# 工大标志

# GPU计算实验报告

## 实验二 基于昇腾CANN的TBE算子开发（TIK）

学院：计算学部

姓名：冯开来

学号：1190201215

### 一、实验预习（10分）

1. 注册华为云账号：https://www.huaweicloud.com/
2. 课程内容预习：<https://www.hiascend.com/edu/courses>



图1 昇腾学院异构计算架构CANN中TBE算子开发（高级）



图2 本次实验主要涉及模块一到模块五

1. 问题（10分）：

（回答请使用红色文字）

1. 昇腾芯片（310或910）中的达芬奇架构具体是哪一部分？（ ）

A、整颗芯片 B、AI CPU C、AI Core D、DVPP

答：C

1. 昇腾310中有2个AI Core，8个ARM A55（有一部分部署为AI CPU）。算子开发也有两类，使用TBE算子开发运行在AI Core上，使用AI CPU算子开发运行在AI CPU上。什么样的任务适合在AI Core上执行，什么样的任务适合在AI CPU执行？

答：

AI Core适合执行：常用于各种算术逻辑简单向量运算，或内置支持的矩阵运算及池化运算，例如elewise类操作。

AI CPU适合执行：适用各类算子的开发，对于无法通过lambda表达描述的复杂计算场景也有很好的支持，例如排序类操作。

1. Unified Buffer(UB)的size为多少？最小访问粒度？

答：

size为：：256KB

最小访问粒度为：32B

1. 向量计算单元一次能够处理的数据量为多少？（ ）

A、512字节 B、256字节 C、128字节 D、64字节

答：B

1. 简述TIK算子开发过程

答：

1. 算子分析
   1. 算子开发前进行算子分析，明确算子的功能、输入、输出和逻辑表达，其中重点分析算子算法的原理，提取出算子的数学表达式。
   2. 查询TIK API接口，是否存在算子满足要求。
2. 算子实现

算子计算逻辑及调度的实现。

1. 算子编译

编译自定义算子工程，生成自定义算子安装包并进行自定义算子包的安装，将自定义算子部署到算子库（OPP）。

## 二、实验目标

1. 了解Davinci架构及其组成
2. 了解核上的计算单元和存储单元以及它们之间的数据流向
3. 了解什么是TIK及它的特点
4. 分析一个简单的TIK程序
5. 了解TIK数据操作和处理
6. 使用TIK的API接口完成一个算子开发

## 三、实验内容

完成链接中的实验：

https://www.hiascend.com/zh/college/onlineExperiment/codeLabTbeTik/tiks



图3 TIK算子开发实验

参考文档：

https://support.huaweicloud.com/tbedevg-cann503alpha2training/atlastikapi\_07\_0001.html



图4 CANN官方文档，TBE TIK相关API

## 四、实验代码（40分）

1. 实验分析（10分）

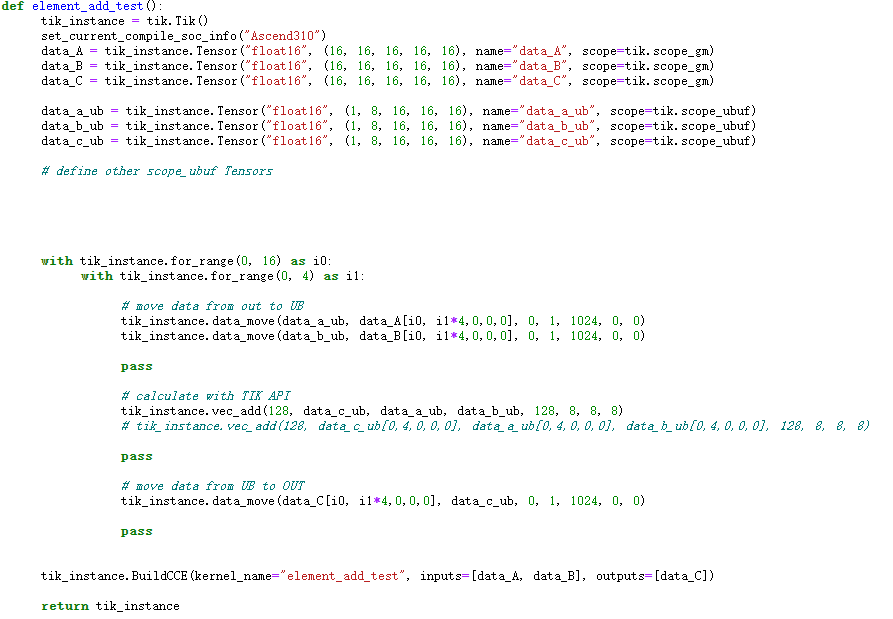
（回答使用红色文字）

我们在代码中使用了for\_range，多次从内存向UB搬运数据，然后计算，为什么？

答：对于Vector计算，一般是用Unified Buffer去存放数据，再进行计算，所以整体数据流应该是从Global Memory>Unified Buffer>Global Memory。而Global Memory的数据大小为221B，而Unified Buffer(UB)的大小为218B，所以在Global Memory>Unified Buffer的过程中，需要的多次搬运才能完成。

2. 实验代码（30分）

（粘贴对应的完整代码截图）



## 五、实验结果（40分）

（运行结果截图）



## 六、TIK商用算子开发（10分）

阅读一个Vadd商用算子开发案例：

<https://www.hiascend.com/zh/college/onlineExperiment/detail/664612>

我们会看到在一个TIK商用算子中，即使是最简单的Vadd算子，也有很多要考虑的细节问题，列举一些细节并简单介绍。

（除下面示例以外，给出1个并简单介绍，合理即给3分；2个6分；3个及3个以上10分）。

示例：

1. for\_range参数使用注意

for\_range有多核（block\_num）和多线程（thread\_num）两个参数，不可同时使用，在外部大循环使用block\_num，内部小循环使用thread\_num

1. 开启多核，在多核循环内使用到的tensor，必须要定义在多核循环内

因为多核循环外和多核循环内的tensor内存分配同时从0开始分配，可能导致地址重叠，最后导致数据错误

1. 开启double buffer时，只有定义在for循环内的tensor ，才会分配2份内存
2. 在开启多核时，begin与endt需要为int类型
3. 使用循环变量时，注意不能使用if i==0: /不能使用else:
4. 循环的执行体内不能修改for\_range的endt值，否则会导致算子执行任务挂起。