# 任务调度系统

## 系统启动情况



系统启动后，会启动3个线程

a. 添加任务的线程

b. 执行具体的任务

c. 清理调度任务后留下的日志

## 集群任务调度

集群通过故障切换和负载平衡的功能，能给调度器带来高可用性和伸缩性。

目前集群只能工作在JDBC-JobStore（JobStore TX 或者JobStoreCMT）方式下，从本质上来说，是使集群上的每一个节点通过共享一个数据库来工作的。

Quartz启动两个维护线程，来维护数据库状态实现集群管理。

通过数据库中加锁的方式实现

对于运行在同一台服务器上的两个进程，可以通过加锁实现互斥执行，而对于运行在多个服务器上的任务仍然可以通过用加锁实现互斥，不过这个锁是分布式锁。这个分布式锁并没有那么神秘，实际上只要一个提供原子性的数据库即可。在数据库的locks表里有一个记录（lock record）

一个持有锁的服务器通过不断的发送心跳，来更新这个记录，心跳的内容就是持有锁的时间戳（update\_time），以及本机ip。也就是说，通过发送心跳来保证当前的服务器是活跃的，而其他服务器通过lock record中的update\_time来判断当前活跃的服务器是否超时，一旦超时，其他的服务器就会去争夺锁，接管任务的执行，并发送心跳更新active\_ip。

### 集群任务调度线程工作原理



## 任务负载均衡

### 模拟场景

有10个任务，3台机器

### 目的

要实现各个服务执行的任务数基本一致

### 具体情况分析

启动A服务时，10个任务全在该服务上调度

接着启动B服务，这时A、B服务执行任务的均值是5个，A服务会释放任务到只剩5个执行任务。B服务会自动的拉取服务执行

然后启动C服务，这时A、B、C服务执行任务的均值是3.3个，A、B服务其中一个会释放到剩4个服务，另外一个服务会释放到剩3个服务，C服务会自动拉起被释放的服务

### 实现

考虑增加leader选举机制

先查询是否存在leader服务，没有则选取活动状态的第一个服务为leader

leader主要完成的任务

分配不同服务调用的任务，来进行负载均衡。

负载均衡时，leader分配之后，先记录服务要删除的任务，删除成功后，变更记录状态，之后leader再来将该任务改为待添加。