End to End Optimization Stack for Deep Learning

Presenter: Tianqi Chen

Paul G. Allen School of Computer Science & Engineering
University of Washington
Collaborators

University of Washington

Tianqi Chen
ML, Software Stack

Thierry Moreau
Hardware Stack

Haichen Shen
GPU

AWS AI Team

Ziheng Jiang
ARM, NNVM pipeline

and many more contributors in the DMLC community

Carlos Guestrin
Luis Ceze
Arvind Krishnamurthy
Deep Learning System Research is Exciting but Hard

Frameworks
- TensorFlow
- PyTorch
- CNTK
- Caffe2

Computational graph

Operator Libraries
- cuDNN
- NNPack
- MKL-DNN

Hardware
Deep Learning System Research is Exciting but Hard

- **Frameworks**: Caffe², CNTK
- **Operator Libraries**: cuDNN, NNPack, MKL-DNN
- **Hardware**
- **Computational graph**
Deep Learning System Research is Exciting but Hard

- **Frameworks**: 
  - TensorFlow, Caffe, CNTK

- **Computational graph**

- **Operator Libraries**: 
  - cuDNN, NNPack, MKL-DNN

- **Hardware**
Deep Learning System Research is Exciting but Hard

- **Frameworks**: CNTK, Caffe2
- **Computational graph**: \[ A \times B + 1 \]
- **Operator Libraries**: cuDNN, NNPack, MKL-DNN
- **Hardware**: Built a new accelerator
Deep Learning System Research is Exciting but Hard

- **Frameworks**: CNTK, Caffe2
- **Computational graph**: A × B + 1
- **Operator Libraries**: cuDNN, NNPack, MKL-DNN
- **Hardware**: 

Need entire software stack on top of it!

- Layout transformation
- Quantization
- Operator kernel optimization
- Benchmarking

Built a new accelerator
Deep Learning System Research is Exciting but Hard

Frameworks
- CNTK
- Caffe2

Computational graph
- Inputs A and B
- Operations: multiplication (×) and addition (+)

Operator Libraries
- cuDNN, NNPack, MKL-DNN

Hardware
Deep Learning System Research is Exciting but Hard

Frameworks
- TensorFlow
- PyTorch
- CNTK
- Caffe2

Computational graph

Operator Libraries
- cuDNN
- NNPack
- MKL-DNN

Hardware
Deep Learning System Research is Exciting but Hard

Frameworks
- Tensorflow
- PyTorch
- CNTK

Computational graph

Operator Libraries
- cuDNN
- NNPack
- MKL-DNN

Hardware

Data Layout Optimization
Deep Learning System Research is Exciting but Hard

Frameworks
- Tensorflow
- Caffe2
- CNTK

Computational graph

Operator Libraries
- cuDNN
- NNPack
- MKL-DNN

Hardware

Data Layout Optimization
Operator Fusion
Deep Learning System Research is Exciting but Hard

Frameworks
- CNTK
- Caffe 2

Computational graph
- A × B + 1

Operator Libraries
- cuDNN, NNPack, MKL-DNN

Hardware

Data Layout Optimization
Operator Fusion

Need optimized hardware kernel for each variant, on each hardware!
Deep Learning System Research is Exciting but Hard

- **Frameworks**
  - CNTK
  - Caffe2

- **Computational graph**

- **Operator Libraries**
  - cuDNN, NNPack, MKL-DNN

- **Hardware**

**Data Layout Optimization**
**Operator Fusion**
**Serving**

Need optimized hardware kernel for each variant, on each hardware!
The End to End System Challenge

Frameworks

Hardware
Back-Ends
The End to End System Challenge

Frameworks

Hardware
Back-Ends
The End to End System Challenge

Frameworks

Hardware
Back-Ends
The End to End System Challenge

Hardware
Back-Ends
The End to End System Challenge

Hardware

Back-Ends
The End to End System Challenge

Frameworks

Hardware
Back-Ends
The End to End System Challenge

Frameworks

Hardware

Back-Ends
The End to End System Challenge

Frameworks

Intermediate representation

Hardware

Back-Ends
Computational Graph IR and Remaining Gap

Examples: NGraph, XLA, NNVM, DLVM …
Computational Graph IR and Remaining Gap

- Computational Graph
- Auto Differentiation
- Memory Plan
- Operator Fusion

Backends
Computational Graph IR and Remaining Gap

too many possible choices:
precision, layout, fused pattern, device, threading … 
Need a low level IR to express them explicitly
TVM: Low Level IR

- Concise and compact description
- Explicit control on codegen
- Ease of deployment
- Support new hardware backends
Tensor Index Expression Declaration

Compute \( C = \text{dot}(A, B.T) \)

\[
\begin{align*}
\text{import } & \text{ tvm} \\
m, n, h = & \text{ tvm.var('m'), tvm.var('n'), tvm.var('h')} \\
A = & \text{ tvm.placeholder((m, h), name='A')} \\
B = & \text{ tvm.placeholder((n, h), name='B')} \\
k = & \text{ tvm.reduce_axis((0, h), name='k')} \\
C = & \text{ tvm.compute((m, n), lambda i, j: tvm.sum(A[i, k] * B[j, k], axis=k))}
\end{align*}
\]
Challenge: Hardware Diversities
Challenge: Hardware Diversities
Challenge: Hardware Diversities

Memory subsystem:
- IR
- CPU: L3, L2, L1D, L1I
- GPU: L2, SM, TX/L1, RF
- Accelerators: Unified Buffer, FIFO, Acc

CPU: implicitly managed
GPU: mixed
Accelerators: explicitly managed
Challenge: Hardware Diversities

- IR
- CPU: L3, L2, L1D, L1I
- GPU: L2, SM, TX/L1, RF, RF
- Accelerators: Unified Buffer, FIFO, Acc
- Memory subsystem: implicitly managed, mixed, explicitly managed
- Compute primitives: scalar, vector, tensor
Challenge: Hardware Diversities

- IR
- CPU Memory subsystem:
  - L3
  - L2
  - L1D, L1I
  - implicitly managed
- GPU Memory subsystem:
  - L2
  - SM
  - TX/L1
  - RF
  - mixed
- Accelerators Memory subsystem:
  - Unified Buffer
  - FIFO
  - Acc
  - explicitly managed

- Compute primitives:
  - scalar
  - vector
  - tensor

- Data type:
  - fp32
  - fp16
  - int8
Unified Schedule Optimizations for Hardwares

- Algorithm described in IR
- Lowering
- Generated code (LLVM, CUDA, OpenCL...)

(Raw Text)
Unified Schedule Optimizations for Hardwares

Algorithm described in IR

Scheduling Optimization

Lowering

Generated code (LLVM, CUDA, OpenCL…)
Unified Schedule Optimizations for Hardwares

**Scheduling Optimizations**

(✔) Data layout

- Algorithm described in IR
- Scheduling Optimization
- Lowering
- Generated code (LLVM, CUDA, OpenCL…)

- Generated code (LLVM, CUDA, OpenCL…)

- Lowering

- Algorithm described in IR

- Scheduling Optimization

- (✔) Data layout

- Scheduling Optimizations
Unified Schedule Optimizations for Hardwares

Scheduling Optimizations

(✔) Data layout
(✔) Tiling

Algorithm described in IR
Scheduling Optimization
Lowering
Generated code (LLVM, CUDA, OpenCL…)

---

- Unified Schedule Optimizations
- Algorithm described in IR
- Scheduling Optimization
- Lowering
- Generated code (LLVM, CUDA, OpenCL…)

---

Scheduling Optimizations

(✔) Data layout
(✔) Tiling
Unified Schedule Optimizations for Hardwares

**Scheduling Optimizations**

- (✔) Data layout
- (✔) Tiling
- (✔) Thread cooperation

Algorithm described in IR

Scheduling Optimization

Lowering

Generated code (LLVM, CUDA, OpenCL...)
Unified Schedule Optimizations for Hardwares

**Scheduling Optimizations**

- ✔ Data layout
- ✔ Tiling
- ✔ Thread cooperation
- ✔ Latency hiding

Algorithm described in IR

Scheduling Optimization

Lowering

Generated code (LLVM, CUDA, OpenCL...)
Unified Schedule Optimizations for Hardwares

**Scheduling Optimizations**

- ✔ Data layout
- ✔ Tiling
- ✔ Thread cooperation
- ✔ Latency hiding
- ✔ Tensorization

Algorithm described in IR → Lowering → Generated code (LLVM, CUDA, OpenCL…)

Scheduling Optimization
Separation of Compilation and Deployment

Compilation Stack

- Framework Frontends
- NNVM
- TVM
- TVM Graph Module

Heavy optimizations

TVM Runtimes

- Deploy
- Lightweight, 300 to 600 KB
- JS
- Java
- Python
- C++
- Android
- iOS
- Raspberry Pi
- Chrome
Remote Execution and Profiling

Server with TVM Compiler

TVM RPC

Devices with TVM Runtime
Performance Portable against state of art

K80, Baseline
MXNet with cuDNN auto tune enabled
One grad student month

Raspberry Pi 3
Baseline: MXNet with OpenBLAS and NNPack
Two undergrad weeks

Credit: Leyuan Wang(AWS/UCDavis), Yuwei Hu(TuSimple), Zheng Jiang(AWS/FDU)
Coming Soon: Target New Accelerators

Tensorization
Latency Hiding
FPGA Example for building new hardware backend
Open-source soon
NNVM Compiler: Open Compiler for AI Systems

Graph Optimizations

TVM Primitives

External Support
Supported
Work in progress

Joint Work with AWS AI Team and DMLC community
Deep Learning System Research is Exciting but Hard

Frameworks: TensorFlow, Caffe2, CNTK

Computational graph

Operator Libraries: cuDNN, NNPack, MKL-DNN

Hardware
Deep Learning System Research is Just Exciting

I can program my new accelerators from python :
My new optimizations works on all platforms !

Hardware

Frameworks

Graph Optimizations

TVM Primitives

Graph Optimizations

TVM

NNVM
Deep Learning System Research is Just Exciting

You can be part of it!

My new optimizations works on all platforms!

I can program my new accelerators from python :)

You can be part of it!