Transparent TVM Backend Acceleration
Boost ML Upstream Frameworks

Tiejun Chen
VMware OCTO, ATG
11/17/2021
Agenda
Transparent TVM Backend Acceleration

- Background
- Project MLInferBooster Introduction
- Summary
Background
Why

- TVM - A compiler stack for deep learning systems
  - Open source
  - TVM supports most AI/ML frameworks
  - TVM targets various types of AI accelerators
    - Including CPU
  - Cross-compiling
    - Host =! Target
  - Good ML inference performance

You have to
- Learn TVM
- Inspect pre-trained ML
- Get AI Acceleration info
- Call TVM APIs
- Build into your platform
  - Relay cache
  - Scheduler
  - AutoTVM
- ...

We love TVM!
Project MLInferBooster
Our solution

- **Target**
  - Power ML upstream frameworks by means of TVM

- **Goal**
  - Build a TVM Serving System
    - Backend
    - Automated
    - Unified server architecture

- **How**
  - Interpose ML framework python API
  - Built-in TVM processing – Auto {detecting, compiling, scheduling, inferencing, etc}
  - Cache
  - Scheduler
Project MLInferBooster
Auto-compiling & inferencing

- Model
  - Auto Detecting Model
  - Configuring Target Accelerator
  - Building model to TVM Relay
  - TVM Compiler

- TVM Runtime
  - Format, Shape – input, output
  - CPU, CUDA, OpenCL, ...

- predict
Project MLInferBooster

Others

- Auto-detecting AI accelerator
- Scheduler
  - Infer task <-> AI accelerator
- Autotvm
  - Flexibility
- Model cache
  - Cache the compiled model information
  - Mapping mechanism
  - Least Frequently Used (LFU) cache replacement policy
Project MLInferBooster

Demo
Project MLInferBooster

Summary

- Supported
  - Tensorflow/Pytorch/ONNX
  - {Nvidia, AMD} GPU, Xilinx FPGA, CPU

- Plan
  - Interpose C++ runtime
  - ML Serving system

Thank you!

@Tiejun_Chen
<tiejunc@vmware.com>