软件定义网络中数据中心虚拟机迁移序列问题的研究

史久根 许辉亮* 陆立鹏
(合肥工业大学计算机与信息学院 合肥 230009)

摘 要：虚拟机迁移是数据中心提供的重要功能之一，可以有效地均衡各个基础设施中的工作负载。为有效地减少虚拟机迁移的总时间和对服务性能的影响，该文提出基于代价评估的启发式算法(Heuristic Algorithm based on Cost Evaluation, HACE)。算法在虚拟机迁移的每一步中综合考虑网络中的剩余带宽和迁移时间，通过有机结合并行算法和启发式算法，解决软件定义网络中数据中心大量虚拟机同时迁移时的迁移序列问题。算法在保证安全、依赖关系和性能要求的同时，减少虚拟机的总迁移时间。实验结果表明，与贪心算法相比，该算法能够减少虚拟机总迁移时间达到 52.1%，提高迁移性能，确保服务质量。

关键词：软件定义网络；虚拟化；虚拟机迁移；迁移序列；资源分配
中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2017)05-1193-07
DOI: 10.11999/JEIT160792

Research on the Migration Queue of Data Center's Virtual Machine in Software Defined Networks

SHI Jiugen XU Huiliang LU Lipeng
(*School of Computer and Information*, *Hefei University of Technology*, *Hefei* 230009, *China*)

Abstract: Virtual machine migration is one of the important features of the data center, which can effectively balance the workload of each infrastructure. In order to reduce the total time of virtual machine migration and impact on service performance, a Heuristic Algorithm based on Cost Evaluation (HACE) is proposed in this paper. The proposed algorithm considers both the residual bandwidth of the network and migration time in every step of the virtual machine migration. And through organic combination of parallel algorithm and heuristic algorithm, it solves migration sequence problem of numerous virtual machines in Software Defined Network (SDN). The algorithm reduces the total migration time of the virtual machine while ensuring the security, dependence and performance requirements. Comparing with the greedy algorithm, experiments show that the algorithm can reduce the total migration time of the virtual machine by up to 52.1%, improve the migration performance and ensure the quality of service.

Keywords: Software Defined Networks (SDN); Virtualization; Virtual machine migration; Migration sequence; Resource allocation

1 引言

随着云计算和软件定义网络(Software Defined Networks, SDN)[1]的发展，虚拟化已经成为数据中心的基础技术，可以有效地减少运行成本、最大化资源利用率、提高性能和可靠性。在不影响服务的前提下，虚拟机在线迁移将服务从源主机迁移到目的主机继续运行，以达到动态负载均衡、在线系统维护等目的。然而，虚拟机迁移仍然是一个具有挑战性的任务，因为虚拟机迁移占用源主机和目的主机的 CPU、内存、网络带宽等资源，一定程度上影响云环境中其他服务的性能；另一方面，虚拟机迁移持续时间较长，且有少量的宕机时间，会使服务产生短暂中断，部分地影响服务质量和用户体验。而金融服务、社交网络、推荐系统和网络搜索服务等不能容忍服务性能的降低和网络错误的出现。

为了减轻这些影响，一个可行的解决方法是找到有效的虚拟机迁移序列，减少虚拟机迁移的总时间。文献[2]最先对虚拟机迁移序列问题进行研究，提出了熵(Entropy)，对数据中心的资源进行动态管理。作者通过实验发现，虚拟机迁移的停机时间与内存的大小有重要关系。文献[3]中强调了虚拟机迁

收稿日期：2016-07-26；改回日期：2017-01-06；网络出版：2017-02-28
*通信作者：许辉亮 xuhuiliang@mail.hfut.edu.cn
基金项目：国家重大科学仪器设备开发专项(2013YQ030595)
Foundation Item: The National Major Scientific Instruments Development Project (2013YQ030595)

移序列对虚拟机迁移性能的影响，并提出了CQNCR算法优化虚拟机迁移序列问题。而文献[4,5]在能耗方面研究了虚拟机迁移问题，并提出相应的迁移计划和模型来减少虚拟机迁移的能量消耗。文献[6]在软件定义网络中研究虚拟机迁移序列问题，提出了启发式的算法，实现 Openflow 网络中虚拟机的迁移，主要目的是发现一个可行的虚拟机迁移序列。文献[7]中研究了虚拟集群中在线迁移的性能和负载，强调了网络中带宽的缺乏和找出最优的迁移序列对网络带宽充分利用的必要性。文献[8, 9]对虚拟网中带宽的调度算法进行了研究。

本文提出了基于代价评估的启发式算法(HACE)来解决软件定义网络中数据中心虚拟机迁移序列问题。该算法综合考虑网络中的剩余带宽和虚拟机迁移需要的时间，在确保安全性、虚拟机迁移之间的依赖关系和服务性能要求的同时，找出一个较优的虚拟机迁移序列。另外，HACE 在迁移的每一步中都考虑了剩余带宽，在满足网络资源约束的前提下，最大化并行迁移的数量。更重要的是，本文的算法可以容易地应用到虚拟机(Virtual Machine, VM)或虚拟数据中心(Virtual Data Center, VDC)的管理架构中，如 Sandpiper[10], VDC Planner[11]和 SecondNet[12]等，进一步地提高网络性能和服务质量。

2 预拷贝迁移算法

虚拟机在线迁移实现将虚拟机从一台物理机迁移到另外一台物理机，迁移过程不影响虚拟机的正常运行，也不会影响虚拟机上的业务。已有大量的工作用于虚拟机在线迁移技术的研究[13,14]。虚拟机在线迁移的目的是尽可能地减少虚拟机服务中断时间。总的来说，虚拟机在线迁移分为两大类：后拷贝迁移和预拷贝迁移。

预拷贝迁移是应用最广泛的虚拟机在线迁移方法，已经成功应用于各种场景，本文也采用预拷贝迁移方法。如图 1 所示，虚拟机预拷贝在线迁移可以分为 3 个阶段。第 1 阶段：磁盘拷贝，虚拟机的磁盘内容首先被复制到目的主机。第 2 阶段：循环复制，在第 1 次循环中，虚拟机的全部内存页被复制到目的主机。但是在第 1 次内存复制中，由于服务在运行，内存页会发生改变(即脏页)，接下来复制脏页到目的主机。循环终止的条件是脏页足够小或达到循环次数的上限。第 3 阶段：停机拷贝，虚拟机的工作被停止，将不会再出现脏页，将最后一次循环中产生的脏页传输到目的主机，虚拟机在目的主机被启动。

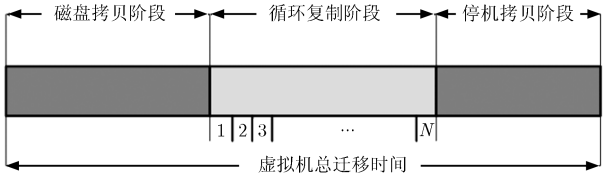

图 1 虚拟机在线迁移过程

Mann 等人[15]为虚拟机在线迁移建立了数学模型，使用下面的参数来计算虚拟机迁移的总时间：W(MB)表示虚拟机磁盘文件大小，M(MB)表示内存页大小，R(Mbps)表示虚拟机的脏页率，L(Mbps)表示虚拟机迁移时使用的总带宽，T(s)表示停机拷贝阶段的持续时间，X(MB)表示每次循环拷贝的内存文件大小。则磁盘拷贝阶段的持续时间表示为

$$T_d = W/L \tag{1}$$

循环复制阶段和停机拷贝阶段的总时间为

$$T_{p+s} = M \cdot \frac{1-(R/L)^n}{1-(R/L)} \Big/ L \tag{2}$$

停机拷贝阶段的时间为

$$T_s = \frac{M}{L} \cdot \left(\frac{R}{L}\right)^n \tag{3}$$

其中，

$$n = \min\left(\left\lfloor \log_{R/L} \frac{T \cdot L}{M} \right\rfloor, \left\lfloor \log_{R/L} \frac{X \cdot R}{M \cdot (L-R)} \right\rfloor\right) \tag{4}$$

则虚拟机的总迁移时间是 $T_d + T_{p+s}$，虚拟机的停机时间是 T_s。

3 问题提出

通过下面的例子，可以看出不同的虚拟机迁移序列对虚拟机迁移的总时间有很大的影响。如图 2 中所示，虚拟机{v0, v1,…, v9}分别放置在服务器{s1, s2,…, s6}中。每个服务器最多可以放置 3 个虚拟机，每个虚拟机要迁移总内容大小设置为 10。{(v1, v3) (v3, v5) (v2, v6) (v6, v9)}表示存在的虚拟链路，每个虚拟链路的带宽设置为 1，所有物理链路的带宽设置为 2。要达到的目标分布如图 3 所示。

在满足网络带宽和服务器性能要求的前提下，一个可行的虚拟机迁移序列是＜(v1, v5, v4, v9, v0)＞，一个不可行的虚拟机迁移序列是＜(v4, v1, v5, v9, v0)＞，因为该迁移序列违反了带宽限制。

为了减少虚拟机迁移的总时间，采用并行迁移，同时考虑剩余带宽对随后迁移序列的影响，根据本文提出的 HACE 算法，得到了图 4 所示的迁移序列。由初始分布和目标分布知，要迁移的虚拟机为{v0, v1, v4, v5, v9}，在初始状态，当前可迁移的虚拟机为{v0, v1, v5, v9}，根据代价评估函数有：T(v0)=10,

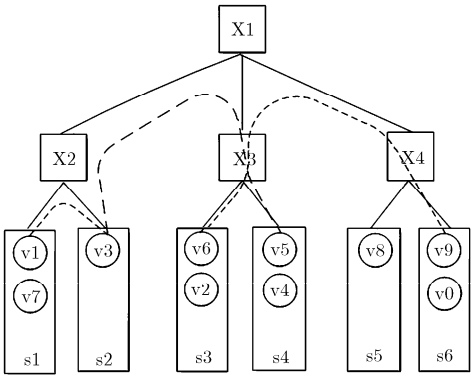

图 2　网络初始分布

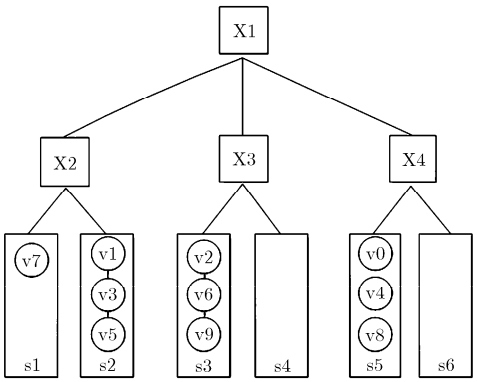

图 3　网络目标分布

T(v1)=10, T(v5)=−5, T(v9)=5，所以第 1 个选择迁移的虚拟机为 v5，因为考虑并行迁移，根据 HACE 算法，依次判断 v9, v0, v1 是否可与 v5 并行迁移，得到 v0 可与 v5 并行迁移。当有虚拟机迁移结束时，即 v0 和 v5 迁移完成，当前可迁移虚拟机被更新，根据代价评估函数选择并行的虚拟机迁移序列为 v9 和 v1。当 v9 结束时，v4 满足迁移条件，与 v1 并行迁移。最后总的虚拟机迁移时间是 20。

图 5 中虽然也考虑了虚拟机并行迁移，但没有考虑剩余带宽，最后总的迁移时间是 30。在这个例子中 HACE 算法减少了 33.3%迁移时间。因此研究虚拟机迁移序列是很有必要的。

4　HACE 算法

4.1 虚拟机迁移序列问题建模

使用图 $\overline{G}=(\overline{N},\overline{L})$ 来表示物理数据中心网络，其中 $\overline{N}$ 和 $\overline{L}$ 分别表示数据中心的节点和链路。定义 $\overline{N}=\overline{N}_H\cup\overline{N}_S$，其中 $\overline{N}_H$ 和 $\overline{N}_S$ 分别表示物理服务器和交换机的集合。每个物理服务器 $\overline{n}\in\overline{N}_H$ 的 CPU、内存和磁盘文件的大小分别表示如下：$\overline{C}_{\overline{n}}\in\mathbb{R}^+$，$\overline{M}_{\overline{n}}\in\mathbb{R}^+$ 和 $\overline{D}_{\overline{n}}\in\mathbb{R}^+$。每个物理链路 $l\in\overline{L}$ 的带宽表示为 $\overline{B}_{\overline{l}}\in\mathbb{R}^+$。

用 I 表示 VDC 的集合，每个 VDC $i\in I$ 用一个图 $G^i=(N^i,L^i)$ 来表示，其中 N^i 和 L^i 分别表示 i 中 VM 和虚拟链路的集合。对于每个虚拟机 $n\in N^i$ 的 CPU、内存和磁盘文件的大小分别表示为 $C_n^i\in\mathbb{R}^+$，$M_n^i\in\mathbb{R}^+$ 和 $D_n^i\in\mathbb{R}^+$。用 R_n^i 表示虚拟机 n 的脏页率。对于每个虚拟链路 $l\in L^i$，使用 $B_l^i\in\mathbb{R}^+$ 表示虚拟链路 l 的带宽需求，$\zeta_{\bar{l}\bar{l}}^i\in\mathbb{R}^+$ 表示物理链路 $\bar{l}\in\overline{L}$ 分配给虚拟链路 $l\in L^i$ 的带宽。

时刻 t, VDC i 的虚拟机 n 和物理主机 $\overline{n}$ 之间的关系用式(5)的参数表示：

$$y_{n\overline{n}}^{it}=\begin{cases}1, & \text{在时刻}t,n\in N^i\text{在物理主机}\overline{n}\in\overline{N}_H\text{内}\\ 0, & \text{其它}\end{cases}\tag{5}$$

在虚拟机迁移过程中，为了获得系统状态，采用了离散时间模型，时间片被分为固定的长度。用 [0，T]表示虚拟机迁移的时间间隔，其中 T 表示虚拟机迁移的最大允许持续时间。在时刻 t, 每个 VDC 的状态可以通过 $(y_{n\overline{n}}^{it})_{i\in I,n\in N_i,\overline{n}\in\overline{N}}$ 获得。

在虚拟机迁移序列问题中，初始和目标虚拟机配置分别用 $(y_{n\overline{n}}^{i0})_{i,n,\overline{n}}$ 和 $(y_{n\overline{n}}^{iT})_{i,n,\overline{n}}$ 表示。则虚拟机迁移序列问题的目标是发现一系列的状态 $(y_{n\overline{n}}^{it})_{i,n,\overline{n}}$，$0\le t\le T$，使系统从初始状态到目标状态转化：

$$\left(y_{n\overline{n}}^{i0}\right)_{i,n,\overline{n}}\to\left(y_{n\overline{n}}^{i1}\right)_{i,n,\overline{n}}\to\left(y_{n\overline{n}}^{i2}\right)_{i,n,\overline{n}}\to\ldots\to\left(y_{n\overline{n}}^{iT}\right)_{i,n,\overline{n}}\tag{6}$$

定义 $z_{n\overline{p}\overline{q}}^{it}\in\{0,1\}$ 作为一个变量表示在时刻 t, n 是否计划从 $\overline{p}$ 到 $\overline{q}$ 进行迁移：

$$z_{n\overline{p}\overline{q}}^{it}=\begin{cases}1, & \text{在时刻}t,n\in N^i\text{被计划从}\overline{p}\text{迁移到}\overline{q}\\ 0, & \text{其它}\end{cases}\tag{7}$$

由于一个迁移要许多时钟周期去完成，定义 $X_{n\overline{p}\overline{q}}^{it}\in\{0,1\}$ 来统计虚拟机的迁移时间：

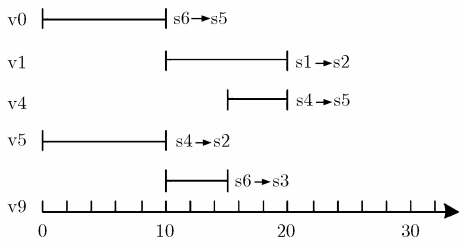

图 4 迁移序列 1

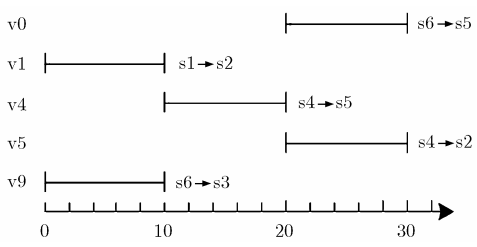

图 5 迁移序列 2

$$X_{n\overline{p}\overline{q}}^{it}=\begin{cases}1, \text{在时刻}t, n\text{正在从}\overline{p}\text{到}\overline{q}\text{进行迁移}\\ 0, \text{其它}\end{cases} \tag{8}$$

为了确保 $z_{n\overline{p}\overline{q}}^{it}$ 触发迁移，且每个服务器中只能有一个虚拟机迁移，给出式(9)和式(10)的约束条件：

$$X_{n\overline{p}\overline{q}}^{it}\ge z_{n\overline{p}\overline{q}}^{it},\ \forall i\in I,n\in N^i,\overline{p},\overline{q}\in\overline{N}_H,0\le t\le T \tag{9}$$

$$\sum_{\overline{q}\in N}X_{n\overline{p}\overline{q}}^{it}\le 1,\ \forall i\in I,n\in N^i,\overline{p},\overline{q}\in\overline{N}_H,0\le t\le T \tag{10}$$

使用 $b_{n\overline{p}\overline{q}}^{it}\in\mathbb{R}^+$ 表示在时刻 t 分配给从 $\overline{p}$ 到 $\overline{q}$ 进行迁移的虚拟机的带宽。使用 $r_{n\overline{p}\overline{q}}^{it}\in\mathbb{R}^+$ 表示虚拟机 n 在时刻 t 剩余要迁移的内容大小，在迁移开始时，有式(11)的约束：

$$r_{n\overline{p}\overline{q}}^{it}=z_{n\overline{p}\overline{q}}^{it}M_n^i,\ \forall i\in I,n\in N^i,\overline{p},\overline{q}\in\overline{N}_H,0\le t\le T \tag{11}$$

当迁移开始以后，满足式(12)的约束条件：

$$r_{n\overline{p}\overline{q}}^{i(t+1)}\ge r_{n\overline{p}\overline{q}}^{it}-b_{n\overline{p}\overline{q}}^{it}X_{n\overline{p}\overline{q}}^{it},\forall i\in I,\ n\in N^i,\ \overline{p},\ \overline{q}\in\overline{N}_H,\ 0\le t\le T-1 \tag{12}$$

为了保证虚拟机迁移正常的进行，给出式(13)的约束：

$$X_{n\overline{p}\overline{q}}^{i(t+1)}\ge\left(r_{n\overline{p}\overline{q}}^{it}-b_{n\overline{p}\overline{q}}^{it}X_{n\overline{p}\overline{q}}^{it}\right)\Big/M_n^i,\ \forall i\in I,\ n\in N^i,\overline{p},\overline{q}\in\overline{N}_H,0\le t\le T-1 \tag{13}$$

此外，运行在物理主机的虚拟机的 CPU、内存和磁盘资源要满足式(14)–式(16)的条件：

$$\sum_{i=1}^{I}\sum_{n\in N^i}y_{n\overline{n}}^{it}C_n^i\le\overline{C}_{\overline{n}},\ \ \forall t,\overline{n}\in\overline{N}_H \tag{14}$$

$$\sum_{i=1}^{I}\sum_{n\in N^i}y_{n\overline{n}}^{it}M_n^i\le\overline{M}_{\overline{n}},\ \ \forall t,\overline{n}\in\overline{N}_H \tag{15}$$

$$\sum_{i=1}^{I}\sum_{n\in N^i}y_{n\overline{n}}^{it}D_n^i\le\overline{D}_{\overline{n}},\ \ \forall t,\overline{n}\in\overline{N}_H \tag{16}$$

用 $v_{\overline{p}\overline{q}\overline{l}}\in\{0,1\}$ 表示从 $\overline{p}$ 到 $\overline{q}$ 的迁移路径中是否包含 $\overline{l}$，则带宽满足式(17)的约束：

$$\sum_{i=1}^{I}\sum_{n\in N^i}\xi_{\overline{l}\overline{l}}^{i}+\sum_{\overline{p}}\sum_{\overline{q}}b_{n\overline{p}\overline{q}}^{it}v_{\overline{p}\overline{q}\overline{l}}\le\overline{B}_{\overline{l}},\ \forall t,\overline{n}\in\overline{N}_H,\overline{l}\in\overline{L} \tag{17}$$

为了保证在迁移的过程中没有空周期，用 w^t 表示是否有迁移发生，则

$$w^t\ge X_{n\overline{p}\overline{q}}^{it},\ \forall i\in I,n\in N^i,\overline{p},\overline{q}\in\overline{N}_H,0\le t\le T \tag{18}$$

$$w^t\ge w^{t+1},\ \forall 0\le t\le T-1 \tag{19}$$

通过整形线性规划来定义 VDC 虚拟机迁移序列问题，从 CPU、内存、带宽、磁盘空间等方面来考虑，最终的目标是：

$$\min\left(\sum_{t=0}^{T}w^t\right)$$

4.2 HACE 算法设计与实现

虚拟机迁移序列问题的最优解可以通过简单排列组合得到，但实际上一般无法实现。尤其对于规模较大的实际问题来说，可行解的数量可能特别大。在没有 CPU 和内存的约束下，文献[16]中已经证明这是一个 NP-难问题。如何选择并行迁移的虚拟机是解决问题的关键。一种解决问题的思路是使用启发式的方法。下面详细描述基于代价评估的启发式算法的设计步骤和处理过程。

单个虚拟机迁移的总时间包括虚拟机磁盘拷贝阶段，循环复制阶段和停机拷贝阶段，使用式(20)计算虚拟机迁移的总时间 $T_M(v)$：

$$T_M(v)=T_d+T_{p+s} \tag{20}$$

为了获得较优的虚拟机迁移序列，对每个虚拟机计算 $T_I(v_j)$ 值，即虚拟机 v_j 迁移完成后，对其他未迁移虚拟机的迁移时间影响值的总和，其中 $T_M(v_k/v_j)$ 表示虚拟机 v_j 迁移完成后，虚拟机 v_k 迁移需要的时间。使用式(21)计算 $T_I(v_j)$：

$$T_I(v_j)=\sum_{k\in\{1\cdots N\}\setminus j}T_M\left(v_k/v_j\right)-T_M(v_k) \tag{21}$$

(1)初始化：获得网络的初始分布 $y_{n\overline{n}}^{i0}$ 和目标分布 $y_{n\overline{n}}^{iT}$。从初始分布到目标分布，如果虚拟机在同一个服务器中，则虚拟机不需要迁移，否则虚拟机需要迁移。根据 $y_{n\overline{n}}^{i0}$ 和 $y_{n\overline{n}}^{iT}$，得到要迁移的虚拟机集合 V_l。

(2)当前可迁移的虚拟机：在时刻 t，对每个虚拟机 $v\in V_l$，根据上文提到的带宽、服务器 CPU、磁盘、内存等约束条件，判断 v 是否可迁移，得到在时刻 t 可迁移虚拟机的集合 V_f。

(3)当前可并行迁移的虚拟机：为了获得较优的迁移序列，优先迁移需要迁移时间较短或在迁移完成后使剩余带宽增加的虚拟机。根据这个原则，首先对每个虚拟机 $v\in V_f$，计算虚拟机 v 的迁移时间 $T_M(v)$，然后计算对其他虚拟机迁移时间的影响值 $T_I(v)$，最后获得 $T_T(v)=T_I(v)+T_M(v)$。根据 $T_T(v)$ 的值，为每个虚拟机 v 设置优先级，$T_T(v)$ 的值越小，优先级越高。首先选择优先级最高的一个虚拟机进行迁移，然后依次按优先级的顺序判断虚拟机是否可以与正在迁移的虚拟机并行迁移，最后获得可并行迁移的虚拟机集合 P。

(4)迭代过程：当有虚拟机迁移完成时，更新 V_l 的值，然后得到在时刻 t 可迁移虚拟机的集合 V_f，根据得到的 $T_T(v)$ 值，判断当前可迁移的虚拟机是否可与正在迁移的虚拟机并行迁移，得到可并行迁移的虚拟机集合 P。迭代终止条件是 V_l 为空，即所有虚拟机迁移完成。

算法中使用 S_q 来记录虚拟机迁移的序列、开始时间和持续时间。使用 T 来记录虚拟机迁移的总时

间。HACE 的详细算法见表 1 的算法 1。

对算法的时间复杂度进行分析，代码 7~10 行对当前可迁移虚拟机进行代价评估，时间复杂度为 $O(|V_f|)$。代码 11~16 行找出可并行迁移的虚拟机，时间复杂度为 $O(|V_f|)$。第 3 行的 While 循环执行 $|V_l|$ 次，时间复杂度为 $O(|V_l|)$。所以 HACE 算法的时间复杂度为 $O(|V_l|^2)$。

表 1 基于代价评估的启发式算法

算法 1：基于代价评估的启发式算法(HACE 算法)
输入：初始分布 $y_{n\bar{n}}^{i0}$，目标分布 $y_{n\bar{n}}^{iT}$
输出：迁移序列 S_q，总迁移时间 T
(1) $S_q \leftarrow \varnothing$，$t \leftarrow 0$，$w_t \leftarrow 0$ //初始化参数
(2) $V_l \leftarrow$ {要迁移的虚拟机的集合}
(3) **while** $V_l \neq \varnothing$ **do**
(4) $P \leftarrow P \setminus v_i$ // v_i 表示已迁移完成的虚拟机
(5) $t \leftarrow t + w_t$
(6) $V_f \leftarrow$ {在时刻 t 可以迁移的虚拟机的集合}
(7) **for** each VM $v \in V_f$ **do**
(8) $T_T(v_j) = T_I(v_j) + T_M(v_j)$
(9) $V_f \leftarrow$ 根据 $T_T(v_j)$ 的值对 $v \in V_f$ 设置优先级
(10) **end for**
(11) **for** each VM $v \in V_f$ **do**
(12) **if** v can be included in P **then**
(13) $P \leftarrow P \cup \{v\}$
(14) $S_q \leftarrow S_q \cup \{\langle v, t, T_M(v)\rangle\}$
(15) **end if**
(16) **end for**
(17) $V_l \leftarrow V_l / P$
(18) **end while**
(19) $T = t + T_M(v_l)$ // v_l 指最后一个迁移的虚拟机
(20) **return** Migration sequence S_q, T

5 仿真实验

5.1 仿真设置

本文使用 MATLAB 进行仿真实验，实验中的数据中心有 1024 个服务器和 320 个交换机，通过树网络拓扑结构来连接[17]。其中每个服务器有 16 核的 CPU 和 64 GB 的内存，服务器与交换机之间的链路带宽是 1 Gbps。

实验在 6 个不同的环境进行，如表 2 所示，给出了每个环境的相关参数，包括 VDCs、VMs、虚拟链路、初始分布、目标分布和需要迁移的虚拟机个数。初始分布的产生是随机地把 VM 分配给服务器。通过虚拟机整合算法产生网络的目标分布，文献[18]中给出了基于预测的整合算法。文中未对虚拟机整合算法深入研究，目标分布的产生是尽可能地整合一个 VDC 中的 VM 到同一个服务器中，同时最小化使用服务器和虚拟链路的数量。例如，在 S1 中，有 12 个 VDCs，包括 48 个虚拟机和 52 个虚拟链路。在初始分布中，48 个服务器被占用，通过虚拟机的整合管理，只有 12 个服务器被占用。为了达到目标分配，需要进行 47 个虚拟机的迁移。在其他的 5 个环境中，考虑更多的 VDCs、VMs 和虚拟链路。

对于 VMs 的虚拟 CPUs 和内存大小，根据数据中心虚拟系统的参数进行随机选择。虚拟机之间的虚拟带宽从 50~250 MB 随机选择。而虚拟机的磁盘内容大小依靠于操作系统和应用程序，随机地从 1~30 GB 进行选择。文中所有的仿真在处理器为双核 2.67 GHz Intel Dell N5010 和 RAM 为 4 GB 的机器上运行。

5.2 实验结果和分析

图 6 所示为 HACE 算法与最小时间贪心算法(Minimum Time Greedy Algorithm, MTGA)[3]、最大带宽影响贪心算法(Maximum Bandwidth Effect Greedy Algorithm, MBEGA)的比较。MTGA 算法优先迁移需要迁移时间最短的虚拟机，然后依次按虚拟机迁移时间递增判断虚拟机是否能并行迁移。MBEGA 算法优先迁移对随后虚拟机迁移时间影响最大的虚拟机，然后依次按虚拟机带宽影响值递增判断虚拟机是否能并行迁移。

表 2 实验相关环境的具体参数

实验环境	VDC 数量	VM 数量	虚拟链路数量	初始分布主机数量	目标分布主机数量	需要迁移的虚拟机个数
S1	12	48	52	48	12	47
S2	30	120	135	120	30	118
S3	50	200	212	200	50	196
S4	100	400	396	400	100	395
S5	150	600	456	600	150	598
S6	250	1000	874	1000	250	995

由图 6 知，在虚拟机迁移数量较少时，如在 S1 中，虚拟机数量为 48, MTGA 算法需要 553.2 s 来完成虚拟机的迁移，MBEGA 算法需要 484.4 s 来完成虚拟机的迁移，HACE 算法完成虚拟机的迁移需要 401.5 s。与 MTGA 算法比较，HACE 算法减少虚拟机迁移总时间达到 27.4%。与 MBEGA 算法比较，虚拟机迁移总时间减少 17.1%。当数据中心要迁移的虚拟机数量较大时，如在 S5 和 S6 中，与 MTGA 算法比较，HACE 算法减少虚拟机迁移总时间达到 52.1%。与 MBEGA 算法比较，虚拟机迁移总时间减少 36.0%。

图 7 所示为 HACE 算法与 Heuristic[6]算法和 CQNCR[3] 算法的比较。Ghorbani 等人提出的 Heuristic 算法为每一个要迁移的虚拟机计算出一个 score 值，根据 score 值的大小给出一个接近最优的虚拟机迁移序列。CQNCR 算法并行地进行虚拟机迁移，将当前可迁移的虚拟机划分为若干个 RIG，即可同时并行迁移的虚拟机集合，然后对每个 RIG 进行评估，找出一个最优的 RIG 进行迁移。这个过程重复，直到所有的虚拟机迁移完成。

由图 7 知，在虚拟机迁移数量较少时，HACE 算法与 Heuristic 算法和 CQNCR 算法的性能很接近，如在 S1 中，虚拟机数量为 48 时，CQNCR 算法需要 435.6 s 来完成虚拟机的迁移，HACE 算法完成虚拟机的迁移需要 401.5 s。与 CQNCR 算法比较，HACE 算法仅减少 7.82%的虚拟机迁移时间。当数据中心要迁移的虚拟机数量较大时，HACE 算法能够有效地减少虚拟机的迁移总时间，如在 S5 和 S6 中，CQNCR 算法需要 1336.3 s 来完成虚拟机的迁移，HACE 算法完成虚拟机的迁移需要 1017.9 s。HACE 算法与 Heuristic 算法和 CQNCR 算法比较，减少虚拟机迁移的总时间分别为 42.0%和 23.8%。

由以上实验结果知，随着虚拟机数量的增加，当达到 200 左右时，虚拟机迁移的总时间增加越来越缓慢。这是因为 HACE 算法采用并行虚拟机迁移，虚拟机数量增加时，可并行迁移的虚拟机数量也增加。与贪心算法、Heuristic 算法和 CQNCR 算法的比较知，HACE 算法能找出更优的虚拟机迁移序列，有效地减少虚拟机迁移的总时间。

6 结束语

在数据中心和云环境中，虚拟机迁移被频繁地用于优化资源分配和获得相应的服务目标。找出一个最优的虚拟机迁移序列，对资源的利用和服务的性能都有重要的影响。本文提出 HACE 算法，解决软件定义网络中数据中心大量虚拟机同时迁移时的迁移序列问题，有效地减少虚拟机迁移的总时间，同时也保证了虚拟机迁移过程中网络状态的一致性。实验结果表明 HACE 算法是可行且有效的。后续工作中，将继续研究和虚拟机在线迁移相关的其他方面，包括复杂网络中虚拟机迁移的带宽分配，以及多虚拟机之间的相互影响等问题。

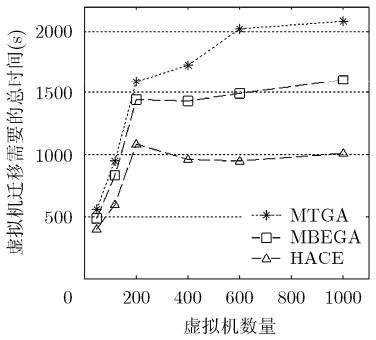

图 6 HACE 算法与贪心算法的比较

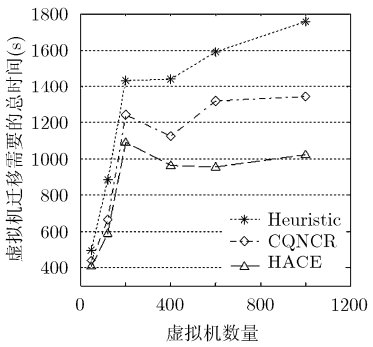

图 7 HACE 算法与 Heuristic 算法和 CQNCR 算法的比较

参 考 文 献

[1] KREUTZ D, RAMOS F M V, and ESTEVES V P. Software defined networking: A comprehensive survey[J]. *Proceedings of the IEEE*, 2015, 103(1): 14–76. doi: 10.1109/JPROC.2014.2371999.

[2] HERMENIER F, LORCA X, and MENAUD J M. Entropy: A consolidation manager for clusters[C]. Proceedings of the ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, Washington, DC, USA, 2009: 41–50. doi: 10.1145/1508293.1508300.

[3] BARI M F, ZHANI M F, ZHANG Q, *et al*. CQNCR: Optimal VM migration planning in cloud data centers[C]. Proceedings of IFIP Networking Conference, Trondheim, Norway, 2014: 1–9. doi: 10.1109/IFIPNetworking. 2014. 6857120.

[4] DUOLIKUN D, WATANABE R, KATAOKA H, *et al*. An

energy-aware migration of virtual machines[C]. 2016 IEEE 30th International Conference on Advanced Information Networking and Application, Crans-Montana, 2016: 557–564. doi: 10.1109/AINA.2016.156.

[5] MAIO V D, KECSKEMETI G, and PRODAN R. A workload-aware energy model for virtual machine migration[C]. 2015 IEEE International Conference on Cluster Computing, Chicago, IL, 2015: 274–283. doi: 10.1109/CLUSTER.2015.47.

[6] GHORBANI S and CAESAR M. Walk the Line: Consistent network updates with bandwidth guarantees[C]. Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN'12, New York, NY, USA, 2012: 67–72. doi: 10.1145/2342441.2342455.

[7] YE Kejiang, JIANG Xiaohong, MA Ran, *et al*. VC-migration: live migration of virtual clusters in the cloud[C]. Proceedings of the 13th ACM/IEEE International Conference on Grid Computing, Beijing, China, 2012: 209–218. doi: 10.1109/Grid.2012.27.

[8] 高先明, 张晓哲, 王宝生, 等. 面向虚拟路由器的基于历史转发开销的资源调度算法[J].电子与信息学报, 2015, 37(3): 686–692. doi: 10.11999/JEIT140491.
GAO Xianming, ZHANG Xiaozhe, WANG Baosheng, *et al*. Historical forwarding overhead based the resource scheduling algorithm for the virtual router[J]. *Journal of Electronics & Information Technology*, 2015, 37(3): 686–692. doi: 10.11999/JEIT140491.

[9] 刘中金, 卓子寒, 何跃鹰, 等. 一种基于动态配额的虚拟网带宽公平调度算法[J]. 电子与信息学报, 2016, 38(10): 2654–2659. doi: 10.11999/JEIT151485.
LIU Zhongjin, ZHUO Zihan, HE Yueying, *et al*. Dynamical Weighted Scheduling Algorithm supporting fair bandwidth allocation of virtual networks[J]. *Journal of Electronics & Information Technology*, 2016, 38(10): 2654–2659. doi: 10.11999/JEIT151485.

[10] WOOD T, SHENOY P, VENKATARAMANI A, *et al*. Sandpiper: Black-box and gray-box resource management for virtual machines[J]. *Computer Networks*, 2009, 53(17): 2923–2938. doi: 10.1016/j.comnet.2009.04.014.

[11] ZHANI M F, ZHANG Q, SIMONA G, *et al*. VDC Planner: dynamic migration-aware virtual data center embedding for clouds[C]. Proceedings of the 13th IFIP/IEEE International Symposiun on Integrated Network Management, Ghent, Belgium, 2013: 18–25.

[12] GUO Chuanxiong, LU Guohan, WANG H J, *et al*. SecondNet: A data center network virtualization architecture with bandwidth guarantees[C]. Proceedings of the 6th International Conference, Philadelphia PA, USA, 2010. doi: 10.1145/ 1921168.1921188.

[13] AMANI A and ZAMANIFAR K. Improving the time of live migration virtual machine by optimized algorithm scheduler credit[C]. Proceedings of the 4th International conference on Computer and Knowledge Engineering (ICCKE), 2014: 346–351. doi: 10.1109/ICCKE.2014.6993374.

[14] CERRONI W and ESPOSITO F. Optimizing live migration of multiple virtual machines[J]. *IEEE Transactions on Cloud Computing*, 2016. doi: 10.1109/TCC.2016.2567381.

[15] MANN V, GUPTA A, DUTTA P, *et al*. Remedy: Network-aware steady state VM management for data centers[C]. Proceedings of the 11th International IFIP TC 6 Networking Conference, Prague, Czech Republic, 2012: 190–204. doi: 10.1007/978-3-642-30045-5_15.

[16] GANDHI R and MESTRE J. Combinatorial algorithms for data migration to minimize average completion time[C]. Proceedings of the 9th International Conference on Approximation Algorithms for Combinatorial Optimization Problems, Barcelona, Spain, 2006: 128–139. doi: 10.1007/11830924_14.

[17] BARI M F, BOUTABA R, ESTEVES R, *et al*. Data center network virtualization: A survey[J]. *IEEE Communications Surveys & Tutorials*, 2013, 15(2): 909–928. doi: 10.1109/SURV.2012.090512.00043.

[18] 魏亮, 黄韬, 陈建亚, 等. 基于工作负载预测的虚拟机整合算法[J]. 电子与信息学报, 2013, 35(6): 1271–1276. doi: 10.3724/SP.J.1146.2012.01131.
WEI Liang, HUANG Tao, CHEN Jianya, *et al*. Workload prediction-based algorithm for consolidation of virtual machines[J]. *Journal of Electronics & Information Technology*, 2013, 35(6): 1271–1276. doi: 10.3724/SP.J.1146. 2012.01131.

史久根：　男，1963 年生，副教授，研究方向为嵌入式系统、计算机网络和无线传感器网络.

许辉亮：　男，1991 年生，硕士生，研究方向为软件定义网络、网络虚拟化和嵌入式系统.

陆立鹏：　男，1991 年生，硕士生，研究方向为无线传感器网络和嵌入式系统.