



(12)发明专利

(10)授权公告号 CN 106165369 B

(45)授权公告日 2019.06.14

(21)申请号 201580015769.3

(72)发明人 蔡继民 李向东 王立鹏

(22)申请日 2015.03.26

T · J · 利特尔

(65)同一申请的已公布的文献号

(74)专利代理机构 中国国际贸易促进委员会专

申请公布号 CN 106165369 A

利商标事务所 11038

(43)申请公布日 2016.11.23

代理人 李晓芳

(30)优先权数据

(51)Int.Cl.

61/971,273 2014.03.27 US

H04L 29/08(2006.01)

14/639,266 2015.03.05 US

G06F 16/23(2019.01)

G06F 9/54(2006.01)

(85)PCT国际申请进入国家阶段日

(56)对比文件

2016.09.23

EP 2746965 A1,2014.06.25,

(86)PCT国际申请的申请数据

CN 1735890 A,2006.02.15,

PCT/US2015/022829 2015.03.26

Bea.BEA elink Data Integration Option

(87)PCT国际申请的公布数据

User Guide.《GOOGLE》.2000,

W02015/148849 EN 2015.10.01

审查员 高悦

(73)专利权人 甲骨文国际公司

权利要求书3页 说明书9页 附图7页

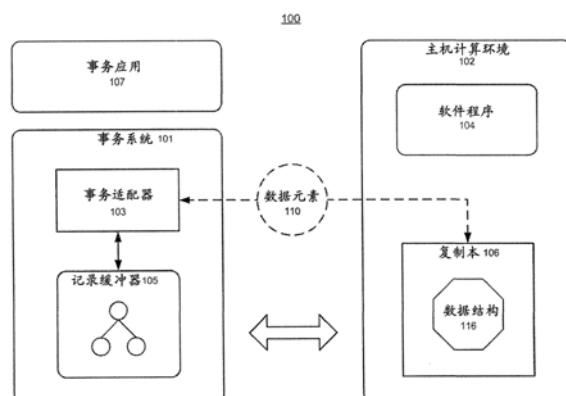
地址 美国加利福尼亚

(54)发明名称

用于在异构计算环境中支持数据类型转换  
的系统和方法

(57)摘要

一种可以在异构计算环境中支持事务系统的  
系统和方法。事务系统可以提供一个或多个缓  
冲器，其中所述一个或多个缓冲器操作以表示异  
构计算环境中的主机计算环境中的一个或多个  
数据结构。事务系统还可以提供事务适配器，该  
事务适配器操作以在主机计算环境中的所述一  
个或多个数据结构和事务系统中的所述一个或  
多个缓冲器之间转换一个或多个数据元素。



1. 一种用于在异构计算环境中支持不同平台之间的事务的系统,包括:  
一个或多个微处理器;  
在所述一个或多个微处理器上运行的事务系统,所述事务系统包括具有记录缓冲器的存储器;以及  
在所述一个或多个微处理器上运行的事务适配器,其中所述事务适配器操作以:  
将主机计算机的数据结构解析成语法树,所述语法树具有多个节点,每个节点表示所述主机计算机的所述数据结构的数据字段,所述语法树表示所述多个节点之间的关系;  
基于所述语法树生成记录描述文件,其中所述记录描述文件描述所述主机计算机的所述数据结构;以及  
根据所述记录描述文件编译所述事务系统的所述记录缓冲器中的记录对象,其中,所述记录对象表示所述记录描述文件中描述的所述主机计算机的所述数据结构;以及  
其中,所述事务系统操作以:  
根据所述记录描述文件对所述主机计算机的所述数据结构的描述访问所述记录对象;  
以及  
操纵所述记录对象。
2. 如权利要求1所述的系统,其中:  
所述主机计算机的所述数据结构在所述主机计算机中的复制本中被定义。
3. 如权利要求1所述的系统,其中:  
所述事务适配器操作以使用树结构来表示所述记录描述文件中描述的所述主机计算机的所述数据结构。
4. 如权利要求1所述的系统,其中:  
所述事务适配器操作以使用序列化的树来在所述记录缓冲器中维护所述记录对象。
5. 如权利要求1所述的系统,还包括:  
记录高速缓存,所述记录高速缓存操作以维护所述记录缓冲器中的所述数据字段。
6. 如权利要求1所述的系统,其中:  
所述事务系统操作以使用一个或多个记录函数来操纵所述记录缓冲器中的所述记录对象。
7. 如权利要求1所述的系统,其中,所述事务适配器操作以生成记录对象文件,其中,所述记录对象文件表示所述记录描述文件中描述的所述主机计算机的所述数据结构。
8. 如权利要求7所述的系统,所述事务系统还包括第一系统变量,所述第一系统变量包括列表,所述列表包括所述记录对象文件和多个其他记录对象文件。
9. 如权利要求8所述的系统,所述事务系统还包括第二系统变量,所述第二系统变量包括目录的列表,所述目录存储所述记录对象文件和所述多个其他记录对象文件。
10. 如权利要求9所述的系统,其中,在所述事务系统上运行的应用使用所述第一系统变量和所述第二系统变量来识别用于所述应用的合适记录对象。
11. 一种用于在异构计算环境中支持不同平台之间的事务的方法,包括:  
由事务适配器将主机计算机的数据结构解析成语法树,所述语法树具有多个节点,每个节点表示所述主机计算机的所述数据结构的数据字段,所述语法树表示所述多个节点之间的关系;

基于所述语法树生成记录描述文件，其中所述记录描述文件描述所述主机计算机的所述数据结构；

由所述事务适配器根据所述记录描述文件编译事务系统的记录缓冲器中的记录对象，其中，所述记录对象表示所述记录描述文件中描述的所述主机计算机的所述数据结构；

由所述事务系统根据所述记录描述文件对所述主机计算机的所述数据结构的描述访问所述记录对象；以及

由所述事务系统操纵所述记录对象。

12. 如权利要求11所述的方法，

其中，所述主机计算机的所述数据结构在所述主机计算机中的副本中被定义。

13. 如权利要求11所述的方法，还包括：

经由所述事务适配器使用树结构来表示所述记录描述文件中描述的所述主机计算机的所述数据结构。

14. 如权利要求11所述的方法，还包括：

经由所述事务适配器使用序列化的树以在所述记录缓冲器中维护所述记录对象。

15. 如权利要求11所述的方法，还包括：

使用记录高速缓存以维护所述记录缓冲器中的所述数据字段。

16. 如权利要求11所述的方法，还包括：

经由所述事务系统使用一个或多个记录函数来操纵所述记录缓冲器中的所述记录对象。

17. 如权利要求11所述的方法，还包括：

经由所述事务适配器创建记录对象文件，其中，所述记录对象文件表示所述记录描述文件中描述的所述主机计算机的所述数据结构。

18. 如权利要求17所述的方法，还包括：

提供第一系统变量，所述第一系统变量包括列表，所述列表包括所述记录对象文件和多个其他记录对象文件。

19. 如权利要求18所述的方法，还包括：

提供第二系统变量，所述第二系统变量包括目录的列表，所述目录存储所述记录对象文件和所述多个其他记录对象文件。

20. 一种非暂态机器可读存储介质，所述非暂态机器可读存储介质具有存储在其上的指令，所述指令在被处理器执行时使得所述处理器执行步骤，所述步骤包括：

由事务适配器将主机计算机的数据结构解析成语法树，所述语法树具有多个节点，每个节点表示所述主机计算机的所述数据结构的数据字段，所述语法树表示所述多个节点之间的关系；

基于所述语法树生成记录描述文件，其中所述记录描述文件描述所述主机计算机的所述数据结构；

由所述事务适配器根据所述记录描述文件编译事务系统的记录缓冲器中的记录对象，其中，所述记录对象表示所述记录描述文件中描述的所述主机计算机的所述数据结构；

由所述事务系统根据所述记录描述文件对所述主机计算机的所述数据结构的描述访问所述记录对象；以及

由所述事务系统操纵所述记录对象。

## 用于在异构计算环境中支持数据类型转换的系统和方法

[0001] 版权声明

[0002] 本专利文档的公开内容的一部分包含受版权保护的材料。版权所有者不反对任何人对专利文档或专利公开进行传真复制,因为它出现在专利商标局专利文件或记录中,但是在其他方面保留所有版权权利。

### 技术领域

[0003] 本公开一般涉及计算机系统和软件,并且更具体地涉及在异构计算环境中支持事务系统。

### 背景技术

[0004] 事务系统包括可以处理组织内的各种事务的企业应用服务器。随着诸如高性能网络和多处理器计算机之类的新技术的发展,存在进一步提高事务系统的性能的需要。这些是本发明的实施例意图解决的一般领域。

### 发明内容

[0005] 本文描述的是在异构计算环境中可以支持事务系统的系统和方法。事务系统可以提供一个或多个缓冲器,其中所述一个或多个缓冲器操作以表示在异构计算环境中的主机计算环境中的一个或多个数据结构。另外,事务系统可以提供事务适配器,该事务适配器操作以在主机计算环境中的所述一个或多个数据结构和事务系统中的所述一个或多个缓冲器之间转换一个或多个数据元素。

### 附图说明

[0006] 图1示出了根据本发明的实施例的在异构计算环境中支持各个应用之间的交互的示例。  
[0007] 图2示出了根据本发明的实施例的在异构计算环境中支持数据类型转换的示例。  
[0008] 图3示出了根据本发明的实施例的复制本(copybook)记录对象描述的示例。  
[0009] 图4示出了根据本发明的实施例的使用序列化的树以支持记录对象结构的示例。  
[0010] 图5示出了根据本发明的实施例的支持记录高速缓存的示例。  
[0011] 图6示出了根据本发明的实施例的在异构计算环境中支持各种事务处理的示例。  
[0012] 图7示出了根据本发明的实施例的在异构计算环境中用于支持数据类型转换的示例性流程图的示例。

### 具体实施方式

[0013] 本发明在附图的图中通过示例的方式而不是通过限制的方式被示出,在附图中相似的附图标记指示相似的元件。应当注意的是本公开中对“一”或“一个”或“某个”实施例(多个实施例)的指代不必然地指代相同的实施例,并且这样的指代意味着至少一个。

[0014] 如下的本发明的描述使用Oracle Tuxedo系统(在下文中简称为“Tuxedo系统”或“Tuxedo”)作为事务系统的示例。对本领域技术人员来说明显其他类型的事务系统可以被使用而没有限制。

[0015] 本文描述的是在异构计算环境中可以支持事务系统的系统和方法。

[0016] 异构计算环境中的数据类型转换

[0017] 异构计算环境中的各种应用可以跨网络使用不同协议彼此通信。

[0018] 例如,主要在主机计算环境中使用的编程语言COBOL使用复制本来定义数据结构。在COBOL中,复制本文件可以被用于定义可以被主机程序引用的数据元素。COBOL记录描述格式是灵活的并且支持各种功能,诸如组、大小可变的表、REDEFINES(重定义)关系。

[0019] 复制本可以包括一个或多个复制本记录基本项和一个或多个复制本记录组项。复制本记录基本项是不具有从属数据项的记录的数据项(即,基本项不被进一步细分)。复制本记录组项包括一个或多个从属数据项。在数据项的一个层级内,基本项可以不属于多于一个组项。另外,组可以包含一个或多个子组。

[0020] 此外,异构计算环境可以包括事务系统(诸如Oracle Tuxedo系统),事务系统可以使用各种缓冲器以支持不同特征。例如,Oracle Tuxedo ATMI(应用到事务管理接口)客户机使用类型化的缓冲器以向ATMI服务器发送消息,类型化的缓冲器是不同类别(或类型)的存储器区域。在Tuxedo系统中,类型化的缓冲器可以可选择地与一个或多个子类别(或子类型)相关联。

[0021] 图1示出了根据本发明的实施例的在异构计算环境中支持事务应用之间的交互的示例。如图1所示,异构计算环境100中的事务系统101可以与在主机计算环境102中部署的各种软件程序104交互。

[0022] 例如,主机软件程序104可以使用复制本106以定义由主机软件程序104使用的一个或多个数据结构116。

[0023] 根据本发明的实施例,事务系统101可以使用记录缓冲器105以表示主机计算环境中的各种数据结构116(诸如复制本106中的记录)。

[0024] 如图1所示,记录缓冲器105可以(在存储器中)使用树结构来映射复制本106中的COBOL记录定义。另外,在记录缓冲器105和复制本106之间可以传送数据。

[0025] 根据本发明的实施例,源数据可以被自动地转换成目的地数据的类型。例如,事务系统101可以使用适配器103(例如Tuxedo主机适配器(TMA))以在主机计算环境102和事务系统101之间自动地转换一个或多个数据元素110。

[0026] 附加地,事务应用107可以直接地访问和操纵记录缓冲器105中的数据。

[0027] 图2示出了根据本发明的实施例的在异构计算环境中支持数据类型转换的示例。如图2所示,异构计算环境200可以支持主机计算环境中的复制本206和事务系统中的记录缓冲器201之间的数据类型转换。

[0028] 根据本发明的实施例,事务系统可以使用工具205以将COBOL复制本206转换为记录对象文件203,记录对象文件203可以由事务应用在运行时使用。

[0029] 如图2所示,转换工具205可以从(用于数据结构216的)COBOL复制本206中生成记录描述文件204。此外,转换工具205可以基于记录描述文件204生成记录对象文件203。在运行时,记录描述文件204可以被用于访问和操纵记录缓冲器201结构,记录缓冲器201结构映

射如记录对象文件203中所表示的COBOL复制本定义。

[0030] 例如,Tuxedo可以使用cpy2record工具来编译源记录描述文件204以创建二进制格式的记录对象文件203。在运行时,Tuxedo可以基于RECORDFILES(记录文件)和RECORDDIR(记录目录)环境变量中的配置来使用记录对象文件203。RECORDFILES变量包含用于应用的对象记录文件的逗号分隔的列表。RECORDDIR变量可以指定可以被用于用相关文件名寻找RECORD(记录)对象文件的目录的冒号分隔的列表。因此,使用C编程语言的Tuxedo应用可以利用COBOL编程语言中的复制本206的记录描述文件204。

[0031] 根据本发明的实施例,转换工具205可以基于不同解析规则将复制本记录类型解析成具有多个节点的语法树。

[0032] 例如,语法树中的节点可以包含不同的Tuxedo VIEW(视图)和RECORD(记录)信息。另外,Tuxedo可以使用DigitParseRule(数字解析规则)以处理COBOL中的外部十进制类型(例如S9999),该外部十进制类型可以被转换为Tuxedo VIEW整数以及RECORD RTYPE\_DIGIT类型。另外,Tuxedo可以使用Comp3ParseRule(Comp-3解析规则)以处理COBOL中的内部十进制类型(即PACKED-DECIMAL(压缩的十进制),例如S9999COMP-3),该内部十进制类型可以被转换为Tuxedo dec\_t类型。此外,Tuxedo可以使用RecordfileGenerator(记录文件生成器)以创建各种记录项。RecordfileGenerator可以确保各种记录项的结构(诸如偏移/类型)被正确地计算和配置。

[0033] 如图2所示,系统可以提供用于直接地定义、访问和操纵存储器210中的记录缓冲器201的一组记录函数202。例如,Tuxedo记录函数202可以包含初始化分配的RECORD缓冲器的函数(例如,Rinit()函数)。记录函数202还可以包含从RECORD缓冲器中检索项或整个记录的函数(例如,Rget()函数)。通过使用Rget()函数,RECORD中的源数据被自动地转换为目的地数据的类型。附加地,记录函数202可以包括将项或整个记录设置到RECORD缓冲器的函数(例如,Rset()函数)。

[0034] 根据本发明的实施例,记录函数202可以通过返回一个或多个错误值支持错误处理。例如,可以通过返回在其他情况下不可能的值(例如,值-1)来指示错误情况。在Tuxedo中,错误消息可以用在Ferror中被定义的错误号码来显示。附加地,Tuxedo可以提供F\_error()函数,F\_error()函数可以产生关于标准错误输出的消息。Tuxedo还可以提供Fstrerror()函数,Fstrerror()函数可以从消息编目中检索错误消息的文本。

[0035] 图3示出了根据本发明的实施例的复制本记录对象描述的示例。如图3所示,包括节点A-J的树300可以被用于表示用于在下面的列表1中示出的复制本的记录对象描述。

**01 A.**

**05 B.**

**10 E PIC S9(9) COMP-5.**

**10 F.**

[0036] **15 K PIC X(4).**

**15 L REDEFINES K PIC S9(8) COMP.**

**05 C PIC S9(4).**

**05 D OCCURS 10 TIMES DEPENDING ON E.**

**10 G PIC S9999 SIGN LEADING.**

**10 H PIC S9(5)V9(2) PACKED-DECIMAL.**

[0037] **10 I USAGE COMP-1.**

**10 J PIC X(10).**

[0038] 列表1

[0039] 如图3所示,在上面的列表1中树300的层级结构表示的复制本中的关系。此外,树300中的节点A-J的每个节点表示复制本中的字段。例如,树300中的节点E、F和K在节点B的下方,这表示在复制本中字段E、F和K在组B下。

[0040] 此外,树300包括链路310,链路310指示节点D依赖于节点E。链路310可以被用于表示可变长度的表,该可变长度的表在字段D的odolink字段中使用OCCURS DEPENDING ON (ODO) (依赖于发生)从句(D OCCURS 10TIMES DEPENDING ON E (D依赖于E发生10次))被定义。

[0041] 附加地,树结构300包括链路320,链路320使用指向字段K的字段L的redefinelink (重定义链路)字段被定义。链路320可以被用于支持REDEFINES (重定义)从句,在上面的列表1中的“L REDEFINES K (L重定义K)”,REDEFINES从句指示不同的数据描述(即,字段L和K)可以描述相同的计算机存储区域。

[0042] 图4示出了根据本发明的实施例的使用序列化的树以支持记录对象结构的示例。如图4所示,对应于树300(如图3所示)的序列化的树410可以包括多个节点(例如,节点A-J)。

[0043] 根据本发明的实施例,使用记录对象文件中的和/或存储器中的序列化的树410实现的记录对象结构400可以被用于表示复制本记录。

[0044] 附加地,节点A-J可以在记录对象结构410中被重新排序。如图4所示,节点A-J可以基于深度优先方案。

[0045] Tuxedo可以使用树状数据结构以表示存储器中的RECORD对象描述。通过使用序列化的树410,Tuxedo中的RECORD的存储器布局可以是相邻的,这反映了复制本布局。因此,Tuxedo应用可以基于提供的RECORD描述容易地操纵RECORD缓冲器。

[0046] 根据本发明的实施例,序列化的树410中的节点A-J中的每个节点表示记录项。例

如,Tuxedo RECORD结构可以与多个RECORDITEM(记录项)元素相关联,该多个RECORDITEM元素中的每个元素可以使用如下面的列表2中示出的C结构被定义。

[0047]

```
struct RECORDITEM {
    unsigned int rname;          /* 在字符串池中到数据项的偏移 */
    unsigned short rhash;        /* 数据项名称的散列值 */
    unsigned int vname;          /* 在字符串池中到相应的视图名称
的偏移 */
    unsigned short type;         /* 数据类型号码 */
    unsigned int flag;           /* 位模式中的标志选择 (依赖意思
类型) */
    unsigned int count;          /* 数据项出现的次数 */
    unsigned int mincount;        /* 数据项的最少出现次数 */
    unsigned int maxcount;        /* 数据项的最多出现次数 */
    unsigned int odolink;         /* 到 OCCURS DEPENDING ON 项的
链路 */
    unsigned int rerulelink;       /* 到重定义规则的链路 */
    unsigned int redfinelink;      /* 到 REDEFINES 项的链路 */
    unsigned int size;            /* 项的大小 */
    unsigned int offset;          /* 项到缓冲器的开头的偏移 */
    unsigned int plink;           /* 到父亲的链路 */
    unsigned int clink;           /* 到孩子的链路 */
    unsigned int fslink;          /* 到前面的兄弟的链路 */
    unsigned int bslink;          /* 到后面的兄弟的链路 */
    unsigned int rec flds[4]; /* 未使用。保留以供未来扩展 */
};
```

[0048] 列表2

[0049] 附加地,各种记录项A-J可以与对应于不同复制本数据类型的不同类型相关联。下面的列表3示出了Tuxedo RECORDITEM元素的不同类型。

[0050]

```
#define RTYPE_RECORD 0 /* RECORD 或组 */
#define RTYPE_DIGIT 1 /* 外部十进制 */
#define RTYPE_UDIGIT 2 /* 无符号外部十进制 */
#define RTYPE_FLOAT 3 /* 单精度浮点型 */
#define RTYPE_DOUBLE 4 /* 双精度浮点型 */
#define RTYPE_LONG 5 /* 8 字节长整型 */
#define RTYPE_ULONG 6 /* 无符号长整型 */
#define RTYPE_INT 7 /* 4 字节整型 */
#define RTYPE_UINT 8 /* 无符号整型 */
#define RTYPE_SHORT 9 /* 2 字节短整型 */
#define RTYPE USHORT 10 /* 无符号短整型 */
#define RTYPE_DEC 11 /* 十进制, dec_t */
#define RTYPE_STRING 12 /* 以空字符终止的字符串 */
串型 */
#define RTYPE_CARRAY 13 /* 字符数组 */
```

[0051] 列表3

[0052] 图5示出了根据本发明的实施例的记录高速缓存的示例。如图5所示，事务系统500可以使用记录高速缓存510以维护不同记录501-505。在Tuxedo中，记录高速缓存510可以被创建以存储用于不同RECORD的概要信息和指针以便在运行时访问每个RECORD。

[0053] 根据本发明的实施例，记录高速缓存510可以使用双链表格式被实现。例如，Tuxedo RECORD高速缓存的大小可以由配置变量（具有默认的128大小的）FRECORDCACHESIZE（记录高速缓存大小）来控制。

[0054] 下面的列表4定义了Tuxedo RECORD高速缓存的结构。

```
#define FRECORDNAMESIZE 32
```

[0055] struct RECORDTBL {

[0056]

```

char rname[FRECORDNAMESIZE];           /* 记录的名称 */
unsigned int totsize;      /* RECORD 的总大小 */
unsigned int nitems;       /* RECORD 中的项的数量 */
unsigned short last;        /* 到上一个 RECORD 的链路 */
unsigned short next;        /* 到下一个 RECORD 的链路 */
char *recordptr;          /* 到 RECORD 的 ptr (指针) */
};


```

[0057] 列表4

[0058] 附加地,记录高速缓存510可以使用近期最少使用(LRU)算法来确定当记录高速缓存510满时,哪个记录可以被丢弃。

[0059] 图6示出了根据本发明的实施例的在异构计算环境中支持各种事务处理的示例。如图6所示,异构计算环境600可以包括可以与一个或多个主机应用607交互的各种事务处理(或处理器),诸如Tuxedo客户机601和Tuxedo客户机602。

[0060] 例如,主机应用607可以基于客户信息控制系统/扩展系统架构(CICS TS)和信息管理系统(IMS)接口或者其他应用程序到程序通信(APPC)或CICS TS接口。

[0061] 如图6所示,诸如Oracle Tuxedo主机适配器(TMA)605之类的适配器可以被用于支持主机应用607与Tuxedo处理(或处理器)601和602之间的通信。TMA 605可以使用通信资源管理器(CRM)来管理通信资源。CRM可以协调在ATMI平台上运行的应用和在主机上运行的应用之间的数据流。

[0062] 根据本发明的实施例,异构计算环境600可以支持在由主机应用607使用的复制本606和由事务处理(处理器)601-602使用的记录缓冲器603(例如,RECORD类型的缓冲器)之间的数据类型转换。

[0063] 如图6所示,基于域的TMA 605允许数据在记录缓冲器603和复制本606之间被自动地转换。因此,TMA 605可以在各种Tuxedo处理601-602和主机应用607之间提供双向连接。

[0064] 例如,Tuxedo可以使用记录缓冲器603以用于表示复制本记录606。记录缓冲器603类型可以具有表示单独的记录结构的子类型。因此,在COBOL复制本606被转换为记录缓冲器603之后,Tuxedo可以利用COBOL复制本606,因为记录缓冲器603提供对COBOL复制本606的访问。Tuxedo客户机601和Tuxedo服务器602还可以基于RECORD描述文件使用如记录缓冲器603中描述的数据结构。

[0065] 此外,Tuxedo允许在记录缓冲器603和其他缓冲器结构(诸如基于C语言结构的Tuxedo FML/VIEW缓冲器604)之间的数据传送。例如,Tuxedo可以使用Fvrtos()函数以将记录缓冲器603转换为FML/VIEW缓冲器604。Tuxedo还可以使用Fvstor()函数以将FML/VIEW缓冲器604转换为记录缓冲器603。

[0066] 根据本发明的实施例,源数据和目的地数据之间的转换可以被自动地执行。例如,记录缓冲器603中的数据使用维持COBOL内部表示的格式。当Rget()函数被调用时,数据被

自动地转换为C语言结构中的数据类型。当Rset()函数被调用时,数据从C语言结构被自动地转换为COBOL内部表示。

[0067] 可替代地,TMA 605可以将复制本606直接地转换为其他缓冲器结构,诸如Tuxedo VIEW/VIEW32缓冲器604.VIEW/VIEW32可能不能正确地表示复制本的所有信息。用户也可能不能直接地操纵这些缓存结构。附加地,如果复制本606中的每个组被翻译成一个VIEW,则系统可以形成大量VIEW。

[0068] 附加地,Tuxedo可以利用不同类型的缓冲器,诸如CARRAY/FML/FML32/STRING/VIEW/VIEW32缓冲器。每个缓冲器可以被指定为特定的类型。附加地,不同类型的缓冲器可能要求不同的例程以初始化、发送和接收消息。不同类型的缓冲器还可能要求不同的例程以对数据编码和解码。Tuxedo可以为特定的类型自动地调用合适的例程而不需要编程介入。

[0069] 图7示出了根据本发明的实施例的用于在异构计算环境中支持数据类型转换的示例性流程图。如图7所示,在步骤701处,在异构计算环境中系统可以在事务系统中提供一个或多个记录缓冲器。此外,在步骤702处,系统可以使用所述一个或多个记录缓冲器以表示异构计算环境中的一个或多个数据结构。然后,在步骤703处,事务适配器可以在主机计算环境中的所述一个或多个数据结构和事务系统中的所述一个或多个记录缓冲器之间转换一个或多个数据元素。

[0070] 本发明可以使用包括一个或多个处理器、存储器和/或根据本公开的教导而被编程的计算机可读存储媒介的一个或多个常规的通用的或专用的数字计算机、计算设备、机器或微处理器被方便地实现。如对于软件领域技术人员将明显的,熟练的程序员可以基于本公开的教导容易地准备合适的软件编码。

[0071] 根据本发明的实施例,提供了异构计算环境中的事务系统。该实施例的事务系统可以是前面的实施例中描述的事务系统(例如,事务系统101、500)中的任何事务系统。此外,事务系统以及它的部件可以基于如在前面的实施例中描述的本发明的原理来执行合适的操作中的任何操作。

[0072] 在该实施例的示例中,事务系统可以包括一个或多个记录缓冲器和事务适配器。一个或多个记录缓冲器可以操作以表示在异构计算环境中的主机计算环境中的一个或多个数据结构。事务适配器可以操作以在主机计算环境中的一个或多个数据结构和一个或多个记录缓冲器之间转换一个或多个数据元素。

[0073] 在该实施例的示例中,一个或多个数据结构可以在主机计算环境中的复制本中被定义。

[0074] 在该实施例的示例中,事务适配器可以操作以使用树结构来表示主机计算环境中的一个或多个数据结构。

[0075] 在该实施例的示例中,事务适配器可以操作以使用序列化的树来维护对象文件或存储器中的树结构。

[0076] 在该实施例的示例中,事务系统还可以包括记录高速缓存,记录高速缓存操作以维护在一个或多个记录缓冲器中存储的一个或多个数据元素。

[0077] 在该实施例的示例中,事务系统可以操作以使用一个或多个记录函数来操纵在一个或多个记录缓冲器中存储的一个或多个数据元素。

[0078] 在该实施例的示例中,事务适配器可以操作以基于主机计算环境中的一个或多个数据结构创建描述文件。

[0079] 在该实施例的示例中,事务适配器可以操作以从描述文件中生成二进制格式的对象文件。

[0080] 在该实施例的示例中,事务系统可以操作以使用基于一个或多个事务数据类型的一个或多个事务缓冲器。

[0081] 在该实施例的示例中,事务系统可以操作以在一个或多个记录缓冲器和一个或多个事务缓冲器之间转换一个或多个数据元素。

[0082] 在一些实施例中,本发明包括计算机程序产品,该计算机程序产品是具有存储在其上/其中的指令的存储介质或计算机可读介质(媒介),该指令可以被用于对计算机编程以执行本发明的处理中的任何处理。存储介质可以包括但不限于任何类型的盘,该任何类型的盘包括软盘、光盘、DVD、CD-ROM、微型硬盘以及磁光盘、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、闪速存储器设备、磁卡或光卡、纳米系统(包括分子存储器IC)或适合于存储指令和/或数据的任何类型的媒介或设备。

[0083] 出于说明和描述的目的提供了本发明的上述描述。它并不旨在是详尽的或者将本发明限制到所公开的精确形式。许多修改和变化对于本领域技术人员来说将是明显的。修改和变化包括所公开的特征的任何相关的组合。实施例被选择并且描述以便于最佳地解释本发明的原理及其实践应用,从而使得本领域其他技术人员能够对于各种实施例以及如适合所设想的特定使用的各种变型来理解本发明。旨在由以下权利要求和它们的等价物来限定本发明的范围。

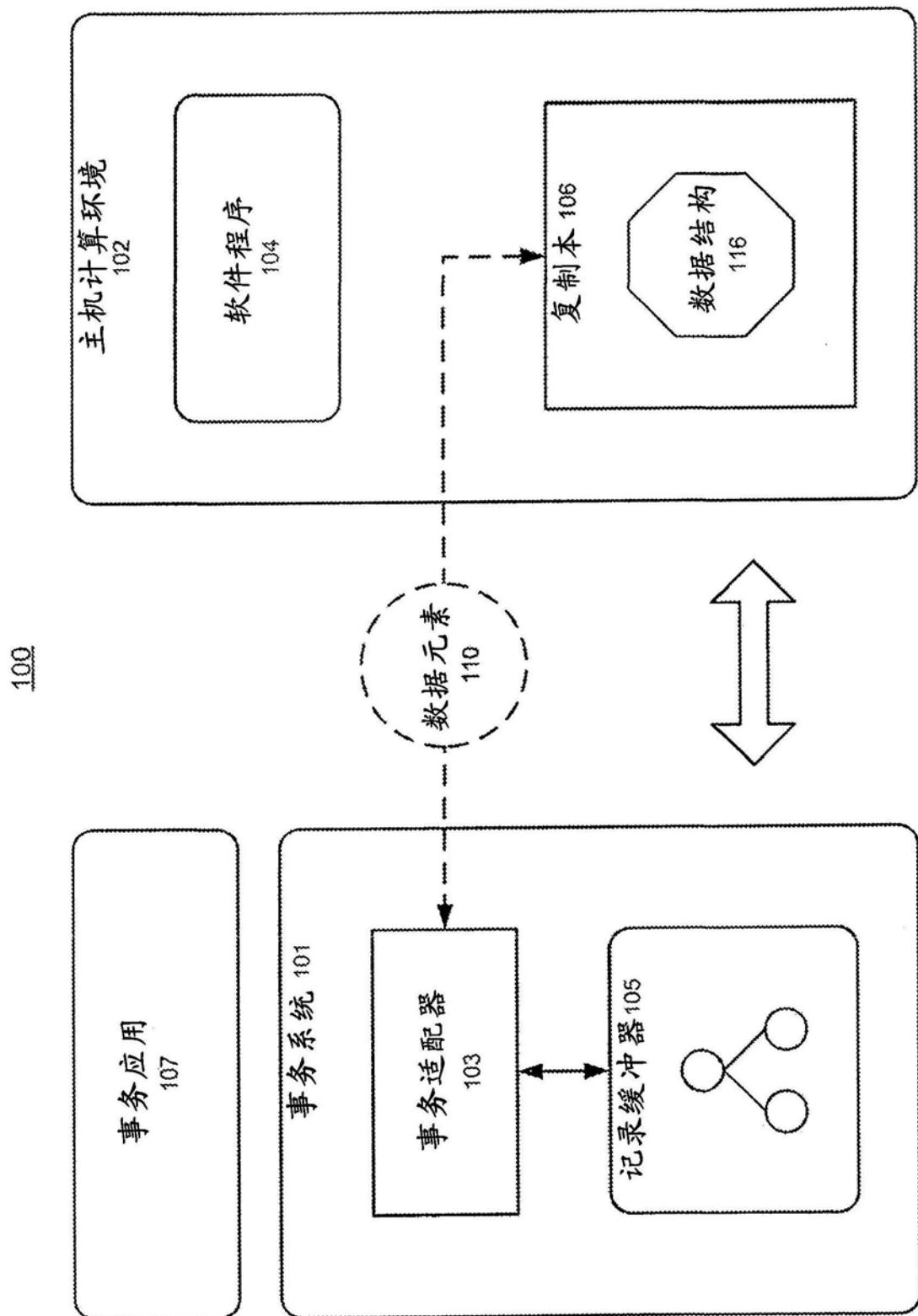


图1

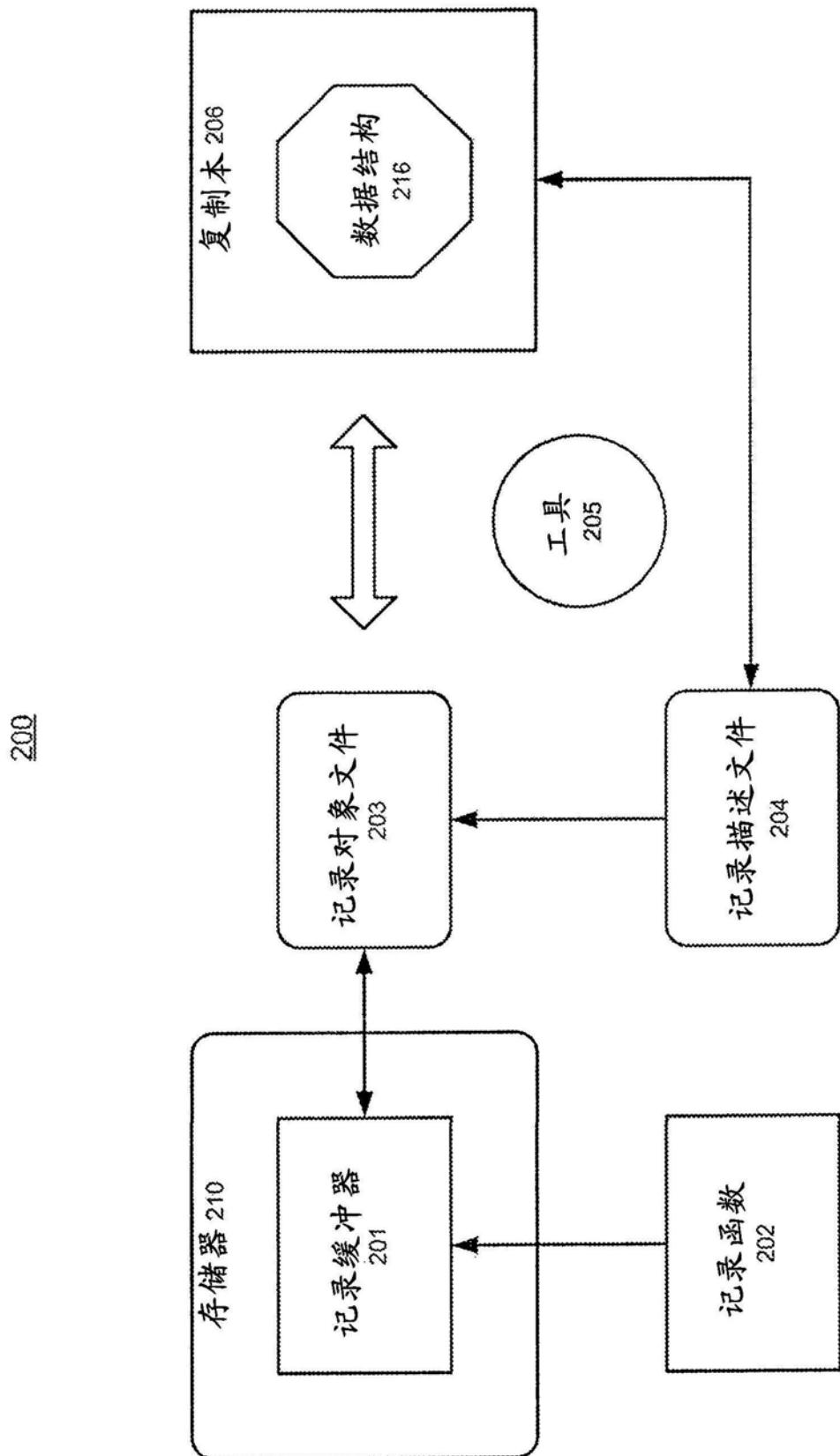


图2

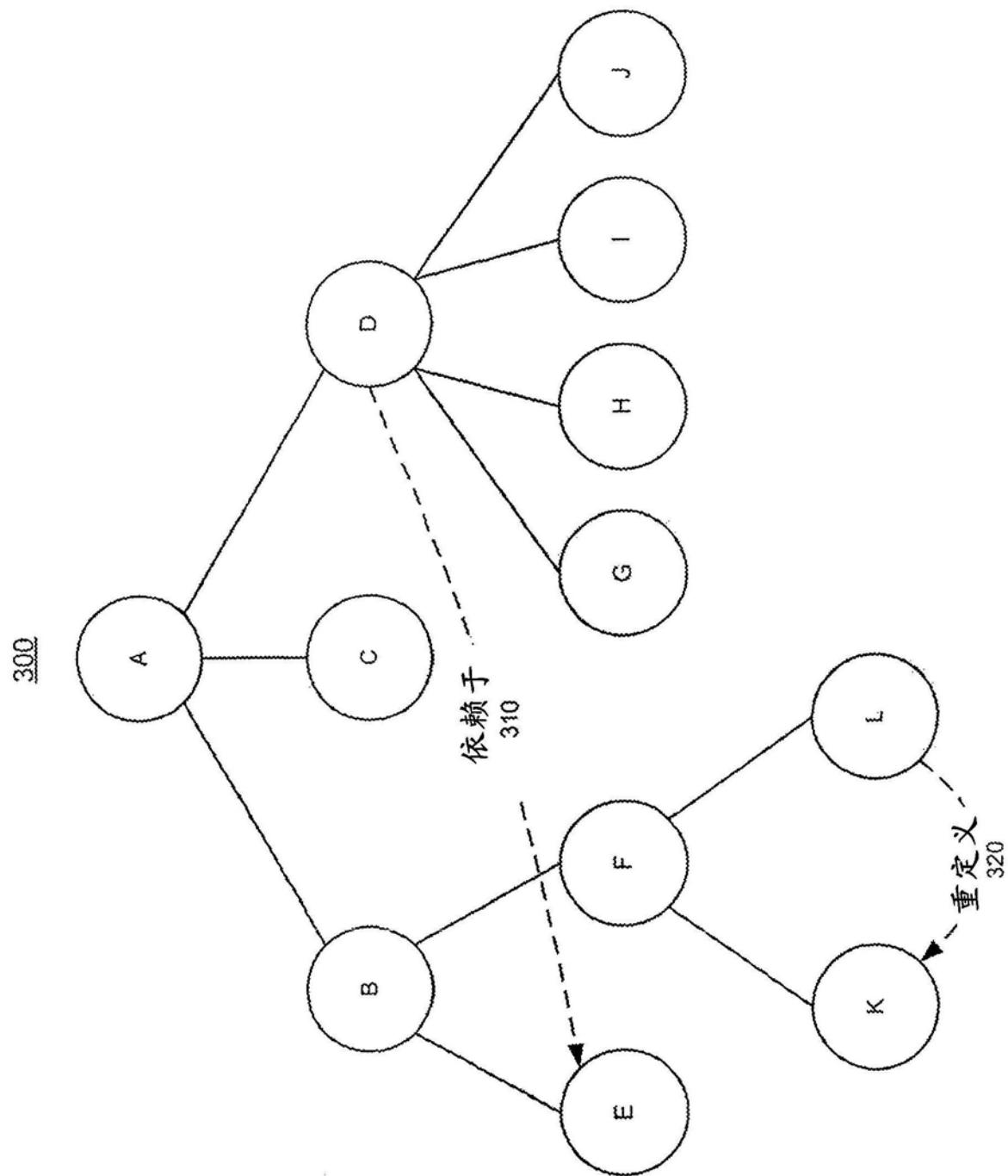


图3

400

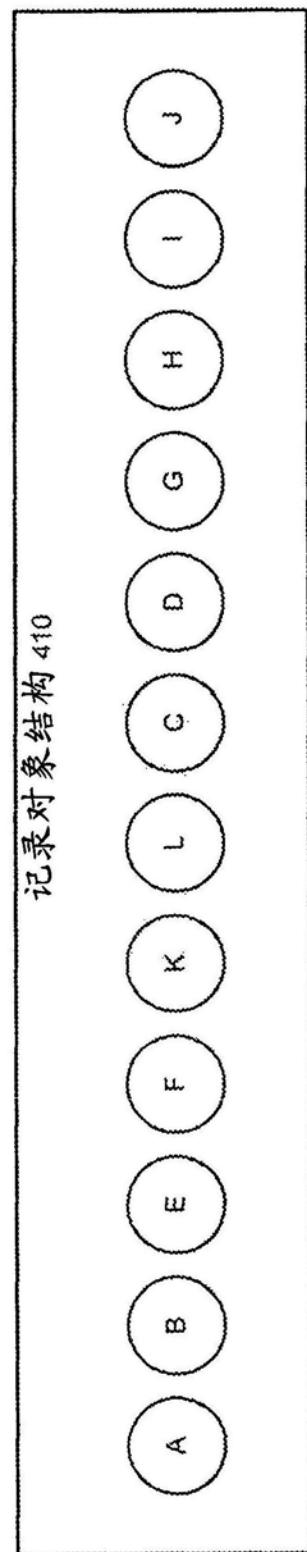


图4

500

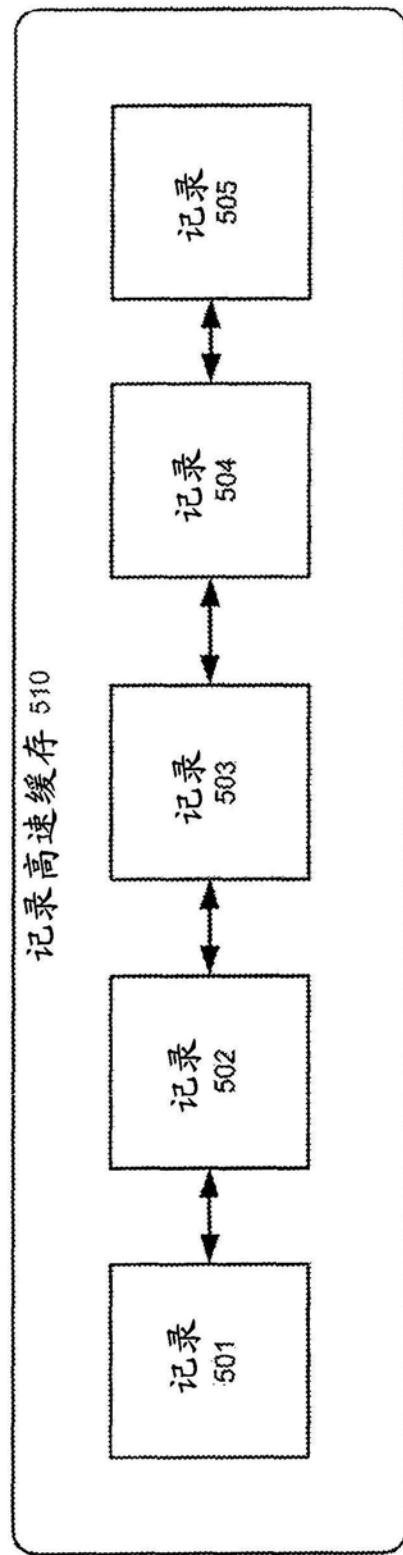


图5

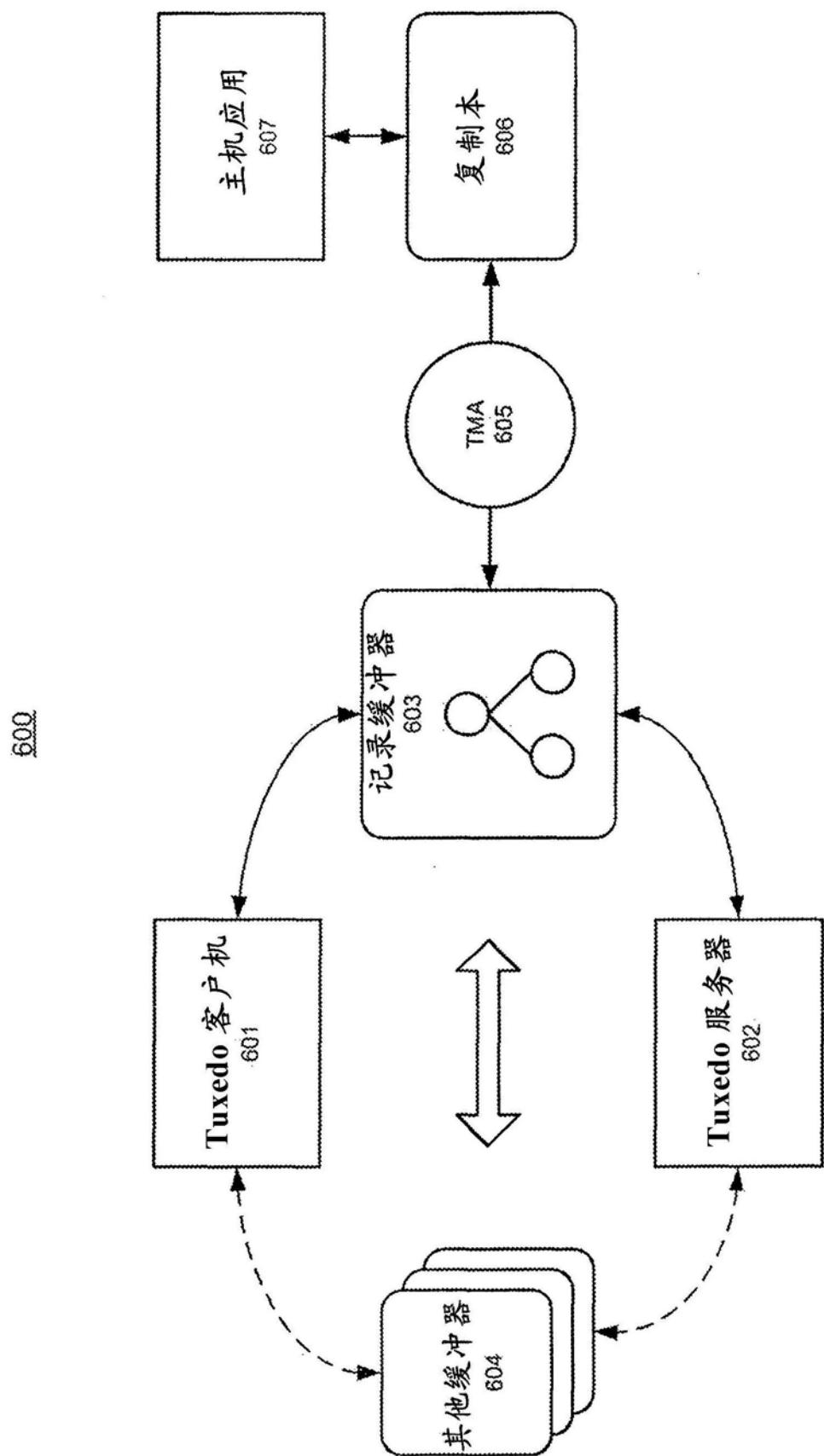


图6

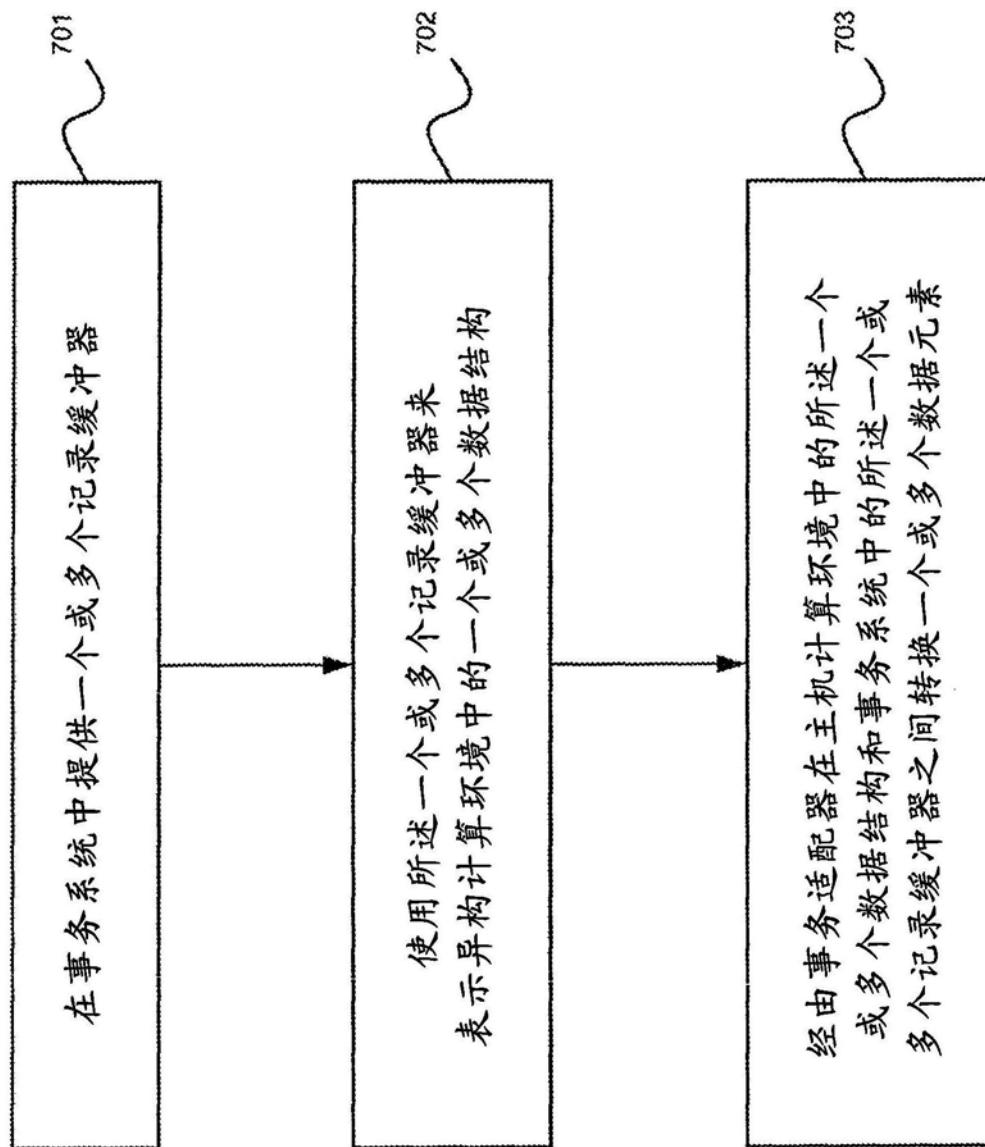


图7