A Hierarchical Model of Reference Affinity

Yutao Zhong

Xipeng Shen Chen Ding

Computer Science Department

University of Rochester

Motivation

- Cache block utilization
- Data reorganization
- Reference Affinity

  a group of data that are often/always used together

Outline

- Motivation
- Model definition
- Data clustering
- Experiment results
- Summary

Basic Definition & Notation

- Address trace

T:	1	2	3	4	5	6	7	…
	w	**x**	**w**	**x**	**u**	**y**	**z**	…
	a_w	a_x	a_w'	a_x'	a_u	a_y	a_z	…

$T[a_u] = 5$

$dist(a_x , a_u) = 2$

- Volume distance
  - $T[a_x] \leq T[a_y]$ => **dist(a_x , a_y)** : the No. of distinct data elements accessed in time range $[T[a_x], T[a_y]-1]$

Definition

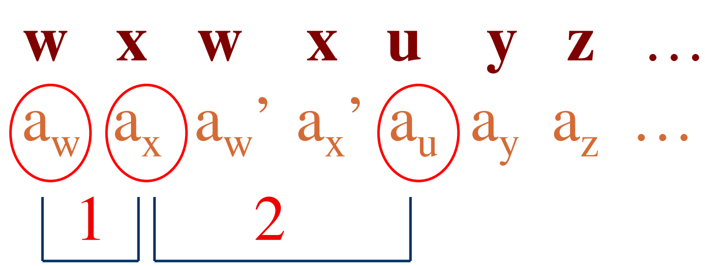
- Linked path & link length

A **linked path from a_x to a_y** with **link length d** exists *iff.*

$\exists$ k accesses $a_{x1}, a_{x2}, a_{x3}, \ldots, a_{xk}$ *s.t.*

1) $dist(a_x, a_{x1}) \le d \wedge dist(a_{x1}, a_{x2}) \le d \wedge \cdots\ dist(a_{xk}, a_y) \le d$

2) $x, x_1, x_2, \ldots, x_k, y$ are all different data elements



A linked path from a_w to a_u with link length 2 ?

Example

w x w x u y z ... z y z y v w x w x ...

- Linked path with d = 2 between all members of data set G = {*w*, *x*, *y*, *z*}
- The same dose not apply to *u* and *v*

Definition - cont'd

- Reference Affinity

Given an address trace, a set of data elements G is a strict **affinity group** with link length d *iff*.

1. for $\forall x \in G$, all its accesses a_x must have a linked path with link length d from a_x to some a_y for any other member $y \in G$ and for all data elements x_i along the linked path, $x_i \in G$

2. adding any other element to G will make Condition (1) impossible to hold

Properties

- Consistency
  - A unique partition of program data
- Hierarchical structure
  - shorter link length => finer partition
- Bounded volume distance

Example

w x w x u y z ... z y z y v w x w x ...

- d = 2, affinity group {w, x, y, z}
- d = 1, affinity groups {w, x} and {y, z}

Identify Affinity Groups

- Detailed analysis
  - Too expensive

w x w x u y z …

a_w a_x a_w' a_x' a_u a_y a_z …

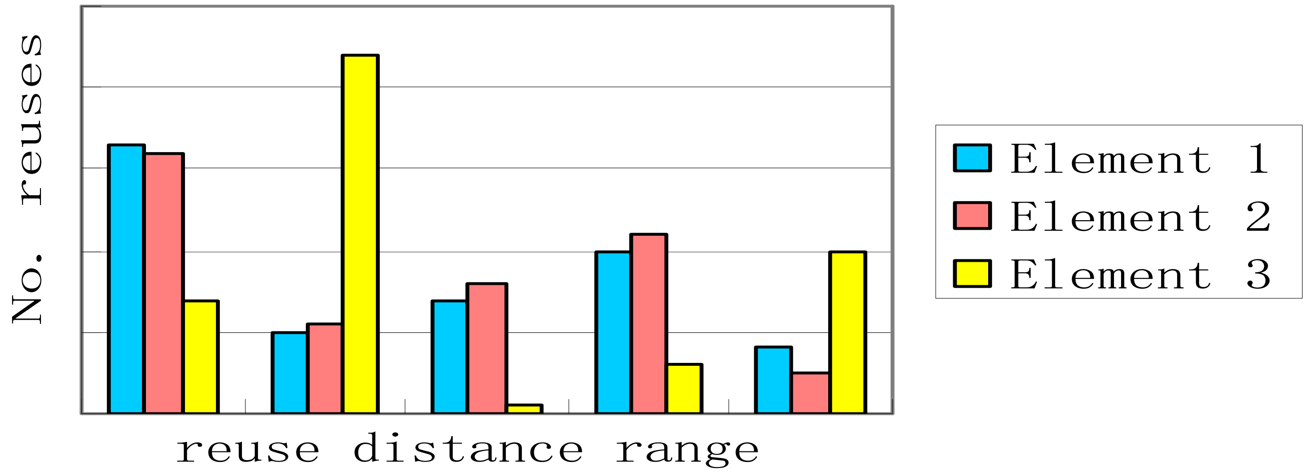
2

- Reuse distance and reuse signature
- Necessary condition:

  Data elements within the same affinity group have similar reuse signatures on long reuse distances

Clustering Analysis

- Estimate affinity groups according to reuse distance distributions of data elements



- X-means and k-means clustering

Evaluation

- Swim
  - Approximation for shallow water equation
  - array grouping
- Cheetah
  - Fully associative cache simulator using splay tree
  - structure splitting
- Methodology
  - Collect reuse signature through profiling
  - Use a different input in evaluation

Swim: clustering

clustering data	clustering method	No. of clusters	clustering
No. of reuses	K-means	8	(cu h)(cv z)(p u v) (pnew unew)(pold uold) (psi)(vnew)(vold)
	X-means	8	(cu cv h z)(p u v)(pnew)(pold uold) (psi)(unew)(vnew)(vold)
Avg. reuse distance	K-means	7	(cu cv h z)(p)(pnew unew vnew)(pold uold vold) (psi)(u)(v)
	X-means	8	(cu cv h z)(p)(pnew unew vnew)(pold uold)(psi)(u)(v)(vold)
Static analysis			(cu h)(cv z)(p) (pnew unew vnew)(pold uold vold) (psi)(u)(v)

[Ding & Kennedy, LCPC, 1999]

Swim: performance

Version	Intel PentiumIV	MIPS R10K	UltraSparc II
Orig	72.7s	156.7s	268.1s
Static	57.5s	153.3s	239.6s
K-means-8-N	59.8s	153.9s	232.5s
K-means-7-A	50.1s	156.0s	226.7s
X-means-8-N	54.1s	155.3s	221.4s
X-means-8-A	50.1s	145.8s	226.7s
Best improvement over orig (%)	31.1	6.96	22.5
Best improvement over static (%)	12.9	6.12	13.2

Cheetah: splitting

Splitting	Version
(addr inum grpno lft prty rt rtwt)	orig
(addr) (inum) (grpno) (lft) (prty) (rt) (rtwt)	naive
(addr) (inum) (grpno) (lft rtwt) (prty) (rt)	v1
(addr) (inum) (grpno) (lft) (prty) (rt rtwt)	v2
(addr) (inum lft rt rtwt) (grpno) (prty)	v3

Cheetah: performance

Version	Intel PentiumIV	MIPS R10K	UltraSparc II
orig	23.4s	59.7s	93.7s
naive	23.7s	71.9s	91.7s
v1	20.7s	59.2s	91.2s
v2	20.1s	58.2s	92.5s
v3	19.3s	56.0s	89.1s
Best improvement over orig (%)	17.5	6.20	4.93
Best improvement over naive (%)	18.6	22.1	2.84

Related Work

- Data reorganization

  [Knuth'71] [Cocke&Kennedy'74] [Sarkar'89] [Seidel&Zorn'98] [Chilimbi+'99] [Petrank&Rawitz'02]

- Computation reordering

  [Abu-sufah+'81] [Gannon+'88] [Wolf&Lam'91] [Ferrante+'91] [Cierniak&Li'95] [McKinley+'96] [Kennedy&Kremer'98]

- Compiler analysis

  [Eggers&Jeremiassen'95] [Ding&Kennedy'99] [Ding&Kennedy'01]

- Dynamic data packing

  [Das+'92] [Al-furaih&Ranka'98] [Ding&Kennedy'99] [Michell+'99] [Han&Tseng'00] [Mellor-Crummey+'01]

Summary

- A model of reference affinity
  - Consistent groups
  - Hierarchical organization
  - Bounded reference distances
- Preliminary results
  - Up to 31% performance improvement
- On-going work

Swim: grouping

Grouping	Version
(cu h)(cv z)(p u v) (pnew unew)(pold uold) (psi)(vnew)(vold)	K-means-8-N
(cu cv h z)(p)(pnew unew vnew)(pold uold vold) (psi)(u)(v)	K-means-7
(cu cv h z)(p u v)(pnew)(pold uold) (psi)(unew)(vnew)(vold)	X-means-8
(cu cv h z)(p)(pnew unew vnew)(pold uold)(psi)(u)(v)(vold)	X-means-8
(cu h)(cv z)(p) (pnew unew vnew)(pold uold vold) (psi)(u)(v)	Static*

* Ding & Kennedy, LCPC 1999

Related Work

- Pair-wise frequency
  - Thabit [Phd Thesis, 1981]
- Group locality
  - Ding & Kennedy[LCPC, 1999]
- Frequent subsequence
  - Chilimbi[PLDI, 2001]

Motivation

- Cache and memory utilization
  - Determined by program data layout and access pattern
- Improvements
  - Computation reordering
  - Data reorganization
    - **Reference affinity model**

Example

Accesses to 1000 other elements

w x w x u y z … z y z y v w x w x …

Datum	Reuse distance		Datum	Reuse distance
w	2 1006 2		x	2 1006 2
y	1002 2		z	1001 2
u	N/A		v	N/A

- d = 2, affinity group {w, x, y, z}
- d = 1, affinity groups {w, x} and {y,z}

Motivation

- Problem: program data placement and access pattern may not match
- Solutions:
  - Computation reordering
  - Data reorganization
    - **Reference affinity model**

Basic Definition & Notation

- Address trace

T:	1	2	3	4	5	6	7	…
	w	**x**	**w**	**x**	**u**	**y**	**z**	…
	a_w	a_x	a_w’	a_x’	a_u	a_y	a_z	…

$T[a_u] = 5$

$dist(a_x , a_u) = 2$

- Volume distance
  - $T[a_x] <= T[a_y]$ => **dist(a_x , a_y)** : the No. of distinct data elements accessed in time range $[T[a_x], T[a_y]-1]$
  - $T[a_x] > T[a_y]$ => **dist(a_x , a_y)** = dist(a_y , a_x)

Future Work

- Affinity group identification
- Partial reference affinity
- Program optimization based on affinity groups

Reference Affinity Models

- Frequency-based models
  - Thabit [PhD Thesis, 1986]
  - Chilimbi [PLDI, 2001]
- Distance-based model
  - Consistency
  - Hierarchical structure

Properties

- Consistency

  Theorem 1. Given an address trace and a link length d, the affinity groups form a unique partition of program data.

- Hierarchical structure

  Theorem 2. Given an address trace and two link length d and d' (d<d'), the affinity groups at d form a finer partition of affinity groups at d'.