Hardware Prediction of OS Run-Length For Fine-Grained Resource Customization

Kshitij Sudan, David Nellans, Rajeev Balasubramonian and Erik Brunvand

UTAH ARCH

THE UNIVERSITY OF UTAH

Background

- Computing moving from emphasizing single thread performance to an energy efficient, throughput oriented chip-multiprocessor (CMP) based model.
- Several studies suggest offloading OS execution to one of the CMP cores.
- To be effective, must balance the cost of offloading versus the benefits.
- Offloading typically implemented by manually instrumenting a few OS routines (out of hundreds).
- Such an effort not sustainable across several operating systems and hardware configurations.

Motivation

- Improve system performance by selectively offloading OS execution.
- Offloading improves performance because:

  I. User threads don't compete with the OS for cache/ CPU/ branch predictor resources.

  II. OS invocations from different threads interact constructively at the shared OS core to yield better cache and branch predictor hit rates.

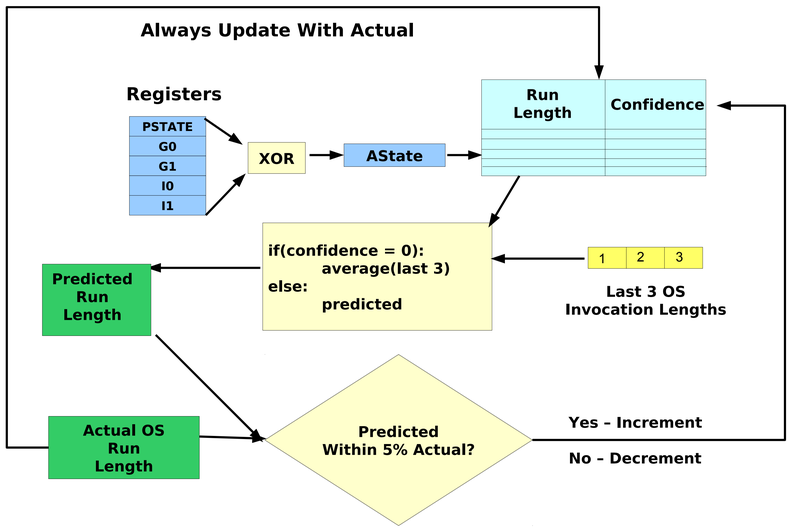
Proposal

Making offloading decisions in software often sub-optimal because it's expensive in terms of run-time overhead ***and*** applications vary in their use of OS features.

We propose offloading decision mechanisms should be supported through a hardware based OS run-length predictor.

Hardware Based Decision Making

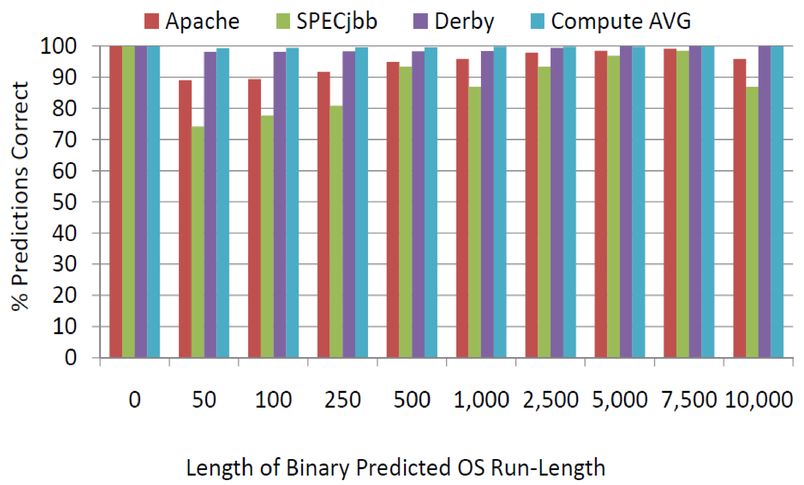
I. Predicting OS Run-Length



II. When to Offload?

- Hardware predictor above provides a discrete prediction of OS run-length.
- The offloading decision making mechanism is distilled into a ***binary prediction*** **:** if run-length exceeds N instructions, then offload OS execution.
- Estimation of ***N can be tuned for optimal behavior*** - either performance or energy-delay product (EDP).
- If the hardware is responsible for ***selecting the value of N at run-time***, then sample behavior at the start of every program phase and employ the optimal configuration until the next program phase change is detected.
- In example experiments, using L2 cache hit-rate as the feedback metric to estimate N, hardware predictor had an average runtime overhead of <1% when instrumenting all possible OS entry points.

Results



- If offloading occurs only on OS invocation run lengths >500 instructions, then the predictor makes correct off-loading decision 94.8%, 93.4%, 96.8%, and 99.6% of the time for Apache, SPECjbb2005, Derby and the average of all compute benchmarks, respectively.

Conclusion

- Averaged across all benchmarks, the predictor, while requiring only 2 KB of storage, is able to ***precisely*** predict the run length of 73.6% of all privileged instruction invocations.
- It is also able to predict ***within ±5%*** the actual run length an additional 24.8% of the time.
- Minimal software instrumentation consumes at least 16 instructions for a single fixed parameter offloading decision.
- Complex instrumentation, similar to what we implement in hardware, would take 250 instructions or more. **Our hardware decision engine is able to make this decision in just a single *cycle*.**

For more details:

Hardware Prediction of OS Run-Length For Fine-Grained Resource Customization

David Nellans, Kshitij Sudan, Erik Brunvand, Rajeev Balasubramonian, ISPASS, March 2010