Time-predictable Memory Hierarchy

Bekim Chilku

Computing the Worst-Case Execution Time (WCET) of a task becomes mandatory when timing guarantees on task completion deadlines have to be given. Unfortunately WCET computation is a complex undertaking, especially for systems that use caches, out-of-order pipelines, and control speculation. The state-of-the-art WCET tools are avoiding the complexity problem by substituting the real hardware with abstracted models. However, abstraction leads to lots of unclassified model states, which in turn results in overly pessimistic WCET bound and poor processor utilization.

Single-path code is another alternative to eliminate complexity on timing analysis by transforming the conventional code into a code that has single execution trace. The approach converts all input-dependent alternatives of the code into pieces of sequential code as well as loops with input-dependent termination condition into loops with constant execution count, thus eliminating all control-flow induced variations in execution time. For obtaining the information about the timing of the code, it is sufficient to run the code once and measure the time. The major drawback of the single-path approach is its potential to end up with a quite long execution time.

In this work we address the problem of long execution time of single-path code. In particular, we focus on narrowing the speed gap between processor and main memory, by proposing a new prefetcher that brings instructions into the cache before they are required. The prefetcher exploits the time-predictable properties of single-path code to accurately predict the target of each prefetch request without polluting the cache at any moment. Another advantage of the prefetcher is its efficiency by issuing prefetch request for every possible cache miss that can occur during the runtime of the code.

In order to make the system design composable and compositional, we propose a new memory hierarchy that provides stable timing through the execution of the code. Although single-path code always runs through the same sequence of instructions, the timing of instructions can vary due to dependencies on the memory hardware states. In order to achieve stability on execution time we need to have repeatability on the history of the hardware states on each layer of the memory hierarchy. Therefore, we have defined a new memory-hierarchy organization that forces the sequence of the hardware states in the memory hierarchy to be repeatable for any iteration of the code. The memory hierarchy is also adapted to allow fetching and prefetching processes to work in parallel without interfering with each other in order to reach the best performance.

To demonstrate the applicability of the new concept, we have implemented the architecture of the system with the new memory hierarchy on an FPGA board and run experimental evaluation. The results prove the benefits that can be achieved in timing performance for single-path code.