

CICE and POP2 Boundary-exchange Performance Optimization (High-resolution and Large-scale Run)

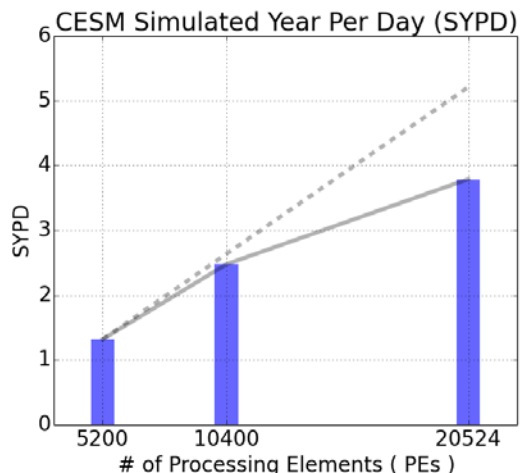
June 19, 2018

Youngsung Kim and John M. Dennis

CESM Workshop 2018

CESM high-resolution simulation

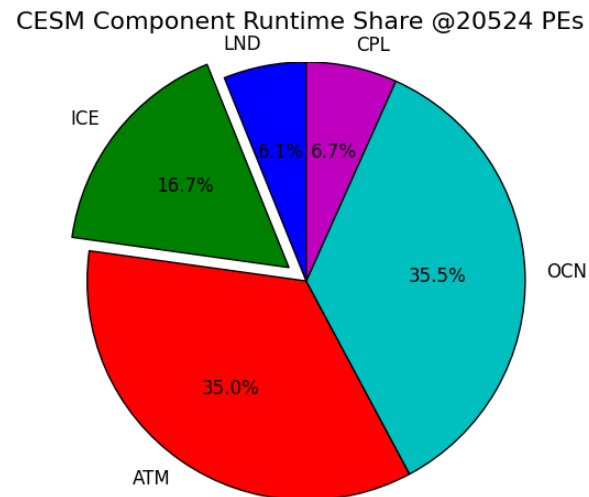
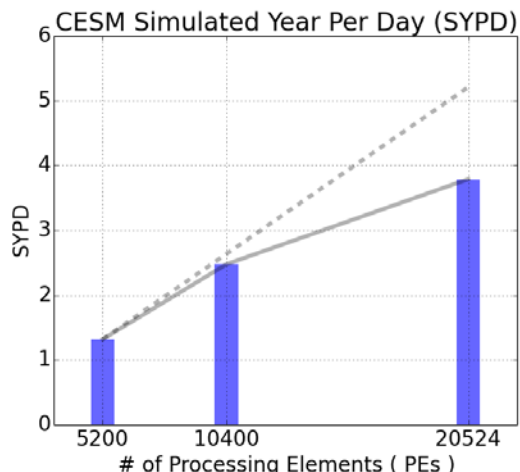
- Adding more resource is getting less efficient at high-resolution simulation.



Data from 1.3.beta17(sehires19.opt), Compset: B1850C5CN, Res.: ne120_t12

CESM high-resolution simulation

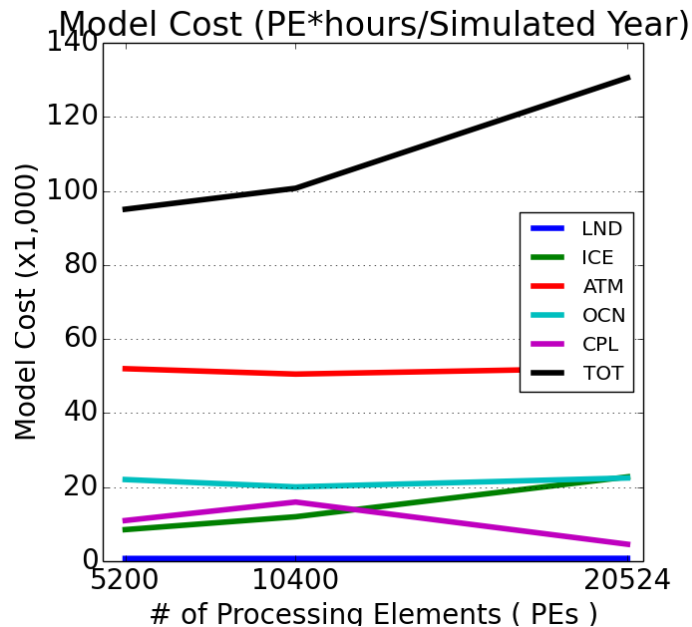
- Adding more resource is getting less efficient at high-resolution simulation.
- CICE does not take the largest chunk of total runtime. But...



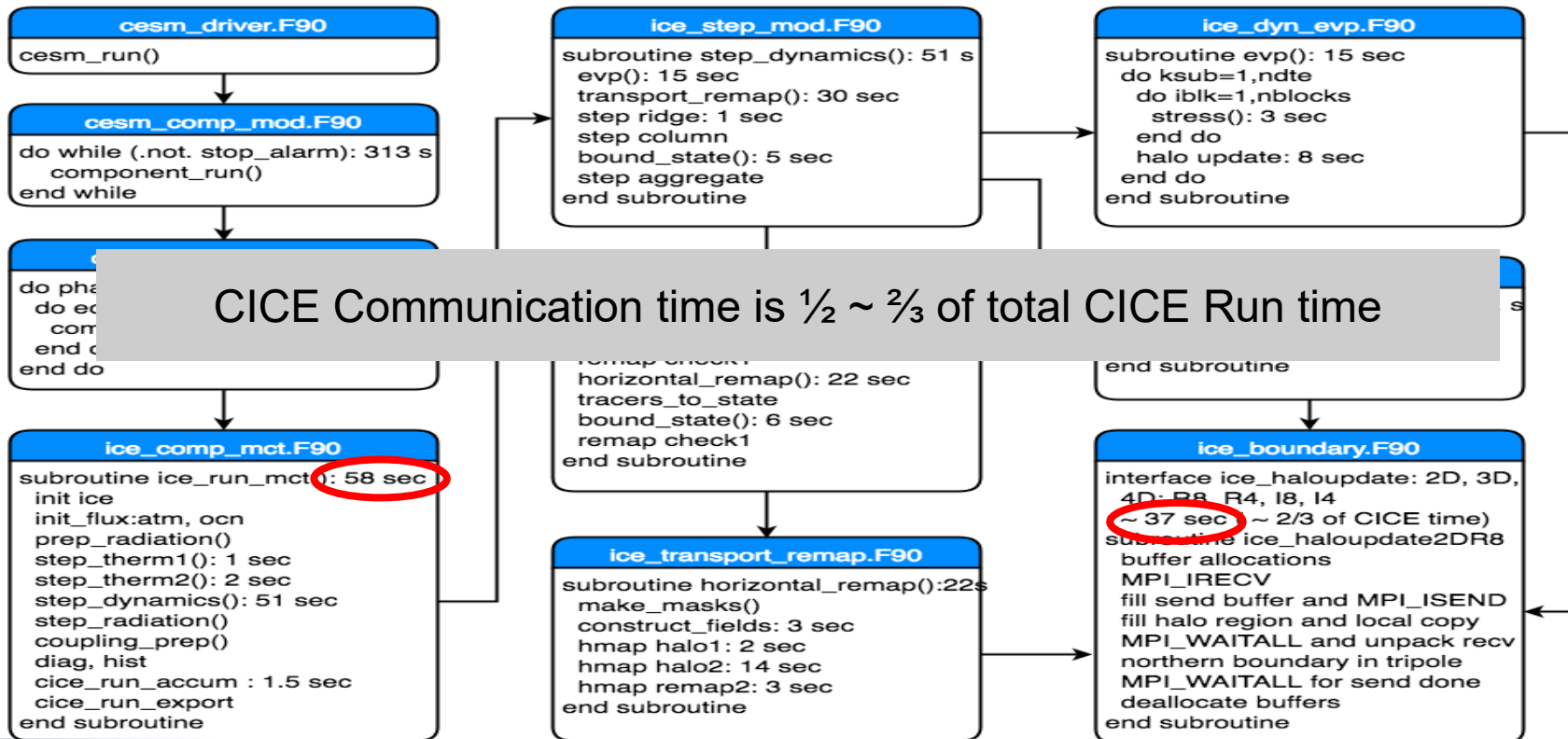
Data from 1.3.beta17(sehires19.opt), Compset: B1850C5CN, Res.: ne120_t12

CICE @ high-resolution

- CICE is increasingly expensive at high-resolution run



CICE in CESM_1_3_beta_17



Tripole Grid @ CICE

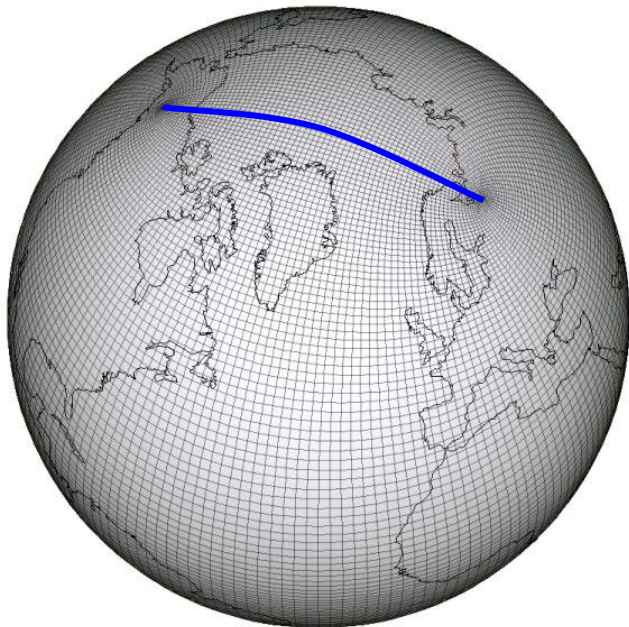
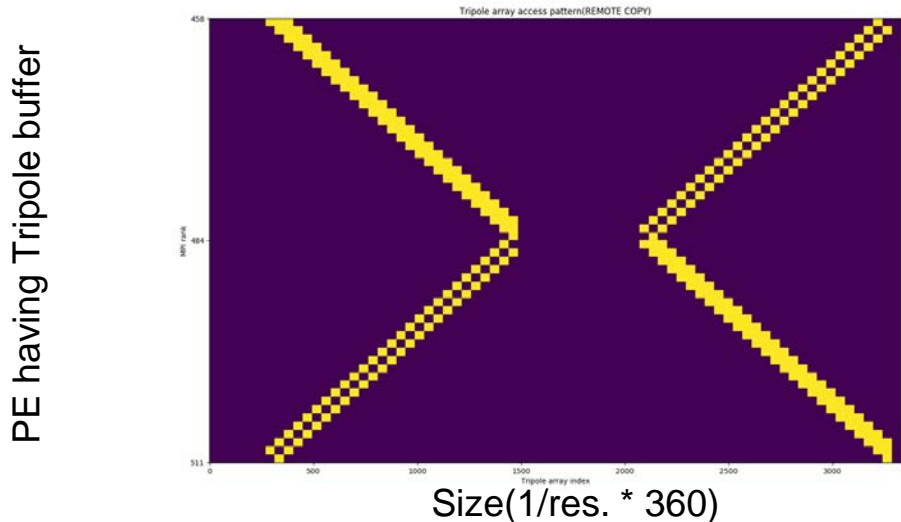


Image source: CESM User Guide
(http://www.cesm.ucar.edu/models/cesm1.0/cesm/cesm_doc_1_0_4/x42.html)

- Tripole Grid
 - Two northern poles on land
 - To avoid numerical instability
- Domain Decomposition
 - Assign a block of cells to a PE
 - Many strategies exist
- Boundary Exchange
 - Data exchanges across boundary of domains
 - In most cases, THE most time consuming reason at high-res. run.
- Tripole Buffer
 - For special handling of cells from one north pole to the other.
 - The number of elements in tripole buffer is $1/\text{resolution} * 360$

Tripole Buffer @ CICE

- Most of elements in the Tripole buffers are wasted
- At higher resolution, the size of wasted memory increased $\sim O(1/\text{resolution}^2)$
- Overall performance is limited by the PEs having the Tripole buffers



Optimizations

1. Reduce the size of Tripole buffer memory allocations

- Allocate only parts that are actually used
- Applies to 3 and 4 dimensional halo updates as they are the most time-consuming routines



1. Move Tripole buffer index calculations to initialization step

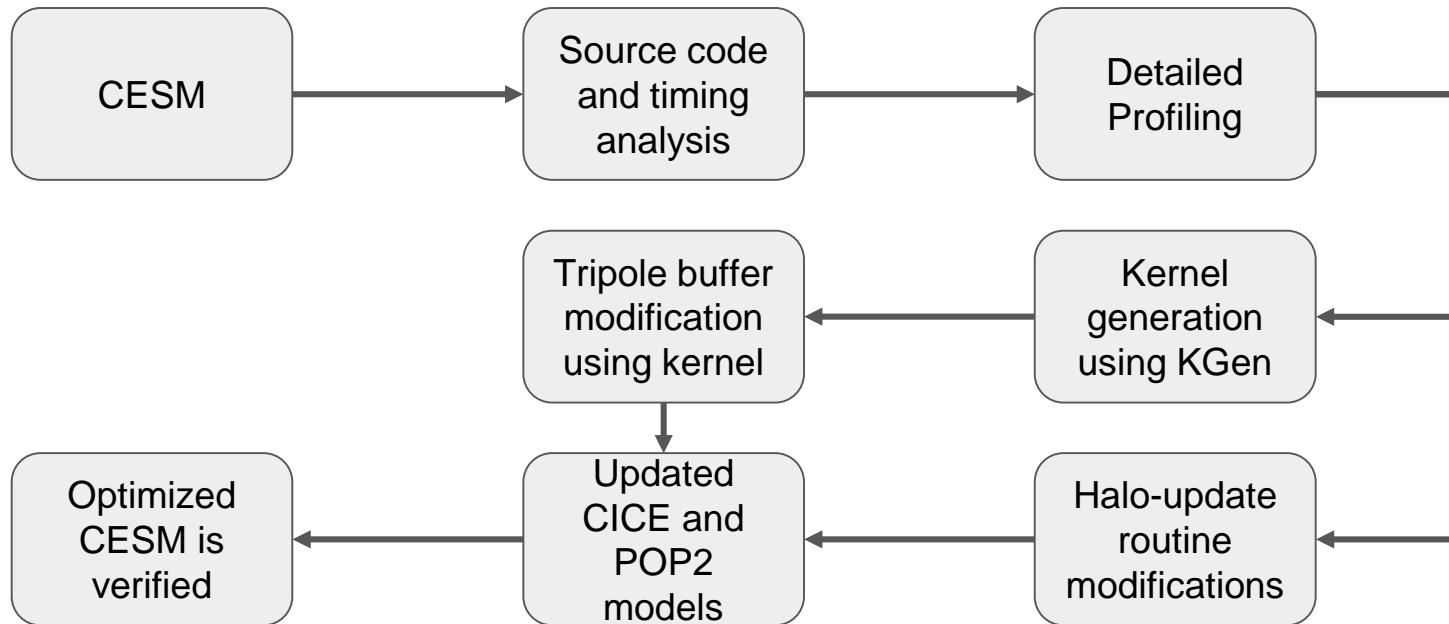
- Tripole buffer index can be pre-calculated once decomposition strategy is selected.
- Isolate complex index calculations in a subroutine

2. Further reduce memory usage by reducing the size of receive and send buffer

- Pre-calculate the indices that are actually used

Optimization Workflow

- A kernel helped to reduce code modification time significantly.



Minimal Source Code Modifications

- Maintained original halo-update code structure
- Modified code produces **bit-for-bit** results

```
if (jSrc <= halo%tripoleRows .and. &
    jSrc>0 .and. jDst>0) then

    do l=1,nt
    do k=1,nz
        array(iDst,jDst,k,l,dstBlock) = isign*   &
            bufTripole(iSrc,jSrc,k,l)
    end do
    end do
endif
```

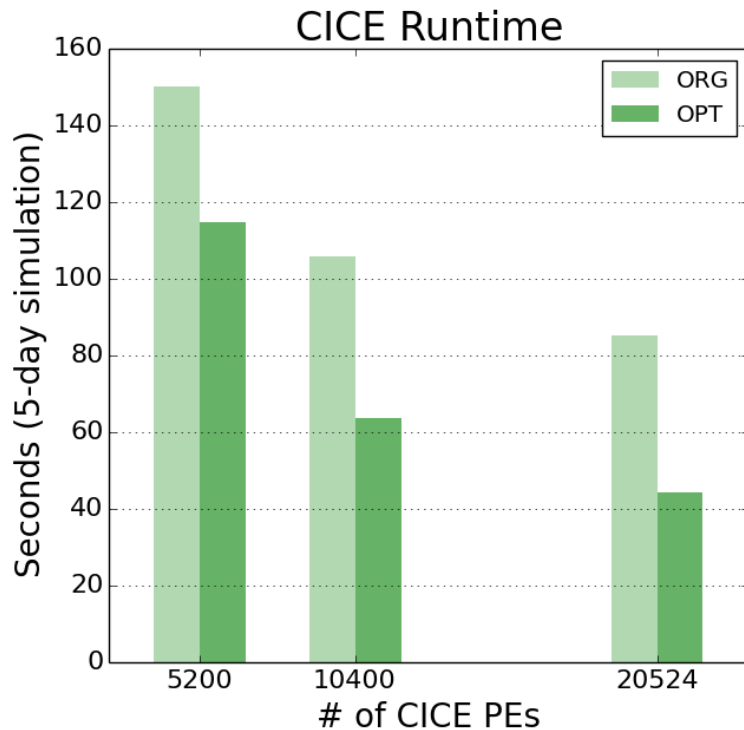


```
if (jSrc <= halo%tripoleRows .and. &
    jSrc>0 .and. jDst>0) then

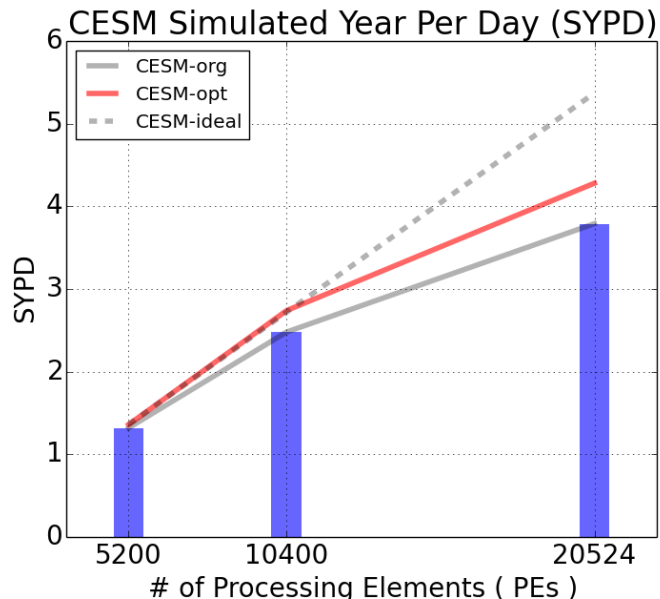
    iSrc = iSrc + sign(halo%offsetTripoleBuf, &
        nxGlobal/2-iSrc)

    do l=1,nt
    do k=1,nz
        array(iDst,jDst,k,l,dstBlock) = isign*   &
            bufTripole(iSrc,jSrc,k,l)
    end do
    end do
endif
```

CICE Runtime is reduced upto 48 % @20524 PEs

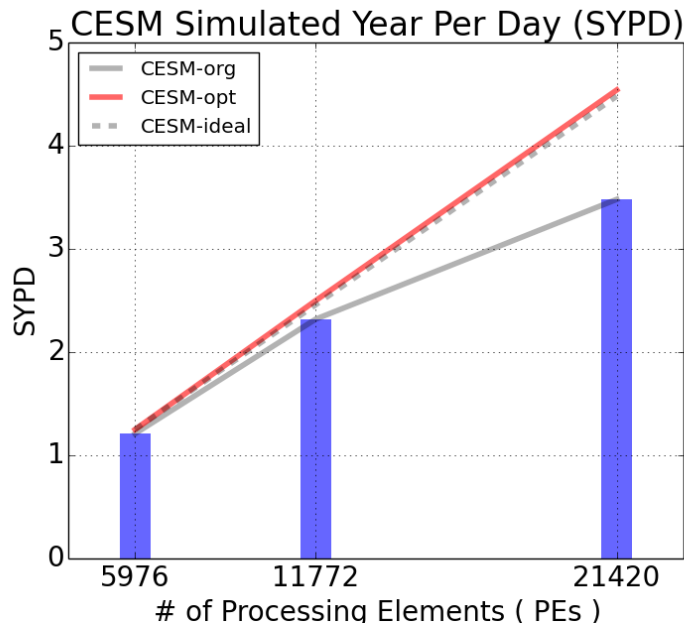


CESM SYPD is increased from 3.79 to 4.28 @5-day run
and 4.29 to 5.21 @1-month run



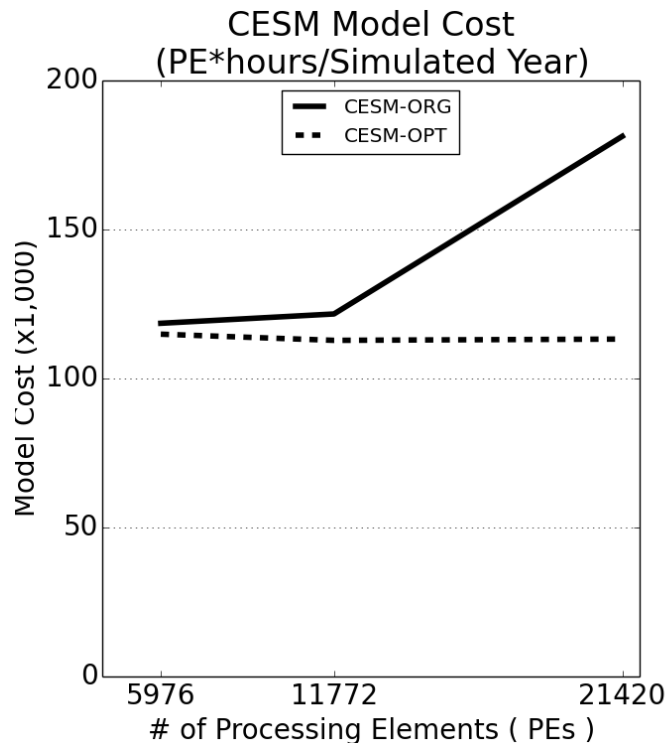
Data from 1.3.beta17(sehires19.opt), Compset: B1850C5CN, Res.: ne120_t12

POP2 has Tripole Buffer too. After optimizing both of CICE and POP2, CESM SYPD is increased near ideally



Data from 1.3.beta17, Compset: B1850C5CN, Res.: ne120_t12

CESM model cost is flat across different # of PEs



Summary

- CICE and POP2 Halo-update routines are optimized
 - The size of Tripole buffer is reduced by ~ 20x
 - From the optimization, a large-scale CICE runtime reduces by 48.1% and CESM SYPD is increased from 4.29 to 5.21 @1-month run.
 - POP2 runtime also significantly reduce similar to a large-scale CICE run
- Please try to avoid unnecessary memory allocation
- The optimization is included in recent releases of CICE5, POP2, and CESM2
- Future work
 - Reducing the number of halo-updates may improve performance further
 - Applying this optimization to models that use this type of tripole buffer may be beneficial

Thank You!

Email: youngsung@ucar.edu and
dennis@ucar.edu