

# Beanstalkd Message Queue

协议·实践·FAQ

# About Me

冯宗宝

基础服务组

Person Home Page : [fzb.me](http://fzb.me)

# Why Queue?

如果有一个工作单元,你想推迟到之后进行,比如,

- 发送短信
- 请求第三方接口
- 生成高质量的产品图像缩略图
- 库存变化通知
- 微信支付银行卡退款状态查询等。

消息队列wikipedia:<https://zh.wikipedia.org/wiki/%E6%B6%88%E6%81%AF%E9%98%9F%E5%88%97>

# Why Beanstalkd?

Beanstalkd, 一个高性能、轻量级的分布式内存队列系统, 最初设计的目的是想通过后台异步执行耗时的任务来降低高容量Web应用系统的页面访问延迟, 支持过有950万用户的Facebook Causes应用。后来开源, 现在有PostRank大规模部署和使用, 每天处理百万级任务。

Beanstalkd是典型的类Memcached设计, 协议和使用方式都是同样的风格, 所以使用过memcached的用户会觉得Beanstalkd似曾相识。

# Beanstalkd and Others

## 性能

---

参考数据:

	入队列	worker处理
delayed job	200 jobs/sec	120 jobs/sec
resque	3800 jobs/sec	300 jobs/sec
rabbitmq	2500 jobs/sec	1300 jobs/sec
beanstalk	9000 jobs/sec	5200 jobs/sec

php进程数	job数量	处理time	效率
1	5366	359sec	14 jobs/sec
10	5366	45sec	119 jobs/sec
50	5366	14sec	383 jobs/sec

# 核心概念

## **job**

一个需要异步处理的任务，是Beanstalkd中的基本单元，需要放在一个tube中。

## **tube**

一个有名的任务队列，用来存储统一类型的job，是producer和consumer操作的对象。

## **producer**

Job的生产者，通过put命令来将一个job放到一个tube中。

## **consumer**

Job的消费者，通过reserve/release/bury/delete命令来获取job或改变job的状态。

# 任务优先级 (priority)

任务 (job) 可以有  $0 \sim 2^{32}$  个优先级, 0 代表最高优先级。beanstalkd 采用最大最小堆 (Min-max heap) 处理任务优先级排序, 任何时刻调用 reserve 命令的消费者总是能拿到当前优先级最高的任务, 时间复杂度为  $O(\log n)$ .

# 延时任务 (delay)

有两种方式可以延时执行任务 (job): 生产者发布任务时指定延时; 或者当任务处理完毕后, 消费者再次将任务放入队列延时执行 (RELEASE with )。这种机制可以实现分布式的 `java.util.Timer`, 这种分布式定时任务的优势是: 如果某个消费者节点故障, 任务超时重发 (time-to-run) 能够保证任务转移到另外的节点执行。



# 任务超时重发 (time-to-run)

Beanstalkd 把任务返回给消费者以后:消费者必须在预设的 TTR (time-to-run) 时间内发送 `delete / release/ bury` 改变任务状态;否则 Beanstalkd 会认为消息处理失败,然后把任务交给另外的消费者节点执行。如果消费者预计在 TTR (time-to-run) 时间内无法完成任务,也可以发送 `touch` 命令,它的作用是让 Beanstalkd 从系统时间重新计算 TTR (time-to-run)。

# 任务预留 (buried)

如果任务因为某些原因无法执行, 消费者可以把任务置为 `buried` 状态让 `Beanstalkd` 保留这些任务。管理员可以通过 `peek buried` 命令查询被保留的任务, 并且进行人工干预。简单的, `kick` 能够一次性把 `n` 条被保留的任务踢回队列。

# Protocol

Beanstalkd Protocol 中文版

# Producer

## 队列投放

```
1 use Pheanstalk\Pheanstalk;
2 // 投放短信发送定时器
3
4 $beanstalk = new Pheanstalk('127.0.0.1', '11300', '2');
5
6 $data = array(
7     'mobile' => '13051662435',
8     'id'      => '11',
9     'time'   => '2015-05-01 10:30:00'
10 );
11 $delay = (int) strtotime($data['time']) - time();
12
13 $beanstalk->useTube('Sms')
14     ->put(json_encode($data), 1024, $delay, 2);
```

# Consumer

```
31     while(1) {
32         $this->job      = $this->beanstalk->pheanstalk->watch($this->tube)->reserve();
33         $payload       = json_decode($this->job->getData(), true);
34         $payload['job_id'] = $this->job->getId();
35         // 分发请求, 调用短信发送接口
36         $this->load->library(
37             array(
38                 'Dachu_request',
39             )
40         );
41         $url = CC('service.s') . '/sms/worker_sms_send';
42         $response = $this->dachu_request->post($url, $payload);
43
44         // 发送成功, 删除Job 失败 buried
45         if($response && isset($response['status']) && $response['status'] == 0) {
46             $this->beanstalk->pheanstalk->delete($this->job);
47             echo 'Job Deleted:' . $this->job->getId() . "\n";
48
49         } else {
50             echo "request failed !\n";
51             echo var_export($response, true);
52         }

```

# Worker

## 1. Supervisor

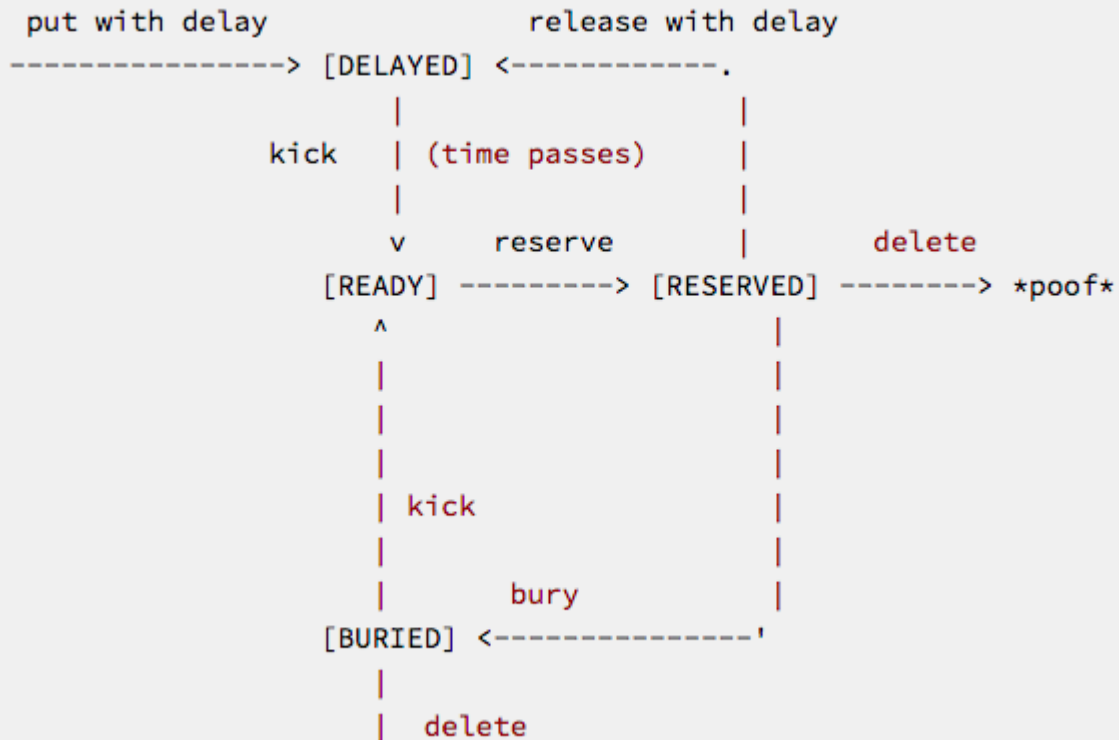
```
8 [program:worker_sms]
9 command=php /data/www/dachuwang/web/applications/worker/index.php worker_sms run
10 priority=1
11 #process_name=%(program_name)s_%(process_num)02d
12 numprocs=1
13 autostart=true
14 autorestart=true
15 stdout_logfile=/data/logs/worker_sms.log ; (main log file;default $CWD/supervisord.log)
16 stdout_logfile_maxbytes=1MB
```

## 2. WorkerManager : <https://github.com/mickeyouyou/YafWorkerManager>

# Normal System

```
put          reserve          delete  
-----> [READY] -----> [RESERVED] -----> *poof*
```

# Delayed System

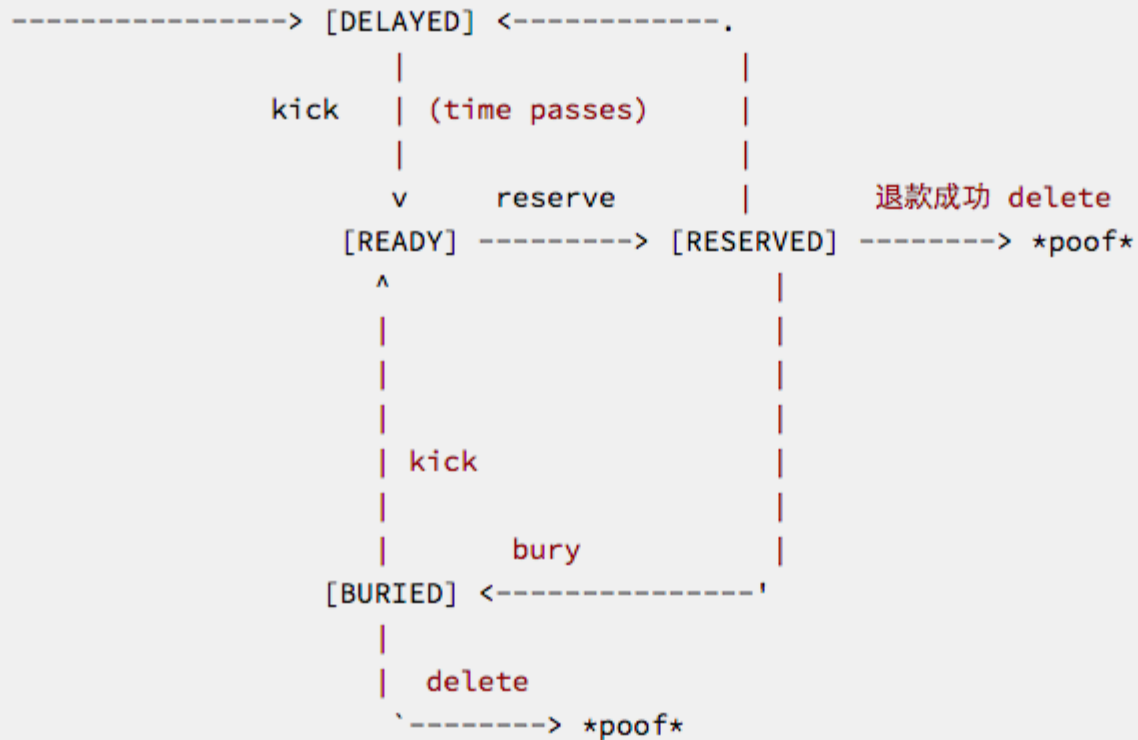




# Poll System

微信退款申请 投放延时任务

退款还未成功 release with delay



# Links

Home page : <http://kr.github.io/beanstalkd/>

Alibaba Core System Team Blog : <http://csrd.aliapp.com/?p=1201>

中文版协议 : <http://www.fzb.me/2015-3-21-beanstalkd-protocol-chinese-version.html>

中文版FAQ: <http://www.fzb.me/2015-7-31-beanstalkd-faq.html>