

# Stage-I Performance Improvement

## Stage-I Code Performance Improvement Suggestions:

### 1.) Use pointers to CLM internal data:

There's code in `csm_share` for a performance improvement of using pointers to the CLM internal data rather than having a separate copy for ESMF import and export states and having to copy in/out on each call to CLM. To do this you check out the CLM branch (`esmfStage1_clm3_expa_69`) and go to `cd src/main`  
edit `Ind_comp_esmf.F90`  
change `setup_states`

```
From
  call eshr_estate_init2DFromList( eGrid, FieldList=seq_flds_x21_fields, &
    eState=x21_l )
  call eshr_estate_init2DFromList( eGrid, FieldList=seq_flds_l2x_fields, &
    eState=l2x_l )
To
  call eshr_estate_init2DFromList( eGrid, FieldList=seq_flds_x21_fields, &
    eState=x21_l , UsePTR=.true.)
  call eshr_estate_init2DFromList( eGrid, FieldList=seq_flds_l2x_fields, &
    eState=l2x_l , UsePTR=.true.)
```

then modify `Ind_comp_convertImport` and `Ind_comp_convertExport` to set the pointers for each field using `eshr_estate_setDataPointer` and only call them once at initialization. `Ind_convertImport` also has some work that needs to be called every run step that operates on internal data types. So this would need to be converted into a subroutine and called each time `Ind_run` is.

Doing this will make about a 1% speedup for ESMF and also reduce memory usage.

### 2.) Improve memory efficiency of copy in atm import/export

The other performance improvement I'd like to do is to make the copy in `atm_comp_esmf.F90` for import/export state data memory efficient.

### 3.) Get IS\_NEEDED logic working

And the final one is to get the `isNeeded` logic working so ONLY the data needed for coupling is actually copied, transferred, and redistributed. The import states do this. But, it also needs to be put in the export states. On creation the export states should set all data to NOTNEEDED

Do this by changing the export setup state calls from

```
From
  call eshr_estate_init2DFromList( eGrid, FieldList=seq_flds_l2x_fields, &
    eState=l2x_l )
to
  call eshr_estate_init2DFromList( eGrid, FieldList=seq_flds_l2x_fields, &
    eState=l2x_l , notNeeded=seq_flds_l2x_fields)
```

then the mergers need to set their input data to `NEEDED`, and the mappers pass this info along as well at initialization.

Once that is done, the `eshr_estate_mod.F90` code can turn make use of this by changing

```
From
  logical,          parameter :: OnlyUseNeededData = .false.
To
  logical,          parameter :: OnlyUseNeededData = .true.
```

You may also need to be able to handle the case where no data is transferred – so mappers and mergers (especially for Ocean) may need to be modified for this case as well.

### 4.) Motivation for getting the improvements in:

If we don't get these improvements in – ESMF is likely to be worse than MCT – on all fronts. But, with them – I think we are likely to see a performance improvement over MCT in both time and memory usage. It also takes full advantage of ESMF functionality and provides a service to the CCSM modeling community.