

	状态	主办方	第 2 赛季截止日期	总奖池	参赛队
阿里巴巴全球调度算法大赛	进行中	<div><div>Alibaba Group</div><div>蚂蚁软件事业部</div><div>Ant Group Software Department</div></div>	2018/09/07	¥ 200000	2115

赛制介绍

赛题与数据

FAQ (/faq.htm?)

排行榜 (/rankingList.htm?)

论坛

文件名称 (报名后可下载)	文件格式
AlibabaSchedulerEvaluatorRun_20180709.java	.java (14KB)
scheduling_preliminary_dataA_20180606.zip	.zip (4MB)
scheduling_preliminary_dataB_20180726.zip	.zip (2MB)
scheduling_semifinal_data_20180815.zip	.zip (4MB)



阿里巴巴全球调度算法大赛

在百万级的宿主机规模环境下，资源调度/管理系统显得愈发重要，如何在保证数据中心业务运行性能的同时，提高资源利用率，降低：施的成本是我们要考虑的核心问题。

此次赛题来自阿里巴巴生产环境中的一个子场景，并做了相应的简化。优秀的解决方案会对我们解决这个场景以及其它场景下的问题帮的启发。我们期待优秀的你和你的团队能够投入进来！

注意：目前的赛题数据主要为了让大家了解复赛的情况。根据我们进一步测试和反馈得到的情况，数据有可能修改。数据会在评测发布（8月20日）最终确定。

在初赛的基础上，增加了以下内容：

在线任务的迁移限制

迁移将更遵循实际，采取多轮并发执行，每一轮并发迁移为多个迁移命令同时执行，一个迁移命令将采取先新建后删除的方式。例如：

```
1,inst1,A
1,inst2,A
2,inst3,B
```

为两阶段迁移计划。在第一轮中，inst1和inst2迁移到了机器A上，这个是分两个阶段执行的：第一阶段，将同时在机器A上新建inst1、例，新建成功后再将inst1、inst2在原来所属的机器删除旧实例，并执行下一轮（inst3到机器B）。

若出现迁移失败，则直接退出在线任务调度阶段。

迁移限制：每轮迁移没有命令数量限制，但限制轮数小于等于k（暂定k=3）。

离线任务调度

初赛仅涉及在线任务的调度，实际场景我们常常需要在离线任务混部才能获得最高的资源利用率。

定义一个离线任务(job_info.x.csv)为：

离线任务id，cpu占用，mem占用，实例数，执行时间，前驱任务id列表

其中

执行时间的单位为分钟。对应到在线任务的cpu/mem曲线的98个点，分别代表该在线任务在[0, 15)、[15, 30)、...、[1455, 1470)的资源用量

每个离线任务的最早启动时刻，必须大于其所有前驱任务的所有实例执行完成的时刻t

在一个完整周期[0, 1470)内，所有离线任务的所有实例必须全部被执行完成离线任务不允许迁移

对于一个离线任务，只有当其所有的instance都执行完毕，这个离线任务才可以被认为是执行完成

所以最终的提交结果分为两个部分：

在线任务阶段：迁移轮数，在线任务实例id，目标宿主机id

离线任务阶段：离线任务id，目标宿id，启动时刻，启动实例数量

一旦开始离线任务的调度，便不允许继续进行在线任务的调度。

评分修改

原评分公式中的a，由a=10，修改为

$a = (1 + \text{该机器上部署的在线任务inst数量})$

赛题数据说明

复赛有a, b, c, d, e一共5份赛题，每份赛题包含5份数据，其中：

app_interference.csv、app_resources.csv为5份赛题的公用数据

instance_deploy.x.csv、job_info.x.csv、machine_resources.x.csv为第x份赛题对应的独有数据

最终提交结果按a,b,c,d,e的顺序拼接，以#分割。

下面是一个例子：

```
1, inst1, machine_a
task_1, machine_a, 10, 31
#
1, inst1, machine_b
2, inst2, machine_c
task_1, machine_c, 10, 28
```

复赛排行榜中，选手的分数为五份赛题答案得分的总和。

为复赛胜出的准备

参加决赛资格由复赛成绩决定，复赛成绩的主要依据是排名，但也包括赛事导师对排名靠前选手的线下代码审核。为了让选手更好的准之后的提交，我们有如下建议：

请按照我们初赛描述中推荐的方式整合您的代码 (link)

请准备一个简短报告，至少包括

您对赛题的理解和解题思路

最后得出方案的计算环境的配置和计算时间（硬件、软件、时间等）

任何您想说的话

复赛结束时（2018.09.07），我们会按照选手提交的复赛成绩，结合初赛时候的一些表现，在复赛结束时要求指定的选手提交代码报告。如果不在规定的时间提交，视作放弃，但我们真诚的希望出色的选手可以和我们分享您的成果。如果有选手没有被邀请提交代告，我们也欢迎您毛遂自荐，我们的导师会看您的代码和报告，并给出答复意见。

初赛赛题描述

重要更新：

1. 7月26日，我们对初赛赛制进行了更新，具体的：

* 我们新开放了新一代版本的数据（Data_B），之前的数据称之为Data_A，作用于初赛

* Data_B和Data_A描述的是两个独立的问题。Data_B和Data_A的格式一致，只有数值不同，主要目的是为了防止参赛选手设计针对Data_合的算法（经我们测试，通用算法无需修改就可以算出合法的解，但Data_B有更多优化空间）

* 由于添加Data_B导致的提交结果的变化，见下面的“提交结果”部分。简单的说，新的提交方式中，参赛选手还是提交一份答案，其格式<Data_A的解>

#

<Data_B的解>

* 评分机制会有相应修改，并在7月30日作用于排行榜。修改后的评分公式为： $\text{final_score} = 0.5 * (\text{score}(\text{Data_A}) + \text{score}(\text{Data_B}))$ ；针对C和Data_B的评分机制score函数，与更新前保持一致，具体见“执行与评分（初赛）”部分；进入复赛的条件保持不变（见“赛制介绍”）

* 对于此次更新给各位选手带来的不便我们感到十分抱歉，但会尽力为选手提供最好的参赛体验。更多关于赛题更新的说明可以参考：

https://tianchi.aliyun.com/forum/new_articleDetail.html?spm=5176.8366600.0.0.52f3311ft3qWk6&raceld=231663&postsId=5802
(https://tianchi.aliyun.com/forum/new_articleDetail.html?spm=5176.8366600.0.0.52f3311ft3qWk6&raceld=231663&postsId=5802;

2. 赛题在6月11日有一次更新（赛题描述和数据），6月7日上线的赛题已经彻底下线，请所有参赛同学以您当前看到的版本赛题为准，7月初评测工作将以目前版本为准。由此给各位选手带来的不便，深感歉意。但我们相信，目前这个版本的赛题无论对于初学者还是资深的研究者，有趣的。

1 赛题描述

7月26日添加了一份新的数据（Data_B xxxx），下面的赛题描述同时适用于之前版本的数据（Data_A xxxx）和更新的数据（Data_B xxxx）二者共同作用的评分机制见上面的“重要更新”部分。注意，Data_A和Data_B描述的是两个独立的问题，只是格式一样。

共约6K台宿主机（machine），包含了若干种型号，约68K个任务实例（instance），其中一部分已经部署在宿主机上，还有一部分部署。

要求：设计调度算法，在满足要求约束的前提下，通过将全部未被调度的任务实例调度到宿主机上以及腾挪部分已经部署的实例的方式到最优的部署方案。最优部署方案指实际使用宿主机数目尽可能少，且宿主机负荷不能过高。请参考“执行与评分（初赛）”部分来初关于最优部署方案的说明。

在解释数据格式、约束条件之前，为防止概念混淆，先给出几个定义，全文出现的概念以此定义为准。

实例（instance）：一个实例是可以被调度到一个机器上的对象，在实际生产中，一个实例可以是一个docker容器

应用分组（App）：一个应用分组包括很多实例（instance）。属于同一个App下的所有实例，具备相同的约束条件。一个实例能且只一个应用分组

机器（Machine）：机器是集群中的服务器，一个实例被可以调度到一个机器上

1.1 约束描述

任务实例到宿主机的分配，需要满足下列约束：

- 每个实例都标明了CPU、memory、disk此3个维度的资源需求，其中CPU、memory以分时占用曲线的形式给出，在任意时刻，在个宿主机A上，所有部署在宿主机A上的实例的任意资源都不能超过宿主机A的该资源容量
- 另外还有P、M、PM三类资源，定义了应用实例的重要程度，任意一台宿主机上的部署数目不能超过该类型宿主机能够容纳的重要数目上限
- 混部集群时刻处于复杂的干扰环境中，所以我们需要满足一些规避干扰约束，一条规避干扰约束被描述为<APP_A, APP_B, k>，代台宿主机上存在APP_A类型的实例，则最多能部署k个APP_B类型的实例。注意，k可能为0。APP_A和APP_B也可能代表同一个APP <APP_A, APP_A, k>），代表同一台机器上最多可以部署的该APP的实例的数目

1.2 数据描述

问题一共包含四份数据表：instance_deploy.csv, app_resources.csv, machine_resources.csv, app_interference.csv

- instance_deploy.csv
 - 实例id
 - 实例所属应用
 - 实例所属宿主机:
 - 注：当前未分配的实例，实例所属宿主机列为空
- app_resources.csv
 - 应用id
 - cpu分时占用曲线（每个点由<|>隔开）
 - mem分时占用曲线（每个点由<|>隔开）
 - disk申请量（标量）
 - P
 - M
 - PM
- machine_resources.csv
 - 宿主机id

- cpu规格
- mem规格
- disk规格
- P上限
- M上限
- PM上限
- app_interference.csv
 - 应用id1
 - 应用id2
 - 最大部署量

1.3 结果提交

1.3.1 提交格式（初赛）

由于7月26日更新了Data_B（具体见本文顶部的“重要更新”），新的提交格式为：

```
<Data_A的解>
#
<Data_B的解>
```

注意把Data_A的解放在前面。

单份数据的解（i.e. Data_A的解或者Data_B的解）的具体格式，以及单份数据的评分方式见下面的描述。具体的例子可以参考我们的submit_sample.csv

1.3.2 单份数据的解的格式（初赛）

单份数据的解（i.e. Data_A的解或者Data_B的解）是一系列对应用实例进行分配或迁移的决策动作，顺序由第一行开始，到最后一行，格式为

<实例ID>, <目标宿主机ID>

<实例ID>, <目标宿主机ID>

<实例ID>, <目标宿主机ID>

<实例ID>, <目标宿主机ID>

... ..

• 要求

- 最终所有实例都要部署到宿主机中
- 不能出现无效的实例id或宿主机id
- 请保存为submit_<YYMMDD_hhmmss>.csv。<YYMMDD_hhmmss>是结果生成时的时间戳，这是我们建议的结果命名方式

例子，

```
<实例的UID>, <目标宿主机的UID>
```

```
<实例的UID>, <目标宿主机的UID>
```

<实例的UID>, <目标宿主机UID>
<实例的UID>, <目标宿主机UID>
... ..

1.3.3 执行与评分（初赛）

- 可执行性

- 决策动作会按文件从上到下的顺序串行执行，若遇到不可执行的操作，评价程序会中断，并直接开始评价当前状态的得分

- 评分

- 根据执行完提交结果的最终状态，计算成本分数`total_cost_score`

- $total_cost_score = \frac{\sum_{t=1}^T \sum_{j=1}^m s_j^t}{T}$, 其中 s_j^t 代表机器 j 在时刻 t 的得分.
- $s_j^t = 0$ 如果机器 j 在时刻 t 没有实例部署
- 否则, $s_j^t = 1 + \alpha * (e^{\max(0, c - \beta)} - 1)$
 - ◊ c 是机器 j 在时刻 t 的 CPU 利用率
 - ◊ α, β 是惩罚系数, 在评分公式中 $\alpha = 10, \beta = 0.5$
- 如果提交结果不符合格式的规范, 成本分数为 `1e9`
- 成本分数越低越好

1.3.4 对方案的评测

- 初赛成绩以提交结果的评分为准
- 复赛阶段会要求排名前10的队伍提交代码与文档，进行方案评测。同时参考提交结果的评分及方案，进入决赛，角逐冠军
- 方案评测的细节，会于复赛前公布，基本会遵循以下原则：
 - 求解时间短(例如1小时以内，具体要求复赛确定)
 - 鼓励策略性模型(可以快速输出部分决策，但效果是前提)
 - 鼓励创新性
 - 鼓励灵活性

1.3.5 推荐的复赛提交格式（暂定）

在第二阶段比赛（复赛）接近尾声时，我们会要求排行榜排名前10的队伍提交针对复赛题目的计算出迁移方案的代码，进行线下评测。方案和线下评测的标准会在“评价标准”中说明，如果参赛队伍不能提供代码、或者提交代码与结果不匹配，相应的排行榜成绩无效。

下面是对提交代码的要求，建议参赛选手按照这个要求组织自己的代码，方便在取得好成绩以后进行提交（代码结构见注1）。

- 数据文件夹`data/*`.csv
 - 选手无需提交数据文件，我们会把初赛复赛用到的所有原始文件（与官网上的文件和文件名一致）放到`data`文件夹下，选手提交中间文件也放入该文件夹。注意的是，初始的时候`data`文件夹会被清空，并只放入原始文件
- 代码文件夹`code/*`.py（也可以用其他语言编写）
 - 读入文件的路径尽量使用相对路径，比如`../data/XX.csv`

- 要有main.py或者main.ipynb去运行所有代码得到最后结果（或其它编程语言的main），并将结果保存到submit文件夹
- 结果输出文件夹submit/*.csv
 - 存储提交的csv文件
 - 提交文件名称submit_Ymd_HMS.csv(e.g.submit_20180203_040506.csv)
- 代码的随机
 - 对于用到随机数的步骤，设定随机数。如果未设置随机数导致结果有随机性，将进行多轮运行取平均的方式，如果随机的误差交结果与答案间的误差将被取消决赛资格。由于代码会运行多次，为避免覆盖结果文件，请选手将每次生成的结果文件以时间方名，如注2所示

注1:提交文件夹结构

- project
- |--README.md
- |--data
- |--code
- |-- main.py or main.ipynb or <其它语言代码>
- |--submit
- |-- submit_20180203_040506.csv

注2:提交文件文件名代码

- # java for example
- import datetime
- data.to_csv("../submit/submit_" + datetime.datetime.now().strftime("%Y%m%d_%H%M%S") + ".csv"), header=N, index=False)

1.4 你可以用这份数据设计其它算法

下面的要求不是比赛的一部分，但同样是数据中心资源调度关心的目标。爱好者可以根据这份数据设计以下面需求之一为目的的调度算法，我们十分欢迎您与我们交流您的想法！

1. 同样是上述数据和问题，设计在线调度算法。所谓在线调度算法，是待调度的任务顺序地被调度器调度，而调度器不知道待调度任务中靠后的任务的信息。实践中，在线算法只能接近，但很难达到离线算法的效果。
2. 让算法更robustness。实际环境中，大量数据为建模预估产生的模型化数据，例如赛题中的cpu, mem分时占用曲线，如何在预估有偏差的前提下进行问题求解，或者如何在已知决策模型的前提下调整预估方法，也是充满挑战的问题。
3. 其它任何你能想到的使用这份数据可以设计的问题和算法。如果你对这个有兴趣，我们相信你会对我们第二阶段的比赛更加有兴趣，持续关注并一定参加我们的正式比赛！

解决方案

成熟解决方案 (https://www.aliyun.com/solution/all) | 专业架构师咨询 (https://www.aliyun.com/solution/all) | 行业架构师咨询

成功客户案例分享 (https://www.aliyun.com/solution/all) | 专属定制服务 (https://www.aliyun.com/solution/all)

热门产品	云服务器ECS (https://www.aliyun.com/product/ecs) spm=5176.8142029.631162.98.0IMSSV&recordcode=33010002000099	云数据库RDS (https://www.aliyun.com/product/rds) spm=5176.8142029.631162.98.0IMSSV&recordcode=33010002000099	云存储OSS (https://www.aliyun.com/product/oss) spm=5176.8142029.631162.98.0IMSSV&recordcode=33010002000099	NAT网关 (https://www.aliyun.com/product/nat) spm=5176.8142029.631162.98.0IMSSV&recordcode=33010002000099	负载均衡 (https://www.aliyun.com/product/slb) spm=5176.8142029.631162.98.0IMSSV&recordcode=33010002000099	域名注册 (https://www.aliyun.com/product/domain) spm=5176.8142029.631162.98.0IMSSV&recordcode=33010002000099	网站建设 (https://www.aliyun.com/product/market) spm=5176.8142029.631162.98.0IMSSV&recordcode=33010002000099	大数据 (https://www.aliyun.com/product/bigdata) spm=5176.8142029.631162.98.0IMSSV&recordcode=33010002000099
用户热搜	网站备案 (https://beian.aliyun.com) 云安全 (https://market.aliyun.com/security)	网安法 (https://market.aliyun.com/security)	CDN加速 (https://www.aliyun.com/product/cdn) 云安全 (https://market.aliyun.com/security)	API网关 (https://www.aliyun.com/product/apigateway) 云安全 (https://market.aliyun.com/security)	企业邮箱 (https://www.aliyun.com/product/enterprise-mail) 云安全 (https://market.aliyun.com/security)	whois查询 (https://www.aliyun.com/product/whois) 云安全 (https://market.aliyun.com/security)	视频直播 (https://www.aliyun.com/product/video-live) 云安全 (https://market.aliyun.com/security)	视频转码 (https://www.aliyun.com/product/video-transcode) 云安全 (https://market.aliyun.com/security)
更多推荐	全民云计算 (https://promotion.aliyun.com/one/qmcc) 合规安全解决方案 (https://www.aliyun.com/solution/security/compliance)	免费套餐 (https://promotion.aliyun.com/one/qmcc) 合规安全解决方案 (https://www.aliyun.com/solution/security/compliance)	学生机 (https://promotion.aliyun.com/one/qmcc) 合规安全解决方案 (https://www.aliyun.com/solution/security/compliance)	IT论坛 (https://bbs.aliyun.com) 合规安全解决方案 (https://www.aliyun.com/solution/security/compliance)	数据可视化 (https://www.aliyun.com/product/data-visual) 合规安全解决方案 (https://www.aliyun.com/solution/security/compliance)	云虚拟机 (https://www.aliyun.com/product/ecs) 合规安全解决方案 (https://www.aliyun.com/solution/security/compliance)	com域名 (https://www.aliyun.com/product/domain) 合规安全解决方案 (https://www.aliyun.com/solution/security/compliance)	cn域名 (https://www.aliyun.com/product/domain) 合规安全解决方案 (https://www.aliyun.com/solution/security/compliance)

关于我们 (https://www.aliyun.com/about)
法律声明及隐私权政策 (http://terms.aliyun.com/legal-agreement/terms/suit_bu1_ali_cloud/suit_bu1_ali_cloud201710161525_98396.html)
廉正举报 (https://jubao.alibaba.com/index.html) 联系我们 (https://www.aliyun.com/contact) 加入阿里云 (https://www.aliyun.com/career)

阿里巴巴集团 (http://www.alibabagroup.com/cn/global/home) 淘宝网 (https://www.taobao.com/) 天猫 (https://www.tmall.com/) 聚划算 (https://ju.taobao.com/) 全球速卖通 (https://www.aliexpress.com/) 阿里巴巴国际交易市场 (https://www.alibaba.com/) 1688 (https://www.1688.com/) 阿里妈妈 (https://www.alimama.com/index.html) 飞猪 (https://www.alitrip.com/) 阿里云计算 (https://www.aliyun.com/) YunOS (https://www.yunos.com/) 阿里通信 (https://aliqin.tmall.com/) 万网 (https://wanwang.aliyun.com/) 高德 (http://www.autonavi.com/) UC (http://www.uc.cn/) 友盟 (https://www.umeng.com/) 虾米 (https://www.xiami.com/) 阿里星球 (https://www.alibabaplanet.com/) 来往 (https://www.laiwang.com/) 钉钉 (https://www.dingtalk.com/?lwfrom=20150205111943449) 支付宝 (https://www.alipay.com/)
© 2009-2018 Aliyun.com 版权所有 ICP证: 浙B2-20080101