Deployment of demo.pyspider.org

demo.pyspider.org is running on three VPSs connected together with private network using tinc.

1vCore 4GB RAM	1vCore 2GB RAM * 2
database message queue scheduler	phantomjs * 2 phantomjs-lb * 1 fetcher * 1 fetcher-lb * 1 processor * 2 result-worker * 1 webui * 4 webui-lb * 1 nginx * 1

All components are running inside docker containers.

database / message queue / scheduler

The database is postgresql and the message queue is redis.

Scheduler may have a lot of database operations, it's better to put it close to the database.

```
docker run --name postgres -v /data/postgres/:/var/lib/postgresql/data -d -p $LOCAL_IP:5
docker run --name redis -d -p  $LOCAL_IP:6379:6379 redis
docker run --name scheduler -d -p $LOCAL_IP:23333:23333 --restart=always binux/pyspider
 --taskdb "sqlalchemy+postgresql+taskdb://binux@10.21.0.7/taskdb" \
 --resultdb "sqlalchemy+postgresql+resultdb://binux@10.21.0.7/resultdb" \
 --projectdb "sqlalchemy+postgresql+projectdb://binux@10.21.0.7/projectdb" \
 --message-queue "redis://10.21.0.7:6379/1" \
 scheduler --inqueue-limit 5000 --delete-time 43200
```

other components

fetcher, processor, result_worker are running on two boxes with same configuration managed with docker-compose.

```
phantomjs:
  image: 'binux/pyspider:latest'
  command: phantomjs
  cpu_shares: 512
  environment:
    - 'EXCLUDE_PORTS=5000,23333,24444'
  expose:
    - '25555'
  mem_limit: 512m
  restart: always
phantomjs-lb:
  image: 'dockercloud/haproxy:latest'
  links:
    - phantomjs
  restart: always

fetcher:
  image: 'binux/pyspider:latest'
  command: '--message-queue "redis://10.21.0.7:6379/1" --phantomjs-proxy "phantomjs:80" '
  cpu_shares: 512
  environment:
    - 'EXCLUDE_PORTS=5000,25555,23333'
  links:
    - 'phantomjs-lb:phantomjs'
  mem_limit: 128m
  restart: always
fetcher-lb:
  image: 'dockercloud/haproxy:latest'
  links:
    - fetcher
  restart: always

processor:
  image: 'binux/pyspider:latest'
  command: '--projectdb "sqlalchemy+postgresql+projectdb://binux@10.21.0.7/projectdb" --r
  cpu_shares: 512
  mem_limit: 256m
  restart: always

result-worker:
  image: 'binux/pyspider:latest'
  command: '--taskdb "sqlalchemy+postgresql+taskdb://binux@10.21.0.7/taskdb"  --projectdb
  cpu_shares: 512
  mem_limit: 256m
  restart: always

webui:
  image: 'binux/pyspider:latest'
  command: '--taskdb "sqlalchemy+postgresql+taskdb://binux@10.21.0.7/taskdb"  --projectdb

  cpu_shares: 512
  environment:
    - 'EXCLUDE_PORTS=24444,25555,23333'
  links:
    - 'fetcher-lb:fetcher'
  mem_limit: 256m
  restart: always
webui-lb:
  image: 'dockercloud/haproxy:latest'
  links:
    - webui
  restart: always

nginx:
  image: 'nginx'
  links:
    - 'webui-lb:HAPROXY'
  ports:
    - '0.0.0.0:80:80'
  volumes:
    - /home/binux/nfs/profile/nginx/nginx.conf:/etc/nginx/nginx.conf
    - /home/binux/nfs/profile/nginx/conf.d/:/etc/nginx/conf.d/
  restart: always
```

With the config, you can change the scale by `docker-compose scale phantomjs=2 processor=2 webui=4` when you need.

load balance

phantomjs-lb, fetcher-lb, webui-lb are automaticlly configed haproxy, allow any number of upstreams.

phantomjs

phantomjs have memory leak issue, memory limit applied, and it's recommended to restart it every hour.

fetcher

fetcher is implemented with aync IO, it supportes 100 concurrent connections. If the upstream queue are not choked, one fetcher should be enough.

processor

processor is CPU bound component, recommended number of instance is number of CPU cores + 1~2 or CPU cores * 10%~15% when you have more then 20 cores.

result-worker

If you didn't override result-worker, it only write results into database, and should be very fast.