计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220602
61(2): 494−502, 2024

一种有状态容器跨集群实时迁移方法

吴 宪 汤红波 赵 宇 许明艳
(解放军信息工程大学 郑州 450001)
(infinity_w@outlook.com)

A Cross-Cluster Real-Time Migration Method for Stateful Container

Wu Xian, Tang Hongbo, Zhao Yu, and Xu Mingyan
(*Information Engineering University*, *Zhengzhou* 450001)

Abstract The real-time migration technology of cross-cluster stateful virtual containers is an important technology in the cloud application of network services. The optimization of stateful container real-time migration technology can achieve the purpose of shortening the downtime of cloud services, which is of great practical significance in practical applications. At present, there are relatively few studies on cross-cluster container real-time migration optimization, of which iterative migration is a migration method that can effectively reduce downtime, but there are still many problems to be solved in practical applications of this technology. Firstly, a container real-time migration method is proposed for the impact of network latency on the service downtime in cross-cluster scenarios, which focuses on the network latency between clusters, and a mathematical model is established accordingly. Secondly, The influence of network latency on migration performance during cross-cluster migration is analyzed by mathematical model theory, which determines the starting conditions and the termination conditions for enabling the iterative migration process, and gives the method to calculate the number of iterations that satisfy the termination condition. The proposed model fully considers the impact of network latency on cross-cluster container real-time migration, and experimental results show that the proposed method can effectively reduce service downtime.

Key words container real-time migration; network latency; service downtime; iterative migration; cross-cluster

摘 要 虚拟容器的跨集群实时迁移技术是网络服务云化应用中的一项重要技术.容器实时迁移技术的优化能够达到缩短云端服务停机时间的目的,在实际应用中有重要的现实意义.目前针对跨集群容器实时迁移优化的研究相对较少,其中迭代迁移是一种能够有效缩短停机时间的迁移方式,但这项技术在实际应用中还存在很多需要解决的问题.针对跨集群场景中网络延迟对服务停机时间产生的影响提出了一种容器实时迁移方法,该方法着重考虑集群间的网络延迟,并据此建立了数学模型.通过数学模型理论分析了跨集群迁移过程中网络延迟对迁移性能的影响,确定了启用迭代迁移方法的起始条件和迭代过程的终止条件,并给出了计算满足终止条件迭代次数的方法.该模型充分考虑了网络延迟对跨集群容器实时迁移的影响.实验结果表明该方法能够有效缩短服务停机时间.

关键词 容器实时迁移;网络延迟;服务停机时间;迭代迁移;跨集群

中图法分类号 TP302

随着网络结构的不断演进,包括边缘计算、物联网甚至5G核心网等网络服务都将会部署在云化基础设施上[1-2].云化为5G等网络服务带来软硬件解耦、切片友好运营等大量优势,但也使得网络服务更加

收稿日期:2022-06-29;**修回日期**:2023-04-14
通信作者:汤红波(tahobo@sina.com)

依赖虚拟容器技术. 在云端虚拟容器面对集群硬件维护停机、集群负载失衡、云平台故障、服务容器升级、更换云供应商以及特殊服务需要时，经常需要使用有状态虚拟容器实时迁移技术[3].

针对虚拟化容器迁移技术的研究主要集中于同一集群内部的迁移方法优化，对跨集群容器迁移方法优化的研究相对较少，而且由于在实际工况下容器迁移应用场景经常面临跨集群的情况，该场景下物理存储介质不同导致容器在集群内部迁移方法无法应用，因此需要一种合理高效的迁移方法来缩短跨集群场景中的有状态虚拟容器迁移造成的服务停机时间. 在有状态虚拟容器跨集群迁移的过程中会存在 3 个问题：

1）当前用户空间中的检查点/还原工具（checkpoint/restore in userspace, CRIU）均采用一次性迁移全部状态文件的方法，当 CRIU 状态文件过大或集群间网络传输速率过慢时，状态文件的迁移耗时会很长，将直接导致容器所提供服务的停机时间变得很长且不可控.

2）同一集群内的虚拟容器迁移依赖底层存储硬件的访问共享，集群内迁移方法的高效执行均依赖硬盘、DRAM 以及内核的共享，而在跨集群场景中无法共享底层存储介质. 因此这些方法均无法在有状态容器的跨集群迁移过程中应用.

3）在有状态容器跨集群迁移过程中，经常遇到集群间距离远、网络延迟长的问题，较高的网络延迟会对当前的一些迁移模型产生很大影响，导致对于迁移所需时间及服务停机时间的计算不准确.

为解决有状态容器跨集群迁移过程中网络延迟会导致服务停机时间长且不可控的问题，本文提出了一种针对网络延迟的迭代迁移算法. 本文具体贡献包括 3 个方面：

1）在有状态虚拟容器跨集群迁移中使用迭代迁移的方式，缩短服务停机时间；

2）在跨集群容器迁移场景中重点考虑网络延迟，提出了一种迭代迁移模型，并确定了迭代过程的起始和终止条件，给出了最优迭代次数的计算方法；

3）在模拟环境中的计算结果表明本文提出的迭代迁移算法能够在满足应用条件的情况下，有效缩短服务停机时间，并根据网络延迟的情况，停机时间可缩短至未采用迭代传输方法服务停机时间的 25% 以下.

1 相关工作

本文主要研究内容为通过迭代迁移的方法对有状态虚拟容器跨集群迁移进行优化以缩短服务停机时间，接下来将对当前虚拟容器迁移方法的优化和容器迭代迁移 2 个方面的相关研究工作进行阐述.

1.1 集群内容器迁移优化相关研究

虚拟容器通过从内核隔离独立的命名空间和用户空间，实现容器中进程的虚拟化隔离. 一般虚拟容器由镜像、容器、网络和卷 4 个部分组成[4].

镜像中包含运行应用程序所需的内容，是一个能够运行应用程序的基础. 容器则是镜像的实例，该实例包含了进程、日志和对原始文件系统的更改. 容器的迁移发生在 2 个镜像实例间，即容器的进程运行状态和数据的迁移. 卷是容器的虚拟存储器，通常在主机上以共享文件夹的方式实现. 这些卷可以在 1 个或多个容器上挂载，能够永久存储所连接容器的程序运行状态和数据. 卷可以采用主机文件挂载的方式，使容器直接访问主机的共享文件夹. 因此容器迁移过程中，如何快速合理地迁移卷的内容就成为了关键. 目前虚拟容器的实时迁移主要通过 CRIU 完成. 该工具可以暂停运行中的容器，将生成的状态文件迁移到目标容器后恢复运行状态[5-6].

Yin 等人[7] 提出了一种根据集群内节点的负载率选择迁移目标节点的容器迁移优化方式，优化容器迁移后的各节点负载均衡，提高迁移后的服务性能. Venkatesh 等人[8] 提出了一种将基于 CRIU 的数据保存在虚拟地址空间（virtual address spaces, VAS）中的方法 VAS-CRIU，利用其读写速度比磁盘快的特点提高容器迁移效率. Xu 等人[9] 提出了对集群内容器迁移 CRIU 功能的改进，使其能够迭代地迁移 CRIU 增量，达到缩短服务停机时间的目的. Lei 等人[10] 则在容器迁移过程中使用马尔可夫模型预测内存访问模式，调整内存页面传输顺序，提高迁移过程中的内存读写效率. Lv 等人[11] 则提出了通信感知最坏适应减少（communication aware worst fit decreasing, CA-WFD）算法作为集群内容器迁移的放置策略，这个算法巧妙地把经典装箱问题的最坏适应算法应用在了容器放置的场景中，该算法结合文献中提出的另一个用于确定容器迁移计划的扫描与搜索（Sweep&Search）算法，能够高效地制定迁移计划并将容器放置到最合适的地方，提高集群的整体性能. 该文献提出的算法在百度大型数据中心进行了应用，实际性能比当时最先进的容器迁移放置方法提高了 70%.

集群内虚拟容器迁移的优化方法均要求源容器与目的容器能够共享、访问物理存储介质. 这是在跨集群容器迁移过程中无法做到的，因此这些优化方

法不能直接应用在跨集群容器迁移的场景中.

1.2 跨集群容器迁移优化相关研究

虽然集群内容器迁移优化方法无法直接应用于跨集群容器迁移场景中,但其思路大体一致.其中一部分方法经过改进摆脱对底层物理存储共享的依赖就可以在跨集群容器迁移过程中应用.同时,相比集群内容器迁移,跨集群场景下多了一种针对网络传输过程的优化方法.

Di 等人[12]通过多线程并行迁移过程的方式更加合理地安排容器迁移的工作流程,提出了一种通过管道(pipeline)进行容器迁移的策略,降低迁移过程中的资源开销,提高资源利用率及容器的迁移性能.赵倩等人[13]利用远程直接内存访问(remote direct memory access, RDMA)技术跳过了迁移过程中内存数据写入磁盘的过程,节约了迁移时间.Stoyanov 等人[14]通过利用 CRIU 的特性,即其独有的缓存/代理技术,达到了缩短容器迁移总时间和服务停机时间的目的.

除针对迁移过程中硬件及性能的优化工作以外,还有大量针对迁移过程自身展开的研究.这部分研究通过改进容器迁移的流程来提高容器迁移效率,达到缩短停机时间的目标,迭代迁移就是这类技术之一.

迭代迁移实现思路是通过 CRIU 等工具,对源容器的运行状态建立检查点(checkpoint)并传输状态文件.但在 checkpoint 建立过程、数据传输过程以及在目标容器内恢复过程中源服务并不停机,在第 1 个 checkpoint 传输完成后,将进程运行增量再次建立 checkpoint 进行传输.此过程不断迭代进行,达到缩短单次传输时间的目的,直至满足迭代终止条件,完成迁移过程.完整迁移过程如图 1 所示.

Junior 等人[15]将迭代迁移的思路应用于跨集群容器迁移中,并通过实验证明迭代迁移能够缩短服务的停机时间.但在研究及实验过程中由于没有考虑集群间网络延迟的影响,未能根据收益制定迭代过程的终止条件,无法确定最优迭代次数.

2 考虑网络延迟的迭代迁移算法

本节首先介绍算法所需的各个参数并根据模型列出计算重要参数的表达式,然后对使用迭代迁移的判定条件和终止条件进行计算,最后根据终止条件计算出最优迭代次数.

2.1 基本参数及表达式

设迁移过程总迭代次数为n,每个迁移过程分为 3 个阶段,分别为快照拍摄阶段 a、快照在集群间网络迁移阶段 b 和快照恢复阶段 c.其中当整数x满足$0 \leqslant x \leqslant n-1$时,第$x+1$个 checkpoint 的快照拍摄、传输和恢复过程的总用时为T_x,单次迭代迁移过程中的源容器进程增量拍摄快照消耗的时间为T_{xa},快照从源容器网络传输到目标容器所消耗的时间为T_{xb},在目标容器中将快照恢复为进程状态所需要的时间为T_{xc}(单位为 ms),需要迁移的状态文件大小为V_x.

开始迁移时需要制作镜像的进程状态文件大小为V_0,在迁移过程中进程状态文件的增长速率为ΔV.本文模型认为在容器迁移过程中目标容器与源容器性能完全相同,因此快照的拍摄与恢复的速率均为v_P,进程状态文件快照的压缩率用K来表示.

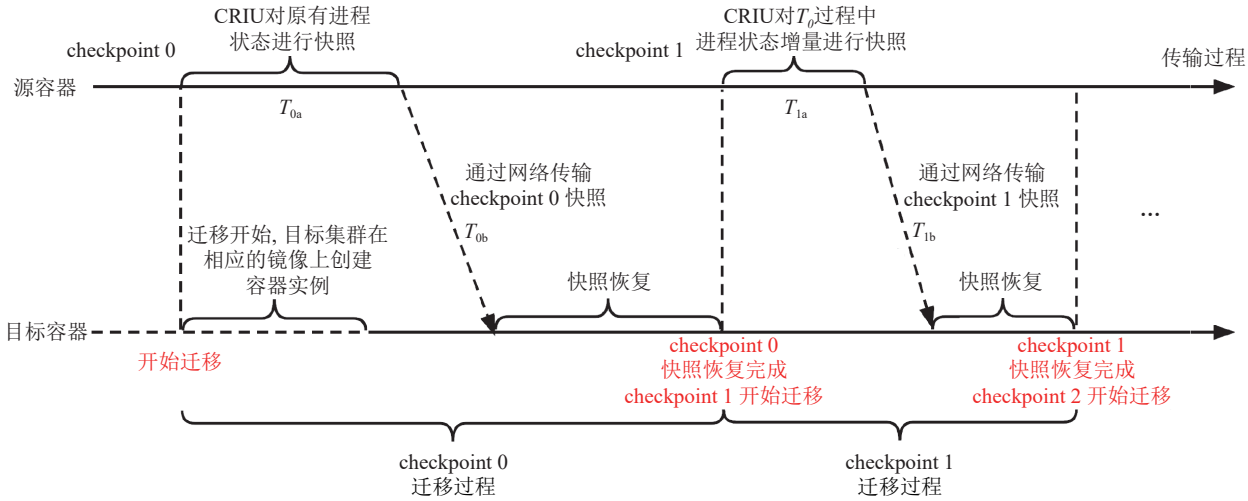


Fig. 1 Illustration of iterate migration process

图 1 迭代迁移过程示意图

根据跨集群迁移的实际情况，网络延迟经常会达到数十至数百毫秒，因此本文模型将网络延迟作为重要参数纳入考虑. 设源集群与目的集群间网络延迟为T_{L}，传输速率为v_{T}. 本模型所用迁移环境参数含义如表 1 所示.

Table 1 Migration Environment Parameters

表 1 迁移环境参数

参数/单位	含义
n	迁移过程总迭代次数
T_x/ms	第 x+1 个迭代过程的单次迭代传输用时
$T_{x\mathrm{a}}$/ms	第 x+1 个迭代过程的快照拍摄用时
$T_{x\mathrm{b}}$/ms	第 x+1 个迭代过程的网络传输用时
$T_{x\mathrm{c}}$/ms	第 x+1 个迭代过程的快照恢复用时
V_x/MB	第 x+1 个迭代过程的状态文件大小
V_0/MB	开始迁移时状态文件大小
ΔV/(MB·ms^{-1})	迁移过程中状态文件的增长速率
v_{P}/(MB·ms^{-1})	快照拍摄速率与恢复速率
K	状态文件的快照压缩率
T_{L}/(MB·ms^{-1})	集群间的网络延迟
v_{T}/(MB·ms^{-1})	集群间的网络传输速率

在本文模型中，我们假设容器迁移过程中两端的源集群和目的集群的性能相同，则可以认为在源容器中拍摄快照的速率与在目的容器中恢复快照的速率相同，即

$$T_{x\mathrm{a}} = T_{x\mathrm{c}}, \tag{1}$$

因此第$x+1$个 checkpoint 迁移过程的用时为

$$T_x = 2T_{x\mathrm{a}} + T_{x\mathrm{b}}, \tag{2}$$

最后 1 次迁移过程，即第n次迁移开始时，源容器停止服务，以防止迁移过程中产生新的进程增量，迁移过程完成后，在目标容器中恢复对用户的服务，整个迁移过程结束. 因此，对用户的服务停机时间即为第n次迁移的用时，可表示为

$$T_{n-1} = 2T_{(n-1)\mathrm{a}} + T_{(n-1)\mathrm{b}}. \tag{3}$$

快照的拍摄速率与恢复速率均为v_{P}，则可计算第$x+1$个快照镜像及恢复的用时为

$$T_{x\mathrm{a}} = T_{x\mathrm{c}} = \frac{V_x}{v_{\mathrm{P}}}, \tag{4}$$

因为在一次连续的网络传输过程中，网络延迟只产生 1 次影响，所以能够得到镜像的网络传输用时:

$$T_{x\mathrm{b}} = \frac{V_x K}{v_{\mathrm{T}}} + T_{\mathrm{L}}. \tag{5}$$

将式(4)和式(5)代入式(2)，可得

$$T_x = 2T_{x\mathrm{a}} + T_{x\mathrm{b}} = V_x\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right) + T_{\mathrm{L}}, \tag{6}$$

特别地，有

$$T_0 = V_0\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right) + T_{\mathrm{L}}, \tag{7}$$

$$T_1 = V_1\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right) + T_{\mathrm{L}}. \tag{8}$$

当$1 \leqslant x \leqslant n-1$时，第$x+1$个进程状态文件大小$V_x$受$T_{x-1}$影响，可以得到:

$$V_x = \Delta V T_{x-1}. \tag{9}$$

将式(9)代入式(6)，可得

$$T_x = \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)T_{x-1} + T_{\mathrm{L}}. \tag{10}$$

令

$$p = \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right), \tag{11}$$

$$q = T_{\mathrm{L}}, \tag{12}$$

则式(10)可化作等比数列:

$$\begin{cases} a_1 = T_0, \\ a_{n+1} = T_n = pT_{n-1} + q = pa_n + q, \end{cases} \tag{13}$$

求其通项公式，可得

$$\begin{cases} a_1 = T_0, \\ a_{n+1} = T_n = p^n\left(a_1 - \dfrac{q}{1-p}\right) + \dfrac{q}{1-p}, \end{cases} \tag{14}$$

将式(12)和式(13)带入式(14)，可解得

$$T_n = T_0\left[\Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)\right]^n + \frac{T_{\mathrm{L}}\left\{1 - \left[\Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)\right]^n\right\}}{1 - \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)}. \tag{15}$$

2.2 迭代迁移的起始和终止条件

实际的跨集群容器迁移场景中，2 个集群性能稳定，因此本文模型以集群间的网络延迟T_{L}的值作为迭代传输是否继续的判断依据.

在该迭代模型中，迭代传输的起始条件依赖的重要指标是T_0与T_1的差值，共存在 3 种情况:

第 1 种情况为满足条件$T_0 - T_1 < 0$，此时有$T_0\Delta V < T_1\Delta V \Rightarrow V_1 < V_2$，则每次迭代传输过程中需要传输的状态文件大小是递增的，即每次迭代传输消耗的时间也是单调递增的，传输过程总用时和对用户的停机时间会随着迭代次数的增加越来越高，迭代传输产生负收益. 因此应当通过一次传输完成整个迁移过程，即迭代次数为 1，服务停机时间为第 1 次的传输时间，即$T = T_0$.

第 2 种情况为满足条件$T_0 - T_1 = 0$，此时有$T_0\Delta V =$

$T_1\Delta V \Rightarrow V_1 = V_2$, 则每次迭代传输过程消耗的时间相同, 迭代并没有起到缩短单次传输时间的效果, 反而增加了集群的其他方面性能压力以及传输总时间. 因此与第 1 种情况相同, 在该情况下也应当通过一次传输完成整个迁移过程, 即迭代次数为 1, 服务停机时间为第 1 次的传输时间, 即$T = T_0$.

第 3 种情况为满足条件$T_0 - T_1 > 0$, 此时有$T_0\Delta V > T_1\Delta V \Rightarrow V_1 > V_2$, 每次迭代的传输过程消耗时间是单调递减的, 迭代传输将缩短单次传输的消耗时间, 对用户的停机时间随迭代次数的增加而缩短, 迭代迁移的使用产生收益, 此时应当使用迭代迁移方法. 因此迭代迁移的起始条件为$T_0 - T_1 > 0$.

令$T_0 - T_1 > 0$, 将其带入式(10), 可得

$$T_0 > \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)T_0 + T_{\mathrm{L}}, \tag{16}$$

解此不等式, 得

$$T_{\mathrm{L}} < T_0\left[1 - \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)\right]. \tag{17}$$

因此, 当集群间网络延迟T_{L}满足式(17)条件时, 容器的迁移应采用迭代的方式. 将式(7)带入式(17), 可得

$$T_{\mathrm{L}} < \frac{V_0}{\Delta V}\left[1 - \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)\right], \tag{18}$$

式(18)即为迭代迁移的起始条件. 当T_{L}的值不满足式(18)时, 则不采用迭代迁移方法, 通过 1 次迁移完成容器的迁移过程, 也可认为是迭代次数为 1 的迭代迁移过程.

当满足条件需要进行迭代传输时, 需要确定传输过程的迭代次数. 本文认为迭代迁移的停止条件为第x次迁移时间与第$x+1$次迁移时间的差不大于网络延迟T_{L}时, 网络延迟时间成为影响单次迭代时间的主要原因, 且不受迭代过程影响, 因此此时应停止迭代过程完成传输. 即有当

$$|T_x - T_{x-1}| \leqslant T_{\mathrm{L}} \tag{19}$$

时, 停止迭代过程完成传输, 总迭代次数为x. 在满足迭代传输的条件下, 一定有$T_{x-1} > T_x$, 因此式(19)可展开为

$$T_{x-1} - T_x \leqslant T_{\mathrm{L}}, \tag{20}$$

因此式(20)为迭代迁移的终止条件.

2.3 迁移过程迭代次数的计算

将式(10)带入式(20), 得到

$$T_{x-1} - \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)T_{x-1} - T_{\mathrm{L}} \leqslant T_{\mathrm{L}}, \tag{21}$$

将T_n的通项式(15)带入式(21), 解得

$$x \geqslant \log_{\Delta V\left(\frac{2}{v_{\mathrm{P}}}+\frac{K}{v_{\mathrm{T}}}\right)}\frac{T_{\mathrm{L}}}{T_0\left[1 - \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)\right] - T_{\mathrm{L}}} + 1. \tag{22}$$

根据式(17), 结合网络延迟的物理意义可知, 网络延迟T_{L}为大于 0 的实数, 因此有

$$0 < T_0\left[1 - \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)\right], \tag{23}$$

即

$$0 < \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right) < 1. \tag{24}$$

同时结合迭代次数n的物理意义, n为不小于 1 的正整数. 令总迭代次数n的值为式(22)结果向上取整, 可得n的计算公式为:

当

$$\frac{T_{\mathrm{L}}}{T_0\left[1 - \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)\right] - T_{\mathrm{L}}} \geqslant 1 \tag{25}$$

时, $n = 1$;

当

$$0 < \frac{T_{\mathrm{L}}}{T_0\left[1 - \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)\right] - T_{\mathrm{L}}} < 1 \tag{26}$$

时,

$$n = \left\lceil \log_{\Delta V\left(\frac{2}{v_{\mathrm{P}}}+\frac{K}{v_{\mathrm{T}}}\right)}\frac{T_{\mathrm{L}}}{T_0\left[1 - \Delta V\left(\frac{2}{v_{\mathrm{P}}} + \frac{K}{v_{\mathrm{T}}}\right)\right] - T_{\mathrm{L}}} \right\rceil + 1. \tag{27}$$

3 实验设置

3.1 实验环境设置

本文提供的迭代迁移算法主要考虑网络延迟对迁移过程的影响, 因此在实验环境的设置中保证了 2 端集群和容器的硬件性能稳定, 将集群间网络延迟的变化作为实验结果的重要影响因素.

本文将实验环境设置为较为典型的云平台性能条件, 令开始迁移时的状态文件大小V_0=100 MB, 迁移过程中状态文件的增长速率ΔV=0.03 MB/ms, checkpoint 快照拍摄速率与恢复的速率v_{P}均为 0.1 MB/ms, 状态文件快照的压缩率K=68%, 集群间的网络传输速率v_{T}=0.2 MB/ms.

在本实验环境中, 认为容器迁移过程中容器性能保持稳定, 容器各性能的值在实验过程中不发生变化.

3.2 评价指标

针对本文主要研究目标, 为验证本文所提出的

迭代迁移方法的贡献，评价指标根据迭代迁移方法是否能够缩短服务停机时间和能够将服务停机时间缩短多少这 2 方面进行划分：

1）当集群间网络延迟满足式(18)时，通过迭代迁移的方式能否缩短服务停机时间；

2）在不同的网络延迟条件下，采用迭代迁移方法能够将服务停机时间缩短多少.

根据本文所建立模型，不采用迭代迁移的一次性迁移方式，也可看作是迭代次数为 1 的迭代迁移方式，其服务停机时间即为第 1 次迭代迁移过程的时间T_0. 迭代迁移过程的服务停机时间即为最后一次迭代迁移过程的时间T_{n-1}. 在本模型中，采用二者的比值

$$R=\frac{T_{n-1}}{T_0}, \tag{28}$$

来表示迭代迁移方法缩短服务停机时间的效率，R 值越低效率越高.

4 实验结果分析

4.1 迭代迁移有效性分析

根据迭代传输的起始条件式(18)，将实验环境中各参数的值带入，计算得到满足条件的网络延迟$T_L<993.33$. 为验证迭代传输判定条件的有效性，分别取 5 种不同网络延迟T_{L_1}=1 200 ms，T_{L_2}=1 100 ms，T_{L_3}=993 ms，T_{L_4}=900 ms 和T_{L_5}=800 ms. 其中T_{L_1}和T_{L_2}不满足式(18)条件，T_{L_3}的值与起始条件值相近，T_{L_4}和T_{L_5}满足式(18)条件. 计算并绘制 5 个条件下迭代过程的单次迭代传输时间T_n随迭代次数n的变化情况如图 2 所示.

从结果中能够看出，当T_L与式(18)计算所得迭代传输起始条件T_L的值相同时，每次迭代的时间几乎没有变化. 图中T_{L_3}=993 ms 时，其对应的T_0=3 333 ms 而后续单次迭代传输时间为 3 332 ms，主要原因为本文假设T_{L_3}=993 ms 与起始条件 T_L=993.33 ms 并非完全相同.

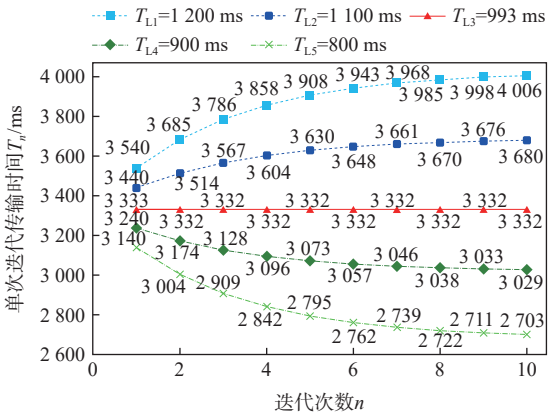


Fig. 2 Effect of the network latency on the iterative transmission time

图 2 网络延迟对迭代传输时间的影响

T_{L_1}与T_{L_2}大于式(18)计算所得迭代传输起始条件的值时，每次迭代传输的耗时是单调递增的. 综上可知，在网络延迟满足T_{L_1}=1 200 ms，T_{L_2}=1 100 ms，T_{L_3}=993 ms 这 3 种情况时，迭代传输并没有缩短单次传输时间，也就是服务的停机时间，同时随着迭代次数的增加，也会增加对集群的性能压力和传输总时间.

T_{L4}与T_{L5}满足式(18)计算所得迭代传输起始条件的值时，能够看到每次迭代传输的耗时是单调递减的，此时迭代传输缩短了单次传输时间，达到了缩短服务停机时间的目的.

我们取 3 个不同的满足迭代传输起始条件的网络延迟的值，分别为T_{L6}=900 ms，T_{L7}=600 ms，T_{L8}=300 ms，计算并绘制单次迭代传输时间T_n随迭代次数n的变化情况如图 3 所示.

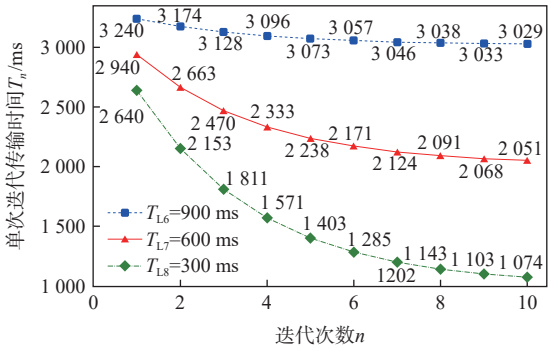


Fig. 3 Effect of the network latency when the iteration conditions are met

图 3 满足迭代条件时网络延迟的影响

结合图 2 与图 3 所示，可以从中得出 2 点结论：

1）当网络延迟T_L满足迭代传输起始条件时，传输过程中单次迭代传输时间随迭代次数的增加而缩短，迭代传输能够达到缩短服务停机时间的目的；

2）网络延迟T_L与起始条件值相差越大，相邻 2 次迭代间减少的传输时间越多，最终停止迭代时单次传输时间更低，即服务停机时间越短，迭代传输的收益越大.

4.2 迭代迁移收益分析

根据式(25)可知，当

$$\frac{T_L}{T_0\left[1-\Delta V\left(\frac{2}{v_P}+\frac{K}{v_T}\right)\right]-T_L}\geqslant 1$$

时, $n=1$, 化简可得

$$T_{\mathrm{L}} \geqslant \frac{T_0}{2}\left[1-\Delta V\left(\frac{2}{v_{\mathrm{P}}}+\frac{K}{v_{\mathrm{T}}}\right)\right], \tag{29}$$

将式(7)带入式(29), 计算并化简后解得

$$T_{\mathrm{L}} \geqslant \frac{V_0\left(\frac{2}{v_{\mathrm{P}}}+\frac{K}{v_{\mathrm{T}}}\right)\left[1-\Delta V\left(\frac{2}{v_{\mathrm{P}}}+\frac{K}{v_{\mathrm{T}}}\right)\right]}{1+\Delta V\left(\frac{2}{v_{\mathrm{P}}}+\frac{K}{v_{\mathrm{T}}}\right)}. \tag{30}$$

将模型中的各项参数值带入式(30), 解得当网络延迟 $T_{\mathrm{L}} \geqslant 409.71$ ms 时, 总迭代传输次数为 1, 即因为每次传输过程受网络延迟 T_{L} 影响较大, 通过 1 次传输完成容器迁移.

当网络延迟 $T_{\mathrm{L}} < 409.71$ ms 时, 迭代次数将大于 1, 可通过式(27)计算总迭代次数 n. 根据计算结果, 绘制总迭代次数随网络延迟 T_{L} 变化的图像如图 4 所示.

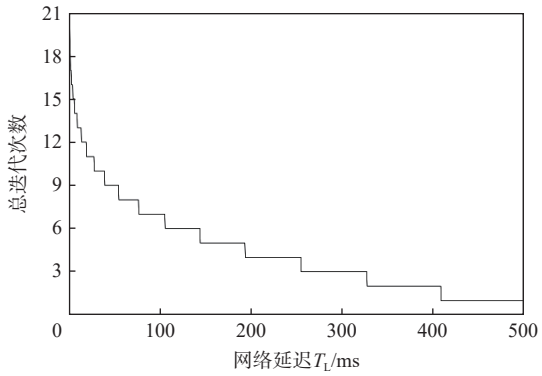


Fig. 4 Trend plot of the number of iterations varying with the network latency

图 4 迭代次数随网络延迟变化趋势图

从图 4 中能够看出, 总迭代次数是随着集群间网络延迟 T_{L} 的增加而不断降低. T_{L} 增加单次传输时间的影响越来越大, 收益越来越低, 因此总迭代次数不断下降, 直至总迭代次数为 1.

实际环境中, 跨集群间的网络延迟通常在 30~500 ms 之间, 可以看到本文建立的迁移模型在这个延迟区间的总迭代次数变化平稳.

在得到不同网络延迟对应的总迭代次数的基础上, 结合式(28)计算不同网络延迟时迭代迁移方法能够提供的服务停机时间缩短的效率. 在网络延迟为 100~400 ms 时, 以 50 ms 作为间隔, 对服务停机时间缩短效率进行分析. 从表 2 中可以看出, 随着集群间网络延迟的降低, 总迭代次数增加会使服务停机时间缩短的效率提高. 在网络延迟足够低时, 甚至可将服务停机时间缩短至不使用迭代迁移方法停机时间的 1/4 以内.

Table 2 Features Comparison of the Main Strategies with TLP Enhancing

表 2 提升 TLP 的主要策略的特点对比

T_{L}/ms	T_0/ms	总迭代次数	T_{n-1}/ms	R
100	2 440.0	7	587.5	0.24
150	2 490.0	5	985.8	0.40
200	2 540.0	4	1 317.7	0.52
250	2 590.0	4	1 444.7	0.56
300	2 640.0	3	1 811.6	0.69
350	2 690.0	2	2 238.4	0.83
400	2 740.0	2	2 323.5	0.85

4.3 真实环境应用可能产生的问题

在真实的有状态容器跨集群迁移过程中, 网络延迟和集群性能并非稳定, 从 4.2 节的分析结果来看, 一次完整的迁移过程可能会持续数秒甚至数十秒, 这个时间对于远距离的网络传输来说, 很难保证性能持续稳定.

针对传输性能不稳定对传输过程造成的影响, 我们分别从网络延迟和集群性能 2 个方面来讨论.

1)在集群性能稳定的情况下, 网络延迟 T_{L} 的波动会对单次迭代传输时间 T_x 产生影响, 进而会影响到下一次迭代过程中的状态文件大小, 对后续的传输过程产生影响. 当传输过程中 T_{L} 增大时, 会使下一次迭代过程中的状态文件大小 V_x 变大, 导致单次迭代传输时间变长. 但是由于网络延迟波动造成的意外情况, T_{L} 会随着后续的迭代过程再次缩短, 最终导致的结果可能是, 按照确定好的迭代次数完成整个容器迁移过程时, 并不能满足迭代过程的终止条件 $T_{x-1}-T_x \leqslant T_{\mathrm{L}}$.

同时, 网络延迟的变化存在着另一种变化方式, 即传输过程中网络延迟 T_{L} 变小. 这种情况可能会导致在达到确定好的迭代次数之前即满足终止条件 $T_{x-1}-T_x \leqslant T_{\mathrm{L}}$. 产生的后果是虽然已经满足了终止条件, 却仍然在进行迭代过程, 直至达到迁移过程开始前确定好的迭代次数为止. 这种情况下, 会造成网络资源和集群性能的浪费.

2)集群性能的波动对传输过程的影响与网络延迟的影响类似, 集群性能同样影响到当前传输过程的单次迭代传输时间 T_x. 因此集群性能的降低所产生的影响与网络延迟增大所产生的影响一样, 而集群性能的提高则与网络延迟降低对迭代迁移过程产生的影响一致.

本文方法中总迭代次数是根据性能参数在容器迁移过程开始之前确定的，因此迁移过程中性能的波动不会对总迭代次数产生影响，但会影响到后续的单次迭代传输过程的时长，可能会造成最后一次迭代过程不能满足终止条件，以及提前满足终止条件而对网络和集群性能造成浪费的情况.

5 结　论

本文提出了一种考虑集群间网络延迟的迭代迁移算法，提高了有状态容器跨集群迁移效率，缩短服务停机时间. 根据提出的迁移方法确定迁移过程的总迭代次数，并通过实验分析了迭代迁移对服务停机时间的缩短效率，证明了本文提出方法能够有效大幅度缩短有状态容器跨集群迁移过程导致的服务停机时间.

未来我们还将继续完善迭代迁移方法、优化方向有 2 个：1)考虑优化迭代迁移过程，通过改进迭代方法提高迁移效率；2)根据实际迁移情况，考虑其他的迭代迁移终止条件. 对真实环境容器迁移过程中的性能波动更进一步分析，定量地研究迁移过程中网络延迟和集群性能对迁移过程终止条件的影响.

作者贡献声明：吴宪负责模型设计并完成论文撰写；汤红波改进模型，确定迭代迁移方式的收益计算方法并指导实验；赵宇提出实验方法并审核实验结果；许明艳指导论文写作，并设计和修改论文的插图.

参　考　文　献

[1] Wu Song, Wang Kun, Jin Hai. Research situation and prospects of operating system virtualization[J]. Journal of Computer Research and Development, 2019, 56(1): 58−68 (in Chinese)
(吴松, 王坤, 金海. 操作系统虚拟化的研究现状与展望 [J]. 计算机研究与发展, 2019, 56(1): 58−68)

[2] Ogbuachi M C, Reale A, Suskovics P, et al. Context-aware kubernetes scheduler for edge-native applications on 5G[J]. Journal of Communications Software and Systems, 2020, 16(1): 85−94

[3] Zu Jiachen, Hu Guyu, Yan Jiajie, et al. Resource management of service function chain in NFV enabled network: A survey[J]. Journal of Computer Research and Development, 2021, 58(1): 137−152 (in Chinese)
(祖家琛, 胡谷雨, 严佳洁, 等. 网络功能虚拟化下服务功能链的资源管理研究综述 [J]. 计算机研究与发展, 2021, 58(1): 137−152)

[4] Lane-walsh S, Stillerman J, Santoro F, et al. Introduction to MDSplus using docker[J]. Fusion Engineering and Design, 2021, 165: 112121

[5] Pickartz S, Eiling N, Lankes S, et al. Migrating Linux containers using CRIU[C]// LNCS 9945: Proc of the 2016 ISC High Performance Workshops. Berlin: Springer, 2016: 674–684

[6] Red Hat. CRIU - Checkpoint/Restore in user space[EB/OL]. [2022-06-12].https://access.redhat.com/articles/2455211

[7] Yin Luxiu, Li Pengfei, Luo Juan. Smart contract service migration mechanism based on container in edge computing[J]. Journal of Parallel and Distributed Computing, 2021, 152(1): 157−166

[8] Venkatesh R S, Smejkal T, Milojicic D S, et al. Fast in-memory CRIU for docker containers[C] //Proc of the Int Symp on Memory Systems. New York: ACM, 2019: 53−65

[9] Xu Bo, Wu Song, Xiao Jiang, et al. Sledge: Towards efficient live migration of docker containers[C]//Proc of the 13th IEEE Int Conf on Cloud Computing (CLOUD). Piscataway, NJ: IEEE, 2020: 321-328

[10] Lei Zhou, Sun Exiong, Chen Shengbo, et al. A novel hybrid-copy algorithm for live migration of virtual machine[J]. Future Internet, 2017, 9(3): 37−50

[11] Lv Liang, Zhang Yuchao, Li Yusen, et al. Communication-aware container placement and reassignment in large-scale internet data centers[J]. IEEE Journal on Selected Areas in Communications, 2019, 37((3):): 540−555

[12] Di Zhanyuan, Shao En, Tan Guangming. High-performance migration tool for live container in a workflow[J]. International Journal of Parallel Programming, 2021, 49(5): 658−670

[13] Zhao Qian, Xie Shangqin, Han Ke, et al. Container migration based on combination of remote direct memory access and check point[J]. Journal of Frontiers of Computer Science and Technology, 2019, 13(12): 1995−2007 (in Chinese)
(赵倩, 谢上钦, 韩轲, 等. 远程直接内存访问与检查点相结合的容器迁移 [J]. 计算机科学与探索, 2019, 13(12): 1995−2007)

[14] Stoyanov R, Kollingbaum M J. Efficient live migration of Linux containers[C]// Proc of the 2018 ISC High Performance Workshops. Berlin: Springer, 2018: 184−193

[15] Junior P S, Miorandi D, Pierre G. Stateful container migration in geo-distributed environments[C/OL]//Proc of the 12th IEEE Int Conf on Cloud Computing Technology and Science (CloudCom). Piscataway, NJ: IEEE, 2020: 49−56



Wu Xian, born in 1989. Master candidate. His main research interest includes mobile communication networks and security.
吴　宪, 1989 年生. 硕士研究生. 主要研究方向为移动通信网络与安全.



Tang Hongbo, born in 1968. Professor, PhD supervisor. His main research interest includes mobile communication networks and security.
汤红波, 1968 年生. 教授, 博士生导师. 主要研究方向为移动通信网络与安全.



Zhao Yu, born in 1984. Assistant professor. His main research interest includes mobile communication networks and security.

赵　宇, 1984年生. 助理研究员. 主要研究方向为移动通信网络与安全.



Xu Mingyan, born in 1974. Associate professor. Her main research interest includes mobile communication networks and security.

许明艳, 1974年生. 副研究员. 主要研究方向为移动通信网络与安全.