



## (12) 发明专利

(10) 授权公告号 CN 101553795 B

(45) 授权公告日 2013. 10. 02

(21) 申请号 200680044632. 1

(51) Int. Cl.

(22) 申请日 2006. 10. 26

G06F 15/16 (2006. 01)

(30) 优先权数据

11/289, 983 2005. 11. 30 US

(56) 对比文件

(85) PCT申请进入国家阶段日

US 6806885 B1, 2004. 10. 19, 第 3 栏第 20 行  
至第 5 栏第 34 行、说明书第 7 样第 9-61 行, 附图  
2、4.

2008. 05. 29

US 6323854 B1, 2001. 11. 27, 说明书第 3 样  
第 66 行至第 6 样第 12 行, 附图 2、6.

(86) PCT申请的申请数据

US 6704024 B2, 2004. 03. 09, 全文 .

PCT/US2006/041684 2006. 10. 26

审查员 赵晓敏

(87) PCT申请的公布数据

W02007/064426 EN 2007. 06. 07

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 N·玛吉利斯

(74) 专利代理机构 上海专利商标事务所有限公  
司 31100

代理人 陈炜

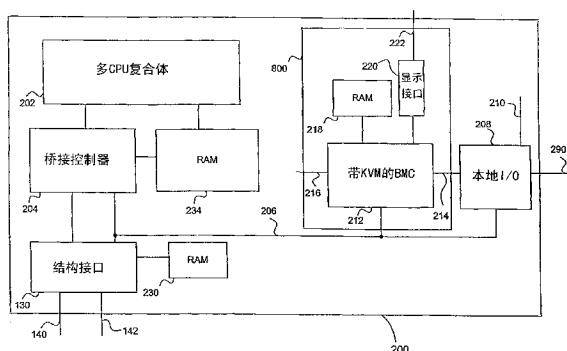
权利要求书3页 说明书23页 附图10页

(54) 发明名称

多用户显示代理服务器

(57) 摘要

多用户主机系统包括与终端服务刀片组合的处理器刀片，以提供加速和代理服务器功能来支持各种远程终端。对于每个远程终端，终端服务刀片和代理服务器功能可改进视频和图形性能。这使得该多用户主机系统能更有效地支持多个用户。终端服务刀片可包括管理每个远程终端的虚拟显示且提供子帧数据的选择性更新的图形处理器。在适当的情况下，子帧数据被编码并经由网络传输到远程终端。终端服务处理器还为预期远程终端及它们相应的网络连接卸载和优化视频数据流。处理器刀片可包括利用先进特征来支持 KVM 管理的基板管理控制器。



1. 一种能够支持多个远程终端的显示代理服务器系统,包括:

图形和显示子系统,具有:

可存储多个终端的显示帧的显示存储器;

用于生成显示帧的一个或更多个绘图引擎,所述显示帧各自可与远程终端上的显示帧相对应,所述显示帧包括子帧,其中所述子帧可被动态设置成块尺寸的任意倍,其中所述块尺寸是数据编码算法的最小实体;以及

用于跟踪显示帧的经修改子帧、以及用于基于此跟踪执行来自所述显示存储器的选择性更新的装置,其中所述跟踪包括生成对应于所述经修改子帧的签名以及对所述经修改子帧执行签名校验;

用于将所述显示代理服务器系统连接到一个或更多个主 CPU 的装置;以及

用于管理来自所述显示存储器的选择性更新从而只有所述选择性更新通过网络子系统被传输到所述远程终端的相应之一的装置。

2. 如权利要求 1 所述的系统,其特征在于,所述远程终端运行因特网浏览器,且通过所述显示代理服务器系统请求 web 更新,所述显示代理服务器系统执行一代理功能以请求和接收来自 web 服务器的 web 更新,将所述 web 更新转换为经编码的部分帧更新,且将所述经编码的部分帧更新提供给所述因特网浏览器。

3. 如权利要求 1 所述的系统,其特征在于,所述远程终端运行因特网浏览器,并对所述显示代理服务器系统执行基于内容的帧请求,并且所述显示代理服务器系统使用已被高速缓存在所述显示代理服务器系统中的数据来合成、编码和供应基于帧的哪些部分已改变的部分帧更新。

4. 如权利要求 1 所述的系统,其特征在于,所述显示代理服务器系统在刀片系统中实现,且所述用于与一个或更多个主 CPU 连接的装置被设置在所述刀片系统的底板上。

5. 如权利要求 1 所述的系统,其特征在于,所述显示代理服务器系统被实现为一装置,并通过网络连接和显示输出路径附连到 web 服务器系统。

6. 如权利要求 1 所述的系统,其特征在于,所述显示代理服务器系统与运行应用程序的 CPU 通信,对所述应用程序作出响应,并执行对所述远程终端的选择性更新。

7. 如权利要求 1 所述的系统,其特征在于,所述显示代理服务器系统在当执行 web 访问时,向所述多个远程终端提供 DNS 查找功能。

8. 如权利要求 1 所述的系统,其特征在于,所述显示代理服务器系统与运行一个或更多个虚拟机的一个或更多个主 CPU 通信,所述虚拟机各自请求图形操作,且其中所述显示代理服务器系统本地地管理多个虚拟机图形请求。

9. 如权利要求 8 所述的系统,其特征在于,所述显示代理服务器系统利用本地管理的所述虚拟机图形请求来产生作出请求的所述虚拟机的虚拟显示,然后将所述虚拟显示的经编码更新经由网络传输到所述远程终端的相应之一。

10. 一种能够支持一个或更多个远程终端的基板管理控制系统,包括:

图形控制和显示子系统,具有:

用于在主 CPU 的请求下在显示存储器中执行图形处理器操作以生成显示帧数据的装置;

用于跟踪显示帧的经修改子帧并基于此跟踪执行选择性更新的装置,其中所述跟踪包

括生成对应于所述经修改子帧的签名以及对所述经修改子帧执行签名校验；

用于编码所述选择性更新以使得没有伪像将会因当前显示窗口的任意对准而产生的装置；以及

用于管理来自所述显示存储器的所述选择性更新从而只有所述选择性更新将通过网络子系统传输到所述远程终端的相应之一的装置。

11. 如权利要求 10 所述的系统，其特征在于，所述选择性更新基于具有固定小块尺寸的子帧。

12. 如权利要求 10 所述的系统，其特征在于，所述图形控制和显示子系统使用 S 缓冲器来管理选择性更新。

13. 如权利要求 10 所述的系统，其特征在于，所述基板管理控制系统在操作系统虚拟机模式下与所述主 CPU 通信，以执行所述图形处理操作的操作系统功能。

14. 如权利要求 13 所述的系统，其特征在于，所述系统与在受保护的管理虚拟机模式下运行的主 CPU 通信，所述受保护的管理虚拟机模式与所述操作系统虚拟机模式隔离，且其中所述管理虚拟机模式管理所述远程终端的部分更新功能。

15. 一种能够支持多个远程终端的基于刀片的多用户系统，包括：

与底板连接的一个或更多个处理器刀片；

包括图形和显示子系统的一个或更多个终端服务加速器刀片，具有：

能存储多个终端的显示帧的显示存储器；

用于生成显示帧的一个或更多个绘图引擎，所述显示帧可各自与远程终端上的显示帧相对应，所述显示帧包括子帧，其中所述子帧可被动态设置成块尺寸的任意倍，其中所述块尺寸是数据编码算法的最小实体；以及

用于跟踪显示帧的经修改子帧以及基于此跟踪执行来自所述显示存储器的选择性更新的装置，其中所述跟踪包括生成对应于所述经修改子帧的签名以及对所述经修改子帧执行签名校验；

用于将所述处理器刀片连接到一个或更多个终端服务加速器刀片的装置；以及

用于管理来自所述显示存储器的选择性更新从而只有选择性更新通过网络子系统传输到所述远程终端的相应之一的装置。

16. 如权利要求 15 所述的系统，其特征在于，所述一个或更多个终端服务加速器刀片与主 CPU 通信，所述主 CPU 运行应用程序软件且利用将所述图形和显示子系统虚拟化的多用户远程客户机协议；

执行作为远程用户客户机的虚拟图形功能；以及然后

将选择性更新从所述图形和显示子系统发送到所述远程终端。

17. 如权利要求 15 所述的系统，其特征在于，所述终端服务加速器刀片与运行一个或更多个虚拟机的一个或更多个 CPU 通信，所述虚拟机各自请求图形操作。

18. 如权利要求 15 所述的系统，其特征在于，所述终端服务加速器刀片利用所述图形和显示子系统来本地管理来自多个独立虚拟机模式 CPU 的虚拟机图形请求，以产生虚拟机的虚拟显示、然后将所述虚拟显示的经编码更新经由网络传输到远程终端，并且所述多个独立的虚拟机模式 CPU 互相没有察觉到对方。

19. 如权利要求 15 所述的系统，其特征在于，所述处理器刀片作为 web 服务器操作。

20. 如权利要求 15 所述的系统，其特征在于，所述基于刀片的多用户系统向访问多个 web 服务器的多个远程终端提供因特网服务管理功能 web 服务器。

## 多用户显示代理服务器

### [0001] 发明背景

#### 技术领域

[0002] 本发明一般涉及多用户主机系统，尤其涉及使用代理服务器支持用于远程客户机的终端服务。

#### [0003] 背景技术的讨论

[0004] 开发有效的多用户主机系统是当代系统设计师和制造商的重要目标。

[0005] 常规的计算机系统可使用本地显示设备来向一个用户直接显示输出。因为由各种使显示设备与计算机系统的输出电耦合的物理连接所强加的限制，本地显示设备通常靠近计算机系统放置。一些计算机系统可以支持具有由该物理连接引起的相似接近限制的第二显示设备。

[0006] 远程用户需要有选择适当的查看位置和与主系统的网络连接的附加灵活性。例如，在公司环境中，企业可能希望将所有的主机保存在既有物理安全性又有诸如空调和电源备用系统的环境管理的安全的中央“计算机机房”。然而，对于用户而言，从他们的办公室和从位于“计算机机房”之外的桌面使用主机系统是有必要的。

[0007] 今天通常的办公室环境包括物理地位于用户位置处的个人计算机和日益增多的瘦客户机。这些个人计算机和瘦客户机在具有用于存储、文件服务、文件共享、网络管理和各种管理服务的集中式系统的网络上操作。起初，当用户在他们的本地桌面上运行应用程序时，系统将所有的与计算机系统相关联的磁盘存储器集于中心。近来，认识到安全性的益处、降低的操作成本、以及集中控制的普遍需求，个人计算机和瘦客户机可在基于服务器计算（SBC）方案中作为远程终端（RT）允许，这些 SBC 解决方案在服务器上运行应用程序。

[0008] 针对 SBC 环境中 RT 的传统方法是主系统使用某种形式的服务器到客户机通信交换，诸如微软的远程显示协议（RDP）。RDP 在服务器上使用它自己的视频驱动程序，且使用 RDP 协议将绘制信息构建成网络分组且经由网络将它们发送到客户机。客户机接收绘制数据且将这些分组转换成相应的微软 Win32 图形设备接口（GDI）API 调用。对将客户机键盘和鼠标命令重定向到服务器以及管理本地音频和本地客户机驱动程序的支持也被包括在内。

[0009] 为了增强主系统与客户机之间的通信，其它系统已经使用主系统的主 CPU 从而为 RT 改进性能。这已经针对瘦客户机和作为远程客户机的传统 PC 进行。这种方法对一次只支持一个用户的主系统有效。然而，对多用户系统，在主系统上使用主 CPU 来为任一用户改进性能的方法具有相当大的局限性。用于针对一个用户作最优化的诸如主存储器和 CPU 周期的计算资源可减少支持额外用户的工作负荷的能力。

[0010] 从单个主机有效地支持多个用户可以降低成本。在通常的办公室环境中，每个人同时且同样地使用他们的计算机是很少的，任何一个用户使用他们计算机的所有计算资源也是很少的。因此例如，有一百个办公室的公司可能只需要在任一时刻支持六十个用户的系统。即使如此，这种系统也可被设计成支持所有的百位用户，从而给他们足够的计算吞吐量以显现出他们每个都有他们自己的主机。在分布式办公室环境中，集中式多用户系统可

经由各种带宽链路连接以支持在各时区的不同工作时间期间在世界不同部分位置上的 RT。

[0011] 基于服务器计算是另一种更有效地为多个用户分配计算资源的方法,其中用户的应用程序在该服务器上运行且只有 RT 服务在该用户终端上得到支持。SBC 使主系统能在多用户操作环境中动态地分配诸如存储器和 CPU 周期的共享资源。SBC 系统可以采用多用户操作系统、虚拟机 (VM)、负载均衡的技术及其它手段来基于多个标准准许不同用户访问不同级别的性能和资源。可使用不同优先级方案分配 SBC 资源。SBC 可实现更高的数据安全性,集中对机构的支持,增强的灾难恢复及业务持续,且减少跨组织的数据存储需求。web 服务器是可以为包括基于浏览器的客户机在内的各种客户机提供多用户平台的一类 SBC。

[0012] 基于刀片的服务器为可缩放多用户主系统实现一有效的体系结构。刀片计算功能、输入 / 输出 (I/O) 功能、底板体系结构以及交换的划分在基于刀片服务器的设计中都是重要的。每个刀片可以构成一完整计算机的功能或者刀片服务器可以跨刀片分享重要功能。随着 CPU 不断通过包括多处理器核来增加它们的性能,将单个用户限制于单个刀片产生了降低的经济意义。刀片服务器系统中的每个刀片可能需要被远程监视。随着刀片系统的多功能本质,一刀片机架可能需要被配置成最佳地匹配不同工作负荷。

[0013] 图 1 示出典型的现有技术多刀片系统级体系结构 100,其包括标记为 1 到 8 (102、104、106、108、110、112、114 和 116) 的刀片和交换机 120 和 112 以及通过每个交换机的路径。交换机还可以包括它们之间以及到链路 124 的路径,该链路 124 可以进一步将一刀片机架扩展到另一刀片机架或一些其它子系统。图 1 的刀片在数量上可有变化且每个刀片可以是处理器刀片、I/O 刀片、交换刀片或该三种的某组合。I/O 刀片可以包括它们自己的外部接口,如由刀片 4108 路径 180 和由刀片 8116 路径 190 所示。外部接口的示例可以包括诸如光纤信道的非网络接口和诸如以太网和 10G 以太网的 iSCSI 或网络接口。

[0014] 交换机 1 120 和交换机 2 122 的示例交换矩阵示出从每个交换机到每个刀片的连接 140 和 142。每个交换机的物理或逻辑传输的类型可以是相同的或不同的。路径 144 可以将交换机 120 和 122 连接或桥接到一起。交换机和连接的组合被称为交换结构 150,该交换结构 150 可以分布在每个刀片上、是底板的一部分、在交换刀片上实现或可以涉及该三个的组合。该交换结构的路径可以是单向的、双向的或它们的组合。每个刀片可以与任何数目的交换机连接,且可以包括作为该刀片功能一部分的针对不同交换机的桥接功能。更先进的系统可包括全网型拓扑结构且可以具有冗余。

[0015] 一些用于该交换结构 150 的连接的示例包括快速外围组件互连 (PCI)、10G 附连单元接口 (XAUI)、无限带宽、高速 I/O、星型结构、先进交换互连 (ASI)、千兆比特以太网、光纤信道以及其它电互连和光互连中的一个或多个信道。在一些情况下,每个刀片的功能芯片可以直接包括结构接口 130。在其它情况下该刀片可以包括执行接口连接的结构接口芯片或桥接芯片。每个刀片内的结构和桥接芯片可以与系统交换机中的一个或多个连接,且可以在这些交换机之间集成附加的桥接功能。

[0016] 然而,基于刀片的多用户主机可能需要增加的复杂度,以对各种服务器软件和基于网络的应用程序及组织可以具有的各种 RT 设备有效地管理、控制和递送丰富的应用程序性能。需要可使基于刀片的多用户主服务器用杰出的计算和显示性能更有效地支持许多远程用户的解决方案。

[0017] 概要

[0018] 本发明提供一种用于基于刀片多用户计算机系统的有效的体系结构,包括一个或多个能够交互式图形和视频处理的远程终端(RT),该有效的体系结构通常管理应用程序并执行基于服务器计算。每个RT具有其自己的键盘、鼠标和显示器,以及可能其它外部设备。这些RT为个体用户提供对服务器上可用的应用程序以及丰富图形用户接口的访问。该多用户计算机系统可以运行多用户操作系统,可以运行单用户操作系统的虚拟化示例,可以运行针对多个用户的web服务器引擎,可以运行代理服务器或可以运行其某种组合。

[0019] 在第一较佳实施方式中,处理器刀片包括允许来自RT的远程管理和控制的基板管理控制器(BMC)。除了提供传感和状态信息之外,BMC包括键盘、视频和鼠标(KVM)能力,从而系统管理者可以像他本地连接到该刀片一样地远程访问该处理器刀片。BMC和KVM起支持“带外”操作的作用,而不对该CPU和板的基本操作作附加的运行时间变化,从而可最有效地执行运行时间问题的诊断。该远程KVM的显示相关特征由利用选择性更新和需要时利用各种形式的显示数据压缩的图形处理器和显示数据编码器支持。该显示数据编码器功能可以被集成到TSA 424内或作为用于组合GPU-TSA的专用数据编码器752。

[0020] 在第二较佳实施方式中,终端服务刀片(TSB)利用软件、图形处理器和数据编码的组合来通过针对每个RT创建虚拟显示环境支持多个RT。与该RT通信的最常见方法包括发送经封装的图形命令或发送经编码的子帧数据。管理RT的软件可以在该主要的主处理器刀片即该TSB的CPU上、在终端服务加速器(TSA)上、在RT上或在其组合上运行。对每个RT的选择性更新可用软件或在多用户图形处理器单元(MU-GPU)或组合TSA-GPU内硬件的协助下进行协调。该图形处理器可以遵循提出的VESA数字分组视频链路(DPVL)标准或使用报头、状态比特和签名用于子帧的经改进方法。其它增强中,高速PCI或另一总线代替DVI用于输出数据,附加的数据编码在该图形处理器内或用附连到该图形处理器的编码器执行,且该软件为多用户支持使用一个或多个图形处理器。

[0021] 该TSB可以执行包括客户机端接的变化级别的代理服务,且甚至可以完全仿真多个RDP客户机,从而在该处理器刀片上运行的RDP主机将在该TSB上运行的进程视为唯一已知的客户机。在此方法中,该TSB可以创建与主处理器未察觉到的RT的完全独立接口。该TSB和RT之间的此类接口可具有分离代理(split proxy)的形式。作为分离代理,该TSB和RT经由专用信道通信,在该专用信道上它们能比使用诸如RDP或网络浏览器协议的更标准协议更为有效地通信。

[0022] 为了附加的效率,该处理器刀片可以运行跟踪软件,该跟踪软件可以与TSA组合以截取诸如视频回放的功能。该TSA可以在由CPU解码前截取视频数据流并可将该本地视频流或诸如经译码或经转换版本的修正版本传送到目标RT,而不是使该主CPU本地地执行视频解码并提供传输到RT的位图。除标准的RDP信道之外,向RT的传送可采用其它专用信道,同时仍然在RDP协议内受管理。

[0023] 在每个实施方式中,在图形处理和选择性更新过程之后,该数据被编码且然后被封装成网络待更新分组。网络I/O刀片、网络处理器、或连同较简单的网络控制器一起工作的CPU,经由有线和/或无线网络将该图形分组传输到RT。在某些KVM配置中,BMC将执行网络处理使得处理器刀片CPU不受该更新分组操作干扰。在另一配置中,CPU利用“虚拟技术”来保护性地分割和执行包括用户任务的操作系统功能以及BMC的带外管理任务。该BMC可以与该系统中的另一网络控制器或网络物理层(PHY)通信。每个RT系统对预期作显示

的图形分组进行解码，管理帧更新并执行针对该显示屏幕的必要处理。其它特征，诸如在网络传输中丢失的掩蔽分组，由该远程显示系统管理。当没有新的帧更新时，该远程显示控制器用来自先前帧的数据刷新显示屏。

[0024] 各种网络系统可反馈来自各种连接到 TSB 的有线和无线网络的网络信息。该 TSB 使用网路信息来影响产生 RT 更新的各种处理步骤，并且基于该网络反馈可以改变针对不同 RT 的帧速率和数据编码。此外，针对包括具有噪音传输信道的网络的系统，编码步骤可与前向纠错保护相组合，以准备传输信道特征的传输数据。这些步骤的组合为每一个 RT 保持具有低等待时间的最佳帧速率。TSA 和 TSA-GPU 可被实现为分离的子系统或者可与其它卸载和加速处理组合，这些卸载和加速处理诸如网络处理器、安全性处理器、XML 加速器、iSCSI 处理器或这些的任何组合。

[0025] 因此，至少出于上述原因，本发明有效地实现了利用各种不同组件来便于系统的互操作性和功能性的灵活的基于刀片的多用户计算机系统。本发明因而有效地实现了增强的基于刀片的多用户服务器。

[0026] 附图简述

[0027] 图 1 是包括多个刀片和交换结构的现有技术基于刀片多用户计算机系统的框图；

[0028] 图 2 是根据本发明第一实施方式的多 CPU 处理器刀片的框图，该多 CPU 处理器刀片具有多 CPU 复合体和带远程键盘、视频和鼠标 (KVM) 支持的基板管理控制 (BMC)；

[0029] 图 3 示出经由网络与图 1 的基于刀片的多用户计算机系统连接的 RT；

[0030] 图 4 是根据本发明第二实施方式的终端服务刀片 (TSB) 的框图；

[0031] 图 5 表示被组织到八个显示区的存储器，该八个显示区之一包括显示窗口且其中的两个用于支持一个大的 RT 显示；

[0032] 图 6A 示出图 5 显示图 536 的更详细视图；

[0033] 图 6B 示出细分为小块的图 6A 的矩形；

[0034] 图 7 是示出图 4 的具有集成图形处理器单元的示例性片上系统 (SOC) 终端服务加速器 (TSA-GPU 700) 的细节的框图；

[0035] 图 8 是包括图形子系统和其它带外处理的带 KVM 的基板管理控制器的框图；

[0036] 图 9 是根据本发明的一个实施方式的用于执行终端服务和显示代理服务器操作的方法步骤的流程图；以及

[0037] 图 10 是根据本发明的一个实施方式的用于执行针对远程终端的网路接收和显示过程的方法步骤的流程图。

[0038] 较佳实施方式的详细描述

[0039] 本发明涉及对基于刀片的多用户计算机系统的改进，该计算机系统带对远程模块的支持。虽然所描述的实施方式涉及基于刀片的多用户计算机系统，但是相同的原理和特征同样可应用于其它类型的单用户和多用户系统以及其它类型的远程终端。

[0040] 如下参照图 3 所述，基于刀片的多用户计算机系统 100，也称为“主机 100”，被设计成支持在远程终端上的多个用户。每个 RT 能够时间共享该主机 100，好像该主机 100 是它们自己的本地计算机且在具有能在本地计算机上实现的同类用户体验的情况下具有对所有类型的图形、文本和视频内容的完全支持。当使用网络连接时，路径 180 和 190 与可连接到图 3 网络路径 390 的图 2 路径 290 和图 4 路径 490 相对应。除了 RT 连接之外，主机

100 可以通过链路 124 连接到 WAN、存储子系统、其它主机或各种其它数据中心连接，这些数据中心连接可采取千兆比特以太网 (GigE)、10G 以太网、iSCSI、光纤信道 (FC)、光纤信道 IP (FCIP) 或另一电连接或光连接。主机 100 可以支持各种多用户操作系统 (OS) 或对可用于处理器刀片中的一个或更多个的单用户 OS 进行虚拟化的软件。诸如 Citrix 或 Windows 服务器的操作系统被设计成多用户 OS。Windows XP，虽然不是为同时的多个用户具体设计的，但在诸如 VMWARE 或 XenSource 的低级虚拟化软件或另一手段的帮助下可在这种配置中用于如此快地执行用户交换从而显现为多用户 OS。不同管理控制可使 RT 和程序静态地或动态地从处理器移到处理器。负载均衡可针对每个处理器由该 OS 执行或者该系统可跨多个处理器执行负载均衡。主机 100 还可运行一类 web 服务器并通过基于网络的接口支持多个用户。该主机 100 可用作针对各个 RT 的代理服务器，并与应用程序服务器或 web 服务器通信。

[0041] 图 2 是可用作基于刀片多用户处理器系统 100 的刀片 102-116 中的一个的处理器刀片 200 的一个实施方式的框图。每个刀片 200 本身可以主机，或者多个刀片可被一起置于机架上以创建能力更强的主机。主机具有的处理器刀片、I/O 处理和 CPU 越多，就可同时支持越多的用户。处理器刀片 200 的基本组件优选地包括，但不限于，多 CPU 复合体 202、总线桥接控制器 204、诸如高速 PCI 的主系统总线 206、本地 I/O 208、主 RAM 234、连接到交换机、底板和其它刀片的结构接口 (FI) 130、以及可任选的基板管理控制 (BMC) 子系统 800。

[0042] 取决于交换结构 150 的传输和物理层接口的类型，结构接口 (FI) 130 可以包括重要处理。例如，该结构接口 130 可包括既针对本地生成分组又针对 ASI 结构接口执行所有必要的分组层过滤和处理的 10G 以太网处理器 (未示出)。这种结构接口处理器可能需要外部 RAM 230 或者可能具有充足的内存。网络物理层接口 (PHY) 可与 FI 处理器结合或可利用外部 PHY 组件。本地 I/O 208 和本地 I/O 连接 290 可由此同一 FI 处理器控制。此外，该结构接口 130 可直接与系统总线 206 附连，而不是通过桥接控制器 204 与系统总线 206 通过接口连接。

[0043] 在一个示例系统中，交换机 1 120 和关联路径 140 利用可允许快速 PCI 分组隧穿的 ASI 总线协议。交换机 2 122 和关联路径 142 可以是 XAUI 类总线，且可针对存储和联网更为优化并且可使用 10G 以太网协议。该系统可主要将 ASI 总线用于不同处理器刀片之间的通信，而将 XAUI 总线用于与联网和存储刀片的通信。每个刀片可包括一个接口，两接口还包括在两个总线之间桥接话务传输的能力。

[0044] 图 2 的多 CPU 复合体 202 可包括一个或更多个处理器芯片，每个处理器芯片具有一个或更多个 CPU 核 (未示出)，每个 CPU 核可执行多个同时线程。在每一层，每个 CPU 核都可包括任一专用高速缓存，而在某些层多 CPU 核可以共享高速缓存。该多 CPU 复合体 202 可独立地控制和访问 RAM 234，或者它可利用桥接控制器 204 来执行对 RAM 234 的访问。在一些其它配置中，该多 CPU 复合体 202 可直接与主系统总线 206 通过接口连接 (未示出)。

[0045] 本地 I/O 208 可包括各种 I/O 接口，并控制外部接口 290 和位于处理器刀片 200 内资源的通路 210。诸如存储和联网的主 I/O 功能可被包括在每个处理器刀片 200 上，或者其它专用 I/O 刀片可经由交换结构 150 得到支持。取决于对交换结构的物理或逻辑连接的选择，接口 140 和 142 可直接结合桥接控制器 204 或者可能需要结构接口 130 芯片。

[0046] 因为每个处理器刀片 200 有效地可以是“计算机系统”，所以具有观察和管理该系

统的方法是非常合乎需要的。因为主机 100 可位于一具体的计算机机房内,所以使该系统能从远程终端 (RT) 300 观察和管理常常是合乎需要的。RT 可以是具体设计的瘦客户机,或者对于管理功能更通常是计算机运行管理软件。此外,RT 可包括用于执行这些管理功能的基于浏览器的软件。处理器刀片 200 可包括提供用于基于浏览器的管理的用户接口的 web 服务器。

[0047] 当处理器刀片 200 正在被管理且不处于来自远程用户的工作负荷下,多 CPU 复合体 202 的主 CPU 经常用于与 RT 通信。然而为了监视各种环境状况且当处理器刀片 200 在执行来自远程用户的工作负荷时对其进行观察和管理,将如图 8 详细说明的基板管理控制 (BMC) 子系统 800 包括在内是合乎需要的。

[0048] 作为 BMC 800 中的独立 CPU 的替代或附件,202 中的主 CPU 可使用虚拟化技术来将管理功能从操作系统和用户功能中隔离。尽管在多核 CPU202 中一具体核可专用于这种管理功能,但适当设计的 CPU 可包括诸如该英特尔范德普尔 (Intel Vanderpool) VT 技术的硬件,以使任务或线程互相隔离而与它们在哪个核上运行无关。这样,单个 CPU 核可以同时既运行用于支持带外管理功能的受保护的虚拟管理机模式,又运行操作系统虚拟机模式以支持各种操作系统和用户任务。

[0049] CPU 核的不同虚拟机各自可以连同 BMC 800 的不同方面进行操作。在一较佳实施方式中,带 KVM 212 的 BMC 包括诸如 TSA-GPU 700 的 MU-GPU 412 的本地图形处理器,该处理器对在虚拟机模式下运行的标准操作系统执行显示处理。对于远程 KVM 操作,远程管理员可能想要观察或执行处理器刀片 200 的带外管理。主 CPU 202 的受保护的虚拟管理机模式可用于协助执行带外远程管理。该远程 KVM 管理可包括从带外网络接口访问相同的本地图形处理器显示。这种对图形处理器显示的访问可利用在图 7 中详细描述的 TSA-GPU 700 的更先进特征。在另一实施方式中,带 KVM 的 BMC 212 内的独立 CPU 808 用于管理带外远程管理,而不是用于在主 CPU 上使用受保护的虚拟管理器模式。

[0050] 图 3 是远程终端 300 的框图,根据本发明的一种实施方式,该远程终端 300 优选地包括,但不限于,显示屏 310、本地 RAM 312、以及远程终端系统控制器 (RTSC) 314。该 RTSC 314 包括键盘、鼠标及 I/O 控制子系统 316,该 I/O 控制子系统 316 具有对鼠标 318、键盘 320 及其它各种设备 322 的相应连接,其它各种设备 322 诸如是用于再现音频的扬声器或可支持各种设备的通用串行总线 (USB) 连接。还可包括其它集成或外围连接,用于通过包括生物鉴定或安全卡的安全手段支持用户认证。这些连接可专用于单一用途,诸如 PS/2 类键盘或鼠标连接,或用于诸如 USB 的更普通用途。在其它实施方式中,该 I/O 可包括游戏控制器、本地无线连接、IR 连接或根本没有连接。远程终端系统 300 还可包括诸如 DVD 驱动器的其它外围设备。

[0051] 本发明的一些实施方式不需要远程终端系统 300 上的任何外部输入。这种系统的一个示例是不同显示器在不同位置可用且可以示出各种信息性和娱乐性信息的零售店或电子告示牌。每个显示器可独立地操作且可以基于各种因素更新。类似的安全系统还可包括一些接受触摸屏幕输入的显示器,例如信息亭或银行里的自动出纳机 (ATM)。其它安全系统,例如娱乐场的游戏机,也可基于这类 RT。

[0052] 图 1 的主机 100 可包括来自每个刀片的网络接口 (例如,180、190) 或者共享的网络控制器可被包括在内。在任一情况下,建立从主机 100 到 RT 300 的输入 390 的网络连接。

网络控制器 336 支持在可以是有线或无线的网络路径 390 上的安全协议,且经由该网络的数据传输可通过密钥交换加密。公用网络示例是以太网,诸如运行某些类型以太网,优选为千兆比特以太网的 CAT 5 布线,其中 I/O 控制路径可与 UDP 传输相组合地使用以太网支持协议,诸如标准的传输控制协议和因特网协议 (TCP/IP) 或某种形式的轻量信号交换。诸如实时流协议 (RTSP) 和实时传输协议 (RTP) 与实时控制协议 (RTCP) 一起的工业成果可用于增强分组传输且可通过添加重发协议进一步增强。其它围绕使用服务质量 (QoS) 成果的较新成果,诸如层 3 差分服务代码点 (DSCP)、作为数字生活网络联盟 (DLNA) 的一部分的 WMM 协议、微软 Qwave、uPnP、QoS 及 802.1P 也是使用现有网络标准的增强方法。

[0053] 除了用于支持 I/O 设备的分组,该网络携带经封装和编码的显示命令和显示所需的数据。CPU 324 与网络控制器 336、2D 绘图引擎 332、3D 绘图引擎 334、数据解码器 326、视频解码器 328 及显示控制器 330 协调以支持可在显示屏 310 上本地地绘制和显示的所有类型的视觉数据表示。没有要求 RT 包括显示处理块的任何特别组合。额外的瘦 RT 可小到只包括显示控制器 330,其中 CPU 进行显示处理,尽管具有至少一类解码器或绘图引擎是更为可能的。

[0054] 各种处理元件可对表示整个帧的分组或表示显示数据的各个子帧的分组进行解码。在一较佳实施方式中,分组包括与显示的固定位置相对应的经编码显示数据的各个数据片。CPU 324 可接收这些分组并通过读取分组的报头信息来确定分组预期的显示的适当位置。诸如数据解码器 326 的适当资源用于对数据片进行解码并将经解码数据传输到显示存储器内的适当位置。数据解码器 326 可被设立成将产生直接进入显示存储器位置的解码数据,或者该数据可被解码到另一区域且通过 2D 绘图引擎 332、CPU 324 或通过另一手段传输到期望位置。

[0055] RT 300 首先可通过将由该主机 100 经由网络 190/390 提供的额外信息引导到本地闪存 (未示出) 之外来初始化。在针对 RT 的初始化序列期间,RTSC 314 与显示屏 310 之间的连接可以反向或双向模式使用,从而利用诸如显示数据信道 (DDC) 接口、扩展显示标识数据 (EDID) 及其它扩展的标准来标识显示监视器的性能。通过键盘、鼠标和 I/O 控制器 316 的 USB 连接还可以用于与显示屏 310 的连接。然后诸如可用分辨率和控制的信息由 CPU 324 处理。系统 300 可实现诸如 uPnP 的协议或能与主机 100 通信的其它发现机制。在初始化通信期间,CPU 324 可将包括显示监视器信息的 RT 信息提供给主机 100,从而每个 RT 可在主机一侧被例示。

[0056] 初始显示屏可来自闪存或来自主机 100。在显示数据的第一全帧之后,该主机 200 只需经由网络 390 发送部分帧信息作为显示更新网络流的一部分。如果与先前帧相比没有显示像素变化,则显示控制器 330 可用来自本地 RAM 存储器 312 的先前帧内容刷新显示屏 310。RTSC 314 的输出可使用诸如 HDCP 的协议加密。如果显示屏 310 由诸如带有 DVI 的电缆连接,HDCP 可以是用于回放具有 DRM 的内容的要求。此外,RTSC 314 可被设计成提供各种加密密钥永远不暴露于芯片之外的高度安全的处理环境。在一更安全实现中,被解密的内容数据不会在该芯片的外部呈现为明码。为了实现这种安全系统,RTSC 314 可以使用与用主机 100 和显示屏 310 执行的密钥交换无关的本地密码术方法。

[0057] 显示更新通过网络流发送,且可由经封装的 2D 绘图命令、3D 绘图命令、编码显示数据或编码视频数据组成。网络控制器 326 接收网络显示流且 CPU 324 根据封装报头确定

对该分组而言需要功能单元 332、334、326 和 328 中的哪一个。功能单元执行必要处理步骤来绘画或解码图像数据且用新图像更新 RAM 312 的适当区域。在下一刷新周期期间，显示控制器 330 将该更新帧用于显示屏 310。

[0058] 显示控制器 330 将来自 RAM 312 的电流图像帧的表示传输到显示器 310。通常，该图像将以准备好显示的格式存储于 RAM 312 中，但是如果 RAM 成本是个问题，则该图像或该图像的各个部分可以编码格式存储。外部 RAM 312 可由远程终端系统控制器 314 内的大缓冲器代替。显示控制器 330 还能将储存于 RAM 312 内的两个或更多个显示表面组合来合成一输出图像以供屏幕 310 显示图像。不同混合操作可与该合成一起执行。

[0059] CPU 324 与 TSB 400（图 4，以下讨论）通信以最佳地设立和管理针对 RT 的全部显示操作。初始设立可包括枚举 RTSC 314 中受支持的功能的类型、显示屏 310 的详细说明、可用于缓冲和高速缓存数据的 RAM 312 的量、由 2D 绘图引擎 332 支持的命令集、由 3D 绘图引擎 334 支持的命令集、由数据解码器 326 支持的格式、由视频解码器 328 支持的格式以及显示控制器 330 的性能。其它运行时的管理优化包括管理和高速缓存 RAM 31 中的显示位图以使得它们不需要被重新发送。

[0060] RT 300 的配置可包括基本数据解码体系结构或在更常用的基于 CPU 的平台上运行的软件，且可基于因特网浏览器体系结构。作为因特网浏览器体系结构的一部分，这些功能单元可由直接调用功能单元的特定浏览器、包括使用功能单元的“插件”或驱动器的浏览器使用，或者更标准的浏览器可使用担当中介的本地代理来确定哪些请求可本地完成，而哪些需要与主系统上的网络应用程序通信。“异步 Java 脚本语言和 XML”（AJAX）及其它派生物，诸如“异步闪存和可扩展标记语言”（AFLAX）是这种基于代理的技术，且可以使用 XHTML、通过 Java 脚本和“XMLHttpRequest”操作的文件目标模型的组合与该 web 服务器异步交换数据以改进在客户机上的用户接口。AJAX 可用来执行选择性的请求，其中客户机一侧的数据解码使用 RT 300 的功能块。

[0061] 来自基于 RT 300 的因特网浏览器的基于代理的选择性请求可以由主处理器刀片 200 和终端服务刀片 400 支持以进一步优化整个系统。附加的通信协议处理可以由 AJAX 代理执行。其它示例 AJAX 操作可以包括安全密钥和 DNS 查找的管理。在该 RT 上的基于 AJAX 的代理可将主处理器刀片 200 或 TSB 400 用作为应用服务器或 web 服务器的代理。基于 AJAX 的代理可以包括特殊通信机制，且作为分离代理与提供该代理的其它部分的主处理器刀片 200 或 TSB 400 一起操作。

[0062] 图 4 示出用于基于刀片的多用户计算机系统 100 的终端服务刀片（TSB）的第二较佳实施方式，虽然 TSB 400 另外可体现为通过网络附连到主处理器的独立计算机或设备。作为设备，TSB 400 可包括到服务器的显示输出的其它非网络连接，诸如 DVI 或其它电缆，且可包括本地 DVI 输出以保持对本地输出设备的支持。这种来自 TSB 400 的 DVI 输出可以是来自服务器的通过模式，或者当该 DVI 上的数据不适于简单通过操作时可以进行一解码步骤。TSB 400 可以独立于主处理器操作以作为多用户 RT 的代理服务器行。

[0063] TSB 400 包括 CPU 子系统 402、存储器 434、桥接控制器 404、带 RAM432 的本地 I/O 428 以及组成终端服务加速器和多用户图形处理器单元子系统（TSA-GPU）700 的组件。桥接控制器 404 可以与交换结构通过接口直接相连，或者可包括用于与交换结构 140 和 142 连接的结构接口 130。该 TSA-GPU 700 内的主要功能单元是多用户 GPU（MU-GPU）412 和终端

服务加速器 (TSA) 424, 这些主要功能单元具有关联 RAM 418 和 430 且可包括经由显示接口 220 的连接 222。在某些配置中, TSA-GPU 具有共享 RAM 且可照图 7 如下所述地被进一步集成。

[0064] 单个 TSB 可以对各自具有一个或更多个 CPU 的多处理器刀片 200 执行图形处理, CPU 各自具有一个或更多个处理器核, 且任何数目的虚拟机在 CPU 和处理器核的混合上运行。在一些多用户或多处理器操作系统中, 对 TSB 400 的请求可以协调方式管理或执行。例如, 在基于微软远程桌面协议 (RDP) 或 Citrix ICA 的系统的情况下, 这些显示命令已被初始化且处理器刀片 200 不会尝试直接访问图形子系统。其它多用户或多处理器操作系统可包括串行化对图形子系统的访问的协调装置, 并且与该操作系统组合的 TSB 400 可使该协调访问被适当映射以使单个 TSA 700 可支持针对一个或更多多个用户操作系统的多个显示。

[0065] 然而, 在虚拟机模式中, 不同虚拟机可能不知道彼此, 且每个虚拟机可假定它具有对专用图形子系统的完全访问权。在这种情况下, TSB 400 需要创建一虚拟抽象层来以有序方式满足来自多个虚拟机的异步命令和请求, 并适当地支持对 RT 的结果显示。TSB 400 可编写可在处理器刀片 200 的不同虚拟机上运行的不同操作系统的图形驱动器。然后 TSB 400 将来自不同操作系统的驱动器调用协调成经协调的多显示 TSA 700 操作。或者, 每个操作系统将利用标准图形驱动器且 TSB 400 需要有效地截取驱动器调用。然后所截取的驱动器调用被管理以在相干多用户模式中操作 TSA 700 并适当地管理针对每个虚拟机的显示。

[0066] 在某些系统中, TSB 400 的功能之一是将该处理器刀片 200 从针对每个 RT 的部分管理中卸载, 并加速部分卸载处理以使每个 RT 具有经改进的显示经验。卸载和加速支持的类型包括: 将图形处理封装为远程图形命令, 帮助确定在每个 RT 上高速缓存什么性能和位图以确定哪些图形命令是最适合的, 将需要传输到 RT 的比特图编码并封装, 以及最佳地管理多媒体比特流。或者, 在传输前, 浏览器和基于 RT 的代理可发起对主机 100 的更多具体请求, 以提供对可由 TSB 400 编码的图形数据的更新。在一种较佳实施方式中, 主系统 100 是 web 服务器, 其中处理器刀片 200 执行 web 服务器功能的数据库和“后端”操作, 而 TSB 400 作为管理用于 web 服务器功能的显示元件的媒体协处理器进行操作。

[0067] 也可支持附加功能, 诸如可扩展标记语言 (XML) 话务、简单对象访问协议 (SOAP)、HTTP 话务、Java 虚拟机 (JVM) 以及其它与基于因特网通信相关联的话务的检查和封装。当执行任何想要的防兜售信息、防病毒、内容过滤、访问限制实施或其它基于分组过滤的算法时, 处理器刀片 200 与 TSB 400 一起可使 RT 有效地执行对整个因特网的远程访问。这种附加功能可对支持将主机用作 web 内容访问的代理的 RT 因特网浏览特别有用。虽然可能在系统中有一些冗余, 此方法可比在主系统与 WAN 之间使用的更普通的网络安全装置提供更多的特定用户控制。TSB 400 可提供诸如 DNS 查找的其它卸载功能。提供 DNS 查找可使需要设立和管理的 TCP/IP 连接的数量下降。TSB 400 还可提供针对诸如 SSL 的安全性的卸载。此外, TSB 400 可提供对安全客户机的基于证书的密码算法支持。

[0068] TSB 400 上的具体代理服务器可将其增强用于基于因特网的话务, 这些增强可包括取决于 RT 显示设备和该 RT 内的执行性能对基于因特网的内容重定格式或重新编码。例如, 如果 RT 设备是具有有限屏幕分辨率的蜂窝式电话或个人数字化助理 (PDA), 则 TSB 400 为了更快和更适当的显示可将高分辨率内容过滤降低为低分辨率图像。该 TSB 400 可

以运行其它更智能的内容过滤和网页解释算法,以执行类似除去标题广告和其它外来信息的功能以使核心信息可被发送到蜂窝式电话。TSB 400 可运行完全 web 浏览器且 RT 运行能力较差的 web 浏览器或微浏览器。TSB 400 的代理功能可将先进 web 格式转化为可被 RT 的浏览器理解的 web 格式。

[0069] 其它类型的 web 内容,诸如那些利用 Active-X 控件、多媒体闪存或其它运行时程序的 web 内容,可与诸如电话或 PDA 的设备不兼容。TSB 400 可以担当中间的代理服务器,且将在 Active-X 控件之后的显示数据传输到等待的 PDA 中。也可以执行应用层正则表达式 (RegEx) 内容处理。还可以执行重定格式和重新编码以增加客户机的安全性。尽管 XML 和 SOAP 可受到黑客或其它格式的病毒传播,TSB 400 可以将 XML 和 SOAP 重新编码成安全的显示格式,以使 RT 客户机不会受到这种危险。

[0070] 在另一实施方式中,主机 100 尤其是 TSB 400 被用作更普通的多格式网络文件代理服务器使得 RT 能查看它们以其它方式无法读取的文件。例如,RT 可以包括用于各种显示格式的查看程序,但可能不包括打开和查看 Adobe PDF 文件或微软 Word 文件的能力。RT 上的查看程序可以是支持各种 HTML 和其它面向 web 格式的浏览器。主机 100 可能能够打开和查看 Adobe PDF 文件和微软 Word 文件。然后该主机 100 可使用 TSB 400 的功能将来自 PDF 文件或 Word 文件的图形输出转化成用于在 RT 上查看的兼容显示格式。该 RT 可具有到许多类型的数据文件的网络链路,并通过使用多格式代理查看程序具有查看许多类型的文件的能力,这些类型的文件用其它方式不能被解码以供在该 RT 本身上查看。此类型的多格式代理查看可与网络文件共享功能组合或与邮件服务器组合。例如,如果该多格式代理查看程序将文件附件转换为蜂窝式电话 RT 可查看的格式,那么它可能能够接收带有各种附件的电子邮件。这些可查看附件可与邮件消息一起被包括在内或到附件的可查看版本的链路可被包括在该邮件消息中。

[0071] 多媒体比特流可包括已经处于压缩格式、且在图 2 处理器刀片 200 或图 4 TSB 400 上接收的视频流。在某些配置中,该多媒体比特流将已经是与该预期 RT 300 兼容的格式。在这种情况下,在处理器刀片 200 上运行的软件跟踪层将指引该比特流进入 TSB 400,该 TSB 400 将把该比特流封装为适当的分组格式以供传输到 RT。封装可包括添加报头信息,例如视频显示窗口的来源,或修改分组构成,诸如将传输流转换为具有不同分组大小的程序流。

[0072] 在一些其它情况下,该多媒体比特流将不是目标 RT 容易处理的格式或不是适于该网络连接的格式。在这种情况下,TSB 400 作为转码代理服务器操作,并执行更复杂的对该多媒体比特流的解码和再编码、转码或译码的步骤。例如,该导入多媒体比特流可以是经编码的 HDTV MPEG-2 流。如果在 RT 上窗口尺寸被设为小的  $320 \times 240$  窗口,则节约网络带宽并使 TSB400 将视频转码和转换为表示期望显示窗口尺寸的较低比特率可以是有意义的。类似地,如果该导入视频是 RT 不能解码的格式,则 TSB 400 可将该视频转码为兼容格式。即使该格式是兼容的,其它诸如数字权限管理 (DRM) 或加密方案的不兼容性可能会存在。TSB 400 还可以从一个 DRM 或加密方案转换为适于目标 RT 的方案。

[0073] 例如,内容拥有者可使用基于专用密钥交换的 DRM 方案,例如由 Apple 的 iTunes 所使用的一个。TSB 400 可检查在处理器刀片 200 或本地 CPU 402 上运行的 iTunes 视频播放器的输出,并捕捉经解密的内容。如果经解码的输出来自 DVI 视频总线,则该输出仍然可

在 HDCP 中加密。在 HDCP 的情况下, TSB 400 需要具有适当的密钥来解码 HDCP 保护内容并且可担当显示设备以便于执行该解密。为了保留该内容拥有者的控制权, TSB 400 然后将把该内容重新加密为可被远程客户机回放设备所理解的协议。这可基于商用协议, 诸如数字传输内容保护 (DTCP) 或微软的当前格式或者可使用一专用协议。一旦被重新加密, 该新的受保护数据流可经由网络传输到具有适当的解密和显示能力的接收设备。

[0074] 微软远程桌面协议 (RDP) 的一些当前版本提供对经压缩的视频比特流的不那么有效的处理。有了 RDP, 该主系统内的驱动器检测比特流且将该比特流解码为设备无关位图 (DIB)。然后该 DIB 被转换为 RDP 传输命令, 且该 DIB 格式数据经由网络被不可靠地传输到 RT。在多数情况下, 只有几个 DIB 数据帧将它传到 RT 用于显示。因此, 在该主 CPU 中执行解码以及经由网络以较低效率格式发送经解码数据中存在低效率。其它基于 RDP 的图形处理也使用 DIB。

[0075] 常规的图形位图, 例如那些来自网站的位图, 也需要从主机 100 传输到 RT 300。TSB 400 可对常规的诸如 DIB 的图形位图执行各种级别的编码。图形位图的编码关于提供原始图形质量的视觉上不可区分的表示的目的可以是无损的或是有损的。TSB 400 的简化软件接口可仅仅包括通过 RDP API 与该主 CPU 连接, 然而一种更为激进的实现是允许 TSB 400 访问下层的 DirectX 驱动器框架。该经编码的 DIB 传输和具体的经压缩视频域传输不是标准 RDP 实现的一部分。因此这些传输可被借道为现存 RDP 传输格式, 作为某类专用 RDP 扩展操作或在该 RDP 框架之外操作。

[0076] 该主操作系统和 RDP 的某些版本需要满足针对该 RDP 协议的附加安全性需求。该 RDP 客户机可被要求与主机交换密钥以便于使用加密分组。因为 TSB 400 正在截取 RDP 客户机分组, 所以 TSB 400 可以包括供密钥交换的适当加速和卸载以及供与主处理器通信的解密。此外, 为了维持该系统的安全性, TSB 400 和网络接口将确保与 RT 的所有通信都被适当加密。

[0077] 在另一实施方式中, TSB 400 可作为 web 服务器卸载引擎操作并支持浏览器和基于代理的 RT, 该 web 服务器卸载引擎作为 web 服务器的一部分或作为多个 web 服务器的集结点。web 服务器加速可为 JAVA、数据编码、数据转码及其它功能执行。在一更佳实施方式中, RT 可运行包括对诸如 AJAX 的多个协议的支持的更为智能的基于代理的浏览器。使用 AJAX 或类似方法的浏览器在用户操作期间能够维持和管理该浏览器内的信息, 且仅当需要新信息时才必须与 web 服务器接触。代替服务器传递整个帧, AJAX 使更具体的信息请求能执行, 并且 web 服务器只提供该新信息, 该新信息然后与先前存储的信息一起用于本地地生成新帧。此连续性可通过预取信息、请求更小的更有效的请求以及管理安全性用于改进该用户接口。由 AJAX 生成的请求可由 TSB 400 管理, 其中安全性可被维持, 数据被 TSB400 有效编码, 并且数据使用 AJAX 和 RT 300 的功能块在客户机一侧解码。

[0078] 精心设计的 AJAXweb 应用程序将使用客户机或代理高速缓存对象且减轻发送整个帧的需要的能力, 其中 TSB 400 仍然可用于对任何帧数据的第一请求进行编码。对于未针对选择性更新设计的 web 应用程序, TSB 400 可用作在这种 web 应用程序和基于 AJAX 的浏览器之间的代理。主机 100 可以不是 web 服务器, 并且可以是位于因特网服务供应商 (ISP) 或入网点 (POP) 上的服务器。作为代理, 该主机 100 或 TSB 400 可与 RT300 协调以执行基于客户机驱动器请求的选择性更新。即使该 web 应用程序请求全部帧更新, TSB 400

代理可处理新帧和旧帧并将两者进行比较，并向客户机提供经编码的更新信息。甚至对于并不是为选择性更新具体设计的 web 应用程序，这也可允许针对 RT 300 的带宽减少和经改进用户接口。

[0079] TSB 400 可通过可编程方案实现，该可编程方案也可解决若干无关操作的普通卸载任务。服务器可从卸载网络、存储器、安全性及其它任务中受益。卸载处理器可被设计成静态或动态地平衡各个卸载任务并针对任何给定工作负荷加速整体系统吞吐量。例如，该服务器可以在白天对瘦客户机执行基于服务器的计算而在晚上运行大的数据库操作。在白天，该卸载引擎将运行对 TSA 描述的操作。在晚上该卸载引擎将运行 iSCSI 加速，以从磁盘存储器系统访问大的数据库。该灵活性可由跟踪各种工作负荷的板上或系统范围的管理程序管理。用于卸载任务之间切换的粒度可以是极小的。该卸载引擎可被设计成执行非常快的环境切换，从而在单个会话内它可对该同一会话执行网络、终端服务、存储、安全性或其它卸载任务。

[0080] 为支持针对不同卸载任务的动态处理，TSB 400 使用可编程和可配置的、且在工作负荷变化时可被快速地任务切换和重新配置的处理块。各种存储器块将被包含于各个处理块中，并且大存储器 434 也可被包括在内。该 CPU 402 是包括其自身的高速缓存的通常可编程的处理器，并且可以为该卸载执行内务处理和管理以及执行某些高级协议和接口处理。桥接控制器 404 可集成网络处理器并管理 TSB 400 的交换结构接口 130 功能，且可管理同时通信的多个管道。诸如内容地址存储器 (CAM) 以及传统存储器的特定内存也可被包括在桥接控制器 404 之内。

[0081] 桥接控制器 404 和 TSA-GPU 700 的处理单元可被实现为可配置数据处理器 (CDP)，这些可配置数据处理器 (CDP) 被设计成容易重新配置以对通常与专用硬件块相关联的吞吐量执行不同处理。通过使用 CDP 而不是专用硬件，不同卸载任务可由同一硬件执行。用于设计 CDP 的现有技术方法，诸如可重新配置的数据路径、动态指令集、超长指令字 (VLIW)、单指令多数据 (SIMD)、多指令多数据 (MIMD)、数字信号处理 (DSP) 以及其它形式的可重新配置计算可被组合以执行极高性能计算。CDP 还可用作针对密码及密钥相关功能的带有或不带附加的专用硬件块的安全性处理器。

[0082] 针对终端服务加速和代理服务器操作，CDP 可被配置成执行针对小块和矩形的数据编码，各种形式的对视频或数据的转码或转换、小块签名的生成和比较、以及其它关于 TSA-GPU 700 的如下描述的任务。对于存储加速，CDP 可对 iSCSI、光纤信道 (FC)、光纤信号因特网协议 (FCIP) 以及因特网协议相关任务的不同方面进行配置。连接 490 可被配置成与 FC 或另一存储协议连接。对于因特网内容加速，CDP 可被配置成处理 XML 话务、SOAP、HTTP 话务、JVM 以及与基于因特网通信相关联的其它话务。其它 web 服务器加速可包括卸载 DNS 查找和处理 TCP/IP 连接，以及为传统的 web 客户机和基于 AJAX 类代理的浏览器执行数据编码和转码。

[0083] TSA-GPU 700 执行 TSB 400 的图形相关卸载和加速功能。多用户图形处理单元 (MU-GPU) 412 包括通过分组对选择性显示更新的支持且可遵守部分或全部所提出的 VESA 数字分组视频链路 (DPVL) 标准，尽管一较佳实施方式包括增强性能。TSA 424 支持通过系统总线 406 来自 MU-GPU 412 的分组显示更新，或优选地支持输入路径 414 和 416，输入路径 414 和 416 可以是串行的数字视频输出 SDV01 和 SDV02 或具有不同总线宽度、信令协议和

频率的一般化端口。示例包括数字视频输出 (DVO)、数字视觉接口 (DVI)、高清晰度多媒体接口 (HDMI)、显示端口或其它低压差分信令 (LVDS)、转换最小化差分信令 (TMDS)、快速 PCI 或另一方案。显示输出路径可全速运行以在高刷新速率下输出多个视频帧，其中这些帧可以是对应于一个以上目标 RT 的选择性更新矩形。TSA 424 可经由专用链路 426、通过主系统总线 4406 或通过片上系统 (SOC) 实现更紧密地结合，与本地 I/O 或网络控制器连接。第二系统总线 408 还可针对附加带宽而被包括在内，并且更直接地支持桥接控制器 404 及其与多个交换结构的接口连接。

[0084] 除执行传统图形处理之外，MU-GPU 412 产生基于显示的选择性更新，这些基于显示的选择性更新指示显示的哪一部分已经改变。选择性更新可以采取经由视频输出路径 414 或 416 或经由主要系统总线 406 输出的矩形或小块的格式。矩形更新包括指示窗口的起源、尺寸和格式的分组报头。起源可用于指示哪个 RT 是目的地。小块也可被使用并且可被标准化为一个或更多个的固定尺寸，以使该报头可需要更少的信息来描述该小块。其它信息，例如矩形或小块在 RT 上是否和如何按比例缩放，也可被包括于该报头内。其它形式的选择性更新包括对位块传送、区域填充和图形填充的支持，其中针对要在 RT 上执行的操作，最小数据量与命令参数一起发送，而不是发送大的数据块。其它报头支持视频流、同步锁相、缩放视频流、伽玛表和帧缓冲器控制的形式的更新。其它增强的和复杂的命令也可以选择性更新的形式置入 RT。所提出的 DPVL 规范详述针对选择性更新连同它们的报头一起的一种可能实现。

[0085] 一个 MU-GPU 412 可由用于所有 RT 300 的系统通过将 RAM 418 组织进各自包含多个 RT 的显示数据的各个显示表面被有效地虚拟化。MU-GPU412 的 2D、3D 和视频图形处理器（未示出）被优选地用来实现较高的图形和视频性能。图形处理单元可以包括 2D 图形、3D 图形、视频编码、视频解码、缩放、视频处理及其它先进的像素处理。MU-GPU 412 的显示控制器还可以执行诸如视频和图形数据的混合和键控的功能，以及整屏刷新操作。除了用于主次显示表面的 RAM 418，还有充足的用以支持各种 3D 和视频操作的屏外存储器。作为管理选择性更新的 DPVL 方法的另一选择，RAM 418 内可以有选择性更新的缓冲存储器（S 缓冲器）404。在一个实施方式中，S 缓冲器 404 存储状态位、签名或对应于每个虚拟显示的每个小块的状态位和签名。在另一实施方式中，S 缓冲器 404 存储带或不带标题、状态位和签名信息的小块本身，其中该小块被安排为选择性更新而输出。

[0086] 图形引擎和显示控制器通常将合成与每个 RT 显示器的主表面相对应的完整显示图像。RAM 418 将有效地包含用于所有 RT 的显示帧的阵列。例如，为了该显示，存储器可被配置为  $16K \times 16K$  像素的虚拟显示。在这种示例应用中， $1K \times 1K$  的 256RT 显示可被映射为  $16K \times 16K$  阵列。类似地，如果每个 RT 都是具有  $256K \times 128K$  显示器的蜂窝式电话，则 8192 个虚拟显示可被映射到  $16K \times 16K$  显示区。附加的屏外和暂时存储器也可类似地被包含在内。因为本申请涉及多个独立的 RT，所以 MU-GPU 412 可添加不同的安全性特征以使不同显示区域安全并防止一个用户获得对另一用户的帧缓冲器的访问权。出于安全性和可靠性考虑，系统将优选地包括硬件锁，该硬件锁防止对显示存储器的受保护部分的未获授权访问。

[0087] 图 5 示出图 4 存储器 418 的一示例配置，其中虚拟显示空间被设为水平方向为 3200 像素而垂直方向为 4800 像素。存储器 418 被分为八个标示为 520、522、524、526、528、530、532 和 534 的  $1600 \times 1200$  显示区。典型的高质量显示模式被配置为每像素 24 比特的

比特深度，尽管配置常常为了更容易对准和额外八个比特在显示器被图形和视频处理器访问时针对其它用途的潜在使用，可使用如在 RAM 418 中组织的每像素 32 比特。该分块存储器的例示作为来自 MU-GPU 412 的视图在本质上是概念性的。实际的 RAM 寻址还将涉及存储器页面尺寸和其它因素。

[0088] 图 5 在显示区 528 进一步示出显示更新矩形 550。 $1600 \times 1200$  显示的虚线 540 对应于称作为分区 (precinct) 的  $256 \times 256$  像素的更粗糙的块界。如从显示窗口 550 显而易见，显示窗口边界的对准并不必与分区边界对直。通常的情况是用户将任意地确定显示屏窗口的尺寸并将其定位。为了支持不要求更新整个帧的远程屏幕更新，受显示窗口 550 影响的每个分区需要被更新。此外，显示窗口 550 内的数据类型和周围的显示像素可以是完全不同的类型并且没有关联。同样地，基于分区的编码算法如果是有损的，则需要确保没有与分区的边缘或显示窗口 550 的边界相关联的视觉伪像。实际编码过程可在比分区小的块上，例如  $8 \times 8$  或  $16 \times 16$  上进行。因此，一较佳实施方式使用确定性编码算法，其中对一组像素生成相同的结果而不管其周围像素如何，并且没有伪像将因该窗口的任意对准而产生。

[0089] 用于该编码方案的块界也是关于小块的考虑因素。例如，编码方案可以要求块界是 8 个像素的倍数。如果该源小块不是 8 像素的倍数，则它将需要用周围数据填补。在另一情况下，较佳地常常是使该块界定向到屏幕，而不是特殊的用户设置的矩形或小块。如果用户操纵  $80 \times 80$  像素的窗口，则即使它在理论上可被放置成在水平方向和垂直方向上各自使用最少的 10 个  $8 \times 8$  块（总共一百个块），更可能的也是在每个方向上跨十一个块（121 块）。该矩形更新和该矩形的任何顺序编码将因此在部分周围像素需要被填补的地方编码  $88 \times 88$  像素（121 块）。

[0090] 也可支持带有不同尺寸显示器的 RT。在一个示例中，MU-GPU 412 可以支持任意数目的任意尺寸显示器。在另一示例中，支持小显示器作为子窗口或支持大显示器作为跨越一个以上显示区的重叠窗口可能更容易些。如矩形 536 所勾勒的， $1920 \times 1080$  窗口需要使用 532 和 534 个区域。尽管这样会浪费区域，但可能比创建每个显示器的定制尺寸简单些。由于 MU-GPU 412 的选择性矩形更新机制，只有该屏幕的相关区域将被传输。当 DVPL 动态地控制 CRTC 控制寄存器以管理选择性更新时，可实现其它需要更少处理器干预和改进系统效率的更灵活的机制，例如 S 缓冲器。

[0091] 更灵活的系统还可将矩形分成更多的常规尺寸实体，诸如小块。具有任意矩形尺寸的报头信息的效率与使用较不灵活小块尺寸但更多屏幕数据的可能更简单的报头之间有折衷。在一较佳实施方式中，小块可被动态地设置成块尺寸的任意倍，其中块尺寸是该数据编码算法的最小实体。这些块可被定向为源图像或屏幕的固定块位置。块的尺寸将被包括于报头信息内。

[0092] 诸如 530 的存储区可被指定为用于管理选择性更新的 S 缓冲器 404。在一实施方式中，S 缓冲器包括与显示帧 520、522、524 和 526 的小块相对应的状态位，其中状态位指示小块是否需要选择性更新。该 S 缓冲器 404 还可存储每个小块的签名，该签名然后用来确定对选择性更新的需要。在另一实施方式中，来自帧 520、522、524 和 526 的需要选择性更新的小块被复制到存储区 530，且排队以供选择性更新输出。这些排队小块可包括各个报头、状态和签名信息。

[0093] 图 6A 示出具有称为 1080P 的高清晰度电视 (HDTV) 分辨率  $1920 \times 1080$  的图 5 显

示图 536 的更详细视图。在图 6A 中,固定尺寸矩形 614 用屏幕位置边界定向。每个矩形是 160 像素宽和 120 像素高。每行有 12 个矩形 ( $12 \times 160 = 1920$ ) 而每列有 9 个矩形 ( $9 \times 120 = 1080$ )。系统可将这些矩形用作为构成选择性更新的基础的小块。在图 6B 中,另一系统进一步将矩形 614 分成包含  $80 \times 40$  像素的小块 620,且系统可选择这些小的小块作为选择性更新的基础。更灵活的系统可以利用由六个小块 620 组成的较大矩形 614 和小块本身,并且使用报头信息来描绘在任何给定时间输出哪种类型。

[0094] 在这两种情况下,构成编码算法的基础的块装入小块或矩形内。假设  $8 \times 8$  块,则每个小块具有  $10 \times 5$  块的配置而每个矩形具有  $20 \times 15$  块的配置。既使用较大矩形又使用较小三角形的系统可使用不同机制来各自确定选择性更新要求。在一较佳实施方式中,较大矩形可具有指示它们是否已经改变的相关联状态位,并且较小小块可将签名用于做这种决定。状态位和签名可如下所述用 S 缓冲器进行管理。

[0095] MU-GPU 412 可结合该处理来直接执行小块的选择性编码,或者每个小块可以使用选择性更新过程检验并输出到 TSA 424,且将包括适当的报头。该报头将由 TSA 424 处理,并且基于报头内的字段,TSA 424 将知道该小块预期哪个 RT 以及在该显示屏上的哪处。在适合时,TSA 424 将把该小块编码为被压缩的格式,调节任何所需报头信息且提供小块和报头以供进一步网络处理。

[0096] MU-GPU 412 和 TSA 424 可以不同地划分该选择性更新过程。在一些情况下,MU-GPU 412 可执行完整的管理且将只发送需要更新到 TSA 424 的小块。在其它情况下,TSA 424 需要执行数据片的进一步滤波以确定哪些数据片真正需要更新。在 MU-GPU 412 内,选择性更新机制可以是硬接线的或需要 CPU 干预,且硬件可跨绘图引擎和选择性更新刷新引擎实现。小块的编码也可在 MU-GPU 412 或在 TSA 424 中执行。MU-GPU 412 还可以将 RT 的图形绘图命令经由数字化视频总线输出到 TSA 424,或者软件驱动器可以直接将命令提供到 TSA 424。

[0097] 针对该选择性小块更新,在第一实施方式中,S 缓冲器用于 MU-GPU412 具有绘图引擎和选择性更新刷新引擎的地方,绘图引擎管理每个小块的状态位,选择性更新刷新引擎在状态位管理针对每个小块的选择性显示更新时监视该状态位。类似于用在 3D 图形中的 Z 缓冲器,S 缓冲器可被实现为数据的分离存储器板。与 Z 缓冲器一样,经增强的 MU-GPU 412 的硬件绘图操作可以在没有附加命令的情况下将 S 缓冲器状态位更新。状态位然后由选择性更新硬件用来确定需要在 RT 上更新哪个小块。类似于显示控制器的刷新周期,该选择性更新硬件可周期性地遍历 S 缓冲器并读取状态位。基于状态位的状态,选择性更新硬件将忽略不需要被更新的小块,或者它将读取用于选择性更新的小块,将该小块与报头信息一起输出且相应地更新状态位。在效率较低的实现中,MU-GPU 412 可使用更多的传统图形绘图操作来生成 S 缓冲器。

[0098] 在另一不需要特定 S 缓冲器硬件的较佳实施方式中,MU-GPU 412 可管理需要更新的级联小块的选择性更新缓冲器。该选择性更新缓冲器可以在独立的存储区中构成。每次 MU-GPU 412 执行改变小块的操作时,它将随后把该小块复制到选择性更新缓冲器中。报头信息可被存储在每个小块的起始处且小块可被打包在一起。显示控制器被设立成使用选择性更新缓冲器,并且使用标准显示控制器输出操作经由刷新端口将其输出。MU-GPU412 可管理作为级联小块的环形缓冲器或链接缓冲器列表的一个或更多个缓冲器,并且提供基于

TSA 424 视作小块列表的 SDVO 输出的连续输出。各种方案可用于 MU-GPU 412 以仲裁列表中的放置优先权。此方法对于使用 MU-GPU 412 而言可能是最有效的，该 MU-GPU 412 具有较少的支持多个 RT 的特定硬件，且具有很少的或没有特殊的选择性更新硬件。

[0099] 在另一较佳实施方式中，TSA 424 与 MU-GPU 412 一起操作以判定在 RT 300 上哪个小块可能需要更新。MU-GPU 412 在每个小块基础上管理状态位的能力可能太困难，且可将小块聚合为大的小块或全虚拟的 RT 显示，并且只具有针对状态位的有限粒度。将大的小块减小为小的小块更新可基于每个小块的跟踪签名执行。该签名通常在第一次处理小块时生成且根据后续签名校验。签名可由 TSA 424 根据导入数据操作或与 MU-GPU 412 的选择性更新硬件一起操作来生成和处理。如果 TSA 424 对每个小块执行签名校验，则可节约到每个 RT 300 的网络带宽。如果 MU-GPU 412 执行签名校验，则节约经由视频路径到 TSA 424 的带宽。MU-GPU 412 能生成和管理对应于小块的签名的存储板，其中状态位可以是签名板或隔离板的一部分。此外，状态位和签名单位可在 RAM 高速缓存中得到管理且由 MU-GPU 412 用链表管理。

[0100] 取决于由主机 400 上的图形操作生成的图形命令的类型和 RT 300 的性能，命令可被封装并发送用于在 RT 上执行，或者命令可由 MU-GPU 412 在本地执行。在许多情况下，尽管命令被发送用于在 RT 上执行，但该命令也由 MU-GPU 412 在本地执行以保持虚拟显示的本地拷贝。理想地，任何作为冗余本地图形命令的结果而改变的小块将用状态位滤出，以防止不必要的小块更新分组被发送到 RT。通常需要较少的带宽来发送命令而不是经编码的小块，但这并不总是可能的。手动地管理选择性更新缓冲器的系统也会考虑正在被发送到 RT 的命令。将由在 RT 上执行的命令更新的小块理想地将不被 MU-GPU 412 放入选择性更新缓冲器。

[0101] 代理服务器或终端服务加速器的虚拟显示存储器可在变化反映到 RT 显示上之前更新。虽然它们与同一显示器相关，但由 MU-GPU 412 和 TSA 424 管理的小块与子帧的位置独立于用户界面操作进行定位。用户界面操作可导致小块内的或跨多个小块的显示变化。用户界面变化可由用户操作、AJAX 代理、浏览器或代理服务器发起并且导致对虚拟显示存储器的更新。小块中的更新或虚拟显示存储器的经更新子帧将在 RT 上反映且由 MU-GPU 412 和 TSA 424 管理。

[0102] 在另一示例中，RT 预期的图形命令由 TSA 424 处理且被分成经编码的数据传输和经修改的图形命令。例如，主系统可能想要执行来自屏外存储器或来自图形到屏上存储器 (pattern to on-screen memory) 的 BitBlt 操作。这可在 MU-GPU 412 子系统上容易地执行。然而，在 RT 上为该 BitBlt 请求的源数据不被高速缓存。因此为了能够发送该图命令，可能首先必需对源数据或图形进行编码、封装并发送到 RT，然后将经修改的图形命令封装并发送到 RT。此过程可由 TSA 424 卸载。尽管 DirectX 驱动器可通过 MU-GPU 412 灌进命令，该 MU-GPU 412 然后将命令输出到 TSA 424，但是对于 DirectX 驱动器而言，将命令直接传输到 TSA 424 常常是更有效的。

[0103] 图 7 示出一较佳实施方式，其中将多用户 GPU (MU-GPU) 412 与终端服务加速器 (TSA) 424 组合成为集成电路 (TSA-GPU-IC) 片上系统 (SOC) 710。该经组合的 TSA-GPU-IC 710 可包括在芯片上的或在芯片外作为子系统 700 的一部分的 RAM 736。TSA-GPU 700 可包括一个或更多个可能相似或不同的系统总线接口 406 和 407。该 TSA-GPU 700 包括 2D 引擎

720、3D 图形处理单元 (GPU) 722、用于像快速 PCI 的各种系统总线的系统总线接口 732 和可包括针对视频的接口的本地 I/O 410 和其它本地 I/O 的控件。此外，SOC 710 可以包括视频压缩机 724 和视频解压机 726 硬件的某组合，或者某种形式的将那些及其它视频相关功能相组合的可编程视频处理器 764。附加处理器 756 也可被包括在内。

[0104] 还被包括在内的是执行选择性更新的带显示控制器的多用户选择性更新 (SUC) 750 以及对所需的子帧或小块进行压缩的数据编码器 752。SUC750 可包括针对本地显示的输出 222 和 758，虽然远程多用户通过系统接口 732 或可能的与网络控制器的直接连接 426 也得到支持。系统总线 760 例示了在各个处理单元以及系统接口 732 和存储器接口 734 之间的连接。该系统总线 760 可包括各种形式的纵横交换、经仲裁传输，并且为了增强性能还可具有从一个单元到另一个的直接路径。

[0105] 在图 4 的多芯片 TSA-GPU 400 中，MU-GPU 412 经由 SDVO 路径 414 和 416 连接到 TSA 424，且 MU-GPU 和 TSA 各自具有它们自己的 RAM。相反在图 7 中，TSA-GPU-IC 710 使用共享的 RAM 736 而不是 SDVO 路径。使用 RAM 736 消除了使用 SDVO 路径来传输的需要且因而消除了 SDVO 带宽问题。此外，通过分享存储器，SUC 750 能够直接从存储器读取帧信息，从而消除通过 MU-GPU 412 对存储器的读取。

[0106] 若干附加的最优化可被包括在 TSA-GPU 700 的 SOC 710 内，诸如直接包括各个功能单元中的 S 缓冲器支持。此外，TSA-GPU 700 可被设计成映射对多个显示的支持，而不是用如图 5 示出的固定尺寸显示器来实现多用户帧支持，该多个显示被匹配到它们对应远程终端中的分辨率和色深度。将存储器内的显示更直接地与对应的远程显示系统匹配可为存储器使用实现更高的效率。

[0107] S 缓冲器支持可用于硬件或与跟踪软件层一起协助针对显示帧的编码选择，这些显示帧已经改变并需要生成选择性更新流。S 缓冲器支持可使用报头、状态位、签名或这三者的组合，并且可基于固定尺寸小块或可变尺寸小块。该 SUC 750 可利用 S 缓冲器支持来确定该显示的哪个子帧需要选择性更新。该报头信息对选择性更新的下游处理可能是有用的。

[0108] 不同系统可针对减小尺寸的报头或减少数目的选择性更新传输进行优化。例如，一较佳实施方式可利用包括一 64 位报头的基于小块的选择性更新系统。该报头的 16 位位可用于指示 64K 可能 RT 的哪一个是该选择性更新的预期接收者。另两位可用于指示有限数目的固定尺寸小块的哪一个被包括在该选择性更新中。这些位的另一段然后指定哪个小块编号将指示显示器上与该选择性更新对应的位置。然后报头之后是该选择性更新编码数据。用于指定优先权、误差校验和其它跟踪信息的附加位也可被包括在内。

[0109] 在另一较佳实施方式中，SUC 750 可不使用小块并且要求更大种类来用于该选择性更新分组。例如，系统可以要求 X 和 Y 偏移量来指定选择性更新矩形的起始位置，而不是提供数字来指定小块位置。然后选择性更新矩形用水平方向和垂直方向的像素或块的数目进行描述。虽然针对这种系统的报头信息可以更大，但执行对远程终端的更新的独特传输的数目可较少。

[0110] 取决于 RT 的性能和主机 100 与 RT 之间的网络特性，TSB 400 可选择将某些图形命令直接发送到 RT 而不是发送选择性更新。这些图形命令可由 TSB CPU 子系统 402 或由处理器 756 管理。这些处理器还可管理代理服务器或其它终端服务功能，包括对 DirectX

和 RDP 命令执行命令解释。作为解释器,处理器 756 或 CPU 子系统 402 对在主机系统上运行的软件驱动器进行卸载,以管理 2D 图形命令、3D 图形命令、视频流和其它窗口功能。解释器功能可与数据编码器 752 组合,以执行许多管理 RT 的计算密集方面,并且还可最优化从主机系统发送到各 RT 的命令、数据和视频流。代理服务器可在 TSB 400 和 RT 之间拆分。

[0111] 例如,各种图形 BitBlt、源到屏幕目的地的 BitBlt 及其它位图传输可得到增强。需要源数据、源图形或源位图的图形命令可通过数据编码器 752 将这些源编码成更有效的格式。经编码的源数据、源图形或源位图与经修改的图形命令一起被传输到 RT 300。目标 RT 将接收经编码源,将其解码,然后在接收经修改图形命令之后执行预期的操作。经编码数据和经修改命令的传输可以用 RDP 传输或用 TSA-GPU 700 和 RT 300 所支持的类似 RDP 的传输。

[0112] 针对 TSB 400 中的视频流,DirectX 解释器可截取和卸载视频流处理并向目标 RT 提供最佳流。卸载中的第一步是确定处理器刀片 200 不对多 CPU 复合体执行视频解码。基于主机的解码具有几个不利方面,最重要的两个方面是,第一,它占用了相当数量的 CPU 周期来执行实际的解码。第二,主机上具有经解码的视频帧并非必然是获得在目标 RT 上显示的帧的最佳方法。代替地,DirectX 解释器软件截取 DirectX 调用,在微软 Windows ® 的某些版本中 DirectX 调用可能需要使用 DirectShow 以获得对该视频流的访问权,尽管其仍然处于被压缩形式。为了使 RDP 继续正常操作,该 DirectX 解释器可能需要用模拟帧来欺骗 RDP 软件接口。

[0113] 同时,TSB 400 知道 RT 能够解码哪种视频流格式、从主机系统到 RT 名义上是哪种网络吞吐量,以及哪种分辨率和显示特征是该视频流想要的。有了此信息,TSB 400 设立 TSA-GPU 来处理导入视频流,从而为了网络、RT 和显示输出需要而产生理想流。这可能需要将一种编码格式代码转换为另一种,将一个比特率转换为另一个、改变帧速率、改变显示格式、改变分辨率或这些的某种组合。然后 TSB 400 封装经处理的位流且将它发送到合适的连接以供网络处理。

[0114] 有几种将代理服务器或运行在系统 100 上的 RDP 服务器软件在处理器刀片 200 和 TSB400 之间划分的方法。两种实施方式在此被详细考虑,第一种是“终止和再生”而第二种是“卸载和增强”。可使用各个实施方式的各个方面各个实施方式的变体也是可能的。

[0115] 在“终止和再生”情况下,RDP 客户机在 TSB 400 上运行。只要涉及在处理器刀片 200 上运行的 RDP 服务器,RDP 操作就被 TSB 400 终止。在此情况下,TSB 400 利用 TSA-GPU 700 来创建虚拟显示空间,从而通过创建单个大的显示图来支持多个虚拟 RT,在该单个大的显示图内每个用户是偏置的或其中每个虚拟显示被视为具有其自己映射的独立显示。RDP 客户机软件可能需要对要求安全客户机通信的 RDP 主机使用处理器刀片 200 与 TSB 400 之间的密钥交换和安全性处理。当 RDP 客户机接收来自 RDP 主机的命令时,该客户机利用 TSA-GPU 700 将显示帧呈现显示子系统。使用“终止和再生”,TSB 400 随后能够将它想要的任何方法和任何协议用于 TSB400 和 RT 之间的通信。

[0116] 在“终止和再生”操作的较佳实施方式中,TSB 400 被配置为各自与 RT 对应的多个 RDP 客户机。处理器刀片 200 使用交换结构来与“虚拟”RDP 客户机通信。然后 TSB 400 担当使用虚拟网络计算 (VNC) 的到 RT 的服务器。RT 与主机 100 之间的所有通信由 TSB 400 管理。来自 RT 的 VNC 键盘和鼠标命令由 TSB 400 译码为到处理器刀片 200 的 RDP 命令。

然后使用 VNC 的任一类客户机能够有效地与主机 100 通信,在该主机 100 中主处理器刀片 200 运行非 VNC 服务器。在“终止和再生”操作的第二较佳实施方式中,TSB 400 担当因特网服务器并与 RT 运行浏览器通信。因为不同平台上的不同浏览器可具有不同性能,所以 TSB 400 可支持不同 HTTP、XML、Java 及其它用于与基于浏览器的客户机通信的元数据协议。

[0117] 对于网络服务,“终止和再生”功能由在 TSB 400 上运行的代理服务器执行。处理器刀片 200 可运行 web 服务器或者可管理与位于其它地方的 web 服务器的连接。或者,TSB 400 可以直接与 web 服务器通信。作为代理服务器,TSB 400 将终止所有的网络操作。TSB 400 可包括代理,诸如连续向上游 web 服务器作出请求的 AJAX 代理。然后 TSB 400 直接与 RT 客户机通信。TSB 400 可使用 RT 所支持的任何协议并且可选择为 RT 通信安排完全独立的协议或可选择重新使用类似的基于 web 的协议。

[0118] 对于基于服务器计算的第二实施方式,“卸载和增强”保持处理器刀片 200 在客户机通信中的更多参与。处理器刀片 200 上的跟踪软件层仍然将 DirectX 视频、图形和数据流重定向到 TSB 400,该 TSB 400 完成 DirectX 调用的功能。卸载该功能使得多 CPU 复合体 202 对该多用户系统的其它用户可用。进一步处理可由 TSB 400 用对显示环境和允许最优化处理的网络带宽的理解完成。

[0119] 当图形命令被本地执行并转发到 RT 以供执行时,TSB 400 上的解释器软件还可以用于与 TSA-GPU 700 一起管理 S 缓冲器。TSA-GPU 700 本地执行该图形命令的原因是使帧缓冲器的当前拷贝可被管理以供未来使用。因为图形命令正在 RT 上执行,所以作为本地图形命令结果在主机上变化的小块不需要使选择性更新硬件发送经编码的小块。为了防止这个,RDP 跟踪软件需要计算哪些小块受图形命令影响。S 缓冲器中与这些小块对应的位状态位可被管理以使基于小块的选择性更新不被执行。

[0120] 跟踪和解释器软件还可用于辅助针对显示帧的编码选择,这些显示帧已经改变并且需要生成显示更新流。回想执行编码是为了减少远程显示系统 300 再生显示所需的数据。跟踪软件层可以帮助识别在帧内或小块内的数据类型以允许执行最优化类型的编码。某些 RT 可能不具有充足的图形处理能力来执行图形命令,并且可能被发送由 TSA-GPU 700 处理的经编码数据。

[0121] 例如,如果跟踪软件层识别到小块的表面是实时视频,则可对那些小块使用对视频更有效的编码方案,该编码方案具有平滑的空间转变和时间局部性。如果跟踪软件层识别到小块的表面主要是文本,则可使用对文本的清晰边缘和充足空白更有效的编码方案。识别哪类数据在哪个区域是一个复杂的问题。然而,跟踪软件层的实施方式允许接口进入主显示系统的图形驱动器和协助此识别的主操作系统体系结构。例如,在微软 Windows ® 中,使用特定 DirectShow 命令的表面可能是视频数据,而使用通常与文本相关联的色彩扩展位块传输(Bit Blits)的表面可能是文本。每个操作系统和图形驱动器体系结构将具有其自己的特征指示器。其它实现可并行地执行多种类型的数据编码,然后选择使用基于编码器反馈产生最佳结果的编码方案。

[0122] 此“卸载和增强”的第二实施方式还可与代理服务器体系结构一起用于基于 web 的服务。在这种实施方式中,TSB 400 可以直接与 web 服务器通信,或者可用作协调 RT 与 web 服务器之间的通信的分离服务器。在 TSB400 上运行的代理和 RT 可对各个 web 服务器作出持续请求。TSB 400 可高速缓存 web 服务器信息直到它真正被 RT 需要用于显示。如

上所述, TSB 400 可对各种图形和视频命令以及数据重新编码或重新格式化以更好地适合到 RT 的信道, 以及更好地匹配 RT 内可用的功能和性能。各种内容管理和内容格式操作可由作为代理或分离代理服务器操作的 TSB 400 来执行。

[0123] 对于各种数据类型, 一些类型的编码方案对具体类型的数据尤其有用, 且一些编码方案与数据不太相关。例如, RLE 对文本特别好而对视频特别差, 基于 DCT 的方案对视频非常好而对文本非常差, 且基于子波变换的方案可以既对视频又对文本好。尽管任何类型的无损或有损编码可用于此系统中, 也可以是无损或有损类型的子波变换编码, 特别是使用决定性算术编码器的渐进子波变换尤其非常适于此应用, 该决定性算术编码器可对每个小块进行编码而不用顾及周围小块。为更好的实时执行而调节处理的 JPEG 2000 子波编码器的派生物是一种可能的实现。

[0124] 图 8 是来自图 2 的 BMC 800 的更详细视图。该 BMC 800 包括 CPU 808, 它可能是简单的微控制器或者可能是带高速缓存器的更强大的 CPU。该 BMC 800 还包括安全性处理器 804、网络处理器和 MAC 控制器或“网络接口控制”(NIC) 806、RAM 218、接口控件 810 和某种形式的 TSA-GPU 700。

[0125] 该 CPU 808 是可“带外”(OOB) 操作的板上处理器, 该“带外”意味着主 CPU 不需要动态的软件或干预。理想的 BMC 800 不需要任何附加的电缆连接且允许处理器刀片 200 的完全远程管理。在某些情况下, BMC 800 可被形成为中心以执行对多个刀片的管理。此处示出的 BMC 800 包括对键盘、视频和鼠标 (KVM) 功能的支持。在某些系统中, CPU 808 可以由在主系统 CPU 上运行的虚拟机进一步增强。虽然不是完全地 OOB, 虚拟机可被设计以不干预其它系统功能。

[0126] 与 BMC 800 的通信可以为了安全性而使用安全性处理器 804 加密, 且可利用本地网络接口 214 或可通过总线 206 和经由适当的图 2 结构连接 140 或 142 传输分组。该网络接口可如图所示部分地包括 NIC 806, 该 NIC 806 包括 MAC 部分和位于任何地方的剩余物理接口 (PHY)。当与外部 PHY (未示出) 通过接口相连时, 接口 814 可以遵循诸如介质独立接口 (MII) 或减少的 MII (RMII) 的标准。该外部 PHY 可以是专用设备或另一网络子系统的一部分。

[0127] BMC 800 的显示方面可从简单到复杂地变化, 且可以支持一个或更多本地或远程用户。当简单系统能利用基本图形控制器和软件来对显示进行编码时, 该较佳实施方式利用图形加速器、选择性更新和对更新的编码的复杂组合以使远程用户具有全性能虚拟存在。这样, BMC 800 示出 TSA-GPU 700 的针对显示支持的用途。取决于同时用户的数量, 针对此类 BMC 800 应用的 TSA-GPU 700 的性能可能不需要与 TSB 400 中一样高, 尽管这些性能可以是相似的。

[0128] 除了支持远程访问的高级虚拟存在特征之外, 该板上处理器可为了各种平台自调系统程序控制通过接口与不同传感器连接, 这些平台自调系统程序控制诸如管理温度、电压、声音、传感器和发光二极管 (LED)。警报的中层远程监视也被监视和管理。为了与外部管理系统通信, BMC 800 可包括 web 服务器且可遵照各种行业成果以使诸如智能平台管理接口 (IPMI) 和活动服务器管理接口 (ASMI) 的远程管理标准化。对同时的多个用户和诸如 DVD 驱动器的虚拟 I/O 设备的支持也可被包括在内。也在 BMC 800 中示出的是接口控件 810, 其可直接与板上传感器、外部接口芯片 850 通过接口相连, 外部接口芯片 850 经由路径

802 与板上传感器通信、或经由路径 214 与另一本地 I/O 控制器通信。接口控件 810 可将 BMC 800 通过接口连接到诸如系统管理总线的另一总线，而不使用如在图 8 中示出的系统总线 206。

[0129] 图 9 是根据本发明的一个实施方式的用于执行终端服务加速和代理服务器过程的方法步骤的流程图。为了清楚起见，参考包括视频的显示数据对过程进行讨论。然而，与音频、键盘、鼠标和其它数据有关的过程是为了结合本发明使用而被等同构想的。最初，在步骤 910，多用户计算机 100 和远程终端系统 300 遵循各个过程来初始化和设立主机一侧和终端一侧以便于各种子系统启用每个 RT。在步骤 912，该应用程序以显示命令、显示数据更新或视频数据流的形式提供经更新的显示数据。应用程序更新可从应用程序本身、客户机上的用户动作、或应用程序、代理服务器或客户机的一些其它代理发起。应用请求可通过在主 CPU 上运行的跟踪软件层截取，或者命令可由在 TSB 400 上运行的代理服务器或终端服务加速器截取。对于步骤 912 中的 BMC 800，没有跟踪软件层被包括在内且 BMC 800 独立于主 CPU 操作。

[0130] 如果图形操作包括 2D 绘图，则在步骤 924 中，2D 绘图引擎 MU-GPU412 优选地将操作处理成 RAM 430 中的适当虚拟显示。类似地，在步骤 926，3D 绘图由 MU-GPU 412 执行为 RAM 中的适当虚拟显示。在步骤 928，TSB400 可确定视频或图形命令将被转发到适当的 RT。直到步骤 940 的流程可以不受支路步骤 928 影响。在步骤 940，MU-GPU 412 将每个虚拟显示合成到适于显示的帧中。此合成可用通过该 CPU 子系统 202、2D 引擎、3D 引擎和 GPU 412 内的任一视频处理元件的操作的任何组合执行。作为该合成步骤的一部分，对于图形操作硬件中包括 S 缓冲器管理的 MU-GPU 412，该绘图引擎为相应小块更新 S 缓冲器。

[0131] 如用返回路径 944 所示，TSB 400 可按需为同一 RT 或不同 RT 处理下一帧。在一较佳实施方式中，TSB 400 可运行整个客户机堆栈，从而只要涉及应用程序服务器，该客户机就完全完成绘图操作。类似地，TSB 400 可以担当代理服务器或分离代理服务器以完成与 webweb 服务器相关的 web 客户机操作。TSB 400 可运行客户机协议以仿真多个用户。一旦 TSB 400 已与服务器通信，则客户机侧操作就完成了，TSB 400 现在可使用任何机制来将预期的命令或显示数据中继到目标 RT。如之前所述地，TSB 400 可重新发送现有命令，重新制定经修改的命令或者使用一不同机制将该显示更新反映到 RT。

[0132] 作为截取命令的一部分，TSB 400 可在本地虚拟显示存储器中完全合成预期的 RT 显示屏。一旦合成操作被执行，步骤 946 就在适当时管理小块和相关联的 S 缓冲器状态位和签名位。步骤 946 考虑任何通过可影响该 S 缓冲器状态位的视频和图形支路步骤 928 处理的图形和视频操作。例如，如果绘图操作既在步骤 924 中执行又通过步骤 928 绕到远程终端，则不需要对受该绘图操作影响的小块执行选择性更新，因为操作将发生在 RT 上。

[0133] 有了在可发生在 MU-GPU 412 或与 TSA 424 的组合内的步骤 946 中处理的小块的状态位和签名，步骤 950 可执行小块的选择性更新。小块可具有固定的或变化的尺寸。与小块一块被包括在内的报头信息将指示格式以及预期 RT 目标。在步骤 954，TSA 424 对从步骤 950 接收的小块进行必要的编码。此编码优选地是一确定性方案，其中在该编码步骤中不需要考虑小块内数据的定向以及周围小块。同样在步骤 954 中，处理步骤 928 之后的视频数据和图形命令。视频数据可在比特率或帧速率被改变、在频域或空间域内按比例缩放、以及在需要时转码成不同的编码标准时被转换。经由返回路径 968 的网络反馈与 RT 信

息一起都可以帮助确定编码步骤 954。

[0134] 步骤 954 还执行任何需要附加处理的图形操作,该图形处理可能需要图形数据编码。在步骤 958, TSA 424 或者 CPU 402 对在先前步骤中处理的图形命令、数据传输或视频传输执行进一步的封装。网络反馈也在此步骤中相关于诸如带宽和等待时间以及具体分组尺寸和传输问题的网络特征进行考虑。在步骤 962, 经封装的分组通过适当的网络控制器进行处理且分组沿该网络被发送到适当的 RT 300。

[0135] 网络处理步骤 962 使用来自系统控件的信息。此信息可包括关于哪个远程显示器需要哪些帧更新流、哪种类型的网络传输协议被用于各个帧更新流,以及用于各个帧更新流的各个部分的是什么优先权和重试特征的信息。网络处理步骤 962 可利用本地 I/O 428 和本地网络连接 490 被本地管理到 TSA。或者,在基于刀片的系统中,该网络就绪分组可经由系统结构总线 140 或 142 中的一个传输,以供通过包括网络连接的处理器刀片 200 或位于另一处理器刀片上的网络处理器处理。各个网络可包括千兆比特以太网、10/100 以太网、输电线以太网、基于同轴电缆的以太网、基于电话线的以太网、或诸如 802.11a、b、g、n、s 的无线以太网标准和进一步的派生物。其它非以太网连接也是可能的且可包括 USB、1394a、1394b、1394c 或诸如超宽带 (UWB) 或 WiMAX 的其它无线协议。

[0136] 图 10 是根据本发明的一个实施方式用于执行网络接收和显示过程的方法步骤的流程图。为了清楚起见,该过程参考包括视频的显示数据进行讨论。然而,与音频和其它数据相关的过程是为了与本发明有关的用途而等同构想的。RT 300 可被配置成运行简单的控制程序以执行功能操作,可以是运行驱动器或应用程序的基于操作系统的处理器系统,或者可以是在某类客户机上运行的浏览器,该某类客户机可以或可以不包括 JAVA 处理或更多包括先进 AJAX 处理的代理。此外,RT 300 可基于导致显示更新的用户动作或代理操作发起请求。

[0137] 在图 10 的实施方式中,最初,在步骤 1012,远程终端 300 优选地经由路径 390 从主机 200 中接收网络传输。然后,在步骤 1014,网络控制器 336 优选地执行网络处理过程来执行网络协议以在无论传输是有线的或是无线的情况下接收所传输数据。

[0138] 在步骤 1020, CPU 324 解释导入传输以确定该传输预期给哪个功能单元。如果该导入传输是 2D 图形命令,则 CPU 324 将通过 2D 绘图引擎 332 初始化操作;如果是 3D 命令,则通过 3D 绘图引擎 334;如果是视频数据流,则通过视频解码器 328;以及如果是经编码的数据小块,则通过数据解码器 326。一些绘图命令可以利用绘图引擎和数据解码器 326。在某些情况下,导入的传输数据可被存储以供需要时使用。各种形式的 AJAX 和代理处理可对数据作出随机请求,该数据最终可以或可以不需要。

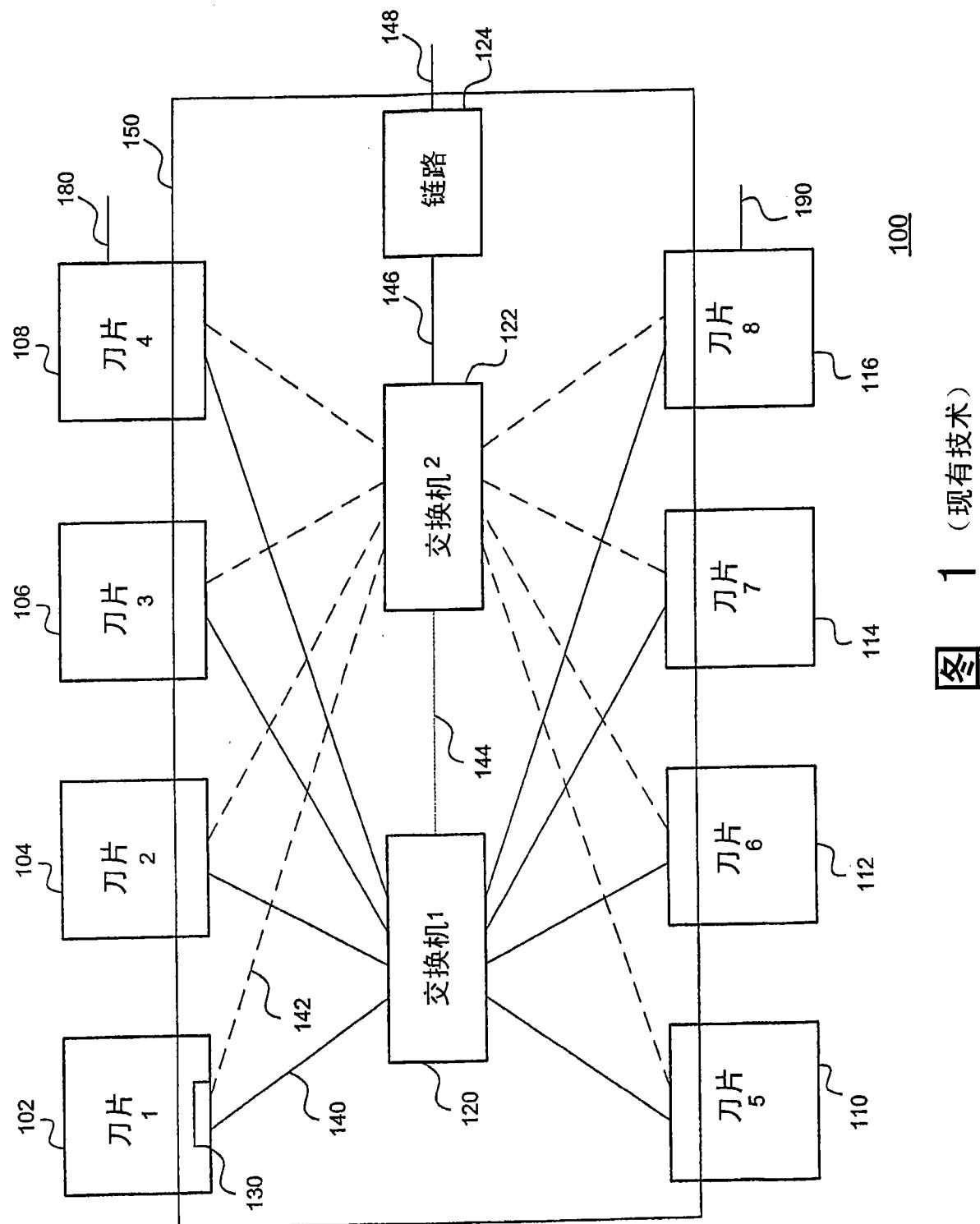
[0139] 可变数量的命令和数据传输可以进行,且各个功能单元操作数据信息,且优选地将其处理为适当地可显示格式。在步骤 1030,来自各个功能单元的经处理数据通过帧管理器 330 集合且可产生更新的显示帧以存入 RAM 312。该经更新的显示帧可包括来自先前帧的显示帧数据、经处理和解码的新帧数据、以及隐藏在该新帧数据传输期间发生的显示数据错误所需的任何处理。

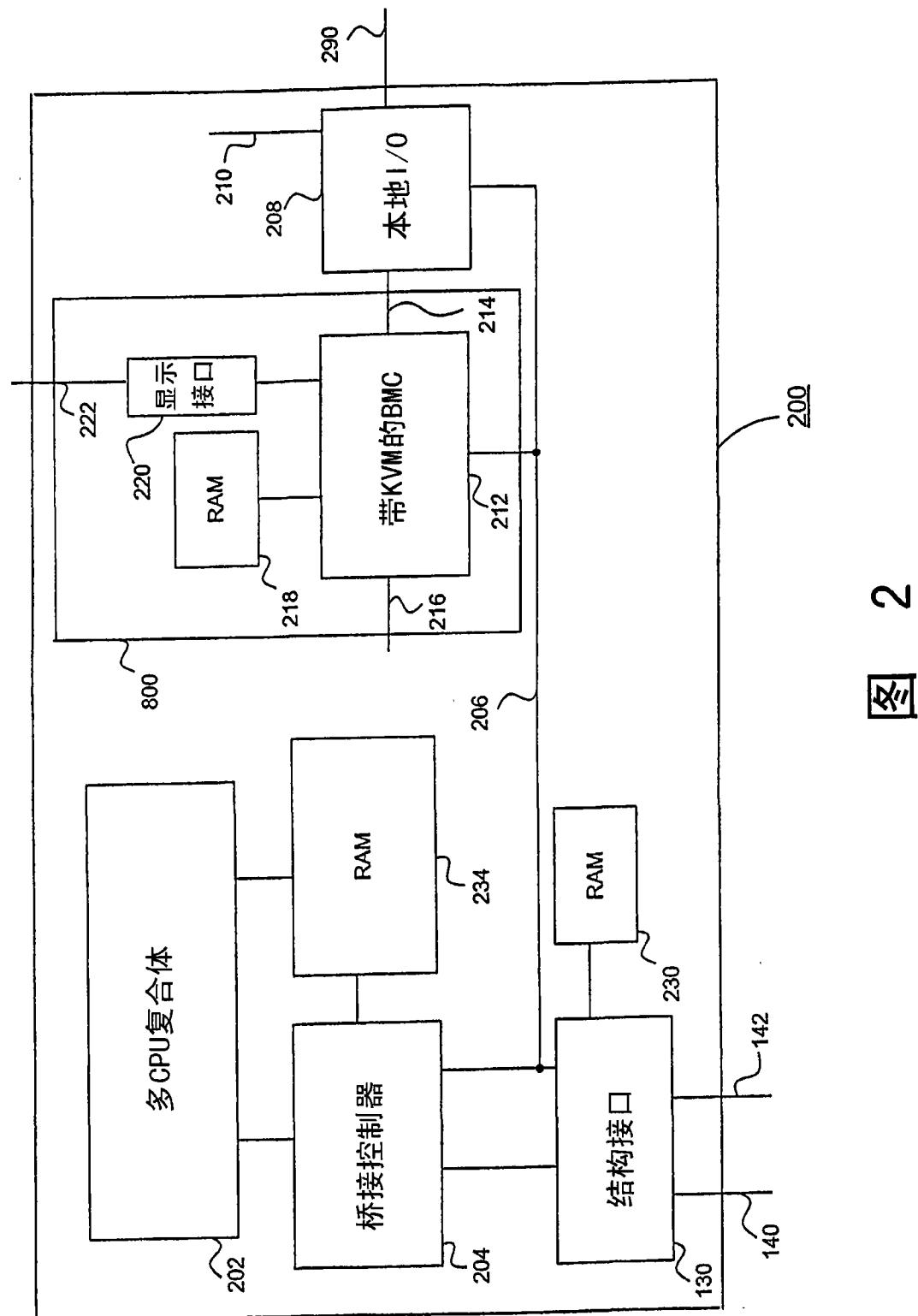
[0140] 最后,在步骤 1040,显示控制器 330 将最近完成的显示帧数据提供给远程终端显示屏 310 以供远程终端系统 300 的用户查看。显示刷新是在远程终端控制器 314 和显示 310 之间的通常是每秒 60 到 72 次操作以避免闪变的异步操作。步骤 1030 中产生新的显示

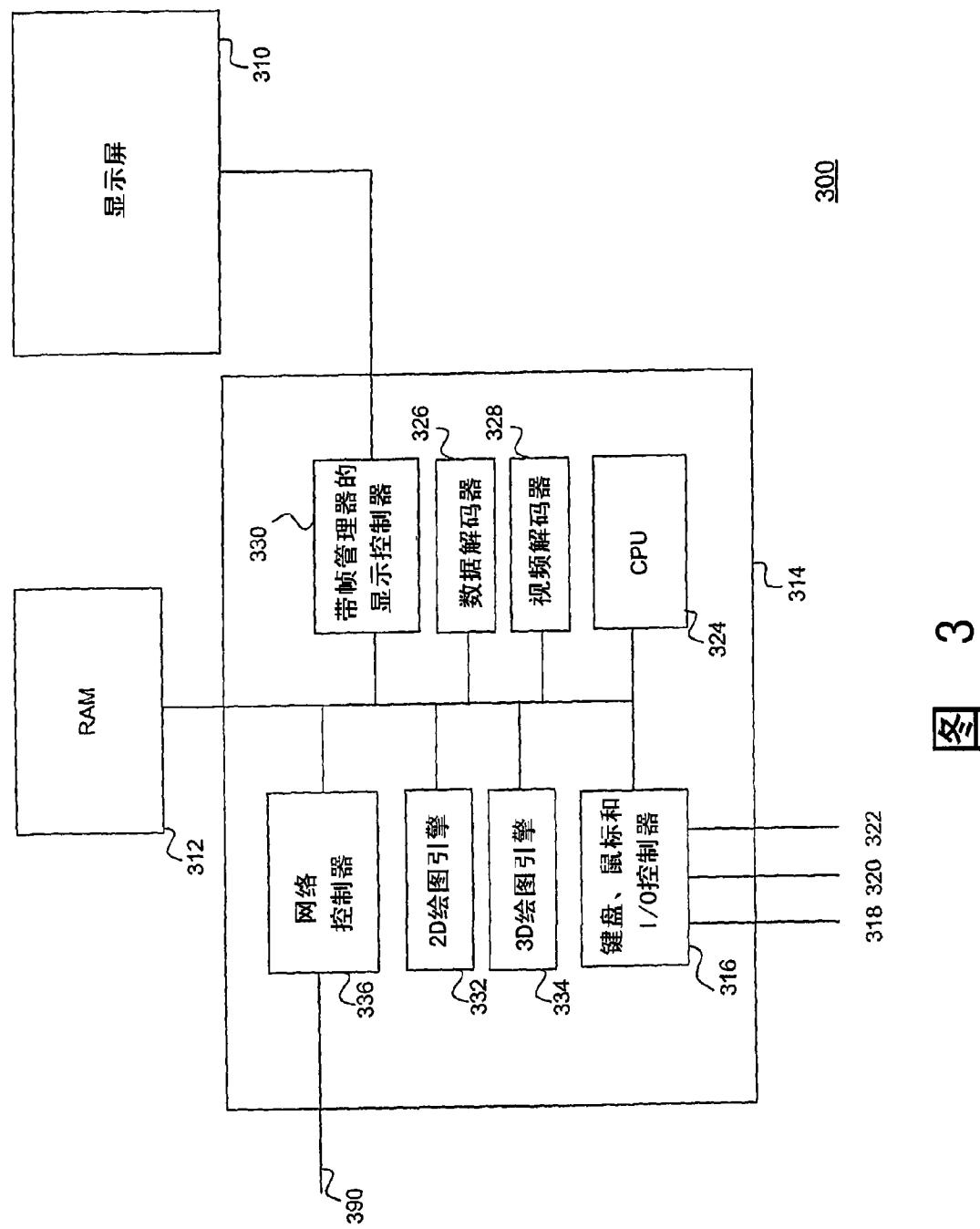
帧通常发生得少得多,尽管必要时可以每秒 30 帧或更多地发生。在缺乏屏幕保护程序或功率下降模式的情况下,该显示处理器在显示刷新过程中将继续用最近完成的显示帧对远程显示屏 310 进行更新,如由反馈路径 1050 所指示的。

[0141] 本发明因此实现基于多用户服务器的计算机系统,该系统支持远程终端从而用户可以有效地使用各种应用程序。例如,公司可以在一个位置使用多个机架的计算机系统,且向处于在远程位置的用户提供在他们桌面上的非常简单和低成本的终端系统 300。不同的远程位置可以基于 LAN、WAN 或通过其它连接进行支持。该 RT 可以是桌面个人计算机或笔记本个人计算机,或者在另一系统中可以是诸如手机、个人数字助理的专业设备或是与诸如便携式视频播放器、游戏机或远程控制系统的其它消费产品的组合。用户可以灵活地利用多用户系统 100 的主机以实现与主系统能够提供给本地用户的一同级别的软件兼容性和相似级别的性能。因此,本发明有效地执行灵活的多用户系统,该系统使用各种不同组件来便于最佳系统的互操作性和功能性。

[0142] 本发明参考一优选的实施方式在以上进行了说明。其它实施方式对本领域的技术人员在查看本公开后是显而易见的。例如,本发明可以使用除了在上述优选实施方式中描述的之外的配置来容易地实现。此外,本发明可有效地结合不同于在上述较佳实施方式中所描述的系统来有效地使用。因此,对较佳实施方式的这些及其它变体旨在由本发明涵盖,本发明只受所附权利要求限制。







3

冬

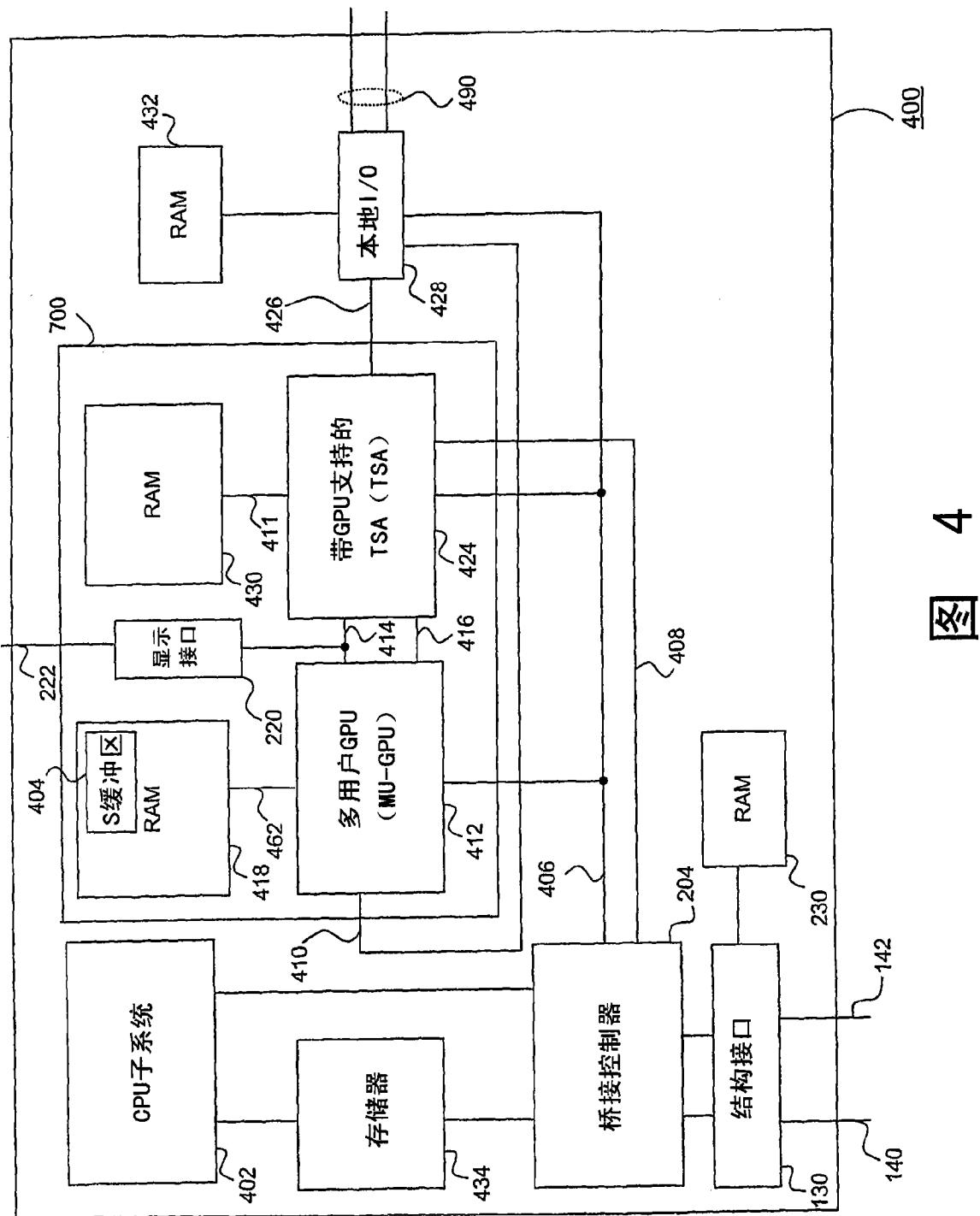


图 4

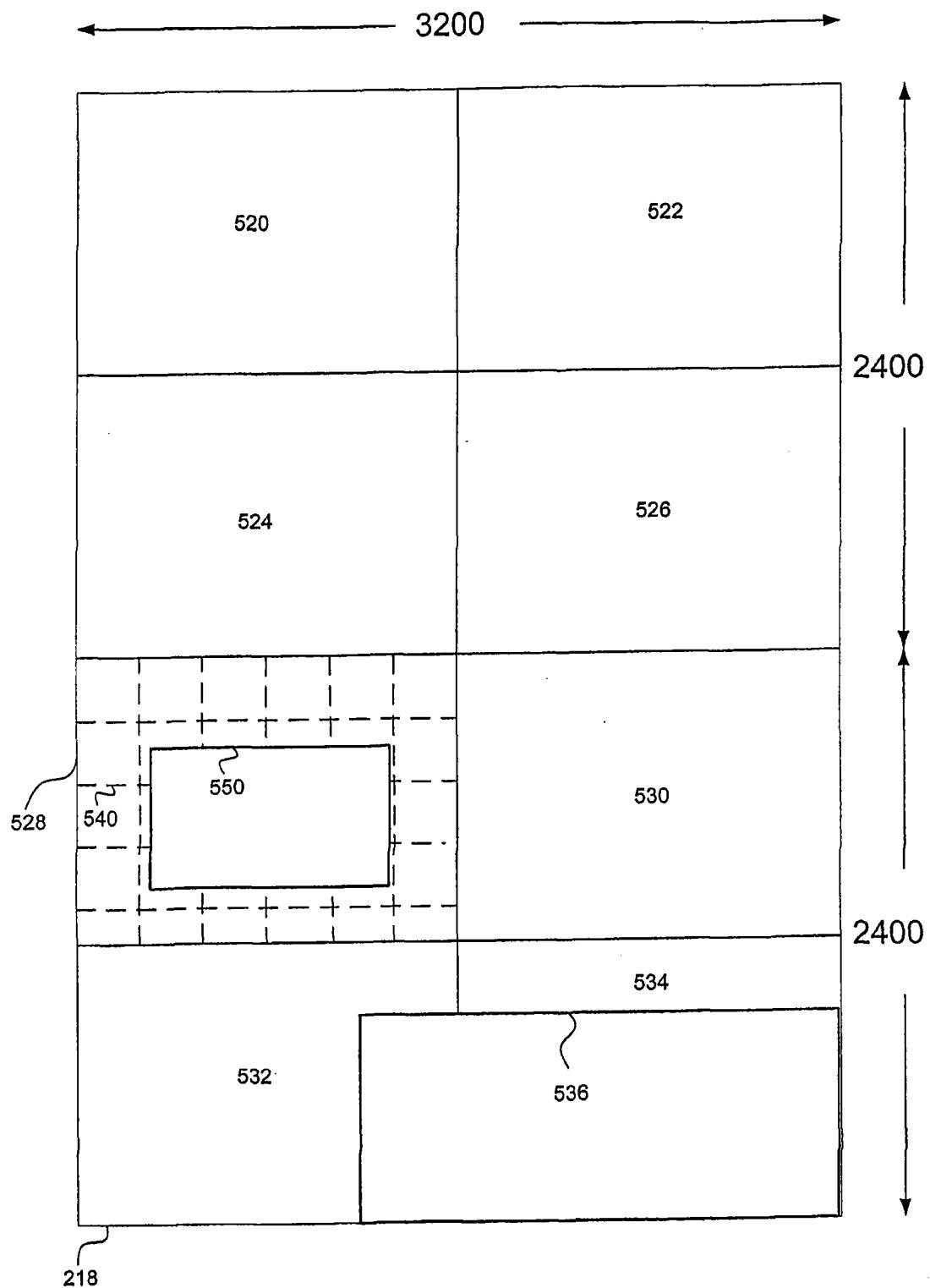


图 5

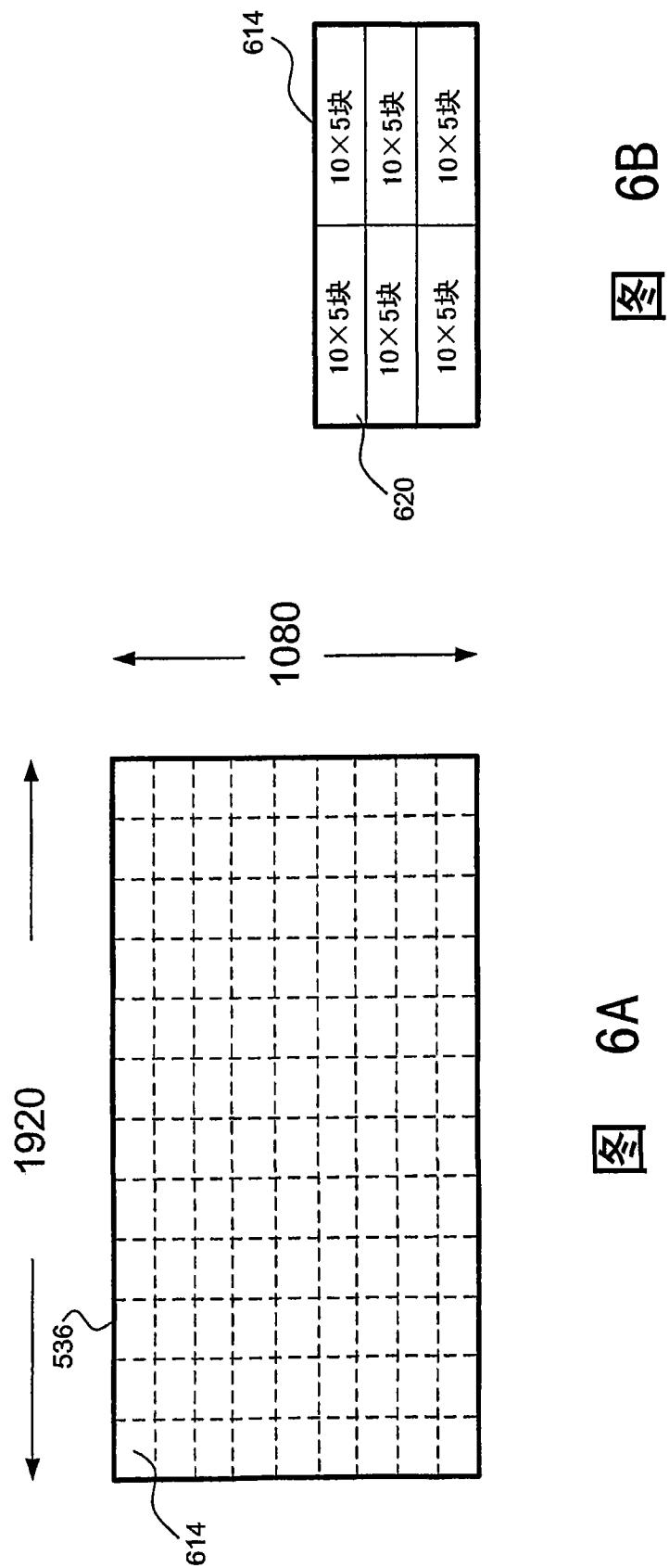


图 6A

图 6B

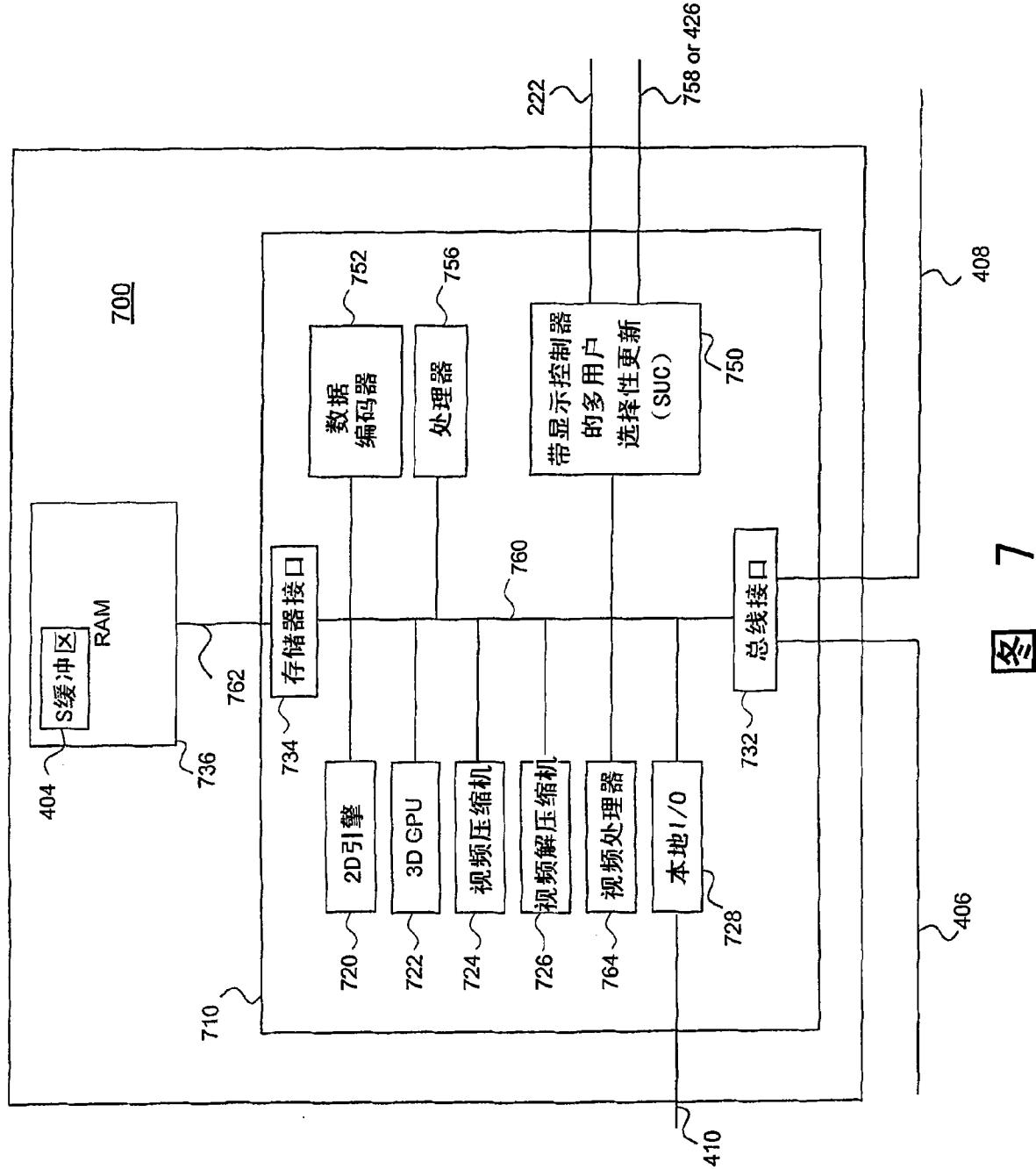
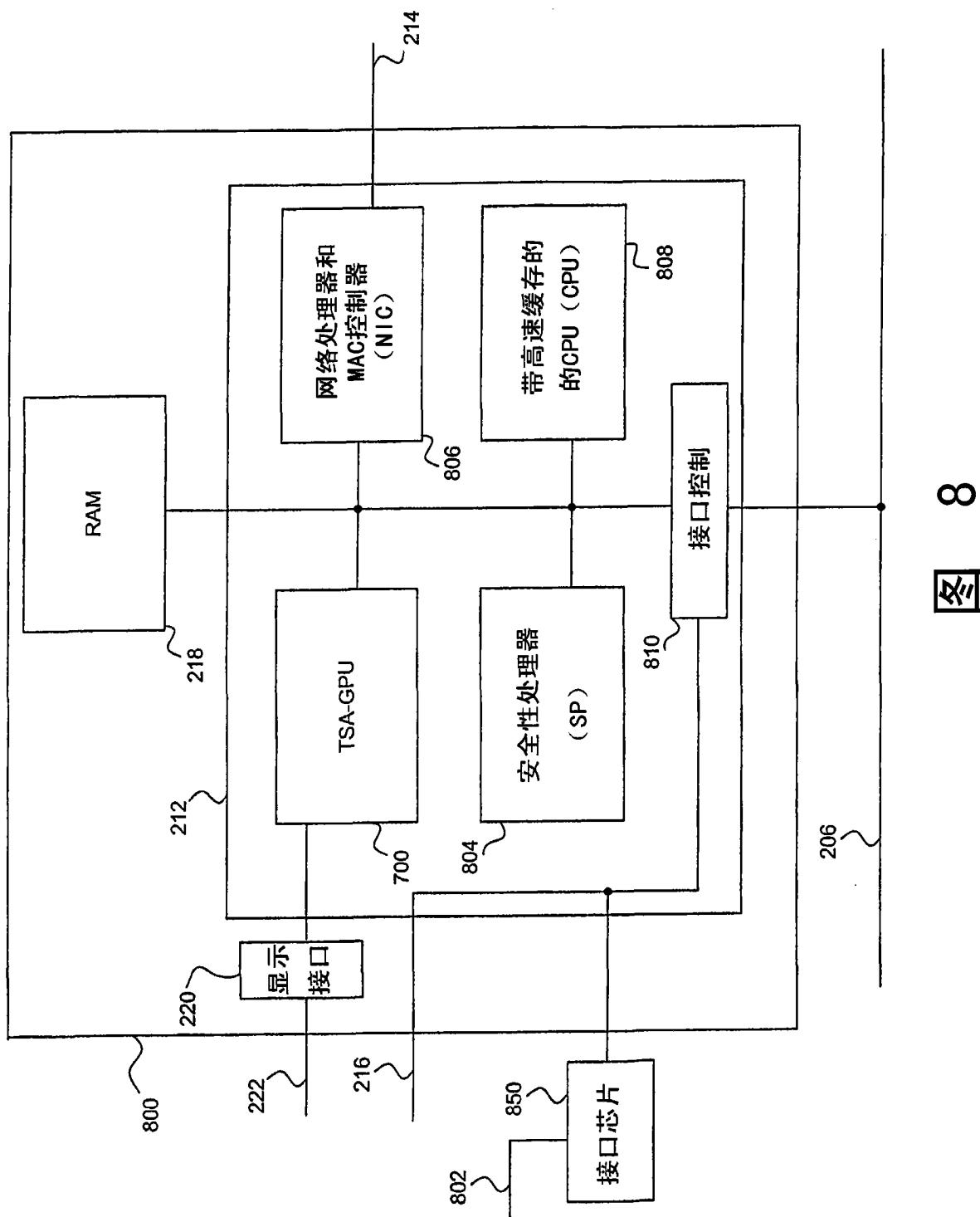


图 7



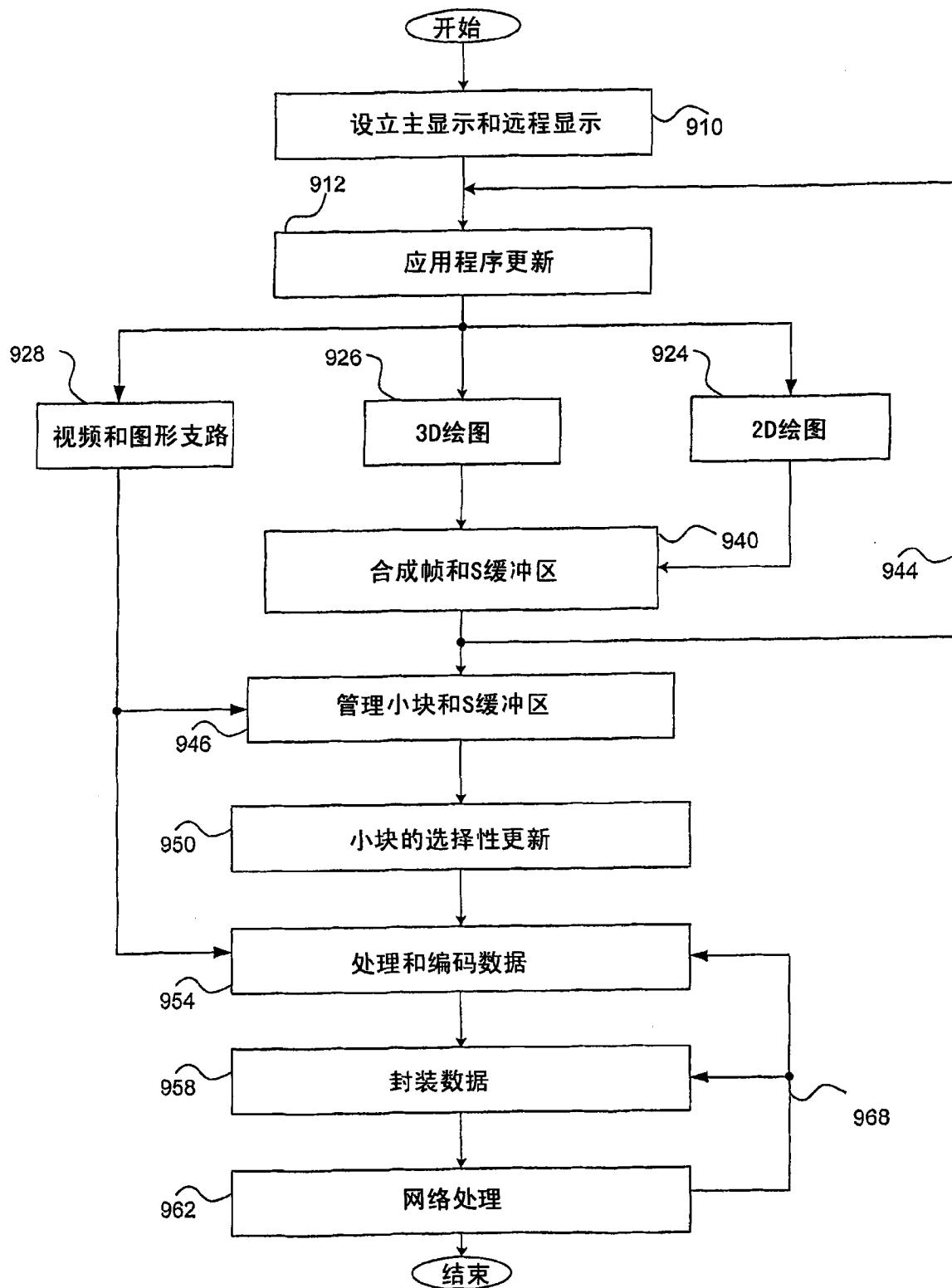


图 9

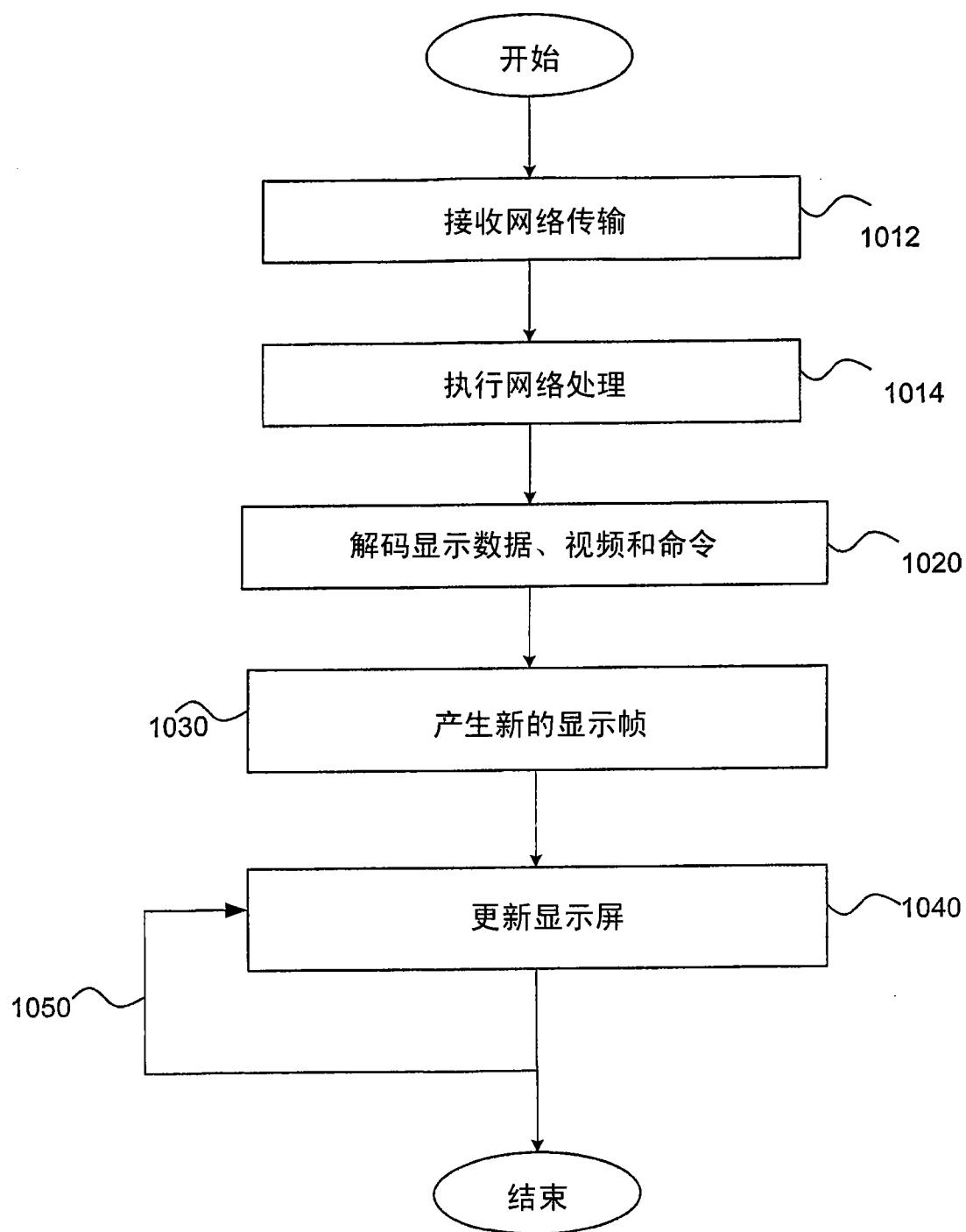


图 10