Locality Phase Prediction

Xipeng Shen
Yutao Zhong    Chen Ding
Department of Computer Science
University of Rochester
# A Lesson from Nature

<table>
<thead>
<tr>
<th>Spring</th>
<th>Fall</th>
<th>Spring</th>
<th>Fall</th>
</tr>
</thead>
</table>

- People adapt, for example, by changing the wardrobe before winter
- Need to predict when winter comes
Motivation

- Memory performance
  - largely determines cost and energy
- Adaptation is increasingly used
  - dynamic data and computation reordering
  - dynamic cache resizing
  - dynamic memory remapping
- Need to predict locality phases
  - manual phase marking in past
Locality Phase

- A period of a program execution that has stable or slowly changing data locality inside but disruptive transition periods between phases
  - [Batson & Madison, SIGMetrics, 1976]
- We want phases with repetitive behavior
  - not necessarily uniform behavior
Locality Phase Examples

- **Unstructured mesh**
  - the computation sweeps through the mesh structure at each time step
    - e.g. the aging of an airplane
  - input-dependent memory behavior
  - repeat or slow-changing across time steps

- **Other scientific & commercial applications**
  - structural, mechanical, and molecular simulations
Outline

- Locality-based phase detection
  - to find repetitive memory behavior
- Program phase marking
  - to mark phases for all inputs
- Run-time prediction
  - to predict machine-dependent behavior
- Evaluation
- Summary
Locality Metric: Reuse Distance

- Defined on each element of a memory-access trace
- It is the number of distinct elements accessed between this and the previous access to the same element

Time: 1 2 3 4 5 6 7 8 9 10 ...
Accesses: b a b b c c b c a c ...
Locality Metric: Reuse Distance

- Defined on each element of a memory-access trace
- It is the number of distinct elements accessed between this and the previous access to the same element

Time: 1 2 3 4 5 6 7 8 9 10 ...
Accesses: b a b b c c b c a c ...
Locality Metric: Reuse Distance

- Defined on each element of a memory-access trace
- It is the number of distinct elements accessed between this and the previous access to the same element

Time: 1 2 3 4 5 6 7 8 9 10 ...
Accesses: b a b b c c b c a c ...
Reuse Distance as a Signal

- Defined on each element of a memory-access trace
- It is the number of distinct elements accessed between this and the previous access to the same element

Time: 1 2 3 4 5 6 7 8 9 10 ...
Accesses: b a b b c c b c a c ...
Reuse distance: ∞ ∞ 1 0 ∞ 0 1 1 2 1 ...
Reuse Distance Signal

One execution of Tomcatv from Spec95

![Graph showing reuse distance over time](chart.png)
Reuse Distance Signal

One execution of Tomcat v from Spec95

Reuse distance

Time

x $10^6$

x $10^7$
Basic Problem & Solution

- A search problem: to find a handful of phase changing points from billions of points

- Solution: to apply signal processing techniques to find disruptive behavior
Variable-length Sampling

- Sample only long reuse distances of representative memory locations
  - for example, 30 thousand samples out of billions of memory accesses
Wavelet Analysis

- The wavelet transform gives temporal-frequency information.
- FFT, in comparison, gives only frequency information.
Wavelet Analysis

- The wavelet transform gives temporal-frequency information.
- FFT, in comparison, gives only frequency information.
Locality Phase Detection

Program binary
Locality Phase Detection

Program binary $\rightarrow$ variable-length sampling
Locality Phase Detection

Program binary

variable-length sampling

Each sampled variable yields a signal
Locality Phase Detection

Program binary -> variable-length sampling

Each sampled variable yields a signal
Locality Phase Detection

Program binary → variable-length sampling

Each sampled variable yields a signal

Wavelet

Find disruptions on each signal as possible phase boundaries
Locality Phase Detection

Program binary sampling

variable-length sampling

Each sampled variable yields a signal

Find disruptions on each signal as possible phase boundaries
Locality Phase Detection

Program binary → variable-length sampling

Each sampled variable yields a signal

Find disruptions on each signal as possible phase boundaries

Candidate phase boundaries → time → combine

wavelet
Locality Phase Detection

Each sampled variable-length sampling yields a signal.

Find disruptions on each signal as possible phase boundaries.

Program binary optimal phase partition (described in paper) combine time.
Locality Phase Detection

Program binary → variable-length sampling

Each sampled variable yields a signal

Find disruptions on each signal as possible phase boundaries

optimal phase partition (described in paper)

Candidate phase boundaries

Actual phase boundaries

combine

time

describe

wavelet

Each sampled variable yields a signal

Find disruptions on each signal as possible phase boundaries
Phase Hierarchy

One execution of Tomcatv from Spec95
Phase Hierarchy

One execution of Tomcatv from Spec95

Reuse distance

Time

\( x \times 10^6 \)

\( x \times 10^7 \)
Reuse Distance  Vs.  Cache Miss Rate

- hardware independent
- defined on each point, no need for windows
- a distance at each access

- hardware dependent
- interval-based, defined on windows
- hit or miss at each access
Outline

- Locality-based phase detection
  - find repetitive memory behavior
- Program phase marking
  - mark phases for all inputs
- Run-time prediction
  - predict machine-dependent behavior
- Evaluation
- Summary
Phase Marking

- **Objective:** to find a basic block that uniquely marks each phase boundary.
Run-Time Prediction

- When an instrumented program runs, we measure the behavior of the first few instances of a phase and use them as prediction for the later instances.
- We assume instances of a phase behave the same in one execution.
Outline

- Locality-based phase detection
- Program phase marking
- Run-time prediction
- Evaluation
  - Phase behavior consistency
  - Prediction accuracy
  - Example uses
- Summary
Cache Miss Rates of Tomcatv
(5250 instances of 7 phases)
Cache Miss Rates of Tomcatv
(5250 instances of 7 phases)

750 instances of Phase 2, whose length are about 2.5M instructions
TOMCATV Miss Rate Distribution (2493 Intervals)
TOMCATV Miss Rate Distribution (2493 Intervals)

2.97% execution inside this BBV phase bounding box
Prediction Accuracy

- Example benchmark: Mesh
  - dynamic mesh structure simulation
  - 5 billion run-time instructions
  - 4691 instances of 31 phases
Run-time Prediction of Mesh
Run-time Prediction of Mesh

First time entering Phase 300. Monitoring. No prediction.
Run-time Prediction of *Mesh*

First time entering Phase 300. Monitoring. No prediction.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
Run-time Prediction of Mesh

First time entering Phase 300. Monitoring. No prediction. Leaving Phase 300. Measured inst. =2918191 cmr=0.276. First time entering Phase 308. Monitoring. No prediction.
Run-time Prediction of Mesh

First time entering Phase 300. Monitoring. No prediction.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
First time entering Phase 308. Monitoring. No prediction.
Leaving Phase 308. Measured inst. = 273589 cmr = 0.125.
Run-time Prediction of Mesh

First time entering Phase 300. Monitoring. No prediction.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.

First time entering Phase 308. Monitoring. No prediction.
Leaving Phase 308. Measured inst. = 273589 cmr = 0.125.

Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Run-time Prediction of Mesh

First time entering Phase 300. Monitoring. No prediction.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
First time entering Phase 308. Monitoring. No prediction.
Leaving Phase 308. Measured inst. = 273589 cmr = 0.125.
Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
Run-time Prediction of Mesh

First time entering Phase 300. Monitoring. No prediction.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
First time entering Phase 308. Monitoring. No prediction.
Leaving Phase 308. Measured inst. = 273589 cmr = 0.125.
Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
Entering Phase 308. Pred. num. inst. = 273589 cmr = 0.125.
Run-time Prediction of Mesh

First time entering Phase 300. Monitoring. No prediction.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.

First time entering Phase 308. Monitoring. No prediction.
Leaving Phase 308. Measured inst. = 273589 cmr = 0.125.

Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.

Entering Phase 308. Pred. num. inst. = 273589 cmr = 0.125.
Leaving Phase 308. Measured inst. = 297334 cmr = 0.125.
Run-time Prediction of *Mesh*

First time entering Phase 300. Monitoring. No prediction.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
First time entering Phase 308. Monitoring. No prediction.
Leaving Phase 308. Measured inst. = 273589 cmr = 0.125.
Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
Entering Phase 308. Pred. num. inst. = 273589 cmr = 0.125.
Leaving Phase 308. Measured inst. = 297334 cmr = 0.125.
Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Run-time Prediction of Mesh

First time entering Phase 300. Monitoring. No prediction.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.

First time entering Phase 308. Monitoring. No prediction.
Leaving Phase 308. Measured inst. = 273589 cmr = 0.125.

Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
Entering Phase 308. Pred. num. inst. = 273589 cmr = 0.125.
Leaving Phase 308. Measured inst. = 297334 cmr = 0.125.
Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
Run-time Prediction of Mesh

First time entering Phase 300. Monitoring. No prediction.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.

First time entering Phase 308. Monitoring. No prediction.
Leaving Phase 308. Measured inst. = 273589 cmr = 0.125.

Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.

Entering Phase 308. Pred. num. inst. = 273589 cmr = 0.125.
Leaving Phase 308. Measured inst. = 297334 cmr = 0.125.

Entering Phase 300. Pred. num. inst. = 2918191 cmr = 0.276.
Leaving Phase 300. Measured inst. = 2918191 cmr = 0.276.
Prediction Accuracy

- **Relaxed prediction**
  - to predict all phases
  - we achieve 86% accuracy, and the predicted code is over 98% of the executions.

- **Strict prediction**
  - to only predict the stable phases, selected from the training run.
  - we achieve over 99% accuracy, and the predicted code is 73% of the executions.
Uses of Phase Prediction

- **Reconfigurable hardware**
  - Adaptive cache resizing [Albonesi et al., 2000]
    - reducing 40% cache size without increasing misses
      (interval-based methods reduce 6%)

- **Dynamic optimizations**
  - Phase-based dynamic memory remapping
    - assuming Impulse memory controller [Carter et al., 2001]
      - 36% speedup than the original program, compared with 13% using the global data reorganization
Other Approaches

- **Program code**
  - input independent and proactive
  - most limited to loops, regions, and subroutines

- **Runtime measurement**
  - exact
  - fixed-length intervals may not match phase length

- **Locality**
  - determines the memory behavior
  - difficult to analyze for complex programs
Conclusions

- Predicting hierarchical memory phases
  - combined locality, program, and run-time analysis
- Effective for programs with consistent phase behavior
- Accurate run-time prediction with low overhead
- Reducing 40% cache size without increasing misses
- Up to 35% speedup when used for memory remapping
The End

Thanks!