Energy-aware multi-threaded software systems
Measuring and modelling software energy consumption on a multi-threaded embedded processor architecture

Steve Kerrison, steve.kerrison@bristol.ac.uk; Supervisor: Dr. Kerstin Eder, kerstin.eder@bristol.ac.uk

DATE 2013 PhD Forum

Implementing the model: Analysis

Combination of XMOS hardware

Research question
What additional challenges do multi-threaded architectures pose?

Solution
Examine multi-threaded software and architectures vs. existing techniques.

Our contribution

Measurement HW/SW framework: XMProfile
Combination of XMOS hardware\textsuperscript{6}, current sense hardware and a custom software framework.

- 1V power supply
- BKPSF current sense
- INA 219
- Power samples
- Control and Sync
- Vcore supply
- XMProfile control SW
- 219 driver
- Test info, power data
- Test processor
- XProcessor 3600 test processor
- INA 219
- Test run
- Test kernels
- Host PC datastore
- Data collection: Test construction

Data collection: Test construction
Tiwari method\textsuperscript{4}:

\[ E_p = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j}) + \sum_k E_k \]

Considerations for XS1-L\textsuperscript{7}:
- Thread count
- Idle periods (event waiting).
- Instruction overhead is between threads.
- Trace simulation is slow.

Solution:
- Tightly-coupled threads.
- Odd/even threads used to measure instruction overheads.
- Thread count (0 – 8) used to establish thread and idle costs.

XS1-L multi-threaded pipeline

Step | 1 thread | 2 threads | 3 threads | 4 threads | 5+ threads
-----|-----------|-----------|-----------|-----------|----------
1    | T0        | T1        | T0        | T1        | T1       |
2    | -         | T1        | T2        | T1        | T1       |
3    | -         | T1        | T2        | T1        | T1       |
4    | -         | -         | T3        | T3        | T3       |
5    | T0        | T0        | T0        | T0        | T4       |

Additional XMProfile features:
- Constrained random number generation.
- Auto-generate large sets of test loops (ALU).
- Minimise loop head/thead overlapped.

Testing and evaluation

A set of benchmarks – traditional and custom – were used to test model performance with various levels of concurrency.

Benchmarks were run through the XMProfile framework to acquire real device energy measurements.

- Worst case error: 16% standard model, 26% grouped model.
- Average error: 7% standard model, 15% grouped model.
- ISA simulation\textsuperscript{6} runs ~100x slower than real time, statistics processing time is negligible.

Continuing and future work

- Complete ISA model based on established base facts and more complex test kernels to improve accuracy.
- Swallow project: Many-core XS1 system grid (100s of cores).

Incorporate comms costs into model.
- Contributing to ENTRA (EEnergy TRANsparency) EU FP7 project.
- Static analysis of compiled code rather than simulation.
- Use model for design space exploration & guided optimisation (tool assisted & fully automated).

References

8. XS1-LSA-TQ288 protocol
9. Steve Kerrison, steve.kerrison@bristol.ac.uk

Software energy modelling: ISA level\textsuperscript{1,2}, device blocks\textsuperscript{3}, library level\textsuperscript{3}.

Steve Kerrison, steve.kerrison@bristol.ac.uk; Supervisor: Dr. Kerstin Eder, kerstin.eder@bristol.ac.uk

DATE 2013 PhD Forum

Solution

Software energy modelling: ISA level\textsuperscript{1,2}, device blocks\textsuperscript{3}, library level\textsuperscript{3}.

Multi-threaded model using simulation statistics:

\[ E_p = P_{\text{idle}} N_{\text{idle}} T_{\text{idle}} + \sum_{i=1}^{N} \left( P_{\text{inst}} N_{\text{inst}} T_{\text{inst}} \right) + \sum_{i,j} \left( P_{\text{com}} O_{i,j} N_{i,j} T_{i,j} \right) \]

- Instruction overhead smaller than data overhead.
- Use instruction execution statistics, rather than trace, for speed: ~16x faster.
- Implement model per-instruction (standard) and by operand count (grouped).
- Consider concurrency levels (number of active threads).

Implementation of the model: Analysis

- Instruction overhead is smaller than data overhead.
- Use instruction execution statistics, rather than trace, for speed: ~16x faster.
- Implement model per-instruction (standard) and by operand count (grouped).
- Consider concurrency levels (number of active threads).

Multi-threaded model using simulation statistics:

- Instruction overhead smaller than data overhead.
- Use instruction execution statistics, rather than trace, for speed: ~16x faster.
- Implement model per-instruction (standard) and by operand count (grouped).
- Consider concurrency levels (number of active threads).

References

8. XS1-LSA-TQ288 protocol
9. Steve Kerrison, steve.kerrison@bristol.ac.uk

Software energy modelling: ISA level\textsuperscript{1,2}, device blocks\textsuperscript{3}, library level\textsuperscript{3}.

Multi-threaded model using simulation statistics:

\[ E_p = P_{\text{idle}} N_{\text{idle}} T_{\text{idle}} + \sum_{i=1}^{N} \left( P_{\text{inst}} N_{\text{inst}} T_{\text{inst}} \right) + \sum_{i,j} \left( P_{\text{com}} O_{i,j} N_{i,j} T_{i,j} \right) \]

- Instruction overhead smaller than data overhead.
- Use instruction execution statistics, rather than trace, for speed: ~16x faster.
- Implement model per-instruction (standard) and by operand count (grouped).
- Consider concurrency levels (number of active threads).

Multi-threaded model using simulation statistics:

\[ E_p = P_{\text{idle}} N_{\text{idle}} T_{\text{idle}} + \sum_{i=1}^{N} \left( P_{\text{inst}} N_{\text{inst}} T_{\text{inst}} \right) + \sum_{i,j} \left( P_{\text{com}} O_{i,j} N_{i,j} T_{i,j} \right) \]

- Instruction overhead smaller than data overhead.
- Use instruction execution statistics, rather than trace, for speed: ~16x faster.
- Implement model per-instruction (standard) and by operand count (grouped).
- Consider concurrency levels (number of active threads).