一种基于机器学习的在线容器迁移预判方法

李　岩
深圳证券交易所　广东　深圳　518000

摘　要：在过去的十年里见证了容器等云计算技术的重要发展，移动边缘计算（Mobile Edge Computing，MEC）中的网络和计算环境的资源局限性给容器服务实时迁移切换带来了重大挑战，导致了容器在线迁移的代价过高给用户带来不好的体验，甚至是迁移失败，因此在切换前对环境的预判，从而减小容器的downtime，减少网络开销，从而改善用户体验，保证切换成功率就显得尤其有意义。本文提出了一种机器学习方法用于容器在线迁移代价的预判，从而识别出不好的迁移决策。目标是通过降低迁移能耗和在线迁移的迭代次数来优化迁移过程的选择。实验结果表明，所提出的解决方案可以显著识别出风险和代价较高的迁移决策。

关键词：机器学习；容器迁移；云计算

引言

广泛应用的云计算技术是当代信息技术的重要进步之一，云计算背后的主要思想是可伸缩性，而使这一目标成为可能的关键技术之一是容器化。Docker是一种容器化运行技术或平台，它以容器的形式将应用程序及所有的依赖项打包在一起，以确保你的应用程序在任何环境中无缝运行。

移动边缘计算（Mobile Edge Computing, MEC）可利用无线接入网络就近提供用户所需的云端计算功能，让消费者享有不间断的高质量网络体验。边缘计算的主要挑战之一是保持比传统云服务相当的服务质量保证，同时将服务卸载到最终用户最近的边缘服务器。然而，当最终用户离开附近的边缘服务器时，由于网络连接恶化，服务质量将显着下降。理想情况下，当最终用户移动时，边缘服务器上的服务也应该实时迁移到附近的新服务器。因此，成功率高且高效的实时迁移对于在边缘计算环境中实现边缘服务的移动性至关重要。[1]

迁移是一种有效的重构数据中心性能的策略，改变了数据中心的组成和运行方式以便更高效地共享和控制数据中心资产。容器迁移技术在容器仍在运行时将容器从一个物理主机转移到另一个物理主机，它特别适用于现代数据中心，使计算动态、灵活、平台独立和有效地利用和共享资源。此外，在数据中心内部/跨数据中心的容器迁移是为了电源管理、负载平衡、可用性和降低SLA违规率[1]。为此，容器迁移过程需要传输CPU状态、内存状态、网络状态和磁盘状态。

一般来说，现有的容器迁移优化技术可分为压缩（compress）、去重（deduplication）和检查点（checkpoint）[2]。此外，还有许多不同的容器迁移策略，如pre-copy和post-copy。然而，没有任何一种技术尝试将机器学习应用于容器迁移，这促使我们去思考一个基于机器学习模型来解决手头的问题。

本文的具体贡献在于，提出一种基于机器学习模型，以在容器的在线迁移时进行预判，从而尽量避免出现失败或者用户体验很差（SLA违规，如迁移时间过长）的容器迁移。

1 相关工作

很多文献研究了云计算中的动态资源分配，其中的一个领域的重要研究趋势是容器迁移问题[3,4]。在[5]中提出的方法是用于管理容器集群根据其内存和CPU参数的值。然后，它处理同一台物理机器上、共享内存页面的容器迁移问题，然后迁移决策是基于对整个系统的评估而做出的。

2 问题描述

容器系统中的后端服务器是分布式的，并配备了云计算资源，容器从一个物理服务器移动到另一个。在这项工作中，我们针对pre-copy（预复制容器迁移）类型来讨论问题；试图通过传输更少的数据来保持更短的停机时间停止和复制阶段。容器迁移需要考虑CPU状态，内存状态，网络状态，和磁盘状态。在此期间，容器将在源主机上checkpoint，并在目标主机上restore。因此复制内存页面，确保容器在恢复后在目的服务器处于相同的运行环境中）。容器迁移是根据云环境的状态和执行期间每个容器的状态产生的任务，因此，容器迁移应考虑以下属性：

到达服务率（ASR, Arrival Service Rate）每秒钟的新产生的服务请求数，比如nginx的http/https的每秒请求数；

（1）容器脏数据比率（DR, Dirty Ratio），当容器中

的内存数据页和磁盘数据页上的内容不一致时，我们称这个内存页为脏页；内存数据写入磁盘后，内存页上的数据和磁盘页上的数据就一致了，此时这个内存页成为干净页；DR = dirty_memory_size / total_memory_size,

（2）集群负载平衡率（LB），对于提供容器服务来说，每个后端的服务处理流量可能不同，这个差异代表了集群中服务的负载平衡率，具体的可以用以下的公式来计算：(max_flow_rate-min_flow_rate)/max_flow_rate

（3）源主机CPU利用率（SCU，SRC_CPU_UTILIZATION）

（4）源主机内存的使用率（SMU，SRC_MEM_UTILIZATION）

（5）源主机IO资源的可用性（SDU，SRC_IOSTAT_UTILIZATION）

（6）源主机网络资源的可用性（SNU，SRC_IOSTAT_UTILIZATION）

（7）目标主机的CPU利用率（DCU，DST_CPU_UTILIZATION）

（8）目标主机已分配的内存（DMU，DST_MEM_UTILIZATION）

（9）目标主机IO资源的可用性（DDU，DST_IOSTAT_UTILIZATION）

（10）目标主机网络资源的可用性（DNU，DST_NET_BW_UTILIZATION）

3 容器迁移的机器学习模型和实验

我们所提出的迁移策略是基于两个主要阶段。第一个阶段是创建一个学习模型并使用数据进行训练，然后第二个阶段是使用这个模型来进行容器迁移的决策（如下图1）。通过海量的容器迁移数据，监督学习方法来训练模型，待模型收敛后直接应用于在线迁移的风险分类。

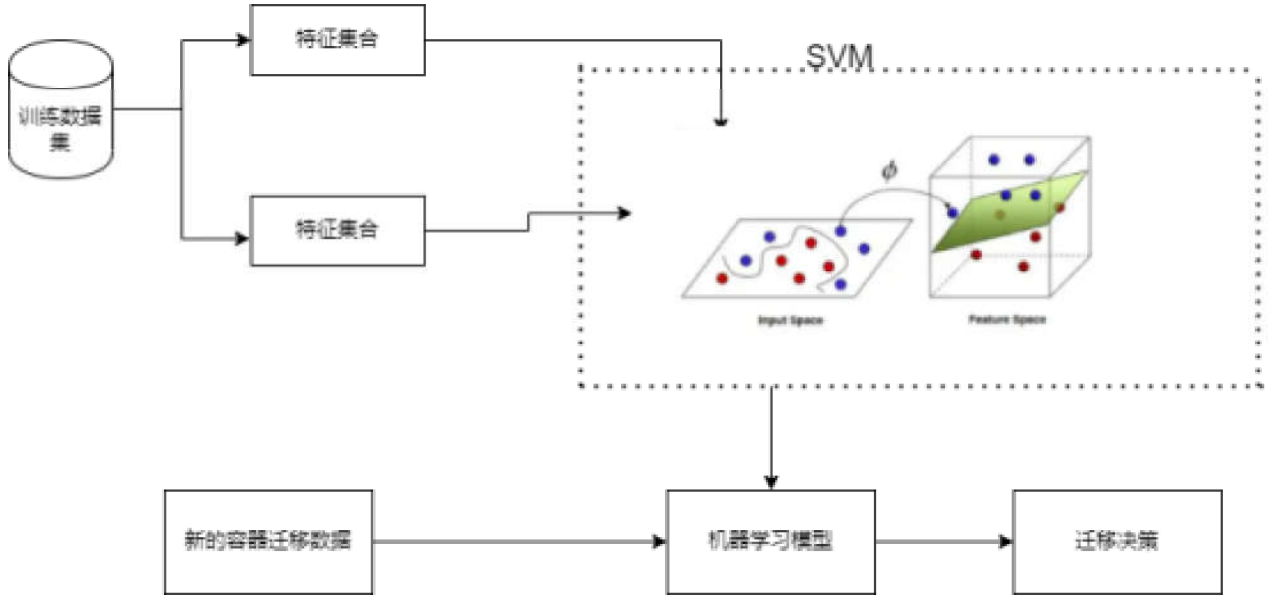

图1 容器迁移决策的机器学习模型

3.1 机器学习准备阶段

本节解释了创建能够在下一阶段中使用的训练模型的基本步骤。这是一个这样的过程：首先准备有关容器迁移环境的原始数据，并使其适合于构建机器学习模式并以格式化的方式放置数据的强制条件。

3.2 特征选择

在选择数据之后，我们需要提取适合于机器学习的适当属性并做出预处理，使之适合从原始数据中提取特征的过程，以使算法得以应用。这可以通过给研究一组m个实例的项目s {(x1, y1), (x2, y2), ...,(xm, ym)}，来构建假设的分类器G：x→y，将一个特征值集合x∈X映射到一个类标签y∈Y。因此，这个阶段包括选择合适的算法（比如XgBoost、决策树、SVM等）来构建分类器。我们主要采用了SVM（support Vector Machine）的监督学习算法。支持向量机SVM可以分为线性核非线性两大类。其主要思想为找到空间中的一个更够将所有数据样本划开的超平面，并且使得本本集中所有数据到这个超平面的距离最短。比如给定一系列的数据样本，每个样本都有对应的一个标签。为了使得描述更加直观，我们可以很简单地找到一条直线使得这两类不同标签的数据正好能够完全分开。

3.3 标签生成

在监督学习中，我们把成功的迁移实例的标签定义为Good(1)，不成功的迁移实例的标签定位为Bad(0)。通

过海量的容器迁移数据，监督学习方法来训练模型，待模型收敛后直接应用于在线迁移的风险分类。

3.4 学习模型（LM）的创建

在从训练数据中创建分类器后，对其性能进行评估并使用独立的测试数据（也称为验证数据）进行预测，以确定它对未知实例进行决策的准确性

3.5 数据分组和交叉验证

分类器的性能通过交叉验证技术来评估的子集的质量分数特性的。交叉验证对不同数据子集的性能度量，并结合预测提供了更准确的模型性能评估。我们的模型将在最终稳定收敛之前在这个样本上进行测试。

在这里，已经使用了k-fold方法来进行交叉验证。这种技术有助于减少差异和避免过度实现，并深入了解模型在实践中的一般行为。在k倍交叉验证中，数据集D被随机分成k个大小大致相似的子集（fold）：D1、D2、...，Dk。在每个试验中，部分数据子集为测试集，其余的数据成为训练集。换句话说，我们重复这个过程K次，每个子集（fold）都是被选择的测试集，其余的k-1个子集构成训练集。

机器学习模型评估之后，我们得到了分类统计数据的概述，如错误实例的数量。记录了在每个预测中看到的误差。这些统计数据被恢复为单个统计数据值，这是精度Ai。最终精度A是每轮精度除以回合数（参考等式1和等式2）。

$$A = \frac{1}{\text{Rounds}} \sum A_i \quad (1)$$

$$A_i = \frac{TN + TP}{TN + TP + FN + FP} \quad (2)$$

其中

-真阳性（TPs）：是判断出来适合容器迁移的情况，实际迁移也成功的场景；

-真否定（TNs）：是判断出来适合容器迁移的情况，实际迁移失败的场景；

-假阳性（FPs）：是判断出来不适合容器迁移的情况，实际迁移却成功的场景；

-假阴性（FNs）：是判断出来不适合容器迁移的情况，实际迁移也失败的场景；

3.6 数据集

通过对我们使用docker17.03搭建的实验环境中的数据集进行收集容器迁移的7652次容器原始数据，形成了测试数据集，其中有227次失败的迁移实例。并把这些数据集按70%和30%的比例划分为训练集和测试集。

3.7 实验结果（如下表2）

表2 机器学习的测试数据集的实验结果

环境	准确度	标签	精确度	召回率	F1分数	样本数
Docker 17.03	0.8691	1	0.98	0.88	0.93	7425
		0	0.12	0.51	0.19	227

通过这个SVM的运行结果来看，我们的机器学习模型在标签为1的精确性方面表现还是比较满意的，但是在标签为0的精确性方面的表现有明显的不理想，这个可能是样本数据不够丰富导致，一定程度上限制了模型的应用。相信随着数据集的补充，模型的准确度可以得到提升，从而提升模型的实用价值。

4 结论

本文提出了一种新颖的机器学习模型，根据容器迁移的环境特征，进行容器迁移的成功率预判，如果成功率过低或者因迁移代价过大导致迁移的用户体验太差，则会给出不适合迁移的提示。这种模型在资源受限且主题在不断移动的边缘计算场景下尤为重要。

参考文献

[1]Y. Wang等，针对具有多层应用程序的虚拟化数据中心的性能控制服务器整合，可持续性计算：信息科学与系统4(1)(2014),52–65

[2]边缘计算中的容器实时迁移：实时性能改进，应用科学与工程杂志，19(3)(2022),1~8

[3]使用用户空间中的检查点和恢复的实时迁移（CRIU）：网络、内存和CPU的使用情况分析，电气工程与信息公报,10(2)(2021),837~847

[4]一种基于资源位置的容器实时迁移算法，W. Fan等，信号处理系统杂志91(4)(2019),1~13

[5]计算中虚拟机迁移的智能弹性调度算法,H. Nashaat等，超级计算杂志75(7)(2019),3842–3865