

Adaptive Compilation and Thread Level Speculation

Ian Watson & Mikel Lujan
Advanced Processor Technologies Group
<http://www.cs.manchester.ac.uk/apt>

Managed Runtime Systems

- Working with JVMs for over 10 years
 - Members of the core development team of Jikes RVM
- Interested in all aspects of adaptive compilation environments
 - Understanding how to improve the internals of JVMs
 - Thread level speculation
 - Optimizing runtime heuristics using machine learning
 - Improving and exploiting Garbage Collection
 - Optimizations for Transactional Memory programs
 - Hardware support for highly threaded applications on JVMs

Thread Level Speculation (HPCC 2007)

- **Co-inspector thread** (collaboration with Sun Microsystems)
 - Reducing the cost of mispeculation
 - Less than 5% overhead
 - We would like sequential execution to proceed at in parallel with the speculation
 - However speculation needs to know about the read/write set of the sequential execution
 - Co-inspector thread is a compiler generated thread with only the set of instructions necessary to record the read/write set (cf. helper threads)
 - The least speculative thread offloads into the co-inspector

Machine Learning applied to Garbage Collectors

(ISMM 2007)

- Security and productivity is improved by automatically managed memory environments
- Different garbage collection exists in the literature with different sweet points
- We studied for the DaCapo benchmarks (and others) the relation between each benchmark and a set of garbage collection algorithms
- Machine learning problem:
 - Given a set of features characterising an application select the best garbage collection algorithm
 - We were able to generate a predictor and select in application specific manner the appropriate garbage collection algorithm