CANN 开源项目预告

2025年12月20日

PyPTO开源预告

AI 加速器编程的挑战

1. 传统算子开发的复杂性

高性能算子开发一直是 AI 加速器编程的核心挑战。算子开发人员不仅需要理解算子的数学计算属性，还必须考虑如何将其转换为对硬件友好的执行方式。这种双重知识要求使得算子开发成为一个高度专业化的领域。

2. 算法开发与算子开发的鸿沟

传统的模型开发通常分为算法开发人员和算子开发人员两个角色。这种分工模式的根源在于高性能算子开发的复杂性：算法开发者专注于算法逻辑和数学表达，而算子开发者需要将算法映射到硬件执行。

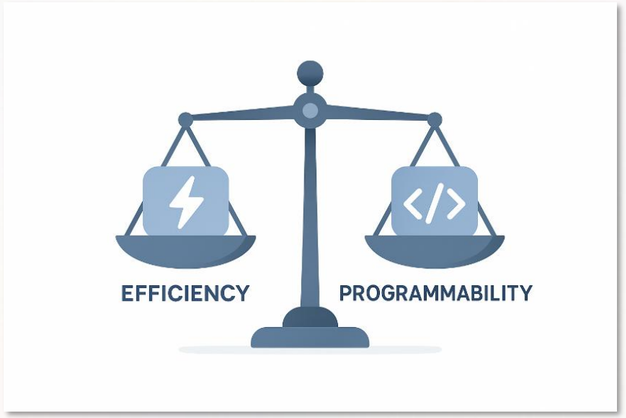
3. 硬件多样性与编程抽象的矛盾

AI 加速器硬件呈现出显著的多样性。不同硬件平台（NPU、GPU 等）具有不同的硬件特性，包括内存层次结构、执行单元类型、并行模式等。这种多样性给编程抽象带来了根本性挑战。

4. 性能优化与开发效率的平衡

AI 加速器编程对性能要求极高，需要充分利用硬件资源以实现最佳性能。然而，性能优化涉及多个复杂维度：内存管理、数据布局、算子融合、调度策略等。每个维度都需要深入的专业知识和大量的调优工作。与此同时，快速迭代和验证算法需要高效的开发工具。

PyPTO是什么?



PyPTO（发音：pai p-t-o）是 CANN 推出的一款面向 AI 加速器的高性能编程框架，旨在简化算子开发流程，同时保持高性能计算能力。该框架采用创新的 **PTO（Parallel Tensor/Tile Operation）编程范式**，以 **基于 Tile 的编程模型** 为核心设计理念，通过多层次的中间表示（IR）系统，将用户通过 API 构建的 AI 模型应用从高层次的 Tensor 图逐步编译成硬件指令，最终生成可在目标平台上高效执行的可执行代码。

PyPTO 的定位不仅限于单个算子的开发，更强调**复杂融合算子，甚至于整个模型网络**的开发能力。框架支持**分布式执行**、**自动微分**等高级特性，使其能够适应从单算子优化到大规模模型训练的多样化场景。

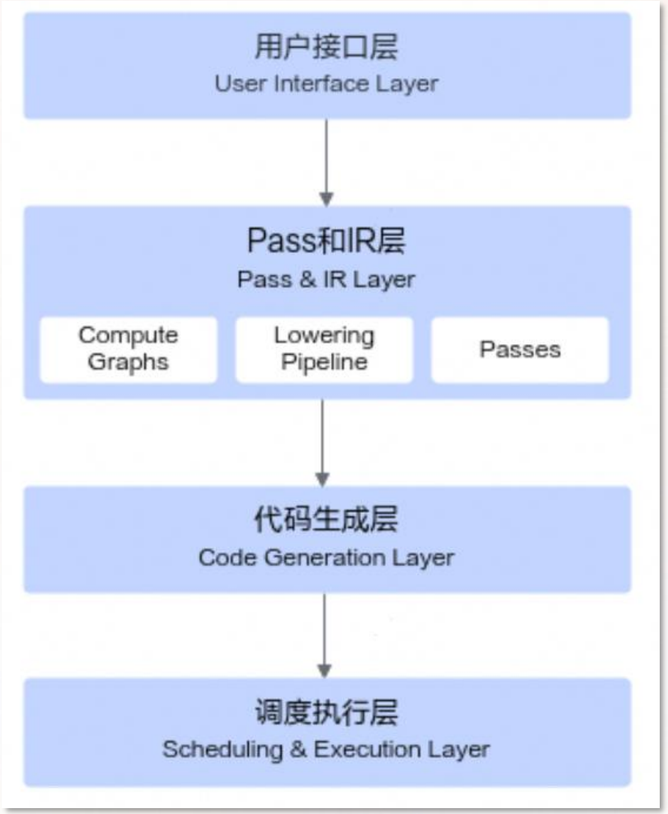
PyPTO 作为已有编程框架的**补充而非替换**，为开发者提供了另一种选择，特别适合需要深度性能优化和细粒度控制的场景。

PyPTO开源预告

PyPTO目标用户群体

PyPTO 框架面向不同层次的开发者，提供相应的抽象层次和工具支持：

- **算法开发者：**使用 Tensor 层次编程，快速实现和验证算法。他们可以专注于算法逻辑，无需关心底层硬件细节，即可获得性能良好的实现。
- **性能优化专家：**可使用 Tile 或 Block 层次，根据他们对抽象层次和控制力需求而定，进行深度性能调优。他们可以在保持易用性的同时，获得对硬件资源的细粒度控制。
- **系统开发者：**可在 Tensor/Tile/Block 和 PTO 虚拟指令集层次上进行三方框架对接或集成，以及工具链开发。他们可以利用 PyPTO 的多层次抽象，实现灵活的框架集成和扩展。
- **研究人员：**探索新的编程模型和优化技术。PyPTO 的模块化设计和开放的 IR 系统为研究提供了良好的实验平台。



本次开源主要内容

- PyPTO 框架的第一个发布版本（v0.1.0）已经完成，标志着框架从设计到实现的重大里程碑：
- **第一个发布版本（ v0.1.0 ）完成**：框架核心功能已实现，可以支持实际应用开发
- **硬件平台支持**：支持 Ascend 910B、910C 硬件平台，充分利用硬件特性
- **Tensor 层次编程**：实现 Tensor 层次编程（主要支持），为大多数开发者提供易用的编程接口
- **R 系统初步实现**：完成 Tensor Graph、Tile Graph、Block Graph 等抽象和对应的 Pass 初步实现，为后续优化奠定基础
- **完整工具链支持**：提供计算图可视化、性能分析、调试等工具，支持完整的开发工作流

开源时间 2025年12月30日

AMCT开源预告

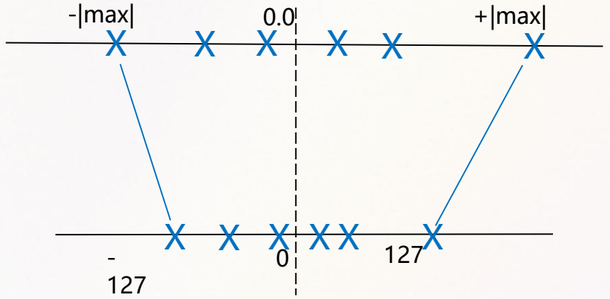
✓ 项目简介

- AMCT全称Ascend Model Compression Toolkit，是一款昇腾AI处理器亲和的模型压缩工具包。
- AMCT致力于模型的“瘦身”和“提速”，解决模型参数规模日益庞大带来的显存占用极高和计算需求极大的痛点问题。

✓ 压缩原理

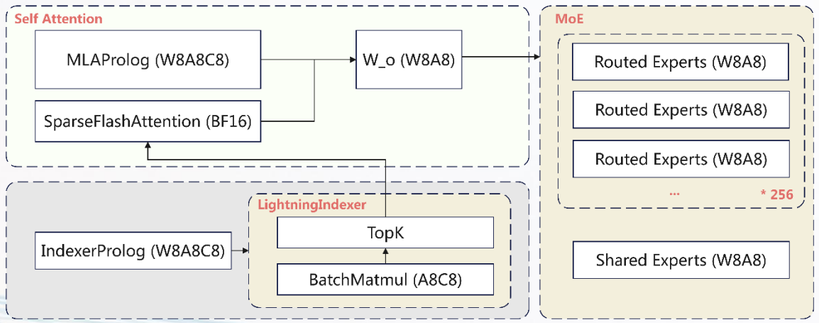
以量化为例，量化本质是使用低精度数据表示高精度数据，计算公式可以表示为 $x_{\mathrm{q}} = \mathrm{clip}(\mathrm{round}\left(\frac{x}{s} + z\right), \alpha_q, \beta_q)$

其中，x是高精度数据；x_{q}是量化后的数据；s是量化缩放因子；z是量化偏移因子；α_q, β_q是量化上下限范围



量化效果与数据分布有关，
如果数据中存在outlier，直接量化因为无法兼顾outlier和普通数据，往往会导致较大的精度损失。
算法通过调整数据分布或调整量化方式，可以提升最终的量化效果。
AMCT将这些算法集成在工具中，满足不同场景的使用述求。

✓ 应用场景



DeepSeek-V3.2-Exp模型量化部署示例

DeepSeek-V3.2-Exp模型参数量高达671B，典型的“大体积，高算力”部署场景。

相对于BF16推理，Int8量化通过降低权重位宽减小了内存占用（1200GB->600GB），同时，低比特的矩阵运算也可以有效降低端到端时延，提升系统吞吐。

A3 64卡128K长序列测试
BF16精度无损场景，TPOT小于30ms
W8A8C8量化场景，TPOT小于20ms

AMCT开源预告

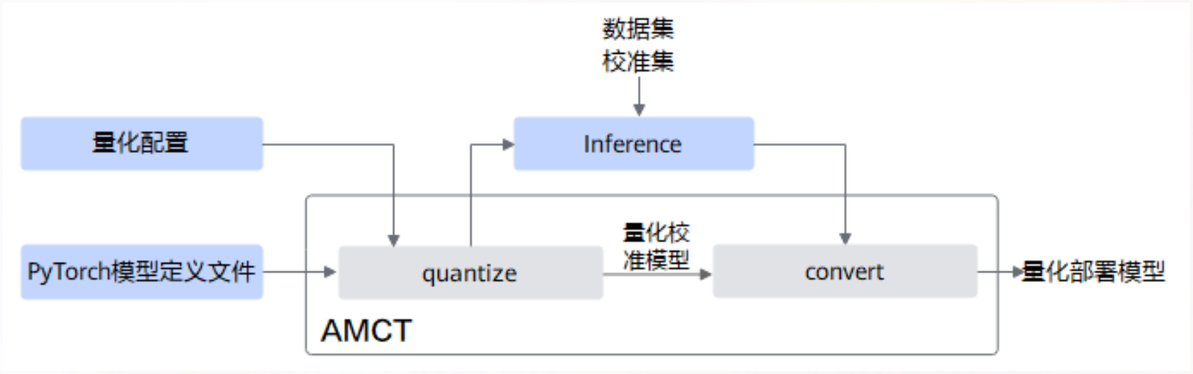
✓ 用户群体

AMCT工具面向不同的使用对象都提供了功能上的支持

- 算法开发者：AMCT提供了算法注册接口，开发者可以专注于创新性算法功能的开发，模型结构的修改等功能均由工具完成，方便开发者快速进行算法验证。
- 性能优化专家：AMCT提供了模型转换接口，量化后模型能够转换成昇腾亲和的量化部署模型，支持开发者高性能快速部署。
- 个人开发者/爱好者：AMCT提供了多种低比特量化算法，即使开发者硬件资源有限，也能够轻松完成模型部署。

✓ 开源内容

- 基于Pytorch框架的模型量化，易用的python API
- min-max/gptq/awq/smooth_quant多种量化算法，满足不同场景的量化述求
- 逐block量化算法，单卡完成deepseekV3.2大尺寸模型量化校准



开源时间 2025年12月30日