REACT: A Framework for Rapid Exploration of Approximate Computing Techniques

Mark Wyse, André Baixo, Thierry Moreau, Bill Zorn
James Bornholt, Adrian Sampson, Luis Ceze, Mark Oskin

University of Washington
Motivation

Understand current research

Investigate new techniques

Evaluate impact of existing techniques
Taxonomy

Determinism

\[ |P(x) - A(x)| \leq \varepsilon \forall x \]
\[ \Pr(|P(x) - A(x)| > \varepsilon) < P \forall x \]

Granularity

Hardware/Software

Computational Resource(s)
<table>
<thead>
<tr>
<th></th>
<th>Nondeterministic</th>
<th>Deterministic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fine Grained</strong></td>
<td>DRAM Refresh Rate</td>
<td>Bit-Width Reduction</td>
</tr>
<tr>
<td></td>
<td>SRAM Soft Error Exposure</td>
<td>Float-to-Fixed Conversion</td>
</tr>
<tr>
<td></td>
<td>Approximate Storage (PCM)</td>
<td>Fuzzy Memoization</td>
</tr>
<tr>
<td></td>
<td>Soft Fault Tolerance</td>
<td>Hierarchical FPU</td>
</tr>
<tr>
<td></td>
<td>Synchronization Elision</td>
<td>Load Value Approximation</td>
</tr>
<tr>
<td></td>
<td>Voltage Overscaling</td>
<td>Lossy Compression and Data Packing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Precision Scaling ALU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reduced-Precision FPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Underdesigned Multiplier</td>
</tr>
<tr>
<td><strong>Coarse Grained</strong></td>
<td>Error Rate</td>
<td>Algorithm Selection</td>
</tr>
<tr>
<td></td>
<td>Neural Acceleration (Analog)</td>
<td>Code Perforation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interpolated Memoization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Neural Acceleration (ASIC, FPGA, GPU)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parallel Pattern Replacement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter Adjustment</td>
</tr>
</tbody>
</table>
REACT

A Framework for Rapid Exploration of Approximate Computing Techniques
Application Profiler & Energy Model

Intel Pin tool
Insn Count + Arch Events

Custom, linear model
Simple, understandable
Validated against McPAT
Error Injection

ACCEPT
Runtime error injection
Simple API
Arbitrary error models

```
int i, p;
APPROX int a;
APPROX int data[N];
a = data[i] * p;
```
Approximation Models

- Load Value Approximation
- Drowsy SRAM
- Neural Acceleration
- Reduced Precision FPU
- Low refresh rate DRAM
- Voltage Overscaled ALU
Early Results - Sobel
Early Results – FFT1D

![Graph showing energy savings vs. signal-to-noise ratio for different techniques. The graph includes markers for DRAM Refresh, Load Value Approximation, Neural Acceleration, Reduced-Precision FPU, Spatial Accelerator, Voltage Overscaling; DRAM; FPU, and Voltage Overscaling; FPU. The x-axis represents the signal-to-noise ratio, and the y-axis represents energy savings. The graph highlights Precise SNR.]
Conclusions

Coarse-grained superior to fine-grained

Coarse-grained, Nondeterministic!
Thank you!

Questions?