计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200300044

企业云服务体系结构的参考模型与开发方法

蒋慧敏[1,2]　蒋哲远[1]
1 合肥工业大学计算机与信息学院　合肥 230601
2 中国科学技术大学信息科学技术学院　合肥 230027
(jianghm@mail.ustc.edu.cn)

摘　要　服务与云计算范型的融合有助于大规模分布式软件的开发和应用,同时也为面向服务的软件工程带来了新的挑战。云计算的最大挑战是缺少事实上的标准或单一的体系结构方法,以满足企业将关键产品作为 Internet 上的云服务发布的应用需求。首先,针对企业云计算的业务特点,提出了一种企业云服务体系结构(Enterprise Cloud Service Architecture,ECSA)风格的通用和抽象参考模型,分析了该模型中的云服务、服务模式、服务消费者、管理、流程、质量属性、服务构件模型、服务匹配和交互模式匹配 9 个组件及其之间的关系,并讨论了它们中的角色。然后,提出了一个四阶段的 ECSA 迭代改进过程,该过程把云服务视为首要的类建模元素,通过解除云服务模型和来自目标构件配置之间的耦合,可实现相同云服务集的多种不同体系结构。最后,给出了一种基于该模型的期货程序化交易的私有云服务应用实例,用以展示该方法的可行性和有效性。

关键词:企业云服务;云计算;参考模型;软件体系结构;开发方法
中图法分类号　TP311

Reference Model and Development Methodology for Enterprise Cloud Service Architecture

JIANG Hui-min[1,2] and JIANG Zhe-yuan[1]
1 School of Computer and Information,Hefei University of Technology,Hefei 230601,China
2 School of Information Science and Technology,University of Science and Technology of China,Hefei 230027,China

Abstract　Service-oriented software engineering with the fusion of the services and cloud computing paradigms not only offers many advantages for large-scale distributed software development and applications,but also brings new challenges. The biggest challenge in cloud computing is the lack of a de facto standard or single architectural design method,which can meet the requirements of an enterprise cloud approach to help deliver software as a service over the Internet. First,according to the business characteristics of enterprise cloud computing,a generic and abstract model for Enterprise Cloud Service Architecture (ECSA) is proposed. The model consists of nine components,including the cloud services,service mode,service consumers,management,processes,quality attributes,service matching and interactive matching. The model components and their relationships are analysed,and their roles are discussed. Then,a four-phase software architecture improvement process that considers cloud services as the first class modeling elements is also presented. By decoupling the cloud service mode from their implementation on target component configurations,the process supports exploration of multiple architectures utilizing the same set of services. Finally,the application instance of ECSA is introduced,which hopes to provide recommendations and reference for enterprise cloud service system development and application integration.

Keywords　Enterprise cloud service,Cloud computing,Reference model,Software architecture,Development methodology

1　引言

服务与云计算越来越受到产业界和学术界的关注,这得益于它们能在众多领域的大规模分布式软件的快速开发中得到应用,例如企业资源计划、金融计算、电子商务、电子政务、协同研究与开发等[1]。服务计算关注体系结构设计,并通过服务发现和组合进行应用开发。云计算关注通过柔性的、可扩展的资源虚拟化和负载均衡支持用户在 Internet 上有效地发布服务。面向服务软件工程(Service-Oriented Software Engineering,SOSE)集成了上述两种范型的优势[2]。最初,SOSE 是基于服务计算,现已演化成包括云计算。在 SOSE 中,面向服务体系结构(Service-Oriented Architecture,SOA)为应用开发提供体系结构风格、标准协议和接口,而云计算是通过虚拟化和资源池为用户发布必要的服务。在软件工程框

到稿日期:2020-03-09　返修日期:2020-05-30　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61403116)
This work was supported by the National Natural Science Foundation of China(61403116).
通信作者:蒋哲远(jiangzy@hfut.edu.cn)

架中整合服务和云计算将有助于应用开发者和服务提供者满足单一范型各自面临的挑战。尽管 SOSE 的应用前景光明，但在企业应用领域的具体实现中还需要软件工程来克服融合服务和云计算所面临的新挑战，例如领域云服务软件的特征建模和开发方法、领域云服务软件的参考模型、安全和服务质量(Quality of Service，QoS)等[3-5]。

本文在研究现有企业业务需求的基础上，结合云计算技术和面向服务的思想，给出了一种企业云服务体系结构(ECSA)，包括公式、详细说明、理解和评价。ECSA 是一个借鉴面向服务体系结构风格和云计算体系结构风格的混合体系结构风格。ECSA 将面向服务思想、云计算技术和企业计算紧密融合，采用面向服务架构来降低耦合性，提高灵活性和可扩展性，利用云计算技术在提升企业计算规模的同时降低成本，从而使得企业计算在具有丰富应用的基础上，兼具云计算和面向服务的优点。其次，为了有效地支持基于体系结构的企业云服务系统软件开发和集成全过程，还需要探索一组给定服务集的多种部署体系结构，这对确认 ECSA 的支撑能力是非常有用的。为此，本文提出了一个从云服务规范阶段就逐步设计软件体系结构的开发方法。

本文第 2 节给出企业云概念和企业云服务体系结构的形式化定义，并在此基础上从云服务模型、云服务视图、云服务消费者、云管理、云流程、云质量属性、云服务构件模型、云服务匹配和云服务交互模式匹配等 9 个方面对企业云服务体系结构风格进行分析和总结；第 3 节提出一种企业云服务体系结构的系统化开发过程；第 4 节把所提及到的开发过程应用到运行实例中；第 5 节对相关工作进行比较；最后总结全文和展望未来。

2 企业云服务的参考体系结构

企业云服务(Enterprise Cloud Service，ECS)一般以 Web 服务的形式来实现，可以在面向服务的架构下进行云服务的组织和协同管理，同时，在 SOA 架构下开发的云服务可以部署到各种分布式平台上，也可以通过网络访问各种服务。

定义 1 企业云服务是企业应用软件供应商结合 SOA 和云计算技术开发的满足企业领域特定业务需求和非功能需求的服务，是一种可以托管在云基础设施上的服务，使用者能在 Internet 上通过一组服务接口定义的 API 来访问该服务，可形式化表示为如下四元组：

$$ECS=\langle CID, SI, CP, ESQ\rangle \quad (1)$$

其中，CID 是云基础设施标识；SI 是服务接口，它包括了服务描述语言 WSDL2.0 中的核心元素，即 $SI=\langle$类型，接口，绑定，服务$\rangle$；CP 是一组协同业务流程，每个业务流程定义一个动作编排或协奏；ESQ 表示企业云服务质量属性。

例如，Amazon 的 EC2 云服务可表示为：$ecs=\langle ec2\ wsdl, acp, aws, sq\rangle$，其中 $acp=\langle createVM, startVM, connectVM, stopVM, cancelVM\rangle$。

企业应用软件供应商使用软件即服务(Software as a Service，SaaS)交付应用软件云服务平台，企业用户通过该平台选择 Web 服务或者根据特定需求个性化定制一个企业应用软件系统，这正成为企业应用软件部署的一种主要方式[6]。在 SaaS 架构环境下，组合企业云服务被看作是一个大粒度的软构件集。将 SaaS 设计为一套软构件集的主要原因是从中可以清楚地看到从云特征到设计的联系。UML 的构件模型表示法能提供一个构件服务及其接口的可视化视图。图 1 显示了一个企业云服务的构件化模型。在该模型中云服务构件被表示为一个标有关键字《service》的矩形，并可以在它的右上角包含一个构件图标以代替或者补充关键字。云服务构件向外的交互接口根据其嵌入角色的属性分为供给接口和需求接口，分别用通过一条实线连接到矩形上的小圆圈和小半圆表示。

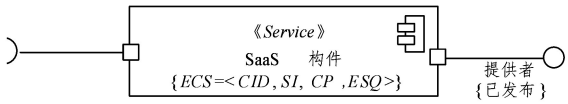

图 1 云服务构件的模型

Fig. 1 Cloud service component model

然而，由于服务产品的无形性、集成性和易逝性，描述一个服务产品要比描述一个传统产品复杂得多。企业云软件设计师在给企业云服务软件应用系统建模时，不仅要抓住特定客户的调用接口操作的句法列表，而且要对隐藏在其后面的交互和集成模式以及系统所部署的基础设施进行建模。

定义 2 ECSA 为企业云服务软件系统提供了一个结构、行为和属性的高级抽象，可以视为论域 U 中一个组合的企业云服务，并可形式化地描述为如下 12 元组：

$$ECSA=\langle S, C, D, M, I, SM, SP, SQ, SD, SC, P, L\rangle \quad (2)$$

其中，S 是公有云(public cloud)、私有云(private cloud)、混合云(hybird cloud)和社区云(Community cloud)等所提供的 Web 服务组件组成的非空集合，即 $S=\{s_i \mid 1\leqslant i\leqslant m, s_i\notin\phi, m\in U\}$，它是企业对外发布的资源对象的一种抽象和封装；$C$ 是 ECSA 中业务逻辑单元(资源对象)组成的非空集合，即 $C=\{c_i \mid 1\leqslant i\leqslant k, c_i\notin\phi, k\in U\}$；$D$ 是 ECSA 中 SOA 数据元素和云元数据组成的非空集合，即 $D=\{d_i \mid 1\leqslant i\leqslant k, d_i\notin\phi, k\in U\}$；$M$ 是 ECSA 中云服务构件模型(包括领域数据模型和工作流)组成的非空集合，即 $M=\{m_i \mid 1\leqslant i\leqslant k, m_i\notin\phi, k\in U\}$；$I$ 是 ECSA 中 SOA 基础设施或云基础设施组成的非空集合，即 $I=\{i_j \mid 1\leqslant j\leqslant k, i_j\notin\varphi, k\in U\}$；$SM$ 是 ECSA 中 SOA 管理和云管理组成的非空集合，即 $SM=\{sm_i \mid 1\leqslant i\leqslant k, sm_i\notin\phi, k\in U\}$；$SP$ 是 ECSA 中 SOA 流程和云流程组成的非空集合，即 $SP=\{sp_i \mid 1\leqslant i\leqslant k, sp_i\notin\phi, k\in U\}$；$SQ$ 是 ECSA 中 SOA 质量属性和云质量属性组成的非空集合，即 $SQ=\{sq_i \mid 1\leqslant i\leqslant k, sq_i\notin\phi, k\in U\}$；$SD$ 是 ECSA 中开发构建元素、服务部署元素和服务交付元素组成的非空集合，即 $SD=\{sd_i \mid 1\leqslant i\leqslant k, sd_i\notin\phi, k\in U\}$；$SC$ 是 ECSA 中云服务的消费者组成的非空集合，即 $SC=\{sc_i \mid 1\leqslant i\leqslant k, sc_i\notin\phi, k\in U\}$；$P$ 是 ECSA 中的协议栈组成的非空有限集合，即 $P=\{p_i \mid 1\leqslant i\leqslant k, p_i\notin\phi, k\in U\}$，其中 UDDI，WSDL，SOAP 是 P 中的核心协议；L 是服务行为约束和交互匹配规则的集合，包括服务业务构件行为的初始条件、前置条件、后置条件和协同时序关系，即 $L=Cons(init, pre\text{-}cond, post\text{-}cond, action)$。

图 2 给出了 ECSA 风格的域本体。

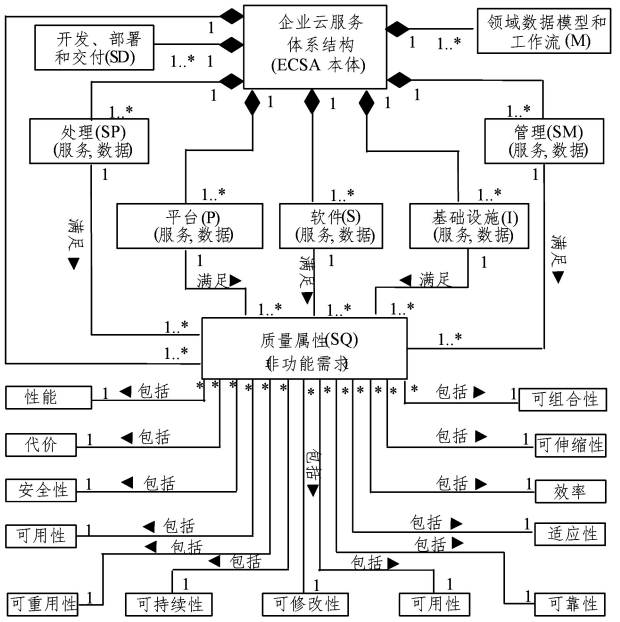

图 2 ECSA 域本体

Fig. 2 ECSA domain ontology

从式(2)每个元素的定义不难看出,ECSA 结合了 SOA 风格和云计算风格。以下小节将详细描述式(2)中的各关键元素。

2.1 云服务模型

云计算是一种能够在 Internet 上实现普适、便捷、按需访问一个共享的可配置计算资源池的模型,它可以使用尽可能少的管理或者与服务提供者的交互就能快速地配置和发布服务。云服务模型包括 3 种服务模式和 4 种部署模型,如图 3 所示。

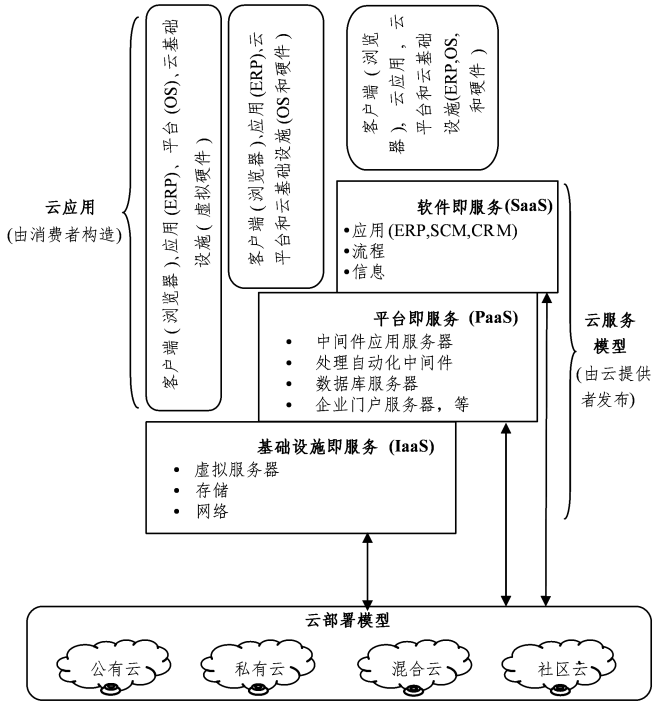

图 3 ECSA 的服务模型和部署模型

Fig. 3 Cloud service model and deployment model of ECSA

从服务类型的角度来看,企业云服务包括 SaaS、平台即服务(Platform as a Service,PaaS)和基础设施即服务(Infrastructure as a Service,IaaS) 3 类服务,可形式化表达为 $T=T^{\mathrm{I}}\cup T^{\mathrm{II}}\cup T^{\mathrm{III}}$,其中,$T^{\mathrm{I}}=\{t\mid t$ 是 SaaS 类型云服务$\}$,$T^{\mathrm{II}}=\{t\mid t$ 是 PaaS 类型云服务$\}$,$T^{\mathrm{III}}=\{t\mid t$ 是 IaaS 类型云服务$\}$。

(1)SaaS 是通过 Internet 为最终用户提供所需要的云服务。终端用户可通过浏览器使用云服务,从而减少在用户系统中安装和运行软件的麻烦。用户只需要为自己所用的服务付费,降低了开发和维护成本。同时客户可以根据自己的实际需求向服务提供者定制所需的应用软件服务。主要的供应商包括 IBM 和 SAP。

(2)PaaS 使用底层的云基础设施构建计算平台。PaaS 主要包含应用开发、软件工具、数据库工具、存储和测试等,并通过远程托管的方式交付给用户。用户可以根据需要安装和部署软件工具和开发环境,不必关注底层的网络、存储、操作系统等技术问题。该层服务是一个开发、托管服务的平台。代表性的有 Google App Engine 和 Microsoft Azure。

(3)IaaS 在服务层次上是底层服务,可以提供物理硬件资源和计算能力。IaaS 通过虚拟化技术,为企业提供计算、存储、网络及其他资源的服务。企业无需自己购买所需的服务器、数据中心或网络资源。服务提供者如 Oracle 和 Amazon EC2 等。

企业云服务部署方式主要有:公有云、私有云、社区云和混合云等[7]。此外,还有一些云服务是通过企业遗留系统中的 Web 服务部署,可形式化表达为 $D=D^{\mathrm{I}}\cup D^{\mathrm{II}}\cup D^{\mathrm{III}}\cup D^{\mathrm{IV}}\cup D^{\mathrm{V}}$,其中,$D^{\mathrm{I}}=\{d\mid d$ 是私有云服务$\}$,$D^{\mathrm{II}}=\{d\mid d$ 是公有云服务$\}$,$D^{\mathrm{III}}=\{d\mid d$ 是混合云服务$\}$,$D^{\mathrm{IV}}=\{d\mid d$ 是社区云服务$\}$,$D^{\mathrm{V}}=\{d\mid d$ 是传统 Web 服务$\}$。

(1)公有云通常由一个组织来管理和维护,提供公共云服务。公有云允许用户只使用简单的浏览器通过提供商提供的接口即可访问到云服务。用户无需知道云服务的内部结构和实现技术等,只需为他们所使用的服务付费。因此,云服务可以像传统的水、电、气等社会服务那样给用户提供按需计算服务,可以减小中小企业开发应用所需要的平台和环境等底层的成本,使其专注于开发。

(2)私有云由一个组织自己构建或租赁。该组织自己负责维护和管理等,并向组织内部提供数据等云服务。企业可以新建或改造原有基础设施,拥有自己的内部数据中心,使操作更易于管理、维护和升级。与公有云相比,私有云安全性高。缺点是基础设施、人力和维护等成本较大,同时服务空闲时可能造成资源浪费。

(3)社区云是大的公有云范畴内的一个组成部分。它是指一些由有着共同利益(如任务、安全需求、策略目标考虑等)并打算共享计算资源、网络资源、软件和服务能力的企业联盟组织共同创立的云。即基于企业联盟内的网络互连优势和技术易于整合等特点,通过对企业联盟内各种计算能力进行统一服务形式的整合,结合企业联盟内的用户需求共性,实现面向企业联盟用户需求的云计算服务模式。社区云可以由该用户/机构或第三方管理,存在预置(on premise)和外置(off premise)两个状态。

(4)混合云是私有云和公有云的混合。即同样的计算资

源既可以向企业内部提供业务服务，也可以向外界提供云服务。在这种云模型中，私有云可以通过互联网向外界提供访问，同时在计算资源紧缺时可以临时购买公有云计算资源。

企业云服务应用作为一种新型的面向服务的应用，它是通过重用和组合所有三层云栈，即 SaaS，PaaS 和 IaaS 中的服务来开发实现。企业云服务应用允许企业在一个多租户(multi-tenant)环境中通过 Internet 更灵活、广泛地购买和共享所有必要的云计算资源。ECSA 和传统的企业面向服务体系结构相比，加入了一个新元素 *SD*。*SD* 实际上是一个云服务和企业应用开发的标准部件和工具集合，因此，它是在{PaaS}上开发服务的基础。

2.2 云服务视图

企业云服务在上文已被形式化和非形式化地描述为一个业务功能和技术，或管理基础设施的自包含的软件抽象，并定义为一个良好的 Web service 接口。定义企业云服务作为功能服务来负责完成某些操作业务，如购物交易 Web 服务和旅馆订阅 Web 服务。同时它们还包括一些组合和流程服务，如工作流服务。假如一个功能服务并不暴露于 Internet 上（离开企业防火墙），或者不能被 Internet 访问，那么$s \in I$。本文关注云环境下的被管理服务（或企业服务），并定义了企业云服务是作为的一个特定受管理服务，它具有服务等级协议(Service Level Agreement，SLA)[8]、伸缩性和灵活性(Elasticity/Dynamism)、可追究性和效用性(accountability/utility)、松耦合(loose-coupled)等特性，且能通过 Internet 访问和发布。

企业云服务从纵向和横向两个方面将 SOA 服务的概念和能力扩展到更广泛的领域，如图 4 所示。

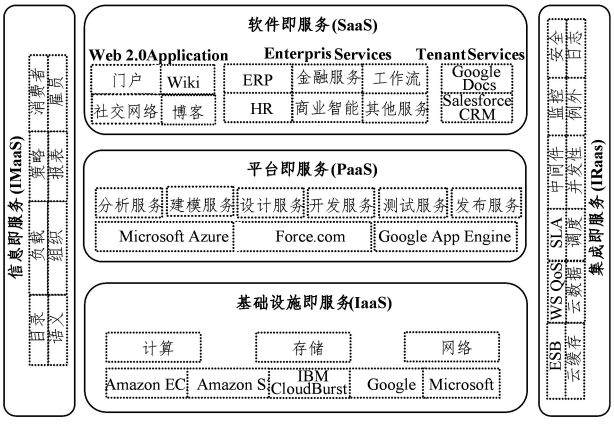

图 4 云服务视图

Fig. 4 Cloud service view

企业云服务还定义了如下几个方面的属性：

(1)云服务接口类型。包括用户交互接口、Web 服务接口、REST 接口、Web 应用接口、事件接口。

(2)云服务访问类型。是一个客户访问协议方法，如 Web 用户交互协议(HTTP)、Web 服务 API (SOAP)、REST API (HTTP)、Web 应用 API、事件触发器、分布式设备(无线设备)。

(3)云服务供给类型。包括应用、业务操作、资源、信息、平台、集成。

(4)云服务控制/拥有者类型。包括买/租赁和拥有、公有云提供者拥有和现收现付。

2.3 云服务消费者

ECSA 体系结构风格中企业用户通过应用软件云服务平台选择 Web 服务或者根据特定需求个性化定制一个企业应用软件系统。企业可根据实际情况选用不同的 ECSA 体系结构风格。

(1)没有自己的数据中心的企业。它可以是提供公有云的企业的一个消费者。大部分中小企业都属于或将成为这种企业。

(2)企业本身是一个公有云提供者。它提供公有云服务，如 Amazon 云、Google 云、Saleforce 云、IBM 云和 Microsoft Azure 云。

(3)企业具有私有云服务的数据中心。它的消费者是通过 Internal 访问的云应用消费者，如注册用户、员工和合作伙伴。私有云可能是其他公有云服务(别人的数据中心)的消费者。

(4)企业有许多数据中心和混合云。它的公有云服务的消费者有可能是企业内部的私有云，以及外部客户访问的内部和外部云应用，包括外部终端用户和其他企业的云应用。它的私有云服务的消费者可能是内部的应用，被内部客户端访问，包括内部终端用户和内部分配的公有云服务。大部分大型企业都将成为这类企业。

云服务消费者也依赖于云服务的类型。假如 ECS 是在{PaaS}上，例如 Google App Engine，那么 Web 软件开发者、IT 管理者和应用系统管理员是 ECS 消费者。假如 ECS 是在{IaaS}上，那么系统和数据库管理员是它的消费者。

特别地，公有或私有云服务的消费者有如下特征：

(1)自助服务。用户能访问它们提供的服务，或直接在云中获取服务，用户也可从自助服务门户管理和监控云服务。

(2)访问云服务的标准 API。

(3)快速的服务供应。

(4)使用付费。

2.4 云服务管理

云服务管理是由云提供者从为云消费者提供高质量服务的视角所推出的一系列流程、活动和方法。云服务管理为云消费者管理和操作云服务提供了必要的功能，从而有助于云服务提供者实现其业务目标。

云计算从两个方面把 SOA 管理提高到一个新的水平，分别是提升 SOA 管理和增加新的特定的云管理。

(1)云计算要求对服务和弹性基础设施实行更高要求的实时动态自动化管理。

(2)云计算要求在运行时支持 SLA 感知的自动化策略管理。

(3)云计算要求其使用的模型能自动进行服务供给和订阅管理。有两种新的云服务，即云供给服务和云订阅服务，用来管理云供给和订阅。

(4)云计算要求为云服务面临的安全和可信挑战提供新的身份管理，如 Amazon 云安全方法。

随着企业云服务的发展，灵活有效地管理云服务成为亟需解决的问题。通常服务管理包括云服务管理和传统 IT 服

务管理,可形式化表达为 $SM=SM^{\mathrm{I}}\cup SM^{\mathrm{II}}$,其中 $SM^{\mathrm{I}}=\{m|m$ 是传统 IT 服务管理$\}$,$SM^{\mathrm{II}}=\{m|m$ 是云服务管理$\}$。传统 IT 服务管理主要包括 SOA 服务管理和 Web 服务管理。云服务管理 SM^{II} 主要涉及到物理资源管理、虚拟资源管理,SLA 管理和 QoS 管理等。

作为企业云服务的核心部分,企业云服务管理对不同类型的云服务设计了不同的管理项目,并据此给云服务提供者参考和量化。不同类型的云服务提供者在云服务管理方面有共通之处,企业云服务管理项目如表 1 所列。

表 1 企业云服务管理项目

Table 1 Enterprise cloud service management items

管理项目	描述
服务生命周期	对云服务的开发、发布和部署等进行管理
业务支持管理	包括合同、账单和客户账户管理等,其中包括向消费者提供的服务的定价
数据管理	云服务使用过程中存储的静态数据、动态数据,以及一些必需的临时数据
安全管理	对于云用户来说,云服务安全包括认证、授权和加密等
风险管理	识别、评估和管理云服务使用过程中的服务需求和客户需求风险
监控管理	管理云服务的使用以及运行情况
QoS 管理	对云服务的性能、可用性及可靠性等进行管理
虚拟资源管理	管理云服务实例中使用的虚拟资源,如虚拟机等
物理资源管理	管理和维护云服务使用的物理资源,如网络、CPU 等

不同的服务类型有不同的侧重点,因此它们的管理项目有所不同。

(1)关于常见的服务类型,SaaS 主要向用户提供各种应用服务,如 Email、办公软件和各种应用管理软件等。通过比较和总结,给出一个对应于此类服务的管理项目。SaaS 的管理项目主要包括需求管理、访问方式、接口管理、个性化、数据分析和集成功能。

(2)设计针对 PaaS 的管理项目。PaaS 主要向用户提供开发、发布和部署应用与软件所需的各种工具、环境以及硬件支持等。PaaS 的管理项目是用户需要考虑的管理项目,包括开发工具、部署环境、服务器、数据管理和网络环境。

(3)IaaS 的管理主要是关于底层的硬件,如存储器和 CPU、网络和管理软件等,这类服务是云服务的基础。IaaS 需要关注的管理项目主要包括多租户、虚拟机、物理资源、数据备份、网络带宽和监控。

2.5 云流程

通常企业服务处理过程涉及两类服务:云服务和传统 Web 服务,可形式化表达为 $SP=SP^{\mathrm{I}}\cup SP^{\mathrm{II}}$,其中 $SP^{\mathrm{I}}=\{p|p$ 是传统服务处理过程$\}$,$SP^{\mathrm{II}}=\{p|p$ 是云服务处理过程$\}$。

$SP^{\mathrm{II}}=SP_S^{\mathrm{II}}\cup SP_P^{\mathrm{II}}\cup SP_{\mathrm{I}}^{\mathrm{II}}$,$SP_S^{\mathrm{II}}=\{p|p$ 是 SaaS 云服务处理过程$\}$,$SP_P{}^{\mathrm{II}}=\{p|p$ 是 PaaS 云服务处理过程$\}$,$SP_{\mathrm{I}}^{\mathrm{II}}=\{p|p$ 是 IaaS 云服务处理过程$\}$。本文主要关注云服务的处理过程,涉及服务请求者、云服务中心和服务提供者,具体如图 5 所示,主要包括需求分析器、服务搜索、服务信息中心、服务提供过程等模块。服务请求者发布需求信息,需求分析器根据需求信息在服务注册中心查找服务,经过服务搜索、服务评估和服务分配调用等过程返回合适的云服务,而服务提供者经过需求分析、服务实现、部署和测试等过程提供用户需要的云服务。

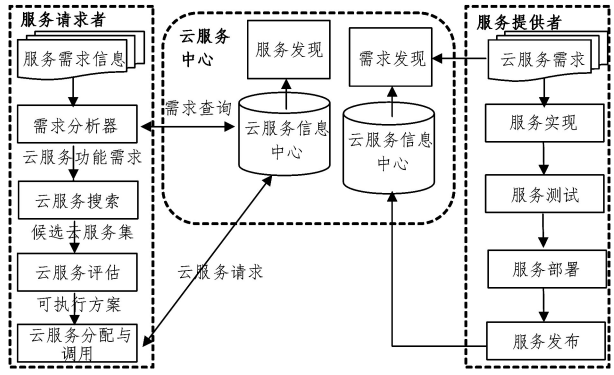

图 5 云服务处理过程

Fig. 5 Steps to cloud service process

在 SOA 中,确定服务处理模块的方法是将现有应用系统中所能提供的功能以列表的形式列出,按照功能域进行划分,然后封装成服务。虽然该方法可以作为开发新系统的依据,但由于其过多地考虑了软件系统的功能要求,而不是从业务需求出发,因此不能满足定制和个性化需求。企业云服务通常需要从业务需求角度管理流程,因此云服务过程可以使用业务流程管理(Business Process Management,BPM)来描述。每个流程是通过编制(orchestration)和编排(choreography)从而由多个服务组合以便完成整个或部分业务流程或任务[13]。就云计算来说,云使得 SOA 流程不仅能够提供新的 BPM 分发模式,也能改善 SOA 流程的可伸缩性、性能和可利用性。

2.6 云质量属性

软件体系结构质量属性不仅包括系统的体系结构设计原理,也包括任何软件体系结构和行为的非功能性约束。因此体系结构质量属性是 ECSA 一部分,参见图 2。现有文献中已经定义的一些常用的 SOA 质量属性也可作为云计算的质量属性,特别是私有云。它们共享许多质量属性,如性能、安全、可伸缩性、可利用性。然而,云质量有别于 SOA 质量属性,具体如下:

(1)SOA 的质量属性和公有云有不同的成熟度。一般地,云质量属性不如 SOA 质量属性成熟;

(2)一些云质量属性的规约与传统的 SOA 不同,如可弹性伸缩;

(3)云质量属性包括一些特定的云质量属性和云服务特色,如云可见性和订阅。

随着越来越多的云消费者把他们的工作负载调度到云服务提供者的基础设施上,不论云环境如何动态变化,消费者和提供者之间的 SLA 都能确保服务质量保持在令人满意的水平,这一点尤为重要。SLA 包含服务协议解释、服务等级参数、QoS 保证、所有违反协议的安排和补救方法。此外,动态云环境下还必须考虑根据环境变化自动的折中方案和动态 SLA 组织。文献[8]指出了云计算和其他 Web 服务的 SLA 之间的主要差异。

2.7 云服务构件模型

企业云体系结构的层次性以及构件和服务的组合要求其能被设计为具有可扩展性并能重新配置,以便提供服务并在

企业与客户之间达成服务协议(如服务等级协议)。考虑到成熟的基于构件的软件工程(Component-Based Software Engineering,CBSE)具有适当的安全控制方法和可靠的重用技术,结合 CBSE 和 SOSE 两个范型的优势满足非功能性需求,设计云应用作为 Web 服务构件是必要的[7,9-10]。

以期货程序化交易数据的实时采集与处理应用场景为例,图 6 给出了一种行情数据云服务提供者的部署体系结构。

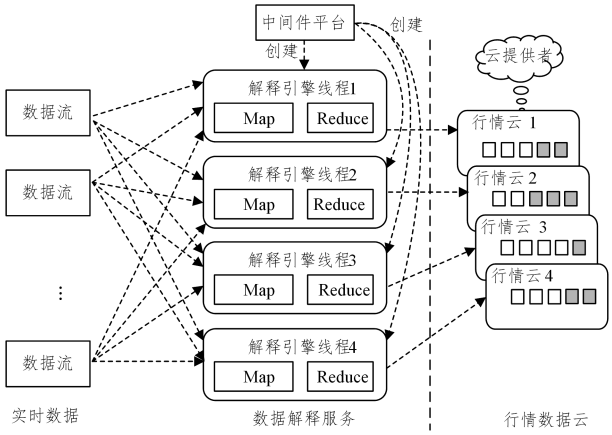

图 6 行情数据云服务提供者的部署体系结构

Fig. 6 Deployment architecture of market data cloud service provider

不同学者从不同的角度和侧面给出了不同的构件模型。以下是构件模型关键元素的一个概要:

(1)支持云设计特征和相关的体系结构层。

(2)显性的导出和导入接口。特别地,显性和形式化导入接口使构件更加脱离语境,只对必需的服务和构件属性进行规定。

(3)服务的语义描述。除了句法信息,如服务基调(signature),服务行为的抽象规约对重用软构件是必要的。

(4)交互模式。交互模式描述了构件用户必须遵从的服务激活协议,以便其能正确地使用该构件。

图 7 说明了本文构件模型的行情数据构件[9]。它由一个行情云服务请求者和云服务提供者组成。接口允许用户打开和保存行情数据,同时需要一个来自行情数据服务器构件的服务去创建、检索和更新行情数据。该服务器实时采集各交易的实时序列数据,经过 MapReduce 大规模数据处理后得到各交易合约和分析周期的实时行情数据并提供云服务[14]。一个空行情能使用 crtMarketData 创建。请求服务 selMarketData 在检索一个合约行情数据时并不改变服务器构件的状态。同样地,更新服务 updMarketData 在更新某个合约行情数据高速本地存储时并不返回任何值。某个合约行情数据也能被删除。对一个合约行情更新服务,其服务用户需要符合特定格式的需求规约。如果行情数据格式文档使用 XML 文档,那么这些文档必须是良定的(正确的标记嵌套)或有效的(良定的且符合一个 DTD 文档类型定义)。为客户 MarketDataInterface 指定一个导入交互模式,同时为提供者 MarketDataServer 指定一个导出模式。导入模式意味着期望首先执行 create 服务,接着是重复调用 retrieve 或 update。

```
Component MarketDataInterface
interfaceLogin { }
interfaceFinish { }
…
interface CloudRequester {
import services
    create(id:ID);
    select(id:ID):MarketData;
    update(id:ID,upd:MarketData);
    preCond valid(upd);
    postCond retrieve(id)=upd;
export services
    openMarketData (id:ID);
    saveMarketData (id:ID,data:MarketData);
import interaction pattern
    create;!(select+update);}
```

```
Component MarketDataServer
interface ConnectionThread { }
interfaceMap { }
interface Reduce { }
…
interface CloudProvider {
import services
…
export services
    crtMarketData (id:ID);
    selMarketData (id:ID):MarketData;
    updMarketData (id:ID,upd:MarketData);
    preCond wellFormed(upd);
    postCond selMarketData (id)=upd∧wellFormed(upd);delDoc(id:ID);
export interaction pattern
    crtMarketData;!(selMarketData+updMarketData);delMarketData;}
```

图 7 行情数据处理构件

Fig. 7 Market data processing components

2.8 云服务和交互模式匹配

交互模式匹配和基于契约描述的服务匹配一直是构件匹配的基础[11]。描述逻辑系统中概念包含关系是推理的基础,可应用于云服务的构件匹配推理。在功能上,一个云服务可通过输入(inSign)、输出(outSign)、前置条件(preCond)和后置条件(postCond)进行描述。云服务的匹配可通过有关前置和后置条件的蕴涵术语和基于广泛接受契约设计方法中的基调匹配来定义。

定义 3 如果 $\frac{\forall\, inSign.in_R \cap \forall R.\ \forall\, outSign.out_R}{inSign.in_P \cap \forall P.\ \forall\, outSign.out_P} < inP \equiv inR \wedge outP \equiv outR$(基调是兼容的:对应参数的类型是一样的)和 $\frac{\forall\, preCond.pre_R \cap \forall R.\ \forall\, postCond.post_R}{\forall\, preCond.pre_P \cap \forall P.\ \forall\, postCond.post_P} < preR \subseteq preP \wedge postP \subseteq postR$(被请求的服务前置条件被弱化,而后置条件被加强),那么称一个提供服务 P 细化为一个请求服务 R,或者服务 P 匹配 R。

例 1 行情数据服务器的服务 *updMarketData* 匹配 *update* 的需求。在接口提供的方法上,*update* 可作一个服务,且基调是一致的。*updMarketData* 要求不高,较少地约束前置条件($valid(MarketData)$蕴含 $wellFormed(MarketData)$)和一个较强的后置条件($select(id)=MarketData \wedge wellFormed(MarketData)$蕴含 $select(id)=MarketData$)。这意味着所提供的服务能满足需求。

交互模式是通过组合转换角色的模式转换图，即代表所有可能的模式执行图来解释。客户构件和提供者构件基于它们的导入和导出接口中的服务描述参与交互过程。客户将展示一个确定的导入交互模式，即执行提供者服务的一个明确的请求次序。另一方面，提供者将对所提供的服务执行次序施加某种约束。交互模式的规约描述了构件进程的可见活动次序。进程演算提出模拟和互模拟作为处理与交互模式等价的构造。下面使用一个进程的模拟概念来定义请求者和提供者之间的交互模式匹配。

定义 4 假如存在一个从 R 转换图到 P 转换图的同态，即对每一个 $R_g \xrightarrow{T_i} R_h$，有一个 $P_k \xrightarrow{S_j} P_l$，并满足 $R_g = \mu(P_k)$，$R_h = \mu(P_l)$，以及 S_j 细化 T，则称一个提供者的交互模式 $P(S_1,\cdots,S_k)$ 模拟一个请求者的交互模式 $R(T_1,\cdots,T_l)$，或模式 P 匹配 R。

交互模式的匹配是模拟，模拟包含服务匹配。提供者需要能够模拟请求，即需要满足请求者期望的交互模式。该定义蕴涵 S_i 和 T_j 的关联是不固定的，即对任何 S_i，S_i 细化一个被请求的服务 T_j 都是适当的。对一个给定的 T_j，在其进程执行期间原则上有几个不同的提供者服务 S_i 能提供实际的服务执行。

例 2 提供者要求 *crtMarketData*；!(*selMarketData* + *updMarketData*)；*delMarketData*，而请求者期望 *create*；!(*select*+*update*)作为命令。假定服务序列对：*crtMarketData*/*create*，*selMarketData*/*select* 和 *updMarketData*/*update*，基于它们的描述匹配，不难看到提供者匹配(即模拟)请求者服务器交互模式，不过 *delMarketData* 服务没有被请求。

3 企业云服务软件开发方法

企业云服务的开发范型是基于服务计算，集成了云计算而演化成的新范型。当企业云服务应用被映射为这两种技术或其任意一种技术的统一服务抽象时，它是作为一种需要被理解和采纳的、重要的新开发模型而出现。图 8 从体系结构设计和配置角度给出了一个企业云服务软件的系统化开发过程。

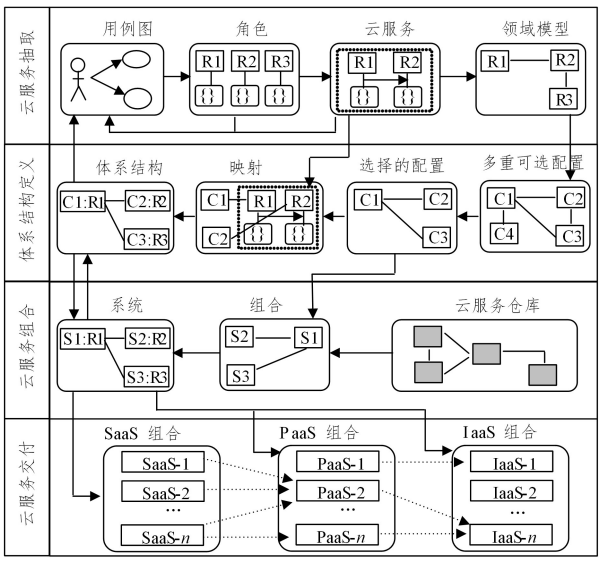

图 8 企业云服务体系结构的开发过程

Fig. 8 Enterprise cloud service architecture development process

该开发过程主要包括 4 个阶段：1)研究现有云市场的云平台策略(包括云服务消费者和管理)和业务场景，抽取主要的云服务构件，定义感兴趣的云服务构件之间关系(互操作性是本阶段的一个重要设计元素)，称之为构建云服务库；2)映射云服务系统到构件配置，可以得到系统的体系结构描述；3)云服务系统构件配置的组装和集成；4)云服务软件交付和部署。

第一个阶段是识别相关用例之间的关系，即：构建云服务系统用例图，描述系统的静态功能；建立活动图，显示系统的活动流程和并发行为。这些用例图和活动图中的用例、角色和它们的交互将被作为定义服务元素的来源之一。接下来是生成包含角色的领域模型(domain model)。

第二阶段中，角色领域模型将被细化为多个可选择构件配置。选择一个局部优化配置，在该配置基础上，系统包含的云服务将被映射成为一个体系结构配置和描述，其中定义了云服务构件的规约、构件之间的交互(即连接器类型)、构件与连接器的拓扑结构和配置组成。这些体系结构描述和配置易于实现，且经过考查和评估可作为目标系统的体系结构。

第三阶段是根据目标系统的体系结构配置和描述来组装和集成现有的构件，形成云服务的企业应用系统。

第四阶段是根据目标系统的体系结构配置描述，使用多租户模式部署云服务软件系统。上述开发过程实际上是一个四阶段的反复迭代优化过程，且跨越从角色和服务的抽取，到构件选取、包装、集成和部署并反馈到用例图的定义；体系结构能被细化和重构，从而产生新的体系结构配置，进而可形成用例的一个全新视角。

该开发过程的生命周期中涉及到如下 3 个主要角色：

(1)市场：在云服务知识库中存储和管理云服务规范。它还可以实现云服务规格的发布和查询。一些市场也支持完成供应商和消费者之间的云服务采购和契约流程。

(2)云服务提供商：指定他们的云服务使用一种统一的云服务规范语言。然后，他们可以使用一种云服务的操作技术将云服务规范发布到市场。

(3)ECSA 架构师：由于企业云服务软件系统在广义上可被视为一个组合的 SaaS，一个 ECSA 架构师可以使用云服务规范语言指定其云服务系统为云服务规范。然后，通过云服务的操作技术来查询市场，以便检索 ECSA 要求的云服务规范和组合云服务规范。组合云服务规范导致了几个替代 ECSA 配置。一个 ECSA 配置是由配置一个 ECSA 部署环境所需的所有云服务规范组成的。然后，基于 ECSA 预定义的标准，如许可或成本，选择一个最佳 ECSA 配置。最后，选定的 ECSA 配置作为 ECSA 配置部署环境的一个部署清单。生命周期中 ECSA 的设计和配置强调需要横跨三层云堆栈的统一的云服务规范语言，以便各供应商所创建的云服务规范可被无缝地组装成一个完整的 ECSA 配置。由于 ECSA 可被视为一个组合 SaaS，云服务规范语言也应该支持 ECSA 规范。此外，既然云服务和 ECSA 可以用一个统一的方式指定，针对配置一个 ECSA，生命周期也意味着需要一套云服务操作技术来支持发布、查询和构成云服务规范。

4 企业云服务软件应用实例与讨论

在证券期货领域，期货企业使用计算机完成自动化交易

即程序化交易的过程中，存在服务资源异构、服务灵活度和可靠性低等现象，期货服务商迫切需要引入按需供给、操作灵活、快速部署的云服务作为服务支撑[12]。因此，结合某期货公司的期货程序化交易云服务平台[9]研发项目，应用上述参考模型与方法，本文设计了一种多租户的期货程序化交易私有云体系结构，以验证企业云服务参考模型和开发方法的可行性。该体系结构主要以 SaaS 和 PaaS 模式为期货交易企业提供可扩展的云服务，实现对期货程序化交易的实时监控和统一管理。

图 9 给出了期货程序化交易私有云体系结构。

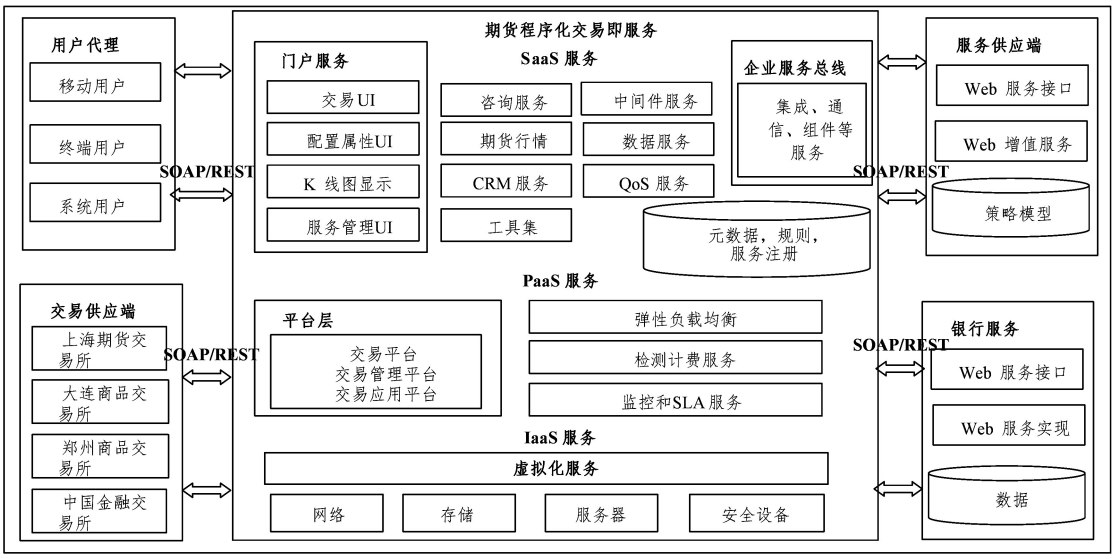

图 9 期货程序化交易私有云体系结构

Fig. 9 Private cloud architecture of program trading

该体系结构支持行情数据采集和历史数据生成服务中间件、行情数据广播服务中间件、行情数据订阅/发布服务中间件、业务交易中间件等多构件协作，支持管理和调度资源，支持 SOA，支持互操作性和资源共享，等等。

作为私有云的最底层，基础设施层主要为上层提供硬件基础。通过虚拟化物理资源（服务器、存储、网络资源、安全设备）屏蔽底层设备，将物理资源封装成服务，并提供服务访问接口。在虚拟化的物理资源上搭建期货交易平台，透明可视化地管理、监控数据资源，实现基础设施资源动态按需分配。中间件和应用平台安装在云操作系统上，主要是为了方便用户实时操作，实现在线交易，同时降低期货企业开发应用的复杂程度，并以中间件的形式将通信和业务服务提供给上层的期货交易应用层使用，支持多环境、多语言环境下企业应用的开发和部署。期货交易应用层可部署企业级应用软件，如门户服务、客户关系管理（CRM）、咨询服务等。同时云端通过服务接口与代理等方式，以 SOAP 和 REST 协议与用户、交易中心、服务供应端和银行等各方进行通信和交互。

基于云/客户（Cloud/Client）模式所开发实例的一种云服务适应性部署场景如图 10 所示。客户能通过该平台与系统交互，集中了实时行情，K 线技术分析，交易模型设计、编写、评测和优化，多账户管理，自动下单并辅以手动的闪电下单，套利交易等功能。

云基础设施服务器通常要求能同时处理多个用户应用程序，但工作负载可能会暂时显著减少。负载平衡将允许调整系统架构并将应用程序重新定位到一台服务器，从而缩小服务器的部署规模。在云服务系统被理解为应用程序和平台组件的集成时，部署系统的架构和/或配置会因质量管理活动而改变，例如，通过在多云中进行负载平衡。因此，云服务系统架构要求能对外部因素（如减少的负载）作出反应，并调整配置以降低成本（非功能性需求），同时保持足够的性能（也是非功能性需求）。

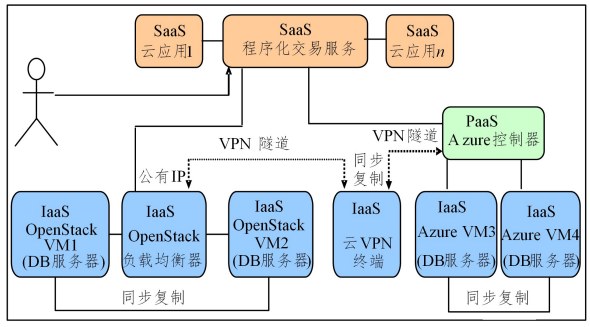

图 10 云服务应用软件部署案例

Fig. 10 Cloud service application deployment use case

以下是云服务系统实施的两个关键：

（1）工作负载和 QoS（如工作负载突发性下降或性能下降）决定云服务系统的适应能力。成本和质量是架构决策的驱动因素，即架构决策是基于非功能性需求做出的。

（2）在多层云服务应用架构中，用户应用程序可能运行在第三方提供的远程基础设施服务器上。因此，影响向下扩展的适应性因素包括：用户的应用程序性能；基础设施提供商的系统工作负载。

另外，云服务软件开发和运营需要将软件需求与云本机平台服务联系起来，例如，在整个软件生命周期中的管理性能和成本。上面介绍的案例系统是一个多云应用软件，涉及在不确定的环境中管理备份资源，提供容错作为一个非功能性需求。除了混合基础设施和程序化交易系统外，还需要考虑层之间下游集成涉及到的其他系统。

5 相关工作——行业最佳实践

目前,SOA 和云计算相融合的工作可分为如下几类:1)解释和分析面向云服务的体系结构风格或框架;2)面向企业的云计算和 SOA;3)创立新方法绑定 SOA 方法和云计算,包括 SOA 在云计算中的最佳实践,以及采用云计算能力改进已有的面向企业的 SOA[4]。

文献[15]提出了一种云计算开放体系结构(Cloud Computing Open Architecture,CCOA)。CCOA 是一个以云计算为中心,并桥接了 SOA 集成能力和云计算环境虚拟技术的面向服务体系结构的框架。同时,提出了构造云计算体系结构的 7 个原则。本文提出了一种由面向企业云服务公式说明的新的混合体系结构风格,其包括基本体系结构元素、面向服务和云计算的设计原理。

许多 SOA 软件销售商,如 IBM,SUN,Oracle 和 Microsoft,都提出了各自的新软件产品体系结构参考模型和框架,并融合了 SOA 和云计算能力。本文提出的模型可用来评价不同软件销售商的体系结构方法。

文献[4]提出了一种企业云服务体系结构,它通过混合体系结构风格来描述 SOA 和云计算的关系。文献[7]描述了 Web 服务构件体系结构模型和 SOA,可以成为我们构造企业云服务参考体系结构和体系结构驱动的应用开发设计的一个集成方面。此外,许多云计算服务提供者,如 Amazon 云、Google 云、IBM 云和 Saleforce 云,它们的体系结构都可看作是混合风格 ECSA 的一个例子。特别地,Amazon Web Services的最佳实践工程师已经使用形式化规约和模型检查,在关键系统的设计时帮助解决困难问题。Amazon 的形式化方法还经常被应用于复杂的实时软件设计中,包括公有云服务[16]。文献[17]分析了云计算和 SOA 的基本原理,并分别以基于 SOA 集成 SaaS 服务和 PaaS 服务方法为例,指导企业怎样一步一步地把云计算和 SOA 融合到自己的企业中。文献[18]提出了一种面向服务的云开发框架,可实现跨多个云计算平台的部署。部署在不同云中相互依赖的构件之间的互操作性是通过自动生成服务和服务客户实现的。文献[19]提出了一种面向电子政务企业的体系结构框架,以支持云计算环境下大数据和开放链接数据的新需求。

本文所定义的 ECSA 同上述企业云服务体系结构建模方法相比较,不同之处体现在以下几个方面:

(1)ECSA 作为一个形式化混合参考体系结构,同时具有 SOA 风格和云计算风格,且是建立在一个 SOA 和云计算上的企业体系结构家族的抽象。ECSA 已被应用于复杂的期货程序化交易私有云软件设计。

(2)ECSA 模型和风格规约强调质量约束,关注约束的 Web 服务构件方面以及隐藏在它们后面的交互和集成模式,不仅能帮助分析和评价 ECSA 系统,也能为 ECSA 体系结构的设计提供指导。

(3)ECSA 是一个四阶段迭代改进过程,该过程把云服务视为首要的类建模元素,通过解除云服务模型和来自目标构件配置之间的耦合,可实现相同云服务集的多种不同体系结构。

结束语 云计算是以虚拟化计算资源、令人难以置信的能力和可扩展性、动态配置、多租户、自助服务和按使用付费的定价模型为主要特征。对基于 SOA 构建信息系统的企业来说,云计算可能承载用户的业务应用和价值信息,通过扩展已有的企业 SOA 体系结构并引入云计算有助于缓解新业务带来的压力并迎接新的机遇。另一方面,对云服务企业来说,采用 SOA 原理和管理方法能提升云服务质量,并可为其他企业构造云计算。因此,如何在现有的 SOA 和云计算研究工作基础上提出新的 SOA 和云计算混合的体系结构风格已引起学术界和产业界的关注。

本文给出了一种 ECSA 体系结构风格的形式化定义,它融合了 SOA 和云计算两种风格。ECSA 体系结构风格规定了 ECSA 体系结构元素的基本词汇,封装了 ECSA 体系结构元素的重要设计决策,以及强调了元素之间的重要约束和关系。因此,ECSA 研究有助于研究人员和参与者更好地理解 ECSA 体系结构并做出正确的体系结构设计决策。从交互和集成的观点对云服务概念进行通用的形式化描述具有如下好处:从体系结构的抽象级为云服务提供了一个精确的语义,从而为云服务的组合、细化和验证提供了一种有价值的处理方法,并且也为它们提供了一流的建模元素,而不仅是一流的实现要素。

本文的主要贡献如下:

(1)通过公式定义和描述 ECSA,深入分析了 ECSA 体系结构功能和非功能方面的设计原理、结构和行为;

(2)从体系结构设计和配置角度给出了一个企业云服务软件的系统化开发过程,集中讨论了特定云中研究较少的设计问题;

(3)所提方法和模型可用于分析和评价各种类型的具体企业体系结构,并能指导云使能 SOA 系统设计或 SOA 风格云计算系统的开发。

未来的工作将包括:

(1)ECSA 体系结构支持的领域建模、描述和应用实现,特别是支持容器和微服务(microservices)的云服务软件平台虚拟化和持续开发,将会形成企业云软件系统开发的新趋势;

(2)ECSA 体系结构的动态 QoS 感知质量属性权衡分析和优化方法;

(3)提供 ECSA 体系结构的领域服务、构件配置和体系结构映射的细化,并重构相关的自动化支持工具。

参 考 文 献

[1] MORENO-VOZMEDIANO R,MONTERO R S,LLORENTEI M. Key challenges in cloud computing:Enabling the future internet of services[J]. IEEE Internet Computing,2013,17(4):18-25.

[2] YAU S,AN H G. Software engineering meets services and cloud computing[J]. Computer,2011,44(10):46-52.

[3] SHAW M,CLEMENTS P. The golden age of software architecture[J]. IEEE Software,2006,23(2):31-39.

[4] TANG L J,DONG J,ZHAO Y J,et al. Enterprise cloud service architecture[C]// Proceedings of the 2010 IEEE 3rd Interna-

tional Conference on Cloud Computing (CLOUD 2010). New York, NY, USA: IEEE Press, 2010: 27-34.

[5] ROUHANI B D, MAHRIN M N, NIKPAY F, et al. A systematic literature review on enterprise architecture implementation methodologies[J]. Information and Software Technology, 2015, 62(1): 1-20.

[6] DEMIRKAN H, KAUFFMAN R J, VAYGHAN J, et al. Service-oriented technology and management: Perspectives on research and practice for the coming decade[J]. Electronic Commerce Research and Applications, 2008, 7(4): 356-376.

[7] ZHANG X L, YANG J H, SUN X Q, et al. Survey of geo-distributed cloud research progress[J]. Journal of Software, 2018, 29(7): 1-18.

[8] ALJOUMAH E, AL-MOUSAWI F, AHMAD I, et al. SLA in cloud computing architectures: A comprehensive study[J]. International Journal of Grid and Distributed Computing, 2015, 8(5): 7-32.

[9] JIANG Z Y, CHI X J, SHANG G G. A distributed futures program trading platform based on middleware[C]// Proceedings of the 2012 IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS 2012). New York, USA: IEEE Press, 2012: 41-46.

[10] PARAISO F, MERLE P, SEINTURIER L. soCloud: a service-oriented component-based PaaS for managing portability, provisioning, elasticity, and high availability across multiple clouds [J]. Computing, 2016, 98(5): 539-565.

[11] PAHL C. An ontology for software component matching[J]. International Journal on Software Tools for Technology Transfer, 2007, 9(2): 169-178.

[12] NUTI G, MIRGHAEMI M, TRELEAVEN P, et al. Algorithmic trading[J]. IEEE Computer, 2011, 44(11): 61-69.

[13] JULA A, SUNDARARAJAN E, OTHMAN Z. Cloud computing service composition: A systematic literature review[J]. Expert Systems with Applications, 2014, 41(8): 3809-3824.

[14] QI K Y, ZHAO Z F, FANG J, et al. Real-time processing for high speed data stream over large scale data[J]. Chinese Journal of Computers, 2012, 35(3): 477-490.

[15] ZHANG L J, ZHOU Q. CCOA: Cloud Computing Open Architecture[C]// Proceedings of the 2009 IEEE International Conference on Web Services (ICWS 2009). New York, USA: IEEE Computer Society Press, 2009: 607-616.

[16] NEWCOMBE C, RATH T, ZHANG F, et al. How amazon web services uses formal methods[J]. Communications of the ACM, 2015, 58(4): 66-73.

[17] YANG X L, ZHANG H M. Cloud computing and SOA convergence research[C]// Proceedings of the 2012 5th International Symposium on Computational Intelligence and Design (ISCID 2012). New York, USA: IEEE Computer Society Press, 2012: 330-335.

[18] GUILLÉN J, MIRANDA J, MURILLO J M, et al. A service-oriented framework for developing cross cloud migratable software [J]. Journal of Systems and Software, 2013, 86(9): 2294-2308.

[19] MARTIN L, JITKA K. Developing a government enterprise architecture framework to support the requirements of big and open linked data with the use of cloud computing [J]. International Journal of Information Management, 2019, 46(6): 124-141.

JIANG Hui-min, born in 1997, M. S. candidate. Her main research interests include cloud computing and knowledge graph.

JINAG Zhe-yuan, born in 1965, Ph. D, associate professor, is a senior member of China Computer Federation. His main research interests include service-oriented software engineering, software architecture and software engineering environment.