Combining Branch Predictors

黄科乐 201828013229084

Time: 10-18

1 Introduction

Both superscalar and superpipelining techniques are becoming increasingly popular recently.With both these techniques,branch instructions are increasingly important in determining overall machine performance.The branch performance problem can be divided into two subproblems. First, a prediction of the branch direction is needed. Second, for taken branches, the instructions from the branch target must be available for execution with minimal delay.Hardware branch prediction strategies have been studied extensively.Local branch prediction technique and global branch prediction technique have their own advantages.

This paper introduces a new technique that allows the distinct advantages of different branch predictors to be combined. The technique uses multiple branch predictors and selects the one which is performing best for each branch. This approach is shown to provide more accurate predictions than any one predictor alone. This paper also shows a method of increasing the utility of branch history by hashing it together with the branch address.

2 Bimodal Branch Prediction

Since the behavior of typical branches is far from random, bimodal branch prediction takes advantage of this bimodal distribution of branch behavior and attempts to distinguish usually taken from usually nottaken branches.As the simplest method, the predictor can tolerate a branch going an unusual direction one time and keep predicting the usual branch direction by using a 2-bit counter, but may result in degraded prediction accuracy. One alternate implementation is to store a tag with each counter and use a set-associative lookup to match counters with branches. We will use the SPEC’ 89 benchmarks.

3 Local Branch Prediction

One way to improve on bimodal prediction is to recognize that many branches execute repetitive patterns.With 3 bits of history and 23 counters, the local branch predictor will be able determine the current iteration and always make the correct prediction after some initial settling of the counter values.For very small predictors, the local scheme is actually worse than the bimodal scheme. If there is excessive contention for history entries, then storing this history is of no value. However, above about 128 bytes, the local predictor has significantly better performance. For large predictors, the accuracy approaches percentage of 97.1 correct, with less than half as many misspredictions as the bimodal scheme.

4 Global Branch Prediction

In the local branch prediction scheme, the only patterns considered are those of the current branch.Global branch prediction is able to take advantage of two types of patterns. First, the direction take by the current branch may depend strongly on other recent branches. A second way that global branch prediction can be effective is by duplicating the behavior of local branch prediction. This can occur when the global history includes all the local history needed to make an accurate prediction.

5 Global Predictor with Index Selection

Global history information is less efficient at identifying the current branch than simply using the branch address. This suggests that a more efficient prediction might be made using both the branch address and the global history.As we would expect, gselect-best performs better than either bimodal or global prediction since both are essentially degenerate cases.

6 Global History with Index Sharing

If there are enough address bits to identify the branch, we can expect the frequent global history combinations to be rather sparse. We can take advantage of this effect by hashing the branch address and the global history together. In particular, we can expect the exclusive OR of the branch address with the global history to have more information than either component alone. Moreover, since more address bits and global history bits are in use, there is reason to expect better predictions than gselect.

7 Combining Branch Predictors

The different branch prediction schemes we have presented have different advantages. A natural question is whether the different advantages can be combined in a new branch prediction method with better prediction accuracy. In this combination, global information can be used if it is worthwhile, otherwise the usual branch direction as predicted by the bimodal scheme can be used.

8 Conclusions

we have presented two new methods for improving branch prediction performance. First, we showed that using the bit-wise exclusive OR of the global branch history and the branch address to access predictor counters results in better performance for a given counter array size. We also showed that the advantages of multiple branch predictors can be combined by providing multiple predictors and by keeping track of which predictor is more accurate for the current branch.

9 Suggestions for Future Work

There are a number of ways this study could be extended, First, there are a large number of parameters such as sizes, associativities, and pipeline costs that were not fully explored here.Second, other sources of information such as whether the branch target is forward or backward might be usefully added to increase accuracy. Third, the typically sparse branch history might be compressed to reduce the number of counters needed. Finally, a compiler with profile support might be able to reduce or eliminate the need for branch predictors as described here.