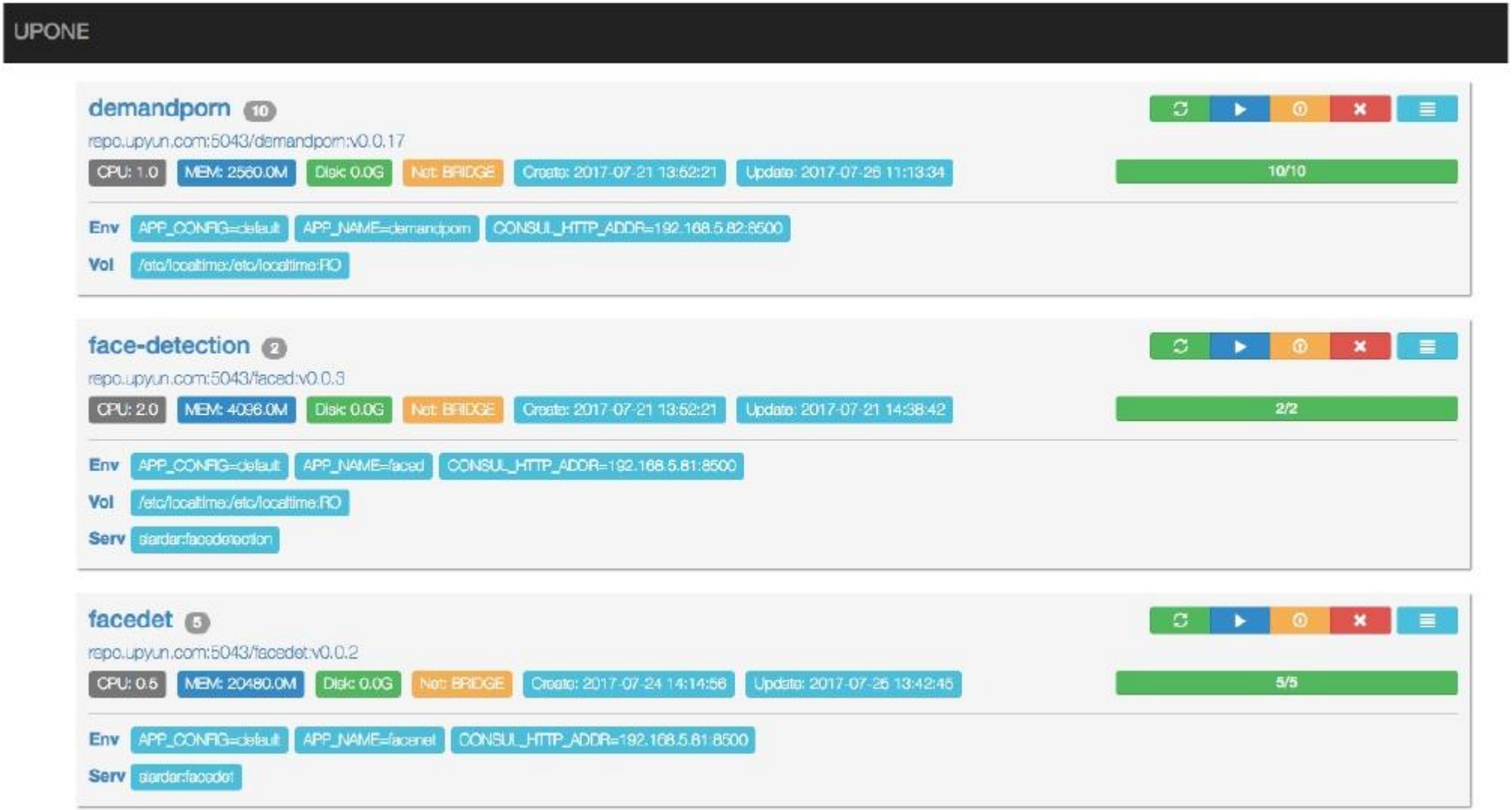
IT大咖说 知识分享平台

基于 mesos的容器调度框架

Gopher杭州 meetup
5 August 2017

黄励博(huangnauh)
又拍云

What's Upone


UPONE
demandporn 10
repo.upyun.com:5043/demandporn:v0.0.17
CPU: 1.0
MEM: 2560.0M
Disk: 0.0G
Net: BRIDGE
Create: 2017-07-21 13:52:21
Update: 2017-07-26 11:13:34
10/10
Env
APP_CONFIG=default
APP_NAME=demandporn
CONSUL_HTTP_ADDR=192.168.5.82:8500
Vol
/etc/localtime:/etc/localtime:RO
face-detection 2
repo.upyun.com:5043/faced:v0.0.3
CPU: 2.0
MEM: 4096.0M
Disk: 0.0G
Net: BRIDGE
Create: 2017-07-21 13:52:21
Update: 2017-07-21 14:38:42
2/2
Env
APP_CONFIG=default
APP_NAME=faced
CONSUL_HTTP_ADDR=192.168.5.81:8500
Vol
/etc/localtime:/etc/localtime:RO
Serv
qiardar:facedetection
facedet 5
repo.upyun.com:5043/facedet:v0.0.2
CPU: 0.5
MEM: 20480.0M
Disk: 0.0G
Net: BRIDGE
Create: 2017-07-24 14:14:56
Update: 2017-07-25 13:42:46
5/5
Env
APP_CONFIG=default
APP_NAME=facenet
CONSUL_HTTP_ADDR=192.168.5.81:8500
Serv
qiardar:facedet

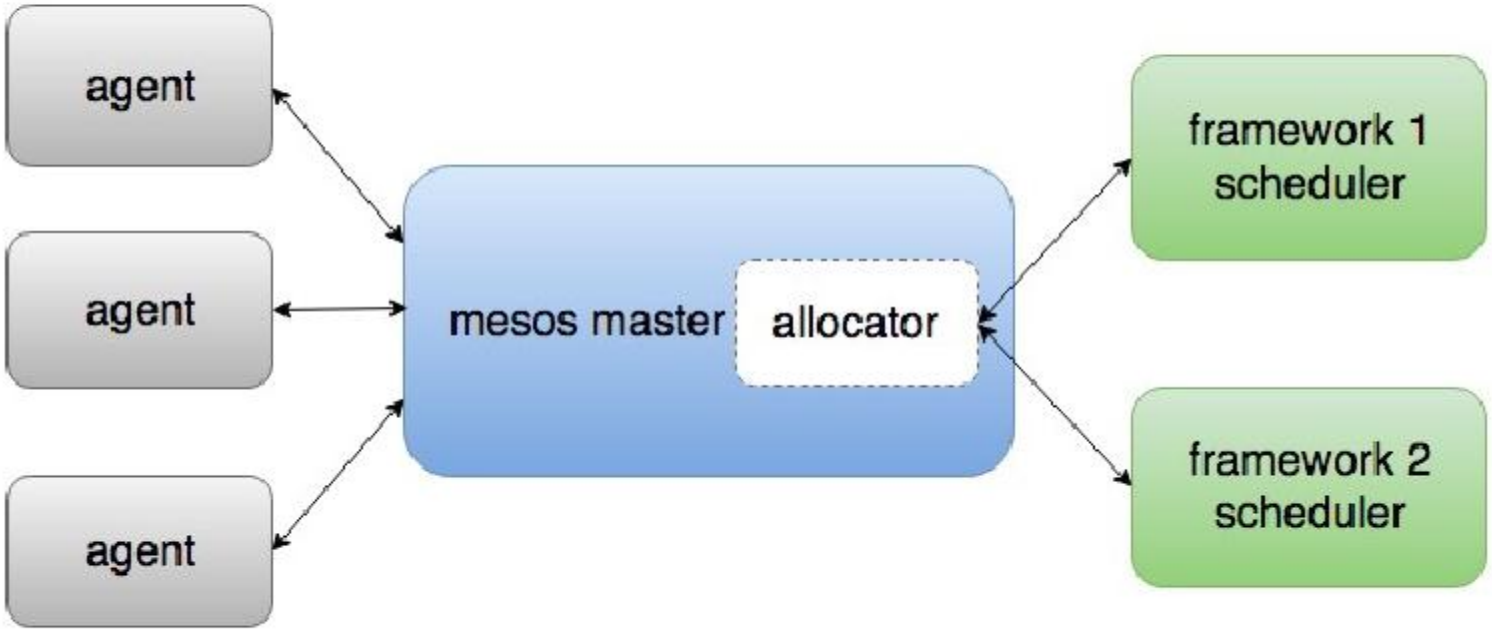
Mesos介绍

MESOS

Image credit: mesos.apache.org (http://mesos.apache.org/)

官方称之为分布式系统内核, 它把数据中心的 CPU、内存、磁盘等抽象成一个资源池

Mesos调度



各个 Agent启动后, 向 Master注册， 携带统计资源, 由 Master决定给每个框架多少资源, 默认采用分级主导资源公平算法

每个框架收到资源后, 根据自身任务需求, 调度任务的资源分配

upone

为云处理服务定制的容器调度框架，支持长期服务和定时任务

处理流程

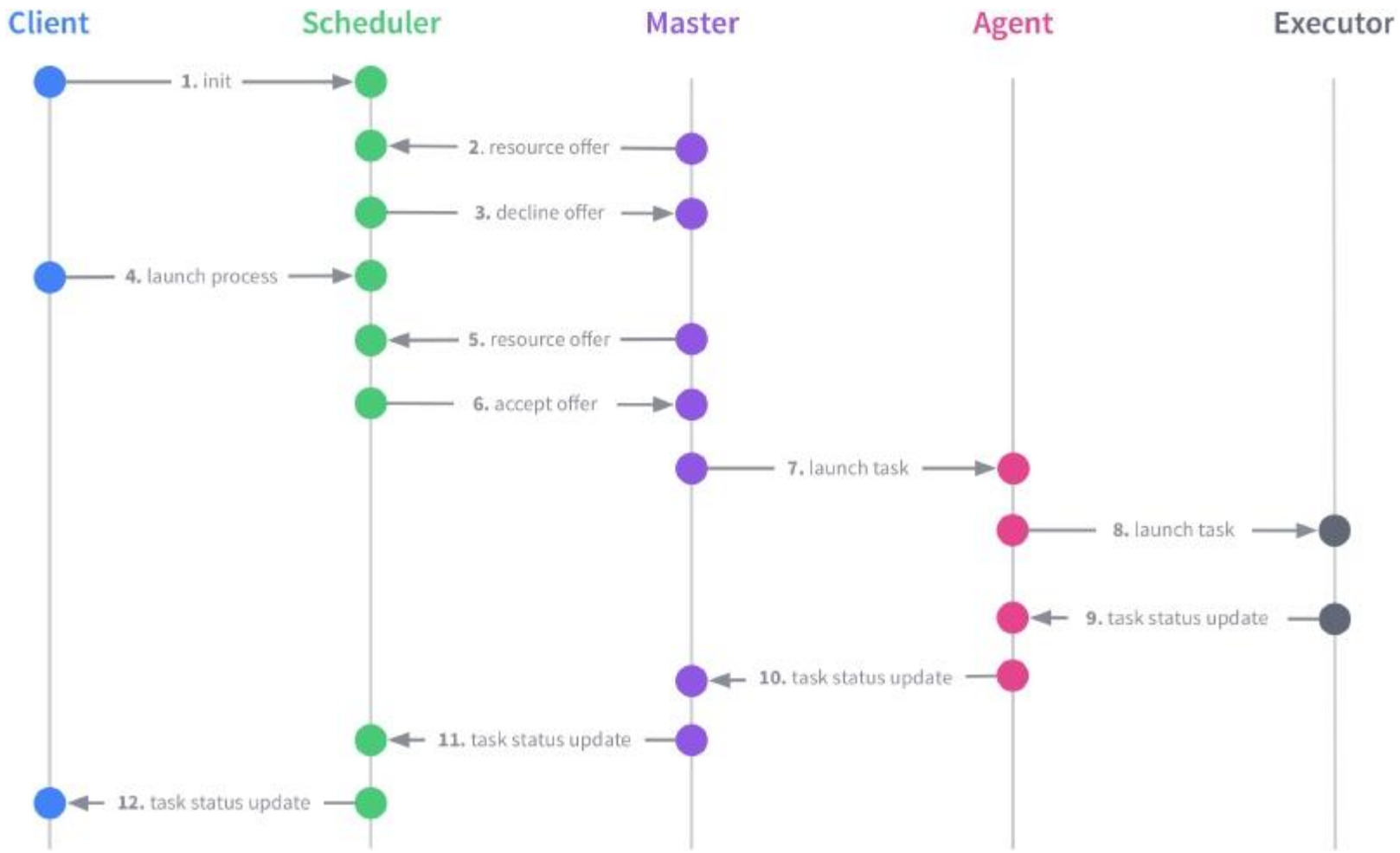
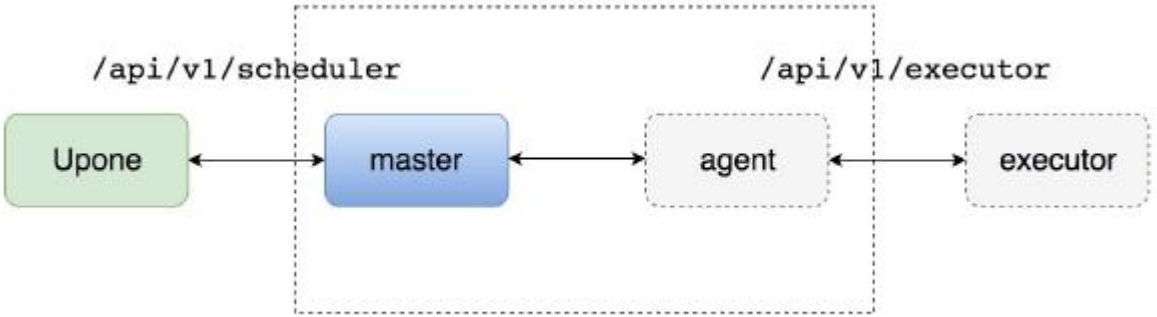


Image credit: [dcos architecture](https://dcos.io/docs/1.7/overview/architecture/)

消息交互

upone通过 /api/v1/scheduler(/api/v1/scheduler) 和 mesos master完成交互



订阅

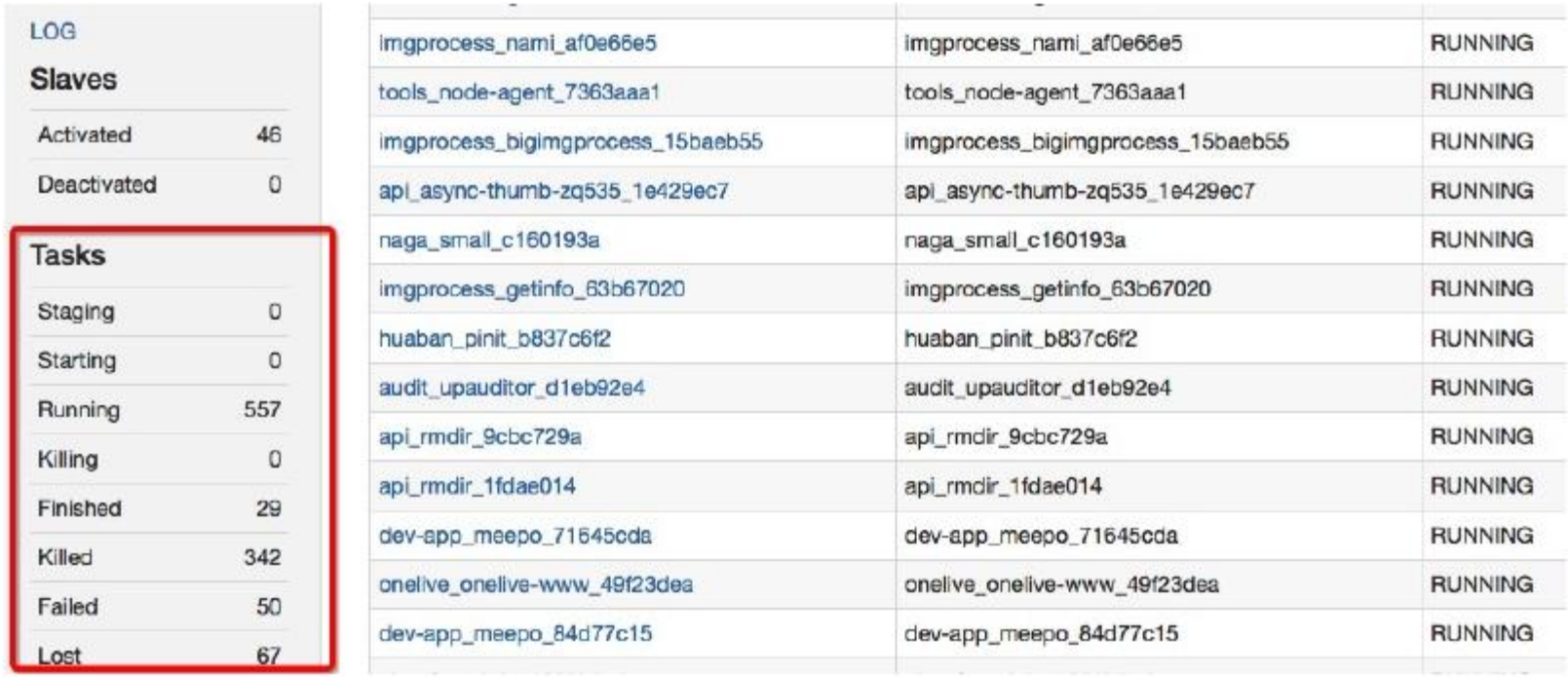
通过 SUBSCRIBE 消息, 开启一个长链接来订阅事件 (Event), 常用事件类型:

事件	简介
SUBSCRIBED	成功订阅
OFFERS	提供资源的信息
RESCIND	通知资源不再有效
UPDATE	任务状态的改变
FAILURE	执行器终止或者代理失去连接, 通常伴随着 UPDATE 等事件
HEATBEAT	心跳

OFFERS事件处理

- ACCEPT消息： 接受此 o er， 包括启动任务，预留资源等调度操作
- DECLINE消息： 明确拒绝此 o er
- Filter :可以设定资源的拒绝时间， 默认是 5s
- REVIVE消息： 取消之前设定 lter

UPDATE事件处理



LOG

Slaves

Activated	46
Deactivated	0

Tasks

Staging	0
Starting	0
Running	557
Killing	0
Finished	29
Killed	342
Failed	50
Lost	67

imgprocess_nami_af0e66e5	imgprocess_nami_af0e66e5	RUNNING
tools_node-agent_7363aaa1	tools_node-agent_7363aaa1	RUNNING
imgprocess_bigimgprocess_15baeb55	imgprocess_bigimgprocess_15baeb55	RUNNING
api_async-thumb-zq535_1e429ec7	api_async-thumb-zq535_1e429ec7	RUNNING
naga_small_c160193a	naga_small_c160193a	RUNNING
imgprocess_getinfo_63b67020	imgprocess_getinfo_63b67020	RUNNING
huaban_pinit_b837c6f2	huaban_pinit_b837c6f2	RUNNING
audit_upauditor_d1eb92e4	audit_upauditor_d1eb92e4	RUNNING
api_rmdir_9cbc729a	api_rmdir_9cbc729a	RUNNING
api_rmdir_1fdae014	api_rmdir_1fdae014	RUNNING
dev-app_meepo_71645cda	dev-app_meepo_71645cda	RUNNING
onelive_onelive-www_49f23dea	onelive_onelive-www_49f23dea	RUNNING
dev-app_meepo_84d77c15	dev-app_meepo_84d77c15	RUNNING

Image generated by Mesos web interface

状态更新与 upone的联动, 例如:

- Running状态, upone更新负载均衡
- Lost状态, 任务迁移

其他消息

消息	简介
KILL	通知执行器删除任务
SHUTDOWN	停止执行器
RECONCILE	用于校准调度器的任务状态
TEARDOWN	删除框架, 关闭所有相关任务和执行器

More Features

基于以上的 mesos http api, upone可以实现一个具备基本功能的调度框架

那么除此之外, upone还可以到哪些

特性

负 载 均 衡

零停机更新

自定义策略

高 可 用

UI + 命令行

负载均衡

基于 ngx_lua 的动态负载均衡方案： Slardar

Hub, Inc. [US] https://github.com/upyun/slardar

Description

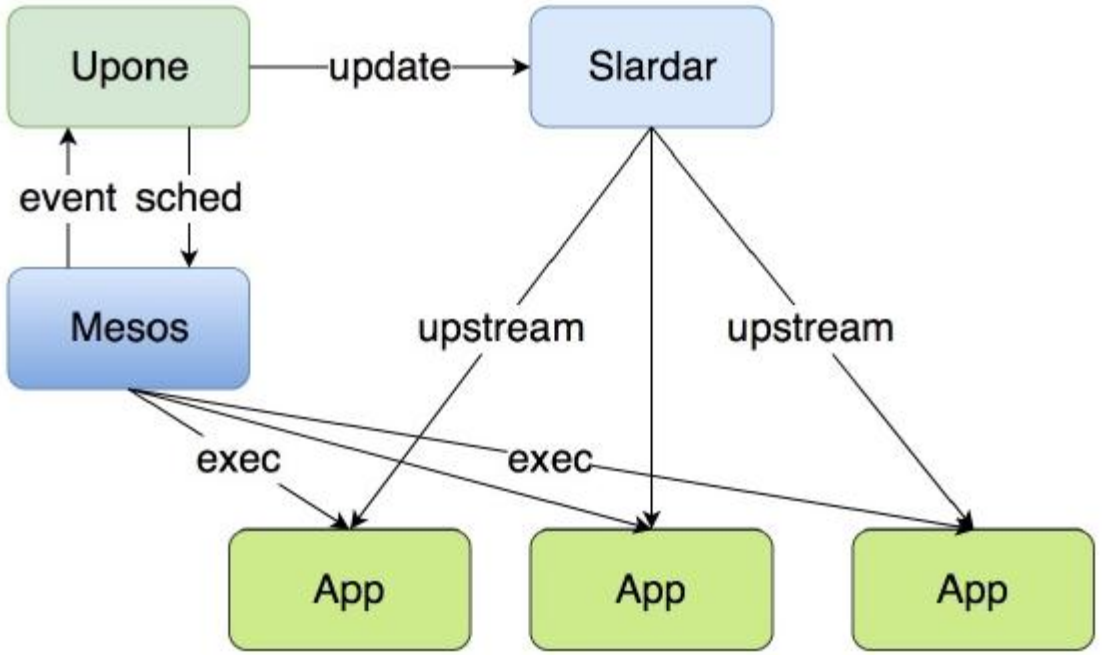
Slardar is a HTTP load balancer based on Nginx and lua-nginx-module, by which you can update your upstream list and run lua scripts without reloading Nginx.

This bundle is maintained by UPYUN(又拍云) Inc.

Because most of the nginx modules are developed by the bundle maintainers, it can ensure that all these modules are played well together.

The bundled software components are copyrighted by the respective copyright holders.

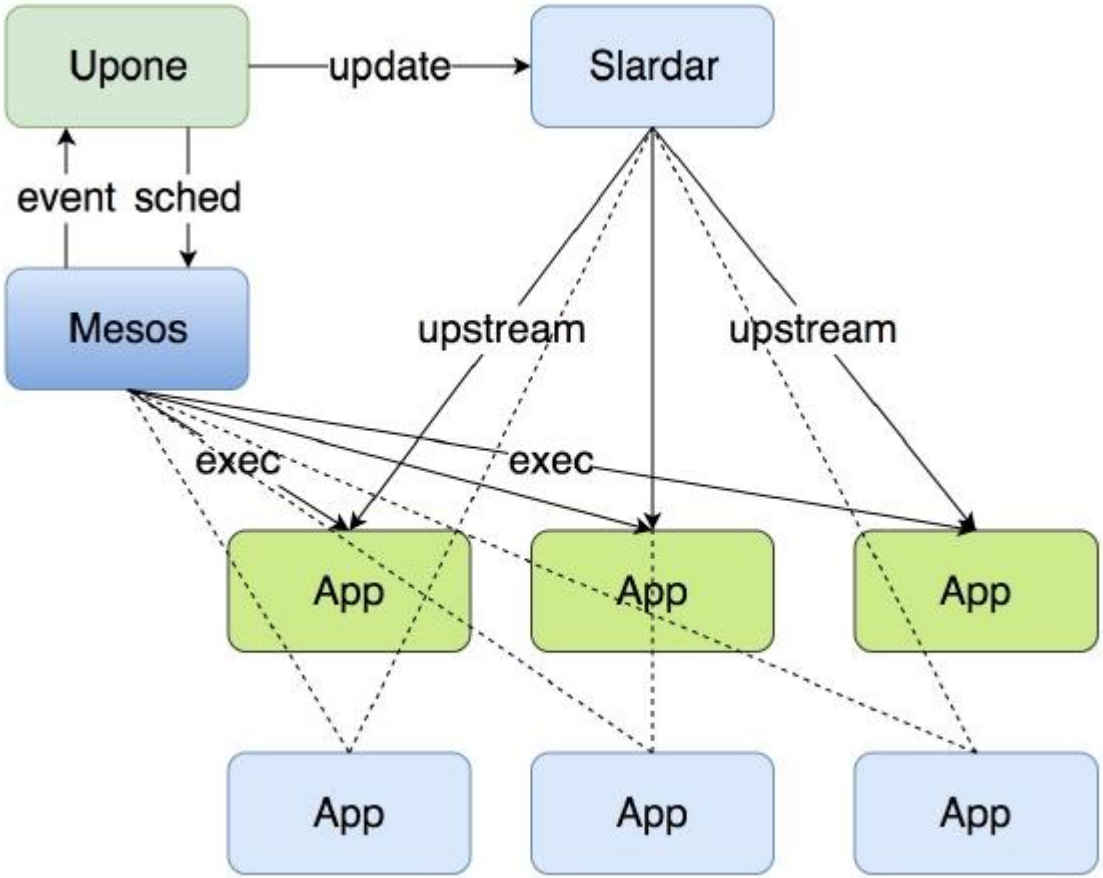
负载均衡



update slardar:
curl 127.0.0.1:1995/upstream/node-dev.example.com -d \
{"servers":[{"host":"10.0.5.108", "port": 1234}]}

零停机更新

通过允许两个不同版本的 App同时运行来实现蓝绿更新



健康检查

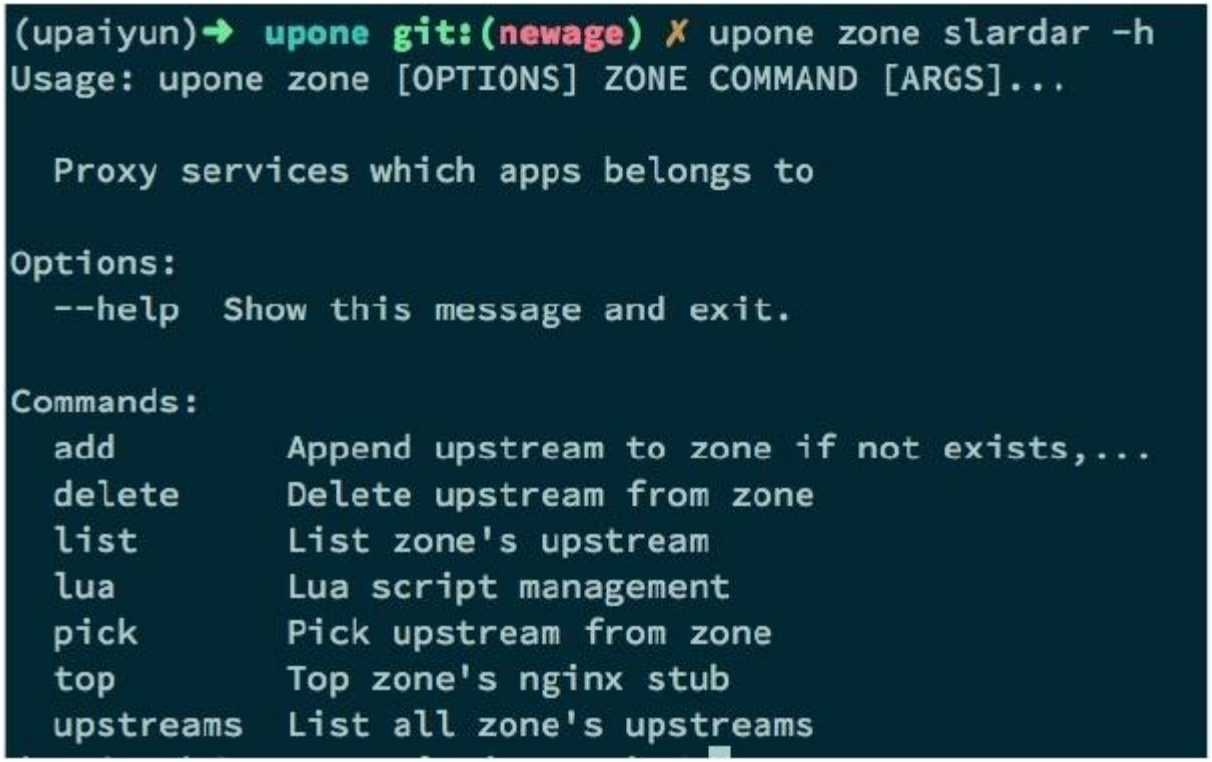

```
{
    -- checkups heartbeat timer is alive.
    "checkup_timer_alive": true,

    -- last heartbeat time
    "last_check_time": "2016-08-12 13:09:40",

    -- status for node-dev cluster
    "cls:node-dev": [
        [
            {
                "server": "node-dev:10.0.5.108:1234",
                "weight": 1,
                "fail_timeout": 30,
                "status": "ok",
                "max_fails": 6
            }
        ]
    ]
}
```

预处理

以频率限制为例, 可以在访问 app前载入 limit相关脚本完成处理

```
"modules": [
    {
        "time": "2017-08-04 13:09:40",
        "version": "aed4a968ef14f8db732e3602c34dc37a",
        "name": "modules.limit"
    },
    {
        "time": "2017-08-04 13:09:40",
        "version": "302f9bf40fcd3734cab120b97f18edf3",
        "name": "script.limit"
    }
]
```

命令行工具



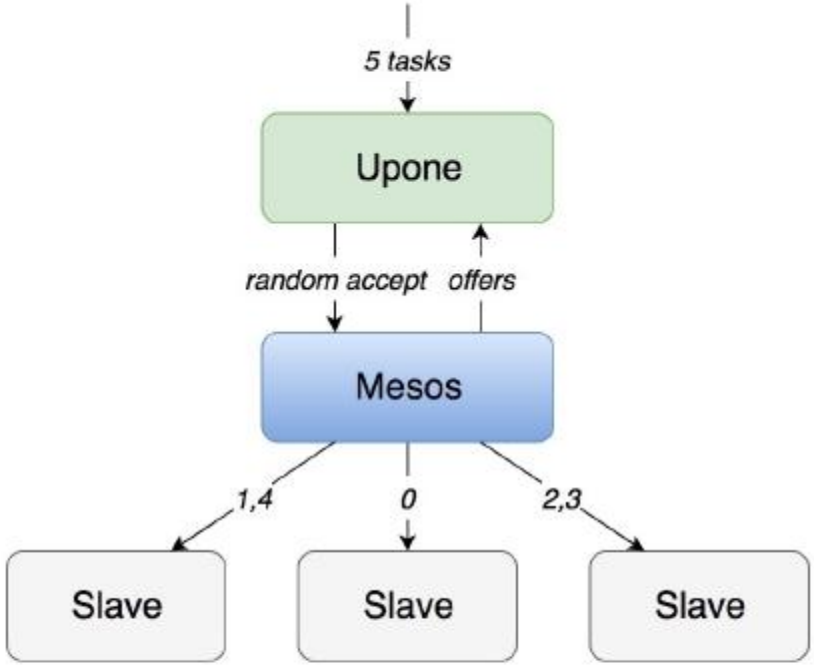
同时, 我们还提供命令行工具, 以便 App所有者通过 upone手动操作上述负载均衡和更新部署的相关功能

自定义策略

收到 mesos o er 后, 可以自定义策略来决定任务调度方案

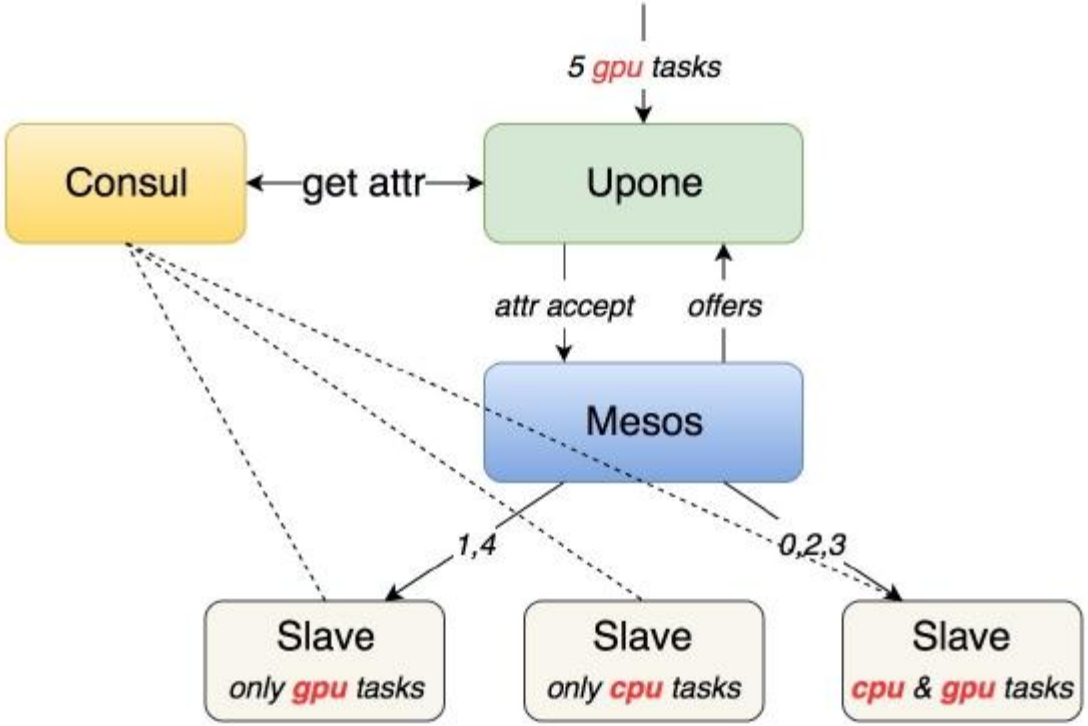
- 随机调度

每次调度尽可能让同一应用的各个任务分布到不同机器



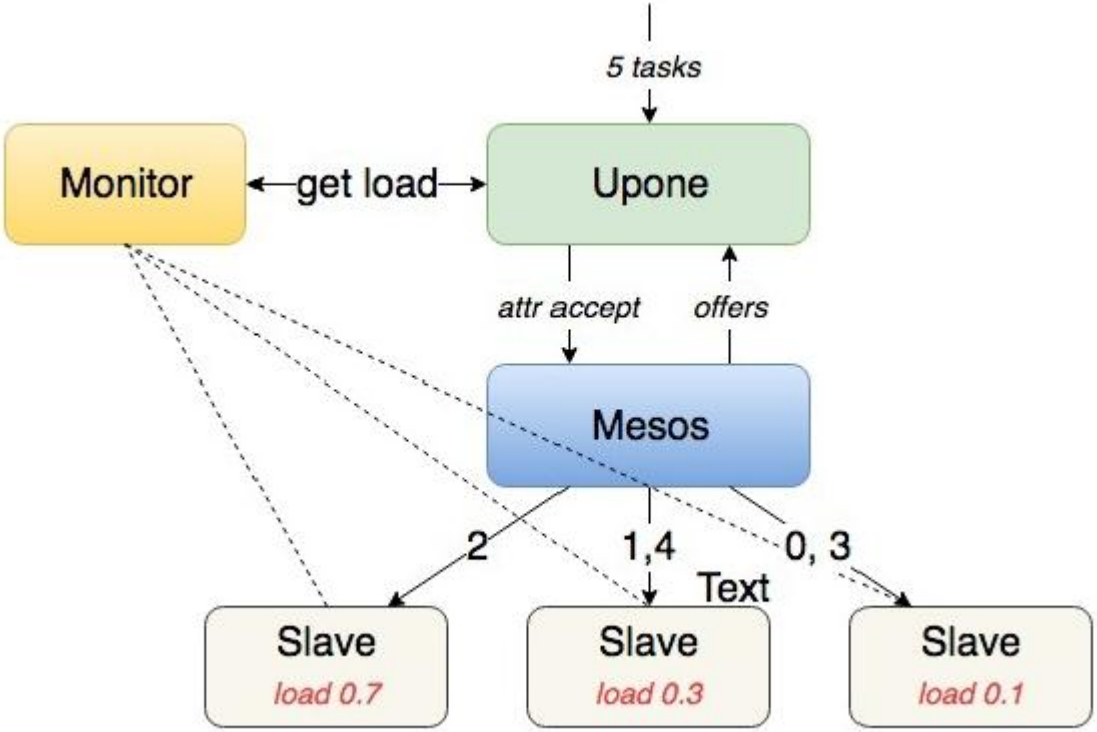
自定义策略

- 根据机器和任务的属性进行调度

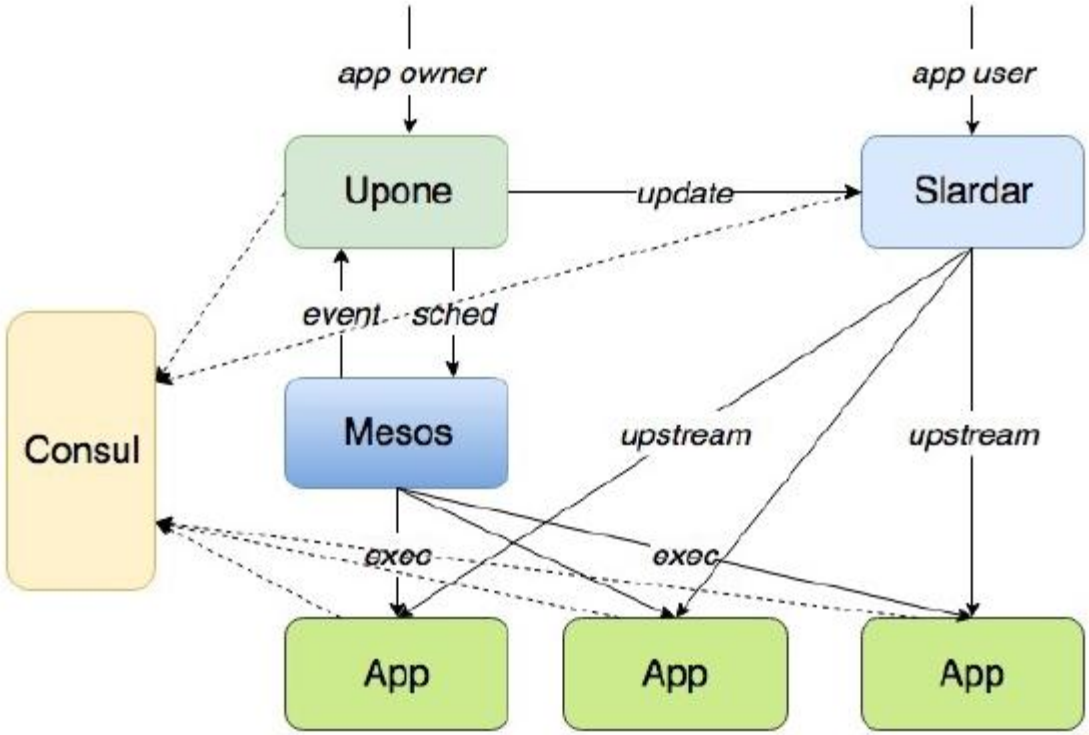


自定义策略

- 根据机器 weight/load 动态调度



配置与服务发现



- upone动态加载属性
- 注册服务
- 应用配置统一管理

高可用

通过 **Raft**分布式一致性协议实现高可用

[hashicorp/raft](https://github.com/hashicorp/raft)(https://github.com/hashicorp/raft)

- 领导选举: 心跳机制来触发选举, term充当逻辑时钟的作用
- 日志复制: 领导者把一条指令(能被复制状态机执行)附加到日志中，发起附加条目 RPC请求给其他角色
- 强领导者：日志条目只从 leader发送给其他的服务器

raft领导选举

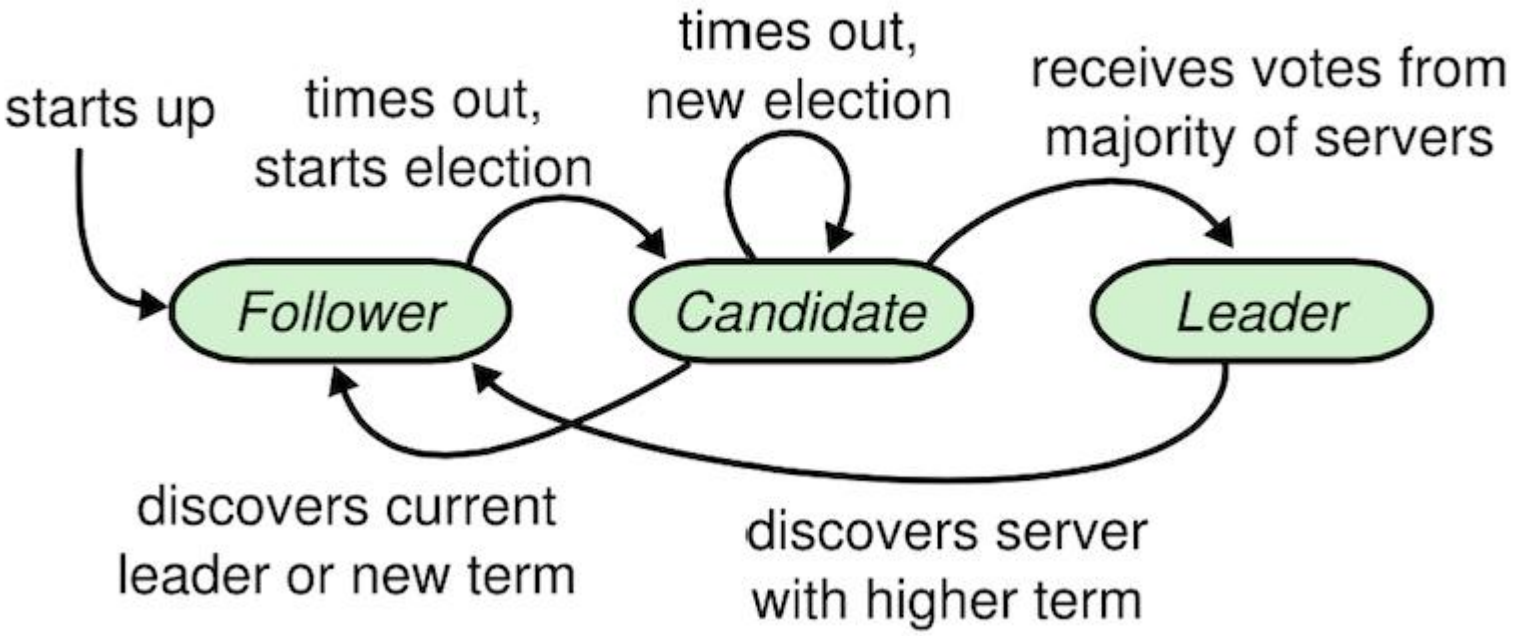


Image credit: the Raft paper (https://raft.github.io/raft.pdf)

raft日志复制

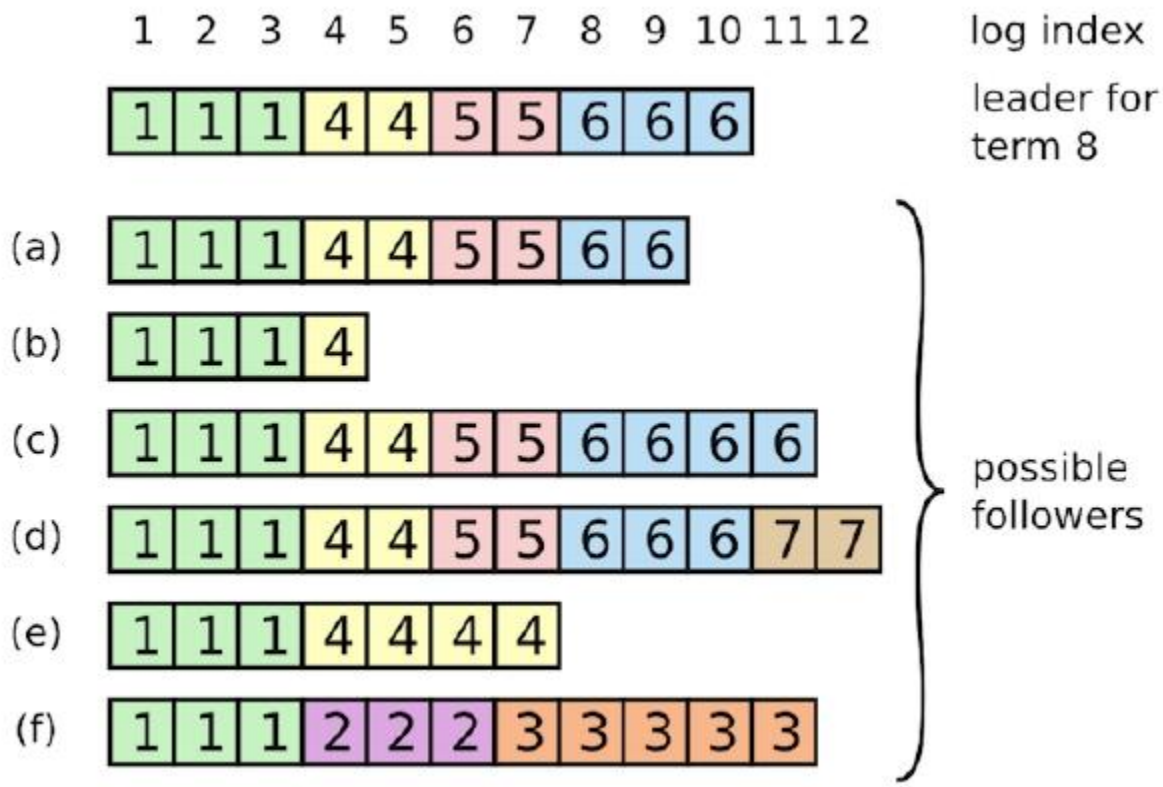


Image credit: the Raft paper (https://raft.github.io/raft.pdf)

raft可视化

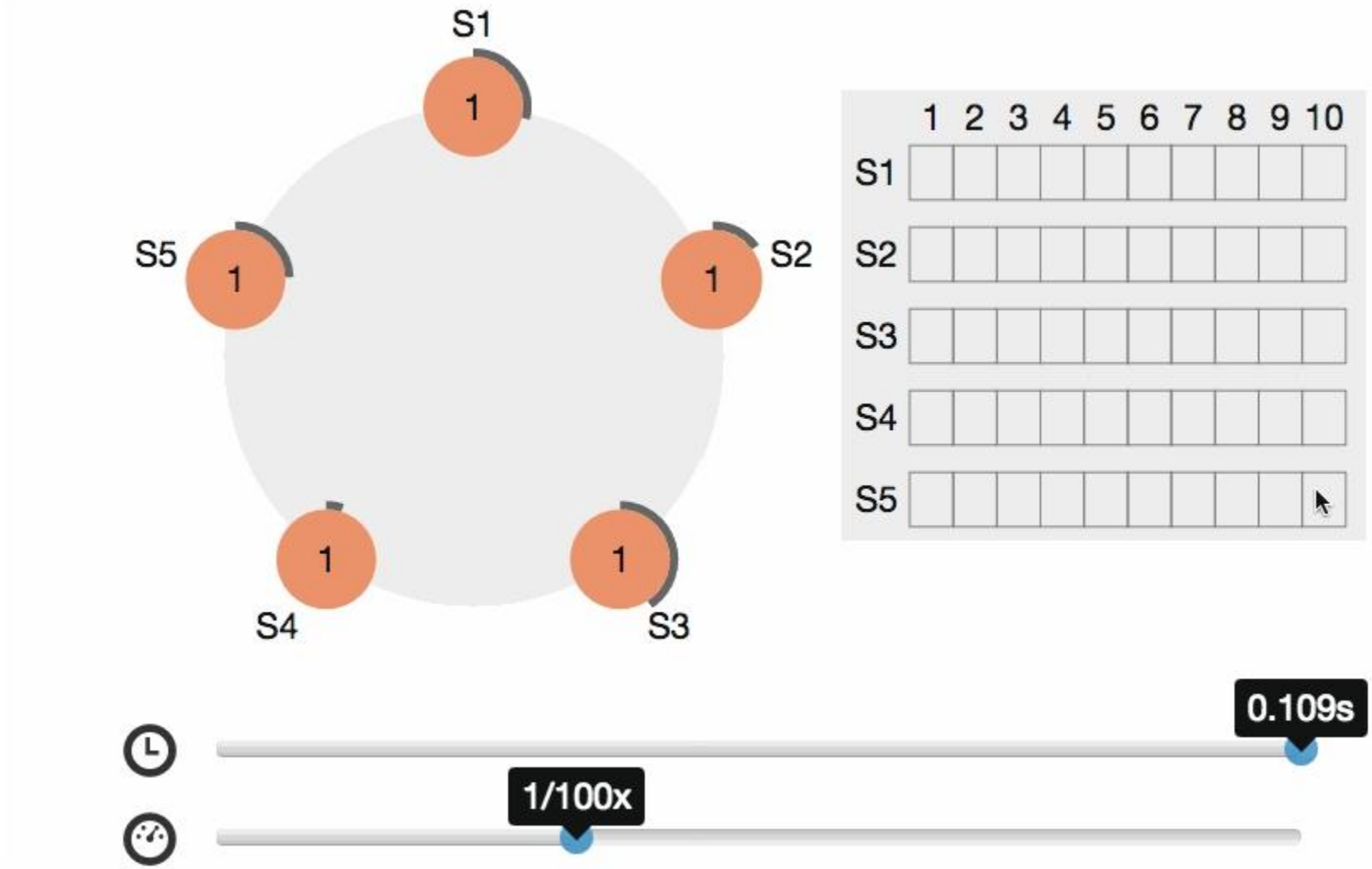


Image generated by [RaftScope](https://raft.github.io/raftscope/index.html)

upone定时任务

mesos/chronos(https://github.com/mesos/chronos) 很多遗留问题没有解决

不再维护:

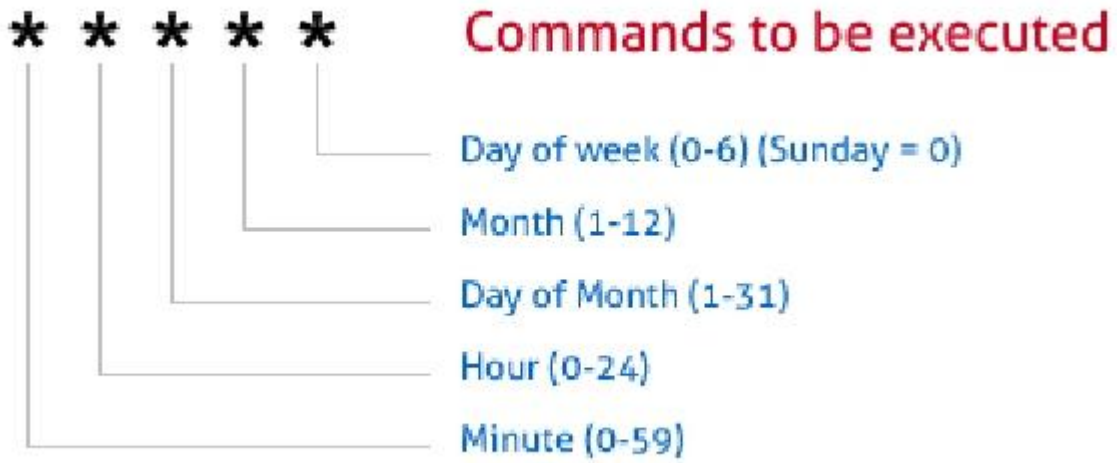
- is:issue is:open 188
- Latest commit on 1 Mar

遇到的问题:

- cpu利用率达到 100%
- 共享数据缺乏锁保护

定时任务特性

- 支持 upone的上述特性(自定义调度等)
- 支持 linux定时任务 crontab的配置格式



组件化

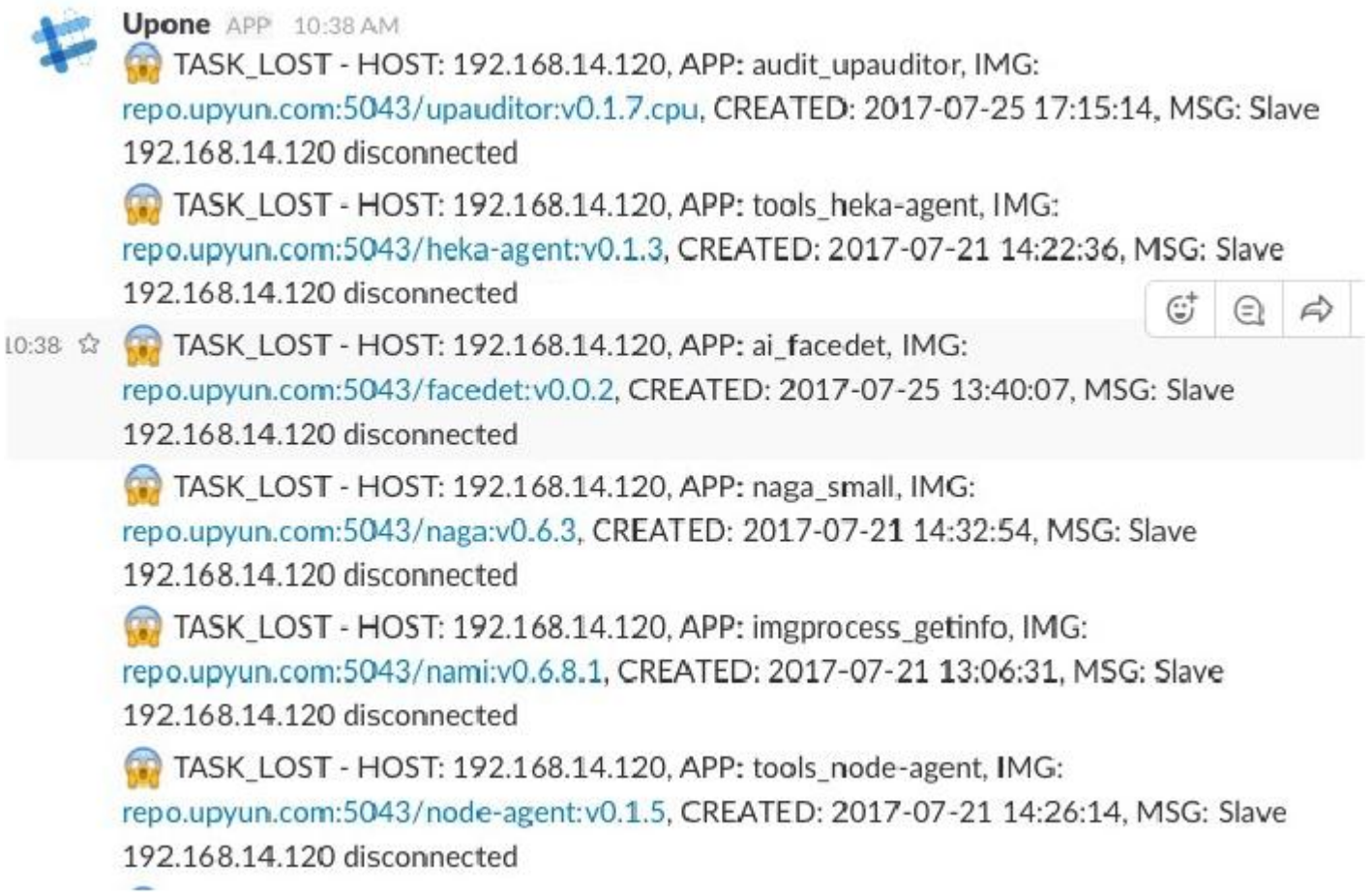
- store组件, 已支持 hashicorp/raft-boltdb(https://github.com/hashicorp/raft-boltdb)数据库,
- drive组件, upone可以选择 master-agen模式, 不依赖于 mesos独立使用

告警

各个应用可以在创建任务的时候指定自己的 slack channel

当任务出现异常时, 发送实时通知

告警


Upone APP 10:38 AM
TASK_LOST - HOST: 192.168.14.120, APP: audit_upauditor, IMG: repo.upyun.com:5043/upauditor:v0.1.7.cpu, CREATED: 2017-07-25 17:15:14, MSG: Slave 192.168.14.120 disconnected
TASK_LOST - HOST: 192.168.14.120, APP: tools_heka-agent, IMG: repo.upyun.com:5043/heka-agent:v0.1.3, CREATED: 2017-07-21 14:22:36, MSG: Slave 192.168.14.120 disconnected
10:38
TASK_LOST - HOST: 192.168.14.120, APP: ai_facedet, IMG: repo.upyun.com:5043/facedet:v0.0.2, CREATED: 2017-07-25 13:40:07, MSG: Slave 192.168.14.120 disconnected
TASK_LOST - HOST: 192.168.14.120, APP: naga_small, IMG: repo.upyun.com:5043/naga:v0.6.3, CREATED: 2017-07-21 14:32:54, MSG: Slave 192.168.14.120 disconnected
TASK_LOST - HOST: 192.168.14.120, APP: imgprocess_getinfo, IMG: repo.upyun.com:5043/nami:v0.6.8.1, CREATED: 2017-07-21 13:06:31, MSG: Slave 192.168.14.120 disconnected
TASK_LOST - HOST: 192.168.14.120, APP: tools_node-agent, IMG: repo.upyun.com:5043/node-agent:v0.1.5, CREATED: 2017-07-21 14:26:14, MSG: Slave 192.168.14.120 disconnected

遇到的问题

- 孤儿任务

mesos上未注册的 framework,没有办法 teardown
只能等 failover_timeout,或者再次注册同一 id,主动删除
最近发现 failover_timeout之后, mesos上的孤儿任务依然保留着

- tcp_tw_recycle

在请求不能保证时间戳单调递增的情况下(比如 NAT)会出现丢包

IT大咖说 知识分享平台

References

- docs

[1] scheduler-http-api(http://mesos.apache.org/documentation/latest/scheduler-http-api/)

[2] mesos tech report(http://mesos.berkeley.edu/mesos_tech_report.pdf)

[3] the Raft paper(https://raft.github.io/raft.pdf)

[4] RaftScope(https://raft.github.io/raftscope/index.html)

- projects

[1] apache/mesos(https://github.com/apache/mesos)

[2] upyun/slardar(https://github.com/upyun/slardar)

[3] hashicorp/consul(https://github.com/hashicorp/consul)

[4] hashicorp/raft(https://github.com/hashicorp/raft)

[5] mesos/chronos(https://github.com/mesos/chronos)

[6] mesosphere/marathon(https://github.com/mesosphere/marathon)

Thank you

黄励博(huangnauh)
又拍云
ihuangnauh@gmail.com (mailto:ihuangnauh@gmail.com)
https://github.com/huangnauh (https://github.com/huangnauh)