Coarray Status

Dan Nagle

PGAS

- Partitioned Global Address Space
- Examples include:
  - UPC (upc.gwu.edu)
  - Titanium (titanium.cs.berkeley.edu)
- Original Paper at ftp://ftp.numerical.rl.ac.uk/pub/reports/nrRAL98060.ps.gz

Other Sources

- WG5 N1824.pdf (John Reid's summary)

(www.nag.co.uk/sc22wg5) (see also N1830)

- Rice University (caf.rice.edu)
- FDIS J3/10-007r1.pdf (www.j3-fortran.org)
- Modern Fortran Explained by Metcalf, Reid, Cohen

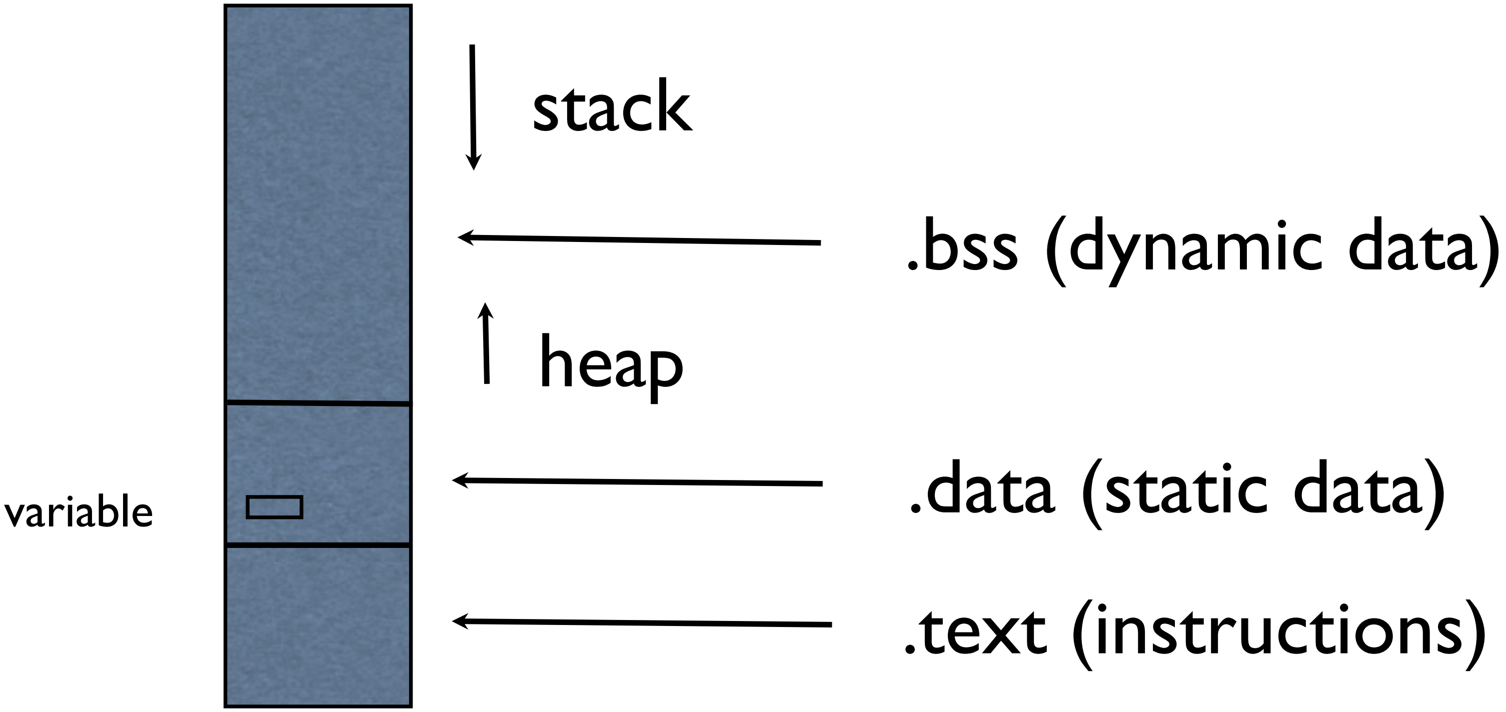
Brief History

- Agreed at Delft 2005
- Discussed at Fairfax 2006, at London 2007
- Las Vegas Compromise (core & more)
- Tokyo Further Compromise 2008
- Finally Agreed Las Vegas 2010
- Most recent draft of “more” is 12-136r2

Coarray Concepts

- A program is treated as if it were replicated at the start of execution, each replication is called an image.
- Each image executes asynchronously.
- A coarray is indicated by trailing [ ].
- A data object without trailing [ ] is local.
- Explicit synchronization statements are used to maintain program correctness.

Memory Basics (Single Image)



stack

.bss (dynamic data)

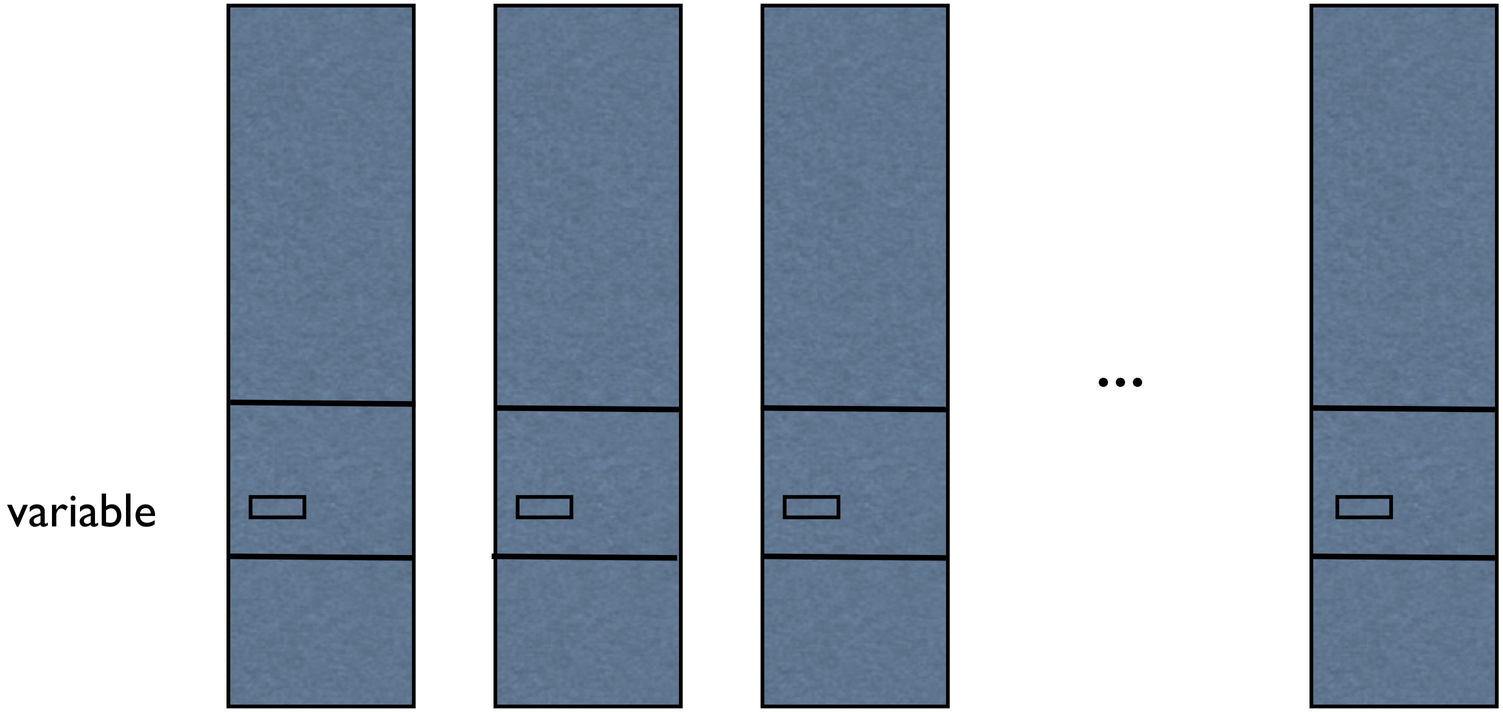
heap

variable

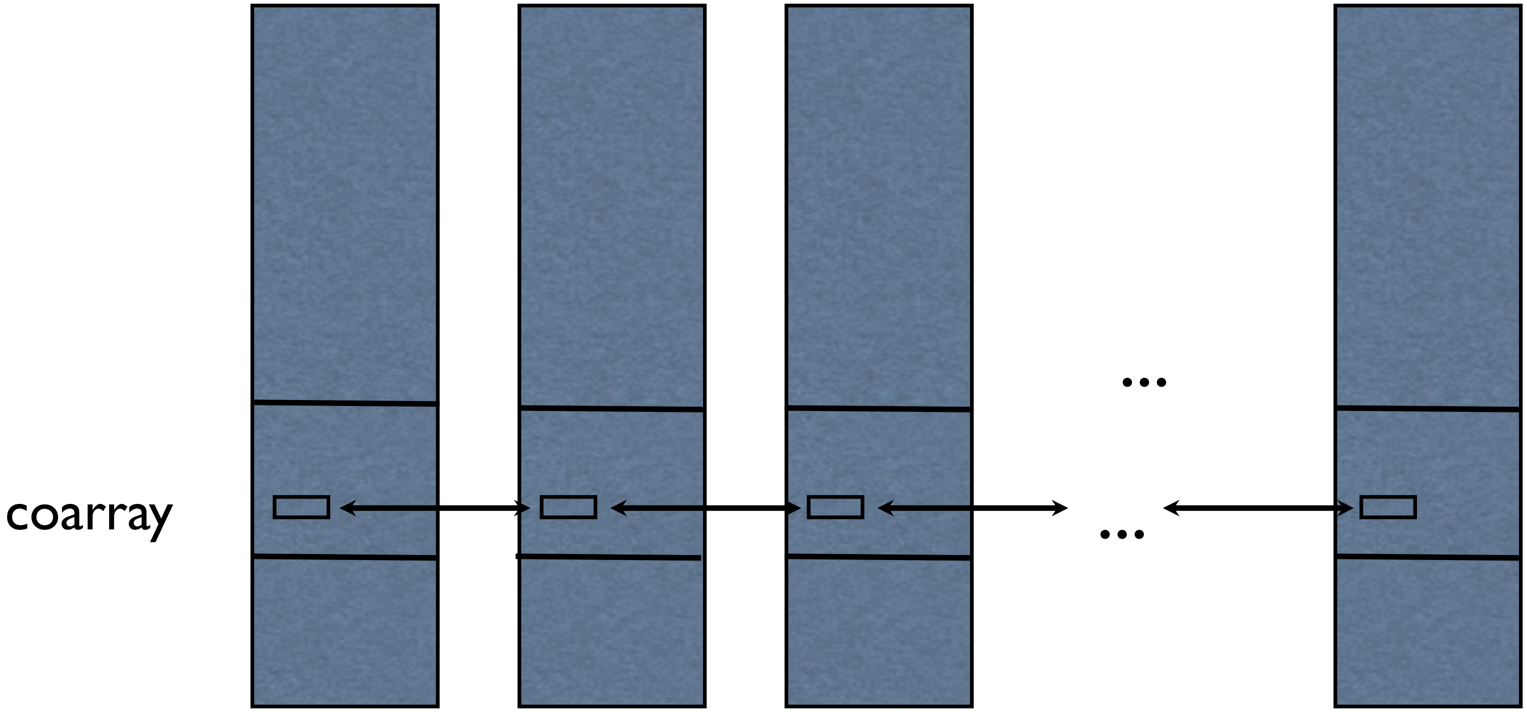
.data (static data)

.text (instructions)

Memory Basics (Multiple Images)



Memory Basics
Coarrays



Core & More Compromise

- Reduce the coarray features in the Fortran 2008 standard (“core”)
- Add some of those features as a TS between revisions of the Fortran standard
- The standards committees are now writing the work list for the TS (“more”)
- The current draft of the work list is 12-136r2

12-136r2

- Teams
- Collective procedures
- Atomics
- Events
- Limited I/O support

Excluded by 12-136r2

- coarray pointers or coscalars
- reduction of OOP restrictions on derived types with coarray components
- asymmetric allocatable variables
- asynchronous copy procedure

Defining Teams

- There is an “all images” team at the start of execution
- An image is a member of only one team at a time (within a nested hierarchy)
- Teams are identified by team variables of type team_type defined in iso_fortran_env
- Further teams are formed by a “split team” or a “form team” statement or intrinsic

Using Teams

- Supply a “with team … end with team” or a “select team … end select team” construct
- Within the construct, image numbers are relative to the specified team
- Collective activities are team-wide
- At the end of the team construct, the images revert to the previous team

Limitations on Teams

- Access to data outside the team is not allowed (how to synchronize?)
- Synchronization is team-wide
- Collective procedures apply team-wide
- Allocatable coarrays allocated during the team execution must be deallocated before the end of the team execution

Collective Procedures

- The currently expected set is
  - co_sum
  - co_min
  - co_max
  - co_broadcast
  - co_reduce

Applicability

- If teams are accepted, collective procedures are synchronous within the current team
- If not, some other means to specify the images partaking in a collective operation will be defined
  - For example, a list of image numbers

Atomic Operations

- Fortran 2008 defines atomic_define and atomic_ref
- Proposed further atomic operations are:
  - atomic_cas (compare and swap)
  - atomic_add, _and, _or, _xor, _swap
  - atomic_ax (masked xor)

Events

- The original notify/query pair morphed into an event variable feature
- Event variables (a standard-defined type?)
- Define the operations:
  - post
  - query
  - wait

Limited Parallel I/O

- Opening a file for parallel I/O would
  - if teams are accepted, have a team= specifier on the open list
  - otherwise, have an images= specifier
  - synchronizes the team/images
  - be opened by all members of a team/ images

More Parallel I/O

- Parallel I/O will be for direct access
- Explicitly closing the file is optional
    - if explicitly closed, all images must close the file and a synchronization occurs
- Use a flush statement to make data available to other images

Coarray Status

Dan Nagle
Thank you