Triton调试方法及工具（上）

嘉宾：杜玉博

Triton调试方法及工具

常见的调试工具有pdb、gdb，在triton官方还有triton API可以对内核代码进行调试，下图是triton官网的Debug Ops。

Debug Ops

`static_print`	Print the values at compile time.
`static_assert`	Assert the condition at compile time.
`device_print`	Print the values at runtime from the device.
`device_assert`	Assert the condition at runtime from the device.

Triton调试方法及工具

PDB和GDB调试工具的主要区别：

1.应用场景

PDB专用于Python程序调试。

GDB主要用于C/C++程序调试，也支持其他编译语言（如Fortran）。

2.调试功能

PDB功能较为简单，适合调试Python代码，支持逐行执行、查看变量、设置断点等。

GDB功能强大，支持设置条件断点、查看内存、跟踪函数调用栈等复杂调试操作。

Triton调试方法及工具

PDB是Python内置的调试工具，可以帮助开发者逐行执行代码，查看变量值，设置断点等。以下是一些常用的PDB命令及其使用方法：

1. **启动PDB**

   - 在代码中插入`import pdb; pdb.set_trace()`，程序执行到此行时会自动进入调试模式。

2. **常用命令**

   - `l` (list): 列出当前代码行。
   - `n` (next): 执行下一行代码。
   - `s` (step): 进入函数内部执行。
   - `c` (continue): 继续执行直到遇到下一个断点。
   - `p` (print): 打印变量值，例如`p variable_name`。
   - `q` (quit): 退出调试模式。

Triton调试方法及工具

GDB是用于调试C/C++程序的强大工具，能够设置断点、查看内存、跟踪函数调用等。以下是GDB的一些基本用法：

1.启动GDB

由于是triton程序，启动命令是 gdb -args python xxx.py

2. **常用命令**

- `break` 或 `b`：设置断点，例如 `b main` 在`main`函数处设置断点。
- `run` 或 `r`：运行程序。
- `next` 或 `n`：执行下一行代码。
- `step` 或 `s`：单步执行，进入函数内部。
- `continue` 或 `c`：继续执行直到遇到下一个断点。
- `print` 或 `p`：打印变量值，例如 `p variable_name`。
- `quit` 或 `q`：退出GDB。

Triton调试方法及工具

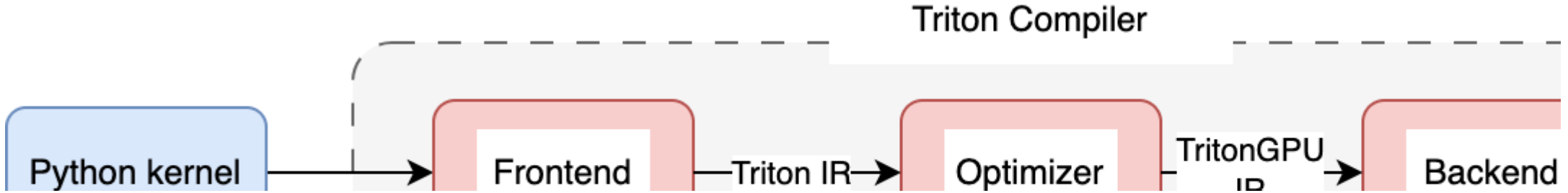
在演示调试程序之前，首先简单介绍一下Triton 新代码中的架构

总体上可以分为三大块

1.Frontend，将用户的 Python kernel code 转换为 Triton IR，以及维护 kernel launch 的 Runtime

2.Optimizer，通过各类 pass 将 Triton IR 逐步转换为优化过的 TritonGPU IR

3.Backend，将 TritonGPU IR 逐步转换为 LLVM IR，并最终通过 ptxas 编译为 cubin



Triton调试方法及工具

一、常用的调试命令，如MLIR_ENABLE_DUMP=1等

二、triton-opt介绍及使用

三、triton-translate介绍及使用

官方文档：https://github.com/triton-lang/triton/tree/main?tab=readme-ov-file#tips-for-hacking

AdvancedCompiler

Tel: 13839830713

Triton调试方法及工具（下）

嘉宾：杜玉博

Triton调试方法及工具

一、常用的调试命令，如MLIR_ENABLE_DUMP=1等

二、triton-opt介绍及使用

三、triton-translate介绍及使用

官方文档：https://github.com/triton-lang/triton/tree/main?tab=readme-ov-file#tips-for-hacking

Triton调试方法及工具

triton官方提供了一些调试方法：

1.MLIR_ENABLE_DUMP=1 打印kernel的每一个MLIR Pass前后的IR变化,如果不起作用需要清理triton cache

2.LLVM_IR_ENABLE_DUMP=1 对每个pass运行LLVM IR之前时打印kernel的IR

3.TRITON_PRINT_AUTOTUNING=1 打印kernel的最优配置和总时间

官方文档：https://github.com/triton-lang/triton/tree/main?tab=readme-ov-file#tips-for-hacking

Triton调试方法及工具

triton源码编译后的工具triton-opt、triton-translate，调试之前需要现将triton-translate、triton-opt路径添加到自己的环境变量中。

```
export PATH=/public/home/fangzx/triton-2.1.0/build/bin:
```

triton-opt：

triton-opt XX.ttir -convert-triton-to-tritongpu &> XX.ttgir （将ttir降级为ttgir）
triton-opt XX.ttgir –特定优化的编译选项 &> XX-opt.ttgir （在ttgir层面执行各种优化）

在gdb调试使用triton-opt:

gdb triton-opt
打断点b AAA.cpp:123
r XX.ttir –convert-triton-to-tritongpu &> XX.ttgir （将ttir降级为ttgir）
r XX.ttgir –特定优化的编译选项 &> XX-opt.ttgir （在ttgir层面执行各种优化）

Triton调试方法及工具

triton-translate：

triton-translate XX.ttgir –target=llvmir &> XX.llir (将ttgir文件降级到llir文件)

在gdb调试使用triton-translate:

gdb triton-translate
打断点b AAA.cpp:123
r XX.ttgir –target=llvmir &> XX.llir (将ttgir文件降级到llir文件)

先进编译实验室
Advanced Compiler

AdvancedCompiler

Tel: 13839830713