Programmable Smart Machines: A Hybrid Neuromorphic approach to General Purpose Computation

Speaker: Jonathan Appavoo
Amos Waterland, Katherine Zhao, Schuyler Eldridge, Ajay Joshi, Steve Homer, Margo Seltzer

PSM: Hybrid computing systems that behave as programmed but transparently learn and automatically improve their operation
A Unification Principle of Information Processing Systems “Programmable Smart Machines”

DSRC/DARPA 2008 Summer Conference

Jonathan Appavoo & Amos Waterland
IBM Research – T.J. Watson Research Center NY
It seems possible to recognize structure independent of syntax.

A form of information processing exists that extracts Statistical Structure which is Multi-Scale, Multi-modal, and correlated in both time and space.

Our Goal:
Explore exploitation of structure in execution (in an ISA independent way), which is correlated over time in a consistent, way in order accelerate the programmed function and potentially enable new un-programmed function.
1. Convert ENTIRE operation of a Modern Computer into an Execution Signal.
2. Extract, Represent and Express Structure in Signal.
3. Exploit Structure to potentially:
   1. Accelerate Future Computation,
   2. Expose Unknown/Un-Expected Correlations and
   3. Enable Extrapolation to generate new Behaviour.
ASC: Automatically Scaling Computation
(A. Waterland et. al, ASPLOS 2014)

Exploit Learning of Structure to enable a form of Auto-Parallelization
Execution Signal

See our *HotPAR'12* Paper, "Parallelization by Simulated Tunneling", Waterland et al. For a more technical view of execution as state space traversal (Dynamical Systems Interpretation).
Computer As a State Vector

\[
x_t = (00000000000000000000000000000000, 00000000000000000000000000000000, \ldots, 00000000000000000000000000000000, \ldots, 00000000, \ldots, 00000000)
\]

- eax
- ecx
- gs
- ram

0 1

All Bits Zero

All Bits One

Random 1 or 0 (flip a coin for each bit \(\Pr(b=1) = 0.5\), \(\Pr(b=0) = 0.5\))

The Task

initial state

past/history

current

future?
LASC

- Classic Sequential Execution process
- Sequential Execution State
- Common Introspection backplane
- Pluggable suite of parallel Recognizers
- SE State
  - Updates of sequential execution state from state cache
- State Cache
  - Cache of (start,end) State Pairs
- Parallel stream of predicted start,end pairs
- Parallel publication of recognized features
- Scalable Ensemble of Predictors
  - Constant parallel stream of predicted start states
- Farm of Execution Processes

Graphs:
- LASC scaling for Ising on 32-core server
- LASC scaling for Ising on Blue Gene/P
- LASC scaling for Polybench on 32-core server

Graph elements:
- IDEAL scaling
- Hand-parallelized scaling
- LASC cycle count scaling
- LASC+oracle scaling
- LASC scaling
A different role for Neuromorphic devices in a general purpose Computer Model

A Computation Cortex — A massive, power efficient, autoassociative memory of execution patterns
A micro-scale Experiment

1. Filter
2. Train
3. Predict

\[
\text{int inner(bool } u[], \text{bool } v[], \text{int } d) \{ \\
\quad \text{int } i, E = 0; \\
\quad \text{for } (i = 0; i < d; i++) \{ \\
\qquad E += u[i] \times v[i]; \\
\quad \} \\
\quad \text{return } E; \\
\}\]

Probability of "1" is Independent and Identically Distributed
Filter

inner 0.1 Execution

265 Changing Bits out of 1345216 State Bits

Byte 128

4001ca 4831ed  xor rbp, rbp
4001cd 5f    pop rdi
4001ce 4889e6  mov rsi, rsp
4001d1 4803e4f0 and rsp, 0xfffffffffffffff0
4001d5 e8c6ffffff call 0x4001a0
4001da 55    push rbp
4001a1 4889e5  mov rbp, rsp
4001a4 4883ec10 sub rsp, 0x10
4001a8 897fc  mov [rbp-0x4], edi
4001ab 480975f0 mov [rbp-0x10], rsi
4001af ba000000 mov edx, 0x2000
4001b4 be000000 mov esi, 0x603000
4001b9 bf000000 mov edi, 0x601000
4001be e881ffffff call 0x400144

Instructions Executed

Time

Probability of 1 as vector element

Number of instructions
.0000000000400144 <inner>:
  400144: 55                    push   %rbp
  400145: 48 89 e5              mov    %rsp,%rbp
  400148: 48 89 7d e8           mov    %rdi,-0x18(%rbp)
  40014c: 48 89 75 e0           mov    %rsi,-0x20(%rbp)
  400150: 89 55 dc              mov    %edx,-0x24(%rbp)
  400153: c7 45 f8 00 00 00 00  movl   $0x0,-0x8(%rbp)
  40015a: c7 45 fc 00 00 00 00  movl   $0x0,-0x4(%rbp)
  400161: eb 30                 jmp    400193 <inner+0x4f>
  400163: 8b 45 fc              mov    -0x4(%rbp),%eax
  400166: 48 63 d0              movslq %eax,%rdx
  400169: 48 8b 45 e8           mov    -0x18(%rbp),%rax
  40016d: 48 01 d0              add    %rdx,%rax
  400170: 0f b6 00              movzbl (%rax),%eax
  400173: 48 63 c8              movslq %eax,%rcx
  400176: 48 8b 45 e0           mov    -0x24(%rbp),%eax
  400179: 48 63 c8              movslq %eax,%rcx
  40017c: 48 8b 45 e0           mov    -0x20(%rbp),%rax
  40017f: 48 01 c8              add    %rcx,%rax
  400183: 0f b6 00              movzbl (%rax),%eax
  400186: 0f b6 c0              movzbl %al,%eax
  400189: 0f af c2              imul   %edx,%eax
  40018c: 01 45 f8              add    %eax,-0x8(%rbp)
  40018f: 83 45 fc 01           addl   $0x1,-0x4(%rbp)
  400193: 8b 45 fc              mov    -0x4(%rbp),%eax
  400196: 3b 45 dc              cmp    -0x24(%rbp),%eax
  400199: 7c c8                 jl     400163 <inner+0x1f>
  40019b: 8b 45 f8              mov    -0x8(%rbp),%eax
  40019e: 5d                    pop    %rbp
  40019f: c3                    retq
Train — Online Update

\[
\text{MASK} = \text{XOR} (\text{LAST}, \text{CUR}) \mid \text{MASK}
\]
\[
\text{BITS} = \text{CONDENSE} (\text{CUR} \& \text{MASK})
\]
\[
\text{FANN\_TRAIN} (\text{NET}, \text{OLDBITS}, \text{BITS}, \ldots)
\]
\[
\text{OLDBITS} = \text{BITS}
\]
Predict

```
400163:  mov    -0x4(%rbp),%eax
400166:  movslq %eax,%rdx
400169:  mov    -0x18(%rbp),%rax
40016d:  add    %rdx,%rax
400170:  movzbl (%rax),%eax
400173:  movzbl %al,%edx
400176:  mov    -0x4(%rbp),%eax
400179:  movslq %eax,%rcx
40017c:  mov    -0x20(%rbp),%rax
400180:  add    %rcx,%rax
400183:  movzbl (%rax),%eax
400186:  movzbl %al,%eax
400189:  imul   %edx,%eax
40018c:  add    %eax,-0x8(%rbp)
40018f:  addl   $0x1,-0x4(%rbp)
400193:  mov    -0x4(%rbp),%eax
400196:  cmp    -0x24(%rbp),%eax
400199:  jl     400163 <inner+0x1f>
```

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>40018f</td>
<td>0.0,0.1</td>
<td></td>
</tr>
<tr>
<td>40016d</td>
<td>0.2-0.7</td>
<td></td>
</tr>
<tr>
<td>400180</td>
<td>0.8,0.9</td>
<td></td>
</tr>
<tr>
<td>400193</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>
400163: mov  -0x4(%rbp),%eax
400166: movslq %eax,%rdx
400169: mov  -0x18(%rbp),%rax
40016d: add   %rdx,%rax
400170: movzbl (%rax),%eax
400173: movzbl %al,%edx
400176: mov  -0x4(%rbp),%eax
400179: movslq %eax,%rcx
40017c: mov  -0x20(%rbp),%rax
400180: add   %rcx,%rax
400183: movzbl (%rax),%eax
400186: movzbl %al,%eax
400189: imul  %edx,%eax
40018c: add   %eax,-0x8(%rbp)
40018f: addl  $0x1,-0x4(%rbp)
400193: mov  -0x4(%rbp),%eax
400196: cmp   -0x24(%rbp),%eax
400199: jl    400163 <inner+0x1f>

| 40018f  | 0.0,0.1 |
| 40016d  | 0.2-0.7 |
| 400180  | 0.8,0.9 |
| 400193  | 1.0     |
- Number of instructions
- Probability of 1 as vector element
- Expected value of boolean inner product
- Variance of boolean inner product
- Final MISS rate on best trial

Best Network for each Probability:
- 18f:4
- 18f:20
- 18f:89
- 16d:4
- 16d:12
- 16d:7
- 16d:44
- 16d:25
- 16d:40
- 180:44
- 180:22
- 193:74

Colors:
- RBP (purple)
- RAX (orange)
- RCX (yellow)
- RDX (red)
- RSI (blue)
- RDI (brown)
- RIP (green)
- EFLAGS (pink)
- RSP (skyblue)
- Stack (salmon)
Bridge to Some (or No) Where?

Will the promise of Neuromorphic technologies be on the other side?
“Spare a little Silicon?”