The Effect of Speculatively Updating Branch History on Branch Prediction Accuracy, Revisited

Eric Hao, Po-Yung Chang, and Yale N. Patt
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122

Abstract

Recent research [6] has suggested that the branch history register need not contain the outcomes of the most recent branches in order for the Two-Level Adaptive Branch Predictor to work well. From this result, it is tempting to conclude that the branch history register need not be speculatively updated. This paper revisits this work and explains when the most recent branch outcomes can be omitted without significantly affecting performance. It also explains why this result does not imply that speculative update is not important. This paper shows that because the number of unresolved branches present in the machine varies during program execution, branch predictors without speculative update perform significantly worse than branch predictors with speculative update.

Keywords: Two-Level Adaptive Branch Prediction, dynamic branch prediction, speculative execution, superscalar processors, out-of-order execution

1 Introduction

Very accurate branch prediction is a critical requirement for high performance wide-issue, deeply pipelined processors. To address this need, many different branch prediction algorithms have been developed [2, 3]. In this paper, we will be examining the global variation of the Two-Level Adaptive Branch Predictor [7, 8, 4]. Its predictions are based on the outcomes of previously issued branches which are stored in the branch history register. Because the processor may be speculatively issuing instructions from a point far ahead of where it is executing in the dynamic instruction stream, the predictor may be basing its predictions on branch outcomes that have yet to be resolved. Past research [8] has stated that because the outcomes of the most recent branches are crucial for accurate branch prediction, the predictor should be speculatively updated with their predicted outcomes whether they are resolved or not.

0

Recent research [6] has questioned the importance of including such branches in the branch history register. They examined the performance of a version of the Two-Level Adaptive Branch Predictor (the *skipped* model) which excluded a fixed number of the most recently issued branches from the branch history register. As the number of branches excluded was increased, the prediction accuracy of the skipped model remained fairly constant. Based on this result, they concluded that branch prediction accuracy is not significantly affected when unresolved branches are excluded from the branch history register (i.e. speculative update does not significantly improve performance). In addition, they showed that in the presence of unresolved branches, speculatively updating the branch history register actually lowers prediction accuracy.

In this paper, we revisit the previous study [6]. In section 2, we present the results of further experiments that provide an explanation for when excluding the most recent branches from the branch history register does not significantly affect prediction accuracy. In section 3, we reexamine the usefulness of speculatively updating the branch history register. We show that the prediction accuracies of predictors with speculative update do not degrade in the presence of unresolved branches. Furthermore, we show that because the number of unresolved branches present in the machine varies, branch predictors that omit unresolved branches from their history registers perform significantly worse than branch predictors that speculatively update their history registers. Section 4 provides some concluding remarks.

2 Branch Prediction Based on Older Histories

In this section, we revisit the results of the skipped model experiment [6]. The skipped model predictor is a speculatively updated predictor that bases its predictions on branch histories that are older than the ones used by the standard predictor. It exchanges some fixed number of the most recently issued branches from its branch history register for an equal number of older branches (see figure 1). The original results of the skipped model experiment found that varying the number of recent branches omitted from one to four has little affect on the prediction accuracy.

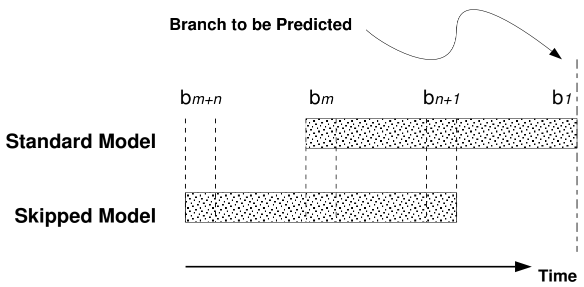

Figure 1: m-bit branch history registers for the standard and skipped models with n unresolved branches.

We repeated the skipped model experiment using a trace-driven simulator that modeled the global variation of the Two-Level Adaptive Branch Predictor. The predictor devoted one pattern history table to each static branch in the program. This eliminated from our experiment any branch mispredictions due to pattern history table conflicts. This way, any mispredictions that occurred in the skipped model predictor that did not occur in the standard predictor would only be due to the difference in recorded branch histories. The pattern history tables were updated immediately after the prediction[1]. For each predictor configuration, we measured prediction accuracy for five of the six SPEC92 integer benchmarks: espresso, xlisp, eqntott, compress, and gcc[2]. Each benchmark was simulated for 10 million conditional branch instructions. Our results confirmed those of the original experiment [6]: prediction accuracy remains fairly constant as the number of skipped branches is increased. The results are omitted from this paper due to space considerations.

The results of the skipped model experiment indicate that the the outcomes of recently issued branches are not significantly more useful for predicting branches than the outcomes of older branches. For a given branch prediction, the outcomes of the most recently issued branches either provide useful information for the prediction or do not provide useful information. If they do not provide useful information, excluding them from the branch history register should not affect prediction accuracy which is consistent with the experimental results. If they do provide useful information, excluding them should lower prediction accuracy which is not consistent with the experimental results.

To resolve this contradiction, we reran the skipped model experiments using the same predictor configurations and recorded the recently issued branch outcomes that were omitted for each branch prediction. For each prediction of a given static branch instruction, we recorded the contents of the branch history register and the sequence of omitted branch outcomes. If the predictor was skipping n branches, then for a given static branch and history register pattern, there would be 2^n possible sequences of omitted branch outcomes. The results showed that for most pairs of static branches and history register patterns, a single sequence would occur an overwhelmingly large majority of the time. We call this sequence the *dominant skipped sequence*. Figures 2— 6 plot for each predictor configuration the fraction of branch predictions in which a dominant skipped sequence occurred. If we only consider predictor configurations with branch history register lengths of at least eight, this fraction never dropped below 93%.

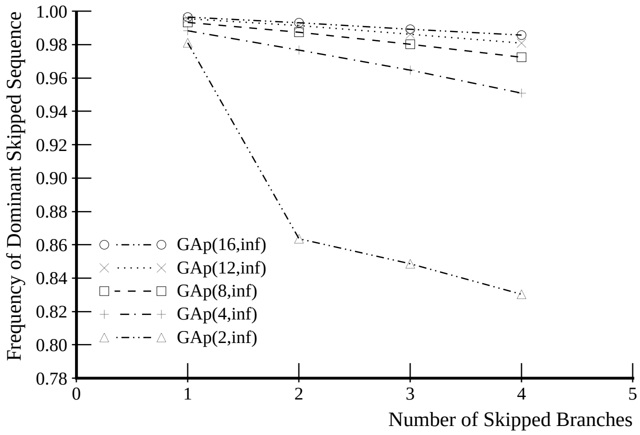

Figure 2: The fraction of branch predictions in which a dominant skipped sequence occurred — espresso.

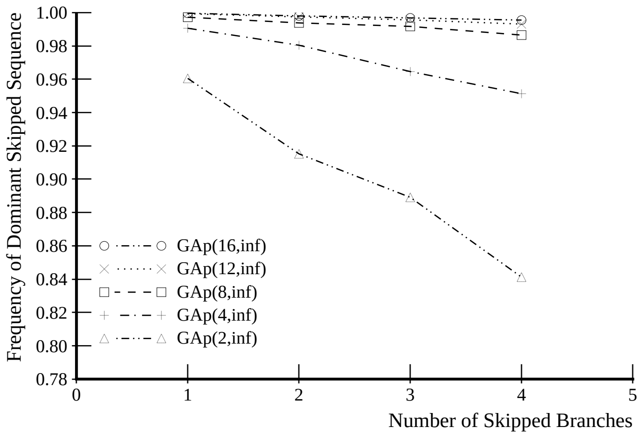

Figure 3: The fraction of branch predictions in which a dominant skipped sequence occurred — xlisp.

[1] A real predictor would probably wait until after the branch was retired to update the pattern history tables. A previous study[9] has shown that updating the tables immediately provides no appreciable gain in prediction accuracy.

[2] Sc was omitted due to problems with the simulator.

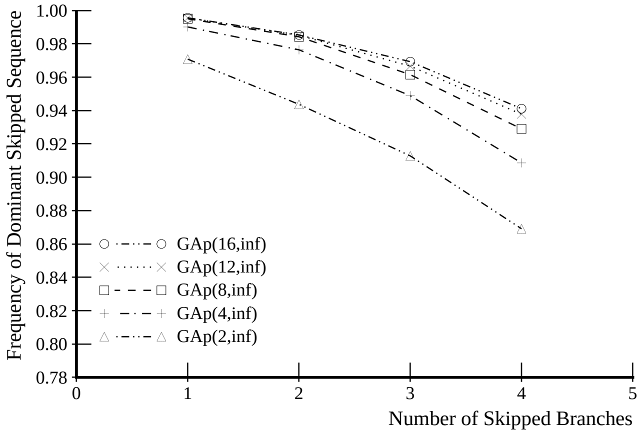

Figure 4: The fraction of branch predictions in which a dominant skipped sequence occurred — eqntott.

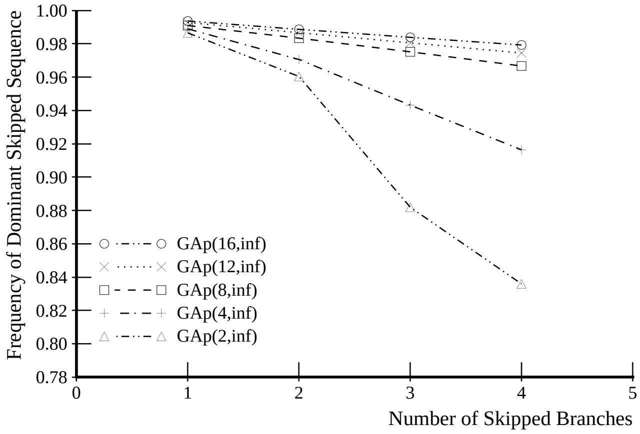

Figure 5: The fraction of branch predictions in which a dominant skipped sequence occurred — compress.

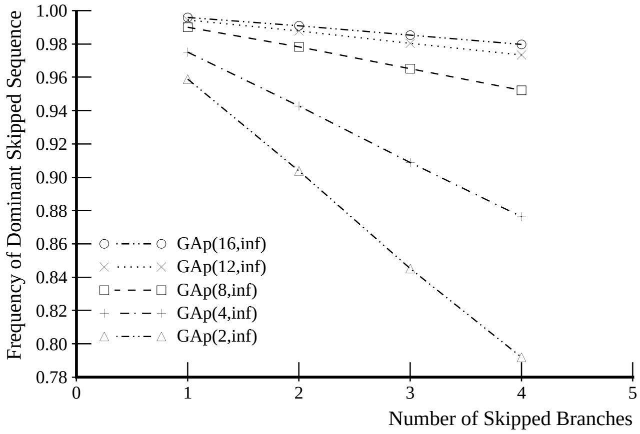

Figure 6: The fraction of branch predictions in which a dominant skipped sequence occurred — gcc.

Because the dominant skipped sequence occurs such a large majority of the time, the value of the sequence can be thought of as being implicitly represented by its associated static branch and history register pattern pair. This implicit representation of the omitted branch outcomes allows the skipped model predictor to achieve a prediction accuracy comparable to that of the standard predictor. As a result, the skipped model predictor does not exchange information about recently issued branch outcomes in return for information about older branch outcomes for every branch prediction it makes. Such exchanges are performed for the rare instances in which the omitted branch outcomes do not match the dominant skipped sequence. For the remaining instances, the skipped model predictor receives the information for free.

3 Speculative Update

In this section, we show the positive impact speculatively updating the branch history register makes towards accurate branch prediction. We show that the prediction accuracies of predictors with speculative update are independent of the number of unresolved branches present in the machine. Furthermore, we show that because this number varies during program execution, the prediction accuracies of predictors without speculative update are significantly lower than those of predictors with speculative update.

Recent research [6] erroneously reported that the prediction accuracy of speculatively updated predictors dramatically decreased as the number of unresolved branches present in the machine increased. The study's author reports that the incorrect result was due to an error in their simulator[5]. The correct result shows that the prediction accuracy is not affected by the number of unresolved branches present. Because only branch predictions made while the machine is speculatively executing down the correct path of the program affect the execution time, only those predictions contribute to the calculation of prediction accuracy [9]. For such predictions, all the branch outcomes contained in the branch history register are guaranteed to be correct, regardless of whether they are resolved or not.

The skipped model experiment showed that omitting a fixed number of branches from the branch history register does not significantly affect prediction accuracy. This result does not apply to a predictor that omits all unresolved branches, because the number of unresolved branches can vary during program execution. To determine the effect of omitting unresolved branches, we simulated such a predictor and compared its performance to that of a predictor with speculative update.

Each predictor's performance was measured using a trace-driven simulator. The simulator modeled a dynamically-scheduled machine that could issue up to

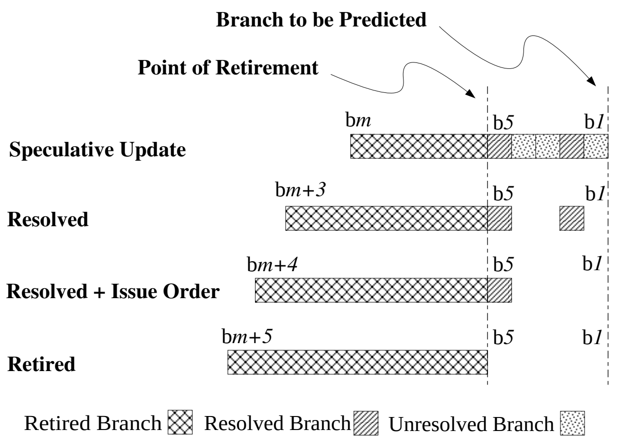

Figure 7: m bit branch history registers for predictors with speculative update and the three variations of predictors without speculative update.

eight instructions per cycle with at most one branch issued per cycle. The machine had eight functional units and a branch predictor with a 16 bit global branch history register and eight pattern history tables. The three least significant bits of the branch's word-aligned address specified the pattern history table to be used for the prediction. Branches were not resolved until after all the instructions upon which they depended were executed and the branch instruction itself was executed. Upon resolving a mispredicted branch, the machine was able to recover immediately to the correct path via checkpointing [1]. The same set of benchmarks used in section 2 was used to measure each predictor's performance. Each benchmark was simulated for 100 million instructions.

We considered three variations in which a predictor without speculative update could be modeled: *resolved*, *resolved+issue order*, and *retired*. In the resolved variation, the branch history register is updated with the outcome of a branch as soon as that branch is resolved. In the resolved+issue order variation, the branch history register is updated with the outcome of a branch as soon as that branch is resolved and all the branches issued before it are resolved. This variation is identical to the skipped model with the exception that we allowed the number of unresolved branches to vary during the execution of a program. The retired variation updates the branch history register with the results of a branch as soon as the checkpoint associated with the branch is retired (i.e. all the instructions issued before it have been executed). Figure 7 illustrates the differences between the variations.

We measured the prediction accuracy and number of instructions retired per cycle (IPC) for each benchmark. The simulation results are shown in figures 8 and 9. All

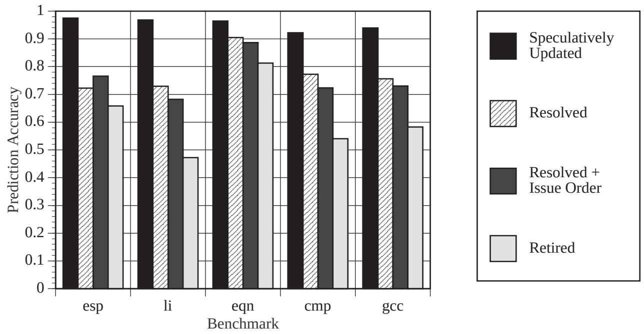

Figure 8: Branch prediction accuracies for the four branch history update variations of the out-of-order model.

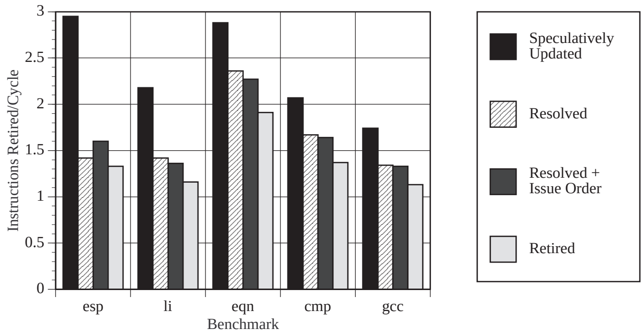

Figure 9: IPC's for the four branch history update variations of the out-of-order model.

three variations of the predictor without speculative update had significantly lower prediction accuracies than the predictor with speculative update. The resolved, resolved+issue order, and retired variations respectively suffered average decreases of 19%, 21%, and 36% in prediction accuracy. They had corresponding average decreases of 29%, 30%, and 41% in IPC.

The decrease in prediction accuracy was not due to the exclusion of the most recently issued branches from the branch history registers of the predictors without speculative update; the skipped model experiment showed that those branches were not needed for accurate branch prediction. The decrease was due to the number of unresolved branches present in the machine varying from cycle to cycle. The branch predictor uses the branch history register to identify what state the program is in and bases its predictions on that state. By allowing the number of unresolved branches to vary, the branch history register can no longer be guaranteed to contain the same sequence of branches for every dynamic occurrence of a particular program state, thereby removing the predictor's ability to identify distinct states in the program. Without this ability, the predictor can no longer make accurate predictions.

Consider the example of the resolved variation making predictions for different dynamic occurrences of the

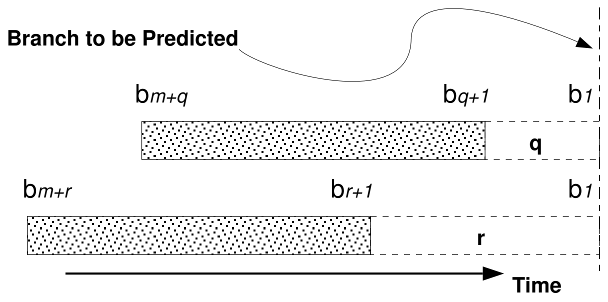

Figure 10: The branch history register contents used by the resolved variation for predicting different dynamic instances of branch b_0.

same branch (see figure 10). Suppose the unresolved branches in the machine at the time of the first prediction are such that the q most recently issued branches must be omitted from the branch history register. Suppose for the second prediction, the r most recently issued branches must be omitted, where q and r are different. If the actual branch histories of the program for the two dynamic occurrences are the same, then the predictor should be making the same prediction for both of them. However, because the number of unresolved branches present varied for the two predictions, the contents of the branch history register used were different. This causes the predictor to use a different pattern history table entry to make each prediction, potentially making different predictions and lowering prediction accuracy.

4 Conclusion

In this paper, we reexamined the skipped model experiment and the use of speculative update in Two-Level Adaptive Branch Prediction. We showed that the outcomes of recently issued branches can be omitted from the branch history register without significantly affecting prediction accuracy provided that the number of outcomes omitted is fixed because the skipped model can most often infer those outcomes from the older branch history and the static branch instruction to be predicted. We also showed that speculatively updated branch predictors are not adversely affected by the occurrence of unresolved branches. Their prediction accuracies are independent of the number of unresolved branches present in the machine. More importantly, predictors with speculative update were shown to far outperform predictors without speculative update, because branch history registers that are not speculatively updated lose the ability to identify distinct program states and hence the ability to make accurate predictions.

Acknowledgments

This paper is one result of our ongoing research in high performance computer implementation at the University of Michigan. The support of our industrial partners: Intel, AT&T/GIS, Motorola, Hewlett-Packard, and Scientific and Engineering Software is greatly appreciated. We would also like to thank Adam Talcott for the clear explanations he provided about his work and the reviewers of this paper for their helpful suggestions.

References

[1] W.-M. W. Hwu and Y. N. Patt, "Checkpoint repair for out-of-order execution machines," in *Proceedings of the 14th Annual International Symposium on Computer Architecture*, pp. 18–26, 1987.

[2] J. K. F. Lee and A. J. Smith, "Branch prediction strategies and branch target buffer design," *IEEE Computer*, pp. 6–22, January 1984.

[3] S. McFarling and J. Hennessy, "Reducing the cost of branches," in *Proceedings of the 13th Annual International Symposium on Computer Architecture*, pp. 396–403, 1986.

[4] S.-T. Pan, K. So, and J. T. Rahmeh, "Improving the accuracy of dynamic branch prediction using branch correlation," in *Proceedings of the Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 76–84, 1992.

[5] A. R. Talcott, June 1994. Personal communication.

[6] A. R. Talcott, W. Yamamoto, M. J. Serrano, R. C. Wood, and M. Nemirovsky, "The impact of unresolved branches on branch prediction scheme performance," in *Proceedings of the 21st Annual International Symposium on Computer Architecture*, pp. 12–21, 1994.

[7] T.-Y. Yeh and Y. N. Patt, "Two-level adaptive branch prediction," in *Proceedings of the 24th Annual ACM/IEEE International Symposium on Computer Microarchitecture*, pp. 51–61, 1991.

[8] T.-Y. Yeh and Y. N. Patt, "Alternative implementations of two-level adaptive branch prediction," in *Proceedings of the 19th Annual International Symposium on Computer Architecture*, pp. 124–134, 1992.

[9] T.-Y. Yeh and Y. N. Patt, "A comprehensive instruction fetch mechanism for a processor supporting speculative execution," in *Proceedings of the 25th Annual ACM/IEEE International Symposium on Computer Microarchitecture*, pp. 129–139, 1992.