

分布式调度器设计文档

共轭空间开源组

2017 年 9 月 24 日

Contents

1	简介	3
2	使用指南	4
2.1	如何安装	4
2.1.1	下载	4
2.1.2	安装	4
2.1.3	配置	4
2.2	如何启动	4
3	设计详情	5
3.1	任务生命周期	5
3.1.1	简介	5
3.1.2	状态周期	6
3.1.3	特殊事件	8
3.2	系统组件	9
3.2.1	master	9
3.2.2	worker	9
3.2.3	db	9
3.2.4	client	9
3.3	负载均衡	10
3.3.1	均衡策略	10
3.3.2	worker 判活	10
3.4	数据表设计	10
3.4.1	任务表	10
3.4.2	实例表	10
3.4.3	执行日志表	10
3.4.4	操作日志表	11
3.4.5	用户表	11
3.4.6	机器表	11
4	API	12
4.1	任务	12
4.1.1	新增任务	12
4.1.2	删除任务	12
4.1.3	修改任务	13
4.1.4	查询任务	13

4.2	执行日志	14
4.2.1	查询执行日志	14
4.3	操作日志	15
4.3.1	查询操作日志	15
4.4	用户	15
4.4.1	新增用户	15
4.4.2	删除用户	16
4.4.3	修改用户	16
4.4.4	查询用户	16
4.4.5	用户登陆	17
4.4.6	用户登出	17
4.5	机器	17
4.5.1	删除机器	17
4.5.2	查询机器	18
5	常见问题解答	19

Chapter 1

简介

chin 是一个支持定时 & 依赖的轻量级分布式调度系统，相比于 Linkin 开源的 azkaban 调度系统，它占用的内存更少；相比于 Airbnb 开源的 AirFlow 调度系统，它占用的 CPU 更少；相比于 Luigi 以及上述两者，它管理界面更友好。

由于是轻量级系统，因此它很容易移植到 ARM 系列的处理器上 (事实上我正是为了解决上述系统无法在 ARM 架构的机器下正常工作而写的这个调度系统)。它可以执行所有的 linux 命令，包括但并不只局限于

- command
- shell
- python
- java
- etc...

我们的后端语言选择 golang，因为 golang 部署容易，不需要额外的依赖。前端框架选择 vue，数据库选择 mysql，客户端选择 android。

Chapter 2

使用指南

2.1 如何安装

2.1.1 下载

2.1.2 安装

2.1.3 配置

2.2 如何启动

启动命令：

`./chin args`

其中，args 可选的参数为

- master: 启动 master 服务
- worker: 启动 worker 服务
- webserver: 启动 web 服务
- build_db: 清空数据表后重新建表，并新建 root 用户
- clean_db: 清空表中的所有数据
- mock_db: mock 数据

Chapter 3

设计详情

3.1 任务生命周期

3.1.1 简介

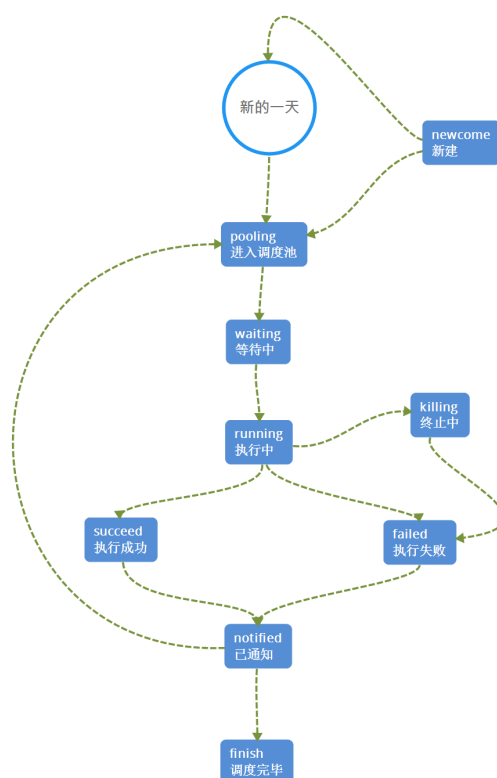


Figure 3.1: 任务生命周期

任务的生命周期共有八个状态：新建 (newcome)、进入调度池 (pooling)、等待中 (waiting)、执行中 (running)、执行成功 (succeed)、执行失败 (failed)、终止中 (killing)、已通知 (notified)、调度完毕 (finish)。其中，newcome 是一个逻辑上存在的状态，在物理上并不存在。各状态的转换流程如图所示。

3.1.2 状态周期

新建 (newcome)

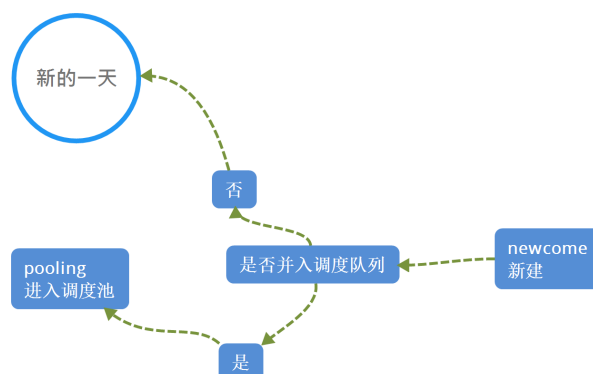


Figure 3.2: newcome 状态

newcome 只是逻辑上存在的状态，在任务新建时，其状态为 newcome 状态。处于 newcome 状态下的任务会有两种行为，如果用户点击并入队列，那么就会新建一个任务实例放进调度队列，进入到 pooling 状态，否则，会一直处于 newcome 状态一直到往后一天进入例行的调度。

进入调度池 (pooling)

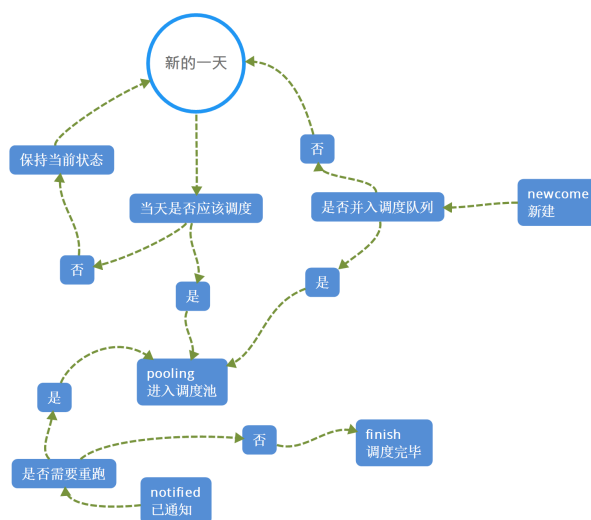


Figure 3.3: pooling 状态

有三种情况会使得任务实例进入 pooling 状态：第一种，正如 newcome 状态里所说的，如果一个任务新建后，用户点击并入队列，那么就会新建一个任务实例，并进入到 pooling 状态。第二种，当新的一天来临，如果这个任务的 `valid = true`，即例行调度为真，并且，当天日期符合调度周期和调度格式，那么会新建一个任务实例，状态为 pooling。第三种情况，如果某个任务实例处于 notified 状态，并且需要重跑，那么状态会更新为 pooling。

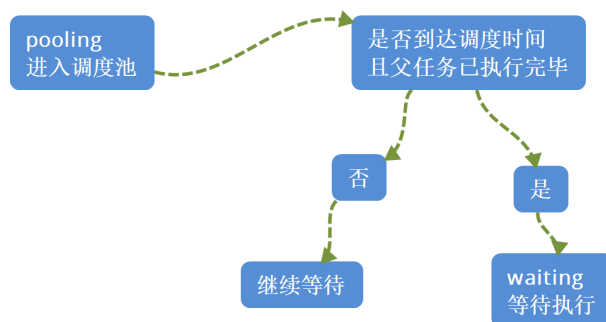


Figure 3.4: waiting 状态

等待执行 (waiting)

当某个任务实例处于 pooling 状态时，一旦它依赖的父任务执行完毕，并且到达它的调度时间，则会进入到 waiting 状态，等待调度器给它分配执行机器。

执行中 (running)

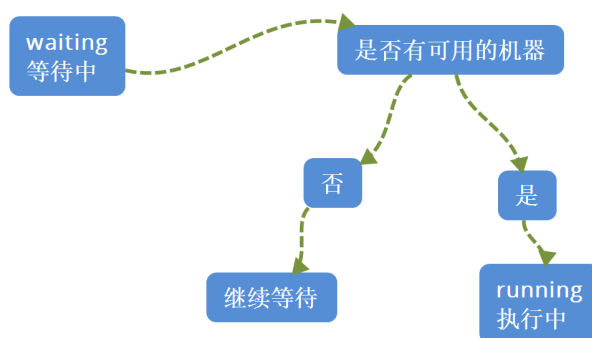


Figure 3.5: running 状态

当某个任务实例处于 waiting 状态时，一旦有可用的机器，则开始执行任务，并将状态更新为 running。

终止中 (killing)

当任务实例处于 running 状态时，用户手动点击终止执行操作，则会使任务进入 killing 状态，等待后台杀死进程。

执行成功/失败 (succeed/failed)

任务执行完毕后，根据任务的执行结果，将状态更新为成功或失败，由于被杀死的任务其返回值也是失败，所以状态也是失败。

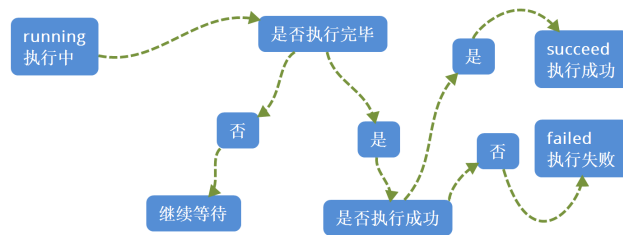


Figure 3.6: succeed/failed 状态

已通知 (notified)

执行成功或执行失败的任务，判断其是否需要通知，如果需要，则将信息通过邮件发送给用户，然后进入 notified 状态；如果不需要，则直接进入 notified 状态

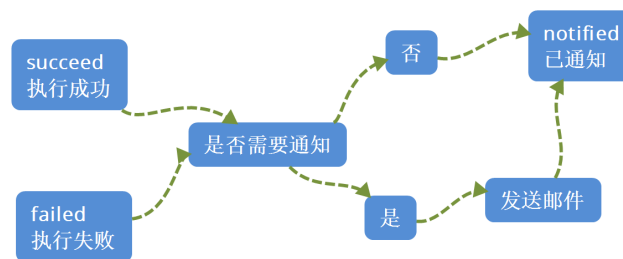


Figure 3.7: notified 状态

调度完毕 (finish)

通知完毕后的任务进入调度完毕状态。

3.1.3 特殊事件

调度器重启

调度器重启后，会检查有没有老任务（即非当天新建的任务）没有被创建实例，如果有，则会创建实例。

worker 重启

worker 重启后，会把属于自己并且状态为 running 的任务实例其状态置为 failed

并入队列

如果一个任务在当天本不应该执行，比如，某个任务调度周期为每周三，而今天是周五，我希望这个任务今天也调度，那么我点击并入队列，则会新建一个实例，进入到 pooling 状态

立即执行

如果某个任务不存在实例，则新建一个实例，然后跳过 pooling 状态，直接进入 waiting 状态，如果存在实例，则直接将实例置为 waiting 状态。

置为成功

立即将一个任务实例的状态设置为成功，如果一个任务位于 running 状态，则需要手动 killing 后才能强制成功

凌晨跨天时当天任务没有执行

如果跨天时，某个任务实例的状态为 pooling 或 waiting，那么调度器将直接把任务状态置为 failed

3.2 系统组件

3.2.1 master

用于分配任务、跟踪任务执行情况、worker 判活等。它要是挂了，client 还能正常工作，但不不会再有任务调度

3.2.2 worker

用于执行任务。它要是挂了，master 会报警，如果任务可以在别的机器上执行，那么 master 会把任务分配到别的机器上。

3.2.3 db

用于存储元数据和系统日志。整个系统中,MySQL 是核心，所有的组件都围绕 MySQL 进行通讯，如果 MySQL 挂了，那所有组件都不能工作了,我们称之为：真·药丸。

3.2.4 client

用户终端，两个版本，PC 端和 Android 端，用于可视化的对任务进行增删改查。

3.3 负载均衡

3.3.1 均衡策略

3.3.2 worker 判活

3.4 数据表设计

3.4.1 任务表

task		
task_id	bigint	任务 ID
task_name	string	任务名
command	string	任务执行命令
father_id	string	父任务 ID, 以逗号间隔, -> task.task_id
valid	tinyint	是否例行调度, 0-不调度 1-调度
machine_pool	string	可执行机器 ID, 以逗号间隔, -> machine.machine_id
owner_id	string	任务所属者 ID, -> user.user_id
schedule_type	string	调度周期
schedule_format	string	调度格式
create_time	string	创建时间
update_time	string	更新时间

3.4.2 实例表

task		
instance_id	bigint	实例 ID
task_id	string	任务 ID, -> task.task_id
status	string	任务实例所在的生命周期状态
create_time	string	创建时间
update_time	string	更新时间

3.4.3 执行日志表

log		
log_id	bigint	日志 ID
instance_id	bigint	实例 ID, -> instance.instance_id
machine_id	bigint	执行机器 ID, -> machine.machine_id
output	string	日志输出
create_time	string	创建时间
update_time	string	更新时间

3.4.4 操作日志表

action		
action_id	bigint	操作日志 ID
user_id	bigint	操作者 ID, -> user.user_id
content	string	操作内容
create_time	string	创建时间
update_time	string	更新时间

3.4.5 用户表

user		
user_id	bigint	用户 ID
user_name	string	用户名
password	string	登陆密码
email	string	用户邮箱
create_time	string	创建时间
update_time	string	更新时间

3.4.6 机器表

machine		
machine_id	bigint	机器 ID
machine_name	string	机器名
ip	string	机器 IP
mac	string	机器 MAC 地址
cpu_load	int	CPU 负载
memory_load	int	内存负载
alive	int	是否活着
create_time	string	创建时间
update_time	string	更新时间

Chapter 4

API

4.1 任务

4.1.1 新增任务

Method & URL

POST /api/task/new

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user_id:** bigint, required, 用户 ID
- **task_name:** string, required, 任务名
- **command:** string, required, 任务执行命令
- **father_id:** array[bigint], required, 任务的父任务
- **valid:** {true, false}, required, 任务是否例行调度
- **machine_pool:** array[bigint], required, 可执行该任务的机器列表
- **schedule_type:** {'once', 'day', 'week', 'month'}, required, 调度周期
- **schedule_format:** string, required, 调度格式

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息

4.1.2 删除任务

Method & URL

POST /api/task/delete, 本操作只能由任务所有者或 root 用户操作

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID
- **task__id:** bigint, required, 任务 ID

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息

4.1.3 修改任务

Method & URL

POST /api/task/modify, 本操作只能由任务所属者或 root 用户操作

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID
- **task__name:** string, required, 任务名
- **command:** string, required, 任务执行命令
- **father__id:** array[bigint], required, 任务的父任务
- **valid:** {true, false}, required, 任务是否例行调度
- **machine__pool:** array[bigint], required, 可执行该任务的机器列表
- **schedule__type:** {'once', 'day', 'week', 'month'}, required, 调度周期
- **schedule__format:** string, required, 调度格式

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息

4.1.4 查询任务

Method & URL

POST /api/task/query

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID
- **batch:** {true, false} required, 是否分页查询
- **task__id:** bigint, optional, 任务 ID, 当 batch=false 时必选
- **page:** int, optional, 页数, 当 batch=true 时必选
- **page__size:** int, optional, 页大小, 当 batch=true 时必选

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息
- **data:** {array[json], json}, 当 batch=true 时返回任务信息的 json, 当 batch=false 时返回任务信息 json 的 array

4.2 执行日志

4.2.1 查询执行日志

Method & URL

POST /api/log/query

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID
- **batch:** {true, false}, required, 是否分页查询
- **given__task:** {true, false}, required, 是否只查询某个任务的日志
- **task__id:** bigint, optional, 任务 ID, 当 given_task=true 时必选
- **log__id:** bigint, optional, 日志 ID, 当 given_task=false 并且 batch=false 时必选
- **column:** array[string], optional, 查询的字段, 若不提供则查询所有的字段
- **page:** int, optional, 页数, 当 batch=true 时必选
- **page__size:** int, optional, 页大小, 当 batch=true 时必选

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息
- **data:** {array[json], json}, 当 batch=true 时返回日志 json 的 array, 当 batch=false 时返回任务信息 json

4.3 操作日志

4.3.1 查询操作日志

Method & URL

POST /api/action/query

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID
- **batch:** {true, false}, required, 是否分页查询
- **page:** int, optional, 页数, 当 batch=true 时必选
- **page__size:** int, optional, 页大小, 当 batch=true 时必选

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息
- **data:** {array[json], json}, 当 batch=true 时返回操作日志的 json, 当 batch=false 时返回操作日志 json 的 array

4.4 用户

4.4.1 新增用户

Method & URL

POST /api/user/new, 本操作只能由 root 用户操作

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__name:** string, required, 用户名
- **password:** string, required, 登陆密码
- **email:** string, required, 用户邮箱

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息

4.4.2 删除用户

Method & URL

POST /api/user/delete, 本操作只能由 root 用户操作

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息

4.4.3 修改用户

Method & URL

POST /api/user/modify, 非 root 只能修改自身, root 用户可以修改其它用户

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID
- **password:** string, required, 登陆密码
- **email:** string, required, 用户邮箱

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息

4.4.4 查询用户

Method & URL

POST /api/user/query

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID
- **page:** int, optional, 页数, 当 batch=true 时必选
- **page__size:** int, optional, 页大小, 当 batch=true 时必选

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息
- **data:** array[json], 用户信息 json 的 array

4.4.5 用户登陆

Method & URL

POST /api/user/login

请求 JSON 字段

- **user__name:** string, required, 用户名
- **password:** string, required, 登陆密码

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息
- **token:** string, 会话 session 的 token

4.4.6 用户登出

Method & URL

POST /api/user/logout

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息

4.5 机器

4.5.1 删除机器

Method & URL

POST /api/machine/delete, 本操作只能由 root 用户操作

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID
- **machine__id:** bigint, required, 机器 ID

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息

4.5.2 查询机器

Method & URL

POST /api/machine/query

请求 JSON 字段

- **token:** string, required, 用户登陆后获取到的 session
- **user__id:** bigint, required, 用户 ID
- **page:** int, optional, 页数, 当 batch=true 时必选
- **page__size:** int, optional, 页大小, 当 batch=true 时必选

返回 JSON 字段

- **status:** {'SUCCEED', 'FAILED'}, 返回状态
- **info:** string, 返回信息
- **data:** array[json], 机器信息 json 的 array

Chapter 5

常见问题解答