CSDN 博客 学院 下载 图文课 论坛 APP 问答 商城 VIP会员 活动 招聘 ITeye GitChat 搜博主文章 写博客 赚零钱 消息
关注和收藏在这里

分布式异步任务神器-Celery

年09月03日 06:52:49 清如許 阅读数：416

研究了下异步任务神器-Celery，发现非常好用，可以说是高可用，假如你发出一个任务执行命令给 Celery，只要 Celery 的执行单元 (worker) 行，那么它一定会执行；如果执行单元 (worker) 出现故障，如断电，断网情况下，只要执行单元 (worker) 恢复运行，那么它会继续执行你已经发出的命令。这一点有很强的实用：假如有交易系统接到了大量交易请求，主机却挂了，但前端用户仍可以继续发交易请求，发送交易请求后，用户无需等待。待主机恢复后，已发出的交易请求可以继续执行，过用户收到交易确认的时间延长而已，但并不影响用户体验。

lery 简介

一个异步任务调度工具，用户使用 Celery 产生任务，借用中间人来传递任务，任务执行单元从中间人那里消费任务。任务执行单元可以单机部署，也可以分布式部署，因此 ry 是一个高可用的生产者消费者模型的异步任务队列。你可以将你的任务交给 Celery 处理，也可以让 Celery 自动按 crontab 那样去自动调度任务，然后去做其他事情，你可以查看任务执行的状态，也可以让 Celery 执行完成后自动把执行结果告诉你。

用场景：

. 高并发的请求任务。互联网已经普及，人们的衣食住行中产生的交易都可以线上进行，这就避免不了某些时间极高的并发任务请求，如公司中常见的购买理财、学生缴费，在理财产品投放市场后、开学前的一段时间，交易量猛增，确认交易时间较长，此时可以把交易请求任务交给 Celery 去异步执行，执行完再将结果返回给用户。用户提交后不需要等待，任务完成后会通知到用户（购买成功或缴费成功），提高了网站的整体吞吐量和响应时间，几乎不需要增加硬件成本即可满足高并发。

. 定时任务。在云计算，大数据，集群等技术越来越普及，生产环境的机器也越来越多，定时任务是避免不了的，如果每台机器上运行着自己的 crontab 任务，管理起来相当麻烦，例如当进行灾备切换时，某些 crontab 任务可能需要单独手工调起，给运维人员造成极大的麻烦，有了 Celery ，你可以集中管理所有机器的定时任务，而且灾备无论何时切换，crontab 任务总能正确的执行。

. 异步任务。 一些耗时较长的操作，比如 I/O 操作，网络请求，可以交给 Celery 去异步执行，用户提交后可以做其他事情，当任务完成后将结果返回用户即可，可提高用户体验。

lery 的优点

. 纯 Python 编写，开源。这已经是站在巨人的肩膀上了，虽然 Celery 是由纯 Python 编写的，但协议可以用任何语言实现。迄今，已有 Ruby 实现的 RCelery 、node.js 实现的 node-celery 以及一个 PHP 客户端 ，语言互通也可以通过 using webhooks 实现。

. 灵活的配置。默认的配置已经满足绝大多数需求，因此你不需要编写配置文件基本就可以使用，当然如果有个性化地定制，你可以选择使用配置文件，也可以将配置写在源代码文件里。

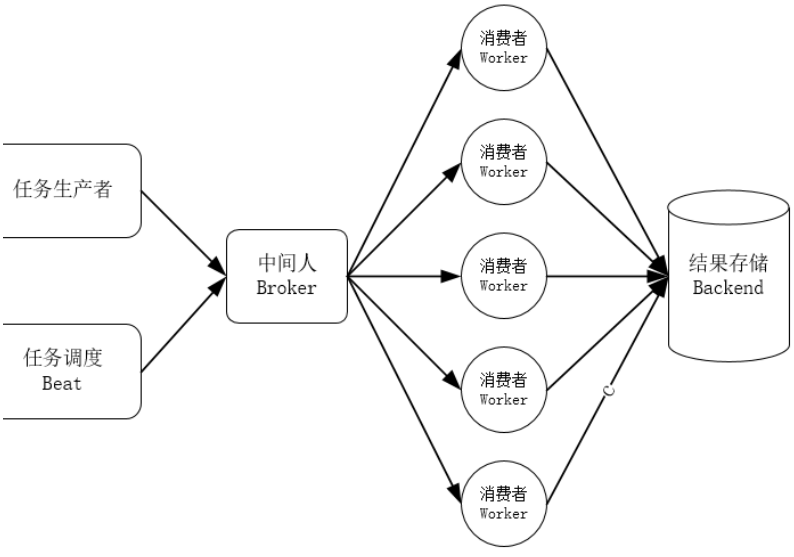
. 方便监控。任务的所有状态，均在你的掌握之下。

. 完善的错误处理。

. 灵活的任务队列和任务路由。你可以非常方便地将一个任务运行在你指定的队列上，这叫任务路由。

lery 的架构

一个工具，最好先从它的架构理解，辅以快速入门的代码来实践，最深入的就是阅读他的源码了，下图是 Celery 的架构图。



生产者： 调用Celery提供的API，函数，装饰器而产生任务并交给任务队列的都是任务生产者。

调度 **Beat**：Celery Beat进程会读取配置文件的内容，周期性的将配置中到期需要执行的任务发送给任务队列

0
2018 博客之星
关闭
VIP 免广告

人（**Broker**）：Celery 用消息通信，通常使用中间人（Broker）在客户端和 worker 之前传递，这个过程从客户端向队列添加消息开始，之后中间人把消息派送给 worker。官方的实现Broker的工具有：

名称	状态	监视	远程控制
RabbitMQ	稳定	是	是
Redis	稳定	是	是
Mongo DB	实验性	是	是
Beanstalk	实验性	否	否
Amazon SQS	实验性	否	否
Couch DB	实验性	否	否
Zookeeper	实验性	否	否
Django DB	实验性	否	否
SQLAlchemy	实验性	否	否
Iron MQ	第三方	否	否

际使用中我们选择 RabbitMQ 或 Redis 作为中间人即可。

单元 **worker**：worker 是任务执行单元，是属于任务队列的消费者，它持续地监控任务队列，当队列中有新地任务时，它便取出来 只要向同一个中间人即可，worker还可以监控一个或多个任务队列， Celery 是分布式任务队列的重要原因就在于 worker 可以分布在多 台er，它会自动生效。

结果存储**backend**：用来持久存储 Worker 执行任务的结果，Celery支持不同的方式存储任务的结果，包括AMQP，Redis，mem

lery 的使用示例：

ython3.6.5 版本为例。

安装 python 库：celery，redis。

```
pip install celery #安装celery
pip install celery[librabbitmq,redis,auth,msgpack] #安装celery对应的依赖
```

y其他的依赖包如下：
化：
y[auth]：使用auth序列化。
y[msgpack]：使用msgpack序列化。
y[yaml]：使用yaml序列化。
：
y[eventlet]：使用eventlet池。
y[gevent]：使用gevent池。
y[threads]：使用线程池。
和后端：
y[librabbitmq]：使用librabbitmq的C库.
y[redis]：使用Redis作为消息传输方式或结果后端。
y[mongodb]：使用MongoDB作为消息传输方式（实验性），或是结果后端（已支持）。
y[sqs]：使用AmazonSQS作为消息传输方式（实验性）。
y[memcache]：使用memcache作为结果后端。
y[cassandra]：使用ApacheCassandra作为结果后端。
y[couchdb]：使用CouchDB作为消息传输方式（实验性）。
y[couchbase]：使用CouchBase作为结果后端。
y[beanstalk]：使用Beanstalk作为消息传输方式（实验性）。
y[zookeeper]：使用Zookeeper作为消息传输方式。
y[zeromq]：使用ZeroMQ作为消息传输方式（实验性）。
y[sqlalchemy]：使用SQLAlchemy作为消息传输方式（实验性），或作为结果后端（已支持）。
y[pyro]：使用Pyro4消息传输方式（实验性）。
y[slmq]：使用SoftLayerMessageQueue传输（实验性）。

关闭

安装 Redis，以 ubuntu 操作系统为例（如果使用 RabbitMQ，自己装一下就可以）。

源码安装：

```
$ wget http://download.redis.io/releases/redis-4.0.11.tar.gz
$ tar xzf redis-4.0.11.tar.gz
$ cd redis-4.0.11
$ make
```

redis 配置文件 redis.conf，修改bind = 127.0.0.0.1为bind = 0.0.0.0，意思是允许远程访问redis数据库。

redis-server

```
$ cd src
$ ./redis-server ../redis.conf
```

育一个 **celery** 应用程序。

：模拟一个耗时操作，并打印 worker 所在机器的 IP 地址，中间人和结果存储都使用 redis 数据库。

```
#encoding=utf-8
#filename my_first_celery.py
from celery import Celery
import time
import socket

app = Celery("tasks", broker='redis://127.0.0.1:6379/0',backend ='redis://127.0.0.1:6379/0' )

def get_host_ip():
    """
    查询本机ip地址
    :return: ip
    """
    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        s.connect(('8.8.8.8', 80))
        ip = s.getsockname()[0]
    finally:
        s.close()
    return ip


@app.task
def add(x, y):
    time.sleep(3) # 模拟耗时操作
    s = x + y
    print("主机IP {}: x + y = {}".format(get_host_ip(),s))
    return s
```

这个 worker：

```
celery -A my_first_celery worker -l info
```

，-A 表示我们的程序的模块名称，worker 表示启动一个执行单元，-l 是批 -level，表示打印的日志级别。可以使用 celery –help 命令来查看celery命令的帮助文档。执行命令worker界面展示信息如下：

```
aaron@ubuntu:~/project$ celery -A my_first_celery worker -l info

 -------------- celery@ubuntu v4.2.1 (windowlicker)
---- **** -----
--- * ***  * -- Linux-4.10.0-37-generic-x86_64-with-Ubuntu-16.04-xenial 2018-08-27 22:46:00
-- * - **** ---
- ** ---------- [config]
- ** ---------- .> app:         tasks:0x7f1ce0747080
- ** ---------- .> transport:   redis://127.0.0.1:6379/0
- ** ---------- .> results:     redis://127.0.0.1:6379/0
- *** --- * --- .> concurrency: 1 (prefork)
-- ******* ---- .> task events: OFF (enable -E to monitor tasks in this worker)
--- ***** -----
 -------------- [queues]
                .> celery           exchange=celery(direct) key=celery


[tasks]
  . my_first_celery.add

[2018-08-27 22:46:00,726: INFO/MainProcess] Connected to redis://127.0.0.1:6379/0
[2018-08-27 22:46:00,780: INFO/MainProcess] mingle: searching for neighbors
[2018-08-27 22:46:02,075: INFO/MainProcess] mingle: all alone
[2018-08-27 22:46:02,125: INFO/MainProcess] celery@ubuntu ready.
```

相当清晰了。
你不想使用 celery 命令来启动 worker，可直接使用文件来驱动，修改 my_first_celery.py （增加入口函数main）

```
if __name__ == '__main__':
    app.start()
```

行

```
python my_first_celery.py worker
```

2018 博客之星 关闭
VIP 免广告

。

调用任务

y_first_celery.py 的同级目录下编写如下脚本 start_task.py如下。

```
from my_first_celery import add #导入我们的任务函数add
import time
result = add.delay(12,12) #异步调用，这一步不会阻塞，程序会立即往下运行

while not result.ready():# 循环检查任务是否执行完毕
    print(time.strftime("%H:%M:%S"))
    time.sleep(1)

print(result.get()) #获取任务的返回结果
print(result.successful()) #判断任务是否成功执行
```

```
python start_task.py
```

如下所示：

```
22:50:59
22:51:00
22:51:01
24
True
```

等待了大约3秒钟后，任务返回了结果24，并且是成功完成，此时worker界面增加的信息如下：

```
[2018-08-27 22:50:58,840: INFO/MainProcess] Received task: my_first_celery.add[a0c4bb6b-17af-474c-9eab-407d593a7807]
[2018-08-27 22:51:01,898: WARNING/ForkPoolWorker-1] 主机IP 192.168.195.128: x + y = 24
[2018-08-27 22:51:01,915: INFO/ForkPoolWorker-1] Task my_first_celery.add[a0c4bb6b-17af-474c-9eab-407d593a7807] succeeded in 3.067237992000173s: 24
```

的信息非常详细，其中a0c4bb6b-17af-474c-9eab-407d593a7807是taskid，只要指定了 backend，根据这个 taskid 可以随时去 backend 去查找运行结果，使用方法如下：

```
>>> from my_first_celery import add
>>> taskid= 'a0c4bb6b-17af-474c-9eab-407d593a7807'
>>> add.AsyncResult(taskid).get()
24
>>>#或者
>>> from celery.result import AsyncResult
>>> AsyncResult(taskid).get()
24
```

说明：如果想远程执行 worker 机器上的作业，请将 my_first_celery.py 和 start_tasks.py 复制到远程主机上（需要安装
y)，修改 my_first_celery.py 指向同一个中间人和结果存储，再执行 start_tasks.py 即可远程执行 worker 机器上的作业。my_first_celery.add函数的代码不是必须的，你也要以这
用任务：

```
from my_first_celery import app
app.send_task("my_first_celery.add",args=(1,3))
```

将一个 celery 项目

产环境中往往有大量的任务需要调度，单独一个文件是不方便的，celery 当然支持模块化的结构，我这里写了一个用于学习的 Celery 小型工程项目，含有队列操作，任务调
操作，目录树如下所示：

```
eleryProj
 app.py
 __init__.py
 readme.txt
 settings.py
 tasks.py
```

2018 博客之星
关闭

init.py是空文件，目的是告诉 Python myCeleryProj 是一个可导入的包.
py

```
from celery import Celery

app = Celery("myCeleryProj", include=["myCeleryProj.tasks"])

app.config_from_object("myCeleryProj.settings")

if __name__ == "__main__":
    app.start()
```

VIP 免广告

ıgs.py

```python
from kombu import Queue
import re
from datetime import timedelta
from celery.schedules import crontab


CELERY_QUEUES = (  # 定义任务队列
    Queue("default", routing_key="task.#"),  # 路由键以“task.”开头的消息都进default队列
    Queue("tasks_A", routing_key="A.#"),  # 路由键以“A.”开头的消息都进tasks_A队列
    Queue("tasks_B", routing_key="B.#"),  # 路由键以“B.”开头的消息都进tasks_B队列
)

CELERY_TASK_DEFAULT_QUEUE = "default"  # 设置默认队列名为 default
CELERY_TASK_DEFAULT_EXCHANGE = "tasks"
CELERY_TASK_DEFAULT_EXCHANGE_TYPE = "topic"
CELERY_TASK_DEFAULT_ROUTING_KEY = "task.default"

CELERY_ROUTES = (
    [
        (
            re.compile(r"myCeleryProj\.tasks\.(taskA|taskB)"),
            {"queue": "tasks_A", "routing_key": "A.import"},
        ),  # 将tasks模块中的taskA,taskB分配至队列 tasks_A ,支持正则表达式
        (
            "myCeleryProj.tasks.add",
            {"queue": "default", "routing_key": "task.default"},
        ),  # 将tasks模块中的add任务分配至队列 default
    ],
)


# CELERY_ROUTES = (
#   [
#       ("myCeleryProj.tasks.*", {"queue": "default"}), # 将tasks模块中的所有任务分配至队列 default
#   ],
# )

# CELERY_ROUTES = (
#   [
#       ("myCeleryProj.tasks.add", {"queue": "default"}), # 将add任务分配至队列 default
#       ("myCeleryProj.tasks.taskA", {"queue": "tasks_A"}),# 将taskA任务分配至队列 tasks_A
#       ("myCeleryProj.tasks.taskB", {"queue": "tasks_B"}),# 将taskB任务分配至队列 tasks_B
#   ],
# )

BROKER_URL = "redis://127.0.0.1:6379/0"  # 使用redis 作为消息代理

CELERY_RESULT_BACKEND = "redis://127.0.0.1:6379/0"  # 任务结果存在Redis

CELERY_RESULT_SERIALIZER = "json"  # 读取任务结果一般性能要求不高，所以使用了可读性更好的JSON

CELERY_TASK_RESULT_EXPIRES = 60 * 60 * 24  # 任务过期时间，不建议直接写86400，应该让这样的magic数字表述更明显


CELERYBEAT_SCHEDULE = {
    "add": {
        "task": "myCeleryProj.tasks.add",
        "schedule": timedelta(seconds=10),
        "args": (10, 16),
    },
    "taskA": {
        "task": "myCeleryProj.tasks.taskA",
        "schedule": crontab(hour=21, minute=10),
    },
    "taskB": {
        "task": "myCeleryProj.tasks.taskB",
        "schedule": crontab(hour=21, minute=12),
    },
}
```

2019 博客之星
关闭

s.py

```python
import os
from myCeleryProj.app import app
import time
import socket


def get_host_ip():
    """
```

VIP 免广告

```
    查询本机ip地址
    :return: ip
    """
        try:
            s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
            s.connect(("8.8.8.8", 80))
            ip = s.getsockname()[0]
        finally:
            s.close()
        return ip


@app.task
def add(x, y):
    s = x + y
    time.sleep(3)  # 模拟耗时操作
    print("主机IP {}: x + y = {}".format(get_host_ip(), s))
    return s


@app.task
def taskA():
    print("taskA begin...")
    print(f"主机IP {get_host_ip()}")
    time.sleep(3)
    print("taskA done.")


@app.task
def taskB():
    print("taskB begin...")
    print(f"主机IP {get_host_ip()}")
    time.sleep(3)
    print("taskB done.")
```

me.txt

```
#启动 worker
#分别在三个终端窗口启动三个队列的worker，执行命令如下所示：
celery -A myCeleryProj.app worker -Q default -l info
celery -A myCeleryProj.app worker -Q tasks_A -l info
celery -A myCeleryProj.app worker -Q tasks_B -l info
#当然也可以一次启动多个队列，如下则表示一次启动两个队列tasks_A，tasks_B。
celery -A myCeleryProj.app worker -Q tasks_A,tasks_B -l info
#则表示一次启动两个队列tasks_A，tasks_B。
#最后我们再开启一个窗口来调用task: 注意观察worker界面的输出
>>> from myCeleryProj.tasks import *
>>> add.delay(4,5);taskA.delay();taskB.delay() #同时发起三个任务
<AsyncResult: 21408d7b-750d-4c88-9929-fee36b2f4474>
<AsyncResult: 737b9502-77b7-47a6-8182-8e91defb46e6>
<AsyncResult: 69b07d94-be8b-453d-9200-12b37a1ca5ab>
#也可以使用下面的方法调用task
>>> from myCeleryProj.app import app
>>> app.send_task(myCeleryProj.tasks.add,args=(4,5)
>>> app.send_task(myCeleryProj.tasks.taskA)
>>> app.send_task(myCeleryProj.tasks.taskB)
```

)

需要请关注微信公众号 somenzz，回复 celery 下载源码。

2019 博客之星
关闭
VIP 免广告
收藏 分享

k中celery介绍及使用celery实现异步任务

1124

资料： Celery 官网：http://www.celeryproject.org/ Celery 官方文档英文版：http://docs.celeryproject.org/en/late...

来自： weixin_40612082

想对作者说点什么

celery异步处理以一个查询任务 1012
个绍通过celery实现一个异步任务。有这样一个需求，前端发起一个查询的请求，但是发起查询后… 来自： spur_man的博客

ery+python+redis异步执行定时任务 1.4万
y+python+redis异步执行定时任务我之前的一篇文章中写了【Celery+django+redis异步执行任务… 来自： AFei0018-博客

任务神器 Celery 简明笔记 305
任务 异步任务是web开发中一个很常见的方法。对于一些耗时耗资源的操作，往往从主应用中隔… 来自： IAlexanderI的专栏

广告 一插上电,50平米内都暖和了!3天一度电,今日特惠!
优诺 · 顶新

asticsearch遇到的bug以及python对接elasticsearch 140
错误 Could not find any executable java binary. Please install java in your PATH or set JAVA_… 来自： wang785994599的博客

ery 异步分布式 171219 558
是Python开发的分布式异步任务调度系统，Celery支持的消息服务有rmq、redis等 Celery 是一… 来自： yanyangjie的专栏

ion使用celery实现异步任务执行 765
celery在django项目中实现异步发送短信 在项目的目录下创建celery_tasks用于保存celery异步任… 来自： gllaxq的博客

了解celery-分布式异步任务调度系统 2682
是Python开发的分布式异步任务调度系统，Celery支持的消息服务有rmq、redis等 以下代码使… 来自： phypor的博客 ®

ngo配置celery执行异步任务和定时任务 2071
演示均基于Django2.0 celery是一个基于python开发的简单、灵活且可靠的分布式任务队列框架，… 来自： 运维咖啡吧

广告 一插上电,50平米内都暖和了!3天一度电,今日特惠!
优诺 · 顶新

ry配合rabbitmq任务队列实现任务的异步调度执行 4627
这东西在任务调度方面，很有一套的，学习他有段时间了，自己也试图在项目中用，但苦于没… 来自： jazywoo_在路上

热词
on分布式爬取网数据配置 Python分布式爬取网数据实例 Python分布式爬取网数据介绍 Python分布式爬取网数据部署 Python分布式爬取网数据

热词
布式 bootstrap3.0的布局神器 android异步任务强制结束 c#什么是分布式 c++ 分布式队列系统 人工智能学习神器 人工智能学习神器多少

ython下比celery更加简单的异步任务队列RQ 7217
/ 编程 / 2013-8-27 19:33 / 阅读: 21 Celery是Python开发的分布式任务调度模块，今天抽空看… 来自： 测者陈磊

范桂飓 333篇文章 排名:1000+
self-motivation 222篇文章 关注 排名:3000+
武坤 47篇文章 关注 排名:千里之外
xsj_blog 270篇文章 关注 排名:5000+

k+celery常见问题及解决方法 6261
oot@ansible flask_celery]# celery -A app worker --loglevel=info Traceback (most recent call l... 来自： lixingdefengzi的专栏

redis又比celery更加简单的异步任务队列RQ 2485
这里介绍一个python下，比celery更加简单的异步工具，真的是很简单，当然他的功能没有c… 来自： permike的专栏

式任务队列 Celery 795
目录 前言 简介 Celery 的应用场景 架构组成 Celery 应用基础 前言 分布式任务队列 Celery，Pyt… 来自： 范桂飓（烟云的计算）

发现了一个免费的云服务器,号称是永久的
百度广告

ngo 使用celery做异步任务处理 703
/ 分布式任务框架,处理异步的 安装 sudo apt-get/yum install rabbitmq-server（使用redis：redis-… 来自： zelinhehe的专栏

任务神器 Celery 快速入门 646
程序运行过程中，要执行一个很久的任务，但是我们又不想主程序被阻塞，常见的方法是多线… 来自： 一个程序员的成长之…

2018 博客之星 关闭
VIP 免广告

任务神器 Celery 236
y 在程序的运行过程中，我们经常会碰到一些耗时耗资源的操作，为了避免它们阻塞主程序的运… 来自： 3569

任务神器 Celery 快速入门教程 249
y在程序的运行过程中，我们经常会碰到一些耗时耗资源的操作，为了避免它们阻塞主程序的运… 来自： 张昆的博客

使用celery异步任务 436
使用celery异步任务 1. 关于celery Celery是基于Python开发的一个分布式任务队列框架，支持使… 来自： wang725的专栏


陈小春坦言：这游戏不充钱都能当全服大哥，找到充值入口算我输！
贪玩游戏 · 顶新

ion Django Celery 实现异步任务 1566
上一篇Python Django 实现restful API ，本次目的是为了实现异步任务 先从需求说起 接口实现之… 来自： xie_0723的博客

igo —— Celery实现异步和定时任务 2485
o celery 实现异步和定时任务 来自： 业余的博客

式消息队列 Celery 的最佳实践 2649
目录 不使用数据库作为 Broker 不要过分关注任务结果 实现优先级任务 应用 Worker 并发池的动… 来自： 范桂飓（烟云的计算）

igo通过 Celery添加异步任务 218
异步任务的重要性 大家在做web项目的时候经常会遇到一些耗时的操作， 比如： 发送邮件、发… 来自： zelinhehe的专栏

lask 来写个轻博客 (26) — 使用 Flask-Celery-Helper 实现异步任务 7828
项目源码：https://github.com/JmilkFan/JmilkFan-s-Blog 目录 目录 前文列表 扩展阅读 Celery 将… 来自： 范桂飓（烟云的计算）


陈小春坦言：这游戏不充钱都能当全服大哥，找到充值入口算我输！
贪玩游戏 · 顶新

ion定时任务/异步任务工具celery与Scheduler 813
n定时任务/异步任务工具celery与Scheduler python定时任务/异步任务工具celery与Scheduler 1. … 来自： AyoCross的专栏

igo基于celery的异步任务实现 575
Celery 依赖包： celery 、 django-celery增加celery 配置， 主要是配置好工程的名称 在settings.p… 来自： chengfangang的专栏

在 Dango 中使用 Celery 配置异步执行与定时任务 156
ango 中使用 Celery 各位小伙伴注意，Celery大于3.1.25的版本不再支持Windows。Windows下… 来自： Dolphin's Blog

ion 并行分布式框架 Celery 2.7万
ery 官网：http://www.celeryproject.org Celery 官方文档英文版：http://docs.celeryproject.org/en/l… 来自： freeking101的博客

ion 爬虫实践(分布式部署) 219
个绍 celery分布式框架是我接触的第一个分布式框架，现学现卖，将我上一篇博客介绍的爬虫，… 来自： caca95的博客


真钻镶嵌,一表两戴!上海牌六十周年纪念机械表售完即止!
上海牌手表 · 顶新

ry 分布式框架详解 1178
y 结构图如果没有celery，让你自己设计一个异步任务队列你怎么做。首先，要有一个发起任务… 来自： 一个程序员的成长之…

式任务队列与任务调度系统Celery进阶——分布式爬虫 2372
文件crawlertask.py，用于执行数据抓取任务，代码如下。 #coding:utf-8 from celery import Celer… 来自： 帮助别人等于帮助自…

ly：高性能异步分布式事务TCC框架 201
来自： 程序猿DD

igo框架上使用Celery异步执行任务 37
Django框架编程时， 由于框架的封装限制，在没有编写底层代码的情况下，要把一段代码放到一… 来自： Best_fish的博客

igo+Celery+xadmin实现异步任务和定时任务 269
go+Celery+xadmin实现异步任务和定时任务 一、celery介绍 1、简介 【官网】http://www.celeryp… 来自： 梁某


真钻镶嵌,一表两戴!上海牌六十周年纪念机械表售完即止!
上海牌手表 · 顶新

igo中异步任务celery 1051
go中异步任务celery 本文主要讲述怎么在Django中应用Celery来完成异步任务. 最近公司在做支… 来自： niekunhit的博客

2018 博客之星
关闭
VIP 免广告

ıgo通过celery添加异步任务 0 533
o通过celery添加异步任务 0 来自： 王冠hurt的博客

Celery Once 来防止 Celery 重复执行同一个任务 124
Celery Once 来防止 Celery 重复执行同一个任务 在使用 Celery 的时候发现有的时候 Celery 会… 来自： 是什么让你停止不前?

ry占用大内存问题记录 557
worker占用大内存记录 定位 原因 celery worker占用大内存记录 定位 命令参考 top命令下，M… 来自： bbhe_work的博客

ry 分布式实现 250
分布式实现 原理很简单。 当你讲任务队列rabbitmq 或redis启动后。再启动celery的work程序… 来自： ding1991as的博客


居住证积分专栏
百度广告

ıon 并行分布式框架 Celery 详解 227
y 官网：http://www.celeryproject.org/ Celery 官方文档英文版：http://docs.celeryproject.org/en/l… 来自： cuomer的博客

ıon 并行分布式框架：Celery 超详细介绍 2.5万
python 并行分布式 框架 超详细 来自： liuxiaochen123的专栏

ython】基于Celery的分布式应用 2494
安装Celery pip install celery 编写Task-Func#-*-coding:utf-8-*- #download.pyfrom celery import C… 来自： ns2250225

rnado中使用celery实现异步任务处理之二 1897
ornado结合RabbitMQ实现异步任务处理 3.1 安装环境 1. 安装tornado 见文章《CentOS6.4安装p… 来自： 北雨南萍

ıgo中使用django-celery完成异步任务(1) 372
Django应用需要执行异步任务, 以便不耽误http request的执行. 我们也可以选择许多方法来完成异… 来自： Demo_3的博客


发现了一个免费的云服务器,号称是永久的
百度广告

ıgo完成异步工具——celery 809
用户发起request，并等待response返回。在本些views中，可能需要执行一段耗时的程序，那… 来自： 幸福清风：人生苦短…

ıgo中的celery异步任务 42
go中使用celery创建异步队列 创建celery任务 - 安装celery必不可少 pip install celery – 如果报错… 来自： passion

o与异步任务神器celery集成 1728
地址：https://github.com/sunliang1163/celery-odoo 背景：公司ODOO运行一年多，随着业务不… 来自： 好修养ODOO团队

ry分布式 1574
y的分布式实际包含两个层次： Distribute work on a given machine across all CPUs Distribute w… 来自： 一个程序员的成长之…

ıgo中celery执行任务结果的保存 648
nstall django-celery-resultsINSTALLED_APPS = (..., 'django_celery_results',) # 注意这个是下… 来自： Areigninhell的博客


爆款取暖器，一插电50平米都暖和了，3天1度电，立即抢购!
淘缘鑫 · 顶新

ery：分布式任务队列 简单上手 1126
介绍 Celery 的基本用法。适合上手。 内容纲要： 0.概念。 1.选择并安装 Broker（消息传送）。 … 来自： zelinhehe的专栏

gl第五课-绘制一个点的另外一种写法 5398
源码可以Q群：828202939 或者点击这里 希望可以和大家一起学习、一起进步！！ 纯手打！！ … 来自： 谷子的博客

ıstorm 最新激活码 多种破解方式(持续更新...) 122463
License server 注册 安装完成，打开Webstorm，在弹出的License Activation窗口中选择"Licen… 来自： 老妖儿的博客

gl第九课-绘制多个顶点 5465
果我们说到一些绘制多个顶点的基础知识，这节课我们来看看绘制多个顶点的代码是怎么实现的 … 来自： 谷子的博客

gl第12课-图形的变换之平移 5579
我们学习了一些点、三角形、矩形等一些基础的图形的绘制的方法 本节课开始我们开始学习对图… 来自： 谷子的博客

关闭

VIP 免广告

:信息隐藏的世界，全面讲解信息隐藏——第1节：信息隐藏技术简介 3105
言息隐藏的世界，全面讲解信息隐藏——第1节：信息隐藏技术简介 专栏题记：奥斯卡优秀电影… 来自： qq_26464039的博客

gl第六课-通过鼠标点击绘点 5749
原码可以Q群：828202939 或者点击这里 希望可以和大家一起学习、一起进步！！纯手打！！ … 来自： 谷子的博客

:现某东移动web轮播图 926
web轮播图 参考实例，可以从中保存轮播图的图片，这里我们放入uploads的文件夹下，将八张图… 来自： weixin_41105030的…

storm 2018 激活破解方法大全 735456
torm 作为最近最火的前端开发工具,也确实对得起那个价格,但是秉着勤俭节约的传统美德,我们肯… 来自： 唐大帅的编程之路

.最简单的 SpringCloud 教程 | 终章 1295248
请标明出处： http://blog.csdn.net/forezp/article/details/70148833 本文出自方志朋的博客 错过了… 来自： 方志朋的专栏

gl第13课-图形变换之旋转 5122
节课我们学习了图形的变换之平移 这一节课我们将学习图形的变换之旋转 如果你学会了图形的平… 来自： 谷子的博客

harm最新2019激活码 1357044
呈对jetbrains全系列可用例：IDEA、WebStorm、phpstorm、clion等 因公司的需求，需要做一个… 来自： 昌昌

gl第七课-鼠标分象限绘制不同颜色的点 5491
原码可以Q群：828202939 或者点击这里 希望可以和大家一起学习、一起进步！！纯手打！！ … 来自： 谷子的博客

:理收集】那些神器级别的BT磁力搜索网站 97687
我喜欢和常用的网站我会特别介绍一下的，其他大家自己看着办吧。 还有一点，一般专门的BT论… 来自： roslei的博客

eJS后处理-星空 20199
电子档书籍或者源码可以Q群：828202939 希望可以和大家一起学习、一起进步！！ 如有错别… 来自： 谷子的博客

，不再成为2018！ 4549
2019年1月4号，上午9：57。 突然看到了新年flag征文活动，不妨一起来展望一下吧。 不做积… 来自： 青春不作伴

ell6 中文不限时版下载(免密匙)（笔记） 90558
ll6免费版 下载 Xshell6下载链接：原有的资源链接csdn积分自调整太高了，没办法降。这边给你… 来自： qq_31362105的博客

了10个干净、好用的BT、磁力链搜索网站给大家 252757
越来越流行在线看视频了，但是对于我得收藏癖爱好者，还是希望可以有比较好的资源网站的，… 来自： YXAPP的技术分享

缩招，AI跻身2019年最赚钱职业榜首！（附薪酬表） 13710
年底，互联网正在经历寒冬，不少公司出现了裁员新闻，也有很多人纷纷转型、跳槽。那么，201… 来自： CSDN学院

tman 使用方法详解 216110
Postman背景介绍 用户在开发或者调试网络程序或者是网页B/S模式的程序的时候是需要一些方… 来自： fxbin123的博客

个字符串的前缀与另一个字符串的后缀的最大相同子串 994
符串ptr的前缀与str的后缀的最大相同子串，若不存在，输出0。 样例输入 mike aniom kiava dvak… 来自： 紫芝的博客

说设计模式——外观（门面）模式 19949
请出今天的主人公——“黑旋风”李逵 李逵：“我是不是萌萌哒？”一部《水浒传》说尽了一群英雄… 来自： 青衣煮茶

）MyBatis核心组件（配图详解＆代码实现） 32115
atis的核心组件分为4个部分 SqlSessionFactoryBuilder（构造器）：根据xml或java代码生成SqlS… 来自： 青衣煮茶

病怎么能治好，看我用Python对接 98240
见代社会中年轻人常见病例，很显然“颈椎病”一定是排在第一的。年轻人长期伏案工作、长期面对… 来自： john_dung的博客

理论课答案（西安交大版） 1238278
【单选题】我国陆地领土面积排名世界第几？（C） A、1 B、2 C、3 D、4 2 【单选题】以下哪… 来自： ling_wang的博客

e2016永久免费激活码（office2016密钥） 823011
soft Toolkit(Win10激活工具/Office2016激活工具) V2.6B4 绿色版人气:42008 下载 Microsoft Tool… 来自： 老K的博客

peg详解及常用命令使用 41928
PEG简介 FFMPEG堪称自由软件中最完备的一套多媒体支持库，它几乎实现了所有当下常见的数… 来自： qq_26464039的博客

最好用百度云破解版，百度网盘不限速下载，教你如何解决百度网盘限速的方法。亲… 250803
网盘不限速 点击下载 提取码：jsk0 百度网盘不限速 点击下载 提取码：jsk0 对于大多数人来说，… 来自： qq_41925894的博客

2018 博客之星 关闭
VIP 免广告

gl第15课-矩阵变换之平移 5552
市课我们学习了矩阵的变换之旋转 这一节课我们将学习矩阵的变换之平移 在这之前，你得了解前… 来自： 谷子的博客

gl第19课-动画基础-旋转 3245
市课我们学习了高级变换--图形的复合变换 这一节课我们将学习动画的基础 ... 来自： 谷子的博客

搭建自己的风格迁移 2473
环境，我用的是ubuntu16.04. 安装TensorFlow，这个按照官网的指导一步步来就可以，需要注意… 来自： 岁月静好

框架NLog简单配置使用 52884
日志管理工具 一、获得NLog 这里介绍最简单的获得方式 1.管理NuGet程序包 2.在打开页面中搜… 来自： Maybe_ch的博客

最全Java面试题（带全部答案） 153912
要谈的主题是关于求职，求职是在每个技术人员的生涯中都要经历多次。对于我们大部分人而言… 来自： 林老师带你学编程

部分展示-3D在线试衣系统解决方案 25217
声明：发布此博客纯属技术展示和交流！未得本人同意，禁止转载！禁止商业目的！ 需要电子档… 来自： 谷子的博客

NTERS ON C【C和指针】 909
ude<iostream> #include<algorithm> #include<cstdio&… 来自： 紫芝的博客

和激活Office 2019 73159
牛请支持正版！相比费尽力气找一个可能不太安全的激活工具，直接买随时随地更新的Office 365… 来自： 过了即是客

的实参和形参、作为值的函数 671
的实参和形参 JavaScript中的函数定义并未指定函数形参的类型，函数调用也未对传入的实参值… 来自： wuyufa1994的博客

员上班没事做该怎么办 2082
作为一名程序员，工作强度不稳定是比较正常的，忙的时候会埋怨，闲的时候会发慌。合理的安… 来自： AikesLs的博客

o2016下载激活破解 55107
载Visio2016 可以在官网下载，也可以在我的网盘下载 链接：https://pan.baidu.com/s/1Mre… 来自： 天狼星的博客

gl第14课-矩阵变换之旋转 5743
市课我们学习了图形的变换之旋转 这一节课我们将学习矩阵的变换之旋转 对于简单的变换，我们… 来自： 谷子的博客

gl第16课-矩阵变换之缩放 5264
市课我们学习了矩阵的变换之平移 这一节课我们将学习矩阵的变换之缩放 在这之前，你得了解前… 来自： 谷子的博客


清如許 关注

原创	粉丝	喜欢	评论
41	11	4	6

等级： 博客 3 访问： 1万+
积分： 564 排名： 11万+
勋章：

最新文章

Python-排序-归并排序中如何用哨兵来追求极致的性能?
Python 排序-插入排序-优化
Python-排序-选择排序-优化
Python-排序-冒泡排序-优化
工作后，为什么还要学习数据结构与算法

关闭

VIP 免广告

个人分类

技术 32篇
运维 11篇
python 17篇
邮件 1篇
微信 1篇
展开

归档

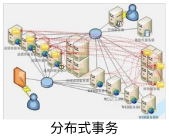
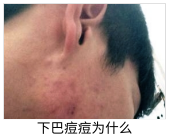
2019年1月 1篇
2018年12月 9篇
2018年11月 3篇
2018年10月 9篇
2018年9月 7篇
展开

热门文章

解决 Ubuntu 18.04 无法关机的问题
阅读量：2469
任务调度神器 airflow 之初体验
阅读量：1493
airflow 安装，部署，填坑
阅读量：1253
Python 多线程操作
阅读量：924
10 分钟搭建一个超好用的 CMDB 系统
阅读量：868

最新评论

工作后，为什么还要学习数据结构与算法
java_zzzz：一起进步吧[reply]somezz[/reply]
工作后，为什么还要学习数据结构与算法
somezz：[reply]java_zzzz[/reply] 我也是菜，在不断输入
运维必备--如何彻底解决数据库的锁...
java_zzzz：大神 可以加个好友吗
工作后，为什么还要学习数据结构与算法
java_zzzz：大佬
Python-排序-冒泡排序-优化
liuzhixiong_521：感谢分享，学习了


分布式事务

下巴痘痘为什么

样板房设计

白丝
广告

2018 博客之星
关闭

联系我们

微信客服 QQ客服
QQ客服 kefu@csdn.net
客服论坛 400-660-0108
工作时间 8:30-22:00

关于我们 | 招聘 | 广告服务 | 网站地图
百度提供站内搜索 京ICP证09002463号
©1999-2019 江苏乐知网络技术有限公司
江苏知之为计算机有限公司 北京创新乐知信息技术有限公司版权所有

网络110报警服务 经营性网站备案信息
北京互联网违法和不良信息举报中心
中国互联网举报中心

关闭

VIP
免广告