FAQ-cuda 以及 hip 移植常见问题处理经验

问题一、纹理内存报错：

```
Bash
1 /data/wkx/develop/ppl.cv/src/ppl/cv/cuda/warp.hpp:31:8: error: no templ
  ate named 'texture'
2 static texture<float4, cudaTextureType2D,
3        ^
4 /data/wkx/develop/ppl.cv/src/ppl/cv/cuda/warp.hpp/data/wkx/develop/ppl.
  cv/src/ppl/cv/cuda/warp.hpp::3131::88::
5 error: error: no template named 'texture'no template named 'texture'
6
7 static texture<float4, cudaTextureType2D,
8 static texture<float4, cudaTextureType2D,
```

解决方法：

CUDA 的 `texture` 类型在较新版本（CUDA 12 及以上）中已被弃用或移除。旧版 CUDA（如 CUDA 10 或 11）中可以使用 `texture<T, ...>` 这种全局变量声明方式，但在 **CUDA 12+ 中，这种语法不再支持**，必须改用 `cudaTextureObject_t` + `cudaResourceDesc` / `cudaTextureDesc` 的方式来创建纹理对象

使用 DCU 的 cuda-11.8 编译老旧代码即可顺利通过；

问题二、 launch bounds (256) 报错：

Launch params (1024, 1, 1) are larger than launch bounds (256) for kernel _ZL12rms_norm_f32ILi1024EEvPKfPfif please add launch_bounds to kernel define or use --gpu-max-threads-per-block recompile program !

解决方法：

解决方法 1：

```
Bash
1  所有的核函数 __global__ 替换为 __global__ __launch_bounds__(1024)
```

解决方法 2：

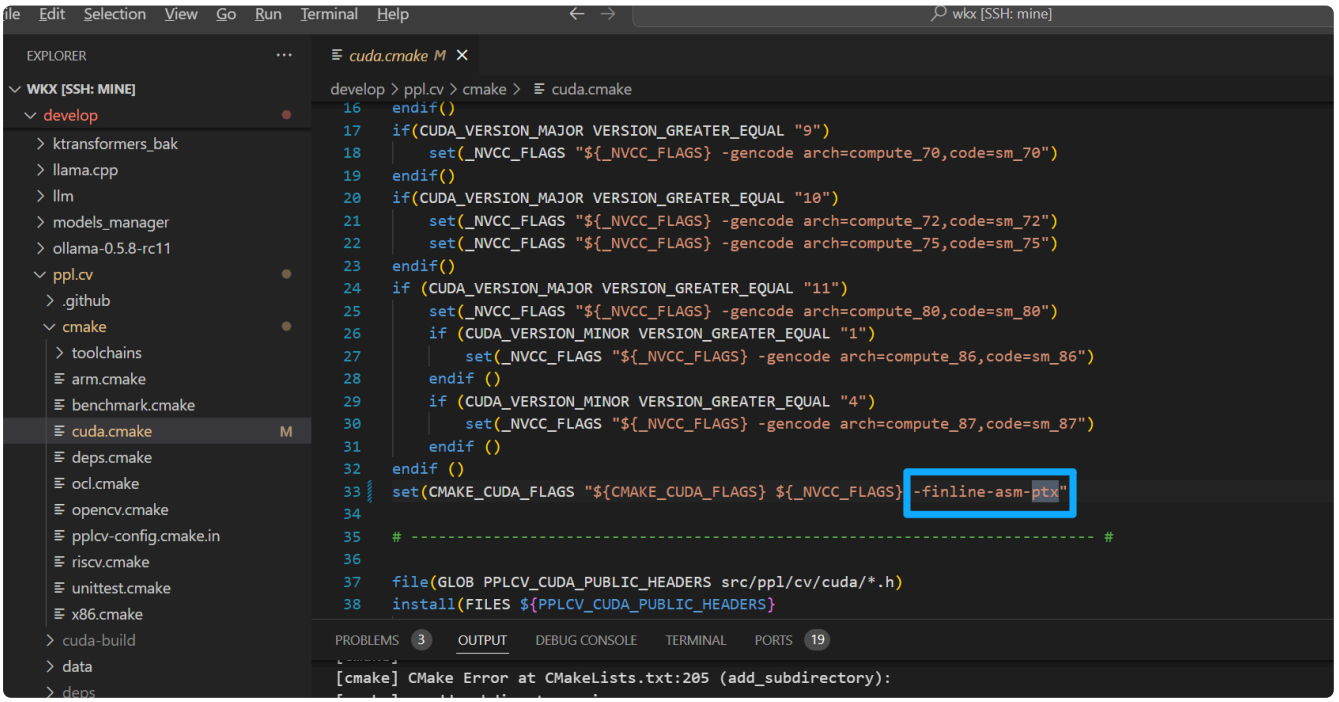
nvcc 或者 hip 编译增加：`--gpu-max-threads-per-block=1024`

问题三、asm 代码，内联汇编代码编译报错；

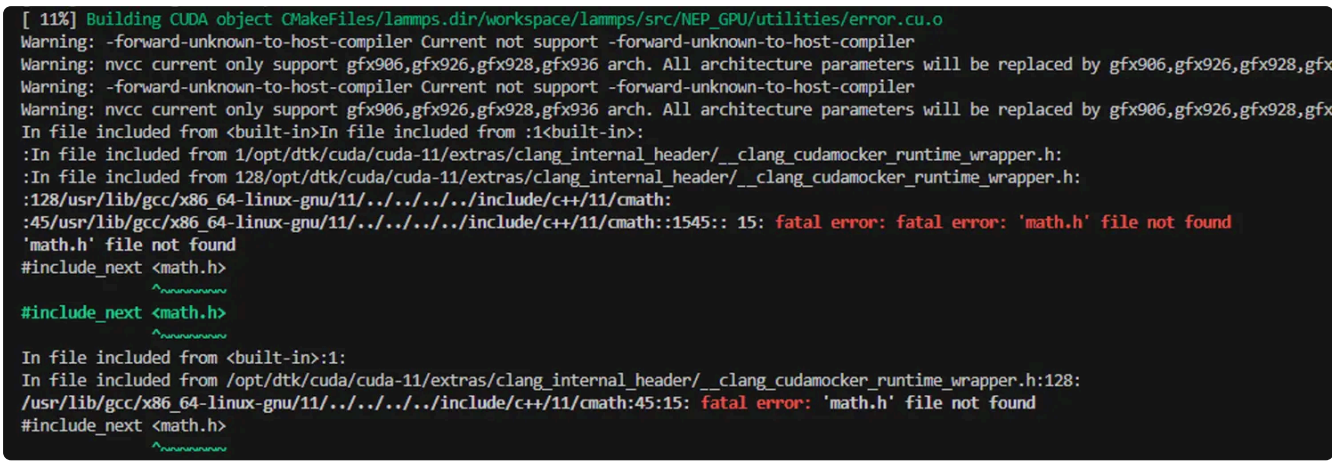
```
__DEVICE__
uchar saturateCast(int value) {
  unsigned int result = 0;
  asm("cvt.sat.u8.s32 %0, %1;" : "=r"(result) : "r"(value));
  return result;
}

__DEVICE__
uchar saturateCast(short value) {
  unsigned int result = 0;
  asm("cvt.sat.u8.s16 %0, %1;" : "=r"(result) : "h"(value));
  return result;
}
```

解决方法：

内嵌 PTX 功能开启需要主动加“-fnline-asm-ptx”选项。



问题四、 cuda 应用不转码适配找不到 math.h 头文件



解决方法：

cmake 编译中增加的 -isystem /usr/include 与 nvcc 编译器同时使用会存在冲突。

开启打印，关注编译过程的 完整头文件、库文件的依赖，去掉 -isystem /usr/include 即可编译成功。

make VERBOSE=1 <project>

问题五、使用开源的 pycuda 无法编译 cu 文件

解决方法：

参考这个，更改下 compiler.py 适配 hip 编译；

https://ontrack.hygon.cn/browse/CSD-10705

问题六、如何针对一个文件夹的 cu 代码进行转码

详细可以参考：

DCU应用移植介绍-程顺延

解决方法：

```Bash
hipconvertinplace-perl.sh <cuda代码文件夹>
```

cuda 文件夹下原有的代码，转码后以 org-name.h/cu.prehip 形式存储在当前目录

由于要使用 hip 编译 , 因此所有的 cu 后缀 , 修改为 hip 或者 cpp;

问题七、hip 转码后部分宏定义不规范不会被转换，可能导致出现问题：

解决方法：

- CublasHandleManager.h

```
Plain Text
1 #if !defined(ROCM_SYMLINK_HIPBLAS_H)
2 #error hipblas.h must be included at the very top of any file includin
  g CublasHandleManager.h
3 #endif
4
5 从 CUBLAS_V2_H_ 更改为  ROCM_SYMLINK_HIPBLAS_H
```

问题八、 math_constants.h 找不到：

解决方法：

DTK 的 cuda 下有 math_constants.h 会被别的工程依赖；

hip 下不存在对应的代码，可以直接拷贝 math_constants.h 到工程中使用；

math_constants.h 仅仅是一些数学值的定义；

问题九、转码后部分 hip 核函数不识别 min：

解决方法：

EddyMatrixKernels.cpp 中不支持 min 的问题解决

Bash

```
__global__ void QR(// Input
                   const float  *K,     // Row-first matrices to decomp
ose
                   unsigned int m,      // Number of rows of K
                   unsigned int n,      // Number of columns of K
                   unsigned int nmat,   // Number of matrices
                   // Output
                   float        *Qt,    // nmat mxm Q matrices
                   float        *R)     // nmat mxn R matrices
{
  extern __shared__ float scratch[];

  if (blockIdx.x < nmat && threadIdx.x < m) {
    unsigned int id = threadIdx.x;
    // unsigned int ntpm = min(m,blockDim.x); // Number of threads per
matrix
    unsigned int ntpm = (m < blockDim.x) ? m : blockDim.x;
    float *v = scratch;
    float *w = &scratch[m];
    const float *lK = &K[blockIdx.x*m*n];
    float *lQt = &Qt[blockIdx.x*m*m];
    float *lR = &R[blockIdx.x*m*n];
    qr_single(lK,m,n,v,w,id,ntpm,lQt,lR);
  }
  return;
}
```

问题十、使用 DTK-25.04 之后的软件栈编译报头文件错：

解决方法：

尽量尝试使用 `-std=c++17` \ `-std=c++14`

问题十一、g++ 编译 hipRuntime（hipMalloc、hipMemcpy）等接口代码，编译报错：

解决方法：

编译时增加宏定义，

__HIP_PLATFORM_AMD__

链接依赖增加 -l galaxyhip