CSE240A Reading Assignment

Xinhao Luo

TOTAL POINTS

24 / 24

QUESTION 1

1 Paper Summary **5 / 5**

✓ **- 0 pts** **Correct**

QUESTION 2

2 Strengths **5 / 5**

✓ **- 0 pts** **Correct**

QUESTION 3

3 Weaknesses **5 / 5**

✓ **- 0 pts** **Correct**

QUESTION 4

4 Novelty **1 / 1**

✓ **- 0 pts** **Correct**

QUESTION 5

5 Writing Quality **1 / 1**

✓ **- 0 pts** **Correct**

QUESTION 6

6 Experimental Methodology **1 / 1**

✓ **- 0 pts** **Correct**

QUESTION 7

7 Overall Merit **1 / 1**

✓ **- 0 pts** **Correct**

QUESTION 8

8 Detailed Comments for Author **5 / 5**

✓ **- 0 pts** **Correct**

Paper title: Whisper: Profile-Guided Branch Misprediction Elimination for Data Center Applications

==+== A. Paper summary

Modern data center applications experience frequent branch mispredictions, and current predictors can't handle these applications properly since the code footprints of these applications are too large. To solve this problem, the author proposed whisper, it investigates the in-production profile of data center applications to identify precise program contexts that lead to branch mispredictions. These hints are then inserted into codes to avoid mispredictions.

The idea behind Whisper is profile-guided branch prediction, but instead of traditional ways, it combines the three novel techniques. When doing run-time profiling, whisper also makes use of the advanced collection technique (Intel PT and LBR) and does branch analysis accordingly. Whisper's offline analysis identifies branches with the history-based Boolean formulas and injects hints into the binary.

The author then compares the result with the traditional predictors and finds Whisper outperform them by twice on average. It also has ~44% speedup compared with MTAGE-SC with unlimited storage.

==+== B. Strengths (2~3 bullets)

1. Whisper provides a combination of three novel profile-guided techniques(hashed history correlation, randomized formula testing, and extended ROMBF) that prove effective in branch prediction, reaching a profoundly accurate branch prediction.
2. Instead of an unlimited storage approach. Whisper used history hash and correlation to reduce the size of required length by using various history lengths.

==+== C. Weaknesses (2~3 bullets)

1. The hint is encoded into the binary after offline analysis. It requires modification on the binary, and well as a stage to do offline analysis of the binary.
2. The added implication and converse non-implications require minor micro-architectural modification to the original hardware implementation of ROMBF, which adds costs to adopt Whisper.

==+== D. Novelty:

4

==+== E. Writing quality:

4

1 Paper Summary 5 / 5

✓ - 0 pts Correct

Paper title: Whisper: Profile-Guided Branch Misprediction Elimination for Data Center Applications

==+== A. Paper summary

Modern data center applications experience frequent branch mispredictions, and current predictors can't handle these applications properly since the code footprints of these applications are too large. To solve this problem, the author proposed whisper, it investigates the in-production profile of data center applications to identify precise program contexts that lead to branch mispredictions. These hints are then inserted into codes to avoid mispredictions.

The idea behind Whisper is profile-guided branch prediction, but instead of traditional ways, it combines the three novel techniques. When doing run-time profiling, whisper also makes use of the advanced collection technique (Intel PT and LBR) and does branch analysis accordingly. Whisper's offline analysis identifies branches with the history-based Boolean formulas and injects hints into the binary.

The author then compares the result with the traditional predictors and finds Whisper outperform them by twice on average. It also has ~44% speedup compared with MTAGE-SC with unlimited storage.

==+== B. Strengths (2~3 bullets)

1. Whisper provides a combination of three novel profile-guided techniques(hashed history correlation, randomized formula testing, and extended ROMBF) that prove effective in branch prediction, reaching a profoundly accurate branch prediction.
2. Instead of an unlimited storage approach. Whisper used history hash and correlation to reduce the size of required length by using various history lengths.

==+== C. Weaknesses (2~3 bullets)

1. The hint is encoded into the binary after offline analysis. It requires modification on the binary, and well as a stage to do offline analysis of the binary.
2. The added implication and converse non-implications require minor micro-architectural modification to the original hardware implementation of ROMBF, which adds costs to adopt Whisper.

==+== D. Novelty:

4

==+== E. Writing quality:

4

2 Strengths 5 / 5

✓ **- 0 pts** Correct

gradescope

Paper title: Whisper: Profile-Guided Branch Misprediction Elimination for Data Center Applications

==+== A. Paper summary

Modern data center applications experience frequent branch mispredictions, and current predictors can't handle these applications properly since the code footprints of these applications are too large. To solve this problem, the author proposed whisper, it investigates the in-production profile of data center applications to identify precise program contexts that lead to branch mispredictions. These hints are then inserted into codes to avoid mispredictions.

The idea behind Whisper is profile-guided branch prediction, but instead of traditional ways, it combines the three novel techniques. When doing run-time profiling, whisper also makes use of the advanced collection technique (Intel PT and LBR) and does branch analysis accordingly. Whisper's offline analysis identifies branches with the history-based Boolean formulas and injects hints into the binary.

The author then compares the result with the traditional predictors and finds Whisper outperform them by twice on average. It also has ~44% speedup compared with MTAGE-SC with unlimited storage.

==+== B. Strengths (2~3 bullets)

1. Whisper provides a combination of three novel profile-guided techniques(hashed history correlation, randomized formula testing, and extended ROMBF) that prove effective in branch prediction, reaching a profoundly accurate branch prediction.
2. Instead of an unlimited storage approach. Whisper used history hash and correlation to reduce the size of required length by using various history lengths.

==+== C. Weaknesses (2~3 bullets)

1. The hint is encoded into the binary after offline analysis. It requires modification on the binary, and well as a stage to do offline analysis of the binary.
2. The added implication and converse non-implications require minor micro-architectural modification to the original hardware implementation of ROMBF, which adds costs to adopt Whisper.

==+== D. Novelty:

4

==+== E. Writing quality:

4

3 Weaknesses 5 / 5

✓ **- 0 pts** Correct

Paper title: Whisper: Profile-Guided Branch Misprediction Elimination for Data Center Applications

==+== A. Paper summary

Modern data center applications experience frequent branch mispredictions, and current predictors can't handle these applications properly since the code footprints of these applications are too large. To solve this problem, the author proposed whisper, it investigates the in-production profile of data center applications to identify precise program contexts that lead to branch mispredictions. These hints are then inserted into codes to avoid mispredictions.

The idea behind Whisper is profile-guided branch prediction, but instead of traditional ways, it combines the three novel techniques. When doing run-time profiling, whisper also makes use of the advanced collection technique (Intel PT and LBR) and does branch analysis accordingly. Whisper's offline analysis identifies branches with the history-based Boolean formulas and injects hints into the binary.

The author then compares the result with the traditional predictors and finds Whisper outperform them by twice on average. It also has ~44% speedup compared with MTAGE-SC with unlimited storage.

==+== B. Strengths (2~3 bullets)

1. Whisper provides a combination of three novel profile-guided techniques(hashed history correlation, randomized formula testing, and extended ROMBF) that prove effective in branch prediction, reaching a profoundly accurate branch prediction.
2. Instead of an unlimited storage approach. Whisper used history hash and correlation to reduce the size of required length by using various history lengths.

==+== C. Weaknesses (2~3 bullets)

1. The hint is encoded into the binary after offline analysis. It requires modification on the binary, and well as a stage to do offline analysis of the binary.
2. The added implication and converse non-implications require minor micro-architectural modification to the original hardware implementation of ROMBF, which adds costs to adopt Whisper.

==+== D. Novelty:

4

==+== E. Writing quality:

4

4 Novelty 1 / 1

✓ **- 0 pts** Correct

Paper title: Whisper: Profile-Guided Branch Misprediction Elimination for Data Center Applications

==+== A. Paper summary

Modern data center applications experience frequent branch mispredictions, and current predictors can't handle these applications properly since the code footprints of these applications are too large. To solve this problem, the author proposed whisper, it investigates the in-production profile of data center applications to identify precise program contexts that lead to branch mispredictions. These hints are then inserted into codes to avoid mispredictions.

The idea behind Whisper is profile-guided branch prediction, but instead of traditional ways, it combines the three novel techniques. When doing run-time profiling, whisper also makes use of the advanced collection technique (Intel PT and LBR) and does branch analysis accordingly. Whisper's offline analysis identifies branches with the history-based Boolean formulas and injects hints into the binary.

The author then compares the result with the traditional predictors and finds Whisper outperform them by twice on average. It also has ~44% speedup compared with MTAGE-SC with unlimited storage.

==+== B. Strengths (2~3 bullets)

1. Whisper provides a combination of three novel profile-guided techniques(hashed history correlation, randomized formula testing, and extended ROMBF) that prove effective in branch prediction, reaching a profoundly accurate branch prediction.
2. Instead of an unlimited storage approach. Whisper used history hash and correlation to reduce the size of required length by using various history lengths.

==+== C. Weaknesses (2~3 bullets)

1. The hint is encoded into the binary after offline analysis. It requires modification on the binary, and well as a stage to do offline analysis of the binary.
2. The added implication and converse non-implications require minor micro-architectural modification to the original hardware implementation of ROMBF, which adds costs to adopt Whisper.

==+== D. Novelty:

4

==+== E. Writing quality:

4

5 Writing Quality 1 / 1

✓ **- 0 pts** Correct

==+== F. Experimental methodology:

4

==+== G. Overall merit:

Accept

==+== H. Detailed comments for author:

Whisper combines various profile-guided techniques to improve branch prediction. First, it mentioned about hased history correlation, which effectively encoded high misprediction branches with lightweight Boolean formulas. Randomized formula testing further narrows the search space by selecting a series of locally-optimal formulas among all possible formulas. The extended ROMBF with implication and converse nonimplication improves the branch history coverage of formulas with minimal overhead. The extended ROMBF handles the case where its original version cannot consider variables in formulas that appear twice. With the extended operator, it improves the prediction accuracy while the storage requirements increase linearly.

The history correlation considers the case where some branches' outcome depends on the most recent branch while some others may depend on older one. Instead of fixed history length, Whisper considers various potential lengths for each static branch and selects the best possible outcome. For each branch, Whisper does not operate on raw, full-length histories. The hashed histories allow comparison histories of different original lengths. A Boolean formula is selected to be the one best for the substream. Whisper then counts the total number of mispredictions with different history length and formula. In the case where none of them are better, Whisper hints that this branch should be predicted dynamically, handled by the underlying branch predictor. The author also mentions the parameter they find best for the correlation technique.

However, it is worth mentioning that the usage model mentioned in the paper requires binary updates. Whisper first runs a profile over existing binary, doing branch trace and prediction accuracy evaluation, and then injects hints to the binary to produce an optimized binary for the next deployment cycles. This means that the analysis does not take effect immediately and has to be postponed until the next release. The binary update adds cost for more generic adoption, as data center application deployment cycles might be really long in order to maintain stability and support critical services. Also, during this period there may also be workload changes that might invalidate some previous optimization conclusion. The extended ROMBF is also an obstacle for wider adoption as it requires hardware modifications. Data centers have extra cost to deploy this solution immediately and might need to wait until the next device update to enjoy the benefit from this feature.

6 Experimental Methodology 1 / 1

✓ - 0 pts Correct

==+== F. Experimental methodology:

4

==+== G. Overall merit:

Accept

==+== H. Detailed comments for author:

Whisper combines various profile-guided techniques to improve branch prediction. First, it mentioned about hased history correlation, which effectively encoded high misprediction branches with lightweight Boolean formulas. Randomized formula testing further narrows the search space by selecting a series of locally-optimal formulas among all possible formulas. The extended ROMBF with implication and converse nonimplication improves the branch history coverage of formulas with minimal overhead. The extended ROMBF handles the case where its original version cannot consider variables in formulas that appear twice. With the extended operator, it improves the prediction accuracy while the storage requirements increase linearly.

The history correlation considers the case where some branches' outcome depends on the most recent branch while some others may depend on older one. Instead of fixed history length, Whisper considers various potential lengths for each static branch and selects the best possible outcome. For each branch, Whisper does not operate on raw, full-length histories. The hashed histories allow comparison histories of different original lengths. A Boolean formula is selected to be the one best for the substream. Whisper then counts the total number of mispredictions with different history length and formula. In the case where none of them are better, Whisper hints that this branch should be predicted dynamically, handled by the underlying branch predictor. The author also mentions the parameter they find best for the correlation technique.

However, it is worth mentioning that the usage model mentioned in the paper requires binary updates. Whisper first runs a profile over existing binary, doing branch trace and prediction accuracy evaluation, and then injects hints to the binary to produce an optimized binary for the next deployment cycles. This means that the analysis does not take effect immediately and has to be postponed until the next release. The binary update adds cost for more generic adoption, as data center application deployment cycles might be really long in order to maintain stability and support critical services. Also, during this period there may also be workload changes that might invalidate some previous optimization conclusion. The extended ROMBF is also an obstacle for wider adoption as it requires hardware modifications. Data centers have extra cost to deploy this solution immediately and might need to wait until the next device update to enjoy the benefit from this feature.

7 Overall Merit 1 / 1

✓ **- 0 pts** Correct

==+== F. Experimental methodology:

4

==+== G. Overall merit:

Accept

==+== H. Detailed comments for author:

Whisper combines various profile-guided techniques to improve branch prediction. First, it mentioned about hased history correlation, which effectively encoded high misprediction branches with lightweight Boolean formulas. Randomized formula testing further narrows the search space by selecting a series of locally-optimal formulas among all possible formulas. The extended ROMBF with implication and converse nonimplication improves the branch history coverage of formulas with minimal overhead. The extended ROMBF handles the case where its original version cannot consider variables in formulas that appear twice. With the extended operator, it improves the prediction accuracy while the storage requirements increase linearly.

The history correlation considers the case where some branches' outcome depends on the most recent branch while some others may depend on older one. Instead of fixed history length, Whisper considers various potential lengths for each static branch and selects the best possible outcome. For each branch, Whisper does not operate on raw, full-length histories. The hashed histories allow comparison histories of different original lengths. A Boolean formula is selected to be the one best for the substream. Whisper then counts the total number of mispredictions with different history length and formula. In the case where none of them are better, Whisper hints that this branch should be predicted dynamically, handled by the underlying branch predictor. The author also mentions the parameter they find best for the correlation technique.

However, it is worth mentioning that the usage model mentioned in the paper requires binary updates. Whisper first runs a profile over existing binary, doing branch trace and prediction accuracy evaluation, and then injects hints to the binary to produce an optimized binary for the next deployment cycles. This means that the analysis does not take effect immediately and has to be postponed until the next release. The binary update adds cost for more generic adoption, as data center application deployment cycles might be really long in order to maintain stability and support critical services. Also, during this period there may also be workload changes that might invalidate some previous optimization conclusion. The extended ROMBF is also an obstacle for wider adoption as it requires hardware modifications. Data centers have extra cost to deploy this solution immediately and might need to wait until the next device update to enjoy the benefit from this feature.

8 Detailed Comments for Author 5 / 5

✓ **- 0 pts** Correct