

HPC Challenge Benchmarks in CAF2.0

Department of Computer Science, Rice University
<http://caf.rice.edu>



HPC Challenge Benchmarks

- Measure performance of HPC systems
 - ❖ processor performance
 - ❖ memory subsystem performance
 - ❖ system interconnect performance

- Benchmarks
 - ❖ RandomAccess
 - ❖ EP STREAM Triad
 - ❖ Global FFT
 - ❖ Global HPL
 - ❖ PTRANS
 - ❖ DGEMM
 - ❖ b_eff Latency/
Bandwidth

RandomAccess

- hypercube-based routing of updates
- overlap computation with synchronization

```
event, allocatable :: delivered(:)[*], received(:)[*]
integer(8), allocatable :: fwd(:, :, :)[*]
...
do i = world_logsize-1, 0, -1
    ...
    call split(ret(:, last), retsizes(last), &
               ret(:, current), retsizes(current), &
               fwd(1:, out, i), fwd(0, out, i), bufsize, dist)

    if (i < world_logsize-1) then
        event_wait(delivered(i+1))
        call split(fwd(1:, in, i+1), fwd(0, in, i+1), &
                   ret(:, current), retsizes(current), &
                   fwd(1:, out, i), fwd(0, out, i), bufsize, dist)
        event_notify(received(i+1) [from])
    endif

    count = fwd(0, out, i)
    event_wait(received(i))
    fwd(0:count, in, i) [partner] = fwd(0:count, out, i)
    event_notify(delivered(i) [partner])
    ...
end do
```

EP STREAM Triad

- outlined STREAM triad with explicit shape array declarations for performance

```
double precision, allocatable :: a(:)[*]
double precision, allocatable :: b(:)[*], c(:)[*]
! allocate with the default team
allocate(a(ndim) [], b(ndim) [], c(ndim) [])
...
do round = 1, rounds
    do j = 1, rep
        call triad(a, b, c, n, scalar)
    end do
    call team_barrier()
end do
...
subroutine triad(a, b, c, n, scalar)
    double precision a(n), b(n), c(n), scalar
    a = b + scalar * c
end subroutine triad
```

Global FFT

- all-to-all collective in permutation step
- asynchronous copy in global FFT rounds

```
complex, allocatable :: c(:, 2)[*]
event, allocatable :: ready(:)[*]
event, allocatable :: prefetch(:, :)[*], copied(:, :)[*]
...
do l = lcomm, levels
    ....
    event_notify(ready(l-lcomm) [partner])
    event_wait(ready(l-lcomm))
    ! prefetch blocks
    do outer = 0, (n_local_size/2)-1, blksize
        copy_async(buf(lo:hi), c(lo:hi, last) [partner], &
                  prefetch(outer/blksize, l-lcomm))
    end do
    do outer = 0, (n_local_size/2)-1, blksize
        event_wait(prefetch(outer/blksize, l-lcomm))
        ! perform computation
        ...
        ! send result back to partner
        copy_async(c(lo:hi, curr) [partner], buf(lo:hi), &
                  copied(outer/blksize, l-lcomm))
    end do
    event_notify(ready(l-lcomm) [partner])
    event_wait(ready(l-lcomm))
end do
```

Global HPL

- block-cyclic data distribution
- row/column team-based communication
- asynchronous broadcast of panels

```
type(paneltype) :: panels(1:NUMPANELS)
event, allocatable :: delivered(:)[*]

allocate(delivered(1:NUMPANELS) [])
event_init(delivered, NUMPANELS)
...
do j = pp, PROBLEMSIZE - 1, BLKSIZE
    cp = j / BLKSIZE + 1
    cp = mod(cp - 1, 2) + 1
    ...
    event_wait(delivered(3-cp))
    ...
    if (mycol == cproc) then
        ...
        if (ncol > 0) then
            ! update part of the trailing matrix
            if (NPCOL == 1) call update(m, n, BLKSIZE, ncol, 3-cp)
            if (NPCOL /= 1) call update(m, n, 0, ncol, 3-cp)
        end if
        call fact(m, n, cp) ! factorize the next panel
        ...
    end if
    ...
    call team_broadcast_async(panels(cp)%buff(1:ub), &
                              panels(cp)%info(8), &
                              delivered(cp))

    ! update rest of the trailing matrix
    if (nn-ncol > 0) call update(m, n, col, nn-ncol, 3 - cp)
    ...
end do
```

Benchmarks	SLOC	# of cores	RandomAccess (GUP/s)	STREAM Triad (TByte/s)	Global FFT (GFlop/s)	Global HPL (TFlop/s)
RandomAccess	409	64	0.08	0.14	3.66	0.36
STREAM Triad	58	256	0.24	0.54	11.7	1.36
Global FFT	439	1024	0.69	2.18	38.2	4.99
Global HPL	786	4096	2.01	8.73	125	18.3



Development of Coarray Fortran 2.0 is supported by the Department of Energy's Office of Science under cooperative agreements DE-FC02-07ER25800 and DE-FC02-06ER25754. Contributors: John Mellor-Crummey (PI), Laksono Adhianto, Guohua Jin, Mark Krentel, Karthik Murthy, Dung Nguyen, William N. Scherer III, Scott K. Warren, Chaoran Yang