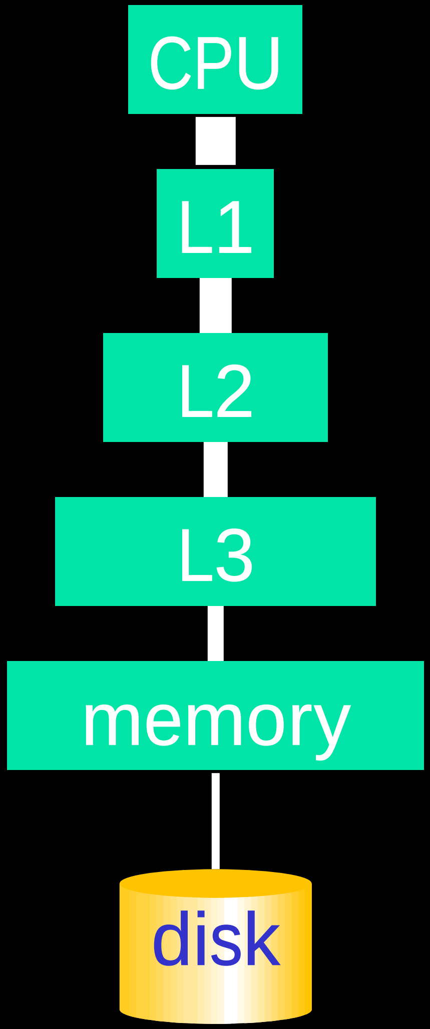
A Theory Temporal and Spatial Locality

Marc Snir
(with Jing Yu & Sara Sadeghi-Baghsorkhi)

Caching & Locality



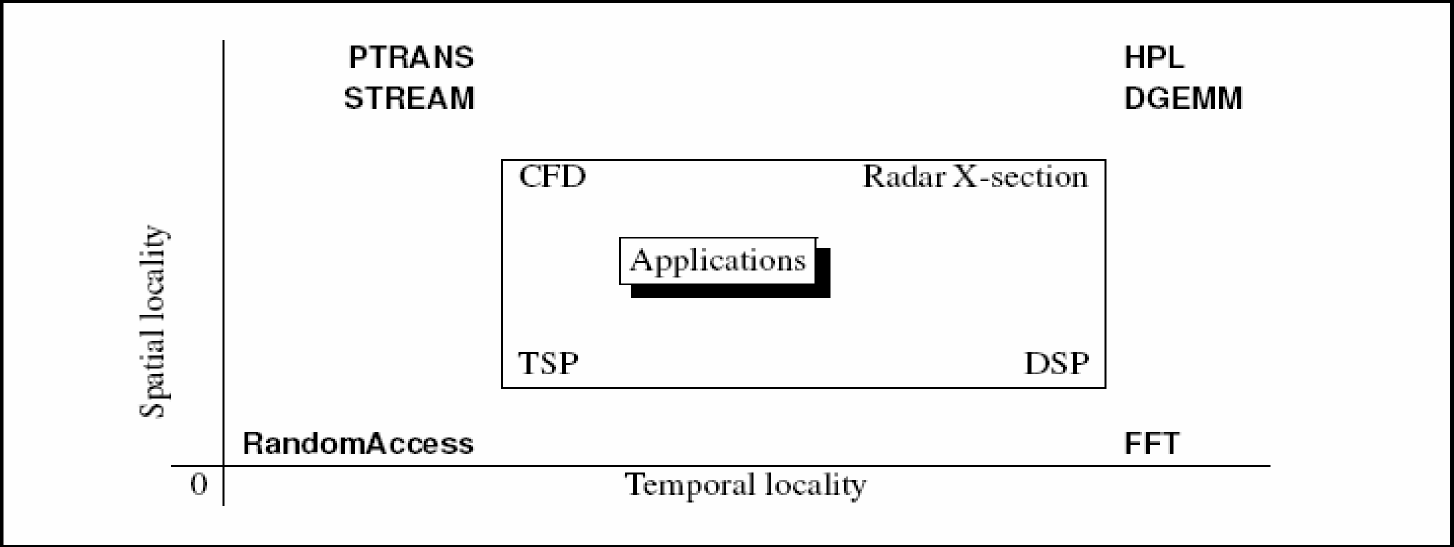
- Caches & memory hierarchy
  - higher levels are smaller and faster
  - maintain copies of data from lower levels
  - provide illusion of fast access to larger storage, provided that most accesses are satisfied by cache
  - work if memory accesses exhibit good locality of references

Locality of References (Wikipedia)

- Temporal locality
  - The concept that a resource that is referenced at one point in time will be referenced again sometime in the near future.
- Spatial locality
  - The concept that likelihood of referencing a resource is higher if a resource near it was just referenced.
- Assume that temporal and spatial locality can be defined intrinsically as a property of the reference stream, with no connection to cache features

An Operational Definition of Locality

- Good temporal locality ⇒ cache miss traffic decreases fast when cache size increases
- Good spatial locality ⇒ cache miss traffic does not increase much when line size increases



The Naïve View

1. Temporal locality and spatial locality are intrinsic to the reference stream – do not depend on cache parameters
2. Each can be defined using one (or few parameters)
3. Temporal locality determine sensitivity to cache size and spatial locality determine sensitivity to line size.

- We show that 1 and 3 contradict and 2 is false

Definitions

- $Hit^A(M,L)$ – Hit rate (fraction of accesses that are cache hits) for sequence A of accesses for a (LRU, fully associative) cache of size M, with cache lines of length L (L divides M)
- $Miss^A(M,L) = 1\text{-}HIT^A(M,L)$ – Miss rate
- $MT^A(M,L) = L \times Miss^A(M,L)$ – Miss traffic
- Good temporal locality = $MT^A(M,L)$ decreases fast when M increases
- Good spatial locality = $MT^A(M,L)$ does not increase much when L increases (the miss rate decreases almost by a factor of L)

Is “LRU” Essential to the Definition?

- Def: Online cache management algorithm

  Offline cache management algorithm

  OPT (optimal offline cache management algorithm) – furthest future use

Assume $L=1$

Def: Cache management ALG is $c(M,m)$ – competitive if for any sequence of accesses A

$$Miss^{A}_{ALG}(M) \leq c(M,m) \times Miss^{A}_{OPT}(m)$$

Competitive Analysis

- Thm (Sleator & Tarjan):
  - LRU is *$M/(M-m+1)$* competitive
  - If an online algorithm is c-competitive then $c \geq M/(M-m+1)$
- Same result is valid for other common cache replacement algorithms, e.g. FIFO, RANDOM, etc.

$\Rightarrow$ “LRU” is not an essential part of the definition

Is “Fully Associative” Essential?

- Direct mapped cache of size M can behave, in the worst case, as a fully associative cache of size 1.
- Problem can be avoided if addresses are hashed
- Def: H is a Universal Hash Function if it is a class of functions $h:A\rightarrow B$ such that, for any $x, y \in A$, and randomly chosen $h \in H$, $Prob[h(x)=h(y)] = 1/|B|$.

  H is k-independent if for any $a_1, \ldots, a_k \in A$, $b_1, \ldots, b_k \in B$, $Prob[h(a_1)=b_1, \ldots, h(a_k)=b_k] = 1/|B|^k$

Hashed Dircet Mapped Cache

- $Miss^A_{Hash}(M)$ – miss rate on direct mapped cache of size M, assuming that addresses are hashed using a universal hash function.
- Thm:

1. $E[Miss^A_{Hash}(M)] \leq M/(M-m+1)\ (Miss^A_{OPT}(m)+m)$

2. If H is m-independent and $Miss^A_{LRU}(m) < \varepsilon$ then $Miss^A_{Hash}(m/\varepsilon) < 2\varepsilon$

Outline of Proof for 1

- Use potential function
  - Φ^i – number of elements that are both in the cache of Hash and in the cache of OPT at step i
  - $\Delta \Phi^i = \Phi^i - \Phi^{i-1}$
  - δ^i_{ALG} indicator function for misses
  - $X^i = \delta^i_{Hash} - M/(M-m+1)\ (\delta^i_{OPT} + \Delta \Phi^i)$

$$\sum X^i = Miss_{Hash} - M/(M-m+1)\ (Miss_{OPT} + \Phi_n - \Phi_0)$$

Show that $E[X^i] \leq 0$ by simple case analysis

Synthetic Traces

- Hypothesis: Memory traces can be meaningfully characterized using a few parameters
- Informal Def: two memory traces are *similar* $(T_1 \approx T_2)$ if, for any cache size, both traces result in approximately the similar me hit rates.
- The synthetic trace hypothesis: One can define a parametric family of stochastic traces $T(\theta_1, \ldots, \theta_k)$ so that, for any application trace T, there is a set of parameter values $a_1,\ldots,a_k$ so that $T \approx T(a_1,\ldots,a_k)$.
  - (Stone, Eeckhout, Stromaier,…)

Hypothesis is Wrong

- Intuition: any monotonic nonincreasing graph can represent miss traffic (miss rate) as function of cache size for some memory trace

- Formal theorem: Given $\varepsilon > 0$, $0 < m_1 < \ldots < m_k$ sequence of increasing cache sizes and $1 \geq p_1 \geq \ldots \geq p_k \geq 0$ sequence of decreasing miss rates, one can build a memory trace so that miss rate for cache of size m_i is within ε of p_i.

Informal Proof

A sequence of accesses of the form

$$(a_1^1 \ldots a_{m_1+1}^1)^{n_1} \ldots (a_1^i \ldots a_{m_i+1}^i)^{n_i} \ldots$$

will have $m_i(n_i - 1)$ accesses that miss for cache of size m_i but do not miss for larger cache. Theorem follows by choosing n_i so that $(n_i - 1) \approx \lambda\,(p_{i-1} - p_i)$, for suitable constant λ.

Reuse

Theorem

- A measure of temporal locality that predicts sensitivity of cache miss bandwidth to cache size must depend on line size
- A measure of spatial locality that predicts sensitivity of cache miss bandwidth to line size must depend on cache size

Proof Outline

	small cache	large cache
short line	**m,1**	**M,1**
long line	**m,L**	**M,L**

- We disprove the following:
  - Increased cache size significantly decreases miss traffic for line size 1 iff it does so for line size L
  - Increased line size significantly increases miss traffic for cache size m iff it does so for cache size M

Four counterexamples

$$\begin{tikzcd} \mathbf{m,1} \arrow[r] & \mathbf{M,1} \\ \mathbf{m,L} \arrow[r, leftrightarrow] & \mathbf{M,L} \end{tikzcd}$$

good T.L. for short lines but
bad T.L. for long lines

$$\begin{tikzcd} \mathbf{m,1} \arrow[r, leftrightarrow] & \mathbf{M,1} \\ \mathbf{m,L} \arrow[r] & \mathbf{M,L} \end{tikzcd}$$

bad T.L. for short lines but
good T.L. for long lines

$$\begin{tikzcd} \mathbf{m,1} \arrow[d, leftrightarrow] & \mathbf{M,1} \\ \mathbf{m,L} & \mathbf{M,L} \arrow[u] \end{tikzcd}$$

bad S.L. for small cache but
good S.L. for large cache

$$\begin{tikzcd} \mathbf{m,1} & \mathbf{M,1} \arrow[d, leftrightarrow] \\ \mathbf{m,L} \arrow[u] & \mathbf{M,L} \end{tikzcd}$$

good S.L. for small cache but
bad S.L. for large cache

Proof Outline

$(a,a+L,\dots,a+(m-1)L)^{\alpha}$ … (misses in ***m,L***)

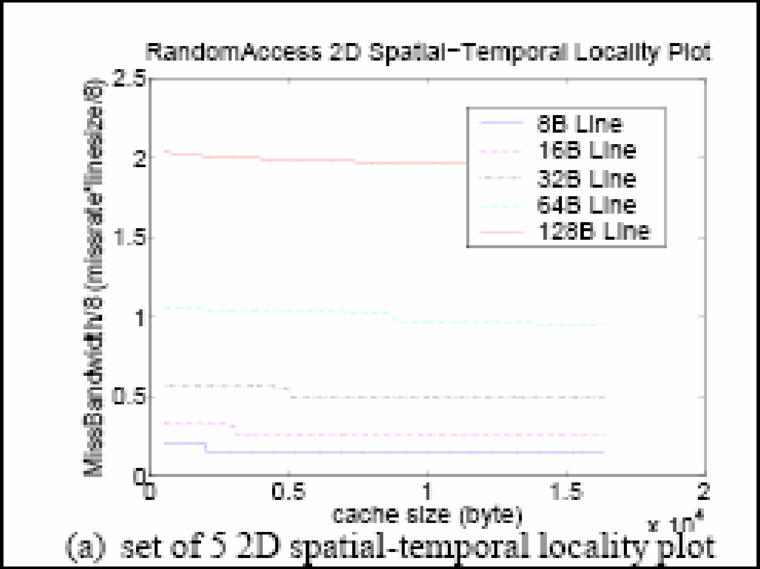
$(b,b+1,\dots,b+M-1)^{\beta}$ … (misses in ***m,1*** and ***m,L***)

$(c,c+L,\dots,c+(M-1)L)^{\gamma}$ … (misses in ***m,1*** ***m,L*** and ***M,L***)

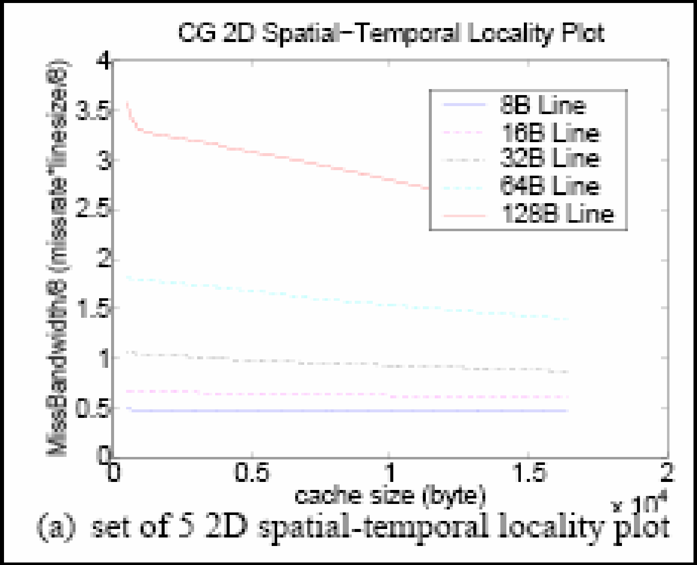
$(d,d+1,\dots,d+M)^{\delta}$ (misses in all)

- Large α: bad spatial locality for ***m*** but good spatial locality for ***M***
- Large γ: good spatial locality for ***m*** but bad spatial locality for ***M***
- Etc.

This is not only theory



(a) set of 5 2D spatial-temporal locality plot



(a) set of 5 2D spatial-temporal locality plot

- RandomAccess benefits from larger cache for small line size but not large line size
- CG benefits from larger cache for large line size but not small line size

Spatial Locality Revisited

- Previous analysis assumed that variables cannot be remapped in memory during execution; in practice such remapping is possible (e.g. during garbage collection and memory compaction). Can this be used to improve spatial locality?
- Model: Caching with Rearrangements. When a line has to be evicted can pick L arbitrary items from the cache and store them in a memory line frame
- RMiss – miss rate in rearrangeable caching.

Rearrangeable Model

- Thm:
- LRU is $L(M-L)/(M-m)$ competitive in rearrangeable model
- If ALG is a c-competitive online algorithm in the rearrangeable model then

$c \geq L(M-L^2+2L)/(M-m+L)$

Online algorithms “loose” a factor of L in the rearrangeable model – spatial locality cannot be managed online

Open Problems

- Close gap between u.b. and l.b in last theorem
- Describe what is the optimal offline algorithm for the rearrangeable model.

References

- Marc Snir and Jing Yu. On the Theory of Spatial and Temporal Locality . Technical Report No. UIUCDCS-R-2005-2611, Department of Computer Science, UIUC, July 2005.
- Jing Yu, Sara Baghsorkhi and Marc Snir. A New Locality Metric and Case Studies for HPCS Benchmarks . Technical Report No. UIUCDCS-R-2005-2564, Department of Computer Science, UIUC, April 2005