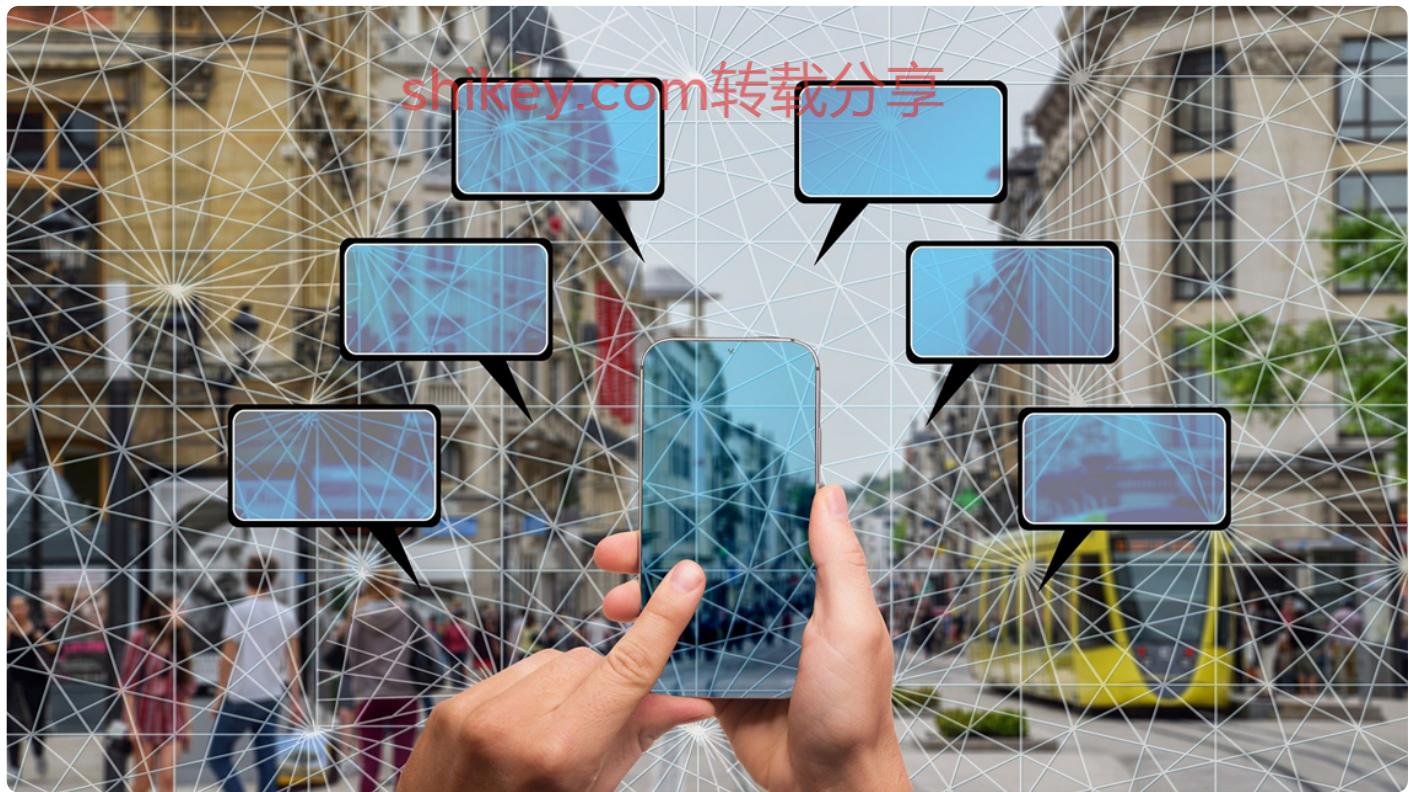


27 | 部署：如何在Linux上配合定时任务部署推荐系统服务？

2023-06-16 黄鸿波 来自北京

《手把手带你搭建推荐系统》



你好，我是黄鸿波。

在前面的课程中，我们对推荐系统的数据获取、数据处理、规则召回、模型召回、排序、重排序都做了比较全面地讲解，可以说，用现有的知识已经能够进行一套企业级的推荐系统开发了。接下来就是推荐系统进行工程化部署，这里就会引入 Linux 部分的知识。

我把本节课分为了以下三大模块。

1. 什么是 Linux 系统？它的优势是什么？
2. Linux 系统中的定时任务——Crontab。
3. 如何把推荐系统项目部署到 Linux 中。

Linux 系统概述

Linux 采用 GNU 通用公共许可证的条款，意味着它可以自由使用、复制、修改和分发。

Linux 系统基于类 Unix 操作系统模式和结构，由许多开源的软件组成，以其优秀的网络性能、稳定性、灵活性、支持多用户和多任务管理等特点而被广泛使用。

先来整体了解一下 Linux 的各大优点。

shikey.com转载分享

开源性。 Linux 系统的源代码是公开的，意味着任何人都可以查看代码和进行修改。这样开发者可以轻松地对系统进行二次开发和优化，从而满足不同用户的需求和要求。此外，用户还可以自由选择他们喜欢的软件应用程序，从而更好地满足其需求和兴趣。

安全性。 由于其开源的特点，Linux 系统拥有更多的安全升级和修复。开发者们不断升级和维护 Linux 系统的安全性，因此用户可以更安全地使用和存储数据。虽然 Linux 和 Windows 都可以作为服务器来进行项目的部署，但是相比于 Windows 系统而言，Linux 系统更加适合于作为服务器使用。

稳定性。 相比于 Windows 系统的频繁升级和更新，Linux 系统更为稳定，通常不需要经常重启系统。这可以有效提高系统的运行效率和稳定性，让用户更安心地使用。

兼容性。 Linux 系统对多种硬件设备的支持比 Windows 系统更广泛。因此，用户不必担心系统和硬件设备不兼容的问题。

自定义。 Linux 系统具有高度的自定义能力，可以根据不同用户的需要进行自由定制和配置。用户可以自行配置系统环境、自由选择软件工具等，从而更好地满足自己的需求和兴趣。

网络性能。 Linux 系统的网络性能非常优秀，它可以支持多个网络协议和多种网络设备，提供高速和高稳定的网络连接，这对于服务器来说非常必要。

在服务器领域，使用 Linux 系统可以保证服务器的稳定运行和高效性能，受到广泛的应用和认可。

Linux系统的优势

| | |
|------|-------------------------------|
| 开源性 | 开发者可以轻松地对系统进行二次开发和优化 |
| 安全性 | 用户可以更安全地使用和存储数据 |
| 稳定性 | 系统更为稳定，通常不需要经常重启 |
| 兼容性 | 用户不必担心系统和硬件设备不兼容 |
| 自定义 | 用户可以自行配置系统环境、自由选择软件工具 |
| 网络性能 | 支持多个网络协议和多种网络设备，提供高速和高稳定的网络连接 |



Linux 系统中的定时任务——Crontab

简单看过 Linux 的特性之后，接下来我们来讲解 Linux 部署时最常用到的一个命令——Crontab，这个命令主要的功能就是做定时任务。

Crontab 是 Linux 系统中用于管理周期性定时任务的工具，其服务进程名称为 Crond（即周期任务的英文缩写）。通常情况下，Linux 系统安装完成后已默认开启 Crond 服务，而 Crontab 则用于呈现 Crond 任务的任务列表。使用 Crontab 可以很方便地实现定时备份数据、清理日志、重启自动命令或者定时执行自定义脚本等任务。

想一下，在我们的这套推荐系统中，哪里需要用到定时任务呢？

如果最开始想要获得源源不断的的数据，就需要每隔一段时间来运行一次爬虫程序，所以爬虫是一个定时任务的强需求。

接下来，应该及时把爬虫爬到的内容变成内容画像的一部分，但爬虫和数据处理是两个项目，因此在这里也需要一个单独的定时任务。

做完数据处理之后，需要定时去更新这个召回集的列表，这样才有助于内容保持及时更新。基于机器学习和基于深度学习的召回模型在数据量比较大的时候，也需要用定时任务来做。

shikey.com转载分享

接下来我们来看看如何使用 Crontab 命令来做定时任务。

在 Linux 中使用下面的命令来写 Crontab 命令。

复制代码

```
1 Crontab -e
```

我们先来看一下 Crontab 的格式，常用的 Crontab 的表达式语法格式如下。

复制代码

```
1 * * * * command
2 - - - - -
3 | | | | |
4 | | | | ----- day of the week (0 ~ 6) (Sunday=0 or 7)
5 | | | | ----- month (1 ~ 12)
6 | | | ----- day of the month (1 ~ 31)
7 | | ----- hour (0 ~ 23)
8 | ----- min (0 ~ 59)
```

你可以分别看下其中各个字段的含义。

min: 分钟，可选值为 0~59。

hour: 小时，可选值为 0~23。

day of the month: 一个月的第几天，可选值为 1~31。

month: 月份，可选值为 1~12。

day of the week: 一周的第几天, 可选值为 0~6 或者使用名称 (0 表示周日, 1~6 表示周一到周六) 。

以下是一个表达式的例子。

shikey.com转载分享

复制代码

```
1 0 0 * * * /root/mybackup.sh
```

这行代码表示每天的 0 点 0 分 (即每天晚上 12 点) , 执行 /root/mybackup.sh 脚本。

注意, 在使用 Crontab 时, 要先设置好环境变量, 执行的命令要使用绝对路径, 否则会因为环境变量不完整, 或路径错误而执行不成功。

举个例子, 如何用 Crontab 写 Python。

复制代码

```
1 0 0 * * * env PYTHONPATH=/path/to/python/file python /path/to/python/file/script.
```

简单解释下这行代码。

Crontab 命令: 0 0 * * * (表示每天 0 点执行) 。

设置 Python 运行时环境变量的命令: env PYTHONPATH=/path/to/python/file。

执行 Python 文件的命令: python /path/to/python/file/script.py。

如何把推荐系统项目部署到 Linux 中

现在我们已经知道了什么是 Linux, 以及如何在 Linux 上使用 Crontab 命令进行定时任务, 接下来, 我们把项目部署到 Linux 系统中。

先回顾一下用到的项目和资源。

爬虫项目

我们的数据集就是使用爬虫项目来进行爬取的，我在之前爬虫项目基础上，做了两个小小的改动。

首先，之前的 main.py 文件现在直接贴出来，你可以直接在 sina 的主目录下建立一个 main.py 文件，然后输入下面内容。

 复制代码

```
1 from scrapy import cmdline
2
3 cmdline.execute('scrapy crawl sina_spider -a page=10 -a flag=0'.split())
```

这个文件向我们的爬虫文件传递两个参数，第一个参数是 page，表示一次爬多少页（默认是 10 页）。第二个参数是 flag，0 表示全爬下来，1 表示增量爬取。

然后，我们也把爬虫文件的 Parse 函数做了如下更改。

 复制代码

```
1 def parse(self, response):
2     driver = webdriver.Chrome(chrome_options=self.option)
3     driver.set_page_load_timeout(30)
4     driver.get(response.url)
5     for i in range(self.page):
6         while not driver.find_element_by_xpath("//div[@class='feed-card-page']")
7             driver.execute_script("window.scrollTo(0,document.body.scrollHeight)")
8             title = driver.find_elements_by_xpath("//h2[@class='undefined']/a[@ta")
9             time = driver.find_elements_by_xpath("//h2[@class='undefined']/..div")
10            for i in range(len(title)):
11                eachtitle = title[i].text
12                eachtime = time[i].text
13                item = DataItem()
14                if response.url == "https://ent.sina.com.cn/zongyi/":
15                    item['type'] = 'zongyi'
16                elif response.url == "https://news.sina.com.cn/china/":
17                    item['type'] = 'news'
18                else:
19                    item['type'] = 'film'
20                item['title'] = eachtitle
21                item['desc'] = ''
```

```

22         href = title[i].get_attribute('href')
23         today = datetime.datetime.now()
24         eachtime = eachtime.replace('今天', str(today.month) + '月' + str(
25             if '分钟前' in eachtime:
26                 minute = int(eachtime.split('分钟前')[0]))
27                 t = datetime.datetime.now() - datetime.timedelta(minutes=minute)
28                 t2 = datetime.datetime(year=t.year, month=t.month, day=t.day,
29             else:
30                 if '年' not in eachtime:
31                     eachtime = str(today.year) + '年' + eachtime
32                     t1 = re.split('[年月日:]', eachtime)
33                     t2 = datetime.datetime(year=int(t1[0]), month=int(t1[1]), day
34                                         minute=int(t1[4]))
35
36             item['times'] = t2
37
38             if self.flag == 1:
39                 today = datetime.datetime.now().strftime("%Y-%m-%d")
40                 yesterday = (datetime.datetime.now() + datetime.timedelta(day
41                 if item['times'].strftime("%Y-%m-%d") < yesterday:
42                     driver.close()
43                     break
44                 if yesterday <= item['times'].strftime("%Y-%m-%d") < today:
45                     yield Request(url=response.urljoin(href), meta={'name': i
46             else:
47                 yield Request(url=response.urljoin(href), meta={'name': item}
48         try:
49             driver.find_element_by_xpath("//div[@class='feed-card-page'] /span
50         except:
51             break

```

这里我们增加了页数和对应的 flag，同时也做了增量时间的处理，我们主要看下面这一部分代码。

 复制代码

```

1 if self.flag == 1:
2     today = datetime.datetime.now().strftime("%Y-%m-%d")
3     yesterday = (datetime.datetime.now() + datetime.timedelta(days=-1)).strftime(
4         if item['times'].strftime("%Y-%m-%d") < yesterday:
5             driver.close()
6             break
7         if yesterday <= item['times'].strftime("%Y-%m-%d") < today:
8             yield Request(url=response.urljoin(href), meta={'name': item}, callback=
9         else:
10            yield Request(url=response.urljoin(href), meta={'name': item}, callback=self.

```

当程序运行时，它会检查 `self.flag` 是否等于 1。如果等于 1，则获取当前时间的日期作为 `today`，获取昨天的日期作为 `yesterday`。

然后，程序会检查 `Item` 字典中存储的时间是否早于昨天的日期，如果是，关闭浏览器 (`driver.close()`) 并停止循环 (`break`)。如果 `Item` 字典中存储的时间在昨天和今天之间，则执行一个名为 `self.parse_namedetail` 的回调函数。

如果 `self.flag` 不为 1，则直接执行 `self.parse_namedetail` 回调函数。在执行回调函数时，会将当前的 URL 和 `Item` 字典传递给 `parse_namedetail` 函数。

当然，我们还要去稍微对 `__init__` 函数做一个改变，具体如下。

 复制代码

```

1  def __init__(self, page=None, flag=None, *args, **kwargs):
2      super(SinaSpiderSpider, self).__init__(*args, **kwargs)
3      self.page = int(page)
4      self.flag = int(flag)
5      self.start_urls = ['https://news.sina.com.cn/china/', 'https://ent.sina.c
6          self.option = webdriver.ChromeOptions()
7          self.option.add_argument('no=sandbox')
8          self.option.add_argument('--blink-setting=imagesEnable=false')
9

```

这段代码相对于之前的代码主要增加了两个变量：`page` 和 `flag`，都是由 `main` 文件传进来的。

这时就可以将整个文件放到 Linux 服务器中，然后尝试运行它。关于 Linux 环境中如何搭建 Python 环境，你可以参考 [这篇文章](#)。

假设我们的环境已经搭建完成，并且已经把项目放到了对应的目录下（这里我放到了 `/data/sina` 目录下），这个时候，我们可以在 Linux 下输入如下命令。

复制代码

```
1 Crontab -e
```

然后再编辑如下内容。

shikey.com转载分享

复制代码

```
1 0 6 * * * /usr/local/python3 /data/sina/main.py
```

退出后，程序自动生效。建议第一次手动运行全量爬取，然后再每天定时运行。

推荐系统主项目

推荐系统主项目的部署和爬虫程序从大体上来说是相同的，只是在定时文件运行上略有差异。

我们在推荐系统主项目中，主要涉及下面三个部分的内容。

1. 处理爬虫爬取下来的内容，也就是对画像的处理。
2. 处理召回层的各个算法。
3. 排序层的内容。

因为定时任务的基本原理类似，为避免重复，我们展开来讲一讲**在每小时的第 20 分钟刷新热度池到 Redis 数据库**这个定时任务。

一般来讲，热度池一般会由于某些特殊的原因暴增，往往这些热度可能是在几天甚至几分钟内发酵起来的，为了不错过热度，又不想在推荐系统刚建立的初期给服务器造成太大的压力，因此会选择每小时刷新一次。

我们可以单独写一个 Python 文件进行热度池的刷新，将 MongoDB 中的热度最高的 Top50 或者 Top100 刷新到 Redis 中形成热度池。在 Python 中实际上就是从 MongoDB 中读取数据，然后再取相应的内容存到 Redis 中。当项目文件放到服务器上后，就可以使用如下 Crontab 命令来做定时任务。

[复制代码](#)

```
1 20 * * * * source /data/ai/venv/bin/activate;cd /data/ai/recommendation-class/sch
```

简单解释一下这行命令。

shikey.com转载分享

20 * * * * : 在每小时的第 20 分钟执行后面的命令。

source /data/ai/venv/bin/activate: 先激活 /data/ai/venv 虚拟环境。

cd /data/ai/recommendation-class/scheduler/: 切换到 /data/ai/recommendation-class/scheduler/ 目录下。

python sched_refresh_redis_hot_pool.py >> /data/ai/logs/scheduler.log: 执行 sched_refresh_redis_hot_pool.py 脚本，并将脚本执行的输出追加到 /data/ai/logs/scheduler.log 文件中。

除了上面讲的这个定时任务外，我们还会有下面这些常用的定时任务，你可以根据自己的需求灵活变通。

每小时的第 20 分钟刷新最新池到 Redis 数据库。

每小时的第 0 分钟刷新增量的内容画像到 MongoDB 数据库。

每 3 小时跑一次协同过滤算法的流程。

每天 0 点跑一次 YouTubeDNN 的召回模型。

每 2 小时跑一次 GBDT+LR 算法。

每 30 分钟刷新一次负反馈内容到 Redis 数据库。

每 30 分钟跑一次冷启动算法。

总结

到现在为止，我们大体了解了如何在 Linux 系统上使用 Crontab 命令做定时任务。除了定时任务之外，剩下的就是服务的部署。服务部署一般是利用 Nginx 配合后端程序来进行部署，这一部分更像是服务端开发工程师的工作，如果你感兴趣的话，我们留言区一起交流讨论。

接下来我对这节课的内容做一个简单总结。

1. Linux 系统是一个自由、开源的操作系统，它拥有良好的稳定性、安全性和可靠性，可以运行在不同的计算机硬件和架构中。
2. Linux 中的 Crontab 命令是一种用于定期执行某项任务的命令。在它的命令中第一个星号表示分钟，第二个星号表示小时，第三个星号表示日，第四个星号表示月，而第五个星号表示星期。
3. 熟悉在一个推荐系统中哪部分需要定时任务，以及如何去设置它。

课后练习

最后，给你留一道课后练习题：尝试把我们的项目部署到 Linux 系统，并贴出你的 Crontab 命令。

期待你的分享，如果今天的内容让你有所收获，也欢迎你推荐给有需要的朋友！

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (1)



peter

2023-06-16 来自北京

Q1：网站后端都部署在Linux上而不用windows，主要原因是性能问题吗？即windows太慢，而Linux处理速度快。

Q2：Crond与Crontab是什么关系？

Crond与Crontab是两个独立的软件，两者之间是类似于client-server的关系，即Crond负责处理，Crontab主要是显示，可以这样理解吗？

Q2：Crontab与Java的定时器是什么关系？

Java体系中有自己的定时器，其与Crontab是什么关系？Java的定时器是基于Crontab实现的，即Java定时器底层其实是通过Crontab实现的，可以这样理解吗？



shikey.com转载分享