S. Seidell
March 19, 2009
PGAS Seminar

So - what's the problem, anyway?

Multi-dimensional Arrays in UPC
Based on -

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

Multidimensional Blocking in UPC, Barton; Cașcaval; Almăsi; et al.,

1 U. of Alberta
2 IBM Watson
This is very restrictive.

is 1-or block cyclic.
The only data distribution scheme
one-dimensional.
UPC shared memory is inherently
The problem is...
A [31] has thread = 2, phase = 1, course = 2


- See the Partition paper

shared (b] type] A [a] [d] .. [c] .. ;
- See first example in Barton.

Examples
- Productivity suffers

- Implementation on Gilbert

- Use a 2-d block struch

- See second example in Barton

- Example 2 -
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
(If \( b^i \geq d^i \)) use 
\[
\left( \frac{\mathcal{B}^i}{D^i} \right)
\]
blocks each of dimension \( b_0 \times b_1 \times \ldots \times b_{n-1} \).

This is an array of 
\[
\frac{b_0}{d_0} \times \frac{b_1}{d_1} \times \ldots \times \frac{b_{n-1}}{d_{n-1}}
\]
shared \([b_0] \ldots [b_{n-1}]\). 

type \( A [d_0] \ldots [d_{n-1}]\) shared memory.

Proposal 2: A language extension that provides a titled layout for
Example: shared \([3][2]\) int \(A[6][8]\).

THREADS = 3

\[
A[4][3]
\]

\[
\text{Lin-block } [4][3] = 3
\]

\[
L [4][3] = 5
\]

thread = 2

phase = 3 = \(\text{Lin-block } [4][3]\)

course = ? (should be 2)
\[ 0 = \int_{\frac{3x}{5}}\text{THEEARS} \cdot \frac{3x}{5} \, dx = \left[ \frac{x^2}{2} \right]_{\frac{3x}{5}}^5 = \frac{5^2}{2} - \frac{\left(\frac{3x}{5}\right)^2}{2} \]

\[ \text{course} = \frac{L^2}{v} \]

\[ L = 5 \]

\[ 1 = 1 \times 1 = 1 \times \left[ \frac{t_0}{v_0} \right]_{t_0}^{1\text{e}} \]

\[ t_0 = 4 \]

\[ \left( \int_{\frac{t_0}{v_0}}^{1\text{e}} \frac{1}{1 - \frac{t_0}{v_0}} \, dt \right)_{\frac{t_0}{v_0}}^{1\text{e}} = 2 \]

\[ n = 2 \]

\[ \begin{array}{c}
\gamma = 3 \\
\dot{\gamma} = 2 \\
\dot{\gamma} = 2 \\
\gamma = 6 \\
\dot{\gamma} = 6 \\
\gamma = 4 \\
\dot{\gamma} = 6 \\
\end{array} \]
- upcall:: alloc still works.
- dimensions are padded when needed.
- g:: they can be cast to private.
- pointer operations work the same as usual.

Claims: