# Task | Open-Falcon

https://book.open-falcon.org/zh/install\_from\_src/task.html

# Task

task是监控系统一个必要的辅助模块。定时任务，实现了如下几个功能：

* index更新。包括图表索引的全量更新 和 垃圾索引清理。
* falcon服务组件的自身状态数据采集。定时任务了采集了transfer、graph、task这三个服务的内部状态数据。
* falcon自检控任务。

## 源码编译

我们提供了本模块的[release包](https://github.com/open-falcon/task/releases)，你可以直接下载并部署。或者，你也可以按照如下方式进行源码编译，

# update common lib

cd $GOPATH/src/github.com/open-falcon/common

git pull

# compile

cd $GOPATH/src/github.com/open-falcon/task

go get ./...

./control build

./control pack

最后一步会pack出一个tar.gz的安装包，拿着这个包去部署服务即可。

## 服务部署

服务部署，包括配置修改、启动服务、检验服务、停止服务等。这之前，需要将安装包解压到服务的部署目录下。

# 修改配置, 配置项含义见下文

mv cfg.example.json cfg.json

vim cfg.json

# 启动服务

./control start

# 校验服务,这里假定服务开启了8002的http监听端口。检验结果为ok表明服务正常启动。

curl -s "127.0.0.1:8002/health"

...

# 停止服务

./control stop

服务启动后，可以通过日志查看服务的运行状态，日志文件地址为./var/app.log。可以通过调试脚本./test/debug查看服务器的内部状态数据，如 运行 bash ./test/debug 可以得到服务器内部状态的统计信息。

## 配置说明

配置文件默认为./cfg.json。默认情况下，安装包会有一个cfg.example.json的配置文件示例。各配置项的含义，如下

debug: true/false, 如果为true，日志中会打印debug信息

http

- enable: true/false, 表示是否开启该http端口，该端口为控制端口，主要用来对task发送控制命令、统计命令、debug命令等

- listen: 表示http-server监听的端口

index

- enable: true/false, 表示是否开启索引更新任务

- dsn: 索引服务的MySQL的连接信息，默认用户名是root，密码为空，host为127.0.0.1，database为graph（如有必要，请修改）

- maxIdle: MySQL连接池配置，连接池允许的最大空闲连接数，保持默认即可

- cluster: 后端graph索引更新的定时任务描述。一条记录的形如: "graph地址:执行周期描述"，通过设置不同的执行周期，来实现负载在时间上的均衡。

eg. 后端部署了两个graph实例，cluster可以配置为

"cluster":{

"test.hostname01:6071" : "0 0 0 ? \* 0-5", //周0-5,每天的00:00:00,开始执行索引全量更新;"0 0 0 ? \* 0-5"为quartz表达式

"test.hostname02:6071" : "0 30 0 ? \* 0-5", //周0-5,每天的00:30:00,开始执行索引全量更新

}

- autoDelete: true|false, 是否自动删除垃圾索引。默认为false

collector

- enable: true/false, 表示是否开启falcon的自身状态采集任务

- destUrl: 监控数据的push地址,默认为本机的1988接口

- srcUrlFmt: 监控数据采集的url格式, %s将由机器名或域名替换

- cluster: falcon后端服务列表，用具体的"module,hostname:port"表示，module取值可以为graph、transfer、task等

## 补充说明

部署完成task组件后，请修改collector配置、使task能够正确采集transfer & graph的内部状态，请修改monitor配置、使task模块能够自检控Open-Falon的各组件(当前支持transfer、graph、query、judge等)。

### 关于自监控报警

因为多点监控的需求，自版本v0.0.10开始，我们将自监控报警功能 从Task模块移除。关于Open-Falcon自监控的详情，请参见[这里](http://book.open-falcon.org/zh/practice/monitor.html)。

### 关于过期索引清除

监控数据停止上报后，该数据对应的索引也会停止更新、变为过期索引。过期索引，影响视听，部分用户希望删除之。

我们原来的方案，是: 通过task模块，有数据上报的索引、每天被更新一次，7天未被更新的索引、清除之。但是，很多用户不能正确配置graph实例的http接口，导致正常上报的监控数据的索引 无法被更新；7天后，合法索引被task模块误删除。

为了解决上述问题，我们在默认配置里面，停掉了task模块自动删除过期索引的功能(autoDelete=false)；如果你确定配置的index.cluster正确无误，可以自行打开该功能。

当然，我们提供了更安全的、手动删除过期索引的方法。用户按需触发索引删除操作，具体步骤为:

1.进行一次索引数据的全量更新。方法为: 针对每个graph实例，运行curl -s "127.0.0.1:6071/index/updateAll"，异步地触发graph实例的索引全量更新(这里假设graph的http监听端口为6071)，等待所有的graph实例完成索引全量更新后 进行第2步操作。单个graph实例，索引全量更新的耗时，因counter数量、mysql数据库性能而不同，一般耗时不大于30min。

2.待索引全量更新完成后，发起过期索引删除 curl -s "$Hostname.Of.Task:$Http.Port/index/delete"。运行索引删除前，请务必**确保索引全量更新已完成**。

+