ESMF Offline Regrid Weight Generator Performance Comparison with SCRIP

Peggy Li

December 7, 2010

Objective

In the latest ESMF internal release, ESMF 5.1.0, a new application ESMF_RegridWeightGen is provided for generating regridding weights. This application reads in two grid files and outputs weights for interpolation between the two grids. The input and output files are in NetCDF format. The grid files are in the same format as is used as input to SCRIP and the weight file is the same format as is output by SCRIP [1]. SCRIP (Spherical Coordinate Remapping and Interpolation Package) is a software package developed by Los Alamos National Laboratory. It is used to generate interpolation weights for remapping fields from one grid to another in spherical geometry. SCRIP is a sequential program and ESMF offline regrid weight generator runs in parallel on multiple processors.

In this benchmark, we compare the timings and the quality of the weights generated by these two approaches. The software versions used in this benchmark are ESMF_5_2_0_beta_snapshot_07 and SCRIP 1.4. The benchmark programs were run on the Cray XT4 (jaguar) at Oak Ridge National Lab.

Benchmark Grids

As depicted in table 1, we used four pairs of input grids for the benchmarks. Ne40np4 is a cubed sphere grid provided by Mark Taylor of Sandia National Lab. Each plane of the cube contains 180 x 180 grid cells. The grid cells are combination of quadruples, triangles and pentagons. fv0.47x0.63 and fv1.9x2.5 are high-resolution and medium-resolution CAM finite volume grids used by CCSM. gx1v6 and tx1v2 are medium-resolution and high-resolution POP grids used by CCSM. The 1/8th degree and 1/10th degree regularly rectangular grids were created by ESMF for testing purpose.

Table 1. Configurations of the benchmark grids

Grid #1	Type	Size	Grid #2	Type	Size
fv0.47x0.63	CAM Finite volume	576x384	**ne60np4**	cubed sphere	180x180x6
fv1.9x2.5	CAM Finite volume	144x96	**gx1v6**	POP DP masked	320x384
fv0.47x0.63	CAM Finite volume	576x384	**tx1v2**	POP Tripole masked	3600x2400
1/8 degree	regularly rectangular	2880x1440	**1/10 degree**	regularly rectangular	3600x1800

ESMF supports bilinear, first-order conservative and higher-order patch recovery interpolation methods. SCRIP supports bilinear, first and second-order conservative, bicubic and distance-weight average remappings. In this benchmark, we generated bilinear and conservative interpolation weights using both ESMF and SCRIP. For case #2, we also generated weights using ESMF patch interpolation. It takes very long to generate the conservative weights for some of the large grids, such as the tx1v2 grid, the 1/8th degree and 1/10th degree grids using SCRIP. As a consequence, some of the timings for SCRIP conservative remapping are missing.

Timing Results

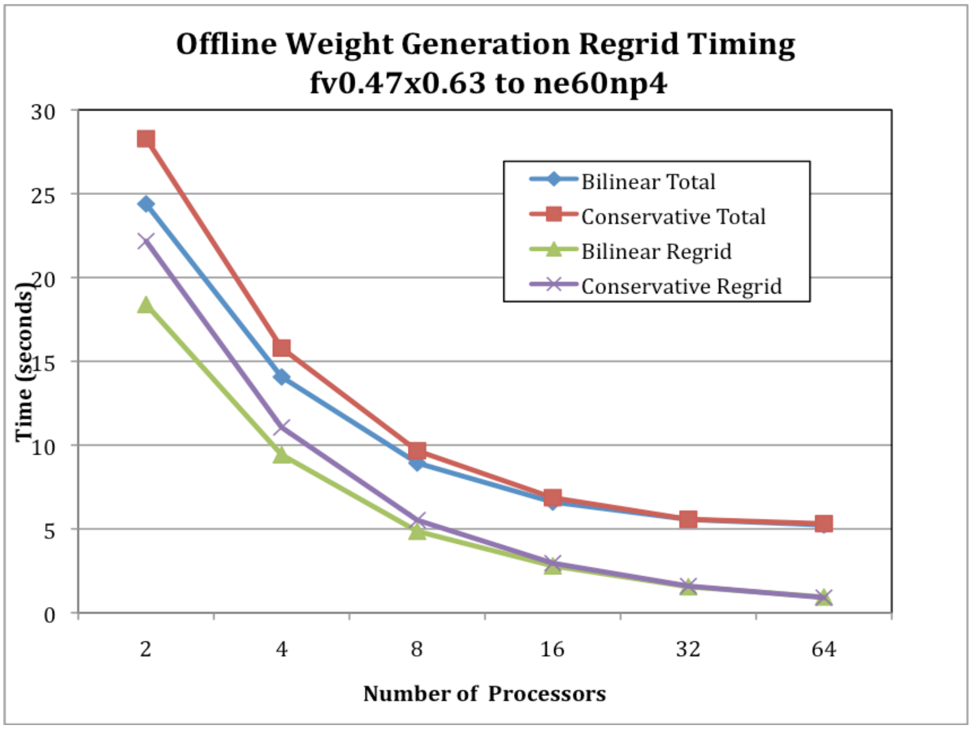
I measured the time to run ESMF_FieldRegridStore() and the total time to run the offline regrid weight generator application. The latter time includes the time to read in the source and the destination grid files, to construct the ESMF_Mesh or ESMF_Grid object, and to write out the SCRIP format weight file. For a cubed sphere grid, it also includes the time to convert the SCRIP format grid file into an ESMF unstructured grid file. We used NetCDF 3.6 Fortran 90 API to read the input grid files and write out the weight files. The IO part is sequential and thus not scalable with the increase of the number of processors.

SCRIP may be set up to run the remapping in both directions. In other words, we can generate the remapping weights from grid 1 to grid 2 and vice versa at the same time. However, ESMF offline regrid weight generator can only perform interpolation in one direction. In this benchmark, we only generate one set of remapping weights at a time using SCRIP.

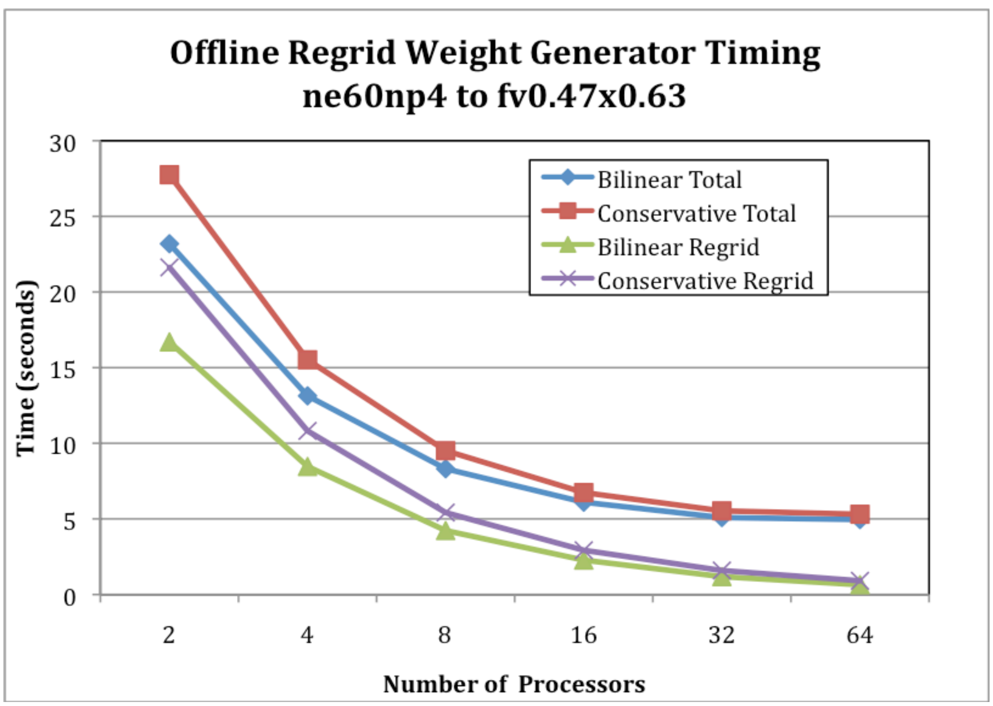
Case 1: fv0.47x0.63 and ne60np4

Figure 1 shows the ESMF FieldRegridStore() and ESMF offline regrid total times using either bilinear interpolation or conservative interpolation from 2 PEs up to 64 PEs. Contrary to SCRIP, the bilinear and the conservative regridding in ESMF takes comparable time. Conservative regridding is a bit slower with small number of processors (<= 8) but scales better. Hence, the timing for both regridding methods is almost the same with 16 or more processors. The difference between the FieldRegridStore() time and the total time remains constant regardless of the number of processors used. The regrid weight generation times from fv0.47x0.63 to ne60np4 and from ne60np4 to fv0.47x0.63 are similar, so is the scalability.

Figure 2 compares the timing between the ESMF offline regrid program and SCRIP. Note the timing for SCRIP is measured on one PE. The ESMF timing is the total time as explained above. SCRIP does not support bilinear interpolation for an unstructured source grid. Therefore, there is only conservative timing from ne60np4 to fv0.47x0.63. It is interesting to see that SCRIP bilinear remapping is much faster then ESMF bilinear interpolation and vice versa for the conservative remapping.

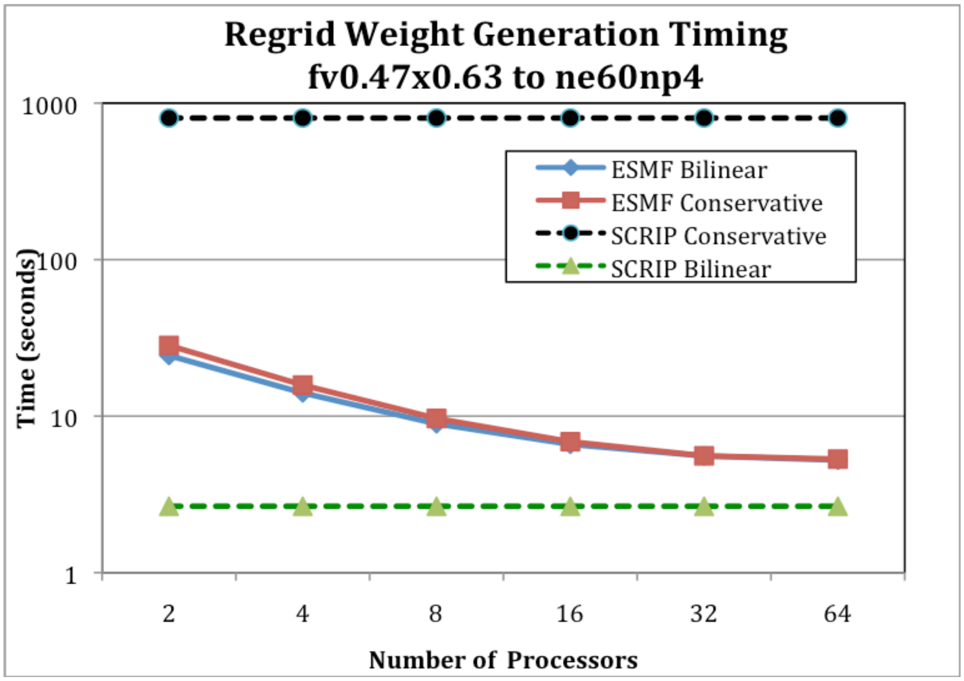


(a)

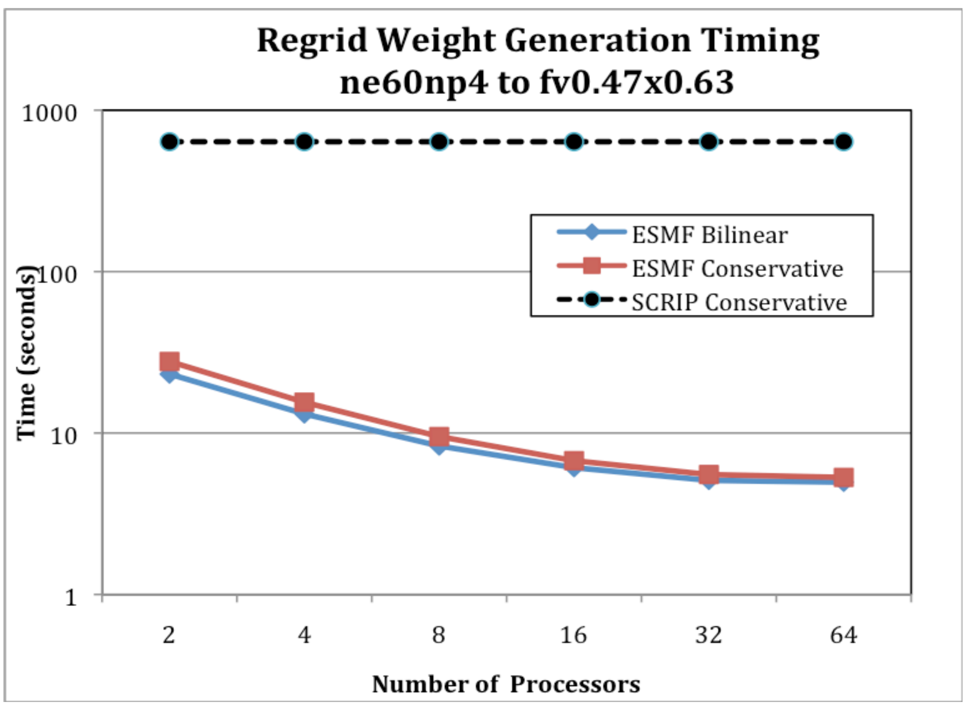


(b)

Figure 1: Case 1 - ESMF Offline Regrid Weight Gen Timing (a) fv0.47x0.63 to ne60np4 (b) ne60np4 to fv0.47x0.63



(a)



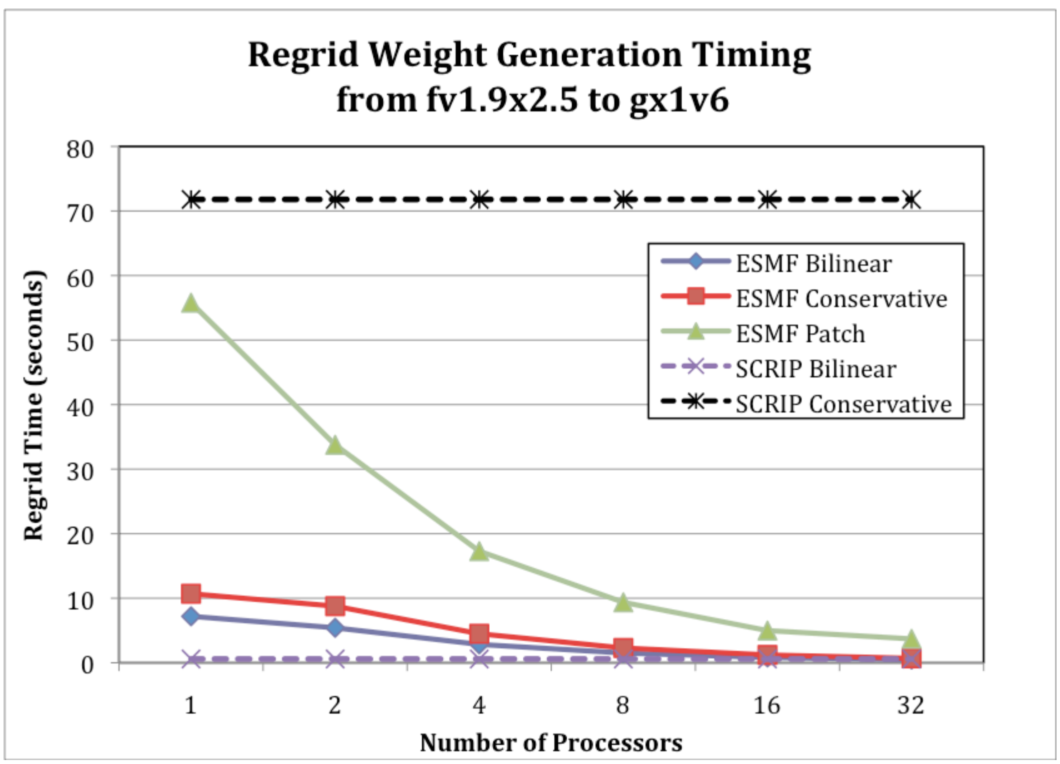
(b)

Figure 2 ESMF vs SCRIP regrid weight generation timing (a) fv0.47x0.63 to ne60np4 (b) ne60np4 to fv0.47x0.63

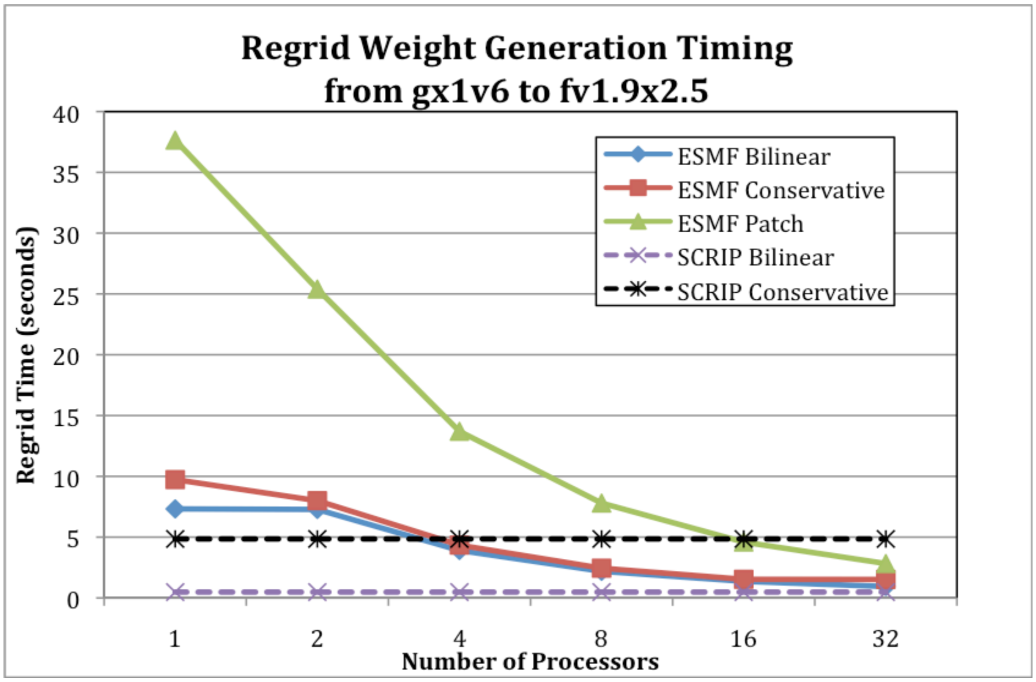
Case 2: fv1.9x2.5 vs gx1v6

These two grids are the smallest grids among all four cases. The timings were measured using 1 to 32 PEs. The ESMF patch interpolation is included for completeness. The patch interpolation timing is not included in the other three cases because it usually takes much longer time and uses more memory space. Therefore, for larger grids, the patch interpolation does not run on small number of processors.

The ESMF conservative and bilinear interpolation timing has similar pattern as shown in case 1, i.e., conservative is slower at lower PE counts but scales better thus takes about the same time as the bilinear interpolation at larger PE counts. The ESMF patch interpolation is about 5 times slower than the bilinear interpolation on a single PE. It scales well with increasing number of processors. It is not clear why the conservative remapping from gx1v6 to fv1.9x2.5 is much faster than the remapping in the opposite direction using SCRIP.



(a)



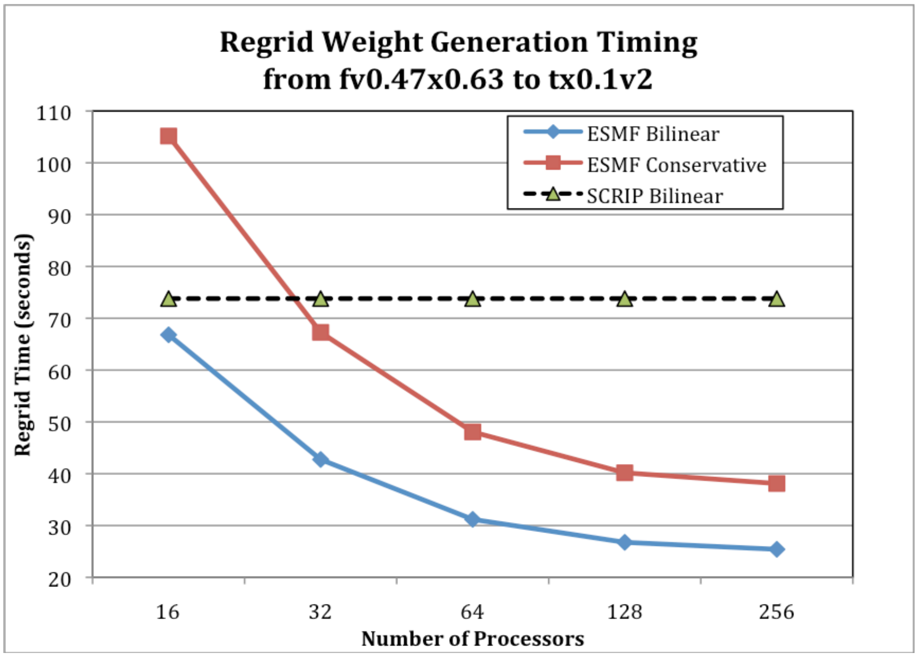
(b)

Figure 3. ESMF vs SCRIP regrid weight generation timing (a) fv1.9x2.5 to gx1v6 (b) gx1v6 to fv1.9x2.5

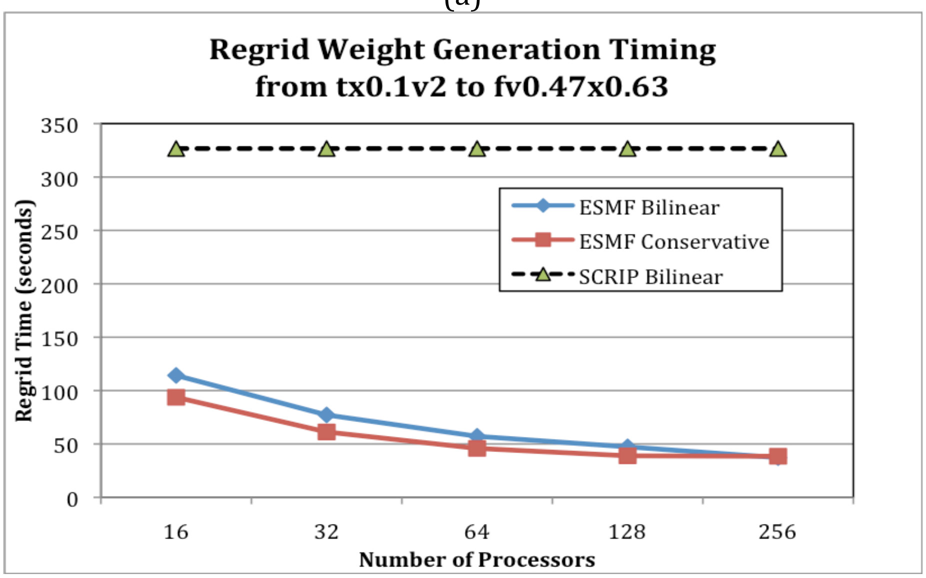
Case 3: fv0.47x0.63 vs tx1v2

In this case, the two grids have huge resolution difference; fv0.47x0.63 grid has a resolution of 0.47 degree x 0.63 degree and tx0.1v2 has a resolution of 0.1 degree x 0.08 degree. The minimal number of PEs to run the ESMF regridding is 16 PEs. It takes too long to run the SCRIP conservative remapping. I killed the job after 8 hours.

The SCRIP bilinear remapping from tx1v2 to fv0.47x0.63 on single PE is slower than the ESMF regridding for the first time. Also notice that the ESMF bilinear regridding is slower than the ESMF conservative regridding from tx1v2 to fv0.47x0.63 (Figure 4.b)



(a)

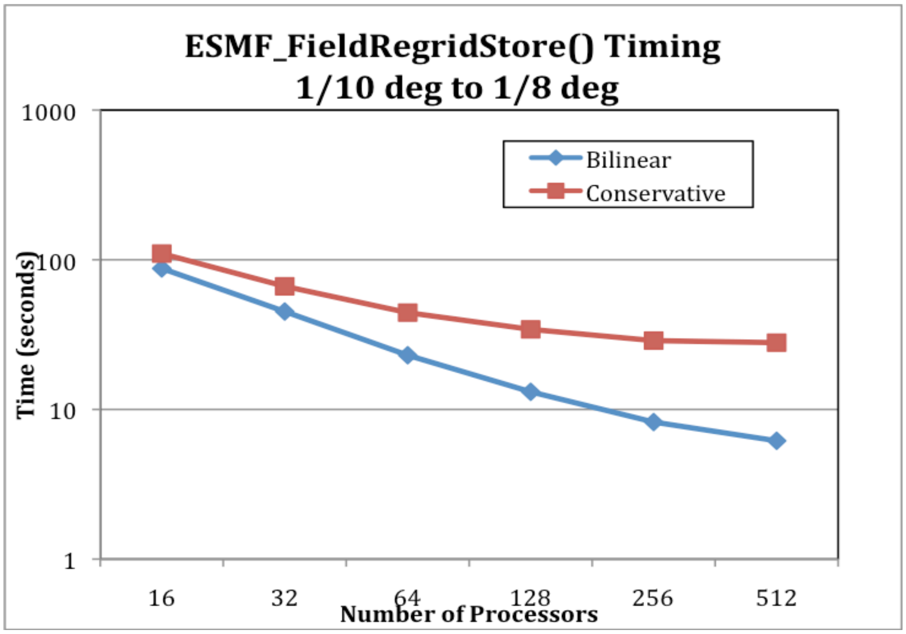


(b)

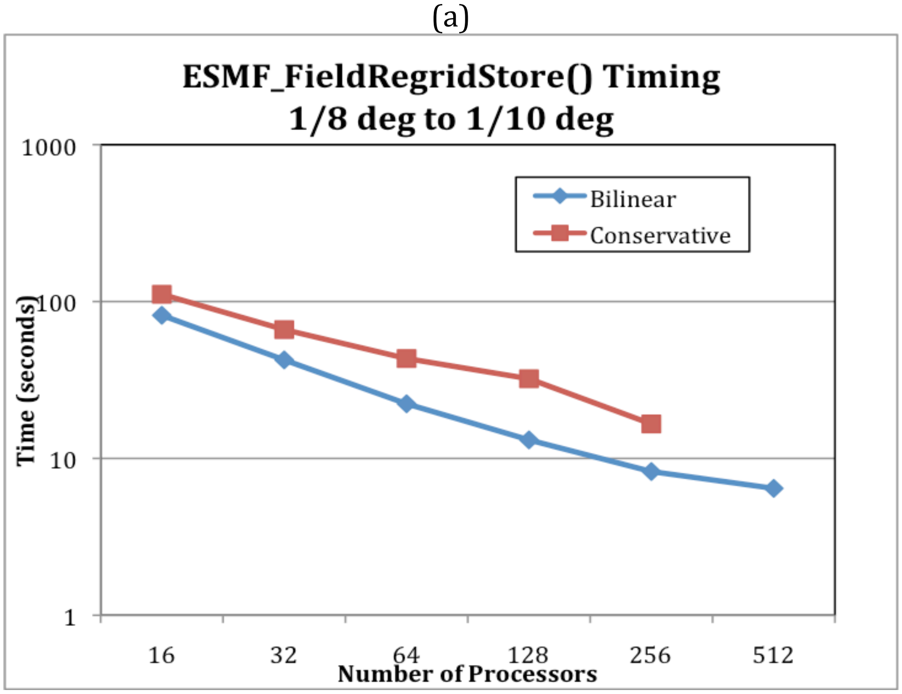
Figure 4. The ESMF vs SCRIP regrid weight generation timing (a) from fv0.47x0.63 to tx0.1v2 (b) from tx0.1v2 to fv0.47x0.63

Case 4: 1/8th degree vs 1/10th degree regularly rectangular grids

Being the largest configuration of all four, this case was used to measure the scalability of the ESMF_FieldRegridStore() function. The offline regrid application ran out of memory on 8 and less PEs. Therefore, the timing results were collected using 16 to 512 PEs. The timing plotted in Figure 5 is the time to run ESMF_FieldRegridStore() only. The plots are in log scale. The 1/8th to 1/10th conservative interpolation failed on 512 PEs because the system ran out some internal MPICH buffer space. One significant difference in this case is that the ESMF conservative regridding does not scale as well as the bilinear regridding as observed in the other three cases. The SCRIP timing is not included in Figure 5. SCRIP bilinear remapping from 1/10th degree to 1/8th degree grid takes 1242 seconds and 1003 seconds in the other direction.



(a)



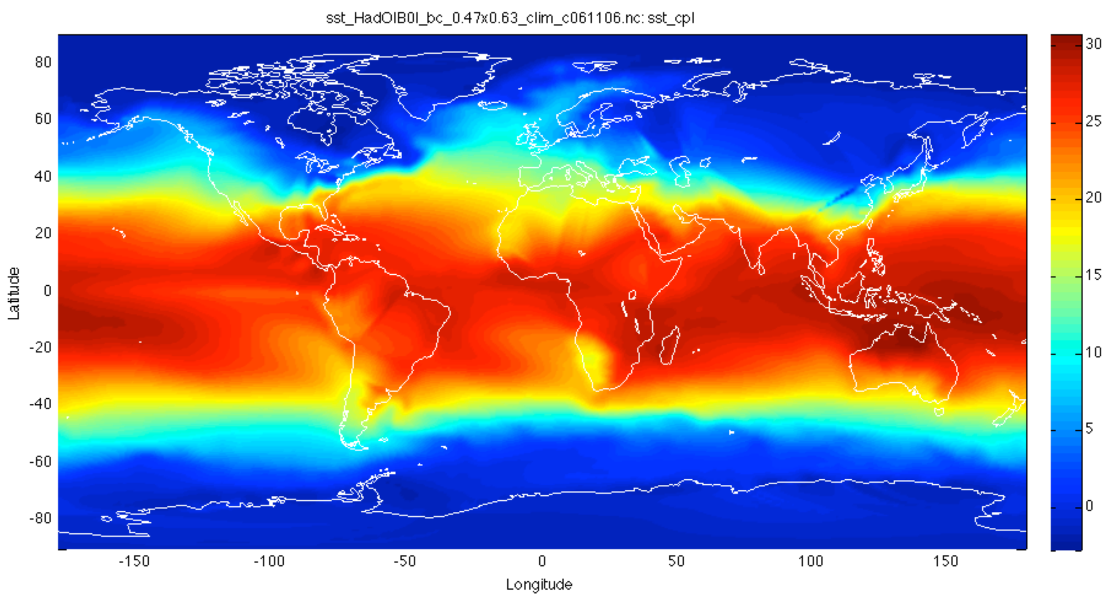
(b)

Figure 5. ESMF_FieldRegridStore() timing (a) 1/10th degree grid to 1/8th degree grid (b) 1/8th degree to 1/10th degree

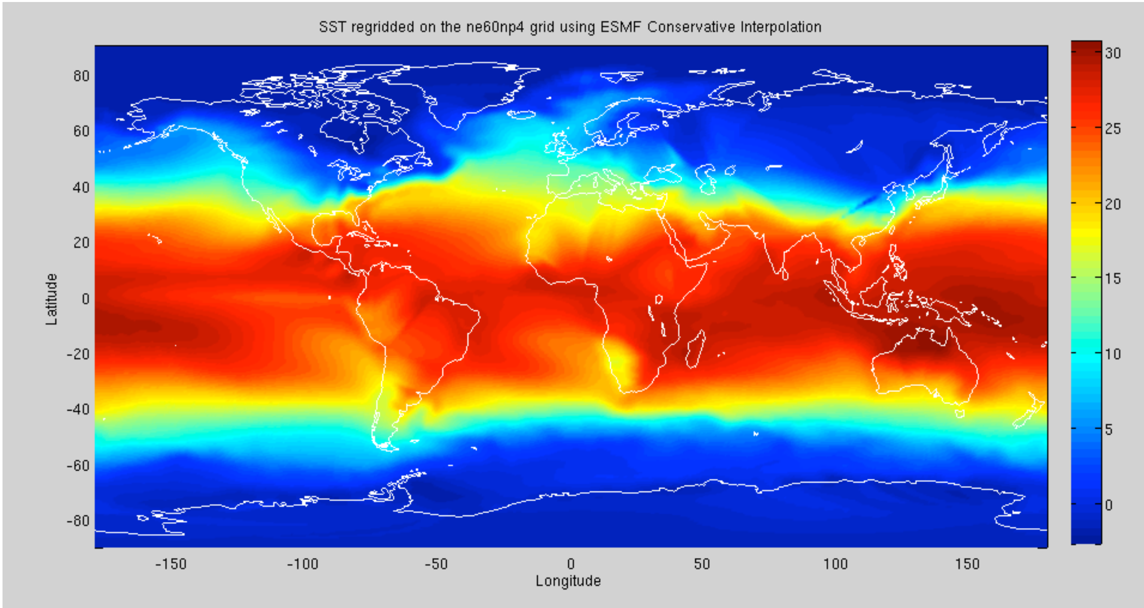
Weight Quality

We compare the quality of the weights generated by the ESMF and SCRIP conservative remapping using three methods. First, we interpolated a realistic data field using the interpolation weight table and calculate the mean errors between the original field and the interpolated field. Figure 6.a is a Sea Surface Temperature field used by CAM 3.0 as the boundary condition. The original data is on the fv0.47x0.63 grid. This field was first remapped to the ne60np4 cubed sphere cube using the weight file, and then the interpolated field on the cubed sphere grid was interpolated back to the original grid using the weight file in the reverse direction. Figure 6.b shows the interpolated field using the ESMF weight files. Figure 6.c is the interpolated field using the SCRIP weight files. Visually, it is hard to find any difference between the original and the interpolated fields.

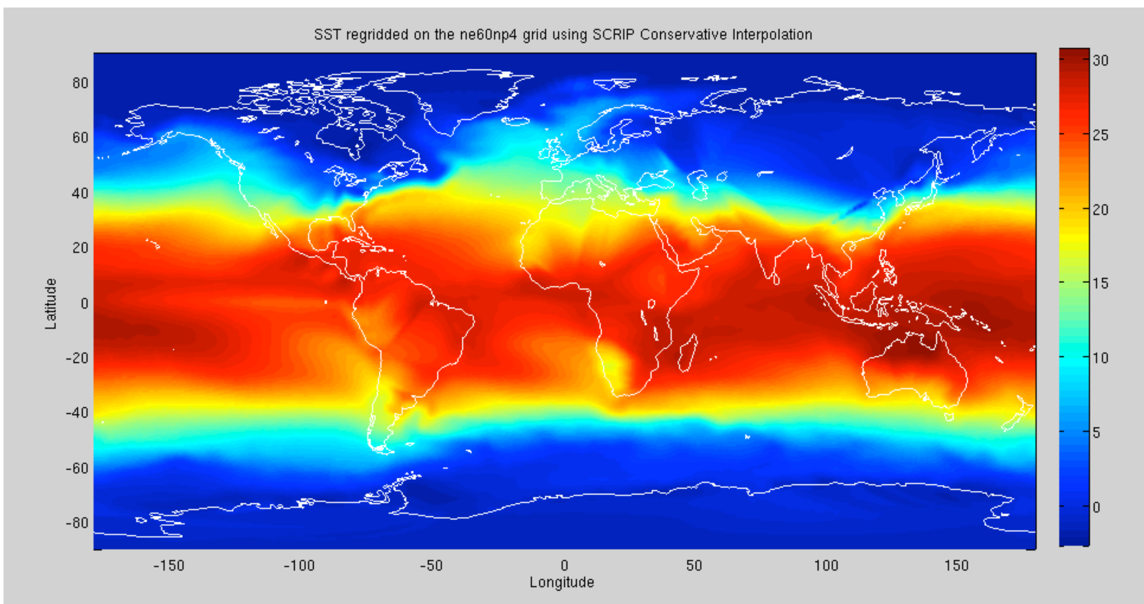
The difference plots between the original and the interpolated fields are shown in Figure 7. In both cases, the mean errors are around 2.8E-4 degrees. The minimal and maximal errors produced from the two interpolation programs differ by only 1E-3 degrees. From the difference plots, the biggest errors occur when there are big gradients in the data field. Big errors also show up at higher latitude around 0 degree longitude. The fan shape lines match with the cell distribution in the cubed sphere grid.



(a)

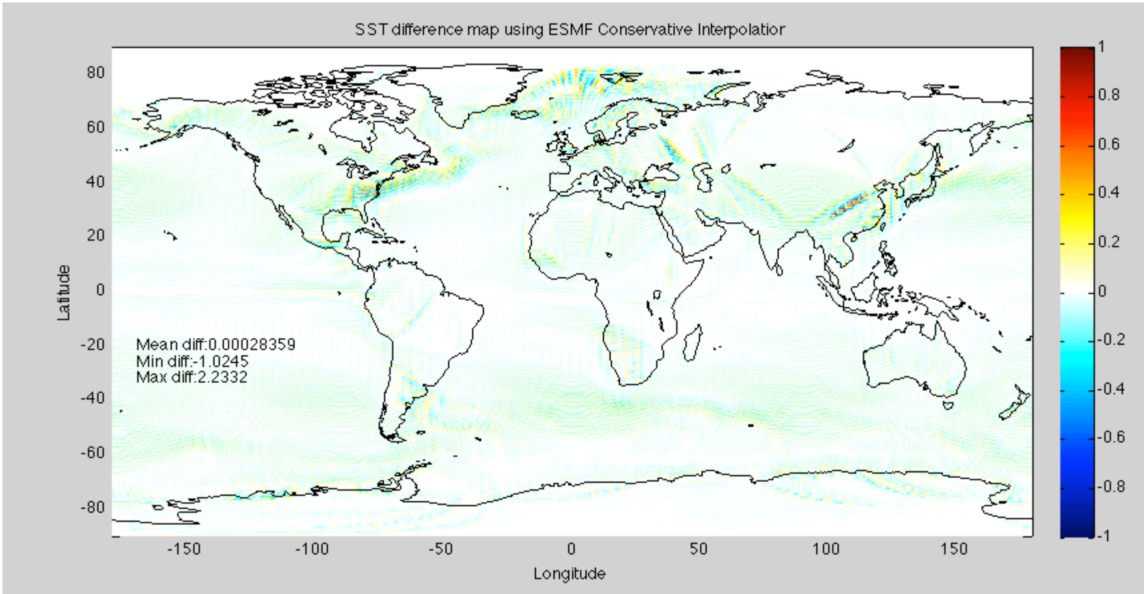


(b)



(c)

Figure 6. A SST field before and after interpolation (a) the original field on the fv0.47x0.63 grid, (b) the interpolated field using ESMF (c) the interpolated field using SCRIP



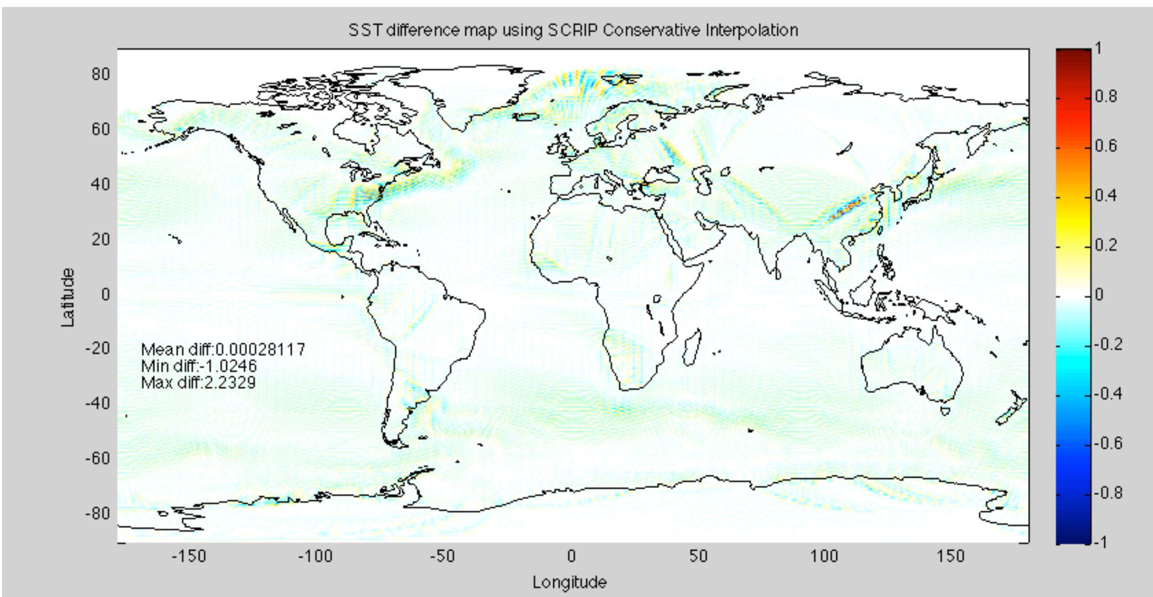


Figure 7. The difference map between the interpolated field and the original field

Secondly, we ran SCRIP_TEST program on the conservative weight files generated by both ESMF and SCRIP. SCRIP_TEST is a simple test program included in the SCRIP package. It reads in a weight file and remaps an analytic field. The field we chose is a spherical harmonic wave function $f = 2+\cos^2\theta\cos(2\phi)$. Table 2 shows the average relative errors of these interpolated fields. The error ranges from both methods are within the same order of magnitude.

Table 2. The mean errors from SCRIP_TEST

Mean Error Comparison (from scrip_test's Err2)				
src/dst grids	bilinear		conserv	
	scrip	esmf	scrip	esmf
fv0.47->ne60np4	1.09E-05	1.02E-05	4.10E-04	4.10E-04
ne60np4->fv0.47	not supported	8.60E-06	3.27E-04	3.27E-04

Finally, we examined the individual weight to see if there is any out-of-bound weight. Out-of-bound weights are either negative weights or the weights with values greater than 1. Table 3 is a summary of out-of-bound weights from all the conservative remappings we produced in this benchmark exercise. ESMF does not generate any out-of-bound weights. There are some negative weights generated by SCRIP using the fv1.9x2.5 and gx1v6 grids. Note the ESMF snapshot used in this benchmark does not support grid masking.

Table 3. the out of bound weight counts

grids	total weights	negative wgts	>1 wgts
fv2ne (SCRIP)	898880	0	0
ne2fv (SCRIP)	898880	0	0
fv2ne (ESMF)	914862	0	0
ne2fv (ESMF)	914862	0	0
gx2fv1.9(SCRIP)	160834	39	0
fv1.92gx(SCRIP)	228595	39	0
gx2fv1.9(ESMF)*	228601	0	0
fv1.92gx(ESMF)	228601	0	0
fv2tx (ESMF)	11847263	0	0
tx2fv (ESMF)	11847263	0	0

Note:

ne: ne60np4 (180x180x6)
fv: fv0.47x0.63 (576x384)
gx: gx1v6 (320x384) masked
fv1.9: fv1.9x2.5 (144x96)
tx: tx1v2 (3600x2400) masked

* Mask is ignored in ESMF conservative regrid

Conclusion

In summary, ESMF generates equal or better qualify weights as SCRIP does for both the bilinear and conservative remappings. The ESMF conservative interpolation is faster than the SCRIP conservative remapping algorithm by a big margin. The ESMF bilinear interpolation runs slower than the SCRIP counterpart because of the additional cost to convert the grid objects into ESMF mesh objects. Since ESMF regridding is fully parallel, it can always generate the interpolation weights within minutes in contract to hours or even days using SCRIP. Functionally, ESMF is still lacking the second-order conservative remapping as SCRIP provides. The masking for the unstructured grid and for the conservative regridding is not supported currently. In addition, it will be nice to provide further optimization in ESMF_FieldRegridStore() and the offline regrid weight generator by generating the interpolation weights in both directions.