

服务器信息

VPN :banyaguang/123456

服务器:

172.16.20.10

bjdx

Bjdx@20220329

产经新闻、报告库

ChatGLM2部署

- 下载模型源码
 - 下载模型文件<https://github.com/THUDM/ChatGLM2-6B>，所有文件放在ChatGLM2-6B文件夹里
 - 下载量化模型参数文件<https://huggingface.co/THUDM/chatglm2-6b>，所有文件放在chatglm2-6b文件夹里（chatglm2-6b是ChatGLM2-6B的子文件夹）
- 配置环境
 - 安装anaconda，测试版本为在<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>处下载的Anaconda3-2023.03-0-Windows-x86_64.exe文件
 - 安装CUDA 11.8（必须是11.8）https://developer.nvidia.com/cuda-11-8-0-download-architecture?target_os=Windows&target_arch=x86_64，根据系统版本选择并下载（windows 10 - exe local）
 - 安装与CUDA 11.8适配的CUDNN，<https://developer.nvidia.com/rdp/cudnn-download#a-collapse894-118>（for CUDA 11.x）
- 配置conda环境
 - 配置镜像源，在终端输入：(1) pip config set global.index-url <https://mirrors.aliyun.com/pypi/simple/> (2) pip config set install.trusted-host mirrors.aliyun.com
 - 打开Anaconda Powershell Prompt，输入conda create -n chatglm2 python=3.10（chatglm2是环境的名称）
 - 输入conda activate chatglm2激活环境，并cd到模型文件夹，在终端输入pip install -r requirements.txt（没配置镜像源就很慢）
 - requirements.txt中torch>=2.0版本不对，需要手动安装pytorch 2.0.1，经测试下面的版本可以，在激活chatglm2环境的情况下，终端输入pip install torch==2.0.0+cu118 torchvision==0.15.1+cu118 torchaudio==2.0.1 --index-url <https://download.pytorch.org/whl/cu118>
- 测试是否部署成功
 - 打开Anaconda Powershell Prompt，输入conda activate chatglm2
 - cd到文件夹，终端输入python web_demo.py，如果ChatGLM网页出来就是成功了

环境安装

安装DOCKER

##没有curl命令通过

```

sudo apt install curl

## 安装docker
curl -fsSL https://test.docker.com -o test-docker.sh
sudo sh test-docker.sh

#配置数据源
第一步：新建或编辑daemon.json
vi /etc/docker/daemon.json

第二步：daemon.json中编辑如下
{
    "registry-mirrors": ["http://hub-mirror.c.163.com"]
}

第三步：重启docker
systemctl restart docker.service

第四步：执行docker info查看是否修改成功
docker info

```

Docker 安装Mysql

```

docker pull mysql:5.7

docker images

mkdir -p /docker/mysql/{log,conf,data}

docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456 -e
TZ=Asia/Shanghai -d -v /docker/mysql/data:/var/lib/mysql -v
/docker/mysql/conf:/etc/mysql/conf.d -v /docker/mysql/log:/var/log/mysql
mysql:5.7 --lower_case_table_names=1

```

Docker Redis 安装

Ubuntu安装Conda

```

# 上传文件 Anaconda3-2022.10-Linux-x86_64.sh
rz -y

bash Anaconda3-2022.10-Linux-x86_64.sh

#一直enter然后输入yes

配置环境变量
sudo vim ~/.bashrc

```

```
## 根据安装目录添加环境变量/root/anaconda3/bin, 添加到 ~/.bashrc 最后一行
export PATH=/root/anaconda3/bin:$PATH

## 使配置生效
source ~/.bashrc

##配置数据源

conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/linux-64/
conda config --set show_channel_urls yes
```

项目运行

使用conda创建一个虚拟环境

```
conda create -n ReportKeeper python=3.8.16

conda create -n IdeaBank python=3.8.16

source activate ReportKeeper

source activate IdeaBank

conda deactivate
```

遇到文件传输不了或者权限不允许

```
### 将可读可写可执行的权限赋予该目录
chmod 777 directory_name
```

项目环境安装

```
###当前的虚拟环境需要时ReportKeeper
cd /home/bjdx/idea-bank-workspace/ReportKeeper
pip install -r requirements.txt
```

项目运行

```
# 部署flask服务
nohup python3 -u app.py > nohup.log 2>&1 &

## 查看项目运行的日志
tail -400f nohup.log

#有安装包未安装则安装
pip3 install [package name]
```

开机自启动脚本

商机库、政策库

商机库数据源

- 公众号
 - [CMKT咨询圈](#)：咨询行业概况
 - 关键字：政府咨询、企业咨询、产业规划
 - [中国智库要闻](#)
- 网址
 - [中国政府采购网](#)：招标采购信息公示
 - 爬取招标信息
 - 忽略工程建设招标
 - 爬取成交信息，查看当地产业是否也可以进行规划升级
 - [湖南政府采购网](#)
 - [长沙政府采购网](#)

部署

前端

```
cd /home/bjdx/idea-bank-workspace/vue
```

default.conf

```
server {
    listen      80;
    server_name 172.16.20.10; # 修改为docker服务宿主机的ip

    location / {
        root    /usr/share/nginx/html;
        index   index.html index.htm;
        try_files $uri $uri/ /index.html =404;
    }
}
```

```

    location /u/v1/{
        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_pass http://172.16.20.10:8000/u/v1/; #注意：使用代理地址时末尾记得加上斜杠"/"

        proxy_buffer_size 1280k;
        proxy_buffers 32 1280k;
        proxy_busy_buffers_size 1280k;
        client_max_body_size 20m;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}

```

Dockerfile

```

FROM nginx

RUN rm /etc/nginx/conf.d/default.conf

ADD default.conf /etc/nginx/conf.d/

COPY dist/ /usr/share/nginx/html/

```

构建镜像

```

docker build -t intelligence-bank-web:v1.0 .

docker stop intelligence-bank-web

docker rm intelligence-bank-web

```

运行

```

docker run -d -p 80:80 --name intelligence-bank-web intelligence-bank-web:v1.0

```

后端

安装Kong

创建容器网络

```

docker network create kong-net

```

启动postgre数据库

```
docker run -d --name kong-database \  
    --network=kong-net \  
    -p 5432:5432 \  
    -v $HOME/kong/postgres-data:/var/lib/postgresql/data \  
    -e "POSTGRES_USER=kong" \  
    -e "POSTGRES_DB=kong" \  
    -e "POSTGRES_PASSWORD=kong" \  
    postgres:9.6
```

初始化数据库

```
docker run --rm \  
    --network=kong-net \  
    -e "KONG_DATABASE=postgres" \  
    -e "KONG_PG_HOST=kong-database" \  
    -e "KONG_PG_USER=kong" \  
    -e "KONG_PG_PASSWORD=kong" \  
    kong:latest kong migrations bootstrap
```

启动Kong

```
docker run -d --name kong \  
    --network=kong-net \  
    -e "KONG_DATABASE=postgres" \  
    -e "KONG_PG_HOST=kong-database" \  
    -e "KONG_PG_USER=kong" \  
    -e "KONG_PG_PASSWORD=kong" \  
    -e "KONG_PROXY_ACCESS_LOG=/dev/stdout" \  
    -e "KONG_ADMIN_ACCESS_LOG=/dev/stdout" \  
    -e "KONG_PROXY_ERROR_LOG=/dev/stderr" \  
    -e "KONG_ADMIN_ERROR_LOG=/dev/stderr" \  
    -e "KONG_ADMIN_LISTEN=0.0.0.0:8001, 0.0.0.0:8444 ssl" \  
    -p 8000:8000 \  
    -p 8443:8443 \  
    -p 172.16.20.10:8001:8001 \  
    -p 172.16.20.10:8444:8444 \  
    kong:latest
```

安装konga

启动konga PostgreSQL数据库。

```
docker run -d --name konga-database \  
    --network=kong-net \  
    -p 5433:5432 \  
    -v $HOME/kong/konga/postgres-data:/var/lib/postgresql/data \  
    -e "POSTGRES_USER=konga" \  
    -e "POSTGRES_DB=konga" \  
    -e "POSTGRES_PASSWORD=konga" \  
    postgres:9.6
```

初始化数据

```
docker run --rm \
    --network=kong-net \
    pantsel/konga:latest \
    -c prepare \
    -a "postgres" \
    -u "postgres://konga:konga@konga-database:5432/konga"
```

启动Konga

```
docker run -d --name konga \
    --network kong-net \
    -e "TOKEN_SECRET=secret123" \
    -e "DB_ADAPTER=postgres" \
    -e "DB_URI=postgres://konga:konga@konga-database:5432/konga" \
    -e "NODE_ENV=development" \
    -p 1337:1337 \
    pantsel/konga
```

konga访问地址

<http://172.16.20.10:1337/>

Connection

```
admin
1234567
```

安装Nacos

```
docker run --name nacos-standalone -e MODE=standalone -e JVM_XMS=512m -e
JVM_XMX=512m -e JVM_XMN=256m -p 8848:8848 -d nacos/nacos-server:latest

docker container update --restart always nacos-standalone
```

[访问地址](#)

```
nacos
nacos
```

安装Consul

```
docker run --name consul -d -p 8500:8500 -p 8300:8300 -p 8301:8301 -p 8302:8302 -
p 8600:8600/udp consul:1.8.8 agent -dev -client=0.0.0.0

docker container update --restart always consul
```

[访问地址](#)

数据库连接 172.16.20.10:3306

使用go部署

部署srv

```
cd /home/bjdx/idea-bank-workspace/service/intelligence-bank-srv

go build -o user-service ./user-srv/main.go

nohup ./user-service > user-service.log 2>&1 &

go build -o policy-service ./policy-srv/main.go

nohup ./policy-service > policy-service.log 2>&1 &
```

部署api

```
go build -o business-opportunity-api-interface ./business-opportunity-api/main.go

nohup ./business-opportunity-api-interface > business-opportunity-api-
interface.log 2>&1 &

go build -o policy-api-interface ./policy-api/main.go

nohup ./policy-api-interface > policy-api-interface.log 2>&1 &

go build -o user-api-interface ./user-api/main.go

nohup ./user-api-interface > user-api-interface.log 2>&1 &
```

路由配置

```
172.16.20.10

/u/v1/business/crawlingBusinessOpportunity

/u/v1/business/getByCondition

/u/v1/policy/getRegionByCondition

/u/v1/policy/getPolicyDataByCondition

/u/v1/policy/getChinaDepartmentList

/u/v1/policy/getFunctionalRegionList

/u/v1/policy/getSpecialRegionList

/u/v1/user/login
```


项目进度

商机库开发完成，政策库开发到 top50 市政府中的扬州市，商机库已经完成了定时任务，政策库还未开发完，开发完成之后需要加入定时任务，可以参考一下商机库的实现方式。