

云计算仿真平台的构建与改进^①

韩珂¹, 蔡小波², 容会¹, 王朝岗³, 张鑫³

¹(昆明冶金高等专科学校 电气学院, 昆明 650033)

²(云南农业大学 基础与信息工程学院, 昆明 650201)

³(云南经济管理职业学院 人文学院, 昆明 650300)

摘要: 由于应用服务受到复杂的组件、配置和部署条件等多维因素的影响, 在实际的云计算数据中心配置不同参数进行测试分析是不现实的. 因此使用模拟工具合理有效的进行资源和任务的调度分配策略仿真实验就尤为重要. 基于此, 本文介绍了三种主流云计算仿真平台, 分别是: CloudSim、CloudAnalyst、CloudReports, 对其各自的特点进行了分析比较, 通过常用的任务调度策略分别对三种仿真平台进行了仿真实验, 对其性能进行评估分析, 并针对其存在的缺陷进行了改进和优化.

关键词: 云计算; 仿真平台; 资源调度

Construction and Optimization of Simulated Platform in Cloud Environment

HAN Ke¹, CAI Xiao-Bo², RONG Hui¹, WANG Chao-Gang³, ZHANG Xin³

¹(School of Electrical Engineering, Kunming Metallurgy College, Kunming 650033, China)

²(College of Basic Science and Information Engineering, Yunnan Agricultural University, Kunming 650201, China)

³(School of Humanities, Yunnan College of Business Management, Kunming 650106, China)

Abstract: It is unrealistic to configure the different parameters and test analysis in a real cloud computing data centers. Because of the multi-dimensional factors such as the application service are effected from the complexity components, configuration, and deployment; it is most important to use Simulation tools to effective for resources and task scheduling strategy simulation experiment. Based on these, we introduce three major cloud computing simulation platform in this paper: CloudSim, CloudAnalyst, CloudReports, analysis and comparion on the characteristics of its own, separately proceed the three kinds of simulation experiment of simulation platform by the usual task scheduling strategy, evaluates and analyzes its performance. The existing defects are improved and optimized.

Key words: cloud computing; simulation platform; resource scheduling

云是以互联网为基础, 对基于网络连接的资源统一进行管理和调度, 构成一个服务资源池, 为基于互联网的应用服务提供可靠、安全、容错、可持续、可扩展的基础设施, 向用户提供按需服务^[1-2]. 云计算需要满足不同用户在不同时间点的服务请求, 这些请求有着不同的配置、部署条件和服务资源数量. 在这种多维因素的影响下, 如何科学的对虚拟资源进行调度就显得尤为重要, 而在实际的云计算数据中心中配置不同参数进行测试是不现实的. 由于不同的应用可能

存在不同的组成、配置和部署要求, 云端基础设施上的应用及服务模型的负载、能源性能和系统规模都在不断变化, 它的首要目标是在云基础设施(软件、硬件、服务)上, 对不同应用和服务模型的调度和分配策略的性能进行量化和比较, 以达到控制使用云计算资源的目的.

1 引言

目前基于云计算建模仿真引起了商业界和学术界广泛的关注, 但是大部分研究工作都集中在云资源管

① 基金项目: 云南省教育厅科学研究基金(2014Z076, 2014Y495, 2011FZ202)

通讯作者: 蔡小波, Email: xbc918@qq.com

收稿时间: 2015-04-21; 收到修改稿时间: 2015-06-08

理策略的设计上,而没有关注仿真平台本身的性能与优化.本文在对云计算仿真平台的研究分析及其应用实例的探究下,遵循云计算的资源分配原则---通过互联网将有效资源分配给有需求的云应用,从而就仿真平台存在的缺陷,例如系统资源利用率和运行效率低等进行改进,使不同用户在不同时间的服务请求的总完成时间最小,而且使资源得到充分利用,提高云计算的资源利用率和运行效率.

因此,使用模拟工具作为实验环境来进行虚拟资源仿真实验来评估虚拟资源调度算法,既可以考察算法所得的结果,也可以测试不同规模数据中心的运行效果.同时,基于云计算仿真器,用户可以反复测试自己的服务,在部署服务之前调节性能瓶颈既节约了大量资金,也给用户的开发工作带来极大的方便.为了建立如此的这种服务,许多工具是可行的,其中目前流行且开源的云仿真平台包括: CloudSim、CloudAnalyst、CloudReports.许多研究机构用云仿真平台建立他们自己的模拟测试计算环境.这里,我们针对三种开源云仿真平台进行模拟分析测试,通过完成复杂云资源管理调度建模,并以图形化的方式直观地表现出虚拟机迁移过程中状态迁移流程和虚拟机迁移策略,实施有效的逻辑结构分析检验,来比较分析三种仿真平台的各自性能.

本文第 2 节综合全面的阐述了目前流行的三种云环境仿真平台的体系结构、工作模型、仿真过程、以及他们的缺陷.第 3 小节章详细展示了缺陷改进后的平台,并在改进前后的云平台上分别设计实现几种资源调度算法,进行实验分析得出结论.最后是对本论文工作进行了总结以及对本论文的未来的工作及扩展进行了展望.

2 云仿真平台

2.1 云体系结构

云计算分为 IaaS、PaaS、SaaS 三种类型,目前还没有一个统一的技术体系,主要采用以下云计算体系结构,如图 1 所示.

云计算技术体系结构分成四层^[3]: 物理资源层、资源池层、管理中间件层和 SOA 构建层.各层之间分工明确,构成云计算系统.物理资源层包括计算机、存储器、网络设施、数据库和软件等.资源池层是将大量相同类型的资源构成同构或接近同构资源池.管理中

间层负责对云计算的资源进行管理,并对众多应用任务进行调度,使资源能够高效安全地为应用服务. SOA 构建层将云计算能力封装成标准的 Web Services 服务,并纳入到 SOA 体系进行管理和使用.云计算技术的最关键部分是管理中间层和资源池层, SOA 构建层的功能依靠外部设施提供.

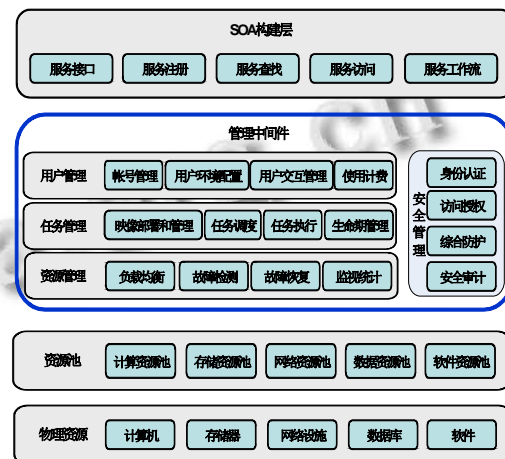


图 1 云计算技术体系结构

2.2 Cloudsim 仿真

CloudSim 云计算仿真工具是澳大利亚墨尔本大学 Rajkumar Buyya 教授领导的网络实验室和 Gridbus 项目推出的云计算仿真软件.它是一个通用、可扩展的,在 GridSim 模型基础上发展而来^[4],继承了 GridSim 的编程模型,提供了云计算的特性,支持云计算的资源管理和调度模拟的研究开发. CloudSim 扩展了部分实现一系列功能的接口,提供基于数据中心的虚拟化技术、虚拟化云的建模和仿真功能.将数据中心的资源虚拟化为资源池,打包对外向用户提供服务, Cloudsim 体现了此特点,扩展部分实现了一系列接口,提供基于数据中心的虚拟化技术、虚拟化云的建模和仿真功能. Cloudsim 平台有助于加快云计算的算法、方法和规范的发展.

2.2.1 Cloudsim 体系结构

Cloudsim 软件框架的多层设计特征以及其体系结构的软件结构框架和体系结构组件包括 SimJava、GridSim、Cloudsim、UserCode 四个层次^[5].如图 2 所示.

Cloudsim 仿真层为云计算的虚拟数据中心环境的配置和仿真提供支持,包括虚拟机、内存、容量及带宽的接口,该层用于主机分配到虚拟机的策略研究,

并通过扩展核心的虚拟机调度函数实现。

Cloudsim 最上层是用户代码层, 该层提供一些基本的实体, 如主机、应用、虚拟机, 用户数和应用类型, 以及代理调度策略等。通过扩展这些实体, 云应用开发人员可以在该层开发各种用户需求分布、应用配置、云可用性场景等应用调度技术, 并执行 Cloudsim 支持的云配置的 Robust 测试。



图 2 Cloudsim 软件框架的多层体系结构

2.2.2 Cloudsim 的仿真过程

CloudSim 模拟器可以对数据中心使用虚拟化技术, 提供主机到虚拟机的映射功能以及资源的检测功能。下图所示为 CloudSim 的工作方式。

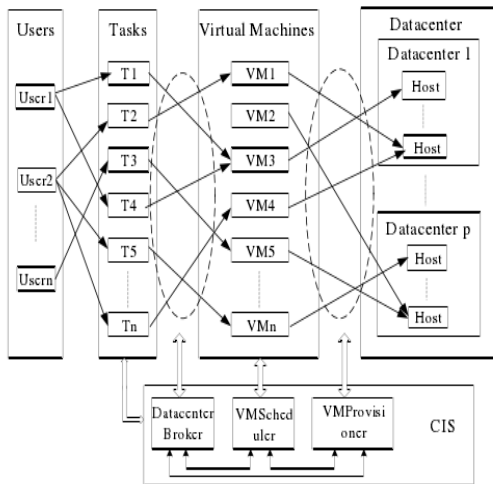


图 3 CloudSim 工作方式

其中, CIS(Cloud Information Service)将用户请求映射到合适的云服务提供商, DatacenterBroker 模拟 SaaS 提供商代理, 根据 QoS 的需求协商资源和服务的分配策略。VmScheduler 是实现主机组件的抽象类, 模

拟虚拟机的分配调度策略, 扩展此抽象类能调整处理器的共享策略。VmAllocationPolicy 代表虚拟机监视器调度策略, 该策略用于将虚拟机分配给主机。通过扩展 DataCenterBroker 的这个类来实现任务调度算法的开发, 为用户分配合适的虚拟机资源。

实现 CloudSim 模拟器仿真的步骤如下:

1) GridSim 库的初始化。

初始化的参数包括: 用户数量(num_user)、日期(calendar)、追踪标志(trace_flag)、文件列表(exclude_from_file)、进程列表(exclude_from_processing)、输出报告(report_name)。

2) 创建数据中心

在 CloudSim 仿真平台中, 一个数据中心由一个或多个 Machine 组成, 一个 Machine 是由一个或多个 PE 或 CPUs 组成。CreateDatacenter 函数用来创建数据中心利用此函数可以创建多个数据中心。

3) 创建代理 Broker

数据中心代理(DatacenterBroker)的功能是利用用户的身份通过云信息服务(CloudInformation Services, CIS)为用户选择合适的云服务提供商, 并要求云服务提供商按照用户的服务质量要求对其分配合适的资源。此外, CloudSim 模拟器提供了很好的云任务调度算法仿真平台, 用户可以调用适当的 API 函数来开发自己的任务调度算法, 来进行模拟实验。

4) 创建虚拟机

首先创建存放虚拟机的虚拟机列表 vmList; 然后创建虚拟机 vm1, 并对其参数进行设置, 包括虚拟机 ID 号(vmid)、指定 Broker 的 ID 号(BrokerId)、镜像大小(size)、内存大小(memory)、带宽(bw)、CPU 数量(vcpus)、优先级(priority)、虚拟机名(vmm)以及调度策略; 最后将已经创建成功的虚拟机 vm1 加入虚拟机列表 vmList 里并向数据中心代理 broker 提交虚拟机列表。

5) 创建云任务

首先创建存云任务的云任务列表 cloudletList; 然后创建云任务 cloudlet1, 并对其参数进行设置, 包括云任务 ID 号(id)、云任务长度(length)、文件大小(file_size)以及输出大小(output_size); 最后将已经创建成功的云任务 cloudlet1 加入任务列表 cloudletList 里并向数据中心代理 broker 提交云任务列表。

6) 启动仿真

7) 在仿真结束后统计结果。

2.3 CloudAnalyst 仿真

针对云计算的用户群的分布、用户群的动态属性、各个区域的网络设施状况等影响云计算服务的开销的问题, CloudAnalyst^[7]可以方便地开展大量实验并且节约成本。CloudAnalyst 是基于 Cloudsim 的基础上而开发的, 它具有更加人性化的用户界面。作为云计算仿真工具, CloudAnalyst 将模拟实验与编程分开, 因此建模可以专注于模拟实验, CloudAnalyst 通过改变参数就可以反复执行模拟, 进行一系列的仿真实验, 实现各个用户群和数据中心之间的最优调度。

CloudAnalyst 不仅用户界面简单易用; 而且可以灵活的配置模拟中使用到的属性。同时还可以重复进行数据的测试, 测试结果以图形化方式输出。另外 CloudAnalyst 是基于模块化设计的, 可以很容易地实现扩展, 具有较高可靠性和实用性。

2.3.1 CloudAnalyst 主要模型和组件

1) Region

CloudAnalyst 将世界分为 6 个地区, 分别代表世界上主要的大洲。其他的模型如 User Bases 和 Data Centers 都是位于在这六个地区中。用户和数据中心的网络属性与它们所在的地区有关。

2) Internet

CloudAnalyst 中的网络属性是对真实世界网络属性的模拟, 包含了真实世界比较重要的属性, 主要包含了数据的传输时延和带宽。

3) Cloud Application Service Broker

该组件主要实现的功能是把用户请求分配到某个具体的数据中心, 具体分配算法有 3 种, 即最小时延算法、最佳性能算法、动态路由算法。

4) User Base

用户群是单个用户的集合, 以便一起提交任务来提高系统性能。

5) InternetCloudlet

用户请求任务组: 一组用户请求的集合。

6) Data Center Controller

主要是管理数据中心, 如虚拟机的创建和销毁, 接受用户请求并分配到虚拟机上。该组件是 CloudAnalyst 中最主要的部分。

7) VmLoadBancer

负载均衡器。它决定了用户请求分配到哪个虚拟

机上。具体的分配算法有三种, 轮转算法、动态监控算法、节流分配算法^[6]。其中, 轮转算法采用 Round-robin 算法分配虚拟机; 动态监控算法则是监控每个虚拟机上的任务数, 分配用户请求使每个虚拟机的任务数保持均衡。而节流分配算法需要先为数据中心里的虚拟机设定一个最大用户请求数, 当用户数量达到该值后, 则不能继续接受用户请求。

8) GUI

该组件主要是实现图形界面的功能。

CloudAnalyst 的主要模块和组件如下图 4 所示。

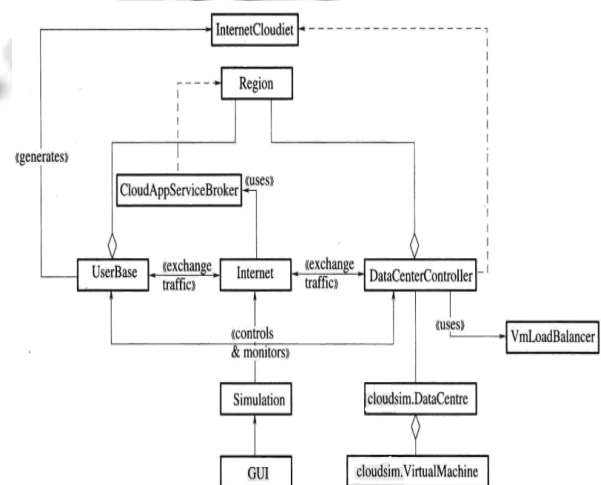


图 4 CloudAnalyst 主要模块和组件

2.3.2 CloudAnalyst 仿真过程

1) CloudAnalyst 运行及配置方法

找到 main 函数的位置 source->cloudsim. ext. gui->GuiMain.java, 右键运行。

2) 需要详细配置物理机的具体信息包括内存大小、存储大小、网络带宽大小、CPU 数量、处理器速度、虚拟机策略。

4) 配置好一个数据中心后可以点击 Add New 增加数据中心, 具体配置同数据中心类似。

5) 数据中心的配置, 用户群的配置信息包括: 所在地区、请求数量、请求大小、高峰期开始和结束时间、高峰期数量、一般时期用户数量。

6) 选择服务代理的策略。其中的数据中心, 只能从开始配置好的数据中心里选择。

7) 高级特性配置。其中的负载均衡器决定了用户请求分配到哪个虚拟机上。具体的分配算法有三种, 轮转算法、动态监控算法、节流分配算法。

8) 配置网络信息

回到主界面后, 点击 Define Interent Characteristics 配置网络信息, 包括的信息有时延矩阵和网络带宽矩阵, 可以设置每两个地区间的网络的带宽和时延。

9) 运行配置后的 CloudAnalyst

配置好后可以进行模拟, 点击 Run Simulation, 此时在确保配置保存后, 只需重新运行并加载刚刚的配置即可。

最后的结果还可点击 Export Results 保存为 PDF 文件, 最后显示的信息有所有数据中心的平均响应时间、最小响应时间、最大响应时间、数据中心处理请求的平均时间、最小时间、最大时间, 数据中心处理时间用画图表示。

2.4 CloudReports 仿真

CloudReports 是一个模拟了基于云计算模式下分布式计算环境的图形化操作工具。它采用 CloudSim 作为仿真引擎, 提供了一个易于使用的用户界面, 以及报表生成和扩展的功能。CloudReports 可以模拟提供任意数量数据中心基础设施的服务(IaaS), 每个数据中心是完全可定制的^[8]。用户可以很容易的计算节点的数量(主机)和他们的资源配置, 包括处理能力, 内存, 可用带宽, 功耗和调度算法。客户也可以模拟和定制提供商提供的服务。用户可以设置虚拟机的数量, 每个客户拥有一个代理, 负责分配这些虚拟机和资源消耗的算法。每个虚拟机都有自己的配置, 包括它的管理程序、图像大小、任务调度算法、所需的处理能力、内存和带宽。此外, cloudreports 生成的 HTML 文件可以报告模拟和原始数据的情况, 该文件可以很容易地导入至第三方软件如 MATLAB。通过 CloudReports 图形界面, 可以很灵活的创建数据中心、主机以及配置不同的虚拟机, 具有可扩展性和可定制性。

2.4.1 在 Eclipse 中配置 CloudReports

1) <http://travis-ci.org/thiagotts/CloudReports> 上下载 CloudReports 项目。

2) 在 Eclipse 中选择文件->新建项目->Java 项目。

3) 在创建 Java 项目窗口中选择“使用缺省位置”, 并将目标定位至 CloudReports 目录下的 src 文件夹下, 并填上项目名。

4) 选择在 Eclipse 中选择 cloudreports 文件夹右键后选择“构建路径”->配置构建路径->添加外部 JAR。

5) 选择 CloudReports\src\main\java\cloudreports

\gui\ MainView.java, 并右键运行即可。运行界面如图 5。

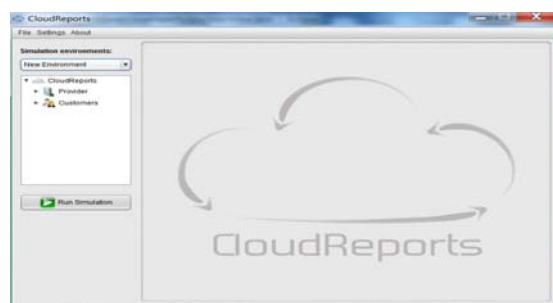


图 5 CloudAnalyst 运行界面

2.4.2 CloudReports 扩展

CloudReports 可以创建独立的扩展, 而无需重新编译其源代码。扩展必须从一个基类或实现接口中继承而来。目前, 支持的扩展类型如下:

- 1) 虚拟机分配策略(Virtual machines allocation policies): 类型: vmallocationpolicy
- 2) 代理策略(Broker policies): 类型: Broker
- 3) Processinglementsprovisioners: 类型 : PeProvisioner
- 4) RAM provisioners:类型: RamProvisioner
- 5) Bandwidth provisioners:类型: BwProvisioner
- 6) Cloudlets schedulers:类型: CloudletScheduler
- 7) Power consumption models:类型: PowerModel
- 8) Resource utilization models:类型: Utilization Model
- 9) Virtual machines schedulers:类型: VmScheduler

使用 Eclipse 扩展 CloudReports

1) 在 src/main/java 下新建包 brokers, 并在包里新建类 MyBrokers。

2) 选择导出文件, 选择 Java->JAR 文件, 选择扩展项目文件 CloudReports, 点击完成。生成的 JAR 文件包含了所有文件。本文中生成的 JAR 包为: MyExtensions.jar。

3) 加载扩展。新建文件夹 Cloudextension, 将导出的 MyExtension.jar 放入此文件夹, 并在此文件夹下新建目录 extensions, 在此文件夹下新建文件 classnames.xml。

4) 将文件夹 Cloudextension 导入至 Eclipse, 并运行, 此时的 CloudReports 将出现扩展后的代理策略 MyBroker。

本节介绍了 CloudSim、CloudAnalyst 和 CloudReports 三种现有的云计算数据中心模拟系统,并对其各自的特点进行了概述。基于 CloudSim 的仿真模拟过程都是通过编写代码实现,仿真结果的呈现也很单一,扩展需要一定的 Java 编程技巧。CloudAnalyst 是基于 CloudSim 的基础上而开发的界面式仿真工具。所以使用其进行仿真具有很大的优势,用户只要操作界面就能进行云计算的仿真,仿真过程不涉及编程。因此研究者建模时可以专注于模拟实验,无需花费太多的时间在编程上。CloudAnalyst 通过改变参数就可以反复执行模拟,进行一系列的仿真实验,实现各个用户群和数据中心之间的最优调度,测试结果以图形化方式输出。CloudReports 是一个模拟了基于云计算模式下分布式计算环境的图形化操作工具。它采用 CloudSim2.1.1 作为仿真引擎,提供了一个易于使用的用户界面,以及报表生成和扩展的功能,易于使用图形用户界面和利用 CloudSim 框架提供的功能进行模拟创建一个云环境。此外,cloudreports 生成的 HTML 文件可以报告模拟和原始数据的情况。通过 CloudReports 图形界面,可以很灵活的创建数据中心、主机以及配置不同的虚拟机。

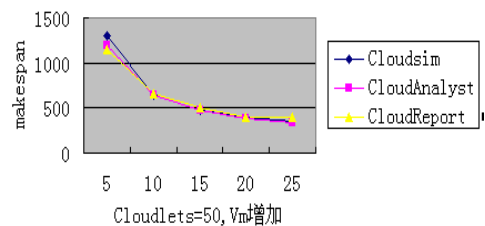
3 实验设计与分析

本实验采用贪心(Greedy)调度算法分别在云计算仿真平台 CloudSim、CloudAnalyst 和 CloudReports 进行模拟,重写了 DataCenterBroker 类中提供的 bindCloudletToVm(int cloudletId, int vmId)进行测试模拟,编写新方法 bindCloudletsToVmsGreedy(),编程工具为 Eclipse-SDK-1.7.6。评价指标采用 makespan,即 Maximum Completion Time(MCT),虚拟机上的所有任务的最大完成时间,期望 makespan 最小。

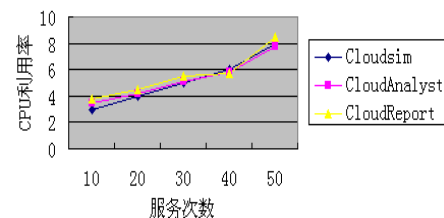
此次实验的 VM 分别为 5、10、15、20、25, Cloudlets 分别为 10、20、30、40、50,仿真实验在云任务个数固定即 Cloudsimlets=50 时,虚拟机个数从 5、10、15、20 到 25,在三个仿真平台上运行,最后统计每次的 makespan 的平均值,测试结果如图 6 所示。

本文在 CloudSim、CloudAnalyst 和 CloudReports Cloudsim 进行虚拟资源调度实验,采用的算法为轮转算法、动态监控算法、节流分配算法。请求服务为 10、20、30、40、50 次,仿真运行的实验结果如图 32 所示。从图 6(a)可以看出,当我们设定情景云计算任务为 50

个,虚拟机个数为 20 台时,Cloudsim 上所有虚拟机上的所有任务的最大完成时间为 757.59s,Cloudsim+上所有虚拟机上的所有任务的最大完成时间为 346.01s;由图 6(b)可知,当请求服务次数为 20 次时,Cloudsim 的 CPU 利用率为 11.36%,Cloudsim+的 CPU 利用率为 4.23%。



(a) Cloudlets=50, Vm 增加, 所有任务的最大完成时间



(b) 请求服务次数递增时, 仿真平台的 CPU 利用率

图 6 测试结果

实验可知使用 CloudSim 进行仿真的过程中随着实验规模的增大而导致效率低下以及资源占用率高,有着运行时间长、使用内存多和 CPU 利用率高等缺点,这是因为很多仿真无用的事件和实体增加了系统开销。经过优化后,使用改进的事件管理和实体管理机制来除去不必要的操作,使得仿真运行时间和系统资源占用率大幅得到改善,让仿真平台更为接近真实的云计算环境,具有较高的实用性。

参考文献

- 1 Armbrust M, Fox A, Griffith R, et al. Above the clouds: a Berkeley view of cloud computing, mimeo, UC Berkeley, RAD Laboratory, 2009.
- 2 Erl T, Puttini R, Mahmood Z. Cloud computing: concepts, technology, & architecture. Pearson Education, 2013.
- 3 Kumar R, Sahoo G. Cloud computing simulation using CloudSim. arXiv preprint arXiv:1403.3253, 2014.
- 4 Wickremasinghe B, et al. CloudAnalyst: A CloudSim-based tool for modeling and analysis of large scale cloud computing

- environments. Proc. of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010). Perth, Australia. April 20–23, 2010.
- 5 Calheiros RN, Ranjan R, Beloglazov A, et al. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 2011, 41(1): 23–50.
- 6 田文洪等. 云计算: 资源调度管理. 北京: 国防工业出版社, 2011.
- 7 Wickremasinghe B, Calheiros R N, Buyya R. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA). IEEE. 2010. 446–452.
- 8 陈志波, 陆雍森. Surfer 在环境评价和规划中的应用. *同济大学学报(自然科学版)*, 2005, 33(2): 191–195.
- 9 Sá TT, Calheiros RN, Gomes DG. CloudReports: An extensible simulation tool for energy-aware cloud computing environments. *Cloud Computing*. Springer International Publishing, 2014: 127–142.
- 10 章坚民, 徐爱春, 李海翔等. 基于SVG/XML/CIM的变电站自动化工程配置系统. *电力系统自动化*, 2004, 28(14): 53–56.
- 11 Zhang JM, Xu AC, Li HX, et al. An automatic engineering configuration system for substation automation based on SVG/XML/CIM. *Automation of Electric Power Systems*, 2004, 28(14): 53–56.