

Instruction Based Memory Distance Analysis and Its Application to Optimization

Changpeng Fang, Steve Carr, Soner Onder, Zhenlin Wang

Department of Computer Science
Michigan Technological University
September 19, 2005

Outline

- **Instruction-based Memory Distance Prediction**
- **Critical Instruction Prediction**
- **Memory Disambiguation**

Motivation

- **Memory System Performance**
 - Cache optimization: **data locality**
 - Load speculation: **memory dependence**
- **Static analysis**
 - Limited applicability
 - e.g. pointers, complex control flow
- **Program profiling**
 - Cannot predict behavior change

Memory Distance

➤ Memory distance

A quantifiable distance in terms of memory references between two accesses to the same memory location

• Reuse distance (LRU stack distance)

The number of **distinct** memory references between two consecutive accesses to the same memory location (address, **cache line** , ...)

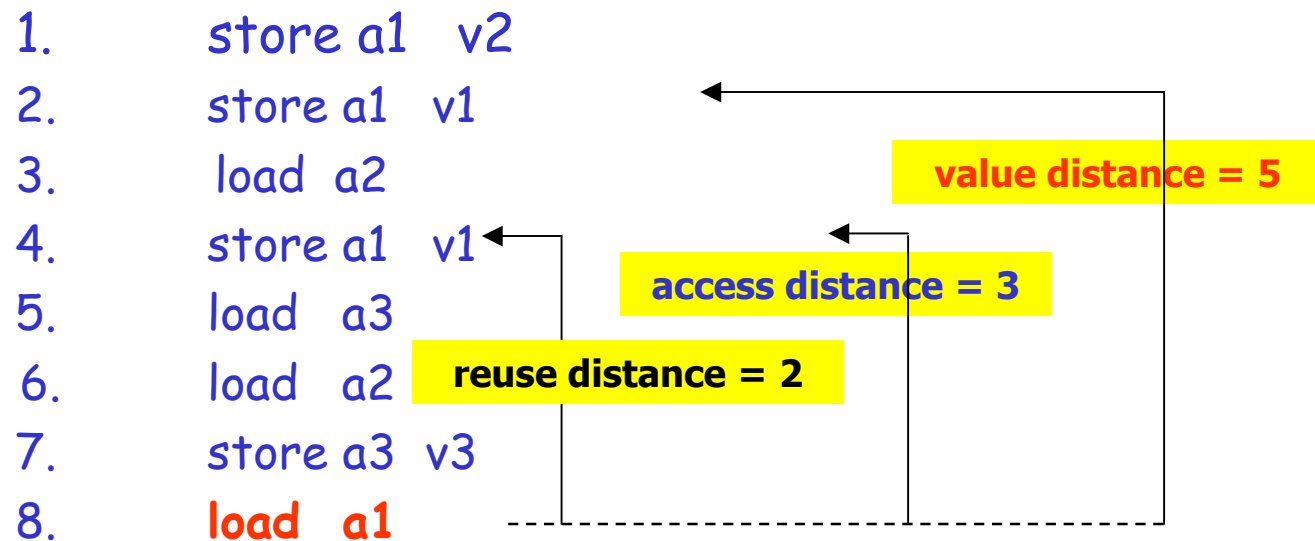
• Access distance

The number of memory references between a store to and a load from the same address

• Value distance

The access distance of a load to the first store in a sequence of stores writing the **same value to the same address**

Memory Distance Example



Research Questions

- Can we predict the instruction based memory distance for both floating point and integer programs?
- Can we use memory distance to identify instructions that will produce most of the cache misses?
- Can we use memory distance for memory dependence prediction?

Outline

- Instruction-based Memory Distance Prediction
- Critical Instruction Prediction
- Memory Disambiguation

Reuse Distance Collection

➤ Intervals

- 0, 1, 2~3, 4~7, ..., 512~1K, 1~2K, 2~3K, ...

➤ Reuse Distance Recording

- Max, min, mean, frequency

➤ Training runs and Target run

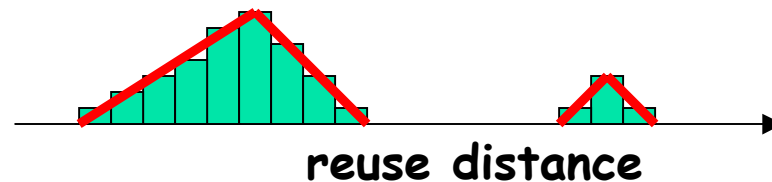
- A few training runs with small inputs
- A target run with large input

Reuse Pattern

➤ Reuse Pattern

- A cluster of references with similar reuse distance
- Models of up to two linear distributions in each pattern
- Details on pattern analysis are in the paper

➤ Example



Reuse Distance Prediction

- **Reuse distance is a function of data size**

for $I = 2, N$

for $J = 2, N$

$$m.get(I).get(J) = m.get(I-1).get(J) + n.get(I).get(J)$$

Data size: $O(N^2)$, Distance of $m.get(I-1).get(J)$: $O(N)$

- **Curve fitting (Ding and Zhong, PLDI'03, PACT'03)**

constant, ..., sqrt, ..., linear

- **Predict pattern**

min, max, mean

Outline

- Instruction-based Memory Distance Prediction
- Critical Instruction Prediction
- Memory Disambiguation

Critical Instructions

➤ Critical Instructions

- The set of high miss instructions whose cumulative misses account for 95% of the **Level 2 cache misses** in a program

➤ Importance

- Target optimizations only to those instructions
- Static analysis may not be sufficient

Prediction Procedure

- Predict reuse distance
- Calculate L2 miss rates
- Calculate the number of L2 misses
- Sort the instructions to find critical ones

Miss Rate Calculation

- Reuse distance predictable
use predicted reuse distance
- Reuse distance not predictable
use one training run's reuse distance
- Instructions not appearing in the training runs
0.0

Evaluation I

- **Tools**
 - ATOM
 - Reuse Distance Tool, Ding and Zhong, PLDI'03
- **Programs**
 - CFP2000: 11 programs
 - CINT2000: 11 programs
- **Coverage**
 - Percent of instructions with predictable patterns
- **Accuracy**
 - Percent of covered instructions predicted correctly

Evaluation II

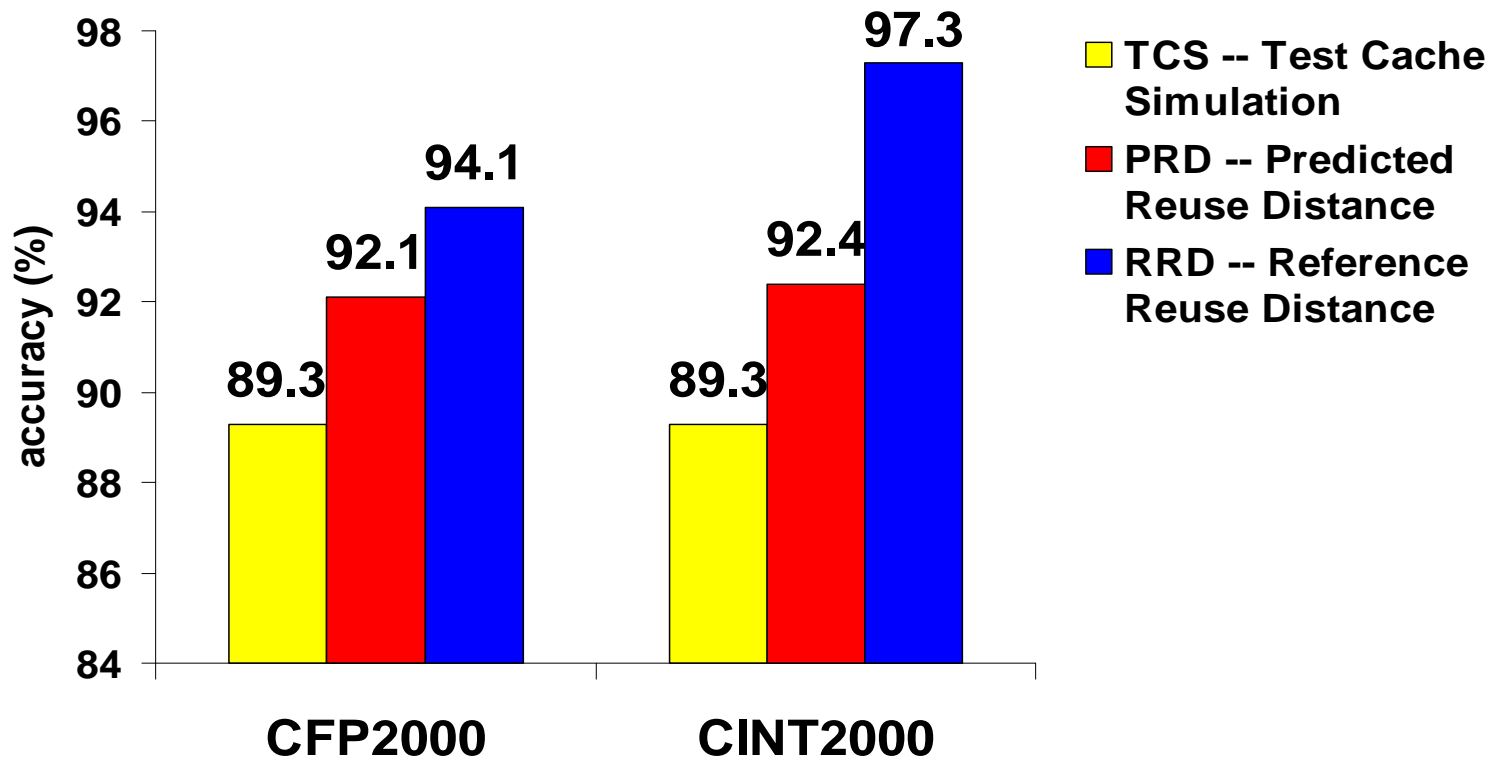
➤ Caches

- L1: 32KB, 2-way set-associative, 64-byte line, LRU
- L2: 1MB, 4-way set-associative, 64-byte line, LRU
- Results for other configurations in the paper

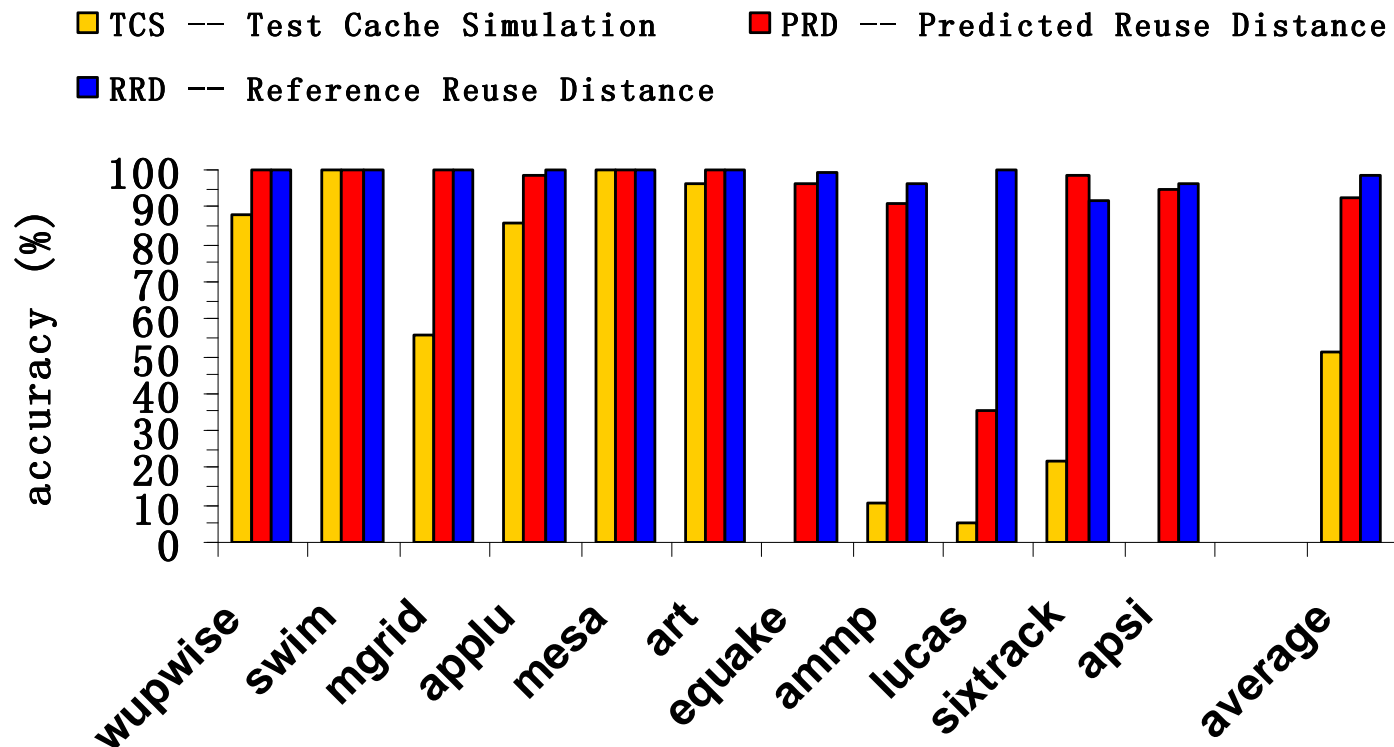
➤ Prediction schemes

- PRD - Predicted Reuse Distance
 - Uses test and train to predict reference
- RRD - Reference Reuse Distance
 - Uses actual reference (unfair, too expensive)
- TCS - Test Cache Simulation
 - Cache simulation on test

L2 Cache Miss Rate Prediction Accuracy

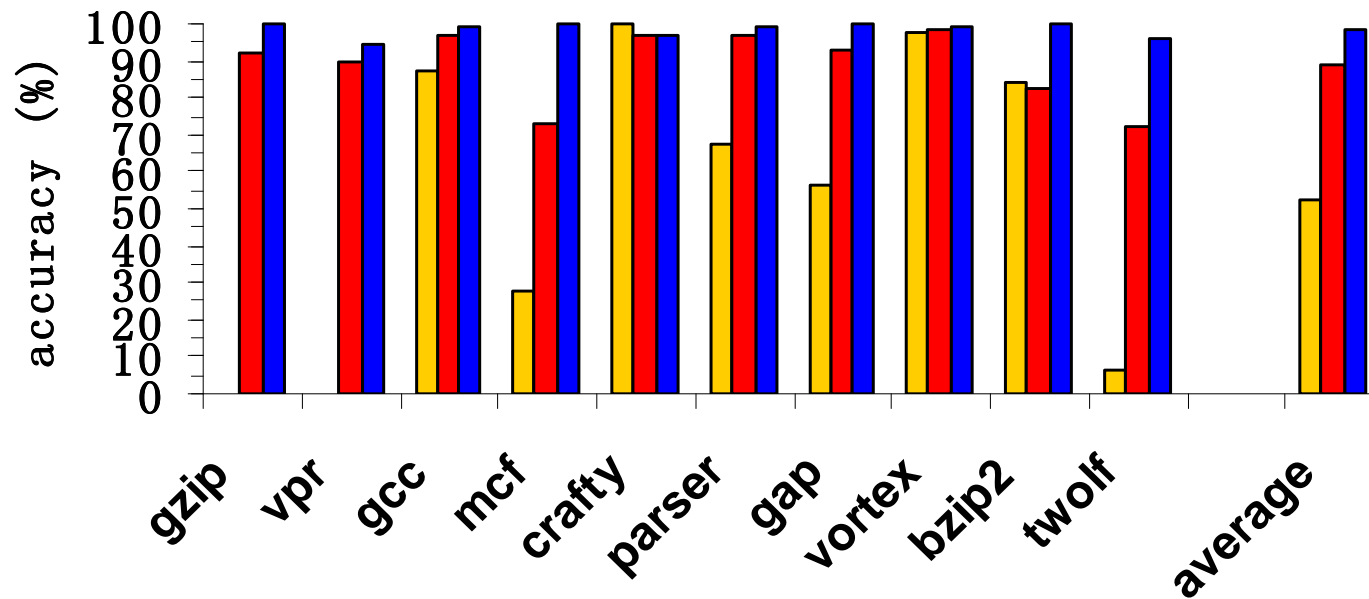


CFP2000 Critical Instruction Prediction Accuracy



CINT2000 Critical Instruction Prediction Accuracy

■ TCS -- Test Cache Simulation ■ PRD -- Predicted Reuse Distance
■ RRD -- Reference Reuse Distance



Reuse Distance Prediction Coverage and Accuracy

Suite	Coverage(%)	Accuracy (%)
CFP2000	93.0	97.6
CINT2000	91.6	93.8

Outline

- Instruction-based Memory Distance Prediction
- Critical Instruction Prediction
- Memory Disambiguation

A Compiler/Architecture Cooperative Approach

- **Compiler's responsibility**
 - Predict access (value) distance
 - Mark loads as **speculative** or **non-speculative**
- **Architecture's responsibility**
 - Schedule a load according to its marker
 - Check and recover from memory order violations
 - **Address based**
 - **Value based**

Speculative and Non-speculative

- **Load Marking**
 - Distance sufficiently large : speculative
 - Empirical threshold: 10
- **Distance changes with inputs**
 - Use two profiling runs with small inputs to predict distance for a third input
- **Multiple distances**
 - Mark a load as **speculative** if 95% of its reference instances with distance greater than or equal to 10

Experimental Setup

➤ Schemes

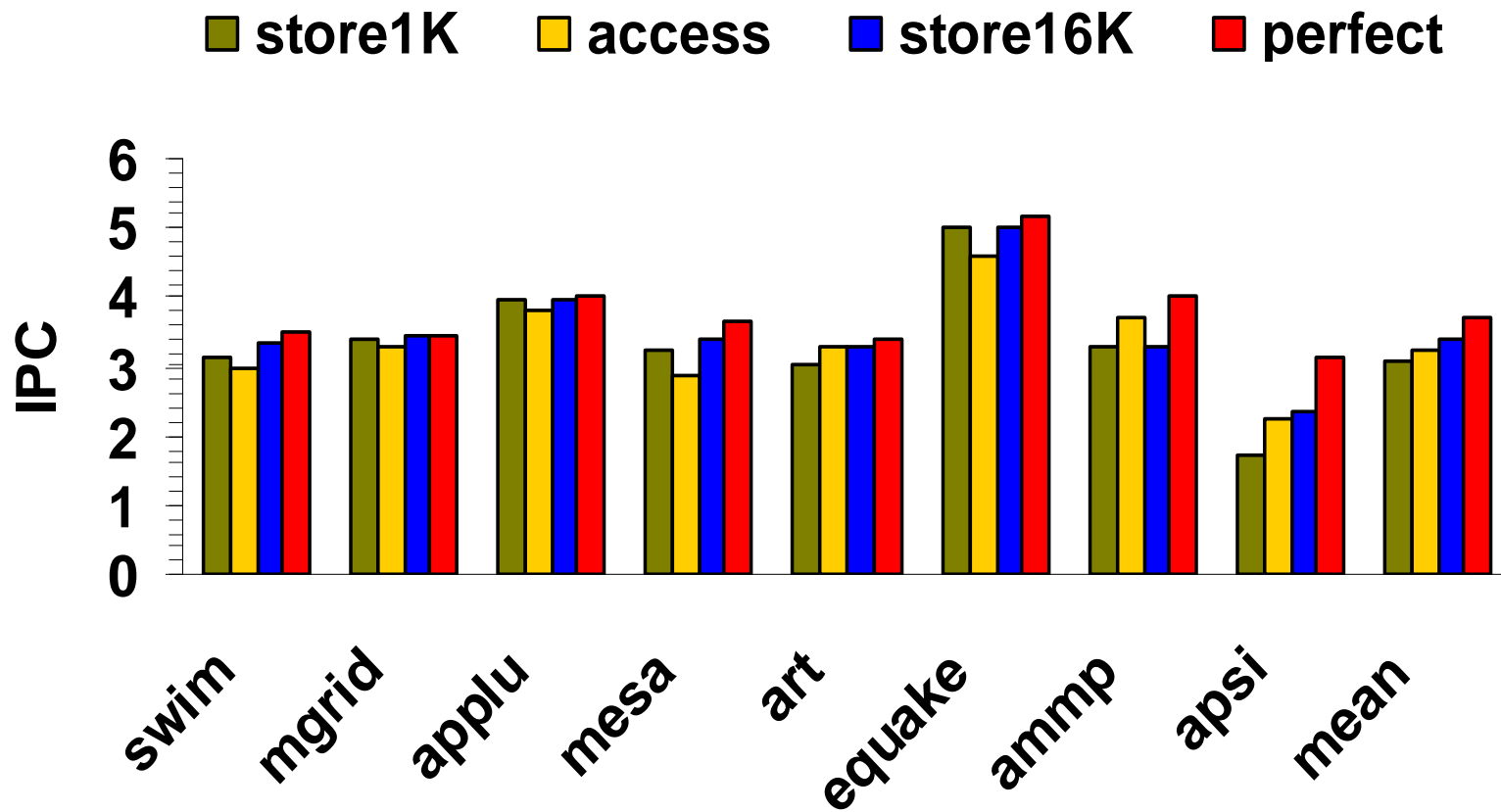
- **access** - access distance based
- **value** - value distance based
- **perfect** - perfect memory disambiguation
- **store set**
 - **store1K** - store set with a 1KB predictor
 - **store16K** - store set with a 16KB predictor

➤ Simulator

- **FAST** - a cycle accurate simulator
- 8 issue; 2 memory ports; perfect cache

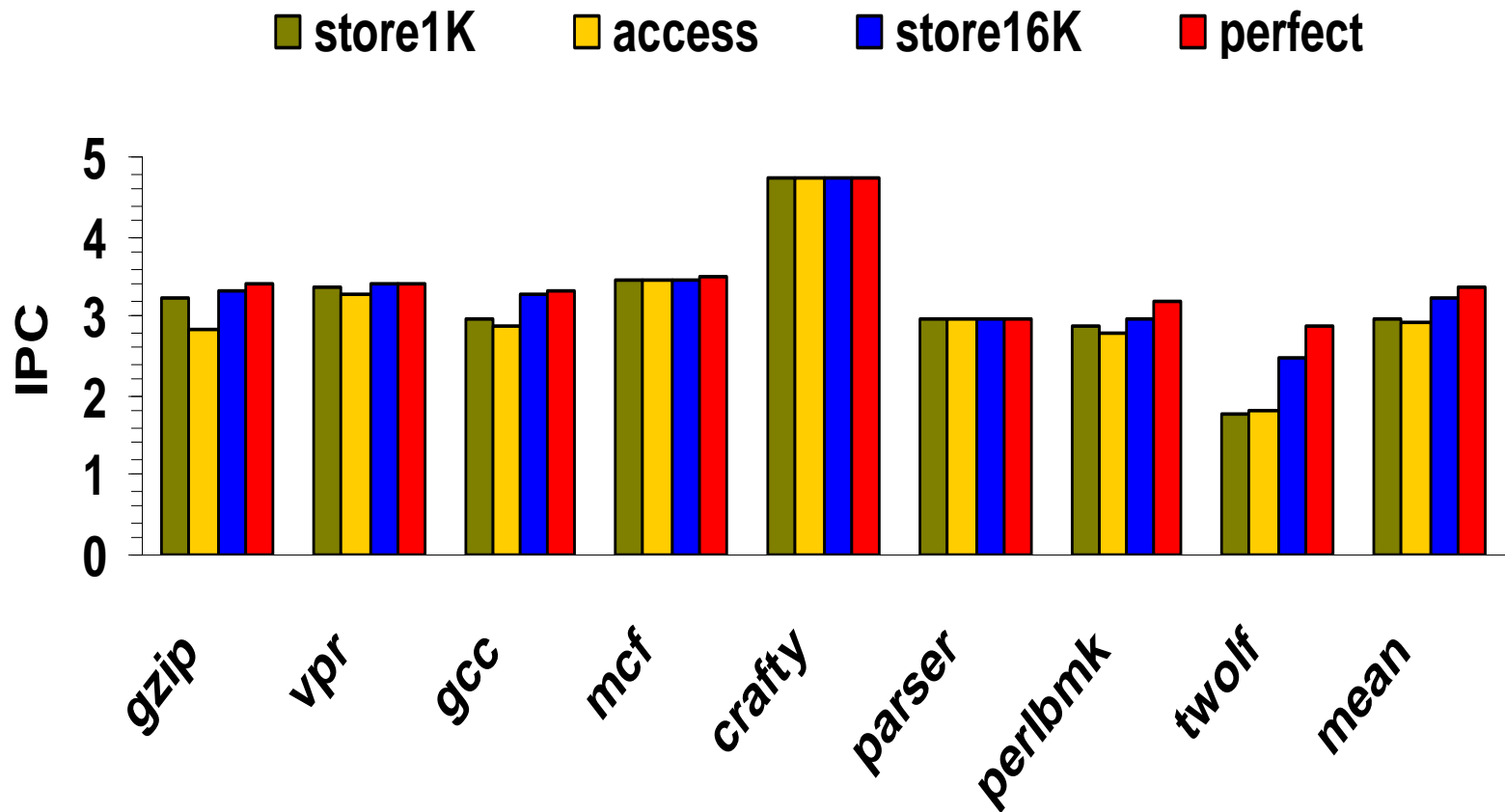
CFP2000 IPC

Address Based Exception Checking



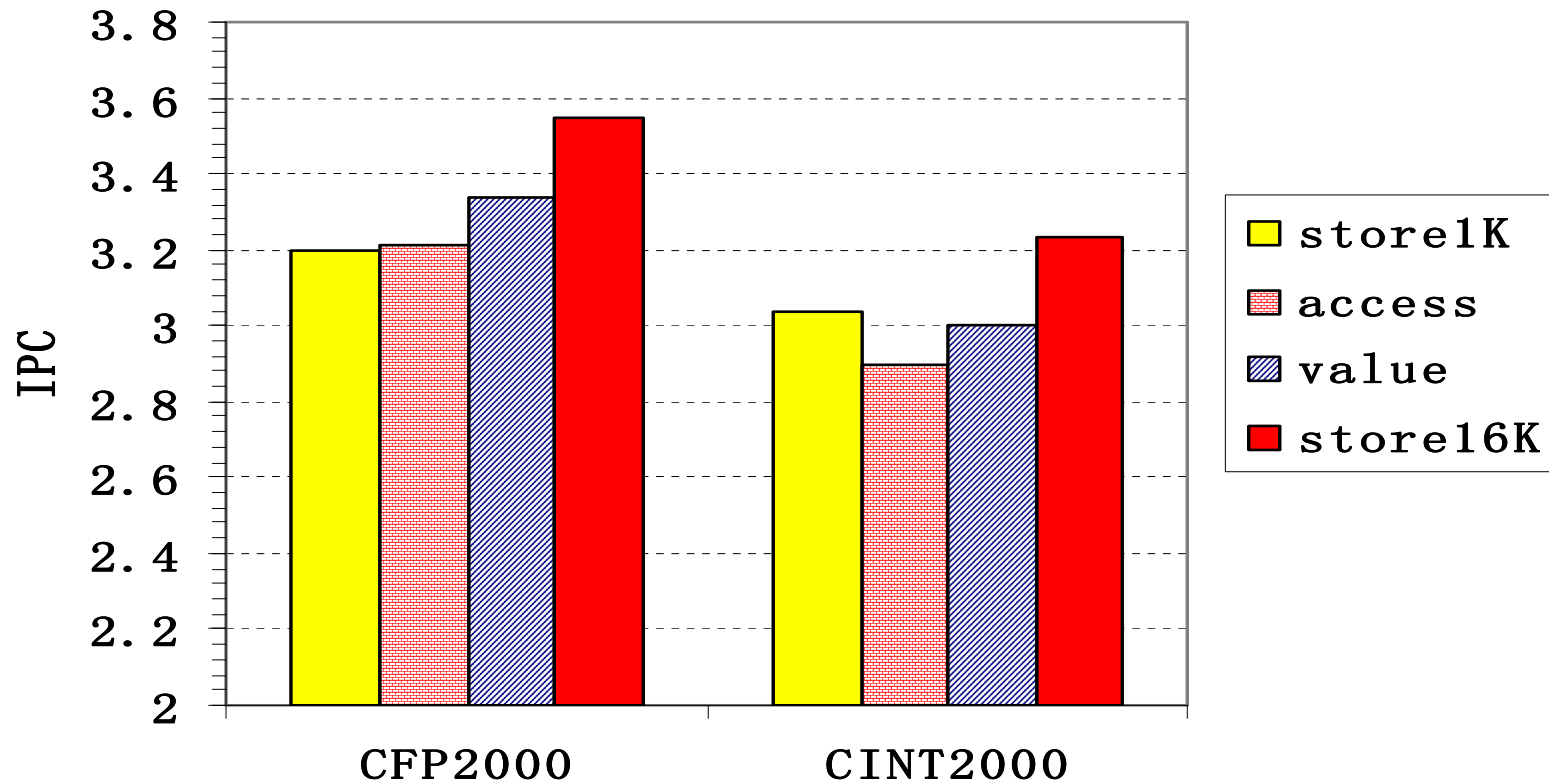
CINT2000 IPC

address based exception checking



SPEC CPU2000 IPC

Value based exception checking



Previous Work

- **Reuse distance prediction**
 - Marin and Mellor-Crummey, MMCS'04
 - Fang, et al. MSP'04
 - Ding and Zhong, et al., PLDI'03, PACT'03
- **Reuse distance profiling**
 - Beyls and D'Hollander, Euro-Par'02
 - Mattson, et al. IBM Systems Journal' 70
 - Many others
- **Static analysis**
 - Cascaval and Padua, ICS'03

Conclusions

- Instruction based memory distance is highly predictable.
- Reuse distance analysis can be used to accurately predict cache misses and identifying critical instructions.
- Memory distance can be used to guide load speculation. Good results have been achieved with limited hardware resource requirement

Future Work

- Critical instruction prefetching
- Cache replacement
- Application of memory distance analysis for instruction scheduling
- Other optimizations