

基于随机森林的虚拟机性能预测与配置优化

张彬彬 王娟 岳昆 武浩 郝佳

(云南大学信息学院 昆明 650500)

摘要 在目前的 IaaS 云计算服务中,用户可租用不同资源配置的虚拟机,然而用户很难根据资源配置准确估计虚拟机的性能,从而较难根据待部署的应用的性能需求选择恰当配置的虚拟机,这种使用方式使得云主机的资源未得到最充分的利用。因此,文中提出基于随机森林回归模型预测特定配置的虚拟机性能,并在此基础上,根据性能需求,利用遗传算法求解较优的符合性能需求的虚拟机配置,用随机森林性能模型获取种群中各个体的性能预测值以选出最接近性能需求的个体进行交叉操作。实验结果表明,随机森林回归模型能准确预测特定配置的虚拟机的性能,利用遗传算法搜索得出的虚拟机配置的实测性能与性能需求非常接近,并且该算法可以在较短时间内达到收敛。

关键词 云计算,虚拟机,性能预测,配置优化,随机森林,遗传算法

中图分类号 TP302 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.09.011

Performance Prediction and Configuration Optimization of Virtual Machines Based on Random Forest

ZHANG Bin-bin WANG Juan YUE Kun WU Hao HAO Jia

(School of Information Science and Engineering, Yunnan University, Kunming 650500, China)

Abstract In IaaS cloud computing, users rent one or more virtual machines with different resource configurations. However, it is difficult for users to accurately estimate the performance of the virtual machine according to the resources allocated. Thus it is hard for them to select an appropriate virtual machine according to the performance requirement of the applications. Therefore, this paper proposed to predict performance of the virtual machine according to their resources and configurations based on random forest. Further, it proposed to use genetic algorithm to search the optimal configuration of the virtual machine which can meet the performance requirement. The difference of the prediction result and the target performance are used as the fitness function. The experimental results show that the random forest model can accurately predict performance of the virtual machine. And the actual performance of the virtual machine configured according to the configuration obtained by the genetic algorithm is very close to the performance requirement, and the convergence can be achieved in a short time.

Keywords Cloud computing, Virtual machine, Performance prediction, Configuration optimization, Random forest, Genetic algorithm

1 引言

在基础设施即服务 (Infrastructure as a Service, IaaS) 的云计算服务模式中,用户以虚拟机 (Virtual Machine, VM) 的形式租用计算、存储、网络等基础资源,并根据租用的资源数量和持续时间缴付租金。用户可在虚拟机上运行客户操作系统,并在操作系统之上部署自己的应用。亚马逊弹性云 (Elastic Compute Cloud, EC2)、微软 Azure、阿里云、腾讯云等都属于 IaaS 服务模式。

目前,上述各平台均提供若干不同配置的虚拟机供用户选择,虚拟机之间的配置差异主要是虚拟机占用的资源(即虚拟资源)的数量差别。如微软 Azure 提供的可选虚拟机具有不同的 CPU 核心数量、内存容量、硬盘容量。腾讯云与微软 Azure 类似,用户可定制 CPU 核心数量、内存容量、硬盘容量、网络带宽等虚拟资源配置。阿里云为不同类型的资源需求提供不同类别的可选虚拟机,如计算型、内存型、I/O 型、通用型等,每种类型提供一组不同配置的虚拟机,可选配置包括 CPU 核心数量、内存容量、硬盘容量和类型、网络带宽等。亚

到稿日期:2018-07-21 返修日期:2018-09-16 本文受国家自然科学基金项目(61402398, U1802271, 61562090), 云南大学青年英才培育计划项目(WX173602)资助。

张彬彬(1982-),女,博士,讲师,CCF 会员,主要研究方向为虚拟化、云计算;王娟(1992-),女,硕士生,主要研究方向为数据分析、虚拟化;岳昆(1979-),男,博士,教授,博士生导师,CCF 高级会员,主要研究方向为海量数据分析与服务、大数据知识工程,E-mail:kyue@ynu.edu.cn (通信作者);武浩(1979-),男,博士,副教授,主要研究方向为 Web 信息处理、服务计算;郝佳(1993-),女,博士生,主要研究方向为数据分析、虚拟化。

亚马逊 EC2 服务为用户提供了更细化的虚拟机配置信息,包括虚拟 CPU(vCPU)个数、内存容量、硬盘容量和类型、网络性能、CPU 类型、主频,以及其他硬件特性。各平台主要根据资源数量对虚拟机进行定价,用户根据待部署应用对虚拟机的性能需求,选择租用一个或多个特定资源配置的虚拟机。

然而,仅根据上述各平台展示给用户的虚拟资源数量和少量可选的资源类型,用户很难按照待部署应用的性能需求选择恰当的虚拟机配置。其主要原因有以下两个方面。

(1)除虚拟机占用的虚拟资源数量之外,虚拟机的性能也会受到虚拟化环境中多方面特征的影响。在上述 IaaS 平台上,用户可见的各资源特征只是虚拟机的一部分资源配置特征,如 vCPU 数量、内存容量、磁盘容量和类型、网络带宽等,以及少量硬件特征,如 CPU 类型、主频等。而在虚拟化环境中,物理主机的硬件特性、虚拟机监控器(Virtual Machine Monitor, VMM)的软件特性、虚拟机资源配置特性、虚拟机运行时的环境特性都会影响虚拟机的实际运行性能。

(2)虚拟机的各个特征与其性能之间的关系难以量化。主要原因在于:影响虚拟机性能的特征较多;特征之间存在相关关系,各特征之间并非完全独立,如 CPU 虚拟化方式受到硬件平台的 CPU 类型的影响;特征与性能之间不是简单的线性关系。

因此,本文提出根据用户的性能需求搜索合适的虚拟机配置。一方面避免用户过度预留资源造成浪费,另一方面,提高整个虚拟化平台的资源利用率,优化云平台的整体性能。为了达到这一目标,本文完成了以下两个方面的工作。

(1)利用随机森林回归模型对虚拟机性能及虚拟机特征之间的关系进行建模,当给定一个虚拟机的配置及运行时环境等特征时,能较为准确地预测该虚拟机运行特定类型应用程序时的性能。随机森林回归可以看作由很多弱分类器集成的预测器,近年来被广泛应用于多个领域。随机森林算法能够处理特征之间的非线性关系,能很好地适应高维度的数据、处理连续型和离散型数据,其训练数据集的规模对预测结果准确率的影响比多层神经网络模型要小。因此,我们选用随机森林来预测虚拟机性能。

(2)基于随机森林性能预测模型,本文提出一种利用遗传算法根据特定性能搜索得到满足用户性能需求的虚拟机配置的方法。由于虚拟化环境中,影响性能的特征较多,搜索空间大,使得满足性能需求的虚拟机配置优化问题难以用全局搜索算法在合理的时间内求解。而遗传算法能同时对搜索空间中多个可能的方向做并行搜索,利用适应度信息使搜索向最优的方向推进,并且采用了概率变迁规则避免其陷入局部最优解。因此,本文采用遗传算法求解配置优化问题。在遗传算法中选择父本产生下一代虚拟机特征取值时,将适应度函数设计为利用随机森林预测出的各虚拟机性能值与目标性能值的距离,采用轮盘赌算法选出一批较为接近目标性能的个体,经过多轮遗传迭代,搜索得到较为接近目标性能值的虚拟机配置,以满足用户在租赁虚拟机时的性能需求,使用户得到更好的用户体验,同时指导资源提供方优化虚拟机

的部署和资源的动态分配。

实验表明,本文实现的随机森林回归模型能准确预测虚拟机的性能,在此基础上设计的虚拟机配置优化算法能快速找到较为接近目标性能值的虚拟机配置。

本文第 2 节介绍相关研究工作;第 3 节提出基于随机森林预测虚拟机性能的方法;第 4 节介绍基于随机森林的性能预测模型,以及利用遗传算法搜索虚拟机配置的原理和实现;第 5 节展示了实验的结果;最后总结全文,并指出后续的研究方向。

2 相关工作

对于虚拟机的性能建模和预测研究,一种途径是发现虚拟机的资源利用率与虚拟机性能或虚拟机上运行的应用性能之间的关系,辅助实现虚拟机的性能评估和预测。文献[1]利用人工神经网络对应用的性能与硬件资源利用率之间的关系进行建模。文献[2]基于排队论构造性能与虚拟机个数、CPU 需求、VMM 的性能开销等特征之间的关系模型,并将该模型用于预测虚拟机性能。文献[3]通过收集 I/O 请求响应时间的变化信息,构造排队模型,以预测在虚拟化环境中由于虚拟机间资源竞争而带来的磁盘 I/O 性能下降的程度。文献[4]根据虚拟机的资源利用率构造线性模型来预测虚拟机性能。这一类工作根据资源利用率预测虚拟机性能,适用于虚拟机运行时的性能分析。而本文的性能预测模型是虚拟机资源分配决策的基础,其更关注虚拟资源数量、虚拟机运行时环境、虚拟机配置等特征与性能之间的关系。

另一种途径是通过寻找虚拟机的部分配置参数与虚拟机性能或虚拟机上运行的应用性能之间的关系,从而根据虚拟机配置评估虚拟机性能。文献[5]收集了 CPU 时间周期权重、虚拟 CPU 的数量、内存以及 I/O 竞争 4 种硬件资源单独对虚拟机应用性能的影响作为建模参数,利用基于奇异值分解的特征拓展与非线性模型的建模方法,对硬件与虚拟机性能的关系进行建模。文献[6]提出了一种虚拟机性能预测方法,其中虚拟机监控模块根据预设的资源配置标准,随机地改变虚拟机的资源配置,将每一种资源配置下的云计算系统的响应时间和当前虚拟机配置的资源集合合并生成向量,将一组资源配置生成的向量作为输入,使用支持向量机算法并根据输入的向量生成性能预测模型,该模型被用来预测在任意资源配置下的云计算系统的响应时间。所设置的资源配置标准涉及内存、CPU、网络带宽 3 种资源,包括每台虚拟机的最低资源耗费率及每台物理主机的资源容量。文献[7]把不同类型应用的不同表现考虑进去,提出了一个简单的排队模型来评估不同的应用在虚拟化环境中的性能。然而,虚拟机的性能受到从硬件到运行时环境中的若干特征的影响,以上工作仅关注其中的部分特征。

在虚拟机的资源分配和任务调度研究中,通常将虚拟机的资源分配问题抽象化为最优化问题,优化目标包括负载均衡、性能最优、开销最低等。这类问题的搜索空间巨大,一般采用遗传算法等近似算法求得近似解。典型的虚拟机迁移目标主机决策和虚拟机放置决策问题等通常被抽象化为装箱问

题,即将主机当作有容量的箱子,将虚拟机当作有一定体积的物品,采用遗传算法求得近似解^[8-12]。文献[13]关注任务的执行时间与处理器之间的负载均衡,提出了一种基于关键路径的遗传算法,实现任务在不同处理器之间的动态转移决策。文献[14]为了满足云计算中用户的需求并改善资源利用率,采用改进的遗传算法求解任务调度决策,采用贪心法初始化种群,并设计了综合考虑总时间及平均时间的适应度函数。文献[15]也采用遗传算法求解迁移方案,利用适应度函数的设置试图找到迁移次数最少和剩余资源最为均衡的方案。针对虚拟化数据中心的用户请求的动态变化,文献[16]基于遗传算法搜索配置空间,能够快速搜索出最优配置。本文的虚拟机配置搜索工作也是一个资源分配问题,但与上述工作的出发点不同,本文的优化目标是在配置空间中找到能达到用户要求的目标性能的虚拟机配置。

3 基于随机森林预测虚拟机性能

3.1 虚拟机特征选取

在虚拟化环境中,存在多种可能影响虚拟机性能的因素。从硬件到运行时环境,我们可以把这些因素分为 4 个层次:1)硬件特征,如 CPU 微架构、主频、Cache 容量和访问延迟、内存容量和访问延迟、总线带宽、硬盘类型等;2)软件特征,如虚拟化平台采用的 CPU 调度方式、为各虚拟 CPU 配置的调度优先级、内存的虚拟化方式、I/O 设备的虚拟化方式等;3)配置特征,如为虚拟机配置的 vCPU 数量、内存容量、I/O 带宽等;4)运行时环境特征,如同一台物理主机上同时运行的虚拟机数量、同时运行的虚拟机上的应用类型、多主机之间进行远程内存访问或虚拟机迁移的频度和持续时间等。表 1 列出了其中典型的影响虚拟机性能的特征。

表 1 影响虚拟机性能的特征

Table 1 Properties affecting performance of VM

硬件特性	VMM 软件特征	虚拟机资源 配置特征	运行时 环境特征
CPU 类型	VMM 类别	vCPU 数量	同时运行的 虚拟机数量
CPU 主频	CPU 虚拟化 方式	vCPU-CPU 绑定方式	同时运行的 虚拟机的负载类型
内存类型	内存虚拟化方式	虚拟机 内存容量	主机的资源 利用率
硬盘类型	I/O 设备 虚拟化方式	虚拟块 设备类型	主机是否正在执行 虚拟机迁移
网络带宽	调度算法	虚拟机网络带宽	...
...

我们调整这些特征的取值,配置不同类型的 n 个虚拟机,在其上运行基准测试程序并记录运行结果,用于量化虚拟机的性能。将虚拟机各项特征的取值、虚拟机的性能取值记录下来,得到虚拟机特征-性能数据样本。该样本形如 $D_i = (X_i, Y_i)$,其中 D_i 表示第 i 种虚拟机特征-性能值的样本,其中 $X_i = (X_{i,1}, X_{i,2}, \dots, X_{i,F})$ 表示第 i 台虚拟机的特征, $X_{i,j}$ 表示第 i 台虚拟机的第 j 个虚拟机特征 x_j 的取值, Y_i 表示第 i 台虚拟机 X_i 对应的性能指标值, $i=1, 2, \dots, n, j=1, 2, \dots, F$ 。

3.2 基于随机森林回归模型的虚拟机性能预测

首先根据 3.1 节获取的虚拟机特征-性能数据样本构造虚拟机性能预测模型。该模型以某一组特定的虚拟机配置参数作为输入,以虚拟机的性能指标取值作为输出。

随机森林回归算法是 Breiman 提出的基于决策树分类器的融合算法,从实际观测到的含有 n 个样本的数据集中有放回地随机抽取 $s (s \leq n)$ 个样本组成一个新样本,该样本称为 Bootstrap 样本。对每个 Bootstrap 样本构建决策树。预测时,对所有决策树分别进行预测,将预测结果的平均值作为最终的预测结果。

如 3.1 节所述,虚拟化环境中存在很多可能影响虚拟机性能的特征,各个特征变量之间的关系复杂且呈非线性,广义线性模型难以描述特征之间的复杂关系,无法得出比较精确的回归模型。随机森林算法能够处理特征之间的非线性关系,对高维度数据的适应度较好,且不需要做特征选择。其在训练过程中还可以得到变量的重要性排序,便于提取影响虚拟机性能的关键特征。

因此,本文基于随机森林回归算法,构建虚拟机特征与虚拟机性能的回归模型。通过在典型软硬件配置的虚拟机上运用基准测试工具获取其性能数据,对虚拟机的软硬件配置特征及虚拟机性能之间存在的依赖关系进行建模,基于模型分析虚拟机各特征对性能影响的重要性排序、虚拟机性能与特征之间的关系,并通过构建的模型来预测特定配置下的虚拟机的性能。

本文以分类回归树 (Classification And Regression Tree, CART) 算法对每个 Bootstrap 样本构建决策树,每棵决策树的样本是随机抽取的,每棵决策树的分裂变量是随机的,如此生成大量决策树,构成随机森林。与决策树相比,随机森林回归模型不容易出现过拟合的情况。该算法的思想如下所示。

(1)在原始数据 n 个样本中,采用 Bootstrap 方法有放回地多次随机采样,抽取 $\gamma \times n$ 个数据样本作为第 t 棵决策树的训练集 D_t , γ 的取值范围为 $0 < \gamma \leq 1$ 。把第 t 棵决策树的训练集 D_t 作为该棵决策树根节点处的样本。

(2)当每个样本有 F 个特征时,在决策树的每个节点依次按照以下方法进行节点分裂:记节点的样本集合为 S ,如果样本集合中的样本数目小于预设阈值 TS ,则本节点作为叶子节点,不再分裂;否则从 F 维虚拟机特征中无放回地随机选取 f 维特征构成集合 P_s , f 的取值范围为 $1 \leq f \leq F$ 。一般地,在回归模型中, f 的取值为 F 的 $1/3$ 。对于虚拟机特征集合 P_s 中的每个特征 x_j ,即 $x_j \in P_s$,分别选取 K 个分裂点 $\lambda_j(i) (i=1, 2, \dots, K)$, K 表示分裂点数量, K 个分裂点一般在特征 x_j 的取值范围内随机选取。计算出节点样本集合 S 以不同特征 x_j 和分裂点 $\lambda_j(i)$ 进行分裂时的方差,筛选出最小方差,如果最小方差小于设定的最小分裂阈值 T_λ (通常取 $T_\lambda \ll 0.01$),则将本节点作为叶子节点,不再分裂;否则选择最小方差对应的特征 \tilde{x} 和分裂点 $\tilde{\lambda}$ 作为本节点的分裂特征和分裂点,该节点即为最佳分裂节点,将节点样本集合 S 中分裂特征 \tilde{x} 的取值小于或等于 $\tilde{\lambda}$ 的样本划分至左子节点,大于 $\tilde{\lambda}$ 的样本划

分至右子节点,子节点对应的性能指标值为子节点样本集中所有样本的性能指标的平均值。如此进行下去,一直到不能够再分裂为止。在整个决策树的生成过程中没有进行剪枝。

(3)重复步骤(1)步骤(2),建立 nt 棵决策树。

上述思想的一种实现如算法 1 所示。

算法 1 随机森林回归模型的构建算法

输入:训练数据集 D ,虚拟机特征集合 P ,构建的随机森林中决策树的数量 nt ,一个节点的最少样本数量 T_s

输出:随机森林回归模型 RF

```

1. For  $i \leftarrow 1$  to  $nt$  Do
2.   $D_s \leftarrow \text{bootstrap}(D)$  //用 Bootstrapping 方法从虚拟机特征-性能数据样本  $D$  中抽取  $D_s$  个样本
3.   $\text{root} \leftarrow \text{makeNode}(D_s)$  //将  $D_s$  中的全部样本作为根节点
4.   $\text{insert}(\text{FIFO}, \text{root})$  //把根节点加入到先进先出队列中
5.  WHILE(1) Do
6.    IF  $\text{empty}(\text{FIFO})$  THEN
7.      break //如果队列为空,则该棵决策树已构建完毕
8.    ENDIF
9.     $\text{node} \leftarrow \text{pop}(\text{FIFO})$ 
10.   IF( $\text{sampleSize}(\text{node}) \geq T_s$ ) THEN //如果在该节点的样本数量少于  $T_s$ ,则该节点将不再分裂,产生一个叶子节点,否则分裂该节点
11.      $P_s \leftarrow \text{randomSubset}(P)$  //随机从特征集合  $P$  中选择  $f$  个特征作为集合  $P_s$ 
12.      $\text{bestSplit}(\text{node}, P_s)$  //用最佳分裂特征分裂该节点
13.      $\text{insert}(\text{FIFO}, \text{node. left})$  //把左孩子加入到 FIFO 队列中
14.      $\text{insert}(\text{FIFO}, \text{node. right})$  //把右孩子加入到 FIFO 队列中
15.   ENDIF
16. ENDWHILE
17.  $\text{addtoRF}(\text{RF}, \text{root})$  //把新生成的树加入随机森林
18. ENDFOR
19. Return RF //得到随机森林

```

由算法 1 得到随机森林回归模型,利用该模型可以预测任意特征取值的虚拟机性能。

对于待预测性能的虚拟机配置 $\mathbf{X}_i = (x_1, x_2, \dots, x_F)$,将特征向量 \mathbf{X}_i 输入决策树的根节点,从根节点出发,若 \mathbf{X}_i 对应当前节点的分裂特征的 x_i 取值小于或等于分裂点的值,则 \mathbf{X}_i 进入左子树,若 x_i 的取值大于分裂点的取值,则 \mathbf{X}_i 进入右子树,直到 \mathbf{X}_i 到达某叶子节点。输出在该叶子节点下的性能值作为单棵决策树对虚拟机的性能预测值。对 nt 棵决策树的预测值取平均值作为虚拟机性能的预测值。图 1 是具有虚拟机内存、vCPU 个数两个特征的一棵决策树的示例,其输出是性能值,由基准测试程序的运行时间量化。

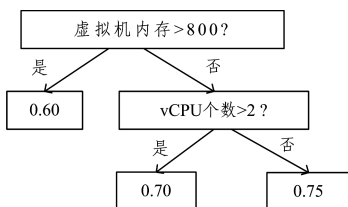


图 1 随机森林中的决策树示例

Fig. 1 Example of regression tree in random forest

图 1 中,该决策树有 2 个内部节点和 3 个叶子节点,每个内部节点对应一个输入特征,即虚拟机内存容量和 vCPU 个数。如果有一个虚拟机配置为 3 个 vCPUs 和 600 MB 内存,则在根节点处,该虚拟机内存 600 MB 小于 800 MB,将其分配到右节点处,然后在右节点处,判断该虚拟机的 vCPUs 个数 3 大于 2,将其分到左孩子节点。因此该棵决策树预测的虚拟机性能值为 0.7,从而得到 nt 棵决策树的预测值,取平均值即为该台虚拟机的性能预测值。

4 虚拟机配置优化

当给定性能需求时,在虚拟机特征取值范围内搜索能满足性能需求的虚拟机配置是一个最优化问题。求解时,可根据特定的性能需求,从虚拟机各特征的取值范围中找到一个或多个能够满足性能需求的组合。在基于随机森林的虚拟机性能预测模型的基础上,利用遗传算法实现问题的快速求解。其基本思路为:随机生成一组虚拟机配置,将其作为初始种群,预测初始种群中每个虚拟机的性能,将其与目标性能比较,挑出最好的一部分虚拟机配置作为父本,进行交叉和变异操作,得到下一代种群。对每一代种群进行上述操作,逐渐向最接近目标性能的方向收敛,在达到终止条件时,得到预测性能值接近目标性能值的虚拟机配置。

4.1 虚拟机配置编码

首先对虚拟机配置进行编码以形成染色体,方便后续的交叉与变异操作。本文将每个虚拟机特征作为染色体中的一个基因,其值为该特征的取值,则虚拟机配置染色体的长度值等于我们选取的影响虚拟机性能的特征个数。因此,虚拟机配置染色体可表示为 $R = \{x_1, x_2, \dots, x_F\}$ 。其中 x_i 为虚拟机的第 i 个特征的取值。表 2 列出了从某一个虚拟化环境中提取出的虚拟机特征及各特征的取值范围。

表 2 各虚拟机特征及其取值范围

Table 2 VM properties and their value ranges

特征	x_1	x_2	x_3	x_4	x_5
含义	vCPU 个数	内存(MB)	CPU 的微架构	运行环境	vCPU-CPU 绑定形式
取值范围	1,2,3,4	400~2000	1,2	0,1	1,0

如表 2 所列, x_3 取值为 1 表示 AMD 的 Streamroller 微架构,取值为 2 表示 Intel Skylake 微架构; x_4 表示虚拟机运行环境,取值为 1 表示物理主机上多虚拟机同时运行,取值为 0 表示单个虚拟机运行; x_5 取值为 1 表示 vCPU 绑定到固定的 CPU 核心,取值为 0 表示未绑定。

根据前述编码方式,一条虚拟机配置染色体的长度为 5,某一个虚拟机配置的染色体形如:

$\{1,400,1,0,1\}$

这条虚拟机配置染色体代表的虚拟机配置为 1 个 vCPU,400MB 内存,vCPU 架构为 Streamroller,物理主机上只有一台虚拟机在运行,vCPU 绑定到 CPU 核心。在遗传算法初始化时,可随机产生一组虚拟机配置染色体作为初始种群。

4.2 初始虚拟机配置种群的生成

在虚拟机配置优化问题的求解中,最优解的概率和可信

度会随着虚拟机配置种群的增大而变大。然而,虚拟机配置种群的增大会显著增加计算时间,因此在实际应用中应依据不同情况来设定虚拟机配置种群的大小。

本文中,根据训练数据集的大小设置初始种群规模 S 的取值范围为 $20 \sim 200$ 。可按照以下方法产生初始虚拟机配置种群:系统按 4.1 节的编码方式随机产生 S 个虚拟机配置染色体。

4.3 适应度函数

在选择种群中的特定个体作为父本进行遗传操作时,为了保证搜索不断向最优方向进行,本文利用随机森林模型预测特定个体的性能值,选出最接近性能需求的个体作为父本进行遗传操作。因此,设计适应度函数公式如下:

$$Fit(\mathbf{X}_i) = |prediction(\mathbf{X}_i - p^{target})| \quad (1)$$

其中, $\mathbf{X}_i = (x_1, x_2, \dots, x_F)$ 表示虚拟机的配置, $prediction(\mathbf{X}_i)$ 表示虚拟机在 \mathbf{X}_i 配置下的随机森林预测的性能值, P^{target} 表示用户需求的目标性能值。在选定初始虚拟机配置种群以后,算法依次对每个染色体进行性能预测,按式(1)计算性能预测值与性能目标值之间的差距,最接近目标性能的预测性能的虚拟机配置适应度函数值最小,其染色体 \mathbf{X}_i 最优。

4.4 遗传操作

从当代种群中选择较好的虚拟机配置进行交叉操作,并以一定的概率进行变异。由于 4.3 节中的适应度函数保证了性能越接近目标性能的虚拟机越优,因此,遗传操作能够向接近目标性能的方向收敛。

本文采用轮盘赌法选中较接近性能需求的虚拟机配置染色体来进行交叉操作,算法的思路如下:

(1) 根据适应度函数(式(1))计算得到种群中每个虚拟机配置的染色体的适应度 $Fit(\mathbf{X}_i)$, $i = 1, 2, \dots, M$, M 为群体大小。

(2) 按式(2)计算每个虚拟机配置的染色体被选中进行交叉操作的概率。

$$P_i = \frac{1 - Fit(\mathbf{X}_i) / P^{target}}{\sum_{i=1}^M 1 - Fit(\mathbf{X}_i) / P^{target}} \quad (2)$$

其中, $Fit(\mathbf{X}_i) / P^{target}$ 度量虚拟机性能预测值与目标值的差距占性能目标值的比例, $1 - Fit(\mathbf{X}_i) / P^{target}$ 即为虚拟机性能预测值与性能目标值的比值,虚拟机配置越接近目标性能值,则该比值越大。因此,式(2)保证了越符合目标性能需求的虚拟机被选中进行交叉操作的概率越大。

(3) 按式(3)计算每个虚拟机配置染色体的累积概率:

$$q_i = \sum_{j=1}^i p_j \quad (3)$$

(4) 在 $[0, 1]$ 区间内产生一个均匀分布的伪随机数 r , 若 $r < q_1$, 则选择虚拟机配置染色体 1, 否则, 选择虚拟机配置染色体 L , 使得 $q_{L-1} < r \leq q_L$ 成立。

(5) 重复执行步骤(4) M 次, 选择概率越大的虚拟机配置染色体被选中的次数越多, 以选出的 Q 条虚拟机配置染色体完成后续的交叉操作。

在交叉操作中, 首先随机选取一对代表虚拟机配置的染色体 \mathbf{X}_a 及 \mathbf{X}_b ; 接着通过单点交叉方式交换配对的虚拟机配置染色体特征, 即随机选取某个交叉点 i , 交换 \mathbf{X}_a 和 \mathbf{X}_b 的

$(x_1, x_2, \dots, x_{i-1})$ 这 $i-1$ 个特征的值, 完成虚拟机配置染色体的交换并生成新的虚拟机配置染色体; 之后, 采用均匀变异算子实现变异操作, 即在某个变量取值的范围内, 用符合该范围的均匀分布的一个随机数, 以某一较小的概率替换虚拟机配置染色体 \mathbf{X}_i 中基因位 x_j 上的原有基因值。

因此, 用遗传算法求解虚拟机配置优化问题的具体执行流程如下:

(1) 初始化算法中所涉及到的参数, 包括虚拟机配置种群的大小、终止迭代的进化代数或适应度值、满足用户需求的目标性能值;

(2) 对虚拟机配置进行编码, 生成初始虚拟机配置种群, 包含 S 条虚拟机配置染色体, 设置用于遗传操作的虚拟机配置染色体为 Q 条;

(3) 利用式(1)计算虚拟机配置种群中虚拟机配置染色体的适应度;

(4) 根据终止条件, 设置最大迭代次数或适应度函数的最优值, 判断是否迭代完成, 如果满足终止条件, 则输出适应度函数值最大的虚拟机配置染色体, 即最接近目标性能的虚拟机配置, 否则进入步骤(5);

(5) 应用轮盘赌选择法对虚拟机配置种群进行选择操作, 选出 Q 条虚拟机配置染色体进行遗传操作;

(6) 对 Q 条虚拟机配置染色体进行遗传操作, 包括单点交叉和均匀变异, 然后将生成的新的虚拟机配置染色体加入到原来的虚拟机配置种群中;

(7) 计算新虚拟机配置染色体的适应度函数值, 删除其中适应度函数值最大的 Q 个虚拟机配置染色体, 回到步骤(4)。

5 实验

5.1 实验环境

主机 1: Intel Core i5-6600, DDR4 2133 16 GB 内存, 西部数据 SATA3 500 G 硬盘;

主机 2: AMD A10-7850K, DDR3 1866 16 GB 内存, 金泰克 SSD 100 G 硬盘;

两台主机均运行 Xen-4.6, 在每一台主机上创建两个虚拟机, 每个虚拟机都安装 CentOS7 系统, 内核版本都为 Linux kernel 3.18.34-20.el7.x86_64。

5.2 训练数据集的获取

从实验环境中选取 10 个对虚拟机性能影响较大的特征, 如表 3 所列。

表 3 虚拟机特征
Table 3 VM properties

特征分类	特征	特征取值与简化表示
硬件特征	CPU 类型	Intel(1), AMD(2)
	硬盘的类型	SATA(1), SSD(2)
软件特征	调度算法	credit(1), credit2(2)
	调度权重	低(1), 中(2), 高(3)
配置特征	vCPU 个数	1, 2, 3, 4
	vCPU-CPU 绑定	不绑定(1), 绑定(2)
	L2Cache 共享	不共享(0), 共享(1)
	虚拟机内存容量(MB)	400~2000
环境特征	同时运行的虚拟机	无(0), 有(1)
	其他虚拟机上运行的负载类型	无负载(1), I/O 密集型应用(2), CPU 密集型应用(3)

选取表 3 中 10 个影响虚拟机性能的特征,在每个特征的值域范围内对该特征随机选取一个值,组合构成一台虚拟机的配置。如此多次取值,共配置 400 个不同的虚拟机。在基准测试程序套件 PARSEC 中选取 facesim, x264, bodytrack, ferret, streamcluster 5 个程序模拟不同的应用,其包含了 CPU 密集型应用、I/O 密集型应用及混合型应用。实验中,以基准测试程序的实际运行时间来表示虚拟机的性能,由于选取的以上 5 个程序在计算、访存、I/O 等方面都各有侧重,因此,它们的实际运行时间能够表达虚拟机在计算、访存、I/O 等方面的综合性能。将这 5 个程序分别运行在 400 台不同的虚拟机上,记录其特征取值及性能值,并将这些值作为构造随机森林模型的训练数据集。

5.3 预测的准确性评估

为了评估随机森林模型预测虚拟机性能的准确性,本文将随机森林模型的预测结果与多层感知器(Multilayer Per-

ceptron, MLP)的预测结果进行比较。MLP 是一种前馈人工神经网络,由多层节点组成。实验中采用了单个隐藏层的 MLP 模型,输入层的节点表示虚拟机的各个特征,输出层的节点表示虚拟机性能指标不同的可能取值,其值表示虚拟机性能取该值的概率。

使用 5.2 节中在 400 台虚拟机运行测试程序时所获取的训练数据集分别训练随机森林模型和 MLP 模型,得到用于虚拟机性能预测的两种不同的模型。

通过为表 3 中虚拟机各特征随机赋值,配置 16 台不同的虚拟机,分别表示为 v_1, v_2, \dots, v_{16} ,分别用随机森林模型和 MLP 模型对 16 台虚拟机进行性能预测。同时,在每一台虚拟机上分别运行 5 次基准测试程序,取 5 次运行结果的平均值作为实测值。同时,将 16 种配置分别输入随机森林模型和 MLP 模型,得到两种模型预测的虚拟机性能。图 2 给出了两种模型的预测值与实测值的比较。

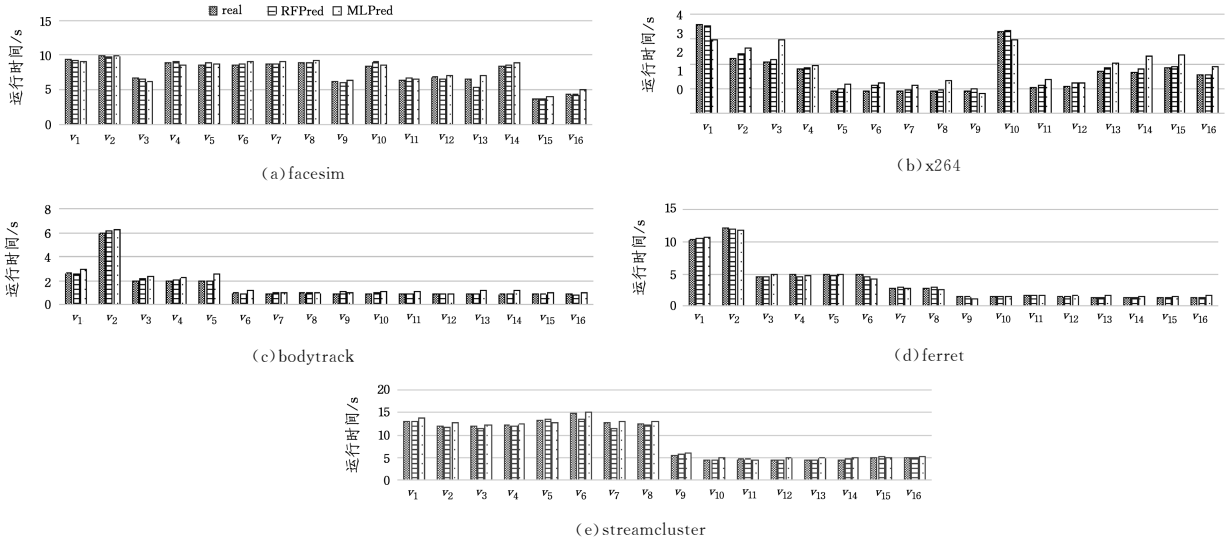


图 2 随机森林、MLP 的预测结果与实测值的比较

Fig. 2 Experimental results of random forest and MLP results

MLP 模型的输出表示虚拟机性能指标为不同取值的概率,因此按式(4)计算虚拟机的性能预测值 Y :

$$Y = \sum_{i=1}^k y_i * p_i \tag{4}$$

其中, k 是虚拟机性能可能取值的个数, y_i 表示虚拟机性能指标可能的取值, p_i 表示性能为该取值的概率。

从图 2 可以看出,随机森林回归模型较 MLP 模型预测的虚拟机性能值更加接近虚拟机性能实测值。同时也可以看出,运行 5 个不同的基准测试程序时,同一个虚拟机呈现出不同的性能。这也说明了不同的基准测试程序在计算、访存、I/O 等方面有不同的需求。因此,可以通过搜索合理的虚拟机配置,为不同的应用程序选取能达到其性能要求的资源配置,保证主机资源的合理分配。

计算 2 种方法预测得到的性能值与实测值之间的误差,本节使用 3 个指标来评估误差大小,分别是 MAE(平均绝对误差)、RMSE(均方根误差)、MAPE(平均绝对百分误差)。表 4 列出了 5 个基准程序的性能预测结果的误差。

表 4 列出了 5 个基准程序的性能预测结果的误差。

表 4 误差评价

Table 4 Errors evaluation

模型	facesim			x264			bodytrack			ferret			streamcluster		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Random forests	0.21	0.36	0.03	0.09	0.11	0.07	0.06	0.10	0.05	0.12	0.19	0.03	0.30	0.49	0.03
MLP	0.33	0.39	0.05	0.38	0.43	0.25	0.22	0.27	0.16	0.22	0.29	0.07	0.37	0.42	0.05

从表 4 可以看出,对于 5 个基准测试程序,用随机森林模型预测得到的性能结果中,3 个误差评价指标都较小,这说明

在当前数据集规模下,随机森林性能预测模型相较于 MLP 模型具有较小的预测误差。

5.4 虚拟机配置搜索的有效性收敛性测试

为了验证满足用户性能需求的虚拟机配置优化方法的有效性,我们设定遗传算法的初始条件:初始虚拟机配置种群 S (虚拟机配置染色体数目)为 200,交叉概率默认为 1.0,变异概率为 0.01。算法终止条件为:达到最大迭代次数(设为

20)。通过实验验证求得的解是否能达到用户的性能需求。

实验采用基准测试程序 x264 和 facesim。在搜索得到符合性能需求的虚拟机配置之后,在实验平台上按搜索得出的虚拟机配置设置虚拟机,测试其实际性能,比较性能需求及实测性能,测试结果分别如表 5 和表 6 所列。

表 5 虚拟机配置搜索有效性测试(x264)

Table 5 Efficiency of VM configuration search (x264)

配置编号	目标性能值 P_{target}	搜索得到的虚拟机配置										性能预测值	性能实测值
		CPU 微架构	内存容量 /MB	CPU 主频	vCPU 个数	调度算法	核心绑定	多虚拟机	负载类型	调度权重	L2 Cache 是否共享		
X_1	2.30	1	450	3.3	3	2	2	1	3	3	0	2.36	2.38
X_2	1.00	1	698	3.3	2	2	2	1	2	1	0	1.01	0.97
X_3	0.90	1	1193	3.3	2	1	1	1	1	3	0	0.91	0.90
X_4	0.95	2	530	3.6	2	2	1	0	1	1	1	0.99	0.95
X_5	0.68	2	1531	3.6	4	1	1	0	2	2	1	0.68	0.61

表 6 虚拟机配置搜索有效性测试(facesim)

Table 6 Efficiency of VM configuration search (facesim)

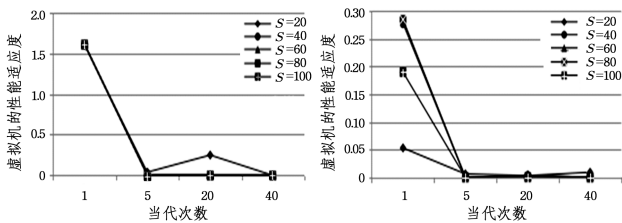
配置编号	目标性能值 P_{target}	搜索得到的虚拟机配置										性能预测值	性能实测值
		CPU 微架构	内存容量 /MB	CPU 主频	vCPU 个数	调度算法	核心绑定	多虚拟机	负载类型	调度权重	L2 Cache 是否共享		
X_6	25	1	522	3.3	3	2	2	1	2	3	0	25.17	25.10
X_7	76	1	427	3.3	4	2	2	1	1	1	0	76.54	75.70
X_8	10	1	720	3.3	3	1	2	1	2	2	0	10.00	9.40
X_9	15	2	555	3.6	1	1	2	0	1	2	1	14.97	15.57
X_{10}	7	2	609	3.6	2	1	2	0	1	1	1	7.05	8.11

由表 5 和表 6 可以看出,当我们给定一个性能需求,使用遗传算法能得到一个较接近目标性能值的虚拟机配置,搜索得到的虚拟机配置用基准测试程序测试得到的实际性能值与目标性能值也非常接近,这说明本文提出的方法是有效的,用遗传算法可以搜索得到较为接近目标性能值的虚拟机配置。

响不明显,适应度函数的值越来越小,表明搜索得到的虚拟机配置的适应度越来越高,其虚拟机性能更接近目标性能值,且可以看出本文算法的收敛速度很快,到第 5 代时就基本收敛。这表明使用提出的方法搜索符合性能需求的虚拟机配置是高效的。

为验证不同参数下遗传算法的收敛速度,测试不同虚拟机配置种群规模对虚拟机配置搜索问题中适应度的影响。在实验中使用 x264 基准测试程序的样本数据集,遗传算法中适应度函数中的 P_{target} 分别设为 2.6 和 1,交叉概率默认为 1.0,变异概率为 0.01,初始虚拟机配置种群大小 $S=20,40,60,80,100$ 。实验结果如图 3 所示。

接下来通过改变遗传算法的变异概率来观察其对遗传算法收敛到全局最优解的影响。图 4 给出了适应度函数中的 $P_{target}=1$,初始虚拟机配置种群大小 $S=20$,交叉概率为 1.0,变异概率为 0.01,0.1,0.5 和 0.7 时,遗传算法收敛速度的变化。



(a) $P_{target}=2.6$ 时虚拟机的性能适应度 (b) $P_{target}=1$ 时虚拟机的性能适应度

图 3 虚拟机性能适应度

Fig. 3 Fitness of VM performance

从图 3 可以看出,整体上,由于遗传算法是一个逐步得到最优解的求解过程,对于虚拟机配置搜索问题,适应度函数的值越小越好,适应度函数的值越小说明该虚拟机配置的适应度越高;就不同的虚拟机配置种群规模来说,在遗传操作初期,虚拟机配置种群规模的大小会对适应度有一定的影响,之后随着遗传代数的增加,虚拟机配置种群规模对适应度的影

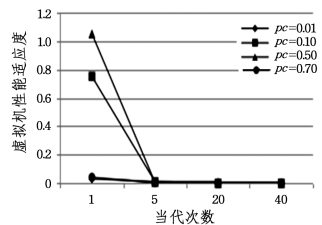


图 4 变异概率与虚拟机性能适应度

Fig. 4 Mutation probabilities and fitness of VMs

从图 4 可以看出,变异概率的大小对算法的收敛速度也有一定的影响。对于本文中满足性能需求的虚拟机配置优化问题,只需要设置较小的变异概率,就可能得到较快的收敛速度,当变异概率为 0.01 时,遗传算法的适应度最佳,收敛速度最快,这说明遗传算法处理满足性能需求的虚拟机配置优化问题是快速且有效的,该方法可以用于虚拟机资源动态配置等云计算平台的资源管理决策。

结束语 本文针对 IaaS 平台目前的服务模式,提出基于随机森林回归模型预测虚拟机的性能,支持根据任意虚拟机

的各项特征取值预测虚拟机运行特定应用程序时呈现的性能,并在此基础上,提出利用遗传算法在虚拟机特征空间中搜索最符合用户性能需求的虚拟机配置,利用随机森林模型的性能预测结果评估每代种群中各个体的适应度,以保证搜索过程向接近用户性能需求的方向收敛。实验结果表明,随机森林性能预测模型能较为准确地预测虚拟机的性能,其预测结果接近实际测试结果。根据性能需求搜索得出的虚拟机配置在实测时能达到目标性能。本文提出的算法能应用于 IaaS 云计算服务中,用于指导用户选择租用恰当的虚拟机,也能指导资源提供方合理分配资源。

在未来的工作中,我们将继续探讨多虚拟机同时运行时的配置优化,即当多个虚拟机同时有性能需求时,在现有的资源约束条件下如何实现资源的最优分配。进一步,将本文方法应用于云计算环境中资源的动态调配,在用户虚拟机上出现突发的性能需求时,及时决策新的资源需求,动态调整资源分配,进一步提高资源利用率。

参 考 文 献

- [1] KUNDU S,RANGASWAMI R,DUTTA K. Application performance modeling in a virtualized environment[C]//16th International Symposium on High Performance Computer Architecture (HPCA), 2010.
- [2] BROSIG F, GORSLER F, HUBER N. Evaluating Approaches for Performance Prediction in Virtualized Environments[C]// Proceedings of 21st International symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2013.
- [3] KRAFT S,CASALE G,KRISHNAMURTHY D. I/O performance prediction in consolidated virtualized environments[J]. Acm Sigmetrics Performance Evaluation Review, 2011, 39(3): 17-18.
- [4] MENG F, DU G, HE H. Performance Modeling on the Basis of Application Type in Virtualized Environments[J]. Journal of Software, 2013, 8(11): 2847-2854.
- [5] LI F Z, YANG D, ZHOU P, et al. Modeling Application Performance in a Virtualized Environment[J]. Computer Systems & Applications, 2015, 24(9): 9-15. (in Chinese)
黎丰泽, 杨达, 周鹏, 等. 虚拟环境下虚拟机应用性能建模[J]. 计算机系统应用, 2015, 24(9): 9-15.
- [6] 贝振东, 喻之斌, 熊文, 等. 一种云计算系统中虚拟机的性能预测方法及系统: 中国, CN104536829A[P]. 2018-04-22.
- [7] CHEIKH B, DONCEL J, BRUN O, et al. Predicting Response Times of Applications in Virtualized Environments[J]. International Journal of Humanoid Robotics, 2016, 12(3): 83-90.
- [8] XU J, FORTES J A. Multi-objective virtual machine placement in virtualized data center environments[C]// IEEE/ACM International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing, 2010.
- [9] WU G, TANG M, TIAN Y C, et al. Energy-efficient virtual machine placement in data centers by genetic algorithm[C]// Neural Information Processing. Springer, 2012: 315-323.
- [10] TANG M, PAN S. A Hybrid Genetic Algorithm for the Energy-Efficient Virtual Machine Placement Problem in Data Centers [J]. Neural Processing Letters, 2015, 41(2): 211-221.
- [11] YANG T, LEE Y C, ZOMAYA A Y. Energy-Efficient Data Center Networks Planning with Virtual Machine Placement and Traffic Configuration[J]. IEEE 6th International Conference on Cloud Computing Technology and Science, 2014.
- [12] KAAOUACHE M A, BOUAMAMA S. Solving bin Packing Problem with a Hybrid Genetic Algorithm for VM Placement in Cloud[J]. Procedia Computer Science, 2015, 60(1): 1061-1069.
- [13] JIANG B, LI R, et al. An improved genetic algorithm for load balance in multiprocessor systems[C]// Proceedings of the 14th International Conference on Advanced Communication Technology, 2012.
- [14] WANG T, LIU Z, CHEN Y, et al. Load balancing task scheduling based on genetic algorithm in cloud computing[C]// Proceedings of 12th International Conference on Dependable, Automatic and Secure Computing, 2014.
- [15] SUNDARARAJAN P K, FELLERY E, FORGEATY J, et al. A Constrained Genetic Algorithm for Rebalancing of Services in Cloud Data Centers[C]// Proceedings of the 8th International Conference on Cloud Computing, 2015.
- [16] MI H B, WANG H M, YIN G, et al. Resource On-Demand Reconfiguration Method for Virtualized Data Centers[J]. Journal of Software, 2011, 22(9): 2193-2205. (in Chinese)
米海波, 王怀民, 尹刚, 等. 一种面向虚拟化数字中心资源按需重配置方法[J]. 软件学报, 2011, 22(9): 2193-2205.