Extending X-Ray: A Tool for Automatic Measurement of Architectural Parameters

Contact: Kamen Yotov (kamen@yotov.org)

March 11, 2004

As we saw in class, the process of designing and developing a self-tuning system requires accurate knowledge of the underlying architecture. Arguably, the most portable way of obtaining this knowledge is through micro-benchmarks, which allow you to measure its parameters. For this reason we have started developing a tool called X-Ray, along with the required infrastructure that allows easy design/implementation of new micro-benchmarks.

Currently, X-Ray measures and reports a large number of architectural parameters such as the frequency of the CPU, the latency and throughput of instructions, the existence of instructions such as a fused multiply-add, the number of registers, the organization of different levels of the memory hierarchy etc. In principle, some of this information can be obtained by consulting the appropriate manuals, but in practice, these manuals are either non-existent or obsolete. Therefore, a tool like X-Ray is useful in many contexts such as performance analysis of programs, and in self-tuning software systems like ATLAS and FFTW.

X-Ray is written in C for maximum portability, and it is based on accurate timing of a number of carefully designed micro-benchmarks. A novel feature of X-Ray is that it has a micro-benchmark generator that can be used to automatically produce the large number of micro-benchmarks needed for architecture parameter measurement.

As part of this project you will be working on the following:

• Do a better job of measuring higher cache level and TLB parameters
  The algorithms that X-Ray uses for obtaining L1 data cache parameters are useful, but not directly applicable for higher cache levels. There is a need for more robust approach at measuring combined instruction/data caches and TLBs.

• Measure other cache latency and other cache parameters.
  Other cache parameters such as latency, write mode, replacement policy, unified code and data organization, virtual/physical indexing, etc. might be useful for certain compiler transformations to optimize performance. X-Ray currently does not attempt to measure them.

• Measure instruction cache parameters
  Knowing the exact size of the L1 instruction cache is useful for certain optimizations like loop unrolling. Currently X-Ray does not successfully measure it.

• Detect SMP/SMT, features of SIMD ISA extensions
  Having accurate measurement of the parallelism capabilities of a platform is essential for generating parallel code that performs best. SMP/SMT and SIMD instructions are just two examples, where potential for parallelization can be revealed.

• Test X-Ray in measuring parameters of challenging architectures like Intel Itanium and AMD Opteron

References

1. MOB (http://www.nmsl.cs.ucsb.edu/mob/)
3. ATLAS (http://math-atlas.sourceforge.net)
4. LMBench (http://www.bitmover.com/lmbench/)
5. X-Ray (as presented in HW1)