

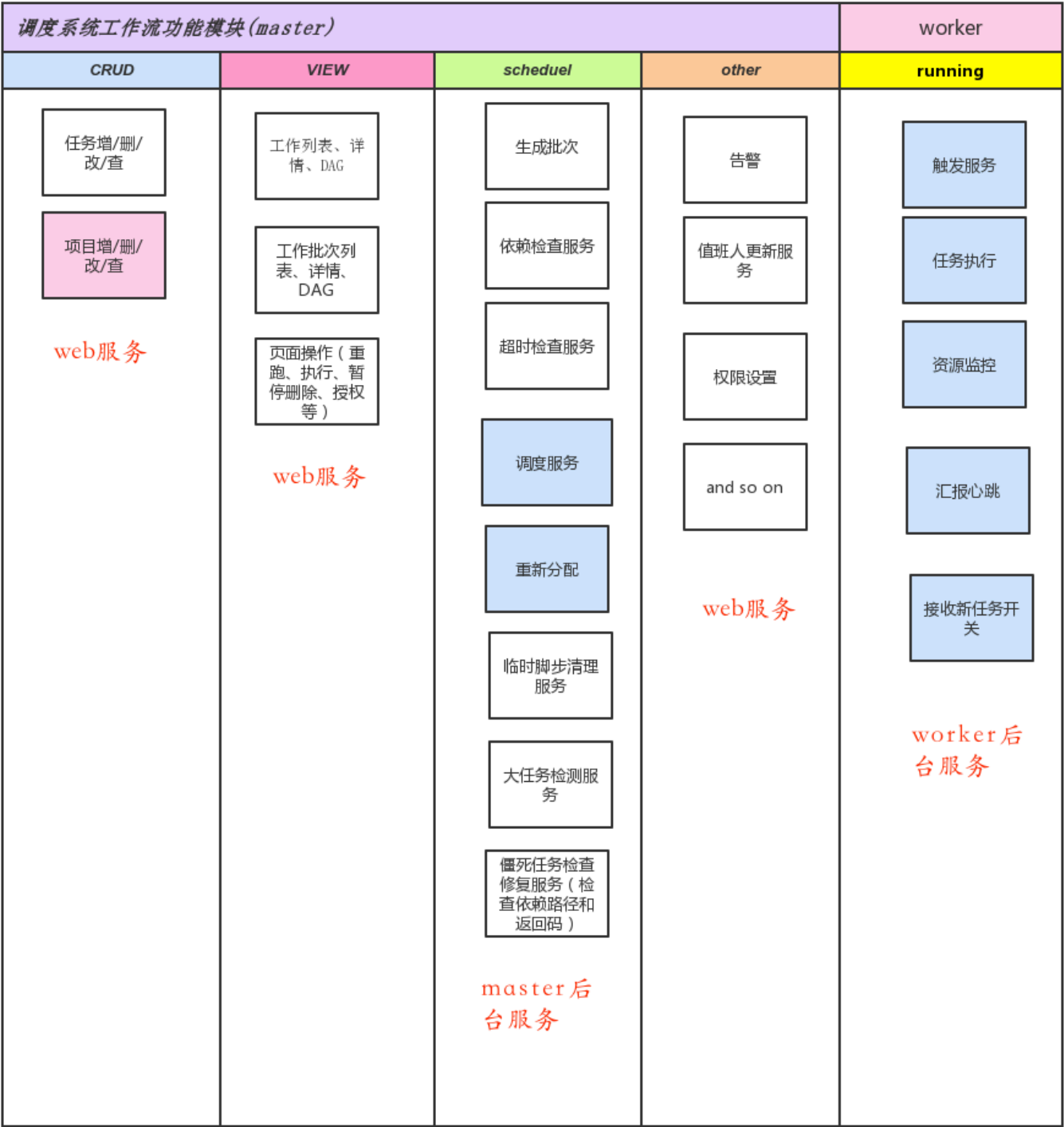
调度三期系统架构

2. 系统主要架构

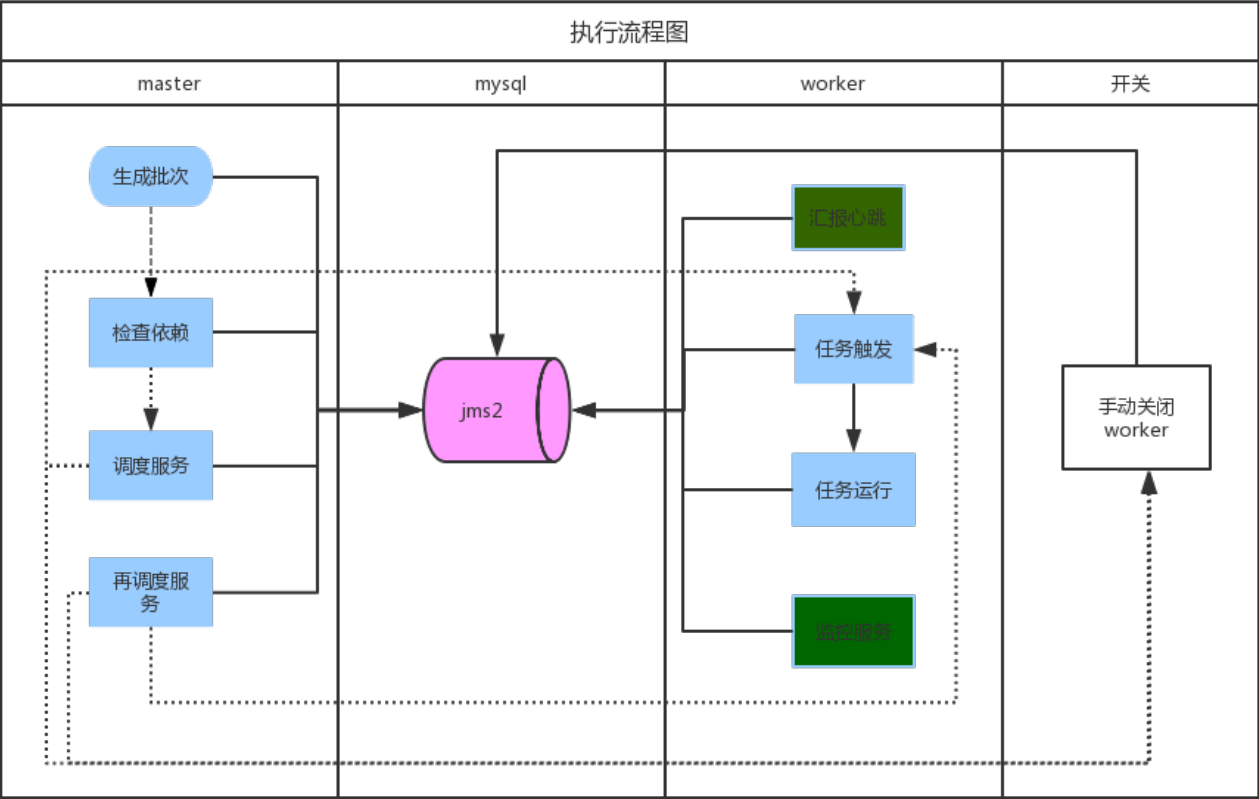
模块关系图、各模块功能、操作完成后涉及的数据表字段修改、操作完成后涉及的任务状态修改

一、调度三期实现master/slave架构

功能模块图如下：

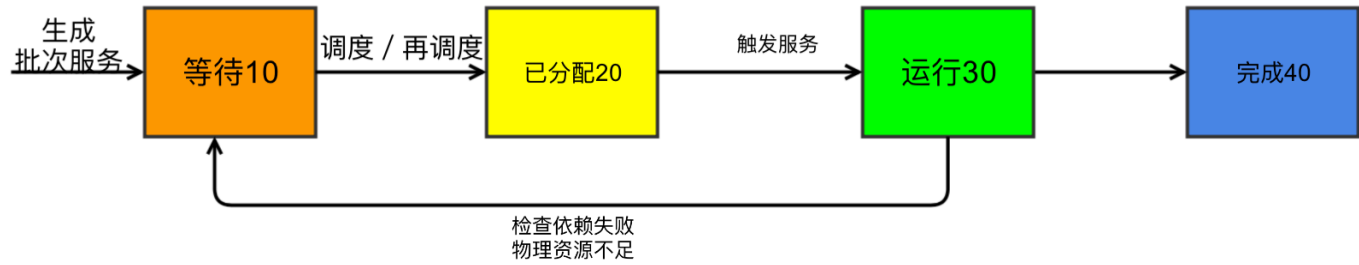


后台调度执行流程图：



二、任务状态转换图

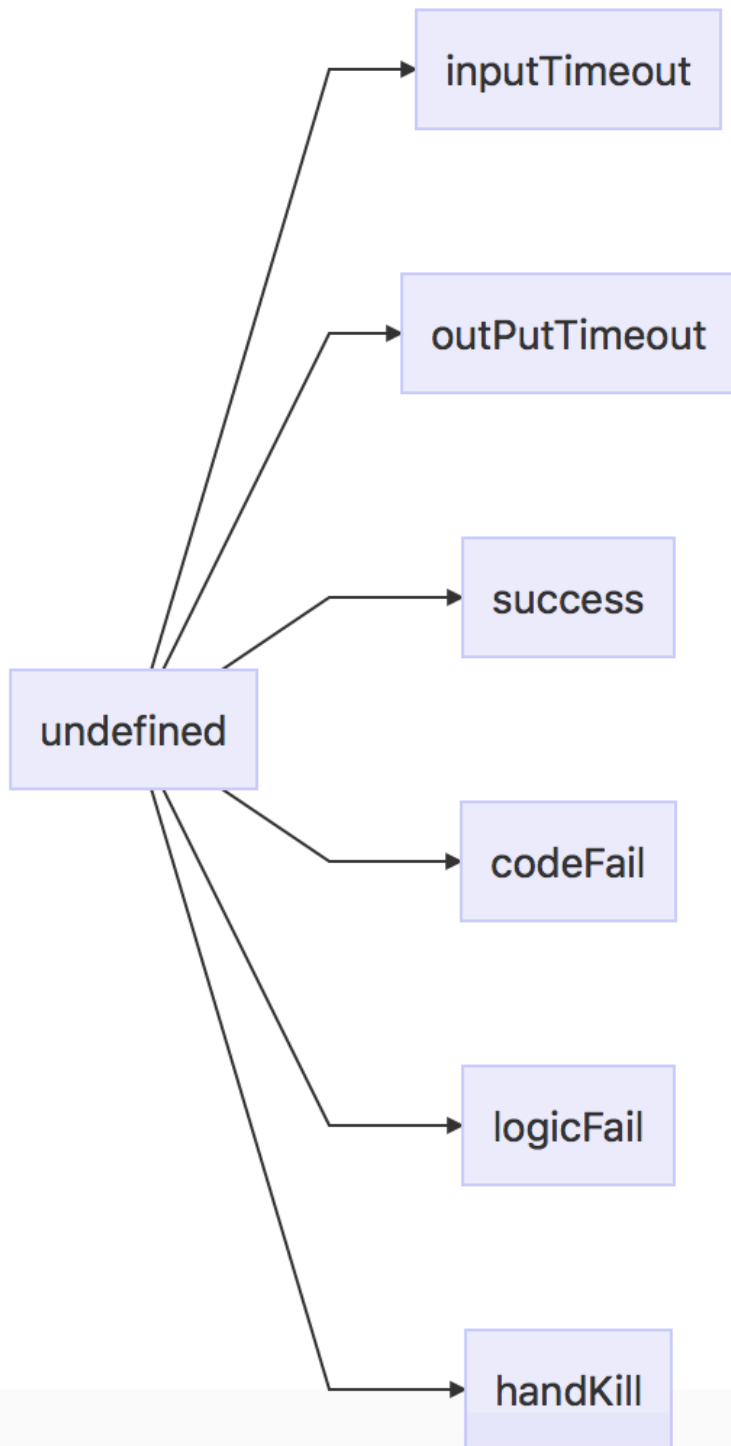
运行状态（0：等待，1：Y已分配，2：运行，3：完成）

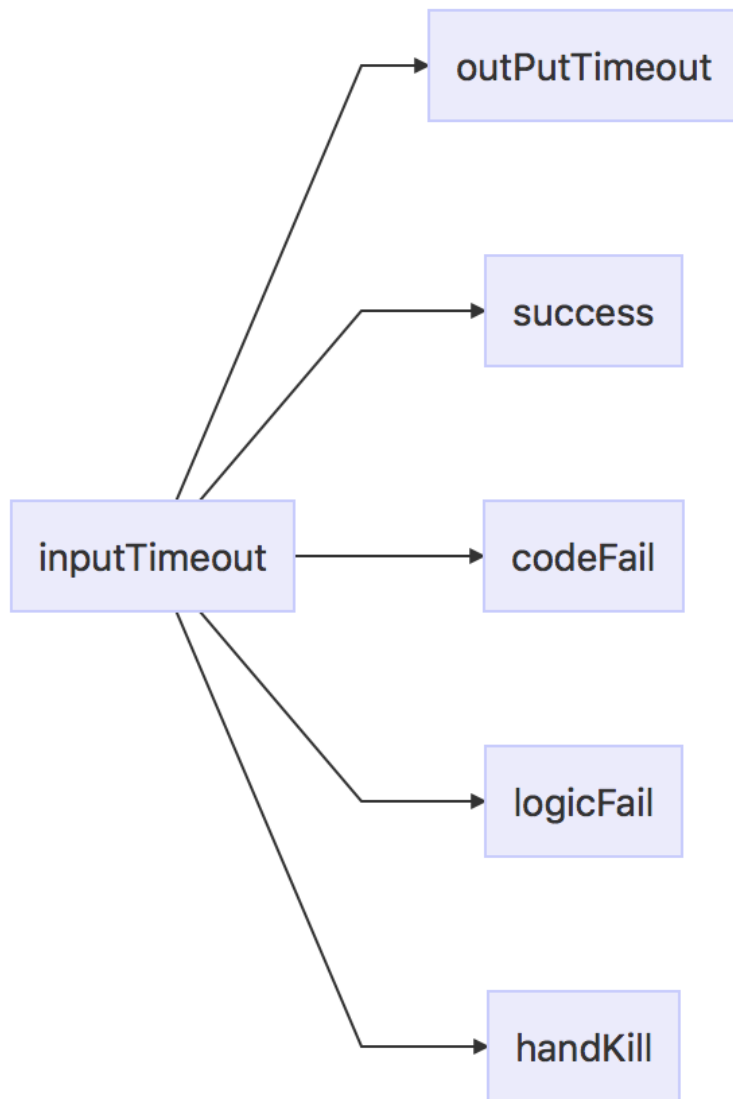


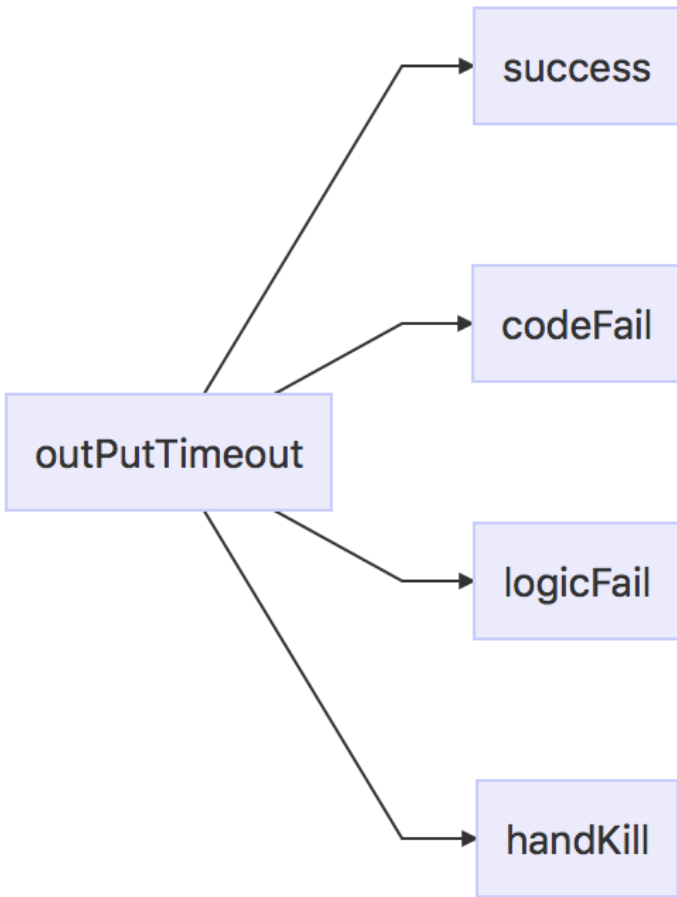
运行状态（state）：



结果状态（finishState）：







三、后台关键模块

说明：此部分主要包括master和worker后台服务关键模块的功能说明、逻辑实现、操作完成后涉及的数据表字段修改、操作完成后涉及的任务状态修改

任务的状态转换流程图如下：

1. master后台服务模块

1.1 生成批次服务

1.1.1 功能描述

通过周期定时扫描任务表（task）生成任务批次。仅生成批次，不真正执行任务。

1.1.2 性能要求

- 高可用：集群调度，某台机器宕机不影响任务批次生成。
- 可恢复：集群全部宕机，重启后可重新生成宕机期间未生成批次；并具有冷启动功能
- 准确性：生成的可执行批次必须精确，不能有重复或遗漏生成任务批次的情况
- 健壮性：当调度异常时，任务生成失败重试3次，再次失败发送告警给任务负责人，告警方式根据任务优先级和告警规则而定；大批量生成失败（失败任务数大于10个）电话、短信、邮件告警给调度系统管理员及值班人员。

1.1.3 程序实现

1.1.3.1 生成批次

任务调度基于quartz集群，设置4种定时触发任务，通过配置保证性能要求。

- 月

每月1号凌晨0点05分，触发以月为产出周期的任务，生成本月需要执行的任务的批次和相关配置信息存入任务批次表（task_work），并根据任务的定义解析出任务批次的依赖路径和产出路径存入任务批次从表（task_data）表，解析第三方依赖数据存入任务批次从表（tthird_data）表，解析任务逻辑父依赖批次存入（tparent_data）表。生成完的工作流批次此时处于等待状态。

- 周

每周一凌晨0点02分，触发以周为产出周期的任务，生成当天需要执行的任务的批次和相关配置信息存入任务批次表（task_work），并根据任务的定义解析出任务批次的依赖路径和产出路径存入任务批次从表（task_data）表，解析第三方依赖数据存入任务批次从表（tthird_data）表，解析任务逻辑父依赖批次存入（tparent_data）表。生成完的任务批次此时处于等待状态。

- 天

每天凌晨0点01分，触发以天为产出周期的任务，生成当天需要执行的任务的批次和相关配置信息存入任务批次表（task_work），并根据任务的定义解析出任务批次的依赖路径和产出路径存入任务批次从表（task_data）表，解析第三方依赖数据存入任务批次从表（tthird_data）表，解析任务逻辑父依赖批次存入（tparent_data）表。生成完的任务批次此时处于等待状态。

- 小时

每个小时01分，触发以小时为产出周期的任务，生成当天需要执行的任务的批次和相关配置信息存入任务批次表（task_work），并根据任务的定义解析出任务批次的依赖路径和产出路径存入任务批次从表（task_data）表，解析第三方依赖数据存入任务批次从表（tthird_data）表，解析任务逻辑父依赖批次存入（tparent_data）表。生成完的任务批次此时处于等待状态。

1.1.3.2 手动触发

用户可在调度系统中手动触发任务生成批次。

1.1.3.2.1 执行

用户可通过调度系统手动执行任务，调度时间为当前时间。当用户选择执行某个任务时，调度系统首先会生成一个此任务当前调度时间的批次代码，调度系统检查此任务是否有相同的批次代码并且这个批次代码对应的批次还未完成，如果有则不允许执行，手动执行失败；反之和定时调度一样生成任务批次以及相关的从表信息。生成完批次处于等待状态。

1.1.3.2.1 重跑

用户可手动重跑某任务，重跑时间区间用户可自行输入，但是重跑区间的最晚时间不得晚于当前时间。重跑功能与执行功能主要有以下两点不同。

执行：一次只能调度某个任务的一个周期即只会生成一个批次且这个批次的调度时间为当前时间；

重跑：用户可以输入一个重跑时间段，一次可以重跑多个周期，调度时间为重跑的时间区间内每个周期对应的时间。

执行：只能调度任务本身，不能调度其他依赖于此任务的任务；

重跑：可以选择只调度此任务本身（默认），也可以选择级联调度（数据级联/逻辑级联），选择级联调度时除了生成此任务的相应批次还触发依赖（子孙任务）于此任务的任务生成对应的批次。每个任务批次的生成规则和执行一样。

1.1.3.3 冷启动

冷启动可支持某个时间段内任务批次手动触发生成，用于定时生成批次服务失效时或需要批量生成过去某个时间段的批次的场景，冷启动功能只有系统超级管理员有权限。

1.1.3.3.1 冷启动某个周期内全部启用任务

传入一个时间段和调度周期（月、周、日、小时），系统自动生成这个时间段内此周期全部启用任务的批次，如果某个任务在这个时间段有未完成的批次，则这个批次不在重新生成但是需要记录和发邮件告知管理员；

1.1.3.3.2 冷启动某个周期内某个组全部启用任务

传入一个时间段和调度周期（月、周、日、小时）和组id，系统自动生成该组这个时间段内此周期全部启用任务的批次，如果某个任务在这个时间段有未完成的批次，则这个批次不在重新生成但是需要记录和发邮件告知管理员；

1.1.3.3.2 冷启动某个周期内指定任务

传入一个时间段和调度周期（月、周、日、小时）和英文任务名称列表，系统自动生成这批任务这个时间段内此周期全部启用任务的批次，如果某个任务在这个时间段有未完成的批次，则这个批次不在重新生成但是需要记录和发邮件告知管理员；

1.1.4涉及的数据表字段修改

1.1.5涉及的任务状态修改

1.2依赖检查服务

1.2.1 功能描述

周期性检查任务批次的依赖状态，包括检查路径依赖、第三方依赖、逻辑依赖。

1.2.2 性能要求

- 及时性：整个依赖检查延时不得超过10分钟，延时超过10分钟需要短信、邮件给调度系统管理员及值班人员并电话告警值班人员
- 高可用：集群调度，某台机器宕机不影响依赖检查及时性
- 可恢复：集群全部宕机，重启后可立即恢复依赖检查服务；并具备冷触发功能
- 健壮性：当异常时，单个依赖检查内部重试3次，再次失败发送短信、报警邮件给调度系统管理员及值班人员，并发送电话告警给值班人员

1.2.3 程序实现

调度系统程序每30秒触发依赖检查服务，依赖检查服务首先查询启用任务处于等待状态的批次（taskwork），并到任务批次从表(task_data/tparent_data/tthird_date)分别检查依赖是否满足。任务批次从表（taskdata）获取到当前所有不存在的hdfs依赖路径，通过调用hadoop api查询当前的依赖路径是否已经存在，若存在则更新taskdata表中对应路径的状态为已存在；查询任务批次从表（tthird_data）获取当前所有不满足的第三方依赖，通过查询第三方依赖数据表（thriddependdata）获取当前第三方依赖数据是否已经满足，若已经满足则更新任务批次从表(tthird_data)中对应的第三方依赖为已满足；查询任务批次从表（tparent_data）获取当前所有不满足的任务依赖批次，通过查询任务批次表（taskwork）获取当前任务依赖是否已经满足，若已经满足则更新任务批次从表(tparent_data)中对应的任务依赖为已满足。

1.2.4涉及的数据表字段修改

1.2.5涉及的任务状态修改

1.3超时检查服务

1.3.1 功能描述

周期性检查出调度系统中依赖超时和产出超时的任务批次，并将超时的任务批次根据此任务的告警规则选择告警方式告警给相关负责人，并将状态置为依赖超时或产出超时状态

1.3.2 性能要求

性能：告警延迟不得超过2分钟即超时告警从调度系统发出不得晚于应该告警时间后2分钟；

健壮：不得漏掉一条超时告警，告警发送需要增加失败重试机制，如果失败重试后仍然无法发送需要记录详细日志并发送邮件告警给调度系统管理员

1.3.3 程序实现

每分钟第10秒，触发依赖超时检查和任务执行超时检查。

- 依赖超时检查：查询任务批次表(taskwork)中状态为等待且配置了依赖超时，并且调度时间+依赖超时设置早于当前时间的批次，查询出的批次即为依赖超时了的批次，需要发送告警给相关人员并更新相关任务批次的finishState为依赖超时状态。
- 产出超时检查：查询任务批次表(taskwork)中完成状态finshstate为未成功且配置了产出超时，并且调度时间+产出超时设置早于当前时间的批次，查询出的批次即为产出超时了的批次，需要发送告警给相关人员并更新相关任务批次的finishState为产出超时状态。

1.3.4涉及的数据表字段修改

1.3.5涉及的任务状态修改

1.4调度服务

1.4.1 功能描述

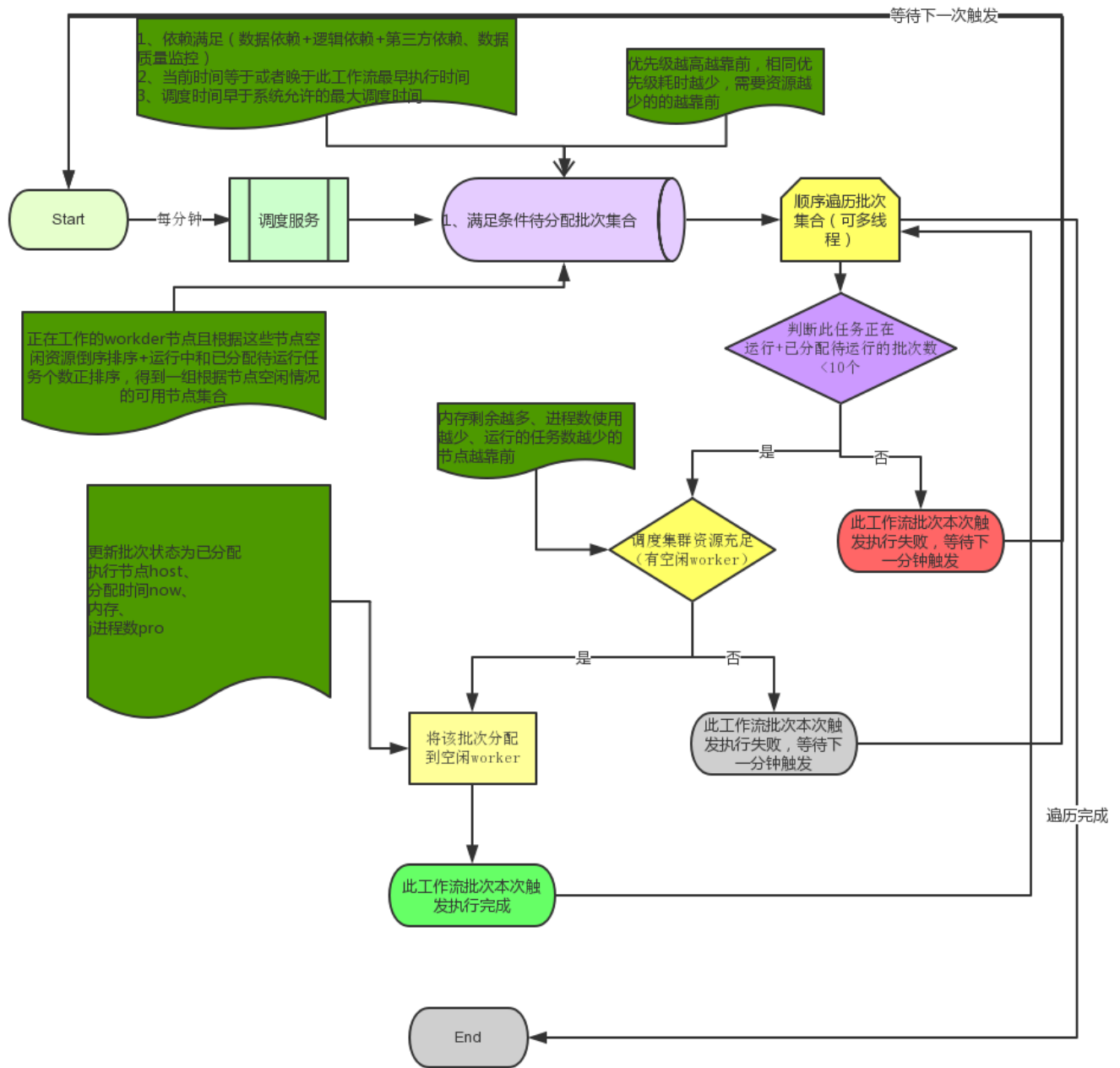
调度服务负责将处于运行状态(state)为等待状态且满足依赖(数据依赖、任务逻辑依赖、第三方依赖、熔断监控、最早执行时间)的批次分配可执行的节点并将任务运行状态更新为已分配待执行状态

1.4.2 性能要求

性能：满足依赖的任务分配延迟不得大于5分钟

1.4.3 程序实现

调度服务负责查询出所有启用任务且处于等待的任务批次，调度服务同时只能有一个执行。



1. 查询出系统允许的最晚调度时间；
2. 查询出满足数据依赖和逻辑依赖和第三方依赖和任务为启用状态和调度时间早于系统允许的最晚调度时间的批次列表 SQL (监控这一步还未加上，等二期监控加上后再在三期加上)；
3. 获取当前启用的worker列表【这步去掉，可以直接在第7步实时获取】；
4. 依此遍历从第二步查询出来的满足依赖的批次，判断此任务当前运行数是否达到最大，如果达到最大则更新工作流批次消息，告诉用户【此任务同时运行的任务数达到最大？个，等待下次调度】，此任务本次调度结束等待下次调度；
5. 判断任务的执行账号是否可以使用。如果没有权限使用则直接让这个任务批次失败，并提示失败内容为【调度失败，失败原因：此任务负责人没有权限使用此账号？运行任务】，此任务本次调度结束等待下次调度；
6. 判断此任务是否达到任务最早执行时间。此任务本次调度结束等待下次调度
7. 获取可执行此任务机器列表
8. 将任务批次分配到执行节点，更新任务状态和执行节点。此任务批次本次成功分配到执行节点，此任务调度完毕。继续进行其他批次调度。从第4步开始循环，直到第2步查询出来的批次处理完成。

1. 4. 4涉及的数据表字段修改

1. 4. 5涉及的任务状态修改

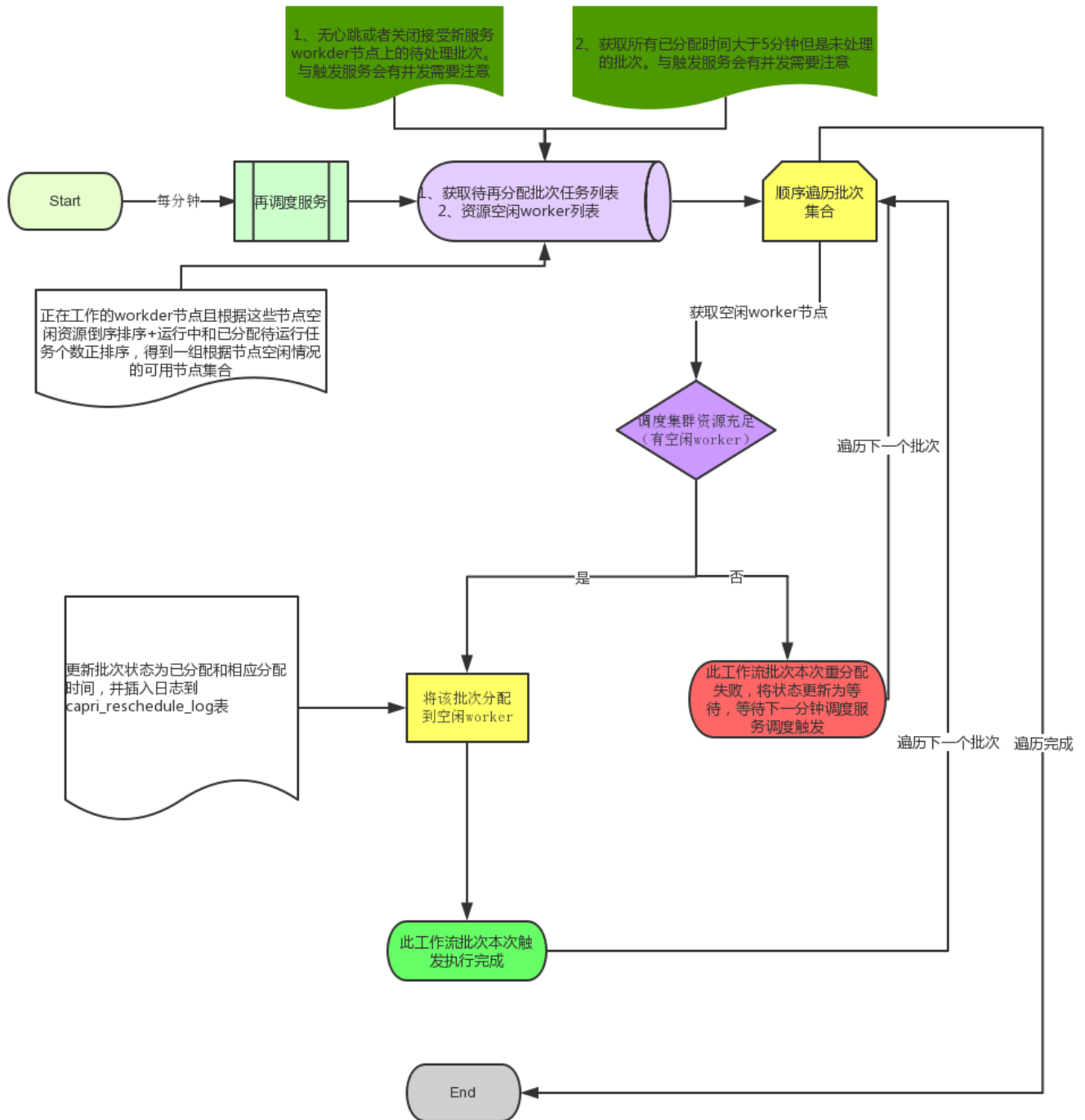
1.5重新分配（再调度）服务

1.5.1 功能描述

再调度服务负责将处于已调度但是未运行状态的批次重新分配可执行的节点并将任务运行状态更新为已分配、分配时间更改为当前时间，分配节点更改为新的节点，且需要在再分配日志表（`capri_reschedule_log`）表中级联这个批次上一次分配的信息。

1.5.2 性能要求

1.5.3 程序实现



1. 查询出系统允许的最晚调度时间;
2. 查询出已分配但是未执行且已分配时间大于系统允许的最大时间 dic_data:task_need_reschudle_time 默认5分钟;
3. 获取可执行此任务机器列表排除之前分配的机器;
4. 将任务批次分配到执行节点, 更新任务状态和执行节点、调度时间并插入调度日志到capri_reschedule_log表。此任务批次本次成功分配到执行节点, 此任务调度完毕。继续进行其他批次调度。从第3步开始循环, 直到第2步查询出来的批次处理完成。(再调度服务更新记录需要与worker节点的执行服务触发服务防止好并发问题。可以通过数据库层面做锁也可以通过java层面做)。

1. 5. 4涉及的数据表字段修改

1. 5. 5涉及的任务状态修改

1.6 监控服务

1.6.1 功能描述

监控服务负责监控整个调度集群的健康状况。包括各服务（依赖检查服务、超时服务、调度服务等）监控、可用worker节点监控、外部依赖环境（hadoop集群、QA全链路监控等）监控。监控服务中各个具体的监控实现可在不同地方，可以在master中，也可以在worker节点上或者是一个调度系统的任务；实现方式可以是Java也可以是其他脚本。

1.6.2 性能要求

及时性：调度集群中各种异常情况监控需要及时，告警需要准实时

1.6.3 程序实现

1.6.3.1 检查依赖服务监控

检查依赖服务监控主要检查监控依赖服务一次执行的耗时情况，通常情况一次执行耗时应该在1分钟以内，5分钟内的延迟用户和系统可接受的。但是大于5分钟的延迟说明检查依赖服务有问题，需要触发管理员告警，告警级别重要；大于十分钟的延迟需要触发管理员告警，告警级别严重。具体告警规则参见告警服务。

1. 详情参考调度二期检查依赖服务监控

http://wiki.intra.xiaojukeji.com/download/attachments/97835258/jms2_moni_taskInputListener.sh?version=1&modificationDate=1495091434000&api=v2

1.6.3.2 调度集群各节点负载监控（cpu/内存/存储等）

此监控通常由大数据op团队提供监控方案

1.6.3.3 调度master和worker进程监控

1. master :同时运行的master节点少于5台触发管理员告警，告警级别重要；少于3台触发管理员告警，告警级别严重。
2. worker: :同时运行的worker节点少于10台触发管理员告警，告警级别重要；少于5台触发管理员告警，告警级别严重。
3. worker: 监控在调度日志表，发现一定周期内某台机器上的任务多次被触发再调度服务监控。30分钟内单机触发在调度服务任务数大于10个触发管理员告警，告警级别重要；大于15个触发管理员告警，告警级别严重；10分钟内单机触发再调度服务次数大于3次触发管理员告警，告警级别重要；大于5个触发管理员告警，告警级别严重。

1.6.3.4 调度worker健康状况监控

详情参见 worker资源监控服务

1.6.3.5 生成批次服务监控

1. 监控生成批次服务单次运行耗时情况。通常情况一次执行耗时应该在5分钟以内。单次耗时大于10分钟的延迟说明服务有问题，需要触发管理员告警，告警级别重要；大于15分钟的延迟需要触发管理员告警，告警级别严重。具体告警规则参见告警服务。生成批次服务监控粒度根据调度周期（天、月、周、小时）而定。

http://wiki.intra.xiaojukeji.com/download/attachments/97835258/jms2_moni_taskInputListener.sh?version=1&modificationDate=1495091434000&api=v2

2. 监控哪些任务本该生成批次而没有生成批次

http://wiki.intra.xiaojukeji.com/download/attachments/97835258/jms2_moni.sh?version=1&modificationDate=1495091308000&api=v2

1.6.3.6 hdfs下载程序文件夯筑监控

hdfs
http://wiki.intra.xiaojukeji.com/download/attachments/97835258/check_hdfs_get_long.sh?version=1&modificationDate=1495091415000&api=v2

1.6.3.7 用户任务监控

worker

1.6.3.7 整个调度平台监控

capricornusMonitor,

1.6.4涉及的数据表字段修改

1.6.5涉及的任务状态修改

1.7 临时脚本清理

1.7.1 功能描述

每天2点10分，触发临时脚本清理任务，负责清理一周前的临时脚本和执行日志

1.7.2 性能要求

1.7.3 程序实现

每天2点10分，触发临时脚本清理服务，负责清理各worker节点上一周前的临时脚本和执行日志

1.7.4涉及的数据表字段修改

1.7.5涉及的任务状态修改

1.8 大任务监控服务

1.8.1 功能描述

每天早上10点，触发大任务检测服务，负责将前一天实际耗时与预计耗时差别较大的任务的预计耗时更新成前一天的实际耗时，并发送告警给任务的负责人；负责查询出前一天单个任务执行实际耗时大于120分钟的大任务，并发送告警给相关负责人。

1.8.2 性能要求

1.8.3 程序实现

每天早上10点，触发大任务检测服务，负责将前一天实际耗时与预计耗时差别较大的任务的预计耗时更新成前一天的实际耗时，并发送告警给任务的负责人；负责查询出前一天单个任务执行实际耗时大于120分钟的大任务，并发送告警给相关负责人。

1.8.4涉及的数据表字段修改

1.8.5涉及的任务状态修改

1.9 僵死任务修复服务

1.9.1 功能描述

每小时触发一次僵死任务检查修复服务，主服务负责查询出僵死任务，若任务此时无产出，则将任务置为失败，根据任务失败告警规则，告警给相关人员；运行在各worker节点上的检查服务主要负责检查与hdfs交互僵死的进程并对其进行修复。

1.9.2 性能要求

1.9.3 程序实现

每小时触发一次僵尸任务检查修复服务，主服务负责查询出僵死任务即处于运行状态且运行时间长且所在机器上并没有该进程的任务批次，将这类任务标记为完成，是否成功根据检查任务成功方式而定。如果此任务检查任务成功的方式为产出路径，则检查hdfs输出是否有产出，若任务此时无产出，则将任务置为失败，根据任务失败告警规则，告警给相关人员，有产出则标记为成功。如果是返回值方式，则直接标记为失败。

1.9.4涉及的数据表字段修改

1.9.5涉及的任务状态修改

1.10 告警模块

1.10.1 功能描述

告警模块用于支撑调度系统告警，包括系统层面管理员告警和用户侧面任务告警。告警方式要求支持电话、短信、邮件三种方式。

1.10.2 性能要求

- 及时性：从系统发出的告警应该准实时，对第三方依赖的延迟不包括在内，如因公司电话告警、短信服务、邮件服务导致的异常不包括在内
- 健壮性：当发送告警异常时，单个告警需要增加内部重试3次，如果再次失败需要发送邮件告警给调度系统管理员记录失败情况

1.10.3 程序实现

1.10.3.1 告警方式

告警方式有电话、短信、邮件三种。每个场景具体采用哪个告警方式根据具体业务而定。

1.10.3.2 告警规则

1.10.3.2.1 管理员告警

当调度系统出现异常时根据告警级别选择不同的告警方式告警给调度系统管理员及值班人员。

- 严重： 电话告警给主、次值班人+短信告警给主、次值班人+邮件告警给所有管理员；
- 重要： 短信告警给主、次值班人+邮件告警给调度系统所有管理员；
- 关注： 邮件告警给调度系统所有管理员；

1.10.3.3 用户告警

当用户任务异常、超时等需要告警时告警给用户。告警方式根据任务优先级和勾选是否电话告警而定。用户可设置任务是否需要电话告警并设置电话告警接收人，短信和邮件告警根据任务优先级确定，任务优先级为0-4的有短信和邮件告警，有限级为5-9的只有邮件告警

1.10.3.2.2 用户告警规则

- 默认告警规则：

当任务的优先级为0-4时触发短信和邮件告警，邮件发送给任务负责人所在部门的所有在调度系统中启用的用户和有此任务短信告警权限的用户；短信仅发送给此任务负责人所在部门的当前值班人和有此任务短信告警权限的用户；

优先级为5-9的仅发送邮件告警，发送给任务负责人所在部门的所有在调度系统中启用的用户和有此任务邮件告警的用户；

如果此任务配置了启用电话告警并配置了电话告警接收人则此任务在需要告警的地方会有电话告警。

- 辅助告警：

用户可以在任务上直接配置电话、短信、邮件接收人并可选择是否关闭默认告警规则。如果默认告警规则开启则告警接收人为默认告警规则接收人+辅助告警接收人；如果默认告警规则关闭则告警接收人仅为辅助告警规则接收人。

- 仅告警给负责人：

若任务配置了仅告警给负责人，则这个任务相关的告警仅发送给负责人，默认告警规则和辅助告警规则将失效。

1.10.4涉及的数据表字段修改

1.10.5涉及的任务状态修改

1.11 数据质量监控

1.11.1 功能描述

触发已经满足依赖的批次的数据质量监控，调用数据质量监控api告知此任务依赖满足请求数据质量监控回答是否可运行此批次，并将批次数据质量监控状态更改为已发送

1.11.2 性能要求

1.11.3 程序实现

查询出所有配置了数据质量监控的启用任务且任务批次的运行状态为等待且所有依赖都满足的批次，依次请求数据质量api，要求数据质量监控服务检查此任务是否可运行，调度系统只负责请求告知不需要等待结果返回。

数据质量监控发现批次可运行则调用调度系统api，更改批次数据质量监控状态为可运行。

1.11.4涉及的数据表字段修改

1.11.5涉及的任务状态修改

2.worker后台服务模块

worker是真正执行调度任务的部分。

2.1 触发服务

2.1.1 功能描述

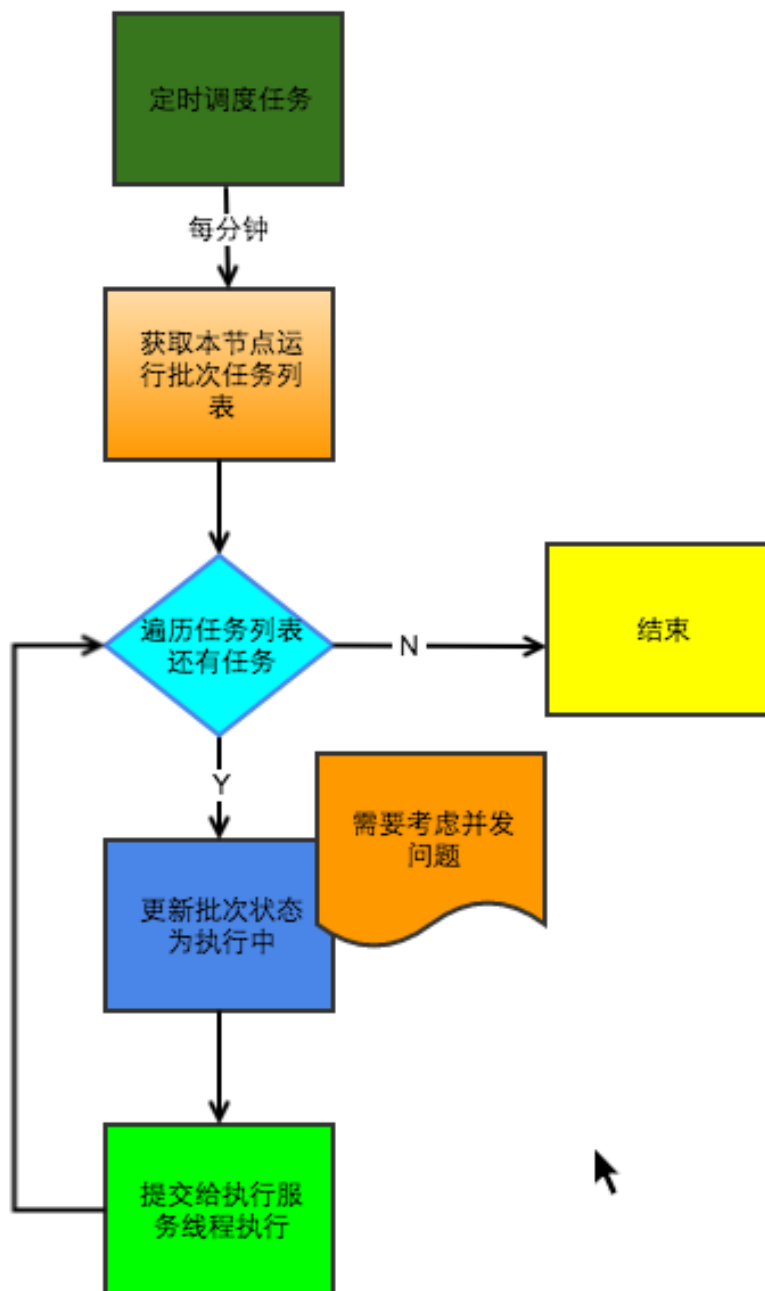
从数据库中找到本节点可以运行的任务，提交给执行服务模块执行。

本节点可以运行的任务由master的调度服务和再调度服务分配。

2.1.2 性能要求

2.1.3 程序实现

流程图：



2.1.4涉及的数据表字段修改

2.1.5涉及的任务状态修改

2.2任务执行

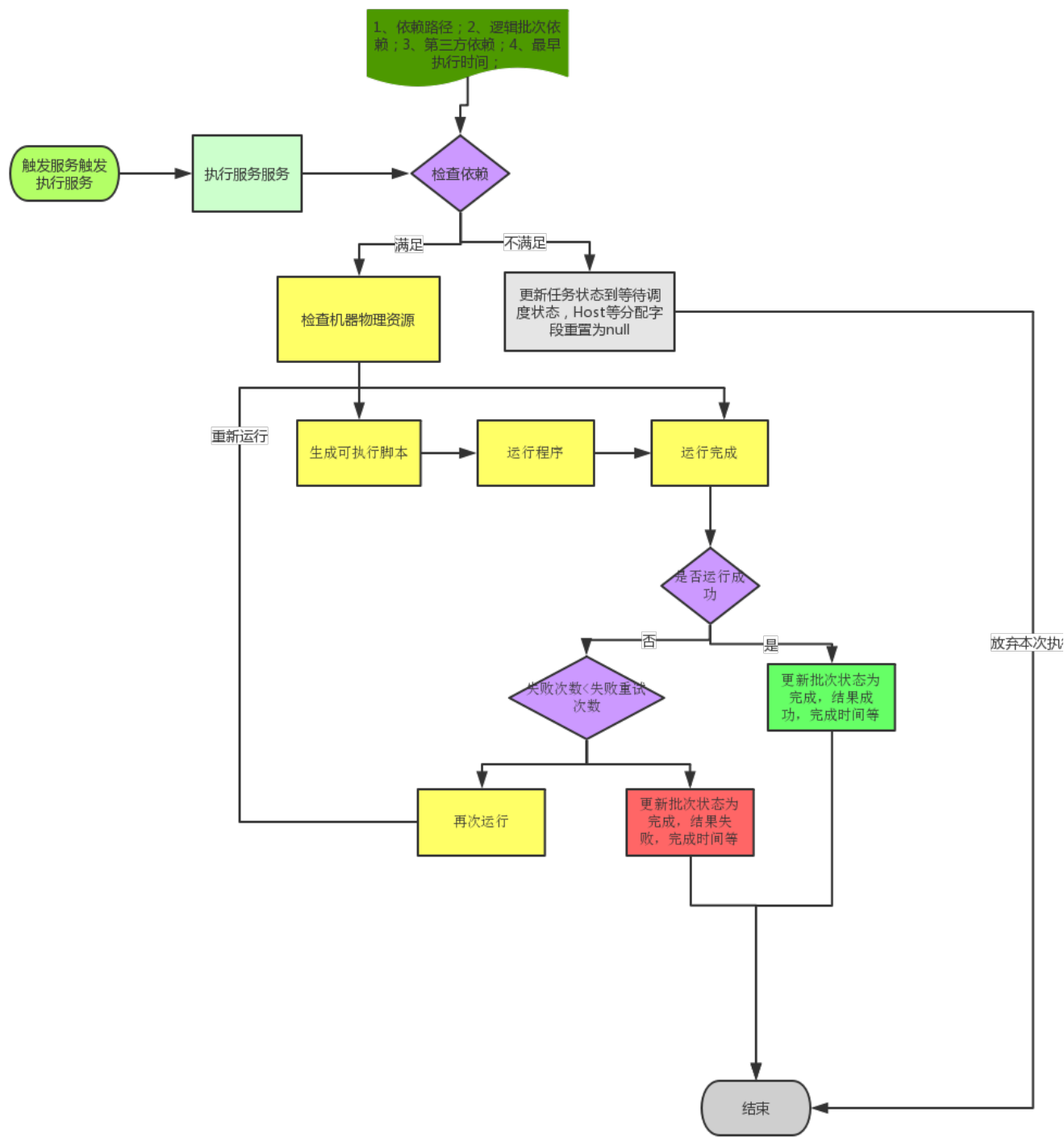
2.2.1功能描述

执行任务

2.2.2 性能要求

2.2.3 程序实现

流程图：



检查依赖失败要回滚到初始状态。回到分配任务时候的状态。

依赖分类



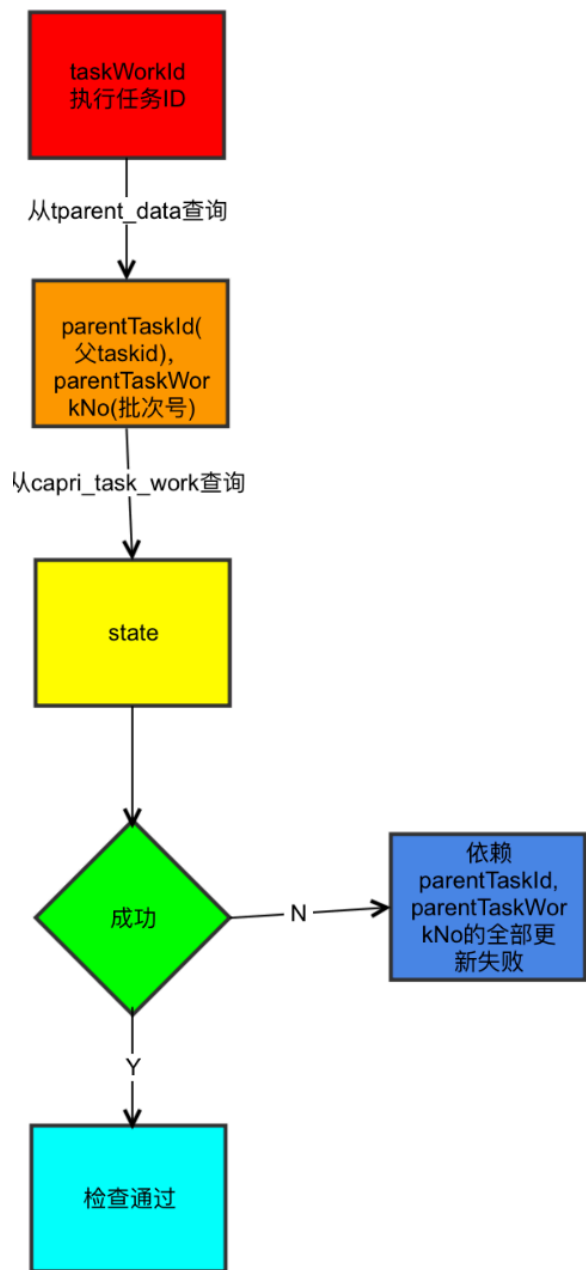
依赖检查思路

路径：检查依赖的路径是否有_SUCCESS文件。没有_success文件，更新task_data表字段为不可用，并且更新task_data表依赖的所有状态都为不可用and type =输入。task_data表。

逻辑：从tparent_data表中根据taskWorkId(批次号) 查出parentTaskId(父taskid)，parentTaskWorkNo(批次号)。

拿parentTaskId(taskid)，parentTaskWorkNo(批次号)去capri_task_work查询出type（表示批次号）最大的task_work，判断依赖的该任务是否成功。

如果成功则依赖检查通过，否则更新tparent_data的state的字段设置为失败并且更新所有依赖parentTaskId(父taskid)，parentTaskWorkNo(批次号)都为失败。



第三方

检查第三方依赖

依赖第三方通知我们

tthird_item存储结果

查询item表看结果，OK更新data表为成功，否则更新data表所有依赖这个code的第三方任务都为失败。

最早执行时间检查

由master控制 这里不控制了。

背景：

调度系统开始执行时间，月任务：每月1号凌晨0点05分。周任务：每周一凌晨0点02分。天：每天凌晨0点01分。小时：每个小时01分。这些称为基准时间。

task表相关字段

字段	含义	备注
----	----	----

outputPeriod	int(11) NOT NULL COMMENT '产出周期（2：小时，3：天，4：月，5：周）’，	
taskBeginTime	int(11) DEFAULT '-1' COMMENT '任务最早执行时间分钟,配置规则与依赖超时产出超时一样’，	这个存的是分钟数目

判断最早执行时间的标准

- 天的任务：基准时间（调度时间当天凌晨0点01分）+ taskBeginTime 换算成当天时间是否小于当前时间。
- 月任务：基准时间（调度时间当月1号凌晨0点05分）+ taskBeginTime 换算成当月时间是否小于当前时间。
- 周任务：基准时间（调度时间当周一凌晨0点02分。）+ taskBeginTime 换算成当周时间是否小于当前时间。
- 小时任务：基准时间（调度时间所在小时01分）+ taskBeginTime 换算成当时时间是否小于当前时间。

tasktime存入的时候要转化。比如在天任务 10:00执行，转换称9:59*60=559分钟。

检查物理资源

- 检查当前机器物理内存是否能满足欲执行任务的内存需求，load机器负载等。
- 如果失败则放弃执行本次任务。 更新状态（回滚，回到分配任务时候的状态）。

执行服务，更新状态，重试机制

- 生成可执行脚本，运行程序，重试机制和二期差不多，可以拿来复用。
- app.sh 加返回状态，加一个表示状态的从表。（状态码执行服务程序
- app.sh加上一条调用我们api返回状态码的代码。用来检查用户脚本执行结果）。

task 表 check_type字段判断 是否是依赖用户code
0表示成功
非0表示失败

- shell 建立分支（依赖用户code）专门用来调用controller层接口记录用户code返回情况(insert操作)。
- 这个只有在runjar里运行。worker里运行的化，worker杀了的化，这段逻辑就没了。

数据库表capri_user_code_log

名字	类型	not null	含义	
id	bigint(20)	y	主键	
task_instance_code	varchar(400)	y	实例code	
user_code	bigint(20)	y	用户自己返回code值：0为成功，非0失败	task_instance_code和code为0 具有唯一性
exexecute_state	bigint(20)	y	执行服务状态：0:完成，1.未完成	这个用来给僵死服务判断是否需要僵死修复
create_time	TIMESTAMP	y, default CURRENT_TIMESTAMP	创建时间	

- 只有一个成功
- 加一个状态表示执行服务是否执行完毕

2.2.4涉及的数据表字段修改

2.2.5涉及的任务状态修改

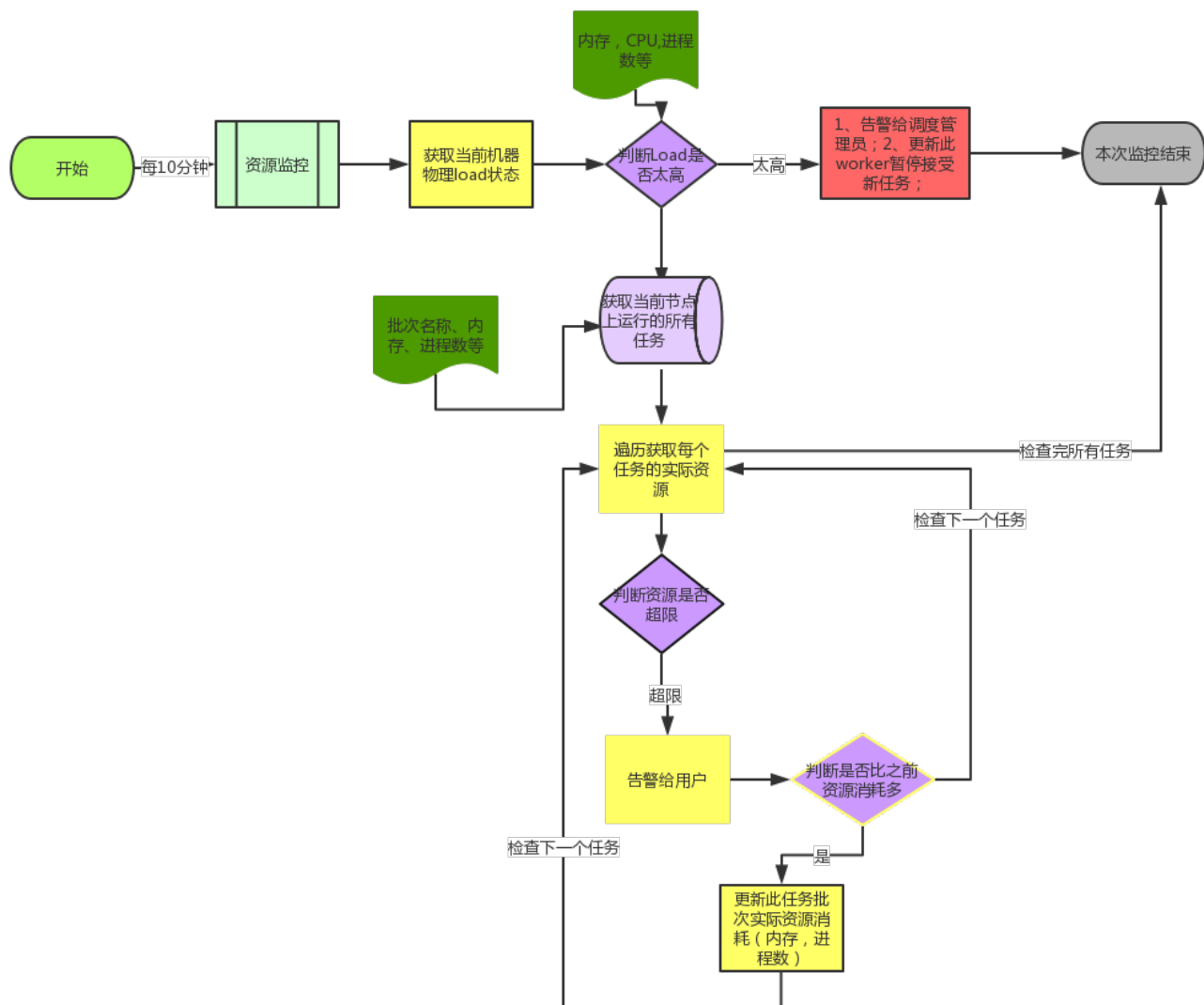
2.3资源监控

2.3.1功能描述

2.3.2 性能要求

2.3.3 程序实现

流程图：



实现思路

- shell脚本
- 告警接口使用封装的告警http接口。
- 更新capri_task_work表实际资源消耗字段。

告警给管理员的级别为严重级别。

2.3.4涉及的数据表字段修改

2.3.5涉及的任务状态修改

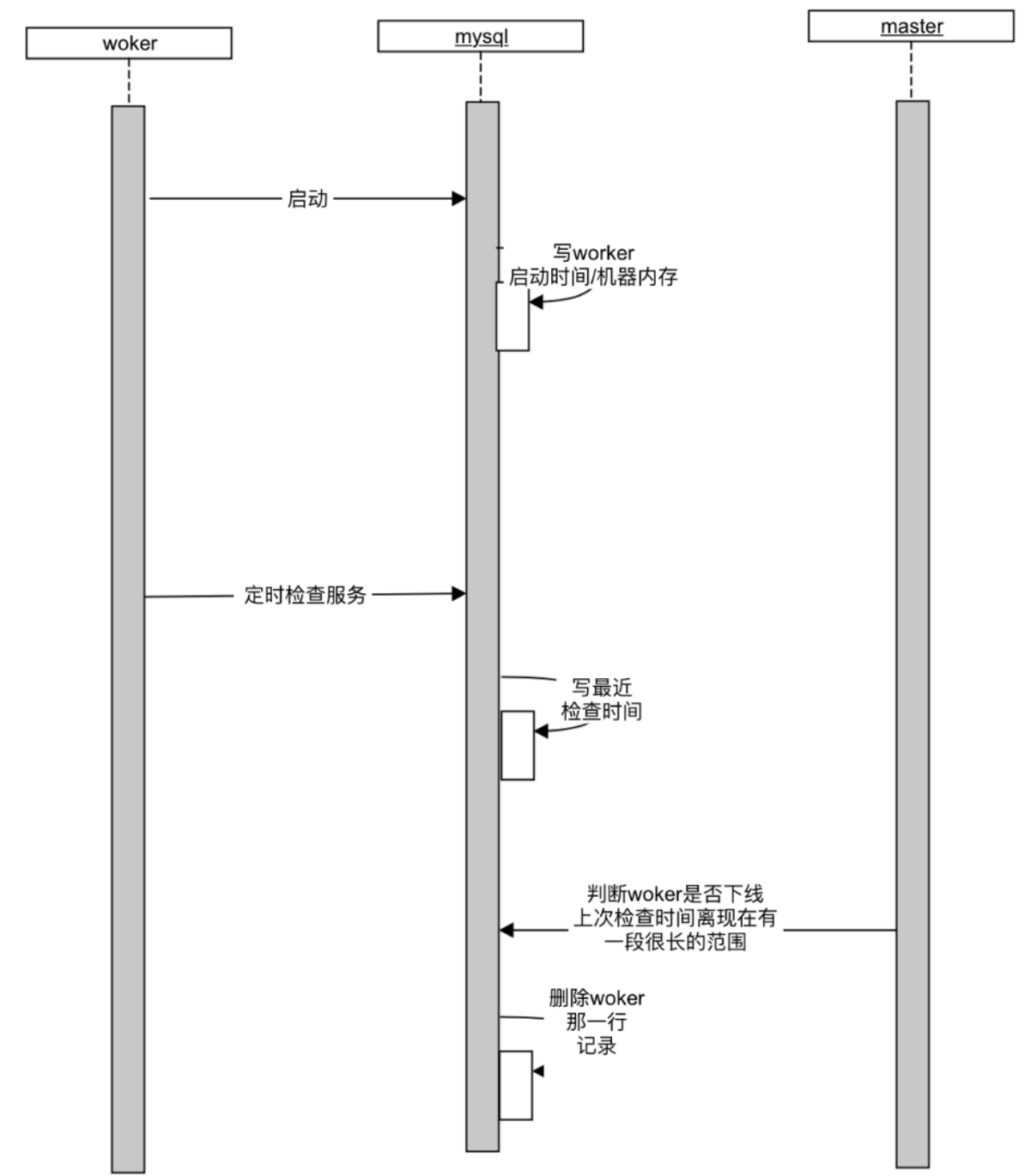
2.4汇报心跳

2.4.1功能描述

2.4.2 性能要求

2.4.3 程序实现

流程图：



数据库表capri_schedul_cluster。

字段	含义	备注
host	主机	XXXXX01. xg
lastCheckTime	上次检查时间	由汇报心跳服务不断刷新
checkInterval	检查间隔	定时检查时间小于checkinterval
isStop	是否不接受服务	由接收新任务开关控制
stopTime	停服时间	由接收新任务开关控制
starttrtime	服务启动时间	
memory	启动时机器总内存	

一大于才更新task_instance的实际内存 / 实际进程。

2. 4. 4涉及的数据表字段修改

2. 4. 5涉及的任务状态修改

2. 5接收新任务开关

2. 5. 1功能描述

2. 5. 2 性能要求

2. 5. 3 程序实现

- 1. 修改数据表capri_schedul_cluster将isStop修改为停止(1)，设置stoptime为当前时间。提供shell脚本修改数据库。
- 2. 在master分支上controller层添加一个接口。

2. 5. 4涉及的数据表字段修改

2. 5. 5涉及的任务状态修改