Speculations about Computer Architecture in Next Three Years

shuchang.zhou@gmail.com
Jan. 20, 2018
About me

https://zsc.github.io/

- Source-to-source transformation
- Cache simulation
- Natural Language Question & Answer
- Indoor Navigation with INS
- Group Orbit Optimization
- OCR
- Quantized Neural Network
- Smart Camera
- Reinforcement Learning

Compiler Optimization | Machine Learning | Neural Network

Deep Learning Revolution in Vision & Speech

End-2-end Neural Network
Deep Learning Revolution in Vision & Speech

Differentiable Forward & Backward
Implications of Deep Learning

● Unification of Algorithms in Vision & Speech
  ○ Deep Learning v.s. “Traditional methods”

● Graph execution engine as the new platform
  ○ For CNN / RNN

● A new wave of data centers
  ○ Google / Facebook: millions of GPU
  ○ Startups: thousands of GPU

● Adjoint of Neural Networks
  ○ Image augmentor
  ○ Simulators
Computation Stack

Silicon
- Partitioning & Planning
- Place & Route
- Timing Closure

Verilog
- Karnaugh map
- Finite State Machine

Architecture
- ISA
- Micro-code
- Resource allocation

Operating System
- Page table
- File system
- Interrupts

Compiler
- Parallelism mining
- Memory latency hiding

Computation Graph Engine
- Kernels
- Execution Plan
Computation Stack

Silicon
- Partitioning & Planning
- Place & Route
- Timing Closure

Verilog
- Karnaugh map
- Finite State Machine

Architecture
- ISA
- Micro-code
- Resource allocation

Operating System
- Page table
- File system
- Interrupts

Compiler
- Parallelism mining
- Memory latency hiding

Computation Graph Engine
- Kernels
- Execution Plan

How will this stack deal with changes?
Case study: Large Neural Networks

Characteristics: many channels + side-branches + many layers

AlexNet

GoogLeNet

ResNet
Case study: Large Neural Networks

- Instructions for convolutions & non-linearity
- Systolic Array
- Large page-table
- Auto-SIMD

On-Chip-Memory for caching feature maps

Static analysis + dynamic profiling for kernel selection + execution plan
Case study: Small Neural Networks

Characteristics: few channels + 1x1 convolutions

MobileNet

Lack of shortcut hurts its transfer learning ability.

ShuffleNet

The unique shuffle operation slows its adoption.
Case study: Small Neural Networks

- Specialized support for few channel layers and 1x1 convolutions.
- Different batching

On-Chip-Memory may be more important.

Fusion of layers + handcrafted kernels

Lower overhead

Auto-SIMD
When a Neural Network Designers, a Computer Architect, a Compiler Expert and an OS Guru meet

- **Designer wants**
  - A reliable performance model
    - Open architecture design and assembly/microcode level exposure
  - Better profilers for runtime diagnostics and analyzers
  - Support for sparse matrices, dynamic operations

- **Architect wants**
  - Batch operations with constant delays
  - Regular memory access pattern subject to locality and many reuses
  - Streamlined memory/computation usage, no overwhelming peaks
  - Less number of operators

- **Compiler Expert and OS Guru wants**
  - To broker between the Designer and the Architect
    - Have a slow fallback for bizarre operators
    - Cutting peaks
Case study: Quantum Computing Simulator on FPGA

Clash/FPGA: implement Complex Number

type CC = Vec 2 RR

c0 = 0 ⇒ 0 ⇒ Nil
c1 = 1 ⇒ 0 ⇒ Nil

sqr_norm :: CC → RR
sqr_norm (a ⇒ b ⇒ Nil) = a * a + b * b

cadd :: CC → CC → CC
cadd = zipWith (+)

cmul :: CC → CC → CC
cmul (a ⇒ b ⇒ Nil) (c ⇒ d ⇒ Nil) = (a * c - b * d) ⇒ (a * d + b * c) ⇒ Nil

dotProduct xs ys = foldr cadd c0 (zipWith cmul xs ys)
matrixVector m v = map (\dotProduct v) m
Case study: Quantum Computing Simulator on FPGA

HLS may be sufficiently efficient and flexible

Deutsch-Jozsa’s algorithm

\[ \begin{array}{c}
|0\rangle \\
|1\rangle
\end{array} \quad \xrightarrow{\mathcal{H} \otimes \mathcal{H}} \quad \xrightarrow{\mathcal{U}_f \otimes \mathcal{H}} \quad \xrightarrow{\mathcal{H} \otimes \mathcal{I}} \]

**Deutsch u** :: \( \text{Vec} \ 2 \ \text{RR} \to \text{Vec} \ 4 \ \text{CC} \to \text{Vec} \ 4 \ \text{CC} \)

\[ \text{deutsch u} \ (f_0 : > f_1 : > \text{Nil}) = \]

\[
\text{matrixVector} \ (\text{make complex} \ (\begin{array}{c}
(1 - f_0) : > f_1 : > 0 : > 0 : > \text{Nil}) : > \\
(f_0 : > (1 - f_1) : > 0 : > 0 : > \text{Nil}) : > \\
(0 : > 0 : > (1 - f_0) : > f_1 : > \text{Nil}) : > \\
(0 : > 0 : > f_0 : > (1 - f_1) : > \text{Nil}) : > \text{Nil})
\]

**Hadamard I** :: \( \text{Vec} \ 4 \ \text{CC} \to \text{Vec} \ 4 \ \text{CC} \)

\[ \text{hadamard I} = \]

\[
\text{matrixVector} \ (\text{make complex} \ (\begin{array}{c}
h : > 0 : > h : > 0 : > \text{Nil}) : > \\
(0 : > h : > 0 : > h : > \text{Nil}) : > \\
h : > 0 : > -h : > 0 : > \text{Nil}) : > \\
(0 : > h : > 0 : > -h : > \text{Nil}) : > \text{Nil})
\]

\[ H \otimes I = \begin{pmatrix}
    h & 0 & h & 0 \\
    0 & h & 0 & h \\
    h & 0 & -h & 0 \\
    0 & h & 0 & -h
\end{pmatrix} \]

**where** \( h = \frac{1}{\sqrt{2}} \)
A possible future

Design Silicon Compiler!

How Google and Qualcomm exploit real world HLS and HLV

By Paul Dempsey | No Comments | Posted: June 1, 2016
Topics/Categories: IP - Assembly & Integration, Design Management, EDA - ESL, IC Implementation, Verification | Tags: high level verification, high-level synthesis (HLS), hls, hlv | Organizations: Google, Mentor Graphics, Qualcomm

By taking a pragmatic approach, the two technology giants have comfortably adopted high-level synthesis and verification – and have shared their experiences.
Case study: Reinforcement Learning

Characteristics: require fast & complex simulations

OpenSim

A human skeleton model for locomotive task modeling.

GTA 5
AirSim

Simulation for self-driving car/ADAS and Drones.
Case study: Reinforcement Learning

Typical CPU load, but need to integrate with Neural Network Accelerator
A possible future

Revival of Compiler Optimizations!

Should we prepare a benchmark of simulators?
The Age of Instant Response

- **Old School**
  - Compiler cannot change code
  - Developer as the dictator
  - Batch operation and buffering
  - Conference & Journal

- **New School**
  - Compiler can offer suggestions
  - User Community
    - User code contributions
    - Peer-to-peer helping
  - Low latency is critical
The combined future ...

Performance critical

TPU / FPGA

Complex coordination

TensorFlow
Backup after this slide