

卷积算子优化赛题示例程序附加说明

赛题提供了 slurm 作业脚本，选手修改代码后，直接执行 `sbatch job.sh`。脚本包括程序编译执行环境配置、编译及执行过程。脚本中的队列 `xahdexclu04` 可根据比赛时给定的队列修改。脚本内容如下：

```
#!/bin/bash
#SBATCH -J test                                # 作业名
#SBATCH -p xahdexclu04                        # 队列名 使用 whichpartition 查看
#SBATCH -N 1                                  # 节点数量
#SBATCH --ntasks-per-node=1                  # 每节点运行进程数
#SBATCH -c 8                                  # 每个进程所用 cpu 核数
#SBATCH --gres=dcu:1                          # 每个节点申请的 dcu 数量
#SBATCH -o %j.out                             # 作业标准输出
#SBATCH -e %j.out                             # 作业错误输出，这里两种输出放在了一个文件中显示
module list

module load anaconda3/5.2.0
module load compiler/devtoolset/7.3.1
module load mpi/hpcx/gcc-7.3.1
module load compiler/dtk/25.04
module list

cd ${HOME}/SUGON_PRA/conv2d-2025/ # 根据实际情况修改
make clean gpu=1
make gpu=1

./conv2ddemo 128 3 225 225 32 3 3 2 2 0 0
./conv2ddemo 49 128 35 35 384 3 3 2 2 0 0
./conv2ddemo 16 128 105 105 256 3 3 2 2 0 0
./conv2ddemo 128 3 230 230 64 7 7 2 2 0 0
./conv2ddemo 2 3 838 1350 64 7 7 2 2 0 0
./conv2ddemo 256 256 28 28 256 2 2 2 2 0 0
```

1. 环境配置

加载 DTK-25.04 环境：

```
module load anaconda3/5.2.0
module load compiler/devtoolset/7.3.1
module load mpi/hpcx/gcc-7.3.1
```

```
module load compiler/dtk/25.04
```

查看所有已加载的环境：

```
module list
```

2. 示例程序编译及运行

程序编译：

进入示例程序目录，执行 `make gpu=1` 命令进行编译。

编译后的可执行程序为 `conv2ddemo`。

```
make
```

程序修改需要重新编译时，执行 `make clean gpu=1` 清理之前编译生成的文件再编译。

```
make clean
```

程序运行：

示例程序运行需要传入相关 `n`、`c`、`h`、`w`、`k`、`r`、`s`、`u`、`v`、`p`、`q` 参数

参数说明：

`argv[1]`: 输入张量的 `batchsize`，对应 `A(n)` 维度；

`argv[2]`: 输入张量的通道数，对应 `A(c)` 维度；

`argv[3]`: 输入张量的高度，对应 `A(h)` 维度；

`argv[4]`: 输入张量的宽度，对应 `A(w)` 维度；

`argv[5]`: 卷积核张量的数量，对应 `B(k)` 维度；

`argv[6]`: 卷积核张量的高，对应 `B(r)` 维度；

`argv[7]`: 卷积核张量的宽，对应 `B(s)` 维度；

`argv[8]`: 卷积核在高方向上的滑动步长，对应步长 `u`；

`argv[9]`: 卷积核在宽方向上的滑动步长，对应步长 `v`；

`argv[10]`: 卷积核在高方向上的补边，对应补边 `p`；

`argv[11]`: 卷积核在宽方向上的补边，对应补边 `q`；

3. 正确性验证

正确性验证主要时测试选手实现的计算函数返回的结果，与 `verfiy.h` 中出题方给出的对照计算函数的对比结果，相对误差不大于 $1e-9$ 。

验证函数：

```
void conv2dcpu(float* pin, float* pwei, float* pout, int n, int c, int h, int w, int k, int  
r, int s, int u, int v, int p, int q)
```

4. 计算耗时显示

程序运行后会循环计算五次，给出每次的计算耗时和平均耗时，评分是主要查看平均耗时。

5. 代码修改要求

参考赛题要求。