|  |
| --- |
|  |
| AMiner |
| |  | | --- | | 推荐系统部署方案 | |
|  |
|  |
|  |
|  |
|  |
|  |

目录

1系统部署流程 1

1.1. 网络环境部署 1

1.1.1. 网络配置 1

1.2. 系统服务部署 1

1.2.1. Nginx服务部署 1

1.2.1.1. 安装 1

1.2.1.2. 配置 1

1.2.2. Meili检索服务部署 3

1.2.2.1. 安装 3

1.2.3. Postgres数据库部署 7

1.2.3.1. 安装 7

1.2.3.2. 配置 7

1.2.3.3. 创建用户 7

1.2.4. Milvus向量搜索数据库服务部署 8

1.2.4.1. 安装 8

1.2.4.2. 配置 8

1.2.5. Redis 内存数据库部署 9

1.2.5.1. 安装 9

1.2.5.2. 配置 9

1.3. 推荐算法部署 10

1.3.1. Python部署 10

1.3.2. RMS API部署 10

1.3.3. RMS 离线任务部署 11

1.3.4. RMS Recall任务部署 12

1.3.5. Topic Embedding服务部署 13

1.3.6. CF\_recommendation 协同推荐 13

# 1系统部署流程

## 网络环境部署

系统应部署于云服务器网络环境中。此节介绍系统服务器上的网络环境部署流程。

### 网络配置

每台服务器作为局域网的节点，具体网络配置在文件 /etc/sysconfig/network-scripts/ifcfg-em1中进行设置。

使用ifconfig或者ip addr命令查看当前网络状态。

### 机器配置

推荐系统各模块所在机器位置如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **资源名称** | **用途** | **所在机器位置（IP）** |
| 1 | GPU | 一台用于训练，一台用于预测 | 192.168.0.51 |
| 2 | CPU | 离线任务 | 192.168.0.53 |
| 3 | CPU | 分布式容灾方案，部署API接口、任务worker | 192.168.0.54~192.168.0.57 |
| 4 | Redis | 内存加速 | 192.168.0.51 |
| 5 | Milvus | 向量引擎 | 192.168.0.50 |
| 6 | Postgres | 存储元数据 | 192.168.0.49 |
| 7 | Meili | 存储搜索数据 | 192.168.0.52 |

## 系统服务部署

### Nginx服务部署

系统使用反向代理服务器Nginx进行反向代理、负载均衡。

#### 安装

安装192.168.0.54上。Nginx应通过官方软件源进行安装，具体流程见<http://nginx.org/en/linux_packages.html>。

#### 配置

默认情况下，Nginx 配置文件位于系统 /etc/nginx 下。目录 /etc/nginx/conf.d 目录下所有 \*.conf 配置文件在 Nginx 启动时也会自动载入。

系统 HTTP 服务的主要配置位于/etc/nginx/conf.d/ngx\_pingback.conf目录下，服务配置涉及API以及日志上报pingback服务。

ngx\_pingback.conf内容如下：

|  |
| --- |
| upstream rec\_servers {  server 192.168.0.54:4091 weight=2;  server 192.168.0.55:4091 weight=10;  server 192.168.0.56:4091 weight=10;  server 192.168.0.57:4091 weight=10;  }  upstream pingback\_servers {  server 192.168.0.54:4091 weight=10;  }  server {  listen 80;  server\_name pingback.aminer.cn;  #log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  # '$status $body\_bytes\_sent "$http\_referer" '  # '"$http\_user\_agent" "$http\_x\_forwarded\_for" $upstream\_response\_time';    add\_header X-Forward-For $proxy\_add\_x\_forwarded\_for;    location /static {  alias /data/webapp/rms/appserver/recsys/recsys/static;  }    location / {  access\_log /data/weblog/pingback.access.log main;    proxy\_pass http://pingback\_servers/pingback/;  proxy\_redirect off;  #proxy\_next\_upstream error timeout http\_502 http\_500;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  #proxy\_set\_header Content-Type "application/x-www-form-urlencoded";  proxy\_set\_header Content-Type "$content\_type";  }  }  server {  listen 80;  server\_name 192.168.0.54 backend.aminersz.cn;  access\_log /data/weblog/apiv2.access.log main;  add\_header X-Forward-For $proxy\_add\_x\_forwarded\_for;  location /static {  alias /data/webapp/rms/appserver/recsys/recsys/static;  }  location /pingback {  access\_log /data/weblog/pingback.access.log main;  proxy\_pass http://pingback\_servers;  proxy\_redirect off;  #proxy\_next\_upstream error timeout http\_502 http\_500;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  #proxy\_set\_header Content-Type "application/x-www-form-urlencoded";  proxy\_set\_header Content-Type "$content\_type";  }  location / {  proxy\_pass http://rec\_servers;  proxy\_redirect off;  proxy\_next\_upstream error timeout http\_502 http\_500 http\_504;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  #proxy\_set\_header Content-Type "application/x-www-form-urlencoded";  proxy\_set\_header Content-Type "$content\_type";  }  } |

默认情况下，Nginx 的日志位于 /var/log/nginx 与/data/weblog目录下。

当 Nginx 的配置被修改后，应当使用如下命令重新加载配置，使其生效：

* sudo systemctl reload nginx

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

### Meili检索服务部署

系统使用Meili-v0.26.1提供检索服务，可以参考文档：https://docs.meilisearch.com

#### 安装

服务部署在192.168.0.52上，操作系统须为CentOS-8，否则将无法运行。

下载二进制文件meilisearch-linux-amd64，并放到目录/data/cache/meili

创建启动脚本supervisord-meili.sh，内容如下：

|  |
| --- |
| #!/bin/bash  #  # Init file for supervisord-meili  #  # chkconfig: 2345 55 25  # description: supervisor server daemon  #  # processname: supervisord-meili  #  NAME="supervisord-meili"  CONFIG=/data/cache/meili/supervisord.ini  CMD="/data/envs/rms/bin/supervisord -c ${CONFIG}"  PID="/tmp/${NAME}.pid"  wait\_for\_pid () {  try=0  while test $try -lt 30 ; do  case "$1" in  'created')  if [ -f "$2" ] ; then  try=''  break  fi  ;;  'removed')  if [ ! -f "$2" ] ; then  try=''  break  fi  ;;  esac  echo -n .  try=`expr $try + 1`  sleep 1  done  }  case "$1" in  start)  echo -n "Starting ${NAME} "  ulimit -n 10240    ${CMD}  if [ $? != 0 ] ; then  echo " failed"  exit 1  fi  wait\_for\_pid created "$PID"  if [ -n "$try" ] ; then  echo " failed"  exit 1  else  echo " done"  fi  ;;  stop)  if [ -f ${PID} ]; then  pid=`cat ${PID}`  else  pid=`ps aux | grep "supervisord" | grep "meili" | grep -v grep | awk -F " " '{print $2}' | head -n 1`  fi  kill ${pid}  wait\_for\_pid removed ${PID}  rm -f ${PID}  echo "Gracefully shutting down ${NAME}, PID ${pid}"  ;;  reload)  if [ -f ${PID} ]; then  pid=`cat ${PID}`  else  pid=`ps aux | grep "supervisord" | grep "meili" | grep -v grep | awk -F " " '{print $2}' | head -n 1`  fi  echo "Reload ${NAME}: ${pid}"  kill -s HUP ${pid}  ;;  status)  ps -aux | grep "supervisord" | grep "meili"  ;;  restart)  $0 stop  sleep 2  $0 start  ;;  \*)  echo "Usage: $0 {start|stop|restart|reload|status|check}"  exit 1  ;;  esac |

创建配置文件supervisord.ini，内容如下：

|  |
| --- |
| [supervisord]  logfile = /data/weblog/supervisord-meili.log  logfile\_maxbytes=50MB  logfile\_backups= 1  loglevel = debug  pidfile = /tmp/supervisord-meili.pid  directory=/data/weblog  user=aminer  [program:meili]  #command=/data/cache/meili/meilisearch\_028 --http-addr "0.0.0.0:7700"  command=/data/cache/meili/meilisearch-linux-amd64 --http-addr "0.0.0.0:7700"  environment=  directory=/data/cache/meili  user=aminer  process\_name=%(program\_name)s\_%(process\_num)02d  numprocs=1  stdout\_logfile=/data/weblog/meili.log  stderr\_logfile=/data/weblog/meili.log  autostart=true  autorestart=true  startsecs=10  priority=20 |

启动服务

|  |
| --- |
| ./supervisord-meili.sh start |

停止服务

|  |
| --- |
| ./supervisord-meili.sh stop |

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

关闭selinux，修改文件/etc/selinux/config，然后重启。修改内容如下:

|  |
| --- |
| SELINUX=disabled |

服务比较稳定，暂未出现过异常崩溃。

### Postgres数据库部署

系统使用Postgresql-14.x作为数据库。

#### 安装

使用yum安装，部署在192.168.0.49上。

|  |
| --- |
| sudo yum install -y net-tools  sudo yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86\_64/pgdg-redhat-repo-latest.noarch.rpm  sudo yum install -y postgresql14-server  sudo /usr/pgsql-14/bin/postgresql-14-setup initdb |

加入自启动服务。

|  |
| --- |
| sudo systemctl enable postgresql-14 |

启动postgresql

|  |
| --- |
| sudo systemctl start postgresql-14 |

停止postgresql

|  |
| --- |
| sudo systemctl stop postgresql-14 |

#### 配置

Postgresql使用默认配置文件: /data2/pgsql/14/data/postgresql.conf

数据存储路径配置为：

|  |
| --- |
| /data2/pgsql/14/data |

#### 创建用户

开启权限认证功能后需要用户名密码才能够访问数据库。

执行sudo -u postgres psql进入psql终端，然后执行：

|  |
| --- |
| cho CREATE USER cacti SUPERUSER;  ALTER USER cacti WITH ENCRYPTED PASSWORD 'Cacti\_12345';  ALTER USER cacti WITH CREATEDB;  ALTER USER cacti WITH LOGIN;  create database rms;  grant all privileges on database rms to cacti;  CREATE USER reporter;  ALTER USER reporter WITH ENCRYPTED PASSWORD 'report\_1232\*()';  ALTER USER reporter WITH LOGIN;  grant pg\_read\_all\_data to reporter;  CREATE USER feature\_dev;  ALTER USER feature\_dev WITH ENCRYPTED PASSWORD 'howtodoit12345\_\*&%';  ALTER USER feature\_dev WITH LOGIN;  grant pg\_read\_all\_data to feature\_dev;  echo "Append 'host all all 0.0.0.0/0 md5' to pg\_hba.conf" |

修改下/data2/pgsql/14/data/pg\_hba.conf，增加如下内容：

|  |
| --- |
| host all all 0.0.0.0/0 md5 |

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

关闭selinux，修改文件/etc/selinux/config，然后重启。修改内容如下:

|  |
| --- |
| SELINUX=disabled |

服务比较稳定，暂未出现过异常崩溃。

### Milvus向量搜索数据库服务部署

系统使用Milvus-2.0.x来进行向量检索，支持语义级别的检索。Attu作为管理终端，浏览milvus中的数据。

#### 安装

部署在192.168.0.50上，Web管理入口是<http://192.168.0.50:8000/>。

通过官方软件源进行安装，具体参见：

<https://milvus.io/docs/v2.0.x/install_standalone-aptyum.md>

|  |
| --- |
| sudo yum install <https://github.com/milvus-io/milvus/releases/download/v2.0.2/milvus-2.0.2-.el7.x86_64.rpm> |

#### 配置

配置文件为 /etc/milvus/configs/milvus.yml

监听所有网络地址：

|  |
| --- |
| proxy:  address: 0.0.0.0  port: 19530 |

启动服务：

|  |
| --- |
| Sudo systemctl start milvus |

停止服务

|  |  |
| --- | --- |
| Sudo systemctl stop milvus |  |

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

关闭selinux，修改文件/etc/selinux/config，然后重启。修改内容如下:

|  |
| --- |
| SELINUX=disabled |

服务比较稳定，暂未出现过异常崩溃。

### Redis 内存数据库部署

系统使用Redis来存储热点数据，以提升整体的响应性能。

#### 安装

服务部署在192.168.0.51上。

先使用yum安装系统自带的redis service

|  |
| --- |
| Sudo yum install -y redis |

从官网下载Redis-5.0.14

|  |
| --- |
| https://redis.io/download/ |

进入redis 5.0.14目录，执行命令：

|  |
| --- |
| make PREFIX=/usr/local/redis5 install  mkdir /usr/local/redis5/conf  cp redis.conf /usr/local/redis5/conf |

#### 配置

修改redis.conf，监听全部地址：

|  |
| --- |
| bind 0.0.0.0 |

关闭硬盘同步

|  |
| --- |
| save ""  #save 900 1  #save 300 10  #save 60 10000 |

设置密码

|  |
| --- |
| requirepass UEWA21CzY3Gwclf5pTXPdpsEUehFFiNQF+tl6NwADl4= |

修改/usr/lib/systemd/system/redis.service，内容如下：

|  |
| --- |
| [Unit]  Description=Redis persistent key-value database  After=network.target  After=network-online.target  Wants=network-online.target  [Service]  #ExecStart=/usr/bin/redis-server /etc/redis.conf --supervised systemd  #ExecStop=/usr/libexec/redis-shutdown  ExecStart=/usr/local/redis5/bin/redis-server /usr/local/redis5/conf/redis.conf --supervised systemd  ExecStop=/usr/local/redis5/bin/redis-shutdown  Type=notify  User=redis  Group=redis  RuntimeDirectory=redis  RuntimeDirectoryMode=0755  [Install]  WantedBy=multi-user.target |

启动redis

|  |
| --- |
| Sudo systemctl start redis |

停止redis

|  |
| --- |
| Sudo systemctl stop redis |

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

关闭selinux，修改文件/etc/selinux/config，然后重启。修改内容如下:

|  |
| --- |
| SELINUX=disabled |

服务比较稳定，暂未出现过异常崩溃。

## 推荐算法部署

### Python部署

部署在192.168.0.53~57有服务器上。

从官网下载Python-3.9.16.tgz，下载地址 <https://www.python.org/downloads/>

下载完成后执行如下命令：

|  |
| --- |
| sudo yum install -y gcc openssl-devel bzip2-devel libffi-devel sqlite-devel lzma-devel postgresql-devel && \  cd /home/aminer && \  tar zxf Python-3.9.16.tgz && \  cd Python-3.9.16 && \  ./configure --enable-optimizations --prefix=/usr/local/python39 --with-ensurepip=install --enable-loadable-sqlite-extensions && \  sudo make clean install && \  mkdir /data/envs |

建立虚拟环境

|  |
| --- |
| sudo /usr/local/python39/bin/pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple virtualenv && \  sudo mkdir -p /data/envs && \  sudo chown -R aminer:aminer /data/envs && \  /usr/local/python39/bin/virtualenv /data/envs/rms |

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

### RMS API部署

部署在192.168.0.54~57上。

下载源代码rms，并放到/data/webapp/rms。安装程序依赖包，如下命令:

|  |
| --- |
| cd /data/webapp/rms/appserver/conf/shenzhen  ./install.sh local |

建立日志目录/data/weblog

|  |
| --- |
| mkdir /data/weblog |

启动服务

|  |
| --- |
| cd /data/webapp/rms/appserver/conf/shenzhen  ./supervisord-rms.sh start |

停止服务

|  |
| --- |
| cd /data/webapp/rms/appserver/conf/shenzhen  ./supervisord-rms.sh stop |

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

每次更新代码，需要重启服务，执行如下命令

|  |
| --- |
| ./supervisord.sh stop  ./supervisord.sh start |

关闭selinux，修改文件/etc/selinux/config，然后重启。修改内容如下:

|  |
| --- |
| SELINUX=disabled |

当前系统有一个问题，每次启动服务的时候，pid文件总是无法立刻落盘，会导致脚本会误报启动失败，实际服务已经启动了。需要运维同学研究解决一下。

### RMS 离线任务部署

部署在192.168.0.53上。

下载源代码rms，并放到/data/webapp/rms。安装程序依赖包，如下命令:

|  |
| --- |
| cd /data/webapp/rms/appserver/conf/shenzhen  ./install.sh local |

建立日志目录/data/weblog

|  |
| --- |
| mkdir /data/weblog |

挂载crontab, 执行crontab -e，内容如下：

|  |
| --- |
| # RMS  5 1 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh update\_pub\_ids  30 1,13 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh update\_pub  55 12,23 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh export\_user\_event\_log  30 1 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh paper\_quality\_update  40 1,15 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh subscribe\_stat\_update  45 14,2 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh ann\_update --mode train --type pub  45 15,4 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh ann\_update --mode train --type pub\_vector  15 3 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh ann\_update --mode train --type topic  5 \* \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh recall\_favorite\_update --mode train  30 2 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh ads\_update  15 \* \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh schedule  # RMS algorithm  30 12 \* \* \* cd /data/webapp/rms/appserver/confs/shenzhen; ./command.sh algorithm\_update --algorithm native --mode train |

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

关闭selinux，修改文件/etc/selinux/config，然后重启。修改内容如下:

|  |
| --- |
| SELINUX=disabled |

### RMS Recall任务部署

部署在192.168.0.53上。

建立虚拟环境recall

|  |
| --- |
| sudo /usr/local/python39/bin/pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple virtualenv && \  sudo mkdir -p /data/envs && \  sudo chown -R aminer:aminer /data/envs && \  /usr/local/python39/bin/virtualenv /data/envs/recall |

下载源代码rms\_recall，并放到/data/webapp/rms\_recall，使用git切换到rms\_recall\_sz分支。安装程序依赖包，如下命令:

|  |
| --- |
| cd /data/webapp/rms\_recall  /data/envs/recall/bin/pip install -r requirements.txt |

上传模型/data/cache/aminer/meta/embedding，当前模型文件放在192.168.0.53上的/data/cache/aminer/meta/embedding下

|  |
| --- |
| [aminer@53 embedding]$ ll  total 0  drwxr-xr-x. 2 aminer aminer 87 Feb 20 04:46 oagbert-v2-sim  drwxr-xr-x. 2 aminer aminer 55 Feb 20 05:42 tfidf-200 |

建立日志目录/data/weblog

|  |
| --- |
| mkdir /data/weblog |

挂载crontab, 执行crontab -e，内容如下：

|  |
| --- |
| # RMS Recall  45 1 \* \* \* cd /data/webapp/rms\_recall; ./safeRun-pro.sh cold  20 \*/2 \* \* \* cd /data/webapp/rms\_recall; ./safeRun-pro.sh latest\_paper  20 1 \* \* \* cd /data/webapp/rms\_recall; ./safeRun-pro.sh scholar\_pool  15 1 \* \* \* cd /data/webapp/rms\_recall; ./safeRun-pro.sh email\_follow\_new  5 3 \* \* \* cd /data/webapp/rms\_recall; ./safeRun-pro.sh follow\_recall  5 6 \* \* \* cd /data/webapp/rms\_recall; ./safeRun-pro.sh miniprogram\_recall  5 4 \* \* \* cd /data/webapp/rms\_recall; ./safeRun-pro.sh subscribe\_children\_kg  20 1 \* \* \* cd /data/webapp/rms\_recall; ./safeRun-pro.sh embedding  20 3 \* \* \* cd /data/webapp/rms\_recall; ./safeRun-pro.sh subscribe\_oag |

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

关闭selinux，修改文件/etc/selinux/config，然后重启。修改内容如下:

|  |
| --- |
| SELINUX=disabled |

### Topic Embedding服务部署

部署在192.168.0.53上。

建立虚拟环境recall

|  |
| --- |
| sudo /usr/local/python39/bin/pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple virtualenv && \  sudo mkdir -p /data/envs && \  sudo chown -R aminer:aminer /data/envs && \  /usr/local/python39/bin/virtualenv /data/envs/recall |

下载源代码rms\_recall，并放到/data/webapp/rms\_recall，使用git切换到rms\_recall\_sz分支。安装程序依赖包，如下命令:

|  |
| --- |
| cd /data/webapp/rms\_recall  /data/envs/recall/bin/pip install -r requirements.txt |

上传模型/data/cache/aminer/meta/embedding，当前模型文件放在192.168.0.53上的/data/cache/aminer/meta/embedding下

|  |
| --- |
| [aminer@53 embedding]$ ll  total 0  drwxr-xr-x. 2 aminer aminer 87 Feb 20 04:46 oagbert-v2-sim  drwxr-xr-x. 2 aminer aminer 55 Feb 20 05:42 tfidf-200 |

建立日志目录/data/weblog

|  |
| --- |
| mkdir /data/weblog |

启动服务

|  |
| --- |
| cd /data/webapp/rms\_recall/topic\_embedding  ./ supervisord-topic.sh start |

停止服务

|  |
| --- |
| cd /data/webapp/rms\_recall/topic\_embedding  ./ supervisord-topic.sh stop |

#### 注意事项

需要将系统的max open file与max process number设置在40K以上，越大越好。修改文件/etc/security/limits.d/20-nproc.conf，增加如下内容：

|  |
| --- |
| # Default limit for number of user's processes to prevent  # accidental fork bombs.  # See rhbz #432903 for reasoning.  \* soft nproc 40960  \* soft nofile 40960  root soft nproc unlimited |

关闭selinux，修改文件/etc/selinux/config，然后重启。修改内容如下:

|  |
| --- |
| SELINUX=disabled |

### CF\_recommendation 协同推荐

CF\_recommendation 的功能是根据输入的用户点击历史日志对用户推荐论文.

克隆项目，git clone http://jira.sciencecores.com:7990/scm/aminersz/get-paper-embedding.git

安装本项目所需的包

|  |
| --- |
| pip install -r requirements.txt |

进入子目录 /CF\_reecommendation/GFCF\_rec/

|  |
| --- |
| cd CF\_recommendation/GFCF\_rec/ |

修改文件 world.py，修改 ROOT\_PATH 变量为 CF\_recommendation 目录所在的绝对路径 (e.g. ROOT\_PATH = "/data/webapp/get-paper-embedding/CF\_recommendation")；修改PROD\_PATH 变量为数据所在的绝对路径 (e.g. PROD\_PATH = "/data/cache/aminer")

|  |
| --- |
| vi world.py |

进入PROD\_PATH路径，新建文件夹 run\_log

|  |
| --- |
| mkdir run\_log  mkdir output\_recommendation |

执行程序时，程序从PROD\_PATH加载日志文件，形如 $PROD\_PATH/input\_log/YYYY-MM-DD-log.json, 日志的参考样例格式为 /CF\_recommendation/input\_log/. 本程序运行日志会存放于 $PROD\_PATH/run\_log/ ，并且推荐结果存放于 $PROD\_PATH/output\_recommendation/. 推荐结果会是一个 json.gz 格式的文件，命名格式为 cf\_XXX\_YYYY\_MM\_DD.json.gz ，其中XXX 是推荐的类型([report, topic, person, 推荐物是论文时为空]).本文封装了4个推荐任务以产生不同任务的推荐场景.假设环境装在conda中，在确定脚本可正常运行后，推荐按如下方式设置定时任务。

|  |
| --- |
| # get-paper-embedding CF\_recommendation  SHELL=/bin/bash  PATH=/data/envs/get\_paper\_embedding/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/aminer/.local/bin:/home/aminer/bin  0\*\*\*\*source/data/envs/get\_paper\_embedding/bin/activate;/data/webapp/get-paper-embedding/CF\_recommendation/GFCF\_rec/production\_run\_pub.sh; deactivate  15\*\*\*source/data/envs/get\_paper\_embedding/bin/activate;/data/webapp/get-paper-embedding/CF\_recommendation/GFCF\_rec/production\_run\_report.sh; deactivate  30\*\*\*\*source/data/envs/get\_paper\_embedding/bin/activate;/data/webapp/get-paper-embedding/CF\_recommendation/GFCF\_rec/production\_run\_topic.sh; deactivate  45\*\*\*\*source/data/envs/get\_paper\_embedding/bin/activate;/data/webapp/get-paper-embedding/CF\_recommendation/GFCF\_rec/production\_run\_person.sh; deactivate |

异常情况

1. 安装requirements.txt中的sparsesvd==0.2.2时报错。

解决方案：先注释此package，再次安装requirements.txt

参考https://pypi.org/project/sparsesvd/，手动安装sparsesvd

1. 报错No such file or directory:

解决方案：参考上述文档，前往相应路径创建该文件夹

1. 报错 ValueError: `k` must be an integer satisfying `0 < k < min(A.shape)`.

解决方案：将用户日志文件放在 PROD\_PATH/input\_log,当前日志文件放在192.168.0.51上的/data/cache/aminer/input\_log下, 建议存放30天以上日志。

若仍报错，进入production\_run\_pub.sh，production\_run\_report.sh，production\_run\_topic.sh，production\_run\_person.sh，手动调低topks的值，直至成功。如改为--topks="[20]"