CMAQ 2021

For Linux (Optimum cluster)

- **NOTE:** This updated UBC guide is for installing the latest (at the time of writing) stable version of CMAQ (V5.3.3) on the Department of Earth, Ocean and Atmospheric Sciences "Optimum" cluster
    - Specifically, using PGI 19.10 (pgcc, pgfortran) with OpenMPI-3.1.3
    - CMAQ can also be installed with gfortran and ifort, though instructions on how to do so are not included here
    - Likewise, CMAQ can be installed on any (reasonably recent and capable) Linux machine, provided it can support parallel computing

- UBC is not affiliated with the USEPA and CMAQ; these instructions are meant for pedagogical purposes for the atmospheric dispersion modelling course **ATSC 595D**

- If you are reading these instructions outside of ATSC 595D, please note that these instructions may not apply to your specific systems, and UBC is under no obligation to provide support
    - CMAQ repository on Github: https://github.com/USEPA/CMAQ

- Bolded entries are individual commands to be placed on the command line; they should be written and entered as a single line in the terminal

- Main CMAQ site: https://www.epa.gov/cmaq/access-cmaq-source-code

- Full CMAQ user's guide: https://github.com/USEPA/CMAQ/blob/main/DOCS/Users_Guide/README.md

Outline

1. Install (or have access to) NETCDF

2. Download and install IOAPI, to interface with NETCDF

3. Download CMAQ; stage it for installation and running by setting the right environment variables and sourcing a config file

4. Install the pre-processors (ICON = chemical initial conditions; BCON = chemical boundary conditions; MCIP = Meteorology-Chemistry Interface Processor [WRF-to-chemistry pre-processor])

5. Install the main model (CCTM = CMAQ Chemical Transport Model)

6. Test the installation with a benchmark case study over southeastern US

Install IOAPI

- Before installing CMAQ, we have to first install IOAPI, which is the interface between NETCDF and CMAQ

- Log on to Optimum
    - **ssh** username@optimum.eos.ubc.ca
        - Replace “username” with your username

- Carefully read over the login message!!! NOTE ESPECIALLY THAT RUNS AND DATA STORAGE SHOULD OCCUR IN **$SCRATCH,** NOT IN **$HOME**

```
********************************************************************************
DISCLAIMER: THIS IS A PRIVATE COMPUTER SYSTEM !!!
This is OPTIMUM, an EOAS computer system for authorized use only.
EOAS computer systems may be monitored for all lawful purposes, including
to ensure that their use is authorized, for management of the system, to
facilitate against unauthorized access, and to verify security procedures,
survivability, and operational security. Using this system constitutes consent
to monitoring. All information, including personal information, placed on or
sent over this system may be obtained during monitoring. Unauthorized use could
result in criminal prosecution.
--------------------------------------------------------------------------------

USER DIRECTORIES:
       $HOME for applications, setups, and scripts. (daily backup)
              DO NOT USE for storing any data.
       $ARCHIVEDIR for storing semi-permanent input and results. (no backup)
              DO NOT USE for applications, setups, and scripts.
              !!! file lifetime of 3 years based on recent-access time
       $SCRATCHDIR for working/running jobs; routinely cleaned up (no backup)
              DO NOT USE for applications, setups, and scripts.
              !!! file lifetime of 1 year based on recent-access time
You can check your disk usage using 'check_quota' command.

QUICK WEB ACCESS to public files (do not use for sensative content):
       https://www.optimum.eos.ubc.ca/~$USER (points to $HOME/public_html)

REMEMBER!!! DO NOT RUN HEAVY JOBS/APPLICATIONS ON THE LOGIN NODE.

BEWARE!!! Offending applications of any user on a head node(s) will be killed.
```

- **

- Compilers have already been pre-installed on CMAQ with modules; you can list out what's available with the command **`module avail`**

- We will load up the "latest" PGI compiler suite (that also contains OpenMPI, needed for running jobs across compute cores in parallel):
  - **`module load PGI/19.10/nollvm`**
  - ^This command sets the paths needed for working with PGI 19.10; to make sure you did this correctly, you should see the paths from running:
    - **`which pgfortran`**
      - /home/Software/system/PGI/linux86-64-nollvm/19.10/bin/pgfortran
    - **`which mpifort`**
      - /home/Software/system/PGI/linux86-64-nollvm/19.10/mpi/openmpi-3.1.3/bin/mpifort

- CMAQ installation scripts are written in csh (c-shell), and NOT bash ---> we will need to switch over to csh before moving on
  - **`csh`**
  - You won't see any output, but you'll know you've converted to csh if you type **`export`** and get "Command not found"

- IOAPI and CMAQ require NETCDF; a version compiled with PGI 19.10 has already been pre-installed, and was made available in the previous CALPUFF tutorial for installing CALWRF on Optimum
  - **`setenv NETCDF /scratch/rstull/shared/netcdf_2021`**
    - Libs: ${NETCDF}/lib
    - Headers/"includes": ${NETCDF}/include

- Now we're ready to install IOAPI; make a new directory to include all CMAQ-related files; download IOAPI-3.2; unzip it, and head into the base directory
  - **`mkdir CMAQ`**

  - **cd CMAQ**
  - **wget http://github.com/cjcoats/ioapi-3.2/archive/20200828.tar.gz**
  - **tar -xvzf 20200828.tar.gz**
  - **cd ioapi-3.2-20200828**

- You should now be in $HOME/CMAQ/ioapi-3.2-20200828

- Set some necessary environment variables
  - **setenv BIN Linux2_x86_64pg**
    - Install with PGI
  - **setenv BASEDIR $cwd**
    - Set base directory
  - **setenv CPLMODE nocpl**
    - No PVM (parallel virtual machine) coupling

- Copy the Makefile template to the main Makefile we'll build with
  - **cp Makefile.template Makefile**
  - Makefiles contain all of the variables/options needed to compile an entire software package

- Open up Makefile
  - **vi Makefile**
  - If you want to see line numbers, type **:set nu**

- On line 193, make the following edit:
  - **NCFLIBS = -L${NETCDF}/lib -lnetcdff -lnetcdf**
  - Save and exit with **<esc>:wq**

- Go into the source directory
  - **cd ioapi**

- Open up Makefile.nocpl

    - **vi Makefile.nocpl**
    - If you want to see line numbers, type **:set nu**

- On line 81, make the following edit:
    - **BASEDIR = ${HOME}/CMAQ/ioapi-3.2-20200828**
    - Save and exit with **<esc>:wq**

- Copy the nocpl Makefile template to the main Makefile (within ioapi)
    - **cp Makefile.nocpl Makefile**

- Go back to the base directory
    - **cd ..**

- Go into the tools directory
    - **cd m3tools**

- Open up Makefile.nocpl
    - **vi Makefile.nocpl**
    - If you want to see line numbers, type **:set nu**

- On line 41, make the following edit:
    - **BASEDIR = ${HOME}/CMAQ/ioapi-3.2-20200828**
    - Save and exit with **<esc>:wq**

- Copy the nocpl Makefile template to the main Makefile (within m3tools)
    - **cp Makefile.nocpl Makefile**

- Go back to the base directory
    - **cd ..**

- Build IOAPI
    - **make all**

- You should not see any errors; finished object files (i.e. libraries) and binaries are found in newly created directory Linux2_x86_64pg
    - **`ls Linux2_x86_64pg`**

- For installing CMAQ, specify that this (i.e. the base directory, where you are now) is the location of the IOAPI installation directory
    - **`setenv IOAPI_PGI ${cwd}`**
    - Confirm with: **`echo $IOAPI_PGI`**

Stage CMAQ

- Return to the CMAQ directory
    - **`cd ~/CMAQ`**

- Download CMAQ (can also clone from Github if you'd like):
    - **`wget`** https://github.com/USEPA/CMAQ/archive/main.zip
    - **`unzip main.zip`**
    - **`cd CMAQ-main`**

- Make a copy directory (CMAQ_Project) containing only stuff that's pertinent to our build by editing and running bldit_project.csh; keep CMAQ-main as the "clean" copy

- Edit bldit_project.csh
    - **`vi bldit_project.csh`**
    - Line 20: **`CMAQ_HOME = ~/CMAQ/CMAQ_Project`**
    - **`<esc>:wq`**

- Run bldit_project.csh
    - **`./bldit_project.csh`**

- Head into CMAQ_Project
  - **`cd ../CMAQ_Project`**

- Set a separate NETCDF variable ($NETCDF is used for another variable within CMAQ)
  - **`setenv NETCDF_PGI /scratch/rstull/shared/netcdf_2021`**

- You will also need to set the linking paths to NETCDF as well; not so much for installation, but certainly for running the compiled executables later on:
  - **`setenv LD_LIBRARY_PATH ${NETCDF_PGI}/lib:${LD_LIBRARY_PATH}`**

- Edit config_cmaq.csh
  - **`vi config_cmaq.csh`**
  - Under “case pgi” (Lines 119 - 153), make the following edits (line number shown immediately after bullets; each bullet is one line):
    - 122 **`setenv IOAPI  ${IOAPI_PGI}`**
    - 123 **`setenv NCDIR  ${NETCDF_PGI}`**
    - 124 **`setenv NFDIR  ${NETCDF_PGI}`**
    - 129 **`setenv IOAPI_INCL_DIR ${IOAPI}/ioapi`**
    - 130 **`setenv IOAPI_LIB_DIR ${IOAPI}/Linux2_x86_64pg`**
    - 132 **`setenv NETCDF_LIB_DIR   ${NCDIR}/lib`**
    - 133 **`setenv NETCDF_INCL_DIR  ${NCDIR}/include`**
    - 134 **`setenv NETCDFF_LIB_DIR  ${NFDIR}/lib`**
    - 135 **`setenv NETCDFF_INCL_DIR ${NFDIR}/include`**
    - 136 **`setenv MPI_INCL_DIR /home/Software/system/PGI/linux86-64-nollvm/19.10/mpi/openmpi-3.1.3/include`**
    - 137 **`setenv MPI_LIB_DIR /home/Software/system/PGI/linux86-64-nollvm/19.10/mpi/openmpi-3.1.3/lib`**

```
119     case pgi:
120
121         #> I/O API and netCDF for WRF-CMAQ
122         setenv IOAPI  ${IOAPI_PGI}
123         setenv NCDIR  ${NETCDF_PGI}
124         setenv NFDIR  ${NETCDF_PGI}
125         setenv NETCDF netcdf_combined_directory_path # Note only for  WRF-CMAQ as it requires combining the netcdf
126         setenv WRF_ARCH 3                           # [1-75] Optional, ONLY for WRF-CMAQ
127
128         #> I/O API, netCDF, and MPI library locations
129         setenv IOAPI_INCL_DIR   ${IOAPI}/ioapi    #> I/O API include header files
130         setenv IOAPI_LIB_DIR    ${IOAPI}/Linux2_x86_64pg    #> I/O API libraries
131         if ( $NETCDF == "netcdf_combined_directory_path" ) then
132             setenv NETCDF_LIB_DIR   ${NCDIR}/lib                    #> netCDF C directory path
133             setenv NETCDF_INCL_DIR  ${NCDIR}/include                    #> netCDF C directory path
134             setenv NETCDFF_LIB_DIR  ${NFDIR}/lib          #> netCDF Fortran directory path
135             setenv NETCDFF_INCL_DIR ${NFDIR}/include          #> netCDF Fortran directory path
136             setenv MPI_INCL_DIR     /home/Software/system/PGI/linux86-64-nollvm/19.10/mpi/openmpi-3.1.3/include
137             setenv MPI_LIB_DIR      /home/Software/system/PGI/linux86-64-nollvm/19.10/mpi/openmpi-3.1.3/lib
138         endif
139
```

- Save and quit
  - **<esc>:wq**

- Source the configuration file
  - **source config_cmaq.csh pgi**

##

EACH TIME YOU WORK WITH CMAQ IN A NEW TERMINAL WINDOW OR SUBSHELL

- **module load PGI/19.10/nollvm**
- **csh**
- **setenv IOAPI_PGI ${HOME}/CMAQ/ioapi-3.2-20200828**
- **setenv NETCDF_PGI /scratch/rstull/shared/netcdf_2021**
- **setenv LD_LIBRARY_PATH ${NETCDF_PGI}/lib:$LD_LIBRARY_PATH**
- **source ${HOME}/CMAQ/CMAQ_Project/config_cmaq.csh pgi**

##

Install ICON

- Go into PREP/icon/scripts
  - **`cd ${HOME}/CMAQ/CMAQ_Project/PREP/icon/scripts`**

- Build ICON and send the output to a logfile
  - **`./bldit_icon.csh pgi | tee build_icon.log`**

- Check that the executable works
  - **`cd BLD_ICON_v532_pgi`**
  - **`./ICON_v532.exe`**
  - Should get (at the bottom):
  - *** ERROR ABORT in subroutine ICON
  - *** Failure defining horizontal domain

Install BCON

- Go into PREP/bcon/scripts
  - **`cd ${HOME}/CMAQ/CMAQ_Project/PREP/bcon/scripts`**

- Build BCON and send the output to a logfile
  - **`./bldit_bcon.csh pgi | tee build_bcon.log`**

- Check that the executable works
  - **`cd BLD_BCON_v532_pgi`**
  - **`./BCON_v532.exe`**
  - Should get (at the bottom):
  - *** ERROR ABORT in subroutine BCON
  - *** Failure defining horizontal domain

Install MCIP

- Go into PREP/mcip/src
  - **`cd ${HOME}/CMAQ/CMAQ_Project/PREP/mcip/src`**

- Edit the Makefile
  - **`vi Makefile`**
  - Uncomment (i.e. remove the #) lines 26 to 36
  - Comment (i.e. add the #) lines 50 to 59
  - Edit lines 26 to 28 with the following:
    - **`FC = pgf90`**
    - **`NETCDF = ${NETCDF_PGI}`**
    - **`IOAPI_ROOT = ${IOAPI_PGI}`**
  - Save and quit with **`<esc>:wq`**

```
24 #...Portland Group Fortran
25 #...also need to setenv LM_LICENSE_FILE from /usr/local/apps/pgi
26 FC      = pgf90
27 NETCDF = ${NETCDF_PGI}
28 IOAPI_ROOT = ${IOAPI_PGI}
29 FFLAGS = -g -O0 -Ktrap=unf -Ktrap=ovf -Ktrap=divz -Ktrap=inv -Ktrap=fp  \
30          -Ktrap=inexact -pc 32 -Mbounds -Mchkfpstk -Mchkptr -Kieee     \
31          -Minform,inform -Mfree -byteswapio -I$(NETCDF)/include \
32          -I$(IOAPI_ROOT)/Linux2_x86_64pg
33 FFLAGS = -O4 -fastsse -pc 32 -Mfree -byteswapio -I$(NETCDF)/include \
34          -I$(IOAPI_ROOT)/Linux2_x86_64pg
35 LIBS    = -L$(IOAPI_ROOT)/Linux2_x86_64pg -lioapi \
36          -L$(NETCDF)/lib -lnetcdff -lnetcdf
37
38 #...gfortran
39 #FC      = gfortran
```

```
48
49 #...Intel Fortran
50 #FC      = ifort
51 #NETCDF = /usr/local/apps/netcdf-4.7.3/intel-19.0
52 #IOAPI_ROOT = /usr/local/apps/ioapi-3.2_20181011/intel-19.0
53 ###FFLAGS  = -g -O0 -check all -C -traceback -FR -I$(NETCDF)/include  \
54 ###          -I$(IOAPI_ROOT)/Linux2_x86_64ifort
55 #FFLAGS  = -O3 -traceback -FR -I$(NETCDF)/include -I$(IOAPI_ROOT)/Linux2_x86_64ifort
56 #LIBS    = -L$(IOAPI_ROOT)/lib -lioapi \
57           -L$(NETCDF)/lib -lnetcdff -lnetcdf
58
59 #DEFS   =
60
61
62 MODULES =\
```

- Compile MCIP
  - **`make`**

- Check that the executable works
  - **`./mcip.exe`**
  - Should get (at the bottom):
  - ********** ********** ********** ********** ********** ********** ********** ********** ********** **********
  - *** SUBROUTINE: READNML
  - *** ERROR OPENING NAMELIST FILE ON UNIT 8
  - *** NAMELIST FILE NAME = namelist.mcip
  - *** IOSTAT = 209
  - ********** ********** ********** ********** ********** ********** ********** ********** ********** **********
  -
  - *** ERROR ABORT in subroutine READNML
  - ABNORMAL TERMINATION IN READNML

Install CCTM

- Go into CCTM/scripts
  - **`cd ${HOME}/CMAQ/CMAQ_Project/CCTM/scripts`**

- Build CCTM and send the output to a logfile
  - **`./bldit_cctm.csh pgi |& tee build_cctm.log`**
  - If you run **`grep ERROR build_cctm.log`**
    - **ERROR** while running make command

- This error is because of an issue with ${HOME}/CMAQ/CMAQ_Project/lib/x86_64/pgi/ioapi/include_files/STATE3.EXT
  - There are “common blocks” within STATE3.EXT that run across multiple lines with & ---> PGI doesn’t like that, so we have to put those common blocks into one massive line
  - In particular:
    - Lines 174 - 187 must have all & removed and placed into a single line
      - COMMON / BSTATE3 / P_ALP3,...,PN_MODE
    - Lines 191-192 must have all & removed and placed into a single line
      - COMMON / CSTATE3 / EXECN3,...,VERSN3

- The end result (line 174 runs offscreen):

```
170         !!  non-character portion of current I/O API state
171         !!  Note that DOUBLE components are listed first, in order
172         !!  to ensure appropriate (64-bit, usually) alignment.
173
174         COMMON  / BSTATE3 /  P_ALP3, P_BET3, P_GAM3,  XCENT3, YCENT3, XORIG3, YORIG3, XCELL3, YCELL3, VGTYP3, VGTOP3,
175
176         !!  character portion of current I/O API state
177
178         COMMON  / CSTATE3 / EXECN3, SCNDSC, FLIST3, GDNAM3, VLIST3, UNITS3, VERSN3
179
180         SAVE / BSTATE3 / , / CSTATE3 /
```

- To save time, you can just copy in a fixed version of STATE3.EXT that’s been placed in ${NETCDF_PGI} for your convenience

    - **cp ${NETCDF_PGI}/STATE3.EXT ${HOME}/CMAQ/CMAQ_Project/lib/x86_64/pgi/ioapi/include_files/STATE3.EXT**

- Re-build CCTM
    - **./bldit_cctm.csh pgi |& tee build_cctm_fixed.log**
    - If you run **grep error build_cctm_fixed.log** or **grep ERROR build_cctm.log**, you should now see nothing

- Check that the executable works
    - **cd BLD_CCTM_v533_pgi**
    - **./CCTM_v533.exe**
    - Should get (at the bottom):
    - *** An error occurred in MPI_Init
    - *** on a NULL communicator
    - *** MPI_ERRORS_ARE_FATAL (processes in this communicator will now abort,
    - *** and potentially your MPI job)
    - [delta:82242] Local abort before MPI_INIT completed completed successfully, but am not able to aggregate error messages, and not able to guarantee that all other processes were killed!

- ^This is because CCTM was compiled with MPI, and so you must run with mpirun:
    - **mpirun -np 1 ./CCTM_v533.exe**
    - Should get (at the bottom):
    - EMIS_SYM_DATE | F (default)
    -
    - |> Process Analysis Parameters:
    - +==========
    - --Env Variable-- | --Value--
    -

- CTM_PROCAN | F (default)
  - PA_BCOL_ECOL | (default)
  - PA_BROW_EROW | (default)
  - PA_BLEV_ELEV | (default)
- ---
- Primary job terminated normally, but 1 process returned
- a non-zero exit code. Per user-direction, the job has been aborted.
- ---
- ---
- mpirun detected that one or more processes exited with non-zero status, thus causing
- the job to be terminated. The first process to do so was:
-
- Process name: [[55754,1],0]
- Exit code: 1
- ---

Test CMAQ

- Test cases and relevant input files can be found here: https://www.epa.gov/cmaq/cmaq-inputs-and-test-case-data

- We'll be using "CMAQv5.3.2_Benchmark_2Day_Input.tar.gz" (Two day input data → used for v5.3.3 as well)

- To save time downloading (it's a big file!), a copy of the .tar.gz has been placed in /scratch/rstull/shared/CMAQ/CMAQv5.3.2_Benchmark_2Day_Input.tar.gz

- Go into your scratch directory; make a new CMAQ directory; make a COPY of your CMAQ_Project directory from home, so that we can run everything in scratch
  - **`cd ${SCRATCHDIR}`**

- **mkdir CMAQ**
- **cd CMAQ**
- **cp -r ${HOME}/CMAQ/CMAQ_Project .**
- **cd CMAQ_Project**

- Go into the data directory; LINK in the .tar.gz; unzip
  - **cd data**
  - **ln -s /scratch/rstull/shared/CMAQ/CMAQv5.3.2_Benchmark_2Day_Input.tar.gz .**
  - **tar -xvzf CMAQv5.3.2_Benchmark_2Day_Input.tar.gz**

- Run ICON; edit run_icon.csh
  - **cd ../PREP/icon/scripts**
  - **vi run_icon.csh**
    - Line 16: **setenv compiler pgi**
    - Line 33: **set ICTYPE = profile**
    - Line 41: **setenv GRID_NAME SE52BENCH**
    - Line 43: **setenv GRIDDESC ${CMAQ_DATA}/CMAQv5.3.2_Benchmark_2Day_Input/2016_12SE1/met/mcipv5.0/GRIDDESC**
    - Line 104: **setenv MET_CRO_3D_FIN ${CMAQ_DATA}/CMAQv5.3.2_Benchmark_2Day_Input/2016_12SE1/met/mcipv5.0/METCRO3D_${YYMMDD}.nc**

  - **./run_icon.csh |& tee run_icon.log**
    - Value for IOAPI_LOG_WRITE: F returning FALSE
    - Warning: ieee_inexact is signaling
    - FORTRAN STOP
    -
    -
    - >>----> Program ICON completed successfully <----<<

        -
        -
        - 0.238u 0.363s 0:00.84 70.2% 0+0k 0+544664io 0pf+0w
        - exit ( )

- Run BCON; edit run_bcon.csh
    - **cd ../../bcon/scripts/**
    - **vi run_bcon.csh**
        - Line 16: **setenv compiler pgi**
        - Line 33: **set BCTYPE = profile**
        - Line 41: **setenv GRID_NAME SE52BENCH**
        - Line 43: **setenv GRIDDESC ${CMAQ_DATA}/CMAQv5.3.2_Benchmark_2Day_Input/2016_12SE1/met/mcipv5.0/GRIDDESC**
        - Line 105: **setenv MET_BDY_3D_FIN ${CMAQ_DATA}/CMAQv5.3.2_Benchmark_2Day_Input/2016_12SE1/met/mcipv5.0/METBDY3D_${YYMMDD}.nc**
    - **./run_bcon.csh | & tee run_bcon.log**
        - Time-independent data.
        - Value for IOAPI_LOG_WRITE: F returning FALSE
        - Warning: ieee_inexact is signaling
        - FORTRAN STOP
        -
        -
        - >>----> Program BCON completed successfully <----<<
        -
        -
        - 0.302u 0.065s 0:00.42 85.7% 0+0k 0+24936io 0pf+0w
        - exit ( )

- Edit run_cctm_Bench_2016_12SE1.csh
    - **cd ../../../CCTM/scripts**

- **vi run_cctm_Bench_2016_12SE1.csh**
  - Line 25: **setenv compiler pgi**
  - Line 58: **setenv INPDIR ${CMAQ_DATA}/CMAQv5.3.2_Benchmark_2Day_Input/2016_12SE1**
  - Line 88: **@ NPCOL = 5; @ NPROW = 4**
  - If you want a shorter run, edit line 81 (i.e. set NSTEPS = 060000)

- Submit an interactive queuing job: request a 1-hour job, with 1 node, and 20 processors per node
  - **Iqsub 1 1 20**

- Wait until you get your prompt back; then you'll have to reload everything to work with CMAQ (i.e. copy/paste the following):
  - **module load PGI/19.10/nollvm**
  - **csh**
  - **setenv IOAPI_PGI ${HOME}/CMAQ/ioapi-3.2-20200828**
  - **setenv NETCDF_PGI /scratch/rstull/shared/netcdf_2021**
  - **setenv LD_LIBRARY_PATH ${NETCDF_PGI}/lib:$LD_LIBRARY_PATH**
  - **source ${HOME}/CMAQ/CMAQ_Project/config_cmaq.csh pgi**

- Run CCTM
  - **./run_cctm_Bench_2016_12SE1.csh |& tee run_cctm.log**
  - If successful, for a 24-hour run, you should get:
  - =--> Data Output completed... 0.3 seconds
  -
  -

- ==========
- |>--- PROGRAM COMPLETED SUCCESSFULLY ---<|
- ==========
- Date and time 0:00:00 July 2, 2016 (2016184:000000)
- The elapsed time for this simulation was 1583.7 seconds.
- real 1584.16
- user 31062.39
- sys 471.55
- CMAQ Processing of Day 20160701 Finished at Thu Nov 11 22:38:16 PST 2021
- \\\\\=====\\\\\=====\\\\\=====\\\\\=====/////=====/////=====/////=========/////
- ==========
- ***** CMAQ TIMING REPORT *****
- ==========
- Start Day: 2016-07-01
- End Day: 2016-07-01
- Number of Simulation Days: 1
- Domain Name: 2016_12SE1
- Number of Grid Cells: 280000 (ROW x COL x LAY)
- Number of Layers: 35
- Number of Processes: 20
- All times are in seconds.
- Num Day Wall Time
- 01 2016-07-01 1584.16
- Total Time = 1584.16

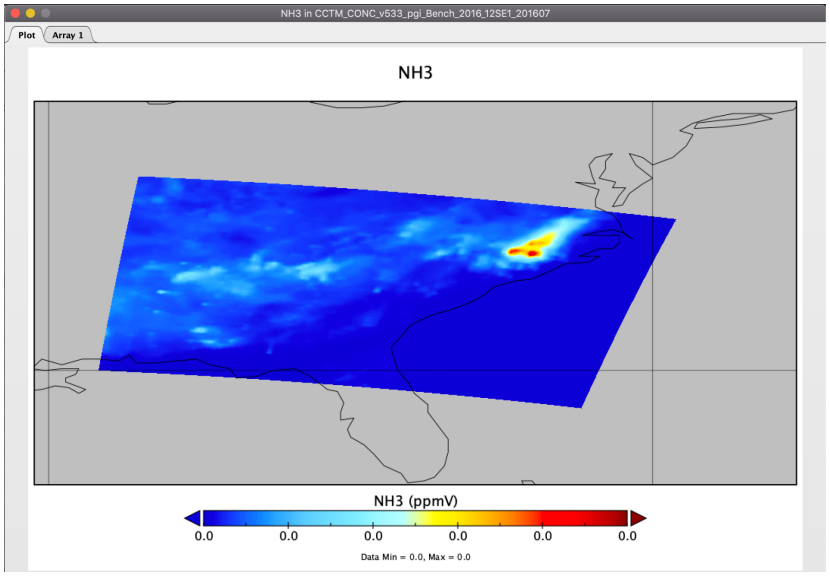
- Avg. Time = 1584.16

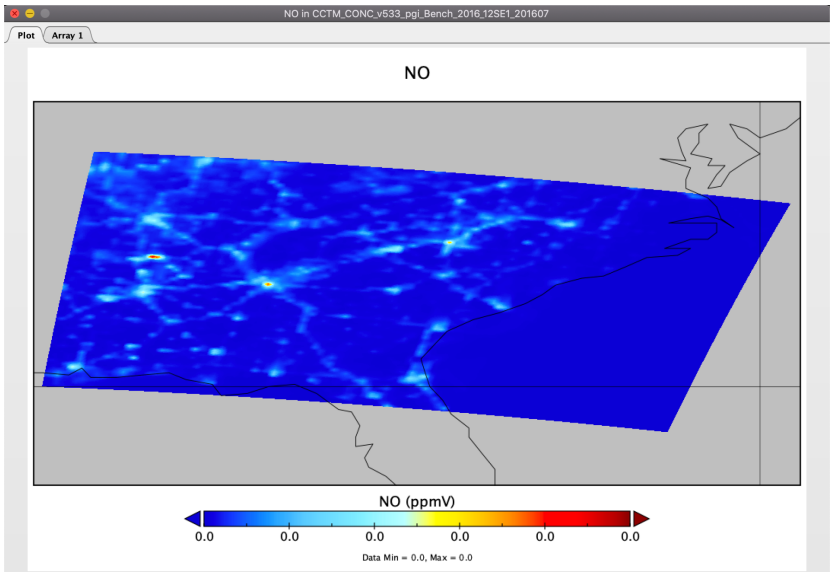
- When done, release the node
  - **`exit`**

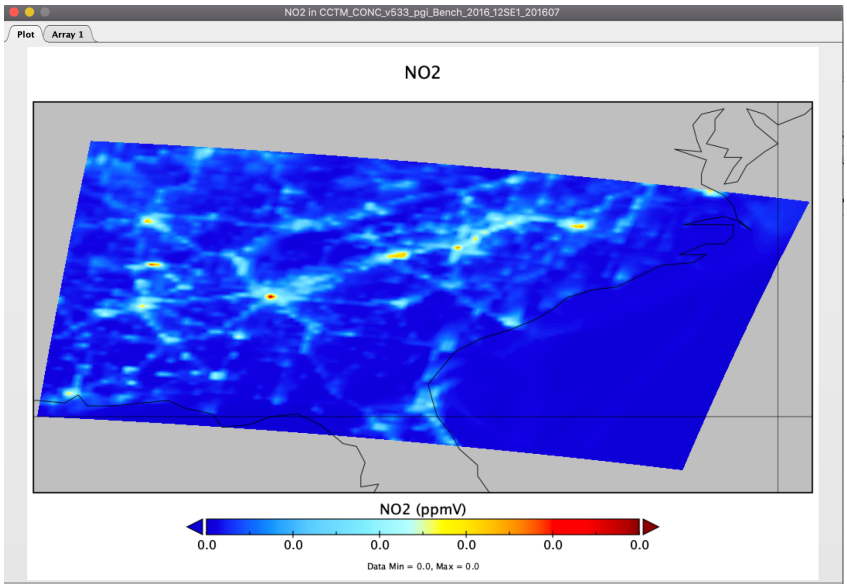
- Output is found in ${SCRATCHDIR}/CMAQ/CMAQ_Project/data/output_CCTM_v533_pgi_Bench_2016_12SE1

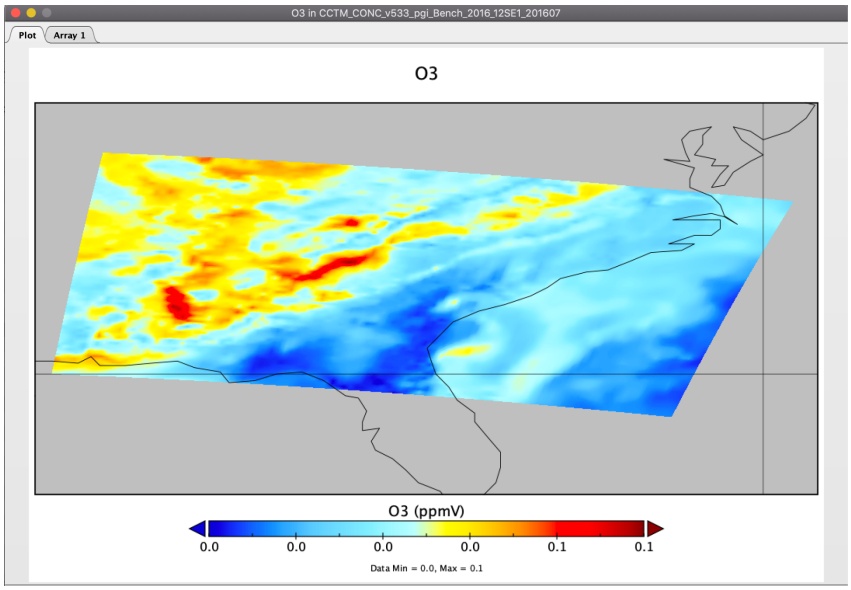
- Ground concentrations are found in CCTM_CONC_v533_pgi_Bench_2016_12SE1_20160701.nc
  - Can send this back to your local computer for plotting
  - From your local computer:
    - **`scp <username>@optimum.eos.ubc.ca:/scratch/rstull/<username>/CMAQ/CMAQ_Project/data/output_CCTM_v533_pgi_Bench_2016_12SE1/CCTM_CONC_v533_pgi_Bench_2016_12SE1_20160701.nc .`**

- You can then plot the output with Panoply
  - Output shown at hour 24 (didn't bother adjusting colorbar labels...but colours are scaled according to each species' molar mixing ratios, so red for O3 isn't the same as red for NO)


NH3 in CCTM_CONC_v533_pgi_Bench_2016_12SE1_201607
Plot
Array 1
NH3
NH3 (ppmV)
0.0
0.0
0.0
0.0
0.0
0.0
Data Min = 0.0, Max = 0.0


NO in CCTM_CONC_v533_pgi_Bench_2016_12SE1_201607
Plot
Array 1
NO
NO (ppmV)
0.0
0.0
0.0
0.0
0.0
0.0
Data Min = 0.0, Max = 0.0


NO2 in CCTM_CONC_v533_pgi_Bench_2016_12SE1_201607
Plot
Array 1
NO2
NO2 (ppmV)
0.0
0.0
0.0
0.0
0.0
Data Min = 0.0, Max = 0.0


O3 in CCTM_CONC_v533_pgi_Bench_2016_12SE1_201607
Plot
Array 1
O3
O3 (ppmV)
0.0
0.0
0.0
0.0
0.1
0.1
Data Min = 0.0, Max = 0.1