



OpenCoarrays: A Coarray Fortran API and ABI

A. Fanfarillo, S. Filippone
D. Nagle, D. Rouson

fanfarillo@ing.uniroma2.it, salvatore.filippone@cranfield.ac.uk
dnagle@ucar.edu, damian@sourceryinstitute.org



What is coarray Fortran? Why use it?

- Coarray Fortran (CAF) is a collection of features in the Fortran 2008 standard with expanded support in Fortran 2015.
- CAF enables programmers to express parallel algorithms without tying an algorithm's implementation to a specific, lower level communication library (e.g., MPI) or set of directives (e.g., OpenMP).
- A CAF executable launches one or more images (replicas of itself) in a Single Program Multiple Data (SPMD) fashion.
- Each image manages its own portion of what is logically a Partitioned Global Address Space (PGAS).
- The language rules facilitate one-sided communication: one image may access a second image's coarray without the second image's involvement.
- A coarray may be a scalar or an array in statically or dynamically allocated memory and may be of intrinsic or derived type.

Fortran is now a PGAS language that facilitates SPMD parallel programming without direct reference to anything outside the language!

OpenCoarrays API and ABI

- With an OpenCoarrays-aware compiler, such as GCC 5.1 or later, OpenCoarrays functions as an application binary interface (ABI) that the compiler calls for parallel communication, computation and synchronization.
- OpenCoarrays also provides an application programmer interface (API) that provides users access to CAF features even with non-CAF compilers.
- OpenCoarrays provides a compiler wrapper ("caf") that can translate the image number tally program at the lower left into a program with API calls that non-CAF compilers can process.

Users of GCC 4.9 can compile some CAF codes via the "caf" wrapper, which will eventually support non-GCC compilers as well.

Fortran 2008 CAF at work

```
real, dimension(10), codimension[*] :: x, y
integer :: num_img, me
num_img = num_images()
me = this_image()
x(2) = x(3)[7] ! get value from image 7
x(6)[4] = x(1) ! put value on image 4
x(:)[2] = y(:) ! put array on image 2
sync all
x(1:10:2) = y(1:10:2)[4] ! strided get from image 4
```

Every transfer is performed using the regular Fortran array syntax. The basic transfer coarray operations are Put, Get, Strided Put and Strided Get. Synchronization primitives are required to maintain program correctness.

The Cray and Intel compilers support Fortran 2008 CAF. GCC 5.1 or later use OpenCoarrays to support most Fortran 2008 CAF.

Fortran 2015 Collective Subroutines

- Fortran 2015 adds collective subroutines for efficient global communication and computation: `co_sum`, `co_min`, `co_max`, and `co_broadcast` perform global summation, minimum, and maximum reductions, and broadcasts.
- The `co_reduce` collective subroutine allows users to extend the set of collective subroutines by writing a user-defined, functionally pure, binary operator for use in a global reduction.

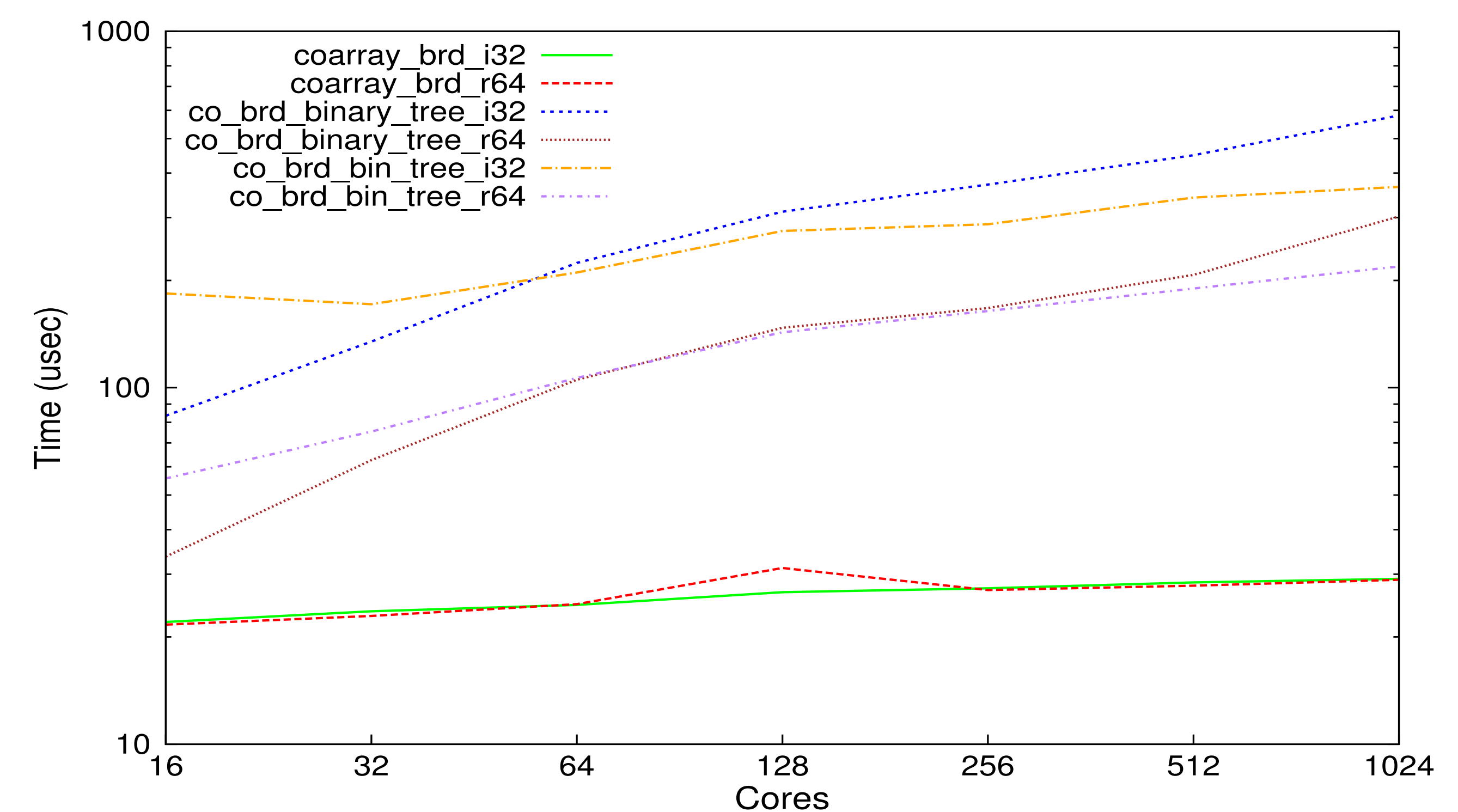
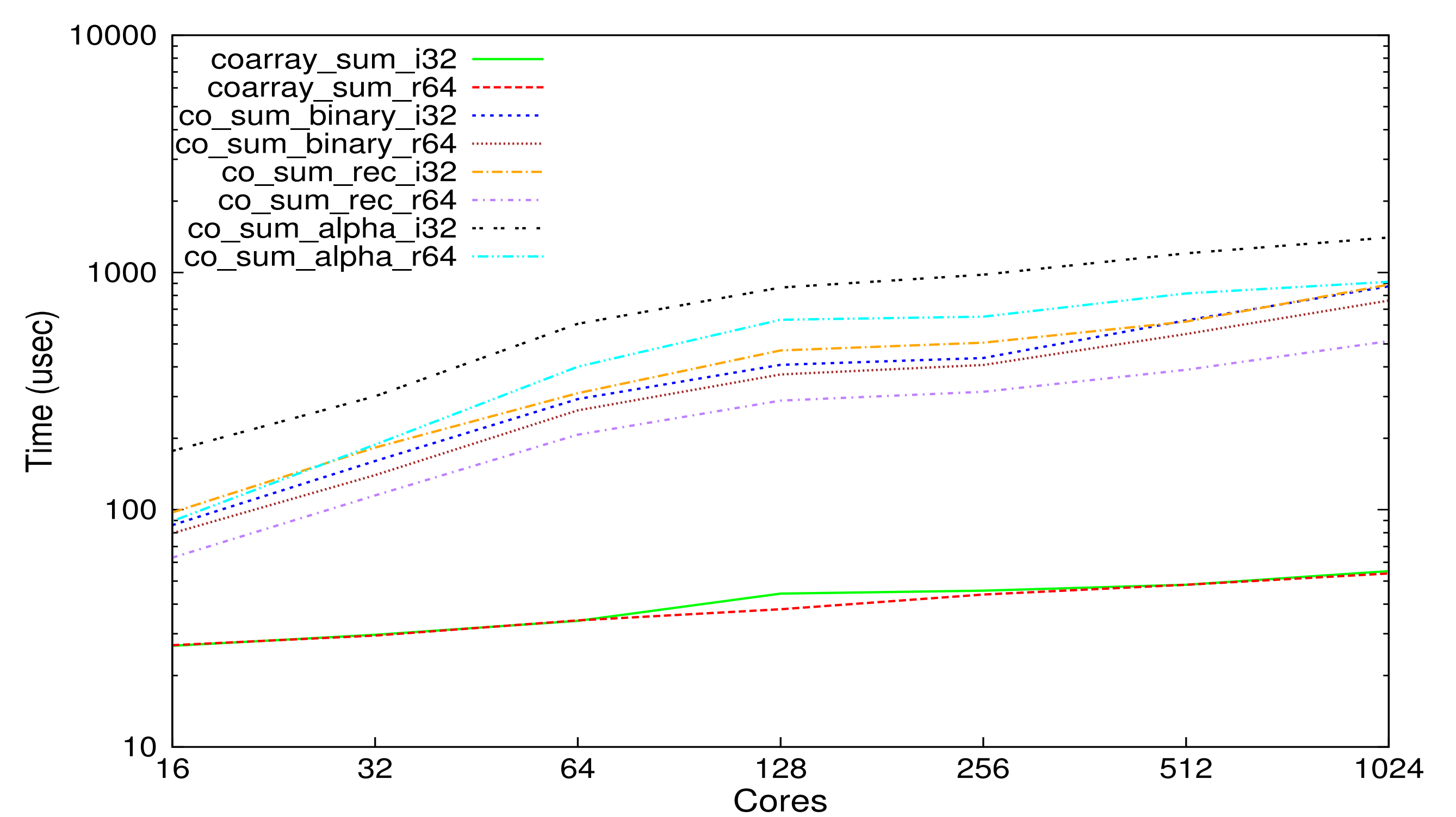
```
integer :: me,i
me=this_image()
sync all
call co_sum(me,result_image=1)
if (this_image() == 1 .and. me /= sum( [ (i,i=1,num_images()) ] )) error stop
print *, "image ",this_image(), " received the correct tally "
```

Error termination halts all images.

Goal

To compare the performance of the intrinsic collectives to analogous, reasonably sophisticated collective subroutines that were manually coded using only Fortran 2008 features.

Intrinsic vs. User-Defined Collectives



Execution times for sum reductions (top) and broadcasts (bottom). From top to bottom in the legend: intrinsics (red and green), binary tree, recursive doubling (binomial tree), and alpha tree (alpha=0.4). Platform: NERSC Hopper Cray XE6, peak performance of 1.28 Petaflops/sec, 153,216 compute cores, 212 Terabytes of memory, and 2 Petabytes of disk.

Summary and Conclusions

Fortran 2008 CAF features adds a PGAS capability to the language PGAS facilitates SPMD-style programming within the language. Fortran 2015 adds intrinsic collective subroutines that greatly outperform even reasonably sophisticated collectives that a user might write in Fortran 2008. The OpenCoarrays compiler wrapper translates a subset of CAF features into calls to the OpenCoarrays API for users of non-CAF compilers. Such users invoke the intrinsic collective subroutines through C bindings in the API, whereas OpenCoarrays-aware CAF compilers directly invoke the OpenCoarrays ABI.