一、复现平台

阿里云 GPU 计算云服务器，配置如下：

1.vCPU：4 核 8 线程（ ecs.gn6i-c8g1.2xlarge）

2.内存：30GB

3.GPU：NVIDIA T4（支持 CUDA 11.7.1）。

4.镜像：Ubuntu 22.04 LTS

5.系统盘：ESSD 云盘，100GB。

二、复现步骤

1. 编译并安装项目定制内核 6.0-lake

①首先克隆了 LAKE 代码库和 linux-6.0，本步执行了 linux-6.0 目录下的 full_compilation.sh。

②full_compilation.sh 首先复制当前内核的配置 (/boot/config-$(uname -r)) 到 .config。

③调用 set_configs.sh，该脚本会修改 .config，启用 CMA 等特性，并设置 LOCALVERSION="-lake"。make olddefconfig 根据这些更改更新 .config 文件，使用默认值填充所有未明确设置的新选项。

④调用 compile_install.sh，该脚本会启动实际的内核编译，安装模块，安装内核，并更新 GRUB。

```
  HDRINST usr/include/asm/sembuf.h
  HDRINST usr/include/asm/bitsperlong.h
  HDRINST usr/include/asm/sigcontext32.h
  HDRINST usr/include/asm/mman.h
  HDRINST usr/include/asm/siginfo.h
  HDRINST usr/include/asm/posix_types.h
  HDRINST usr/include/asm/debugreg.h
  HDRINST usr/include/asm/unistd.h
  HDRINST usr/include/asm/svm.h
  HDRINST usr/include/asm/byteorder.h
  HDRINST usr/include/asm/amd_hsmp.h
  HDRINST usr/include/asm/msr.h
  HDRINST usr/include/asm/ptrace.h
  HDRINST usr/include/asm/statfs.h
  HDRINST usr/include/asm/mce.h
  HDRINST usr/include/asm/kvm_para.h
  HDRINST usr/include/asm/sgx.h
  HDRINST usr/include/asm/processor-flags.h
  HDRINST usr/include/asm/prctl.h
  HDRINST usr/include/asm/sigcontext.h
  HDRINST usr/include/asm/errno.h
  HDRINST usr/include/asm/bpf_perf_event.h
  HDRINST usr/include/asm/unistd_64.h
  HDRINST usr/include/asm/ioctls.h
  HDRINST usr/include/asm/unistd_x32.h
  HDRINST usr/include/asm/fcntl.h
  HDRINST usr/include/asm/ipcbuf.h
  HDRINST usr/include/asm/poll.h
  HDRINST usr/include/asm/sockios.h
  HDRINST usr/include/asm/param.h
  HDRINST usr/include/asm/resource.h
  HDRINST usr/include/asm/termios.h
  HDRINST usr/include/asm/termbits.h
  HDRINST usr/include/asm/types.h
  HDRINST usr/include/asm/socket.h
  HDRINST usr/include/asm/unistd_32.h
  HDRINST usr/include/asm/ioctl.h
  INSTALL /usr/include
Success!
root@iZ2ze4yygw64fe5ve842g4Z:~/linux-6.0#
```

2. 配置 GRUB 引导新内核

①查找新内核的 GRUB ID：

cat /boot/grub/grub.cfg | grep submenucat /boot/grub/grub.cfg | grep option | grep 6.0.0-lake

得到一个高级菜单 ID 和一个内核 ID。

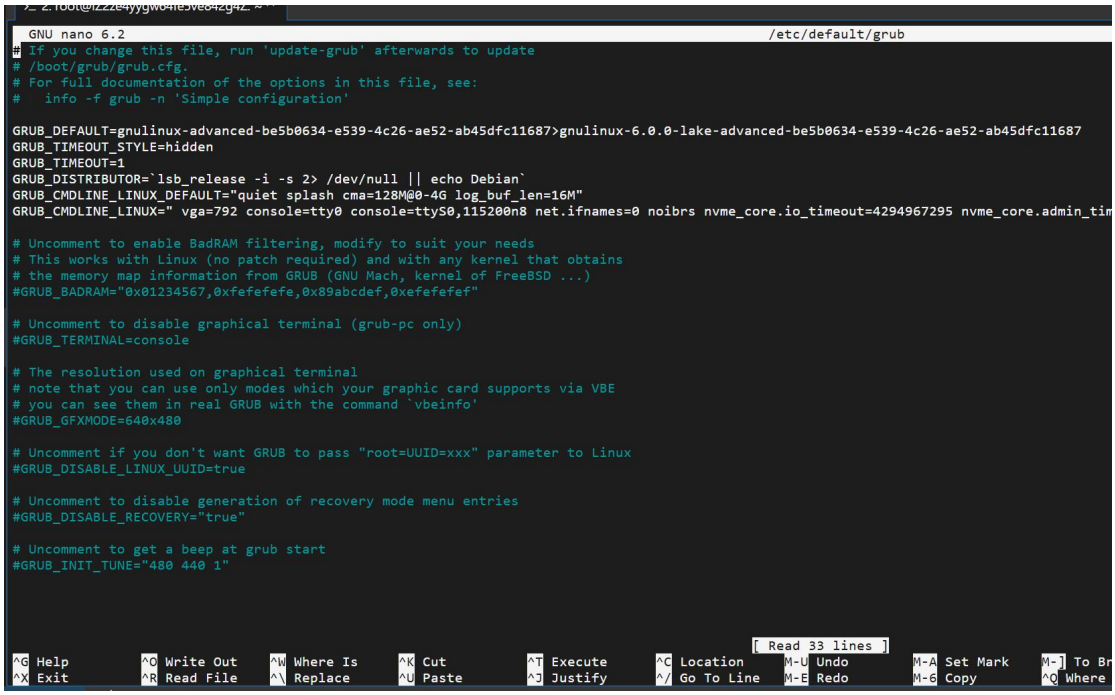
②修改 /etc/default/grub 文件：

sudo nano /etc/default/grub

添加 GRUB_DEFAULT 行 GRUB_DEFAULT="高级菜单 ID>6.0.0-lake 内核 ID"

修改 GRUB_CMDLINE_LINUX_DEFAULT 行

GRUB_CMDLINE_LINUX_DEFAULT="quiet splash cma=128M@0-4G log_buf_len=16M"



③更新 GRUB 配置

sudo update-grub # 应用 GRUB 配置更改

验证新内核是否生效：

uname -rcat /proc/cmdline | grep "cma=128M@0-4G"

3. 安装 CUDA 11.7 和 NVIDIA 驱动

在新编译并成功启动的 6.0.0-lake 内核上进行，NVIDIA 驱动与内核版本相关。禁用或卸载开源的 nouveau 驱动，以避免冲突。

①下载 CUDA 安装包：

wget

https://developer.download.nvidia.com/compute/cuda/11.7.1/local_installers/cuda_11.7.1_515.65.01_linux.run

②安装 CUDA 工具包和驱动：

sudo sh cuda_11.7.1_515.65.01_linux.run --toolkit --driver --silent

验证 CUDA 和驱动安装：

nvidia-smi

nvcc --version

输出：

```
root@iZ2ze4yygw64fe5ve842g4Z:~# nvidia-smi
Mon May 26 22:31:57 2025
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.76       Driver Version: 515.76       CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:07.0 Off |                    0 |
| N/A   34C    P0    27W /  70W |      2MiB / 15360MiB |      5%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

```
root@iZ2ze4yygw64fe5ve842g4Z:~# nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2021 NVIDIA Corporation
Built on Thu_Nov_18_09:45:30_PST_2021
Cuda compilation tools, release 11.5, V11.5.119
Build cuda_11.5.r11.5/compiler.30672275_0
```

4. 测试编译好的内核

首先运行 create_venv.sh 来创建一个 python 虚拟环境。

运行项目仓库中提供的基本测试脚本：./basic_test.sh。

```
>_ 2. root@iZ2ze4yygw64fe5ve842g4Z: ~/LAKE    create_venv.sh

rmmod: ERROR: Module lake_shm is not currently loaded
> All unloaded. Loading them now.
> Loading shared memory module
~/LAKE/src/kapi/kshm ~/LAKE/src/kapi ~/LAKE
~/LAKE/src/kapi ~/LAKE
> Done.
> Loading kernel API remoting module
~/LAKE/src/kapi/kernel ~/LAKE/src/kapi ~/LAKE
~/LAKE/src/kapi ~/LAKE
> Done.
> *************    PLEASE READ    ****************
> We are about to run the user space daemon.
> If you see repeating netlink errors, press ctrl+c and start again. This means the
> user space application cannot communicate with the shared memory module.
> If you still can't run when trying this script again, run lsmod and check if the lake_* modules are loaded,
> if they are and are in use by one more modules, ** you need to restart your machine. **
> Make sure you added cma=128M@0-4G to your kernel parameters.
> ************************************************
~/LAKE/src/kapi/uspace ~/LAKE/src/kapi ~/LAKE
~/LAKE/src/kapi ~/LAKE
> Done. Waiting for things to settle..
Starting uspace lake kapi with pid 60390
Netlink connected, message sent to kernel
Request shared memory size: 32 MB   33554432
mmap shared memory region to 0x7fbfbd800000, size=0x2000000
> Checking if the user space daemon is running...
                                                > Looks like it is.
                                                                ~/LAKE
                                                                     ~/LAKE/src/hello_driver ~/LAKE
                                                                                                  > ************************************

**********
          > Running hello world kernel module that uses CUDA.
                                                               > Success! run dmesg if you want to see the output
                                                                                                                 > ******************************

******************
                ~/LAKE
                       > Unloading everything...
                                                 > Finished!
```