

SHMA:一种云平台的监控框架

陈林 应时 贾向阳

(武汉大学软件工程国家重点实验室 武汉 430072) (武汉大学计算机学院 武汉 430072)

摘要 由于云平台架构的复杂性和工作负载的不可预测性,云监控对保证云平台的高可用性有重要意义,云平台上资源的动态性、多样性以及资源规模巨大给云平台监控带来了一定的困难。研究并提出一种面向云平台的可扩展多层次监控框架 SHMA,SHMA 采用微服务架构构建监控系统中可扩展的各个相互独立的服务组件,实现对云平台不同层次的应用服务、中间件和基础设施资源的监控。通过对部署并运行在 CloudStack 基础设施云平台上的某综合减灾空间信息服务应用系统进行案例分析,验证了该监控框架的有效性。

关键词 云监控,微服务,可扩展,多层次

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.01.002

SHMA: Monitoring Architecture for Clouds

CHEN Lin YING Shi JIA Xiang-yang

(State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China)

(Computer School, Wuhan University, Wuhan 430072, China)

Abstract Monitoring plays a significant role in ensuring the availability of cloud computing, because of the complex architecture of cloud and the unpredictability of workload. It provides the real-time monitoring data, the status of the running system, and helps to discover the defects and optimize system performance. However, due to the heterogeneity of cloud, the resources on cloud are often dynamic, diverse and enormous scale, which bring about a great challenge for cloud monitoring. To solve this problem, we proposed a scalable and hierarchical monitoring architecture for clouds (SHMA). SHMA uses microservices to create each independent component of the scalable monitoring system. It realizes the goal of offering cloud providers and cloud consumers to have an overview of the resources on three levels of clouds, including the runtime information of application services, middleware and infrastructure. Finally, we used a case of comprehensive disaster reduction system on CloudStack to evaluate the effectiveness of the architecture.

Keywords Cloud monitoring, Microservices, Scalable, Hierarchical

1 引言

云计算是一种新型的计算和服务模式,它基于分布式计算、并行计算、网格计算和虚拟化等技术,通过建立计算资源共享池,以按需付费的方式向用户提供广泛的计算、存储、数据库、分析、应用程序和部署等多种类型的云服务。根据 NIST^[1]的定义,云计算可以提供多个不同层次的服务模式:基础设施即服务(Infrastructure as a Service, IaaS)、平台即服务(Platform as a Service, PaaS)和软件即服务(Software as a Service, SaaS)。

在云计算平台上,有许多应用对性能有严格的要求,例如某综合减灾空间信息服务应用系统,当灾害发生时,该系统应在最短的时间内从空间和时间两个维度对自然灾害的风险和损失进行可视化展示,为各项灾害管理工作提供直观的信息。

为了更好地满足系统性能的要求,云平台应根据需求的动态变化快速地调整资源供给策略,实施云监控,可以实时量化云计算平台资源的使用情况,提供直观的性能监控数据,使云服务提供者和消费者第一时间了解系统性能,并及时掌握整个系统的运行状态信息,发现系统缺陷、指导负载均衡和资源调度,优化系统性能。因此,对云计算平台进行监控是十分必要的。

目前,云计算平台的监控框架必须考虑的一个问题是:在云计算平台中,不同的资源分布在不同的层次上,硬件资源分布在 IaaS 层,中间件资源分布在 PaaS 层,应用服务资源分布在 SaaS 层,在运行过程中,各层资源都会对系统性能产生影响,单独监控某一层次上的资源不能全面地展示系统的性能特征,因此监控服务需要采取多层次的监控方法对云平台 3 个层次上的资源进行全面的监控。此外,云计算环境中的资

到稿日期:2015-12-07 返修日期:2016-01-20 本文受国家自然科学基金面上项目:面向性能改善的 SaaS 软件部署方案自优化方法(61373038),国家 863 高技术研究发展计划:网构化软件运行支撑和在线管理技术与机制(2012AA011204-01),国家自然科学基金重点项目:可信软件构造理论与方法研究(91118003)资助。

陈林(1992-),女,硕士生,主要研究方向为云平台性能监控,E-mail:1450825830@qq.com;应时(1965-),男,博士,教授,主要研究方向为面向服务的软件工程、语义智能软件、高可信软件、云计算与物联网;贾向阳(1972-),男,博士,讲师,主要研究方向为服务计算、中间件技术、高可信软件。

源可能随时加入或离开,这使得云平台上的资源具有动态性、多样性和规模巨大的特点,这就要求监控系统必须具有良好的可扩展性。

针对云计算平台的这些特点,提出了一种面向云平台的可扩展多层次监控框架 SHMA (Scalable and Hierarchical Monitoring Architecture for clouds), 并采用微服务^[2,3] 架构构建监控系统中各个相互独立的服务组件, 实现系统原型。SHMA 具有多层次和可扩展的特点, 可以适应云平台中各个层次上不同的服务内容和运行机制, 为不同层次上的被监控资源提供不同的监控服务, 实现对云平台不同层次的应用服务、中间件和基础设施资源的监控。对部署并运行在 Cloud-Stack 基础设施云平台上的某综合减灾空间信息服务应用系统进行的案例研究表明, SHMA 监控框架能够实现有效监控。

2 相关工作

随着云计算技术的不断发展, 云平台提供者和消费者越来越关注云计算平台的运行状态。许多研究学者对云监控做了大量的研究工作。DARGOS^[4] 是一个分布式云平台资源管理监控框架, 它为多租户云平台提供物理基础设施资源和虚拟资源的监控, 但该框架无法满足大型应用场景中监控服务高度可扩展的需求。GridICE^[5] 以监控网络资源为目标, 为用户提供故障监测报告、服务级协议违规以及用户定义事件机制。He Huang^[6] 提出了一种推拉混合模式监控云平台资源, 在保证准确性和一致性的前提下调整数据传输策略, 并根据用户的需求过滤监控数据, 它主要解决了性能监控数据的传输问题, 但没有考虑云平台监控资源的分布和监控方法的可扩展性。Hyuck^[7] 介绍了一种使用 RESTful 的方法来监控云计算平台上的基础设施资源, 如计算、存储以及网络资源等, 但是这样的监控架构只适合监控云计算平台上的基础设施资源, 对云平台上运行的服务和应用的监控则存在不足。Rak^[8] 提出了一个可以监控云计算基础设施的集中式的监控架构, 但该架构由于集中监控的方式缺乏可扩展性, 因此只适用于小型的部署方案。Gregory^[9,10] 介绍了一种分层的自适

应云监控框架, 但其不支持动态的扩展监控软件, 存在监控不足的情况。候永东^[11] 提出了一种可扩展集成化的云平台监控机制, 该监控机制可根据需求集成多种监控软件, 但没有考虑监控服务本身的可扩展性, 无法进行动态伸缩。

许多开源项目也对云计算平台上的资源监控进行了研究开发工作。这些企业级项目可以实现对大型集群系统的监控, 但通常是针对物理基础设施进行的监控, 缺乏对云平台中应用服务的监控。Ganglia^[12,13] 是由加州伯克利分校发起的开源集群监视项目, 可以监控 CPU、内存、硬盘利用率、I/O 负载、网络流量等, 通过网页可以查看集群中节点的信息, 做到对集群系统的实时监控。Nagios^[14] 是一款可运行在 Windows, Linux 和 Unix 平台之上的开源网络监视系统, 可以用来监控系统的运行状态和网络信息。Zabbix^[15] 是一个提供网络监控和分布式系统监控功能的企业级开源解决方案, 主要完成对硬件信息或与操作系统有关的内存、CPU 等信息的收集。Monitis^[16] 云监控包括云的系统监控、性能测试和配置管理, 使用 Monitis 可以消除监控动态云资源过程的复杂性。

为了解决以上工作中存在的问题, SHMA 设计了一个面向云平台的可扩展多层次监控框架, 可以对云平台中 IaaS 层物理基础设施、PaaS 层运行的中间件以及的 SaaS 层的应用服务进行多层次的监控, 并采用微服务架构构建监控系统中各个相互独立的监控服务组件, 实现监控系统高度的可扩展性。

3 云平台的监控框架

如图 1 所示, SHMA 是一个面向云平台的可扩展多层次监控框架。图 1 的左边部分为云计算平台的 3 个层次, IaaS 层包括服务器、存储、网络等物理基础设施, PaaS 层包括了软件运行的中间件, SaaS 层提供了各种运行的应用服务; 右边部分为本文提出的云平台监控框架, 该框架针对左边云计算平台的 3 个层次提供 3 个层次的监控服务, 包括 IaaS 层的基础设施数据采集服务, PaaS 层的全局数据采集管理服务、中间件数据采集服务、全局数据存储服务, 以及 SaaS 层的局部数据采集管理服务、应用数据采集服务和局部数据存储服务。

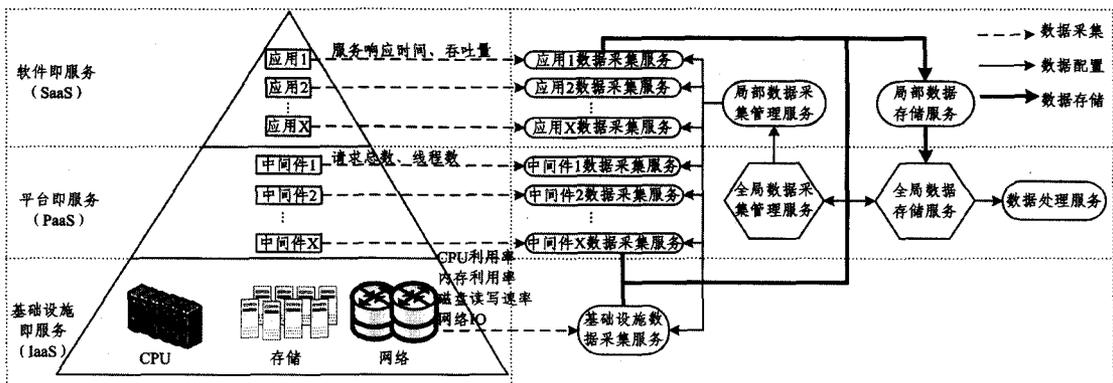


图 1 云平台监控框架

如图 2 所示, 云平台的监控流程分为 5 个步骤: (1) 云平台监控通过全局数据采集管理服务将各虚拟机节点上被监控资源的相关监控信息发送到局部数据采集管理服务; (2) 局部数据采集管理服务接收全局数据管理服务的配置信息, 开启对基础设施数据采集 (2a)、中间件数据采集服务 (2b) 和应用数据采集服务 (2c); (3) 局部数据存储服务接收基础设施数据采集服务 (3a)、中间件数据采集服务 (3b) 和应用数据采集服

务 (3c) 监控的性能数据, 并对这些数据进行存储; (4) 局部数据存储服务每隔一定时间将性能数据传输至全局数据存储服务中进行统一的存储管理; (5) 全局数据采集管理服务从全局数据存储服务中获取性能数据, 对性能数据进行评估, 根据预先定义的监控策略来调整监控采集的时间间隔。下面从 SHMA 监控框架的多层次性和可扩展性两个部分对框架的各个服务进行详细叙述。

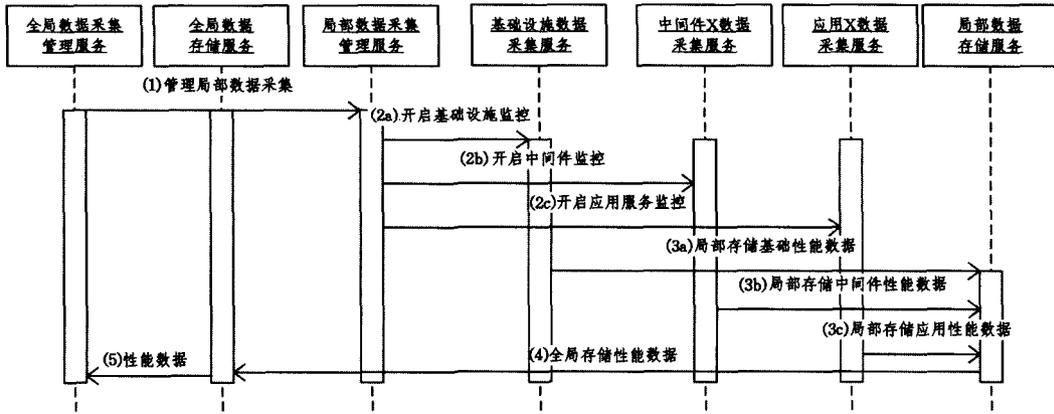


图 2 监控时序图

3.1 监控框架的多层次性

3.1.1 IaaS 层

基础设施数据采集服务:此服务位于虚拟机节点上,负责采集 IaaS 层面上各个虚拟机的基础性能数据,包括虚拟机的基本信息如,CPU 核数、内存大小、磁盘空间大小,还有运行时的情况如 CPU 利用率、内存使用率、磁盘读写速率、网络速率,这些采集到的性能数据按 JSON 数据格式发布给局部数据存储服务进行存储。

3.1.2 PaaS 层

全局数据采集管理服务:全局数据采集管理是 SHMA 监控框架的核心服务,负责管理云计算平台中所有被监控的资源,它以虚拟机节点为单位并按表 1 所列的 XML 配置管理该虚拟机节点上的所有被监控资源,配置文件的内容包括被监控的虚拟机节点的 IP 地址、采集服务名称、被监控资源类别、被监控资源名称、采集时间间隔、监控性能指标。全局数据采集管理服务还可以对监控采集的性能数据进行评估,根据预先定义的监控策略调整监控采集间隔,尽量减小性能监控带来的系统负载。

表 1 数据管理服务配置

```

<DataCollectionConfiguration>
  <IP>IP 地址</IP>
  <CollectName>采集服务名称</CollectName>
  <ResourceCategory>类别名称</ResourceCategory>
  <ResourceName>资源名称</ResourceName>
  <CollectInterval>xx</CollectInterval>
  <MetricProperty>{xx,yy,zz...}</MetricProperty>
</DataCollectionConfiguration>

```

中间件数据采集服务:不同的中间件使用不同的中间件数据采集服务实现性能监控,根据局部数据采集管理服务的定义采集中间件产品如 Mysql、Tomcat 的线程数、连接数等性能数据,按 JSON 数据格式发布到局部数据存储服务中存储。

全局数据存储服务:提供持久化数据存储服务接口,接收来自 SaaS 层中局部数据存储服务存储的每个虚拟机节点上的性能监控数据,对性能数据进行统一的管理存储,并对外提供历史性能数据和实时性能数据的查询接口。

3.1.3 SaaS 层

局部数据采集管理服务:云平台中运行着大量的虚拟机实例,仅仅通过全局数据采集管理服务来管理所有虚拟机节点的资源会给她带来极大的负担,因此本文采用局部数据采集管理服务来缓解压力。局部数据采集管理服务位于每个虚拟机节点中,接收全局数据采集管理服务对该虚拟机节点上

被监控资源的配置信息,管理该虚拟机上资源的性能数据采集器的名称、数据采集方法、采集时间间隔和采集性能指标,并开启应用性能数据采集、中间件性能数据采集和基础设施性能数据采集。

应用数据采集服务:通常应用开发人员提供应用的性能监控接口,应用数据采集服务调用性能监控接口,根据局部数据采集管理服务的定义按规定的的时间间隔自动采集应用服务的响应时间、吞吐量等性能指标,并将产生的性能数据按 JSON 数据格式发布到局部数据存储服务中进行存储。

局部数据存储服务:云平台中采集的所有性能数据如果直接由数据采集服务发布到 PaaS 层的全局数据存储服务进行存储管理,则全局数据存储服务需要同时处理大量的监控数据,这会造成服务器的负载过大,并且,传输大量性能数据也会占用很大的网络带宽。本文采用局部数据存储服务来解决这一问题,该服务以一个虚拟机为单位,应用数据采集服务、中间件数据采集服务和基础设施数据采集服务监控采集到的性能监控数据传输到局部数据存储服务中进行存储,每隔一定时间再传输至全局数据存储服务进行存储。局部数据存储服务中存储的是该虚拟机节点上的性能监控数据,该性能数据随着数据采集不断变化。

3.2 监控框架的可扩展性

本文使用微服务架构实现监控框架的可扩展性。微服务架构使用一系列小服务来开发应用,每个服务运行在自己的进程中,通过轻量级的协议进行通信,这些服务基于业务能力构建,能够独立部署在一台或者多台服务器上。微服务架构最大的优势在于它提供了灵活性和可扩展性。如图 1 所示,监控系统中的每个微服务都有自己的业务逻辑,完成特定的功能,并且相互独立。

云平台监控服务的可扩展性包含了横向扩展和纵向扩展两个方面。横向扩展是指功能的扩充,如增加新的监控流程和支持新的监控运行环境。云平台上的被监控资源具有动态性,当加入新的资源如新部署的应用服务 X 时,如果需要对应用服务 X 进行监控,可以通过向全局数据采集管理服务进行注册,注册配置信息如表 1 所列,包括应用服务 X 部署的虚拟机节点 IP 地址、性能数据采集器名称、采集时间间隔、性能采集指标相关信息,然后将配置信息发送到应用服务 X 所在虚拟机节点的局部数据采集管理服务,以开启对应用服务 X 的监控。如果监控系统需要新增预警服务功能,在不修改原有监控系统的前提下,只需根据系统预留的服务接口加入预警服务,即可从全局数据存储服务中获取性能监控数据,从

而实现对云平台的性能预警。

纵向扩展是指在功能不变的情况下提高系统的性能和扩大用户的支持规模。当云平台中某一类型的资源数量巨大时,仅仅通过一个该资源的数据采集服务实例监控性能状态无法满足监控的需求,由于监控系统采用的微服务架构设计使得不同服务之间相互独立,可以根据被监控资源的规模动态地扩展该监控服务的实例数量来满足监控需求,以提高监控效率。同时,如果要修改某一服务的处理能力或处理模式,可以通过重写该服务来进行扩展维护,无需对其他服务做任何修改。

4 系统实现

4.1 性能数据

JSON(JavaScript Object Notation)^[17]是一种轻量级的数据交换格式,采用完全独立于语言的文本格式,易于阅读和编写,同时也易于机器解析和生成。传输 JSON 格式的性能数据带来的传输负载很小,非常符合云监控系统的要求。本文采用表 2 定义的 JSON 数据格式,将各个数据采集服务采集监控的性能数据发送给局部数据存储服务或全局数据存储服务进行存储,性能数据包括资源采集时间,资源类别,性能数据的名称、数据值和数据单位。

表 2 JSON 性能数据

```

{
  "MonitoringData": [
    {
      "timestamp": "采集性能数据的时间点",
      "ResourceCategory": ["类别名称 1", "类别名称 2", ...],
      "Metric": [
        {
          "类别名称 1": [
            {
              "name": "指标名",
              "value": "指标值",
              "valueType": "指标值单位", ...
            }
          ]
        }
      ]
    }
  ]
}

```

4.2 监控系统实现

由于被监控资源的多样性,本文的监控框架采用了多种不同的监控技术来满足云平台的不同监控需求,如图 3 所示。

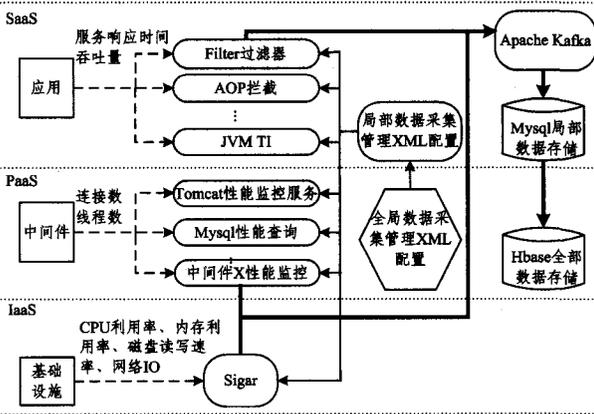


图 3 云平台监控框架的技术实现

• 局部/全局数据采集管理服务

XML(Extensible Markup Language)^[18]是一种可扩展标记语言,具有平台无关性、语言无关性、系统无关性,被广泛地作为跨平台之间交互数据的形式。采用表 1 所列的 XML 格式对全局数据采集管理服务进行资源配置,并按虚拟机节点将 XML 配置文件发布到局部数据采集管理服务。

• 基础设施数据采集服务

在框架实现中,使用跨平台收集系统资源的类库 Sigar (System Information Gatherer and Report)^[19,20],通过这个本地类库,可以收集云计算平台中虚拟机的资源信息,包括虚拟机的基本信息如 CPU 核数、内存大小、磁盘空间大小,以及运行时的情况如 CPU 利用率、内存使用率、磁盘读写速率、网络速率等。

• 中间件数据采集服务

对于不同的中间件,需要使用不同的监控方法和技术实现性能数据的采集。例如通过 Mysql 性能查询服务获取 Mysql 服务器运行的线程数、TPS(每秒事务量)、QPS(每秒 Query 量)等各种状态值,通过 Tomcat 性能监控服务获取 Tomcat 的请求数、网络流量、线程数、平均处理时间等各种参数。

• 应用数据采集服务

目前,在云平台上的部分服务运行在 Java 虚拟机(JVM)上,所以 JVM 的运行状态对服务的质量十分重要,因此,使用 JVM Tool Interface(JVM TI)实现对 JVM 的运行状态和运行在 JVM 上应用状态的监控。基于 Servlet 容器的 Filter 拦截器可以对客户请求进行拦截处理,处理完成后将请求转发给下一个过滤器处理。请求在过滤链里逐个处理,直到最后一个过滤器处理完成。另一种常用的拦截技术是面向切面技术(Aspect Oriented Programming, AOP)^[21],AOP 采用动态代理技术在运行期动态生成代理类,将切面织入到代理类中实现监控,对软件系统性能的影响小。通过以上技术,可以收集应用服务的性能监控数据。

• 全局/局部数据存储服务

发布/订阅^[22]模式是指事件发布者向服务器发送事件,事件订阅者订阅感兴趣事件。Kafka^[23]是一种高吞吐量的分布式发布订阅消息系统,为处理实时数据提供一个统一、高通量、低等待的平台。各个数据采集服务监控到的性能数据通过 Kafka 发布,数据存储服务订阅相关性能数据进行存储管理。

5 案例研究

为了对本文提出的监控框架进行验证,在一个真实的云环境中进行了案例的部署和研究。云计算平台的环境由 1 台管理节点服务器、7 台计算节点服务器组成。管理节点是 DELL PowerEdge M630,它有 2 个 6 核 E5-2609v3 处理器,1.9GHz,15M 缓存,64GDDR4 内存,2 块 300G 10K 2.5` SAS 硬盘。计算节点是 DELL PowerEdge M630,它有 2 个 8 核 Xeon E5-2640 v3 处理器,2.6GHz,20M 缓存,128GDDR4 内存,2 块 300G 10K 2.5` SAS 硬盘。这些硬件设备之间通过万兆网卡相连,在这些硬件设备上安装了 CloudStackIaaS 平台。在云平台上部署了某综合减灾空间信息服务应用系统,该应用系统的目标是从空间和时间的维度可视化自然灾害的风险和损失,为各项灾害管理工作的各阶段提供直观的信息,并提供产品、技术、决策等服务,保障了防灾减灾工作的有效进行。该应用系统采用面向服务的体系架构(SOA),包含一系列具有独立功能的 Web 组件。将本文提出的监控框架应用到这个系统所在的云平台中,如图 4 所示,对运行在云计算平台中各个层次上的资源进行监控。

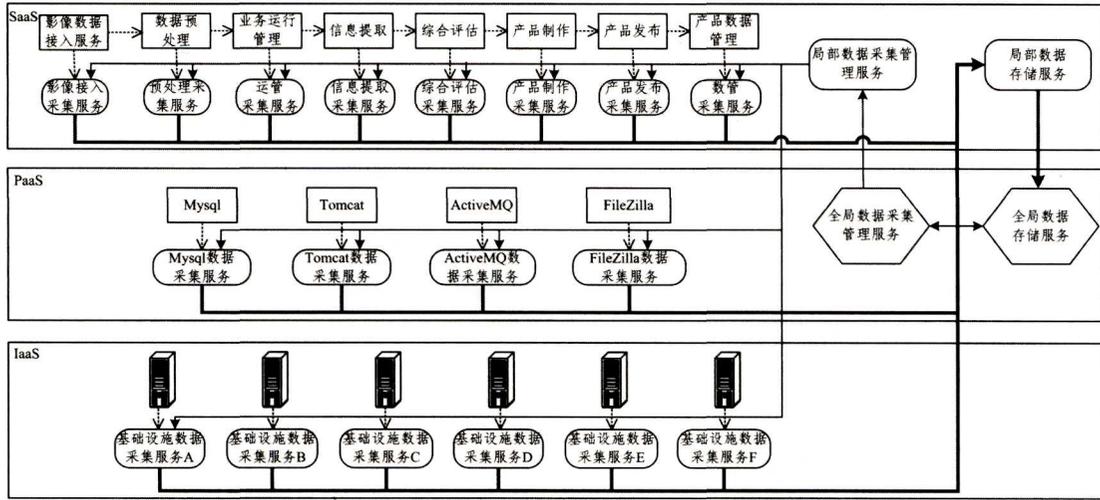


图 4 云平台监控框架应用案例

在本应用案例中,需要被监控的资源包括 IaaS 层的虚拟机节点 A—F,PaaS 层中的 Mysql, Tomcat, ActiveMQ, FileZilla 等中间件,SaaS 层中的影像数据接入服务、数据预处理服务、业务运行管理、信息提取等服务。首先按表 1 所列的内容向全局数据采集管理服务注册需要被监控资源的信息,各个虚拟机节点上的局部数据采集管理服务接收全局数据管理服务的配置信息,开启基础设施数据采集服务采集虚拟机节点 A—F 的 CPU、内存、磁盘和网络的使用情况,开启各中间件数据采集服务采集 Mysql 的线程数及 Tomcat 的 JVM 空闲内存、请求总数和线程数等,开启各个应用数据采集服务采集应用服务的响应时间、吞吐量。局部数据存储服务接收该虚拟机节点上监控的性能数据,每隔一定时间将性能数据传输至全局数据存储服务中进行统一的存储管理。

5.1 监控结果

对业务运行管理所在的虚拟机节点 A 及其上的中间件资源进行监控,实验中对业务运行管理中的某一服务 S 以 100ms 的服务平均请求间隔进行调用,执行时间戳为 2015-11-2 10:28:16,得到的监控结果如表 3 所列。

表 3 性能监控结果

Property	Name	Value	ValueType
SaaS	服务	响应时间	328.41 ms
		吞吐量	297.82 counts/s
PaaS	Mysql	线程数	16 counts
		JVM 空闲内存	316.4 MB
	Tomcat	请求总数	3000 counts
		线程数	107 counts
CPU	核数	2 counts	
	利用率	93.25 %	
内存	大小	8 GB	
	利用率	31.043 %	
IaaS	磁盘	空闲空间大小	16.3 GB
		Read	17.2 kBps
	网卡	Write	95.23 kBps
Receive		7 kbps	
	Send	19 kbps	

通过表 3 可以看出,本文提出的 SHMA 监控方法可以精确地监控 IaaS 层、PaaS 层和 SaaS 层上的资源,并获取各项性能数据,使云平台系统管理员准确掌握系统的运行状态,快速调整资源供给策略,指导负载均衡,优化系统性能。云平台消费者可以查看详细的性能指标数据,了解云平台的运行状态,查看云平台及部署的应用服务是否正常运行。

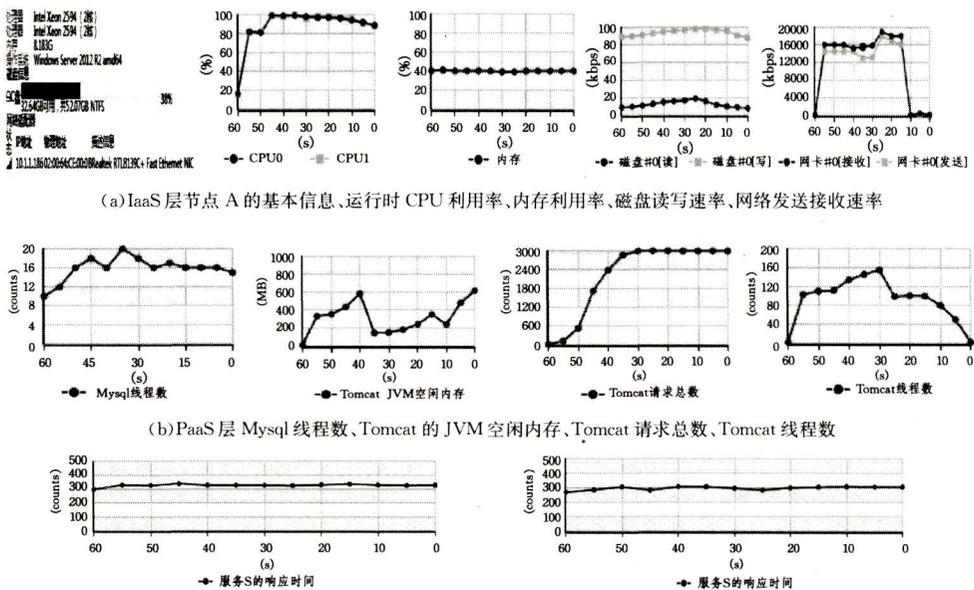
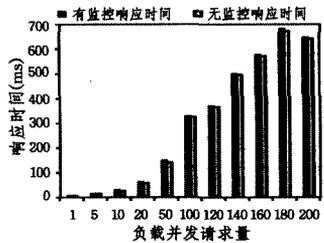


图 5 虚拟机节点 A 的性能指标

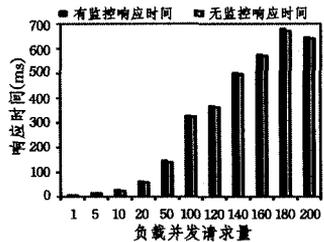
SHMA 监控框架的原型系统资源管理界面如图 5 所示。云服务的提供者和消费者通过原型系统的资源管理界面曲线可以很容易地查看系统各个节点的工作状态,对合理调整、分配系统资源,提高系统整体性能起到重要作用。

5.2 监控代价分析

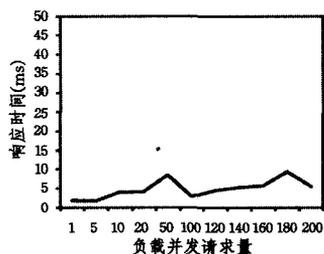
为了评价方法的有效性,对执行监控对性能本身带来的代价进行评估。在进行监控和不进行监控的情况下对虚拟机节点 A 的 CPU 利用率和服务 S 的响应时间进行对比,得出监控方法本身的执行代价。图 6 给出了服务并发负载请求从 1 到 200 变化时服务 S 的响应时间和虚拟机节点 A 的 CPU 利用率的情况。



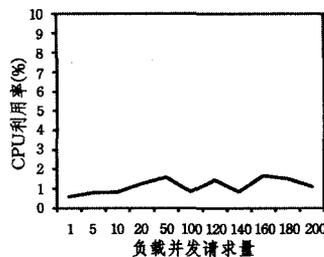
(a)有监控和无监控情况下服务响应时间



(b)有监控和无监控情况下CPU利用率



(c)监控响应时间代价



(d)监控CPU利用率代价

图 6

由图 6(a)可知,随着负载并发请求的增加,服务的响应时间也逐渐增加,当负载并发请求大于 100 时,服务的响应时间急剧增大,结合图 6(b)可知,当负载请求大于 100 时 CPU 的利用率高达 90%,说明节点 A 无法同时处理这么多的请求,是节点 A 的性能瓶颈造成了服务 S 响应时间的急剧变化。同时,由图 6(c)可以看到,监控造成的服务响应时间的性能代价在 1~10ms 时,对服务自身响应时间的影响非常小(1%以下),在图 6(d)中,监控占用的 CPU 在 1%左右。由以

上可知,监控服务本身的性能代价非常小,说明本文的监控方法能对云计算平台的性能进行有效的监控。

结束语 本文提出了一种面向云平台的可扩展多层次监控框架 SHMA,解决了云服务平台中 IaaS 层、PaaS 层和 SaaS 层中被监控资源的动态性、多样性以及资源规模巨大的问题。SHMA 采用微服务架构实现监控系统中可扩展的相互独立的服务组件,通过对部署并运行在 CloudStack 云平台上的某综合减灾空间信息服务应用系统进行案例研究,验证了框架的有效性,同时框架本身的执行代价较小。在本文中,对于局部数据存储服务中的性能数据推送给全局数据存储服务进行存储这个问题,本文采用的策略十分简单,在接下来的研究工作中,将进一步考虑如何提高传输数据的有效性,避免无效数据传输造成监控的额外负载。

参考文献

- [1] MELL P, GRANCE T. The NIST definition of cloud computing [J]. National Institute of Standards and Technology, 2009, 53 (6): 50.
- [2] NAMIOT D, SNEPS-SNEPPE M. On micro-services architecture [J]. International Journal of Open Information Technologies, 2014, 2(9): 24-27.
- [3] THONES J. Microservices [J]. Software, IEEE, 2015, 32(1): 116-116.
- [4] POVEDANO-MOLINA J, LOPEZ-VEGA J M, LOPEZ-SOLER J M, et al. DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds [J]. Future Generation Computer Systems, 2013, 29(8): 2041-2056.
- [5] ANDREOZZI S, DE BORTOLI N, FANTINEL S, et al. Grid-ICE: a monitoring service for Grid systems [J]. Future Generation Computer Systems, 2005, 21(4): 559-571.
- [6] HUANG H, WANG L. P&P: A Combined Push-Pull Model for Resource Monitoring in Cloud Computing Environment [C] // 2010 IEEE 3rd International Conference on Cloud Computing. IEEE Computer Society, 2010: 260-267.
- [7] HAN H, KIM S, JUNG H, et al. A RESTful approach to the management of cloud infrastructure [C] // IEEE International Conference on Cloud Computing, 2009 (CLOUD'09). IEEE, 2009: 139-142.
- [8] RAK M, VENTICINQUE S, MÁHR T, et al. Cloud application monitoring: The mOSAIC approach [C] // 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom). IEEE, 2011: 758-763.
- [9] KATSAROS G, KOUSIOURIS G, GOGOUVITIS S V, et al. A Self-adaptive hierarchical monitoring mechanism for Clouds [J]. Journal of Systems and Software, 2012, 85(5): 1029-1041.
- [10] KATSAROS G, SUBIRATS J, FITÓ J O, et al. A service framework for energy-aware monitoring and VM management in Clouds [J]. Future Generation Computer Systems, 2013, 29(8): 2077-2091.
- [11] HOU Yong-dong, WANG Jing, ZHAO Zhuo-feng, et al. A scalable and Integrated Cloud Platform Monitoring Mechanism [J]. Journal of Chinese Computer Systems, 2014, 35(9): 2024-2028. (in Chinese)

参考文献

- [1] AKBARALLY H, KLEEMAN L. 3D robot sensing from sonar and vision[C]// 1996 IEEE International Conference on Robotics and Automation, 1996. IEEE, 1996: 686-691.
- [2] PIERACCINI M, LUZI G, MECATTI D, et al. A microwave radar technique for dynamic testing of large structures[J]. IEEE Transactions on Microwave Theory Techniques, 2003, 51(5): 1603-1609.
- [3] RAJAGOPALAN A N, CHAUDHURI S. Space-Variant Approaches to Recovery of Depth from Defocused Images[J]. Computer Vision & Image Understanding, 1997, 68(3): 309-329.
- [4] SUBBARAO M. Parallel Depth Recovery By Changing Camera Parameters[C]// Second International Conference on Computer Vision. IEEE, 1988: 149-155.
- [5] OLIENSIS J, GENC Y. Fast and Accurate Algorithms for Projective Multi-Image Structure from Motion[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2001, 23(6): 546-559.
- [6] NG A Y, SUN M, SAXENA A. Make3D: Learning 3-d scene structure from a single still image[C]// PAMI. 2008: 824-840.
- [7] PENTLAND P. A New Sense for Depth of Field [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1987, 9(4): 523-531.
- [8] SUBBARAO M, GURMOORTH N. Depth recovery from blurred edges[M]. IEEE, 1988.
- [9] NAMBOODIRI V P, CHAUDHURI S. Recovery of relative depth from a single observation using an uncalibrated (real-aperture) camera[C]// 2013 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008: 1-6.
- [10] SU Qing-hua, ZHAO Yan, YANG Kui, et al. Calculation Method of Depth in Single Defocused Image[J]. Infrared, 2013, 34(5): 16-22. (in Chinese)
苏庆华, 赵剡, 杨奎, 等. 单幅散焦图像深度计算方法[J]. 红外, 2013, 34(5): 16-22.
- [11] ZHUO S, SIM T. Defocus map estimation from a single image. [J]. Pattern Recognition, 2011, 44(9): 1852-1858.
- [12] MING Ying, JIANG Jing-yu. Depth recovery from a single defocused image using a Cauchy-distribution-based point spread function model[J]. Journal of Image and Graphics, 2015, 20(5): 708-714. (in Chinese)
明英, 蒋晶珏. 使用柯西分布点扩散函数模型的单幅散焦图像深度恢复[J]. 中国图象图形学报, 2015, 20(5): 708-714.
- [13] ENS J, LAWRENCE P. An Investigation of Methods for Determining Depth from Focus[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1993, 15(2): 97-108.
- [14] SUBBARAO M, Choi T S, Nikzad A. Focusing techniques[M]. Optical Engineering, 1992.
- [15] LAN K T, LAN C H. Notes on the Distinction of Gaussian and Cauchy Mutations[C]// Eighth International Conference on Intelligent Systems Design and Applications. IEEE Computer Society, 2008: 272-277.
- [16] MING Ying, JIANG Jing-yu. Cauchy Distribution Based on Statistical Change Detection for Visual Surveillance[J]. Journal Of Image And Graphics, 2008, 13(2): 328-334. (in Chinese)
明英, 蒋晶珏. 视觉监视中基于柯西分布的统计变化检测[J]. 中国图象图形学报, 2008, 13(2): 328-334.
- [17] CANNY J. A Computational Approach to Edge Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1986, 8(6): 679-698.
- [18] KIMELFELD B, SENELLART P. Probabilistic XML: Models and complexity[M]// Advances in Probabilistic Databases for Uncertain Information Management. Springer Berlin Heidelberg, 2013: 39-66.
- [19] REDDY P, RAJAMANI L. Performance comparison of different operating systems in the private cloud with KVM hypervisor using SIGAR framework[C]// 2015 International Conference on Communication, Information & Computing Technology (IC-CICT). IEEE, 2015: 1-6.
- [20] REDDY P, RAJAMANI L. Virtualization overhead findings of four hypervisors in the CloudStack with SIGAR[C]// 2014 Fourth World Congress on Information and Communication Technologies (WICT). IEEE, 2014: 140-145.
- [21] MUNOZ F, BAUDRY B, DELAMARE R, et al. Usage and Testability of AOP: an empirical study of AspectJ[J]. Information and Software Technology, 2013, 55(2): 252-266.
- [22] ZHENG Xiao, LUO Jun-zhou, CAO Jiu-xin, et al. A Publish/Subscribe Based Information Dissemination Model for QoS of Web Services[J]. Journal of Computer Research and Development, 2010, 47(6): 1088-1097. (in Chinese)
郑啸, 罗军舟, 曹玖新, 等. 基于发布/订阅机制的 Web 服务 QoS 信息分发模型[J]. 计算机研究与发展, 2010, 47(6): 1088-1097.
- [23] KAFKA A. A high-throughput, distributed messaging system [OL]. <http://kafka.apache.org/design.html>.

(上接第 12 页)

侯永东, 王菁, 赵卓峰, 等. 一种可扩展集成化的云平台监控机制 [J]. 小型微型计算机系统, 2014, 35(9): 2024-2028.

[12] MASSIE M L, CHUN B N, CULLER D E. The ganglia distributed monitoring system: design, implementation, and experience [J]. Parallel Computing, 2004, 30(7): 817-840.

[13] The Ganglia Project, Ganglia monitoring service, 2012 [OL]. <http://www.ganglia.info>.

[14] Nagios Enterprises LLC, Nagios-the industry standard in IT infrastructure monitoring, 2012[OL]. <http://www.nagios.org>.

[15] <http://www.zabbix.com/>.

[16] <http://www.monitis.com/>.

[17] LIU Z H, HAMMERSCHMIDT B, MCMAHON D. JSON data management: supporting schema-less development in RDBMS [C]// Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. ACM, 2014: 1247-1258.

[18] KIMELFELD B, SENELLART P. Probabilistic XML: Models and complexity[M]// Advances in Probabilistic Databases for Uncertain Information Management. Springer Berlin Heidelberg, 2013: 39-66.