



(12) 发明专利

(10) 授权公告号 CN 101188627 B

(45) 授权公告日 2012. 11. 21

(21) 申请号 200710307383. 7

代理人 陈炜

(22) 申请日 2003. 10. 28

(51) Int. Cl.

(30) 优先权数据

H04L 29/08 (2006. 01)

10/285, 315 2002. 10. 30 US

(62) 分案原申请数据

(56) 对比文件

200380107078. 3 2003. 10. 28

US 6415329 B1, 2002. 07. 02, 全文.

US 6449658 B1, 2002. 09. 10, 全文.

(73) 专利权人 河床技术股份有限公司

审查员 于峰

地址 美国加利福尼亚州

(72) 发明人 S·麦克坎尼 M·J·德玛

(74) 专利代理机构 上海专利商标事务所有限公司
司 31100

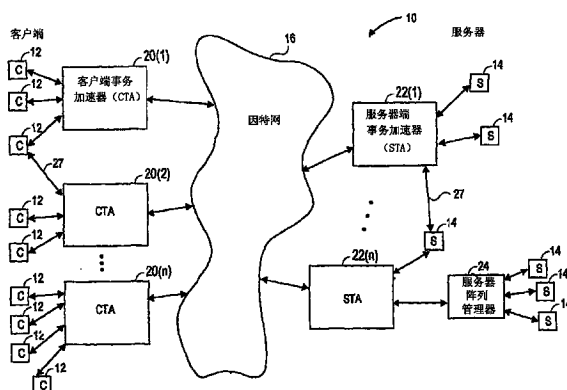
权利要求书 6 页 说明书 21 页 附图 11 页

(54) 发明名称

用于客户-服务器通信系统的事务加速器

(57) 摘要

在具有事务加速的网络中,对于已加速的事务来说,客户直接给客户端事务处理器一个请求,其转发这个请求给服务器端事务处理器,这个处理器依次提供请求或它的表示到用于响应此请求的服务器。服务器发送响应到服务器端事务处理器,其转发这个响应到客户端事务处理器,这个处理器依次提供响应给客户。通过在服务器端事务处理器可存取永久性段存储器中,和客户端事务处理器可存取的永久性段处理器中存储用于事务的数据段,事务可以被事务处理器加速。当数据要在事务处理器之间发送时,发送事务处理器比较要发送的数据段和存储在它永久性段存储器中的段,并关于存储在它永久性段存储器中,与所替代的数据段匹配或近似匹配的条目,来替代数据段。接收事务存储器重构由参考对应来自它永久性段存储器的段数据的替代段发送的数据,按照需要请求遗漏来自发送者的段。事务加速器可以处理多个客户和/或多个服务器,且存储在永久性段存储器中的段可以涉及不同的事务、不同的客户和/或不同的服务器。永久性段存储器可以预先提供来自其他事务加速器的段数据。



1. 在一个网络中,其中客户启动与服务器的任务,网络运送来自客户的请求消息给服务器,并运送来自服务器的响应消息给客户,其中请求消息和响应消息中至少一个包含在网络上从发送端到接收端传送的净负荷,一种任务加速系统包括:

客户端任务加速器包括:

a) 与客户耦合的代理,用于接收用于传输到服务器的请求消息和关联从服务器到客户的重构响应消息;

b) 任务变换器,用于变换从代理接收到的请求消息,包括:

i) 分割器,用于根据基于内容的分割过程分割请求消息净负荷成至少一个段;和

ii) 替换器,用于确定表示所述至少一个段中每个段的段数据作为未参考段还是作为参考段;

c) 永久性段存储器,用于存储段数据和针对请求消息净负荷的参考段的段参考,其中带有不同段数据的段具有不同段参考,不同段参考可以与创建段参考的任务无关;和

d) 反相任务变换器,用于当响应消息已经被加速时,去变换响应消息净负荷成重构的响应消息,包括:

i) 确定段参考出现在响应消息净负荷中位置的记号赋予器;和

ii) 用于为每个被记号赋予器检测到的段参考替换来自永久性段存储器的段数据的反参考器;

服务器端任务加速器包括:

a) 与服务器耦合的代理,用于接收用于传输到客户的响应消息和中继从客户到服务器的重构请求消息;

b) 任务变换器,用于变换从代理接收到的响应消息,包括:

i) 分割器,用于根据基于内容的分割过程分割响应消息净负荷成至少一个段;和

ii) 替换器,用于确定表示所述至少一个段的每个段的段数据作为未参考段还是作为参考段;

c) 永久性段存储器,用于存储段数据和针对响应消息净负荷的参考段的段参考,其中带有不同段数据的段具有不同段参考,不同段参考可以与创建段参考的任务无关;和

d) 反相任务变换器,用于当请求消息已经被加速时,去变换请求消息净负荷成重构的请求消息,包括:

i) 确定段参考出现在请求消息净负荷中位置的记号赋予器;和

ii) 用于为每个被记号赋予器检测到的段参考替换来自永久性段存储器的段数据的反参考器,

其中所述请求消息净负荷和响应消息净负荷均包括至少一个参考段和至少一个未分割数据。

2. 在一个网络中,其中客户启动与服务器的任务,网络运送来自客户的请求消息给服务器,并运送来自服务器的响应消息给客户,其中请求消息和响应消息中至少一个包含在网络上从发送端到接收端传送的净负荷,一种加速任务的方法包括:

根据基于内容的分割过程,分割净负荷成至少一个段;

对所述至少一个段的每个段来说,确定发送在净负荷中的段数据是作为未参考段还是作为参考段;

对于每个参考段来说,将至少一些段数据替换为对于所替换数据的参考,并存储与其参考相关联的替换后的参考数据;

从发送端发送净负荷到接收端作为加速净负荷,其包括段参考、对应于未参考段的符号和未分割数据的至少一个;

在网络上接收加速净负荷;

如果有,在加速净负荷中标识段参考;

用接收端可用的段数据替换加速净负荷中的段参考,来形成重构的净负荷;和

提供重构的净负荷到接收端作为所传送的净负荷;

当段数据被存储作为段参考的一部分时,通过段数据的可逆功能变换段数据,并存储这个变换的结果;和

当段数据被用于解参考段时,使用可逆功能的反相来去变换存储的结果。

3. 在一个网络中,其中客户启动与服务器的任务,网络运送来自客户的请求消息给服务器,并运送来自服务器的响应消息给客户,其中请求消息和响应消息中至少一个包含在网络上从发送端到接收端传送的净负荷,一种任务加速器包括:

用于变换待发送的消息的净负荷的任务变换器,包括:

a) 用于根据基于内容的分割过程,分割净负荷成至少一个段和至少一个未分割数据的分割器;和

b) 用于确定表示所述至少一个段的每个段的段数据作为非参考段还是作为参考段的替换器;

用于存储参考段的段数据和段参考的永久性段存储器;

用于去变换来自远程任务加速器的任务变换器的加速净负荷的反相任务变换器,包括:

a) 确定段参考出现在加速器净负荷中位置的记号赋予器;和

b) 用于为每个被记号赋予器检测到的段参考替换来自所述永久性段存储器的段数据的反参考器;

用于当遇到对于不在所述永久性段存储器中出现的段的段参考时,解析由所述反参考器需要的参考的参考解析器。

4. 在一个网络中,其中客户启动与服务器的任务,网络运送来自客户的请求消息给服务器,并运送来自服务器的响应消息给客户,其中请求消息和响应消息中至少一个包含净负荷,所述净负荷包括根据基于内容的分割过程分割的一个或多个段,其中所述净负荷在网络上从发送端传送到接收端,一种加速任务的方法包括:

对所述至少一个段的每个段来说,确定发送在净负荷中的段数据是作为未参考段还是作为参考段;

对于每个参考段来说,将至少一些段数据替换为对于所替换数据的参考,并将与其参考相关联的替换后的参考数据存储永久性段存储器中;

从发送端发送净负荷到接收端作为加速净负荷;

在网络上接收加速净负荷;

如果有,在加速净负荷中标识段参考;

用接收端可用的段数据替换加速净负荷中的段参考,来形成重构的净负荷;和

提供重构的净负荷到接收端作为所传送的净负荷；

当段数据被存储作为段参考的一部分时，通过段数据的可逆功能变换段数据，并存储这个变换的结果；和

当段数据被用于解参考段时，使用可逆功能的反相来去变换存储的结果。

5. 在一个网络中，其中客户启动与服务器的任务，网络运送来自客户的请求消息给服务器，并运送来自服务器的响应消息给客户，其中请求消息和响应消息中至少一个包含净负荷，所述净负荷包括根据基于内容的分割过程分割的一个或多个段，其中所述净负荷在网络上从发送端传送到接收端，一种事务加速器包括：

用于变换待发送的消息的净负荷的事务变换器，包括用于确定表示每个段的段数据作为非参考段还是作为参考段的替换器；

用于存储参考段的段数据和段参考的永久性段存储器；

用于去变换来自远程事务加速器的任务变换器的加速净负荷的反相事务变换器，包括：

a) 确定段参考出现在加速器净负荷中位置的记号赋予器；和

b) 用于为每个被记号赋予器检测到的段参考替换来自所述永久性段存储器的段数据的反参考器；

用于当遇到对于不在所述永久性段存储器中出现的段的段参考时，解析由所述反参考器需要的参考的参考解析器。

6. 如权利要求 5 所述的事务加速器，其特征在于，还包括用于在所述事务加速器截取发送端和接收端之间的事务的代理。

7. 在一个网络中，其中客户启动与服务器的任务，网络运送来自客户的请求消息给服务器，并运送来自服务器的响应消息给客户，其中请求消息和响应消息中至少一个包含在网络上从发送端到接收端传送的净负荷，一种事务加速系统包括：

客户端事务加速器包括：

a) 与客户耦合的代理，用于接收用于传输到服务器的请求消息和中继从服务器到客户的重构请求消息；

b) 事务变换器，用于变换从代理接收到的请求消息，包括：

分割器，用于将所述请求消息净负荷分割为至少一个段；以及

替换器，用于确定表示每个段的段数据作为未参考段还是作为参考段；

c) 永久性段存储器，用于存储段数据和针对请求消息净负荷的参考段的段参考，其中带有不同段数据的段具有不同段参考，不同段参考可以与创建段参考的任务无关；和

d) 反相事务变换器，用于当响应消息已经被加速时，去变换响应消息净负荷成重构的响应消息，包括：

i) 确定段参考出现在响应消息净负荷中位置的记号赋予器；和

ii) 用于为每个被记号赋予器检测到的段参考替换来自永久性段存储器的段数据的反参考器；

服务器端事务加速器包括：

a) 与服务器耦合的代理，用于接收用于传输到客户的响应消息和中继从客户到服务器的重构请求消息；

- b) 事务变换器,用于变换从代理接收到的响应消息,包括:
 - 分割器,用于将所述响应消息净负荷分割为至少一个段;以及
 - 替换器,用于确定表示每个段的段数据作为未参考段还是作为参考段;
- c) 永久性段存储器,用于存储段数据和针对响应消息净负荷的参考段的段参考,其中带有不同段数据的段具有不同段参考,不同段参考可以与创建段参考的事务无关;和
- d) 反相事务变换器,用于当请求消息已经被加速时,去变换请求消息净负荷成重构的请求消息,包括:
 - i) 确定段参考出现在请求消息净负荷中位置的记号赋予器;和
 - ii) 用于为每个被记号赋予器检测到的段参考替换来自永久性段存储器的段数据的反参考器。

8. 如权利要求 7 所述的系统,其特征在于,还包括:

客户端参考解析器,用于当遇到对于不在客户端的永久性段存储器中出现的段的段参考时,解析由客户端的反参考器需要的参考;和

服务器端参考解析器,用于当遇到对于不在服务器端的永久性段存储器中出现的段的段参考时,解析由服务器端的反参考器需要的参考。

9. 在一个网络中,其中客户启动与服务器的连接,网络通过所述连接运送来自客户的请求消息给服务器,并通过所述连接运送来自服务器的响应消息给客户,其中客户通过第一连接与第一代理进行通信,服务器通过第二连接与第二代理进行通信,并且所述第一和第二代理通过第三连接互相通信,一种在客户和服务器之间传送数据的方法包括:

在所述第二代理通过第二连接从服务器接收净负荷;

对于每个净负荷,根据基于内容的分割过程标识净负荷内的一个或多个数据段;

对于所述一个或多个数据段的每个,确定发送在净负荷中数据段的段数据是作为未参考段还是作为参考段;

对于每个参考段,将至少一些段数据替换为对于所替换数据的参考,并将与其参考相关联的替换后的参考数据存储在永久性段存储器中,以形成加速净负荷;

通过第三连接从第二代理发送加速净负荷到第一代理;

在第一代理通过第三连接接收加速净负荷;

如果有,在加速净负荷中分割段参考;

用客户或第一代理可用的段数据替换加速净负荷中的段参考,来形成重构的净负荷;和

通过第一连接从第一代理提供重构的净负荷到客户作为所传送的净负荷;

当段数据被存储作为段参考的一部分时,通过段数据的可逆功能变换段数据,并存储这个变换的结果;和

当段数据被用于解参考段时,使用可逆功能的反相来去变换存储的结果。

10. 在一个网络中,其中客户启动与服务器的连接,网络通过所述连接运送来自客户的请求消息给服务器,并通过所述连接运送来自服务器的响应消息给客户,其中客户通过第一连接与第一代理进行通信,服务器通过第二连接与第二代理进行通信,并且所述第一和第二代理通过第三连接互相通信,一种客户端事务加速器,包括:

a) 与客户耦合的第一代理,用于通过第一连接接收用于传输到服务器的请求消息、通

过第一连接中继从服务器到客户的重构请求消息、通过第三连接发送加速净负荷到第二代理并通过第三连接从第二代理接收加速净负荷；

b) 事务变换器,用于变换从第一代理接收到的请求消息,包括：

分割器,用于将所述请求消息净负荷分割为至少一个段；以及

替换器,用于确定表示每个段的段数据作为未参考段还是作为参考段；

c) 永久性段存储器,用于存储段数据和针对请求消息净负荷的参考段的段参考,其中带有不同段数据的段具有不同段参考,不同段参考可以与创建段参考的事务无关；和

d) 反相事务变换器,用于当响应消息已经被加速时,去变换响应消息净负荷成重构的响应消息,包括：

i) 确定段参考出现在响应消息净负荷中位置的记号赋予器；和

ii) 用于为每个被记号赋予器检测到的段参考替换来自永久性段存储器的段数据的反参考器。

11. 如权利要求 10 所述的客户端事务加速器,其特征在于,还包括用于当遇到对于不在所述永久性段存储器中出现的段的段参考时,解析由所述反参考器需要的参考的参考解析器。

12. 在一个网络中,其中客户启动与服务器的连接,网络通过所述连接运送来自客户的请求消息给服务器,并通过所述连接运送来自服务器的响应消息给客户,其中客户通过第一连接与第一代理进行通信,服务器通过第二连接与第二代理进行通信,并且所述第一和第二代理通过第三连接互相通信,一种服务器端事务加速器,包括：

a) 与服务器耦合的第二代理,用于通过第二连接接收用于传输到客户的响应消息、通过第二连接中继从客户到服务器的重构请求消息、通过第三连接发送加速净负荷到第一代理并通过第三连接从第一代理接收加速净负荷；

b) 事务变换器,用于变换从第二代理接收到的响应消息,包括：

分割器,用于将所述响应消息净负荷分割为至少一个段；以及

替换器,用于确定表示每个段的段数据作为未参考段还是作为参考段；

c) 永久性段存储器,用于存储段数据和针对响应消息净负荷的参考段的段参考,其中带有不同段数据的段具有不同段参考,不同段参考可以与创建段参考的事务无关；和

d) 反相事务变换器,用于当请求消息已经被加速时,去变换请求消息净负荷成重构的请求消息,包括：

i) 确定段参考出现在请求消息净负荷中位置的记号赋予器；和

ii) 用于为每个被记号赋予器检测到的段参考替换来自永久性段存储器的段数据的反参考器。

13. 如权利要求 12 所述的服务器端事务加速器,其特征在于,还包括用于当遇到对于不在所述永久性段存储器中出现的段的段参考时,解析由所述反参考器需要的参考的参考解析器。

14. 在一个网络中,其中客户启动与服务器的连接,网络通过所述连接运送来自客户的请求消息给服务器,并通过所述连接运送来自服务器的响应消息给客户,其中客户通过第一连接与第一代理进行通信,服务器通过第二连接与第二代理进行通信,并且所述第一和第二代理通过第三连接互相通信,一种事务加速系统,包括：

客户端事务加速器,包括:

a) 与客户耦合的第一代理,用于通过第一连接接收用于传输到服务器的请求消息、通过第一连接中继从服务器到客户的中构请求消息、通过第三连接发送加速净负荷到第二代理并通过第三连接从第二代理接收加速净负荷;

b) 事务变换器,用于变换从第一代理接收到的请求消息,包括:

分割器,用于将所述请求消息净负荷分割为至少一个段;以及

替换器,用于确定表示每个段的段数据作为未参考段还是作为参考段;

c) 永久性段存储器,用于存储段数据和针对请求消息净负荷的参考段的段参考,其中带有不同段数据的段具有不同段参考,不同段参考可以与创建段参考的事务无关;和

d) 反相事务变换器,用于当响应消息已经被加速时,去变换响应消息净负荷成中构的响应消息,包括:

i) 确定段参考出现在响应消息净负荷中位置的记号赋予器;和

ii) 用于为每个被记号赋予器检测到的段参考替换来自永久性段存储器的段数据的反参考器;以及

服务器端事务加速器,包括:

a) 与服务器耦合的第二代理,用于通过第二连接接收用于传输到客户的响应消息、通过第二连接中继从客户到服务器的中构请求消息、通过第三连接发送加速净负荷到第一代理并通过第三连接从第一代理接收加速净负荷;

b) 事务变换器,用于变换从第二代理接收到的响应消息,包括:

分割器,用于将所述响应消息净负荷分割为至少一个段;以及

替换器,用于确定表示每个段的段数据作为未参考段还是作为参考段;

c) 永久性段存储器,用于存储段数据和针对响应消息净负荷的参考段的段参考,其中带有不同段数据的段具有不同段参考,不同段参考可以与创建段参考的事务无关;和

d) 反相事务变换器,用于当请求消息已经被加速时,去变换请求消息净负荷成中构的请求消息,包括:

i) 确定段参考出现在请求消息净负荷中位置的记号赋予器;和

ii) 用于为每个被记号赋予器检测到的段参考替换来自永久性段存储器的段数据的反参考器。

15. 如权利要求 14 所述的系统,其特征在于,还包括:

客户端参考解析器,用于当遇到对于不在客户端的永久性段存储器中出现的段的段参考时,解析由客户端的反参考器需要的参考;和

服务器端参考解析器,用于当遇到对于不在服务器端的永久性段存储器中出现的段的段参考时,解析由服务器端的反参考器需要的参考。

用于客户 - 服务器通信系统的事务加速器

[0001] 本申请是申请日为 2003 年 10 月 28 日申请号为第 200380107078.3 号发明名称为“用于客户 - 服务器通信系统的事务加速器”的中国专利申请的分案申请。

[0002] 相关应用参照

[0003] 美国专利申请系列 10/285,330, 名为“在存储和传输中数据压缩的基于内容分段方案, 包括分级分段表示”[代理记录摘要号 021647-000200US] (在下文中“McCanne II”) 和本申请同一日期申请, 在此为所有目的结合和参考。

[0004] 发明背景

[0005] 本发明通常涉及用于通过有限的带宽信道有效移动数据的系统, 尤其是涉及使可用数据在有限信道上响应对数据请求比数据未处理就发送来响应请求更快。

[0006] 许多在高速连接上运行良好的应用和系统需要适合于在较低速连接上运行。例如, 在局域网 (LAN) 上操作一个文件系统工作良好, 但常常需要在诸如 LAN 的高速链路上访问文件, 不适合于从需要访问文件的客户和服务文件的文件服务器沿着整个路径访问。对其它网络服务, 诸如 e-mail 服务、计算服务、多媒体、食品会议、数据库查询、办公协作等等, 类似的设计问题同样存在。

[0007] 例如, 在联网文件系统中, 在一个地方被应用程序使用的文件可能被存储在另外一个地方。在一个典型的情景中, 大量用户在联网的计算机上操作, 贯串一个组织和 / 或地理区域共享一个文件或存储在一个文件系统中的文件组。文件系统可能接近用户中的一个, 但典型的它远离大多数的用户, 但用户常常期望文件出现在靠近他们的站点。

[0008] 正如在此使用的, “客户”通常指的是可以提出对数据或动作的请求的计算机、计算设备、外围设备、电子设备或类似的, 而“服务器”通常指的是响应由一个或更多客户提出的对数据或动作的请求而操作的计算机、计算设备、外围设备、电子设备或类似的。

[0009] 请求可以是针对计算机、计算设备、外围设备、电子设备或类似设备的操作, 和 / 或针对正在由客户执行或控制的应用程序。一个例子是计算机运行一个需要存储在计算机外部的文档的字处理程序, 它使用网络文件系统客户在网络上对文件服务器发出请求。另外一个例子是对直接在服务器的动作的请求它本身执行一个动作, 诸如打印服务器、处理服务器、控制服务器、设备接口服务器和 I/O (输入 / 输出) 服务器等等。

[0010] 请求常常通过提供请求数据或执行请求动作的响应消息, 或指示无法服务此请求的响应消息, 诸如错误消息或失败或不正确请求的监控系统的警报来满足。服务器还可能阻塞请求、传送请求、变换请求或类似的, 然后响应请求或不响应请求。

[0011] 在有些情况, 通常当作服务器的设备可以担当客户并提出请求, 通常当作客户的设备可以担当服务器并响应请求。此外, 单个设备对于其它服务器 / 客户或它自己来说, 可以既是服务器也是客户。例如, 桌上型计算机可以运行数据库客户和针对数据库客户的用户界面。如果桌上型计算机用户操作数据库客户, 使它提出数据请求, 则数据库客户将发布这个请求, 大概到数据库服务器。如果数据库服务器正在相同桌上型计算机上运行, 桌上型计算机将有效地提出请求给它自己。应该可以理解的是, 如在此所用的, 客户和服务器常常是不同的且被网络、物理距离、安全测量和其它障碍分离, 但那些都不需要客户和服务器的

特征。

[0012] 在有些情况下,客户和服务端不是必须唯一的。例如,在点对点网络中,一个点可能对另一个点请求,但也可能提供对那个点的响应。因此,应该可以理解的是,当术语“客户”和“服务器”典型地在此使用分别作为提出“请求”和提供“响应”的角色的时候,那些元素可能扮演其它不清晰地通过客户-服务器范例描绘的角色。

[0013] 通常,请求-响应环可以被称为“事务”,对于一个给定的事务,有些物体(物理的、逻辑的和/或虚拟的)可以被说成是那个事务的“客户”,有些其它物体(物理的、逻辑的和/或虚拟的)可以被说成是那个事务的“服务器”。

[0014] 通常客户-服务器事务经过分组网络在客户和服务端之间直接流动,但在有些环境中,这些事务可以通过称为“代理”的传输层或应用层设备被截取和转发。在这种情况下,代理是客户连接的终点和另一个连接到代表客户的服务器的起点。可替换地,代理连接到一个或更多依次连接到服务器的其它代理。每个代理可能当事务从客户流到服务器时,转发、修改或不同的变换事务,反之亦然。代理的例子包括(1)Web代理,其通过高速缓存或通过控制访问服务器加强安全来加强性能,(2)邮件中继,其转发来自客户的邮件到其它邮件服务器,(3)DNS中继,其隐藏DNS名字解析等。

[0015] 如在此使用的,术语“近”、“远”、“本地”和“远程”可能指的是物理距离,但更典型的是他们指的是有效距离。二台计算机、计算设备、服务器、客户、外围设备等之间的有效距离至少近似于二台计算机之间获得数据困难的测量。例如,文件数据存储在直接连接到使用这个文件数据的计算机处理器的硬盘驱动器上,通过专用高速总线连接,硬盘驱动器和计算机处理器彼此有效的“近”,但硬盘驱动器和计算机处理器之间的业务在一个低的总线上,具有更多的可能截住数据的插入事件,硬盘驱动器和计算机处理器被说成是更远地分离。

[0016] 较大和较小的物理距离不需要对应较大和较小的有效距离。例如,被高质量和高带宽光纤英里分离的文件服务器和桌上型计算机,与被几英尺分离和通过嘈杂环境中无线连接耦合的文件服务器和桌上型计算机相比,可能具有更小的有效距离。

[0017] 通常,在有效距离大的地方,需要更多的努力来造成较短有效距离的效果。发展了许多方法来造成这个效果。例如,当由于有限的带宽增加有效距离时,使用压缩或通过高速缓存来改善这个限制。压缩是使用较少比特来表示大量数据比特的处理,以这样的方式作,在大多数情况下,可以从压缩处理的反相来恢复原始比特或原始比特的足够近似值。高速缓存是存储先前发送结果的处理,以期望用户将再次请求结果,并比结果来自原始提供者更快速地从高速缓冲存储器接收响应。

[0018] 压缩使得有限带宽更有效使用,且可能导致更少的等待时间,但在有些情况下,没有等待时间改善发生。关于客户-服务器事务的等待时间,是当提出数据请求和请求数据被接收的时间之间的延迟的测量。在有些情况下,如果在提出请求之后压缩数据需要时间,在接收之后解压缩数据需要时间,则压缩可能增加等待时间。如果在进行请求之前的时间之前压缩数据,则可以改善,但如果数据没必要适合压缩时间之前,或如果提供请求的数据量相对于所使用的类似数据量来说太大,则没有可行性。

[0019] 高速缓存还提供有些帮助来减小有效距离,但在有些情况它帮助不了这么多。例如,在单个处理器正在从它控制的存储器检索数据,并以重复的方式这样作的地方,可能是

这样的情况,当从存储器读取处理器指令时,高速缓存可以大大加快处理器的任务。在典型地高速缓冲存储器设置中,请求者请求来自有些存储器的数据,设备等的数据,结果提供给请求者,并存储在比原始设备提供数据具有更快响应时间的高速缓存存储器中。然后,当请求者再次请求这个数据时,如果它仍然在高速缓存存储器中,则高速缓存存储器可以在原始设备能够返回数据之前,响应请求返回数据,并快速满足请求。

[0020] 高速缓存有它的难点,其中之一是数据可能在源头改变,高速缓存存储器将提供“陈旧”数据给请求者。这是“高速缓存一致性”问题。高速缓存的另一个问题是数据的原始源头可能想跟踪数据的使用,将不知道来自高速缓存存储器服务而非来自原始源头。例如,Web 服务器在远离运行“指向”Web 服务器的 Web 浏览器的大量计算机的地方,Web 浏览器可能高速缓存来自它们浏览的站点的 Web 页面,来避免再次下载 Web 页面发生的延迟。虽然这可以改善很多情况下的性能,并降低 Web 服务器的负荷,但是 Web 服务器操作员可能试图跟踪“页面浏览”的总数量,但是将对由高速缓存存储器服务的那些不知情。在有些情况下,因特网服务提供者可能操作远离浏览器的高速缓存存储器,并为大量浏览器提供高速缓存的内容,这样 Web 服务器操作员可能甚至完全错过唯一的用户。

[0021] 另外,在下面 Web 高速缓存的机制为原始数据和高速缓存的数据之间的一致性仅提供自由的模式。通常,基于与原始数据的变化无关的事务中的直观推断或暗示,Web 数据被高速缓存一段时间。这意味高速缓存 Web 数据有时变得和原始服务器不一致,且这样的不一致被 Web 站点操作者、服务提供者和用户简单容忍作为合理的性能平衡。不幸地,这个自由一致性的模式完全不适合于通常的如同联网文件系统的客户-服务器通信。当客户与文件服务器相互作用时,一致性模式必须全部正确,并精确地确保使用文件系统的应用程序的正确操作。

[0022] 网络响应的有些解决方案处理在文件系统或网络层的问题。一个建议的解决方案是低带宽网络文件系统的使用,诸如在操作系统原理 (SOSP' 01) 上的 18th ACM 研讨会, pp. 174-187 (Chateau Lake Louise, Banff, Canada, 2001 年 10 月) (ACM 出版社在 ACM SIGOPS 操作系统回顾的 vol. 35, 5 中) 的会议记录中, Muthitacharoen, A., et al “的带宽网络文件系统”中所描述的。在那个称为 LBFS 的系统中,客户采用“整个文件”高速缓存,由此在文件打开操作上,客户从服务器取来文件中所有的数据,然后操作在本地高速缓存文件数据的拷贝。如果客户对文件进行了改变,则当客户关闭文件时,那些改变被传送回服务器。为了最优化这些传输, LBFS 使用无用信息替换文件片,收件人使用与存储本地文件相关联的无用信息来解析无用信息到文件的原始部分。这样的系统由于他们受文件系统的束缚,所以有限制,且通常要求改善客户和服务器之间响应的修改。此外,无用信息机制在相对大(平均)大小的信息块上运行,当文件受有细密纹理的变化超时支配时,其工作得很差。最后, LBFS 被设计得密切受网络文件系统协议的束缚。不能优化或加速其它类型的客户-服务器事务,例如 e-mail、Web、流媒体等等。

[0023] 另一个提议的解决方案在 ACM SIGCOMM (2000 八月) 的会议记录中, Spring, N., et al, “用于消除冗余网络业务的协议独立技术”被提出。如在那个参考中所描述的,类似于最近发送的分组网络分组可以通过标识重复字符串和使用标记替换重复字符串来减小大小,标记由在网络任一端的共享分组高速缓存存储器解析。这个方法,虽然有利,但具有大量的缺点。因为它在个别的分组上单独运行,所以增加的性能增益被分组净负荷大小

对分组报头的比例限制（因为分组报头通常不能使用上述技术压缩）。同样，因为机制是在分组等级实现，所以它仅仅应用到通信路径的二端已经配置设备的网络区域。这个结构难于实现，且在有些环境中可能不切实际。同样，通过使用具有先进先出替换策略的相对小的基于高速缓冲的存储器，来高速缓存网络分组（例如，没有大磁盘备份存储的帮助），此方法的功效被限制于检测和使用被及时本地化的通信冗余。最后，因为这个方法没有任何束缚于产生（冗余）的网络业务的应用或服务，所以不能预期数据被使用的地方和远端高速缓冲存储器中数据提供网络业务潜在更加加速和优化的前一阶段。

[0024] 在跨越到广域网络操作的企业中，应对上述的问题做了许多欠理想的补丁。例如，有些企业采取购买越来越多的带宽来保持响应。组织中的个体将试图通过求助于特别的 e-mail 协作来本地解决（其可能使一个文件更加容易被一个用户取得，但增加了版本控制问题和增加了整个网络的负荷）。其它解决问题的试图可能包括手工创建数据拷贝来运行或推进只读复制品给远程服务器。

[0025] 考虑到上面的问题和目前解决方案的限制，可以对如何在网络上为事务传送数据进行改善。

[0026] 发明简述

[0027] 在具有事务加速器的网络实施例，对于加速事务来说，客户直接向客户端事务处理器提出请求，其转送这个请求给服务器端事务处理器，服务器端处理器依次提供请求或它的表示给用于响应请求的服务器。服务器发送响应给服务器端事务处理器，其转发响应给客户端事务处理器，客户端事务处理器依次提供响应给客户。通过在服务器端事务处理器可存取的永久性段存储器中，和客户端事务处理器可存取的永久性段处理器中存储用于事务的数据段，事务可以被事务处理器加速。当数据要在事务处理器之间发送时，发送事务处理器比较要发送的数据段和存储在它永久性段存储器中的段，并参照永久性段存储器中与所替代的数据段匹配或近似匹配的条目，来替代数据段。将被发送的数据可能从客户发送到服务器、从服务器发送到客户、从一个点发送到一个点等等。然后接收事务存储器通过用其永久性段存储器的对应段数据来替换段参考，而重构发送数据。如果段是指向，但不存在于接收器的永久性段存储器中，则接收方可以经由端信道或经由用于发送参考到段的链路来发布对来自发送方的丢失段的请求。在每个端的永久性段存储器可能填充着可能被重复的段，这样将常常发生段替换，结果在网络上使用更少的带宽，从而加速了事务。

[0028] 事务加速器可以是专用的，诸如客户端事务加速器仅仅与一个客户交互，服务器端事务加速器仅仅与一个服务器交互，但事务加速器还可能处理超过一个的客户和 / 或超过一个的服务器。在处理多个事务的地方，不是针对同样的客户和服务器，就是遍及可能不同的客户和可能不同的服务器，存储在永久性段存储器的段可以涉及不同的事务、不同的客户和 / 或不同的服务器。例如，如果事务加速器遇到数据段，并存储它到处理给定事务中它的永久性段存储器，数据段的参考可能在不同的事务中再次被使用，涉及不同客户或相同客户和不同服务器或相同服务器，或涉及整个不同客户 - 服务器应用。

[0029] 在有些实施例中，事务加速器的永久性段存储器预先填充了来自其它事务加速器的段数据，这样当事务发生时，更多的段可在发送端被用于替换参考，且更多段可用于在接收端从参考重构。

[0030] 考虑到下面的详细描述和较佳实施例，本发明的其它特点和优点将变得明显。

[0031] 附图简述

[0032] 图 1 是根据本发明第一实施例的联网客户 - 服务器系统的方框图。

[0033] 图 2 是图 1 系统的方框图,更详细地显示客户端事务加速器 (“CTA”) 和服务器端事务加速器 (“STA”),考虑到空间条件,整个系统稍微简略显示。

[0034] 图 3 是可用于图 1 所示系统的永久性段存储器 (“PSS”) 的实施例中数据组织的图示说明。

[0035] 图 4 是可用于图 2 的事务变换器 (“TT”) 的编码器的方框图。

[0036] 图 5 是可用于图 2 的反相事务变换器 (TT-1) 的解码器的方框图。

[0037] 图 6 是通过参考数据段来分割和表示输入数据的编码处理的图示说明。

[0038] 图 7 是图示说明用于解码从图 4 的编码器输出的数据的处理过程的流程图。

[0039] 图 8 是实现事务加速器并使用预发式段分配器 (“PSD”) 的联网系统的方框图。

[0040] 图 9 是根据本发明实施例的联网点对点系统的方框图。

[0041] 图 10 是实现事务加速器且客户端事务加速器集成在客户里的联网系统的方框图。

[0042] 图 11 是实现事务加速器且服务器端事务加速器集成在服务器里的联网系统的方框图。

[0043] 图 12 是实现事务加速器且 PSS 在大量事务加速器中共享的联网系统的方框图。

[0044] 图 13 是显示多点传送被用于更新和读取共享 PSS 的图 12 的系统的多点传送实现的方框图。

[0045] 图 14 是显示大量本地耦合的客户经由 LAN 到 WAN 的多点传送实现的方框图。

[0046] 图 15 是实现事务加速器且网络处理多种协议和服务的联网系统的方框图。

[0047] 发明详述

[0048] 本发明有许多应用,阅读这个说明后将变得明显。在描述根据本发明的事务加速器系统实施例中,仅仅描述少数几个可能的变化。本领域的普通技术人员的一个将明白其它应用和变换,所以发明将不仅于例子所说明的,而是结合附加的权利要求书。

[0049] 事务作为在这里使用的术语,是导致数据从一个地方移动到另一个地方的一组逻辑步骤。在有些情况下,被移动的数据存在于它与事务无关的发端,诸如文件存在服务器的磁盘中的文件读取事务。在其它情况下,在发端产生用于事务的数据,诸如响应计算请求、查询等等。典型地,启动事务的计算机、计算设备等等称为“客户”,响应或被期望响应的计算机、计算设备等等称为“服务器”。数据可以以任一方向流动。例如,文件系统客户可以通过请求文件读取来启动事务。相应数据将从响应此请求的服务器返回,所以在这个情况下,大多数数据从服务器流动到客户。然而,在客户启动文件写入事务的地方,大量数据从客户流动到服务器,或者作为启动请求的一部分或作为后续消息。事务可以在多个部分中,但在简单事务中,客户发送请求 (数据、详细、信号等等明确地是请求或是请求的表示或指示) 给服务器,服务器使用响应 (数据、详细、信号等等明确地是响应或请响应表示或指示) 来响应给客户。例如更复杂的事务可能包括一些服务器需要用来解释请求、验证客户接收请求的响应的权限、获得用于准备响应所需的附加信息等等的来来回回。

[0050] 在此,客户和服务器之间连接的典型例子是分组网络,但其它连接方法也可能被使用,诸如点对点有线或无线信道。将归纳这些组件,在此称为具有假设节点之间通信的信

道的“节点”。

[0051] 事务可能首先由客户在一个节点直接对在另一个节点的服务器发出一个用于文件数据的请求开始,随后有包含所请求的文件数据响应的传输。其它事务可能是对文件特指部分、所有文件、另外数据构造的全部或有些的请求,或者事务可能涉及从请求者流动的数据或涉及一个命令。事务的例子包括“读取数据块”、“读取文件”、“读取流”、“写入携带这个数据的数据块”(从请求者流动数据的例子)、“打开文件”、“对这个数据执行计算”、“获得携带这些特征的电子邮件”、“发送电子邮件”、“检验新电子邮件”、“列出目录内容”等等。

[0052] 有些事务可能包括大量以一个方向或二个方向流动的数据。有些事务甚至可能包括具有超过一个的请求者和 / 或超过一个的接收者的事务。为了清楚地说明,按照典型简单事务来描述这些许多的事务类型,一个客户提出一个服务器请求且一个服务器以客户期望的某种方式响应请求。然而,基于阅读这个说明,普通技术熟练人士中的一个能够应用这些概念到客户和服务器之间或更通常的二个节点之间的一对多和多对多事务。在描述一个方向数据流动的地方,应该可以理解的是数据可能以其它方向流动和 / 或信息可能仅仅以一个方向流动,但数据和 / 或信号以二个方向流动来完成信息的移动。

[0053] 使用在此描述的一些系统,客户访问服务器(在需要的地方反之亦然)可以通过事务加速器“穿过”,其匹配事务到具有内容感应段切点的可变长度段序列。段可以存储在各种位置,典型地在客户和服务器的访问中,使用可缩放存储段在持久性命名系统中。段可以从文件系统或其它系统数据块和结构中去耦,使得可以在多个上下文中找到匹配段。作为替代高速缓冲存储文件、数据块或其它系统依靠的结构,段可以被存储和被用于表示段内容的参考约束。

[0054] 图 1 使根据本发明实施例的联网客户-服务器系统 10 的方框图,在此可能发生这样的事务。如那里所示,客户 12 在网络 16 上经由客户端事务加速器 (“CTA' s”) 20 和服务端事务加速器 (“STA' s”) 22 与服务器 14 耦合。在事务加速器的位置不是特指的地方,它在此称为“TA”,指示它可被称为客户端事务加速器、服务端事务加速器、点事务加速器或可能甚至是客户和服务器(也可能是点)使用的事务加速器。

[0055] 虽然图 1 中没有显示,客户和服务器之间(也可能客户和客户之间以及服务器和服务器之间)的附加路径存在并旁路 TA' s。这样附加路径可以用于承载传统业务,诸如不可能受益于事务加速器的事务。通过发送在 TA' s 周围这样的事务,TA' s 的状态可以保持集中在加速的事务上,例如,通过不使 TA 的永久性段存储器(下面描述的)存储来自不可能受益于事务加速器的事务的段。

[0056] 如图所示,CTA20 可能服务一个或更多客户,多个 CTA' s20 可能在一个网络上实现。如在此使用的且除非另外指出,索引“n”指的是不定整数,每个索引的不同使用指的是不同的不定整数。例如,图 1 图示说明可以有一些数量的 CTA' s 和一些数量的 STA' s,不必一个对一个对应。通常,CTA' s 的数量可能基于客户的数量、期望客户的数量、网络设计等等,而 STA' s 的数量可能基于服务器的数量、期望服务器的数量、网络设计等等。在一些实现中,每个服务器连接到专用于那个服务器的 STA。在一些实现中,大量服务器可能耦合到服务器阵列管理器 24,并依次经由 STA 耦合到因特网 16。在一些情况下,客户可能与多于一个的 CTA 相互作用,如图 1 线 27 所图示说明的,服务器可能和多于一个的 STA 相互作用,如图 1 线 29 所图示说明的。

[0057] 在一个 CTA 和一个 STA 相互作用并从连接到那个 CTA 的多个客户接收到请求的地方,对应 STA 发送每个客户的请求给请求所指向的服务器。然而,TA' s 可能更加接近耦合它们的客户 / 服务器,这样所有或近似所有对一个服务器的加速事务通过一个服务器的 STA。另外,在一些实现中,TA' s 共享状态,使得在一个 TA 的事务可能受益于存储在另一个 TA 的段。

[0058] 客户连接可能以许多的方法被发送到 CTA,类似于关于客户的多么优先技术代理功能。例如,使用域名服务器 (DNS) 重新指向可以用于导致客户来分析 CTA 而不是服务器的 IP 地址,从而发送请求给 CTA。可替换地,客户或客户的应用程序可以固定配置来使用特殊的 CTA 或在每个应用程序基础上的一组 CTA' s。一旦客户连接到达 CTA,然后 CTA 可以经由能以许多方法工作的查询处理联系适当的 STA。例如映像表 (保持在集中且可查询数据库上或配置到 CTA) 可用于指向 CTA 到适当的 STA;或在事务中传送的信息可以允许 CTA 发现使用哪个 STA;或可配置政策可以被编程到 CTA 指示哪个传送端口应该转播到哪个 STA' s。同样,STA 可以使用类似的查询处理来决定对从 CTA 到达的新客户连接联系哪个服务器。STA 也可以使用事务中的数据来推断服务器连接什么服务器 (例如在对 CIFS 文件服务器连接进行一个连接建立请求时,HTTP Web 请求包含服务器的身份)。

[0059] 应该可以理解的是,当图 1 所示的网络是因特网,如今普遍使用网络的全球因特网网络的时候,为此可以替代其它网络。例如,因特网上的网络业务可以经过公共网络传播,并主要基于 TCP/IP (传输控制协议 / 因特网协议) 分组交换。然而,在此显示的发明的实施例还可能在不是公共的网络上使用,诸如内联网、外联网和虚拟私有网络。实施例还可能用于 WAN' S、LAN' S、WAN/LAN 耦合、无线连接、移动链路、卫星链路、蜂窝电话网络 and 任何其它关注响应的网络。此外,当如今 TCP/IP 是最通用的分组交换协议且因此作为一个好的例子时,可能使用其它网络协议 (以太网等等)。为了覆盖那些协议,在此描述的客户和服务器 (和点,如下面描述的) 可能使用 HTTP,FTP,SNMP,POP3,IMAP,SMTP,NFS,CIFS,RPC 或其它用于数据传输的开放或私有的协议。

[0060] 在一个公用事务中,客户发送对文件、数据块或其它数据单元的请求给服务器,如果可能的话,服务器使用数据响应来响应此请求。例如,在客户是运行计算机辅助设计 (CAD) 程序的计算机,且需要存储在文件服务器上的 CAD 文件的地方,客户可能阐明对文件的请求,压缩此请求为消息并在网络上发送这个消息给适当的文件服务器。然后文件服务器可能实现鉴别,检验在文件服务器上文件的存在,如果客户被批准拥有文件且文件存在,则文件服务器可能创建一个消息或一组消息或包含所请求文件数据的分组,并发送这些消息 / 分组到提出请求的客户。

[0061] 使用 TA' s 可能改善事务的响应,即加速事务。在典型环境中,客户和 CTA' s 之间的链路 27 是快速链路,诸如本地区域网络 (LAN) 链路及等待时间和带宽是较低的在网络 16 上的链路。“等待时间”指的是发送消息时间和接收到它时间之间的时间 (通常以时间单位度量),“带宽”指的是为特定任务在一个链路上可以承载多少容量 (通常以每单位时间比特数量来度量)。在许多情况下,低带宽可能导致高等待时间,但那些因素可以是独立的,这样可能有高带宽但仍然有高等待时间。其它因素可能影响响应和 / 或带宽开销,诸如链路的可靠性和带宽使用率。

[0062] 在使用 TA' s 的典型文件请求事务中,客户 12 通过发送请求消息启动与服务器

14 的事务。如上面所说明的,如果事务包括小数量的比特或其它因素存在,则使用 TA' s 不能使得事务完成得有任何加快,因此事务可能在传统分组路径上进行。然而,无论如何事务可能经过 TA' s,其可能是有用的,如下面所说明的,这样 TA' s 对业务有更为完整的视图。作为一个例子,如果客户请求通过 CTA,则 CTA 可以记住请求并协调对请求的响应,来提供附加服务给客户。CTA 还可能使用请求来猜测在将来什么时间事情可能发生,并使用这些猜测来进一步优化事务加速器的处理。

[0063] 当服务器 14 接收一个请求时,它配置一个响应给请求,并经由与它耦合的 STA22 发送它给客户。在基本实现中,每个客户耦合到一个 CTA,每个服务器耦合到一个 STA,但在更复杂的实现中,服务器可能耦合到多于一个的 STA,并使用一些优化逻辑来确定在哪个时间用哪个 STA。客户可能耦合到多于一个的 CTA,并使用一些优化逻辑来确定在哪个时间用哪个 CTA。

[0064] CTA20 可以无变化地发送请求到适当的 STA 和 / 或接收 STA22 可以接收来自服务器的响应并无变化地发送这个响应到适当的 CTA20。然而,在请求或响应包括大量数据的地方,可能期望有效的事务加速器在这样的情况,数据是否如在此所述的通过在接收端存储数据段,在发送端使用存储段的参考替换数据来被“压缩”。在一些情况下,这样的替换不会加速事务,但可能仍然有好处,诸如对数据“采取措施使其发展”,这样接收端具有能够被后面用于重构参考那些段的发送数据的段数据。参考图 2 将更加清楚的描述这样的概念。

[0065] 如在图 2 和其它图中详细所示,事务请求和响应经由 TA' s 发送,而不是直接从客户到服务器。当然,在一些结构中,CTA 和客户和 / 或 STA 和服务器紧密结合,这样不要明确重新选择路由。但是,假设数据被发送是有用处的,至少因为它阐明来自客户的业务可以经由 CTA 发送,来自服务器的业务可以经由 STA 发送,但业务还可以旁路 TA' s。因为 TA' s 可以在传输网络层操作,他们可以作为工作单元在事务上操作。

[0066] 容易发送被加速业务的一个结构是通过连接代理的使用。因此,CTA 作为对客户和其进入一个事务的服务器的连接代理服务,STA 作为对服务器对其响应的客户的连接代理服务。可以理解的是,TA 系统可以用均衡的 TA' S 实现,例如在 CTA 和 STA 被非常相似地设置,可能除非 CTA 被建立为了期望遇到来自客户但不是来自 STA 的新事务,STA 被建立为了不期望遇到来自服务器但期望它们来自 CTA 的新事务。

[0067] 图 2 是系统 10 部分的方框图,更加详细地显示 CTA20、STA22 和它们互相连接。当仅仅显示一个客户和一个服务器时,可以理解的是,即使没有显示,图 1 的各种组件也可能存在。例如,CTA20 可以处理来自多于一个客户的事务,STA22 可以处理和多于一个服务器的事务。如图 2 所图示说明的,客户 12 耦合到 CTA20 的客户代理 30。在这个例子中,当使用到达和来自客户的复用和解复用业务的其它形式时,客户代理用于接收来自一个或更多客户对 CTA20 的数据,并发送对 CTA20 的数据到一个或更多客户。图 2 中所示 CTA20 的其它组件包括事务变换器 (TT) 32、反相事务变换器 (TT⁻¹) 34、持久性段存储器 (PSS) 36 和参考解析器 (RR) 38。服务器 14 耦合到 STA22 的服务器代理 40,其显示包括类似与 CTA20 那些的组件,诸如事务变换器 (TT) 42、反相事务变换器 (TT⁻¹) 44、持久性段存储器 (PSS) 46 和参考解析器 (RR) 48。

[0068] 客户 12 耦合到与 TT32 和 TT⁻¹34 耦合的客户代理 30。TT32 耦合到 PSS36 和 CTA20 与 STA22 之间的网络。TT⁻¹34 耦合到 PSS36、客户代理 30、RR38 和 CTA20 与 STA22 之间的网

络。如所示,RR38 还耦合到 PSS36 和 CTA20 与 STA22 之间的网络。

[0069] 在图的另一侧,服务器 14 耦合到与 TT42 和 TT⁻¹44 耦合的服务器代理 40。TT42 耦合到 PSS46 和 STA22 与 CTA20 之间的网络。TT⁻¹44 耦合到 PSS46、服务器代理 40、RR48 和 STA22 与 CTA20 之间的网络。如所示,RR48 还耦合到 PSS46 和 STA22 与 CTA20 之间的网络。

[0070] 可以理解的是,CTA20 的一些或全部组件和 / 或 STA22 可以集成到 CTA20 或 STA22 中,这样不需要组件间外在连接,但仍然存在逻辑耦合。例如,CTA20 可能整个地实现为具有数据存储器、程序存储器的单个程序,和具有包含用于实现客户代理、TT、TT⁻¹ 和 RR 的指令的程序存储器的处理器,当这些指令由处理器实现时。在这样的实现中,数据存储器可以逻辑地区分来保持对处理器执行指令、客户代理状态、TT、TT⁻¹ 和 RR 以及 PSS 内容的的可变需要。STA22 同样如此。

[0071] PSS 可以是磁盘子系统、存储器子系统或其部分。PSS 也可以是具有磁盘支持存储器、数据库服务器、数据库等的存储器子系统。

[0072] 所显示连接的,箭头指示最共用的方向或信息流动的方向,但信息可以在附加的方向流动,在单个方向流动的信息可能包括同样以相反方向流动的数据。例如,TT32 通常在 TT⁻¹44 的方向发送信息,但诸如证实、握手等数据可以从 TT⁻¹44 到 TT32 流动。

[0073] 一些连接用在 CTA20 和 STA22 之间跨越的虚线显示 (例如 TT' s 和 TT⁻¹' s 和 RR' s 之间)。尽管它们用特殊的线表示,可以理解的是,这些线可以表示不同的网络连接,或区分在公共网络连接上的分组流动,或甚至显示的逻辑连接中共享分组。因此,虚线连接可能是包括超过一个端口数量和 / 或超过一个 IP 地址的独立连接,但它们也可能是一个分组交换连接上三个逻辑连接,诸如经由使用公共端口数量和公共 IP 地址的公共路径。

[0074] 客户和 CTA 以及服务器和 STA 之间的无点线标记为“LAN/ 直接”,来指示那些连接是可能比标记“因特网 /WAN/ 等等”的 TA' s 之间的连接更高的性能 (等待时间、带宽、可靠性等)。前面的例子包括 LANs、电缆、主板、CPU 总线等。如果 TA' s 之间的连接是更高性能的连接,系统仍然可操作,但可能看不见事务加速器的一些好处。

[0075] 操作中,CTA' s 和 STA' s 检查它们事务的净负荷,其中能与事务无关的唯一命名机制从那些净负荷中获得授权的存储 / 高速缓存字符串或数据 (“段”) 的其它序列。当从一个 TA 发送净负荷到另一个时,TA 可以使用段数据的参考替换段数据。一个指示是这个替换应该在段数据是发送者期望接收者拥有唯一命名的段数据的时候发生,或者因为它在早先的事务中出现,或通过其它处理被发送到接收者,然而其它指示或没有指示可用于确定是否用参考替换段数据。在一些情况分段和替换在不期望加速的地方将不执行,诸如在包含的数据数量小的地方。事务的分段部分可以是发送数据的任何部分,只要事务在接收端仍然可以被足够辨认以被重构。

[0076] 因为段可以是唯一命名的且名字与事务无关,所以出现在一个事务中的段可以存储在 TA' s 并用作加速其它事务。例如,在客户启动大量文件请求事务的地方,如果文件拥有共有数据,则共有数据可能作为一个段形成,且在发送第一个这样的段之后,所有对具有共有数据的文件的更多请求将拥有替换共有数据的段参考,来在发送重构文件到提出请求的客户之前由 CTA 替换。类似地,在一个 CTA 处理多于一个的客户的地方,一个客户的段可以用于其它客户。

[0077] 事务是不同于文件事务的地方,类似的加速是可能的。例如,在 CTA 耦合到邮件客

户以及 STA 耦合到邮件服务器的地方,CTA 已获得附加的内容之后,许多客户经由 CTA 请求的邮件附件可以表现为一个段,然后在客户请求附件的每个序列时间,响应 STA 将使用段参考替换附件,接收 CTA 将用存储的附件替换参考。因为附件作为与事务无关的段存储,所以可能在文件事务、附加邮件事务或其它事务中找到相同段数据,在每个情况,发送者使用段参考替换数据,接收者使用段数据替换段参考。

[0078] 在此值得注释的是,这样的方法有几个优点。不象高速缓存,在当事务被重复且高速缓冲存储器不是无效的时候,事务结果被高速缓存并重复使用的地方,段可以用于几个无关的事务,且段不需要束缚在任意的切点。因为段名字和内容可以与任何特殊的比特流或事务无关,它们可在任意数量的时间持久性存储器中存在,即使系统成分失效和重新启动,新的成分被加入混合,段存储被擦除等。

[0079] 接收者可以为包括在它持久性存储器中和 / 或为在从发送者接收参考序列的传输之前、期间或之后获得段数据。最好是在可能的时候以改善事务响应的方法获得段数据。例如,如果可以预期对段的需要,则可以在需要它之前发送段数据,这样当它被需要时,它可以更快的被获得。然而,在一些情况下,诸如在接收 TA 没有任何存储的段且在事务期间必须获得它们的全部的地方,事务加速可能不发生,因为需要发送的数据总数量没有减少。

[0080] 假设请求流过 TA' s,客户 12 将发送请求给客户代理 30,其然后发送它到 TT32,修改请求或仅仅转播它。TT32 确定如何变换请求,如所需要的在 PSS 存储段和参考(在下面更加详细的解释),并发送变换后或未修改的请求给 TT⁻¹44,其实现任何需要的反相变换(下面更加详细的解释),并发送请求给服务器代理 40,依次到服务器 14。响应经过同样的路径。

[0081] 在消息(诸如客户请求消息或服务器响应消息)已被变换的地方,反相事务变换器使用它 PSS 的内容来重构消息。在简单情况下,发送者的 TT 通过识别消息的段来变换消息,用参考替换所识别的段,并在 PSS 中存储参考一段对。在 McCanneII 中描述了针对基于内容聪明地分割数据的一些技术。通过发送参考而不是段数据,事务期间 TA' s 之间的总业务被减少,或许大多数业务被移到较小的临界时间或较小的临界路径。

[0082] 在接收 TA 在它 PSS 中有发送 TT 使用的参考-段对的地方,接收 TT⁻¹ 可以通过使用它们对应的段数据替换参考来再生发送数据。接收 TA 可以从一侧信道或来自发送 TA 的部分业务获得存储在它 PSS 的段数据。因此,从发送 TA 传送到接收 TA 的数据可能包括段参考和表示从参考到段数据映像的“联系”。当然,如果每次用参考替换段,并发送参考和联系,则没有节省许多带宽,事实上增加了带宽。然而,在发送者猜想接收者已经有联系的地方,发送者可以省略联系,导致充分地更少业务。在此值得注释的是,从这个过程获得好处不要求精确知道接收者有什么。

[0083] 在一些情况下,实现请求所需的所有数据在客户的 PSS 存在,使得如果代替地使用高速缓存机制,则用 PSS 作为高速缓冲存储器,根本没有任何消息需要发送给服务器。然而,这要求 CTA 是足够聪明以至于理解客户事务并构建来自于存在于 PSS 的数据的适当响应。通常这是个困难的问题,尤其当许多不同应用类型和客户-服务器方案通过 CTA 代理且某些客户-服务器事务不简单地服从高速缓存的时候(例如文件系统写入操作、数据库更新事务、文件删除等)。因此,最后是使用在此描述的 TA' s 并任何形式发送消息到服务器,这样使得避免高速缓存机制的矛盾。例如,如果文件服务器接收对文件的所有请求,即

使在客户存在整个文件内容,服务器可以跟踪请求,服务器可以实现诸如文件锁定协议的复杂操作,即使没有真实地文件数据需要在网络传送。

[0084] 在 TA 系统的较佳实施例中,上面的好处自动产生。例如,STA 分割每个事务净负荷并用参考替换段。针对 STA 猜想 CTA 拥有的段数据,STA 对那些段使用它知道 CTA 拥有的参考。当数据在服务器变化时,STA 创建表示变化后数据的段,并可以假设 CTA 没有那些段,而非试图修改 PSS 中存在的段。在这种情况下,STA 对表示变化后数据的新段使用新参考。在接收 TA(在这个例子中是 CTA),对旧数据的参考可以从存储在接收者 PSS 中的联系分析出来,但对于新的数据、变化后的段,参考从包括在来自发送者的数据流中的联系分析出来。然后那些联系被接收者的 TT^{-1} 存储到接收者的 PSS 中,这样它们可被那个接收者的后续事务利用。

[0085] 此外,因为参考总体上是唯一的(如下面所描述的),它们可以被网络中任何 TA 使用,不仅仅如在这个例子中描述的 STA 和 CTA 对。例如,CTA 可能和不同的 STA 通信,并使用前面 STA 分配的参考。如果将来二个 STA' s 通信,则它们立即享受已经散布到二个设备的段联系的好处。

[0086] 几个机制可用于确保在整个系统中任何给定时间每个命名的段拥有唯一名字(即没有二个具有不同数据的段被错误的分配相同的名字)。在一个方法中,每个段参考被产生作为大的随机数字,在唯一参考的数字比所有可能大小随机数字的空间低得多。这个机制比理想的更小,因为存在小的可能性二个段拥有相同的段参考,但不同的段数据,其导致接收 TA 错误地重构具有错误数据的消息。

[0087] 另一个机制是从段数据产生,使得每个段参考是段数据的一个无用信息,不同的段数据将导致不同的无用信息,除了在非常罕见的情况下。但是罕见的情况总是有问题的,只要系统中存在二个具有相同参考但不同段数据的降级段。不象随机数字的情况,每当特殊数据模式存在于数据流中,这个问题将重现。

[0088] 一个简单避免上面问题的方法是针对每个发送 TA 从唯一 ID(诸如当普遍唯一 IP 地址在整个网络使用时主机的 IP 地址、主机 MAC 地址、分配的唯一标识符或其他方法)和连续数字的组合产生段参考。在大多数实现中,唯一连续数字的最大数字被绑定,因此将最终需要被重新使用。然而,通过使用可以持续数百万年的供应且不需要特殊处理的足够标签数字,可以有效地不绑定名字空间。打标签可以压缩来未标签提供小的脚标。

[0089] 因为标签将被顺序分配,且因为对应的段将常常出现在相同的序列里,所以实际上可以实现好的标签压缩(不仅仅穿过网络而且在表示整个系统中总是采用的标签字符串的数据结构中)。附加压缩还可能在发送 TA' s 输出数据流上。例如,在接收 TA 可以标识发送 TA 且发送 TA' s 参考包括发送 TA' s 唯一 ID 的地方,当接收 TA 知道在形成参考中使用什么 ID 时,ID 不需要出现在发送数据中(尽管通常当发送 TA 参考不仅仅它自己绑定而且绑定和标签起源于其他 TA' s 时,必须告知附加信息)。这个方法的一个其他的优点是,TA 可以标识来自段参考的 ID 成分的它 PSS 中每个段的来源以用于统计分析、诊断和类似的。

[0090] 在系统所期望的生命期期间打算重新使用标签的系统中,系统最好包括一个机制来“终止”参考绑定,这个终止普及到网络中所有的 TA' s。一个方法是时间戳每个段,使得它拥有在使用段的系统中每个成分可容易地推断的固定生命期。如果以粗糙的方式分配

时间戳给标签（例如时间戳一天变化一次），则标签压缩消除与分配和告知时间戳相关的大多数协议报头。从而 TA 可以猜想什么时候它安全地重新使用特殊组标签。

[0091] 然而管理段名字空间的另一个替换是拥有唯一参考的中心分配。在这样的情况下，发送 TA 从保证参考是唯一的源头中请求一个参考或一批参考。此外，每个分配可以被分配最大的生存时间，使得参考或参考块的分配可以绝对地被重新使用。

[0092] 可能发生的是，发送 TA 在某一绑定在接收 TA 不存在时候，假设它存在。在接收 TA 拥有 PSS 溢出、信息损坏、功率损耗等或接收 TA 故意移出绑定的情况，这个可能发生。在这样的情况下，接收 TA 可以无需终止或不得不报告事务故障而获得段数据。这允许系统适度地处理由于磁盘满了、磁盘故障、网络故障、系统崩溃等遗漏的数据。如果发送 TA 假设接收 TA 拥有绑定，则发送 TA 将使用那个绑定的参考发送消息，但将不包括绑定的段数据。当接收 TT^{-1} 设法分析参考时，它将失败。在那些情况下，接收 TT^{-1} 发送结构请求给它的 RR，其然后提出一个请求给发送者的 RR。当接收到所需数据时，可能由于可用数据的事件触发信号， TT^{-1} 可以正好阻塞并重新启动；这个处理对 TT^{-1} 是透明的（除了在获得响应中的延迟）。一旦接收者接收到段数据，接收者的 RR 可以提供数据给接收者的 TT^{-1} 或仅仅放在接收者的 PSS，在此它可以被接收者的 TT^{-1} 访问。当发送者的 TT 适当地增加绑定到它的 PSS 时，它保留那些保证最小时间数量的绑定；同样当它用参考替换段数据时，当接收者的 RR 为那个段数据提出发送者的 RR 请求时可以保证它，它将存在于发送者的 PSS，在发送者的段的所提供和保证的“生命期”比接收者可能要求提出段请求的最大时间数量更大。

[0093] 图 3 包括简单 PSS 的绑定表数据构造的图示说明。如所示，绑定表存储大量绑定，诸如 (R_1, S_1) 、 (R_2, S_2) 等，在此 R_i 是第 i 绑定的参考标签， S_i 是第 i 绑定的参考数据。每个绑定的时间戳可能用于老化绑定。绑定记录可能包括图 3 中没有显示的其他字段，诸如那些在表 1 列出的和 / 或类似或附加的字段，可能除了其他表、数据结构、物体和 / 或代码。

[0094] 表 1

[0095] - 访问的时间数量

[0096] - 最后的访问时间

[0097] - 最后的修改时间

[0098] - 生命期

[0099] - 编码方法标识符（例如未编码的原始数据、运行长度编码、MD5 编码、加密）

[0100] - 指针

[0101] - 错误校正数据（如果没有段数据打断）

[0102] - 创建绑定的发送者的标识（绑定“所有者”）

[0103] - 创建时间（对超时段有用，诸如通过使用生命期字段）

[0104] - 其他字段

[0105] 一些附加数据结构可能包括参考的索引、其他字段的索引、段的指示等，用于搜寻或另外处理 PSS 的内容。可以用许多能用于编码处理的方法来索引段，但一个实施例建立段的索引，众所周知的在包含段的整个数据计算的无用信息被用作键。如果使用编码方法标识符，段数据可以被编码来错误校正、加密等。

[0106] 对于一些段数据来说，可能适合于压缩段数据来减少 PSS 的存储需求和所需的经过周围绑定的传输花费。在一些情况下，编码方法是固定的已知的，所以不需要编码方法标

识符。例如,发送 TA 可以发送段的逐字功能,例如前向错误校正已编码的段数据块、段加密、段签名或类似的,而非传送和高速缓存逐字段(即逐字的表示事务数据的子字符串或子序列的段)。这将允许接收 TA 解码来自已适当编码的数据的共用存储池的不同段,如果编码数据的某些段被丢失或破坏,则仍然可以重构原始段,从而增加错误校正到链路,无需在客户或服务器变化。

[0107] 为了跟踪哪个接收者知道哪个段,在 PSS 可能存在其他字段。在一些实现中,发送者仅仅分割数据并创建与接收者可能按结果做无关的参考,但在其他实现中,发送者保留不能确定接收者是否可能拥有特殊绑定的信息,诸如通过跟踪哪个接收者以前接收哪个段。通过记录哪个接收者在 Bloom 滤波器中拥有哪个段可以优化对这样信息的存储(即通过目的文件的无用信息索引的比特向量到给出罕见错误正极但从来不给错误负极的向量中)。

[0108] 一些实现可能使用探索法,这样服务器代理包括段绑定,仅仅当它创建新条目和其他需要段的客户代理将不得不请求它的时候,因为仅仅第一个客户代理将自动获得针对新段的绑定。

[0109] TA 可能包括用于 PSS 清除的例行程序,诸如当客户关闭文件时,删除涉及特殊服务器上特殊文件的客户端 PSS 中所有段的探索法。服务器端 PSS 也可能删除对应的段,或推迟对那些段的清除,直到所有的客户关闭文件。其他清除可能包括删除已经超过它们生命期或已经一段时间没有使用的段条目。其他探索法可能指示特殊段绑定被使用和放弃的时间。

[0110] PSS 的设置有许多好处,通过阅读这个说明,其中一些将变得明显。因为在变化切点可以发生分割,段与事务无关,段可能在任意长度时间的 PSS 内生存,并用于完全不涉及其中创建和存储段事务的事务。因为段参考对于唯一的段数据来说是唯一的,接收者可以总是正确识别对段参考的段数据(如果接收者有段的话)。这仅仅比高速缓存结果更好。它还比使用本地化信号统计压缩更好,诸如建立适当的码本和类似的。段名字和内容与任何特殊的比特流无关,即使系统成分崩溃和重新启动,新成分被增加到混合里,擦去持久性段存储等等。应该可以理解的是,“持久性”用于描述 PSS 不意味着段被永久不变地存储,因此能够从来不被清除;仅仅是段中至少一些坚持至少超过一个事务。

[0111] 图 4 图示说明编码器 140 和 PSS142。用于 TA 的 TT 可能正好是编码器 140,但 TT 还可能包括其他功能或元件。如所示,编码器 140 有一个用于编码数据的输入,并控制用于编码控制参数和带宽外面关于输入数据的信息的输入。编码器 140 被显示有用于编码数据和 PSS142 中存储的段绑定的输出。在操作中,编码器 140 将处理输入数据,标识数据的段,用参考替换段的数据,已绑定的方式提供段数据和段参考给 PSS142 并输出编码后数据。如图 4 所示,结果编码后数据可能包括参考、绑定和剩余数据(诸如不能有效用参考表示的数据)。在此,一块剩余数据还称作“未参考段”。在一些实施例,被分段但没有参考的数据和没有分段的数据之间存在差异。在前者中,有一个定义的段开始和结束,但段内容没有用段参考替代,然而在后者中,同样可能没有段开始或结束。为了便于下面描述,忽略此差异。

[0112] 编码器 140 的另一个输出是对 PSS142 用在解码引入数据中的段绑定(或者为了提供给其他要求的 TA's)。控制到编码器 140 的输入可能包括目标段大小,带宽外信息可能包括指示这样的事情作为段默认生命期、关于数据源的信息等的参数。目标段大小是控

制通过分段处理产生的段的平均大小的参数。通常,段长度随着大小的某个特定分布变化,目标段大小控制通过分段处理产生的平均大小。当段大小能被固定时,最好允许段大小变化,使得段配合比如果系统处理的数据被分段成任意固定的段更经常。

[0113] TT 把它创建的绑定放到它自己的 PSS 中用于解码,但同样所以绑定的“所有者”(即创建绑定的 TA)可以保持跟踪它,提供它给其他并同样当编码后续数据时查阅它(所以段参考可能在段数据重复的地方被重新使用)。

[0114] 绑定的所有者的 TT^{-1} 常常重新使用那些绑定,诸如当段数据的序列走圆形行程时,即从 STA 流动到 CTA 并返回,或反之亦然。例如,用户编辑文件情况这可能发生。用户的文件客户将请求文件数据,服务器将服务文件,且当用户编辑文件时,文件数据的绑定将在 CTA' s 的 PSS 和 STA' s 的 PSS 中存在。如果用户写回文件数据,则没有变化的部分可以完全用在文件数据第一次发送到用户的客户时候创建的参考标签来表示。在这个情况下,CTA 简单参考相同 STA 创建的旧的绑定,而非在发送数据返回 STA 的时候创建新绑定。其他例子包括电子邮件,客户可以请求电子邮件(经由类似 IMAP 或 POP 的一个协议),然后在网络上转播它返回(经由类似 SMTP 的另一个协议),在这个情况,STA' s TT^{-1} 可以使用在电子邮件第一次发送到客户的时候由 STA' s 创建的绑定,假设 SMTP 事务和 IMAP 或 POP 事务流过 STA/CTA 对。另一个例子是用户经由 CIFS 从 Web 站点(通过 HTTP)拷贝信息到文件系统,假设 HTTP 事务和 CIFS 事务流过 STA/CTA 对。

[0115] 由于 PSS 的这个特性,客户和服务器可以使用非常小的带宽且无需改变客户或服务器,来回有效地发送大块数据。这个在大文件被移动且仅仅稍微有点改变的情况特别有用,诸如二个或更多用户在一个大 CAD 文件上合作的情况。使用在此显示的系统,网络性能可以足够导致用户放弃其他工作区网络瓶颈,诸如远程访问、存储本地文件拷贝、推出文件的只读拷贝等。

[0116] 如果根据内容分割输入数据,则相同比特序列可能导致相同的段,而不管比特序列在什么地方发生。这有个优点是重复的比特序列有效的被注意和参考。然而,在有强制需求或改善性能的情况,外部因素可能开始活动。例如,如果有抵消好处的话,则关于可以使用的事务的一些参数可能导致多余一个的为一个比特序列创建的段。在一种方法中,外部因素是什么段存在于 PSS 中,以及基于在 PSS 中已经存在什么段来决定段分界。虽然这不象前面描述的更基础的方法可以升级,但是它可能导致段更多的重新使用,从而拥有一些好处。

[0117] 这可以通过一个例子来图示说明。如果通常地单向切割净负荷,但不同组的切点将更近似匹配已经在 PSS 中的段,则将导致更大的压缩。然而,为了能够保持增益,发送 TA 将不得有一些想法接收 TA 可能具有哪个段,使得如果发送者知道大多数发送者的 PSS 段不存在与接收者的 PSS,则发送 TA 不基于发送者的 PSS 进行它的切割。

[0118] 图 5 图示说明了解码器 150 和 PSS152。用于 TA 的 TT^{-1} 可能仅仅是解码器 150,但 TT^{-1} 也可能包括其他功能或元件。解码器 150 接收编码后的数据,其可能已经通过图 14 所示解码器 140 输出。如上所述,编码后数据可能包括参考、绑定和剩余数据。当解码器 150 遇到它接收数据中的绑定时,它可以使用绑定中的段数据来重构原始数据,它还能够存储绑定到它的 PSS 中。当解码器 150 遇到没有绑定的参考时,它能够使用参考从 PSS152 获得段数据来重构段。如果在 PSS152 中没有找到段参考,则解码器 150 可以发送对段数据的请

求。

[0119] 图 6 是编码处理的图示说明,其中输入数据被分割和用参考表示成为段数据。如所示,原始输入数据被存入缓冲器 160(尽管如果需要的话可以不要缓冲)。然后分割原始数据成段。几个技术适用于确定什么地方定义为分离每个段和它邻近邻居的“切割线”。一些分割的方法如 McCanne II 中所描述的。可能使用的其他方法是以规则的时间间隔,或诸如标记的结束线的在原始输入数据中发现的数据固定序列的一些关系,来放置切割线的简单方法,尽管这样的方法可能不会产生最好的执行分割机制。

[0120] 然而在图 6 的例子中确定切割线,缓冲器 160 中原始输入数据被分割成段 S_A , S_B , S_C , S_D , S_E , 和 S_F 。在这个例子中,第一个五个段用参考替换,参考是 R_{15} , R_{16} , R_{17} , R_3 , 和 R_8 。在此值得注释的是,参考不必是顺序的,这个例子图示说明了一些参考(例如 R_3 和 R_8)可能是已经遇到过的段数据,在这种情况下不使用新的段,但参考先于段存在。这儿还图示说明了段(例如 S_F)不需要用参考替换。

[0121] 原始输入数据可以完全用从原始输入数据产生的输出数据和绑定来表示。提供绑定给用于产生绑定的 TA 的 PSS,和其他以及一些或所有绑定可以作为输出数据的一部分被发送。在这个例子中,新绑定是 (R_{15}, S_A) , (R_{16}, S_B) 和 (R_{17}, S_C) 。在这个例子中,因为用于段 S_D 和 S_E 的段数据已被知道并用参考 R_3 和 R_8 存储,所以不需要绑定 (R_3, S_D) 和 (R_8, S_E) 。

[0122] 图 7 是图示说明用于可能由图 4 的编码器输出和由图 5 的解码器解码的解码数据的处理过程的流程图。处理步骤标志为“S1”,“S2”等,除非另外标识,通常步骤按顺序进展。在第一个步骤(S1)中,参考后的数据(例如带参考编码后的数据)被接收并分析标记。如果检查标记(S2)且它不是参考,则它必须是未参考段,所以直接输出标记(S3)。然而,如果标记是参考,则解码器检查(S4)在支持解码器的 PSS 中是否存在参考。如果是 YES,则解码器从 PSS 参考段(S5)。如果是 NO,则解码器发送一个解决请求(S6)给支持解码器的参考分析器,并接收从参考分析器返回的已解决已参考的段(S7)。在参考标签编码段数据的源头的情况,标签可以被参考分析器用于帮助找到参考后的段。

[0123] 一旦解码器访问参考后段的段数据(进展到步骤 S3 或步骤 S7),它输出段数据(S8)。然后解码器检查附加的标记(S9)。如果有更多的标记,则处理过程携带下一个标记在步骤 S2 重复,除非处理完成了。

[0124] 上面的描述仅仅表示使用 PSS 编码和解码段绑定和参考的一个特殊实施例。其他方法可能包括标签和数据直接关系更完善的表示,诸如那些在 McCanne II 中所显示的。

[0125] 图 8 是联网系统的方框图,其中事务加速被实现并使用预发式段分配器(“PSD”)。如所示,PSD210 包括 PSD 控制器 212、它自己的 PSS214 和其他用于 PSD 可变的存储器 216。在一些实现中,使用多个 PSD's,尽管仅仅在图中显示了一个。

[0126] 通过 PSD210 的操作,段更加可能在需要它们的时候存在,从而需要更少的段解决请求。在需要将段从 PSS 移动到 PSS 的情况,PSD210 可以在对这个段的实际需要之前触发这个处理过程,使得事务返回更加快速,当接收到净负荷时,接收 TA 不必为接收 TA 阻塞来发布对段的请求给发送 TA。PSD210 可以自己分配或仅仅指示段的所有者(或其他占有者)传送它们。在一些状况下,PSD210 可能保留他们自己的 PSS214,但在一些实现中,PSD 仅仅指示 PSS's 之间绑定的流动且不保留它自己的 PSS。

[0127] PSD210 可能监控来自 CTA's 20 和 STA's 22 事务流动,并确定可能需要哪个段和

什么地方需要。当 PSD210 确定需要一个段时,它可以发送消息给发送 TA,诸如 STA 服务文件系统或电子邮件系统。消息将指示发送 TA 执行分割,存储绑定在它自己的 PSS,且甚至传送绑定给其他 PSS' s,使得当发送 TA 接收到将导致发送 TA 发送净负荷的消息时,进行分割。当很好的做完时,接收 TA 将获得在接收 TA 接收带有参考的净负荷时需要的绑定,在带宽不是临界的时候可以发送那些绑定。更加典型地,发送 TA 是 STA,但 PSD 还可能指示 CTA' s “预先存入”绑定到系统中。

[0128] 在一些情况下,服务器代理增加到服务器中来识别用于预存的选择物。例如,诸如 Microsoft Exchange™ 服务器的邮件服务器可能耦合到网络并用 STA 和关联服务器代理。服务器代理检测电子邮件和附件到达的时间,并基于过去观察或操作员的政策,预先存入具有相应段数据的特殊 CTA。这可能通过跟踪哪个客户读取来自哪个位置的它们的电子邮件来完成,或者通过静态配置或者最好是测量。然后,当远程用户读取电子邮件时,大多数电子邮件的数据已经在用户的远程站点了,但事务仍然返回交换邮件服务器来确保协议的正确性。

[0129] 除了预先触发段产生,PSD210 还可能帮助“预先填充”各种具有已经存在使得那些 TA' s 拥有在接收参考时准备好的段数据的绑定的 TA PSS' s。在一个实现中,PSD210 在传播模式上操作,如利用 USENET 消息条目所做的那样,通知新绑定给 PSD210,PSD210 然后传播来自通知 TA 的新绑定给所有或一些其他的 TA' s,其可能依次传播绑定。可替换地,或者除了 PSD 触发的预先填充,发送 TA 可能预期哪个段需要被发送到接收 TA,并发送它们在任何时间之前或“带宽之外”,这样接收 TA 不需要发布附加的请求来解决未知的段。

[0130] 在不加选择的传播可能导致网络超载或提高相对于仅仅发送完成的带宽利用率的地方,可以使用每个事务的原始数据更加改进的方法。在更加有效方法的例子中,PSD 使用启发式来确定哪个 TA' s 可能需要哪个段。在另一个方法中,服务器包括提供具有在允许 PSD 确定哪个 CTA' s 将需要哪个来自“代理”服务器的段的高层的信息的 PSD 的服务器代理。在一些实施例中,使用上面方法的组合。

[0131] 具有服务器代理的 PSD 的另一个实施例包括一种类型的文件系统监控。在此,服务器代理监控文件系统活动,且每当新的数据写入文件系统,代理命令 PSD 复制适当的段绑定给一个或更多 CTA' s。用户或定义的操作者政策可以指示整个文件系统的数据是否被复制或仅仅复制配置部分。此外,这些政策可以通过访问模式的测量增加,使得来自文件系统的最频繁访问的部分的段数据被复制(且这些测量可以在预先 CTA 基础上实现)。结果,每个这样的 CTA 有效地包括所有(或部分)文件系统数据的镜像。然后,当客户通过网络文件-系统协议(例如 CIFS 或 NFS)和 CTA 联系时,事务一路上流动返回到原始文件服务器,然而这些事务被完全压缩到参考的纯字符串中。这个方法确保原始文件系统语义被保存,好像所有客户共享单个文件服务器,但是客户-服务器通信行为的性能表现为好像所有数据在本地预先客户文件服务器。

[0132] 虽然上面略述的段机制可以充分地减少客户-服务器通信的带宽要求,但是事务仍然受广域网固有通信等待时间的支配。这些等待时间瓶颈可以不利地影响性能,且可以使用类似文件前面读取和后面写入的补充技术来处理。然而由于数据的压缩和分级,前面读取和后面写入技术变得极端有效,因为所有数据已经在 CTA 存在,所以在网络上他们导致非常小的开销。

[0133] 所有这些方法可以用机制补充来对各种种类的 CTA/STA 通信采用带宽政策。例如,某些带宽限制可强加到 PSD 来限制分级算法的积极性。在另一个例子中,带宽政策可应用于不同级别的分级数据(例如文件系统段复制可以具有高于电子邮件附件段复制的优先权)。

[0134] 图 9 是根据本发明实施例的联网点对点系统的方框图。如所示,各种点 180 经由点事务加速器 (PTA's) 182 彼此互相作用。点 180 可能直接相互作用,尽管没有显示这样的连接。运行中,一个点 180 可能经由每个点的 PTA182 和网络 184,请求来自另一个点的数据。如所示,每个 PTA182 可能包括点代理 190、TT192、TT⁻¹194、PSS196 和 RR198。在点对点系统中,点本质上功能如同用于一些事务的客户和用于另一些事务的服务器,这样事务加速机制将以类似的方式运行。

[0135] 图 10 是联网系统的方框图,其中事务加速被实现且客户端事务加速器集成到客户中,与成为独立实体相反。如所示,客户系统 302 经由网络 304 直接耦合到服务器 306,经由服务器加速器 STA310 耦合到服务器 308。所显示的客户系统 302 包括通信处理 320、直接网络 I/O 处理 322、CTA 处理 324 和包括持久性段存储器 328 的存储器 326。通信处理 320 与直接网络 I/O 处理 322、CTA 处理 324 和存储器 326 耦合。CTA 处理 324 耦合到 PSS328。

[0136] 运行中,通信处理 320 典型地在应用层执行功能,要求和客户系统 302 外面的服务器相互作用。例如,通信处理可能包括 web 浏览器、电子邮件客户、Java 程序和交互式网络程序、聊天程序、FTP 程序等等。在通信处理直接和服务器相互作用的地方,通信处理将和直接网络 I/O 处理 322 相互作用,来与服务器相互作用,但在事务要被加速的地方,通信处理将和 CTA324 相互作用。在一些实施例中,通信处理 320 和 CTA324 可能是单个应用程序的组成成分,而在另一些实施例中它们可能是分离的应用处理。CTA 处理 324 可以使用存储器 326 的一部分作为 PSS 来加速事务,非常类似前面所描述的各种孤立 CTA's 所作的。在一些变化中,PSS328 是来自存储器 326 的独立存储器,其用于客户系统 302 的其他处理,诸如通信处理 320 的需要。

[0137] 直接网络 I/O 处理 322 通过和网络 304 上服务器的相互作用,满足通信处理 320 的网络 I/O 需要。在一些情况下,直接网络 I/O 处理 322 和与 CTA324 相同的服务器相互作用,通过到服务器 308 的虚线来描述。客户系统 302 可能包括没有显示的其他处理,包括涉及事务加速的处理。例如,通信处理 320 可能依靠确定何时直接发送事务给服务器以及何时试图加速它的分离处理。

[0138] 图 11 是联网系统的方框图,其中事务加速被实现,服务器端事务加速器集成到服务器里。图显示了服务器系统 352、网络 354、客户 356、客户 358 和客户事务加速器 (CTA) 360。所显示的服务器系统 352 包括通信处理 370、直接网络 I/O 处理 372、STA 处理 374 和包括持久性段存储器 378 的存储器 376。通信处理 370 与直接网络 I/O 处理 372、STA 处理 374 和存储器 376 耦合。STA 处理 374 耦合到 PSS378。如通过从客户 356 到直接网络 I/O 处理 372 的线所图示说明的,客户 356 直接耦合到服务器系统 352,其处理不通过 STA 处理 374 的事务。客户 358 经由 CTA360 和 STA 处理 374 耦合到服务器系统 352,但也可能用于其他事务直接连接到直接网络 I/O 处理 372。

[0139] 运行中,通信处理 370 执行功能,诸如响应来自客户的请求的服务器处理。服务器系统 352 和客户直接相互作用,事务将经由直接网络 I/O 处理 372 在通信处理和客户之间

流动。在服务器系统 352 和客户经由 TA' s 相互作用的地方,事务将经由 STA 处理 374 在通信处理和客户之间流动。STA 处理 374 可以使用存储器 376 的一部分作为 PSS 来加速事务,非常类似前面所描述的各种孤立 STA' s 所作的。在一些变化中,PSS378 是来自存储器 376 的独立存储器,其用于服务器系统 352 的其他处理,诸如通信处理 370 的需要。

[0140] 直接网络 I/O 处理 372 通过和网络 354 上服务器的相互作用,满足通信处理 352 的网络 I/O 需要。在一些情况下,直接网络 I/O 处理 372 和与 STA374 相同的服务器相互作用,通过到客户 358 的虚线来描述。服务器系统 352 可能包括没有显示的其他处理,包括涉及事务加速的处理。例如,通信处理 370 可能依靠确定何时直接发送事务给服务器以及何时试图加速它的分离处理。

[0141] 应该可以理解的是,图 10 和 11 的元件可能被组合,这样具有内部 CTA' s 的客户系统可以和具有内部 STA' s 的服务器系统通信。还应该可以理解的是,使用箭头线的信号的地方,还可能存在双向信息或数据流动。

[0142] 在客户和 / 或服务器中嵌入 TA 的一个缺点是,每个设备以它们自己的 PSS 告终,减少了高速缓存代表在给定位置的大量客户 (或服务器) 的相同段数据的好处。然而这个问题可以在另一个实施例中克服,其允许 PSS 逻辑地跨过多个 TA' s,最好位于公共 LAN 段 (内部高速链路互连的公共网络区域,例如一个楼或近似接近的多个楼里面多层互连的高速校园网)。在这种情况下,逻辑共享的 PSS 可以是隶属于网络的另一个设备或它可以是嵌入到每个 CTA 中的几个 PSS' s,这样通过协同操作协议 (例如 IP 多点传送) 这些 PSS' s 作为单个的逻辑的实体运转。

[0143] 图 12 是联网系统的方框图,其中事务加速被实现,且在大量事务加速器之间共享 PSS。如所示,客户耦合到用于事务加速的本地 CTA402。替换保留的分离 PSS,本地 CTA' s402 耦合到共享的 PSS404。最好地是,本地 CTA' s 和共享的 PSS 之间的连接是相对于经由在客户和服务器之间存在的网络 405 的连接,具有更高性能的连接。还可能存在共享参考分析器 406,并耦合到共享的 PSS404 和共享 PSS 的本地 CTA' s。

[0144] 当每个本地 CTA402 是用请求消息启动一个事务或接收一个响应消息时,本地 CTA402 将使用用于存储和段数据检索的共享 PSS404。由于作为一个本地 CTA 的事务的结果被存储的段可以用于另一个本地 CTA 的事务,所以在系统上使用对于每个本地 CTA402 分离的 PSS' s 有一个优点。例如,如果本地 CTA402 (1) 最近处理包括从服务器 S 获得数据的客户事务,则服务器 S 为那个事务创建的段将可能在共享 PSS404 中存在。如果本地 CTA402 (2) 然后处理在服务器 S 指示的不同客户 (或相同配置的相同客户,诸如循环法本地 CTA 共享机制) 的事务,本地 CTA402 (2) 将发送请求给用于服务器 S 的 STA。如果用于第二个事务的段匹配本地 CTA402 (1) 更早事务的一些,则不管它们实际上是表示相同的请求还是结果净负荷拥有共有的相同数据的不相关请求,本地 CTA402 (2) 将接收对那些段的参考,而不是段数据本身。

[0145] 当本地 CTA 接收不能从共享 PSS404 解决的段参考时,本地 CTA 可以发送用于解决的请求给共享参考分析器 406。在一些实施例中,每个本地 CTA 拥有它自己的共享参考分析器,其传送它的参考解决给共享 PSS404 和本地 CTA 的其他成分。其他实施例可能采用所有客户使用的单个共享参考分析器。

[0146] 当在图 12 中所描述的共享 PSS 作为在客户端的时候,可以在服务器端配置类似的

设置,拥有客户端的共享或单独的 PSS' s。同样,具有共享 PSS' s 的 TA' s 可能在 TA' s 具有单独 PSS' s 的相同网络上存在。尽管图 12 显示共享 PSS404 是不同于本地 CTA' s,可以是共享 PSS 包含在本地 CTA' s 的一个中,尽管它在共享那个 PSS 的其他 CTA' s 的外部。

[0147] 可能使用本地的网络多点传送通信在它服务的本地 CTA' s 之间连接 PSS。在这种方法中,每个事务加速器预定众所周知的本地范围的多点传送群。通过使用本地的范围,系统可以保证仅仅用本地高速网络连接的事务加速器通过这个机制彼此协调。每个主机可以产生发送给这个群的定期会话消息分组(或未交换会话分组的另一个配置群),允许对预定那个群的其他事务加速器的环程时间估计的计算。众所周知的技术可以用于这个处理过程,诸如在 1997 年 12 月卷 5,6 号 pp. 784-803 的联网上 IEEE/ACM 事务中 Floyd, S., et al.,“对轻-重会话和应用层框架的可靠多点传送架构”中所显示的那些。会话协议允许群中所有的成员获悉每个其他的存在,还能够推断来自成员总和的群的大小。

[0148] 使用这个多点传送结构,用于高速缓存段数据的系统可以用许多方法加强。在一个方法中,每当事务加速器接收到新的段绑定,它可以多点传送段绑定给本地范围群内所有其他事务加速器。这个可以减轻上面用每个具有分离 PSS 的客户来略述的问题,因为在本地事务加速器组中的每个 PSS 将被彼此复制,从而任何给定的数据段将在 WAN 连接上仅仅发送一次。

[0149] 为了确保在网络多点传送连接上传输的可靠性,许多针对可靠多点传送的众所周知的机制被采用作 Floyd et al. 可靠多点传送协议中引用的证件中。假设这个多点传送通信在同类的高速本地区域或校园区域网络中处理,则拥塞控制和 WAN 多点传送的困难问题都完全避免了。

[0150] 图 13 是显示图 12 系统的多点传送实现的方框图,其中多点传送通信用于更新或读取共享的 PSS。如图 12 所示的设置,本地 CTA' s412 连到客户和网络 405,并和其他本地 CTA' s 共享所共享的 PSS414。共享 RR416 配置在相同多点传送群 417 上,作为共享 PSS414 的每个实例(标识为 414(1),414(2),...)。逻辑地,如果本地 CTA' s 处理所需的 I/O 读取和写入共享 PSS,则可能称为多点传送群包含共享 RR416 和本地 CTA' s。用图中线 418 来图示说明多点传送业务。

[0151] 在另一个方法中,PSS 不是如上面说描述的预先激活复制,但更合适的事务加速器可以发布本地请求给分析未知的段。换句话说,当事务加速器接收一个不在它 PSS 里面的数据的参考,它在本地范围多点传送群上发送解决请求消息。因此所有的其他本地事务加速器将接收请求消息,除非错误发生。在它 PSS 中有请求的数据的接收者然后可以用数据响应。通过使用众所周知的开槽和阻尼技术(如 Floyd et al. 中),当发生小延迟的时候,仅仅一个响应消息典型地将在网络上发送。

[0152] 如果请求者没有接收到响应(从会话消息环程时间计算的一些讨论延迟之后),请求者假设数据不存在于本地环境,并在 WAN 上发送解决请求给原始的在询问中产生数据参考的事务加速器。在此值得注释的是,因为本地环程时间与 WAN 环程时间(典型地好几毫秒或更多)相比,是相当的小(典型地小于 1 毫秒),所以由这个启动检查导致的外部延迟被忽略(即典型地在少许百分比下),而由于较高的本地网络性能的好处是真实的。

[0153] 但是在另一个方法中,上面所描述的二个方法之间的混合消除了关于本地解决请

求的延迟。在这个混合方法中,每当事务加速器接收到新的段绑定,作为多点传送的整个段的替代,它简单地多点传送段的名称。这样,所有的本地事务加速器获悉段呈现的是什么,无需必须拥有所有段数据的拷贝。然后,当接收到用于不在 PSS 中但它的名字被记录作为本地已知的段的参考时,事务加速器可以发送对数据的本地请求,且本地请求可以之间到发送出新段绑定的事务加速器,如果可以标识发送者的话。另外,加速器可以假设数据不是本地存在且立即穿过 WAN 发送请求。即使当段被推断时本地存在的,它也可能已经被从所有其他本地加速器的 PSS' s 驱逐出去。在这种情况下,请求加速器将仍然终止连接并回复穿过 WAN 发送它的解决请求。

[0154] 但是在另一个方法中,穿过本地加速器群的 PSS' s 存储的段不需要完全复制。在此,每个加速器对使用合作的高速缓冲技术的段高速缓冲存储器的一部分负责。如上面所描述的,当由另一个加速器管理的段数据的参考需要被分析时,请求可以在多点传送群上被直接或间接的发送到那个设备。一旦数据已经重新集合并传达给客户(或服务器),它可以被放弃且不需要进入本地 PSS(因为段数据被其他事务加速器积极地管理)。

[0155] 图 14 显示了大量具有集成 CTA' s 的客户 502。客户 502 耦合到 LAN504,其依次经由 LAN-WAN 链路 508 耦合客户 502 到 WAN506。不是在 LAN504 上所有的客户需要包括 CTA512,但至少显示二个客户包括集成的 CTA' s512。显示的每个 CTA 包括 PSS514 和 RR516。通过这个实现,CTA 所有的功能可以作为运行在客户上的软件来实现。

[0156] 客户的 CTA512 处理被运行在那个客户上的客户应用程序 510 所请求的事务加速。例如,运行在客户 502(2) 上的应用程序启动和服务器的被加速的事务,将连接到 CTA512(2)。然后在 LAN504 和 WAN506 上 CTA512(2) 将打开和相应 STA 的连接,如上面所说明的。当 CTA512(2) 接收包括已经被加速的净负荷的响应消息时,CTA512(2) 将使用 PSS514(2) 的内容来服从加速净负荷中参考的标签。

[0157] 为了实现可能已经在 LAN504 上服务器和其他客户之间的段的好处,PSS' s514 可以协作 PSS' s。通过协作,每个 CTA 可以使用来自它们自己 PSS 和在 LAN504 上其他 CTA' s 的 PSS' s 的段绑定。然后,如果段绑定不能在本地找到,则 CTA' sRR 可以在 WAN 上发送绑定的请求给 STA。

[0158] 在一些情况下,当 RR 接收新的绑定时(或它的 CTA 创建的一个),它分配新的绑定给 LAN 上其他 RR' s 中的每个,这样每个客户的 PSS 填充了在 LAN 上创建的可用绑定,且当 CTA 服从净负荷时,CTA 已经拥有在 LAN 上可用的每个绑定的拷贝。在此这称为“规范协作”。

[0159] 在其他情况,没有提早分配绑定,但在请求上发送绑定。因此,当 RR 需要它所没有的绑定时,它为绑定提出 LAN 上其他 RR' s 的请求。在此这称为“要求协作”。

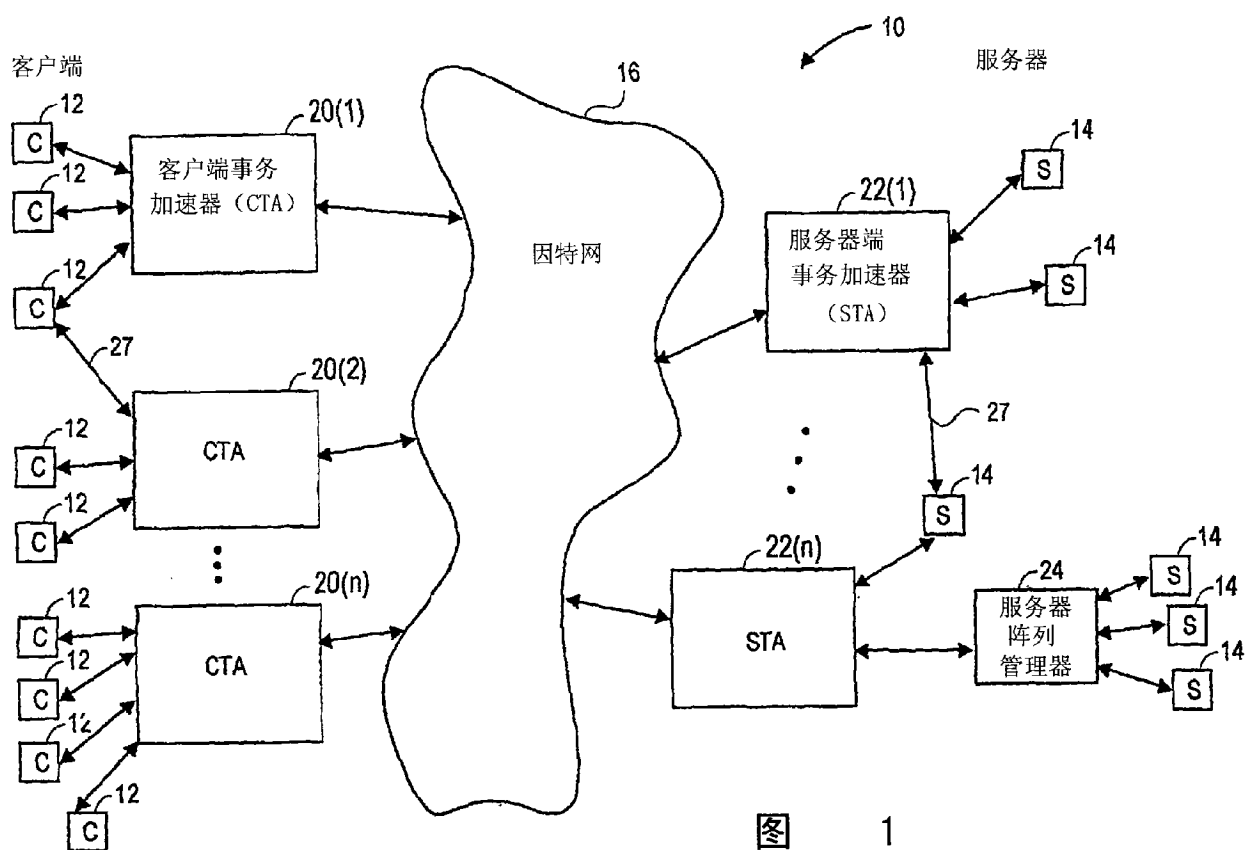
[0160] 在这些方法的混合中,当 RR 接收新绑定或它 CTA 创建的一个,它分配指示新段的参考的“绑定通知”和原始 CTA 给 LAN 上其他的 CTA' s。当另一个 CTA 确定它在它自己的 PSS 中没有所需要的绑定时,CTA' sRR 检查先前接收到的绑定通知的列表。如果所需的绑定在列表中,则请求 RR 通知始发站 CTA 获得绑定。如果 RR 确定它没有绑定且没有来自 LAN 上另一个 CTA 的绑定通知,则 RR 在 WAN 上为绑定发送请求。在此这称为“通知协作”。

[0161] 应该可以理解的是,给定 LAN 可以实现多于一个的上面所描述的协作机制。RR' s 之间为协作的通知可以使用多点传送来实现。例如,协作的客户中的每个(或它们的

CTA' s 或 RR' s) 可以是多点传送群中的一个成员。对于“规范协作”来说,每个原始 CTA 多点传送它接收或创建的新绑定。对于要求协作来说,请求 RR 可以多点传送请求且响应 CTA(s) 可以单播或多点传送它们的回答。多点传送它们的回答允许不请求绑定的其他 CTA 来接收它且可能存储它到其他 CTA' s PSS。对于通知协作来说,可以多点传送通知,但对于请求来说,可以单播那些,因为请求者将知道哪个 CTA 拥有所请求的绑定。当然,通知协作系统可以在绑定通知没有指示原始 CTA,或者没有存储那个信息的情况实现,在这种情况下,可能多点传送绑定请求,但当使用通知协作时最好的方法是跟踪哪个 CTA 发送哪个通知。

[0162] 图 15 是联网系统的方框图,其中实现事务加速且网络处理各种协议和服务。所显示的 CTA 和 STA 耦合到计算 CIFS、NFS、SMTP、IMAP 和 HTTP 事务。在其他设置中,服务器在不同位置且客户在不同位置。在每个情况下,加速协议的事务通过 CTA 和 STA,且能够如上面所述加速,并对参加事务的客户和服务端是透明的。除了图中图示说明的开放协议之外,CTA' s 和 STA' s 可以为诸如 MicrosoftExchange™、Lotus Notes™ 等等的专利协议加速事务。因为具有在此所描述的其他变化,TA' s 可能被集成到客户和服务端中。例如,一些软件卖主可能包括事务加速作为它们的客户-服务器软件套中的一部分。

[0163] 上面的描述被图示说明但没有被限制。根据对上述说明的回顾,对本领域的普通技术人员来说发明的许多变化将变得明显。因此将不参考上面的描述来确定本发明的范围,而是以附加的权利要求书及其等同的整个范围来确定。



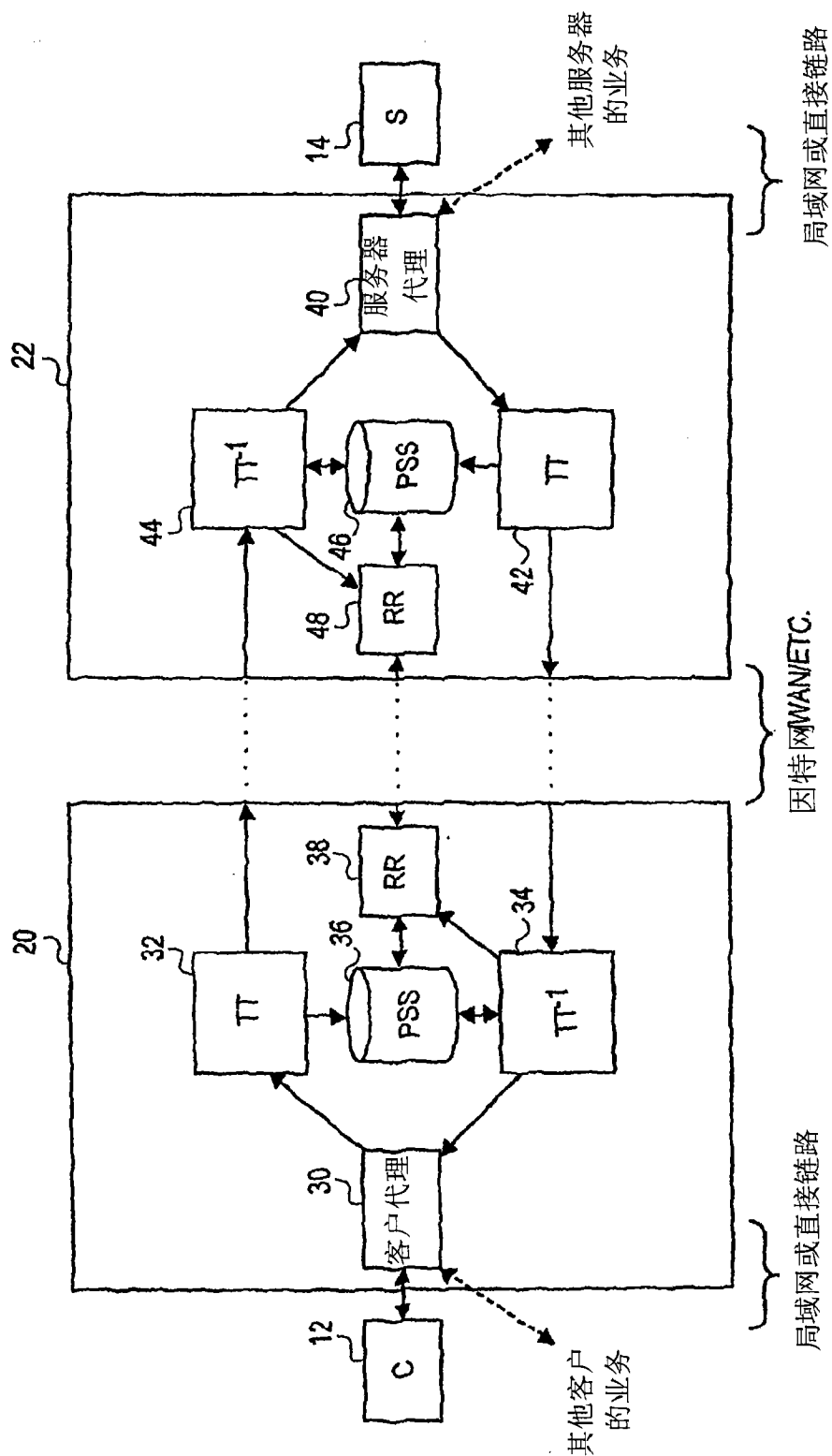


图 2

参考数据		时间戳 N	
R ₁	段1数据	00/00/0000, 00:00:00	...
R ₂	段2数据	...	...
R ₃	段3数据	...	...
...	...		

图 3

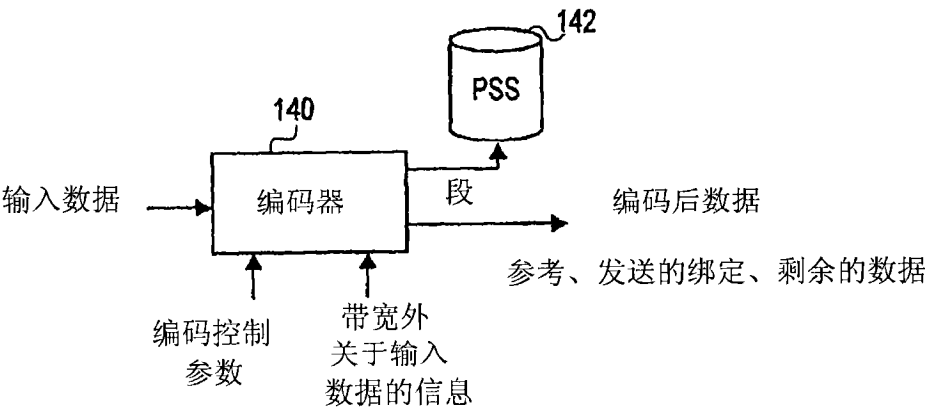


图 4

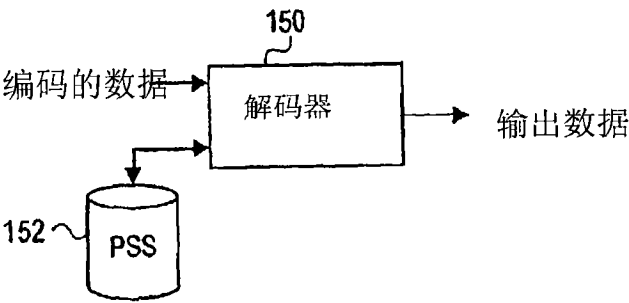


图 5

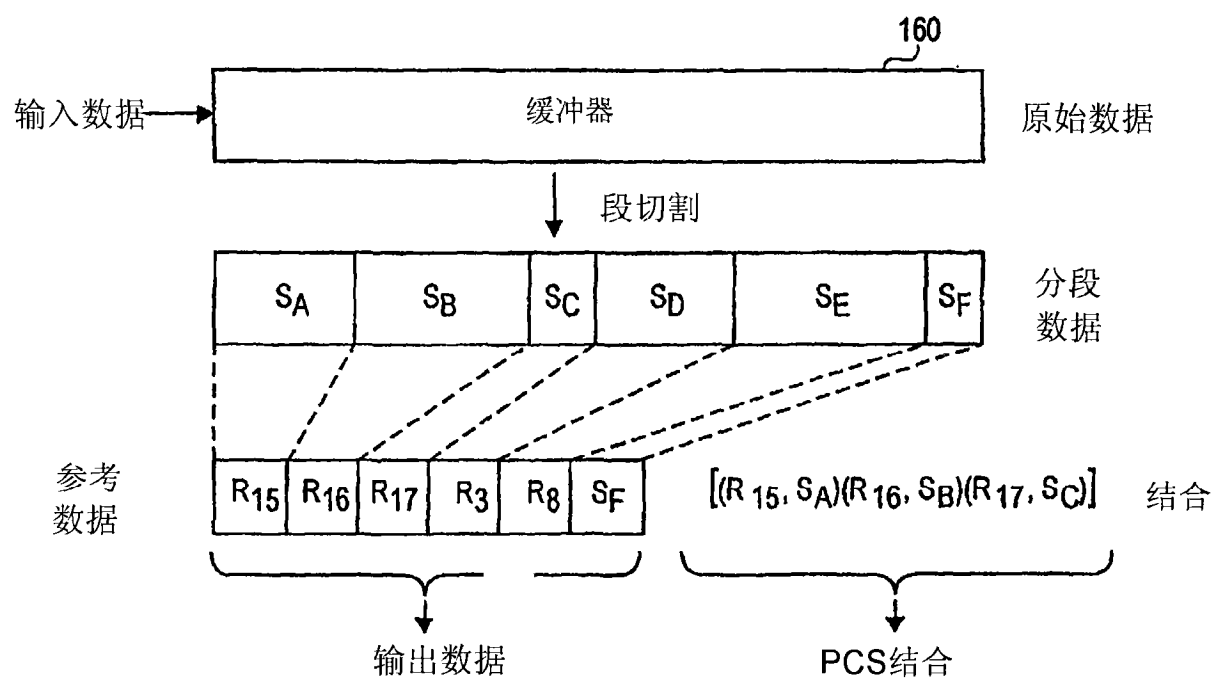


图 6

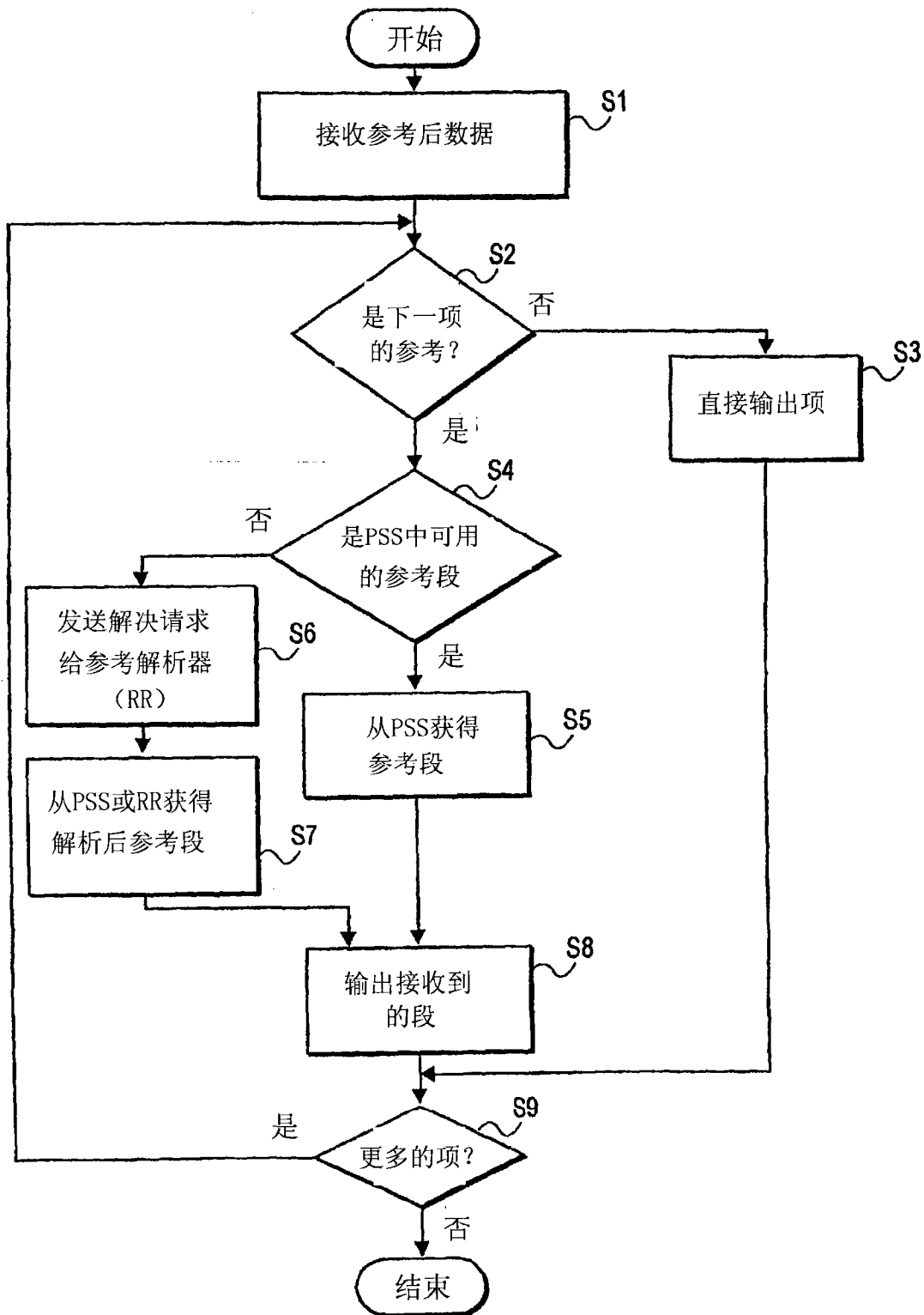


图 7

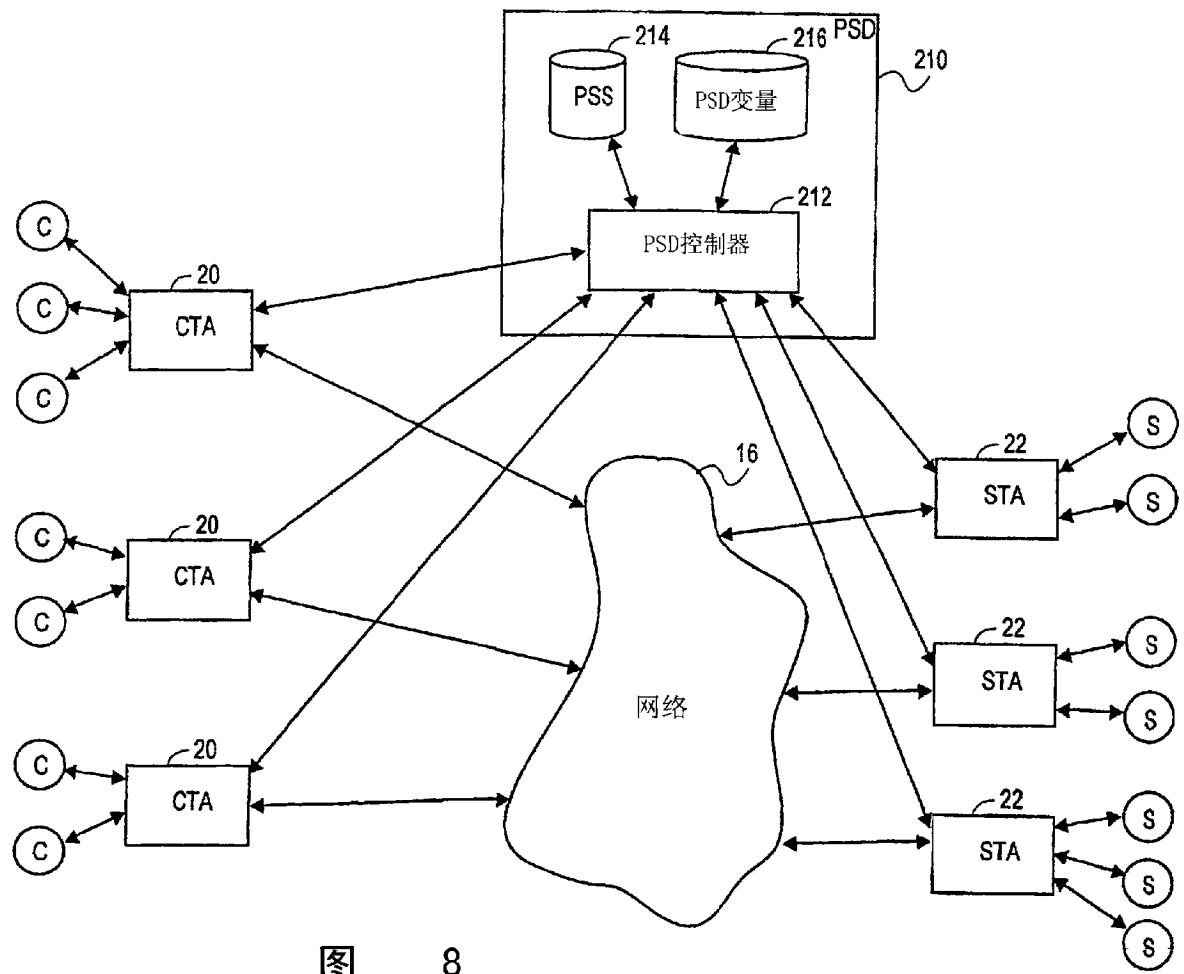
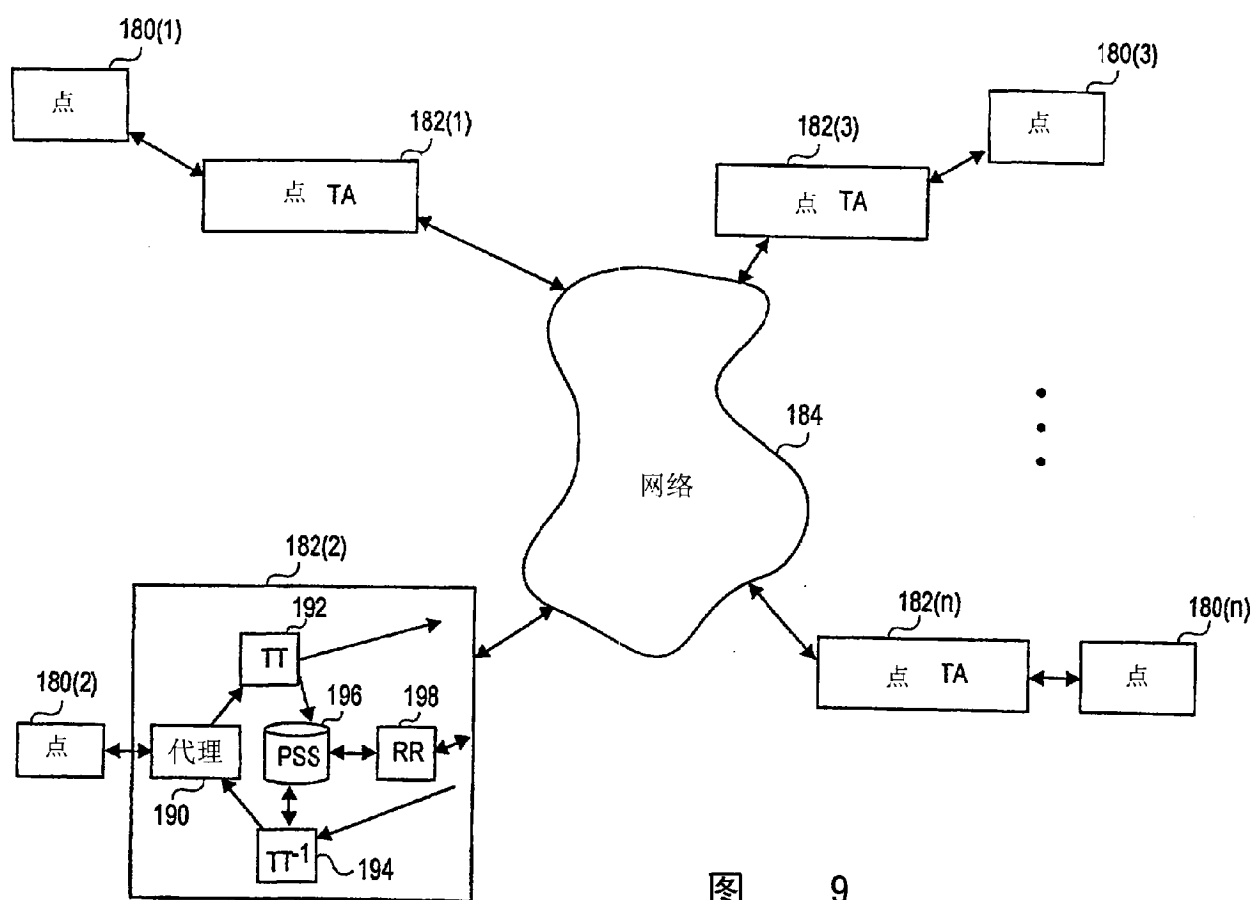


图 8



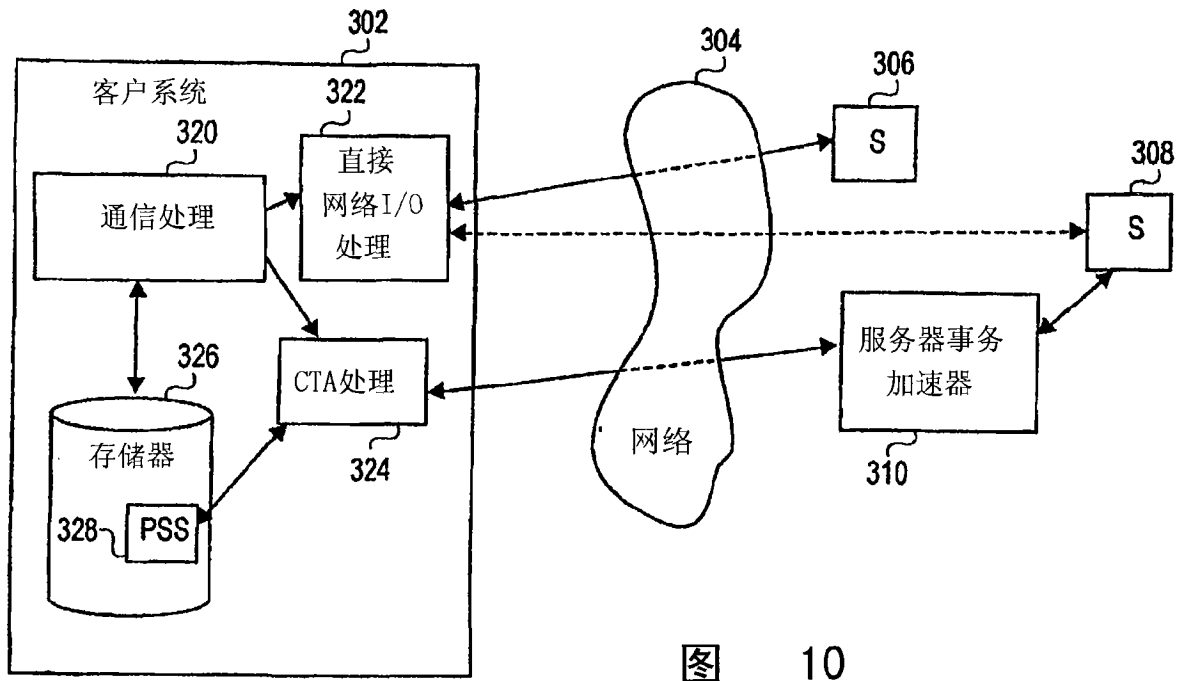


图 10

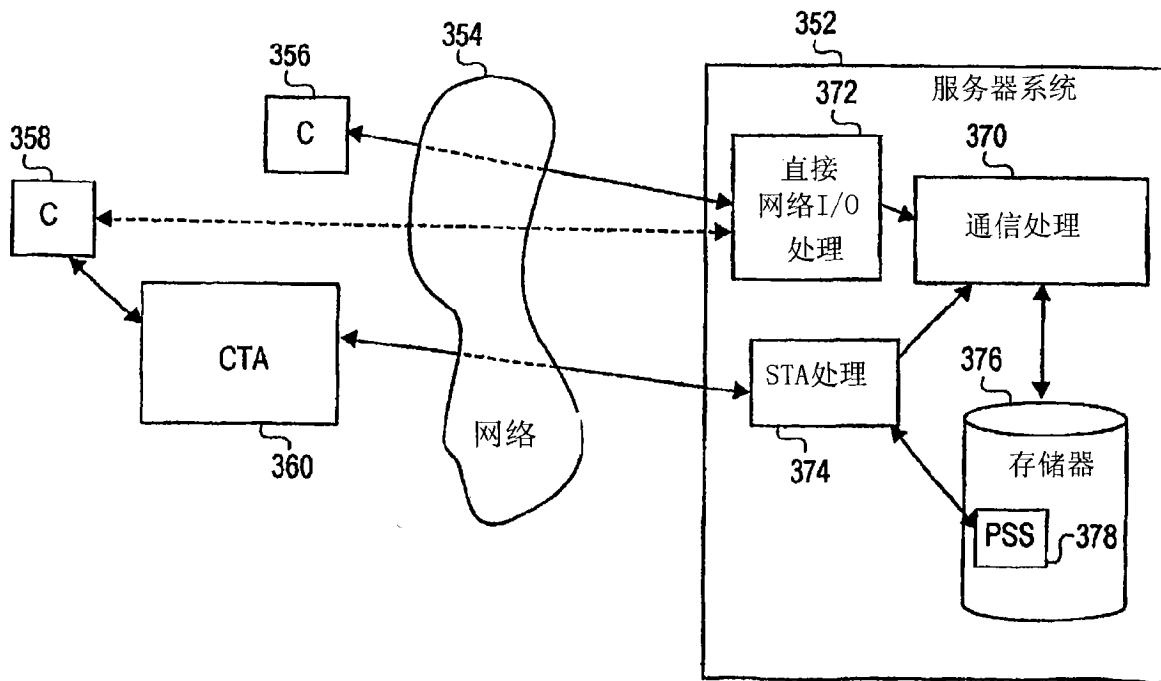


图 11

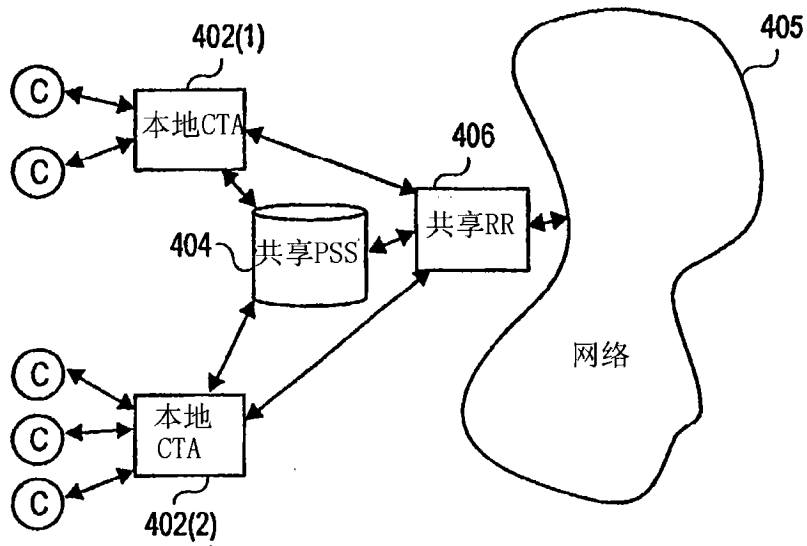


图 12

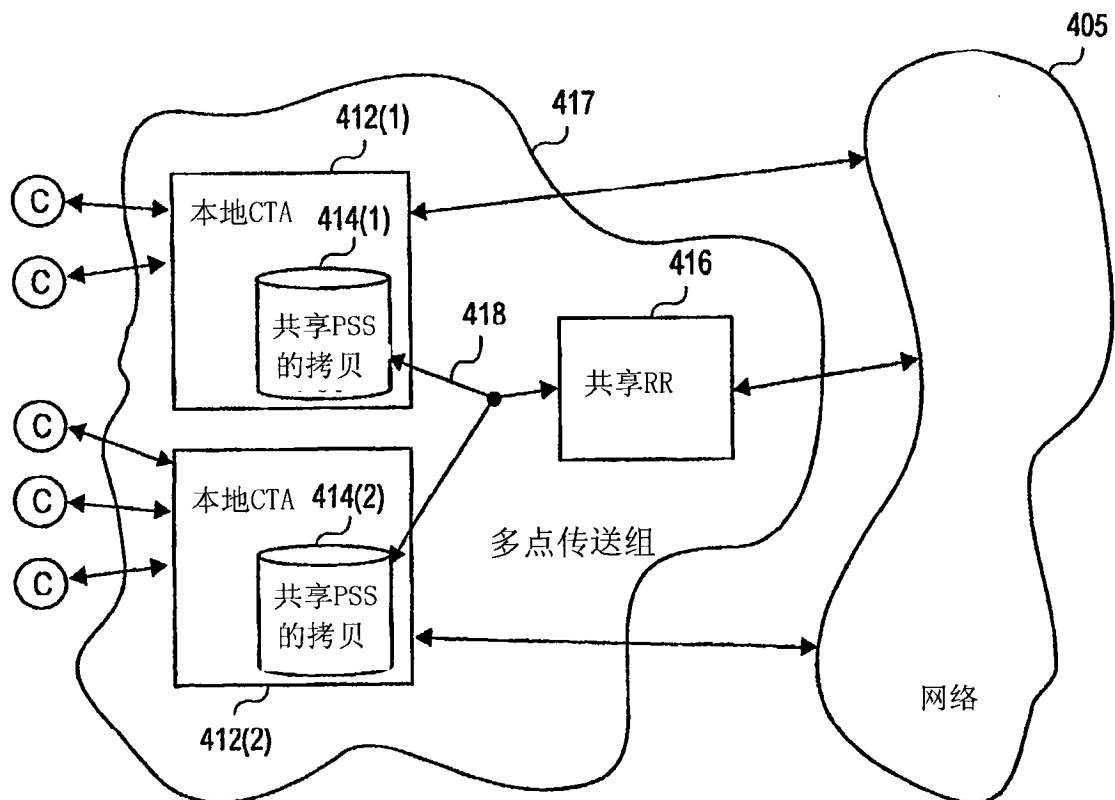


图 13

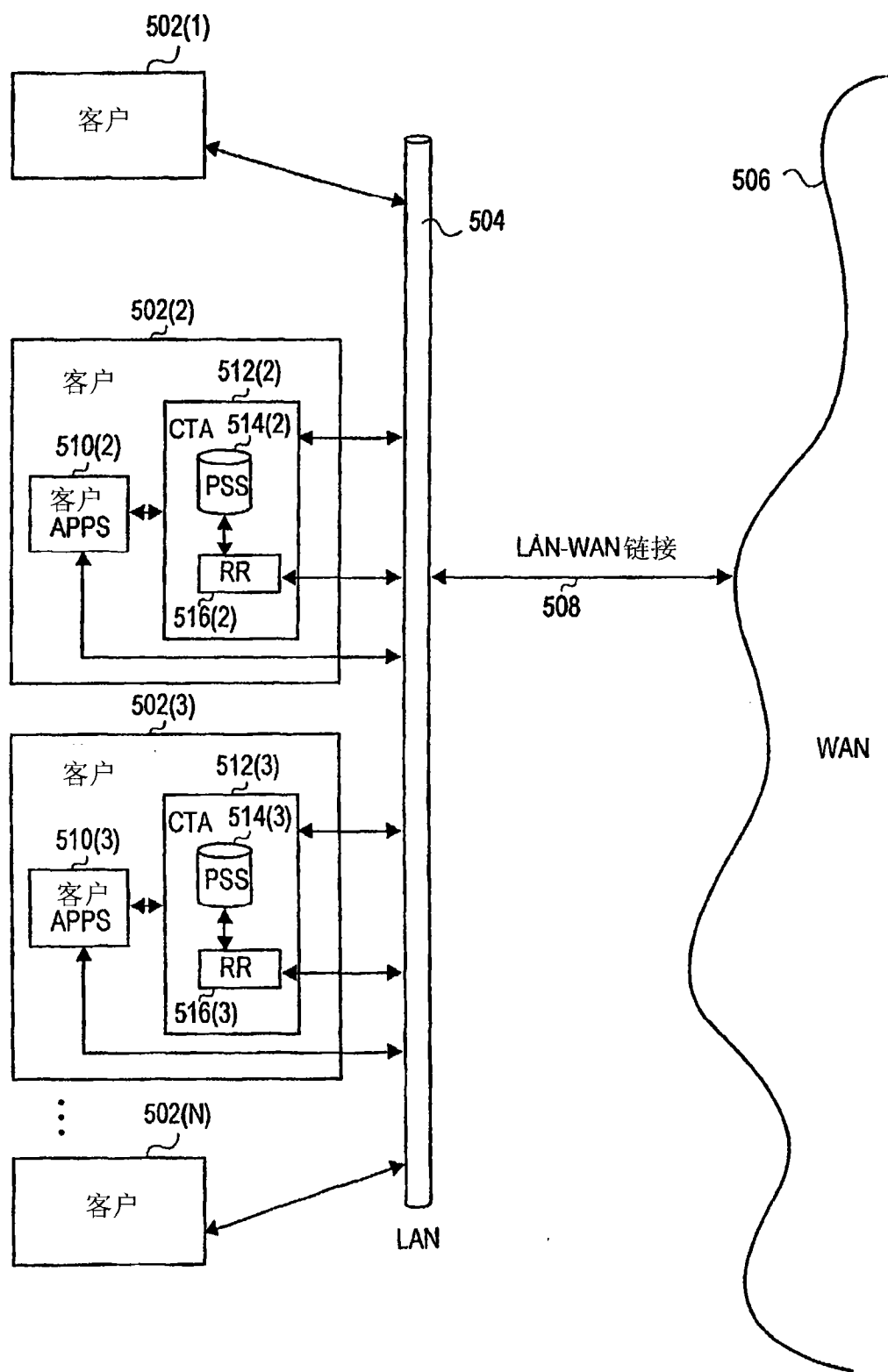


图 14

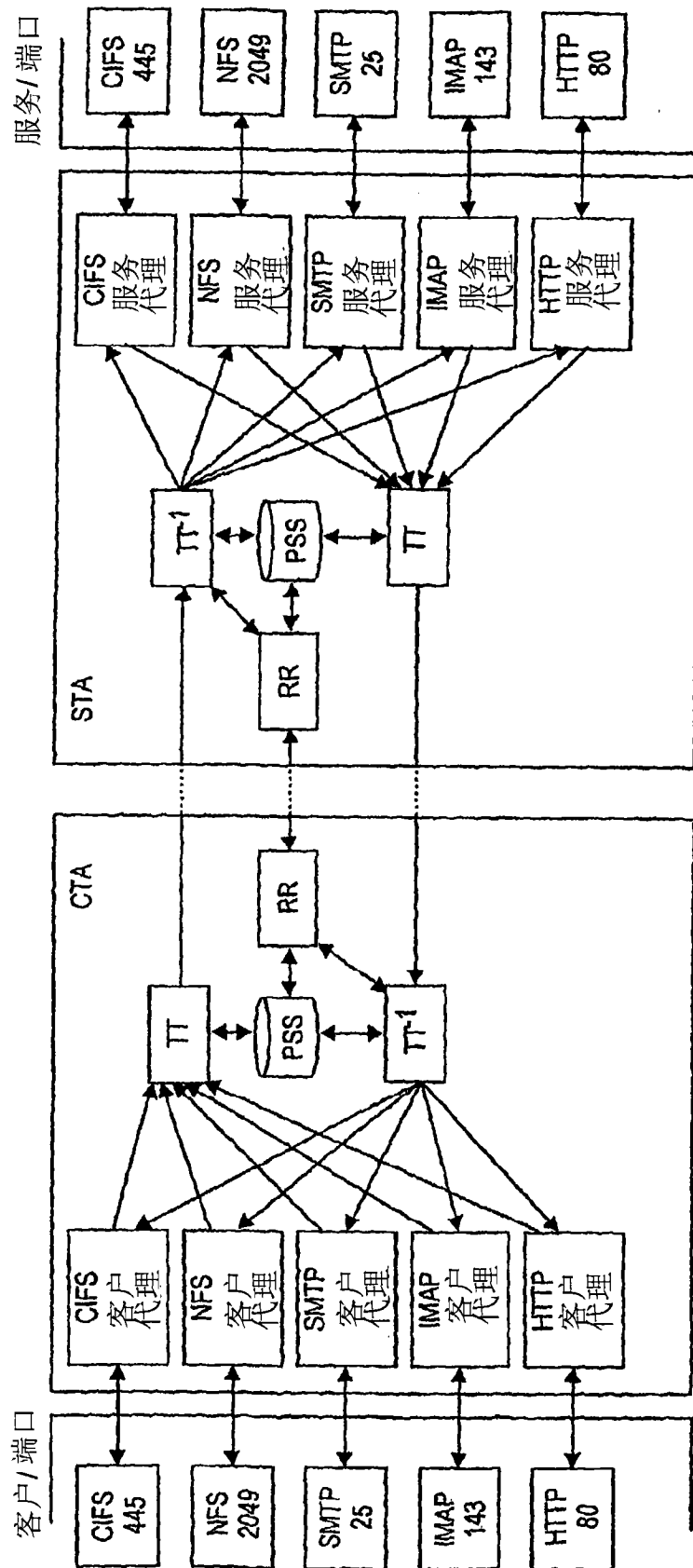


图 15