UPC Interoperability with Fortran

Robert W. Numrich
Minnesota Supercomputing Institute
University of Minnesota, Minneapolis
and
Goddard Space Flight Center
Greenbelt, Maryland
Where are We?

- Fortran 2003 defines interoperability with C.
- Co-arrays will be part of Fortran 2006.
- There is an overlap of co-array objects and UPC objects.
- Do we want to define an interface between the intersection of objects?
- Can we do it without changing anything on either side?
Fortran Interoperability with C

- Fortran
  
  ```fortran
  use iso_c_binding
  real(c_double) :: x(18,3:7,*)
  Integer(c_int) :: index
  ```

- C
  
  ```c
  double x[][][5][18];
  int index;
  ```
Co-array Interoperability with UPC

• Fortran
  
  use iso_c_binding
  real(c_double) :: x(18,3:7,*)[*]
  Integer(c_int) :: index[*]

• C
  
  shared double x[][5][18][THREADS];
  shared int index[THREADS];
Derived Types and Structures

• Fortran
  
  type, bind(c) :: myType
  integer(c_int) :: i,j
  real(c_float) :: s
  end type myType
  
  type(myType) :: x[*]
  z = x[p]%s

• C
  
  typedef struct{
    int m,n;
    float r;
  } myctype
  myctype y[THREADS];
  w = y[p].r
Procedure Interfaces

- **C Interface**
  ```c
  int C_Function(shared void* x, int y)
  ```

- **Fortran Interface**
  ```fortran
  interface
    integer(c_int) function cFtn(z,j), bind(c, name='C_Function')
    type(c_ptr),value :: z[*]
    integer(c_int) :: j
  end interface

  real(c_float), target :: z[*]
  integer(c_int) :: k
  call cFtn(c_loc(z),k)
  ```
Allocatable Variables

- **Fortran**
  
  ```fortran
  real, allocatable, target :: x[:,]
  allocate(x[*,])
  type(c_ptr), value :: p2x
  p2x = c_loc(x)
  ```

- **C**
  
  ```c
  upc_malloc(x, ...);
  ```
The Co-Array Fortran Standard

• Co-arrays will be added to Fortran 2006

• Additional information on the web:
  – www.co-array.org
  – www.pmodels.org