编译方向的探索

传统编译器 & 深度学习编译器

胡临天

2025 年 11 月 21 日

目录

1. 编译的本质与核心任务
2. 传统编译器与深度学习编译器
3. 现存技术瓶颈
4. 编译技术的未来发展方向

编译的本质与核心任务

编译的本质

编译的本质是对计算任务的重新认识，将程序中蕴含的计算过程转换为一种更适合分析、优化和执行的表达形式。

编译的本质

编译的本质是对计算任务的重新认识，将程序中蕴含的计算过程转换为一种更适合分析、优化和执行的表达形式。

核心任务包括：

编译的本质

编译的本质是对计算任务的重新认识，将程序中蕴含的计算过程转换为一种更适合分析、优化和执行的表达形式。

核心任务包括：

- 对计算的重新解释

编译的本质

编译的本质是对计算任务的重新认识，将程序中蕴含的计算过程转换为一种更适合分析、优化和执行的表达形式。

核心任务包括：

- 对计算的重新解释
- 对计算的形式化建模

编译的本质

编译的本质是对计算任务的重新认识，将程序中蕴含的计算过程转换为一种更适合分析、优化和执行的表达形式。

核心任务包括：

- 对计算的重新解释
- 对计算的形式化建模
- 构建从“源代码”到“执行计划”的映射

传统编译器与深度学习编译器

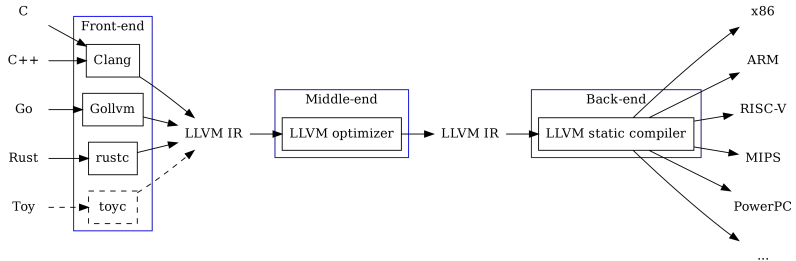
传统编译器架构

传统编译器采用三段式架构：

- 前端（Front-End）
- 中端（IR 优化）
- 后端（CodeGen）

LLVM IR 的意义：

- 跨语言的统一抽象层
- 支持多 CPU 架构目标
- 降低编译器开发复杂度



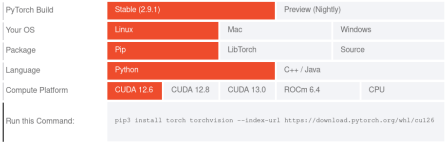
深度学习编译器的出现

深度学习模型快速增长，硬件异构化严重：

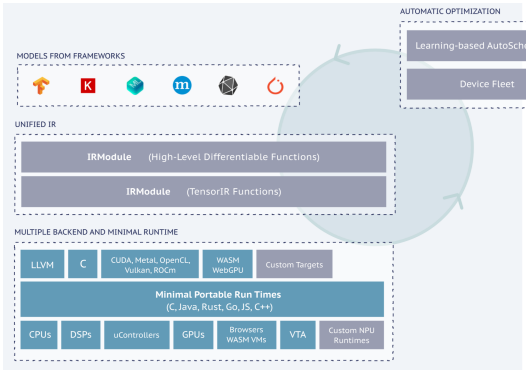
- 多框架（PyTorch / TF / ONNX / PaddlePaddle）
- 多硬件（CPU/GPU/NPU/TPU/FPGA）
- 手写算子库不可持续（cuDNN/MKL）

因此需要新的编译体系：

- 多层中间表示：Graph IR / Tensor IR
- 图优化、算子融合、布局变换
- 自动调优系统



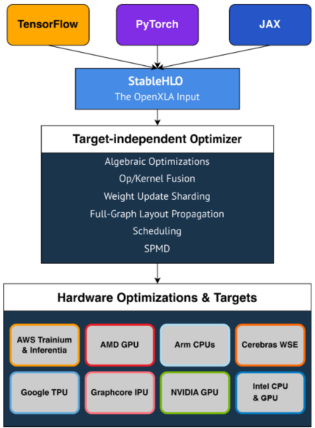
深度学习编译器



将用不同深度学习框架写的模型自动转换为不同硬件上的高性能执行代码。

- TVM：最早出现的一批深度学习编译器之一

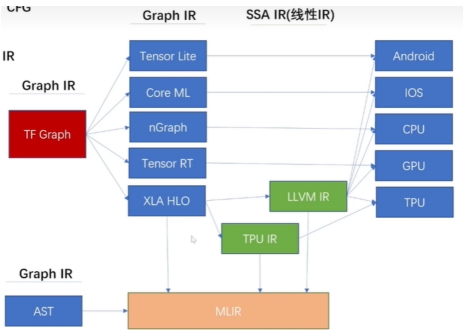
深度学习编译器



将用不同深度学习框架写的模型自动转换为不同硬件上的高性能执行代码。

- TVM：最早出现的一批深度学习编译器之一
- XAL：Google 为 TPU 做的深度学习编译器，openXAL 为其开源版

深度学习编译器



将用不同深度学习框架写的模型自动转换为不同硬件上的高性能执行代码。

- TVM：最早出现的一批深度学习编译器之一
- XAL：Google 为 TPU 做的深度学习编译器，openXAL 为其开源版
- MLIR：多层 IR 表达体系

现存技术瓶颈

传统编译器瓶颈

- 异构硬件缺乏统一抽象

传统编译器瓶颈

- 异构硬件缺乏统一抽象
- 硬件复杂度指数级提升，启发式规则越来越难写

传统编译器瓶颈

- 异构硬件缺乏统一抽象
- 硬件复杂度指数级提升，启发式规则越来越难写
- LLVM IR 难表达张量计算与层级内存结构

深度学习编译器瓶颈

- 生态碎片化：各厂商维护私有的编译器

深度学习编译器瓶颈

- 生态碎片化：各厂商维护私有的编译器
- 调度空间巨大，自动调优成本高

深度学习编译器瓶颈

- 生态碎片化：各厂商维护私有的编译器
- 调度空间巨大，自动调优成本高
- 自动生成 kernel 仍难匹配加速器供应商提供的手写库性能

编译技术的未来发展方向

未来方向：统一抽象

- 构建具备动态 Shape 与稀疏语义的 IR
- MLIR/Relax/TensorIR 的融合
- 跨硬件平台的统一语义层

未来方向：自动化优化

- 大模型驱动的调优
- 端到端可学习的编译器

未来方向：软硬件协同

- 编译器介入硬件设计
- 自动为 NPU/GPU/TPU 生成算子的最优实现
- 构建统一的可移植的运行时

Questions