Mesos Go Stateful

An Abstraction for frameworks running stateful workload

Dhilip & Amit - PaaS Team, Huawei

Contents

- Why Abstraction
- Available solution in Kubernetes
- Available solution in Mesos
- Mesos Go Stateful

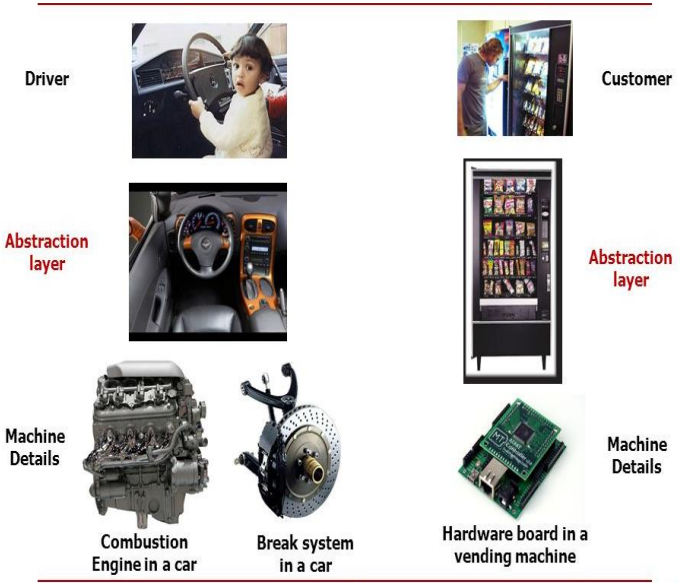
HUAWEI

Design Patterns

- Four essential element Pattern, Problem, Solution and Consequences
- Program to an interface not an Implementation
- General reusable solution to a commonly occurring problem
- Not a finished design that can be transformed directly into source or machine code
- Description or template for how to solve a problem that can be used in many different situations
- Design patterns can speed up the development process by providing tested, proven development paradigms
- Design patterns reside in the domain of modules and interconnections
- Mostly there are 23 types of design patterns categorized in Behavioral design patterns,Creational design patterns,Structural design patterns...etcd
- Example : Factory pattern , Singleton Pattern, Adaptor Pattern etc

Why Abstraction

- Reducing the complexity of the systems
- Key elements of good software design
- Decouple software modules
- More self-contained modules
- Makes the application extendable in much easier way
- Code Reusability
- Refactoring much easier



We are Proposing a Design Pattern for writing Framework for Stateful workload along with abstracted modules on top of mesos-go

HUAWEI

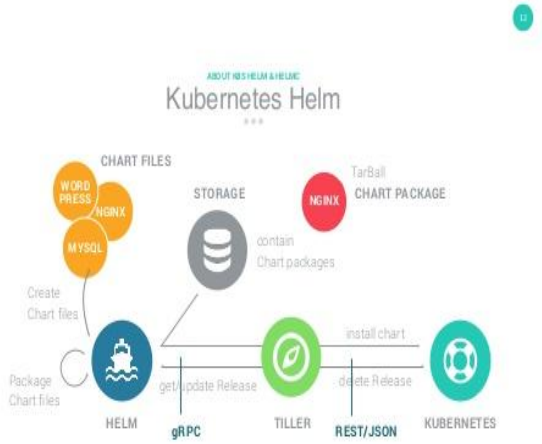
Similar Projects

Kubernetes charts and helm

- Helm is a tool for managing Kubernetes applications
- Charts are packages of pre-configured Kubernetes resources

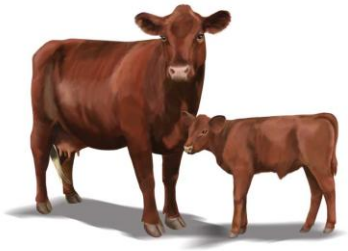
Helm can be used to

- Create reproducible builds of your Kubernetes appli
- Intelligently manage your Kubernetes manifest files
- Share your own applications as Kubernetes charts



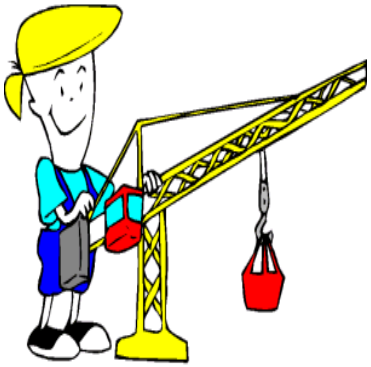
Kubernetes PetSet

- Typically, pods are treated as stateless units, so if one of them is unhealthy or gets superseded, Kubernetes just disposes it.
- So Petset will be used in contrast ,is a group of stateful pods that has a stronger notion of identity.
- It assigns unique identities to individual instances of an application
- PetSet requires {0..n-1} Pets
- Each Pet has a deterministic name, PetSetName-Ordinal, and a unique identity
- The identity of a pet set comprised of
  - A stable DNS hostname
  - An ordinal index
  - Storage linked to ordinal and hostname



CoreOs Operator (for K8s)

- Introduced on 3rd Nov 2016
- An Operator is an application-specific controller .
- That extends the Kubernetes API to create, configure, and manage instances of complex stateful applications on behalf of a Kubernetes user
- An Operator builds upon the basic Kubernetes resource and controller concepts and adds a set of knowledge or configuration that allows the Operator to execute common application tasks



K8s Operators defines some set of rules

- Operator as scheduler
- Operator create types (application specific task)
- Operator leverage built-in primitives like Service and ReplicaSet
- Decouple Operator lifecycle with workload life cycle
- User can declare desired version
- Operators should be tested against a "Chaos Monkey"

HUAWEI

DCOS Commons

- It is a collection of classes and utilities necessary for building a DCOS service
- It is written in Java and is Java 1.8+ compatible.

Spring Cloud

- Provides tools for developers to quickly build some of the common patterns in distributed systems
- It is written in Java
- Main Projects
  - Spring Cloud Config
  - Spring Cloud Netflix
  - Spring Cloud for Cloud Foundry
  - Spring Cloud Security

Analysis of Different Stateful Workload

MySql	Kafka	ETCD	PostgreSql	Redis
Master config: vi /etc/mysql/my.cnf bind-address=12.34.56.789 server-id = 1 log_bin=/var/log/mysql/mysql-bin.log binlog_do_db = newdatabase	Leader and follower config: vi ~/kafka/config/server1.properties broker.id=1 port=9092 host.name=ec2-<IP1>.amazonaws.com num.partitions=4 zookeeper.connect=ec2-<IP1>.amazonaws.com:2080,ec2-<IP2>.amazonaws.com:2080	Master and Slave config: vi /etcd/etcd.conf --name = infra0 --initial-advertise-peer-urls = http://10.0.1.10:2380 --listen-peer-urls = http://10.0.1.10:2380 --listen-client-urls = http://10.0.1.10:2379,http://127.0.0.1:2379 --advertise-client-urls= http://10.0.1.10:2379 --initial-cluster-token = etcd-cluster-1 --initial-cluster = infra0=http://10.0.1.10:2380,infra1=http://10.0.1.11:2380,infra2=http://10.0.1.12:2380 --heartbeat-interval=100 --election-timeout=500 --initial-cluster-state = new	Master config: vi pg_hba.conf host replication rep slave_ip/32 md5 vi postgresql.conf listen_addresses = 'localhost,master_ip' wal_level = 'hot_standby' archive_mode = on archive_command = 'cd .' max_wal_senders = 1 hot_standby = on	Master config: vi /etc/redis/redis.conf tcp-keepalive = 60 bind = 12.34.56.789 requirepass = master_password appendonly = yes appendfilename = redis-staging-ao.aof
Slave config: vi /etc/mysql/my.cnf bind-address= 12.23.34.456 server-id = 2 binlog_do_db = newdatabase mysql>CHANGE MASTER TO MASTER_HOST='12.34.56.789',MASTER_USER='slave_user', MASTER_PASSWORD='pa		Note:It automatically handles leader election via Raft Consensus protocol.	Slave config: vi pg_hba.conf host replication rep master_ip/32 md5 vi postgresql.conf listen_addresses = 'localhost,slave_ip' wal_level = 'hot_standby' archive_mode = on archive_command = 'cd .' max_wal_senders = 1 hot_standby = on	Slave config: vi /etc/redis/redis.conf bind = 12.23.34.456 requirepass = slave_password slaveof = redis_master_ip 6379 masterauth = master_password

The Problem

As a Framework Developer

Need to expose endpoints

Need to deal with offers

Need to write custom executor

Need to maintain state of the tasks

Need to distribute Workload optimally

May require higher degree of control over Docker

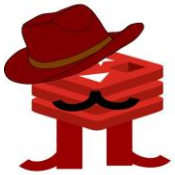
HUAWEI

What is Mesos Go Stateful

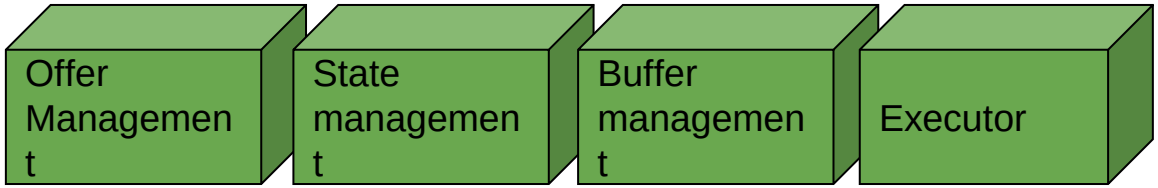
High level abstraction on top of frameworks language bindings
which makes framework development for stateful workloads more easier

https://github.com/huawei-cloudfederation/mesos-go-stateful

Service Framework



Abstraction



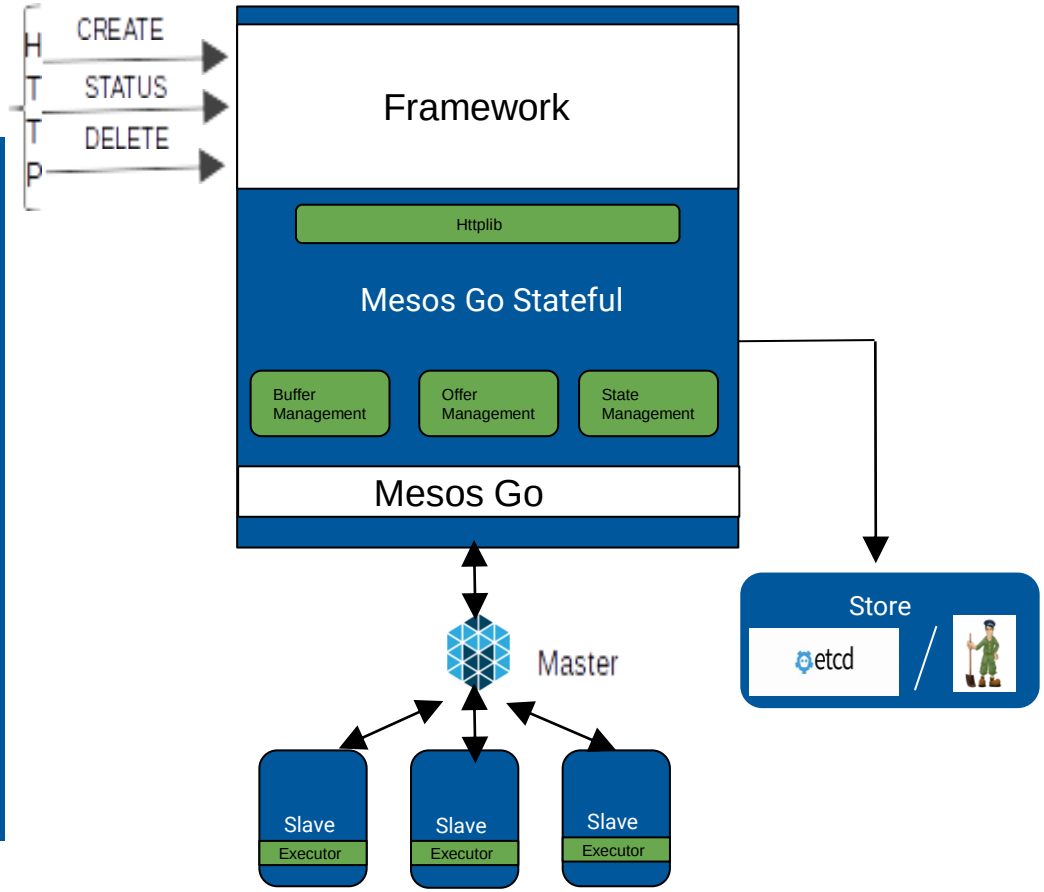
Language BInding

Mesos

HUAWEI

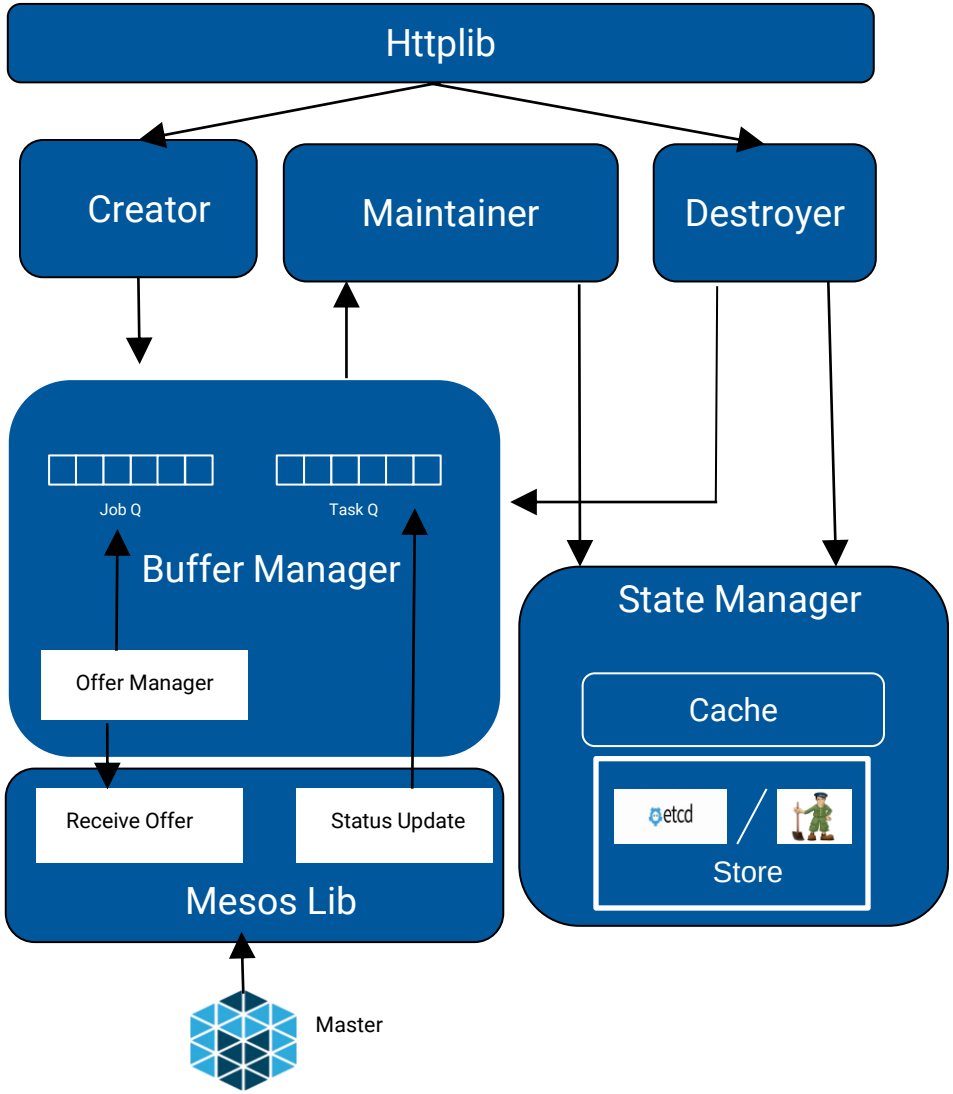
Overall Design

- 1000 feet Overview
- HttpLib handles CRUD operation
- Abstract out complexity of Offers and events from mesos-go
- Decouple framework with language binding with buffer management.
- Abstract out the Store (key / value) management

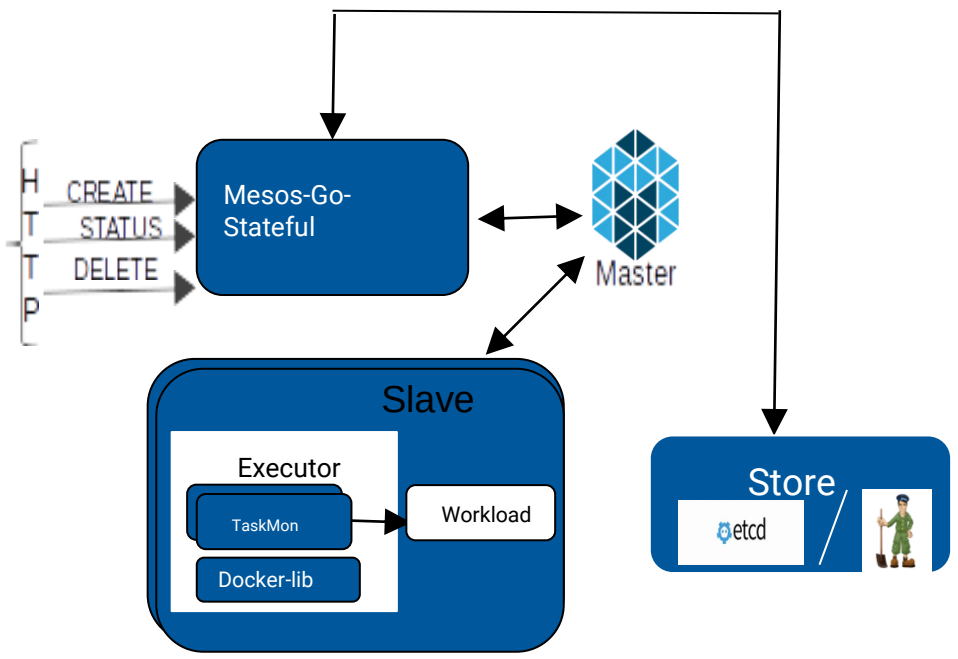


Design Cont…

- HttpLib maintains controller with user routes to schedule/destroy workload
- Creation request to Creator for getting it scheduler as workload.
- Delete request for Destroyer for deleting workload
- Buffer Manager maintains Queues for Scheduled Job and Task update.
- Offer manager watches Job queue and optimally manages the offers
- TaskQ gets updated by Status update event
- Maintainer keep watch on TaskQ and Update status of each task in Store.
- State manager provides interface for Store interactions. It maintains Cache for faster transactions.



Executor



- Pull the docker images from docker daemon.
- Create docker containers
- Start the containers
- Launch the workload
- Collects stats from docker container
- Update stats to store
- Monitor the workloads
- Stop the workload

Callbacks

CALL BACK	DESCRIPCITION
`func (S *TestFWScheduler) Config(I *typ.Instance, IsMaster bool) []string { …. }`	Will be called before the Instances/Tasks are created, can be used to auto-generate config files or command line arguments for each task
`func (S *TestFWScheduler) Start(I *typ.Instance) error { …. }`	General call back for starting a workload regardless of it being a master or slave
`func (S *TestFWScheduler) StartMaster(I *typ.Instance) error { …. }`	Specifically a call back to start MASTER/LEADER type of workloads, perform master related work like configuring PROXY / Updating service discovery etc. Will talk to 'CREATOR'
`func (S *TestFWScheduler) StartSlave(I *typ.Instance) error { …. }`	Simlar config call backs for Slaves / Peers to help service discovery will talk to 'CREATOR'
`func (S *TestFWScheduler) MasterRunning(I *typ.Instance) error { …. }`	Will be invoked when 'TASK_RUNNING' update is recived by the framework.
`func (S *TestFWScheduler) SlaveRunning(I *typ.Instance) error { …. }`	Will be invoked when 'TASK_RUNNING' update is recived by the framework.
`func (S *TestFWScheduler) MasterLost(I *typ.Instance) error { …. }`	Will be invoked when 'TASK_RUNNING' update is recived by the framework. This could internally call 'StartMaster'
`func (S *TestFWScheduler) SlaveLost(I *typ.Instance) error { …. }`	Will be invoked ind if TASK_LOST / TASK_ERROR / TASK_FAILED task updates, this could internally call

Project Development Status

Module	Progress
Httplib	90%
CMD	40%
Offer Manager	90%
Executor	40%
Mesoslib	90%
Dockerlib	90%
StateManager	30%
BufferManager	90%

Demo

Screen Shot: Code Generation

```
$./codegen -name MConAsia -path $HOME
I1116 07:03:02.223101   14354 gen.go:173] Creating Sub-directories at /home/ubuntu/MConAsia.....
I1116 07:03:02.223265   14354 gen.go:197] Generating Scheduler.go...
I1116 07:03:02.223629   14354 gen.go:229] Generating autofilled config file
I1116 07:03:02.223799   14354 gen.go:250] Project Generation Completed

~/MConAsia$ ls -lrt
total 12
drwxrwxr-x 2 ubuntu ubuntu 4096 Nov 16 07:03 Scheduler
drwxrwxr-x 2 ubuntu ubuntu 4096 Nov 16 07:03 Executor
drwxrwxr-x 2 ubuntu ubuntu 4096 Nov 16 07:03 Config

~/MConAsia/Scheduler$ go build .
~/MConAsia/Scheduler$ ls -lrt
total 24716
-rw-rw-r-- 1 ubuntu ubuntu     1829 Nov 16 07:03 Scheduler.go
-rwxrwxr-x 1 ubuntu ubuntu 25302776 Nov 16 07:03 Scheduler

~/MConAsia/Executor$ go build MConAsiaExecutor.go
~/MConAsia/Executor$ ls -lrt
total 22164
-rw-rw-r-- 1 ubuntu ubuntu      884 Nov 16 07:03 MConAsiaExecutor.go
-rwxrwxr-x 1 ubuntu ubuntu 22688896 Nov 16 07:05 MConAsiaExecutor
```

Screen Shot: Offer Management

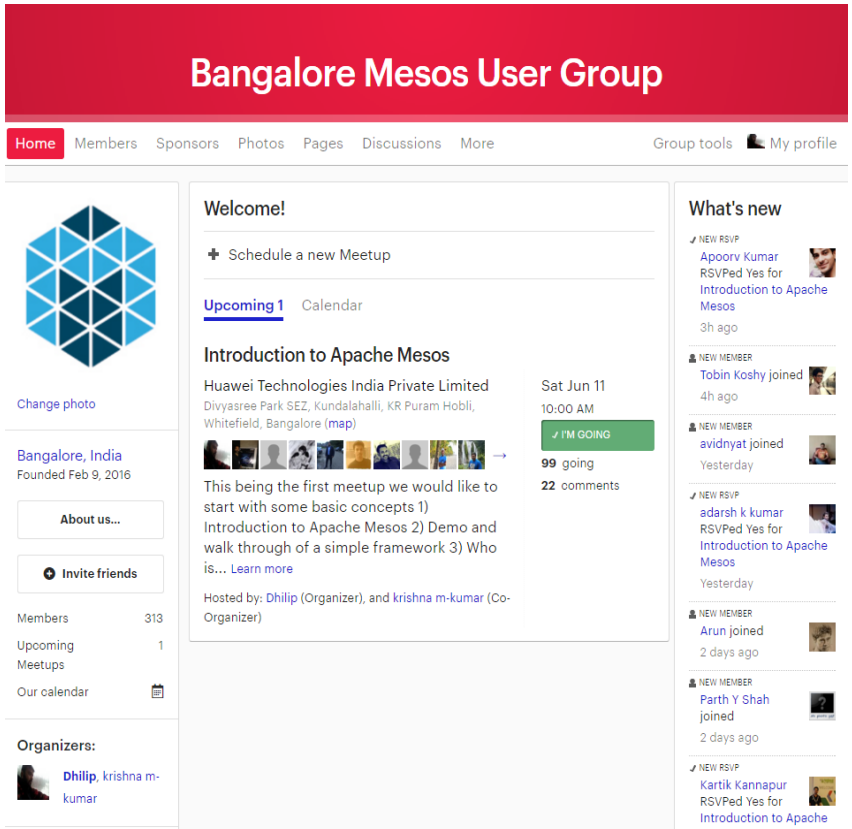
```
I1116 11:51:00.863705    6620 workloadscheduler.go:29] Framework Tet2 Registered
&FrameworkID{Value:*998fec17-c85e-4fd1-b090-6c421a3e286b-0006,XXX_unrecognized:[],}
I1116 11:51:02.796815    6620 workloadscheduler.go:65] DECLINE OFFERS for 1 Next Hour
I1116 11:52:15.879995    6620 httplib.go:27] HTTP: CREATE request for instance test1
I1116 11:52:15.879995    6620 httplib.go:48] Request Accepted, test1 Instance will be created
I1116 11:52:15.881996    6620 cmd.go:58] CREATOR: Recived {test1 3 {1 100 1 host redis:3.0-alpine}} from
HTTP
I1116 11:52:15.882996    6620 JobList.go:87] JOBLIST: Call NewEvent()
I1116 11:52:15.882996    6620 workloadscheduler.go:188] OfferLIST Queued
I1116 11:52:16.169012    6620 workloadscheduler.go:99] Received Offer with CPU=8 MEM=6960 OfferID=998fec17-
c85e-4fd1-b090-6c421a3e286b-O99
I1116 11:52:16.169012    6620 workloadscheduler.go:143] Launched 1 tasks from this offer
I1116 11:52:16.169012    6620 workloadscheduler.go:99] Received Offer with CPU=8 MEM=6960 OfferID=998fec17-
c85e-4fd1-b090-6c421a3e286b-O100
I1116 11:52:16.169012    6620 workloadscheduler.go:143] Launched 0 tasks from this offer
I1116 11:52:16.169012    6620 workloadscheduler.go:99] Received Offer with CPU=8 MEM=6960 OfferID=998fec17-
c85e-4fd1-b090-6c421a3e286b-O101
I1116 11:52:16.170012    6620 workloadscheduler.go:143] Launched 0 tasks from this offer
I1116 11:52:16.170012    6620 workloadscheduler.go:145] workload Receives offer
I1116 11:52:16.608037    6620 workloadscheduler.go:155] workload Task Update received
I1116 11:52:22.358366    6620 workloadscheduler.go:65] DECLINE OFFERS for 1 Next Hour
```

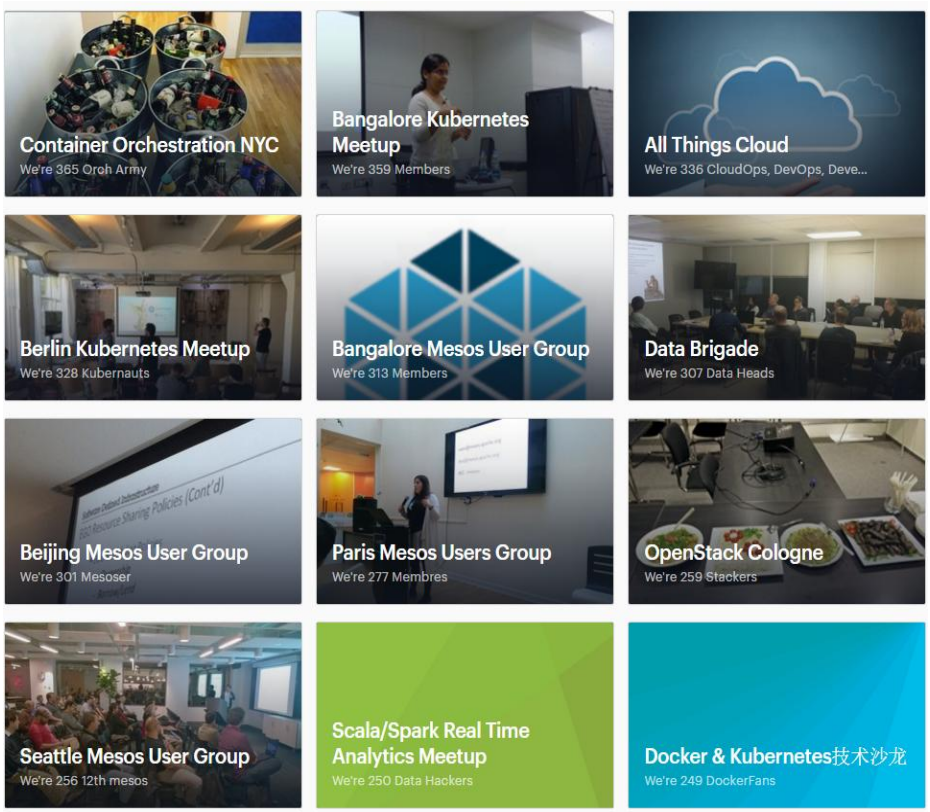
Future Work

- Add generic UI capability
- Reimplement Mr-Redis Framework
- Implement Regression suit to test SDK
- Test with different stateful workload

HUAWEI

Mesos Community Info





http://www.meetup.com/Bangalore-Mesos-User-Group/

Krishna M Kumar <krishna.m.kumar@huawei.com>
Dhilip Kumar S <dhilip.kumar.s@huawei.com>
Amit Kumar Roushan <amit.roushan@huawei.com>

HUAWEI

Thank You

HUAWEI