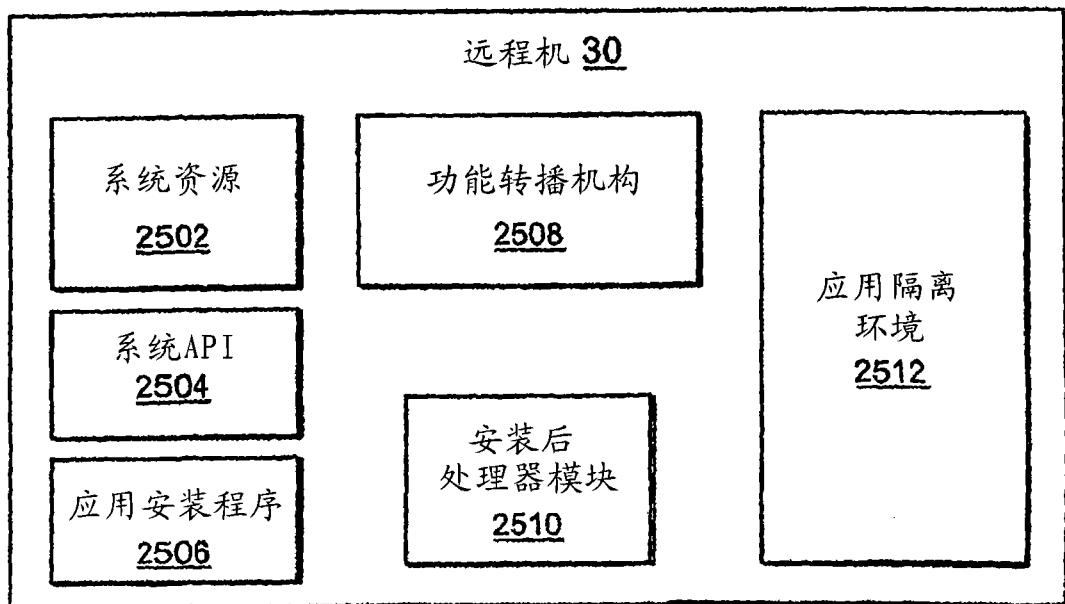


图 25

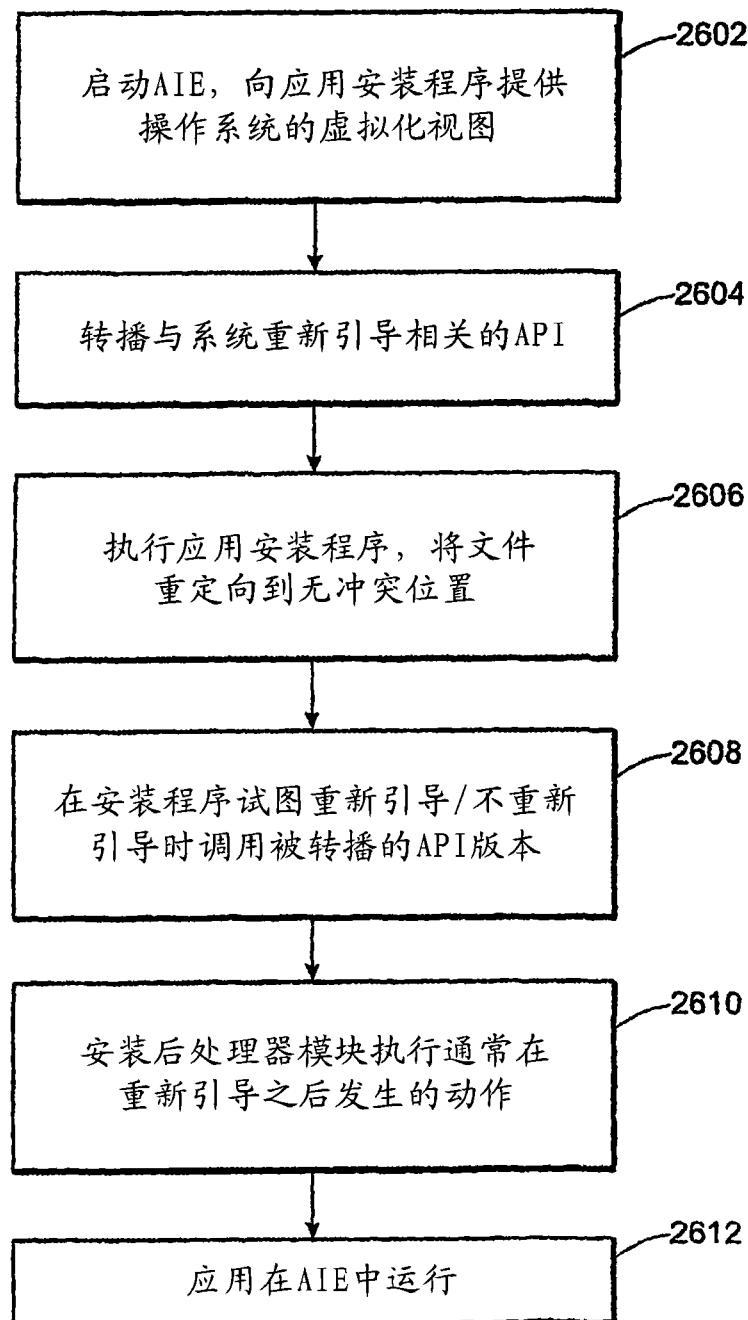


图 26