NorESM code efficiency

Chandan Basu, Hamish Struthers

NorESM workshop, Stockholm
23rd - 24th October 2014

NorESM code analysis and optimization (SNIC Project based user support)

Project organization

- Requester:
- Ilona Riipinen, Department of Applied Environmental Science, Stockholms University
- Juan-Camilo Acosta, Department of Applied Environmental Science, Stockholms University
-
- Project responsible for SNIC: Chandan Basu, NSC, SNIC
-
- Project member: Hamish Struthers, NSC, SNIC
-
- Project manager: Torben Rasmussen, NSC, SNIC
-
- SNIC project name: SNIC 2014/1-155 and SNIC 2014/8-18
-
- Duration: 1 month

NorESM code analysis and optimization (SNIC Project based user support)

Expected enabling benefit

- Short-term benefit
- Better understand the bottlenecks of our code and learn about performance optimization potentials.
- Set of recommendations regarding focus points for code optimization changes.
-
- Long-term benefit:
- A speedup of this code will enable researchers to perform more and/or longer simulations within a range of scientific projects.

NorESM code analysis and optimization (SNIC Project based user support)

Project objective

- Review the MPI pe mapping of the different sub-models (atmosphere, land, sea ice and ocean) currently used for model simulations
-
- Analyze the NorESM code to identify code sections and routines to be further evaluated for performance optimization. We will do this by running suitable test cases through analysis tools such as TAU and VTune
-
- Propose a set of recommendations for code optimization changes based on the performance analysis.

Benchmark cases

We are using two benchmarks cases

- N20TRAERCNTRCH & NF2005MOZNPF
-
- The case setup, compiling, and running is complicated
-
- The standard CCSM works fine
-
- NorESM changes are copied by hand or edited on the file
- This is prone to mistakes
- Confusing for a new user
- Took me almost 2 weeks to start running
-
- A tool for NorESM on top of CCSM will be helpful for users
- Can be a by product of our project
-
- Done some testing with N20TRAERCNTRCH
-

NorESM tool example

Building the case

```
rm -rf noresm-ver1-cmip5/cases/N20TRAERCNTRCH;
cd noresm-ver1-cmip5/scripts/;
cp ${MY_SRC}/config_machines.xml ccsm_utils/Machines/config_machines.xml;
cp ${MY_SRC}/Macros.triolith.new ccsm_utils/Machines/Macros.triolith;
cp ${MY_SRC}/config_pes.xml.new ccsm_utils/Machines/config_pes.xml;
cp ${MY_SRC}/config_compsets.xml.N20TRAERCNTRCH ccsm_utils/Case.template/config_compsets.xml;
./create_newcase -case ../cases/N20TRAERCNTRCH -mach triolith -res f19_g16 -compset N20TRAERCNTRCH -pecount M;
cd ../cases/N20TRAERCNTRCH;
rm -rf SourceMods/src.cam/;
cp -R ${MY_SRC}/src.cam.N20TRAERCNTRCH SourceMods/src.cam/;
rm -rf ../../models/atm/cam/bld/namelist_files/use_cases;
cp -R ${MY_SRC}/use_cases.N20TRAERCNTRCH ../../models/atm/cam/bld/namelist_files/use_cases;
cp ${MY_SRC}/env_conf.xml.N20TRAERCNTRCH env_conf.xml;
./configure -case;
cp ${MY_SRC}/config_cache.xml.N20TRAERCNTRCH Buildconf/camconf/config_cache.xml;
cp ${MY_SRC}/cam.buildexe.csh.N20TRAERCNTRCH Buildconf/cam.buildexe.csh;
rm LockedFiles/env_conf.xml.locked;
./N20TRAERCNTRCH.triolith.build
```

Case specific files kept in a separate folder

NorESM tool example

Running the case

```
cd noresm-ver1-cmip5/cases/N20TRAERCNTRCH;
vim -p env_mach_pes.xml;
cp ${MY_SRC}/env_conf.xml.N20TRAERCNTRCH env_conf.xml;
./configure -cleanall;
./configure -case;
cp ${MY_SRC}/cam.buildexe.csh.N20TRAERCNTRCH Buildconf/cam.buildexe.csh;
cp ${MY_SRC}/config_cache.xml.N20TRAERCNTRCH Buildconf/camconf/config_cache.xml;
rm LockedFiles/env_conf.xml.locked;
./N20TRAERCNTRCH.triolith.build;
cp ${MY_SRC}/cam.buildnml.csh.N20TRAERCNTRCH Buildconf/cam.buildnml.csh;
cp ${MY_SRC}/clm.buildnml.csh.N20TRAERCNTRCH Buildconf/clm.buildnml.csh;
sbatch N20TRAERCNTRCH.triolith.run
```

- Actual changes can be done in the ${MY_SRC} folder
- This will keep The two code separate
- Chances of accidental delete is less
- More efficient

Profiling NorESM with TAU

- TAU is an open source profiling tool for MPI applications
-
- TAU can be used in different modes
- We used source code instrumentation mode
-
- Compiled the code with TAU compiler
mpif90 --> tau_f90.sh

- Tau compiler puts profiling calls in each subroutine
- temporary copy created on the fly
-
- Run as usual
- Some TAU variables can be set in the environment
-
- At the end of the run profile files will be generated
-
- profiles can be seen by paraprof tool

Scaling and load balancing of NorESM

N20TRAERCNTRCH

We tested 3 PE layouts

<table>
<tr><td rowspan="3">run1</td><td colspan="2">CPL (44)</td><td rowspan="3">OCN (36)</td><td rowspan="3">~10m</td></tr>
<tr><td>LND (20)</td><td>ICE (24)</td></tr>
<tr><td colspan="2">ATM (44)</td></tr>
</table>

<table>
<tr><td rowspan="3">run2</td><td colspan="2">CPL (60)</td><td rowspan="3">OCN (36)</td><td rowspan="3">~9m</td></tr>
<tr><td>LND (30)</td><td>ICE (30)</td></tr>
<tr><td colspan="2">ATM (60)</td></tr>
</table>

<table>
<tr><td rowspan="3">run3</td><td colspan="2">CPL (92)</td><td rowspan="3">OCN (36)</td><td rowspan="3">~</td></tr>
<tr><td>LND (52)</td><td>ICE (40)</td></tr>
<tr><td colspan="2">ATM (92)</td></tr>
</table>