

# 基于多云架构进行应用编排混合部署的研究

程伟华, 王纪军, 章路进

(江苏电力信息技术有限公司, 南京 210024)

**摘要:**针对单云架构存在的技术不足,提出了多云架构的编排混合部署方案;构建出包括多云存储层、云软件平台、数据传递层和应用层的架构体系,在多云网络架构中,云软件平台包括数据层、控制层、数据接口层和应用层,并在网络设备的控制平面设置SDN控制器,通过VPC(Virtual Private Cloud 虚拟私有云)对底层网络进行计算、分析、处理、存储和共享等;并在编排器内引入自适应蚁群算法,应用基于SDN/NFV的通讯网络实现数据重构,进而实现数据网络的集中控制和智能编排;通过试验,本研究的方法数据包整体调度时延误差短,达0.15秒。

**关键词:**编排混合部署;多云网络架构;SDN控制器;编排器;自适应蚁群算法

## Research on Big Data Driven Method of Cloud Computing Smart Grid

Cheng Weihua, Wang Jijun, Zhang Lujin

(Jiangsu Electric Power Information Technology Co., Ltd., Nanjing 210024, China)

**Abstract:** Aiming at the technical shortcomings of the single cloud architecture, a hybrid deployment scheme of multi-cloud architecture was proposed. Build an architecture system that includes a multi-cloud storage layer, a cloud software platform, a data transfer layer, and an application layer. In a multi-cloud network architecture, a cloud software platform includes a data layer, a control layer, a data interface layer, and an application layer. The control plane is provided with an SDN controller, and the underlying network is calculated, analyzed, processed, stored, and shared through a VPC (virtual private cloud). An adaptive ant colony algorithm is introduced into the orchestrator, and the communication network based on SDN / NFV is used to implement data reconstruction, thereby achieving centralized control and intelligent orchestration of the data network. Through experiments, the overall delay of packet scheduling in this method is short, which up to 0.15 seconds.

**Keywords:** orchestration hybrid deployment; multi-cloud network architecture; SDN controller; orchestrator; adaptive ant colony algorithm

## 0 引言

应用编排作为 Devops 方向的研究技术之一,用户能够使用它进行配置文件的编制,也能够使用户在操作界面上,通过控件的拖拽来定义正在使用中组件之间的辅助关系,使得应用程序能够自动地部署、延伸,使得本领域的普通技术人员能够通过简易的方式实施该操作系统,大大提高了运维效率和水平<sup>[1]</sup>。随着电力系统和网络技术的飞速发展,如何管理服务配置、寻求各种环境下的配置差异以及灵活运维管理系统,则是对服务配置管理问题提出的挑战。传统技术中,编排技术的应用大多采用 Internet 网络架构实现应用编排部署,比如借助于 OpenStack 等虚拟机管理工具,管理、部署、启动这些应用编排。这种技术虽然在一定程度上具有技术进步性,但是虚拟机的运行程序比较慢,伸缩速度难以满足当前的需求,单一的云服务(例如,私有云或公有云)存在诸多的局限性<sup>[2]</sup>,比如:

1) 单一的云平台面对的服务面窄,布局呆板,不灵活,云盘能够实现的解决方案、底层基础和运维服务都通

过单一的服务商提供,严重制约了应用编排的适用范围。

2) 单云架构性能较差;单云网络系统输出性能受广域网波动的影响,不同地区的带宽差异较大,在网络运行速度在高峰期和闲余期能相差几千倍,用户满意度较差。

3) 供应商锁定性较强;单一的云平台服务使得企业或者其他用户大量的数据信息被锁定在单一的云盘上,数据转移、交互性能差。

4) 成本高,可靠性差;单一的云架构造价高,一旦出现故障或者无法正常运行,则造成整个系统处于瘫痪状态,容错性、安全性能差<sup>[3]</sup>。

## 1 整体架构设计

基于上述描述,这就需要一种新型的架构来解决上述技术问题。在本研究设计中,架构体系包括多云存储层、云软件平台、数据传递层和应用层,具体架构图如图1所示。具体而言,多云部署架构是利用至少两个云架构平台实现数据计算、存储等。多云架构能够从不同的角度和网络路径实现应用编排的混合部署,云计算编程性能强,出错率低,可靠性性能高<sup>[4]</sup>,下文将详细描述。

在本研究中,由于多云架构集群中无单节点故障,文件传递、下载速度快,因此,很有必要启动网络服务编排(network service orchestration, NSO),在多云环境下如何

收稿日期:2020-03-19; 修回日期:2020-04-27。

基金项目:江苏省科技厅重点科技项目(783400KJ521700H9)。

作者简介:程伟华(1987-),男,江西南昌人,硕士,高级工程师,主要从事云计算,企业信息化等方向的研究。

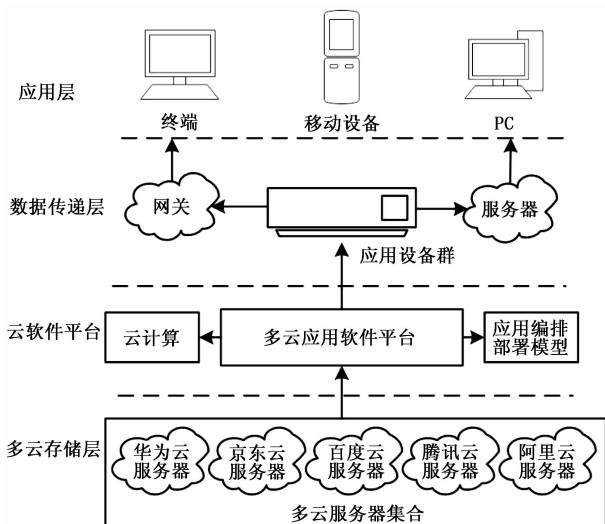


图1 整体架构设计

实现网络服务请求，是其重要的工作过程，其能请求多云架构平台中的资源规划、应用设计、部署、状态监控等管理，以实现云资源的合理应用。本研究在云网络架构中融入了 Software - defined networking (SDN)<sup>[6]</sup>，通过利用 SDN 控制器，在多云网络架构中以分层的方式实现网络数据的可扩展、可传递应用，这种方式能够提高网络数据的应用率。在云软件管理平台中，还融入了统一编排器 (Conductor)，对多云架构中的网络信息资源进行分配、编排<sup>[6]</sup>。由于不同云平台和云服务器的差异，在对不同云平台的资源进行应用时，首先抽取不同云平台的信息，将这些不同的网络信息放置在一个编排区内，以便于后期的编排。然后再将云平台上的相关资源映射到该平台上，通过生成编排模板，进一步生成堆栈，创建出云资源及其网络拓扑<sup>[7]</sup>。通过统一编排器将不同的多云平台内的资源信息集中在一个编排视图内。由于云平台不同，各云之间能够通过不同的网络（比如公网 IP、专线或 VPN 隧道等）进行信息交互。本研究应用的通讯网络为 SDN/NFV，控制能力比较集中，智能化水平比较高，进一步实现网络各层网元功能的弹性容量伸缩、自动化部署与运维等<sup>[8]</sup>。

## 2 多云网络方案设计

基于上述描述，在本研究中的多云网络架构中，其包括数据层、控制层、数据接口层和应用层。大量的网络设备设置在数据层内，SDN 控制器作为网络设备的控制平面<sup>[9]</sup>，能够有效地将网络数据从平面中分离出来，同时借助于应用软件，通过软件内设置的逻辑控制，独立地控制网络设备的运行。SDN 控制器外设有网络接口，内设有应用软件，使其通过 API (Application Programming Interface, 应用程序接口) 与应用层连接、通讯。在应用层中，用户通过应用接口实现数据接口层的信息互通。用户在使用时，通过中央控制的方式规划网络<sup>[10]</sup>。多云网络方案示意图如图 2 所示。

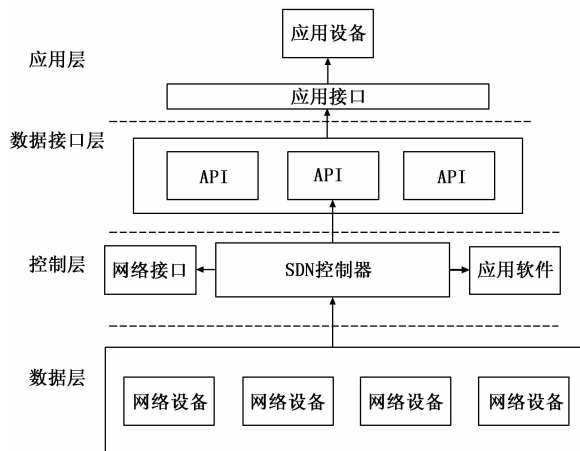


图2 多云网络方案示意图

云管理平台：

在上述的多云网络架构设计中，通过 VPC (Virtual Private Cloud 虚拟私有云) 对底层网络进行计算、分析、处理、存储和共享<sup>[10]</sup>。这种方式可以使用户对网络进行自定义设置，在其内设置各种云服务资源，比如云主机、网络负载均衡参数、存储数据库、Nosql 快存储等<sup>[11]</sup>。云管理平台架构示意图如图 3 所示。

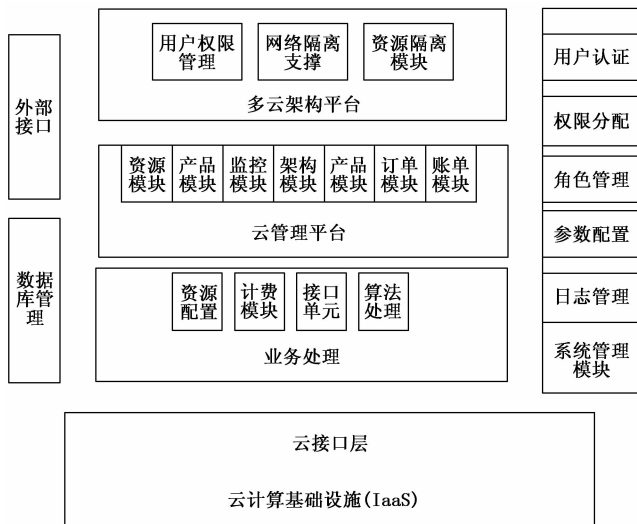


图3 云管理平台架构示意图

总体架构包含 IaaS (Infras-tructure as a Service, 云计算基础设施) 层、云计算应用基础设施层、网络数据业务处理层、网络数据应用核心云管理层、网络系统管理模块、网络数据库管理、外部接口等。在该云管理平台设计中，在云计算核心云管理层中设置有核心服务层、服务管理层和用户访问接口层。核心服务层在整个架构体系中以核心部件存在，能够使用户直接应用系统架构的硬件基础设施、软件运行环境等各种程序<sup>[12]</sup>。该层结构中聚集了多种服务类型，诸如基础设施即服务层 (IaaS)、平台即服务层 (PaaS) 和软件即服务层 (SaaS) 等。用户应用 IaaS 能够处理、保存和传递网络信息中设置的基础设施部署设备，并传递设备、个人或

者网络数据<sup>[13]</sup>。PaaS在云计算应用程序中能够部署、管理服务应用程序,并充分应用软件工具和开发语言访问底层程序,实现多种功能的信息管理。用户通过SaaS能够利用云计算基础平台,所开发的应用程序、服务管理层能够管理、控制核心服务层在运行过程中的各项活动,使得用户能够应用并操作其操作类型,用户访问接口层能够提供用户端与云系统之间的交流访问通道。通过上述软件布局,能够实现多种数据在云平台中的应用和计算<sup>[14]</sup>。

### 3 编排混合部署方法及关键技术

#### 3.1 编排混合部署方法

下面对编排混合部署方法进行描述,其具体步骤为:

1) 启动编排混合部署管理平台,配置多云服务平台架构,将不同的云平台的资源添加到相应的平台内,使得云平台内均有相应的云资源。在云平台环境中,存储设置好各种不同的云环境信息。为了更好地衡量云环境的工作情况,用户可以试图对云网络进行连接测试,当通讯成功时,则说明云环境构建成功,如果无法接收到通讯数据信息,则表示云环境创建失败。则需要重新创建,或者用户需要检测通讯失败的原因<sup>[11]</sup>。这样能够提高用户在在编排时的网络通讯性能。

2) 配置多云平台,并在多云管理平台中增加配置指令,启动平台的资源编排页面,在该页面能够展示不同云环境和资源信息。在不同云的目录下,创建不同的云资源的图标,在编排区内,对云平台内的云资源之间的网络拓扑关系进行编排<sup>[15]</sup>。本研究采用统一编排器进行编辑,在下文中,有详细描述。

3) 启动编排器内编制好的程序,对任务指令进行编排,本研究中引入了一种自适应蚁群算法,下文将详细描述。通过该算法在在平台的应用、服务页面展示应用等方面完成最佳路径的搜索,最终完成对应用及服务的编排<sup>[16]</sup>。

4) 输出编排结果,保存编排模板,对编排结果进行保存并生成编排模板;

5) 在编排模板上生成数据堆栈,创建出多云管理平台兼容的资源拓扑关系,应用编排模板生成堆栈,创建出云环境中的云资源以及资源之间的网络拓扑,并在云资源上部署容器应用。

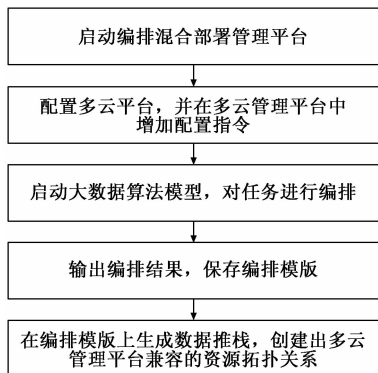


图 4 编排混合部署方法流程示意图

#### 3.2 编排部署的关键技术

在多云架构中的数据集合进行编排部署时,NSO系统在前端设备和后端设备之间的连接和通讯也很重要,应用时,将NSO系统的两端分别与IT基础设施OSS和网络基础设施连接<sup>[17]</sup>。一般来讲,IT基础设施上设置有计算单元、存储单元和网络系统等。NSO系统能够将资源请求、管理方法、数据参数等通过各种不同的数据节点发送到网络基础设施设备。在NSO系统内,所有服务请求通过数据接口、数据转换接口转换成其他设备能够接收的数据,采用的通讯网络为SDN/NFV,该网络能够实现数据的重构,能够实现数据网络的集中控制和智能编排,通讯接口兼容性较好。

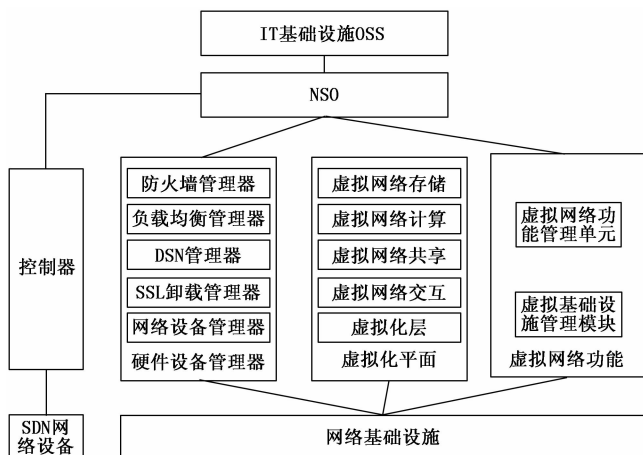


图 5 NSO设备与前端设备和后端设备之间的连接

在本研究中,还引入了统一编排器(Conductor),能够有效地控制资源编排和业务编排。其中资源编排具有网络信息资源管理、网络资源拓扑、网络映射、资源交互、资源监控和共享等功能,包含但不限于SDN域资源、NFV域资源和DC域云资源等<sup>[18]</sup>。其中的业务编排能够对资源网络业务规划过程实现可视化处理,比如业务生成可视化、部署过程自动化等一系列的活动。通过这种方式的安排,能够大大地缩短网络部署时间,提高网络资源应用率。其中统一编排器分层部署示意图如图6所示。

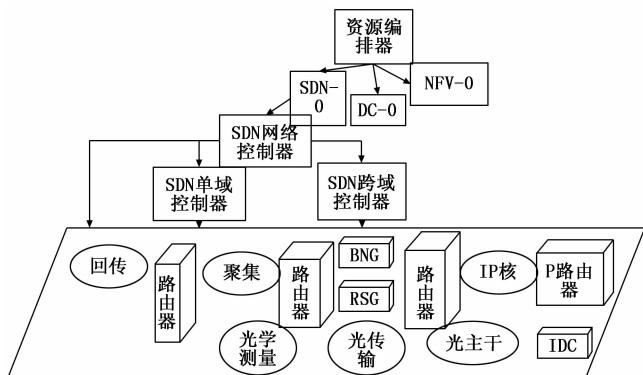


图 6 统一编排器分层部署示意图

为了在网络资源的精确匹配,在NFV资源最低的情况

下,能够保证寻找最优的云网络、云资源。在统一编排器内采用自适应蚁群算法<sup>[19]</sup>。在采用该算法时,假设 pathMatrix 表示为二维矩阵,矩阵数据为云服务内的多个云服务平台内的访问路径,假设 pathMatrix 的行数表示为  $\sum_{i \in [1, |S|]_z} |F_i|$ ,  $|M|$  表示为列数,用  $M$  表示为多云管理平台中的个数<sup>[20]</sup>。单个蚂蚁在一次循环中,在某个平台上检索 NFV,利用算法表示如下。

自适应蚁群算法多云平台通讯搜索算法:

1. 程序初始化:设置蚂蚁元素集合,寻找信息因子  $\vartheta$ ,启发因子  $\beta$  等;

2. for it Count=1;iterator Num do

3. for ant Count=1;ant Num do

4. [NFVs 的调度 \* \

5. for i=1:I do

6. for j=1:J do

7. 根据蚁群算法寻找最佳路径

8. 更新寻找路径,寻找信息因子  $\vartheta$ ,启发因子  $\beta$  等

9. end

10. end

11. end

12. [多云网络平台节点 \* \

13. //对每个云平台节点上的路径进行资源配比和分配//

14. for m=1:M do

15.  $C_{\max}(f_{ij}) = C_{total}(m) \cdot C_{pr}(f_{ij}) l \sum_{i=1}^I \sum_{j=1}^J C_{pr}(f_{ij}) \cdot x_{f_i}^m$

16. end

17. //计算每个云平台访问路径需要的最大访问时间时,需要分配的 NFV 资源//

18 for i=1:I do

19  $C_{f_i} = \min_{j \in J} \left\{ \frac{C_{\max}(f_{ij})}{\eta_{f_i}} \right\} \cdot \eta_{f_i}$

20 end

在自适应蚁群算法时,寻找信息因子  $\vartheta \in [\vartheta_{\min}, \vartheta_{\max}]$ ,启发因子  $\beta \in [\beta_{\min}, \beta_{\max}]$ ,寻找信息素浓度挥发因子  $\rho \in [\rho_{\min}, \rho_{\max}]$ 。在进行数据更新时,存在以下公式:

$$\rho(iN+1) = \begin{cases} \rho_{\min}, & \text{其它} \\ 0.95\rho(iN), & 0.95\rho(iN) \geq \rho_{\min} \end{cases} \quad (1)$$

$$\vartheta(iN+1) = \begin{cases} 1.05\vartheta(iN), & 1.05\vartheta(iN) \leq \vartheta_{\min}, \text{其它} \\ \vartheta_{\max}, & \text{其它} \end{cases} \quad (2)$$

$$\beta(iN+1) = \begin{cases} 1.1\beta(iN), & 1.1\beta(iN) \leq \beta_{\min} \\ \beta_{\max}, & \text{其它} \end{cases} \quad (3)$$

在多云平台中寻找云平台路径时,前一个云平台路径转移到后一个云平台路径的概率分布为:

$$P(k) = \frac{\zeta_{ck}^{\vartheta} * \delta_k^{\beta}}{\sum_{m \in N_{f_i}} \zeta_{cm}^{\vartheta} * \delta_m^{\beta}} \quad (4)$$

在公式(4)中,  $c$  表示进行  $f_{ij}$  的云平台路径,  $k$  表示为下一个云平台(或者云资源)访问路径。  $N_{f_i}$  表示能够进行  $f_{ij}$  云平台(或者云资源)的集合,访问信息公式可以表示为:

$$\zeta_{ck}(iN+1) = (1 - \rho\%) \zeta_{ck}(iN) + \Delta\zeta_{ck}(iN) \quad (5)$$

$$\Delta\zeta_{ck}(iN) = \sum_{s=1}^{aN} \Delta\zeta_{ck}^s(iN) \quad (6)$$

$$\Delta\zeta_{ck}^s(iN) = \begin{cases} \frac{C_{ac}(s)}{Q, (1)} \\ 0, (2) \end{cases} \quad (7)$$

在公式(7)中,第一种情况是第  $s$  只蚂蚁元素在本次路径选择中先后选择了不同云平台路径  $c, k$ 。  $C_{ac}(s)$  为第  $s$  只蚂蚁元素路径在寻求平台路径完成之后,分配资源信息的总和。  $Q$  表示为系数常数,  $iN$  表示为迭代系数,  $aN$  表示信息蚂蚁的规模大小<sup>[21]</sup>。

## 4 方案实验及分析

在试验时,云服务器主机的软件配置为:将 9 台 PC 机都安装 RedHatLinux 操作系统, Hadoop 版本为 1.0.4, JRE 环境为 1.6, 用户只要继承 MapReduce-Base, 分别实现 Map 和 Reduce 的两个类,即可编写 Map 和 Reduce 的程序<sup>[22]</sup>。其中通过 PC 机组成的 SP-DPP 平台的节点配置信息为:节点 IP 地址分别为 172.16.0.1、172.16.0.2、172.16.0.3、172.16.0.4 和 172.16.0.5,角色分别为 Master 和 Slave 两种类型,操作系统为 Linux,运行的进程分别为 Namenode, JobTracker 和 DataNode, TaskTracker<sup>[23]</sup>。然后定义参数如表 1 所示。

表 1 仿真参数表

仿真参数	仿真值
云平台数	6 个
最短周期时长	20 秒
最长周期时长	8 小时
数据包到达表示	泊松分布
数据包大小	指数分布
队列缓存大小	20MB
仿真时间	24 小时
数据训练窗口	10 个
搜索路径	3 个

为了验证本研究的计算精度,采用 MAPE 算法 (Mean Absolute Percentage Error) 对系统的准确度进行计算。计算公式为:

$$MAPE = \left( \frac{1}{N} \sum_{t=1}^N \left| \frac{O_t(t) - \xi_t(t)}{\xi_t(t)} \right| \right) \quad (8)$$

在公式(8)中,  $O_t(t)$  表示为预测值,  $\xi_t(t)$  表示为真实值,  $N$  表示为平台样本的数量值,  $MAPE$  数值越小,就说明本文研究的方案精确度越高。

在进行云平台网络部署时,先从整体端到端时延、CPU 资源利用率到网络吞吐量和数据比特率丢失率这个方面来试验。在以下方案图中,方案 1 为采用本文研究的多云架构和蚂蚁算法相结合的 CPU 资源利用率情况,方案 2 为仅仅采用本文研究的多云架构的 CPU 资源利用率情况,方案 3 为未采用多云架构和蚂蚁算法的 CPU 资源利用率情况,方案 4 为仅仅采用蚂蚁算法的 CPU 资源利用率情况。

图 7 为 CPU 资源利用率结构示意图, 该图通过试验 10 次, 利用 MAPE 算法求出最终值, 绘制而成。

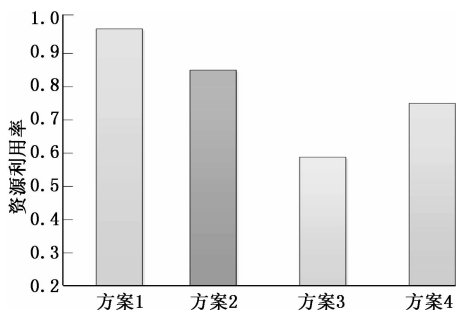


图 7 CPU 资源利用率结构示意图

通过试验, 采用本文研究的多云架构和蚂蚁算法相结合的 CPU 资源利用率较高。图 8 为数据包整体调度时延对比图。该图通过试验 10 次, 利用 MAPE 算法求出最终值, 绘制而成。

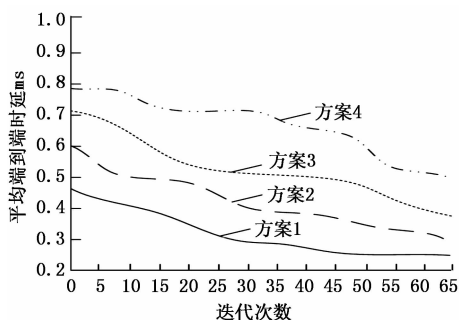


图 8 数据包整体调度时延对比图

通过图 8 可以看出, 方案 1 的数据包整体调度时延对比比较小, 误差也少。图 9 为平均端到端时延对比图, 该图通过试验 10 次, 利用 MAPE 算法求出最终值, 绘制而成。

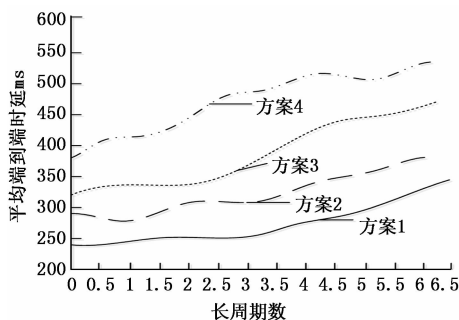


图 9 平均端到端时延对比图

通过图 9 可以看出, 方案 1 的平均端到端时延 ms 最小。图 10 为不同方案平均比特丢失率对比图, 该图通过试验 10 次, 利用 MAPE 算法求出最终值, 绘制而成。

通过图 10 可以看出, 图 1 中的方案平均比特丢失率最小。图 11 为不同方案网络平均吞吐量平均对比图, 该图通过试验 10 次, 利用 MAPE 算法求出最终值, 绘制而成。

通过图 11 可以看出, 方案 1 的网络平均吞吐量最大, 工作性能最好。多云联合概念的提出弥补了单一云存储产

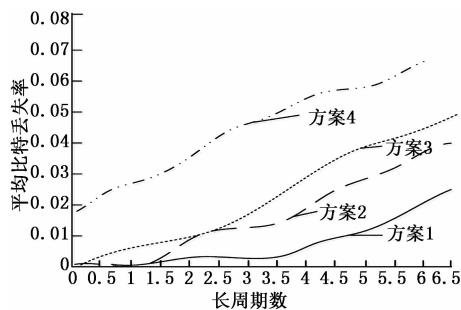


图 10 不同方案平均比特丢失率对比图

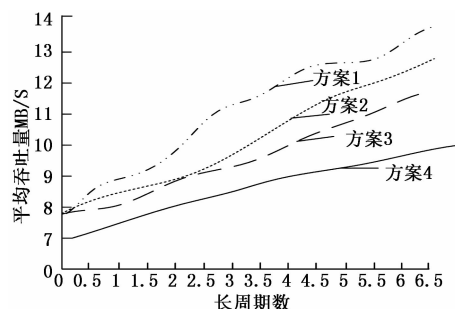


图 11 不同方案网络平均吞吐量平均对比图

品所存在的不足。通过将数据冗余地储存在多个云存储服务服务器上, 多云架构的系统能达到更高的可靠性和安全性, 并且能有效地降低运营商锁定风险。

### 5 结束语

针对单一的云平台面对的服务面窄、运行效率差以及整体性能不高等问题, 采用了多云架构部署方案, 该架构在安全性和可靠性方面具有突出的优越性能, 网络传输性能也很卓越, 基于多云架构的特点, 设计出多云网络架构管理平台, 并应用基于自适应蚁群算法实现多云平台的搜索, 实现多云架构云管理平台的搜索和优化, 提高了网络路径的搜索速度, 通过试验, 本文研究的方案在多方面均具有突出的特性, 本研究能够按照编排的流程自动执行服务请求, 自动化程度高, 无需人工干预, 减少网络云管理平台中网络服务交付的时间, 减少了人力投入成本<sup>[24]</sup>。多云部署架构顺应技术发展的需要, 符合用户的消费心理。但是, 在未来的应用过程, 也将存在诸多其他缺点, 需要在实践中逐步推进。

### 参考文献:

[1] 李佳, 徐胜超. 基于云计算的智能电网大数据处理平台 [J]. 计算机工程与设计, 2018, 39 (10): 81-87.

[2] 郝然, 艾芊, 肖斐. 基于多元大数据平台的用电行为分析构架研究 [J]. 电力自动化设备, 2017, 37 (8): 26-33.

[3] 孙庚泽. 基于 Docker 的混合云应用编排方案研究 [J]. 西安电子科技大学, 2015, 37 (3): 1-101.

[4] 房秉毅, 杨绍光. SDN 技术在混合云网络架构中的应用研究 [J]. 邮电设计技术, 2015, 2 (7): 1-6.