On Speculative Optimizations Using Alias/Dependence Profiling and Heuristics For C programs

Jin Lin, Tong Chen, Wei-Chung Hsu, Pen-Chung Yew
Department of Computer Science and Engineering
University of Minnesota

1

Outline

- Motivation
- Background review
  - Framework for speculative optimizations
  - Alias/data dependence profiling
- Further studies
  - Input sensitivity
  - Different approaches to guide speculative optimizations
  - Multi-level speculation v.s single-level speculation

2

Motivation

- Compiler optimizations are usually hampered by imprecise compiler analysis information because complete alias/dependence analysis is very costly and difficult
- Alias/dependence profiling and compiler heuristics can compensate for imprecise alias/dependence information
- Use data speculation to verify such information at run time

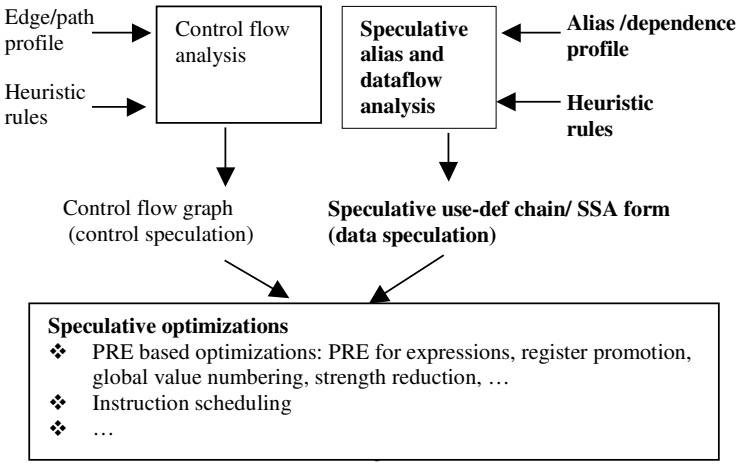
3

Approach

- A general compiler framework to support both control speculation and data speculation
- Extends SSA form/dependence graph to incorporate probabilistic aliasing/dependence information, thus enabling more aggressive compiler optimizations such as PRE
- Use Advance Load Address Table (ALAT) in IA64 to provide the necessary support for data speculation

4

A General Framework of Speculative Analysis and Optimiazations



Data Speculation – Redundancy Elimination

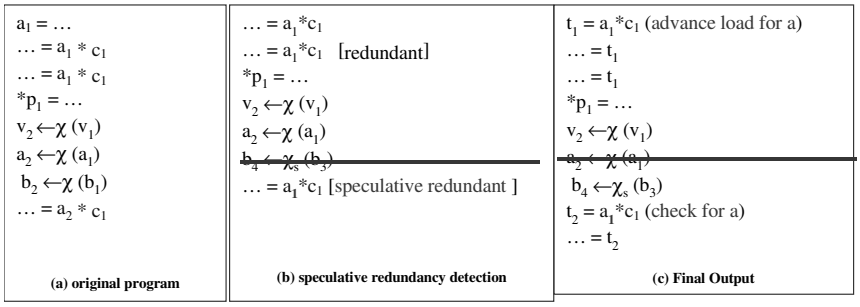
```
… = a[i]*b
*q = ..
… = a[i]*b
```

(a) original program

```
ld.a  r3 = [r8]
mult …
st …
chk.a r3, recovery
```

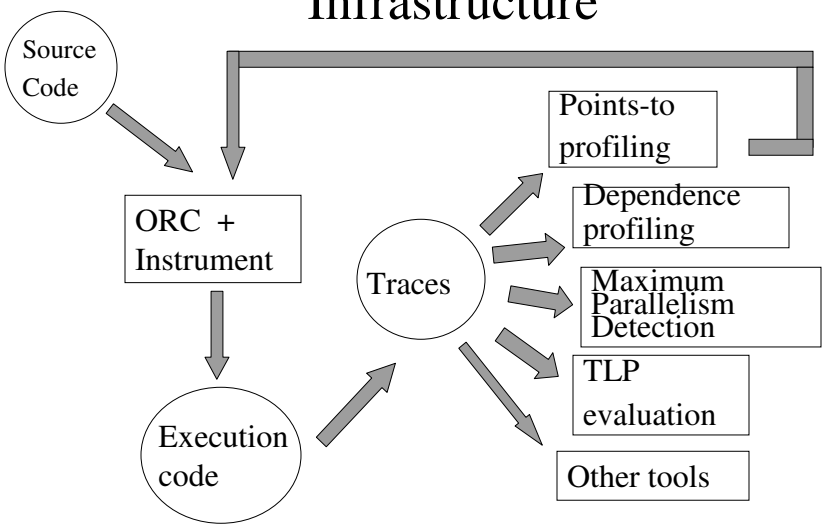
(b) speculative version

Enhanced SSAPRE



7

An Instrumentation-based Profiling Infrastructure



8

Outline of instrumentation based profiling approach

- Use ORC as an instrumentation and profiling tool to study alias, dependence, thread-level parallelism for speculative multithreaded architectures.
- Feed the profiling information back into ORC to guide compiler analyses and optimizations.
- Use ORC to generate code to exploit speculative thread-level parallelism.

9

Input Sensitivity

- Since alias profiling can be sensitive to the program inputs, how sensitive the performance is to the program inputs in this framework?
- How to answer this question?
  - Study the profile collected from different input set
  - Compare the performance impact using different profile

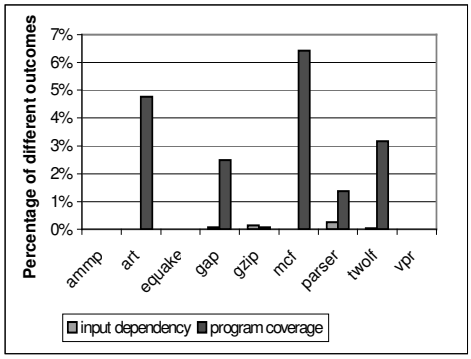
10

Two Effects of Input Sensitivity

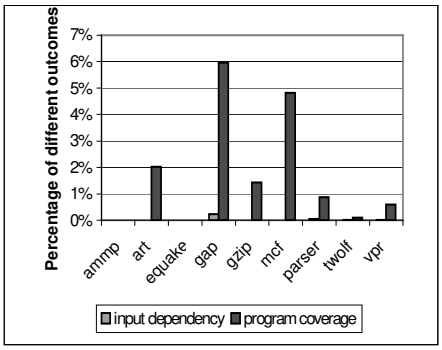
- Input dependency
- Program coverage

11

Difference In Alias Queries Attributed to Input Dependency and Program Coverage



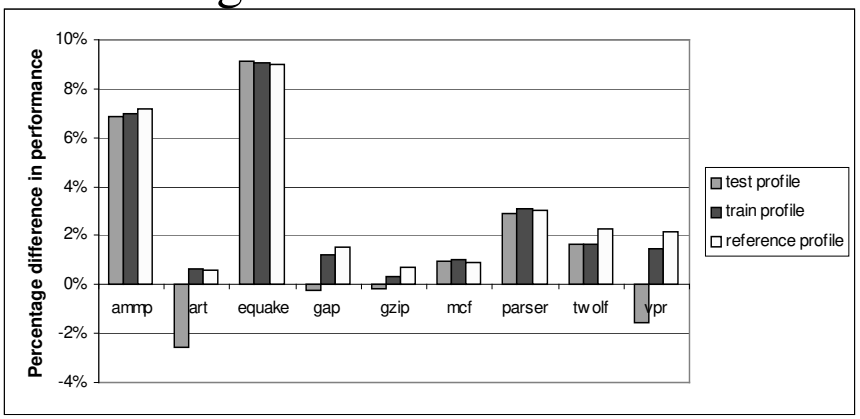
(a) *test* vs. *reference* input set



(b) *train* vs. *reference* input set

12

Performance Improvement from Speculative Register Promotion Using Different Alias Profiles



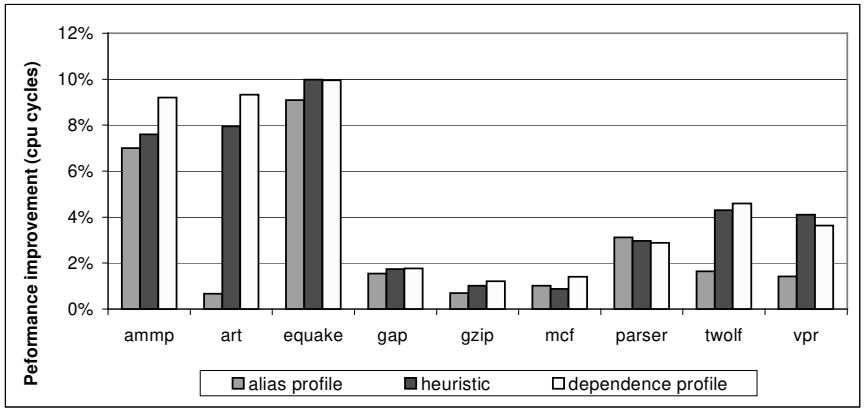
13

Different Approaches to Guide Speculative Optimizations

- Use three different approaches to guide speculative optimizations and study their impact on performance
  - Alias profile
  - Dependence profile
  - Heuristic rules

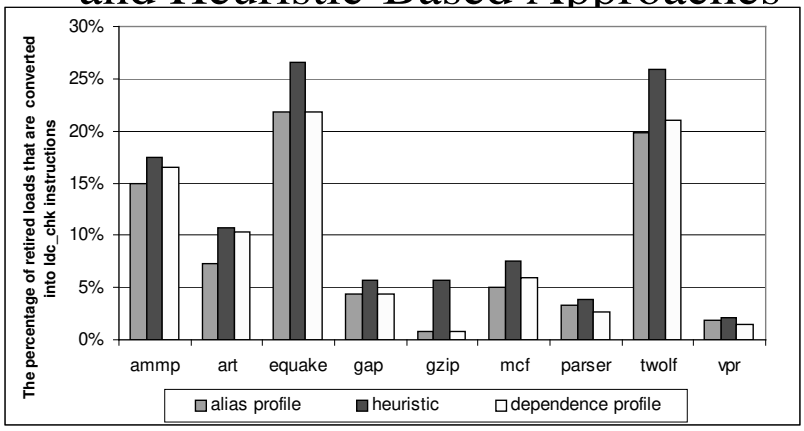
14

Performance Comparison in the CPU Cycles



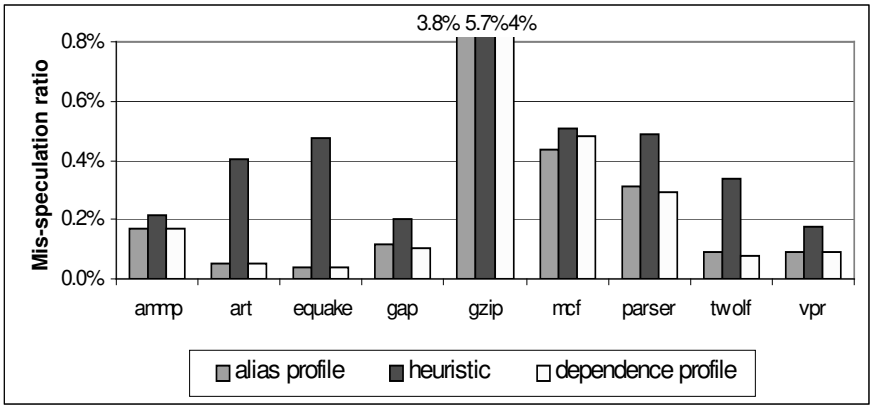
15

Effectiveness of Speculative Register Promotion for Profile-based and Heuristic-Based Approaches



16

Mis-speculation Rates for Profile-based and Heuristic-Based Approaches



17

Multi-level Data Speculation

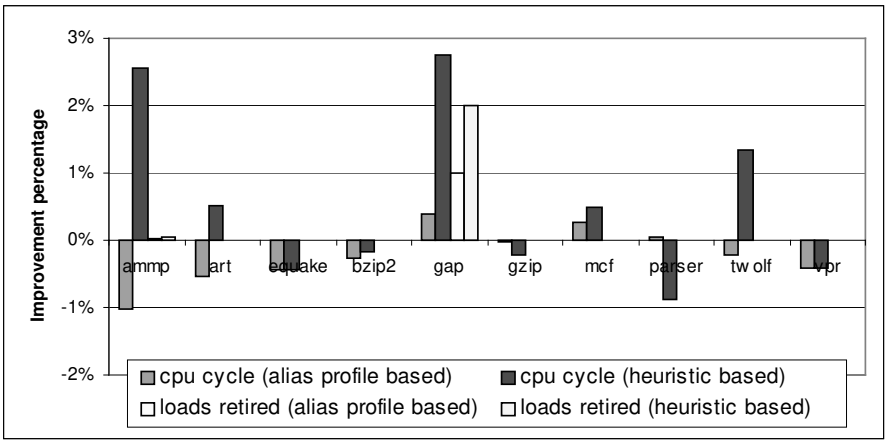
- Expose more redundant opportunties
- Examples

```
 = **p
*q =
 //*q may be aliased with *p
and **p
= **p
```

```
  =  sin(*p)
*q =
//*q may be aliased with *p
  = sin(*p)
```

18

The Performance of Speculative Register Promotion with Multi-Level Data Speculation over Single-Level Data Speculation



19

Conclusions

- Data speculation can further improve program performance for compiler optimizations other than instruction scheduling
- Compiler heuristic rules can be very cost-effective
- Little performance impact on SPEC2000 benchmarks using the three provided input sets: test, train and reference
- Multi-level data speculation may not contribute much to performance improvement if the type-based alias analysis is used.

20

Multi-level Speculation

- An example

… = **p *q = … (may modify *p) … = **p (a) original program	r = *p (ld.a flag) …= *r *q = … r = *p (ld.c flag) … = *r (b) output after one level data speculation	s1: r = *p (ld.a flag) s2: s = *r s3: …= s s4: *q = … s5: r = *p (ld.c flag) s6: s = *r (remove flag) s7: … = s (c) output after two level data speculation	ld.a r = [p] ld s = [r] …. st [q] =… chk.a r, recovery label: … recovery: ld r = [p] ld s = [r] br label (d) output of assembly code

21

Alias/Data Profiling

- Provide the probabilistic information for memory disambiguation
- Instrumentation-based
- Multiple possible places for instrumentation and feedback
- Feedback maintained in compiler

22

Alias Profiling

- Target set for indirect references
  - variables or heap blocks
- Read/write set for function calls
- Additional information about a target
  - Probability: (# occurrences of a target)/(# occurrences of this reference)
  - Field tag for structure
  - Calling context: mimic context-sensitive points-to analysis

23

Data Dependence Profiling

- Data dependence edges among memory references and function calls
- Detail information
  - type: flow, anti, output, or input
  - probability: frequency of occurrence
- When loops are targeted
  - dependence distance: limited

24

Compare Performance

- Profiling with test, train and ref input sets
- Optimizations guided by different profiles
- The optimized code is run with ref input
- Performance difference is rather small

25

Performance Comparision

- pfmon is used to observe the differences
- Detail behavior are reported
  - PerformancCPU cycle
  - Speculative load
  - Mis-speculation rate

26

Support Multiple-level Speculation

- Extension in the renaming step: assign the version number based on the speculation that has been performance
- Extension in the code motion: introduce a new flag to generate recovery code correctly.

27

References

- J. Lin, T. Chen, W.C. Hsu, P.C. Yew, R. D.-C. Ju, T.-F. Ngai, S. Chan, A Compiler Framework for Speculative Analysis and Optimizations, PLDI, 2003.
- J. Lin, T. Chen, W. Hsu, P. Yew. Speculative Register Promotion Using Advanced Load Address Table (ALAT), CGO, 2003.
- T. Chen, J. Lin, W. Hsu, P.C. Yew. An Empirical Study on the Granularity of Pointer Analysis in C Programs, LCPC, 2002.
- T. Chen, J. Lin, X. Dai, W. Hsu, P.C. Yew. *Data Dependence Profiling for Speculative Optimizations*, CC, 2004.

28