**分布式任务调度框架**

**考试提醒**

1. **编程前请仔细阅读需求文档,理解需求文档预计需要花费10~30分钟来理解,请务必看清楚要求再动手,比赛试题包括需求描述文档和代码框架。**
2. **编程时建议经常保持能编译运行，确保可以获得已经实现功能的得分。**
3. **在阅读题目后请同时阅读随代码框架提供的单元测试用例代码和注释以帮助更好的理解题意，在完成答题后一定要确保随代码框架提供的单元测试用例执行通过。**
4. **每道题已经提供初步的框架，请在此框架上继续完成代码。随试题提供的代码框架\*.java文件使用UTF-8编码，请调整IntelliJ、eclipse等IDE的编码字符集为UTF-8，以防止源文件中的中文出现乱码。**
5. **试题框架提供的原有代码,除需要实现的方法体外的其他的内容(包括常量定义,方法签名等)均不能做任何修改。**
6. **可以根据需要在代码框架中新增属性,方法等，也可以在本类中调用其他自己开发的类。不建议使用其他第三方lib库，如果确实需要使用，请务必加到pom文件中。**
7. **除题目明确要求需要考虑的异常情况、输入校验外，其他异常情况和输入校验一律不需要考虑。**
8. **关于代码提交提交到你自己的github从fork过去的工程下。**

**比如路径为https://github.com/OwenTse/ExamDemo.git 其中OwenTse为gitHub Username**

**考试结束截止时间考试程序会根据各位提供的GitHub Username去拉取代码，后进行自动的单元测试**

## 1总体说明

本系统实现简易的分布式任务调度框架。

“分布式任务调度框架”是定时运行在分布式集群系统主控节点上,进行任务调度的控制程序。调度程序定义物理服务器(服务节点)上任务分配的规则。用户可以很容易地添加任务到分布式系统中,也可以从分布式系统中删除任务。如果有一个或多个物理服务器(服务节点)的任务占用资源量大幅度变化，分布式任务调度框架根据指定的任务调度策略将在服务节点之间重新分配任务。

本题目实现系统初始化、服务节点注册、服务节点注销、添加任务、删除任务、任务调度等功能

## 系统规则

**任务务调度策略**

分布式任务调度框架执行的调度策略为：

任意两台不同服务节点上的任务资源总消耗率的差值如果超过了用户定义的阈值，则启动任务迁移方案，按照迁移方案将任务重新分配到服务节点上，使所有服务器两两的占用率差值降到阈值以下。

符合阈值的迁移方案会有很多种，我们主要关心**最佳迁移方案**：

**最佳迁移方案要求**

※ **约束**：

* 总消耗率差值最小的迁移方案（迁移后，所有的物理服务器的总消耗率相同，或者两两消耗率差值最小）
* 尽量把任务平均分配到各个服务器上。（这里指任务数量，也就是两两服务器上的任务总数差值最小）
* 如果存在资源消耗率相同的任务，则优先将编号小的任务迁移到编号小的服务器上；（暂时不考虑）
* 如果迁移后，有任意两台服务器的总消耗率相同，则应保证编号小的服务器的运行任务总数量少；（大任务尽可能多分小）
* 如果迁移后，所有的物理服务器的总消耗率不相同，保证编号大的服务器的总消耗大于编号小的服务器的总消耗。（当任务超过平均平均值时，尽可能分大）
* 如果迁移后，满足以上要求的方案有多个，则应选择编号小的服务器上的任务编号升序序列最小。（假设有四个任务编号：1、2、3、4，分配到2台服务器, 满足以上要求有两个方案,

方案1 : 1,2任务在服务器1 ; 3,4任务在服务器2

方案2 : 2,3任务在服务器1 ; 1,4任务在服务器2

则 方案1 的 服务器1 的升序排列 12 <方案2 的 服务器1 的升序排列 23)

如下表是一个简单的任务调度场景，有两台物理服务器1和2，上面分别运行了1个和3个任务，用户设置的阈值为10：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 服务节点编号 | 消耗率1 | 消耗率2 | 消耗率3 | 总消耗率 |
| 1 | 任务1：30 | 无 | 无 | 30 |
| 2 | 任务2：30 | 任务3：30 | 任务4：30 | 90 |

表格1 任务资源消耗场景

在表格1中，服务器1的总消耗率为30，服务器2的总消耗率为90，差值超过了阈值10，所以启动任务调度程序，重新将这些任务分配到两台服务器上，使总消耗率的差值小于10。

如下表，是最佳迁移方案。任务调度启动后，迁移方案将任务2迁移到服务器1上，两服务器的总消耗率差值为0，小于阈值10，并且任务被平均到两台服务器上，每个服务器有两个任务：

|  |  |  |  |
| --- | --- | --- | --- |
| 服务节点编号 | 消耗率1 | 消耗率2 | 总消耗率 |
| 1 | 任务1：30 | 任务2：30 | 60 |
| 2 | 任务3：30 | 任务4：30 | 60 |

表格2 任务调度后任务分配

**系统初始化**

系统初始化后清空所有数据，已经注册到系统的服务节点信息和任务信息需要清理。在系统运行过程中也可以根据需要随时进行系统初始化工作。

**服务节点注册**

系统初始化后，服务节点可以通过注册接口注册到本系统，注册接口包括服务节点编号.

**服务节点注销**

删除服务节点。删除服务节点时，对于服务节点上运行的任务，将移到挂起队列中。

**添加任务**

通过任务添加接口加到系统的任务挂起队列中，等待任务调度程序来调度。 添加任务接口包括任务编号 和 资源消耗率。

**删除任务**

将在挂起队列中的任务 或 运行在服务节点上的任务删除。

**任务调度**

如果挂起队列中有任务存在，则进行根据上述的任务调度策略，获得最佳迁移方案，进行任务的迁移， 返回调度成功。

如果没有挂起的任务，则将运行中的任务，则根据上述的任务调度策略，获得最佳迁移方案；如果在最佳迁移方案中，任意两台不同服务节点上的任务资源总消耗率的差值小于等于调度阈值， 则进行任务的迁移，返回调度成功。如果在最佳迁移方案中，任意两台不同服务节点上的任务资源总消耗率的差值大于调度阈值，则不做任务的迁移，返回无合适迁移方案。

**查询当前任务状态**

查询获得所有已添加任务的任务状态列表。

## 系统约束

参赛者需要实现系统初始化、服务节点注册、服务节点注销、添加任务、删除任务、任务调度等功能。具体每个功能的规则，请参见下述具体的接口说明。

为了计算简便，任务资源消耗率以一个整型数字来表示，在任务添加时提供的任务资源消耗率。

简化假设，任务在服务器上的迁移没有代价。

系统规模范围为: 服务器节点数 \* 调度任务数 < 50。

# 操作说明

试题预先给出Schedule.java, TaskInfo.java及 ReturnCodeKeys.java类；其中， Schedule.java为该系统核心实现类，由参赛者在原有框架内实现；

ReturnCodeKeys.java为系统预先提供的操作返回错误码，

TaskInfo.java 为用于保存查询任务状态结果,

**请参赛者务必不要修改类中已有的资源**，否则导致评分错误，由自己负责。

## 系统初始化

**函数命名：** public int init()

**功能说明**:

系统初始化，会清空所有数据，包括已经注册到系统的服务节点信息、以及添加的任务信息，全部都被清理。执行该命令后，系统恢复到最初始的状态。

**参数说明：**

无

**输出说明：**

初始化成功，返回E001初始化成功。

未做此题返回E000方法未实现。

## 服务节点注册

**函数命名：** public int registNode(int nodeId)

**功能说明**:

系统初始化后，服务节点可以通过注册接口注册到本系统。

**参数说明：**

nodeId 服务节点编号, 每个服务节点全局唯一的标识, 取值范围： 大于0；

**输出说明：**

注册成功，返回E003:服务节点注册成功。

如果服务节点编号小于等于0, 返回E004:服务节点编号非法。

如果服务节点编号已注册, 返回E005:服务节点已注册。

## 服务节点注销

**函数命名：**public void unregistNode(int nodeId)

**功能说明**:

1. 从系统中删除服务节点；
2. 如果该服务节点正运行任务，则将运行的任务移到任务挂起队列中，等待调度程序调度。

**参数说明：**

nodeId服务节点编号, 每个服务节点全局唯一的标识, 取值范围： 大于0。

**输出说明：**

注销成功，返回E006:服务节点注销成功。

如果服务节点编号小于等于0, 返回E004:服务节点编号非法。

如果服务节点编号未被注册, 返回E007:服务节点不存在。

## 添加任务

**函数命名：** public int addTask (int taskId, int consumption)

**功能说明**:

将新的任务加到系统的挂起队列中，等待服务调度程序来调度。

**参数说明：**

taskId任务编号；取值范围： 大于0。

consumption资源消耗率；

**输出说明：**

添加成功，返回E008任务添加成功。

如果任务编号小于等于0, 返回E009:任务编号非法。

如果相同任务编号任务已经被添加, 返回E010:任务已添加。

## 删除任务

**函数命名：** public int deleteTask (int taskId)

**功能说明**:

将在挂起队列中的任务 或 运行在服务节点上的任务删除。

**参数说明：**

taskId任务编号；取值范围： 大于0。

**输出说明：**

删除成功，返回E011:任务删除成功。

如果任务编号小于等于0, 返回E009:任务编号非法。

如果指定编号的任务未被添加, 返回E012:任务不存在。

## 任务调度

**函数命名：** public int scheduleTask(int threshold)

**功能说明**:

如果挂起队列中有任务存在，则进行根据上述的任务调度策略，获得最佳迁移方案，进行任务的迁移， 返回调度成功

如果没有挂起的任务，则将运行中的任务则根据上述的任务调度策略，获得最佳迁移方案；

如果在最佳迁移方案中，任意两台不同服务节点上的任务资源总消耗率的差值小于等于调度阈值， 则进行任务的迁移，返回调度成功，

如果在最佳迁移方案中，任意两台不同服务节点上的任务资源总消耗率的差值大于调度阈值，则不做任务的迁移，返回无合适迁移方案

**参数说明：**

threshold系统任务调度阈值，取值范围： 大于0；

**输出说明：**

如果调度阈值取值错误，返回E002调度阈值非法。

如果获得最佳迁移方案, 进行了任务的迁移,返回E013: 任务调度成功;

如果所有迁移方案中，总会有任意两台服务器的总消耗率差值大于阈值。则认为没有合适的迁移方案,返回 E014:无合适迁移方案;

## 查询任务状态列表

**函数命名：** public int queryTaskStatus(List<TaskInfo> tasks)

**功能说明**:

查询获得所有已添加任务的任务状态, 以任务列表方式返回。

**参数说明：**

Tasks 保存所有任务状态列表；要求按照任务编号升序排列,

如果该任务处于挂起队列中, 所属的服务编号为-1;

在保存查询结果之前,要求将列表清空.

**输出说明：**

未做此题返回E000方法未实现。

如果查询结果参数tasks为null，返回E016:参数列表非法

如果查询成功, 返回E015: 查询任务状态成功;查询结果从参数Tasks返回。

## 附录

### 3.1错误码汇总

|  |  |
| --- | --- |
| **错误码** | **错误描述** |
| E000 | 方法未实现 |
| E001 | 初始化成功 |
| E002 | 调度阈值非法 |
| E003 | 服务节点注册成功 |
| E004 | 服务节点编号非法 |
| E005 | 服务节点已注册 |
| E006 | 服务节点注销成功 |
| E007 | 服务节点不存在 |
| E008 | 任务添加成功 |
| E009 | 任务编号非法 |
| E010 | 任务已添加 |
| E011 | 任务删除成功 |
| E012 | 任务不存在 |
| E013 | 任务调度成功 |
| E014 | 无合适迁移方案 |
| E015 | 查询任务状态成功 |
| E016 | 参数列表非法 |