Multi-Dimensional Arrays in UPC

Note Title 3/18/2009

So - What's the problem, anyway?

PGAS Seminar

March 19, 2009

S. Seidel

Based on –

Z-ordering and UPC, Phil Merkey, June 2009, http://www.upc.mtu.edu/applications/appl.html

Multidimensional Blocking in UPC, Barton[1], Caşcaval[2], Almasi[2], et al., LCPC 2007, LNCS 5234, pp. 47-62, 2008.

[1] U. of Alberta
[2] IBM Watson

The problem is ...

- UPC shared memory is inherently one-dimensional.
- The only data distribution scheme is 1-d block cyclic.
- This is very restrictive.

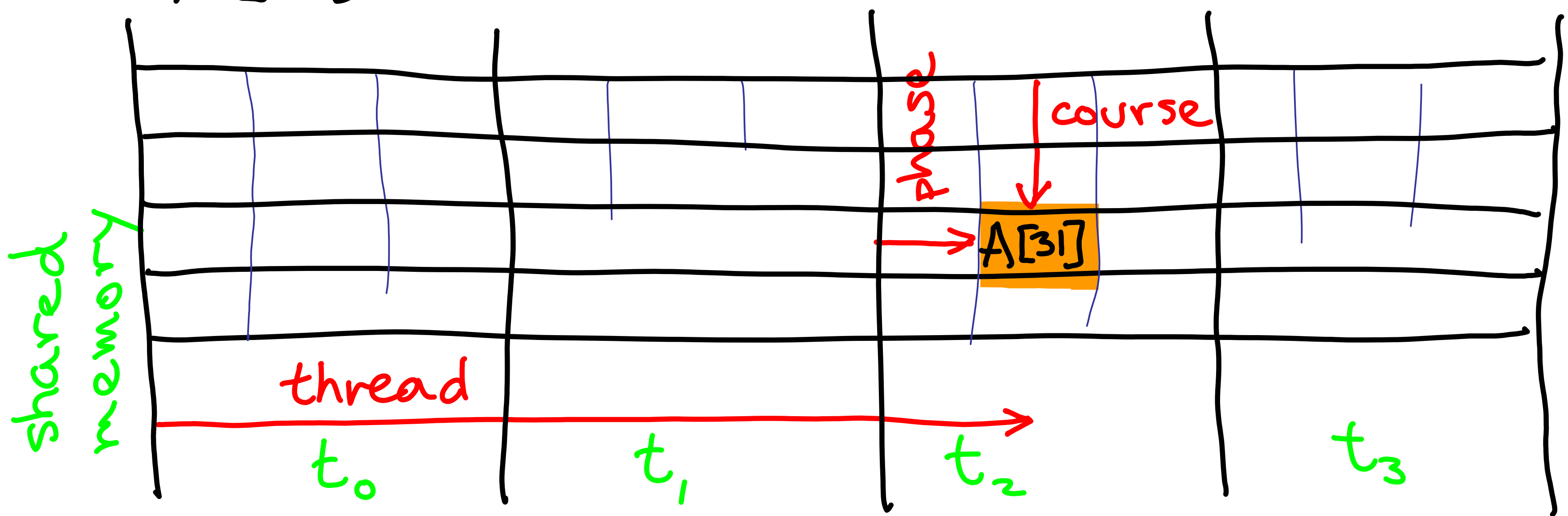
The details —

shared [b] <type> A $[d_0][d_1]..[d_{n-1}]$;

- See the Barton paper.

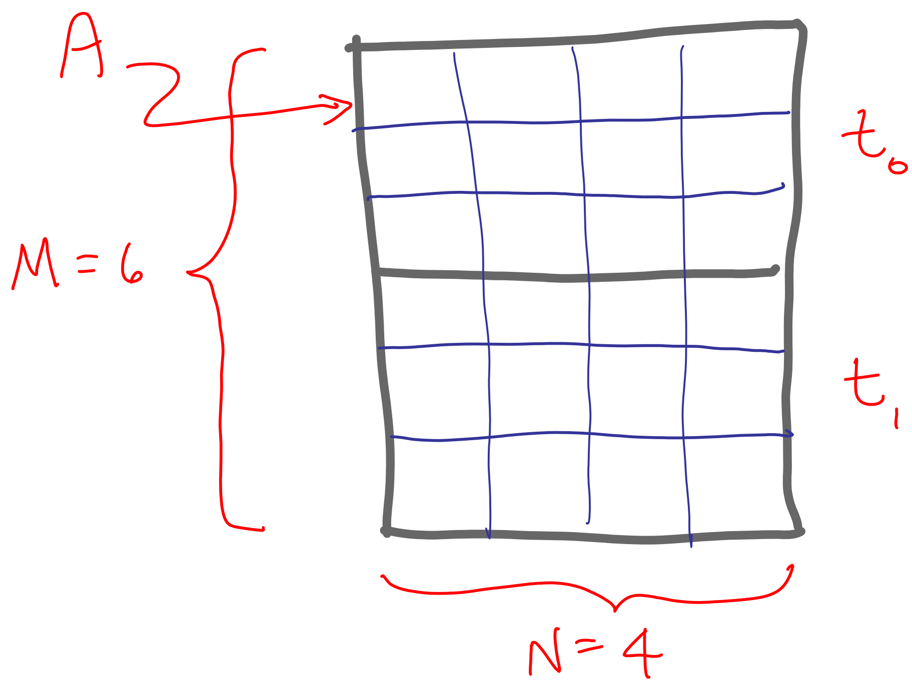
shared [3] int A [48];

A[31] has thread = 2, phase = 1, course = 2



Examples

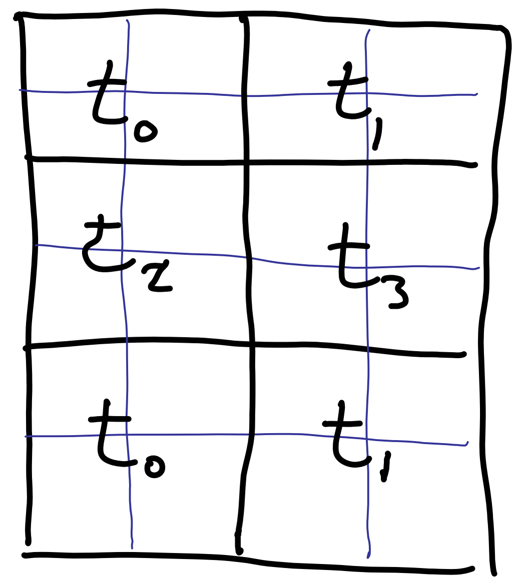
- See first example in Barton,
  - implementation on gilbert.cse



$M/b = 3$

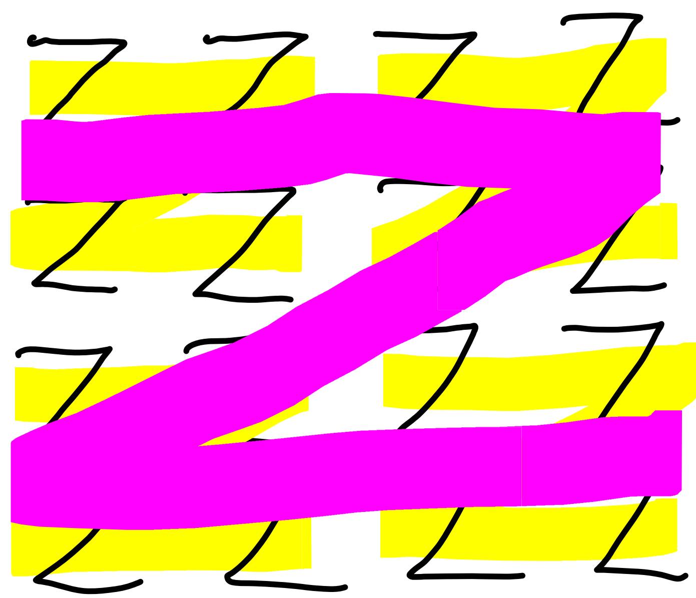
Example 2 –

- See second example in Barton
  - use a 2-d block struct
  - implementation on gilbert.cse
- productivity suffers



Proposal 1: Morton Z-ordering

- Creates a 1-to-1 correspondence between $\{0..2^n-1\} \times \{0..2^n-1\}$ and $\{0..2^{2n}-1\}$
- See Phil's TR



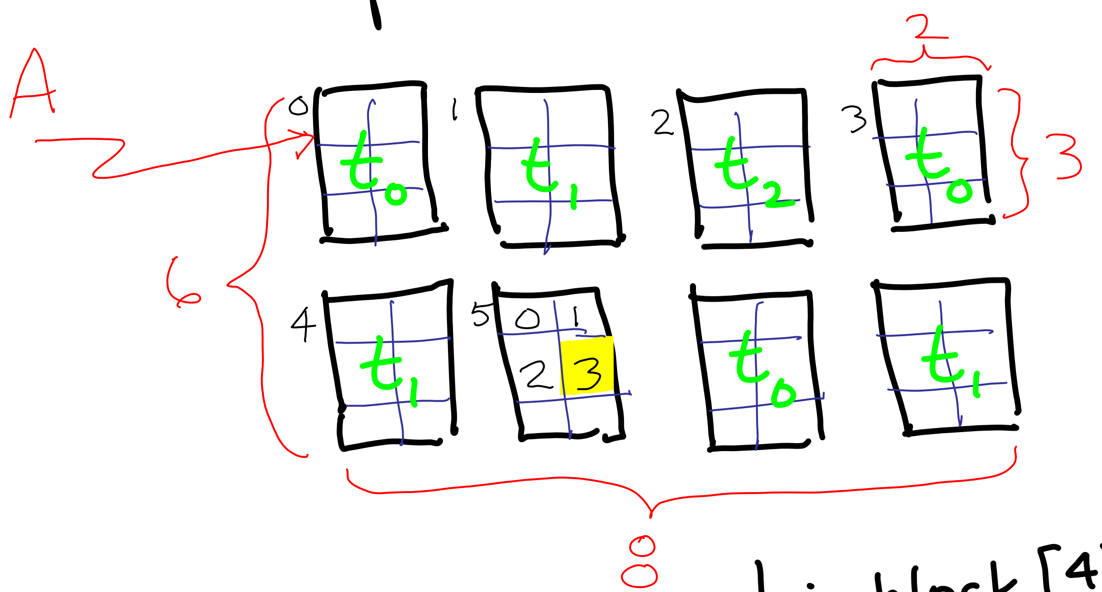
Proposal 2: A language extension that provides a tiled layout for shared memory.

shared $[b_0] \ldots [b_{n-1}]$ <type> $A[d_0] \ldots [d_{n-1}]$

This is an array of $\frac{d_0}{b_0} \times \frac{d_1}{b_1} \times \cdots \times \frac{d_{n-1}}{b_{n-1}}$ blocks each of dimension $b_0 \times b_1 \times \cdots \times b_{n-1}$.

(If $b_i \nmid d_i$, use $\left\lceil \frac{d_i}{b_i} \right\rceil$.)

Example: shared [3][2] int A[6][8];



THREADS = 3

Lin-block [4][3] = 3

$A[4][3]$

L [4][3] = 5

thread = 2

phase = 3 = Lin-block [4][3]

course = ? (should be ~~2~~ 1)

$v_0 = 4$ $\quad b_0 = 3$ $\quad d_0 = 6$ $\quad n = 2$

$v_1 = 3$ $\quad b_1 = 2$ $\quad d_1 = 8$

$$L(v) = \sum_{k=0}^{n-1} \left(\left\lfloor \frac{v_k}{b_k} \right\rfloor \times \prod_{j=k+1}^{n-1} \left\lceil \frac{d_j}{b_j} \right\rceil \right)$$

$$k=0 \quad \left\lfloor \frac{v_0}{b_0} \right\rfloor \times \left\lceil \frac{d_1}{b_1} \right\rceil = 1 \times 4 = 4$$

$$k=1 \quad \left\lfloor \frac{v_1}{b_1} \right\rfloor \times 1 = 1 \times 1 = 1$$

$$L(v) = 5$$

$$\text{course} = \left\lfloor \frac{L(v)}{\text{THREADS} \prod_i b_i} \right\rfloor = \left\lfloor \frac{5}{3 \times 6} \right\rfloor = 0$$

Claims:

- pointer operations work the same as usual
  e.g., they can be cast to private.
- dimensions are padded when $b_i \nmid d_i$.
- upc_all_alloc still works