

基于带宽预测与自适应压缩的容器迁移方法

罗成^{1,2,3}, 崔勇⁴, 林子松^{1,2,3}

(1. 郑州大学 软件学院, 郑州 450002; 2. 郑州大学 汉威物联网研究院, 郑州 450002;

3. 郑州大学 互联网医疗与健康服务河南省协同创新中心, 郑州 450052;

4. 郑州轻工业大学 计算机与通信工程学院, 郑州 450002)

摘要: 随着数字化技术的发展与工业自动化应用范围的扩大,在万物互联环境下边缘设备数量快速增长,这些设备产生的数据量激增,导致网络带宽逐渐成为边缘计算的瓶颈。针对移动边缘服务迁移过程中传输数据量过大以及边缘节点网络环境不稳定等问题,结合带宽预测和数据压缩技术,提出一种面向 Docker 容器的服务迁移方法。通过预测网络带宽动态调整数据压缩算法的压缩速度以及压缩强度,从而充分利用网络带宽和多核处理器的计算能力,最大限度地减少网络传输的数据量以及服务的停机时间。实验结果表明,该方法对网络环境变化具有较强的适应性,能有效平衡数据传输和压缩计算的时间开销,提高服务迁移性能,相比于容器本地服务迁移、基于 Docker 基础镜像的服务迁移等方法,迁移时间、传输数据量和停机时间至少减少了 23.7%、19.4% 和 17.6%。

关键词: 边缘计算; 服务迁移; 带宽预测; 自适应压缩; Docker 容器

开放科学(资源服务)标志码(OSID):



中文引用格式: 罗成, 崔勇, 林子松. 基于带宽预测与自适应压缩的容器迁移方法[J]. 计算机工程, 2022, 48(5): 200-207, 214.

英文引用格式: LUO C, CUI Y, LIN Y S. Container migration method based on bandwidth prediction and adaptive compression[J]. Computer Engineering, 2022, 48(5): 200-207, 214.

Container Migration Method Based on Bandwidth Prediction and Adaptive Compression

LUO Cheng^{1,2,3}, CUI Yong⁴, LIN Yusong^{1,2,3}

(1. School of Software, Zhengzhou University, Zhengzhou 450002, China;

2. Hanwei IoT Institute, Zhengzhou University, Zhengzhou 450002, China;

3. Collaborative Innovation Center of Internet Medical & Healthcare in Henan, Zhengzhou University, Zhengzhou 450052, China;

4. School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China)

[Abstract] With the development of digital technology and the expansion of industrial automation applications, the number of edge devices in the Internet of Everything environment is growing rapidly, and the amount of data generated by these devices is increasing exponentially, making network bandwidth a bottleneck for edge computing. To address the large amount of data transmitted during mobile edge service migration and the nonstationary characteristics of edge node network environments, a Docker container-oriented edge service migration method is proposed based on bandwidth prediction and data compression technology. Through bandwidth prediction technology, the compression speed and strength of the data compression algorithm are adjusted dynamically to fully utilize the network bandwidth, and the power of the multicore processor is computed to minimize service downtime and the amount of data transmitted on the network. The experimental results show that this method is more adaptable to changes in the network environment and can effectively balance the time overhead of data transmission and compression calculation, thereby improving the performance of service migration. Compared with Pre-copy, LZ4-ACM, and other migration methods, this method reduces the migration time by at least 23.7%, the amount of data transferred by at least 19.4%, and downtime by at least 17.6%.

[Key words] edge computing; service migration; bandwidth prediction; adaptive compression; Docker container

DOI: 10.19678/j.issn.1000-3428.0061758

基金项目: 国家自然科学基金面上项目(81772009); 河南省科技攻关计划(212102210409); 郑州市协同创新重大专项(20XTZX06013, 20XTZX05015)。

作者简介: 罗成(1995—), 男, 硕士研究生, 主研方向为边缘计算; 崔勇, 讲师、博士; 林子松, 教授、博士。

收稿日期: 2021-05-26 **修回日期:** 2021-07-26 **E-mail:** 2425399887@qq.com

0 概述

随着第五代移动通信系统(5G)和传感器行业的发展,智能汽车、智能手机等物联网设备可通过蜂窝网、低功耗广域网等接入互联网,并且可使用传感器来感知周围环境^[1]。根据IDC预测,到2025年,全球物联网设备数量将达到416亿台,每年产生的数据量将达到79.4 ZB,其中超过半数的数据需要在边缘侧分析处理。在这种情况下,传统的云计算模式已经无法满足高时延计算的要求。边缘计算的出现有效地解决了传统云服务距离网络边缘过远的问题,实现了应用的实时响应、分析和操作,尤其是在网络条件有限的地区。通过将繁重的计算任务分流到附近的边缘服务器,可使用户体验到更高的带宽、更低的延迟和更强的计算能力^[2-3]。

边缘计算面临的主要挑战是在非平稳的网络环境下,满足用户移动时的服务质量(Quality of Service, QoS)要求^[4]。用户的移动往往伴随着服务迁移,由于服务迁移需要在网络上传输大量数据,因此导致网络和计算开销的相应增加,当传输的数据量过大时,会导致服务的停机时间过长,服务质量将会显著下降^[5-6]。此外,如果服务频繁迁移,会对边缘网络造成更多负面影响,恶化边缘网络,严重影响用户的服务体验。因此,高效的服务迁移对于实现边缘计算环境中服务的移动性至关重要^[7-8]。通过对国内外服务迁移文献的研究发现,Docker容器因轻量级和具有保持一定程度隔离的能力在边缘计算服务迁移领域得到广泛应用^[9]。文献[10-11]提出一种容器本地服务迁移方法(Pre-copy),该方法通过考虑容器间距离、可用带宽和成本,提高了资源利用率,缩短了总迁移时间,但服务迁移过程中传输的数据量过大,导致服务的停机时间过长,降低了服务质量。文献[12]提出Teddybear系统,该系统能够在网络带宽基础设施较差的地区为移动用户继续提供超低延迟的服务,并在其更改位置时将服务停机时间降至最低,但由于边缘网络的不稳定性,无法提前根据网络状况进行服务迁移,经常导致服务丢失,甚至使得服务无限期停机,影响移动用户的服务体验。文献[13]提出一种带宽分片机制,该机制可以动态有效地提供迁移带宽和非迁移带宽,以满足不同服务的延迟要求。该机制仅通过带宽分片来保证服务迁移的正常运行,但服务迁移过程中传输的数据量问题并未解决,导致服务的迁移时间过长,影响服务迁移效率。文献[14]提出一种基于Docker基础镜像的服务迁移方法(LZ4-ACM),并设计一种适用于自适应网络环境的压缩算法,对镜像等数据进行压缩,但压缩算法单一,数据压缩不彻底,并且未考虑压缩计算的时间开销,使得服务迁移时间过长,降低了服务质量。

针对上述问题,本文提出一种基于带宽预测和自适应压缩的容器迁移方法(BP-ACM)。利用

CRIU技术为服务生成检查点,并对检查点数据应用压缩去重算法。考虑到网络带宽和计算开销之间的折衷,自适应压缩算法能够根据网络带宽预测结果动态调整压缩速率以及压缩强度,从而充分利用网络带宽和处理器的计算效率。通过结合带宽预测和自适应压缩技术,BP-ACM方法能够有效利用网络带宽,提高服务迁移性能,进而保证边缘节点为移动用户提供服务质量。

1 服务迁移问题分析

1.1 相关知识

Docker是一个开源的容器引擎,基于Go语言并遵从Apache2.0协议开源。Docker将集装箱思想运用到软件打包上,伪代码提供了一个基于容器的标准化运输系统。Docker可将任何应用及其依赖打包成一个轻量级、可移植、自包含的容器。

CRIU是一个软件冻结和恢复执行工具。CRIU能够冻结或者部分冻结一个应用程序的执行,并将其执行状态保存到磁盘的一系列文件中。然后使用这些文件来恢复运行该应用程序,并从之前被冻结的点开始执行。

实时迁移是指迁移期间不会中断正在运行的应用程序的连接,通过持续提供服务以获得无缝连接。

迭代迁移是指通过使用pre-dump命令将一部分内存数据提前传输到目标服务器,pre-dump阶段不会停止容器的运行,然后使用dump命令将剩余的内存数据传输到目标服务器,dump阶段将导致容器服务停止。迭代迁移大幅减少了服务的停机时间。

停机时间是指迁移容器提供的服务不可用或无法满足用户请求的时间。

总迁移时间是指从启动迁移过程到容器可供目标服务器使用的这段时间。

传输数据量是指迁移期间传输的总数据量。当迁移的容器正在运行网络密集型服务时,这是一项关键的衡量标准,因为它将在迁移过程中争用网络带宽。

1.2 问题分析

服务的实时迁移是指在服务不中断的前提下跟随用户的移动,当服务任务从当前服务器迁移到其他服务器时,为在保证服务性能的同时快速平滑迁移,需要通过服务迁移性能指标进行衡量,迁移性能指标主要包括总迁移时间、停机时间以及传输数据量。

由于服务是随着终端用户进行移动的,当用户不在边缘网络服务器的覆盖范围内时,网络服务性能将降低,QoS将急剧下降,并且运行的边缘服务将被中断^[15]。当正在运行服务的终端远离服务器的服务区时,则需要进行服务迁移,找寻距离更近的服务器,将未完成的用户服务迁移到该服务器,继续进行用户服务。在服务迁移过程中,需要传输的数据(例

如内存状态、网络状态、磁盘状态、程序镜像数据、数据集等)相对较大(可以达到几百兆字节甚至几千兆字节),如果传输过程花费的时间过长则会降低服务质量,影响服务迁移的效率,并且数据的频繁传输需要耗费大量的网络资源,导致服务迁移时间以及停机时间过长,降低服务质量,严重影响用户体验^[16-17]。本文提出BP-ACM方法来解决这些问题,预测网络带宽,调整压缩算法,使服务迁移过程中的传输数据更加紧凑,从而有效利用网络资源,减少服务迁移时间并改善QoS。

2 Docker 容器迁移方法

BP-ACM方法首先使用基于小波变换的ARMA带宽预测模型进行网络带宽预测^[18],然后根据网络带宽预测结果选择合适的数据压缩算法,实现服务的快速迁移。通过带宽预测预先获取迁移时的带宽,能够减少数据压缩算法适应网络带宽的时间及服务迁移的数据量,降低服务迁移的停机时间。

2.1 基于小波变换的ARMA带宽预测模型

根据小波变换具有多分辨率的特点,首先将网络带宽表示为一个低频成分与不同分辨率下的高频成分,然后根据不同分辨率下的频率成分的各自特性,分别采用ARMA模型对低频成分和不同分辨率下的高频成分进行预测,最后把各分层数据的预测结果重组生成网络带宽的最终预测结果。基于小波变换的ARMA预测模型架构如图1所示。

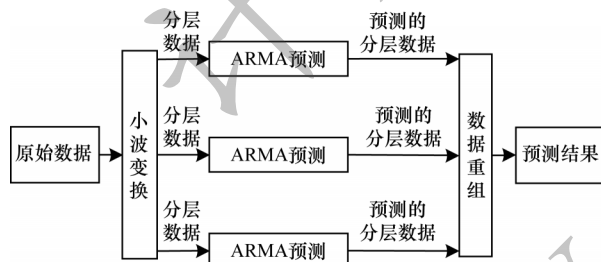


图1 基于小波变换的ARMA预测模型架构

Fig.1 ARMA prediction model architecture based on Wavelet transform

2.1.1 小波变换滤波网络带宽

由于实际网络带宽时间序列的非平稳特性,因此需要通过小波变换将其分解为多个平稳分量。本文使用小波Mallat^[19]对网络带宽数据进行分解,Mallat算法原理如下:

$$c_k^j = \sum_n h(n-2k)c_n^{j-1} \quad (1)$$

$$d_k^j = \sum_n g(n-2k)c_n^{j-1} \quad (2)$$

其中: $h(n)$ 和 $g(n)$ 分别是低通和高通滤波器。将原始的网络带宽数据 c_k^0 分解为 c_k 和 d_k , c_k 为带宽数据的近似分量,包含 c_k^1 、 c_k^2 、 c_k^3 , d_k 为带宽数据的细节分量,包含 d_k^1 、 d_k^2 、 d_k^3 。Mallat数据分解如图2所示。

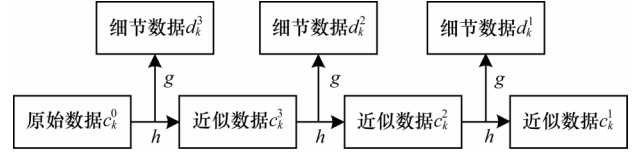


图2 Mallat数据分解

Fig.2 Mallat data decomposition

在原始网络带宽数据进行分解后,使用ARMA模型对各分层数据进行预测,再通过式(3)重构算法,将各分层数据的预测结果进行重构,得到最终的网络带宽预测结果^[20],如图3所示。

$$c_n^{j-1} = \sum_n h^*(n-2k)c_k^j + \sum_n g^*(n-2k)d_k^j \quad (3)$$

其中: $h^*(n)$ 和 $g^*(n)$ 分别是滤波器 $h(n)$ 和 $g(n)$ 的共轭的时序反转。

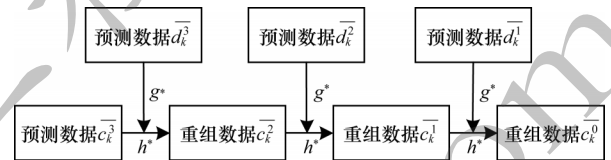


图3 Mallat数据重构

Fig.3 Mallat data reconstruction

2.1.2 基于ARMA的网络带宽预测

ARMA模型^[21]是一种典型的短期时间序列预测模型。时间序列 $\{X_t\}$ 的自回归滑动平均模型(简记为ARMA(p, q)模型)定义如下:

$$X_t = \varphi_0 + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \dots + \varphi_p X_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (4)$$

其中: X_t 为时间序列 $\{X_t\}$ 在 t 时刻的元素;参数 $\varphi_1, \varphi_2, \dots, \varphi_p$ 和参数 $\theta_1, \theta_2, \dots, \theta_q$ 为自回归系数和滑动平均系数; $\{\varepsilon_t\}$ 为噪声序列,即 $\varepsilon_t \sim wn(0, \sigma_\varepsilon^2)$,且与前期变量 $X_{t-1}, X_{t-2}, \dots, X_{t-p}$ 相独立。

本文带宽预测模型在小波分解时,采用Mallat算法将带宽数据分解为3层数据分别进行预测,然后进行预测结果的重构,具体步骤如下:

1)将原始网络带宽数据 c_k^0 分解为近似分量 c_k^3 和细节分量 d_k^1, d_k^2, d_k^3 。

2)将分解后的各层分量 $c_k^3, d_k^1, d_k^2, d_k^3$ 分别进行参数估计,并预测得到 $\bar{c}_k^3, \bar{d}_k^1, \bar{d}_k^2, \bar{d}_k^3$ 。

3)预测后的各层分量 $\bar{c}_k^3, \bar{d}_k^1, \bar{d}_k^2, \bar{d}_k^3$ 使用重构算法进行分量重构得到 \bar{c}_k^2, \bar{c}_k^1 以及最终的带宽预测结果 \bar{c}_k^0 。

2.2 自适应迁移数据压缩

由于边缘节点网络环境的不稳定性,导致服务迁移过程中数据延迟而影响服务质量,自适应迁移数据压缩就是根据网络带宽的变化调整压缩算法,使服务迁移过程中的传输数据更加紧凑,更有效地利用网络资源。边缘网络带宽有限,迁移任务传输的数据量太大,增加了服务中断时间,影响了用户的服务质量。该算法考虑了服务器的计算资源,充分

利用了网络带宽资源,使压缩时间和传输时间达到平衡,以最短时间完成服务迁移^[22]。

2.2.1 压缩算法的选择

在BP-ACM方法中,使用2种压缩算法:网络带宽较低时采用压缩速度相对较慢但压缩率较高的Zstd压缩算法^[23],网络带宽较高时采用压缩速度较快但压缩率较低的LZ4压缩算法^[24],采用迭代测试来确定高低带宽的临界值。通过分别运行内核编译、Web服务、Zeusmp这3种负载,获得LZ4和Zstd压缩算法在不同网络带宽下的服务迁移速度^[25]临界值,如表1所示。

表1 LZ4与Zstd在不同网络带宽下的迁移速度对比

Table 1 Comparison of migration speed between LZ4 and Zstd under different network bandwidths (MB·s ⁻¹)					
压缩算法	迁移速度	网络带宽为50 MB/s	网络带宽为60 MB/s	网络带宽为70 MB/s	网络带宽为80 MB/s
LZ4	最大值	104	118	137	149
	最小值	71	83	99	106
Zstd	最大值	129	130	132	145
	最小值	74	87	96	101

从表1中可以看出:当网络带宽为60 MB/s时,相比于LZ4压缩算法,Zstd压缩算法的迁移速度更快;当带宽为70 MB/s时,LZ4压缩得到的迁移速度更快。可见,网络带宽为60 MB/s是选择压缩算法的标准。

2.2.2 Zstd压缩算法的优化

由网络迁移速度的临界值可知,在网络带宽小于或等于60 MB/s的情况下,采用Zstd压缩算法。通过采用多种负载来测量Zstd压缩算法各压缩级别的压缩率和压缩速度,发现不同负载下各压缩级别的压缩率和压缩速度没有明显差异,所以本文通过运行多种负载来获取压缩率及压缩速度的平均值,并且与压缩算法的压缩级别一一对应。根据服务迁移速度是由网络带宽、压缩速度以及压缩率决定,当服务迁移过程中加入压缩算法时,压缩后需要传输的数据量大小是原始数据大小的 $1/\theta$ 。这意味着服务迁移速度间接增加了一个 θ 因子,减少了数据的传输,提高了服务迁移的速度。

服务迁移数据的压缩和压缩内容的传输是并行的,在传输压缩数据的同时也在进行数据的压缩,因此,具有压缩算法的迁移速度是压缩速度和增加的传输速度中的最小值,即:

$$V_{mr} = \min(V_c, B_w \times \theta)$$
 (5)

其中: V_{mr} 是服务迁移速度; V_c 是压缩速度; B_w 是网络带宽; θ 是压缩率。 V_{mr} 越大,总的迁移时间越短,当给定一个 B_w 值时,调整压缩算法的压缩率 θ 和压缩速度 V_c ,可得到最优的服务迁移速度 V_{mr} 。

通过分别计算Zstd压缩算法不同压缩级别的压缩率和压缩速度,得到最大迁移速度下的最佳压缩级别。由于压缩率 θ 和压缩速度 V_c 是随着压缩级别近似平滑变化的,因此压缩级别可以只取20个压缩级别中的奇数值,将10对 θ 和 V_c 值保存在数组 $V[\]$ 中,伪代码如算法1所示,其中 $V[i][0]$ 为压缩率 θ , $V[i][1]$ 为压缩速度 V_c , B_L 代表最优压缩级别。

算法1 压缩算法的优化

```
输入  获取的预测带宽  $B_w$ 
输出  Zstd最优压缩级别  $B_L$ 
1.max  $V_{mr} = 0$ //初始化最大服务迁移速度
2.i = 0//初始化指针
3.While(i < 10)//选择最优压缩级别
4.if  $V[i][1] < V[i][0] \times B_w$ //压缩速度和压缩率与迁移带宽//之积比较取最小值
5. $V_{mr} = V[i][1]$ 
6.else
7. $V_{mr} = V[i][0] \times B_w$ 
8.end if
9.if  $V_{mr} > \max V_{mr}$ //获取最大服务迁移速率
10.max  $V_{mr} = V_{mr}$ 
11.end if
12.i = i + 1
13.end While
14. $B_L = i$ //将最大迁移速率对应的压缩级别赋值给最优//压缩级别  $B_L$ 
15.return  $B_L$ 
```

2.2.3 LZ4压缩算法的优化

本文带宽预测模型在网络带宽大于60 MB/s的情况下采用LZ4压缩算法,服务迁移速度随着LZ4压缩算法压缩级别的增加先增大后减小,压缩级别为7时迁移速度达到最快。由于该算法的压缩级别数量过多,最优压缩级别的寻找需要花费大量的时间,容器服务迁移速度随着压缩级别的增加逐渐稳定,当压缩级别达到25时,迁移速度已经稳定,因此本文通过设置压缩级别为小于25的奇数,以此减少压缩级别的个数,这样既可以得到最优压缩级别,又可以节省CPU开销。同样,由于压缩级别的变化压缩率 θ 和压缩速度 V_c 是近似平滑变化的,因此将对应的压缩率 θ 和压缩速度 V_c 的值保存在数组 $V[\]$ 中,由 $V_{mr} = \min(V_c, B_w \times \theta)$ 可以求出最大迁移速度时的最优压缩级别。

2.3 计算复杂度分析

本文提出的BP-ACM方法涉及的计算主要包括以下3个方面:

1)网络带宽数据处理。首先获取网络带宽数据,从系统proc/net/dev文件中读取网络带宽,复杂度为 $O(m)$;然后通过小波变换对网络带宽数据进行分解和重构,复杂度为 $O(mlb\ m)$,其中 m 为样本数量。

2)网络带宽数据预测。使用ARMA模型对分解后的带宽数据分别进行预测,ARMA模型包括阶数的确定和参数估计,分别使用AIC准则和最小二乘法进行确定,复杂度分别为 $O(m+p)$ 和 $O(m^2)$,其中, p 为参数数量。

3)数据压缩算法选择。根据带宽预测结果选择压缩算法:LZ4数据压缩算法,复杂度为 $O(n)$;Zstd数据压缩算法,复杂度为 $O(l \times n) + O(n \lg n)$,其中, n 为传输数据的字符数量, l 为字符匹配的长度, $n \gg l$ 。

综上,本文提出的BP-ACM迁移方法的计算复杂度为 $O(m^2) + O(n \lg n)$ 。

2.4 迁移方法实现

本文提出的BP-ACM方法通过预测网络带宽自适应地选择数据压缩方法进行 Docker 容器服务迁移,整体系统架构如图4所示。

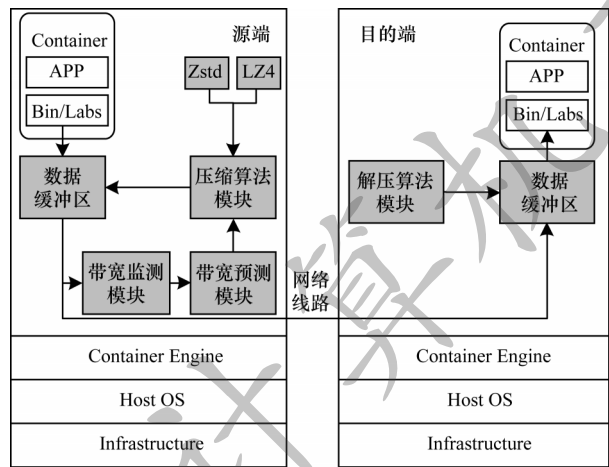


图4 Docker容器服务迁移架构

Fig.4 Docker container service migration architecture

Docker 容器服务迁移架构主要由以下模块构成:1)带宽监测模块,从proc/net/dev文件获取以往的网络带宽数值,为网络带宽预测方法提供数据;2)带宽预测模块,首先通过小波变换对带宽数据进行分解,然后使用ARMA模型进行预测,最后对各层预测数据进行重构,得出最终结果;3)压缩算法模块,结合带宽预测的结果,选择合适的压缩算法,确定最佳的压缩级别以及最大的迁移速度;4)解压算法模块,识别2种压缩算法,然后对压缩数据进行解压。

整个系统的运行流程如下:1)带宽监测,获取网络带宽数据;2)结合带宽数据,使用带宽预测方法进行带宽预测;3)通过带宽预测的结果选择压缩算法,然后代入式(5)中,获得最优压缩级别以及最大迁移速度;4)对比2种压缩算法的性能,分别对压缩算法进行优化;5)对服务迁移数据进行压缩处理;6)对压缩数据进行传输;7)选择对应的解压算法对数据进行解压。

3 实验验证

3.1 实验环境设置

为评估容器迁移场景,实验配置3台独立的服务器,用来模拟边缘服务节点。第1台是源MEC主机,第2台是目标MEC主机,第3台是网络文件系统(Network File System, NFS)服务器,用于存储容器的文件系统。每台服务器使用带有4.4.0-64通用内核的Ubuntu16.04LTS,拥有4核CPU和8 GB主存,2个Intel PRO/1000千兆网络接口卡(Network Interface Card, NIC)。容器环境采用Docker版本为17.06.0和CRIU版本为3.14(均为稳定版本)的设置。实验使用Pre-copy、Zstd-ACM、LZ4-ACM、BP-ACM等4种迁移方法运行不同类型负载进行服务迁移对比,并进行全面的容器迁移性能测试分析。工作负载详细信息见表2。

表2 工作负载详细信息

Table 2 Workload details

工作负载	描述
内核编译	Linux 3.13.2内核,是测试系统虚拟化性能的平衡工作负载
Web 服务	1个客户端配置100个并发连接,每个连接不断向网络服务器下发请求
Zeusmp	浮点基准,使用计算流体力学来模拟天体物理现象

3.2 带宽预测结果分析

以郑州大学校园网络作为测试载体,采用2021年3月21日到2021年3月23日的网络带宽数据进行实验,数据收集间隔为5 min,以4 ms为一个测试时间段,取后800个数据作为原始数据集,将这些数据按时间先后顺序形成一个时间序列,所得网络带宽的数据曲线如图5所示。由图5可以看出,网络带宽具有周期性和不平稳特性,表明用户使用网络时间具有相似性且网络带宽随着时间的变化而变化。

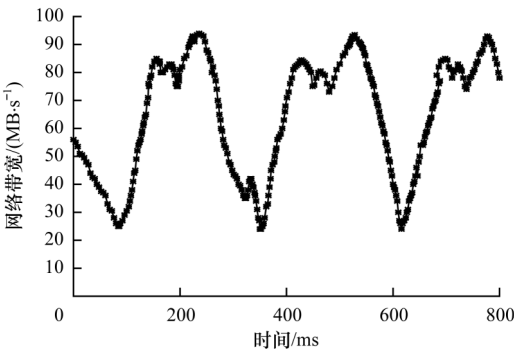


图5 实际网络带宽

Fig.5 Actual network bandwidth

图6是使用带宽预测模型进行预测的结果,可以看出本文模型的预测效果和实际网络带宽数据比较吻合,能够较好地预测现实中的网络带宽情况。

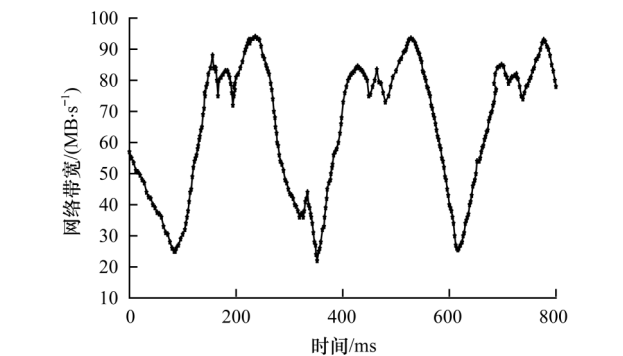


图6 网络带宽预测结果

Fig.6 Prediction results of network bandwidth

另外,本文还采用AR模型、ARMA模型、小波-AR模型的预测结果与本文带宽预测模型(简称为小波-ARMA模型)的预测结果进行比较,采用平均绝对误差(Mean Absolute Error,MAE)和平均相对误差(Mean Relative Error,MRE)来判断模型的预测精确度,MAE是预测值与实际值的绝对误差的平均值,MRE是预测值与实际值的相对误差率的平均值,结果如表3所示。由表3可以看出,本文提出的带宽预测模型大幅降低了预测误差,说明小波分解在预测过程中发挥了重要作用,也证实了该模型完全考虑了网络带宽数据的非平稳和多分辨率特性,对于带宽预测具有重要意义。

表3 网络带宽预测结果数据对比		
Table 3 Data comparison of network bandwidth prediction results		
模型	MAE	MRE/%
AR	1.921	2.31
ARMA	1.827	2.11
小波-AR	1.673	2.04
小波-ARMA	1.324	1.92

3.3 迁移性能分析

在同样的实验环境下,使用Pre-copy、Zstd-ACM、LZ4-ACM、BP-ACM 4种迁移方法以及内核编译、Zeusmp、Web服务3种不同类型的工作负载进行服务迁移对比实验,分别从容器服务迁移传输总时间、容器服务迁移传输数据量以及容器服务迁移停机时间比较容器服务迁移性能。

图7比较了网络迁移带宽为30 MB/s、60 MB/s、90 MB/s时,使用Pre-copy、Zstd-ACM、LZ4-ACM、BP-ACM迁移方法分别运行不同类型负载下的容器服务迁移总时间。由图7可以看出,随着网络带宽的增加,相比于Zstd-ACM和LZ4-ACM方法,BP-ACM方法通过带宽预测模型能够预先地选择压缩算法并对压缩算法进行优化,进一步缩短容器服务迁移总时间。

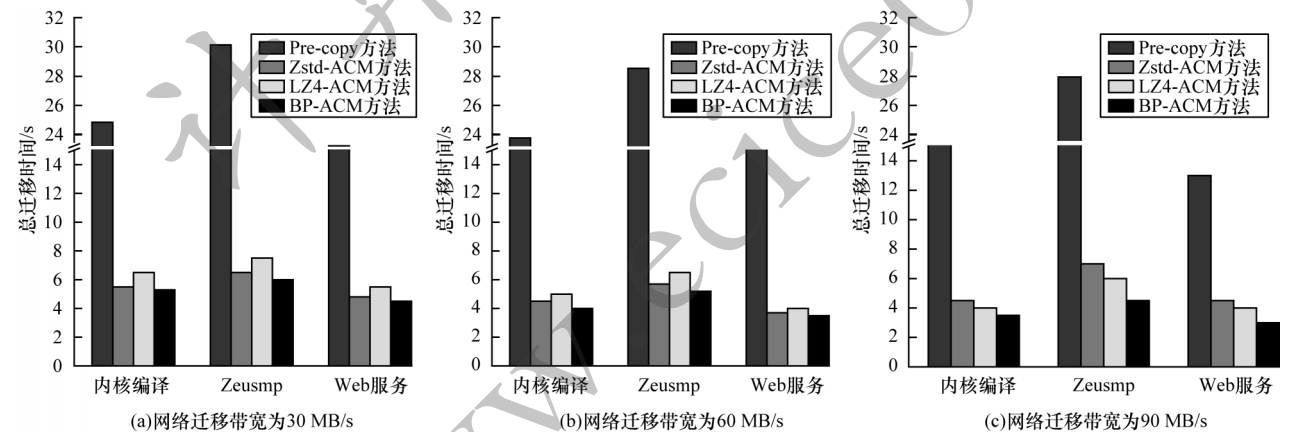


图7 容器服务迁移总时间

Fig.7 Total container service migration time

图8比较了网络迁移带宽为30 MB/s、60 MB/s、90 MB/s时,使用Pre-copy、Zstd-ACM、LZ4-ACM、BP-ACM迁移方法分别运行不同类型负载下的服务迁移传输数据量。由图8可以看出,容器服务迁移中的适应性压缩算法能够大幅减少迁移所传输的传输数据量,相比于Zstd-ACM和LZ4-ACM方法,BP-ACM方法能够有效地平衡数据压缩和数据传输所消耗的时间,在网络带宽低时,尽可能地压缩数据,减少传输时间;在网络带宽高时,降低压缩率,减少压缩时间。

图9比较了网络迁移带宽为30 MB/s、60 MB/s、90 MB/s时,使用Pre-copy、Zstd-ACM、LZ4-ACM、

BP-ACM迁移方法分别运行不同类型负载下的容器服务迁移停机时间。由图9可以看出,与Pre-copy方法相比,Zstd-ACM、LZ4-ACM和BP-ACM方法使得容器服务迁移停机时间从数百毫秒大幅缩短到了数十毫秒。另外,随着网络带宽的增加,容器的服务迁移停机时间均有所减少。

本文还在实际网络环境中进行了容器服务迁移实验,如图10所示,Zstd-ACM、LZ4-ACM和BP-ACM方法均比Pre-copy方法有了显著改进,并且BP-ACM方法比Zstd-ACM和LZ4-ACM方法能够更好地提升容器服务迁移效率。

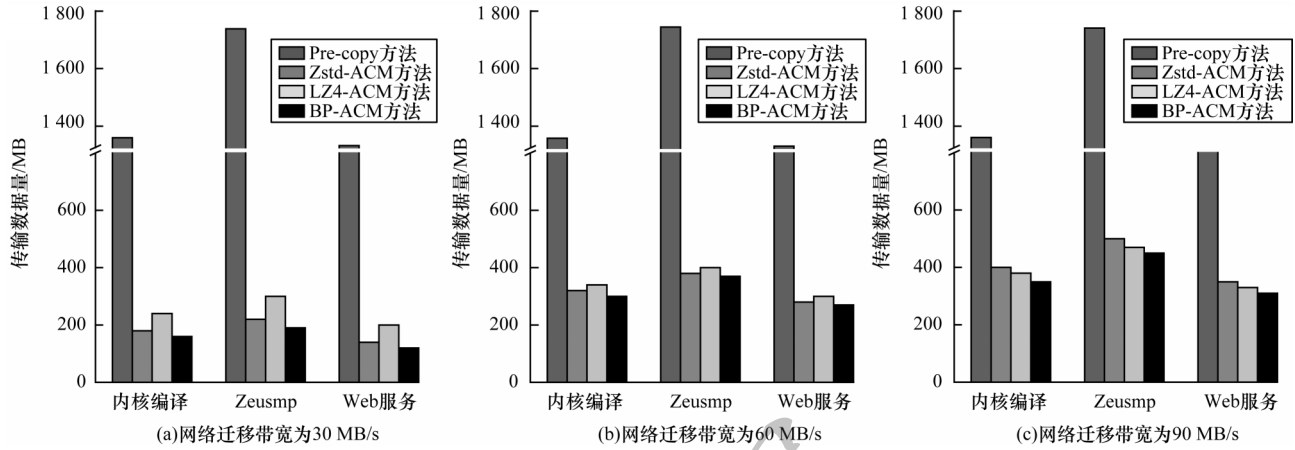


图8 容器服务迁移传输数据量

Fig.8 Container service migration transfer data volume

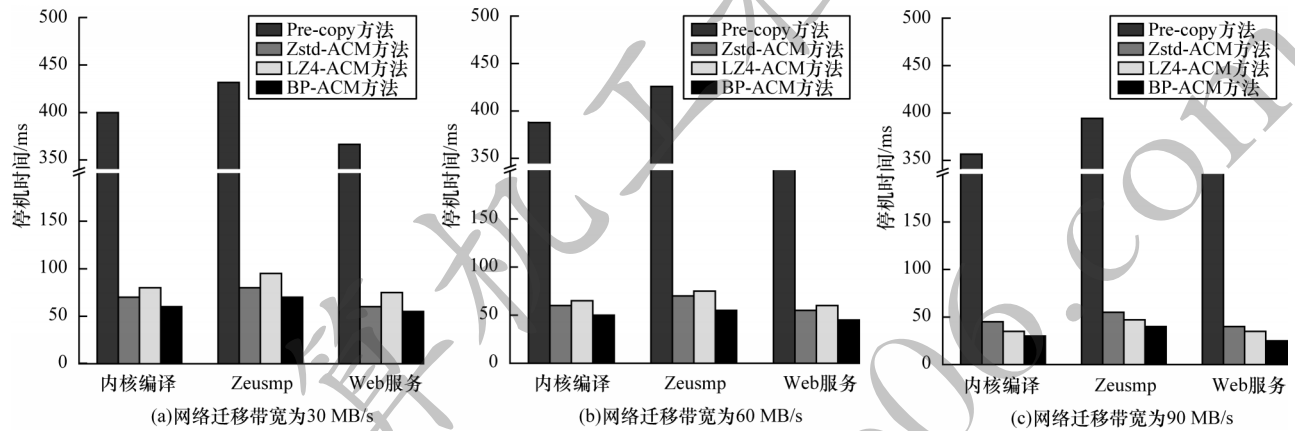


图9 容器服务迁移停机时间

Fig.9 Container service migration downtime

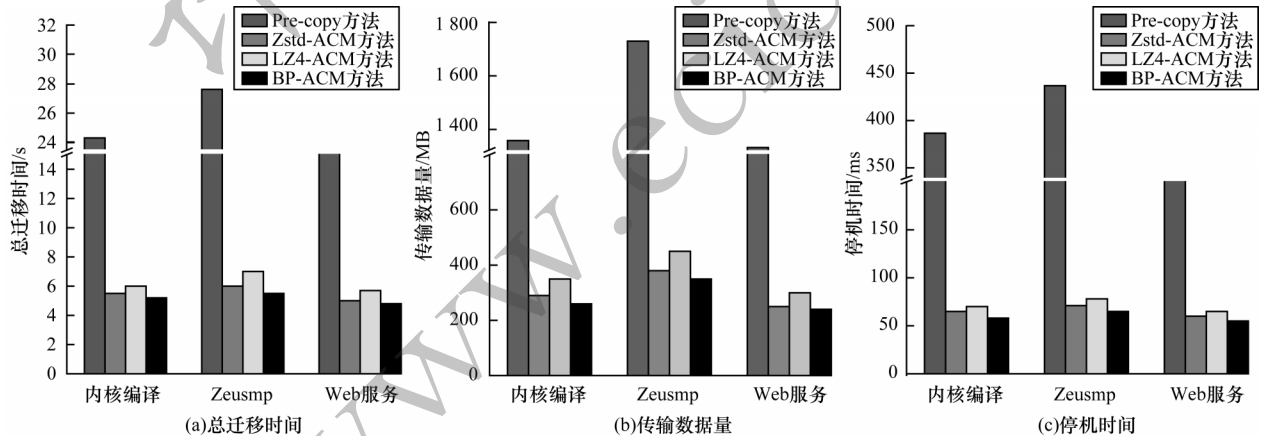


图10 实际网络环境下的容器服务迁移性能对比

Fig.10 Comparison of container service migration performance under actual network environment

从容器服务迁移总时间、传输数据量以及停机时间性能指标的对比结果可以看出,本文所提出的BP-ACM服务迁移方法能够有效地提高 Docker 容器服务迁移性能,从总迁移时间和传输数据量方面可以看出性能有了很大提升,而停机时间则是受网络迁移带宽的影响较大。通过比较4种服务迁移方法可以看出,BP-ACM方法能够在很大程度上提升 Docker 容器服务迁移的综合性能,其中,总迁移时间

至少减少了23.7%,传输数据量至少减少了19.4%,停机时间至少减少了17.6%,充分验证了本文所提出的BP-ACM迁移方法的有效性。

3.4 计算复杂度对比

4种容器服务迁移方法的计算复杂度分析如下:

1) Pre-copy方法:通过循环方式将内存数据传输到目的容器,总的复杂度为 $O(n)$ 。

2) Zstd-ACM方法:循环获取实时的网络带宽,

根据带宽使用 Zstd 压缩算法, 获得最优的压缩速度和压缩强度, 总的复杂度为 $O(n \lg n)$ 。

3) LZ4-ACM 方法: 循环获取实时的网络带宽, 根据网络带宽使用 LZ4 压缩算法, 获得最优的压缩速度和压缩强度, 总的复杂度为 $O(n)$ 。

4) BP-ACM 方法: 预测网络带宽, 选择 Zstd 或 LZ4 压缩算法, 获得最优的压缩速度和压缩强度, 总的复杂度为 $O(m^2) + O(n \lg n)$ 。

由表 4 可以看出, 相比于其他 3 种迁移方法, 本文提出的 BP-ACM 方法的计算复杂度有所提升, 但通过迁移性能结果分析可知其服务迁移性能得到了有效提升, 整体而言 BP-ACM 方法具有较高的实用性。

表 4 容器服务迁移方法的计算复杂度对比

Table 4 Comparison of computational complexity of container service migration methods

迁移方法	复杂度
Pre-copy	$O(n)$
Zstd-ACM	$O(n \lg n)$
LZ4-ACM	$O(n)$
BP-ACM	$O(m^2) + O(n \lg n)$

4 结束语

对于边缘计算服务迁移, 传统的容器服务迁移方法比较繁琐, 给边缘网络带宽带来了负担。本文提出 BP-ACM 服务迁移方法, 通过在评估边缘网络状态后动态调整压缩强度, 并全面平衡计算开销和网络传输开销。实验结果表明, 该方法可在迁移期间提供稳定响应, 并且将网络带宽预测模型与自适应压缩算法相结合相比传统非压缩迁移方法以及单一压缩算法的迁移方法效率更高。后续将改进现有压缩算法, 采用增量压缩方式进一步减少传输数据量, 提高实时迁移性能。

参考文献

- [1] LIU S, LIU L, YANG H, et al. Research on 5G technology based on Internet of Things [C]//Proceedings of the 5th Information Technology and Mechatronics Engineering Conference. Washington D. C., USA: IEEE Press, 2020: 1821-1823.
- [2] 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望 [J]. 计算机研究与发展, 2019, 56(1): 69-89.
SHI W S, ZHANG X Z, WANG Y F, et al. Edge computing: state-of-the-art and future directions [J]. Journal of Computer Research and Development, 2019, 56(1): 69-89. (in Chinese)
- [3] PORAMBAGE P, OKWUIBE J, LIYANAGE M, et al. Survey on multi-access edge computing for Internet of Things realization [J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 2961-2991.
- [4] GRASSI V. Architectural issues for self-adaptive service migration management in mobile edge computing scenarios [C]//Proceedings of 2019 IEEE International Conference on Edge Computing. Washington D. C., USA: IEEE Press, 2019: 27-29.
- [5] PULIAFITO C, VIRDIS A, MINGOZZI E. The impact of container migration on fog services as perceived by mobile things [C]//Proceedings of 2020 IEEE International Conference on Smart Computing. Washington D. C., USA: IEEE Press, 2020: 70-77.
- [6] 吴大鹏, 吕吉, 李职杜, 等. 移动性感知的边缘服务迁移策略 [J]. 通信学报, 2020, 41(4): 1-13.
WU D P, LÜ J, LI Z D, et al. Mobility aware edge service migration strategy [J]. Journal on Communications, 2020, 41(4): 1-13. (in Chinese)
- [7] ZHANG M L, HUANG H Q, RUI L L, et al. A service migration method based on dynamic awareness in mobile edge computing [C]//Proceedings of 2020 IEEE/IFIP Network Operations and Management Symposium. Washington D. C., USA: IEEE Press, 2020: 1-7.
- [8] WANG S G, XU J L, ZHANG N, et al. A survey on service migration in mobile edge computing [J]. IEEE Access, 2018, 6: 23511-23528.
- [9] BELLAVISTA P, CORRADI A, FOSCHINI L, et al. Differentiated service/data migration for edge services leveraging container characteristics [J]. IEEE Access, 2019, 7: 139746-139758.
- [10] HAN Z J, ZHANG Y J, WANG R C. A locality live migration strategy based on docker containers [EB/OL]. [2021-04-12]. https://www.researchgate.net/publication/329318325_A_Locality_Live_Migration_Strategy_Based_on_Docker_Containers.
- [11] FAN W B, HAN Z J, LI P, et al. A live migration algorithm for containers based on resource locality [J]. Journal of Signal Processing Systems, 2019, 91(10): 1077-1089.
- [12] ELGAZAR A, HARRAS K. Teddybear: enabling efficient seamless container migration in user-owned edge platforms [C]//Proceedings of 2019 IEEE International Conference on Cloud Computing Technology and Science. Washington D. C., USA: IEEE Press, 2019: 70-77.
- [13] LI J, SHEN X M, CHEN L, et al. Delay-aware bandwidth slicing for service migration in mobile backhaul networks [J]. Journal of Optical Communications and Networking, 2019, 11(4): 1-9.
- [14] ZHANG X J, WU W G, ZHANG C, et al. Dynamic adaptive network edge service migration method based on a docker container [C]//Proceedings of the 5th International Conference on Computer and Communications. Washington D. C., USA: IEEE Press, 2019: 1398-1405.
- [15] PULIAFITO C, VIRDIS A, MINGOZZI E. Migration of multi-container services in the fog to support things mobility [C]//Proceedings of 2020 IEEE International Conference on Smart Computing. Washington D. C., USA: IEEE Press, 2020: 259-261.
- [16] GAO Z P, MENG J, WANG Q, et al. Service migration for deadline-varying user-generated data in mobile edge-clouds [C]//Proceedings of 2018 IEEE World Congress on Services. Washington D. C., USA: IEEE Press, 2018: 51-52.
- [17] LABRIJI I, MENEGHELLO F, CECCHINATO D, et al. Mobility aware and dynamic migration of MEC services for the Internet of vehicles [J]. IEEE Transactions on Network and Service Management, 2021, 18(1): 570-584.
- [18] TAN H J, YANG Y H. Network traffic prediction based on multi-scale wavelet transform and mixed time series model [C]//Proceedings of the 9th International Conference on Computer Science & Education. Washington D. C., USA: IEEE Press, 2014: 696-699.

(上接第207页)

- [19] BAO Q X, GAO Y, WEI X, et al. An integrated model for traffic flow prediction based on the wavelet transform[C]//Proceedings of the 24th Asia-Pacific Conference on Communications. Washington D. C. , USA: IEEE Press, 2018: 88-93.
- [20] JIAO H J, LI J P, JIANG T C. The cloud resource load security detection based on wavelet packet transform and grey wolf algorithm[C]//Proceedings of the 15th International Computer Conference on Wavelet Active Media Technology and Information Processing. Washington D. C. , USA: IEEE Press, 2018: 1-5.
- [21] 麻书钦, 范海峰. 基于小波变换和时间序列的网络流量预测模型[J]. 河南理工大学学报(自然科学版), 2013, 32(2): 188-192.
- MA S Q, FAN H F. Network traffic prediction based on wavelet transform and time series model[J]. Journal of Henan Polytechnic University(Natural Science), 2013, 32(2): 188-192. (in Chinese)
- [22] TORRE R, URBANO E, SALAH H, et al. Towards a better understanding of live migration performance with Docker containers[C]//Proceedings of the 25th European Wireless Conference. Aarhus, Denmark: VDE Press, 2019: 1-6.
- [23] CHEN J Y, DAVERVELDT M, AL-ARS Z. FPGA acceleration of Zstd compression algorithm[C]//Proceedings of 2021 IEEE International Parallel and Distributed Processing Symposium Workshops. Washington D. C. , USA: IEEE Press, 2021: 188-191.
- [24] 梅发强. 基于 LZ4 数据压缩与格加密的大数据加密算法研究[D]. 南京: 南京航空航天大学, 2019.
- MEI F Q. Research on big data encryption algorithm based on LZ4 data compression and lattice encryption [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2019. (in Chinese)
- [25] CHAKRABORTY S, BANDYOPADHYAY A. A two stage data compression and decompression technique for point cloud data[C]//Proceedings of 2018 World Symposium on Digital Intelligence for Systems and Machines. Washington D. C. , USA: IEEE Press, 2018: 297-302.

编辑 陆燕菲