「AI is Everywhere」专场

透明后端图编译器无缝提升AI上游框架
Boosting AI upstream frameworks with transparent backend graph compilers seamlessly

Tiejun Chen - VMware, OCTO
2023/5/16

Agenda

Building modern AI application platform

- Problem area
- Our solution – Project Yellowstone
- Demo
- Summary

Towards modern AI application centric platform

Problem area

- Heterogeneous AI HW accelerators
- Various upstream ML frameworks
- Hard to exploit the best performance
- No such a modern AI platform with cloud native principle

Project Yellowstone I

Goal

- Build end—to—end ML service on Kubernetes from cloud to edge
  - ❑ Enable CRD based accelerators for ML serving
  - ❑ Boost ML by transparent backend acceleration

Project Yellowstone II

Enable CRD based local accelerators for ML serving

- Node feature discovery
- Device plugins
- NodeSelector
- Kubernetes Scheduler

Background

Graph compilers

- What
  - The high-level computational graph coming from ML frameworks

  - Th operations on AI device

- Graph compilers
  - Apache TVM
  - Nvidia TensorRT
  - Intel OpenVINO
  - AMD ROCM
  - Xilinx vitis AI
  - …

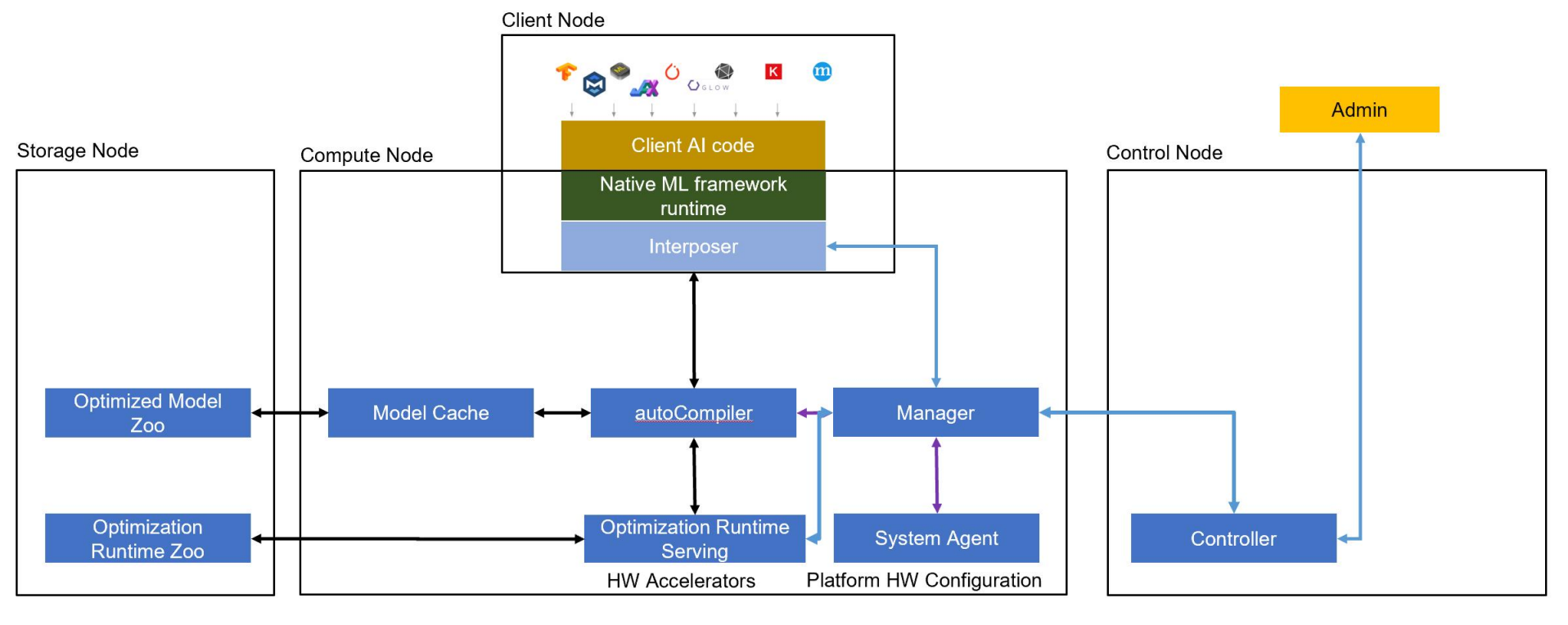
Project Yellowstone III

Overview

- Target
  - Boost ML/AI by enabling ML upstream frameworks seamlessly with graph compilers
- Design
  - Build ML Boost Serving System
    - ❑ Backend
    - ❑ Automated
    - ❑ Unified server architecture
- How
  - Interpose ML framework API
  - Built-in graph compilers processing - Auto {detecting, compiling, scheduling, inferencing, etc}

Project Yellowstone IV

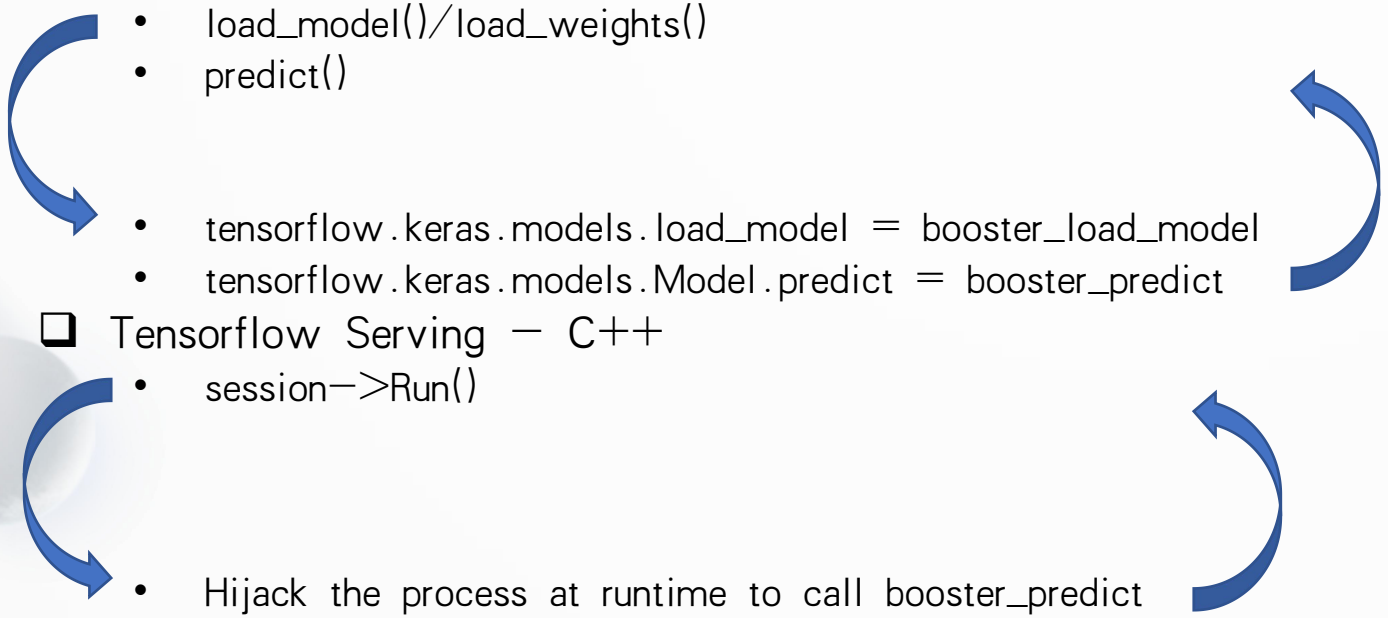
Architecture



Project Yellowstone V

Backend acceleration

- Runtime interposer
  - Target to key APIs
  - Mapping between ML Frameworks APIs and Backend APIs
- Example
  - Tensorflow – Python
    - load_model()/load_weights()
    - predict()
    - tensorflow.keras.models.load_model = booster_load_model
    - tensorflow.keras.models.Model.predict = booster_predict
  - Tensorflow Serving – C++
    - session->Run()
    - Hijack the process at runtime to call booster_predict



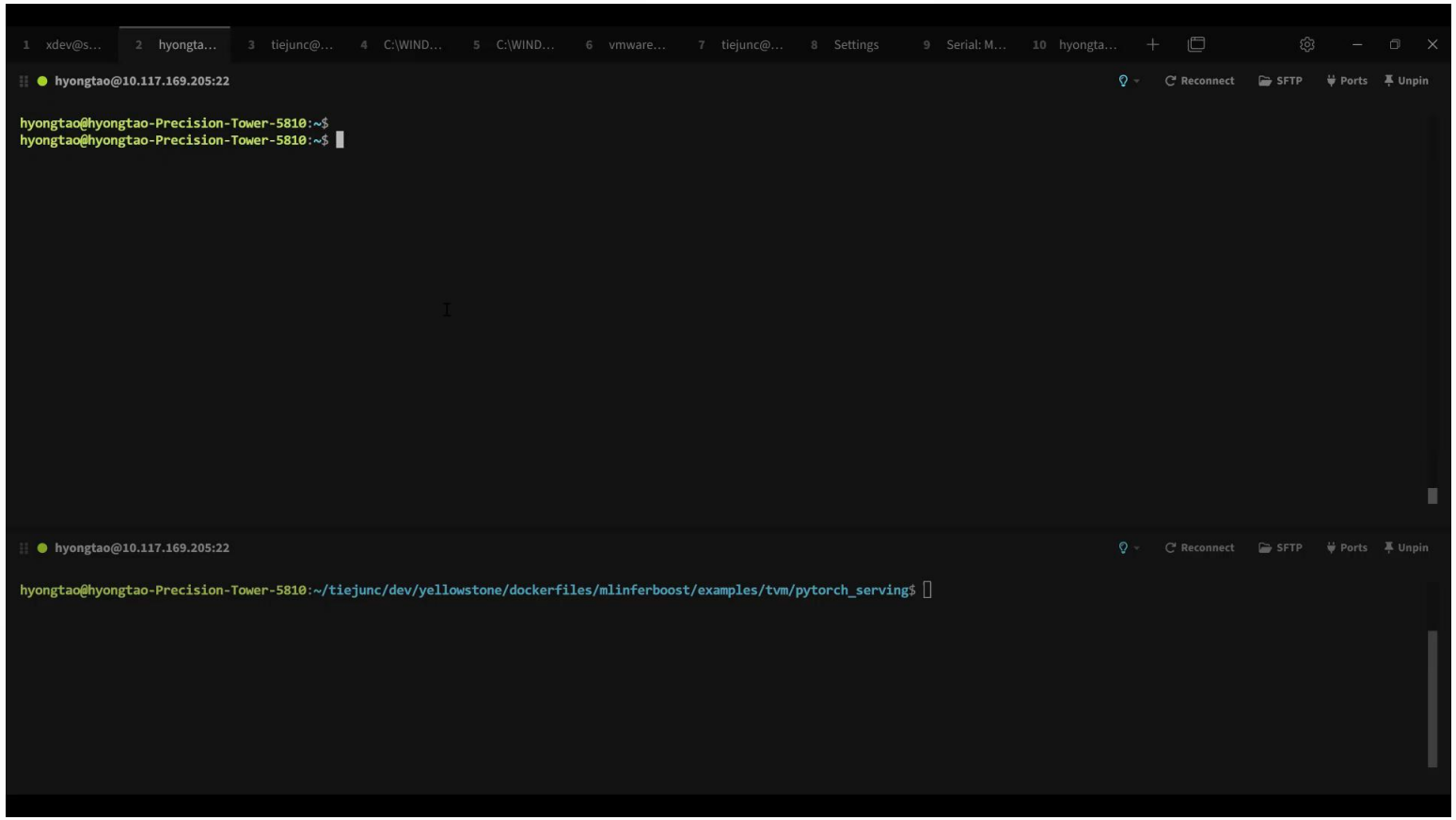
Project Yellowstone VI

Demos

- TorchServe on GPU accelerated by TVM
- Tensorflow Serving on GPU accelerated by TVM

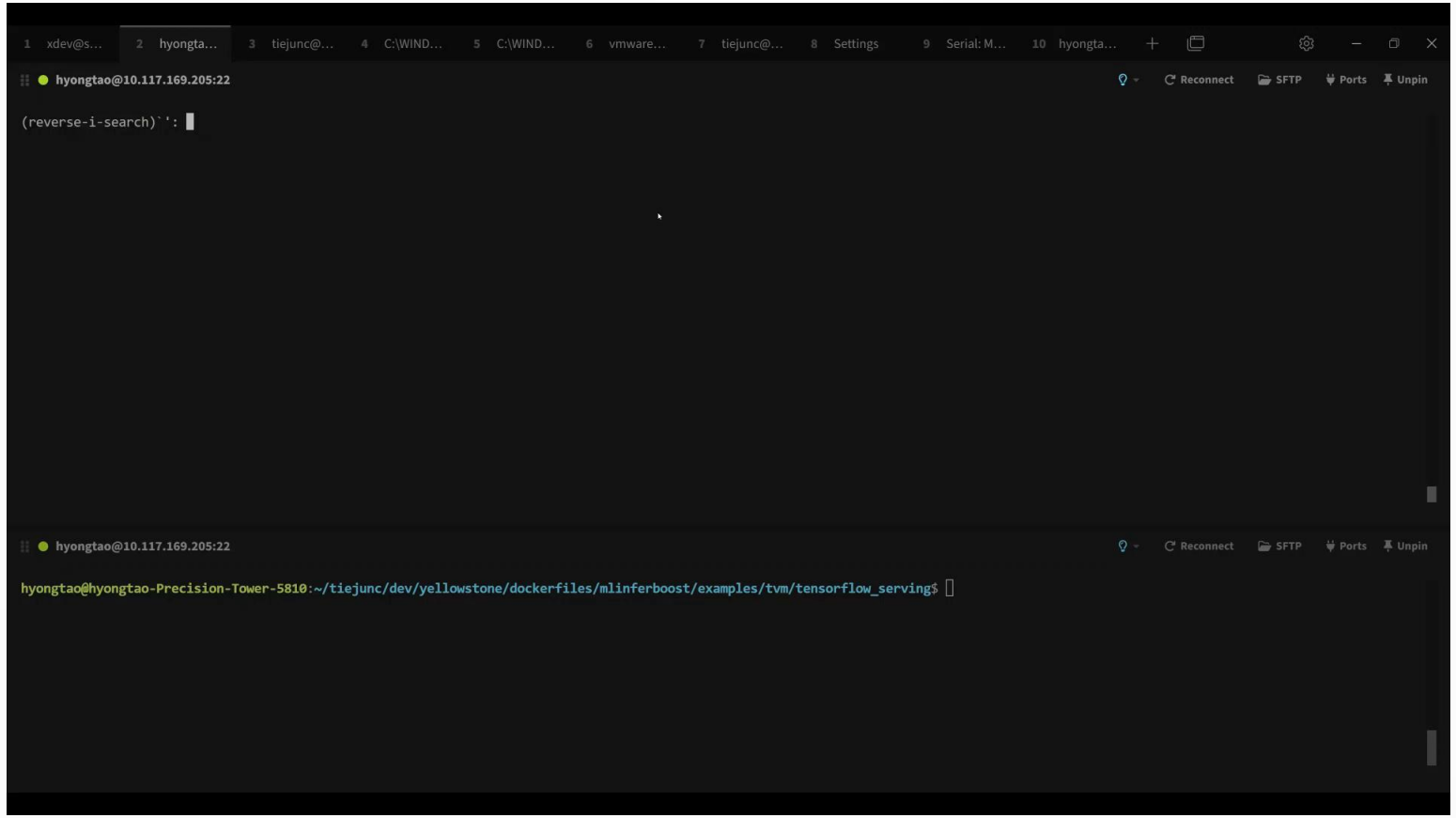
Project Yellowstone VII

Demos – TorchServe



Project Yellowstone VIII

Demos – Tensorflow Serving



Summary I

Now

- ML frameworks
  - Tensorflow
  - Pytorch
  - ONNX
  - Tensorflow Serving, TorchServe, KServe, etc
- Backend acceleration technologies
  - Apache TVM
  - Intel OpenVINO
  - Nvidia TensorRT
  - Xilinx vitis AI
- AI HW accelerators
  - Nvidia GPU
  - AMD GPU
  - Intel GPU
  - Xilinx FPGA
  - CPU

Summary II

Next

- From ML Inference to ML training
- Towards multi—cloud

Thank you!

Tiejun Chen <tiejunc@vmware.com>

GOTC

THANKS

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE