# 队列系统服务化项目详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档版本号** | **日期** | **作者** | **审核人** | **说明** |
| V0.1 |  | 吴正华 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 总体目标

本项目的目标是收敛队列系统的入口，增强队列的扩展性，完善告警，增加统计功能。

## 实现的功能

1. http接口重构：重点，定时队列优化
2. Worker 重构：重点，守护进程，job与task实现，重载配置，无损退出
3. 统计功能：每个队列每天任务数，失败数；每个队列每秒的进队列的任务数和出队列的任务数
4. 完善监控：worker的错误监控
5. 分流算法：组扩容分流算法和单队列扩容分流算法
6. 库：提供给业务方调用的库

## 接口重构设计

### 接口设计

#### 增加任务到队列

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口描述 | | **增加任务到队列** | | | |
| Request | | | | | |
| 线下：  Http Method：post  Host: http://10.10.8.100:8080/bridgequeue/addTask  线上：  Http Method：post  http://10.231.158.59:8000/bridgequeue/addTask | | | | | |
| Request header | | | | | |
| 字段 | | 取值 | 描述 | | 必选 |
| Request param | | | | | |
| 字段 | | 取值 | 描述 | | 必选 |
| 无 | | 无 | 无 | | 无 |
|  | |  |  | |  |
| Request POST | | | | | |
| 字段 | | 取值 | 描述 | | 必选 |
| qid | | 整型 | 由队列系统分配的队列id，8开头表示普通队列，9开头表示定时队列 | | 是 |
| type | | 整型 | 队列类型，1 普通队列 2 定时队列 | | 是 |
| url | | 字符串 | 回调时url，比如  "http://10.10.8.100:8080/bridgequeue/testCallbackPost" | | 是 |
| mode | | 字符串 | 回调时http请求方式，1 表示get 2表示post | | *是* |
| retry | | 数字 | 回调时，重试次数，合法范围[1,3]，默认3 | | 是 |
| timeout | | 数字 | 回调时，超时时间，合法范围[0,3000]ms | | 是 |
| expire | | 数字 | 过期时间，时间到了就不再发起回调 | | 是 |
| timestamp | | 数字 | 时间戳，type=2时必须参数 | | 可选 |
| params | | 数组 | 回调时，参数，格式为json  比如json\_encode(array("order\_id" =>123)) | | 是 |
|  | |  |  | |  |
| Response | | | | | |
| Response header | | | | | |
| 字段 | 取值 | | 描述 | 必选 | |
| Response body | | | | | |
| {  " errno":"0",  " errmsg":"",  } | | | | | |
| Example | | | | | |
| Linux curl命令：  curl -d "qid=800100&url=http:\/\/10.10.8.100:8080/gulfstream\/api\/v1\/raptor\/dTest  &type=1&mode=1&retry=3&timeout=3&params={'order\_id':100}"  "http://10.10.8.100:8080/bridgequeue/addTask"  {  "errno":"0",  "errmsg":"",  } | | | | | |
| PHP code: | | | | | |

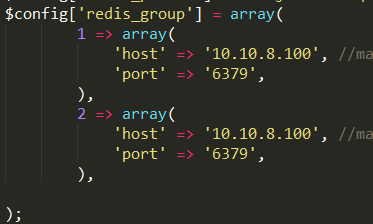
#### 查询队列任务数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口描述 | | **查询队列任务数** | | |
| Request | | | | |
| 线下：  Http Method：get  Host: http://10.10.8.100:8000/bridgequeue/getLength  线上：  Http Method：get  http://10.231.158.59:8000/bridgequeue/getLength | | | | |
| Request header | | | | |
| 字段 | | 取值 | 描述 | 必选 |
| Request param | | | | |
| 字段 | | 取值 | 描述 | 必选 |
| 无 | | 无 | 无 | 无 |
|  | |  |  |  |
| Request POST | | | | |
| 字段 | | 取值 | 描述 | 必选 |
| qid | | 整型 |  | 是 |
| type | | 整型 | 队列类型，1 普通队列 2 定时队列 | 是 |
| end\_time | | 整型 | 时间戳，当type=2时需要，如果是定时队列，是指获取以end\_time为截止时间的任务数 | 可选 |
| Response | | | | |
| Response header | | | | |
| 字段 | 取值 | | 描述 | 必选 |
| Response body | | | | |
| {  "erno":"0",  "errmsg":"",  "data":1  } | | | | |
| Example | | | | |
| Linux curl命令：  curl -G "http://10.10.8.100:8080/bridgequeue/getLength?qid=800200&type=1"  {  "code":"0",  "errmsg":"",  "data":1  } | | | | |
| PHP code: | | | | |

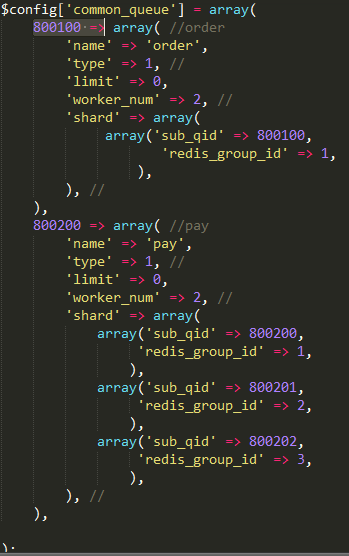
### 内部实现方案

在实现上，分流算法暂不实现，改为在配置文件里建立队列ID到redis组Id的映射关系，然后直接从配置文件读取，配置格式如下

Redis集群配置:



普通队列映射配置：



800100, 800200表示普通队列ID

name是队列别名

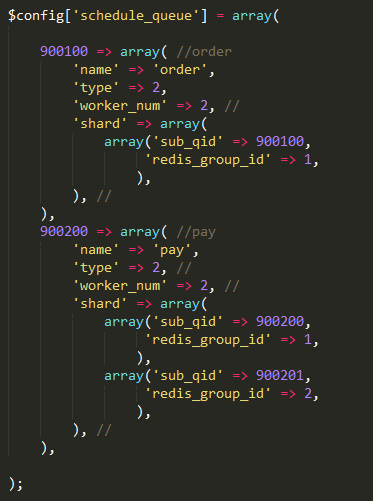
limit表示队列最多容纳多少个任务，到达限制后，每添加一个新任务就删除一个最老的任务。0表示没有限制。

worker\_num表示，负责处理此队列的worker子进程数量。

Shard表示，此队列分裂后的队列ID。表示子队列ID与redis组ID的映射。

redis\_group\_id表示，该队列所在redis的组ID。通过组ID定位redis主机。

定时队列映射配置：



#### 多组请求分流

一开始就部署多组redis主从集群，再根据配置，由client端实现请求分流。



#### 单队列请求分流



单队列扩容的基本思路是，把原来的QueueID映射为两个或多个QueueID(QueueID1, QueueID2, QueueID3…)，然后Client端根据新QueueID来做请求分流。

配置信息的shard字段有子队列ID与redis组ID的映射关系；添加任务到其中一个队列上，采用random策略；获取指定队列的任务长度，如果该队列已经扩容，长度是所有子队列的长度的总和。

## Worker重构设计



Worker方案

设计的核心是，Worker(不论普通队列，还是定时队列)的主进程根据业务配置来启动指定数量的子进程。一个主进程+众多子进程。

### 重载配置文件

单配置文件出现如下变化，worker需要柔性重启，给主进程发送信号SIGHUP。

1) 队列增加

2) worker\_num调整

3）redis增加主机IP+PORT

### 无损退出

为了保证进程退出时不丢失正在处理的任务，需要建立安全退出机制。Worker主进程和所有子进程都需要处理信号SIGTERM。

## Lib库设计

### 接口

配置文件：

config\_bridge\_queue.php



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口描述 | | **业务调用库接口** | | |
| 函数形式 | | | | |
| class BridgeQueueLib {  public function addTaskToQueue($qid, $data)  } | | | | |
| 函数参数 | | | | |
| 字段 | 取值 | | 描述 | 必选 |
| qid | 整型 | | 由队列系统分配的队列id，8开头表示普通队列，9开头表示定时队列 | 是 |
| data | 数组 | | data["qid"] 不设置，就设置传入参赛$qid；  data["type"] 不设置，加载配置文件的callback\_type；  data["url"] 不设置，加载配置文件的callback\_url；  data["mode"] 不设置，加载配置文件的callback\_mode；  data["retry"] 不设置，加载配置文件的callback\_retry；  data["timeout "] 不设置，加载配置文件的callback\_timeout；  data["expire"] 不设置，加载配置文件的callback\_ expire+time()；普通队列的过期时间  data["timestamp "] 不设置，加载配置文件的callback\_ timediff+time()；定时队列的执行时间，  data["params"] 为数组,是业务方回调时需要的参数 | 是 |
| 函数返回值 | | | | |
| bool ture:success, false:failed | | | | |
| PHP code: | | | | |