The Subconscious Mind of a Branch Predictor

Daniel A. Jiménez, Computer Science Departments, UT San Antonio and Rutgers

When a branch predictor goes to sleep at night, what does it dream about? What does it mean for a branch predictor to dream? Normally, when the branch predictor is "awake," it predicts branches from the logic of the program and its inputs. If instead we allow the branch predictor to "sleep" and forget about the program, simply following its predictions as if they are all correct, then the branch predictor is dreaming.

We define a branch predictor dream as the sequence of addresses generated by a conditional branch predictor following its own predictions. We simulated branch predictors on the SPEC CPU 2006 benchmark `403.gcc` having them stop at random intervals to collect dreams of 64K addresses. (Results are similar for other benchmarks.) We simulated 12 different branch predictors ranging from very simple to state-of-the-art, allowing them to all train on the same data for millions of branches, periodically stopping them at the same points to collect their dreams.

In this limited space, we discuss two ways to characterize branch predictor dreams:

Dream prelude and period lengths. A dream has two phases: a *prelude* during which a number of different branches are visited, followed by a *period* where the branch predictor repeats the same several branches forever. We quantify the complexity of a branch predictor dream by measuring the lengths of the prelude and period. Figure 1 shows the average prelude and period lengths. Surprisingly, the BiMode predictor, a predictor that is about in the middle of the pack in terms of accuracy, has by far the longest average prelude and period.

Control flow graph. A branch predictor dream induces an imaginary control-flow graph (CFG), or *dreamscape*. Figure 2 shows four CFGs: a real one and three dreamscapes from different branch predictors. Each of the dreamscapes represent the graph with the largest number of nodes from its branch predictor over the traces simulated. Interestingly, the largest dreamscape for L-TAGE has only two nodes, the same as the largest number for a static not-taken policy (not shown). The BiMode predictor has the largest dreamscape of all predictors, but its dreamscape is dwarfed by the real control flow graph.

Questions:

1. How similar are the dreams to the actual code the branch predictor encounters while it is awake?

2. If we let the branch predictor dream when the processor is idle, will that improve its accuracy, or somehow give us insight into some other aspect of performance?

3. One might expect that a more accurate predictor would have more complex dreams, but this is not the case. Why is that? What can we learn from that?

4. Can we examine the dreams of other structures, such as caches, prefetchers, instruction queues, etc.? What would that mean?

5. A branch predictor dream is just a string. To what extent can we manipulate the initial conditions of the predictor to produce a given string (for e.g. the purpose of compression)?

6. Branch predictor dreams can be thought of as similar to human dreams. The prelude is the "interesting part" of the dream while the period is the part where e.g. you've lost something and you are repetitively trying to find it. So, can we learn anything about human dreaming from machine dreaming?

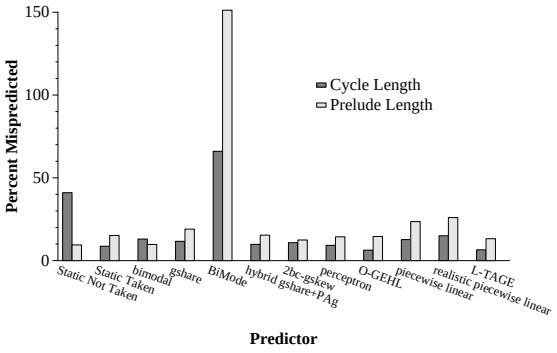

Figure 1: Dream prelude and period lengths.

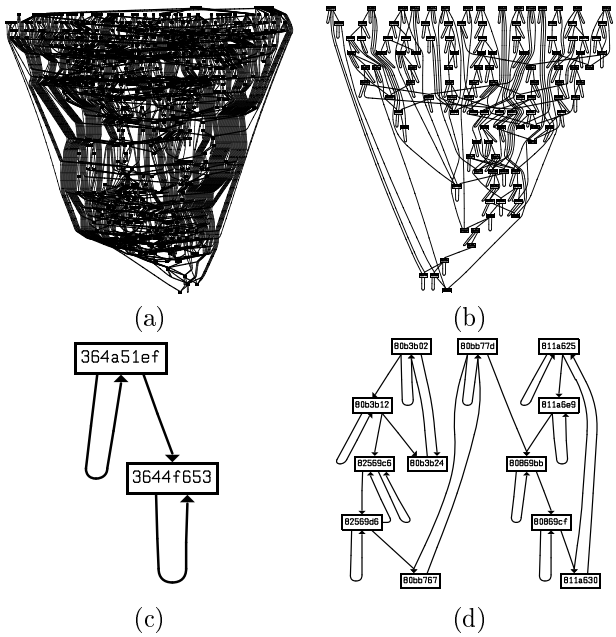

Figure 2: (a): actual control-flow-graph; CFG dreamscapes from (b): BiMode, (c): L-TAGE, and (d): piecewise linear branch predictors