Approximate Software for Accurate Hardware

Jorge Castro-Godínez | September 4, 2019
Outline

- Motivation.
- Proposed project.
- Current progress.
- Results.
- Future work.
Goal of *Approximate* and *Transprecision Computing*: reduce the computational *effort* by exchanging computational *accuracy/quality*.

Many existing computing systems can not afford hardware modifications to embrace proposed non-precise computing techniques.

Remaining exploitable layer: **software**.
But...

- Proposed techniques are isolated (one at the time).
- Results have been driven by *accuracy* rather than *quality*.
- Not available as open-source contributions to be used and to build on top.
Proposed project

- Develop a tool to generate approximate executable code from accurate implementations for a given a quality constraint.
Loop Perforation

- Transform loops to execute only a subset of the original iterations.
- New approaches to dynamically apply this technique.
- Before
  
  ```
  for (i = 1; i < LIMIT; i++)
  ```

- Then
  
  ```
  for (i = 1; i < LIMIT; i=i+2)
  ```
Variable Replacement

- In some functions, variables can be replaced by other variables as their values are very similar.

```c
int foo(int a, int b, int c, int d) {
    int b2 = b * 2;
    int d2 = d * 2;
    int s1 = a + b2;
    int s2 = c + d2;
    ...
}
```

- If $d$ is similar $b$

```c
int foo(int a, int b, int c) {
    int b2 = b * 2;
    int s1 = a + b2;
    int s2 = c + b2;
    ...
}
```
Currently testing on Freedom E310, RISC-V from SiFive (HiFive1 Rev. B)

Using LLVM-based toolchain for code modifications and GCC-based toolchain to generate executable for assembly.
- Image processing kernels.
- Pixel value correlation (spatial correlation)
Results

Considering 3 × 3 kernels

\[
\begin{bmatrix}
a & b & c \\
d & e & f \\
g & h & i \\
\end{bmatrix}
\]
Results

- Gaussian filter (Lena image).
Results

- Gaussian filter (Lena image).
Results

- Sobel filter (Plate image).

![Graph showing Execution Time and PSNR(dB) for different conditions.]

**Exec. Time**
- 1: 0.8
- 2: 0.8
- 3: 0.8
- 4: 0.8
- LP: 0.6
- LP+1: 0.4

**PSNR(dB)**
- 1: 28
- 2: 28
- 3: 26
- 4: 26
- LP: 24
- LP+1: 22

Outline
- Motivation
- Proposed project
- Current progress
- Results
- Future work

Jorge Castro-Godínez – Approximate Software for Accurate Hardware

September 4, 2019
Future work

- Define and implement other individual techniques.
- Design an algorithm to test and determine the best combination of techniques for a given code.
- Perform JIT execution to assess quality degradation.
- If interested, please stay tuned in https://git.scc.kit.edu/CES
Thanks for your attention!