Software Defined Hardware

For data intensive computation

Wade Shen
DARPA I2O
September 19, 2017
Goal Statement

Build runtime reconfigurable hardware and software that enables near ASIC performance (within 10x) without sacrificing programmability for data-intensive algorithms.
Problem

- Processor design trades
  - Math/logic resources
  - Memory (cache vs. register vs. shared)
  - Address computation
  - Data access and flow

The problem: Optimal hardware configuration differs across algorithms

No one hardware efficiently solves all problems well
**SDH: Runtime optimization of software and hardware**

*For data intensive computation*

---

**Today:** HW design specialization

- One chip per algorithm
  - Chip design expensive
- Not reprogrammable
- Can’t take advantage of data-dependent optimizations

**Tomorrow:** Runtime optimization of hardware and software

- One chip many applications
  - One time design cost
- Reprogrammable via high-level languages
- Data-dependent optimization (10-100x)

---

Google’s TPU

**Specialized**

Energy Efficiency [MOP/mW]
**Software-defined hardware**

**Properties:**
1. Reconfiguration times: 300 - 1,000 ns
2. Re-allocatable compute resources - i.e. ALUs for address computation or math
3. Re-allocatable memory resources - i.e. cache/register configuration to match data
4. Malleable external memory access - i.e. reconfigurable memory controller

**Dynamic HW/ SW compilers for high-level languages (TA2)**
1. Generate optimal configuration based on static analysis code
2. Generates optimal code
3. Re-optimize machine code and processor configuration based on runtime data

**Reconfigurable processors (TA1)**

- **Code₁**
- **Code₂**
- **Code₃**
- **Code₄**
- **Code₅**

**Config₁**

**Config₂**

**Config₃**

**Config₄**

**Config₅**

**Time**
TA1: Reconfigurable processors

Graphicionado: graph search engine

- Async Memory Controller
- ALU
- ALU
- ALU
- M
- M
- Scratchpad for active search nodes
- Address calculators for sparse vector lookup

Performance: 157M edges/s/W search (BFS)

Eyeriss: Image neural net engine

- Block read Controller
- M
- M
- ALU
- ALU
- ALU
- M
- M
- Image convolution operators
- Image cache

Performance: 250 images/s/W (AlexNet)

SDH

- Reconfigurable Interconnect
- Programmable Memory Controller
- ALU
- M
- ALU
- M
- ALU
- M
- ALU
- M
- ALU

Plasticine: Stanford Seedling
- Graph search: 102M edges/s/W
- Image recognition: 130 images/s/W

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.
Compilers generate optimal code via static analysis + tracing methods
- Assume static processor configuration, compile code, run, trace, recompile

SDH compilers don’t assume a static processor configuration
- Generates optimal configuration/code given program + data
- Problem: Resources and architecture optimization space is large

Solution:
1. Configure initial processor configuration, compile code, run and trace, then
2. Predict best configuration via reinforcement learning/stochastic optimization
How will TA2 work?

Empirical kernel mining from D3M ML corpus

DARPA D3M Corpus

IR (intermediate representation) component optimization

Pre-compilation to IR (data-independent)

Optimization (data-dependent)

High-level data programs

\[ f(x) = \text{warp}(\text{fft}(\text{window}(x))) \]

\[ f(A, x) = Ax + b \]

Low-level implementations not exposed

Compile time

Run time

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.
Program evaluation and goals

- USG team will create a benchmark suite of machine learning, optimization, graph and numeric applications
  - 500+ programs from D3M program
  - Implementations for GPU and CPU
  - Subset of 100 optimized for ASIC (FPGA proxy)

- Metrics:
  - Speedup/power relative to ASIC and general purpose processors
  - Programmability: time to code solution for SDH languages vs. NumPy/Python

- Target outcomes:

<table>
<thead>
<tr>
<th></th>
<th>vs. CPU</th>
<th>vs. ASIC</th>
<th>vs. ASIC (sparse math, graphs)</th>
<th>Programmability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phase 1</strong></td>
<td>100-300x</td>
<td>within 10x</td>
<td>2x</td>
<td>within 3x</td>
</tr>
<tr>
<td><strong>Phase 2</strong></td>
<td>500-1000x</td>
<td>within 5x</td>
<td>8-10x</td>
<td>~1x</td>
</tr>
</tbody>
</table>

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.
Backup
TA1: Example architecture (U. Michigan Seedling)

- Dramatic performance and energy opportunity by tailoring architectures to applications: e.g. sparse and dense matrix multiplication and graph algorithms
- Cache hierarchy, memory bandwidth, SIMD vs. MIMD, dedicated cores

Sparse Computation

1. Outer product generation
2. Merge outer products

Dense Computation

1. Inner product on individual tiles
2. Merge tiles

- vs. CPU: 20 – 100x performance gain – data reuse, reduced memory bandwidth
- vs. GPU: 10 – 50x performance gain – data movement & placement, async execution
- vs. ASIC: within 2 – 3x of performance but full flexibility / programmability
**ASiCs**

**Graphicionado:** ASiC for graph search
- Element-based memory access
- Specialized indirect address calculator for sparse vectors
- Specialized on-chip scratch pad
- **157K edges/s/mW search (BFS)**

**Eyeriss:** Specialized DNN accelerator
- 168 specialized convolution units
- Specialized implementation of neural non-linearity (ReLU)
- Large block memory access only
- **250 images/s/W on AlexNet**

---

**SDH Opportunities (vs. CPU)**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Dense Advantage</th>
<th>Sparse Advantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 memory control units</td>
<td>30x Perf, 52x Perf/W</td>
<td>8.2x Perf, 55.2x Perf/W</td>
</tr>
<tr>
<td>64 data pattern control units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configurable off-chip access for dense and sparse (scatter/gather)</td>
<td>1x</td>
<td>18x</td>
</tr>
<tr>
<td>Configurable on-chip memory for high BW &amp; coarse-grain pipes</td>
<td>2.1x</td>
<td>18.4x</td>
</tr>
<tr>
<td>Compute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pipelined SIMD</td>
<td>12.6x</td>
<td>NA</td>
</tr>
<tr>
<td>Variable precision</td>
<td>25.2x</td>
<td></td>
</tr>
<tr>
<td>Shift network</td>
<td>7x</td>
<td></td>
</tr>
</tbody>
</table>

**Key Challenges:**
- Data flow: configurable memory control units, data patterns, data storage
- Compute flexibility: compute granularity, modular functionality, high BW malleable interconnect

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.