分布式定时任务

为什么需要?

- 复杂的数据处理需要耗费大量的时间，用户体验差
- 同样的逻辑多次执行，浪费不必要的性能

主要内容：

- Elastic-Job-Lite
- Elastic-Job-Cloud
- LTS

简介

- Elastic-Job是一个分布式调度解决方案，由两个相互独立的子项目Elastic-Job-Lite和Elastic-Job-Cloud组成。

- Elastic-Job-Lite定位为轻量级无中心化解决方案，使用jar包的形式提供分布式任务的协调服务。 Elastic-Job-Cloud使用Mesos + Docker的解决方案，额外提供资源治理、应用分发以及进程隔离等服务。

- Elastic-Job-Lite和Elastic-Job-Cloud提供同一套API开发作业，开发者仅需一次开发，然后可根据需要以Lite或Cloud的方式部署。

- https://github.com/dangdangdotcom/elastic-job

之前

- 使用Quartz

- 缺点：

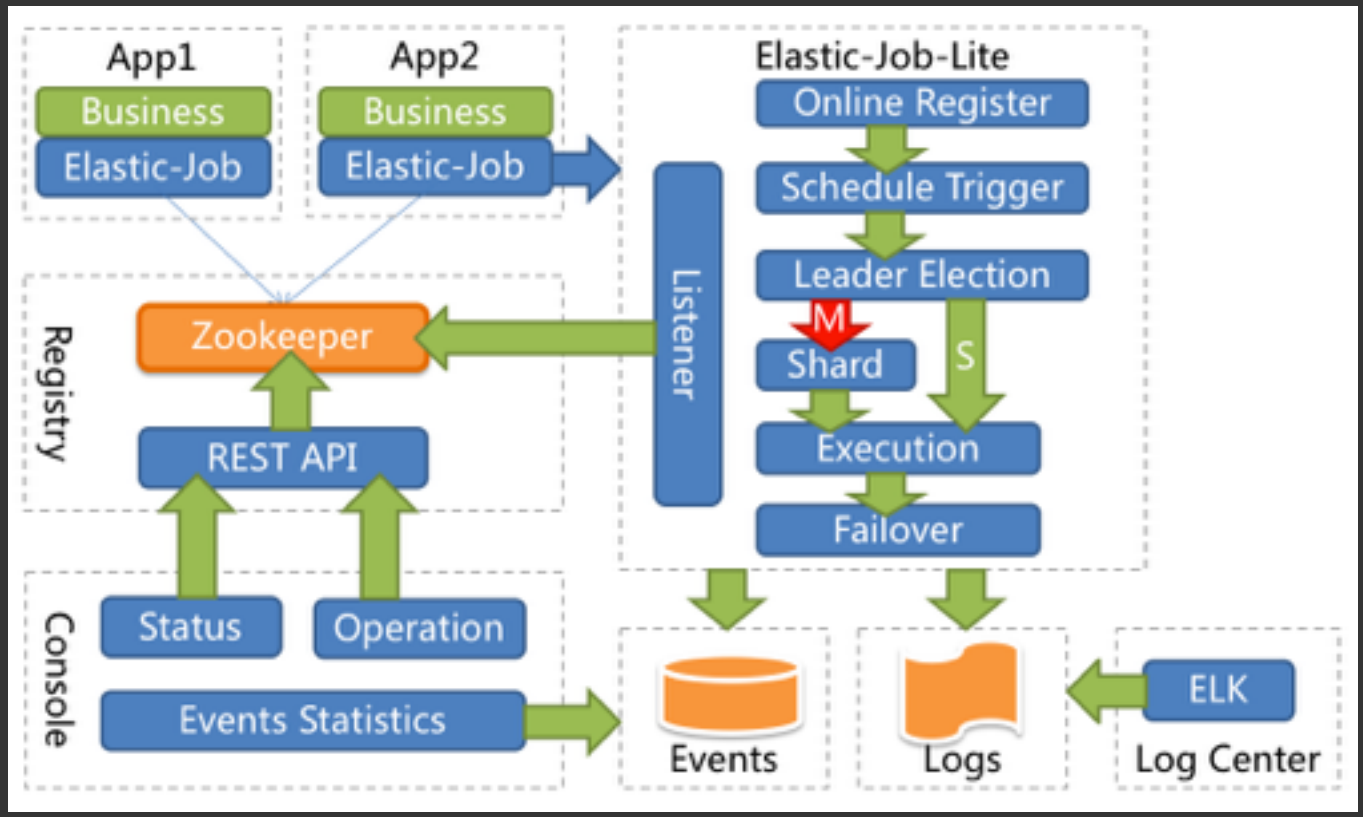
  缺少管理查看的平台

  实现分布式麻烦且对性能影响较大

  并发时，容易死锁

  任务失败后的处理

原理


App1
Business
Elastic-Job
App2
Business
Elastic-Job
Elastic-Job-Lite
Online Register
Schedule Trigger
Leader Election
M
S
Shard
Execution
Failover
Listener
Registry
Zookeeper
REST API
Console
Status
Operation
Events Statistics
Events
Logs
ELK
Log Center

功能

- 分布式调度协调
- 弹性扩容缩容
- 失效转移
- 错过执行作业重触发
- 作业分片一致性，保证同一分片在分布式环境中仅一个执行实例
- 支持并行调度
- 支持作业声明周期操作
- 丰富的作业类型
- Spring整合以及命名空间提供
- 运维平台

作业类型

- Simple类型作业
- Dataflow类型作业
- Script类型作业

运维平台

- 添加注册中心
- 三种维度查看作业运行情况
- 修改作业配置
- 控制作业的暂停、恢复、删除
- 与作业本身没有之间关联

注册中心

- 采用zookeeper作为定时任务配置的持久化。
- 第一台服务器上线出发主服务器选举。主服务器下线重新出发选举。

分片策略

框架提供的分片策略

基于平均分配算法的分片策略，也是默认的分片策略。

根据作业名的哈希值奇偶数决定IP升降序算法的分片策略。

根据作业名的哈希值对服务器列表进行轮转的分片策略。

自定义分片策略

实现JobShardingStrategy接口并实现sharding方法，接口方法参数为作业服务器IP列表和分片策略选项，分片策略选项包括作业名称，分片总数以及分片序列号和个性化参数对照表，可以根据需求定制化自己的分片策略。

例子

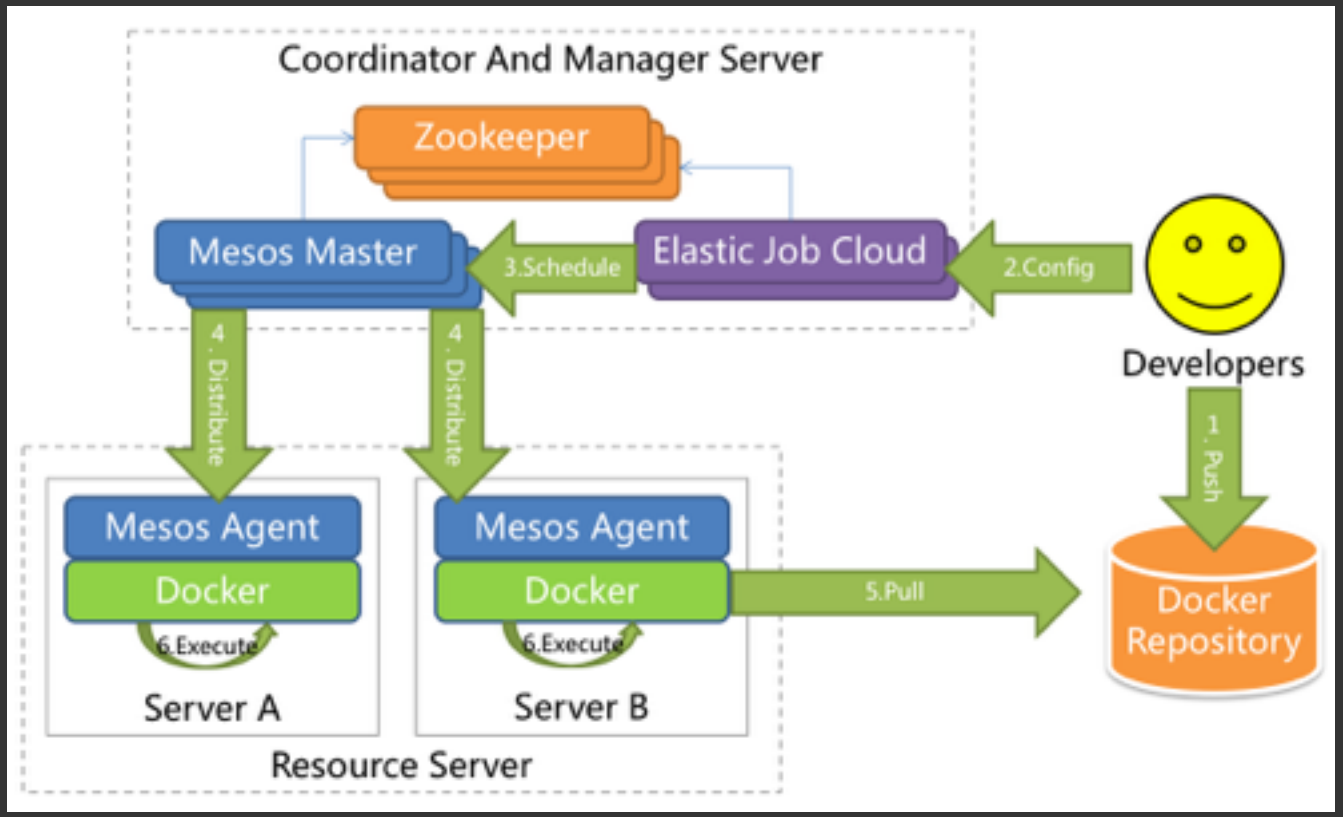
使用限制

- 作业名称无法修改
- 动态添加作业

Elastic-Job-Cloud

- Elastic-Job 2.0新增
- 包含Elastic-Job-Lite的全部功能
- 弹性资源分配(Mesos)
- 应用自动分发
- 基于Docker的进程隔离(TBD)
- Maven部署插件

原理


Coordinator And Manager Server
Zookeeper
Mesos Master
3.Schedule
Elastic Job Cloud
2.Config
Developers
4. Distribute
4. Distribute
1. Push
Mesos Agent
Docker
6.Execute
Server A
Mesos Agent
Docker
6.Execute
Server B
5.Pull
Docker Repository
Resource Server

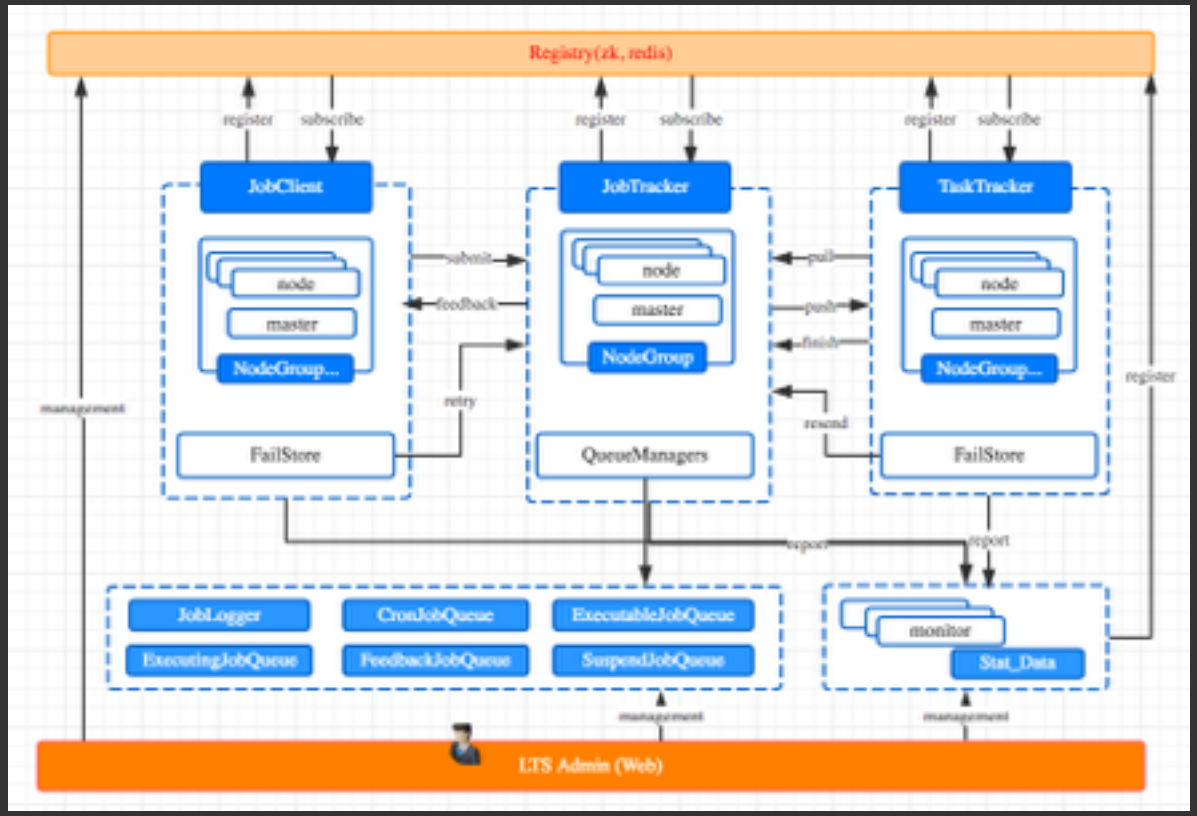
LTS

- LTS(light-task-scheduler)主要用于解决分布式任务调度问题，支持实时任务，定时任务，Cron任务，Repeat任务。

节点

- JobClient :主要负责提交任务, 并接收任务执行反馈结果。
- JobTracker :负责任务调度，接收并分配任务。
- TaskTracker :负责执行任务，执行完反馈给JobTracker。
- LTS-Monitor　:主要负责收集各个节点的监控信息，包括任务监控信息，节点JVM监控信息
- LTS-Admin　:管理后台）主要负责节点管理，任务队列管理，监控管理等。

原理


Registry(zk, redis)
register
subscribe
register
subscribe
register
subscribe
JobClient
JobTracker
TaskTracker
node
master
NodeGroup...
node
master
NodeGroup
node
master
NodeGroup...
submit
feedback
retry
pull
push
finish
resend
register
management
FailStore
QueueManagers
FailStore
report
report
JobLogger
CronJobQueue
ExecutableJobQueue
ExecutingJobQueue
FeedbackJobQueue
SuspendJobQueue
monitor
Stat_Data
management
management
LTS Admin (Web)

选择elastic-job 的理由

- 稳定，可靠
- 分片功能
- 更新速度比较快
- 可快速配置启动，降低新项目搭建框架的成本

谢谢