2009 IEEE International Conference on E-business Engineering

**虚拟化云计算环境中Web应用程序的动态扩展**

Trieu C. Chieu, Ajay Mohindra, Alexei A. Karve and Alla Segal

IBM T. J. Watson Research Center

19 Skyline Drive

Hawthorne, NY, USA

e-mail: {tchieu, ajaym, karve, segal}@us.ibm.com

**概述** - 可扩展性对于目前许多在Web网页上开展业务的企业，以及企业提供一些信息（可能在一定时间内变化很大）时的成功至关重要，但为了满足峰值要求而保留足够的资源所需的成本可能是昂贵的。云计算提供了强大的计算模型，这种模型允许用户按需访问资源。在本文中，我们将介绍一种基于虚拟化云计算环境中阈值的Web应用程序动态扩展的新体系结构。我们将使用前端负载均衡器来说明我们的扩展方法，这种负载均衡用于将安装在虚拟机实例中Web服务器上的Web应用程序的用户请求进行路由和平衡。我们也将介绍一种基于活动会话中阈值数量进行虚拟机资源自动配置的动态缩放算法，以及讨论按需进行云快速配置和动态分配资源给用户的能力。我们的成果已经展示了云的强大优势，它能够处理突发的负载浪潮，按用户需求提供IT资源，并保持更高的资源利用率，从而降低基础架构和管理成本。

**关键词**：云计算; 可扩展性；虚拟化; 虚拟机

**一、 引言**

云计算[1-3]正在成为越来越受欢迎的企业模式。云计算资源可根据需要按需提供给用户，其独特的价值主张为IT和业务目标的整合创造了新的机会。云计算本质上是一个强大的计算范例，通过网络访问的连接、软件和服务的组合来完成其中的任务。通过与VMware ESX Server [4]和Xen [5]等服务器虚拟化软件相结合的分布式大型计算集群和并行处理，云计算的巨大处理能力成为可能。这个服务器和连接网络统称为云。计算资源可以在客户端企业内维护，或由服务提供商提供。云计算可以让用户达到超级计算机级的计算能力。用户可以在他们需要的任何时候访问庞大的弹性资源。因此，云计算也被描述为按需计算。

按需模型是为了克服来自需要能有效满足波动需求的企业的共同挑战而开发的。云计算的模式是从以下实用计算、自主计算、网格计算和软件即服务（SaaS）的概念发展而来[6]。例如，效用计算是将外包计算资源与基于使用的支付结构的基础架构管理设计相结合的按需方法。由于企业对计算资源的需求可以从一个时间到另一个时刻大不相同，因此要保持足够的资源以满足峰值要求的成本可能是昂贵的。相反，如果企业通过只维护最少的计算资源来降低成本，或许将没有足够的资源来满足峰值要求。

另一方面，动态扩充和增加资源的可扩展性或能力对于许多目前在网络上开展业务并提供可能会突然被大量需求的信息的企业的成功与否至关重要。虽然IT部门可以采取很多策略来为更多的客户需求提供服务，但他们设计和实施的方式对业务本身不能确定是好是坏。云计算提供了一个强大的环境来轻松扩展Web应用程序。事实上，云计算可以为Web应用程序需要的许多典型扩展点按需求提供不同的资源，包括服务器，存储空间和网络。云计算的按需性与现收现付模式相结合，这意味着随着应用需求的增长，服务需求所需的资源也越来越多。在这种情况下，只要应用程序设计正确，其体系结构适合扩展，容量就能满足需求。

在本文中，我们将介绍一种动态扩展方案，其中包含一种Web应用程序在动态安装于云端的虚拟机实例中部署的新设计方法，以说明云计算环境的强大扩展功能。 首先我们将描述云计算和可扩展Web应用程序的各种扩展指标。 然后，我们将讨论云的可扩展功能及其虚拟化技术的使用。接下来我们将通过一种扩展场景，以及在这种场景下基于Web应用程序登录用户数量的在线应用程序和动态扩展算法来介绍我们全新的架构设计。

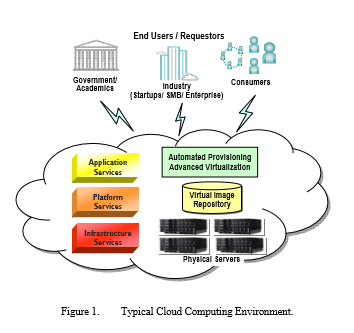
本文其余部分的大纲组织如下所述。 第1节为引言； 第2节介绍了云计算架构和虚拟化； 第3节阐述了在线Web应用程序的扩展场景的架构设计，将对基于登录用户数量的缩放算法进行描述； 相关工作将在第4节讨论；最后，第5节总结全文。

**二、 服务的云计算，虚拟化和可扩展性**

云计算提供了根据需要添加容量的能力，通常具有非常小的提前期。 显然，云计算提供了一个新的引人注目的机制来处理需要可扩展的应用程序服务。 以下部分将简要介绍云计算，虚拟化的关键交付技术，云中Web应用程序的可扩展性和扩展指标。

A．云计算

云计算是通过网络提供服务的一种方式。这种方式将虚拟化技术[7,8]，处理器，磁盘存储，宽带互联网接入以及快速，便宜和强大的服务器的先进技术结合起来，使云计算成为现实中倍受关注的范例。云计算基本上使用虚拟化技术将计算机资源转化为客户虚拟机。这些虚拟机通常位于主机环境中某些网络的物理服务器上。然而，客户虚拟机可以四处移动，从而破坏与物理机器相关联的直接硬件依赖性。随着硬件依赖性不再是问题，客户系统可以与硬件故障隔离，并自动转移到另一个可用的硬件。如图所示，具有自动配置功能、高级虚拟化技术和托管在物理服务器上的虚拟机的典型云计算环境向用户直接提供云应用服务。



在云计算中，新的应用程序通过高效的虚拟化计算，可以以一种安全的方式快速缩放规模来提供高质量的服务。 用户可以通过连接设备随时随地访问他们的应用程序。 用户只看到服务，而不是其交付所需的实现过程或基础设施。 云服务的案例包括诸如存储、数据保护、应用程序、业务流程以及对业务和消费者服务（例如电子邮件和一些办公应用）的技术服务。 云计算允许用户和公司在任何需要时使用他们需要的服务和存储空间，随着无线宽带连接选项的扩充，用户在任何需要的地方都可以使用相应的服务和存储。 可以根据服务器利用率，过程中的功耗或使用的带宽对客户进行计费。

因此，云计算有可能完全颠覆软件行业，因为应用程序是通过网络而不是用户桌面进行购买、许可和运行的。 这种转变将使数据中心及其管理员处于分布式网络的中心，因为处理能力，带宽和存储都是远程管理的。 以下是云计算服务的几种典型类型，具体取决于产品的性质：

* 应用程序服务 - 任何Web应用程序都是云应用程序服务，它位于云中。 Google，Amazon，Facebook，Twitter，Flickr以及几乎所有其他Web 2.0应用程序都是这个意义上的Cloud应用程序。
* 平台服务 - 纯实用计算中的一个步骤就是云平台服务，如Google Apps和Google AppsEngine，以及Salesforce的force.com，它们将虚拟机实例隐藏在更高级别的API之上。
* 基础设施服务 - 亚马逊弹性计算云（EC2）是一种典型的云基础架构服务，它可以按现收现付电价格提供原始虚拟机实例、存储和计算，并且是目前该类别的主要供应商。 开发商是这种云计算服务的典型目标。

B.虚拟化技术

虚拟化技术[7,8]是云计算中的关键交付技术。 在云中，虚拟化主要涉及平台虚拟化，或者是使用它们的用户和应用程序中物理IT资源的抽象化。 虚拟化允许将服务器，存储设备和其他硬件视为资源池而不是离散系统，以便能够按需分配这些资源。 通过一种被称为虚拟机管理程序或虚拟机监视器的控制软件，在给定硬件平台上执行平台虚拟化。 该软件为其客户软件创建一个称为虚拟机的模拟计算机环境。 客户软件通常本身就是一个完整的操作系统，就像安装在独立的硬件平台上运行一样。 目前领先的虚拟化和软件供应商包括VMware [4]，Xen [5]，KVM [9]，Force.com [10]和Microsoft Virtualization [11]。

作为物理资源封装的手段，虚拟化解决了数据中心管理面临的几个核心挑战，并提供了具体的优势，包括：

* 更高的利用率：通过虚拟化，工作负载可以封装并转移到空闲或未充分利用的系统中。 这意味着现有系统可以合并，因此可以延迟或避免购买额外的服务器容量。
* 资源整合：虚拟化允许整合多个IT资源。 除了服务器和存储整合之外，虚拟化还提供了对系统架构、应用程序基础设施、数据和数据库、接口、网络、桌面甚至业务流程整合的机会，从而节省成本并提高效率。
* 较低的电力使用/成本：使用虚拟化进行整合，可以节省总功耗并节约大量成本。
* 节约空间：服务器蔓延在大多数数据中心仍然是一个严重的问题，但对于数据中心扩展昂贵的构建成本和冷却成本，这不总是一个好的选择。而 虚拟化可以通过将许多虚拟系统整合到更少的物理系统上来缓解压力。

C.应用程序的可扩展性

可扩展性对于许多目前在网络上实现业务并提供可能会突然被大量需要的信息的组织的成功来说至关重要。 可扩展性是衡量应用程序扩展以满足企业业务需求的能力的量度。 一般来说，企业可以通过在需要时添加更多或更大的资源来扩展给定的应用程序。 资源可以是许多东西，包括服务器，处理器，存储和网络带宽。 可扩展应用程序能够在扩展同时正常运行，并可随时添加更多资源来满足更多客户需求。 不可扩展的应用程序可能在需求增加时遇到性能和服务可用性问题，这时不可扩展的应用程序可能无法利用更多的资源。

D.扩展指标

虽然不同的Web应用程序可能不会以相同的方式执行，但是资源被限制的可扩展点非常相同。例如，Web应用程序不大可能在单个服务器上扩展以满足业务需求。在某些时候，需要增加更多的服务器来满足日益增长的需求，进而满足服务质量要求。存储、网络和其他扩展点通常也是如此。了解Web应用程序如何消耗资源，及其在高负荷下的表现以及整个系统的潜在扩展点会发生什么变化，对维护Web应用程序的理想性能至关重要。一种了解应用程序在增加的客户负载下如何执行的方法是探究扩展点的位置，并使用容量规划来衡量这些点上的负载，以便在需要时添加更多资源。精心设计的Web应用程序可以让客户在需求增加时，为每个扩展点添加更多的容量，从而提高能力以适应不断增长的需求。为了以动态方式扩展应用程序，通常在扩展点利用一些扩展指标来监视和跟踪性能。对于Web应用程序，Web服务器上的典型扩展指标可能包括：

* 并发用户数
* 活动连接数
* 每秒请求数
* 每个请求的平均响应时间

一旦扩展指标被选择用于扩展应用程序，就会实时收集扩展指标的采样，并定期计算统计数据。根据扩展指标的统计资料得出的历史趋势和预测，可以定义缩放规则来扩大或缩小Web应用程序实例的数量。

**三、 在云中扩展WEB应用**

在下面几节中,我们将提出我们的扩展方案，一种新颖的架构设计通过动态扩展算法来扩展一个安装在云上虚拟机中的web应用程序。该方案是基于使用前端负载均衡器动态地将用户请求路由到后端web服务器主机上的web应用程序。web服务器的数量应根据当前活动会话的数量阈值，自动地在每个web服务器实例中扩展以维护服务对于质量的需求。

A．架构设计

我们考虑这样一个场景, 在任何时间，对一个给定的web应用程序在互联网上提供优质的网络服务。因为可能有潜在的无限数量的用户，所以访问该应用程序的请求数量是不可预知的。通常这种类型的工作负载需求应用程序短响应时间以及高度的可靠性和可用性的。因此，不管多少数量的并发用户访问系统，web应用程序应该提供没有停机时间和保障最快的响应时间。

这样的web应用程序的主要问题是不能提前计划没，特别是预测用户将访问的数量。一种解决方案是通过动态管理web应用程序的方式，根据需求让web服务器和web应用程序组件的数量增加(或减少)。一个可伸缩的架构可以有效地处理这种情况。见图2使用虚拟化的云计算环境。架构设计包括前端负载平衡器，包括许多web应用程序的虚拟机,配置子系统和具备动态扩展算法的服务监控子系统。

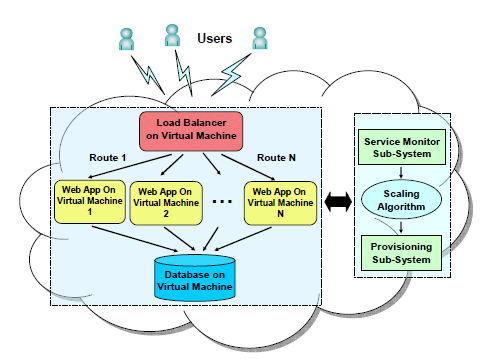


图2.云上扩展WEB应用的架构

基本上,前端负载平衡器是一个Apache HTTP负载均衡器,作为一个单一的web请求,入口点，用于将请求路由到web服务器主机上相同的web应用程序。web应用程序被部署于安装在Linux虚拟机上的Apache HTTP服务器。这些虚拟机根据需要通过配置子系统在云基础设施上配置和启动。

为了控制配置或解除的web服务器虚拟机实例行为，一种基于相关阈值或web应用程序的扩展指标的动态扩展算法被开发出来的。这里的扩展指标指的是在每个web应用程序上，我们选择活动会话或登录会话的数量。据我们对于性能和可伸缩性的测试结果，这里的扩展指标指是一个影响我们部署在Apache HTTP服务器的Linux操作系统中web应用程序性能的关键。在这项研究中,我们发现web应用程序只支持大约40000活跃并发会话,当活动会话数量超过这个阈值仍然增加时,web服务器系统将会开始失败的响应用户请求,然后最终将事件冻结起来。

B．负载均衡器与Web-App虚拟应用

使用如Apache HTTP负载均衡器的类似负载平衡器将允许传入的HTTP请求路由到web服务器主机上的web应用程序。因为Apache HTTP负载均衡器配置可以在系统运行时更新,这使得系统可以自动以及动态添加新的web服务器实例。附加的web服务器启用系统来扩展,从而对于传入的HTTP请求提供更好的响应时间。

为了简化配置额外的资源,web应用程序和其相应的web服务器预先安装在一个虚拟机设备应用镜像模板，并可在云上的镜像仓库中获得。通过系统命令使用相应的应用镜像模板,web应用程序的新实例可以十分容易的配置与创建。

C．服务监视与配置子系统

服务监控子系统负责收集单个web应用程序的比例指标,然后计算他们的平均偏移。在我们目前的设计中,单个web应用程序上的活动会话数量将作为比例指标来使用。监控代理将安装在每个web应用程序上来跟踪活动会话的数量，并定期转发数量给服务监控子系统。

基于扩展的平均偏移,一种动态扩展算法将在下一节中给出,用于触发扩展事件给配置子系统。根据更新的统计显示, 规模向上或向下的行为可能会启动。规模上升或下降意味着一个事件，将触发指示配置子系统启动或关闭运行在云中的虚拟机服务器实例。在扩展web应用程序的情况下,新启动的虚拟机实例将会运行web应用程序。一旦web应用程序实例启动准备就绪,前端负载平衡器将更新配置文件更新，并刷新将它们分配为活动的服务。

供应子系统基本上是通过IBM Tivoli Provisioning Manager(TPM)软件产品[12]构造。这种TPM软件组件通常用于自动完成人工任务，在服务器集群上启动和配置服务器、操作系统、中间件、应用程序、存储和网络设备。为了启用和管理在云中的网络服务器和虚拟机,多种的TPM自动化工作流程已经被开发出来以应对不同的操作系统和虚拟化技术。这些工作流程作为TPM的插件工具以web服务api的方式来被云中的其他组件使用。

D．基于镜像的配置

基于镜像的配置是一种部署和激活机制，它通过克隆一个“真实”的虚拟镜像来创建新的虚拟机实例。克隆虚拟映像的挑战之一是处理不同操作系统、网络和应用程序中特殊的定制。从一个“真实””镜像模板[13]自动化配置新虚拟机可以通过添加自动化功能到模板镜像,并结合外部的自动化脚本来控制部署。

通过我们的工作, 在Red Hat Enterprise Linux系统[5][14]中，基于镜像，我们执行自动化的基于Xen hypervisor技术的Linux虚拟应用的配置。我们利用一个简化的配置过程,它是一种利用Linux桌面镜像挂载磁盘到克隆的Xen镜像，在启动前修正主机名与网络。整个过程可以归纳为两个简单的阶段：

第一阶段是复制“真实”的镜像应用模板到目标Linux主机系统上,主机系统挂载镜像文件,在挂载的文件系统中执行修复主机名和网络。修复涉及的第一步是修改/etc/hosts文件，/etc/sysconfig/network文件和/etc/sysconfig/networkscripts/ifcfg-ethx文件为正确的新主机名和网络设置实体。第二步修复是复制主机系统的RSA公钥到虚拟机的/root/.ssh/authorized\_keys文件中。前一步将允许新的虚拟机启动新的主机名和静态IP地址，从而允许外部网络访问；后者将允许主机系统远程执行虚拟机的secure shell脚本,而不需要提供密码。第二阶段的配置过程涉及到启动虚拟机和VM上secure shell脚本的远程执行，密码更改以及应用的特殊配置。

E．扩展算法

正如前文所言, 扩展算法是通过服务监控子系统实现的,用于在服务监控子系统中控制和触发基于扩展统计指标地扩大或减少虚拟机实例的数量。我们的动态扩展算法,在图3给出了伪程序,基于人工智能在每个扩展指标云中的虚拟机实例。为了演示, 在我们的实现中我们选择每个实例的web应用程序的活动会话数量作为扩展指标。

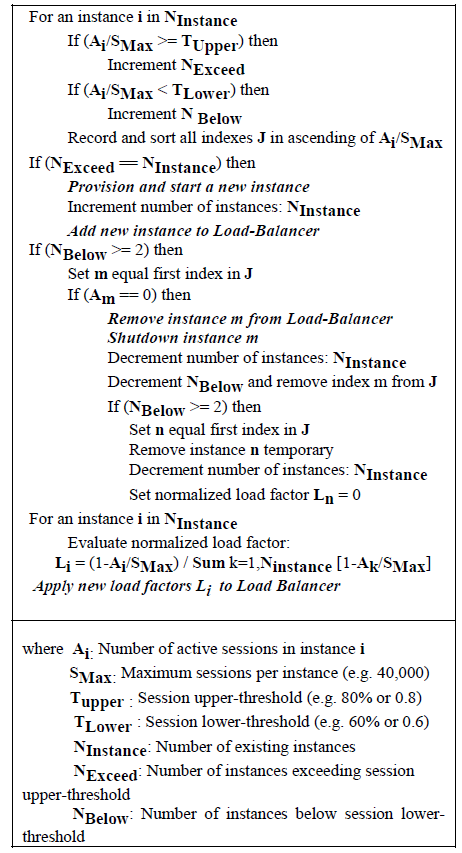


图3.对于云中虚拟机实例的动态扩展算法

算法首先判断当前的web应用程序实例与活跃的会话数量是否高于或低于给定的阈值。如果所有实例有活跃的会议超过给定的阈值,将启动一个新的web应用程序实例,然后将其添加到前端负载平衡器中。如果有与活跃的会话实例低于给定的阈值低并且至少有一个实例没有活跃的会话,这个空闲实例将从负载平衡器删除并关闭系统。在所有情况下,所有活动实例都将重新计算负载因子,然后应用到负载平衡器中来重新分配请求处理到每个实例中。

**四、相关工作**

在过去的十年中,分布式系统的目标都集中在接口与实现的解构,代理模型,基于订阅的计算、面向服务架构(SOA)[15]和社会协作上。最近,依托互联网分布式、多租户[16]应用来连接到内部业务应用程序,被称为软件即服务(SaaS)[6]，越来越受欢迎。企业利用互联网中心托管数据，由第三方供应商的虚拟云计算环境来缓解硬件、软件、维护、可用性、可靠性和可伸缩性的担忧。

特别地,可伸缩性是关于云计算讨论中的主要话题之一。可伸缩性是一种应用程序通过在一个服务器池上扩展以满足复制和分发请求的需求。web应用程序的动态可伸缩性在目前的虚拟云计算概述工作中还没有得到太多讨论。大部分之前的网络可伸缩性工作[19]已经被报告通过静态负载均衡和服务器集群的解决方案来实现。

**五、结论**

总而言之,我们提出了一个扩展方案来解决web应用程序在虚拟云计算环境下的动态可伸缩性。系统由前端负载均衡器来均衡分配用户请求路由到相应的部署在云中虚拟机实例的web服务器上的web应用程序。也介绍了一种动态扩展算法基于活动会话的数量自动配置虚拟机资源。我们的工作已经证明了云的引人注目好处是能够处理突然激增的负载需求,并按需以一个更好的和更便宜的方法向用户提供IT资源。自动配置、动态分配以及云的快速扩展功能的基本要素是提供更高的资源利用率,从而减少基础设施和管理成本。

**引用**

[1] G. Gruman, "What cloud computing really means", InfoWorld, Jan. 2009.

[2] R. Buyya, Y. S. Chee, and V. Srikumar, “Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities”, Department of Computer Science and Software Engineering, University of Melbourne, Australia, July 2008, pp. 9.

[3] D. Chappell, “A Short Introduction to Cloud Platforms”, David Chappell & Associates, August 2008.

[4] VMware ESX Server, VMware Inc., <http://www.vmware.com/products/vi/esx/>

[5] Xen Hypervisor, <http://www.xen.org/>

[6] E. Knorr, “Software as a service: The next big thing”, InfoWorld, March 2006.

[7] Virtualization Technology, <http://www.kernelthread.com/publications/virtualization/>

[8] VMware Inc., "Understanding Full Virtualization, Paravirtualization, and Hardware Assist", VMware, 2007, <http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf>.

[9] Kernel-based Virtual Machine (KVM), http://www.linuxkvm. org/page/Main\_Page

[10] Force.com, <http://www.salesforce.com/platform/>

[11] Microsoft Virtualization, <http://www.microsoft.com/virtualization/>

[12] IBM Tivoli Provisioning Manager Products, <http://www.ibm.com/software/tivoli/products/prov-mgr/>

[13] L. He, S. Smith, R. Willenborg and Q. Wang, “Automating deployment and activation of virtual images”, IBM developerWorks, Aug. 2007. http://www.ibm.com/developerworks/websphere/techjournal/0708\_he /0708\_he.html

[14] Red Hat Enterprise Linux, <http://www.redhat.com/rhel/>

[15] T. Erl, “Service-oriented Architecture: Concepts, Technology, and Design”, Upper Saddle River, Prentice Hall, 2005.

[16] F. Chong, G. Carraro, and R. Wolter, “Multi-Tenant Data Architecture”, Microsoft Corporation, 2006.

[17] V. Ungureanu, B. Melamed, and M.Katehakis, “Effective Load Balancing for Cluster-Based Servers Employing Job Preemption”, Performance Evaluation, 65(8), July 2008, pp. 606-622.

[18] L. Aversa and A. Bestavros. “Load Balancing a Cluster of Web Servers using Distributed Packet Rewriting”, Proceedings of the 19th IEEE International Performance, Computing, and Communication Conference, Phoenix, AZ, Feb. 2000, pp. 24-29.

[19] V. Cardellini, M. Colajanni, P. S. Yu, “Dynamic Load Balancing on Web-Server Systems”, IEEE Internet Computing, Vol. 33, May-June 1999, pp. 28 -39.