

DOI: 10.7652/xjtuxb201601016

单步启发式策略的备份虚拟机复用策略

陈鹏飞, 李昕怡, 齐勇, 张小辉

(西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 针对云环境中的备份虚拟机(VM)利用率过低的问题,提出了基于不停歇多臂赌博机(RM-AB)方法的备份 VM 分时复用策略,并给出了获得最优解的条件。该策略将每个备份 VM 形式化为具有“空闲”(1)和“占用”(0)两种状态的 Markov 过程,将多个备份 VM 的调度问题形式化为具有多个 Markov 过程的 Markov 决策问题(MDP),最终目标是期望在有限的备份 VM 数量下,最大化备份 VM 的利用率同时保证系统整体的可用性不会明显降低。然而,利用传统的动态规划方法求解该问题时会出现维度爆炸的现象,从而导致问题不可解,故将该 Markov 决策问题转化为 RM-AB 问题,然后利用简单易操作的单步启发式算法进行求解,并通过计算单步最优获得长期最优解,在特定条件下该策略可以保证得到的解为最优解。模拟实验结果表明:所提方法将备份 VM 与服务 VM 之间的备份比例从 1:1 扩展成 1:M($M \gg 1$),同时保证失效 VM 的恢复比率不低于 96%,相应地备份 VM 的利用率显著提高;在 VM 失效率较低的条件,备份 VM 利用率比 1:1 备份时提高了 89%;利用该备份 VM 调度策略,有助于减少整个云计算平台的建设和运维费用。

关键词: 云计算;可用性;虚拟机迁移;不停歇多臂赌博机

中图分类号: TP301 **文献标志码:** A **文章编号:** 0253-987X(2016)01-0100-08

Multiplexing of Backup Virtual Machine Based on Single-Step Heuristic Policy

CHEN Pengfei, LI Xinyi, QI Yong, ZHANG Xiaohui

(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Based on the restless multi-arm bandit (RMAB) approach, a multiplexing strategy of backup virtual machines (VMs) is proposed to resolve the problem of low utilization of backup VMs in the cloud environment, and the optimal condition is given. This strategy regards an individual backup VM as a Markov process with two states, namely “idle” (1) and “backup” (0), and models the scheduling of multiple backup VMs as a Markov decision problem (MDP) consisting of multiple Markov processes. The goal of this strategy is to maximize the utilization of backup VMs without obvious reduction in the system availability under the constraint of limited backup VMs. However, this problem is computationally intractable with traditional dynamic programming methods due to the curse of dimensionality. Therefore, this paper transforms the original MDP problem to a RMAB problem and adopts a simple single-step heuristic policy to resolve it. By calculating the single-step optimal solution, the long-term optimal solution can be obtained. Under specific conditions, the optimal solution of this strategy

收稿日期: 2015-05-10。 作者简介: 陈鹏飞(1987—),男,博士生;齐勇(通信作者),男,教授,博士生导师。 基金项目: 国家自然科学基金资助项目(60933003)。

网络出版时间: 2015-11-04

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20151104.2222.002.html>

is guaranteed. The results of simulation experiments show that the proposed policy can achieve the goal of extending the backup ratio between backup VMs to service VMs from 1:1 to 1:M ($M \gg 1$) while the failed VM assurance rate is no lower than 96%. Correspondingly, the utilization of backup resources is significantly enhanced. When the failure rate of service VM is low, the utilization of backup resources can be raised 89% compared with the 1:1 backup. The building and operation costs of a cloud platform can be reduced with the help of this backup VM scheduling strategy.

Keywords: cloud computing; availability; VM migration; restless multi-armed bandit

云基础设施可以通过虚拟化技术弹性地管理计算资源,进行灵活地硬件资源共享,以节省硬件成本和能耗成本。但是,复杂的云计算软件栈中的 bug、瞬息多变的运行时环境以及随机出现的硬件失效等导致云系统的可用性保障难度骤增。为了保障云应用的高可用性,云基础设施往往采用冗余备份的方式,为相关虚拟机(VM)实例提供一个或者多个备份实例。然而,过度的冗余备份会造成计算资源的浪费,增加云基础平台的运维成本。

长久以来学术界和工业界一直聚焦于寻求一种高效的 VM 预留或者整合策略,以提高用于生产即直接提供服务的虚拟资源或者硬件资源的利用率,如文献[1-2]等通过设计新的任务调度器或者虚拟机调度器,达到提高资源利用率的目的。为了达到 VM 的 k -resilient^[3]即允许 k 个物理机同时失效,每个 VM 至少需要提供 k 个备份。假定在最弱的容错条件下即 1-resilient,有 n 个 VM 提供服务,云基础平台共需要分配 $2n$ 个虚拟机,如果采用热备份,计算资源和能耗是没有容错情况的两倍。然而,对于单个 VM 来说失效是稀疏事件,备份 VM 大部分时间处于空闲状态,如果利用空闲的备份 VM 对其他需要备份的 VM 进行备份,即对备份 VM 进行分时复用,可以减少备份 VM 的数量,同时减少能耗。解决该问题的难点包括:①大规模的云环境中包含成百上千的备份 VM,每个备份 VM 都有各自特定的变化规律,如果把每个 VM 形式化为一个变量,问题最终形式化为超高维资源优化分配问题,而对此问题的精确求解几乎是不可能的,因此如何设计算法通过降维或者问题分解的方法解决超高维优化问题是主要难点;②在云环境中,任意时刻都会出现多个 VM 失效,如何保证尽可能多的失效 VM 得到备份,也是问题的难点之一;③由于 VM 失效是动态变化的,因此要求调度策略可以实时在线决策,在每个时间槽内能快速输出调度结果;④调度策略在保证备份 VM 得到充分利用的同时,不能明显影响

到系统的整体可用性,在有限的备份 VM 条件下计算出能够进行备份容错的 VM 上限也是难点之一。

针对以上问题及难点,本文将备份 VM 复用问题形式化为包含‘0’、‘1’两状态的不停歇多臂赌博(RMAB)问题^[4],通过将 m 维的优化问题分解为 m 个一维优化问题,给出了 VM 失效模型已知情况下的优化算法,并给出了算法最优解条件。模拟实验结果表明了本文提出的方法的有效性。

1 虚拟机备份

软件系统运行过程中,由于受到软件自身 Bug、硬件损坏、断电以及其他外界环境变化的影响,会出现性能下降和失效现象。软件失效的特征规律在软件可靠性领域一直是个开放性话题,目前广泛采用的失效模型(两次失效之间的间隔时间分布)有 Weibull 分布^[5]和 Poisson 分布^[6]。软件出现失效后,系统管理员会采取一定的措施,例如重启等恢复系统,恢复时间通常服从 Lognormal 分布^[5]。从软件的长期运行过程看,软件系统一直处于“失效-恢复”的循环过程中。本文主要讨论失效时间符合 Weibull 分布和 Poisson 分布、恢复时间符合 Lognormal 分布时的情况。

为了保障云平台的可用性,大量 VM 用于冗余备份。冗余备份的方式主要分为热备份和冷备份,热备份是指主服务 VM 实例与备份 VM 同时运行在多台物理机上,两者之间的运行状态时刻保持同步,例如 Remus^[7]等;冷备份不需要主服务 VM 与备份 VM 同时运行以保持运行状态的一致性,而是当主服务 VM 需要备份时,备份 VM 再启动,运行状态或者负载从主服务 VM 拷贝到备份 VM。本文只讨论该种备份方式,利用虚拟机在线迁移技术(live migration),将即将失效的 VM 迁移到备份服务器上,等 VM 恢复后备份 VM 关闭,原 VM 提供服务,采用这种方式既能保证软件运行时状态的一致性又能减少软件的悬停时间。

针对备份资源的有效利用问题,文献[8]设计了 Yank 系统,可以利用一个备份服务器对多个 Transient 服务器进行备份,但是当 VM 转移到 Stable 服务器上后,仍然是 1:1 的备份,与本文强调的对备份系统的分时复用策略不同。文献[9]讨论如何通过减少备份供电基础设施,例如 UPS 等,达到降低数据中心建设成本的目的,与本文的核心思想一致。本文第一次利用 RMAB 模型,对备份虚拟资源进行有效利用。

2 单步启发式策略

2.1 问题形式化

本文提出的策略适用于以下场景:一部分 VM 拥有专属的备份 VM,称之为高优先级 VM,当高优先级 VM 需要备份时,可以立即迁移到相应的备份 VM 上;一部分 VM 没有专属的备份 VM,称之为次优先级 VM,当次优先级 VM 需要备份时,依据本文提出的调度方案选择合适的备份 VM 进行迁移,如果出现高优先级 VM 与次优先级 VM 争用备份 VM 情况,高优先级 VM 优先调度。

开始阶段,云系统为特定的 N 个高优先级 VM 按照 1:1 的比例分配了 N 个备份 VM,以保证这些 VM 的高可用性,备份 VM 是这组高优先级 VM 状态的直接反映即高优先级 VM 正常,备份 VM 空闲;高优先级 VM 异常,备份 VM 忙碌,因此在后续部分,本文不再关注这 N 个高优先级 VM 的状态,而只关注 N 个备份 VM 的状态。每个备份 VM 的状态按照独立同分布的 Markov 过程演化,令 $s_j(t)$ 和 S_j 分别表示备份 VM $_j$ ($j=1,2,\dots,N$) 在 t 时刻的状态和状态空间,令 P_j 表示备份 VM $_j$ 的状态转移矩阵, $S(t)=\{s_j(t), j \in \{1,2,\dots,N\}\}$ 表示 N 个备份 VM 在 t 时刻的状态。用状态‘1’表示备份 VM 空闲,‘0’表示备份 VM 被占用,则备份 VM 的状态演化过程可以表述为包含‘0’、‘1’两状态的 Markov 过程。其中, $s_j(t) \in \{0,1\}$, $S(t) \in \{0,1\}^N$, $S_j = \{0,1\}$, 状态转移矩阵为

$$P_j = \begin{Bmatrix} p_{00}^j & p_{01}^j \\ p_{10}^j & p_{11}^j \end{Bmatrix} = \begin{Bmatrix} p_{00}^j & 1 - p_{00}^j \\ p_{10}^j & 1 - p_{10}^j \end{Bmatrix}$$

在 t 时刻感知这 N 个备份 VM 状态,选择当前状态为‘1’的备份 VM,利用这些备份 VM 可以为其他失效 VM 提供备份。

假设存在其他 M 个次优先级 VM,当失效发生时需要感知当前空闲的备份 VM,然后通过迁移的方式占用高优先级 VM 的备份 VM。由于受到计

算资源如 CPU、网络带宽等限制,在每个决策时间槽内,感知备份 VM 的数量受到限制,假设只能感知 K ($1 \leq K \leq N$) 个备份 VM。利用 $A(t)$, $|A(t)| = K$ 表示第 t 个时间槽内做出的决策,即选择出的一组备份 VM,决策 $A(t)$ 会产生相应的回报,回报值是状态 $S(t)$ 的函数定值,即 $R(S(t)) = \sum_{i \in A(t)} s_i(t)$, 当选择备份 VM 的状态为‘1’时可获得 1 个单位的回报,失效 VM 可以占用备份 VM 1 个时间槽,反之不产生回报。在每个决策时间槽内,所有的备份 VM 分成两类:被选择的 VM 称为 Active VM 简记为 VM^a 和没有被选择的 VM 称为 Passive VM 简记为 VM^p 。 VM^a 的状态转移矩阵为 $P_i^a, i \in \{1,2,\dots,K\}$, 即

$$P_i^a = \begin{Bmatrix} p_{i,00}^a & p_{i,01}^a \\ p_{i,10}^a & p_{i,11}^a \end{Bmatrix} = \begin{Bmatrix} p_{i,00}^a & 1 - p_{i,00}^a \\ p_{i,10}^a & 1 - p_{i,10}^a \end{Bmatrix}$$

VM^p 的状态转移矩阵为 $P_i^p, i \in \{1,2,\dots,N-K\}$, 即 $P_i^p = P_i$ 。由于不能完全感知所有备份 VM 的状态,决策算法需要根据以往的决策和观察历史来推断下一时刻的状态,并以此做出决策。为了达到该目的,本文引入在部分可观察 Markov 决策 (POMDP) 领域经常使用的“信念状态”(Belief state) 的概念作为所有备份 VM 的充分统计特征。信念状态定义为:在给定所有备份 VM 的观察历史和决策的前提下每个备份 VM 状态 $s_j(t)=1$ 的概率,用 $\Omega(t)=\{\omega_i(t), i \in \{1,2,\dots,N\}\}$ 表示,其中 $0 \leq \omega_i(t) \leq 1$, 将 $\{0,1\}$ 离散状态转化为 $[0,1]$ 区间上的连续状态空间。由于备份 VM 的状态转移过程符合 Markov 属性,因此每个备份 VM 的信念状态可以利用 Bayes 方法迭代更新,规则为

$$\omega_i(t+1) = \begin{cases} p_{01}^i, & i \in A(t), s_i(t) = 0 \\ p_{11}^i, & i \in A(t), s_i(t) = 1 \\ \tau_i(\omega_i(t)), & i \notin A(t) \end{cases} \quad (1)$$

式中: $\tau_i(\omega_i(t)) = \omega_i(t)p_{11}^i + (1 - \omega_i(t))p_{01}^i$, 表示没有被选择备份 VM 状态的一步更新。如果将每个备份 VM 看成一个摇臂,则该问题演变为不停歇的多臂赌博问题。定义 π 为每个时间槽内从信念状态 $\Omega(t)$ 到决策 $A(t)$ 的映射

$\pi: \Omega(t) \rightarrow A(t), |A(t)| = K, t = 1,2,\dots,T$ 本文目的在于寻找最优策略 π^* , 最大化有限时间区段或者无限时间区段的折现回报值为

$$\pi^* = \operatorname{argmax}_{\pi} E \left[\sum_{t=1}^T \beta^t R_{\pi}(\Omega(t)) \mid \Omega(1) \right] \quad (2)$$

式中: β 为折现因子, $0 \leq \beta \leq 1$, 表示当前的决策产生的后续回报比即时回报的价值小; $R_\pi(\Omega(t))$ 表示信念状态为 $\Omega(t)$ 时, 采取策略 π 产生的回报; $\Omega(1)$ 为初始信念状态, 当 T 为有限值时, 为有限时段的决策, 当 $T \rightarrow \infty$ 时, 为无限时段的决策, 本文只讨论有限时段的决策。

为了评价算法的优劣, 本文提出了失效 VM 的恢复率、备份 VM 的利用率增量和丢失的回报值 3 个度量指标。假设在 t 时刻失效的次优先级 VM 的个数为 f_t , 在 T 个决策时间槽内策略 π 产生的失效 VM 的恢复率定义为

$$A(T, \pi) = \min\left(1, \left(\sum_{i=1}^T R_\pi(S(t))\right) / \sum_{i=1}^T f_t\right)$$

备份 VM 经过分时复用后的利用率比只为高优先级 VM 提供备份时增加了, 在决策时间长度 T 内, 策略 π 产生的备份 VM 的利用率增量定义为

$$\Delta U(T, \pi) = \frac{\sum_{j=1}^N \Delta U^j(T, \pi)}{N}$$

$$\Delta U^j(T, \pi) = \frac{\sum_{t=1}^T \text{ind } s_\pi^j(t)}{T}$$

式中: $s_\pi^j(t)$ 为备份 VM _{j} 在 t 时刻的状态; $\text{ind } s_\pi^j(t)$ 为指示函数

$$\text{ind } s_\pi^j(t) = \begin{cases} 1, & s_\pi^j(t) = 1 \\ 0, & s_\pi^j(t) = 0 \end{cases}$$

丢失的回报值定义为

$$r(T, \pi) = \sum_{i=1}^T \left(\sum_{j=1}^K \mu_j(t) - R_\pi(S(t)) \right)$$

式中: $\mu_j(t)$ 为 t 时刻将所有备份 VM 的回报值按序排列后的第 j 大回报值, 不失一般性地令

$$\mu_1(t) \geq \mu_2(t) \geq \mu_3(t) \geq \dots \geq \mu_N(t)$$

2.2 问题求解

本文将备份 VM 的调度问题转化为不停歇的多臂赌博问题, 解决不停歇的多臂赌博问题的核心思想是将一个 M 维的超高维优化问题分解为 M 个一维优化问题, 计算复杂度大大降低, 本文利用该思想进行求解, 提出了单步启发式算法, 并对解的最优条件进行了分析。

针对 VM 失效模型已知的情况, 本文提出了“单步启发式复用策略”。求解式(2)的最优解, 可以采用动态规划的方法迭代求解, 有限时段 T 内的值函数为

$$V_t(\Omega(t)) = \max_{A(t) \subseteq \{1, 2, \dots, N\}, |A(t)|=K} E[R_\pi(\Omega(t)) +$$

$$\beta \sum_{\epsilon \subseteq A(t)} \prod_{i \in \epsilon} \omega_i(t) \prod_{i \in A(t) \setminus \epsilon} (1 - \omega_i(t)) V(\Omega(t+1))]$$

式中: $V_t(\Omega(t))$ 表示从 $t(1 \leq t \leq T)$ 时刻到 T 时刻期望的最大回报值; ϵ 表示决策 $A(t)$ 中备份 VM 状态为‘1’的集合, $A(t) \setminus \epsilon$ 表示决策 $A(t)$ 中备份状态为‘0’的集合, 信念状态 $\Omega(t+1)$ 按照式(1)更新。由于云系统中备份 VM 的数量巨大, 信念状态 $\Omega(t)$ 的状态空间过于庞大, 导致很难给出值函数的最优解, 因此本文采用单步启发式复用策略, 最大化当前决策产生的即时回报值, 而不考虑决策对系统未来状态产生的影响, 计算复杂度大大降低, 但是解的最优性难以保证, 因此本文讨论了单步启发式复用策略的最优性条件。

单步启发式复用策略的求解目标不再是式(2), 而是退化为

$$\pi^* = \arg\max_{\pi} E[R_\pi(\Omega(t)) | \Omega(1)] \quad (3)$$

只需计算在 t 时刻使得即时回报 $R(\Omega(t))$ 最大化的策略 π , 即将所有备份 VM 在 t 时刻的即时回报 $R(\omega_j(t)) (1 \leq j \leq N)$ 按照降序排列, 选择前 K 个最大回报值对应的备份 VM, 该算法实现了从 N 维优化到一维的降维目标。在利用单步启发式复用策略求解之前, 需要根据 VM 失效模型计算出备份 VM 的状态转移矩阵以及初始信念状态 $\Omega(1)$ 。

当系统达到稳态时, 每个高优先级 VM 进入“正常—失效”的再生循环过程, 其备份 VM 相应地具有“空闲—忙碌”的循环过程。在每个循环周期内, 备份 VM _{j} 处于状态‘1’和‘0’的时间分别为 $T_1^j = E(D_f^j)$, $T_0^j = E(D_r^j)$, D_f^j 、 D_r^j 分别表示备份 VM _{j} 处于空闲状态的时间分布即高优先级 VM 两次失效间隔的概率分布、忙碌状态的时间分布即高优先级 VM 占用备份 VM 的时间分布, $E(\cdot)$ 表示期望值。进而推出其处于状态‘1’和‘0’的概率分别为

$$\pi_0^j = \frac{T_0^j}{T_0^j + T_1^j}; \quad \pi_1^j = \frac{T_1^j}{T_0^j + T_1^j}$$

当高优先级 VM _{j} 的失效模型为参数 $\lambda = 3$ 的 Poisson 分布, 占用备份 VM 的时间分布为参数 $\mu = 1$, $\delta = 0.5$ 的 Lognormal 分布, $T_1^j = 3$, $T_0^j = 2$, $\pi_0^j = 0.4$, $\pi_1^j = 0.6$, 将 T_0^j 和 T_1^j 按照单个时间槽离散化, 如 $T_0^j = 2$ 离散化为“0, 0”, $T_1^j = 3$ 离散化为“1, 1, 1”。稳态情况下可以计算出备份 VM _{j} 的状态转移概率矩阵为

$$P_j = \begin{bmatrix} p_{00}^j & p_{01}^j \\ p_{10}^j & p_{11}^j \end{bmatrix} = \begin{bmatrix} \frac{T_0^j - 1}{T_0^j} & \frac{1}{T_0^j} \\ \frac{1}{T_1^j} & \frac{T_1^j - 1}{T_1^j} \end{bmatrix}$$

通过 Markov 稳态方程计算出稳态情况下, 备份 VM_j 处于‘0’、‘1’的概率为

$$S_i^j = \frac{p_{01}^j}{p_{01}^j + p_{10}^j} = \frac{T_1^j}{T_1^j + T_0^j} = \pi_1^j; S_0^j = 1 - S_i^j = \pi_0^j$$

因此初始信念状态 $\Omega(1) = \{S_i^j, j=1, 2, 3, \dots, N\}$ 。

单步启发式 VM 复用过程如下。

输入: 备份 VM 的数量 N , 可同时感知 VM 数量 K , 决策时间长度 T , 以及高优先级 VM 的失效和恢复时间模型 D_f^j 和 $D_r^j, 1 \leq j \leq N$ 。

输出: 从 t 到 $t+T$ 之间选择出备份 VM 序列。

(1) 计算每个备份 VM 的转移矩阵 $\mathbf{P}$ 以及初始信念状态 $\Omega(1)$;

(2) 在 t 时刻对所有备份 VM 的信念状态进行排序;

(3) 选择前 K 个信念状态最大的备份 VM;

(4) 感知由步骤(3)选出的 K 个备份 VM 的状态, 将失效 VM 调度到状态为‘1’的备份 VM 上;

(5) 根据式(1)更新信念状态, 计算 $t+1$ 时刻的调度决策;

(6) 重复步骤(2)~(5)直至 $t+T$ 时刻。

2.3 最优解条件

文献[10]采用相同的方法求解机会频谱访问问题(OSA), 证明了当 $p_{11} > p_{01}$ 时, 利用启发式算法求解 OSA 问题具有最优解。

定理 1 对于具有两状态‘0’和‘1’的 N 个通信信道, 且每个信道遵从独立同分布的 Markov 过程的 OSA 问题, 无论从 N 个信道中选择一个还是 K ($1 < K \leq N$) 个, 单步启发式算法在条件 $p_{11} > p_{01}$ 时, 解是最优的。

本文假定每个备份 VM 的状态按照独立同分布的 Markov 过程演进, 如果将每个备份 VM 类比成一个通信信道, 则备份 VM 的复用问题等价于文献[10]提出的 OSA 问题, 故定理 1 在本文中成立。但是, 在本文提出的 VM 复用问题中 p_{11} 和 p_{01} 并不是已知条件, 本文给出如下最优解条件。

推论 1 对于备份 VM 的复用问题, 满足条件 $\frac{1}{E(D_f)} + \frac{1}{E(D_r)} < 1$ 时, 单步启发式策略具有最优解。

证明 由于 $T_1 = E(D_f)$, $T_0 = E(D_r)$, $p_{01} = \frac{1}{T_0}$, $p_{11} = \frac{T_1 - 1}{T_1}$ (每个备份 VM 的状态转移矩阵相同, 因此去掉了上标 j), 因此当 $p_{11} > p_{01}$ 时有

$$p_{11} > p_{01} \Rightarrow \frac{T_1 - 1}{T_1} > \frac{1}{T_0} \Rightarrow \frac{1}{T_0} + \frac{1}{T_1} < 1 \Rightarrow$$

$$\frac{1}{E(D_f)} + \frac{1}{E(D_r)} < 1$$

证毕。

2.4 复杂度分析

由单步启发式算法的描述可以看出, 算法主要包括对备份 VM 信念状态的排序过程和对备份 VM 状态的更新过程。给定 N 个备份 VM, 决策时间长度为 T , 采用快速排序方法, 在一个时间槽内排序过程的时间复杂度为 $O(N \log N)$, 状态更新过程的时间复杂度为 $O(N)$, 总的时间复杂度为 $O(N \log N)$, 因此在 T 个时间槽内的时间复杂度为 $O(T^* N \log N)$ 。与 POMDP 等方法的指数复杂度相比, 单步启发式算法的时间复杂度大大降低了。

3 实验结果

首先, 根据高优先级 VM 的失效、恢复规律, 产生了备份 VM 的初始利用规律。为了模拟 VM 失效模型及恢复模型, 利用 Matlab 按照“正常—失效”循环产生符合特定分布的随机数。“正常”随机数表示高优先级 VM 的失效间隔时间, 也是备份 VM 的空闲时间; “失效”随机数表示高优先级 VM 的修复时间, 也是备份 VM 的忙碌时间。然后, 将随机数按照单位时间离散成‘0’、‘1’二值序列, 表示特定时刻某个 VM 所处的状态, 例如利用失效模型产生随机数 4, 二值化为“1111”, 表示在 4 个连续时间槽内备份 VM 是空闲的。本文共模拟了 N ($N=50, 100, 150, 200, 500$) 个备份 VM, 每个备份 VM 的失效及恢复模型都是独立同分布的, 且包含 1 000 个“0, 1”值。在每个时间槽内 (模拟时间单位为 min), 利用单步启发式复用算法选择 K 个备份 VM, 按照感知能力的不同, 分析 K 取不同值时的算法性能, 另外本文针对不同的次优先级 VM 失效率 f , 讨论了有限备份次优先级 VM 的数量上限。模拟实验程序采用 Matlab 实现, 通过实验分析, 回答了以下 3 个问题。

问题一 单步启发式算法产生的调度方案效果。

在此实验中, 备份 VM 数量 N 为 500, 初始阶段每个备份 VM 空闲时间分别服从参数 $\alpha=17, \beta=1$ 的 Weibull 分布, 忙碌时间服从参数 $\mu=1, \delta=0.5$ 的 Lognormal 分布, 每个 VM 的平均效率为 0.05, 模块可感知的备份 VM 数量为 400, 决策时间长度为 1 000 个时间槽。利用单步启发式复用策略得出的 regret 的对数如图 1 所示。由图中观察得到, regret 的对数值随着决策时间增加呈现收敛趋势, 说

明单步启发式复用策略具有对数最优。这是因为随着决策次数增加,单步启发式策略不断感知新的备份 VM 的状态,单步更新的“信念状态”越来越能精确反映每个备份 VM 的“权重”,选择策略也渐进趋向最优策略。

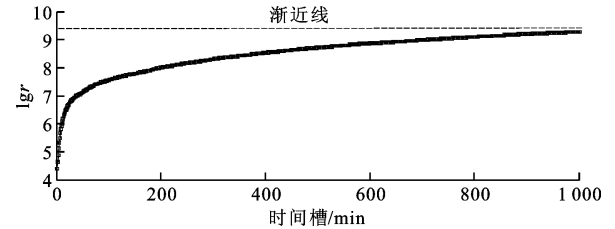


图 1 regret 的对数随决策时间的变化曲线

图 2 展示了采用单步启发式策略对空闲的备份 VM 进行调度后,每个备份 VM 的利用率增量。从图中看出,单步启发式复用策略在上述场景下将备份 VM 的利用率平均提高了 86.5%,使备份 VM 的空闲时段得到了充分利用,节省了资源。

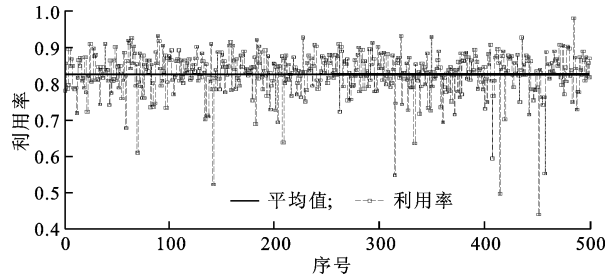


图 2 每个备份 VM 的利用率增量

问题二 当系统参量变化时,单步启发式算法产生的效果变化。

在云系统中,软件运行时的内外环境频繁变动,本文主要探讨了高优先级 VM 的失效率变化、失效模型类型变化、可感知的备份 VM 的数量变化对单步启发式复用策略的影响。

(1)失效率变化。本文通过调整失效模型或者恢复模型的参数达到调整失效率的目的。设定失效模型为 Weibull 分布,恢复模型为 Lognormal 分布, $K=400$,备份 VM 的数量为 500。讨论表 1 中 4 种情况下单步启发式复用策略的效果。

表 1 失效率及相应的模型参数

失效率	模型参数			
	Weibull 分布		Lognormal 分布	
0.01	$\alpha=97$	$\beta=1$	$\mu=1$	$\delta=0.5$
0.05	$\alpha=17$	$\beta=1$	$\mu=1$	$\delta=0.5$
0.10	$\alpha=7$	$\beta=1$	$\mu=1$	$\delta=0.5$
0.20	$\alpha=2$	$\beta=1$	$\mu=1$	$\delta=0.5$

图 3 比较了不同失效率情况下,500 个备份 VM 经过复用后的平均利用率增量。随着失效率的增加,备份 VM 的利用率增量逐渐降低。这是因为失效率增加后,每次经过单步启发式策略选择出的 400 个备份 VM 集合中状态为‘0’的备份 VM 增加,即与高优先级 VM 出现资源争用的情况增多,但即使在高失效率 0.2 的情况下,单步启发式复用策略仍然获得了 71.7%的利用率增量。与利用率增量变化情况相似,regret 也随着失效率增加而增加,如图 4 所示。

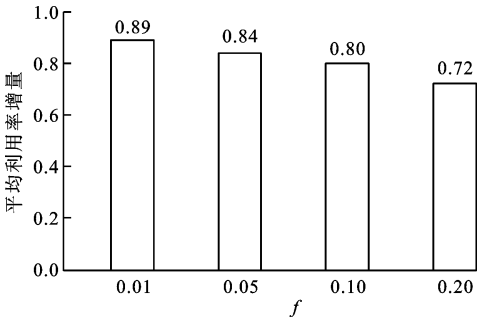


图 3 不同失效率场景下的平均利用率增量

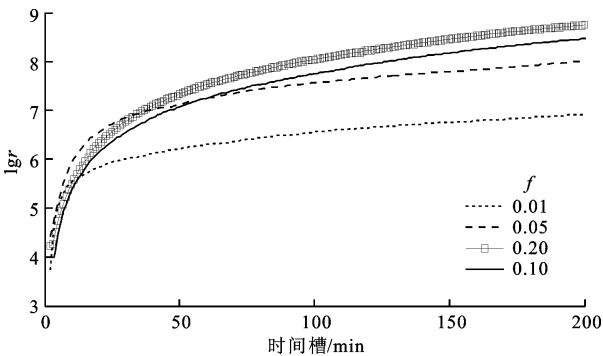


图 4 不同失效率场景下的 regret 的对数的变化情况

(2)失效模型类型变化。固定失效率为 0.05, $K=400$,备份 VM 的数量为 500,讨论两种分布:失效模型采用参数 λ 为 17 的 Poisson 分布($d-1$);失效模型采用参数 $\alpha=17$ 、 $\beta=1$ 的 Weibull 分布,恢复模型均为参数 $\mu=1$ 、 $\delta=0.5$ 的 Lognormal 分布($d-2$)。图 5 展示了 $d-1$ 、 $d-2$ 两种情况下备份 VM 的利用率增量,从图中可以看出,利用率增量相差并不大,平均增量都是 0.84,说明单步启发式策略产生的效果在失效率相同的情况下与失效模型关系不大。这是因为失效和恢复模型都是单步启发式策略的已知条件,并据此进行单步状态更新。图 6 展示了 $d-1$ 、 $d-2$ 两种情况下备份 VM 在 1 000 个时间槽内的 regret 变化曲线,两条曲线几乎重叠,验证了上述结论。

(3) K 值变化。固定失效模型和恢复模型,备

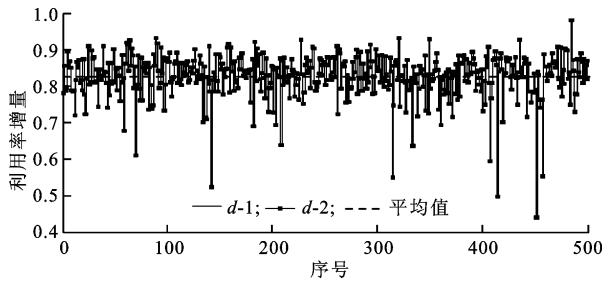


图 5 $d-1$ 、 $d-2$ 两种情况下备份 VM 的利用率增量

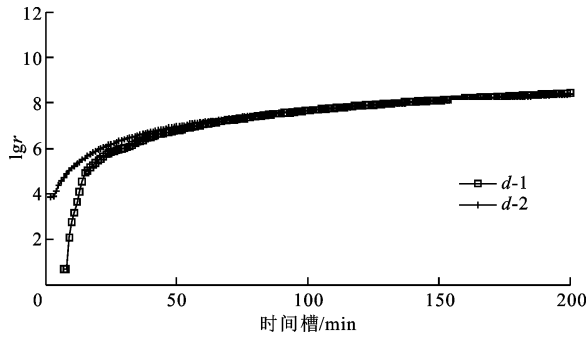


图 6 $d-1$ 、 $d-2$ 两种情况下备份 VM 的 regret 的对数的变化曲线

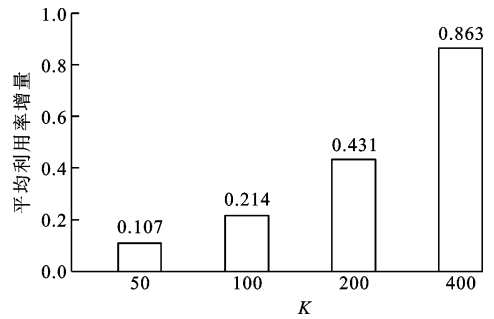


图 7 利用率增量随 K 值变化的情况

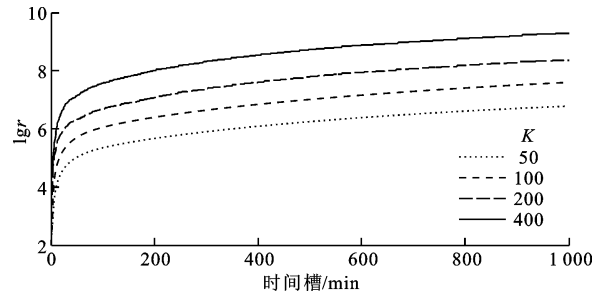


图 8 VM 失效率为 0.05 时 regret 的对数与 K 之间的关系

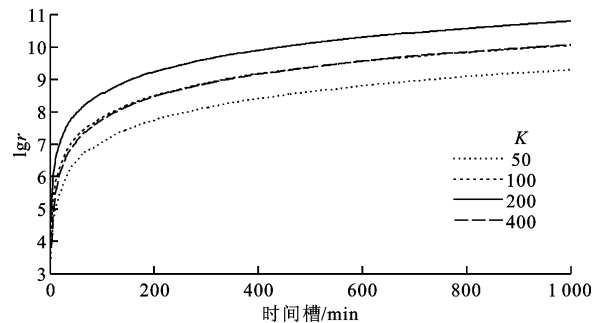


图 9 VM 失效率为 0.2 时 regret 的对数与 K 之间的关系

份 VM 的数量为 500,调整 K 值,观察单步启发式策略的效果。图 7 展示了失效模型参数 $\alpha=17$ 、 $\beta=1$ 的 Weibull 分布,恢复模型参数 $\mu=1$ 、 $\delta=0.5$ 的 Lognormal 分布时备份 VM 的平均利用率随 K 值的变化情况,从图中可以直观看出,利用率增量随着 K 值增加而增加。这是因为 K 值决定了可感知的备份数量以及备份 VM 被调度的数量,被调度的备份 VM 数量越多,利用率增量也就越大,而且 K 与利用率增量之间遵循线性关系。regret 的对数与 K 值之间却没有单调的线性关系,图 8 展示了 VM 失效率较低时的 regret 变化情况,regret 的对数值随着 K 值增加而增加。但是,当 VM 失效率较高时, $K=100$ 、400 时的 regret 对数曲线几乎重叠在一起,与 K 值之间不存在单调线性关系,如图 9 所示。

通过上述分析发现,单步启发式复用策略的决策效果受 VM 失效率以及可感知的备份 VM 数量 K 的影响较大,但是在失效率固定时不受失效模型类型的影响。上述 3 种因素与决策效果之间存在复杂的线性以及非线性关系,本文只给出了定性分析,定量关系将在后续工作中进行讨论。

问题三 在满足可用性要求的前提下,备份 VM 能备份的次优先级 VM 数量的上限。

本实验设定备份 VM 的数量 $N=500$,可感知的备份 VM 的数量 $K=400$,失效模型采用参数 $\alpha=$

17 、 $\beta=1$ 的 Weibull 分布,恢复模型采用参数 $\mu=1$ 、 $\delta=0.5$ 的 Lognormal 分布,讨论当次优先级 VM 在失效率为 0.01、0.05、0.1、0.2 时,备份 VM 可备份的次优先级 VM 的数量上限。图 10 展示了失效 VM 的覆盖率随次优先级 VM 数量变化的曲线。以失效 VM 覆盖率 96% 为基准,从图 10 中可以看出,当失效率为 0.01、0.05、0.1、0.2 时,500 个备份 VM 可以备份的 VM 上限分别为 9 000、8 000、6 000 和 2 000,备份比例为 1:18、1:16、1:12 和 1:4,即使在次优先级 VM 频繁失效的情况下,备份比例仍可以达到 1:4。在保证失效 VM 被 100% 覆盖的情况下,单步启发式策略仍可以在次优先级失效率较高时获得 1:3 的备份比。

本实验讨论单步启发式算法的运行时间与可感知的备份 VM 的数量 K 和备份 VM 的数量 N 之间

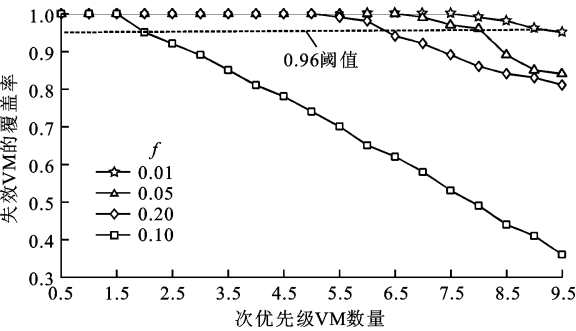


图 10 失效 VM 的覆盖率随次优先级 VM 数量变化曲线

的关系,算法的执行单步决策即 $T=1$ 时的运行时间如表 2 所示。从表 2 可以看出,算法的执行时间处在 ms 级水平,完全可以做到实时决策,而且运行时间随着 N 值增加而增加,两者存在单调变化关系,但与 K 值之间不存在单调变化关系,说明单步启发式算法的执行时间与 N 相关但是与 K 无关。

表 2 单步启发式算法执行单步决策的运行时间 ms

N	运行时间			
	K=50	K=100	K=200	K=500
50	0.05			
100	0.088	0.09		
150	0.13	0.133		
200	0.2	0.21	0.195	
500	0.93	0.932	0.935	0.934

4 结 论

为了充分利用过度冗余的备份虚拟资源,本文提出了基于 RMAB 问题的单步启发式复用策略。该策略将单个备份 VM 描述为独立同分布的 Markov 随机过程,按照特定的概率转移矩阵进行单步状态更新,通过最大化单步回报达到备份 VM 利用率最大化的目的。此外,本文给出了在特定失效、恢复模型下的最优解条件,从理论上证明了单步启发式复用策略的最优性。模拟实验结果表明,单步启发式复用策略可以将传统的 1:1 备份扩展成 1:M($M \gg 1$),同时保证失效 VM 的恢复率不低于 96%,相应备份虚拟资源的利用率显著提高。在 VM 失效率较低条件下,备份虚拟资源利用率比 1:1 备份时提高了 89%,从而可以有效减少云计算平台的建设和运维费用。

参考文献:

[1] JAYASINGHE D, PU C. Improving performance and availability of services hosted on IaaS clouds with

structural constraint-aware virtual machine placement [C]// Proceedings of the 8th IEEE International Conference on Services Computing. Piscataway, NJ, USA: IEEE, 2011: 72-79.

[2] VERMA A, DASGUPTA G, TAPAN K, et al. Server workload analysis for power minimization using consolidation [C]// Proceedings of the 2009 Conference on USENIX Annual Technical Conference. Berkeley, CA, USA: USENIX, 2009: 28.

[3] BIN E, BIRAN O. Guaranteeing high availability goals for virtual machine placement [C]// Proceedings of the 31st International Conference on Distributed Computing Systems. Piscataway, NJ, USA: IEEE, 2011: 700-709.

[4] WHITTLE P. Restless bandits: activity allocation in a changing world [J]. Journal of Applied Probability, 1988, 25(2): 287-298.

[5] SCHROEDER B, GIBSON G A. A large-scale study of failures in high-performance computing systems [J]. IEEE Transactions on Dependable and Secure Computing, 2010, 7(4): 337-350.

[6] OKAMURA H, TADASHI D, SHUNJI O. Software reliability growth models with normal failure time distributions [J]. Reliability Engineering & System Safety, 2013, 11(6): 135-141.

[7] CULLY B, LEFEBVRE G. Remus: high availability via asynchronous virtual machine replication [C] // Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation. Berkeley, CA, USA: USENIX, 2008: 161-174.

[8] SINGH R, IRWIN D, SHENOY P, et al. Yank: enabling green data centers to pull the plug [C] // Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation. Berkeley, CA, USA: USENIX, 2013: 143-155.

[9] WANG D, GOVINDAN S, ANAND S, et al. Under provisioning backup power infrastructure for data-centers [C] // Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems. New York, USA: ACM, 2013: 177-192.

[10] AHMAD S, HAJI A, LIU M. Multi-channel opportunistic access: a case of restless bandits with multiple plays [C] // Proceedings of the 47th Annual Allerton Conference on Communication, Control and Computing. Piscataway, NJ, USA: IEEE, 2009: 1361-1368.

(编辑 赵炜)