Journal of System Simulation

Volume 27 | Issue 5 Article 25

9-1-2020

Design of IaaS Mode “Cloud Training” System

Zhijia Chen
Department of Electronic and Optics, Ordnance Engineering College, Shijiazhuang 050003, China;

Yuanchang Zhu
Department of Electronic and Optics, Ordnance Engineering College, Shijiazhuang 050003, China;

Yanqiang Di
Department of Electronic and Optics, Ordnance Engineering College, Shijiazhuang 050003, China;

Shaochong Feng
Department of Electronic and Optics, Ordnance Engineering College, Shijiazhuang 050003, China;

Design of IaaS Mode “Cloud Training” System

Abstract

Abstract: There are some problems in equipment web simulating training, including poor sense of reality, low efficiency of training and high difficulty of management. To solve those problems, *Infrastructure as a Service mode “cloud training” was proposed*. The architecture and operation flow were explained in detail and the key technologies of “cloud training” were focused on. By *GPU virtualization technology*, the problem of client 3D image processing capability was solved. According to the characters of simulation training, the user requirement model was established. *Fuzzy algorithm was introduced to realize resource dynamic scheduling.* To assure the stability and reliability, *combined with checkpoint, virtual machine backup and migration, dynamic failure tolerance algorithm was given*. The results show that resource-sharing capability and simulating training effect have been improved.

Keywords

simulating training, IaaS, cloud training, GPU virtualization, dynamic resource scheduling, dynamic failure tolerance

Recommended Citation

Chen Zhijia, Zhu Yuanchang, Di Yanqiang, Feng Shaochong. Design of IaaS Mode “Cloud Training” System[J]. Journal of System Simulation, 2015, 27(5): 1095-1104.

第 27 卷第 5 期 系统仿真学报© Vol. 27 No. 5
2015 年 5 月 Journal of System Simulation May, 2015

一种 IaaS 模式"云训练"系统设计

陈志佳，朱元昌，邸彦强，冯少冲
(军械工程学院电子与光学工程系，石家庄 050003)

摘要：针对网络化模拟训练真实感不强，训练效率低等问题，*提出了一种 IaaS (Infrastructure as a Service)模式的"云训练"*。阐述了"云训练"的体系结构和运行模式，研究了"云训练"的 3 项核心技术。通过 *GPU 虚拟化技术*解决云环境中终端用户 3D 图形图像处理能力弱的问题。根据模拟训练特点，建立用户需求模型，将*模糊理论引入资源调度技术中，实现虚拟资源的动态调度*；*结合检查点回滚、虚拟机备份和虚拟机迁移技术，实现系统高效动态容错*，保证了系统的稳定性与可靠性。实验证明，"云训练"系统改善了传统模拟训练系统中资源的按需共享能力，提升了资源利用率，保证了模拟训练效果。

关键词: 模拟训练；IaaS (基础设施即服务)；云训练；GPU 虚拟化；资源动态调度；动态容错
中图分类号：TP391 文献标识码：A 文章编号：1004-731X (2015) 05-1095-10

Design of IaaS Mode "Cloud Training" System

Chen Zhijia, Zhu Yuanchang, Di Yanqiang, Feng Shaochong
(Department of Electronic and Optics, Ordnance Engineering College, Shijiazhuang 050003, China)

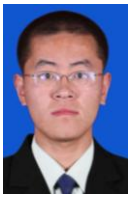
Abstract: There are some problems in equipment web simulating training, including poor sense of reality, low efficiency of training and high difficulty of management. To solve those problems, *Infrastructure as a Service mode "cloud training" was proposed*. The architecture and operation flow were explained in detail and the key technologies of "cloud training" were focused on. By *GPU virtualization technology*, the problem of client 3D image processing capability was solved. According to the characters of simulation training, the user requirement model was established. *Fuzzy algorithm was introduced to realize resource dynamic scheduling.* To assure the stability and reliability, *combined with checkpoint, virtual machine backup and migration, dynamic failure tolerance algorithm was given*. The results show that resource-sharing capability and simulating training effect have been improved.

Keywords: simulating training; IaaS; cloud training; GPU virtualization; dynamic resource scheduling; dynamic failure tolerance

引言

武器装备模拟训练是利用仿真系统完成武器装备的作战指挥、战斗操作和维修保障等的一种训练方式。为了满足规模协同作战的需要，模拟训练向大规模协同训练方向发展，使得模拟器的开发难度变大，而且其组织和开展由于受到场地和模拟器的限制，影响了模拟训练的正常实施，降低了训练效率与效果。

IaaS(Infrastructure as a Service，基础设施即服务)模式云计算[1-2]通过网络为每个用户提供独立的虚拟硬件设备，实现资源的按需分配与高度共



收稿日期：2014-04-22 修回日期：2014-07-03；
基金项目：装备预研基金(9140A04030214JB34001)；
作者简介：陈志佳（1986-），男，河北唐县人，博士生，研究方向为云计算、云仿真；朱元昌（1960-），男，黑龙江哈尔滨人，博士，博导，研究方向为武器系统建模与仿真；邸彦强（1973-），男，河北保定人，博士，硕导，研究方向为武器系统建模与仿真。

享，使得多个用户可以通过网络访问数据中心，完成包括数据计算和图形处理等在内的多种任务。用户不再受地理位置的约束，仅需网络即可获取所需资源，完成相应任务。通过对数据中心的维护管理，保证资源可用性与高效性，有效降低开发人员和终端用户的工作量。结合上述 IaaS 模式云计算的特点，本文提出了一种 IaaS 模式的“云训练”（下文简称“云训练”），为解决武器装备模拟训练中的问题提供了一种新的思路。

1 相关研究

文献[3]针对信息化条件下的装备模拟训练问题提出了一种多用户多任务模拟训练系统的概念，实现了基于网格技术、Web 服务技术和 HLA/RTI 技术的多用户多任务训练的新模式，促进了规模性、多人次的装备教学和训练的发展。文献[4]介绍了一种网格环境下基于 web 的仿真训练系统，使仿真训练系统中的海量异构资源得到了较好的控制，解决了用户交互体验和网页全局刷新问题。针对航空兵训练的特殊性，在云计算强大计算和存储能力的基础上，文献[5]提出了一种新型的航空兵训练仿真系统体系结构，描述了云计算平台应完成的主要内容，在理论上证明了云计算与航空兵仿真训练结合的可能性。在云计算和网络化建模与仿真技术[6]的推动下，文献[7]提出了网络化模拟训练模式，它实质是一种 SaaS(Software as a Service，软件即服务)的服务训练模式，将虚拟训练资源部署在训练中心，用户通过浏览器将训练模型下载至终端 PC 机，实现与其他用户协同训练。上述研究成果在不同方面对解决传统模拟训练存在的资源重用问题起到了一定作用，具体表现在：(1) 改变了传统单机集中式的模拟训练方式，实现了面向网络的训练模式；(2) 实现了多用户、多任务的武器装备模拟训练，解决了多人次同时训练与教学的问题。

上述网络化模拟训练方式在一定程度上提升了装备模拟训练的规模和效率，但是还存在一些弊端：(1) 由于模拟训练系统越来越庞大，包含很多 2D 模型、3D 模型以及大量图形图像，且模型数目及所占空间也随系统变得越来越大，传统的网络化模拟训练在训练前需要将全部模型下载到终端，这种方式使得模拟训练的准备过程变得十分冗长，影响模拟训练效果与效率。(2) 由于训练模型资源需要全部下载到本地，并且需要利用本地计算机硬件资源对模型和图形图像进行处理，因而对本地终端要求比较高，部分单位的训练终端计算机配置较低，在实际部署应用时存在模型真实感不强、训练效果差、训练效率低的问题。(3) 训练终端的设备维护较为复杂，可靠性受到极大影响，而且出现故障后重新配置系统较为繁琐，耗费大量的人力物力，制约着大规模模拟训练的实施与开展。

2 云训练系统结构与运行模式

IaaS 模式云训练基于 IaaS 模式云计算的理念，以桌面云[8]技术为依托，将包括软件资源、硬件资源和仿真资源在内的训练资源进行封装，为用户提供基于计算、存储、图形处理、模型等基础软硬件资源的训练服务，降低对训练终端机器配置要求的同时使服务更加多样化，提升训练用户的自由度。

2.1 云训练的结构框架

云训练的基本结构为“中心—终端”式结构，如图 1 所示。“中心”包含了模拟训练系统的基础设施资源、模型资源及其他软件资源，通过虚拟化技术，将上述资源虚拟化封装为以虚拟机为单位的模拟训练节点，通过远程传输协议为训练终端提供服务。“中心”的资源可用性和系统容错性是保证终端获取高质量服务的保证，是提升训练效果的重要因素，也是本文重点研究内容。“终端”按照功能可分为两大类，分别是训练终端和开发终端，在硬件上均可采用瘦客户端，以减少终端用户的基础设施部署与维护成本，提高开发训练效率。“中心”和“终端”通过网络进行互联，网络可以采用适应军队基础建设的军训网、营区网或者校园网等。终端用户通过远程传输协议与训练中心进行数据交互，开发

终端将开发的训练项目发布到云中心，中心调度服务将训练项目封装到虚拟机中，通过远程传输协议将训练项目传送至训练终端。

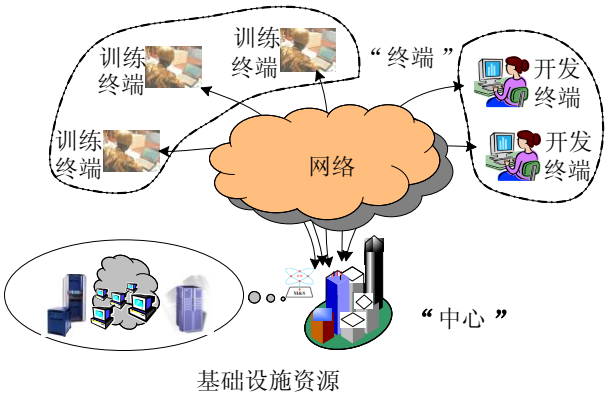


图 1　IaaS 模式云训练示意图

如图 2 所示，云中心结构由下至上分为 5 层，分别是基础设施层、虚拟化层、虚拟资源层、管理层和支撑平台层。基础设施层涵盖了计算资源、图形处理资源、数据资源、模型资源和存储资源等多种类型资源，这些资源融合为一个资源池，为整个云中心的运行提供基本的软硬件设施保证。虚拟化层通过虚拟化技术将这些资源虚拟化封装为多个虚拟机，这些虚拟机构成了虚拟资源层，可以提供基础设施层所包含的全部资源。上述以虚拟机为单位面向用户提供训练资源的方式体现了 IaaS 模式云的思想。管理层作为管理整个云训练系统正常运行的关键层，它提供了创建虚拟机、准备桌面池、用户账户信息管理、虚拟机镜像管理、虚拟资源管理、物理服务器管理、操作系统迁移管理、安全管理等功能。支撑平台层为用户提供训练支撑环境，用户通过服务接口连接至中心，可以定制服务，包括训练科目、节点个数等内容。

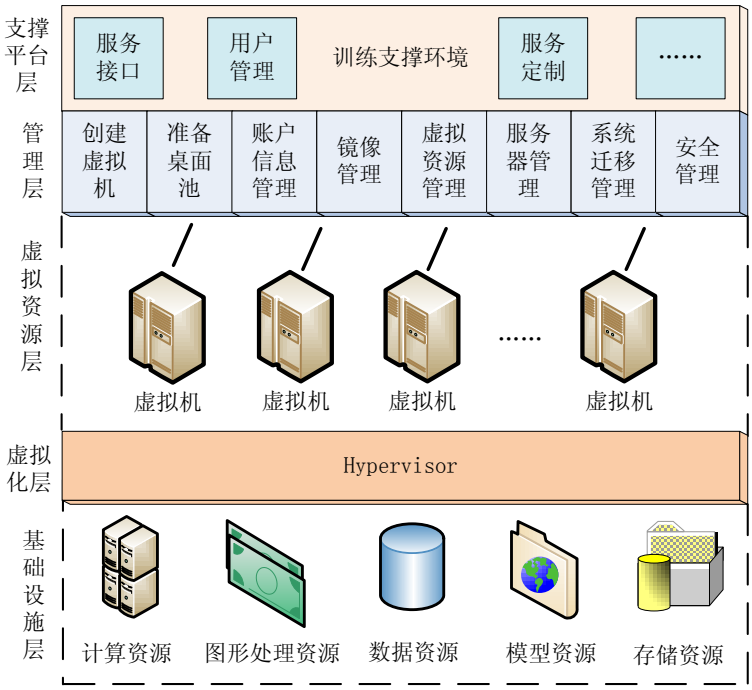


图 2　云中心体系结构示意图

2.2 云训练运行模式

云训练中心硬件基础设施搭建完毕后，由开发终端开发训练系统，并将仿真模型、仿真软件和其他模拟训练资源集成到云训练中心。中心运行时，终端训练用户根据自身需求向云中心提交训练任务请求，包括训练科目名称、虚拟机配置需求、仿真模型资源需求等。中心接到请求后，通过管理系统调度所需软硬件资源，创建虚拟机，为用户动态分配相应的训练资源[9]。用户根据得到的资源进行

模拟训练，训练完毕后，将训练效果、训练体验(真实感，流畅度等)等信息反馈至云中心，中心根据反馈信息为每个用户创建相应的文件，管理用户信息，如用户的虚拟机镜像、用户的训练记录等，根据这些信息，中心提出系统改进建议，开发终端根据这些建议对训练系统进行升级，通过反馈达到完善系统的目的。IaaS 云训练的运行模式如图 3 所示。

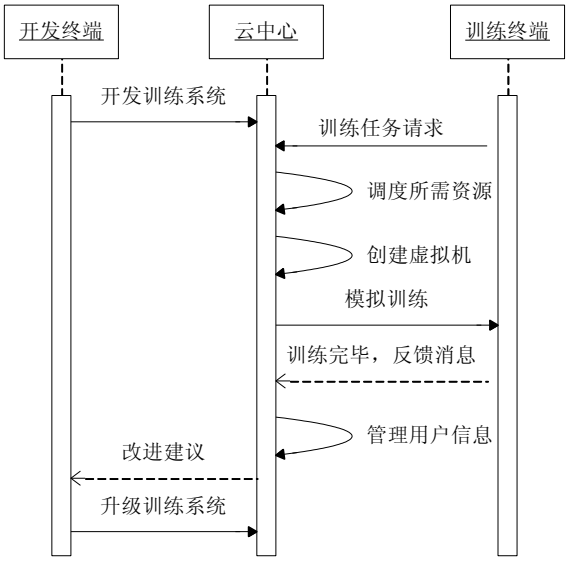


图 3 IaaS 云训练运行模式

3 云训练系统核心技术研究

在云训练系统构建过程中，对系统的核心技术进行了重点研究，包括 GPU 虚拟化技术、虚拟资源按需配置技术和系统动态容错技术。

3.1 GPU虚拟化技术

为了将 3D 模型逼真的呈现给用户，提升模型的模拟效果，使用户获得流畅的操作体验，虚拟 GPU 需要提供强大的图形处理能力。现在部分学者在提升虚拟 GPU 性能方面取得了一些成果，如 Asael Dror[10]等人设计的一种针对 Windows 操作系统的 GPU 虚拟化方案，实现了为多个远程桌面提供图形处理支持能力。文献[11]提出了一种采用 API remoting 法实现的 GPU 虚拟化中间件，无需更改硬件结构，可以满足通用物理 GPU 共享的需求，实现了服务器中多虚拟机节点对 GPU 的共享。VMware 的 GPU 虚拟化方案[12]采用前端-后端(frontend-backend)的实现模式：前端位于客户操作系统，面向应用，由伪显卡驱动和虚拟 GPU 构成，在实现方法上是以设备仿真为主，实现了根据 2D 和 3D 图形指令驱动不同功能单元的目的。但是上述实现方案中，或者引入了大量数据的复制与传输，或者依靠牺牲 CPU 的计算能力实现对 GPU 的模拟，显存受到很大限制，虚拟 GPU 的图形处理能力无法得到有效提升。

在上述 GPU 虚拟化的基础上，提出了一种改进的 GPU 虚拟化实施方案。在方案中，首先以设备独占[13](VMM pass-through)的方式创建一个父虚拟机，称为图形虚拟机(Graphic Virtual Machine，简称 GVM)。以 GVM 作为其他虚拟机的“图形服务器”(Graphics Server)。显卡驱动采用显卡厂商发布的专用显卡设备驱动，因此通过调用该驱动可以充分实现 GPU 的全部特性，包括调用 GPU 以实现高速的 3D 图形处理和数据运算。由于采用了设备独占使用的虚拟化实施方案，因此 GVM 可以获得与物理机相似的图形处理性能。利用 Hypervisor 继续创建多个子虚拟机，也称为桌面虚拟机(Desktop Virtual Machine，简称 DVM)。这些 DVM 分配给终端用户使用，不与物理 GPU 发生直接交互。DVM 与 GVM 之间共享数据和指令存储空间，降低由于数据复制和传输带来的时延。DVM 通过远程桌面协议，如 ICA 和 PCOIP 等传输协议，将图像传输至远程客户端，为客户端提供所需的 3D 图形呈现。系统架构如图 4 所示。

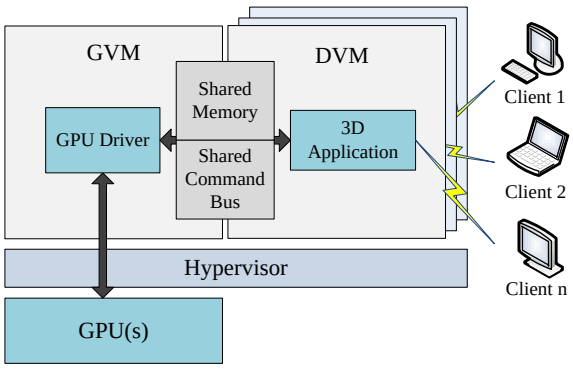


图 4 GPU 虚拟化架构

GVM 主要包括 2 大部分：GPU 驱动和渲染组件，其中，渲染组件又包括捕获、压缩和渲染系统。通过 GPU 驱动，GVM 可以被翻译成一个虚拟图形设备。GVM 根据 GPU 的相关特性，对其进行调度管理，包括 GPU 通道管理、环境管理、内存管理等[14]。在 GVM 上会为每个 DVM 提供相应的图形捕捉、渲染和压缩功能。GVM 根据收到的请求，将任务按照一定的顺序交由物理机(图形服务器)的 GPU 控制器完成，而 GVM 上的图形渲染组件可以不断刷新桌面图像。而 DVM 对物理 GPU 的调用都需要经过 GVM 来进行，通过 API remoting 方法，解释为对 GVM 上的 API 调用。这套 API 可能是 Direct 3D 的 API 或者其他 API。DVM 中的 3D 命令存储于由用户模式驱动构建的命令缓冲器中，经编码后发送至由内核模式驱动构建的 DMA 缓冲器中，然后随着数据变量一起发送至 GVM。通过这种机制，一个物理 GPU 的图形处理功能，包括 3D 图像和多媒体处理功能，可以被多个子虚拟机共享，多个虚拟机可以同时在物理机上的 GPU 进行 3D 渲染任务。子虚拟机与父虚拟机之间共享内存,减少了图形处理数据的复制传输操作，降低了传输时延，提升了图像刷新速率。

通过这种方法，首先减少了数据在虚拟机各层之间的传输和复制的次数，因为 GVM 和 DVM 之间的命令传输方式是直接通过共享内存空间传输的，与传统的 API remoting 方式不同，不需要拦截 GPU 相关指令，也无需进行特殊处理，只需要将其放置到二者的共享空间即可，因此相对传统的 API remoting 方式来说提高了效率。其次，它解决了设备独占法生成的虚拟 GPU 无法为多个虚拟机共享的问题，是对设备独占法的一种完善和优化。

3.2 虚拟机资源动态调度技术

针对训练用户的特殊性，平台为用户提供两种不同的资源申请模式，一种模式与普通云计算的资源申请模式相似，直接提交所需虚拟硬件配置；另外一种则是用户提交训练内容信息、预计使用起止时间、使用人员信息等，由中心评估预测所需资源及软件配置，并动态为用户配置虚拟资源的数目。下面主要对第二种模式进行详细介绍。

3.2.1 虚拟资源配置结构

IaaS 云训练平台中，虚拟资源配置结构如图 5 所示。图中，用户并发要求资源数量是指同时在训练平台上线并进行训练的用户所请求的包括 CPU、内存和 GPU 等在内的资源总量。该总量根据用户训练内容、训练类型以及用户上线数量预测得来。通过资源管理系统实时监控用户虚拟机的资源利用情况，根据每个用户训练任务类型(逻辑运算密集型、图形图像密集型)、资源配置与使用情况和虚拟机性能的关系，推导并建立三者关系模型。在该模型基础上调整各用户虚拟机的配置，提升虚拟机的性能，提升资源利用率。并将调整参数反馈至监控中心，从而不断完善上述模型，达到有效提升模拟训练性能的目的。通过该模型保证云训练资源的可用性与高效性：即在资源需求高峰时，避免过载，保证服务的正常可用；在资源需求低谷时，尽量减少资源占用，避免资源浪费。

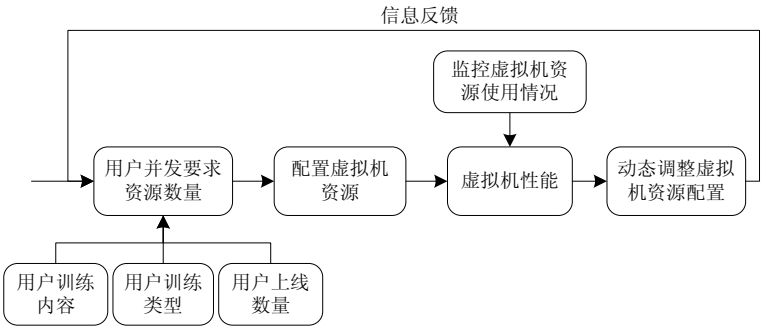


图 5　用户虚拟资源配置模型示意

3.2.2 用户需求预测

结合用户使用特点，采用以天为单位进行分时段预测，在预测时，认为同一天内的不同时段之间的资源请求数量彼此独立，仅需考虑历史记录中当前时段的用户资源请求数，无需考虑同一天内其他时段的数据记录。云计算数据中心的连续长期运行能够为预测算法提供足够的历史数据，因而该预测模型也会更加适应于云计算数据中心的长期资源配置模型。鉴于上述预测模型特点，基于二次移动平均法[15]，降低了预测值与实际值的滞后偏差。第 i 个用户在 $t+\tau$ 时刻的资源需求预测值如下式：

$$x_{t+\tau}(i)=a_t(i)+\tau b_t(i) \tag{1}$$

式中：$x_{t+T}(i)$ 为待预测值；τ 为待预测的时刻序号，

$$a_t(i)=2M_t^{(1)}(i)-M_t^{(2)}(i) \tag{2}$$

$$b_t(i)=\frac{2}{N-1}[M_t^{(1)}(i)-M_t^{(2)}(i)] \tag{3}$$

式中：$M_t^{(1)}(i)$ 和 $M_t^{(2)}(i)$ 分别为第 i 个用户在第 t 个时刻资源需求值的一次移动平均数和二次移动平均数；N 为选择移动平均的时期数，且

$$M_t^{(1)}(i)=\frac{x_t(i)+x_{t-1}(i)+\cdots+x_{t-(N-1)}(i)}{N} \tag{4}$$

$$M_t^{(2)}(i)=\frac{M_t^{(1)}(i)+M_{t-1}^{(1)}(i)+\cdots+M_{t-1}^{(1)}(i)}{N} \tag{5}$$

则全部 m 个用户的资源需求总量可由式(6)得出

$$x_{t+\tau}=\sum_{i=1}^{m}x_{t+\tau}(i) \tag{6}$$

由上述分析可知，在 $t+\tau$ 时刻的预测值仅由 t 时刻的前 N 个时期的值有关，预测结果在 t 时刻开始即可计算得出。在上述预测模型中，预测的精度与实时性是一对冲突的指标，影响二者的参数为历史数据的个数 N 决定。提升 N 值可以在一定程度上提升预测精度，但是同时会降低预测实时性；而降低 N 值则会在一定程度上降低预测精度，但是对预测的实时性则会产生较小影响。

3.2.3 基于模糊控制理论的资源调度

对于用户训练任务而言，不同任务需要不同类型的虚拟机资源，某些任务进行了较多的网络通信，希望分配到带宽较高的虚拟机资源，而有些任务进行大量的数据运算，则更希望分配到有较高计算性能的虚拟机资源[16]。因而，对于用户任务的这些 QoS 需求，我们将虚拟机属性信息 VM_i 抽象化为 4 个特征信息粒 C_{i1}，C_{i2}，C_{i3}，C_{i4}，分别代表 CPU 信息粒，GPU 信息粒，Memory 信息粒以及 Bandwidth 信息粒。

如图 6 所示，当系统有多个用户训练任务到达时，首先对该任务进行排序，根据训练任务的优先级，排序方式可以是先到先服务(FIFO)方式。然后根据每个用户的训练任务进行模糊推理：根据用户的任务大小、类型、持续时间等属性，参照模糊推理知识库对任务所需资源进行模糊逻辑推理，然后按照一定的解模糊策略将推理结果清晰化，转化为清晰的数据量，包括对 CPU，GPU，内存等的具体需求值。系统根据这些具体需求将硬件资源配置为虚拟机，达到动态为每个用户配置资源、提升资源利用率的目的。

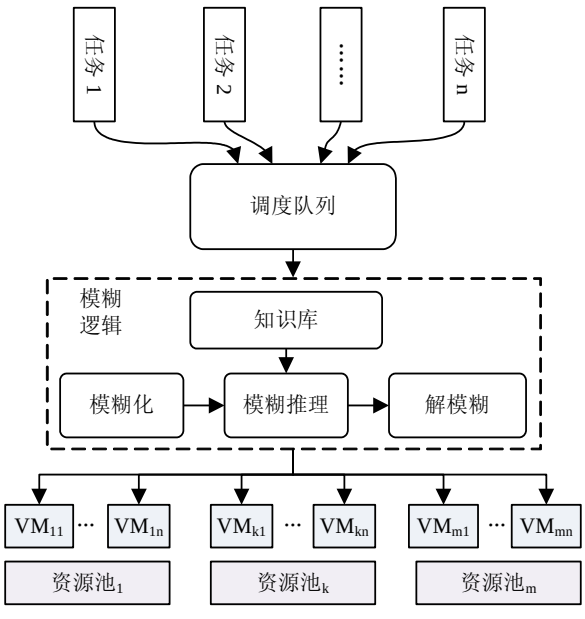


图 6 基于模糊控制理论的资源调度过程

针对模糊推理器的需要，将云训练中心的资源按照提供服务的能力分为 7 个等级，分别定义为很高、高、较高、中等、较低、低、很低。同理，对

资源需求按照上述方法划分为 7 个等级。中心的资源调度根据为用户虚拟机增加或者减少分别将符号定义为正和负，按照调度的数目定义为正大、正中、正小、零、负小、负中、负大。按照模糊推理的一般定义，将上述 7 级服务能力以及资源调度数目由高到低简写为 *PB*，*PM*，*PS*，*ZO*，*NS*，*NM*，*NB*。

构建模糊推理规则时，按照二输入单输出方式进行。输入的前提条件分别为：(1) 用户资源需求等级；(2) 资源可用性等级。输出为虚拟机资源调度值。

假设用户的训练任务类型有 *n* 种，中心的资源类型有 *m* 种，每类任务对于不同的资源需求不同，设第 *i* 类训练任务 *T* 对于第 *j* 类资源的需求为很高(*PB*)，即 T_{ij}= *PB*，如果当前为该虚拟机提供的资源为很低(*NB*)，即 $S_j = NB$ ，那么，资源调度的数目为正大(*PB*)，即 $f_{ij} = PB$ 。采用 Matlab 编程语言可将上述推理过程表示为：

If (T_{ij} is *PB*) and (S_j is *NB*) then (f_{ij} is *PB*)

按照上述定义和推理方式，可以得出基于模糊推理的资源调度规则表，如表 1 所示。

表 1 基于模糊推理的资源调度规则表

S_j	T_{ij}						
	NB	NM	NS	ZO	PS	PM	PB
NB	ZO	PS	PS	PM	PM	PB	PB
NM	NS	ZO	PS	PS	PM	PB	PB
NS	NS	NS	ZO	PS	PM	PM	PB
ZO	NM	NM	NS	ZO	PS	PM	PB
PS	NM	NM	NM	NS	ZO	PS	PM
PM	NB	NB	NM	NS	NS	ZO	PS
PB	NB	NB	NM	NM	NS	NS	ZO

表 1 中的各量为模糊量，根据相关解模糊算法[17]，即可得到系统解模糊后的数字量。通过模糊逻辑进行资源的调度，可以有效提高资源的利用率，提升服务质量。

3.3 系统动态容错技术

为了改善系统容错能力，需要一种稳定的容错策略。常用的检查点(Checkpoint)回卷容错方式存在一定的弊端：仿真程序或者进程本身出现的错误可以通过检查点回卷方式进行容错，但是当仿真节点所在的物理机失效或者崩溃后，那么周期性设置的检查点也会同时失去效果，无法进行有效容错。而在 IaaS 云训练系统中，可以通过虚拟化技术生成多台虚拟机，因此可采用这些虚拟机作为副本(Backup)，方便的为仿真系统提供备份，避免单个仿真节点出现错误而导致的整个分布式仿真系统崩溃的情况。由于数据备份需要在一定时间内进行心跳交互，如果心跳频率过快，会造成大量数据传输和存储的开销，而如果心跳频率过慢，则有可能影响系统容错效果。为减小容错开销，可引入虚拟机迁移(VM Migration)技术作为补充。

本文充分运用 IaaS 模式云计算的特殊优势，借鉴云计算中的容错策略[18-19]，结合检查点回卷容错策略、数据副本策略和虚拟机迁移技术，对仿真系统的容错机制进行了改进，给出了一种仿真系统的低开销动态自适应副本和检查点回卷相结合的容错方案，实现动态自适应仿真系统容错。

为便于分析仿真系统中发生的错误，根据仿真系统出现的故障级别，将仿真系统出现的错误划分为两大类故障[20]：应用层故障和资源层故障。所谓应用层故障是指仿真中间件、各种仿真应用及仿真进程出现的故障；而资源层故障是指在该仿真应用的运行过程中用到的资源，尤指物理资源出现的故障，比如计算机系统崩溃、系统宕机等故障。系统出现故障时，首先检查出现故障的级别，根据故障级别动态选择需采用的具体容错策略：当故障为应用层故障时，可以采用检查点回卷的容错方式，避免采用复制策略时产生的多余开销；而故障发生在物理资源层时，则采用数据副本的策略进行容错，通过物理节点的副本来保证仿真系统的可用性，从而弥补了检查点回卷策略的缺陷。如图 7 所示。

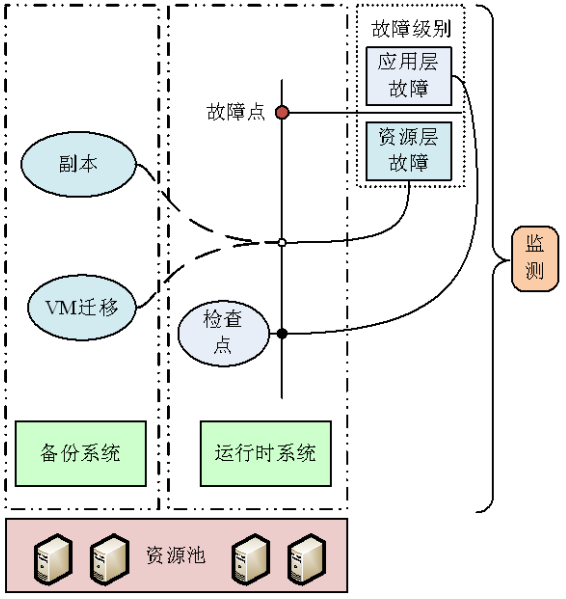


图 7 自适应容错系统结构示意

为实现容错策略的自适应选择，提高容错效率和性能，首先应根据用户数据交交互频率和交互量来确定用户 k 的活跃度 pd_k，然后确定总的数据交互量，根据交互量大小确定容错策略的选择。

为了计算节点 f_k 的活跃度 pd_k，首先定义遗忘参数 ω，它与该节点从初始时刻 t_s 到当前时刻 t_p 的数据收发间隔以及收发的频率呈指数关系，如式(7)所示：

$$\omega(t_p,t_s)=e^{-(t_p-t_s)^k},\ k\in\{1,2,...\} \tag{7}$$

令 $\Delta t=t_p-t_s$，则式(7)可简化表示为：

$$\omega(t_p,t_s)=e^{-(\Delta t)^k} \tag{8}$$

由式(8)可知，ω 的取值范围为[0, 1]。k 值为节点 f_k 在时间区间 Δt 收发数据的次数，决定了活跃度曲线在 Δt 时间内的衰减率。收发数据次数越多，k 取值越大，反之则变小。当 $(\Delta t)^k\to 0$ 时，$\omega=e^0=1$；当 $(\Delta t)^k\to\infty$ 时，$\omega=\lim\limits_{(\Delta t)^k\to\infty}e^{-(\Delta t)^k}=0$。

在初始时刻 t_s 到当前时刻 t_p 的间隔内，活跃度 pd_{bk} 为：

$$pd_k=\sum_{t_i=t_s}^{t_p}[an_k(t_i,t_{i+1})\cdot(1-\omega(t_i,t_p))] \tag{9}$$

式(9)中，$an_k(t_i,t_{i+1})$ 为在时间间隔 $[t_i,t_{i+1})$ 内数据的交互次数。由式(9)可知，节点数据交互次数越多，遗忘参数越小，则节点活跃度越高。

节点 k 的数据交互总量 rf_k，定义为节点的活跃度与给定条件下所有任务对仿真节点总数据量的需求之间的乘积，即：

$$rf_k=pd_k\cdot(rn_k\cdot fs_k) \tag{10}$$

其中：rn_k 和 fs_k 分别表示备份数量和节点文件的大小。

当活跃度低于最小阈值 $rf_{\min}$ 时，则采用检查点回卷方式进行容错；当活跃度处于最小阈值 $rf_{\min}$ 和最大阈值 $rf_{\max}$ 之间时，启动副本策略；当活跃度超过某个最大阈值 $rf_{\max}$ 后，则启动虚拟机迁移策略。如式(11)所示：

$$\begin{cases} Checkpoint & rf_{\min}<rf_k \\ Backup & rf_{\min}<rf_k<rf_{\max} \\ VM\ Migration & rf_{\max}<rf_k \end{cases} \tag{11}$$

其中，影响容错效果的可控参数主要为最小阈值 $rf_{\min}$ 和最大阈值 $rf_{\max}$。系统运行时，可根据实际运行效果进行调整，以保证系统容错效果。

4 系统验证

构建的训练系统原型系统主要配置如下：服务器采用 Dell PowerEdge R720；GPU 采用 Nvidia 公司基于 Kepler 架构的 Grid K2 显卡(以下简称 K2)，其显存容量为 8GB；网络环境为千兆级带宽交换机的局域网。该训练系统原型系统可允许 4 个用户同时训练，每个用户的虚拟机配置如下：操作系统采用 64 位 Windows 7 操作系统；CPU 为双核，主频为 2.2 GHz；内存为 4 GB。

为方便进行对比测试，创建用户组 1，采用本文介绍 GPU 虚拟化方法为用户虚拟机配置虚拟显卡。另外创建一组用户虚拟机，采用 CPU 模拟 GPU 方式作为虚拟显卡，命名为组 2。以 3D 游戏引擎 Virtools 作为测试软件，运行的 3D 模型为某型雷达模拟训练系统，根据需要，同时在虚拟机上加载模型大小为 50 MB，80 MB，120 MB，160 MB 和

200 MB。测试时,以软件运行帧速作为主要指标。测试结果如图 8 所示,可以看到采用本文所述 GPU 虚拟化技术后，帧速较普通“CPU+内存”模拟 GPU 的方式有很大的提升。原因主要有以下几点：(1) 普通模拟 GPU 方式通过 CPU 计算来处理图形，而 CPU 的计算能力与 GPU 是无法比拟的，因而虚拟 GPU 的处理能力受到了很大制约，其显存最大仅能达到 512 M; 本文所述的虚拟化方式直接对 GPU 资源进行调度，可以无损的将一个 8G 的 K2 显卡虚拟化为 8 个虚拟显卡,每个显卡内存可达 1024 M。(2) 普通模式对 CPU 计算能力消耗极大，因而 CPU 处理其他任务的能力会下降，因而会在一定程度影响系统运行帧速;而对 GPU 资源直接调度则降低了 CPU 的负载，不会过多影响 CPU 的任务处理能力。

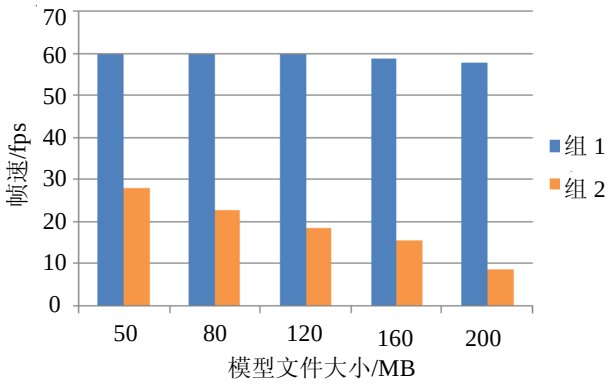


图 8　不同模式虚拟 GPU 环境下的运行帧速对比

图 9 所示为创建 5 个虚拟机时，采用不同策略时的任务处理时间。其中各柱状图分别代表：不采用资源调度策略(NRS)，采用负载均衡策略(LBS)，采用基于模糊控制理论的资源调度策略(FCTRS)时的任务执行时间。进行实验时，系统任务队列为先到先服务(FCFS)单队列。可见不采用任何调度策略时，任务所需处理时间最长，采用基于模糊控制的资源调度策略所需时间最短。这是因为不采用调度策略时，各虚拟机资源可用性一定，用户任务需求不能尽快满足；采用负载均衡策略时，各个处理单元可用资源根据任务长度进行了调整，虚拟机之间的执行时间差距减小；而采用基于模糊控制理论的资源调度策略后，根据任务需求等级动态调整各虚拟机之间的可用性资源，有效缩短了执行时间，各处理单元之间执行时间差距也有显著缩小。

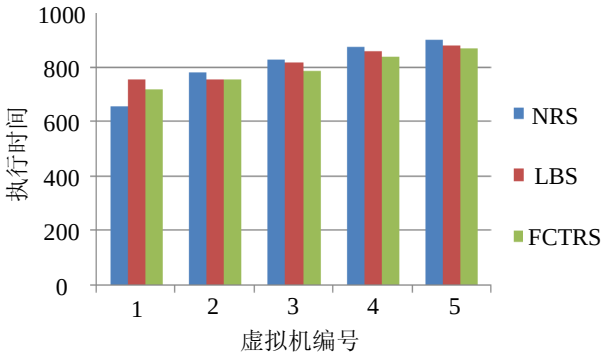


图 9　不同资源调度策略各虚拟处理单元的任务执行时间

上述实验结果是合理的：(1) NRS 几乎没有对执行效率进行任何优化，因而无论是最大执行时间跨度还是各个虚拟机执行时间跨度均最大；(2) LBS 主要作用在于均衡各个虚拟机的执行速度，因而任务之间的执行时间差最小，但是总的执行时间并未有太大改善；(3) FCTRS 并未对任务的执行时间平均化，而是根据用户任务和虚拟机之间的关系进行了最优化，缩短了总的执行跨度，提升了执行效率，而且兼顾了用户与云计算之间的友好性。

5　结论

IaaS 模式的云训练为解决部队传统模拟训练中的问题提供了一种借鉴与思路。文章分析了提升模拟训练效果的 3 个主要因素：虚拟机的 3D 图形处理能力、训练中心的资源利用率和系统可靠性与稳定性。在构建 IaaS 模式云训练架构的基础上，对 GPU 虚拟资源共享、虚拟机资源动态配置和系统容错进行了重点研究，并提出了解决方法。通过对原型系统进行实验测试，表明该系统在高可靠性基础上有效的提升了系统资源利用率，保证了终端用户的 3D 图形呈现能力，达到良好的训练效果。下一步的工作重点是将该系统进一步应用到部队的训练中，在实际应用中完善 IaaS 模式的云训练系统。

参考文献：

[1] Sean Marston, Zhi Li, Subhajyoti Bandyopadhyay, *et al*. Cloud Computing-The Business Perspective [C]// The 44th Hawaii International Conference on System Sciences. Hawaii, USA: IEEE Computer Society, 2011: 176-189.

[2] Ling Qian, Zhiguo Luo, Yujian Du, *et al*. Cloud Computing: An Overview [J]. Lecture Notes in Computer Science (S1804-2724), 2009, 5931(1): 626-631.

[3] 邸彦强, 朱元昌, 孟宪国, 等. 基于网格技术的多用户多任务模拟训练系统[J]. 系统仿真学报, 2008, 20(3): 643-647.

[4] Meng Xianguo, Di Yanqiang, Zhu Yuanchang, *et al*. System Design and Implementation:a Web-based Simulation Training System in Grid Environment [J]. Journal of System Simulation (S1004-731X), 2009, 21(4): 1202-1205.

[5] 黄安祥, 冯晓文, 李劲松, 等. 基于云计算平台的航空兵训练仿真体系结构[J]. 系统仿真学报, 2011, 23(S1): 106-109.

[6] Eva Pajorová, Ladislav Hluchý. Complicated Simulation Visualization Based on Grid and Cloud Computing [C]// The 7th International Conference on Cooperative Design, Visualization & Engineering. Mallorca: Springer Berlin Heidelberg, 2010: 211-217.

[7] 冯少冲. 基于云计算的武器装备网络化模拟训练支撑技术研究[D]. 石家庄: 军械工程学院, 2011.

[8] Su Min Jang, Won Hyuk Choi, Won Young Kim. Client Rendering Method for Desktop Virtualization Services [J]. Electronics Telecommunications Research Inst (S1225-6463), 2013, 35(2): 348-351.

[9] 钱琼芬, 李春林, 张小庆. 云数据中心虚拟资源管理研究综述[J]. 计算机应用研究, 2012, 29(7): 2411-2416.

[10] Asael Dror, Hao Zhang, B.Anil Kumar, *et al*. Virtualized GPU in a Virtual Machine Environment [P]. United States: US 20110102443A1, 2011.

[11] Jose Duato, Francisco D. Igual, Rafael Mayo, *et al*. An Efficient Implementation of GPU Virtualization in High Performance Clusters [J]. Lecture Notes in Computer Science (S0302-9743), 2010, 6043(1): 385-394.

[12] Micah Dowty, Jeremy Sugerman. GPU Virtualization on VMware's Hosted I/O Architecture [J]. SIGOPS Operation Systems Review (S0163-5980), 2009, 43(3): 73-82.

[13] Sunit Parmar, Aniruddh Kurtkoti. An Approach To Graphics Passthrough In Cloud Virtual Machines [J]. International Journal of Engineering Research & Technology (S2278-0181), 2013, 2(6): 746-749.

[14] Shinpei Kato, Scott Brandt, Yutaka Ishikawa, *et al*. Operating Systems Challenges for GPU Resource Management [C]// 7th Annual Workshop on Operating Systems Platforms for Embedded Real-Time Applications. Porto, USA: George Washington University, 2011: 23-32.

[15] 齐平, 李龙澍. 云环境下结合模糊商空间理论的资源调度算法[J]. 小型微型计算机系统, 2013, 34(8): 1793-1797.

[16] 赵淦森, 虞海, 季统凯, 等. 云计算平台的自适应资源供给[J]. 电信科学, 2012, 28(1): 31-37.

[17] Y J Huang, T C Kuo, H K Lee. Fuzzy-PD Controller Design with Stability Equations for Electro-Hydraulic Servo Systems [C]// International Conference on Control, Seoul, Korea. USA: IEEE, 2007: 2407-2410.

[18] Dawei Sun, Guiran Chang, Changsheng Miao. Analyzing, Modeling and Evaluating Dynamic Adaptive Fault Tolerance Strategies in Cloud Computing Environments [J]. Journal of Supercomput (S0920-8542), 2013, 66(1): 193-228.

[19] Bo Yang, Feng Tan, Yuan-Shun Dai. Performance Evaluation of Cloud Service Considering Fault Recovery [J]. Journal of Supercomput (S0920-8542), 2013, 65(1): 426-444.

[20] 刘云生. 大规模分布式仿真系统容错关键技术研究[D]. 长沙: 国防科学技术大学, 2006.