

面向服务的云 workflow 模型与调度研究

于新征, 蒋哲远

(合肥工业大学 计算机与信息学院, 安徽 合肥 230009)

摘 要: 云 workflow 环境具有复杂性, 针对如何在此背景下有效地响应用户的工作流请求和调度 workflow 在云中的执行, 提出了一种面向服务的云 workflow 模型, 利用云计算环境中的计算、存储等服务完成 workflow 实例的执行, 进而满足用户的请求. 基于该模型还设计了一种云 workflow 调度策略和调度算法, 该算法利用 DAG (Directed Acyclic Grapg) 进行建模, 优化资源策略, 保证在最晚期限内使整个 workflow 实例的执行花费尽可能小. 实验结果表明该算法能有效提高 workflow 执行效率, 降低成本.

关键词: 面向服务; 云计算; 云 workflow; 云 workflow 调度

中图分类号: TP393

文献标识码: A

文章编号: 1000—7180(2016)07—0044—05

Service-Oriented Cloud Workflow Model and Scheduling Method

YU Xin-zheng, JIANG Zhe-yuan

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract: In the dynamic environment of cloud workflow, response to the workflow request of users and scheduling cloud workflow is a challenge problem. To solve this problem, a service-oriented cloud workflow model is proposed to satisfy the request of users by the computing and storage service. Then on the basis, the cloud workflow scheduling strategy and scheduling algorithm for workflow were proposed. DAG (Directed Acyclic Grapg) was used in the algorithm, and resource strategy was modified to meet the latest execution time under the premise of minimizing the entire workflow execution costs. The experimental results showed that the efficiency effectively improved and the cost was reduced with the proposed algorithm.

Key words: service-oriented; cloud computing; cloud workflow; cloud workflow scheduling

1 引言

工作流是业务过程的全部或部分自动化流程. 在此过程中, 文档、信息或者任务按照一系列规则在不同的参与者之间流转, 实现组织成员间的协调工作以达到业务的总体目标^[1]. 传统工作流系统结构过于集中, 业务过程和企业资源联系紧密, 系统在灵活性和可用性方面具有一定的缺陷. 利用 Web Service 技术来解决工作流系统的分布性、跨平台性和互操作性, 从而出现了面向服务的工作流管理方法. 当今的工作流通常需要在复杂的分布式计算机系统上执行, 然而构造这样的系统需要付出昂贵的

代价, 申请访问这些系统也需要复杂耗时的过程. 这就要求工作流的运行环境能够进一步扩大, 真正实现无障碍、自由地利用 Internet 资源. 云计算 (Cloud Computing) 的快速发展和成熟为解决工作流运行过程中的系列难题提供了一种新思路, 通过互联网技术形成更大的云计算平台, 将工作流部署到云计算平台, 从而提升工作流的执行性能^[2].

面向服务的云 workflow 技术是云计算应用中提供互联协同功能的核心技术, Xiao Liu^[3]等在集群工作流、网格工作流、云计算等概念基础上, 探讨了一种云计算环境下 P2P 的工作流系统, 并对其中的工作流调度、资源存储问题进行了探讨, 但是未涉及跨

平台问题的研究;Adnan Fida^[4]提出了面向服务的云计算环境(Service Oriented Computing Cloud, SOCC)下 workflow 调度方法,该方法利用服务组合对服务进行重用和扩展以及构造新的应用,李文浩^[5]采用基于 Java 的分布式资源建模和资源管理调度模拟工具包 GridSim 作为实验工具,通过全局 QoS 分解等方法提出了面向社区云的实例密集型 workflow 调度策略.本文提出了一种面向服务的云 workflow 模型,该模型将 workflow 作为云计算环境中的软件服务提供给用户,分别从概念模型和结构模型进行了描述,并设计了该模型中 workflow 调度策略和基于 QoS 约束的调度算法^[6-8].

2 面向服务的云 workflow 概念模型

在面向服务的云 workflow 中,云 workflow 系统负责创建协同服务流程、搭建协同服务执行网络以及管理流程执行过程,是实现云 workflow 系统的核心^[9-10].

2.1 面向服务的云 workflow

面向服务的云 workflow (Cloud Workflow based on Service) 可以描述为一个元组 (Wid, WI, SC, Engine).

(1)Wid 为 workflow 标识.

(2)WI = {wi₁, wi₂, ..., wi_n} 是 workflow 实例的集合,每个实例 wi ∈ WI 实现一个目标相对独立的业务过程.

(3)SC = {sc₁, sc₂, ..., sc_n} 是一个服务云 (Cloud Service) 集合,为所有 workflow 实例提供实现目标所需要的各种服务 (虚拟资源).

(4)Engine = (G_{wi}, G_{sc}) 是面向业务流程目标的工作流引擎,其中 G_{wi} 定义在 workflow 实例集合 WI 和服务云集合 SC 上.该引擎了解服务云集合中的服务状况,能够依据业务流程的总目标以及目标分解规则,规划何时何地以及如何创建工作流实例,并且能够实施对多个 workflow 实例工作的管理和调度. G_{sc} 定义在集合 SC 上,负责服务的组织和管理.

2.2 工作流实例

面向服务的云 workflow 是一个非集中式的系统,服务云 SC 提供了服务的具体位置,要求 workflow 实例能自身调用相应的服务.将业务过程看作一个 workflow 实例,为保证每个 workflow 实例都能得到有效服务,可对云 workflow 实例做出如下定义:

云 workflow 实例 (Cloud Workflow Instance) 可以用一个八元组描述 (wiid, TL, v, SC, p, S, Tol, MC).

(1)wiid 为可认证的实例标识.

(2)TL = ({<v₁, R₁, S₁>}, <V₂, R₂, S₂>, ..., <v_n, R_n, S_n>}, Schedule) 是 workflow 实例携带的任务说明书,包括任务列表 {<v₁, R₁, S₁>}, <v₂, R₂, S₂>, ..., <v_n, R_n, S_n>} 和任务调度 Schedule 两部分组成.其中,任务 v_i 对应业务流程中的活动, R_i 是任务 v_i 的资源需求, S_i 是任务 v_i 的服务需求, Schedule 是依照目标预先定义好的任务执行关系 AR.

(3)v 为云 workflow 实例 wi 当前正在处理的任务, v ∈ TL.

(4)SC 为允许 wi 可利用的服务云集合, SC_{wi} ∈ SC.

(5)p 为 wi 当前所处的云服务, p ∈ SC_{wi}.

(6)S 为 workflow 实例 wi 的当前状态.

(7)ToL 为 workflow 实例 wi 的声明周期.

(8)MC 为 workflow 实例 wi 的工作机,负责任务的执行、协调以及监控等.

2.3 服务云

服务云 (Service Cloud) 可表示为一个二元组形式,即 sc = (Server, ServiceNode).

(1)Server 负责对云服务进行管理,并向服务节点分配流程任务.

(2)ServiceNode 对自身包含的 Web Service、服务器等用于执行云服务的网络资源的抽象化表示,这些服务节点由云成员独立管理和维护.

每个服务云 sc ∈ SC 都负责管理某一类型的服务,服务云是动态的,一局规则对加入、退出的资源进行管理.甚至当云的规模过于庞大时,会依据地理位置、网络配置等主动将其分解为多个云结构.

3 面向服务的云 workflow 结构模型

3.1 云 workflow 系统结构模型

面向服务的工作流系统结构模型包括:

(1)云 workflow 引擎 (Cloud Workflow Engine).

(2)服务云 (Service Cloud).

(3)云服务管理机 (Cloud Service Management Machine).

(4)用户接口 (User Interface).

图 1 给出了面向服务的云 workflow 系统结构模型.

3.2 云 workflow 系统层次描述

从体系结构的角度来看,面向服务的云 workflow 包含四个层次.

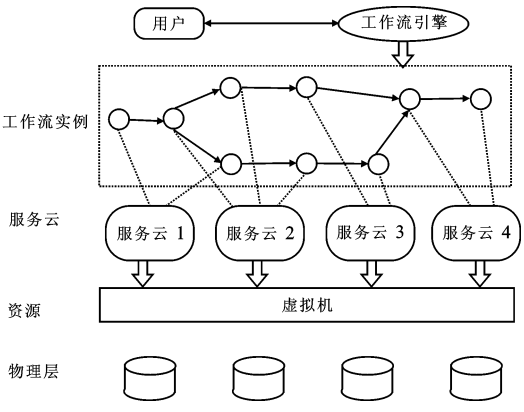


图 1 面向服务的工作流系统结构模型

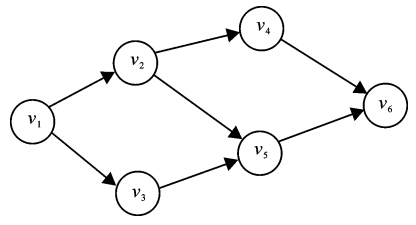


图 2 云工作流 DAG

(1)应用层(Application Layer):主要完成用户或服务注册等用户基本功能,以及工作流任务的提交、管理、监控等功能。

(2)平台层(Platform Layer):主要包含一组资源管理工具和中间件服务. 平台层可以提供一个开发和部署应用程序的环境, 如一个 Web 托管环境, 一个工作流建模和调度服务等. 针对提出的云工作流系统, 这里的平台层即为工作流建模和调度服务提供一个可执行的环境。

(3)统一资源层(Unified Resource Layer):指一些本身虚拟而不是真实的资源,但是可以实现一定操作具有一定功能. 统一资源层为平台层提供系统开发和部署所需的各种资源, 如计算、存储等资源。

(4)基础设施层(Physical Hardware Resource):主要指能够支持计算机正常运行的一些硬件设备及技术,可以使价格低廉的也可以是价格昂贵的服务器或磁盘阵列等设备,利用目前比较成熟的互联网技术(如网络技术、分布式技术等)将遍布各地的计算机组成一个运算、存储功能强大的计算机群,用于计算和存储等云计算操作。

4 云工作流调度策略研究

4.1 工作流模型 DAG 描述

工作流是由一系列的任务和它们之间的依赖关系构成. 在工作流中, 每一个任务都是工作流执行的最小单位, 单个任务执行过程不能被终止. 因此我们使用有向无环图(Directed Acyclic Grapg, DAG)来描述工作流. 图中的节点代表任务, 任务之间的依赖关系由边组成. 如图 2 所示, $V(G)=\{v_1, v_2, v_3, v_4, v_5, v_6\}$ 是 DAG 图中顶点的集合, 表示工作流中单个任务, $E(G)=\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ 表示有向边集

合, 它们表示了工作流的执行方向和任务之间的先后依赖关系. 没有父任务的任务称为初始任务, 表示为 v_{start} , 如图 2 中的 v_1 , 没有后续任务的任务称为结束任务, 表示为 v_{end} , 如图 2 中的 v_6 .

4.2 工作流调度策略

在云计算环境下, 假设工作流的任务可以在任意分配的资源上执行, 每个任务在各资源上的执行时间可以估计, 如果工作流中的任务被分配到不同的资源上则可以并行执行, 但在不同的资源上不可以同时执行同一个任务. 为了更清楚地理解工作流的调度过程, 画了一个简单的调度案例, 如图 3 所示, $v_1 \sim v_5$ 为工作流的任务, $R_1 \sim R_4$ 为工作流调用的服务资源. 工作流的调度过程如下所示。

(1) v_1 作为工作流入口任务, 给 v_1 分配资源后即可执行, 根据相应的调度策略, 任务 v_1 在资源 R_2 上执行。

(2) 根据工作流 DAG 图可以看出, 任务 v_2, v_3 执行需要在 v_1 执行完毕之后, 由于 v_1, v_3 在同一资源上执行, 所以 v_1 执行完成后可以立即执行 v_3 , 而 v_2 则要等到 v_1 执行完后的数据传输到资源 R_3 上才能执行, v_2 和 v_3 可以并行执行。

(3) 任务 v_4 需要在 v_2, v_3 执行完毕之后并且数据传送到资源 R_4 上才可以执行, v_4 在资源 R_4 上执行。

(4) 任务 v_5 需要在 v_4 执行完毕之后并且数据传送到资源 R_1 上才可以执行, v_5 在资源 R_1 上执行。

根据以上调度策略, 最终形成四个任务就绪队列。

4.3 工作流调度算法

在云计算环境中, QoS 作为一种标准来衡量用户对所使用云服务的满意程度. 在云工作流调度中, QoS 的参数主要为成本、时间以及可靠性等. 因此, 我们提出一种减少工作流执行时间和降低成本的工作流调度算法, 即一种基于 QoS 约束的云工作流调度算法, 称为 SACWQ (scheduling algorithm for cloud workflow based on Qos).

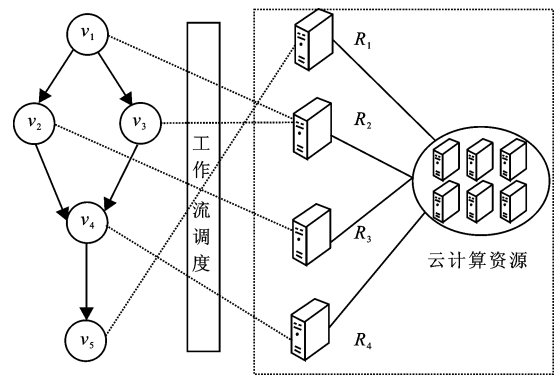


图 3 工作流调度过程

4.3.1 SACWQ 算法描述

该算法需要满足以下几个既定原则.

(1)工作流中的单个任务需要在规定的时间内执行,这样才能保证整个工作流的执行时间在规定时间内.

(2)单个任务分配的最小时间应该大于任务执行所需的最大时间,保证单个任务在规定时间内完成,从而保证工作流所有任务在规定时间内完成.

(3)把用户期望的执行时间按照单个任务最短执行时间比例分配给每个任务,这样可以保证每个任务分配的时间比较合理.

4.3.2 算法执行过程

定义工作流实例执行的最晚期限为 t_d ,可用服务云集合 $SC = \{sc_1, sc_2, \dots, sc_n\}$,任务完成所需时间 t_v 总的花费为 C_r .

第 1 步 输入一组工作流实例和 t_d .

第 2 步 对输入实例进行类型划分

开始执行任务,根据 v_i 是否为 v_{start}, v_{end} 加入到 DAG 中,然后深度优先搜索 v_{start} 到 v_{end} 的关键路径,标记所有任务,并且标记任务的最晚执行时间 t_{start} ,任务的最早完成 t_{end} .

第 3 步 标记在各个可用资源上任务的最短执行时间:

$$t(v_i) = t_{end} - t_{start};$$

(1)

各任务最短处理时间之和为

$$t = \sum_{v_i=1}^n t(v_i)$$

(2)

计算得到

$$t_v = t + R(t')$$

(3)

式中, R 是随机分配函数,将 t' 随机分为 n 个值.

第 4 步 计算出各任务在各可用服务资源上的执行费用 C_r : $C_r = C'_r/[S_s(1 - H)]$, S_s 是服务速度, H 是系统延迟.

$C'_r = (t_d/t)C$, C'_r 是参考的任务费用, C 为执行费用.

第 5 步 为每个就绪任务选取最佳的可用服务资源.

第 6 步 开始执行就绪队列中任务.

5 实验验证及分析

5.1 实验环境与设置

实验环境:Windows 7 操作系统,Eclipse 4.2, JDK1.7 以及 CloudSim 3.0.2. 实验设置 24 种不同类型的服务资源,每种类型的服务资源有 4 个服务提供者,随机产生 500 个工作流实例作为 DAG 实例的输入数据.不同服务提供者提供的服务质量与价格不同,表 1 列出了执行某个任务的可用资源情况.

表 1 模拟云服务资源

可用资源	服务速度/MIPS	所需时间	所需费用
S_1	500	60	6
S_2	1 000	30	12
S_3	2 000	15	24
S_4	4 000	7.5	48

5.2 实验结果与分析

算法执行费用和时间随着最晚期限的变化情况如表 2 和表 3 所示.

表 2 算法执行费用

最晚期限(单位: 基本时间单元)	平均执行费用(单位:基本成本单元)	
	SACWQ	Deadline-MDP
0.5	16.815	19.106
1	13.007	14.796
1.5	11.210	13.467
2	8.632	10.306
2.5	7.378	9.018
3	6.749	8.596
3.5	5.935	7.694
4	5.132	6.896

表 3 算法执行时间

最晚期限(单位: 基本时间单元)	平均执行费用(单位:基本成本单元)	
	SACWQ	Deadline-MDP
0.5	6.016	7.356
1	6.408	7.714
1.5	6.982	8.310
2	7.402	8.796
2.5	7.836	9.307
3	8.219	9.736
3.5	8.505	9.985
4	8.756	10.296

对实验结果数据进行分析,根据表 2、表 3 算法执行时间和执行费用可以对算法的调度性能进行很好的评估.以最后期限为横坐标,分别以执行时间和执行费用为纵坐标对时间结果用如图 4 和图 5 表示.

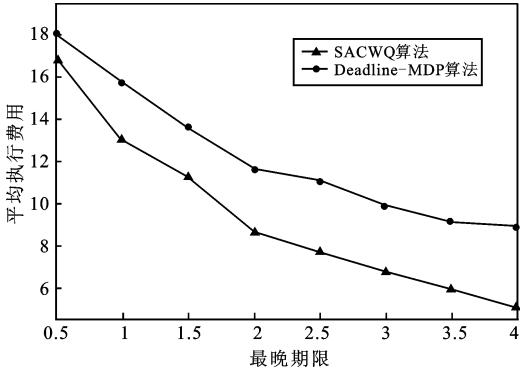


图 4 平均执行费用

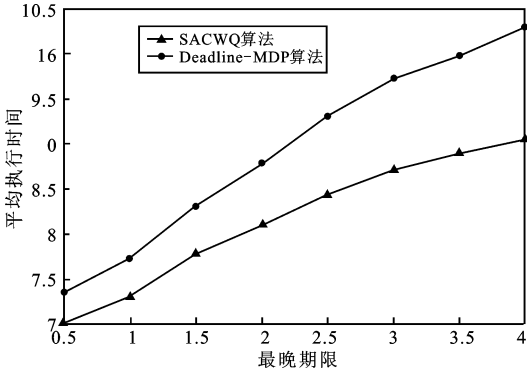


图 5 平均执行时间

由图 5 可知,随着最晚期限的增大,SACWQ 和 Deadline-MDP 的平均执行时间都在升高,但是 SACWQ 算法的平均执行时间明显低于 Deadline-MDP 算法,而且随着最晚期限的增大,差距越明显,证明了 SACWQ 算法减少执行时间的有效性.

6 结束语

本文分析了当前云计算的服务形式以及云 workflow 的研究现状^[11],进而提出了面向服务的云 workflow 概念模型,然后根据概念模型提出了面向服务的云 workflow 结构模型.基于 QoS 的主要参数时间和成本因素,提出了一种减少 workflow 执行时间和降低成本的云 workflow 调度策略,给出了调度算法,并通过实验分析验证了该算法的可行性和有效性,从而提高了 workflow 的执行效率.

参考文献:

[1] Sauter C, Morger O, Sauter C, et al. Die workflow management coalition[J]. Chemical Physics Letters, 1996, 206(5/6):422-428.

[2] Zeng L, Veeravalli B, Zomaya A Y. An integrated task computation and data management scheduling strategy for workflow applications in cloud environments[J]. Journal of Network & Computer Applications, 2015, 50(C):39-48.

[3] Xiao Liu, Dong Yuan, Gaofeng Zhang, et al. SwinD-eW-C: a peer-to-peer based cloud workflow system [M]. Germany:Springer-Verlag, 2009;309-332.

[4] Adnan Fida. Workflow scheduling for service oriented cloud computing[D]. Saskatoon Saskatchewan: University of Saskatchewan,2008.

[5] 李文浩. 面向社区云的实例密集型工作流调度方法研究[D]. 济南:山东大学,2010.

[6] Wang J, Korambath P, Altintas I, et al. Workflow as a service in the cloud: architecture and scheduling algorithms [J]. Procedia Computer Science, 2014, 29 (1):546-556.

[7] Jula A, Sundararajan E, Othman Z. Cloud computing service composition: a systematic literature review[J]. Expert Systems with Applications, 2014, 41 (8): 3809-3824.

[8] Korambath P, Wang J, Kumar A, et al. Deploying kepler workflows as services on a cloud infrastructure for smart manufacturing [J]. Procedia Computer Science, 2014, 29(1):2254-2259.

[9] 卢坤. 面向云计算的工作流系统研究与设计[D]. 北京:北京邮电大学,2013.

[10] 晏婧. 云环境下基于 QoS 约束的工作流任务调度算法研究与实现[D]. 重庆:重庆大学,2011.

[11] Pandey S, Jin C, Voorsluys W, et al. Gridbus workflow management system on clouds and global grids [C]// IEEE International Conference on Escience, Indianapolis, IEEE Computer Society. UK, Oxford, 2008:323-324.

作者简介:

于新征 男,(1990-),硕士研究生.研究方向为智能计算理论与软件.

蒋哲远(通讯作者) 男,(1966-),博士,硕士生导师,副研究员.研究方向为云计算、面向服务软件工程.

E-mail:jzheyuan@sohu.com.