

环境配置文档

本环境配置文档用于配置一个容器来：

- 安装pgvector-baseline
- 安装我们的改良版pgvector
- 配置ANN-benchmark测试pgvector-baseline和各种改进版本

以下的配置用于从头创建一个容器，你也可以联系 2200013088@stu.pku.edu.cn 来获取我们配置完成的autodl容器。

在以下的操作中，我们假设你的工作目录是 {workspace}

安装pgvector-baseline

回到工作目录 `cd {workspace}`

运行 `git clone https://github.com/pgvector/pgvector.git pgvevtor-baseline`

安装我们的改良版pgvector

回到工作目录 `cd {workspace}`

运行 `git clone https://github.com/mindtravel/pgvector.git`

配置ANN-benchmark测试pgvector_ivfflat

环境配置

原版ann-benchmark是依靠在宿主机上为每个查询算法建立一个docker容器来测试的，但是目前我们的服务器本身就是一个docker容器。为了避免docker套docker的麻烦，且我们只需要测试pgvector_ivf，在测试不同版本的pgvector_ivf时，我们不再建立新容器，而是编译不同版本的vector.so扩展，并将其替换到postgresql中，然后运行ann-benchmark进行测试

具体配置步骤如下：

1. 拉取仓库

回到工作目录 `cd {workspace}`

运行 `git clone https://github.com/mindtravel/ann-benchmarks.git`

运行 `cd ann-benchmarks`

2. 创建并激活conda环境

运行 `conda create -n ann-benchmarks python==3.10.6`

运行 `conda activate ann-benchmarks`

3. 安装python依赖

运行 `pip install -r requirements.txt`

4. 安装各个ivf算法的依赖

测试ivf算法：运行 `./scripts/env/pgvector_ivf.sh` 安装pgvector_ivf算法环境（脚本参考 `ann_benchmarks/algorithms/pgvector_ivfflat`）

测试cuvs算法：运行过 `./scripts/env/pgvector_ivf.sh` 的基础上，运行 `conda install -y -c conda-forge -c nvidia -c rapidsai cupy cuvs`

运行测试

测试脚本在目录 `scripts/tests` 中，各个测试的内容如下：

- `compile.sh`：编译pgvector baseline或我们的改良版pgvector
- `pgvector_origin.sh`：测试pgvector的原始版本（ivfflat多线程，ivfflat单线程【可选】）
- `pgvector_ours.sh`：测试pgvector的改良版本（ivfflat多线程，ivfjl多线程）
- `cuvs.sh`：测试cuvs算法（cuvs-ivfflat，cuvs-ivfpq）
- `test_all.sh`：在所有数据集上运行所有测试并绘制图像到 `{workspace}/ann-benchmarks/results`
- `test_simple.sh`：在一个较小的数据集上运行所有测试并绘制图像到 `{workspace}/ann-benchmarks/results`

回到工作目录 `cd {workspace}`

运行 `./scripts/tests/test_all.sh` 即可

测试结果：以glove-100-angular测试为例

