1) 实验中遇到的问题及解决办法说明

在 Linux 环境下安装软件依赖包。

[root@ecs-5045~]# yum install -y autoconf gcc gcc-c++ libaio-devel flex bison ncursoreadline-devel dkms python3-devel pam-devel libffi-devel automake byacc cmake o

在 Linux 环境下安装 Git 工具并配置 Git 环境。

```
[root@ecs-5045~]# yum install git -y ---安装 git 工具
[root@ecs-5045~]# git config --global user.name "Your Name" --- 配置
[root@ecs-5045~]# git config --global user.email "email@example.com" --- 配置
```

下载时报错: error:GPG check Filed

修改

```
[root@ecs-5045 ~]# yum install -y autoconf gcc gcc-c++ libaio-devel flex bison ncurses-devel glibc-devel patch readline readline-devel dkms python3-devel pam-devel libffi-devel automake byacc cmake openssI-devel libtool-devel --nogpgcheck

[root@ecs-5045 ~]# yum install git -y --nogpgcheck ---安装 git 工具
```

(2)实验耗费时长说明(各主要步骤大致所需时间

```
G + 0 13 + 2 1 1 1 7 + C +
                                          ▼ ▶ 附加... ▼ 🛝 🚳 🛫 🔽 ▼ ▶ ▼ 💺 🖆 🖷 🗎 🤼 📜 🐧 剂 🐴 🛫
                                              builtins.h varlena.cpp CMake 概述页
          builtin_funcs.ini ≠ × set_thirdparty_path.cmake
                   AddBuiltinFunc(_0(DEFAULTFORMATFUNCOID), _1("format"), _2(1), _3(false)
    3105
    3106
               AddFuncGroup(
    3107 ⊑
                   "format_type", 1,
    3108
                   AddBuiltinFunc(_0(1081), _1("format_type"), _2(2), _3(false), _4(false)
    3109
    3110
               AddFuncGroup(
    3111 □
    3112
                "gauss_hello", 1,
                   AddBuiltinFunc(_0(GAUSSHELLOFUNCOID), _1("gauss_hello"), _2(1), _3(true
    3113
    3114
    3115 ⊟
             AddFuncGroup(
                   "generate_series", 8,
    3116
                   AddBuiltinFunc (_0 (938), _1 ("generate_series"), _2(3), _3 (true), _4 (true
    3117
                   AddBuiltinFunc(_0(GENERATESERIESFUNCOID), _1("generate_series"), _2(3),
    3118
                   AddRuiltinFuna ( 0(1066)
                                             1 ("ganarata carias")
                                                                   2(3)
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.
:\Windows\System32>git config -global user.name "FufuWei"
error: did you mean `--global` (with two dashes)?
 :\Windows\System32>git config --global user.name "FufuWei"
 :\Windows\System32>git config --global user.email "721385959@qq.com"
 :\Windows\System32>ssh-keygen -t rsa -C "721385959@qq.com"
Generating public/private rsa key pair.

Enter file in which to save the key (C:\Users\HONOR/.ssh/id_rsa):

Enter passphrase (empty for no passphrase):
     same passphrase again:
Your identification has been saved in C:\Users\HONOR/.ssh/id_rsa
Your public key has been saved in C:\Users\HONOR/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ubahc714QXrfwQPANr06S6XCweeiSv7YzaRn185IM2A 721385959@qq.com
The key's randomart image is:
   -[RSA 3072]--
    create mode 100644 out/build/x64-Debug/CMakeFiles/ShowIncludes/foo.h
tem: create mode 100644 out/build/x64-Debug/CMakeFiles/ShowIncludes/main.c
nic/ create mode 100644 out/build/x64-Debug/CMakeFiles/ShowIncludes/main.obj
create mode 100644 out/build/x64-Debug/CMakeFiles/cmake.check_cache
   create mode 100644 out/build/x64-Debug/VSInheritEnvironments.txt
 rirC:\openGauss-server\openGauss-server>git push origin master
 Enumerating objects: 61, done.
   Counting objects: 100% (61/61), done.
  Delta compression using up to 8 threads
   Compressing objects: 100% (37/37), done.
   Writing objects: 100% (47/47), 90.45 KiB | 2.15 MiB/s, done.
推订otal 47 (delta 18), reused 1 (delta 0), pack-reused 0 remote: Powered by GITEE.COM [GNK-6.4]
   To https://gitee.com/fufuyd/openGauss-server.git
       f70af3cd8..965d629ba master -> master
   C:\openGauss-server\openGauss-server>
                Welcome to 4.19.90-2110.8.0.0119.oe1.aarch64
                System information as of time: Tue Jun 6 20:2
                ST 2023
                System load:
                                         0.00
                Processes:
                                         186
                Memory used:
                                         1.4%
                Swap used:
                                         0.0%
                Usage On:
                                         42%
                IP address:
                                         192.168.0.77
                Users online:
```

Froot@ecs-5539 ~1# cd script

- 2.搭建开发环境, 10min
- 3.二次开发(gauss_hello 函数为例),15min,
- 4. 编译环境要求:约 45min,
- 5. 验证二次开发结果 20min
- (3) openGauss 其他二次开发创意说明(实现的或未实现的二次开发创意说明均可,实现的创意可说明具体实现步骤,未实现的创意可说明拟实现思路)。

基于 openGauss 的机器学习平台,可以基于 openGauss 实现机器学习算法和模型的训练和部署。(只是想法)

数据预处理:使用 openGauss 提供的数据处理功能,对原始数据进行清洗、归一化、特征提取等预处理操作。

数据存储:将预处理后的数据存储到 openGauss 数据库中,使用 openGauss 的高性能和可扩展性,为后续的机器学习处理提供支持。

模型训练:使用 openGauss 提供的数据处理和机器学习算法,对存储在 openGauss 数据库中的数据进行模型训练。

模型部署:将训练好的模型部署到 openGauss 数据库中,以便后续的机器学习处理能够直接使用。

数据分析和预测:使用 openGauss 提供的机器学习算法和模型,对存储在 openGauss 数据库中的数据进行分析和预测,为用户提供数据分析和决策支持等功能。