Scalable Cross-Architecture Predictions of Memory Latency for Scientific Applications

Gabriel Marin
mgabi@cs.rice.edu

John Mellor-Crummey
johnmc@cs.rice.edu

Introduction

The gap between processor and memory speeds grows with each new generation of microprocessors, making memory hierarchy response a critical factor limiting performance. This poster presents a method for characterizing the data access pattern of an application in a machine independent fashion. We collect memory reuse distance histograms for each memory reference in a program during repeated executions with small data inputs. Then, we model the structure and scaling of each reference's reuse distance histogram as a function of problem size. This approach enables us to predict the number of compulsory and capacity cache misses for architectures and problem sizes that we did not measure. In conjunction with our reuse distance models, we use a probabilistic model to estimate the number of conflict misses for set-associative caches.

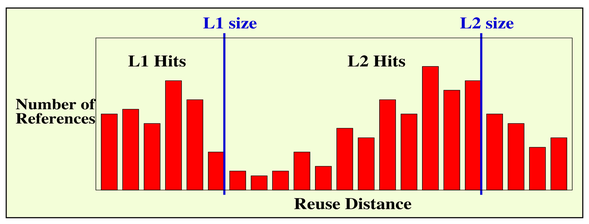
Dynamic Analysis

Memory reuse distance: number of distinct memory blocks accessed by a program between two accesses to the same memory block.

time →

reference	l_1	l_2	l_3	l_1	l_2	l_3
Memory block	A	B	C	B	B	A
MRD	∞	∞	∞	1	0	2

Sample histograms:
l_1: 1 cold miss, 1 distance 1
l_2: 1 cold miss, 1 distance 0
l_3: 1 cold miss, 1 distance 2



Reuse distance data translates directly into cache miss predictions for fully-associative caches:

- cache hit, if reuse distance is less than the cache size
- cache miss otherwise

Data Modeling

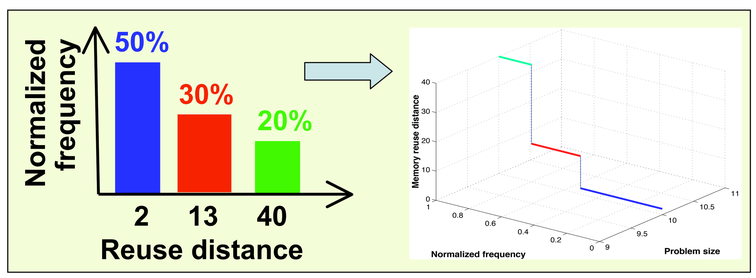
Model the structure and scaling of MRD histograms as a function of problem size.

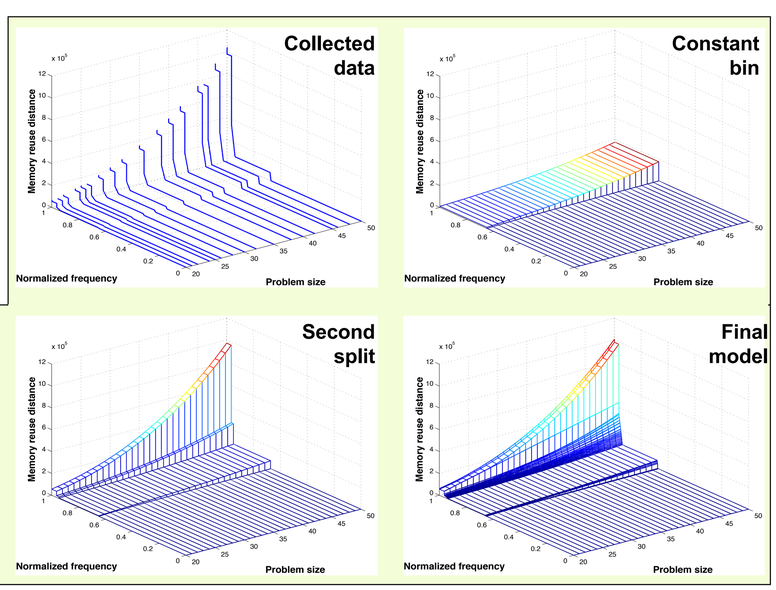
Translate each histogram into 3D Cartesian coordinates:

- *x axis*: problem size
- *y axis*: bins' normalized execution frequency
- *z axis*: reuse distance

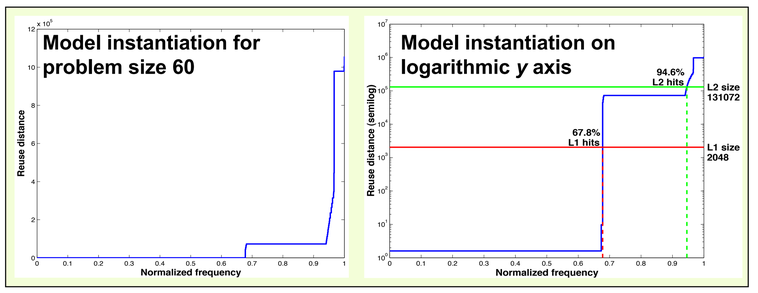
Model bins with constant distance first.
Recursively split rest of data:

- normalized frequency ratio the same across all problem sizes at each split
- select ratio s.t. the two subsets cover equal sized ranges of reuse distance





Model Evaluation



Evaluate the MRD models for the desired problem size.
For a fully-associative cache, the number of cache misses is predicted by counting the number of accesses with reuse distance larger than the size of the cache.

Assuming an uniform distribution of accessed memory blocks, we can combine the MRD predictions with a probabilistic model to approximate the number of cache misses for a set-associative cache with *s* sets and associativity *k*. Probability that a memory access with reuse distance *n* misses in a set-associative cache is:

$$P_{miss}(n,s,k) = 1 - \sum_{i=0}^{k-1} \left(\frac{1}{s}\right)^{i} \left(\frac{s-1}{s}\right)^{n-i} \binom{n}{i}$$

where *n* = memory reuse distance
s = number of sets in cache
k = associativity level

and the number of misses for a single histogram (a set of aggregated references) is:

$$Num_{misses}(Hist,s,k) = \sum_{bin_i \in Hist} \left(P_{miss}(D_{bin_i},s,k)\, F_{bin_i} \right)$$

D_{bin_i} = average MRD for bin_i
F_{bin_i} = execution frequency for bin_i

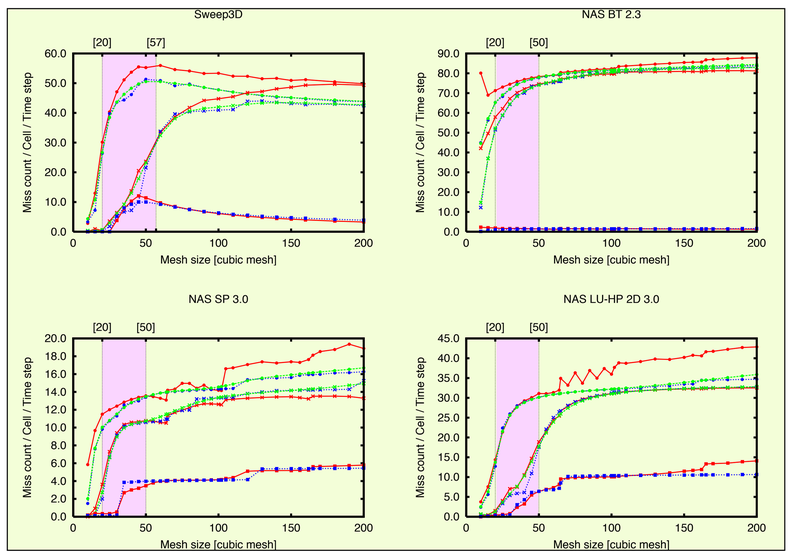
Validation

Compare predictions versus measurements with hardware performance counters. All predictions are based on measurements for mesh sizes highlighted in the pink region.

Itanium2:

- **L2 cache:** 256 KB, 8-way set-associative
- **L3 cache:** 1.5 MB, 6-way set-associative
- **L2 TLB:** 128 entries fully-associative
- **memory page:** 16 KB

Legend
L2 measured
L2 predicted MRD
L2 predicted probabilistic
L3 measured
L3 predicted MRD
L3 predicted probabilistic
TLB measured
TLB predicted MRD



Memory Latency Predictions

Predict observed latency of cache misses

- number of outstanding loads
  –analyze dependency graph
- measured latency of parallel misses