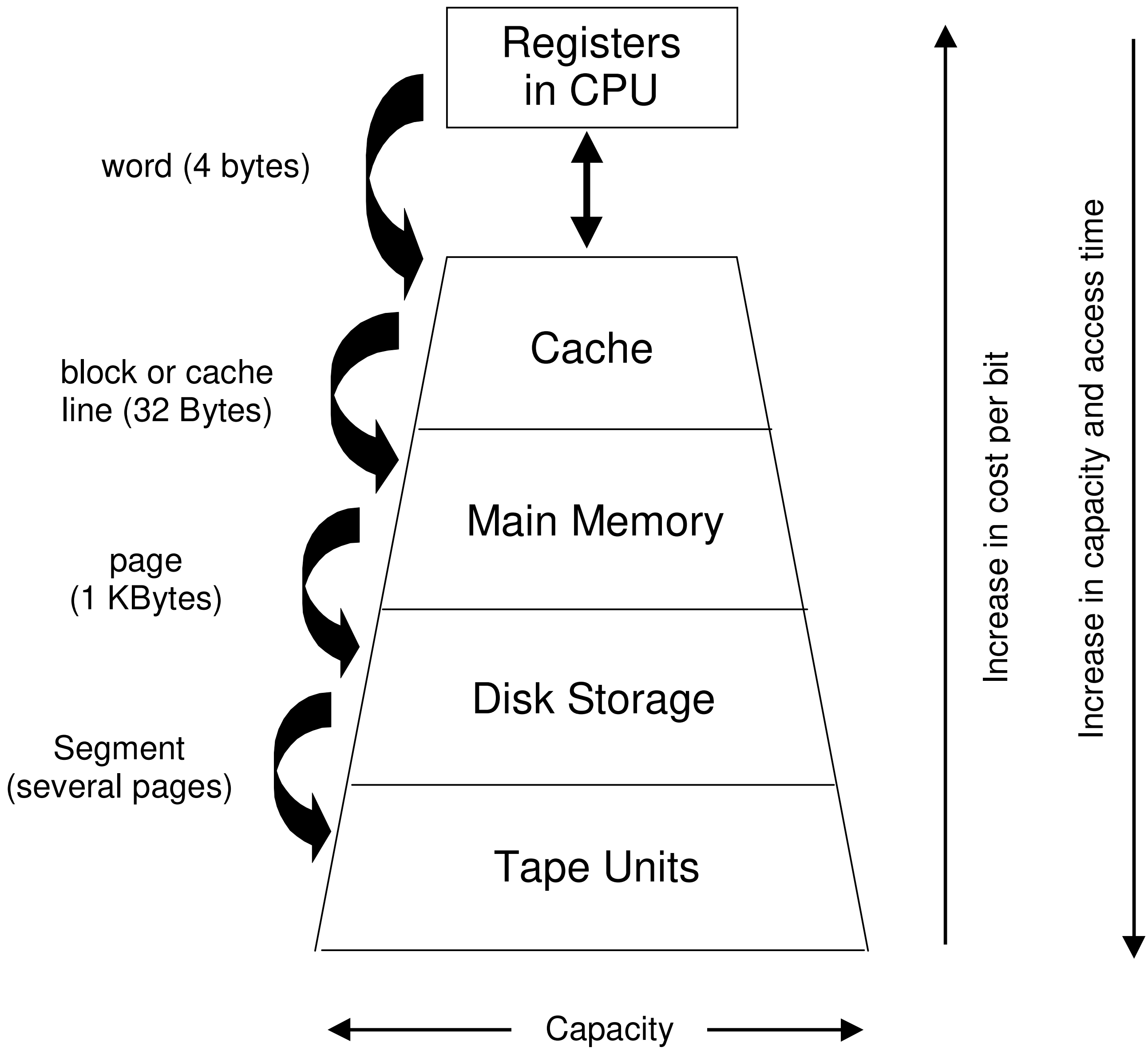
MATHEMATICAL TOOLS

FOR ANALYZING AND OPTIMIZING

MEMORY BEHAVIOR

Philippe Clauss
ICPS
Université Louis Pasteur
Strasbourg
France

Hierarchically-organized Memory Systems



Performance of a Hierarchically-organized Memory System

➤ Underlying assumption

= most of the data accesses from the low or intermediate levels

$\Rightarrow$ *Fast access time*

➤ In practice

performance is variable over the programs

performance depends on the address stream of a program

➤ Efficiency of a program on a given hierarchy

characterized by :

its *temporal locality*

&

its *spatial locality*

Improving Performance ⇔ Improving Data Locality

⇒ *Compiler transformations :*

➤ <u>Program restructuring</u> reordering the data accesses loop-oriented programs : *permutation, tiling, fusion, …*	➤ <u>Data layout optimizations</u>

based on limited or heuristic analyses

leading to poor results
for some programs
and cache organizations

⇒ *necessity of precise analysis frameworks*

Precise analysis frameworks
$\Rightarrow$ mathematical models

2 main mathematical frameworks :

Cache Miss Equations

(S. Ghosh, M. Martonosi, S. Malik – Princeton Univ. 1997-98)

Generalized Reference Windows

(K. Gallivan, W. Jalby, D. Gannon, C. Eisenbeis – INRIA Rocquencourt 1988-1998)

Cache Miss Equations

✓Perfect nested DO-loops

✓Subscript expressions of array references and bounds of a loop index
= affine functions of the enclosing loop indices

✓No conditional expressions

✓Loops with constant step values

<u>Example</u> : Matrix product

```
DO i =1,N
    DO k=1,N
        DO j=1,N
            Z(j,i)+=X(k,i)*Y(j,k)
```

$\Rightarrow$ *Reuse vectors* :

✓Iterations (i,k,j) and (i,k,j+1) can access the same cache line for the reference Z(j,i)

$\Rightarrow$ *spatial reuse*

$\Rightarrow \vec{r}_1 = (0,0,1)$

✓Iterations (i,k,j) and (i,k+1,j) access the same array element Z(j,i)

$\Rightarrow$ *temporal reuse*

$\Rightarrow \vec{r}_2 = (0,1,0)$

Cache Miss Equations (CMEs)

If a reference is reusing a previously accessed line

If an intervening access evicts the line from the cache

$\Rightarrow$ *cache miss*

else

$\Rightarrow$ *cache hit*

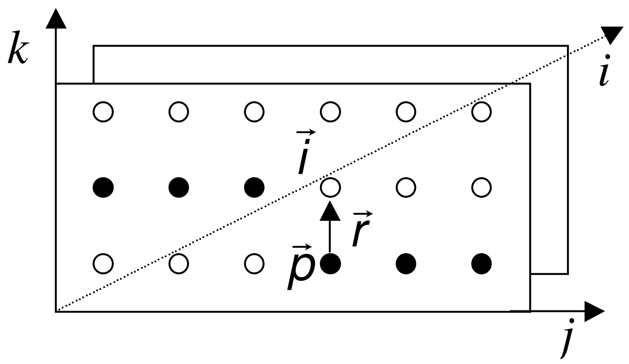
Solutions to CMEs correspond to iterations where a cache miss occurs

Number of solutions = number of occuring cache misses

For each reuse vector :

- cold CMEs = 1st reference to a line
- replacement CMEs = the line has been replaced before its reuse

Example : $\vec{i} = (1,2,4), \vec{r} = (0,1,0), \vec{p} = \vec{i} - \vec{r} = (1,1,4)$



● = potential interfering points $\vec{j}$

interference ⇐ when the cache line (or cache set) accessed by reference R_A of $\vec{i}$ = cache line (or cache set) accessed by R_B of any $\vec{j}$ between $\vec{p}$ and $\vec{i}$

$$\Rightarrow Cache\_Set_{R_A}(\vec{i}) = Cache\_Set_{R_B}(\vec{j})$$

Example : Matrix product

```
DO i = 1,32
    DO k = 1,32
        DO j = 1,32
            Z(j,i) += X(k,i) * Y(j,k)
```

$\rightarrow$ Considering $\vec{r} = (0,0,1)$ (spatial reuse of Z)

$$\vec{i} = (i, k, j) \Rightarrow \vec{p} = (i, k, j-1)$$

$\rightarrow$ Considering a 8KB 2-way associative cache with 256 cache lines and 4 array elements / cache line

$\Rightarrow$ Replacement miss equation for the interferences with X(k,i) along $\vec{r}$:

$$Cache\_Set\_Z(j,i) = Cache\_Set\_X(k',i')$$
where $(i',k',j') \in [(i,k,j-1),(i,k,j)]$

$$\Rightarrow \lfloor (4192 + 32i + j - 1)/4 \rfloor \bmod 128$$
$$= \lfloor (2136 + 32i + k - 1)/4 \rfloor \bmod 128$$

$$\boxed{\Rightarrow 4192 + 32i + j - 1 \;=\; 2136 + 32i + k + 512n + b}$$

where - 4192, 2136 = base adresses of Z and X
- $(i, j) \in [(0,0),(31,31)]$
- $b \in [-3,3]$

Generalized Reference Windows

Reference Window W_p at program point p
=
set of memory locations touched before or at p
that will be accessed again after or at p

If $|W_p| \leq$ cache size $\Rightarrow$ no reload from main memory
(no cache miss)

Case of an <u>associative cache</u> :

(memory line = set of consecutive memory locations mapped into the same cache line)

$$GW_p^s$$
=
set of memory lines mapped into the cache set s
accessed before or at p
that will be accessed again after or at p

Generalized Reference Windows in Loops

Example :

```
      REAL X(0:99,0:99)
C     Start address of X: 0 (in words)
      REAL Y(0:99,0:99)
C     Start address of Y: 10251 (in words)
      DO I=0,N
         DO J=0,M
S:          X(J,I)
T:          X(J,I+1)
U:          Y(J,I)
         ENDDO
      ENDDO
```

$\Rightarrow$ Iteration Space $\{(I, J) | 0 \le I \le N, 0 \le J \le M\}$

$\rightarrow$

Computing the reference window

between T and S

at point (10,10)

Computing the reference window between T and S at point (10,10)

1) Set of points…

- lexicographically inferior to (10,10)

$$INF = \{(10, J) | 0 \le J \le 9\} \cup \{(I, J) | 0 \le I \le 9, 0 \le J \le M\}$$

- lexicographically superior to (10,10)

$$SUP = \{(10, J) | 11 \le J \le M\} \cup \{(I, J) | 11 \le I \le N, 0 \le J \le M\}$$

2)
- memory locations touched by T before (10,10)
  ⇒ affine transformation τ : $(I, J) \rightarrow (J, I+1)$

$$\tau(INF) = \{(J, 11) | 0 \le J \le 9\} \cup \{(J, I+1) | 0 \le I \le 9, 0 \le J \le M\}$$

- memory locations touched by S after (10,10)
  ⇒ affine transformation σ : $(I, J) \rightarrow (J, I)$

$$\sigma(SUP) = \{(J, 10) | 11 \le J \le M\} \cup \{(J, I) | 11 \le I \le N, 0 \le J \le M\}$$

3) Reference window = $\tau(INF) \cap \sigma(SUP)$

$$= \{(J, 10) | 11 \le J \le M\} \cup \{(J, 11) | 0 \le J \le 9\}$$

4) Number of integer points in the reference window = M

Reaching good optimization compromises…

$$\begin{cases} \text{reference window volume} \leq \text{cache size} \\ \text{program latency} \leq T_{\max} \\ \text{maximum parallelism} \leq \text{number of processors} \\ \text{communication volume} \leq f(\text{bandwidth}) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \end{cases}$$

⇒ System of *parametric multi-variate polynomial equations*

⇒ *Integer solutions*

Mathematical Tools Needed

✓Parametric Tools
$\rightarrow$ allows optimization of programs

✓Exact Array Dataflow Analysis
$\rightarrow$ PIP (Parametric Integer Programming)
$\rightarrow$ Polylib (Parametric Vertices – Validity domains)

✓Number of solutions in parametric diophantine equations $\Leftrightarrow$ Number of integer points in parametric polytopes
$\rightarrow$ Parameters in constant parts of the equations
$\rightarrow$ Ehrhart Polynomials
$\rightarrow$ Parametric coefficients in the equations
$\rightarrow$ Ehrhart Polynomials ?

✓Affine transformations of polytopes (projection, …)
$\rightarrow$ Non-regular lattices of integer points
$\rightarrow$ Holes
$\rightarrow$ Intersections of such lattices
$\rightarrow$ Ehrhart Polynomials
$\rightarrow$ need the resolution of parametric multi-variate polynomials

Parametric Multi-variate Polynomial Equations : an investigation direction, *Gröbner Bases*

Term ordering : Lexicographic order $\prec$

Let $t_1 = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ and $t_2 = x_1^{\beta_1} x_2^{\beta_2} \cdots x_n^{\beta_n}$

$$t_1 \prec t_2 \text{ iff } \exists i \leq n \text{ s.t. } \alpha_j = \beta_j \text{ for } i < j \leq n \text{ and } \alpha_i < \beta_i$$

Head Term and Reduction

for every polynomial $f \in Q\lfloor x_1, x_2, \ldots, x_n \rfloor$

head(f) = largest term under $\prec$ that has a non-zero coeff.
ldcf(f) = coeff. of head(f)

$\Rightarrow$ *f = ldcf(f) head(f) + g* where *head(g)* $\prec$ *head(f)*

g = tail(f)

Let *f* and *g* be 2 polynomials ; suppose *g* has a term *t* that is a multiple of *head(f)* :

$$g = at + \hat{g} \text{ where } a \in Q \text{ and } t = t' \, head(f)$$

for some term *t'*

$\Rightarrow$ *g* is *reductible* with respect to *f*

$\Rightarrow$ Reduction operation :

$$g \xrightarrow{f} h$$

$$h = g - \hat{a}t' f = \hat{a}t' \, tail(f) + \hat{g} \text{ and } \hat{a} \, ldcf(f) = a$$

g is *reductible* with respect to $F = \{f_1, f_2, \ldots, f_r\}$
$\Leftrightarrow$ g is reductible with respect to 1 or more polynomials in F

else *g is reduced* or *g is a normal form* w.r.t. F

Through a *finite sequence of reductions*

$$g = g_1 \xrightarrow{F} g_2 \xrightarrow{F} g_3 \ldots \xrightarrow{F} g_s$$

we can obtain a polynomial g_s that is reduced w.r.t. F

$\Rightarrow \forall g_i, g_i - g \in (f_1, f_2, \ldots, f_r)$

<u>Example</u> :
$f_1 = x_1^2 x_2 - 2x_2 x_3 + 1, f_2 = x_1 x_2^2 - x_3^2 + 2x_1, f_3 = x_2^2 x_3 - x_1^2 + 5$

$$g = 3x_1^2 x_2^2 + x_1^3 - 1$$

$\Rightarrow$ $head(f_1) = x_1^2 x_2, head(f_2) = x_1 x_2^2, head(f_3) = x_1^2$

$\Rightarrow$ One possible reduction sequence :

$$\begin{aligned}
g = g_1 &\xrightarrow{f_1} g_2 = x_1^3 + 6x_2^2 x_3 - 3x_2 - 1 \\
&\xrightarrow{f_3} g_3 = 6x_2^2 x_3 - 3x_2 - 1 + x_1 x_2^2 x_3 + 5x_1 \\
&\xrightarrow{f_2} g_3 = x_3^3 - 2x_3 x_1 + 6x_2^2 x_3 - 3x_2 - 1 + 5x_1
\end{aligned}$$

$\Rightarrow$ g_3 is a normal form w.r.t. F

g can be reduced in another way leading to a different normal form !

$$g = g_1 \xrightarrow{f_2} g'_2 = x_1^3 - 6x_1^2 + 3x_1 x_3^2 - 1$$

Gröbner basis : definition

<u>Ideal</u> :

Let $f_1, f_2, \ldots, f_r \in Q[x_1, x_2, \ldots, x_n]$

$$I = \left\{ \sum_{i=1}^{r} g_i f_i \ / \ g_i \in Q[x_1, x_2, \ldots, x_3] \right\}$$

is the ideal generated by $f_1, f_2, \ldots, f_r$

$\{f_1, f_2, \ldots, f_r\}$ is called a basis for I

A basis $G \in Q[x_1, x_2, \ldots, x_n]$ is called *a Gröbner basis* for the ideal it generates iff every polynomial in $Q[x_1, x_2, \ldots, x_n]$ has *a unique normal form w.r.t. G*

introduced by *Buchberger* (1965, 1976)
→ showed that every ideal has a Gröbner basis
→ designed an algorithm

<u>About the ambiguity concerning the normal form</u> :

<u>Example</u> : 2 normal forms g_3 and g'_2

→ ambiguity resolved by augmenting the basis F by the polynomial $g_3 - g'_2$
(same ideal generated since $g_3 - g'_2 \in F$)

Buchberger's algo. ⇒ same work for all terms having more than 1 normal form

⇒ Definition of an *s-polynomial* of 2 polynomials f_1, f_2 :

Let $m = lcm(head(f_1),head(f_2)) = m_1\ head(f_1)$
$= m_2\ head(f_2)$
where m_1,m_2,m_3 are terms

$$s\text{-}poly(f_1,f_2) = m_1\ ldcf(f_2)\ f_1 - m_2\ ldcf(f_1)\ f_2$$

Buchberger's algorithm :

Let $\mathcal{NF}_G(f)$ denote any normal form of f w.r.t. the current elements of G

G is augmented until $\mathcal{NF}_G(s\text{-}poly(g_i,g_j))$ is zero
for every pair of polynomials in G

```
G :=F ;
B :={(fi,fj) / fi,fj ∈ F, i < j} ;
while B is non-empty do
    h := NF_G(s-poly(fi,fj)) ; % for some (fi,fj) ∈ B
    if h ≠ 0 then
        B :=B ∪ {(h,g) / g ∈ G} ;
        G :=G ∪ {h} ;
    fi ;
    B :=B \ {(fi,fj)} ;
od ;
```

Example : Let $F = \{f_1, f_2, f_3, f_4\}$ where

$$f_1 = x_3^4 - 6x_1x_3^3 + 13x_3^2x_1^2 - 12x_3x_1^3 + 4x_1^4$$
$$f_2 = x_2^2 - 2x_2x_1 - 2x_2x_3 + x_1^2 + 2x_1x_3 + x_3^2$$
$$f_3 = x_1^2 + 4x_1 + 3$$
$$f_4 = x_1x_3^3 - 3x_3^2x_1^2 + 3x_3x_1^3 - x_1^4 + x_3^3 - 3x_3^2x_1 + 3x_3x_1^2 - x_1^3 + x_1x_3 - 2x_1^2 + 3x_3 - 6x_1$$

lexicographic ordering with $x_1 \prec x_2 \prec x_3$

$\Rightarrow G = \{g_1, g_{2,1}, g_{2,2}, g_{3,1}, g_{3,2}, g_{3,3}\}$ where

$$g_1 = x_1^2 + 4x_1 + 3$$
$$g_{2,1} = x_1x_2^2 + 3x_2^2 + 9x_1 + 27 + 6x_2x_1 + 18x_2$$
$$g_{2,2} = 2x_2^3 + 36x_2^2 - 27x_2x_1 + 135x_2 - 108x_1 + 108$$
$$g_{3,1} = x_1x_3 + 2x_1 + 3x_3 + 6$$
$$g_{3,2} = 4x_2x_3 + 24x_3 - 2x_2^2 + 4x_2x_1 + 15x_1 + 45$$
$$g_{3,3} = 2x_3^2 + 12x_3 - x_1 + 15$$

$\Rightarrow$
- g_1 depends only on x
- $g_{2,1}$ and $g_{2,2}$ depend on both x_1, x_2
- $g_{3,1}$, $g_{3,2}$ and $g_{3,3}$ depend on x_1, x_2 and x_3

$\Rightarrow$ the variables are separated in the basis G
$\Rightarrow$ <u>similar to a triangular form</u>
(always the case with lexicographic ordering)

$\Rightarrow$ can be used <u>to compute all the common zeros of F</u> through back substitution

Conclusion

Memory behavior analysis and optimization

$\rightarrow$ very precise existing models

$\rightarrow$ which need powerful arithmetical and geometrical tools handling integers and allowing parametric computations

$\rightarrow$ At this time, existing tools handle subsets of the needs and have to be extended :

- Omega library
- PIP
- Polylib / Ehrhart

A possible answer : Gröbner bases

$\rightarrow$ can it be adapted to integers ?

$\rightarrow$ is it expandable to handle inequalities ?

$\rightarrow$ can it be used for parametric computations ?