Using MOM6 in CESM

Alper Altuntas
G.Marques, F.Bryan, G.Danabasoglu, S.Bachman, K.Lindsay, M.Vertenstein, J.Edwards et al.
National Center for Atmospheric Research

MOM6 Webinar Series
Apr 27, 2020

NCAR
UCAR

Updates

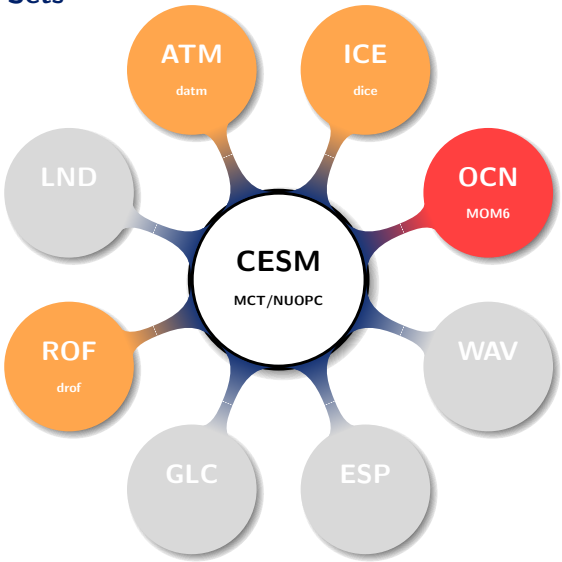
- MOM6 is fully incorporated in CESM testing and tagging workflow.
- MOM6 is to be an *optional* component in CESM 2.2.
  - First CESM 2.2 release tag this summer.
  - CESM 2.2 alpha tags already available.

- What's currently available:

Compsets: C, G, B
Forcings: CORE2 *(default)*, JRA-55
Drivers: MCT *(default)*, NUOPC
Grids: `tx0.66v1` *(workhorse)*
`gx1v6` *(testing only)*
`tx0.25v1` *(testing only)*

CESM Component Sets

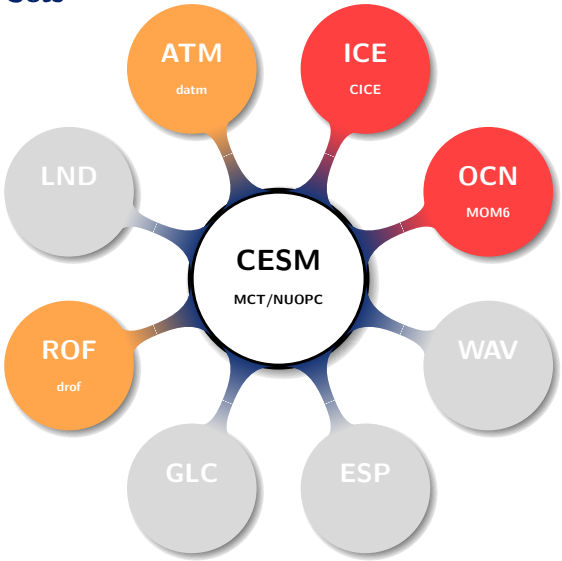
CMOM
COMPSET



- Active component
- Data component
- Stub component

CESM Component Sets

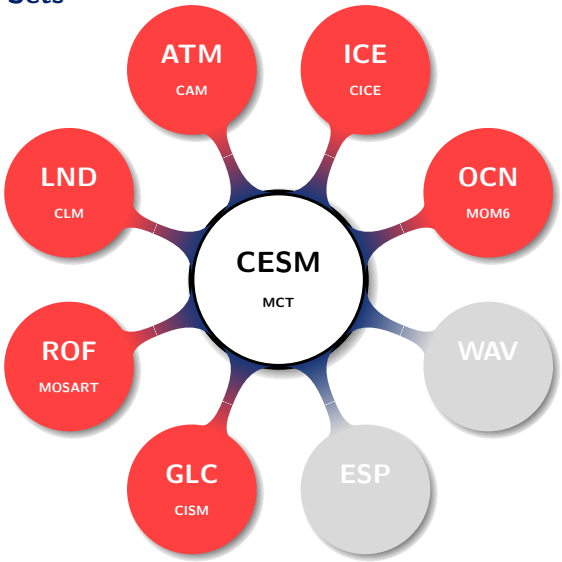
GMOM
COMPSET



- Active component
- Data component
- Stub component

CESM Component Sets

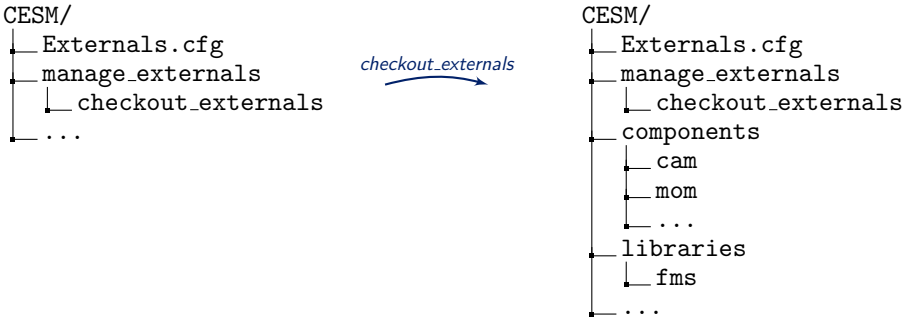
BMOM
COMPSET



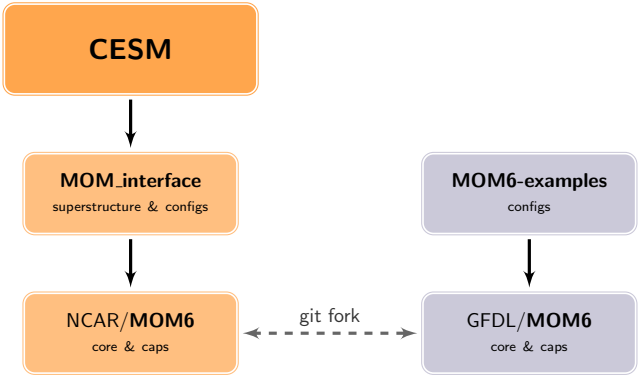
- Active component
- Data component
- Stub component

CESM Repository Structure

- All components and libraries on GitHub and public.
- A custom-made package manager: `manage_externals`
  - external $\approx$ git submodule



CESM & MOM6 Repository Structure

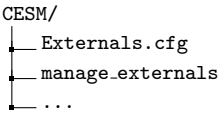


Tutorial: Using MOM6 in CESM

Downloading CESM+MOM6

- Clone CESM Github repository: *(∼5 sec.)*

```
$ git clone https://github.com/ESCOMP/CESM.git
```



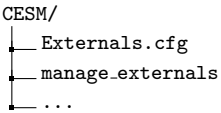
Downloading CESM+MOM6

- Clone CESM Github repository: *(∼5 sec.)*

```
$ git clone https://github.com/ESCOMP/CESM.git
```

- Check out the CESM 2.2 alpha tag including MOM6: *(∼1 sec.)*

```
$ cd CESM/
$ git checkout cesm2_2_alpha04d_mom6
```



Downloading CESM+MOM6

- Clone CESM GitHub repository: *(∼5 sec.)*

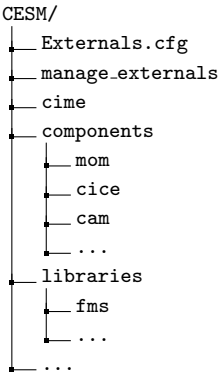
```
$ git clone https://github.com/ESCOMP/CESM.git
```

- Check out the CESM 2.2 alpha tag including MOM6: *(∼1 sec.)*

```
$ cd CESM/
$ git checkout cesm2_2_alpha04d_mom6
```

- Check out externals: *(∼2 min.)*

```
$ ./manage_externals/checkout_externals -o
```



Prerequisite

- Port CESM 2.2+
  - Can be ported to laptops, small clusters, supercomputers.
  - Already ported to some major ESM community machines:
    NCAR machines, lonestar5, stampede2, theia, gaea, aleph, cori, etc.

- *Porting documentation:*
  https://esmci.github.io/cime/versions/maint-5.6/html/users_guide/porting-cime.html

Running a MOM6 case – Available Configurations

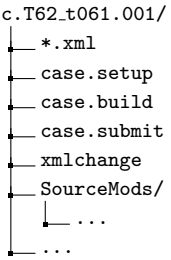
COMPSET	Compatible Resolutions	Description
CMOM	`T62_t061`, `T62_g16`, `T62_t025`	*MOM6 only, CORE2 NYF*
CMOM_IAF	`T62_t061`, `T62_g16`, `T62_t025`	*MOM6 only, CORE2 IAF*
CMOM_JRA	`TL319_t061`, `TL319_g16`	*MOM6 only, JRA55*
GMOM	`T62_t061`, `T62_g16`, `T62_t025`	*MOM6 and CICE only, CORE2 NYF*
GMOM_IAF	`T62_t061`, `T62_g16`, `T62_t025`	*MOM6 and CICE only, CORE2 IAF*
GMOM_JRA	`TL319_t061`, `TL319_g16`	*MOM6 and CICE only, JRA55*
BMOM	`f09_t061`	*Fully Coupled*

- `t061`: `tx0.66v1`
- `t025`: `tx0.25v1`
- `g16`: `gx1v6`

Running a MOM6 case

- Create a new case: *(∼5 sec.)*

```
$ CESM/cime/scripts/create_newcase \
    --run-unsupported              \
    --res T62_t061                 \
    --compset CMOM                 \
    --case c.T62_t061.001
```



Running a MOM6 case

- Create a new case: *(∼5 sec.)*

```
$ CESM/cime/scripts/create_newcase \
    --run-unsupported              \
    --res T62_t061                 \
    --compset CMOM                 \
    --case c.T62_t061.001
```

- Setup and build the case: *(∼5-10 mins.)*

```
$ cd c.T62_t061.001
$ ./case.setup
$ ./case.build
```

```
c.T62_t061.001/
├── *.xml
├── case.setup
├── case.build
├── case.submit
├── xmlchange
├── SourceMods/
│   └── ...
├── user_nl_???
└── ...
```

Running a MOM6 case

- Create a new case: *(∼5 sec.)*

```
$ CESM/cime/scripts/create_newcase \
    --run-unsupported              \
    --res T62_t061                 \
    --compset CMOM                 \
    --case c.T62_t061.001
```

- Setup and build the case: *(∼5-10 mins.)*

```
$ cd c.T62_t061.001
$ ./case.setup
$ ./case.build
```

- Submit the run: *(∼2 sec.)*

```
$ ./case.submit
```

```
c.T62_t061.001/
├── *.xml
├── case.setup
├── case.build
├── case.submit
├── xmlchange
├── SourceMods/
│   └── ...
├── user_nl_???
└── ...
```

Customizing a MOM6 case

- Mechanisms:
  1. **xml changes**:
     - General, model-agnostic settings, e.g., `NTASKS`, `NCPL`
     - High-level MOM6 diagnostics control.
  2. **user_nl_mom**
     - Runtime Parameters
     - Automatically transformed to `MOM_override`.
  3. SourceMods
     - For development only.
     - Overrides auto-generated input files:
       `MOM_input`, `MOM_override`, `diag_table`, `input.nml`

Example xml changes

- **Before building the case**:

```
$ ./xmlchange NTASKS_OCN=360          # change no. of MOM6 MPI tasks
$ ./xmlchange DEBUG=TRUE              # turn on compiler debug mode
$ ./xmlchange NCPL_OCN=48             # change the coupling frequency
```

- **Before submitting the run**:

```
$ ./xmlchange CONTINUE_RUN=TRUE       # make it a restart run
$ ./xmlchange STOP_N=3                # set the run duration
$ ./xmlchange STOP_OPTION=nmonths     # set the run duration
$ ./xmlchange JOB_WALLCLOCK_TIME=01:00:00 # set the job duration
```

- **High-level MOM6 diagnostics control**:

```
$ ./xmlchange OCN_DIAG_MODE=spinup    # spinup,production,development
$ ./xmlchange OCN_DIAG_SECTIONS=FALSE # turn off section diags
```

Example `xmlquery` use cases

```
$ ./xmlquery RUNDIR
        RUNDIR: /glade/scratch/altuntas/c.T62_t061.001/run
```

```
$ ./xmlquery -p DOUT
        DOUT_S: TRUE
        DOUT_S_SAVE_INTERIM_RESTART_FILES: FALSE
        DOUT_S_ROOT: /glade/scratch/altuntas/archive/c.T62_t061.001
```

```
$ ./xmlquery --description DOUT_S
        DOUT_S: Logical to turn on short term archiving. If TRUE,
            short term archiving will be turned on.
```

```
$ ./xmlquery --valid-values OCN_DIAG_MODE
        OCN_DIAG_MODE: ['development', 'spinup', 'production']
```

Example `user_nl_mom` changes

- All MOM6 runtime parameters to be changed via `user_nl_mom`.
- Same syntax as `MOM_override`.
- Example entries:

```
DT = 900                        ! change baroclinic timestep
KHTH = 800.0                    ! bckg horizontal thickness diffusivity
KPP%INTERP_TYPE = 'linear'      ! interpolation to determine OBL depth
```

Note: Unlike `MOM_override`, the `#override` keyword is not needed.

MOM6 Runtime Parameters in CESM

Conventional MOM6 Runtime Parameter System (sans CESM)

- **MOM6 code** - *A typical input parameter initialization:*

```
call get_param(param_file, "MOM", "HFREEZE", CS%HFrz, &
   "The depth over which melt potential is computed." &
   units="m", default=-1.0)
```

- **MOM_input file:** *all non-default params defining a "baseline" experiment.*

```
HFREEZE = 10.0 ! [default=-1.0m] The depth over which
               ! melt potential is computed.
```

- **MOM_override file:** *a few parameters that define a "derived" experiment.*

```
#override HFREEZE = 2.5
```

MOM6 Runtime Parameter System in CESM

Conventional MOM6 parameter files adopted/repurposed within the CESM framework:

- `MOM_input`:
  - Out-of-the-box MOM6 configuration in CESM.
  - Auto-generated.
  - Cannot be changed by the user. *(except, via SourceMods)*
- `MOM_override`:
  - Auto-generated from `user_nl_mom`.
  - Cannot be changed directly by the user. *(except, via SourceMods)*
- `diag_table`:
  - Auto-generated.
  - Can be altered via xml changes *(or, via SourceMods)*
- `input.nml`:
  - Auto-generated, cannot (and should never) be changed.

MOM6 Runtime Parameters in CESM

`DIN_LOC_ROOT`: Directory of all component input data, e.g., grids, forcings, ... } *common*

`CASEROOT`: Control desk, where the user can apply changes.

`RUNDIR`: temporary directory for run. files staged automatically.

`DOUT_S_ROOT`: short-term archive directory.

} *case-specific*

CASEROOT

```
c.T62_t061.001/
├── *.xml
├── case.setup
├── case.build
├── case.submit
├── xmlchange
├── SourceMods/
│   └── ...
├── user_nl_mom
└── ...
```

RUNDIR

```
[SCRATCH]/c.T62_t061.001/run/
└── .
```

MOM6 Runtime Parameters in CESM

`DIN_LOC_ROOT`: Directory of all component input data, e.g., grids, forcings, ... } *common*

`CASEROOT`: Control desk, where the user can apply changes.

`RUNDIR`: temporary directory for run. files staged automatically.

`DOUT_S_ROOT`: short-term archive directory.

} *case-specific*

CASEROOT

```
c.T62_t061.001/
├── *.xml
├── case.setup
├── case.build
├── case.submit
├── xmlchange
├── SourceMods/
│   └── ...
├── user_nl_mom
└── ...
```

./preview_namelists,
./case.build,
./case.submit

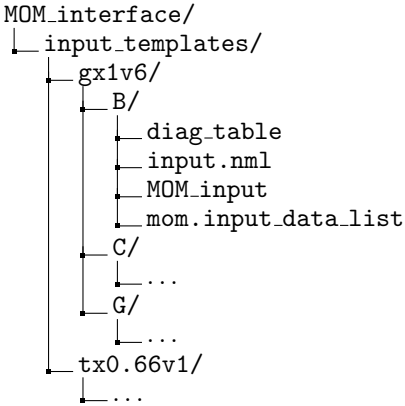
RUNDIR

```
[SCRATCH]/c.T62_t061.001/run/
├── MOM_input
├── MOM_override
├── diag_table
├── input.nml
├── ???_in
└── ...
```

Staging input files – Initially, in CESM...

Maintain copies for each grid × compset:

- Static files: parameter values hard-coded.
- Not extensible/tractable.
- Restrictive: out-of-the-box configurations depend only on OCN_GRID and COMPSET.



Staging input files – Currently, in CESM:

Single yaml file for each file category:

- All possible CESM configurations.
- Dynamic: param values can be tied to any CESM variable, e.g., `OCN_GRID`, `DATM_MODE`, `COMP_WAV`, etc.

```
MOM_interface/
└── param_templates/
    ├── diag_table.yaml
    ├── input_nml.yaml
    ├── MOM_input.yaml
    └── input_data_list.yaml
```

Example MOM_input.yaml entries

```
INPUTDIR:
    value: ${DIN_LOC_ROOT}/ocn/mom/${OCN_GRID}
```

```
TRIPOLAR_N:
    value:
        $OCN_GRID in ["tx0.66v1", "tx0.25v1"]: True
        else: False
```

```
DT_THERM:
    value: |
        = ( ( $NCPL_BASE_PERIOD =="decade") * 86400.0 * 3650.0 +
            ( $NCPL_BASE_PERIOD =="year") * 86400.0 * 365.0 +
            ( $NCPL_BASE_PERIOD =="day") * 86400.0 +
            ( $NCPL_BASE_PERIOD =="hour") * 3600.0 ) / $OCN_NCPL
```

Reminder: `MOM_input.yaml` is to reflect our out-of-the-box MOM6 configurations, and, thus, should not be modified for the purpose of customizing a particular case.

Testing MOM6 in CESM

Testing MOM6 in CESM

- **Continuous Integration – Travis:**
  - MOM6 repository: same as GFDL fork.
  - MOM_interface: unit tests for `MOM_RPS` module, consistency checks, linter.
- **CESM testing infrastructure**
  - Test suites run before every MOM6 PR and tag creation.
    - `aux_mom`: comprehensive
    - `pr_mom`: lightweight
  - Test suite run by CSEG before every CESM alpha tag.
    - `prealpha`: comprehensive, inter-component.

Testing MOM6 in CESM

- Run an exact restart test:

```
$ CESM/cime/scripts/create_test ERS.T62_t061.cheyenne_intel
```

- Create a baseline:

```
$ CESM/cime/scripts/create_test ERS.T62_t061.cheyenne_intel \
    -g [YOUR-BASELINE-DIR]
```

- Compare against a baseline:

```
$ CESM/cime/scripts/create_test ERS.T62_t061.cheyenne_intel \
    -c [YOUR-BASELINE-DIR]
```

- Run a full test suite:

```
$ CESM/cime/scripts/create_test \
    --xml-category aux_mom --xml-machine cheyenne
```

Remarks

Remarks

- A *functional* MOM6 release in upcoming CESM 2.2.
  - Not fully scientifically vetted.
  - Current focus on `GMOM_JRA`. Next, `BMOM`.
- Early versions available for download and experimentation.
  - "Out-of-the-box" configurations change rapidly.
- Mechanisms for customizing a MOM6 case:
  - `xmlchange`
  - `user_nl_mom`
- Check out online user manual: "MOM6 in CESM"
  - https://github.com/ESCOMP/MOM_interface/wiki
- Post your questions to CESM/MOM6 forum.
  - https://bb.cgd.ucar.edu/cesm/forums/mom.148/

Thanks!

altuntas@ucar.edu