

Further Coarrays

Dan Nagle

Welcome

Core and More Compromise

- The entire coarray feature set originally proposed was divided into a *core* functionality subset and a *more* features subset (08-131r1)
- The *core* functionality is in Fortran 2008
- The *more* features will be decided within the next several months
- The starting point for current work is 11-256r2
 - With contributions from Rice U CAF, Germany

Background Documents

- John Reid's Summary N1824
- Bill Long's Draft of the TS N1858 = 11-176
 - Pre Garching
- Metcalf, Reid, & Cohen, Modern Fortran Explained
 - http://www.amazon.com/Explained-Numerical-Mathematics-Scientific-Computation/dp/0199601429/ref=sr_1_1?s=books&ie=UTF8&qid=1326999081&sr=1-1

Basic Ideas

- Collective functions
 - Changed between 08-131r1 and 11-256r2
- Teams
 - Original team proposal versus Rice U proposal
- Notify/Query – How Much Like Events ?
- Parallel I/O – Features and Limitations ?
- Global Data Structures – coscalars or pointers ?
- Atomic Operations

Documents with Suggestions

- 08-131r1 (the original list of deferred features)
- 10-166 (Bill Long's 2010 draft of the Further Coarrays TS based on 08-131r1)
- N1835 (John Reid's 2010 summary list)
- N1856 (Rice U CAF group 2011 list)
- N1883 Comments on list (post Garching)
- 11-256r2 (John Reid's 2011 summary list) **

The Size of the Coarray Additions

- Competing desires:
 - to manage the workload on compiler suppliers
 - to provide useful tools to applications programmers as quickly as possible
- What's missing ?
 - Must judge without extensive application programmer experience with f08 coarrays
- “Useful” is application-dependent

Collective Procedures

- Which ones ?
 - Similar to the existing reduction intrinsics ?
 - Similar to MPI reduction procedures ?
- Synchronous or Asynchronous ?
 - If synchronous, what affect on performance ?
 - If asynchronous, how ?
 - Signal completion via event variables ?

The Original Set

- From 08-131r1 (see also 07-007r3)
 - co_all, co_any, co_count
 - co_maxloc, co_maxval
 - co_minloc, co_minval
 - co_product, co_sum
 - co_findloc ?
- Most Had Arg Lists (source, result [, team])
- Synchronous (Image Control Statements)

The Current Favorites

- As per 11-256r2
- co_bcast, co_max, co_min, co_reduce, co_sum
- asynchronous ?
- copy_async ?
- max-and-copy, min-and-copy ?
- apply to non-coarrays ?

Teams

- Teams were originally lists of image indexes
- Teams might be subdivided from a parent team via the scheme proposed by the Rice U CAF group
- What should teams do ? (synchronize and ...)
 - label blocks ? (the **with team** construct)
 - procedures ? (intrinsic or external ?)
 - I/O ? (**team=** on control list)

Parallel Input/Output

- Sequential Access or Direct Access or Both ?
 - If sequential, control record order ?
 - If direct, control record access ?
 - how to manage races ?
- Linked with Teams how ?
 - **team=** on control list ?

Notify/Query

- As is (identified by image number), or as first class events (identified by event variables) ?
- Synchronous or queued ?
- Applied to a team or global ?
- What about libraries ?
 - Do library writers prefer event variables ?

If First Class Events, Details ?

- Apply to collective intrinsics ?
 - copy_async: Rice U CAF proposal has 3 events: source ready, destination ready, copy ready
- Apply to asynchronous I/O ?
- Apply to any other wait operation ?
- Do events queue or signal ?
 - If they queue, how to distinguish individual events ?

Global Data Structures

- Coscalars (Germany)
- Copointers (Rice U CAF)
 - and cotarget
- Pointer attribute allowed on coarrays (IBM)

Any other business

- Asynchronous copy (Rice U CAF)
- More atomic operations (Cray)
 - compare-and-swap is likely
 - the rest of Cray's current set ?
 - add, and, or, xor
 - and corresponding fetch-and-*op* versions
- Support for irregular grids ?
 - affinity for teams ?

The Process Needs Input

- In February, set the initial feature list to start the discussion for the International meeting
 - might be viewed as the opening US position
- The June International meeting at IBM Toronto Labs will set the work list
- What are your concerns ?
- Discussion at <http://sea.ucar.edu/forums/updates-fortran-2008>

Further Coarrays

Dan Nagle

Thank you