

在轻载节点完成迁移操作后会关闭该节点. 确定待迁移(关闭)轻载节点列表方法如下:

首先将轻载节点按负载度进行升序, 此时轻载节点列表 $\{NL_1, NL_2, \dots, NL_s\}$, 则对应节点负载值列表为 $\{Ld_1, Ld_2, \dots, Ld_s\}$, Ld_{dc} 为此时的全局负载度, n 为正在使用的节点总数, 设需要关闭的节点为 k 个, 则 k 应该满足如下条件的最小正整数:

$$\begin{cases} n(\omega_l - Ld_{dc}) < \sum_{i=1}^k Ld_i \\ 0 < k < n \left(1 - \frac{Ld_{dc}}{\omega_l}\right) < s \end{cases} \quad (11)$$

当 $Ld_{dc} > \omega_l$ 且 $Lr_h > \max Lr_h$ 时, 数据中心负载度偏高, 此时开启一个新的节点.

本文采用的动态阈值调整具体流程如图1所示.

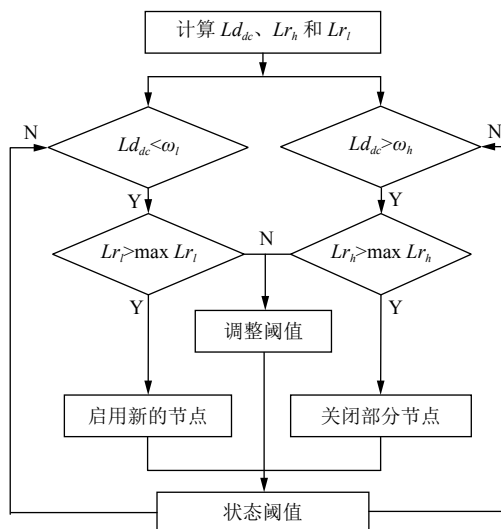


图1 状态阈值动态调整流程图

2.2 迁移触发算法

由状态阈值可以对节点负载状态进行评估, 是判断节点是否需要迁移调整的重要步骤, 其评估结果是迁移触发时机判定的重要依据. 但是节点负载变化时会出现短时间内的负载抖动现象产生负载瞬时值, 进而影响迁移触发时机的判断, 降低虚拟机迁移的有效性. 本文采用乘积指数加权法^[11]对 m 期内的负载进行评估, 乘积指数加权公式如下:

$$y = \prod_{t \in T_1} \frac{x_t^{w_t}}{\max_{t \in T_1} \{x_t^{w_t}\}} \prod_{t \in T_2} \frac{\left(1 + \frac{1}{x_t}\right)^{w_t}}{\max_{t \in T_2} \left\{ \left(1 + \frac{1}{x_t}\right)^{w_t} \right\}} \quad (12)$$

其中, w_t 为第 t 期权重系数. 若此时为负载过载评估, 则 T_1 为负载超过高位阈值的期数, T_2 为负载低于高位阈值的期数; 若为负载轻载评估, 则 T_1 为负载值低于低位阈值的期数, T_2 为负载值高于低位阈值的期数.

根据以上状态阈值动态调整方法结合负载值评估方法, 设计了虚拟机的迁移时机判决算法, 内容如算法1.

算法1. 基于动态阈值的迁移时机判决算法

- 1) 收集各节点负载信息, 计算监测时间内全局负载度与高低负载节点占比, 判断是否需要更新阈值, 是则转第2)步, 否则转第3)步.
- 2) 分析全局负载度与高低负载占比, 调整并更新状态阈值.
- 3) 根据状态阈值, 分别判断各节点负载状态. 若节点负载指标高于高位负载, 开启该节点该负载指标过载评估, 转第4)步; 若节点负载度低于低位负载, 开启该节点负载度轻载评估, 转第5)步; 均未达到高低阈值要求为适载节点, 则转第7)步.
- 4) 对节点超过高位阈值的负载指标进行 m 期的查询并计算其评估值, 若评估值达到过载要求, 则判断此节点为该指标类型过载, 根据定义3, 更新高负载节点矩阵 H . 转第3)步, 继续循环.
- 5) 对负载度低于低位阈值的节点进行 m 期的负载度查询并计算其评估值. 若 m 期内评估值均达到轻载要求, 则判断此节点轻载节点, 加入低负载节点列表. 转第3)步, 继续循环.
- 6) 将低负载节点列表按负载度升序, 结合全局负载度, 判断是否在负载均衡范围 (ω_l, ω_h) 中, 若 $Ld_{dc} < \omega_l$, 则根据式(11)计算得到待迁移操作的轻载节点集合 L , 若 $Ld_{dc} > \omega_l$ 且 $Lr_h > \max Lr_h$, 则启用一个节点.
- 7) 输出高负载节点矩阵 H 与低负载节点矩阵 L .

上述算法可以通过动态阈值判断迁移时机并得到高负载与低负载节点列表, 即确定需进行迁移选择策略的节点.

3 基于负载类型感知的迁移选择算法

基于动态阈值的迁移时机判决算法确定了需进行虚拟机迁移操作的高负载与低负载节点列表, 本节的迁移选择算法包含两个方面: 待迁移虚拟机的选择算法和目的节点的选择算法, 旨在选择合适的待迁移虚拟机与目的节点, 解除节点过载与轻载状态.

3.1 虚拟机选择算法

根据过载节点内所有虚拟机使用资源情况, 选择合适的待迁移在虚拟机迁移过程中至关重要. 本文将过载情况根据过载资源类型进行划分, 以最少迁移次数为目标, 那么待迁移虚拟机的选择即可描述为节点迁移走后某虚拟机后过载资源回到适载范围(即不超过此资源类型的高位阈值), 使用以下公式描述:

$$N_{type} - \overline{V}_{type} \leq Thr_{type} \quad (13)$$

其中, N_{type} 为节点资源为 $type$ 类型的指标, $V = (\overline{v}_{cpu},$

$\overline{v_{ram}}, \overline{v_{band}}$)为虚拟机在 n 时段所使用的平均资源利用率向量. 除了满足此条件, 还需考虑其他方面因素, 如下:

(1) 虚拟机与节点的资源依赖度

考虑到最快解决过载状态, 引入虚拟机与节点的资源依赖度来评价虚拟机使用节点的资源情况, 按过载类型寻找该类型资源依赖度高的虚拟机. 虚拟机与节点的资源依赖度 R_{type} 定义如下:

$$R_{type} = \frac{\sum_{i=1}^n V_{type\ i} N_{type\ i}}{\sqrt{\left(\sum_{i=1}^n V_{type\ i}^2\right) \left(\sum_{i=1}^n N_{type\ i}^2\right)}} \quad (14)$$

其中, V_i 为虚拟机在 i 时刻的某指标资源利用率, N_i 为节点在 i 时刻的该指标资源利用率, R_{type} 越大则表示该虚拟机与此节点的该指标类型资源的依赖越高.

(2) 虚拟机当前内存带宽比

虚拟机迁移的核心考虑因素是虚拟机内存页的迁移, 此外还易受到节点带宽因素的影响, 故而在迁移开销最小化的目的下选取虚拟机当前内存带宽较小的虚拟机, 公式如下:

$$u_{rb} = \sqrt{\left(\frac{R_i}{B_i}\right)^2 - \left(\frac{v_{ram\ j}}{B_i}\right)^2} \quad (15)$$

其中, R_i 为此节点 i 的可用内存; B_i 为此节点 i 可用带宽; $v_{ram\ j}$ 为虚拟机 j 的内存.

(3) 虚拟机负载贡献度

虚拟机负载贡献度表现出此虚拟机使用此节点资源情况, 此值越大则表示该虚拟机使用的资源越多. 已知虚拟机在 n 时段所使用的平均资源利用率向量为 V , 该节点的负载指标的权值为 K , 则该虚拟机在此节点上的负载贡献度用式 (16) 描述:

$$Lc = V \cdot K = k_1 \times v_{cpu} + k_2 \times v_{ram} + k_3 \times v_{band} \quad (16)$$

若限定了某种资源类型 Lc 的值越大则代表另外虚拟机使用的两种资源越多.

由式 (13) 至式 (16) 可以得到影响虚拟机选择的 3 个指标值, 为了获得较少的迁移时间与迁移开销, 使用式 (17) 对这 3 个指标值进行处理.

$$S = \sqrt{Lc^2 + u_{rb}^2 + \left(\frac{1}{R_{type}}\right)^2} \quad (17)$$

选择待迁移虚拟机时将虚拟机按照此参考值升序

排列, 选取此参考值小的虚拟机可获得最少的迁移次数、较少迁移时间与迁移代价.

为了方便为待迁移虚拟机选择目的节点 (即下一小节), 为待迁移虚拟机计算资源比, 即以下公式:

$$\frac{v_{cpu}}{CU} : \frac{v_{ram}}{RU} : \frac{v_{band}}{BU} \quad (18)$$

由虚拟机使用资源比可以得到此虚拟机的资源偏向类型.

考虑到以上问题, 本文以最快解决节点过载问题为目的, 选取迁移时间最短的虚拟机, 算法表达如算法 2.

算法 2. 待迁移虚拟机选择算法

- 1) 根据节点评估得到的高负载节点列表, 获取高负载节点信息.
- 2) 根据此节点过载类型, 根据式 (13) 计算一次迁移可使得该节点解除过载情况的待筛选虚拟机集合.
- 3) 根据式 (14) 计算待筛选虚拟机集合中各虚拟机的此类资源依赖度, 并按递减排序.
- 4) 根据式 (15) 计算各虚拟机的内存带宽比; 根据式 (16) 计算虚拟机的负载贡献度.
- 5) 根据式 (17) 选择虚拟机作为待迁移虚拟机并计算其资源比, 加入待迁移虚拟机列表.

若节点是由于轻载需进行迁移, 则直接获取其上运行的虚拟机, 计算其资源偏向比并加入待迁移虚拟机列表.

3.2 目的节点选择算法

目的节点的选取即为虚拟机选取合适的目的节点, 即虚拟机安置问题, 高效的节点选择算法有利于提高整个平台的稳定性和资源的高效利用. 要考虑的因素主要有虚拟机与目的节点的资源匹配度以及迁移代价.

虚拟机与目标节点的资源匹配度主要用来度量节点剩余资源与待虚拟机所需资源的承接情况. 考虑到资源利用的均衡性, 依据虚拟机资源偏向优先为其配置该资源剩余度高的节点. 匹配度描述如下:

$$M = \frac{\sum_{i=1}^m \sqrt{(Ld_i - Lc_i)^2}}{\sqrt{(T_{best} - m)}} \quad (19)$$

其中, m 表示当前有 m 个虚拟机正在迁移到该台服务器上, T_{best} 为设定的节点承载虚拟机迁移的最大数目; Lc_i 为第 i 个虚拟机的资源需求量;

选择目的节点必须要考虑的一个问题就是虚拟机的迁移代价, 这代表者虚拟机迁移过程中的资源损耗,

若是迁移代价过高,则会造成资源浪费.虚拟机到目的节点的迁移代价与两者距离有关,其迁移代价由式(20)定义:

$$C(V_i, N_i) = \sum_{i \neq j} D(N_{i \text{ target}}, N_{j \text{ src}}) \cdot f(V_i, V_j) \quad (20)$$

其中, $N_{i \text{ target}}$ 指 V_i 的目的节点, $N_{j \text{ src}}$ 为 V_i 的源节点; $D(N_{i \text{ target}}, N_{j \text{ src}})$ 为 V_i 到目的节点和源节点上剩余虚拟机 V_j 的通信距离; $f(V_i, V_j)$ 为虚拟机之间的通信量.

根据以上内容,设计目标节点选择算法表达可见算法3.

算法3. 目标节点选择算法

- 1) 根据节点评估得到的适载节点列表,由负载度递增排序,得到待选节点列表.
- 2) 根据式(19)计算待选节点列表计算虚拟机与节点的匹配度,并按递减排序.
- 3) 按照匹配度顺序根据式(20)分别计算迁移代价.
- 4) 选取匹配度高且迁移代价最小的节点为目的节点.

4 实验验证与分析

4.1 实验准备

为了验证基于虚拟机动态迁移的负载均衡策略的有效性,采用 CloudSim 仿真平台进行仿真验证.建立含有异构节点的数据中心,并配置多规格虚拟机,为使仿真使用的物理机和虚拟机负载更接近现实情况,使用 PlanetLab 项目中提供的虚拟机监测数据作为实验负载,模拟数据中心进行虚拟机迁移的仿真实验,实验配置如表1.

表1 实验配置表

项目	数量(个)	配置	
物理节点	50	CPU(Mips)	{1000, 1800, 2600, 3000}
		内存(GB)	{1, 2, 4, 8}
		带宽(Mb/s)	{500, 700, 1000}
虚拟机	50	CPU(Mips)	{200, 500, 1000, 1500, 2000}
		内存(GB)	{0.5, 1, 2, 3}
		带宽(Mb/s)	{100, 200, 500}

为了比较本文策略的仿真效果,使用以下指标对迁移策略的性能进行分析:

1) 虚拟机迁移次数.实验中,由于虚拟机个数、负载和位置的变化,导致节点的负载动态变化,虚拟机迁

移策略影响着虚拟机的迁移次数,故获取指定时间内的虚拟机迁移次数,用来观测系统的虚拟机的迁移频率,是判断虚拟机迁移策略性能的重要指标.

2) 数据中心平均负载度.数据中心负载度是衡量数据中心负载情况与资源分配情况的有效指标,若负载度保持较低水平则说明较多节点存在过多的空闲资源,即资源分配不均衡,若此指标数值过高,则说明存在大量过载节点.

3) 平均 SLAV (Service Level Agreement Violation, SLAV).平均 SLAV 即平均服务等级协议违例率,用来评价用户服务质量水平的常用指标之一,该值越低则表示用户服务质量越高.由于本文算法综合考虑了节点的 CPU 资源、内存资源和带宽资源,使用多维度平均 SLAV 计算方法,如式(21):

$$\overline{SLAV} = \frac{1}{k} \sum_{j=1}^n \frac{1}{3} \left(\sum_{i=1}^3 \frac{v_{iR}^{type} - v_{iA}^{type}}{v_{iR}^{type}} \right) \times 100\% \quad (21)$$

其中, v_{iR}^{type} 表示虚拟机需求的 $type$ 类型资源量, v_{iA}^{type} 为虚拟机 $type$ 类型资源的实际使用量. K 为一个周期内的违约次数.

4.2 实验分析

为了比较本文算法的仿真效果,将使用两个对比方案进行实验对比.对比方案分别为:瞬时迁移触发策略搭配最大相关性选择策略的虚拟机迁移策略(以下称策略1)与文献[9]中的基于双门限与资源需求相关性匹配的虚拟机迁移算法(以下称策略2).针对虚拟机迁移的两个模块进行两组实验,第一组为迁移触发策略比较实验,用以分析本策略中基于动态阈值的负载评估算法触发迁移时机的合理性;第二组实验为迁移策略性能对比实验,分析比较3种不同的迁移策略效果,并对本文策略进行综合评价.

实验一. 迁移触发策略对比组

合理的迁移触发策略可以有效判断虚拟机迁移时机,并通过触发时机的合理性减少由于节点负载不均衡导致的服务效率问题.通常通过观测虚拟机迁移次数,从而分析迁移触发策略的性能.

本实验组使用3种触发策略并在同一种选择策略的条件下进行实验,通过收集一段时间内不同虚拟机数量下的虚拟机迁移次数进行可视化分析.3种触发策略分别为:策略1中的瞬时触发方式、策略2中的双门限触发方式和本文动态阈值触发方式.实验结果如

图2所示。

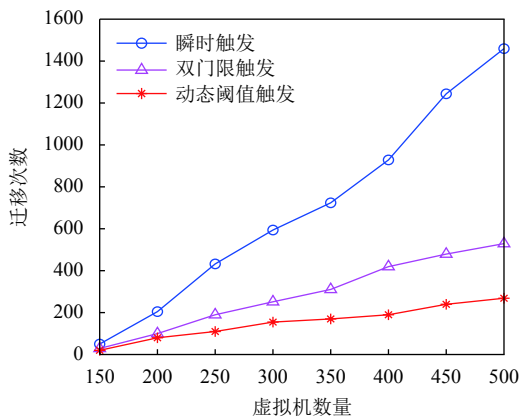


图2 虚拟机迁移次数对比图

由图2可见,随着虚拟机数目的增加,3种触发策略下的虚拟机迁移次数也逐渐增加。其中,由瞬间触发策略下的虚拟机迁移频率曲线斜率最大且增长速度明显高于其他两种策略,动态阈值触发策略下的虚拟机迁移频率最为平缓,由此可见使用基于动态阈值的负载评估算法进行迁移时机判断更能有效减少由瞬时峰值导致的虚拟机迁移,更有效解决由于虚拟机频繁迁移导致的资源浪费问题。

实验二. 迁移策略性能比对照组

将本文迁移策略、策略1和策略2的实验效果进行分析比较,主要从数据中心的负载度和平均SLAV进行比较。不同虚拟机数量下的数据中心的负载度对比图如图3所示。

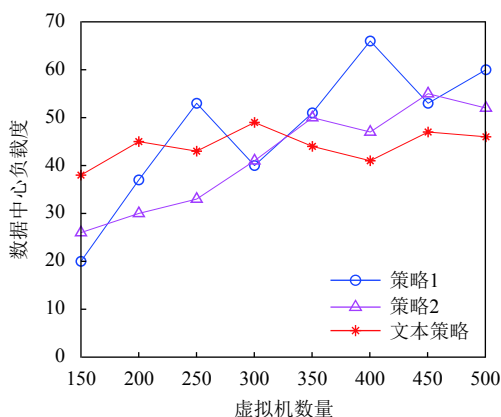


图3 数据中心负载度对比图

随着虚拟机数目的增加,3种迁移策略下的数据中心负载度逐渐增加,由策略1下的数据中心负载度曲

线增长最快且波动最明显,本文迁移策略下的数据中心负载度增长最慢且趋势平缓,可见本文考虑虚拟机和节点过载资源的依赖关系和虚拟机负载贡献度的虚拟机选择算法配合根据虚拟机与目的节点资源匹配度和迁移代价的目的节点选择算法在保持数据中心负载均衡方面是有效的。

保证数据中心处于较高水平的服务质量也是评价虚拟机迁移策略效果的必要的指标。本文通过分析平均SLAV对3种迁移策略下的服务质量进行评价,不同虚拟机数量下的平均SLAV比较图如图4所示。

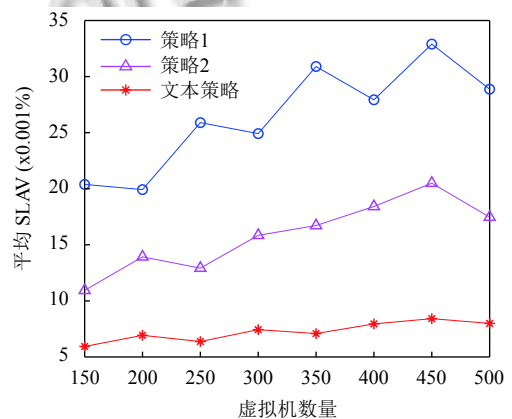


图4 数据中心平均SLAV对比图

由图4可见,随着虚拟机数目的增加,3种迁移策略下的数据中心平均SLAV呈现不同程度的波动趋势,策略1的平均SLAV曲线波动明显,策略2其次,本文策略具有最平稳的变化趋势,由此可见,本文策略可以有效地保证数据中心的SLAV的违背率,并提供较为稳定的服务质量。

综上所述,本文基于动态阈值的负载评估算法可以有效合理地对迁移时机进行判断并有效减少负载瞬时变化引起的迁移问题;基于负载类型感知的选择策略包括根据虚拟机与节点过载资源的依赖关系与虚拟机负载贡献度的虚拟机选择算法与根据待迁移虚拟机与目的节点的资源匹配度与迁移代价的目的节点选择算法在保证服务质量与数据中心负载均衡方面是有效的。

5 结论与展望

本文针对数据中心异构节点设计了一种基于虚拟机动态迁移的负载均衡策略,旨在通过解决异构节点

钟各资源利用率不均衡问题达到数据中心负载均衡. 设计了基于动态阈值的负载评估算法用于判断迁移时机与基于负载类型感知的选择策略. 实验证明, 本文迁移策略可以有效减少虚拟机迁移频率, 保证服务质量并改善数据中心的负载均衡能力.

参考文献

- 1 Chen T, Zhu YM, Gao XF, *et al.* Improving resource utilization via virtual machine placement in data center networks. *Mobile Networks and Applications*, 2018, 23(2): 227–238. [doi: [10.1007/s11036-017-0925-7](https://doi.org/10.1007/s11036-017-0925-7)]
- 2 Choudhary A, Govil MC, Singh G, *et al.* A critical survey of live virtual machine migration techniques. *Journal of Cloud Computing*, 2017, 6(1): 23. [doi: [10.1186/s13677-017-0092-1](https://doi.org/10.1186/s13677-017-0092-1)]
- 3 Dave A, Patel B, Bhatt G. Load balancing in cloud computing using optimization techniques: A study. *Proceedings of 2016 International Conference on Communication and Electronics Systems*. Coimbatore, India. 2016. 1–6.
- 4 Shen HY. RIAL: Resource intensity aware load balancing in clouds. *IEEE Transactions on Cloud Computing*, 2017. [doi: [10.1109/TCC.2017.2737628](https://doi.org/10.1109/TCC.2017.2737628)]
- 5 曲晓雅, 刘真. 基于阈值滑动窗口机制的虚拟机迁移判决算法. *计算机科学*, 2016, 43(4): 64–69.
- 6 李腾耀, 张月玲, 张凤琴, 等. 基于动态阈值和占优资源的云资源调度方法研究. *计算机应用研究*, 2017, 34(4): 1199–1203, 1216. [doi: [10.3969/j.issn.1001-3695.2017.04.055](https://doi.org/10.3969/j.issn.1001-3695.2017.04.055)]
- 7 Paulraj GJL, Francis SAJ, Peter JD, *et al.* A combined forecast-based virtual machine migration in cloud data centers. *Computers & Electrical Engineering*, 2018, 69: 287–300.
- 8 刘鏊, 虞红芳, 郑少平. 面向业务动态变化的虚拟机迁移技术研究. *计算机应用研究*, 2016, 33(2): 534–539. [doi: [10.3969/j.issn.1001-3695.2016.02.050](https://doi.org/10.3969/j.issn.1001-3695.2016.02.050)]
- 9 胡荣辉, 王瑞通. 云环境下虚拟机集群迁移策略研究. *计算机技术与发展*, 2017, 27(11): 33–36. [doi: [10.3969/j.issn.1673-629X.2017.11.007](https://doi.org/10.3969/j.issn.1673-629X.2017.11.007)]
- 10 刘琨. 云计算负载均衡策略的研究[博士学位论文]. 长春: 吉林大学, 2016.
- 11 郭三刚. 多属性对象决策的乘积指数加权法及应用. *陕西理工学院学报(自然科学版)*, 2012, 28(6): 34–39.