

CSC 591C - Cluster Computing Project Home page

Jaydeep Marathe (jpmarath@unity.ncsu.edu)

HW5 – Progress Report

Address Trace Generation for OpenMP programs using Dynamic Instrumentation

Tasks Completed

1. Finalizing Benchmark/Test Suite

- Need good benchmarks to test tracing capability
- Looked at SPEC OMP and NAS NPB-3.0 set of benchmarks
- Categorized OpenMP directives by their type and frequency, for each NAS NPB-3.0 benchmark
- Determined the NAS NPB.30 tests to be better for testing, compared to SPEC-OMP, since they are simpler and have fewer OpenMP directives.

2. Building the Framework

- Started from uniprocessor tracing framework –METRIC [1]
- Analyzed trace instrumentation for thread-safety, modified instrumentation to be thread-safe (with pthread mutex lock/unlock calls)
- Updated tool-chain to better support target load/store instructions (support for new instructions).
- Sorting component to sort huge files – system sort breaks down after 5 Million lines of text.
- Region component to delineate barrier regions in trace, using OpenMP function calls tracing for the IBM xlc/xlf compilers.

3. Initial Testing

- Initial test – benchmark FT from the NAS NPB suite.
- 50 million memory accesses logged – 4 active OpenMP threads.
- Total time required = 45 minutes.
- Analyzed instrumentation, optimized insertion of mutex locks/unlocks. Tested again. Total time required now = 22 minutes.

Tasks Planned

1. Supporting additional OpenMP directives

- Plan to support “omp critical”, “omp barrier”, “omp do ...no wait” directives.

2. Completing traces for NAS NPB suite

- Get memory traces for the 8 NAS NPB benchmarks.

References

1. **METRIC**: a framework for extracting uni-processor address traces.
<http://moss.csc.ncsu.edu/~mueller/ftp/pub/mueller/papers/cgo03.ps.gz>

Project Web Page

<http://www4.ncsu.edu/~jpmarath/index.html>