

基于 NGPD 的 PaaS 平台研究与实现*

谢浩晖, 高 济

(浙江大学 计算机科学与技术学院 计算机应用技术系, 杭州 310027)

摘要: 云计算是一种将计算和存储任务分配到由大量计算机构成的云中的计算模式。云计算提供三种类型的服务:基础设施服务(IaaS)、平台服务(PaaS)和软件服务(SaaS)。虽然 PaaS 在云系统中起到了关键的作用,但是目前的 PaaS 平台存在着服务协同性低、资源匮乏、用户界面不友好等问题。设计并实现了基于准则调控和策略驱动的自治式服务协同模型(NGPD)的 PaaS 平台,其利用自治计算元素实现了自治式服务协同,这是尝试解决现有 PaaS 平台支持提供云平台型服务不足的探索性研究,为云用户提供了更加高效和经济的平台开发环境。

关键词: 云计算; 平台即服务; 服务协同; 自治计算元素; 策略驱动; 准则调控

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2011)05-1839-04

doi:10.3969/j.issn.1001-3695.2011.05.070

Research and implementation of PaaS platform based on norm-governed and policy-driven autonomic model

XIE Hao-hui, GAO Ji

(Dept. of Computer, College of Computer Science & Technology, Zhejiang University, Hangzhou 310027, China)

Abstract: Cloud computing is a computing paradigm that it assigns the computing and storing task to the cloud which is composed by a large number of computers. Cloud computing provides three types of services: infrastructure as a service, platform as a service and software as a service. While PaaS in cloud system plays a key role, there is now PaaS platform for low service coordination, lack of resources, unfriendly user interface and other issues. This paper designed and implemented PaaS platforms based on norm-governed and policy-driven autonomic model of service cooperation, which was using autonomic computing elements to achieve a collaborative self-service and was an exploratory study trying to solve the problems of existing platform to support PaaS, aiming to provide more efficient and economic development environment for the users.

Key words: cloud computing; platform as a service; service cooperation; autonomic computing element; policy-driven; norm-governed

0 引言

云计算是对分布式处理(distributed computing)、并行处理(parallel computing)和网格计算(grid computing)及分布式数据库的改进处理^[1]。云计算包含三个层次的服务,从下至上依次为:由底层硬件或虚拟机资源构建的基础设施服务(IaaS);构建在云基础设施上,提供云计算应用服务开发平台和环境的平台服务(PaaS);基于云平台开发的各类应用服务的软件服务(SaaS)^[2,3]。

虽然这三个层次都基于 Web 技术,但是它与传统应用的结构有着非常多的相似之处,图 1 对比说明了云计算技术的三层结构和传统应用的技术结构。在云计算体系的三层结构中,PaaS 层起到了关键的作用。PaaS 层作为系统软件的角色可以为上层提供丰富的 API,以支持多元化的 SaaS 应用开发。PaaS 层的目标旨在不仅能像传统的系统软件一样提供一些基本的 API,还要提供更为先进的即面向服务 API^[4],上层的应用软件开发者可以非常容易地使用这些服务接口为终端用户设计特

定的应用。目前具有代表性的 PaaS 平台有 Bungee Connect、Google App Engine、Salesforce.com 等。由于目前对于 PaaS 的研究和实现相对较少,利用现有的 PaaS 平台向云用户提供 PaaS 型服务存在以下一些问题^[5]:

a) 现有的 PaaS 平台虽然可以通过网络互联实现网格环境下的服务协同^[6],但这种形式的服务协同主要是基础设施层计算、存储等资源的协同,缺乏对基于基础设施开发的不同功能独立平台以服务形式提供给用户的支持。



b) 共享服务可能导致的低可用性。现有的 PaaS 平台采用共享服务模式,用户环境边界模糊,所有用户登录服务阵列后共享同一个配置和运行环境。不同用户的系统环境需求和操

收稿日期: 2010-10-11; 修回日期: 2010-11-28 基金项目: 国家“863”高技术研究发展计划资助项目(2007AA01Z187)

作者简介: 谢浩晖(1987-),男,浙江杭州人,硕士研究生,主要研究方向为人工智能和网络计算(1051477@qq.com);高济(1946-),男,教授,博导,主要研究方向为人工智能、网络计算、软件工程、信息智能。

作之间可能存在冲突,从而降低服务阵列以及平台的可用性。

c) 缺乏友好、高效的应用开发环境。现有的 PaaS 平台通常使用命令行环境进行程序开发(如程序编辑、编译、调试等),并且大多都不支持线下开发和应用开发同步上传,而大部分云用户不习惯使用非图形化界面进行应用开发。

针对以上问题,本文提出了基于 NGPD(norm-governed and policy-driven autonomic model of service cooperation)的 PaaS 平台——NGPD-PaaS 平台。NGPD-PaaS 平台的目的在于为云用户提供更好的 PaaS 型服务,同时解决现有 PaaS 平台服务协同性较低、开发环境受限的问题;解决引入自治服务计算后产生的“可信”问题。除此之外,平台还向云用户提供了易用性较好的图形化云应用开发环境等。

1 PaaS 平台

PaaS(平台即服务)旨在抽象出目标工具和应用的复杂性,为不同应用共享一个开发平台和常用服务,包括认证、授权和计费。因此,一个 PaaS 平台应该具备以下三个特征^[7]:

a) 网络资源中心。目前,互联网产业的竞争不再是单纯的技术竞争,而是资源的竞争,包括用户、网络组织和在线服务等。PaaS 平台的目的在于以外在增值服务的形式提供 Internet 资源。一些著名的互联网公司重新整合了它们长期积累的自有资源并发布了 PaaS 平台,为特定领域提供外在增值服务。除此之外,其他商业公司也开放了它们的资源,使得平台拥有更多的资源,加速了 PaaS 的开发。

b) 良好的开放性。PaaS 是互联网开放性的具体体现,但在早些年,互联网的资源只对终端用户提供,很少向第三方开发者开放。随着互联网资源的增加和价值的提升,拥有资源的企业不再开发具有附加价值的应用。与此同时,具有开发附加价值应用能力的第三方开发者对于开放资源的要求更为强烈。因此,一部分资源拥有者以 Web API 的形式开放了资源,在其设计、操作和控制过程中,PaaS 平台也应运而生。对于第三方开发者来说,他们不需要积累大量的互联网资源,只需要调用 PaaS 平台提供的 Web API 就可以根据终端用户的需求去提供相应的应用。丰富的网络应用极大地提升了 Internet 资源的价值,这也是 PaaS 平台最为成功的特性所在。

c) 拥有丰富的 Web API。Web API 主要是为第三方开发者所提供的,其通过标准的 Web 协议或技术独立于特定的语言和系统环境。PaaS 平台采用统一标准的 Web 协议封装 Web API,其中大多数都是基于 HTTP 协议的。PaaS 平台的本质在于将服务抽象成为接口,为第三方开发者提供商业资源和开发平台。通过 PaaS 平台的支持,云计算开发者拥有了大量的含有特定业务逻辑的可编程元素,使得应用的开发变得十分便捷,这也给终端用户带来了实在的好处。

2 NGPD-PaaS 平台的设计与实现

NGPD-PaaS 平台主要针对以下四类使用者进行了开发和设计:

a) 平台所有者。平台所有者需要向平台提供充足的 Internet 资源(典型的是基于 IaaS 模型基础之上)以满足客户的需

求,并且提供一些适当的应急响应。

b) 平台开发者。平台开发者需要创建一个合适的开发环境,使得绝大多数的第三方开发者在不需要很深的应用域背景知识以及前端客户端开发或者网站管理的经验下可以方便地创建 Web 应用。

c) 第三方开发者。第三方开发者需要掌握使用基于浏览器的开发环境,无缝部署运行时环境以及使用管理和监控工具的能力。

d) 服务消费者。服务消费者需要向平台提出调用应用的申请,提供相应的服务寻求信息,在使用相关应用后根据应用满足消费需求的程度向 PaaS 平台提供反馈。

2.1 NGPD 模型

目前的面向服务的架构(SOA)已经从以 WSDL、UDDI 和 SOAP 为核心的,面向 Web 服务描述、发布与使用的第一代技术,增强到促使应用系统能够以基于服务协同(面向软件服务的协同)的虚拟组织(virtual organization, VO)形态,按需、快捷和低成本地组建和演化^[8,9]。本文提出的 NGPD-PaaS 平台也正是基于 SOA 架构的云计算体系结构所建立的^[10]。为了解决网络服务面临的非自治性,引入以高度自治的理性实体——自治计算元素(autonomic computing element, ACE)作为网络服务的代理,可以实现自治式服务协同。理性的 ACE 具有依据感知、信念(知识)推理、决策合理目标及行动的能力,使得应用系统能够合理、灵活、快捷地实现自我管理,包括自组织(配置)、自优化、自修复(适应)和自保护。

自治式服务协同面临“自治”和“可信”需求相互冲突的两难困境:相互陌生的 ACEs 动态组建协同系统时,自治性和“黑箱”性将导致 ACEs 行为难以为系统可靠地预测和控制,从而产生对于 ACEs 能否协同完成全局目标的“可信”危机,严重影响了自治式服务协同的实用化。因此,规范调控(norm-governed)和策略驱动(policy-driven)的自治式服务协同模型 NGPD 使得 ACEs 的协同行为因接受规范的约束而变得可控、可预测,进而可信,又给 ACEs 的理性自治留下足够空间^[11]。NGPD 有两个层次如图 2 所示:宏观调控、微观行为,经由宏—微连接模式耦合而成。模型旨在建立 E-机构去指导和调控 ACEs 间服务协同关系的按需、动态组建、实施和维护,并驱动 ACEs 的协同行为接受 E-机构规范的宏观调控,从而变得可预测和可信,又不失高度自治性。



图2 规范调控策略驱动的自治式服务协同模型
NGPD 表示为三元组:NGPD = (MG, MB, LM)。

MG 为宏观调控 (macro-governing) 层,其设计应用域 E-机构作为宏观调控 ACEs 开展服务协同的标准和规范,并提供面向社交促进的 E-机构去规范社交促进型 ACEs 提供的协同促进服务。

MB 为微观行为 (micro-behavior) 层,其建立策略驱动的 ACE 行为自主管理机制,去实现“宏观调控→微观行为”的映射,使 ACE 面向服务供需的协同行为接受 E-机构的宏观调控。

LM 为宏—微连接模式 (link-mode),用于支持 E-机构的操作化,使得 E-机构的宏观调控作用能够施加到 ACE 个体,并驱动 ACEs 理性遵从协同行为规范。

2.2 NGPD-PaaS 平台的体系结构

NGPD-PaaS 平台是现有 PaaS 平台的延伸和扩展,其体系结构如图 3 所示。平台从上至下分为应用开发层、平台服务层、基础设施层。下面将分别对平台体系中各层的构成和功能进行介绍。



图3 NGPD-PaaS平台体系结构

2.2.1 NGPD-PaaS 平台应用开发层

NGPD-PaaS 平台的目标是为用户提供操作方便、易用性好的图形化云应用开发平台。NGPD-PaaS 平台为用户提供了标准的 Web 集成开发环境,它借助统一的 Web API 调用平台的服务制定、编译、调试等服务,在统一图形用户界面下支持程序编辑、编译、调试和加载全过程。应用开发层提供了针对监控和 SLA 管理服务的 Web 界面,开发者可以根据服务消费方使用的反馈配置应用有关政策。应用开发层允许开发者使用线下编辑开发工具,开发者可以线下完成应用开发的一部分,然后再同步上传到平台,有助于提高开发者的程序开发效率。

图 4 从第三方开发者的角度显示了 NGPD-PaaS 平台所提供的图形化界面和实现的部分功能操作。



图4 使用NGPD-PaaS平台的部分图形化操作

a) 在线服务提交申请。图 4(a) 是平台为第三方开发者提供服务申请的 Web 界面。第三方开发者需要向开发平台提交应用域 E-机构、服务应用、角色等申请应用服务信息。

b) 线下服务政策编辑。图 4(b) 是平台提供的线下支持工具之一,第三方开发者可以根据应用使用的反馈来制定相应的政策,并通过 Web 界面同步上传到平台,使得代理开发者开发

的应用服务的理性 ACE 能够根据相应的政策驱动接受调控。

2.2.2 NGPD-PaaS 平台服务层

NGPD-PaaS 平台将被作为服务提供给用户,即将应用和系统软件以服务的形式提供给用户,支持基于 NGPD-PaaS 平台的各种远程应用的开发。平台服务层又被细分为服务工程和服务管理两个部分(图 3)。

1) 服务工程

服务工程由 Web API、理性 ACEs 和应用 QoS 库组成。Web API 采用了扩展的 SOAP 协议,对 NGPD-PaaS 平台各类应用服务进行了抽象,向用户提供了统一的和平台环境交互的抽象接口,同时服务工程还为服务管理提供了相应的连接接口。第三方开发者通过调用 Web API 可以开发满足特定需求的应用,并将应用通过作为服务代理的理性 ACE 提供给服务消费者。

理性 ACEs 是服务工程的核心,由 ACE 知识、ACE 信念和元知识所构成。协同(契约履行)情境、服务供需适用情景、角色承担契约、服务契约和熟人构成 ACE 的信念;管理策略、本地业务过程、应用域 E-机构制定的协同行为规范和本地业务指令作为 ACE 的推理知识;策略描述本、BDI 模型本体、应用域本体、社交促进本体则构成了理性 ACE 的元知识。通过设置的 ACE 行为管理策略,并将协同行为规范、本地业务指令和 ACE 对于当前协同状态的信念作为应用管理策略决策协同行为的依据,能够有效地实现策略驱动的“宏观调控→微观行为”的映射,使得 ACE 可以通过让其社交行为始终接受行为规范(和本地业务指令)的宏观调控,成为可信的理性个体。

应用 QoS 库的建立旨在要求平台所有者(第三方开发者)发布应用时需要向平台提交 QoS 证书申请,SLA (service level agreement) 管理角色收到申请后根据相关策略对质量参数进行检验,检验通过后颁发 QoS 证书并在应用 QoS 库中进行注册。当应用服务出现与 QoS 证书描述不符、证书有效期到期等情况时,应用 QoS 库会撤销应用的 QoS 证书,并告知服务管理 SLA 管理角色停止该应用。

2) 服务管理

为了更好地支持服务协同,解决引入自治计算而产生的“可信”问题,服务管理根据 NGPD 模型的社交促进 E-机构定义了四个社交促进型角色:SLA 管理 (authority)、中介 (match-maker)、仲裁 (arbitrator)、监视 (monitor),以及由这些角色分别提供的 ACE 注册、信誉查询、不良行为制裁、能力广告、协同伙伴推荐、契约公证、违约仲裁、违约监视和分析等促进服务。正是设置了社交促进型角色,使得承担这些角色的 ACEs 能够通过提供促进服务去辅助平台的服务协同的建立和优化,并驱动和迫使 ACEs(包括代理平台所有者所提供资源服务的 ACE 和代理第三方开发者开发应用的 ACE) 理性遵守服务契约,以确保应用域 E-机构定义的宏观调控作用可以有效地施加到 ACEs。

2.2.3 NGPD-PaaS 平台基础设施层

基础设施层由支持服务协同的调度系统和虚拟化模块组成(图 3)。

1) 支持服务协同的调度系统

支持服务协同的调度系统提供了多种模式的系统资源分

配和调度,能够根据用户应用的特点、计算资源和当前系统的可用资源灵活地分配和调度任务。云计算环境中,用户的使用方式和应用模式复杂多样,传统的调度策略已不能适应快速增长的多种类型的应用需求。为了提高资源管理的有效性和高效性,基础设施层采用多目标服务协同的并行调度策略,调度系统设定由诸多可配置因素组成的综合优先级规则,并基于不同因素的综合权重动态计算出任务的优先级。

2) 虚拟化模块

PaaS 平台如果通过传统的软件部署方法进行开发部署,至少需要以下四个步骤^[7]:

- a) 购买和配置硬件;
- b) 安装和配置操作系统;
- c) 安装和配置中间件和数据库;
- d) 安装和配置应用程序。

为了简化部署过程,并且使平台具有跨平台性,独立于特定的系统环境,NGPD-PaaS 平台采用了虚拟化技术,即将资源虚拟化,通过将操作系统、数据库、支持工具集等基础设施封装成虚拟化模块部署到 IaaS 层的虚拟机上,如图 5 所示。



图5 虚拟化模块的封装和部署

3 实例研究

通过建立在线安排小型会议的应用实例验证 NGPD-PaaS 平台的可行性。在线安排小型会议的应用包括旅店预约、酒宴预订、旅游安排、纪念品预订和回程机(车)票购买服务。下面将从应用开发者和平台的交互以及服务消费者和平台的交互说明应用的建立和使用。

3.1 应用开发者和平台的交互

应用开发者通过 Web 界面制定安排小型会议的本地业务过程并定义涉及应用服务的五个应用域 E-机构(旅店预约、酒宴预订、旅行安排、纪念品预订、旅行票务)的服务供、需标准,然后将上述文件(以 XML 文件的形式)配置到平台的理性 ACE 中,使之成为以完成小型会议安排为目标的理性 ACE-MMA。依据开发者制定的本地业务过程, MMA 发现相应并发的下层业务活动需调用其他 ACEs 提供的服务来完成时,就通过平台的中介 ACE 提供的伙伴推荐服务来获取平台上潜在的协同伙伴。在经过协商、应用 QoS 库验证服务质量等过程选定服务提供者后, MMA 通过 Web 界面返回服务提供者的相关服务信息(如服务适用情境、服务分类体系、服务契约等)。开发者在获得这些信息后,通过平台提供的支持服务和 Web API 进行应用开发(包括对 MMA 的开发完善),并在开发完成后将应用配置加载到平台上,然后通过中介 ACE 发布能力广告。

3.2 服务消费者和平台的交互

服务消费者通过平台中应用发布的能力广告获得在线安排小型会议的应用,通过该应用提供的 Web 界面请求安排小型会议后, MMA 根据消费者提供的服务需求从与其签订过服务契约的服务提供者中选择合适的服务提供者来完成相关服

务操作的执行。在服务操作执行的过程中,平台的监控 ACE 会对执行过程进行监控,如果出现违约情况则会向仲裁 ACE 申请仲裁。与此同时,服务提供方如果设置了针对违约的补救性履行规范, MMA 会依据服务提供者的信誉、当前资源情况等因素作出抉择,并将服务执行结果以及仲裁报告返回给服务消费者。最后,消费者根据会议安排的满意程度、服务需求匹配度等向平台提供反馈,平台的 SLA 管理 ACE 或者平台的所有者会根据反馈调整该应用的相关评级。

4 结束语

本文从现有云计算技术出发,针对目前 PaaS 平台存在的问题,提出了基于 NGPD 的 PaaS 平台,并进行了设计和实现。NGPD-PaaS 平台的优点主要集中在以下三个方面:a) NGPD-PaaS 平台采用了理性 ACE 作为服务代理,解决了服务协同的非自治问题,同时通过 NGPD 模型解决了自治计算面临的“自治”和“可信”问题,增强了平台的服务协同性;b) NGPD-PaaS 平台向云用户提供了统一的图形化界面,并且允许用户进行线下开发和同步上传应用,提高了平台的开发效率;c) NGPD-PaaS 平台灵活快捷地支持自主管理和远程动态服务开发,具有良好的跨平台性,有助于向云用户提供 PaaS 型服务。

参考文献:

[1] 陈国良. 并行算法的设计与分析[M]. 北京:高等教育出版社, 2002.

[2] NURMI D, WOLSKI R, GRZEGORCZYK C, et al. The eucalyptus open source cloud-computing system[C]//Proc of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid. Washington DC:IEEE Computer Society, 2008:124-131.

[3] BUYYA R, CHEE S Y, VENUGOPAL S. Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities[C]//Proc of the 10th IEEE International Conference on High Performance Computing and Communications. 2008:5-13.

[4] BUYYA R, CHEE S Y, VENUGOPAL S, et al. Cloud computing and emerging IT platforms: vision, hype, reality for delivering computing as the 5th utility[J]. Future Generation Computer Systems, 2009, 25(6):599-616.

[5] 史佩昌,王怀民,蒋杰,等. 面向云计算的网络化平台研究与实现[J]. 计算机工程与科学, 2009, 31(A1):249-252.

[6] 李伟,徐志伟,唐志敏,等. 国家高性能计算环境的设计与实现[C]//“863”计划智能计算机会议论文集. 北京:清华大学出版社, 2001:46-56.

[7] CHENG Tong-lv, QING Li, ZHOU Lei, et al. PaaS: a revolution for information technology platforms[C]//Proc of International Conference on Educational and Network Technology. 2010:346-349.

[8] ERL T. Service-oriented architecture: concepts, technology, design[M]. Upper Saddle River: Prentice Hall, 2005.

[9] ERL T. Service-oriented architecture: a field guide to integrating XML and Web services[M]. London: Pearson Education, 2004.

[10] 林立. 基于 SOA 的云计算体系研究[J]. 福建电脑, 2009(8): 112-113.

[11] 高济,吕何新,郭航,等. 可信的自治式服务协同模型和应用开发构架[J]. 中国科学 F 辑, 2009, 39(11):1146-1175.