一种基于 OpenStack 的云存储方案

徐芳辰[1],沈苏彬[2]
(1. 南京邮电大学 物联网学院,江苏 南京 210003;
2. 南京邮电大学 计算机学院 软件学院,江苏 南京 210003)

摘 要:云存储作为新兴存储模式,已经被广泛应用于大规模数据的存储中。NoSQL 数据库系统面向特定应用,具有可伸缩性和适应动态需求的灵活性,已经逐步在特定的应用领域取代了关系数据库。CouchDB 作为文档型数据库,具有高可用性、灵活性等特点,适用于各种部署场景。然而 CouchDB 结构简单,较难扩展,可伸缩性较差。通过研究,实现了 CouchDB 在 OpenStack 云平台上的部署方法,利用 OpenStack 虚拟化技术,设计了保证存储资源、计算资源和网络资源的动态分配与管理的云存储方案,以解决 CouchDB 的可伸缩问题;实现了基于 OpenStack 的 CouchDB 实验原型以验证云存储方案的有效性。实验结果表明,该方案能有效满足海量数据的存储需求,提高了存储的可缩放性。

关键词:云计算;数据存储;云平台;非结构化数据库
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2015)11-0076-06
doi:10.3969/j.issn.1673-629X.2015.11.016

A Cloud Storage Solution Based on OpenStack

XU Fang-chen[1], SHEN Su-bin[2]
(1. School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;
2. School of Computer Science & Technology, School of Software, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: The cloud storage as an emerging storage model, has been widely used in large-scale data storage. NoSQL database is oriented for the specific application with horizontal scalability and flexibility to adapt to the dynamic demand, which gradually replaces the relational database in some specific application domains. CouchDB as a document database, with high availability and flexibility, can be applied for various deployment scenarios. However the CouchDB has the disadvantages of simple structure, poor scalability and extensibility. In this paper, the CouchDB deploys in OpenStack cloud platform. The improved application scheme of cloud storage is proposed, using the OpenStack virtualization technology, guaranteeing the storage, computing and network resources to be allocation and management dynamically, which solves the problem of scalability. The CouchDB storage prototype based on OpenStack is implemented. The experimental results show that the prototype can effectively meet the storage demand of mass data, improving storage scalability.

Key words: cloud computing; data storage; cloud platform; NoSQL

0 引 言

随着互联网的发展与数据量的增加,网络中的数据呈指数级增长。越来越多的企业和用户利用云存储[1]解决数据存储问题。云计算[2]是一种新型的网络技术,提供了更低消耗、更可靠和安全的服务。用户不再需要投入大量的物力,也不需要知道服务如何实现,只要接入“云”,就能实现随取随用的计算和存储。

OpenStack[3]作为云计算平台,旨在为所有类型的云提供简单可实现、大规模可伸缩和功能丰富的解决方案。OpenStack 利用标准化的接口为用户提供了虚拟机和云存储服务。

目前,存储在云中的数据量越来越多。这些数据来自不同的领域,各种不同格式的数据以指数形式增长。对于海量数据而言,传统的集中式存储方案已经无法满足现有的存储需求。无论用户还是企业,如何有效地存储和管理不断增长的数据,保持良好的存储

收稿日期:2015-02-04 **修回日期**:2015-05-05 **网络出版时间**:2015-11-04
基金项目:江苏省未来网络前瞻性研究资助项目(BY2013095-1-08)
作者简介:徐芳辰(1990-),女,硕士研究生,研究方向为计算机网络;沈苏彬,博士生导师,研究方向为计算机网络、下一代电信网及网络安全。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20151104.0948.018.html

和读取性能是迫切需求。

针对海量数据的存储和管理,一般解决方案为构建大规模廉价服务器集群,采用分布式存储技术,通过添加节点的方式动态伸缩以保证可扩展性。分布式存储技术包括分布式文件系统、分布式对象存储系统及分布式数据库。其中,NoSQL[4] 没有表结构和数据类型的约束,同时也不需要全部满足 ACID[5] 事务。NoSQL 作为现有云存储的解决方法之一,提供数据实时存储、检索和维护,满足海量数据存储的并发性、高可用性和可扩展性。较传统的关系数据库,其优势包括易于扩展、结构简单、性能高、可用性高。它去掉了数据间的关系特性,能够简单进行增加和删除,极大地节省了开发和维护成本。因此,NoSQL 能够有效地存储和管理海量非结构化数据。

CouchDB[6] 是一个 RESTful 的文档数据库,支持 JavaScript、易于管理及与 Web 交互,并可以扩展移植到移动设备上;支持分布式节点的精确同步复制。CouchDB 是面向文档的数据库。文档数据库较好地支持文档索引,确保大量数据的存储和查询性能。但 CouchDB 集群无法根据整体需求对资源进行分配和管理。OpenStack 云平台能够对云主机的资源进行有效管理。因此,文中提出将 CouchDB 集群构建在 OpenStack 云主机之上。结合 OpenStack 与 CouchDB 技术,快速有效的进行存储和资源管理是需要解决的关键问题,也是文中研究的重点。

文中首先详细介绍了云存储的概念,分析了 OpenStack 的架构和 CouchDB 数据库的特点。其次,针对当前数据存储需求,设计了一个较为完善的数据存储方案,并对每个模块进行了详细设计。然后根据设计方案实现了存储方案的实验原型。最后,对存储原型进行了测试,并对测试结果进行了分析。实验结果表明,存储原型能够较好地完成数据存储需求。

1　相关技术分析

1.1　云存储技术

云存储由成千上万的网络存储设备集群、分布式文件系统和其他存储中间件组成,为用户提供块或文件的存储服务。云存储数据一般由第三方维护,用户通过互联网与远程数据中心进行交互,不需要了解详细的存储架构和机制,也不需要携带物理存储设备。针对不同的目标有许多不同的形式,简单的小规模云存储只有几台计算机组成,大规模云存储系统可能由不同区域的存储集群组成。

文献[1]给出了一般云存储的定义,提出了一个新的分层的存储架构,并分析了云存储的关键技术。云存储在应用中的优势包括易于管理、节省成本、高可用性、简单灵活及灾难恢复。文献[7]主要关注非结构化的数据分析,介绍了非结构化数据的将数据管理和存储分开的优势。数据管理层保证了数据的有效性和完整性,灵活的数据模型可以简单实现应用部署,同时数据结构遵循 ACID 事务。

但云存储也存在一定的挑战,包括安全、数据完整性、复制时间和消耗及可靠性等。文献[8]使用 CouchDB 数据库,设计和实现了 PaaS 日志系统,并利用 MapReduce 实现有效的日志分析。文献[9]设计和实现了基于 CouchDB 数据库服务器的物联网数据存储方案。但 CouchDB 结构简单,不支持自主扩展,可伸缩性较差,应用过程中可能由于资源不足导致节点失效。

为解决海量数据的存储需求,文中提出了一种云存储方案。针对不同格式和未知格式的数据,选用 CouchDB 作为数据库,对海量数据进行存储和管理。考虑到 CouchDB 集群较难扩展,将 CouchDB 集群构建在 OpenStack 平台上,利用 OpenStack 虚拟化技术,提供虚拟资源的管理和监控。

1.2　OpenStack 云平台

OpenStack 是集中大量计算、存储和网络资源,提供大规模可扩展的云操作系统。用户通过 Web 接口控制并管理资源。其优势在于模块松耦合、虚拟化技术全面、组件配置灵活、便于二次开发[10]。OpenStack 的 Grizzly[11] 版本架构包括七个核心组件,共同合作提供了可伸缩的云主机管理功能:

(1)计算(Nova),根据需求提供虚拟服务器。

(2)身份认证(Keystone),提供 OpenStack 身份验证和授权服务。

(3)网络(Quantum),提供网络连接服务。

(4)对象存储(Swift),提供对象存储,允许存储或检索文件。

(5)块存储(Cinder),提供持续的数据块存储。

(6)镜像(Glance),提供虚拟磁盘镜像目录和资源库。

(7)控制面板(Horizon),为所有用户提供基于 Web 的用户接口。

文中的云存储方案构建在 OpenStack 云平台之上,利用其虚拟化技术解决资源控制和管理问题。该功能主要通过计算组件 Nova 实现,通过虚拟化技术,将计算能力通过云主机的方式交付。利用 OpenStack 模块化分明的特性,设计并构建了一个具有良好可扩展性的 OpenStack 架构。

主控节点控制和管理整个系统,计算节点为用户提供云主机和存储空间,通过添加计算节点为云平台添加物理资源。

1.3 CouchDB 数据库

CouchDB 是基于 JSON 与 REST 的面向文档的分布式数据库,存储 JSON 格式的文档。从大型数据中心到智能手机,CouchDB 支持各种场景的部署。CouchDB 利用 JavaScript 和 MapReduce[12] 将进行查询和索引,在数据存储和处理方面有很好的应用前景。

CouchDB 是 NoSQL 的典型代表,遵循 CAP 理论[13] 和 BASE 模型[6],满足可用性和分区容错性。CouchDB 有以下几个特点:

(1)只支持追加的存储模式,保证了数据不会损坏,易于复制、备份和恢复,支持离线存储,灵活性较高;

(2)存储系统分布在多台物理节点上,并保证了数据读写一致性,具有良好的并发性;

(3)面向文档的存储结构,每个文档均由唯一的 ID 标识,适合存储半结构化数据和非结构化数据,具有良好的可扩展性;

(4)用户自定义视图,通过视图查询、汇总和统计数据,支持 JavaScript 和 MapReduce 的索引;

(5)提供基于 RESTful API 访问,使用方便;

(6)通过复制实现备份机制,保证数据的可靠性。

CouchDB 文档数据以 JSON 格式的文档存储在数据库中,每个文档通过唯一 ID 和版本号标识。因此,CouchDB 可以在同一数据库中存储任意格式的数据文档,满足不同格式的数据存储需求。然而,随着数据量源源不断的增加,单一 CouchDB 数据库无法满足海量数据的存储需求,也无法同时处理大量数据。文中构建 CouchDB 数据库集群,通过增加数据库节点实现集群扩展,同时提高了数据并行处理的性能。

2 云存储的总体方案

在不同的场景中,针对数据存储的不同需求,数据的格式也有所不同。文中假定的场景是在物联网中,源源不断的传感器感知数据如 RFID[14],需要一个具有良好的可扩展性和灵活性的存储方案。此方案同样适用于海量不同格式数据的并行处理。

OpenStack 云平台作为基础设施,能够较好地利用现有硬件资源,根据需求对资源进行分配和调度,满足可扩展性的需求。CouchDB 集群通过复制机制同步数据,满足一致性需求,并行处理海量数据的需求。

为了满足存储可伸缩性,文中选择在 OpenStack 平台上构建 CouchDB 数据库,利用 OpenStack 虚拟化技术构建良好的数据存储平台,设计并实现了一个可靠的、灵活的、可扩展的数据存储原型。该原型的设计目标有:

(1)大规模可扩展和伸缩。

原型支持 TB 级数据存储容量,且易于扩展,具有良好的可伸缩性。为了满足数据源源不断实时写入,原型需满足自主扩展,但同时不能影响其他服务持续稳定的运行。

(2)高可靠性和可用性。

在实际应用中,可能发生硬件故障、系统崩溃、网络断开等异常,非常容易导致数据丢失。因此,需要设计一些容错功能保证异常得到处理,数据能够恢复。

(3)易于管理和维护。

单一传感器数据的容量非常小,可能只有几 kB,但单位时间内文件增加的数量可能是海量的。因此,要求存储原型能够对这些数据快速的进行管理和检索,易于维护。

根据实际应用需求,结合当前云计算的关键技术,文中提出了可行的云存储方案。存储方案原型总体结构图如图 1 所示。

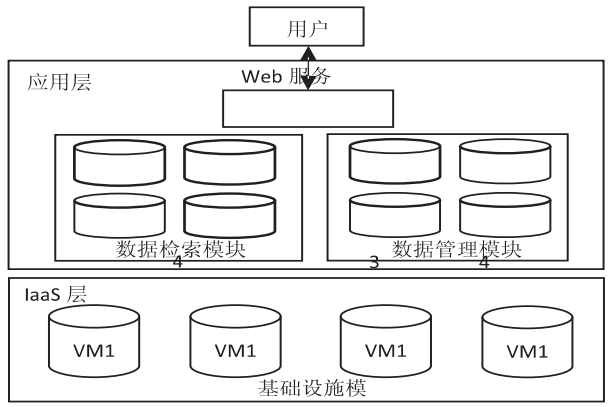

图 1 原型体系结构图

利用 OpenStack 云平台将硬件资源封装在独立的云主机中,通过调节云主机个数和磁盘空间,动态加载存储以满足可缩放性的需求。CouchDB 集群建立在 OpenStack 的云主机之上,实现数据存储和数据检索功能。用户通过可视化的用户界面与 CouchDB 进行交互,根据用户选择完成相应操作。

3 云存储的实现方案

采用模块化设计方法可以将云存储方案的实现分为四个模块:基础设施模块、数据存储模块、数据检索模块和用户界面模块。

基础设施模块包括云主机管理、网络管理、镜像管理等。高效自主管理满足可扩展性和可靠性的需求。

根据 OpenStack 模块松耦合的特点,文中使用多节点的物理结构。一台物理主机作为主控节点,其他物理主机作为计算节点,具体结构设计如图 2 所示。若对平台物理资源进行扩展,只需增加计算节点。

主控节点管理和监控整个 OpenStack 云平台,提供了用户界面、RESTful API、网络管理、安全认证等功

能,并不参与云主机、存储设备及网络的实际操作。计算节点为用户提供云主机和存储空间等资源,分别提供了云主机管理、虚拟网络管理和数据存储功能。

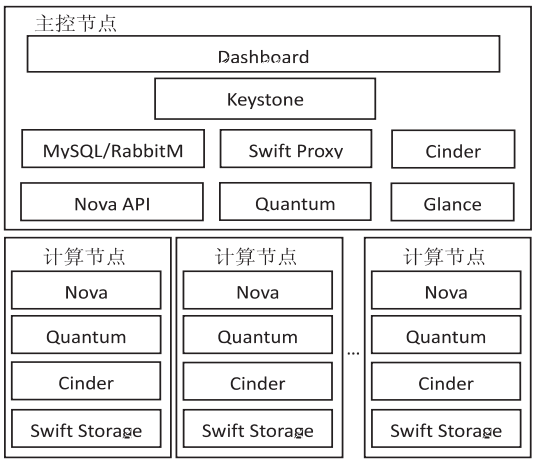

图 2　基础设施模块结构设计图

该设计结构清晰、部署简单、易于维护,满足了可扩展的需求,具有高可靠性,能够持续稳定的运行,较好地满足了实验需求。

数据存储模块利用 CouchDB 集群实现海量数据存储。其中,包括所有对数据的操作以及存储和维护等基本功能。

集群使用扁平化结构部署,节点间关系对等,并通过复制进行同步。多台服务器之间进行信息复制,可以添加负载均衡器实现负载分配,防止某一服务器失效影响系统整体性能,以满足可靠性需求。文中考虑使用分片策略实现扩展,即使用 CouchDB Lounge。

数据检索模块 CouchDB 视图实现文档检索。视图使用 JavaScript 函数定义,并保存在设计文档中,使用时通过 MapReduce 运行。通过 HTTP API 请求数据库、文档及视图,最终完成数据检索。

用户界面模块要为用户提供可视化操作。用户界面作为系统与用户交互的媒介,提供可视化的 UI,使用户能够更方便、简单地使用和操作。利用 Web 页面作为用户界面,由于 CouchDB 操作过程中需要用到 JavaScript 函数,文中选择 JSP 技术实现 Web 页面。

4　实验原型的实现

从功能角度,云存储方案的实验原型可以分为应用层和基础设施即服务(IaaS)层,如图 3 所示。

IaaS 层由一台主控节点和两台计算节点构成,主控节点主要用于 OpenStack 资源管理和控制,计算节点主要执行计算任务,提供虚拟化服务。应用层为 CouchDB 数据库集群,利用 OpenStack 平台提供的云主机构建。每一台云主机中构建一个 CouchDB 数据库,提供数据存储与检索功能,数据库之间通过过滤器和复制实现同步。

IaaS 层的核心是 OpenStack 云平台,为上层应用提供基础设施,实现计算、存储、网络等资源的管理。文中实现的实验原型包括一个主控节点和两个计算节点。若对云平台进行扩展,只需添加计算节点。

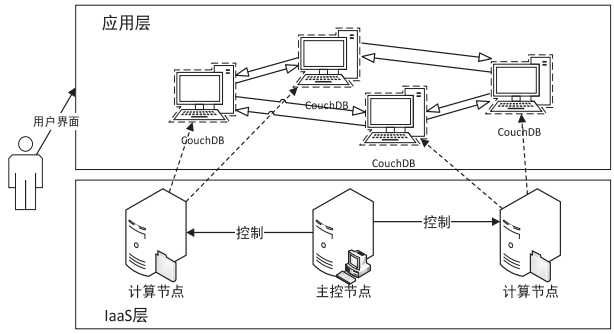

图 3　系统功能结构图

主控节点是 OpenStack 云管理平台的核心,主要通过 Nova 控制 OpenStack 其他组件,包括 Keystone、Glance、Nova、Quantum、Horizon 等。主控节点提供两个网络,一个提供 OpenStack API 访问,用于平台与外部交互,另一个提供 OpenStack 内建局域网,用于平台管理。Nova 提供了自动创建和管理云主机的功能,用于管理平台所有资源和网络。Keystone 提供了 OpenStack 的权限管理,用于用户管理和认证等。Glance 提供了虚拟机镜像的查询与存储服务。Quantum 提供了虚拟网络服务。Cinder 为稳定的数据块存储服务提供保证。Horizon 为用户和管理员提供了可视化的界面,通过界面可以完成虚拟机、网络的查看和创建。MySQL 用于存储主控节点的各组件信息。RabbitMQ 消息队列机制用于主控节点和计算节点间的通信。

计算节点的主要功能包括执行计算任务、与用户交互、提供云主机和存储空间等。计算节点组合起来是一个资源池,通过虚拟化的方式提供服务。计算节点的结构较主控节点的简单,包括 KVM、Quantum、Nova 三个模块。文中采用 KVM 虚拟化技术与 OpenVSwitch 组件实现虚拟交换机服务。计算节点的核心是 Nova,实现了云主机计算服务。

应用层的核心是 CouchDB 集群,利用 Lounge 实现。Lounge 构建于代理之上,主要包括 dumbproxy 与 smartproxy 两个组件,dumbproxy 用于处理文档 GET 和 PUT 请求,smartproxy 向集群所有分区发送请求,处理 MapReduce 视图。

集群中各主机完全对等。通过多处理机的复制机制实现同步,并在复制操作中添加过滤器,有选择地同步数据。

利用 JCouchDB[15] 实现 CouchDB 数据库的操作,包括数据库创建、删除及文档操作。JCouchDB 是 CouchDB 文档数据库的 JDBC 驱动程序,支持从动态

解析到静态的 Java POJO。CouchDB 数据库的操作接口定义如下：

(1)代码实现包含以下关键函数。

```
public Database creatDB(); //创建数据库
public int deleted();//删除数据库
public int creatDocument(String id, String name, String size);
//创建文档
public String deleteDocument(String id);    //删除文档
```

(2)定义数据实体的 Java 类如下：

```
public class Entity{
private String name;
private String info;
}
```

(3)fileStorage 类实现将数据存储到 CouchDB 集群的功能。输入一条数据信息，将其写入 CouchDB 并返回 id。定义如下：

```
public class fileStorage;
```

(4)getInfo 函数实现数据查询功能。输入查询关键字，返回所有包含该关键字的文档信息。函数定义如下：

```
public Entity getInfo(String id);
```

数据检索利用 CouchDB 自定义视图实现。通过编写 CouchDB 的 Map 函数和 Reduce 函数实现相关信息的检索。函数定义如下：

```
function(doc){    //map 函数
emit(null,doc);
}
function(key,values,rereduce){    //reduce 函数
return sum(values);
}
```

MapReduce 使用设计文档的格式存储在数据库中，调用过程如下：

```
for(Document d: result.getResult()){
System.out.println(d.getId());
}
ViewResults resultAdHoc = db.adhoc("function (doc) {
var items;
for(items in doc){
if(doc.[item].service){
emit(doc[item].service.items);
}
else{for(item in doc[item]){
return doc;
}
}
}");
```

5 测试结果及分析

文中云存储方案的实验环境为 3 台物理主机。在物理主机上构建 OpenStack 云平台。其中 2 台作为计算节点，1 台作为主控节点。利用 OpenStack 提供的云主机构建 CouchDB 集群。

物理机配置：Intel core i3，内存 16 G，磁盘 500 GB。

操作系统：Ubuntu12.04。

云系统环境：OpenStack G 版。

第三方软件：MySQL，RabbitMQ，NTP，KVM。

为了验证文中提出的云存储方案中数据存储和检索功能的正确性，进一步发现方案设计和实现的不足之处，分两个部分对实验原型进行测试：OpenStack 云平台功能测试以及 CouchDB 集群功能的测试。

(1)OpenStack 云平台功能测试。

OpenStack 云平台的基础功能包括计算资源、存储资源和网络资源的管理。

测试云主机创建功能，申请一个内存 1 G、磁盘 20 G 的云主机，查看结果；

测试网络创建功能，申请一个 IP 为 198.162.0.30 的子网，查看创建结果；

测试挂载硬盘功能，为云主机挂载 10 G 硬盘空间，查看结果。

(2)CouchDB 功能测试。

测试 CouchDB 存储功能，将本地文件写入 CouchDB 集群，在数据库中查看传入的数据格式和结果是否正确。

测试 CouchDB 检索功能，利用自定义 Map 函数查询数据库中的信息。与数据库中实际信息进行对比，查看结果是否正确。

根据以上测试方案，得出以下测试结果：

(1)OpenStack 云平台功能。

图 4 显示了 OpenStack 平台的资源概况。通过该可视化界面可以对OpenStack中所有资源进行分配和

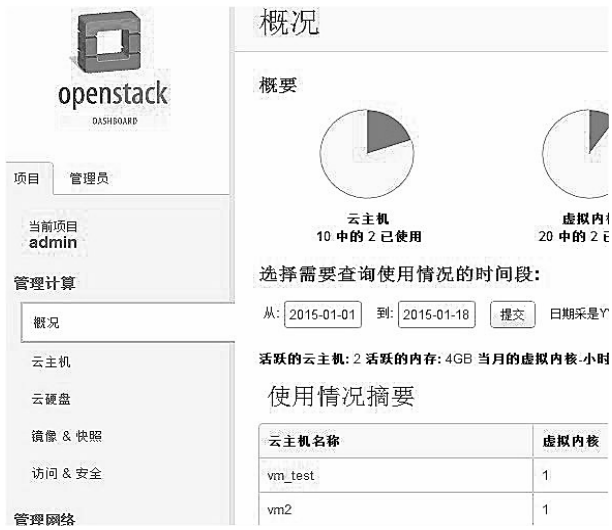

图 4 OpenStack 云平台资源概况

管理。

测试结果表明,利用 OpenStack 平台能够正确地创建云主机、创建网络以及挂载磁盘。由此得知,文中设计的 OpenStack 平台构建方案能够有效管理所有物理资源,并根据平台中云主机的实际情况进行分配。

(2)数据存储功能。

文件系统中存有多个 JSON 文档,需要将这些文档都存入 transport 表中。若上传成功,返回 ID 号;若上传失败,返回提示信息。测试所得结果如图 5 所示,图中标明的三条数据为添加成功的数据。

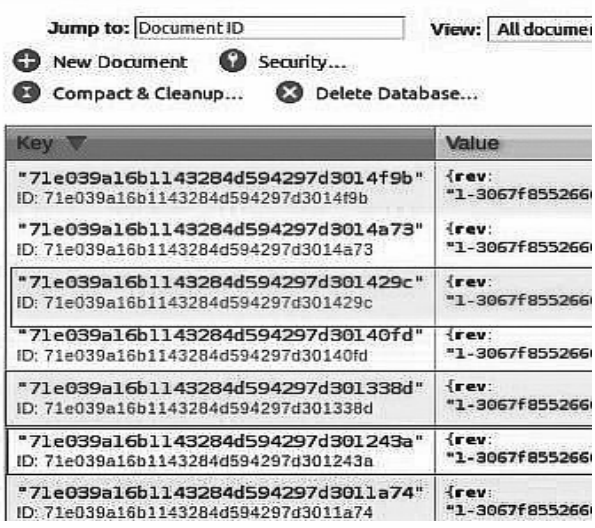

图 5　transport 数据库详情

transport 表中只显示 ID 和版本号,其他信息需要进入文档详细信息才能显示。因此,对于不同格式的数据,该原型都能按照统一的格式存储,满足了存储需求。

(3)数据检索功能。

编写自定义 Map 函数,查询数据库中所有城市名称,若查询到相应信息,则同时显示该城市的详细信息。检索结果如图 6 所示。

实验结果表明,通过该函数,查询到了无锡、温州、苏州等城市的具体信息。根据该结果,可以证明利用视图,编写自定义 Map 函数,能对所需信息进行检索,满足数据检索的需求。

6　结束语

针对海量数据存储中存在的数据格式多样化问题,文中详细分析了 OpenStack 开源云平台以及 CouchDB 的存储机制,设计并实现了一种可行的云存储方案。利用 OpenStack 平台提供资源的可扩展性,利用 CouchDB 数据库实现海量数据的并行处理以及数据检索。最后设计了测试方案,测试证明了该方案的有效性。

虽然文中主要针对源源不断的海量数据实现存储

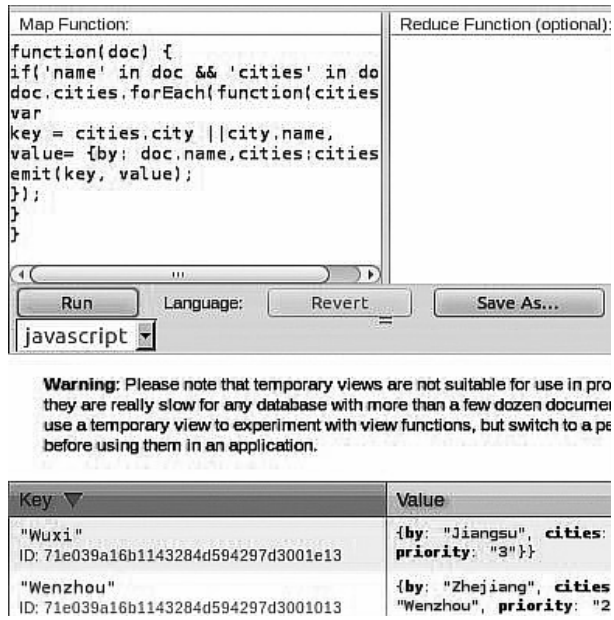

图 6　city 数据库检索结果

原型,但数据格式多样化是云存储中普遍需要解决的问题。因此,该云存储方案同样适用于其他应用场景。文中实验较为简单,只针对 CouchDB 的存储和检索功能进行了测试。该部分仍然存在一些待优化的问题,如数据检索算法等。除此之外,未来工作可以从 OpenStack 平台入手,从虚拟机调度及网络连接等方面,优化现有存储架构。

参考文献:

[1] Zeng W, Zhao Y, Ou K, et al. Research on cloud storage architecture and key technologies[C]//Proceedings of the 2nd international conference on interaction sciences: information technology, culture and human. Seoul, Korea: ACM, 2009: 1044-1048.

[2] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50-58.

[3] OpenStack[EB/OL]. 2013-04-08. http://www.openstack.org/software/grizzly/.

[4] Han J, Haihong E, Le G, et al. Survey on NoSQL database [C]//Proc of 2011 6th international conference on pervasive computing and applications. [s.l.]: IEEE, 2011: 363-366.

[5] Pritchett D. Base: an acid alternative[J]. Queue, 2008, 6(3): 48-55.

[6] Anderson J C, Lehnardt J, Slater N. CouchDB: the definitive guide[M]. [s.l.]: O'Reilly Media, Inc, 2010.

[7] Bakshi K. Considerations for big data: architecture and approach[C]//Proc of IEEE aerospace conference. [s.l.]: IEEE, 2012: 1-7.

[8] Yang C T, Liu Y T, Liu J C, et al. Implementation of a cloud IaaS with dynamic resource allocation method using OpenStack[C]//Proc of international conference on parallel and

(下转第 86 页)

6 结束语

文中通过对 SDN 网络的研究和 Openflow 协议的分析,从应用层 Flash P2P 流媒体应用出发,提出了一种基于 Openflow 的 Flash P2P 流媒体传输框架以及 tagtable 流媒体包分发路由转发技术。随着 SDN 网络的普及和研究的不断深入,文中进一步提出了未来基于 Openflow 的流媒体传输架构的流媒体发布平台以及流媒体分享平台的应用。

参考文献:

[1] 第 31 次中国互联网络发展状态统计报告[R/OL]. 2013. http://tech. hexun. com/2013/cnnic31/.

[2] Civanlar S, Parlakisik M, Tekalp A M, et al. A QoS enabled Openflow environment for scalable video streaming[C]//Proc of GLOBECOM Workshops. Miami, FL: IEEE, 2010: 351 - 356.

[3] 左青云,陈 鸣,赵广松,等. 基于 OpenFlow 的 SDN 技术研究[J]. 软件学报,2013,24(5):1078-1097.

[4] Yang L, Dantu R, Anderson T, et al. Forwarding and Control Element Separation (ForCES) framework[S/OL]. 2004. http://tools. ietf. org/html/rfc3746.

[5] Casado M, Garfinkel T, Akella A, et al. SANE: a protection architecture for enterprise networks[C]//Proc of the 15th conf on USENIX security symposium. Vancouver: USENIX Association, 2006: 137-151.

[6] Casado M, Freedman M J, Pettit J, et al. Ethane: taking control of the enterprise[C]//Proc of the SIGCOMM 2007. Kyoto: ACM Press, 2007.

[7] 梁军学,林昭文,马 严. 未来互联网试验平台[J]. 计算机学报,2013,36(7):1364-1374.

[8] Egilmez H, Gorkemli B, Tekalp A M, et al. Scalable video streaming over openflow networks: an optimization framework for QoS routing[C]//Proc of IEEE international conference on image processing. Brussels, Belgium: IEEE, 2011.

[9] 蔡进科,顾华玺,卢 冀,等. 基于 Openflow 网络的高可靠性虚拟网络映射算法[J]. 电子与信息学报,2014,36(2):396-402.

[10] Farias F, Salvatti J J, Cerqueira E, et al. A proposal management of the legacy network environment using OpenFlow control plane[C]//Proc of IEEE conf on network operations and management symposium. [s. l.]: IEEE, 2012.

[11] 管红光,杨宜镇,任万里,等. 基于 OpenFlow 的网络虚拟化技术研究应用[J]. 电信科学,2014,30(1):96-102.

[12] Karl M, Gruen J, Herfet T, et al. Multimedia optimized routing in OpenFlow networks[C]//Proc of 19th IEEE international conference on networks. [s. l.]: IEEE, 2013.

[13] Ruckert J, Blendin J, Hausheer D. RASP: using OpenFlow to push overlay streams into the underlay[C]//Proc of 2013 IEEE thirteenth international conference on peer - to - peer computing. [s. l.]: IEEE, 2013.

[14] McKeown N, Anderson T, Balakrishnan H. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(4): 69-74.

[15] 黄 韬,张 丽,张云勇,等. 基于 OpenFlow 的 SVC 流媒体时延自适应分级传输方法[J]. 通信学报,2013,34(11):121-128.

(上接第 81 页)

distributed computing, applications and technologies. [s. l.]: IEEE, 2013: 71-78.

[9] Liu Zhihan, Wang Yuhua, Lin Rongheng. A novel development and analysis solution to PaaS log by using CouchDB[C]// Proc of 2012 3rd IEEE international conference on network infrastructure and digital content. [s. l.]: IEEE, 2012: 251 - 255.

[10] Wen X, Gu G, Li Q, et al. Comparison of open-source cloud management platforms: OpenStack and OpenNebula[C]// Proc of 9th international conference on fuzzy systems and knowledge discovery. [s. l.]: IEEE, 2012: 2457-2461.

[11] OpenStack Grizzly[EB/OL]. 2013 - 10 - 17. http://www. openstack. org/software/grizzly/.

[12] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.

[13] Brewer E. Pushing the CAP: strategies for consistency and availability[J]. Computer, 2012, 45(2): 23-29.

[14] Nambiar A N. RFID technology: a review of its applications[C]//Proceedings of the world congress on engineering and computer science. [s. l.]: [s. n.], 2009: 20-22.

[15] JCouchDB[EB/OL]. 2012 - 08 - 06. http://code. google. com/p/jcouchdb/.

一种基于OpenStack的云存储方案

作者： 徐芳辰， 沈苏彬， XU Fang-chen， SHEN Su-bin
作者单位： 徐芳辰,XU Fang-chen(南京邮电大学 物联网学院,江苏 南京,210003)， 沈苏彬,SHEN Su-bin(南京邮电大学 计算机学院 软件学院,江苏 南京,210003)

刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2015,25(11)