# Symbolic Footprint Analysis and Uses

---

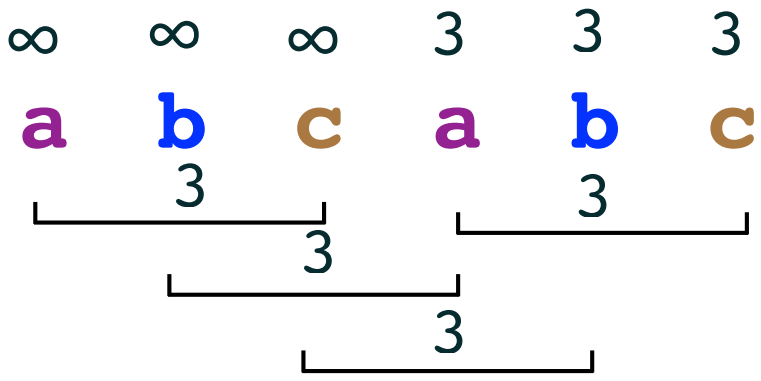
Chen Ding

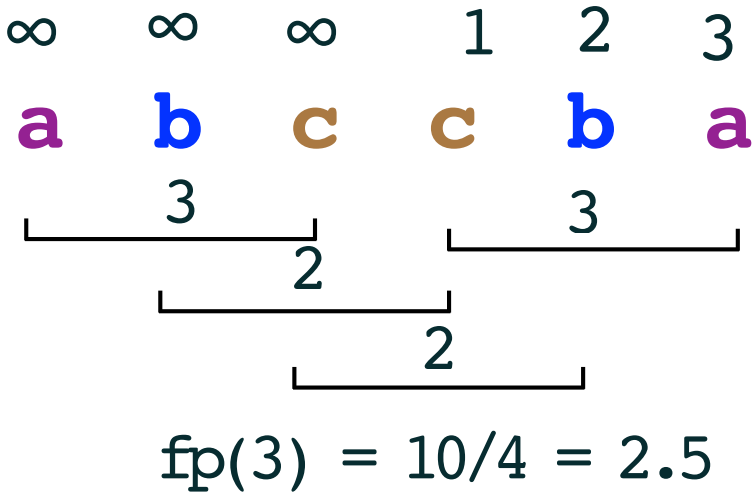
Professor
Department of Computer Science
University of Rochester

# Locality Analysis

- What is locality?
  - Denning's definition
    - working set
  - how to quantify?
- Essence of cache
  - dynamic sharing of (fast) memory
    - memory allocation for working sets
  - within/across threads and programs
    - program optimization, programs management
- Locality analysis
  - program parameters
  - conversion to miss ratio (cache parameters)

- Locality of an access
  - shorter reuse distance -> better locality
- Locality of an execution window
  - smaller working set -> better locality
- Timescale locality: footprint fp(x)
  - average working set size of all windows of length x
    - parameterized by timescale
  - smaller footprint -> smaller average WSS -> better locality



# Reuse Distance (rd) vs Reuse Time (rt)

- Distribution function P(rd) and P(rt)
  - e.g.
    - P(rd=3) = 2/7
    - P(rt=3) = 0/7
  - P(rt) is easy to measure

```
rt: ∞ ∞ ∞ 4 2 4 6
rd: ∞ ∞ ∞ 3 1 2 3
    a b c a a c b
```

- P(rt) can be used to compute footprint
  - precise solution [Xiang et al. PACT'11]
  - approximate solution, Denning 1968
- The higher-order theory of locality (HOTL)
  - miss ratio, reuse distance, fill time [Xiang et al. ASPLOS'13]
- Growing literature [OSDI'14, FAST'15, ATC'16, MEMSYS'16]

# Static Analysis of Footprint

- Building on reuse distance equations
  - Beyls and D'Hollander JSA 2004
  - nested loops and affine array subscript expressions
- Key concepts
  - references and iterations
  - reuse pairs
  - forward reuses
  - iteration polytopes

# Step 1. Reuse Pairs

$$\forall r, s \in \mathscr{R} : \mathsf{reuse}(r \rightarrow s) = \{(I_r, J_s) \in \mathbb{Z}^n \mid \text{subject to conditions (2a)--(2d)}\}, \tag{2}$$

$$I_r \in \mathsf{IS}(r) \wedge J_s \in \mathsf{IS}(s) \quad \text{(iteration space)}, \tag{2a}$$

$$I_r \lessdot J_s \quad \text{(execution order)}, \tag{2b}$$

$$r@I_r = s@J_s \quad \text{(same location)}, \tag{2c}$$

$$\forall t \in \mathscr{R} : \neg(\exists K_t \in \mathsf{IS}(t) : I_r \lessdot K_t \lessdot J_s \wedge t@K_t = r@I_r) \quad \text{(no intervening access)}. \tag{2d}$$

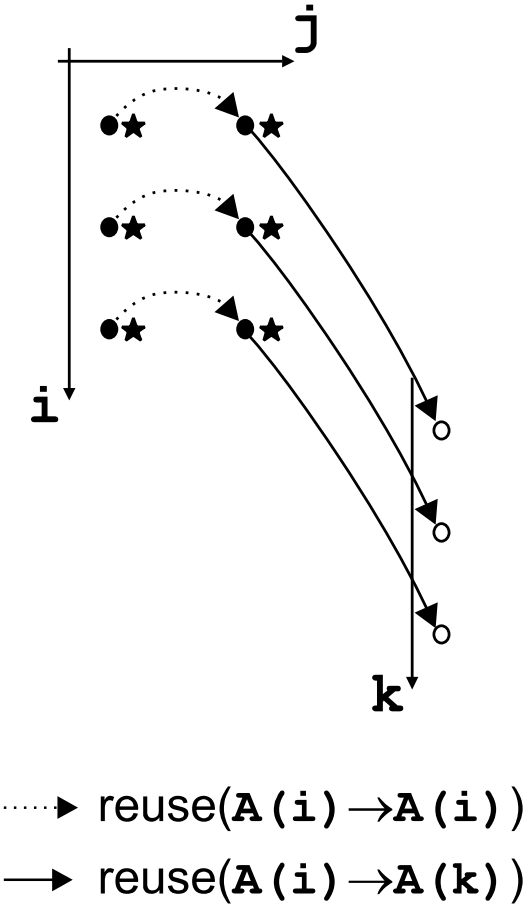
# Step 2. Iteration Polytopes

```
do i=1,N
  do j=1,2
    B(i,j) = A(i)
  enddo
enddo
do k=1,N
  A(k)=0
enddo
```

$$\text{reuse}\,(\texttt{A(i)} \rightarrow \texttt{A(i)}) = \{(i,j,i',j') | 1 \leq i, i' \leq N \wedge j = 1 \wedge j' = 2\}$$

$$\text{reuse}\,(\texttt{A(i)} \rightarrow \texttt{A(k)}) = \{(i,j,k) | 1 \leq i \leq N \wedge j = 2 \wedge k = i\}$$

$$\text{reuse}_F\,(\texttt{A(i)}) = \{(i,j) | 1 \leq i \leq N \wedge 1 \leq j \leq 2\}$$




$$\text{reuse}_F(r) = \bigcup_{\forall s \in \mathscr{R}} \{I_r \mid \exists J_s : (I_r, J_s) \in \text{reuse}(r \rightarrow s)\},$$

# Static Reuse Distance

Bin Bao

[CGO 2013]

$$
\begin{aligned}
&for(jj = 0; jj < N; jj = jj + B_j)\\
&\quad for(kk = 0; kk < N; kk = kk + B_k)\\
&\quad\quad for(i = 0; i < N; i = i + 1)\\
&\quad\quad\quad for(j = jj; j < min(jj + B_j, N); j = j + 1)\\
&\quad\quad\quad\quad for(k = kk; k < min(kk + B_k, N); k = k + 1)\\
&\quad\quad\quad\quad\quad C[i][j] = beta * C[i][j] + alpha * A[i][k] * B[k][j];
\end{aligned}
$$

| Loop | Reuse | Reuse distance (bytes) | Reuse time (accesses) |
|---|---|---|---|
| k | $C[i][j]$ | $8 * 3$ | $3$ |
| j | $A[i][k]$ | $8 * 1 + 8 * B_j + 8 * B_j$ | $3 * B_j$ |
| i | $B[k][j]$ | $8 * B_k + 8 * B_k * B_j + 8 * B_j$ | $3 * B_j * B_k$ |
| jj | $A[i][k]$ | $8 * B_i * B_k + 8 * B_k * B_j + 8 * B_i * B_j$ | $3 * B_i * B_j * B_k$ |
| kk | $C[i][j]$ | $8 * B_i * B_k + 8 * B_k * J + 8 * B_i * J$ | $3 * B_i * J * B_k$ |
| ii | $B[k][j]$ | $8 * B_i * K + 8 * K * J + 8 * B_i * J$ | $3 * B_i * J * K$ |

# Uses

- Locality optimization of GPU kernels
  - select the best order to execute the thread groups in a thread block
    - statically compute the footprint and miss ratio
      - static or dynamic optimization
  - select the best placement of arrays
    - statically compute the footprint and miss ratio
      - for every group of arrays
    - possible use in PORPLE [Chen et al. MICRO 2014]

# Uses (cont'd)

- Optimization of NUMA task and data placement (TPDP)
  - joint work with Robert Ho and Yaoqing Gao
  - tomorrow's workshop
- Write locality
  - write caching [Li+ LCPC'16] in ATLAS [Chakrabarti+ OOPSLA'14]
  - write locality analysis / optimization [Chen+ MEMSYS'16]
    - modeling writebacks in mixed read-write cache
    - write reuse distance
    - fast measurement
  - compiler optimization of writes
- Other suggestions?