

Further coarray features

John Reid

In 2008, WG5 decided to move some coarray features from Fortran 2008 to a Technical Specification (TS).

Last year, WG5 decided that while the overall size of the addition should not be increased, the choice of features should be reviewed.

This choice, in the form of requirements, was made during the June meeting, and will be explained in this talk.

BCS Fortran Specialist Group
London, 27 September 2012.

Teams

Needed for independent computations on subsets of images.

Code that has been written and tested on whole machine should run on a team.

Image indices need to be relative to team.

Collective activities, including syncs, need to be relative to team.

NB Old feature did not do this.

Teams (cont)

- Type for identifying team.
- Possible to split a team into subteams.
- When an image executes a statement it shall be a member of exactly one team.
- Construct for changing the team, involving syncing all images of the teams at beginning and end of construct.
- Can find index of executing image in ancestor team.
- Allocatable coarray allocated in team construct must be deallocated in it (retains symmetric memory within each team).
- Standard input only on root image 1.

Collectives

- SUM, MAX, MIN across images of the team, plus one based on user-written binary operation.
- Invoked by same statement on all images of the team.
- No sync on entry or return, but syncs inside.
- Result provide to one image or all images of team.
- Collective to broadcast value to all images of team.

Events

One-sided ordering of execution segments sometimes needed.

For example, suppose image I executes segments I1, I2,... and image J executes segments J1, J2,... there might be a need for I1 to precede J2 without the need for J1 to precede I2.

NOTIFY/QUERY did this but relied on counts of number of executions.

Tagged events give much clearer associations. Important, for a library routine, for example.

There will be a data type for tags. Each event will be identified by a data item of this type, accessible on all images.

Mechanisms will be provided to post and test an event, and wait for an event to be posted.

More atomics

- Compare and swap
- Integer add
- Bitwise and
- Bitwise or
- Bitwise xor

Parallel Input/Output

A feature for parallel I/O was removed in 2008.

It was decided not to restore this or design a new feature.

This makes a worthwhile reduction in the size of the TS, which should mean that its construction is easier, quicker, and less controversial than it would have been.