saturn 架构文档

1.引言

1.1 背景

Saturn (定时任务调度系统)是唯品会自主研发的分布式的定时任务的调度平台，目标是取代传统的Linux Cron/Spring Batch Job/Quartz的方式，做到全域统一配置，统一监控，任务高可用以及分片。

1.2 术语定义

术语	解释
作业(Job)和作业分片	作业(Job)是可以独立运行的脚本(shell作业)或者具备某项功能的函数实现(java)。作业可并发执行在多个执行节点(Executo
Namespace	域（saturn称之为Namespace)代表一组特定的执行结点和作业。作业必须而且只能属于某一个特定的域。一个域下通常有
组织名	每个namespace可以属于一个组织
Saturn	定时任务调度系统
执行结点(Executor)	执行结点(Executor)是调用并执行作业的程序。它通过定时(quartz)驱动来触发调用事件，并最终调用作业的执行入口(shel
控制台(Console)	统一配置界面，可以使用控制台来查看作业状态，执行结点状态和执行日志，添加、删除作业，修改作业属性。

1.3 运行环境

- Linux(shell作业仅支持linux,java/msg作业支持linux和windows)
- Java 1.7以上(不支持JDK1.6及更低版本)

2 总体设计

2.1 需求说明

编号	说明
1	支持多种作业类型(Shell作业/Java作业)
2	支持作业HA，负载均衡和失败转移
3	支持弹性动态扩容
4	支持Job Timeout处理
5	支持统一监控和告警
6	支持作业统一配置
7	支持资源隔离和作业隔离

2.2 总体设计说明

Saturn的基本原理是将作业在逻辑上划分为若干个作业分片，通过作业分片调度器将作业分片指派给特定的执行结点。
执行结点通过quartz触发执行作业的具体实现（以shell为例，则为shell脚本），在执行的时候，会将分片序号和参数作为参数传入(见图1)。
作业的实现逻辑需分析分片序号和分片参数，并以此为依据来调用具体的实现（比如一个批量处理数据库的作业，可以划分0号分片处理1-10号数据库，1号分片处理11-20号数据库）。

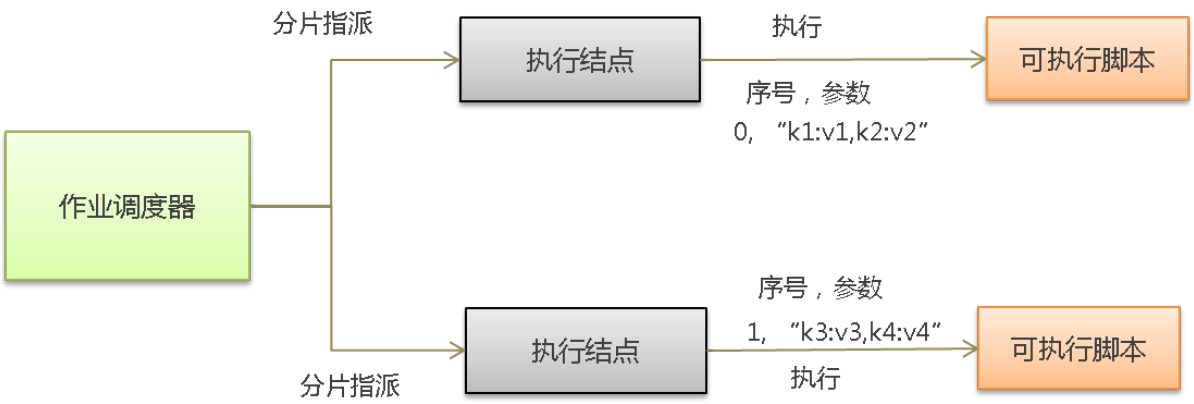


图1 基本原理图

2.3 总体架构设计

2.3.1 系统逻辑架构图

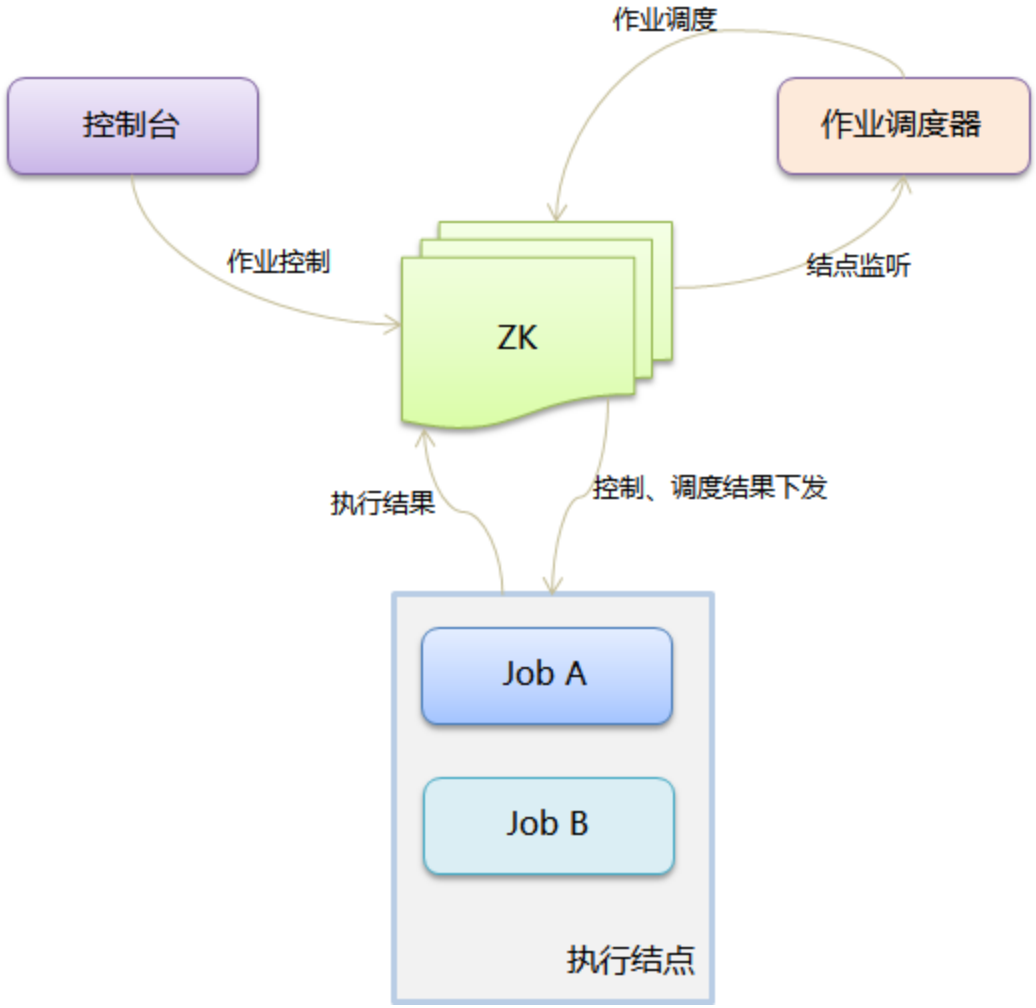


图2 系统逻辑架构图

执行结点

负责作业的触发（定时），作业执行，结果上报，日志上报，告警上报，监控日志写入等功能。可独立运行在业务服务器，也可与业务代码运行在同一个JVM。 使用java开发，提供jar包和可运行的工程两种方式供业务方使用，是业务作业接入saturn最主要的组件。

控制台

负责作业的统一配置，包括作业添加、删除，作业属性配置，作业状态查看，执行日志查看，执行结点监控等功能。
控制台单独部署，提供WEB应用给全域共用，业务接入方根据申请的权限控制对应的业务作业。

作业分片调度器

Saturn的"大脑"，其基本功能是将作业分片指派到执行结点。通过调整分配算法和分配策略，可以将作业合理地安排到合适的执行结点，从而实现HA，负载均衡，动态扩容，作业隔离，资源隔离等治理功能。
作业分片调度器为后台程序，单独部署；它是公共资源，所有域共用同一套作业分片调度器。 接入作业后，会自动接受作业分片调度器的调度。

3 子系统设计

3.1 子系统划分

序号子系统	名称	功能描述
1	saturn-core	公共模块，定义公共类，方法及实现
2	saturn-console-core	saturn控制台公共模块
3	saturn-console	saturn控制台
4	saturn-console-api	saturn控制台业务处理
5	saturn-console-web	页面工程
6	saturn-executor	saturn执行结点
7	saturn-job-sharding	saturn作业分片调度器
8	saturn-job-embed	嵌入方式使用saturn的转换模块（把saturn嵌入其它系统中运行，比如tomcat)
9	saturn-plugin	saturn maven插件
8	saturn-it	saturn 集成测试

3.2 Saturn 执行结点

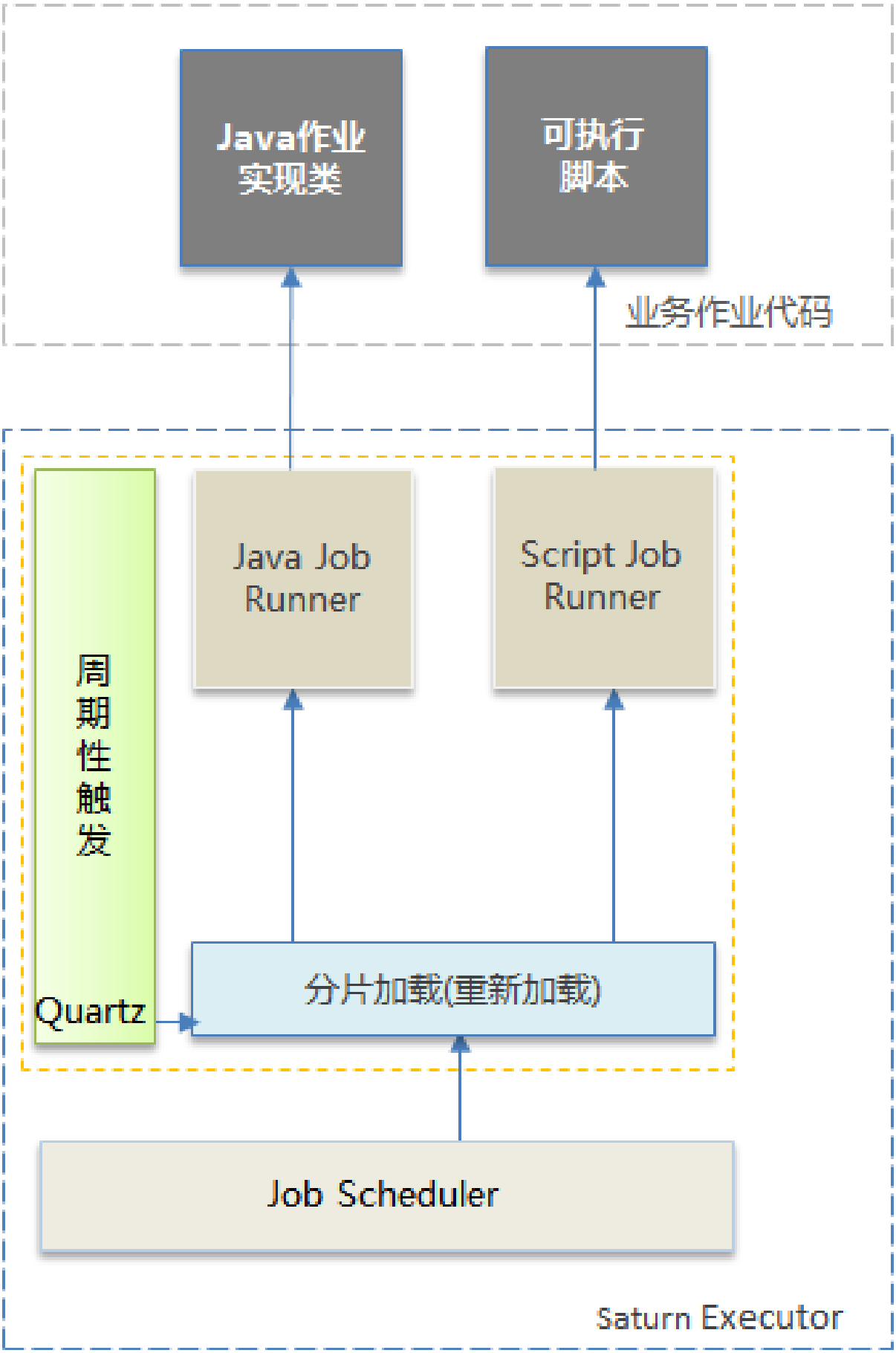

Java作业
实现类
可执行
脚本
业务作业代码
Java Job
Runner
Script Job
Runner
周
期
性
触
发
Quartz
分片加载(重新加载)
Job Scheduler
Saturn Executor

图3 执行结点实现逻辑

Job Scheduler

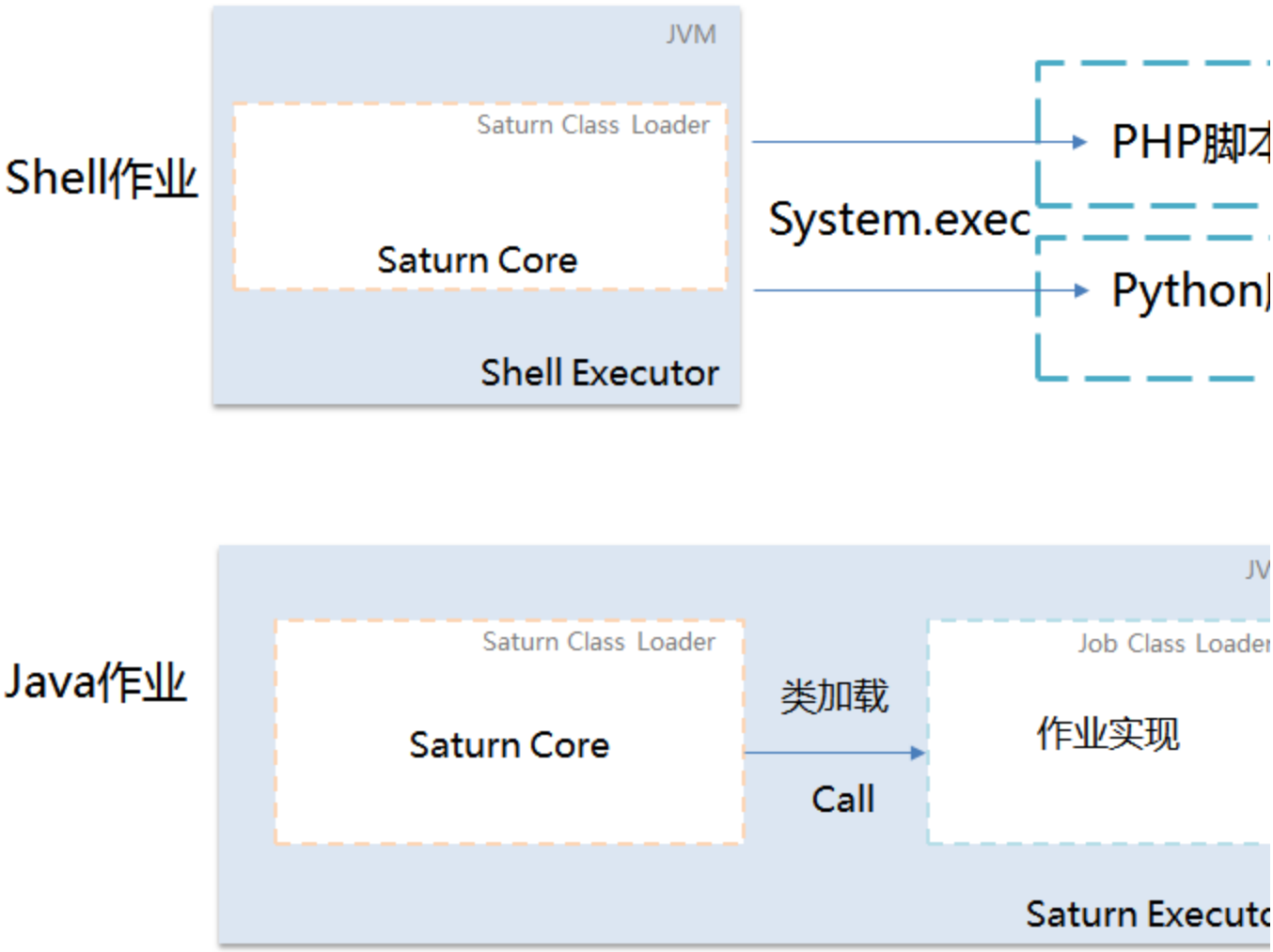
作业处理基础类，负责从ZK中读取作业配置信息（比如作业类型【shell作业/java作业/msg作业】，作业cron表达式，分片参数等），根据作业类型启动不同的处理逻辑。
shell作业和java作业使用cron表达式开启quartz
scheduler，并周期性触发；msg作业启动分片监听，并根据获取到的分片启动消息订阅。

分片加载（重新加载)

检查有没有收到作业分片调度器发出的重新加载指令(通过ZK结点)，如果有，则全部执行结点都须等待全部分片执行完成，然后由其中一个执行结点从任务调度器的调结果中读取本作业的分片指派结果， 并将结果广播（使用ZK的监听机制）给本作业的全部执行结点。

Shell/Java作业触发

Shell作业和Java作业根据cron表达式定义的时间规则进行周期性触发，每次触发周期到达时，先进行分片加载（重新加载），之后根据作业类型调用Java Job Runner或Script Job Runner。
Job Runner取出分片序列，解析分片参数（分片参数可通过控制台配置)；
如果是java作业，启动线程池，将分片序列和分片参数作为调用参数构造作业执行线程，将作业执行线程提交到线程池执行；
如果是script作业，将分片序列和分片参数作为调用参数exec调用可执行脚本，并获取输出流。
Job
Runner启动作业之后，如果作业配置了超时时间，则需进行超时检测，一旦检测到超时，需要做超时处理（作业中止，状态上报）。



3.3 Saturn 作业分片调度器

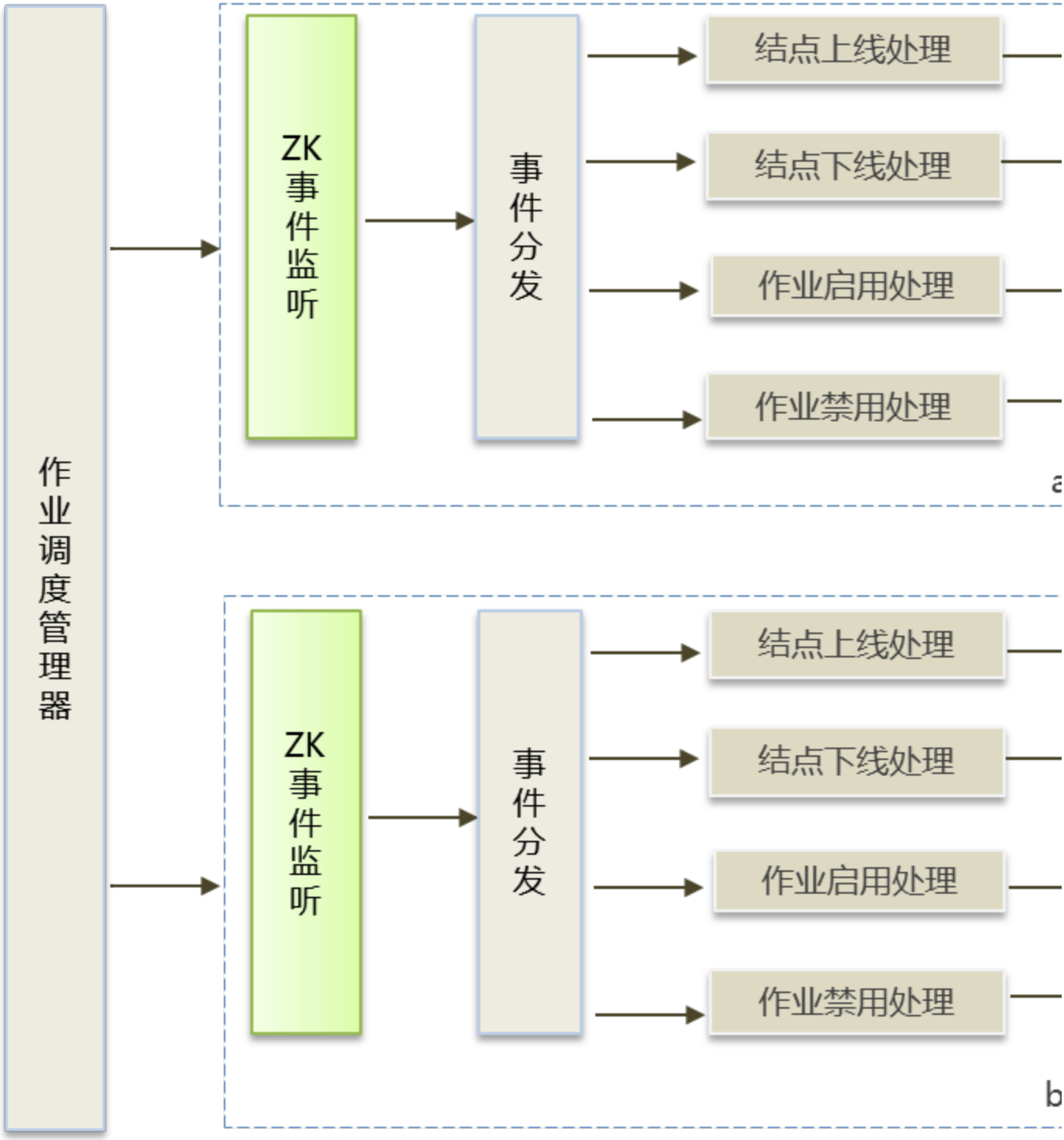


图4 Saturn作业分片调度器实现逻辑

作业分片调度管理器

遍历全部域，为每个域都生成域作业分片调度器。域作业分片调度器的处理步骤为：ZK事件监听，事件分发和处理，结果下发。

ZK事件监听

执行结点上线后会在ZK注册一个临时结点，下线后ZK会话超时临时结点会被清除。
作业启用和禁用状态为持久化结点，由管理员通过控制台修改（从启用改为禁用，或者从禁用改为启用）。
Saturn作业分片调度器监听这些结点的变化，根据结点路径和结点的值判断出事件类型进行事件分发。

事件分发

Saturn作业分片调度器处理以下4类事件：结点上线，结点下线，作业启用，作业禁用。
分别对应结点上线处理类，结点下线处理类，作业启用处理类，作业禁用处理类四个模块。
事件处理类的作用是进行作业分片调度，将作业分片指派给执行结点。并将指派结果保存到ZK。

结果下发

将重新加载指令写入到域下全部作业的指定的ZK结点。

4 功能设计

4.1 作业分片调度算法设计

作业分片调度算法参见《Saturn作业调度算法》

4.2 需求设计

多种作业类型(Java作业/Shell作业)

Saturn
executor使用java语言开发，Java作业天然支持，接入作业只需实现特定的作业执行方法即可接受调度； 对于Shell作业，使用apache common
exec调用Shell执行脚本，并获取输出流，使用timeout系统命令在超时后将脚本进程退出。

动态扩容，HA，负载均衡

Saturn通过作业分片调度器和作业分片调度算法在资源改变时控制作业分片的指派；
资源改变包括：执行结点增加（扩容），执行结点减少（减容），作业上线（启用）
执行结点上线（增加、扩容）时，作业分片调度器会根据平衡算法将现有执行结点上的部分作业分片迁移到新增加的执行结点，作业上线（启用）时，作业分片调度器会根据平衡算法将新上线作业的全部分片分配给域下的执行结点，从而实现动态扩容和负载均衡。
执行结点下线（减少、减容）时，作业调度器会根据平衡算法将被减容（下线）的执行结点的全部作业分片迁移到其它存活的执行结点，从而实现HA；

失败转移

执行结点下线时，如果有作业正在执行且尚未结束，则正在执行的作业分片会被作业调度器指定到其它执行结点并且尝试立刻执行，如果无法立刻执行，则会等到当前执行结点分配到的本作业的其它分片执行完毕后立刻执行。
注意：saturn的失败转移机制并不保证分片迁移到新的执行结点后立刻开始执行，它需等待当前执行结点分配到的本作业的其它分片执行完毕后才会立刻执行。如果作业执行周期较长，就会存在一段相当长的时间内，此分片没有分配到执行结点。

Job Timeout处理

设置了timeout的作业开始执行后会启动超时检测，如果执行超时，则会停止当前作业的执行（JAVA作业停止线程，Script作业停止进程），并将超时事件上报。

资源隔离

有时候需要对执行结点进行功能划分，比如1，2号执行结点处理作业A，B；3，4号作业处理作业C,D；
设置作业的优先列表，当优先列表结点存活时，作业的分片只会指派给这些结点；当优先结点全部下线时，作业的分片才会指派给其它结点。

作业隔离

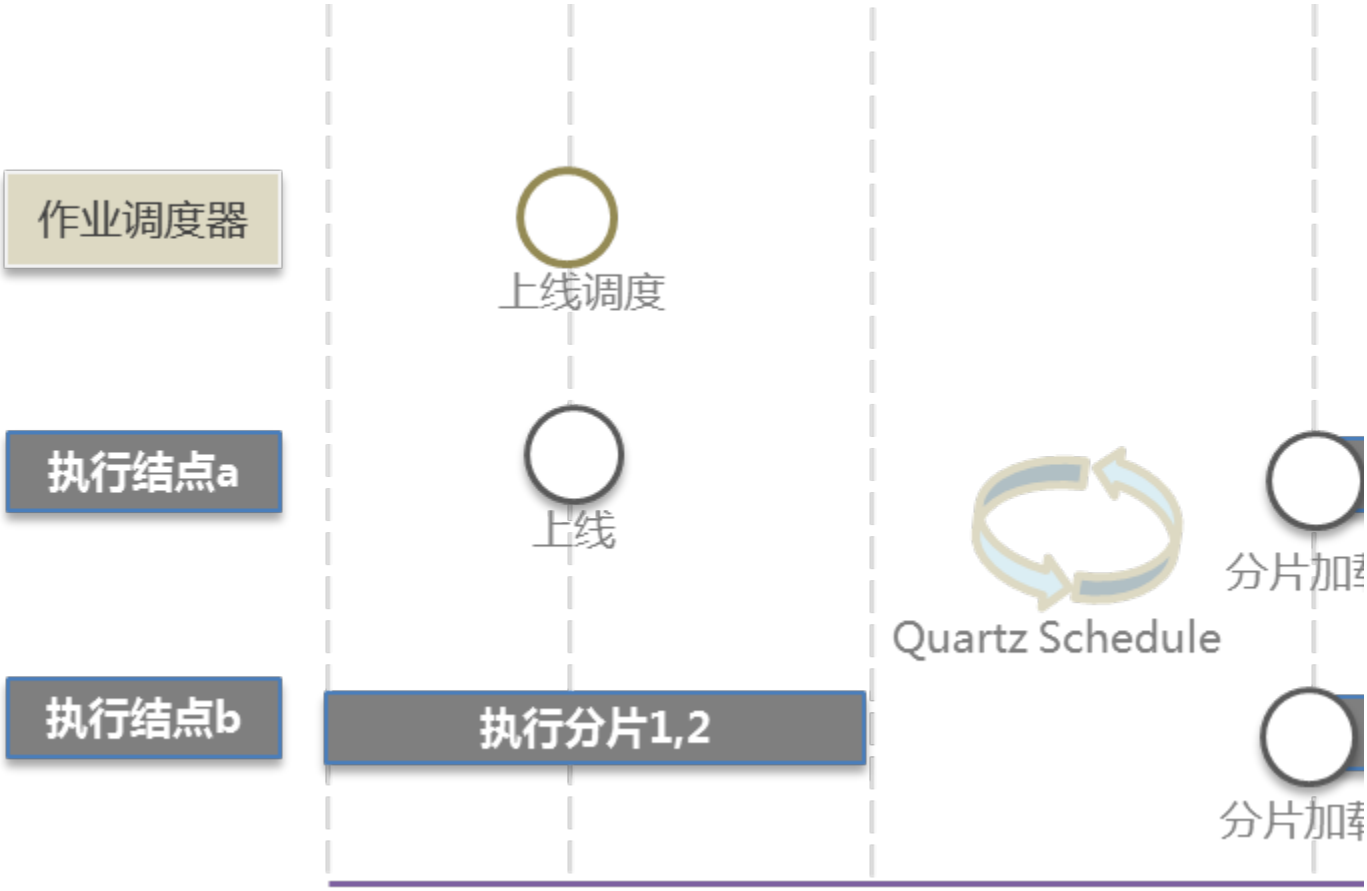
某些作业不希望与其它作业共用同一个执行结点，与其它作业隔离。
将需要隔离的作业的负载值设为远超其它作业的值（比如本作业负载设置为999，其它作业负载1），作业调度器在进行作业调度时会尽量保证执行结点的负载均衡。
负载值大的分片优先分配，而且一旦分配到某个执行结点，将会导致该执行结点的负载值远超其它结点，从而保证不会有其他作业被分配到该执行结点，于是可以达到作业隔离的目的。

统一配置

Saturn通过统一的控制台进行全部域及全部作业的配置，执行情况监控，结点监控等。

4.3 关键流程

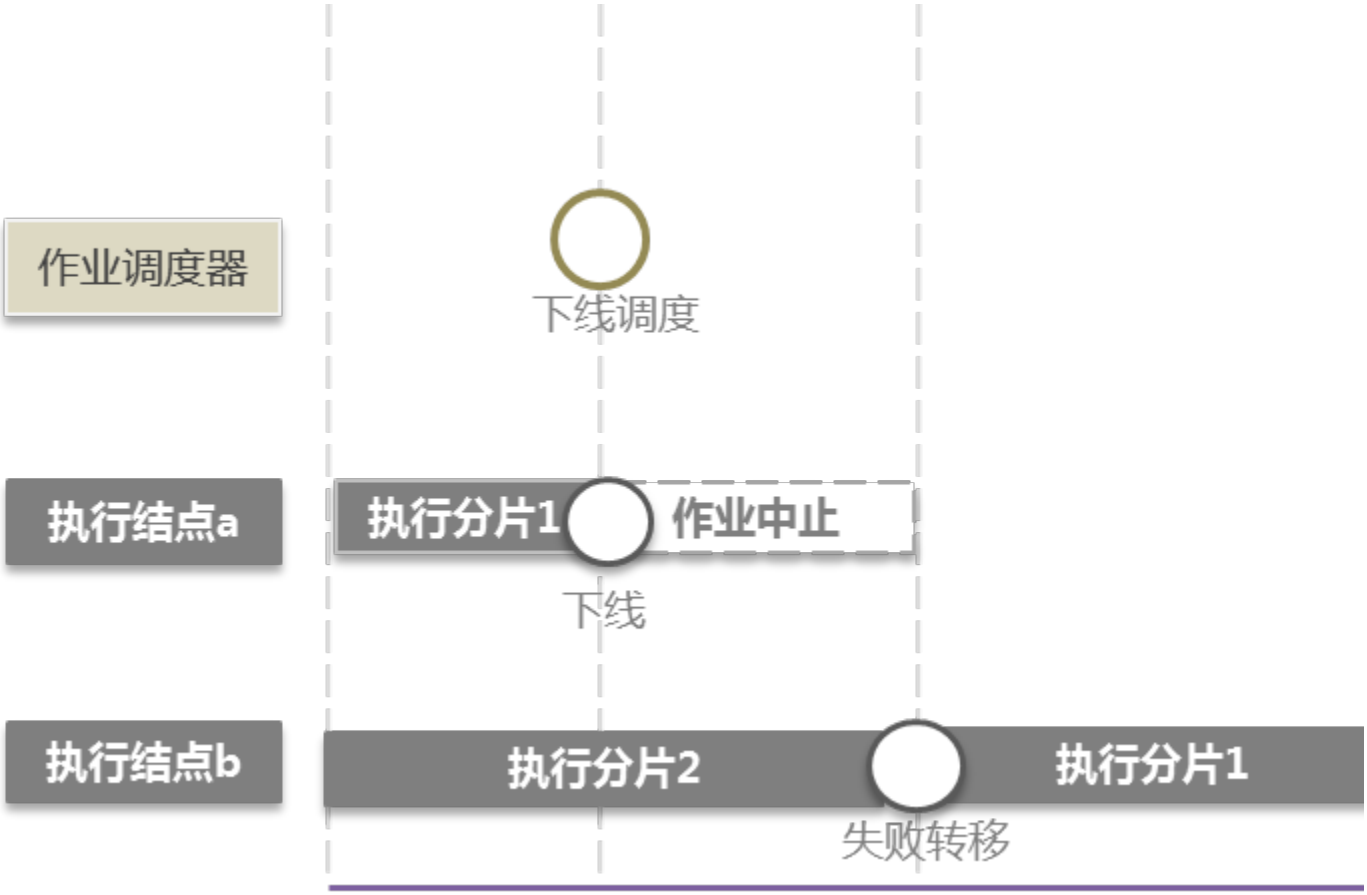
执行结点上线流程



步骤说明：

名称	说明
上线	执行结点启动，往ZK临时结点/a.vip.vip.com/$SaturnExecutors/executors/excutor_001/ip写入本机IP 。
上线调度	作业分片调度器监听ZK路径/a.vip.vip.com/$SaturnExecutors/executors/，当路径为/ip并且事件为ADD，则判断出有结点新上线；作
分片加载	执行结点从/a.vip.vip.com/$Jobs/job_01(这里为jobName)/leader/sharding/necessary结点读取重新加载指令，如果大于0，则重新

执行结点下线流程



步骤说明：

名称	说明
下线	执行结点关闭，临时结点/a.vip.vip.com/$SaturnExecutors/executors/excutor_001/ip自动清除；
下线调度	作业分片调度器监听ZK路径/a.vip.vip.com/$SaturnExecutors/executors/，当路径为/ip并且事件为REMOVE，则判断出有结点新下线
失败转移	执行结点监听/a.vip.vip.com/$Jobs/job_01/execution/0/running结点，当事件为REMOVE，表示有分片执行中止，而且尚未完成执行
分片加载	执行结点从/a.vip.vip.com/$Jobs/job_01(这里为jobName)/leader/sharding/necessary结点读取重新加载指令，如果大于0，则重新

4.4 作业状态


Ready
启用
触发（手工/
停用
执行完成
Stopped
Running
执行完成
停用
Stopping