基于多云服务器的企业网盘设计与实现

王和康[1,2] 王 洋[1] 王锦鹏[1,2] 母宝红[3] 须成忠[1]

[1](中国科学院深圳先进技术研究院云计算技术研究中心 深圳 518055)
[2](中国科学技术大学 合肥 230026)
[3](深圳市瑞驰信息技术有限公司 深圳 518071)

摘 要 针对现有企业网盘存在的安全隐患、传输性能较差、可靠性不高、运营商锁定等问题。该文从网盘存储的机密性、可靠性和访问效率等方面，设计和实现一种基于多云服务器的安全企业网盘系统——SkyDisk，实现了数据的自主可控、高速存取和安全可靠。其中，基于 Tahoe-LAFS 系统将多个云服务器整合成分布式存储集群，为网盘系统提供后端存储服务；文件在存储之前采用 256 位高级加密标准加密，保证数据的机密性；通过纠删编码和分散存储保证数据的可靠性；本地网盘服务器与多个云服务器之间并行传输数据，实现了高速上传和下载。最终，SkyDisk 实现一个 Web 服务，向用户提供 Web 方式的网盘系统。系统测试结果表明，SkyDisk 能够实现安全、可靠的文件存储管理，多云服务器存储集群没有单节点故障。同时，能够满足快速上传、下载和便捷的文件分享等功能性需求，降低了企业文件管理成本、提高了生产效率和企业竞争力。

关键词 云网盘；云存储；多云平台；Tahoe-LAFS；分布式存储系统；纠删码；SkyDisk
中图分类号 TP 302 **文献标志码** A doi: 10.12146/j.issn.2095-3135.20180907001

Design and Implementation of Enterprise Net-Disk Based on Multi-Cloud Servers

WANG Hekang[1,2] WANG Yang[1] WANG Jinpeng[1,2] MU Baohong[3] XU Chengzhong[1]

[1](*Cloud Computing Technology Research Center*, *Shenzhen Institutes of Advanced Technology*, *Chinese Academy of Sciences*, *Shenzhen* 518055, *China*)
[2](*University of Science and Technology of China*, *Hefei* 230026, *China*)
[3](*Shenzhen Virtual Clusters Information Technology Co.*, *Ltd.*, *Shenzhen* 518071, *China*)

Abstract There are several problems such as potential security issues, poor transmission performance, low reliability, and heavily dependent on service providers in existing enterprise net-disk based on public cloud. To address these problems, this paper aimes designs a secure enterprise net-disk system, named SkyDisk, based on

收稿日期：2018-09-07 **修回日期**：2018-12-11
基金项目：中科院先进技术研究院-深圳市瑞驰信息技术有限公司存储新技术联合实验室项目(Y7Z015)；深圳市海外高层次专项项目(KQJSCX20170331161854780)；中国科学院中亚生态与环境研究中心项目(RCEECA-2018-001)；国家自然科学基金面上项目(61672513)；广东省自然科学基金项目(2016A030313183)
作者简介：王和康，硕士研究生，研究方向为分布式存储；王洋(通讯作者)，博士，研究员，研究方向为云存储、云计算，E-mail：yang.wang1@siat.ac.cn；王锦鹏，硕士研究生，研究方向为软件工程；母宝红，高级工程师，研究方向为分布式存储；须成忠，博士，研究员，研究方向为并行分布式系统。

the multi-cloud servers. SkyDisk provides a full control of files, high-speed data accesses, confidentiality, and reliability for storing the files. Before stored, all files have been encrypted with 256 bit advanced encryption standard, and encoded with Reed-Solomon codes. Each cloud server saves a piece of cipher text of original files, and the files can be recovered even if some servers fail. In addition, parallel transfer of data between local net disk servers and multiple cloud servers can achieve high-speed of upload and download. Finally, SkyDisk implements a Web service, maintains file metadata and user information, and provides users with web access. The test results show that the system can achieve secure and reliable file storage management. At the same time, it can satisfy the functional requirements such as quick upload, download and, convenient file sharing, to reduce the cost of file management and improve production efficiency.

Keywords net disk; cloud storage; multi-cloud server; Tahoe-LAFS; distributed storage system; erasure code; SkyDisk

1 引 言

随着云计算技术的迅速发展，云计算已经从概念走向了实际应用，云服务器、云存储、云数据库等新服务逐渐受到重视[1]。云存储技术的发展衍生出了网盘产品，基于云存储的网盘系统，其具备存储空间的可扩展、高速的上传和下载以及便捷的分享等优点，适用于各大中小型企业和个人用户的数据资料存储、备份和归档，成为企业解决文件备份和管理的利器。在“互联网+”[2-3]驱动传统企业转型，实现互联化和信息化的大背景下，企业数据存储、共享的需求持续增长，企业网盘刚好契合了文件服务器、文件传输协议(FTP)、传统文档管理系统的替换需求，它的共享协作也符合企业信息化建设的总体方向。云存储技术的不断发展和成熟，支撑了企业网盘的落地，目前企业网盘正处于发展的黄金时期。

网盘技术经过多年发展已经基本成熟，越来越多的中小型企业已经或准备将服务迁移到云服务上，但仍不时有一些问题暴露出来，使企业用户们的不信任度增加。主要体现为以下 4 点：

(1)数据机密性问题。公有云网盘的存储层对企业用户是完全透明的，文件一旦上传到网盘中，企业就丧失了对数据的绝对控制权，而网盘服务提供商却能完全掌握这些文件，即使网盘服务商能够被信任，但恶意的内部员工或外来入侵人员都可能对企业文件造成危害。此外，文件的传输方式和存储方式直接影响数据机密性，而数据机密性是阻碍企业网盘发展和推广的主要阻力之一。

(2)服务不可靠问题。公有云网盘的解决方案、基础设施和运营维护全部由服务商提供，用户通过网络获取服务。但由于广域网网络不稳定，易造成获取服务异常，间接使得服务不可靠。

(3)性能问题。网盘系统的传输性能直接影响用户的体验——受广域网波动影响，不同地方同一时间或不同时间同一地方的上传下载带宽差异巨大，高峰期可能仅仅几 KB/s，而空闲期可能达到几 MB/s，性能不稳定直接影响用户满意度。

(4)运营商锁定问题。当企业购买某服务商提供的企业网盘服务后，随着越来越多的数据储存在该网盘上后，企业对这个网盘的依赖性就越大，也就越难更换到其他的网盘服务，这就是运营商锁定。若当前网盘出现问题或有更合适的网

盘出现时，企业很难在短时间内将大量文件从一个网盘迁移到另一个网盘中去。

尽管业内已经推出很多的网盘产品，但真正适合企业的网盘却不多。公有云网盘存在上述的 4 大问题，而私有云网盘的造价太高且运维复杂，对中小型企业不适用。因此，有必要针对现有网盘服务存在的不足，从安全性、可靠性和网络性能上进行研究，设计一种新结构的企业网盘系统。

针对上述问题，本文提出一种基于多云服务器的企业网盘系统 SkyDisk。其基本思想是将分布于广域网内的多个云服务器进行整合，建立一个跨广域网的、高容错、安全性独立于服务提供商的存储集群来存放文件内容，而用户信息、访问控制和文件元数据信息则由本地网盘服务器控制，使文件内容与控制信息相分离。这样既享受外部云服务器廉价、可扩展的存储空间，又能利用内网安全的环境保护数据的机密性，且用户对数据具有充分控制权。本地网盘服务器通过本地存储网关与多云服务器组成的存储集群进行数据传输。

SkyDisk 主要实现了以下 3 项内容，用以解决公有云网盘的 4 个问题。

(1) 存储层的框架与设计。在多个云服务器搭建安全可靠的分布式存储系统，为上层网盘应用提供存取数据的服务接口，存取数据时利用多个云服务器的网络带宽并行传输，减少公网波动带来的影响，实现高速地上传下载。存储系统在保存数据之前进行加密处理，保障数据的机密性，通过冗余存储技术保证数据的可靠性，并实现重复数据删除，节省存储空间。

(2) 访问接口层的负载均衡。当有大量文件上传下载时，单个存储网关的性能有限，压力较大，可以使用多个存储网关同时进行数据读写提高性能。多网关就会遇到负载分配的问题，通常由动态负载均衡技术来解决。负载均衡工具和调度策略多种多样，需要分析并确定最适合 SkyDisk 的负载均衡工具和调度策略，从而达到最好的负载均衡效果。

(3) SkyDisk 网盘 Web 系统的设计和实现。存储层和访问接口层为系统提供基础的数据存取服务，而文件元数据管理、业务功能和权限控制皆由 Web 服务层实现。其中，文件元数据管理包含两个方面：一是维护用户文件基本信息，如文件路径、文件大小和文件名等；二是维护文件元数据与存储系统存储的文件内容之间的映射关系，确保通过元数据能够获取到真正的文件内容。业务功能包括资料库管理、共享管理、群组管理、部门管理和员工管理等。每位用户拥有一个独立的资料库来管理自己的文件，同时通过共享管理可以将文件分享给群组或他人，达到知识共享、协同合作的目的。部门管理和员工管理方便系统管理员对公司进行人员安排和调整。

SkyDisk 的设计中，在数据存储前需要先加密，保证数据的机密性。冗余技术和分散存储保证了数据的可靠性。存储网关与多个云服务器间并行传输数据，以及多个存储网关动态负载均衡，大大提高了系统数据传输性能。多云服务器组成的存储集群无单节点故障，而云服务器可以由不同服务商提供，避免了运营商锁定困扰。

2　相关研究现状

近年来，基于云存储技术的网盘系统提供了大容量的在线存储、便捷的分享和高效率的协作，一经推出就深受人们喜爱。学术界也对网盘系统做了大量研究和分析，其中一些研究是对现有的众多网盘系统进行测试评估，如 Hu 等[4]探索了 4 种典型网盘产品的文件备份和恢复功能，简单说明了系统稳定性和数据安全性方面存在的隐患。一些研究致力于如何提升网盘系统的性能。如 Li 等[5]提出了 Update-Batched Delayed

Synchronization 机制，将大量的小文件更新请求进行合并，然后一次性同步，从而达到降低网络带宽消耗的目的，但引入了更大的延迟。还有相当一部分研究是通过多云服务构建一个安全可靠的存储，同时避免运营商锁定问题。如唐皓文[6]针对基于单云的网盘服务的固有问题，利用网盘服务提供的存取文件的应用程序编程接口（Application Programming Interface，API），实现了基于多云架构的网盘中间件。其中，中间件通过纠删编码的方式保护文件的可靠性。同时，该中间件依据不同网盘的实时网络性能动态地调度数据向多云服务的传输策略，提高了传输效率。但该研究尚未提出有效方式来保证数据的机密性。王帅[7]同样是面向多云盘的存储，通过网盘服务的 API，利用编码冗余和透明加密的机制保护数据的可靠性和机密性，但密钥和其他控制信息保存在 USB Key 中，一旦 USB Key 损坏或遗失，所有文件都将丢失。通过多云服务来解决单云网盘所带来的不足时，一般都是通过云存储服务提供存取 API 构建一个中间件服务。这也带来一些问题：首先，存储端不能运行任何代码，所有工作都由中间件承担，存在单节点故障；其次，能提供稳定云存储的服务商数量不多，纠删编码产生的数据块分散程度有限制；最后，各网盘服务提供的 API 不尽相同，给中间件开发带来困难。

尽管网盘发展日趋成熟，但仍有一些可以改进的地方。随着因特网在速度和带宽方面的显著提升，通过广域网远程存储数据在技术上变得可行，也有许多研究致力于此。Zeng 等[8]结合了纠删编码、独立磁盘冗余阵列（RAID）和写时复制技术提出了一套广域网存储模型 RSRAII（Replication-based Snapshot Redundant Array of Independent Imagefiles），在广域网环境下提供较好的存储性能、恢复效率和数据可靠性。基于多云服务融合的广域网存储可能是未来网盘发展的新方向。本文介绍的 SkyDisk 就是基于广域网存储技术而设计的网盘系统，用户文件内容存储在广域网中，而文件的元数据则存储在内网环境，实现用户文件安全可靠存储。

3 SkyDisk 的设计与实现

3.1 总体设计

SkyDisk 的总体结构由存储层、访问接口层、负载均衡层、Web 服务层、用户层组成，具体设计如图 1 所示。其中，存储层是多个不同服务商提供的云服务器组成的 Tahoe-LAFS[9]存储集群，负责存储用户上传的文件。访问接口层则是由 Tahoe-LAFS 的网关节点构成，负责文件的加解密和编解码，为上层应用提供存取文件的接口。负载均衡层则是利用 VS/DR[10-11]技术现实多网关的动态负载均衡，发挥多个网关的性能优势。Web 服务层实现一个面向用户的 Web 平台，提供网盘服务的各种功能。最后用户可以在手机或个人电脑（PC）等终端设备通过 Web 使用网盘服务。SkyDisk 结构的特点是将控制信息与文件内容进行分离，其中文件内容由外部云服务器存储，而控制信息由内部 Web 服务层处理并保存在本地数据库服务器中。这样既享受外部云服务器廉价、可扩展的存储空间，又能利用内网安全的环境保护数据的机密性。

3.2 存储层

存储层是 SkyDisk 中最基础的部分，所有数据最终均由存储层保存。系统存储层包括本地数据库和远端云服务器组成的 Tahoe-LAFS 存储集群，将管理控制数据和文件内容数据分离存储。云服务器组成的 Tahoe-LAFS 集群存储所有用户上传的文件，文件在传送到云服务器之前，在访问接口层中已经按照高级加密标准（Advanced Encryption Standard，AES）加密[12]和里德-所罗门

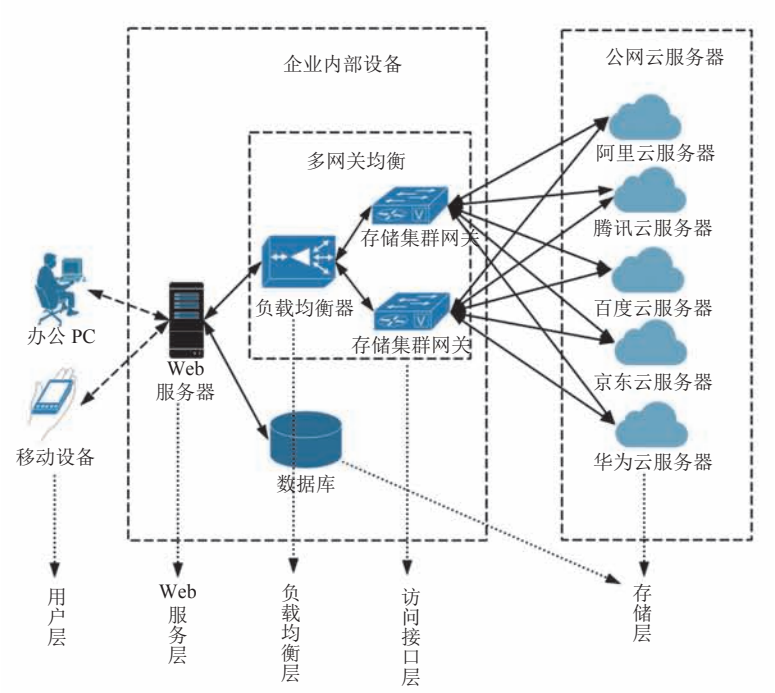

图 1　SkyDisk 系统总体结构图

Fig. 1　Architecture diagram of SkyDisk

码(Reed-Solomon Codes，RS)纠删编码[13]，各个云服务器实际存储的是文件加密后的编码块。因此，即使云提供商或入侵人员能够进入云服务器读取数据，也无法获得真实的明文数据，确保文件安全存储。另外，一个云服务器可能相对容易出现故障，而多个不同服务商的云服务器同时出故障的概率就极低，这得益于纠删编码的可恢复性[13]，存储层具有极强的容错能力。而云服务器分布在各服务商的机房中，形成异地纠删冗余存储，相对于本地纠删具有更强的容灾能力，并且存储服务不会依赖于特定服务商的云服务器，避免了运营商锁定风险。

所有有关认证、安全、权限以及文件元数据的信息存储在本地数据库 MySQL[14]中，如用户信息、部门信息、共享群组信息、访问权限、系统日志以及用户文件元数据信息。其中，文件元数据包含了文件名、目录和文件权限码(Capabilities，CAP)[15]等信息。CAP 是一个包含了加密密钥、密文哈希值、纠删比例以及文件大小等多种信息的长字符串。其中，密钥和纠删比例是恢复原文件的必要信息；哈希值和文件大小用于验证文件完整性，是从 Tahoe-LAFS 中获取文件内容的唯一凭证。因此，只有经授权的用户获取到 CAP 才能访问真正的文件内容。

3.3　访问接口层

访问接口层是指本地存储网关，它运行 Tahoe-LAFS 客户端程序，对上层应用提供存取文件的 Restful API[16]。它负责将文件存储到 Tahoe-LAFS 集群，该过程如图 2 所示：①计算原文件 SHA-256[17]值作为加密密钥；②采用 AES 加密文件得到密文；③计算出密文 SHA-256 值作为密文指纹；④按设定的纠删比编码密文，得出多个编码块 shares；⑤将密钥、密文指纹、纠删比例等信息拼接成权限码 CAP。最终 shares 会被发送到各个云服务器中存储，而 CAP 返回给上层应用管理，因此访问接口层不做任何信息的存储。

网关还负责从存储集群读取文件，该过程和

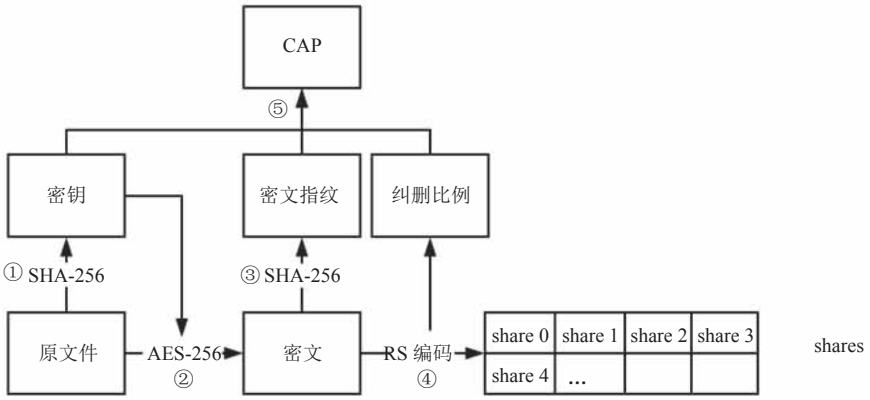

图 2 上传文件处理过程

Fig. 2 Process of uploading file

存储过程刚好相反，如图 3 所示。从多个服务器获取足够的加密文件碎片 share，解析 CAP 得到纠删比例、密文指纹、密钥等信息。根据纠删比例将碎片文件解码得出密文，计算密文的 SHA-256 值和 CAP 中包含的指纹信息对比，验证密文的完整性[17]，最后用 CAP 中包含的密钥 AES 解密得出原文件。

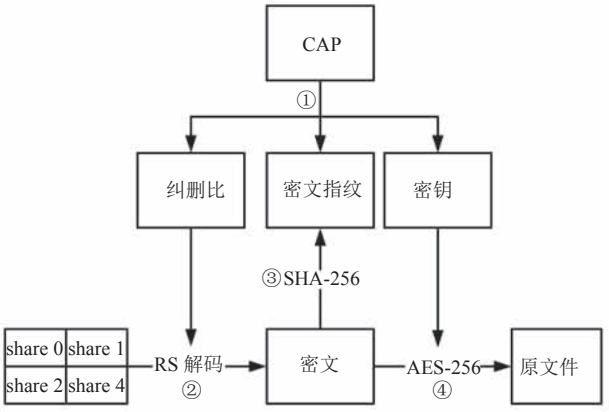

图 3 下载文件处理过程

Fig. 3 Process of downloading file

从存取文件的过程可以看出，每存储一个文件，便会产生并返回一个 CAP，通过 CAP 又可以获取真实的文件内容。访问接口层和存储层忽视了文件名和文件路径，而 CAP 是文件的唯一标识，CAP 和文件内容形成一种键值对关系，并且 CAP 包含了密钥信息。因此，对密钥的管理和对文件的管理，其实就是对 CAP 的管理。访问接口层只负责产生 CAP，管理 CAP 由 Web 服务层的用户资料库虚拟文件系统负责。从 CAP 的生成方式可以知道，内容不同的两份文件的 CAP 必然不同，而内容相同的两份文件的 CAP 必然一致。实际上，对于相同内容的文件，存储层只存储一份数据，实现了文件级别的重复数据删除，提高了存储利用率。此外，存取文件时，网关节点会有大量的计算和文件缓存，对机器的性能和配置有一定要求。

3.4 负载均衡层

本文最终采用的网关节点在存取文件时，需要进行 AES 加解密、RS 编解码和 SHA-256 哈希计算，故 CPU 和内存空间占用大。多用户同时存取文件时，网关的性能可能成为整个网盘系统的瓶颈。Tahoe-LAFS 支持多个网关访问存储集群，可以通过动态负载均衡技术[18]将用户请求压力分散到多个网关，避免出现某个网关负载很高，而其他网关负载较低的情况，提高系统存取性能，减少用户等待时长。

常用的软负载均衡工具有 LVS（Linux Virtual Server）[11]、Ngnix[19]和 Haproxy[20]，三者均为开源免费软件。其中，Ngnix 和 Haproxy 主要在应用层工作，可以保持 session、URL 定向等高级特性，但网关主要是提供存取文件的服务，更注重请求调度效率，无需复杂的负载均衡模式。

LVS 是基于 TCP/IP 的负载均衡技术，在更底层工作，因此转发效率更高，具有处理百万级并发连接请求的能力，且运行非常稳定。另外，以直接路由模式 VS/DR 工作时，数据包不经过负载均衡器直接从真实服务器发送给客户端，降低了均衡器的压力，同时延迟更低。因此，本文 SkyDisk 采用 VS/DR 技术实现多网关间的动态负载均衡。

除了负载均衡工具影响效率外，负载均衡调度策略也和效率息息相关。最简单的均衡策略是随机调度。这种方式在整体上是均衡的，但在某些时刻，可能存在部分服务器忙而部分服务器空闲，从而使得用户的请求响应慢。其他的均衡策略，如简单轮询和加权轮询[21]调度适合所有请求复杂度差不多的服务，但当有些请求处理时间很长，而有些则很短时，就会导致某些服务器一些请求还未做完就又新增了一批请求。如请求网关存储许多大小不一的文件，大文件的处理时间比小文件长太多，此时存储小文件的网关很快处理完就处于空闲状态，而存储大文件的网关还没处理完当前任务就又有新的任务，导致负载均衡效果很差。最小连接（Least-Connection）[22]是指新请求总是发给连接数最少的服务器，当有一批大小不一的文件存储请求时，存储小文件的网关在存储完后连接数就会减少，而处理大文件的网关连接一直保持，连接数没有减少，后面再来请求时就会发给存储小文件的网关，从而提高存储效率。由于多个网关节点的性能不一定完全相同，希望性能更强的服务器处理更多的请求，此时加权最小连接调度具有更好的普适性。因此，加权最小连接调度策略是最适合系统的多网关均衡调度策略。

3.5　Web 服务层

上文 3 层实现文件的安全高效存储，Web 服务层则实现了网盘所有的业务功能。该层要解决两个问题：用户文件及目录的元信息维护和用户数据的隔离。

3.5.1 Web 层功能描述

Web 服务层主要功能模块包括用户认证、资料库管理、共享群组管理、文件分享管理、员工管理和部门管理等。其中，用户认证采用常用的账号和口令形式。账号和口令存储在数据库中，为安全起见，口令以 MD5 值存储。当员工登录企业网盘时，后台计算用户输入的密码 MD5 值与数据库中的值进行比对，MD5 值相同则认证通过。系统为每位员工创建一个独立的资料库，员工可以在资料库中新建目录、上传文件、移动文件、下载文件、删除文件或目录。员工还可以建立共享群组，将有协作关系的其他员工添加为组员，并为他们设置上传、下载权限，共享组内的员工可以轻易分享文件。另外，员工可以为文件生成分享链接和提取码，其他用户可依据链接和提取码获取分享的文件。员工管理主要负责维护员工状态和信息，包括添加、修改、禁用、解禁和删除员工用户。部门管理负责维护部门信息和员工与部门的关系信息。

3.5.2 Web 层架构设计

Web 服务层采用 Django 框架[23]开发，遵循 MTV 模式。Web 服务层的架构图如图 4 所示，用户的请求由框架提供 URL 路由自行解析，交由不同的 View 处理。View 处理完用户的请求后，将结果渲染到 Template 中，以 Http 响应对象返回。但不同 View 在处理不同请求时，可能会用到相同或类似的方法，这些方法可以再抽象出来，这就是 Process Function 模块，其内封装了真正处理请求的函数和类。其中，Process Function 模块提高代码的重用率，简化了 View 的实现。当用户发来上传文件请求时，View 会调用 Proces Function 中的 Upload() 函数，此时 Upload() 先接受客户端的文件，并缓存到 Web 服务器上，然后调用 Connect_Cluster 模块提供的上传文件方法将文件存储到多云服务器组成的

Tahoe-LAFS 集群(其中，Connect_Cluster 是封装了一组访问 Tahoe-LAFS 的函数模块)，最后会构造 Model 中 File 的一个对象用来存储该文件的元数据，利用 ORM 技术将该对象持久化到本地数据库中。在 Upload 处理完成后，View 返回一个上传完成的响应。其他请求采用类似的处理方式。在本文系统中，Process Function 模块包含 Upload()、Download()、Login()等函数和 File_Operate、Dir_Operate、User_Operate 等类，是业务处理的核心模块。主要的 Model 类有 File、Directory、User、Group 等，用来保存各类信息。

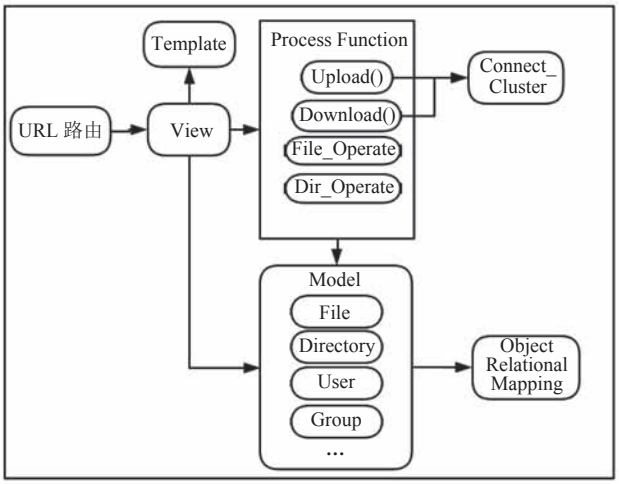

图 4 Web 服务层架构设计

Fig. 4 Architecture of Web service

3.5.3 用户资料库虚拟文件系统设计

SkyDisk 采用 Tahoe-LAFS 的键值对方式存储用户上传的文件，存储一个文件(Value)就会返回一个权限码 CAP(Key)，输入一个 CAP 就会返回文件的内容。但 Tahoe-LAFS 不会存储文件名、文件目录和 CAP 等信息，而用户资料库的文件以类似文件系统的目录树结构展示，并提供类似本地文件目录的操作，如复制、粘贴、移动和删除等。因此，需要在原本 Key-Value 存储的基础上构建一个虚拟的文件系统。SkyDisk 使用数据库表的形式存储文件目录关系、文件与 CAP 的对应关系。用一个目录表(Directory Table)保存目录的基本信息和目录层级关系。其中，目录基本信息包括目录名、目录创建时间和目录创建人，直接以表中的字段表示；目录层级关系采用自关联外键引用的方式表示：为每个目录设定一个唯一标识 DID，并用 PID 记录父目录的 DID，其中根目录的 PID 为空。这样，在查询一个目录(DID＝did)的所有子目录时，以 PID＝did 为条件查询一次目录表就行，且以 DID 建立数据库索引，通过 DID 进行查询会更快。另外，在新建子目录时也非常方便，同时也能实现无限层级，对目录深度没有限制。文件的基本信息和文件所在目录信息以及文件与 CAP 的对应关系用一个文件表(File Table)存储。文件的基本信息包括文件名、创建时间、创建人、文件大小和文件类型等，直接以表中的字段表示；文件所在目录信息以外键关联到目录表来表示，即每个文件设定一个父目录标识 PID，其值为文件父目录的 DID；文件与 CAP 是多对一的关系，文件内容相同的文件在 Tahoe-LAFS 中只存储一份数据，返回的是同一个 CAP，因此文件的 CAP 信息直接以文件表中一个字段表示即可。最终用户资料库文件系统如图 5 所示。

3.5.4 用户资料库隔离和文件分享

用户资料库是用户存储个人文件的地方，只有用户本身能够查看和操作。为保障用户文件安全，需要对用户资料库进行隔离。SkyDisk 为实现用户文件的软隔离，从 Web 服务层的虚拟文件系统上隔离用户文件，为每位用户建立一个根目录以对应其资料库，只有认证通过的用户才能访问自身资料库的根目录，从而访问资料库中的所有文件。

SkyDisk 提供两种文件分享方式：一种是分享到共享群组，共享群组是组员们共有的资料库，系统为每个群组建立一个根目录，只有组内人员才能访问该目录下的文件。当用户将文件分

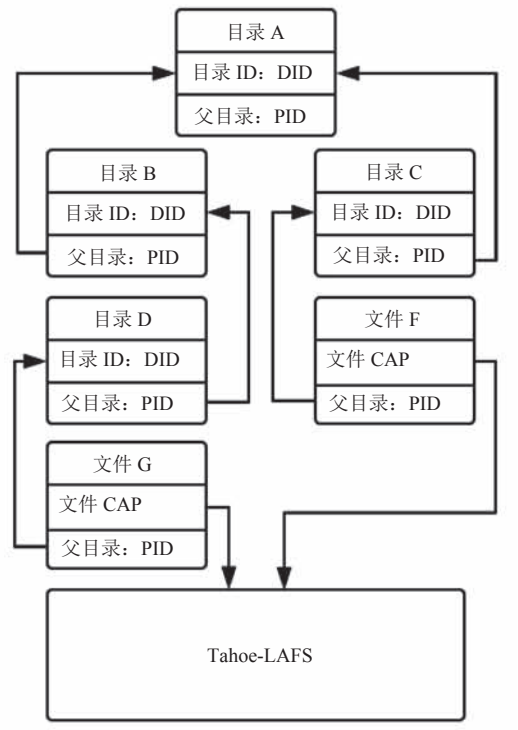

图 5　虚拟文件系统

Fig. 5　Virtual file system

享到共享群组时，只需在群组目录下建立该文件的元信息，并引用相同的文件 CAP，指向存储集群中的文件内容。第二种方式是以链接的方式分享，其关键在于产生一个唯一的链接，并指向原文件。具体地，每个文件有一个唯一标识 FID，加上分享时间和用户名，可组成一个唯一的长字符串，然后计算其哈希值作为链接并在数据库中建立一个分享表(Share Table)来存储链接与文件的对应关系，最后用户访问链接即可查询到被分享的文件的 CAP，通过 CAP 获取被分享文件的内容。获取文件的唯一方式就是通过 CAP 获取，因此两种文件分享方式的底层还是将 CAP 告知给被分享的人。

4　实　验

SkyDisk 采用 Tahoe-LAFS 集群作为后端存储，带来了许多优良特性，如数据机密性、数据容错性，但这些都要一一通过测试来验证。此外，SkyDisk 的主要功能是上传和下载文件，数据传输性能是一个重要特性，需要进行测试来验证性能是否满足需求。SkyDisk 需要运行在多个云服务器和本地服务器上，所需资源较多，本次实验的云服务器是租赁于阿里云、腾讯云和京东的低端服务器，虽然测试结果不能完全反映出系统的真实性能，但可以从中分析出部分性能的趋势。

4.1　实验环境

本小节所描述的测试环境是指测试时系统主要运行环境，其他对照实验环境的控制变量略有不同，在具体的实验中再加以说明。SkyDisk 的后端存储系统 Tahoe-LAFS 由 5 个分布于阿里云、腾讯云和京东云上的 ECS(Elastic Compute Service，弹性计算服务，即云服务器)组成，本地的存储网关节点和网盘 Web 服务节点在同一个本地物理服务器(Local Server)上，其拓扑结构如图 6 所示。限于资源不足，没有做多网关负载均衡测试。详细物理配置数据如表 1 所示，软件配置数据则如表 2 所示。

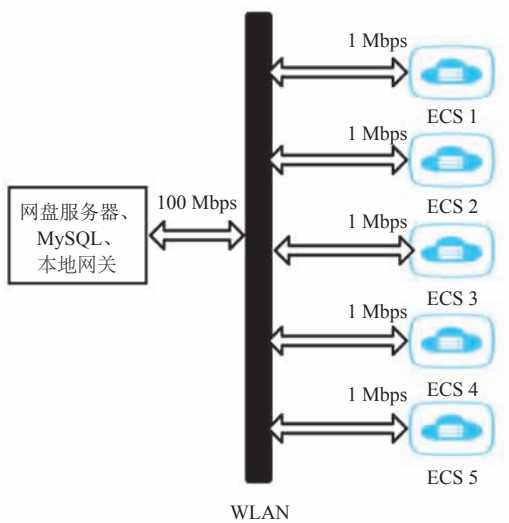

图 6　测试环境拓扑结构

Fig. 6　Topological structure of test environment

4.2　数据机密性

影响数据机密性的因素主要有两个：一个

表 1 系统节点物理配置

Table 1 Nodes' physical configuration

节点	处理器	内存（GB）	网络（Mbps）	公网 IP	所在区域	角色
ECS1	1 核	2	1	有	阿里云	存储服务节点
ECS2	1 核	2	1	有	阿里云	存储服务节点
ECS3	1 核	2	1	有	腾讯云	存储服务节点
ECS4	1 核	2	1	有	腾讯云	存储服务节点
ECS5	1 核	2	1	有	京东云	存储服务节点
本地物理服务器	I3-7100U	4	500	无	本地	存储网关、网盘服务器

表 2 系统节点软件配置

Table 2 Nodes' software configuration

节点	操作系统	Python	存储软件	Django	数据库
ECS1	CentOS 7.2	2.75	Tahoe-LAFS 1.12.1	/	/
ECS2	CentOS 7.2	2.75	Tahoe-LAFS 1.12.1	/	/
ECS3	CentOS 7.2	2.75	Tahoe-LAFS 1.12.1	/	/
ECS4	CentOS 7.2	2.75	Tahoe-LAFS 1.12.1	/	/
ECS5	CentOS 7.2	2.75	Tahoe-LAFS 1.12.1	/	/
本地物理服务器	Windows 10	2.75	Tahoe-LAFS 1.12.1	1.11.6	MySQL5.7

注：“/”表示未安装该软件

是传输过程，另一个是存储。上传时，文件经本地存储网关传输给各个云服务器；下载时，文件碎片从各个云服务器传输给本地网关，中间是经过普通广域网传输，容易被截获，如果是明文传输，将没有机密性可言。另外，云服务商内部人员或外部入侵人员可能侵入云服务器窃取数据，如果文件以明文存储，安全性也很糟糕。本次实验从传输和存储两个角度验证数据的机密性。首先，登录系统，上传 12 MB 小说文件 Book.txt 至用户资料库。上传期间采用 tcpdump 抓取网关发送给各个云服务器的 IP 数据包并解析。所截获的 50 个数据包的内容均为乱码，无任何明文信息。其次，我们也做了下载抓包实验，依然以密文传输。最后，以安全外壳协议(Secure Shell，SSH)登录到云服务器中，查看云服务器保存的分片文件可知，每个云服务器上的分片文件大小均为 4 MB，内容也全部被加密。因此，文件在传输和存储均已加密处理，机密性极高。

4.3 数据可靠性

数据可靠性是指系统能容忍一定的故障并且继续正常存取数据。SkyDisk 的后端存储采用了多云服务器构成 Tahoe-LAFS 存储集群，部分云服务器可能会突然发生异常，如文件丢失、宕机、系统损坏、网络异常、云服务商停止服务等。如果部分服务器异常，用户依然能存储文件，则说明系统具有良好的数据可靠性。随后测试在哪些极端情况下，系统不能正常工作，进而分析系统数据可靠性的极限。

实验中，模拟 3 种时期服务器故障，分别为上传之前、上传时和上传之后出现部分服务器停止服务故障。通过测试系统的上传和下载服务，并通过文件的 MD5 值验证文件的完整性，结果如表 3 所示。根据 Tahoe-LAFS 的配置参数 *K*-*H*-*N*(3-4-5)和实验结果可知：当可用服务器数量大于等于 *H*(4)时，上传就不会出错。当可用服务器数小于 *H* 时，不能保证文件的碎片分散

表 3　数据可靠性测试结果

Table 3　Data reliability test results

故障时期	操作	关闭服务器数			
		0	1	2	3
上传前	上传	正常	正常	失败	失败
	下载	正常	正常	/	/
	MD5 值是否一致	一致	一致	/	/
上传时	上传	正常	正常	失败	失败
	下载	正常	正常	/	/
	MD5 值是否一致	一致	一致	/	/
上传后	下载	正常	正常	正常	失败
	MD5 值是否一致	一致	一致	一致	/

注：下载处的“/”表示因上传失败，无法下载；MD5 值是否一致的“/”表示因上传或下载失败，导致无法计算 MD5 进行比较

存储。如果存储多片数据的服务器异常则文件读取会失败，容错能力弱。因此 SkyDisk 在过多存储服务器异常时禁止上传文件，以维持所有上传文件有较高的冗余可靠性。当剩余服务器内的分片文件数大于或等于 $K(3)$时，都能正常下载文件。这是因为在纠删解码中，任意的 K 片数据都可以恢复出原文件。因此，系统的数据可靠性主要由 K-H-N 三参数决定。在测试环境中(K-H-N＝3-4-5)，任意一个服务器异常都能成功上传文件；任意两个服务器异常，都能成功下载文件，数据可靠性比较高。

4.4　传输性能

上传和下载文件是企业网盘系统的核心功能。上传、下载文件时的传输性能直接影响到系统的用户体验。本小节通过实验分析 SkyDisk 的传输性能，测试性能时需要一个性能基准值作为参考。在测试环境中，云服务器的网络带宽仅为 1 Mbps，单个云服务器的传输带宽不会高于此值。因此，此次测试结果和其他公有云网盘或私有云网盘没有很大的可比性。本实验选用单个云服务器的传输性能作为性能基准值，测试和分析多云服务器集群系统的性能。

4.4.1 基准性能测试

性能测试实验分为基准性能测试和 SkyDisk 性能测试两部分。这里先测试以单云服务器作为存储后端时网盘系统的传输性能作为基准性能。测试时监控云服务器的各项资源。三次测试结果的平均值如表 4 所示。

表 4　单云服务器传输性能

Table 4　Single cloud server transmission performance

操作	CPU（%）	内存（MB）	网络带宽（KB/s）
上传	16	150	3 072
下载	12	150	170

其中，上传和下载过程中，CPU 和内存的使用量均不高，基本可以排除其对传输性能的影响，但传输性能差异非常大。本地上传文件到云服务器时，网络带宽达到 3 MB/s，而从云服务器下载文件到本地时只有 170 KB/s。经过验证，云服务器的网络带宽是非对称的，其标称的 1 Mbps 是其上传带宽，本地下载受制于云服务器的上传带宽，约为 1 Mbps＝128 KB/s，而实测数据约为 170 KB/s，略大于标称值。云服务器的下载带宽没有标称值，实际测得结果约为 3 MB/s。以此二值作为云服务器传输性能的基准值。

4.4.2 SkyDisk 性能测试

得出基准性能值之后，可开始测试多云服务器集群下网盘系统的传输性能。登录网盘系统，上传 4 GB 高清电影文件 Brave.mp4，使用 Nmon

工具监控 5 个云服务器以及 Web 服务器的各项资源。上传完成后，清理内存并重启服务，消除缓存影响。然后下载该文件，同样也监控每个服务器的各项资源。上传测试结果如表 5 所示，下载测试结果如表 6 所示。

在上传测试中，本地服务器将数据同时发送给 5 个云服务器，总的上传带宽为 1.5 MB/s，平均每个云服务器提供约 300 KB/s 的带宽。

在下载测试中，会有 3 个云服务器同时将数据发送给本地服务器，每个云服务器的带宽在 140～180 KB/s。

表 5 多云服务器网盘上传性能

Table 5 Upload performance of net disk base on multi-cloud servers

节点	CPU（%）	内存（MB）	网络带宽（KB/s）
ECS1	2	150	300
ECS2	2	130	280
ECS3	2	180	350
ECS4	2	150	310
ECS5	2	130	260
本地服务器	45	4 096＋4 096	1 536

表 6 多云服务器网盘下载性能

Table 6 Download performance of net disk base on multi-cloud servers

节点	CPU（%）	内存（MB）	网络带宽（KB/s）
ECS1	2	150	160
ECS2	2	130	140
ECS3	2	180	180
ECS4	2	0	0
ECS5	2	0	0
本地服务器	43	4 096＋4 096	480

注：内存的“0”表示未有额外内存消耗；网络带宽的“0”表示未向本地服务器传输流量

4.4.3 性能对比分析

上面两小节测试了基准性能值和 SkyDisk 的性能值，下面分别从上传和下载两个方面对传输性能进行对比分析。

在文件上传时，SkyDisk 会同时与 5 个云服务器进行数据传输，总的上传带宽为 1.5 MB/s，平均每个云服务器贡献 300 KB/s，仅为基准上传带宽的 1/10，基准上传带宽为 3 MB/s。这种现象与设计的目的相违背，必有其他因素制约了 SkyDisk 的上传性能。查看其他监控指标，CPU 在上传时的使用率都不超过 20%，表明 CPU 不是制约因素。而内存的使用量在基准测试和 SkyDisk 测试的差距非常大。基准测试时，内存使用量为 150 MB（共 4 GB），使用量不高，但 SkDisk 测试时，内存使用量为 4 GB＋4 GB（共 4 GB），即进程耗费了系统所有的内存资源，并且还占用了 4 GB 的虚拟内存。其中，虚拟内存是硬盘虚拟出的内存资源，其速度相对物理内存非常慢。4 GB 文件上传时，其加密、编码和指纹计算需要大量的内存资源，当内存不够时，占用虚拟内存，是制约系统上传性能的原因。但可以发现，本地服务器与 5 个云服务器间并行传输数据，理论上，如果内存足够，上传速度将优于基准性能。

在文件下载时，会有 3 个云服务器同时将数据发送给本地服务器，带宽在 140～180 KB/s，和基准性能值相当，也受云服务器的 1 Mbps 上传带宽限制。但是，整个系统的带宽为 480 KB/s，是基准性能的 3 倍，与预期的结果接近。

4.4.4 性能测试结论

测试结果分析表明，当本地服务器内存不足时，系统的上传性能约为基准性能的一半。如果本地资源足够，理论上能达到基准性能的 $N(5)$ 倍，但不会超过本地服务器的总带宽。下载文件时，主要受限于云服务器的带宽，SkyDisk 的下载性能为基准性能的 $K(3)$ 倍，不会高于本地服务器的总带宽。在单个服务器资源有限时，将 Web 服务器和存储网关分离，采用多网关负载均衡能大大提高系统整体性能。

4.4.5 优势与不足

从测试结果来看，相对于传统的基于单云

服务的网盘系统，SkyDisk 融合了加密、编码冗余、并发传输等多种技术以及多云架构的优势，保证了文件的机密性和可靠性，在一定程度上提高了传输性能，并且规避了运营商锁定风险，具有一定的优势。

另外，唐皓文[6]、王帅[7]、苏鹏等[24]等也提出了一些在多云融合方面的相关研究成果，这些文献的基本思路和本文相近，都是利用了纠删冗余编码和多云分散存储来保护用户数据的可靠性，但也有各自特色的地方。

在系统架构方面，上面 3 篇文献中的多云架构是融合多个云存储服务，通过构建网盘中间件调用各个云存储服务的 API 进行文件的上传和下载。而 SkyDisk 是通过在多个云服务器(来自多个云服务商)上搭建广域网存储系统 Tahoe-LAFS 为网盘系统提供文件存储服务。后者具有更强的可定制性。

在文件机密性方面，唐皓文[6]仅仅是在 RS 编码时采用非系统码进行文件机密性的保证，虽然非系统码编码会使得编码后的数据块错乱，但当获取到足额的数据块时就能恢复出原文了，因此仅有较低的机密性。王帅[7]对数据进行了加密处理，密钥由 USB Key 的原始密钥产生，因此具有较高的机密性，但元数据和密钥都由 USB Key 保存，系统的可用性和数据的机密性完全依赖于 USB Key。如果 USB Key 遗失或损坏，那么用户数据也随之失效，各个云盘中的数据将毫无用处。苏鹏等[24]未对文件的机密性进行讨论。SkyDisk 中的所有文件在上传之前进行 AES 加密，密钥由文件的 SHA-256 值产生，保证一文一密，密钥信息由本地产生，也存储于本地数据库中，不上传至外网中，具有较高的机密性。

在传输性能方面，各个文献中都有对其成果的性能测试说明，由于测试环境各不相同，虽具体数值没有太大的可比性，但可以从设计理论上进行分析讨论。SkyDisk 和苏鹏等[24]假设了各个云服务提供基本相同的数据传输能力，在纠删编码后将等量的数据并发分发给不同的云服务。而唐皓文[6]和王帅[7]均考虑到各个云服务的实际状态，如网络带宽、可用空间等，从而动态地调整到达各个云服务的数据量，尽可能多地利用高带宽的云服务器，理论上具有更大的吞吐量和更快的响应时间。

综上，SkyDisk 在系统架构和数据机密性方面具有一定的优势，但在传输性能方面，因未考虑各个云服务的特异性，没有做出动态调整数据分发的优化，相比唐皓文[6]、王帅[7]研究稍有不足。

5　总结与展望

本文介绍了当前企业网盘系统发展中面临的 4 个问题。分析了当前网盘存储的发展现状和研究成果，总结出多云服务融合成为解决这 4 大问题的可行方案，并提出一种基于多云服务器的企业网盘系统——将文件内容与元信息相分离，文件内容由多云服务器组成的 Tahoe-LAFS 集群存储，而获取文件内容的必要信息 CAP 由本地网盘服务器控制。本文方法既能享受外部便捷、可扩展的云服务，又能利用内网安全的环境保护数据的机密性，且企业对数据具有充分控制权。多云服务器融合避免了运营商锁定，并发传输数据提高了系统性能，从某种程度上解决或缓解了企业网盘面临的 4 大问题。

参 考 文 献

[1] Benfenatki H, Kemp G, Silva CFD, et al. Service-oriented architecture for cloud application development [C] // ISPE International Conference on Concurrent Engineering, 2014.

[2] 国务院. 国务院关于积极推进“互联网+”行动的指导意见 [DB/OL]. 2015-07-01[2018-12-10].

http://www.gov.cn/zhengce/content/2015-07/04/content_10002.htm.

[3] 吴秋萍, 黄嵩. “互联网+”促进云计算生态系统发展 [J]. 探求, 2015(3): 102-104.

[4] Hu WJ, Yang T, Matthews JN. The good, the bad and the ugly of consumer cloud storage [J]. ACM Sigops Operating Systems Review, 2010, 44(3): 110-115.

[5] Li ZH, Wilson C, Jiang ZF, et al. Efficient batched synchronization in Dropbox-like cloud storage services [C] // ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, 2013: 307-327.

[6] 唐皓文. 基于多云架构的网盘中间件关键技术研究 [D]. 武汉: 华中科技大学, 2015.

[7] 王帅. 面向多云盘的终端透明加密存储系统研究与实现 [D]. 郑州: 解放军信息工程大学, 2015.

[8] Zeng LF, Veeravalli B, Wei QS, et al. SeWDReSS: on the design of an application independent, secure, wide-area disaster recovery storage system [J]. Multimedia Tools and Applications, 2012, 58(3): 543-568.

[9] Selimi M, Freitag F. Tahoe-LAFS distributed storage service in community network clouds [C] // IEEE Fourth International Conference on Big Data and Cloud Computing, 2015, doi: 10.1109/BDCloud.2014.24.

[10] 刘玉艳. 基于 VS/DR 模式的负载均衡系统实践 [J]. 安庆师范学院学报(自然科学版), 2007, 13(2): 24-26, 38.

[11] 郑灵翔, 刘君尧, 陈辉煌. Linux 下的负载均衡集群 LVS 实现分析与测试 [J]. 厦门大学学报(自然科学版), 2002, 41(6): 726-730.

[12] Daemen J, Rijmen V. The Design of Rijndael: AES, the Advanced Encryption Standard [M]. Springer-Verlag, 2001.

[13] Jiang J, Narayanan KR. Iterative soft-input soft-output decoding of reed-solomoncodes by adapting the parity-check matrix [J]. IEEE Transactions on Information Theory, 2006, 52(8): 3746-3756.

[14] Giacomo MD. MySQL: lessons learned on a digital library [J]. IEEE Software, 2005, 22(3): 10-13.

[15] Selimi M, Freitag F, Cerdà-Alabern L, et al. Performance evaluation of a distributed storageservice in community network clouds [J]. Concurrency & Computation: Practice & Experience, 2016, 28(11): 3131-3148.

[16] Masse M. REST API Design Rulebook [M]. Sebastopol: O’reilly Media, Inc., 2011.

[17] Gilbert H, Handschuh H. Security analysis of SHA-256 and sisters [C] // International Workshop onSelected Areas in Cryptography, 2003: 175-193.

[18] Panwar R, Mallick B. Load balancing in cloud computing using dynamic load management algorithm [C] // International Conference on Green Computing and Internet of Things, 2016: 773-778.

[19] Dani R, Suryawan F. Perancangan dan pengujian load balancing dan failover Menggunakan NginX [D]. Surakarta: Universitas Muhammadiyah Surakarta, 2017.

[20] De la Cruz JEC. Design of a high availability system with HAProxy and domain name service for web services [C] // IEEE Xxiv International Conference on Electronics, Electrical Engineering and Computing, 2017, doi: 10.1109/INTERCON.2017.8079712.

[21] 黄建设. 基于 LVS 集群技术和动态反馈分组加权轮叫调度算法的 SAMC-CDN 网络系统研究 [J]. 现代电子技术, 2008(8): 141-143, 146.

[22] Kim SC, Lee Y. A study of distributing the load of the LVS clustering system based on the dynamic weight [J]. The KIPS Transactions: Part A, 2001, 8A(4): 299-310.

[23] Burch C. Django, a web framework using Python: tutorial presentation [J]. Journal of Computing Sciences in Colleges, 2010, 25(5): 154-155.

[24] 苏鹏, 刘甚灵, 张春元. 多云存储网关设计和实现 [J]. 计算机研究与发展, 2015, 52(S2): 43-48.