



(12)发明专利申请

(10)申请公布号 CN 110912982 A  
(43)申请公布日 2020. 03. 24

(21)申请号 201911139688.0

(22)申请日 2019.11.20

(71)申请人 上海戈登网络科技有限公司  
地址 200940 上海市浦东新区中国(上海)  
自由贸易试验区金科路2889弄3号C座  
11层1107室

(72)发明人 王汉杰 周鸣炜

(74)专利代理机构 上海启核知识产权代理有限公司 31339  
代理人 王仙子

(51)Int.Cl.  
H04L 29/08(2006.01)  
H04L 29/06(2006.01)

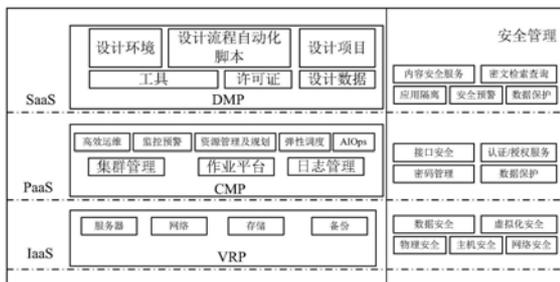
权利要求书1页 说明书11页 附图5页

(54)发明名称

芯片设计云计算系统及方法

(57)摘要

本发明揭示了一种芯片设计云计算系统及方法,所述芯片设计云计算系统包括:软件即服务SaaS、平台即服务PaaS和基础设施即服务IaaS,PaaS做为IaaS和SaaS之间的模块,在IT CAD里主要通过云管理平台对IaaS的硬件资源进行管理,并通过计算集群管理软件对计算资源进行优化,为SaaS中的应用程序和数据库提供运行环境;SaaS包括设计管理平台,集成SaaS层中的所有应用程序并对数据库进行配置。整个体系通过2个管理平台和1个虚拟化资源池对设计项目所需的各种资源进行管理分配并使得成本最优化,使得芯片设计任务不仅能在公有云上高效完成,还能取得很好的ROI。



1. 一种芯片设计云计算系统,其特征在于,包括:软件即服务SaaS、平台即服务PaaS和基础设施即服务IaaS,

PaaS做为IaaS和SaaS之间的模块,在IT CAD里主要通过云管理平台对IaaS的硬件资源进行管理,并通过计算集群管理软件对计算资源进行优化,为SaaS中的应用程序和数据库提供运行环境;

SaaS包括设计管理平台,集成SaaS层中的所有应用程序并对数据库进行配置。

2. 如权利要求1所述的芯片设计云计算系统,其特征在于,所述IaaS包括虚拟化资源池,所述虚拟化资源池包括网络、存储和服务。

3. 如权利要求2所述的芯片设计云计算系统,其特征在于,所述PaaS的云管理平台包括:集群管理模块、作业平台和日志管理模块,通过API接口实现应用场景的监控。

4. 如权利要求3所述的芯片设计云计算系统,其特征在于,所述应用场景的监控包括运维管控、监控预警、资源管理及规划以及弹性调度。

5. 如权利要求3所述的芯片设计云计算系统,其特征在于,所述集群管理模块把IaaS里分配的用于计算的服务器用动态资源管理软件搭建计算集群,并根据设计类型和资源属性定义不同的作业队列和管理参数。

6. 如权利要求3所述的芯片设计云计算系统,其特征在于,所述作业平台主要以运维脚本为基础,实现主机管理,采用集中式管理,控制版本一致性,同时对历史版本统一管理留存,执行时将脚本下发目标机,实现一次操作的完整时序,并且在脚本下发执行上提供审计功能。

7. 如权利要求3所述的芯片设计云计算系统,其特征在于,所述日志管理基于大数据平台对日志数据进行实时捕获和分布式存储,包括IT CAD环境日志,集群资源调度日志,网络传输日志,以及用户操作日志的多维度的信息,同时结合时序数据库实现对日志内容的实时检索。

8. 如权利要求1所述的芯片设计云计算系统,其特征在于,所述SaaS包括设计数据管理模块、设计环境管理模块、设计流程自动化脚本模块和设计项目管理模块,所述设计管理平台结合芯片设计中所需要的应用程序以及数据库,根据不同的设计类型通过设计环境管理模块配置统一设计环境并调用设计流程的自动化脚本完成项目设计。

9. 如权利要求1所述的芯片设计云计算系统,其特征在于,还包括安全管理模块,实现IaaS、PaaS及SaaS的安全监控,涉及基础设施安全和工作负载安全;

优选的,所述基础设施安全包括物理安全、网络安全、存储安全和虚拟化安全;

优选的,所述工作负载安全包括客户端数据加密和数据完整性验证、服务器端加密及网络流量保护;

优选的,还包括数据安全通道,实现数据在跨安全区域之间流动的监控及可追溯。

10. 一种芯片设计云计算方法,其特征在于,采用如权利要求1~9中任一项所述的芯片设计云计算系统进行芯片设计。

## 芯片设计云计算系统及方法

### 技术领域

[0001] 本发明涉及IT技术领域,特别是涉及一种芯片设计云计算系统及方法。

### 背景技术

[0002] 随着人工智能、5G、自动驾驶和HPC等应用的流行,智能终端硬件的升级,厂商对集成电路的需求日益提升,驱使集成电路的规模变得越来越大、电路也日益复杂、集成度也同步提高,这就给相关芯片的设计、测试、验证带来了严峻的考验。

[0003] 从技术层面上看,这些变化不但体现在工具本身上,支撑这些系统运行的计算能力、基础设施甚至成本,也在这个发展过程中给芯片设计者带来了前所未有的挑战。

[0004] 为了解决这个问题,半导体产业链的核心角色(例如EDA Tool/Fabless/Foundry)都开始尝试利用云上几乎无限的计算、存储和资源,从而缩短其产品开发生命周期和上市时间。

[0005] 但是芯片设计的上云,目前尚不存在完善的体系和机制,因此,若能够实现芯片设计的云计算,将从根本上影响甚至变革芯片设计产业。

### 发明内容

[0006] 本发明的一个目的在于,提供一种芯片设计云计算系统及方法,实现高效芯片设计。

[0007] 为解决上述技术问题,根据本发明的第一方面,提供一种芯片设计云计算系统,包括:软件即服务SaaS、平台即服务PaaS和基础设施即服务IaaS,

[0008] PaaS做为IaaS和SaaS之间的模块,在IT CAD里主要通过云管理平台对IaaS的硬件资源进行管理,并通过计算集群管理软件对计算资源进行优化,为SaaS中的应用程序和数据库提供运行环境;

[0009] SaaS包括设计管理平台,集成SaaS层中的所有应用程序并对数据库进行配置。

[0010] 可选的,对于所述的芯片设计云计算系统,所述IaaS包括虚拟化资源池,所述虚拟化资源池包括网络、存储和服务器。

[0011] 可选的,对于所述的芯片设计云计算系统,所述PaaS的云管理平台包括:集群管理模块、作业平台和日志管理模块,通过API接口实现应用场景的监控。

[0012] 可选的,对于所述的芯片设计云计算系统,所述应用场景的监控包括运维管控、监控预警、资源管理及规划以及弹性调度。

[0013] 可选的,对于所述的芯片设计云计算系统,所述集群管理模块把IaaS里分配的用于计算的服务器用动态资源管理软件搭建计算集群,并根据设计类型和资源属性定义不同的作业队列和管理参数。

[0014] 可选的,对于所述的芯片设计云计算系统,所述作业平台主要以运维脚本为基础,实现主机管理,采用集中式管理,控制版本一致性,同时对历史版本统一管理留存,执行时将脚本下发目标机,实现一次操作的完整时序,并且在脚本下发执行上提供审计功能。

[0015] 可选的,对于所述的芯片设计云计算系统,所述日志管理基于大数据平台对日志数据进行实时捕获和分布式存储,包括IT CAD环境日志,集群资源调度日志,网络传输日志,以及用户操作日志的多维度的信息,同时结合时序数据库实现对日志内容的实时检索。

[0016] 可选的,对于所述的芯片设计云计算系统,所述SaaS包括设计数据管理模块、设计环境管理模块、设计流程自动化脚本模块和设计项目管理模块,所述设计管理平台结合芯片设计中所需要的应用程序以及数据库,根据不同的设计类型通过设计环境管理模块配置统一设计环境并调用设计流程的自动化脚本完成项目设计。

[0017] 可选的,对于所述的芯片设计云计算系统,还包括安全管理模块,实现IaaS、PaaS及SaaS的安全监控,涉及基础设施安全和工作负载安全;

[0018] 优选的,所述基础设施安全包括物理安全、网络安全、存储安全和虚拟化安全;

[0019] 优选的,所述工作负载安全包括客户端数据加密和数据完整性验证、服务器端加密及网络流量保护;

[0020] 优选的,还包括数据安全通道,实现数据在跨安全区域之间流动的监控及可追溯。

[0021] 根据本发明的第二方面,提供一种芯片设计云计算方法,采用如上所述的芯片设计云计算系统进行芯片设计。

[0022] 本发明提供的芯片设计云计算系统及方法中,所述芯片设计云计算系统包括:软件即服务SaaS、平台即服务PaaS和基础设施即服务IaaS,PaaS做为IaaS和SaaS之间的模块,在IT CAD里主要通过云管理平台对IaaS的硬件资源进行管理,并通过计算集群管理软件对计算资源进行优化,为SaaS中的应用程序和数据库提供运行环境;SaaS包括设计管理平台,集成SaaS层中的所有应用程序并对数据库进行配置。整个体系通过2个管理平台(DMP:设计管理平台,CMP:云管理平台)和1个虚拟化资源池(VRP)对设计项目所需的各种资源进行管理分配并使得成本最优化,使得芯片设计任务不仅能在公有云上高效完成,还能取得很好的ROI(投资回报率);此外,对不同层次的资源使用不同的安全管理策略,从而保证设计项目数据安全。

## 附图说明

[0023] 图1为本发明一实施例中芯片设计云计算系统的示意图;

[0024] 图2为本发明一实施例中CMP的示意图;

[0025] 图3为本发明一实施例中AIOps的示意图;

[0026] 图4为本发明一实施例中安全管理模块的示意图;

[0027] 图5为本发明一验证例中的云计算系统的示意图;

[0028] 图6a为本发明一验证例中共享存储峰值吞吐情况的示意图;

[0029] 图6b为本发明一验证例中共享存储峰值IO的示意图;

[0030] 图7a为本发明一验证例中监控指标的示意图一;

[0031] 图7b为本发明一验证例中监控指标的示意图二;

[0032] 图8a为本发明一验证例中CPU指标示意图一;

[0033] 图8b为本发明一验证例中CPU指标示意图二。

## 具体实施方式

[0034] 下面将结合示意图对本发明的芯片设计云计算系统及方法进行更详细的描述,其中表示了本发明的优选实施例,应该理解本领域技术人员可以修改在此描述的本发明,而仍然实现本发明的有利效果。因此,下列描述应当被理解为对于本领域技术人员的广泛知道,而并不作为对本发明的限制。

[0035] 在下列段落中参照附图以举例方式更具体地描述本发明。根据下面说明和权利要求书,本发明的优点和特征将更清楚。需说明的是,附图均采用非常简化的形式且均使用非精准的比例,仅用以方便、明晰地辅助说明本发明实施例的目的。

[0036] 请参考图1,本发明提供一种芯片设计云计算系统,包括:软件即服务SaaS、平台即服务PaaS和基础设施即服务IaaS,

[0037] PaaS做为IaaS和SaaS之间的模块,在IT CAD(计算机辅助设计)里主要通过云管理平台对IaaS的硬件资源进行管理,并通过计算集群管理软件对计算资源进行优化,为SaaS中的应用程序和数据库提供运行环境;

[0038] SaaS包括设计管理平台,集成SaaS层中的所有应用程序并对数据库进行配置。

[0039] 在本发明实施例中,所述IaaS包括虚拟化资源池(VRP),所述虚拟化资源池包括网络、存储和服务器。

[0040] 从IT基础架构上云的路径上来说,混合云是本发明一个实施例中主要的半导体上云的方式:基础算力采用传统的私有化部署方式,弹性算力可以利用云端算力来优化成本结构。对于研发人员来说,可以做到透明的作业智能调度,由作业调度工具来分配计算作业分发到本地集群或是云端集群。简单说就是将HPC(高性能计算)区扩展到云端,甚至可以是多云。

[0041] 在公有云计算平台上,将网络、存储和服务器用虚拟技术形成一个资源池,用户根据需求通过自助服务得到这些虚拟资源,这些资源按使用时间收费,例如:云服务商对计算资源使用按秒计费,从而达到成本优化的目的。在云计算平台上可以通过水平扩展,使得用户可以在更短的时间内运行更多计算服务器(license足够的前提下),缩短设计时间,提高设计效率,例如:由于完成每个芯片设计任务的时间是相同的,通过动态地将集群规模扩容到24倍或更多,从而可以在两个小时内完成48小时的RTL验证。

[0042] 另一个好处是利用云计算平台上的丰富的硬件实例,特别是可以获得最新的处理器、存储和网络等技术的使用。用户根据芯片设计的需求将硬件资源调整到最合适的配置,从而以更低的总体成本实现更高的设计性能。用户不需要购买长期的计算集群,而是选择最合适的服务器配置,启动整个服务器集群,并根据应用程序和项目阶段向上或向下扩展这些集群(弹性策略),实现对每个集群的优化。

[0043] 甚至我们可以考虑这样一种情况,用户需要在短短几周内执行gate-level模拟,可能需要一个包含100台机器(意味着超过2000个CPU内核)的集群,具有特定的内存/核心比率和特定的存储配置。使用云计算平台,可以只针对这个任务部署和运行这个集群,仅在模拟需要的时候运行,然后在项目的这个阶段完成时终止集群。同时,云计算平台使得多团队跨区域协作变得更加简单和方便,大大降低了设计数据传输同步以及对各个站点进行维护的成本。

[0044] 计算节点与存储之间的网络吞吐与延迟是影响芯片设计任务运行效果的重要因

素,受益于SRIOV/Nitro等新技术不断的应用在云端的产品上(例如亚马逊云计算AWS的EC2),客户可以实现在自建机房无法做到的技术创新,最新的AWS实例甚至可以做到100Gbps的网络带宽。

[0045] 对于服务器部分,例如在云计算平台上运行EDA软件时,计算优化和内存优化实例类型最适合芯片设计工作负载,可以使用不同的机型配置来满足整个设计任务中不同阶段各个应用程序的需求。

[0046] 对于大规模运行的芯片工作来说,存储会随着大量设计作业的提交而成为瓶颈。为了支持较高的设计吞吐量(后端)与IOPS(前端),设计公司通常要一次性投入巨资购买传统的SAN/NAS存储。在云服务平台上,则可以充分利用云服务商所提供的广泛存储选项,缩短大型批处理工作负载的运行时间。进一步的,如果要尽可能的获得这些优势,须对设计数据进行分类并对设计流程进行一定的重构,但在云上,进行这些优化的好处是十分显著的。

[0047] 根据不同的设计数据类型,用最合适的存储服务来支撑,物尽其用,才能最大化芯片设计作业的效率。

[0048] 进一步的,VRP依据存储,还可以对数据进行备份。

[0049] 例如,可以借助云服务商托管在全球多个位置。这些位置由区域和可用区组成,每个区域都是世界上的独立地理区域,如北京、宁夏、香港。运行的每个区域都设计成与其他区域完全隔离。这可实现最大程度的容错能力和稳定性。通常资源无法跨区域复制,除非用户为他的服务进行特意配置。在每个地理区域内,都有多个相互隔离的地点,这些地点被称为可用区。本发明中可以通过云服务商使客户能够使用这些可用区将资源(如Amazon EC2实例)和数据放置在多个地点。各个可用区相互隔离,但同一区域中的可用区通过低延迟链路相连。通过同时利用多个区域和多个可用区,可以防止出现故障,并确保有足够的容量来运行大多数的计算密集型EDA作业。

[0050] 在本发明实施例中,所述PaaS的云管理平台(CMP)包括:集群管理模块、作业平台和日志管理模块,通过API接口实现应用场景的监控。

[0051] 具体的,云管理平台(CMP)能够充分发挥云计算特性优势,大幅提升生产力,是应对新增混合多云资源管理问题的平台工具。CMP旨在管理云计算所需的资源与环境,充分发挥云计算低成本、弹性、易用、高可靠性、按需服务等特点,同时在提升IT资源利用率的基础上,实现资源的统一视图管理,最终实现云平台的自服务。在IT CAD环境下的CMP,提供对公有云、私有云和混合云整合管理,提高IT的运维效率,实现云上资源的实时监控,控制云上资源的合理分配同时降低云计算成本,实时预警保障SaaS中应用高可用的环境。

[0052] 请参考图2,所述应用场景的监控包括运维管控、监控预警、资源管理及规划以及弹性调度。

[0053] 具体的,在本发明一实施例中,CMP通过可视化界面,让运维人员可以实时动态观测到云服务下所有资源的使用情况。例如统一的信息展示入口,可以从让运维人员能从全局出发快速定位问题,及时决策,保证日常运维任务的高效开展。资源的配置管理,实时服务日志查看,都可以在CMP上方便的操作。在CMP上还可以自定义运维脚本和任务编排,实现对海量云服务远程调度。本发明中的CMP提供多种高效运维方式,基于可视化界面的运维方式适用于非专业运维人员(工程师,资源所有者等),操作简单方便,能更直观的开展运维工作;基于传统命令行的运维方式适用于专业运维人员,可以根据运管平台open API定制需

要的脚本,让运维工作更加高效便捷。

[0054] 在本发明一实施例中,监控预警通过资源监控服务模块实现,可以用于收集主机资源(系统性能、日志等)、HPC、License服务等实时监控指标,探测各项资源及应用服务的可用性。资源监控的各项指标数据统一汇总至CMP,最终以图形化的方式展示一个时间段内的资源池或集群的资源使用情况。同时对资源的各项指标设定阈值,达到实时监控及时预警的目的,以便于IT维护人员及时发现问题进行干预。

[0055] 另外,资源监控数据会在一定时间段存留,并可按照时间回溯,可以帮助IT人员进行分析对比总结规律,能够在未来时间段内更加高效的运维。

[0056] 在本发明一实施例中,CMP的资源管理及规划可依托于资源拓扑图实现资源的编排。资源编排实现了资源及其拓扑关系的自动执行,不仅独立资源可按配置实现自动化部署,各资源间的拓扑关系也将自动执行,帮助运维管理人员省去大量机械重复的手动配置操作。

[0057] 在CMP中可以通过简单的图表拖拽图操作,也可以根据已有的资源关系生成资源拓扑图,从而完成对各种资源的资源属性定义及拓扑依赖关系描述,最终将模板描述的组关系编排出一套实体资源,实现资源的自动化部署和运维。资源编排可以包含多种云服务资源,如主机、路由器/VPC、负载均衡器、共享存储、EDA、集群等,在基于资源拓扑图资源编排过程中,可以清楚的看到所有云资源的分配及占用情况,并能根据资源占用情况模拟出总价,从而为整体资源的容量评估和价格评估提供支持。

[0058] 在本发明一实施例中,弹性调度即弹性伸缩(Auto Scaling,AS),是可提供高效管理计算资源的策略。弹性调度能够实现高效且低成本使用云资源。弹性策略可以设定时间周期性地执行管理策略或创建实时监控策略,来管理计算节点实例或集群数量,并完成对实例的环境部署,保证业务平稳顺利运行。例如,在需求高峰时,弹性伸缩自动增加计算节点实例或集群数量,以保证性能不受影响;当需求较低时,则会减少计算节点实例或集群数量以降低成本。

[0059] 弹性伸缩不仅能够让需求稳定规律的应用程序实现自动化管理,同时能最大程度的保持资源需求与成本消耗互相协调不浪费资源。除了实例和集群数量的伸缩调节外,弹性策略还能根据任务类型(例如计算密集型任务、IO密集型任务等),从CPU、内存、存储等多维度弹性调配适合该任务类型的资源,做到资源的弹性使用和效率最大化。

[0060] 进一步的,弹性调度中还涉及AIOps(智能化运维模块),具体的,请参考图3,本发明中所述AIOps是将人工智能的能力与运维相结合,通过机器学习的方法来提升运维效率。这一点是考虑到,在传统的自动化运维体系中,重复性运维工作的人力成本和效率问题得到了有效解决。但在复杂场景下的故障处理、变更管理、容量管理、服务资源过程中,仍需要人来掌控决策的过程,这阻碍了运维效率的进一步提升。因此,本发明中通过AIOps的引入,使得机器能够代替人来做出决策,从而实现完全自动化真正意义上成为可能。

[0061] 进一步的,AIOps可适用于复杂的运维场景,尤其是一些复合因素的场景,比如磁盘寿命,各物理硬件的实时指标性能监控,服务故障定界定位,根因分析等方面,AIOps借助大数据平台,能够对各种复杂的运维场景给出快速准确的分析。AIOps还能根据历史监测指标、服务模型、规则以及算法智能地及时对云服务的弹性策略做出适当调整,使运维更加自动化和智能化。

[0062] 请继续参考图2,所述集群管理模块把IaaS里分配的用于计算的服务器(计算节点)用动态资源管理软件(DRM,例如:IBM的LSF,SGE等)搭建计算集群,并根据设计类型和资源属性定义不同的作业队列和管理参数。基于此,使得用户的作业能够分配到最优资源尽快完成设计任务并提高整个计算节点的使用效率。

[0063] CMP会根据资源使用监控数据以及项目进度对计算集群的各种参数和队列进行调整,从而取得最大计算资源使用率和最优化设计成本。

[0064] 在本发明实施例中,所述作业平台主要以运维脚本为基础,实现主机管理,采用集中式管理,控制版本一致性,同时对历史版本统一管理留存,执行时将脚本下发目标机,实现一次操作的完整时序,并且在脚本下发执行上提供审计功能。由此,实现集中管理,为云资源管理方面的安全提供多一层的保障。

[0065] 在本发明一实施例中,所述日志管理基于大数据平台对日志数据进行实时捕获和分布式存储,包括IT CAD环境日志,集群资源调度日志,网络传输日志,以及用户操作日志的多维度的信息,同时结合时序数据库实现对日志内容的实时检索。

[0066] 所述日志管理结合实时的资源监控,能最大程度还原真实使用场景,及时发现问题。同时基于对日志的分析可以从多个维度预测未来不同时间周期内的性能趋势,为资源管理和弹性调度配置提供参考依据。

[0067] 在本发明一实施例中,所述SaaS包括设计数据管理模块、设计环境管理模块、设计流程自动化脚本模块和设计项目管理模块,所述设计管理平台(DMP)结合芯片设计中所需的应用程序以及数据库,根据不同的设计类型通过设计环境管理模块配置统一设计环境并调用设计流程的自动化脚本完成项目设计。

[0068] 具体的,所述设计数据管理模块中,所涉及的作为设计项目最为基础的设计数据,根据数据的生命周期以及重要性,基本可以分为以下2个类型:

[0069] 1. 公共数据:此类数据可能在不同的设计项目都会被用到,需要制定标准目录进行统一管理,例如:EDA工具,PDK,IP等。用户在项目设立之初就可以选择不同的工具,工艺文件和IP等,其他没被挑选到的公共数据在项目中不可见。在项目进行之中,用户也可以通过DMP再添加或删除这些公用数据。

[0070] 2. 项目数据:此类数据是为特定的项目生成,并在项目结束是需要进行归档或者删除的,例如:RTL代码,仿真结果,工具使用日志,GDSII文件等。其中,项目会采用文件版本管理工具去对重要数据进行版本管理。

[0071] 此外,在项目建立之初,用户选择好公用数据以后,DMP会自动生成相应的许可证文件并通过设计环境管理模块将许可证文件配置到相应的设计环境中。在项目进行过程中,DMP也会根据公用数据的变化生成新的许可证文件并在项目设计环境中及时更新配置文件从而保证成本最优化。

[0072] 对于所述设计环境管理模块,当项目所有公用数据选择好以后,用户根据项目数据特性搭建好项目数据库,DMP会通过设计环境管理模块为项目自动生成统一的设计环境,使得所有项目参与人员在同一个环境中完成设计项目,减少设计错误。同时,用户也可以通过此模块定制设计环境,使用不同的公用数据达到设计规格。

[0073] 对于所述设计流程自动化脚本模块,DMP会根据不同的设计工艺,IP,芯片类型提供不同的设计流程自动化脚本,这些自动化脚本将会把所有设计工作中用到的点工具链接

起来从而实现项目数据的自动处理。此外,DMP能够不断的优化和升级设计流程自动化脚本去满足更新更复杂的工艺要求。

[0074] 用户在项目设立之初,可以选择适合的项目流程自动化脚本,并根据项目实际进展情况重复或迭代使用这些脚本,也可以根据设计要求修改脚本从而快速完成设计项目。

[0075] 本发明中,考虑到现代的设计项目都是一个设计团队来完成,无论设计团队规模如何,团队成员之间对项目的目标、里程碑、进度、资源调配和成本控制都应该有统一认识并能做到信息共享从而达到共同协作共同完成设计项目。因此,DMP中通过设置设计项目管理模块,从而能够完成设计项目从建立到结束的整个项目管理。

[0076] 进一步的,本发明中的芯片设计云计算系统,还包括安全管理模块,实现IaaS、PaaS及SaaS的安全监控,涉及基础设施安全和工作负载安全。

[0077] 请参考图4,云端负责底层云基础设施的安全,客户端则负责其在云服务商中部署的工作负载的安全。这一设计可以带来灵活性和敏捷性,允许客户为其在云服务商的环境中部署的业务功能实现最恰当的安全控制机制。客户可以严格限制对处理敏感数据的环境的访问,也可以为要公开的信息部署较为宽松的控制机制。

[0078] 在本发明一个实施例中,所述基础设施安全包括物理安全、网络安全、存储安全和虚拟化安全,例如涉及计算、存储、数据库和网络连接;所述工作负载安全包括客户端数据加密和数据完整性验证、服务器端加密及网络流量保护。

[0079] 例如,物理安全涉及:数据中心一般位于没有明显可描述特征的建筑物中,可以在全球范围内分布。物理访问受到严格控制,建筑周边和入口有专业的安保人员,利用视频监控、入侵监测系统和其他电子手段进行全方位监控。授权员工须通过至少两次的双重认证,才能进入数据中心楼层。所有访客和承包商必须出示身份证件,并由授权员工全程护送等等。

[0080] 网络安全涉及:防火墙以及各种边界设备,用于监视和控制网络外部边界和网络内关键内部边界的通信。这些边界设备使用规则集、访问控制列表(ACL)和配置来限制信息流到特定的系统服务的访问。

[0081] 网络安全可以通过云端和客户端共同完成,例如,可以是允许客户端根据自己的工作负载灵活的选择合适的安全级别,以及可以使用内置防火墙创建私有网络和对实例及子网实现访问控制。例如,在所有的服务中针对传输数据进行TLS加密。

[0082] 进一步的,可以通过VPC实现云端的网络隔离。VPC是云端的虚拟网络,可以定义IP地址端、子网、路由表等等网络信息。客户端通过配置VPC,网络ACL以及安全组实现网络、子网以及实例(或Guest)等不能层级的网络安全防护。

[0083] 所述存储安全涉及:存储设备提供给客户端的是未格式化的原始块设备,并且在使用前擦除了数据;提供可用区级别的冗余性,可以启用EBS卷加密或者文件系统级别加密功能,实现数据存储的安全;此外,存储设备在退役之前会采用符合安全规范的技术进行数据擦除和销毁。

[0084] 所述虚拟化安全在多个层级上提供安全性:例如主机操作系统,防火墙和签名API。目标是防止包括计算节点在内的数据被未经授权的系统或用户截获。

[0085] HyperVisor(虚拟机监视器):由于准虚拟化Guest依赖hypervisor为通常需要特权访问的操作提供支持,因此Guest操作系统对CPU没有提升的访问权限。CPU提供四种不同

的特权模式:0-3,称为环。环0是最有特权的,环3是最少的。主机操作系统在环0中执行。但是,本发明中Guest操作系统不是像大多数操作系统那样在环0中执行,而是在特权较低的环1中运行,应用程序在特权最低的环3中运行。从而物理资源的这种显式虚拟化方式使Guest和hypervisor之间被明显的安全隔离。

[0086] 实例隔离:AWS EC2基于分层的安全保护架构,提供完整的实例隔离。

[0087] 在同一物理机上运行的不同实例通过hypervisor管理程序进行隔离。AWS防火墙在hypervisor层中,位于物理网络接口和实例的虚拟接口之间。所有数据包都必须通过这一层,因此一个实例的相邻实例等同与一个位于Internet上的主机一样,可以将它们视为位于不同的物理主机上。物理RAM的隔离机制也是类似的机制。

[0088] 客户端实例服务不能访问物理磁盘设备,通过虚拟磁盘来代替。AWS专有磁盘虚拟化层自动重置客户端使用的每个存储块,以便客户端的数据永远不会无意中暴露给其他人。此外,分配给Guest系统的内存在未分配给Guest时由虚拟机监控程序清理(设置为零)。

[0089] 对于所述主机操作系统:有业务需要访问管理平面的管理员需要使用多因素身份验证来获得访问专门构建的管理主机。这些管理主机可以是专门设计、构建、配置和加固的系统保护云的管理层。所有的访问都被记录并进行审核。当不再需要访问管理平面时,对这些主机的访问权限以及相关的系统将会被及时撤销。

[0090] Guest操作系统:虚拟实例完全由客户端控制。客户端的账号具有完全的root权限或者管理员权限,云端无权访问客户端的实例和Guest操作系统。云端实例操作系统通过各种安全手段和方式实现实例系统的安全访问和控制。首先是分层级的网络访问控制,例如采用的IAM权限管理机制,实现资源的细粒度访问控制;例如AWS EC2实例基于非对称密钥的方式进行登录管理;通过自动化系统分发,系统补丁及时更新保证系统的漏洞可以得到及时的修正;另外,通过集中的监控和审计机制确保系统的异常事件及操作可以得到全面的监控和审计追踪。

[0091] API访问:启动和终止实例的API调用,更改防火墙参数和执行其他功能都由客户端的签名访问密钥,可以是云服务商(例如AWS)帐户的访问密钥或使用云服务商创建的用户访问密钥。如果没有权限使用这些安全访问密钥,这无法进行的API调用。在此外,API调用可以用SSL加密以保持机密性。

[0092] 进一步的,云端还允许客户端进一步控制用户的API的调用权限。

[0093] 优选的,在本发明实施例中,还包括数据安全通道,能够实现数据在跨安全区域之间流动的监控及可追溯。

[0094] 本发明提供的芯片设计云计算系统及方法中,所述芯片设计云计算系统包括:软件即服务SaaS、平台即服务PaaS和基础设施即服务IaaS,PaaS做为IaaS和SaaS之间的模块,在IT CAD里主要通过云管理平台对IaaS的硬件资源进行管理,并通过计算集群管理软件对计算资源进行优化,为SaaS中的应用程序和数据库提供运行环境;SaaS包括设计管理平台,集成SaaS层中的所有应用程序并对数据库进行配置。整个体系通过2个管理平台(DMP:设计管理平台,CMP:云管理平台)和1个虚拟化资源池(VRP)对设计项目所需的各种资源进行管理分配并使得成本最优化,使得芯片设计任务不仅能在公有云上高效完成,还能取得很好的ROI(投资回报率);此外,对不同层次的资源使用不同的安全管理策略,从而保证设计项目数据安全。

[0095] 本发明还提供一种芯片设计云计算方法,采用如上所述的芯片设计云计算系统进行芯片设计。

[0096] 验证例

[0097] 此次测试方法如下:

[0098] 将一个TSMC 28nm后端设计任务分别在本地和云计算系统运行,将得到的运行时间、运行资源和运行成本进行一个比较。

[0099] 验证所涉及的整体构架如图5所示,具体的:

[0100] 1.采用了混合云的模式,保留本地的许可证服务器、NIS以及VDI服务,通过IPSec VPN把本地机房与AWS云上的环境进行网络连通

[0101] 2.利用AWS VPC服务的路由表,除了VPN节点所在的子网以外,其他任何子网都不能访问公网

[0102] 3.为Library和Project数据单独划分出一个VPC,实现数据安全层面的细粒度管控,在这种设计模式下,随着项目的扩展,不会因为所有服务都部署在同一个VPC里,而需要做过于复杂的安全组设置

[0103] 4.VPC Peering能够快速达到跨网段内网互联的目的,而无需任何公网IP或者VPN配置

[0104] 在验证的前期,通过AWS的Snowball服务将超过TB级别的Library/Tools/IP(IntellectualProperty Core)/Project数据传输上云。

[0105] 对比原来机房的HPC环境,本次在AWS平台上采用了一台R5.12xlarge的实例类型,其具体的配置细节,如下表1所示

[0106]

	本地机房	云端
CPU	3.20GHz Intel Xeon E5-2667v4	3.1GHz Intel Xeon Platinum 8175
核心	16	24
线程	32	48
内存	512	384
硬盘	650G	650G

[0107] 表1

[0108] 测试结果如下表2所示:

[0109]

EDA 阶段	本地机房运行时长(min)	云端运行时长(min)	CPU	备注
Placement	550	481	16core	16core, 串行
cts+cts_opt	630	631	16core	16core, 串行
routing	1020	990	16core	16core, 串行
RC extract	23	20	1core(一个 job 占一个 core)	并行 job 数: 4
sta	14	13	16core	并行 job 数: 8

[0110] 表2

[0111] 由此可以看出：

[0112] 1. 此次混合云的架构适用于此项芯片设计任务；

[0113] 2. 对于同一个芯片设计任务，当本地和云上服务器配置差不多的情况下，云计算平台上的运行时间和本地运行时间近似。

[0114] 为了验证芯片设计任务在AWS云平台上的扩展性，我们同时在本地机房与AWS云平台上进行了第二个POC，具体过程如下：

[0115] 1. 本地和云计算平台上分别用一个计算节点来运行12个静态时序验证如下表3所示：

场景	运行阶段	使用工具	服务器数量	线程	运行时间 (min)	备注
本地	STA	Primetime	1	32	27	并行 job 数：12
云端	SAT	Primetime	1	48	20	并行 job 数：12

[0117] 表32. 把96个静态时序验证的作业通过LSF同时提交到云平台上的10台R5.12Xlarge机器上，并模拟本地环境由于受到于计算资源的扩展瓶颈，只能在一台物理服务器上提交96个任务，结果如下表4所示：

场景	运行阶段	使用工具	服务器数量	线程	运行时间 (min)	备注
本地 (on premise)	STA	Primetime	1	32	146	并行 job 数：96
云端	SAT	Primetime	10	480	28	并行 job 数：96，其中一台计算机关闭了超线程

[0119] 表4

[0120] 由此可以看出：

[0121] 1. 同一个并发设计作业在本地和云平台上都能得到充足的计算资源时，运行时间近似。

[0122] 2. 同一个并发设计作业在本地遇到计算资源瓶颈时，云计算平台通过横向扩展能够提供充足的计算资源，其运行时间远远小于作业本地运行时间。

[0123] 要做到横向扩展，随着任务并行的增长，除了计算节点以外，存储也是一块不小的瓶颈，会面临着非常大的吞吐量与随机IO。请参考图6a和图6b所示的POC监测结果，其中，图6a示意了共享存储峰值吞吐情况，图6b示意了共享存储峰值IO。事实证明，在云平台上有能力运行高并发的芯片设计任务，同时提供强大的磁盘IO与吞吐量。

[0124] 从上述第二个测试中，我们也看到结果并非线性，还有方式可以进一步优化。

[0125] 从监测数据上看，如CPU充分利用的情况下，内存实际上只用了50%。如果把计算

节点调整成M系列的机型 (CPU/内存比例为1:4), 并且把tmp目录挂载到NVMe SSD的instance store上, 是有机会实现线性的增长。

[0126] 从成本优化上考虑, 如果把计算节点换成AWS的竞价实例, 那么成本也将会大幅降低。

[0127] 以上测试的所有组件 (包括存储和计算力) 都是按秒计费的, 这也反映了自动化的伸缩 (弹性算力) 在云计算平台上的重要性。

[0128] 为了验证NVMe SSD的优点, 我们利用R5d的实例在Place&Route场景中, 使用ICC2, 做了两次测试:

[0129] 1. 把tmp目录挂载在EBS卷上

[0130] 2. 把tmp目录挂载在两块NVMe SSD本地盘做成的raid0卷上

[0131]

实例类型	每小时费用 (RMB)	Routing 时 间 (min)	vCPU	内存	存储
R5.12xlarge	21.195	870	48	384	EBS SSD
R5d.12xlarge	24.81	660	48	384	2ea 900G NVMe SSD

[0132] 表6

[0133] 图7a显示了Amazon Cloud Watch对这两次测试的监控指标, 图7中左半部分是tmp目录挂在EBS卷上 (14.5小时), 右半部分是tmp目录挂载到两块NVMe SSD本地盘做成的raid0卷上 (11小时)。

[0134] 请参考图7a和7b, 图7b为通过zabbix搭建的监控平台下的指标数据, 可见其数据与Amazon Cloud Watch的结果是一致的。

[0135] 请参考图8a和图8b, 其中图8a示意了R5.12xlarge在zabbix监控平台上的CPU指标; 图8b示意了R5d.12xlarge在zabbix下的CPU指标。

[0136] 在执行任务期间, tmp目录的I/O活动并不高, 导致运行时间产生差异的原因在于I/O的延迟。实例存储的延迟会比走EBS卷的延迟更低, 因为实例存储是对物理机本地存储的直接访问, 而EBS卷则需要通过网络访问。而同样是实例存储, 最新实例R5d又比上两代的实例有四倍的性能提升, 所以虽然在tmp目录上的I/O活动不多, 但是I/O很密集, 每个I/O的响应的延迟越短, 任务完成的时间也就越短。

[0137] 经过上述验证, 可以发现, 本发明提供的芯片设计云计算系统及方法中确实能够实现成本优化, 并且大大提高效率。

[0138] 显然, 本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样, 倘若本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内, 则本发明也意图包含这些改动和变型在内。

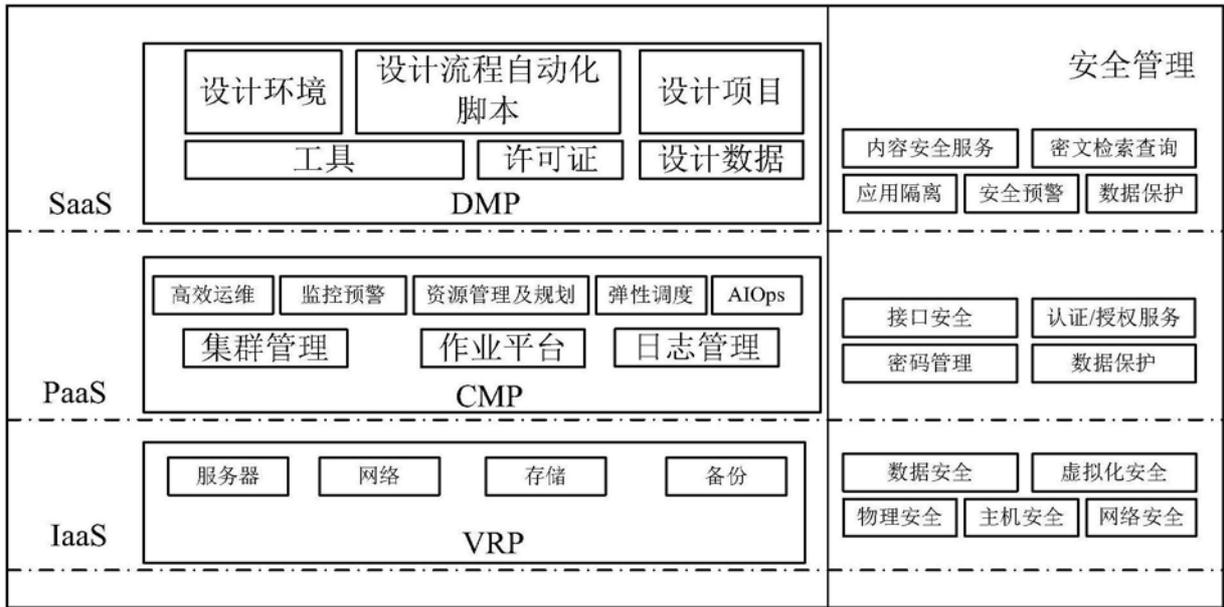


图1

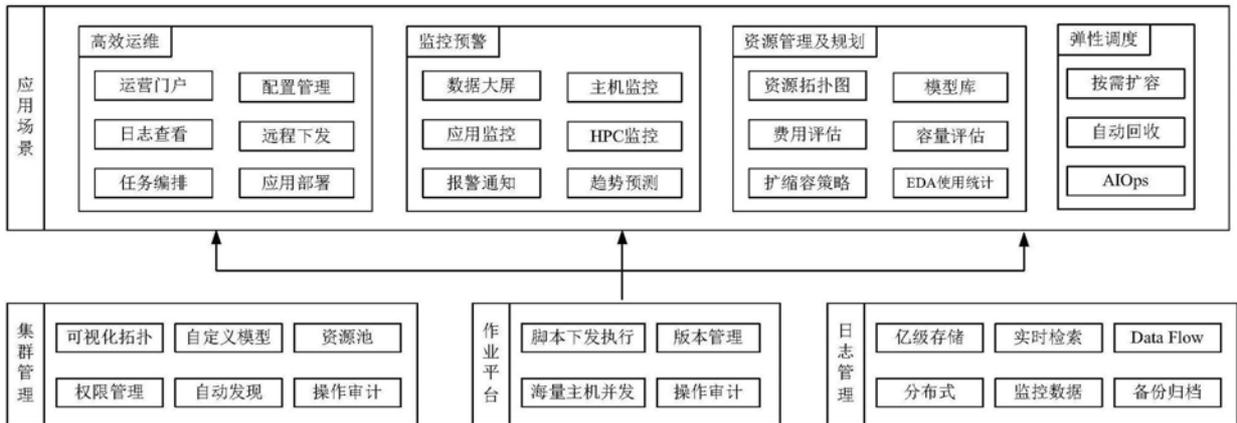


图2

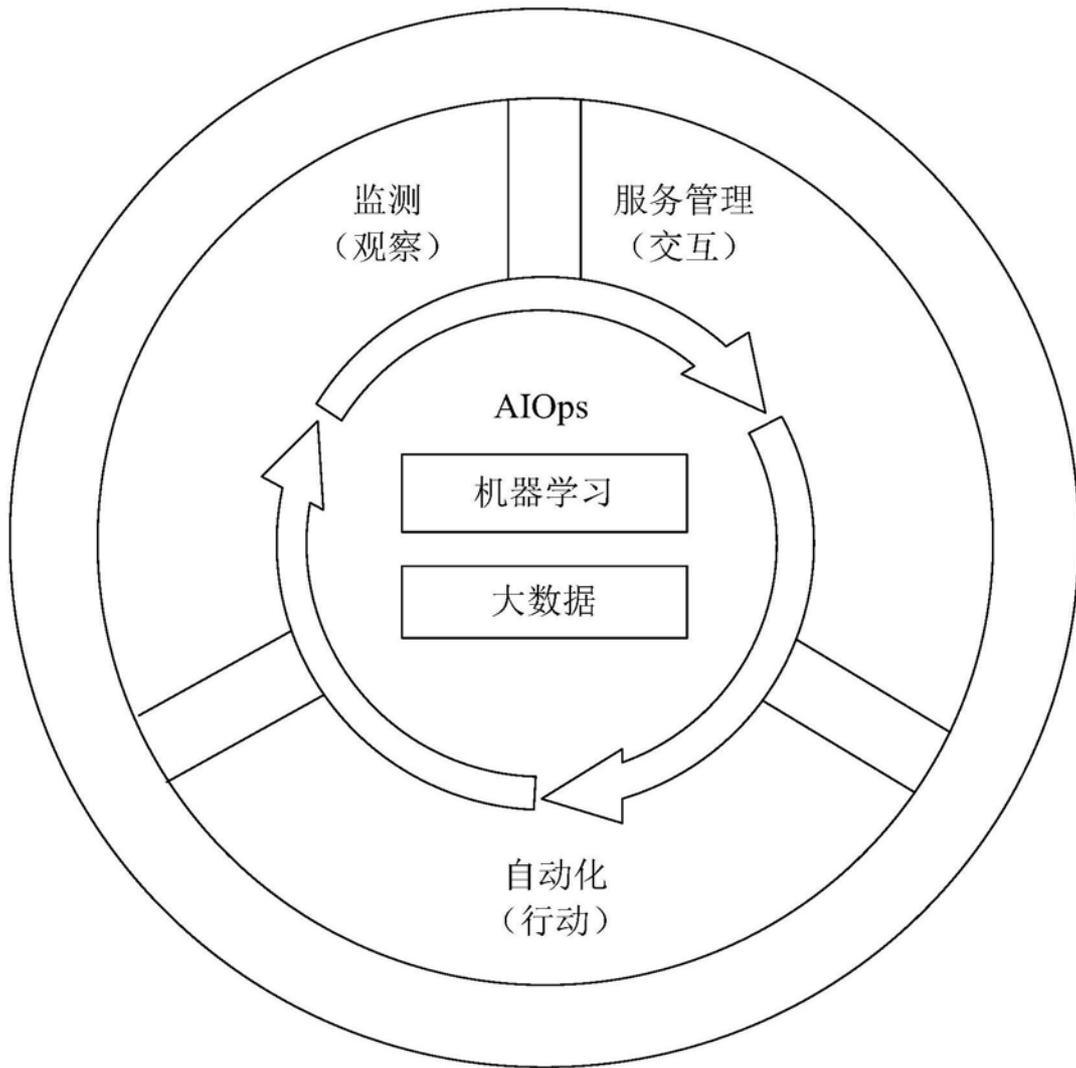


图3

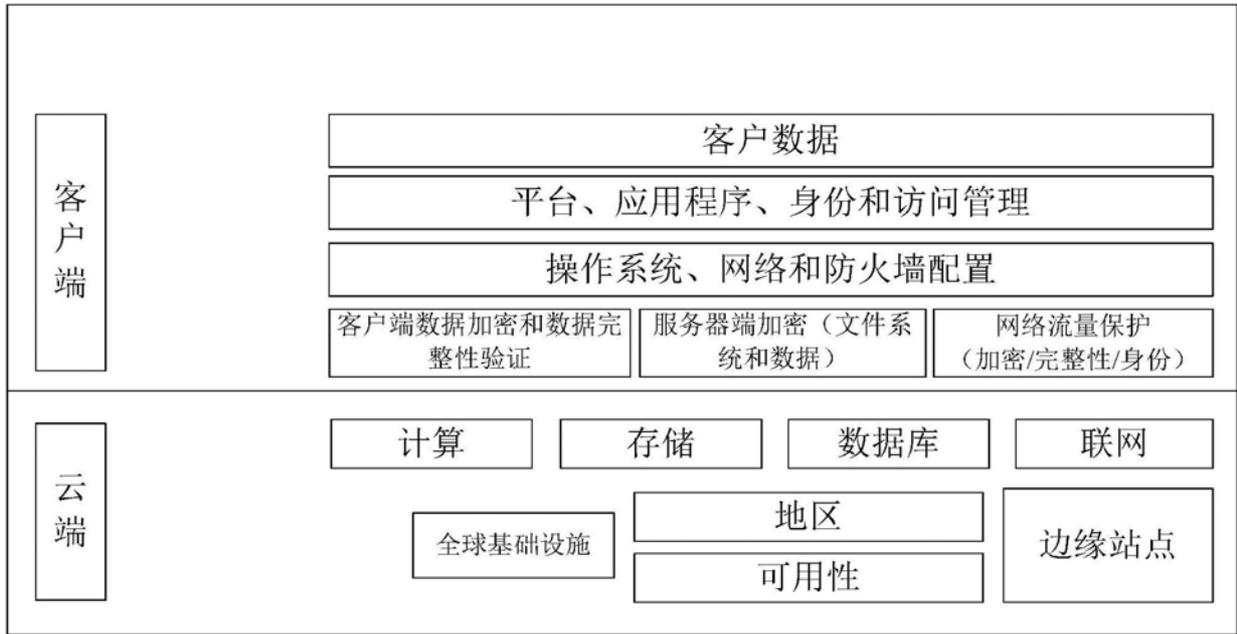


图4

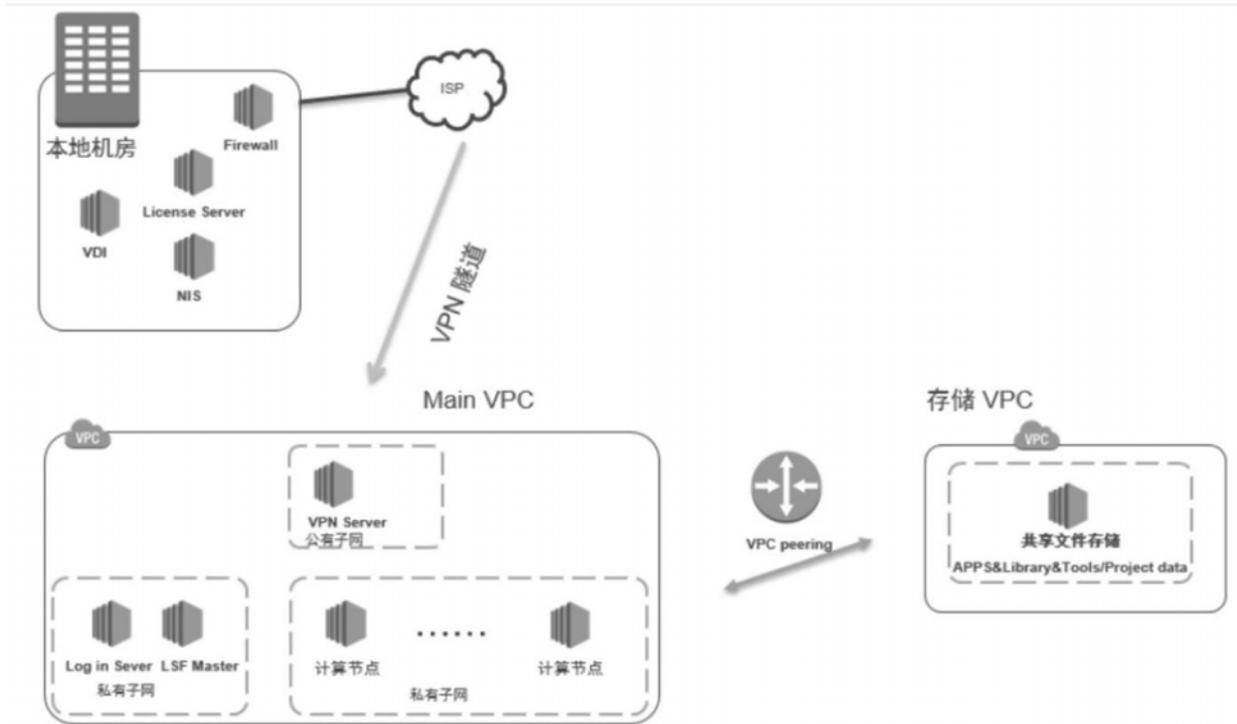


图5

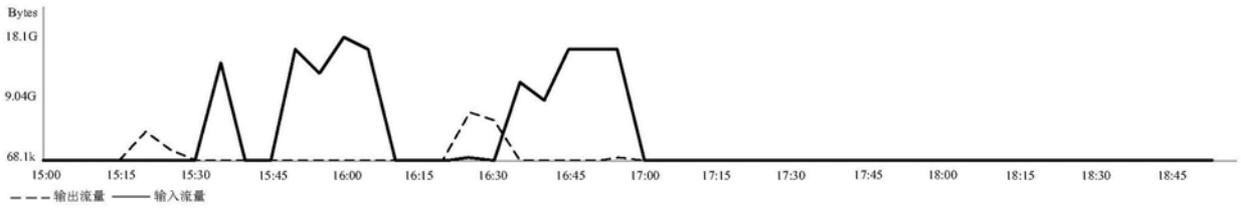


图6a

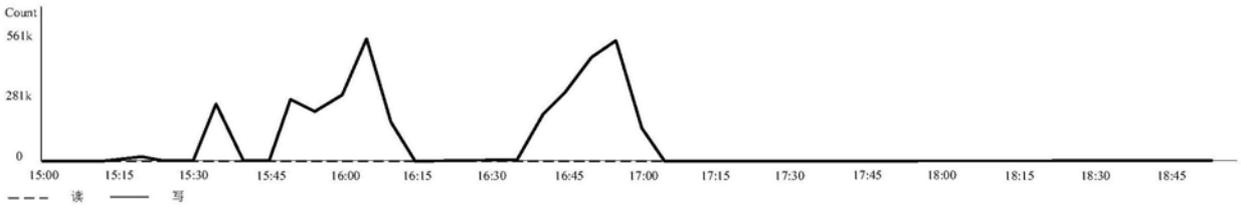


图6b

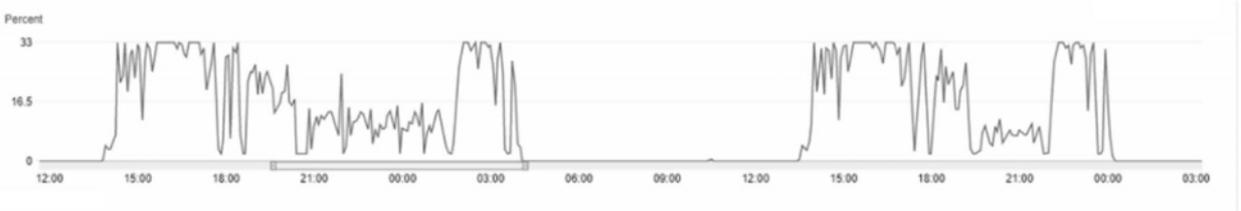


图7a

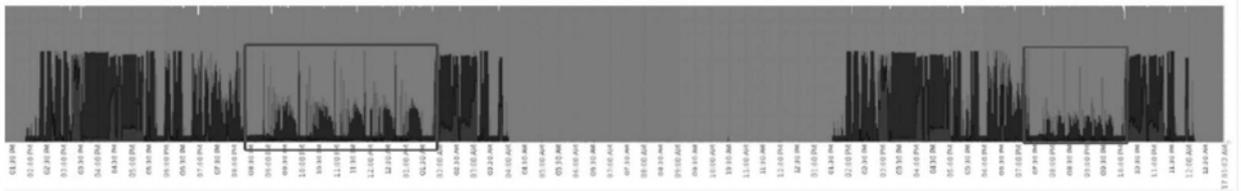


图7b

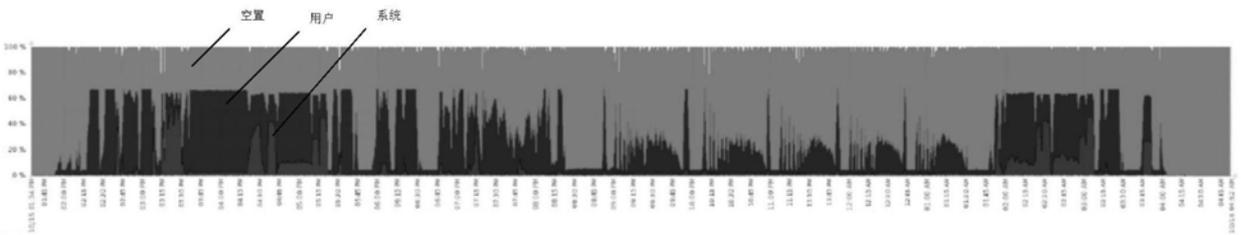


图8a

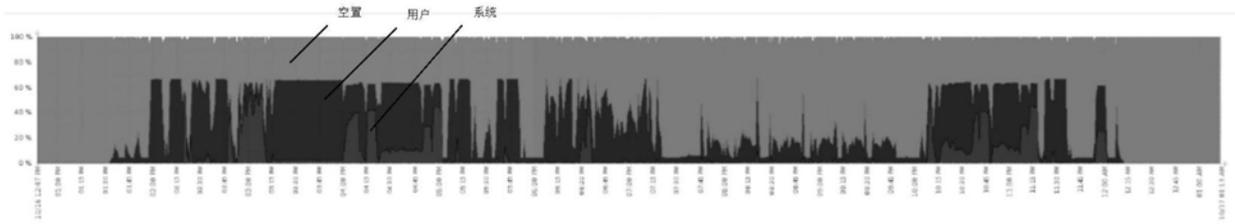


图8b