# 2018 华为软件精英挑战赛初赛赛题介绍

# （智能预测与资源调度）

## 1、 比赛问题定义

背景：云平台为了满足不同租户的需求，提供了一种可随时自助获取、可弹性伸缩的云服务器，即弹性云服务器（Elastic Cloud Server，ECS）。为容纳更多的租户请求、并尽可能提高资源利用率、降低成本，自动化、智能化的资源调度管理系统非常关键。

**本次赛题基本描述**

由于租户对ECS实例（虚拟机，VM）请求的行为具有一定规律，可以通过对历史ECS实例请求的分析，预测到未来一段时间的ECS实例请求，然后对预测的请求分配资源（如图1所示），这样可以找到一个接近最优的分配策略，实现资源最大化利用，同时也能参考预测的结果制定云数据中心的建设计划。

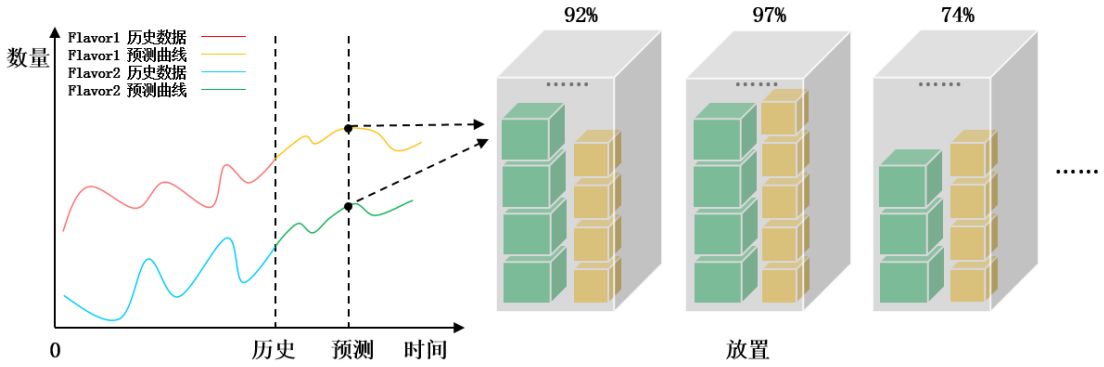


图1 智能预测与资源调度示例

**本次赛题通用性描述**

**物理服务器：**云数据中心通常由大规模的物理服务器集群所组成，通过虚拟化技术，可以对每个物理节点的资源（如CPU、内存、硬盘等）进行隔离，使得每台物理机上可以同时容纳多个虚拟机，这些虚拟机即共享该物理服务器上的所有资源。为了保证每台虚拟机的运行性能，通常物理资源不能“超分”，即每台物理服务器上所有虚拟机的虚拟资源总量不能超过其物理资源总量。这里假设物理服务器只有一种规格大小。

**虚拟机规格：**云平台通常预先定义了各种类型虚拟机的规格（flavor），便于租户选择

购买。例如，如果租户对计算要求低，而内存要求相对高，可以选择4个vCPU、16GB内存的配置。由于不考虑物理资源的“超分”，虚拟机的资源占用可与物理资源进行一比一的映射。当然，不同规格的虚拟机有不同的性能，价格也不一。

**资源维度：**如上述，虚拟机正常工作需要多个维度的资源同时配合，如CPU、内存、硬盘、网络等等，每种资源都可能成为瓶颈，并且每种资源分配不合理都可能产生碎片。这里假设只需要考虑单个维度资源的使用，即尽可能最大化每台物理服务器的CPU或者内存（Mem）的利用率，但在考虑某个维度资源优化的同时，要保证其他资源不能超分。

**历史请求数据：**租户在云平台上每申请一台虚拟机都会在后台的数据库产生一条数据，每条数据包含了虚拟机的ID、虚拟机的规格大小以及创建时间等。如果我们可以获取到云平台在过去一段时间的所有虚拟机请求数据，通过训练这些数据特征，可以预测下一个时间段可能到来的虚拟机请求分布。

**预测时间段：**需要预测的开始时间、结束时间，如开始时间为2017-01-09 00:00:00、结束时间为2017-01-16 00:00:00，那么需要预测在该时间段内每种规格虚拟机的请求数量。

**比赛程序内容：**请你设计一个程序能够精确预测未来某个时间段内的虚拟机的请求情况，并寻找最佳的资源分配方案：对输入的虚拟机历史请求数据进行建模分析以及训练，确定系数给出预测模型，然后对给定预测时间段内不同规格的虚拟机数量进行预测，最后根据预测结果把虚拟机部署到物理服务器上，使得服务器的资源利用率最大化。

**比赛胜负规则**

比较参赛者程序输出的预测结果的精度以及部署的资源利用率的乘积，得分较大者胜出。如果出现得分相同的情况，则比较程序运行时间，时间短者胜出。若运行时间也相同，则根据提交时间先后来区分排名。如输出结果不满足约束条件，得分为零。

（备注：资源利用率只需要考虑单个维度，如CPU或内存的利用率）

**补充说明：**

1. 物理服务器资源认为全部可用，不需要考虑系统占用、资源预留等问题。如一台物理服务器具有56个CPU，即认为这56个CPU可以全部分配给虚拟机使用；
2. 假设物理服务器数量充足，如果当前集群无法放置新的虚拟机，则可以开启新的物理服务器进行放置；
3. 无论预测结果好坏，预测出来的虚拟机需要全部放置，否则得分为零；

4. 不需要考虑虚拟机动态迁移的情况，即会根据虚拟机放置的静态结果进行资源利用率的评分；
5. 不考虑任何资源类型的超分情况，如果虚拟机资源总量超出物理服务器总资源，则得分为零；
6. 只考虑优化单个维度资源利用率的情况，不需要考虑其他资源的碎片率，但要保证其他资源不能超分，否则得分为零；
7. 不需要考虑虚拟机的生命周期，即不需考虑虚拟机被删除的情况；
8. 由于云平台通常使用网络共享存储，任何时候只需要考虑CPU和内存两种资源；
9. 只需要对测试用例输入文件给出的虚拟机规格进行预测并放置，未给出的虚拟机规格但在历史请求数据存在的可不作考虑；初赛要求预测的虚拟机规格数量最多为15种，分别为以下类型：

   *flavor1 1 1024*

   *flavor2 1 2048*

   *flavor3 1 4096*

   *flavor4 2 2048*

   *flavor5 2 4096*

   *flavor6 2 8192*

   *flavor7 4 4096*

   *flavor8 4 8192*

   *flavor9 4 16384*

   *flavor10 8 8192*

   *flavor11 8 16384*

   *flavor12 8 32768*

   *flavor13 16 16384*

   *flavor14 16 32768*

   *flavor15 16 65536*

   备注：flavor名称 CPU核数 内存大小（MB）
10. 需要预测的时间跨度为1~2个星期，且要预测的开始时间紧接着训练数据集的结束时间；
11. 节假日、双十一等特殊日期的历史数据，通常认为是异常点，这个需要参赛者自

已做去噪处理，这也是一个考查的点，测试用例的训练数据集可能会出现异常点，但是测试数据集不会出现异常点；

12. 程序实现不能使用第三方库，编译环境也不支持第三方库，如有发现违规情况，直接取消比赛成绩。

## 2、 程序输入与输出

**输入文件格式**

程序输入为一个以空格分隔的文本文件，文件每行以换行符（ASCII’\n’即0x0a）为结尾。

**文件格式为：**

物理服务器CPU核数 内存大小（GB） 硬盘大小（GB）

（空行）

虚拟机规格数量

虚拟机规格名称1 CPU核数 内存大小（MB）

虚拟机规格名称2 CPU核数 内存大小（MB）

……（如上虚拟机规格信息若干行）

（空行）

需要优化的资源维度名称（CPU或内存）

（空行）

预测开始时间

预测结束时间（时间跨度单位为：天）

（文件结束）

示例：

56 128 1200

（备注：物理服务器资源信息包含硬盘大小，实际调度可不考虑硬盘容量。CPU核数为不超过两位的整数，内存大小为不超过3位的整数，硬盘大小为不超过4位的整数。1GB=1024MB）

flavor5　2　4096

flavor10　8　8192

flavor2　1　2048

（备注：虚拟机规格通常只包含CPU和内存两种信息，硬盘通常为可扩展存储，暂不做考虑。虚拟机规格名称最大不超过10位字符，CPU字段最大不超过2位字符，内存字段最大不超过6位字符。注意：测试用例输入的虚拟机规格通常只是历史数据的一部分，不是全部，参赛者只需要对输入的虚拟机规格进行预测即可，其他虚拟机规格无需考虑。）

CPU

（备注：若是MEM，则表示内存）

2017-01-09 00:00:00

2017-01-16 00:00:00 //注：即7天的时间跨度

（备注：日期与具体时间之间为空格符。需要预测的时间跨度为1~2个星期，且要预测的开始时间紧接着训练数据集的结束时间。）

（文件结束）

**【训练数据集说明】**

用户历史请求数据为一个以制表’\t’分隔的文本文件，文件每行以换行符（ASCII’\n’即0x0a）为结尾。

**文件格式为：**

虚拟机ID　虚拟机规格名称　创建时间（年-月-日 时:分:秒）

示例：

23d7ac3a-1134-46d7-ac88-c3015e11db02　flavor8　2016-09-07 07:45:36

（备注：时间中的日期与具体时间之间为空格符；虚拟机ID字段最大长度不超过36位字符，虚拟机规格名称最大不超过10位字符；另外，数据集的时间跨度会以整天为单位，参赛者只需要检测到开始时间的当天以及结束时间的当天即可。训练数据中每一条已经按照时间先后顺序排序。每个测试用例的训练数据集最大不超过10000条。）

**输出文件格式**

程序输出为一个以空格分隔的文本文件，文件每行以换行符（ASCII’\n’即0x0a）为结尾。

**文件格式为：**

预测的虚拟机总数

虚拟机规格名称1　虚拟机个数

虚拟机规格名称2　虚拟机个数

……（如上预测的不同规格虚拟机名称及数量若干行）

（空行）

所需物理服务器总数

物理服务器1　虚拟机规格名称1　能放置该类型虚拟机个数　虚拟机规格名称2　能放置该类型虚拟机个数 ……

物理服务器2　虚拟机规格名称1　能放置该类型虚拟机个数　虚拟机规格名称2　能放置该类型虚拟机个数 ……

……（如上每台物理服务器的所能放置每种规格虚拟机数量若干行）

（文件结束）

（备注：物理服务器 ID 可用阿拉伯数字 1、2、3、……表示即可）

示例：

6

flavor5　3

flavor10　2

flavor15　1

（备注：如果某种虚拟机规格的预测结果为零，即对应写 0）

4

1　flavor5　2

2　flavor5　1　flavor10　1

3　flavor15　1

4　flavor10　1

（备注：每个字段之间用空格隔开即可。）

## 3、　单个用例的评分机制

**用例的评分机制**

按下面流程对参赛者结果进行评分：

Step1：对递交代码进行编译；

Step2：运行用例，程序运行时间不得超过60s；

Step3：对用例运行的输出结果进行合法性检验（详见题目描述）；

上述步骤有任何失败异常，则本用例得分为0；

Step4：计算预测精度与资源利用率的乘积，结果越大，得分越高。

**单个用例的评分公式如下：**

$$\mathbf{Score} = (\mathbf{1} - \frac{\sqrt{\frac{1}{|N|}\sum_{i=1}^{|N|}(y_i - y_i')^2}}{\sqrt{\frac{1}{|N|}\sum_{i=1}^{|N|} y_i{}^2} + \sqrt{\frac{1}{|N|}\sum_{i=1}^{|N|} y_i'^2}}) \times \frac{\sum_{v \in V} \boldsymbol{r_v}}{\sum_{h \in H} \boldsymbol{R_h}} \times \mathbf{100}$$

$\mathbf{y}_i$：表示第$i$种虚拟机规格的实际数量；

$\mathbf{y}_i'$：表示预测出来的第$i$种虚拟机规格数量；

$\boldsymbol{N}$：表示虚拟机规格的集合；

$\boldsymbol{r_v}$：表示预测到的第$v$个虚拟机所要预测的资源大小；

$\boldsymbol{R_h}$：表示所放置的第$h$个物理主机相应的资源容量；

$\boldsymbol{V}$：表示预测出来的虚拟机集合；

$\boldsymbol{H}$：表示所需要的物理主机集合。

（备注：预测部分精度计算使用了希尔不等式 TIC，希尔系数是评价预测模型常用的评价指标，它总是介于 0 至 1 之间，数值越小表明拟合值和真实值之间的差异越小，预测精度越高。当等于 0 时，表示 100%拟合。单个用例满分为 100 分。）

## 4、 最终得分和排名机制

比赛平台会使用多个训练数据集，每个训练数据集又构造出多个测试用例判题，测试用例分为初级、中级、高级三个等级，每个等级的难度主要根据预测的时间长短以及预测的虚拟机规格数量两个指标来区分。每个等级各3个用例，共9个用例。参赛者对于每个测试用例都会得到一个百分制分数，使用加权总分（初级权重为0.2，中级权重为0.3，高级权重为0.5）作为该参赛者的最终得分。

最终排名机制如下：

Step1：最终得分为所有测试用例得分的加权，根据得分高低进行排名；

Step2：最终得分相同的情况，比较所有测试用例的总运行时间，运行时间越短，排名越靠前；

Step3： 如以上均相同的情况，则根据提交的先后顺序区分排名。

特别说明：在比赛初期（练习阶段），比赛平台只放出初级、中级的测试用例各2个，故加权后满分为100分（初级权重为0.2，中级权重为0.3，高级权重为0.5；下同）；在正式比赛阶段，才会放出高级测试用例（具体发放时间会在网站公告通知），初、中、高级用例各3个，加权后满分为300分。练习阶段每支参赛队伍每天最多只能提交100次，正式比赛阶段每支参赛队伍每天最多只能提交5次。

特别注意：最终成绩以**最后一次**提交的答案为准。请各位参赛者务必注意。

## 5、 运行环境

开发语言支持：C/C++、Java 7/8、Python 2.7

CPU：Intel(R) Xeon(R) CPU E5-2680 V4 @ 2.40GHz

内存：2G

CPU核数：单核

编译器：gcc 4.8.4、java 1.8、python 2.7

操作系统：Ubuntu 14.04.4 LTS 64位，内核版本 Linux version 4.4.0-31-generic

SDK：为方便选手做题，分别提供c++(兼容c）、Java、Python的SDK包供参考（见DevCloud项目），详细描述信息请见SDK目录下的readme.txt。