WRF CMAQ Install Notes

Be sure to purge modules and ENSURE that the SAME module is loaded when compiling each individual program (e.g., openmpi 1.8.3 and intel 15.0.3)

FAQ:

- If I run the two-way-coupled WRF-CMAQ system (available with CMAQv5.0), do I need to run MCIP?
  - No. The functionality of MCIP is handled within the two-way-coupled WRF-CMAQ system in AQPREP.

Original Notes from my first attempt to install the coupled model:

- Downloaded WRF-ARW 3.8 from http://www2.mmm.ucar.edu/wrf/users/download/get_sources.html
  - copied tar file to HPC, extracted, renamed folder to WRFV38 (see coupled model instructions)
    - https://github.com/USEPA/CMAQ/blob/5.2/CCTM/docs/Release_Notes/Two_Way_Coupled_WRF-CMAQ.md
- Copied NETCDF src from Shupeng (/dfs2/dabdub/shupengz/lib/srcNETCDF/)
  - netcdf-4.4.1.1.tar.gz and netcdf-fortran-4.4.4.tar.gz
- Extracted these in my HPC directory (e.g., /dfs2/dabdub/jrhorne/WRFCMAQ/lib) and created a directory structure similar to his
- Copied setting-intel.sh files from Shupengs folders (in same directory as above, the folders reflect the src names without the tar.gz) – one from each folder, into the folders I just created when I extracted the tar.gz in my directory
  - Modified these .sh files to reflect my directory structure
- Executed (source setting-intel.sh) the setting file in the NON Fortran netcdf first (4.4.1.1)
  - Got the "Congratulations! You have successfully installed netCDF!" box
    - See screenshot
- Moved into fortran netcdf directory and executed that setting-intel.sh setting file
  - Got the "Congratulations! You have successfully installed the netCDF Fortran libraries." message.
    - See screenshot
- Copied ioapi-3.2.tar.gz from Shupengs directory to mine (**OR FOLLOW MARC's IOAPI Instructions step by step – SEE Appendix A in the end of this document)**
  - IMPORTANT: Also need to edit the Makeinclude.Linux2_x86_64ifort file in the ioapi directory (this is in addition to the other Makefile edits listed in Marc's instructions)
    - Change FC   = ifort -auto -warn notruncated_source -Bstatic -static-intel
    - To FC   = ifort
      - e.g., remove all the extra flags
  - Before running make all on the main Makefile in the /lib directory, I had to manually export the LD_LIBRARY_PATH variables and LIBRARY_PATH
    - export LD_LIBRARY_PATH=/data/apps/mpi/openmpi-1.8.3/intel/15.0.3/lib/openmpi:/data/apps/mpi/openmpi-1.8.3/intel/15.0.3/lib:/data/apps/intel/parallel_studio-xe/2015-update-3/composer_xe_2015.3.187/mkl/lib/intel64:/data/apps/intel/parallel_studio-xe/2015-update-3/composer_xe_2015.3.187/mkl/lib/mic:/data/apps/intel/parallel_studio-xe/2015-update-3/composer_xe_2015.3.187/compiler/lib/intel64:/dfs2/dabdub/jrhorne/WRFCMAQ/lib/NETCDF_intel_15.0.3/lib
    - export LIBRARY_PATH=/data/apps/intel/parallel_studio-xe/2015-update-3/composer_xe_2015.3.187/compiler/lib/intel64:/data/apps/intel/parallel_studio-xe/2015-update-3/composer_xe_2015.3.187/ipp/../compiler/lib/intel64:/data/apps/intel/parallel_studio-xe/2015-update-3/composer_xe_2015.3.187/ipp/lib/intel64:/dfs2/dabdub/jrhorne/WRFCMAQ/lib/NETCDF_intel_15.0.3/lib

    - After doing that and successfully compiling, I editied my .bashrc file in ~ (home) HPC directory to include these variables
        - export LD_LIBRARY_PATH=/dfs2/dabdub/jrhorne/WRFCMAQ/lib/NETCDF_intel_15.0.3/lib
        - export LIBRARY_PATH=/dfs2/dabdub/jrhorne/WRFCMAQ/lib/NETCDF_intel_15.0.3/lib
        - export LD_LIBRARY_PATH=/dfs2/dabdub/jrhorne/lib/NETCDF_/lib: $LD_LIBRARY_PATH
        - export LIBRARY_PATH=/dfs2/dabdub/jrhorne/WRFCMAQ/lib/NETCDF_intel_15.0.3/lib: $LIBRARY_PATH
        - ***or the opposite order? See file***
    - AFTER following Marc's change instructions for Makefiles AND editing the Makeinclude file and exporting those paths manually, I was able to make all and compile successfully, completing ioapi installation.
-

Downloading and Installing CMAQ
https://www.airqualitymodeling.org/index.php/CMAQ_version_5.2_(June_2017_release)_Technical_Documentation
https://github.com/USEPA/CMAQ/blob/5.2/DOCS/User_Manual/README.md
https://github.com/USEPA/CMAQ
https://github.com/USEPA/CMAQ/tree/5.2

FAQ for WRF Runtime errors:
http://www2.mmm.ucar.edu/wrf/users/FAQ_files/FAQ_wrf_runtime.html

1. Obtained directly from git repository
2. Must go into csh shell to source .csh script
3. Get benchmark data from here: https://drive.google.com/drive/folders/0B3Xmjif6GtQZYmtkczZkM0dndnc
    1. see the README here https://docs.google.com/document/d/16yKV30xgXnfH7_tFDHvFLN2vykdv3V8OmYxknQ_VH9Y/edit
    2. and this page for how to DL large files directly to HPC in the terminal using curl
        1. https://insynout.blogspot.com/2017/01/secret-how-to-download-large-files-from.html
            1. method 1 worked well on HPC
4. to compile MCIP using intel 15.0.3 had to add -openmp flag into makefile
    1. FFLAGS = -openmp $(myFRFLAGS) -I$(NETCDF_DIR)/include -I$(IOAPI_DIR)/include
5. Need to create many links inside CMAQ/lib/x86_64/intel/* that point to the ioapi, NETCDF, and mpi libraries
    1. lrwxrwxrwx 1 jrhorne dabdub 52 Apr 16 16:01 netcdf -> /dfs2/dabdub/jrhorne/WRFCMAQ/lib/NETCDF_intel_15.0.3
    2. lrwxrwxrwx 1 jrhorne dabdub 41 Apr 16 16:01 mpi -> /data/apps/mpi/openmpi-1.8.3/intel/15.0.3
    3. drwxr-xr-x 2 jrhorne dabdub 6 Apr 17 16:38 ioapi
    4. lrwxrwxrwx 1 jrhorne dabdub 52 Apr 16 17:18 include ->

/dfs2/dabdub/jrhorne/WRFCMAQ/lib/Linux2_x86_64ifort/

   5. lrwxrwxrwx 1 jrhorne dabdub 39 Apr 16 17:18 src -> /dfs2/dabdub/jrhorne/WRFCMAQ/lib/ioapi/
   6. lrwxrwxrwx 1 jrhorne dabdub 52 Apr 16 17:18 bin -> /dfs2/dabdub/jrhorne/WRFCMAQ/lib/Linux2_x86_64ifort/
   7. lrwxrwxrwx 1 jrhorne dabdub 52 Apr 16 17:19 lib -> /dfs2/dabdub/jrhorne/WRFCMAQ/lib/Linux2_x86_64ifort/
   8. lrwxrwxrwx 1 jrhorne dabdub 52 Apr 17 12:29 modules -> /dfs2/dabdub/jrhorne/WRFCMAQ/lib/Linux2_x86_64ifort/
   9. lrwxrwxrwx 1 jrhorne dabdub 39 Apr 17 16:38 include_files -> /dfs2/dabdub/jrhorne/WRFCMAQ/lib/ioapi/

6. After running MCIP, ICON, and BCON, many changes will need to be made to the run_cctm script before you can run CMAQ
   1. and you will need to create a separate pbs script to actually submit the job
7. *Probably skipped a bunch of stuff here... kept trying to get benchmark to run but it wasn't working, so I moved to trying to run using Shupengs input files with my compiled executable.*
   1. **This eventually did not work either, so I did NOT run the CMAQ benchmark separate from the coupled model.**

Instructions for previous version of WRF-CMAQ coupled model (this wiki does not yet exist for new one from what I have seen):
https://www.airqualitymodeling.org/index.php/CMAQv5.1_Two-way_model_release_notes

Coupled model source code build scripts link:
https://www.epa.gov/cmaq/access-cmaq-source-code

Building CMAQ for use with coupled model and then assembling and compiling coupled model:

- Did NOT rebuild ICON/BCON after doing so for cmaq benchmark
  - these will not be needed for coupled model if using Shupeng's inputs
- removed/moved all old cctm builds and log files from trying benchmark and shupeng's case
- edited bldit.cctm script as directed (uncomment 2 lines), then built CMAQ
- renamed BLD_* to cmaq and copied into WRFv38 directory, continued with directions
- tried twoway/assemble, but had problems
  - To fix some of them, see this page:https://github.com/USEPA/CMAQ/issues/78
  - NOT ALL CHANGES may be necessary, see the bottom post and only fix/change lines in the assembly/reconfigure file as needed
    - fix *.cfg
    - add / (but doesnt matter for intel)
    - did NOT change LIOAPI
    - MAY have had to make other changes besides these
- Eventually got it assembled, which generates the configure.wrf file
- Then tried to compile and got TONS of errors in compile
  - ***WHEN YOU FIRST TRY TO COMILE, FOLLOW THE "mylog" file and STOP the compilation as soon as you get an error, or else it will run for over an hour and only give errors at the end. You may want to run a "./clean -a" if you have errors, but be SURE that you backup your configure.wrf file FIRST***
    - If you run reassembly after doing a clean -a, it will regenerate the configure.wrf file but seems to add some extra lines in it and extra flags... it may also delete the netcdf_links directory for some reason, which you then need to re-create with links to lib,bin,share,include in your NETCDF_intel_15.0.3 directory. It also may overwrite some makefiles (see modifications below), so you will probabaly want to backup (in a separate directory or name) any Makefiles or configure files that you had to modify.
  - in makefile.intel in cmaq directory had to change MPI_INC = /data/apps/mpi/openmpi-1.8.3/intel/15.0.3/include/
  - also had to change line 3 makefile.twoway.part1 as mentioned on page above
    - IOAPI_INC_PATH = $(IOAPI)/include_files
    - AND made this change in cmaq/makefile.twoway
  - added MPI_INC = /data/apps/mpi/openmpi-1.8.3/intel/15.0.3/include/
    - into: cmaq/makefile.intel and makefile.twoway
    - AND inside of configure.wrf had to change the first line commented out below to 2nd
      - #JRH MPI_INC = /dfs2/dabdub/jrhorne/WRFCMAQ/CMAQ5.2/CMAQ_REPO/lib/x86_64/intel/ -I$(LIB)/mpi
      - MPI_INC = /data/apps/mpi/openmpi-1.8.3/intel/15.0.3/include/

    - in WRFV38/main/Makefile had to add the -openmp flag
        - $(LD) -o wrf.exe -openmp $(LDFLAGS) wrf.o ../main/module_wrf_top.o $(LIBWRFLIB) libcmaqlib.a $(LIB)
- in WRFV38, had to make netcdf_links directory with share, lib, include, and bin links pointing to those directories in the NETCDF_intel_15.0.3 directory where NETCDF was originally compiled
- Had to modify the makefile in WRFV38/main/Makefile and UNCOMMENT the lines for ndown.exe, tc.exe, and real.exe
    - ***later found out that these 3 executables are not necessary – you only need wrf.exe for the coupled model***
        - also had to add the -openmp flag to these files (may not be necessary for all of them)
        - MAKE SURE that these lines start with a TAB (Makefile syntax) after you uncomment them, copy the tab from a different line if necessary
- THEN after those changes I got “Executables successfully built” message after running the “./compile em_real >& my_log” command BUT I got segmentation fault when trying to execute.
- Thus, I backed-up all my modified Makefiles, assembly files, configuration files, etc. (especially configure.wrf)
    - Then did a ./clean -a in the main WRFV38 directory (I think this is master clean)
    - Then re-ran twoway/assemble (said y, y, n in prompt to NOT overwrite the twoway since I know I had to change that one). VERIFY the configure.wrf file did not get extra flags or lines added, do a diff with your original backup.
        - ALSO verify the cmaq/Makefile.twoway because it seems that it APPENDS the same information over and over when you re-run assemble...
    - Then after veryfing configuration and makefiles were correct (YOU WILL ALMOST CERTAINLY NEED TO COPY YOUR BACKUPS INTO THE “main” “cmaq” etc directories), ran the ./compile em_real >& mylog command, and immediately went into a separate terminal and followed the log with tail -f mylog
        - also in a 3rd terminal occasionally searched the mylog for “error”, “abort”, etc, with grep
        - STOP COMPILATION if any errors, fix them, clean, and repeat
-

Coupled model benchmark files:
https://drive.google.com/drive/folders/0B2kjxCwKICxUdzBmT2JCa1MwSW8

Running the coupled model:

1. fixed the variable error in the sample run script as mentioned in link above
2. create a .pbs file like you would normally have for any HPC run
3. TRIED running but get segmentation fault immediately
    1. same thing if I try and just execute the wrf.exe file... immediate segmentation fault with no other information

Attempt at Recompiling....

1. moved/backed up configure.wrf file (and all makefiles as before)
2. ran clean -a in main directory
3. re-ran configure (for WRF), selected 15, 1 for dmpar
4. compared configure.wrf file to old one – this has some addition stuff (see notes)
    1. has additional -f90=$(SFC) on DM_FC and has -cc=$(SCC) -DMPI2_SUPPORT on DM_CC that was not there before...
    2. had to fix MPI path
    3. uncommented flags that have -openmp (OMP and OMPCC)
        1. did NOT manually add -openmp into wrf main Makefile
5. went through and compared all makefiles, left the ndown.exe, tc.exe, and real.exe commented out in wrf main Makefile
6. then tried again... (./compile em_real >& my_log) watching for errors
    1. I can see while watching the log that the -f90=ifort is definitely being used... and so was the -openmp -fpp -auto flags (did not have these before either)
    2. ran this on a compute node with qrsh, bashrc loaded the modules I needed (checked) and the LIBRARY_PATH seemed correct
    3. lots of COMILING WIHTOUT OMP messages in log
    4. made sure I had correct modules loaded (those used for libraries)
7. Finally finished after 2.5 hours → no errors in log BUT it says problem building executables because others besides wrf.exe were commented out.
8. same segmentation fault error immediately. It looks like the start of the program is in module_wrf_top.F and module_configure.F subroutine initial_config
9. When I uncommented real.exe, tc.exe, and ndown.exe and ran compile again I got executables successfully built message
    1. if I try to run real.exe I also get the same crash, immediate segmentation fault in init_resource

<u>MIGHT BE GOOD ONE (issue with optomization):</u>

http://forum.wrfforum.com/viewtopic.php?f=6&t=10356
http://forum.wrfforum.com/viewtopic.php?f=6&t=1625
https://software.intel.com/en-us/forums/intel-fortran-compiler-for-linux-and-mac-os-x/topic/270043#146748

Links for runtime erros (usually about timestep or stack size, not related to mine it seems)
http://forum.wrfforum.com/viewtopic.php?f=6&t=7162
http://www2.mmm.ucar.edu/wrf/users/FAQ_files/FAQ_wrf_runtime.html
https://software.intel.com/en-us/forums/intel-many-integrated-core/topic/393751
http://fortran53.rssing.com/chan-7461311/all_p85.html#item1681
https://software.intel.com/en-us/articles/how-to-get-wrf-running-on-the-intelr-xeon-phitm-coprocessor
http://forum.wrfforum.com/viewtopic.php?f=6&t=10410
http://forum.wrfforum.com/viewtopic.php?f=6&t=9922
http://forum.wrfforum.com/viewtopic.php?f=6&t=3685

Next try with no optimization and -d configure

1. clean -a
2. remake netcdf_links directory
3. ./configure -d
4. twoway/assemble
5. fixed configure.wrf (MPI path, etc)
6. checked Makefiles from JRH2 backups
7. ran compile again....
    1. search for errors and warnings
    2. what is “without OMP”? OMP flag commented out in configure.wrf?
8. deleted output_* folder
9. PRODUCED SAME EXACT SEGEMTNATION FAULT – was able to get some backtrace (see .txt file in WRFV38) and it seems like it was a problem with NETCDF library based on google search.
10. Also tried pointing to Shupeng's NETCDF_intel_15.0.3 and same crash
11. *USE THIS COMMAND BEFOR BUILDING setenv WRFIO_NCD_LARGE_FILE_SUPPORT 1*
    1. *also see tab that had several commands before compiled*
12. **NEXT TRIED A FRESH REBUILD OF CMAQ AND WRF WITH gcc**
    1. **also requires redoing IOAPI, NETCDF, etc... see appendix/marc instructions for that**

Recompiling NETCDF with:
Currently Loaded Modulefiles:

```
  1) intel-parallel-studio-xe/15.0.3   3) hdf5/1.8.14
2) openmpi-1.8.3/intel-15.0.3        4) zlib/1.2.11
```

Becuase of this page matching my segmentation fault backtrack error:
https://www.unidata.ucar.edu/support/help/MailArchives/netcdf/msg13410.html

See this page:
https://www.unidata.ucar.edu/software/netcdf/docs/getting_and_building_netcdf.html#build_parallel under Building with Parallel I/O Support
Top of modified setting-intel.sh file:

```
#!/bin/bash
export H5DIR=/data/apps/hdf5/1.8.14
export ROOTDIR=/dfs2/dabdub/jrhorne/WRFCMAQ/lib
export SRCNCDF=${ROOTDIR}/scrNETCDF
export INSNCDF=${ROOTDIR}/NETCDF_intel_15.0.3
```

note added H5DIR there, ONLY worked with newest 1.8.14 loaded

Essentially used these commands:

```
$ CC=mpicc CPPFLAGS=-I${H5DIR}/include LDFLAGS=-L${H5DIR}/lib \
./configure --disable-shared --enable-parallel-tests --prefix=${NCDIR}
$ make check
$ make install
```

1. first one went OK after loading 1.8.14 and adding path
2. YOU MUST RUN make BEFORE running make check
3. then ran make install
4. did NOT use shupeng's setting-intel.sh script, so not sure how it even found hdf5 directory
5. got Congratulations! You have successfully installed netCDF! Message still... seems OK
6. I had loaded 1.8.14 but my old export for the lib was for 1.8.13 still....

Then went into fortran-4.4.4 folder for netcdf

1. made some changes to setting-intel.sh file, trying a bunch of different crap when it didnt work and googling stuff
2. sourced that file
3. ***FAILED AND GAVE UP – dont think hdf5 on HPC works for shared libraries***

Trying to compile with option 16 (sm+dm), default 1, and mpiifort and mpiicc in configure.wrf (also has the -D_OPENMP flag on)

- normal procedure with renaming netcdf_links, clean -a, configure, fix configure.wrf, twoway/assemble, check makefiles, configuration, wtf
- had to manually export the variables in JRHmpivars.sh one at a time, after exporting
  - I_MPI_ROOT=/data/apps/intel/parallel_studio-xe/2015-update-3/impi/5.0.3.048
  - could not run the mpivars.sh, or at least it would not make a difference in my variables and the compile command could not find mpiifort compiler in the LD_LIBRARY_PATH variable
- finally ran ./compile em_real >& my_log after doing all that
- NOTE SURE about MPI_INC_PATH... maybe I should have used:
  - /data/apps/intel/parallel_studio-xe/2015-update-3/impi/5.0.3.048/include64/
  - but the path I gave is what I compiled NETCDF and stuff with...

STARTINC CMPLETELY OVER in gccWRFCMAQ

following these instructions for isntalling NETCDF and WRF stuff: **http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php#STEP8**

- Ran the “Library Compatibility Tests” and got success for all of them
- Ran ./configure in WRFV38 with 34-1 options

GAVE UP ON THIS because I could not build IOAPI with that old version of NETCDF listed on that page.... see next page

STARTINC CMPLETELY OVER in gccWRFCMAQ attempt #2, all in the same terminal without exiting

```
Currently Loaded Modulefiles:
  1) gcc-libs/5.3.0           3) openmpi-1.8.8/gcc-5.3.0   5) zlib/1.2.8
  2) gcc/5.3.0                4) hdf5/1.8.14
```

- Started by installing NETCDF 4.4.1.1 following Marc's instructions
    - see my .sh settings file for gcc
    - Got congratulations message → ran make check again to be safe
- Then installed NETCDF-fortran-4.4.4 follwing his instructions
    - see my .sh settings file for gcc (the one in the fortran directory)
    - Got congratulations message → ran make check again to be safe
- DID NOT USE wget TO GET A NEW ioapi archive....
    - I tried this but it was a tiny file and would not even extract
    - I manually downloaded this file from the internet and it was slightly bigger than Shupengs archive, so I just used his...
        - ***Why are all of these md5sum different?***
- Followed Marc's instructions pretty much EXACTLY for IOAPI, including the IOAPI directory, commenting out BIN, Line 61 edits, etc.
    - ALSO, in the main Makefile in IOAPI, had to change
    - NCFLIBS   = -L${NCDIR}/lib -lnetcdff -lnetcdf
        - e.g., added the -L part or else it would not find -lnetcdff when compiling m3 tools
    - finally did make all and it seemed to work...
- final message is:
    - "make[1]: Leaving directory `/dfs2/dabdub/jrhorne/gccWRFCMAQ/lib/IOAPI/m3tools'"
- RE-downloaded WRFv3.8.TAR.gz and checked md5sum against my old one, its fine
- Extracted using tar zxvf
- Renamed directory
- Went inside, did printenv, and changed "NETCDF" variable
    - when I first ran configure it was it was using a path that did not actually exist (was based on this variable from some point before... so I exported:
    - export NETCDF=/dfs2/dabdub/jrhorne/gccWRFCMAQ/lib/NETCDF_gcc/
- Then ran ./configure
    - did NOT get warning that Shupeng did when using mpich and earlier gcc
    - NETCDF path looked good
    - generated configure.wrf (this will be changed later after running twoway/assemble)
- Then need to get CMAQ5.2
    - go back out of WRFV38 directory and get it:
        - `git clone -b 5.2 https://github.com/USEPA/CMAQ.git CMAQ_REPO`
- went into CMAQ_REPO directory and changed line 28 of config_cmaq.csh
    - setenv CMAQ_REPO /dfs2/dabdub/jrhorne/gccWRFCMAQ/CMAQ_REPO
- went into CCTM/scripts and made the 2 changes for twoway model
- These seem to be the only 2 file changes that Shupeng made when he compiled using mpich and older gcc compiler that partially ran
- ***Made additional changes not described here, although it may not be necessary because I am not actually compiling CMAQ and it may get this information from the configure.wrf file***

because Shupeng did not make these changes when he built compiled model... just want to be sure

- Had to create symbolic links to libraries
- Then went into CCTM/scripts and ran ./bldit_cctm.csh gcc
  - got message “Makefile generated”
- Renamed BLD_CCTM_v52_gcc to cmaq and copied to WRFV38 folder
- Redownloaded coupled model tarball, same md5sum as old, extracted, moved twoway folder
- BEFORE doing two/assemble, made fixes on bug site
  - 2 changes in reconfigure file (*.cfg and / at end of line)
  - also had to add setenv FC gfort to line 12 of reconfigure for some reason...
- then twoway/assemble ran, and said “y” to all 3 prompts
- got new configure.wrf file, looked through it, compared to old intel one... everything looked OK for paths and whatnot, didnt have to make any modifications (probably because I took time to set up cmaq configuration with paths above)
- tried compile and fialed right away on mpi error → sent MAYDAY to HPC
- ***EVENTUALLY TRIED WITH NEW GCC/OPENMPI COMPILER VERSION AND ABANDONED THIS OLD GCC ONE → see below***

WHILE WAITING FOR HPC TO REPLY: SEE THESE TWO LINKS ABOUT THIS ERROR:
http://forum.wrfforum.com/viewtopic.php?f=5&t=3660
https://translate.google.com/translate?hl=en&sl=ja&u=https://qiita.com/7of9/items/a3c692585bf6b905fff3&prev=search
http://forum.wrfforum.com/viewtopic.php?f=5&t=7093

- Following those suggestions, I added the “`-DMPI2_SUPPORT`” flag to DM_CC in the configure.wrf
- then did a clean -a and tried to recompile to a separate log file (since I have a support open with HPC still)
- Had the wrong mpi_inc path because my “mpi” link in the CMAQ_REPO/lib/x86_64/gcc/ folder was wrong. So I remade the link:
  - mpi -> /data/apps/mpi/openmpi-1.8.8/gcc/5.3.0
  - rather than changing the path in the configure.wrf file since CMAQ was built previous expecting this to be the mpi path (the /include, /lib, etc, directories are inside here).
- Did another clean -a, then recompiled
  - Did not check makefile, or rerun assemble before recompiling.
- Go another error, see this link:
  - https://stackoverflow.com/questions/9685377/undefined-reference-to-omp-get-max-threads
  - Uncommented -fopenmp flag on 2 lines in configure.wrf
- Ran another clean -a, and then started compile AGAIN with those 2 flags uncommented
  - got an internal compiler error...
  - internal compiler error: in gfc_trans_use_stmts, at fortran/trans-decl.c:4605
- Kept replying to HPC emails, probably stopping here for now...
- ***EVENTUALLY TRIED WITH NEW GCC/OPENMPI COMPILER VERSION AND ABANDONED THIS OLD GCC ONE → see below***

HPC replied and built a new compiler version for me to try: openmpi-2.1.3/gcc-6.4.0

```
Currently Loaded Modulefiles:
  1) gcc/6.4.0                 2) openmpi-2.1.3/gcc-6.4.0   3) zlib/1.2.8
```

- moved gccWRFCMAQ to OLD_gccWRFCMAQ to start fresh here with the same install path for all my other config files when I go to rebuild everything
- basically started completely over as before with build.... starting from archives
- Downloaded HDF5-1.10.2 from the first link below, followed instructions for parallel build from 2nd and 3rd links. Other links are for additional information
  - https://www.hdfgroup.org/downloads/hdf5/source-code/
  - https://www.unidata.ucar.edu/software/netcdf/docs/getting_and_building_netcdf.html#build_parallel
  - https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.10/hdf5-1.10.2/src/unpacked/release_docs/INSTALL_parallel
    - https://support.hdfgroup.org/HDF5/PHDF5/
    - https://support.hdfgroup.org/HDF5/release/obtainsrc.html#conf
    - https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.10/hdf5-1.10.2/src/unpacked/release_docs/INSTALL
- Ran HDF5 configure with:
  - `CC=mpicc ./configure --enable-parallel --prefix=/dfs2/dabdub/jrhorne/gccWRFCMAQ/lib/HDF5`
  - Features after configure how that parallel HDF5 is Yes
  - also saw that it found zlib when configure was running
  - After that, export H5DIR=/dfs2/dabdub/jrhorne/gccWRFCMAQ/lib/HDF5
- Then ran make → this took quite a while with a bunch of warnings
- then make check → seemed to pass everything, took quite a long time to run (like 1h or more)
  - got some werid errors on very end about mpiexec exiting but I was on login node... also got this warning ignored a bunch of times (like seen in other Airshed runs and stuff?)
  - [hpc-login-1-3.local:11378] mca_base_component_repository_open: unable to open mca_coll_hcoll: libsmx-1.3.1.MLNX20170625.859dc24.so: cannot open shared object file: No such file or directory (ignored)
- then make install → got this note:

```
Libraries have been installed in:
   /dfs2/dabdub/jrhorne/gccWRFCMAQ/lib/HDF5/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
   - add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
     during execution
   - add LIBDIR to the 'LD_RUN_PATH' environment variable
     during linking
   - use the '-Wl,-rpath -Wl,LIBDIR' linker flag
   - have your system administrator add LIBDIR to '/etc/ld.so.conf'
```

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.

- Copied netcdf4.6.1 and fortran 4.4.4 archives from OLDgcc, and also IOAPI3.2 archive, and then created the same directory structure as before in lib/ directory and extracted these archives
- After finishing with make install, moved on to netcdf installs
- created a custom setting-gcc.sh file for 4.6.1 based off of unidata link above. Basically just followed their command exactly but REMOVED –disable-shared flag, added some of Marc's paths at the top, but did not include all of Marc's flags for compiler settings (these might be required for Fortran? Will see I guess)
    - commented out make all check install (doing separately)
    - ***should I have added -fopenmp flags here? Since used on WRF? Is that what made the intel one work, or was it newer netcdf4.6.1? (note that shupeng used older netcdf on his that crashed in same place)***
- Sourced that file to run configure
    - I can see that it has parallel support and found zlib also (loaded module)
- then ran make → then make check (all passed) → then make install
    - got congratulations message
- used essentially the same fortran settings.sh script as my old GCC (copied here), commented out make lines
    - never exported most of the variables Marc said I needed to in first C netcdf isntall
- ran script to configure
- then ran make → make check → make install
    - ***I didnt export the BIN variable, so where did all this stuff go??***
-
- DAY 2 RESUME HERE by installing IOAPI, extracting WRF, then CMAQ, assemble, etc
-
- Because this terminal exited (broken pipe) over the weekend, created a “pre-setting-gcc.sh” file to export all the variables I created while doing NETCDF install, before moving on to IOAPI3.2
    - some more IOAPI install notes:
    - https://github.com/cjcoats/ioapi-3.2/blob/master/README.txt
- Made a bunch of changes to the Makefiles, mostly following Marc's directions, but also had to hardcode a lot of file/folder paths to get it to compile
    - also exported variable for large file support
- finally ran “make all” and it seemed to go OK
    - I can also see that it is using “-fopenmp” flag during compile
    - **did not manually add HDF5 path anywhere, so it wasn't defined at at this time**
- Seemed to finish OK, final line was:
    - make[1]: Leaving directory `/dfs2/dabdub/jrhorne/gccWRFCMAQ/lib/IOAPI/m3tools'
- Downloaded Jasper from here:
    - http://www.ece.uvic.ca/~frodo/jasper/#download
    - first tried 2.0, but wouldnt work, then went back to earlier 1.900.1 given in WRF instructions
    - GOT THIS MESSAGE: Libraries have been installed in:
    - /dfs2/dabdub/jrhorne/gccWRFCMAQ/lib/Jasper/lib
- **tried to export JASPER variables but still wasnt going to work based on configure message, so just unloaded those variables and configured without**

- Copied CMAQ_REPO directory from previous gcc build into this one → edited build and configure cmaq files to make sure I gave NEW path for gcc/openmpi version I am using.
    - Also had to update the “mpi” link in the cmaq lib folder for new compiler, everything else essentially the same for IOAPI and NETCDF (even though newer version)
- built cmaq as mentioned above and then renamed folder and copied into WRFV38
- extracted twoway folder from coupled model tar → did diff with OLDgcc one → made fixes to files as mentioned above
- ran twoway/assemble
- checked configure.wrf and made the changes above for -fopenmp flag and “-DMPI2_SUPPORT”
    - can also see that this configure.wrf has HDF5 in it
- then ran ./compile em_real >& my_log
    - somehow messed up mpi link in CMAQ lib folder... did clean -a, fixed link, and started compile over
- Ran into another problem when compiling wrf.exe → had to remove -lhdf5_fortran flag from configure.wrf (**I guess I didnt give a fortran option when building hdf5)**
    - **https://translate.google.com/translate?hl=en&sl=zh-CN&u=http://debug.fanzheng.org/post/hdf5-error-when-compiling-WRF.html&prev=search**
- Did NOT clean after this error, just ran compile again → created wrf.exe executable
- Ran into error trying to run, for some reason it was using completely wrong modules and could not find proper include files for gcc/openmpi...
    - FINALLY figured this out → there is a .cshrc file in ~/ that was being sources **EVERY TIME I USE A csh SCRIPT** →This was overwriting the modules I had loaded with incorrect paths for include files and whatnot, causing wrf.exe to fail.
    - **Had to edit ~/.cshrc script and COMMENT OUT the module load Cluster_Defaults command**
        - Also edited ~/.bashrc and did the same since I am always using different ones now
- After making that fix, kept trying to submit runs but they were crashing from within WRF.EXE → seems to be an actual problem with WRF configuration/namelist, not with compilers..
    - http://forum.wrfforum.com/viewtopic.php?f=9&t=5894
    - http://www2.mmm.ucar.edu/wrf/users/phys_references.html
    - http://mailman.ucar.edu/pipermail/wrf-users/2016/004316.html
- Tried changing the NUM_LAND_USE_TYPE variable to all other suggested values but seemed to make it worse (more NAN values when FATAL call happened in rsl error files), went back to 40
- Tried changing physics schemes
    - Then crashed with “cfl” problem → time step issue?
    - http://forum.wrfforum.com/viewtopic.php?f=8&t=357
    - tried reducing time step from 20 to 10 inside two way model run script → then later tried going as low as 1 sec or even 0.2 seconds and STILL crashed with cfl errors toward very beginning of run...
- Tried with a variety of different combinations of physics schemes as suggested here under “examples of namelists”
    - http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3/users_guide_chap5.htm#Namelist

- No matter what still getting cfl errors, seems like it is happenening on edge of domain.
- Also tried with changing/adding/adjusting vertical damping, adding epssm parameter with different values, etc.
- ***THIS VERSION OF THE MODEL ISNT EVEN GETTING AS FAR AS MY INTEL AND SHUPENG's VERSIO THAT WOULD CRASH WHEN CMAQ STARTED***
    - ***going back to try intel***

WOULD GO OFF TOP:

The "WOULD GO OFF TOP" messages mean vertical CFL criteria violations. You do have to reduce your timestep further or decrease your vertical level spacing. It can happen in (primarily) convective situations that there are high vertical velocities and your timestep is too large to be able to resolve those (hence the CFL error).

Where is your domain? If you are runnning the model for mountainous regions these errors are even more likely to occur. In this case you can also opt for smoothing your terrain data first so that you get less steep slopes. For instance, when I did simulations for Switzerland at 1 km resolution, without terrain smoothing, I had to use a timestep of 1 second to avoid violations to the CFL criterium - which makes the model very slooooooow...

In most cases you can re-set the timestep to a larger value after after several hours (so to speed up your model run you could consider writing restart files, running the first say 5 hours with your original (120 sec) timestep, then running as long as needed with a much shorter timestep, then the rest of the simulation again with a larger timestep).

----

as I see from your rsl. file, instability occurs from vertical velocity. I suggest: Put the upper level damping flag damp_opt=1 and dampcoef to something between 0.01 and 0.1. For me it works fine with 0.01 even in steep mountains. If this still doesn't work you can try additionally to put w_damping=1.

<u>SEPRATE TERMINAL – tried to rebuild a newer NETCDF with intel 15.0.3 and then recompiling WRF-CMAQ using that</u>

```
Currently Loaded Modulefiles:
  1) intel-parallel-studio-xe/15.0.3   3) gcc-libs/5.3.0             5) hdf5/1.8.14
  2) openmpi-1.8.3/intel-15.0.3        4) gcc/5.3.0
```

1. found out that HPC HDF5 does not have parallel support so gave up on that but added -openmp flag to lots of stuff and built newer version of NETCDF (see setting-intel.sh files for both newer NETCDF C and the same F version rebuilt after the newer C).
    1. got success on both, good make check
    2. Note the modules I had loaded, not sure if the hdf5 version that autoloaded gcc made a difference
2. then went and clean -a → checked stuff, made netcdf_links, fixed mpi_inc path in configure.wrf, recompiled wrfcmaq before leaving
3. There were some errors when I got here on Friday because I messed up the mpi_inc path in configure.wrf. Fixed that and did a clean -a, make netcdf_links, fixed configure.wrf, and then recompiled.
4. It seemed to work... got wrf.exe → checked run script and .pbs → submitted
    1. had to uncomment the export LD_LIBRARY_PATH and other line in PBS run script since I removed these from my bashrc
5. IT ACTUALLY RAN BRIELFLY → then crashed at the same place as Shupeng's
    1. see rsl.out and rsl.error files in output directory
6. As soon as this run finished, I scp the WRFV38 directory to home on tarrega as a backup
    1. this does not contain the CMAQ_REPO or the input/output files, just the built 2-way model

RESUMING THIS LATER:

- Went back to CMAQ_REPO, edited the configure_cmaq.sh script properly based on how I did gcc flags, for example
    - added the paths to the netcdf, etc
    - added the flag for openmpi that was missing before
    - manually copied mpif.h into the BLD (aka cmaq) folder since I noticed in the build log it could not find it.
    - lots of other stuff
- built this cmaq → renamed → copied to WRFV38
- configured wrf from scratch again using option 15-1 (default nesting). Then did twoway/assemble after EXPORTING netcdf path because it originally could not find it
- checked configure.wrf against previous ones to make sure everything looked good
- then compiled em_real → did successfully first try, no problems
- WOULD JUST HANG ON STARTUP if using 32 cores → try less with gcc?
- Tried with 16 and it works → 25 does not, basically just use 16 or less
- Did LOTS of tests with different # of cores, different nodes (pub8i just crashes)
- Tried changing buffered files to real → runs into problem with those files
- changed ONLY the MET_CRO_3D file to real → did ncdum → I can see DENSA_J in it...
-
- **SENT EMAIL TO DAVID WONG**
-
- He responded and sent me "internal EPA ioapi v3.1"
    - downloaded this → coped to HPC → made new IOAPI directory → extracted → installed

following his instructions (did not modify makefiles like Marc)
    - It seems to have found NETCDF_intel_15.0.3/lib okay, used -lnetcdf -lnetcdff
    - seemed to go OK, final line was leaving directory for m3tools
- remade ioapi links in CMAQ_REPO/*/lib folder
- modified configure_cmaq.csh script to point to new IOAPI paths
- rebuilt cmaq → copied mpif.h as before
- did clean -a in WRFV38
- configured WRF from scratch again, with large file support = 1 exported
- copied cmaq into WRFV38
- ran twoway/assemble
- uncommened openmp flags in configure.wrf
- fixed mpi link in cmaq/*lib* directory (had to do this before also, must be from cmaq build script)
- ran compile and watched log for errors
- got some errors, couldnt find state3 file...
- had to go into CMAQ/lib* folder for ioapi and create "include_files" link to fixed_src
    - see cmaq Makefile.intel for why it needs this path
- then ran compile again, did not clean before → produced executable
- submitted run, IT WORKED FURTHER, then ran into issue writing to another file
- adjusted some filetimestep/TSTEP parameters and it seemed to get further but producing HUGE files
-
- **SENT EMAIL TO DAVID WONG AGAIN**
-
- ***After setting the TSTEP and FILE_TIME_STEP parameters to 000140 (1min 40 sec which is 5x WRF_TSTEP since coupling frequency is set to 5), the model did a complete run but produced HUGE output files, especially C_GRID***
-
- After emailing with David, it seems that C_GRID is the restart/resume file for CMAQ. The model simply would not run with the file time step value set to 010000 (1h) like it is supposed to be because the S_CGRID file needed a timestep of 000140. But setting the file time step for ALL the files to such a small value was not practical.
- Next I looked inside the "cmaq" folder for where this CGRID file is created, which is based on driver.F and wr_cgrid.F files. The driver.F determines when the file gets written to by calling a subroutine in the wr_cgrid.F file. After reading the documentation in wr_cgrid.F it seems that the CGRID file should be a "circular-buffer" file that gets overwritten after 2 steps so that it only saves the most recent 2 time step concentration outputs. BUT L119 of that file that sets the negative time step required to create this type of file was commented out, AND it would have the wrong value for TSTEP (010000) anyways.
    - **So I modified L119 to be: "TSTEP3D = -000140"**
    - See this think about this "circular-buffer" file **https://www.cmascenter.org/ioapi/documentation/all_versions/html/DATATYPES.html**
- After modifying that line of code, set the FILE_TIME_STEP and TSTEP in the twoway run script back to 010000 as they came, then ran "./compile em_real >& mylog_recompile" to rebuild with the modified code
- After recompiling, submitted a run with 16 cores (instead of 8) and it FIXED the problem with

S_CGRID crashing but then crashed later on because of “NEGATIVE or UNDEFINED Dry Deposition Velocity” “ *** ERROR ABORT in subroutine M3DRY on PE 006”

  - Random error that wasnt there before and didnt happen with 8 cores using TSTEP as 000140 so must be related to # of cores...
- Then tried submitting with 24 cores and it ran for a few minutes (000820) and then crashed after some cfl errors with “ Could not determine Courant-condition safe sync step for model step: 010000 HHMMSS (Max vel)/(dX) = 9.140E+01” and “ERROR ABORT in subroutine ADVSTEP on PE 021”
- Tried with 20 cores and got lots of cfl errors after 26:00
- tried with 12 cores and eventually crashed after 2:10:20 due to cfl errors and “RIBX never exceeds RIC”
- pub8i crashes with unknown segmentation fault
- tried changing wrf_tstep in run script to 15 → edited wr_cgrid.F to 000115 accordingly, recompiled, submitted run with 24 cores
  - crashed with cfl errors after 10:45
- tried with 16 cores with this 15 sec tstep → crashed with cfl errors after 2:05:00
  - usually only a few points... like 1-7 have cfl errors
- tried with 16 cores on free32i (15 sec tstep) →ran for 2:05:00 crashed at same spot but got there faster...
  - so this intel node works but pub8i did not
- tried with 24 cores on free32i (15 sec tstep), added epss=0.25 and changed dampcoef to 0.1 --> crashed after 10:45 with lots of cfl errors
  - also get weird processor clock speed warning or something on intel node
- tried same thing but with epssm=0.3 and dampcoef to 0.2 → crashed after 10 mins
- tried 23 cores, 15tstep, epssm=0.3, smooth_cg_topo=.true., crashed at 2:52:15
- tried 15 cores, same as above, started getting cfl errors around 1:32:30 on node 14, but KEPT RUNNING PAST THIS? Maybe there is a threshold for how many, this was just a few points
  - crashed shortly after... looks like the actual problem was RIBX never exceeds RIC, RIB even though there are lots of cfl
  - it actually looks like all the other ones that crashed after cfl were due to this same problem
- NOTHING WORKING so...
- went back to 8 cores, default 20 sec tstep, basically everything that I did on the run that actually completed except I left file time step in the run script as 1h but recompiled the model with the wr_cgrid.F modified to have -000140 time step for the S_CGRID file. This should run to completion.... (although used air64 instead of air but should not make a difference)
  - **THIS RUN COMPLETED SUCCESSFULL (got message in rsl.* files)**
- Comapred my ACONC to the benchmark ACONC and there are huge ozone differences 5-28ppb east of Reno where there seems to be like nothing...
  - This run uses essentially all unchanged parameters in the run script (no change in physics, timestep, etc)
  - But when I look at the two scenarios plots of MD8h ozone side by side they look very similar... either way its probably fine since the goal for our modeling would be to look at deltas versus a reference case
- TRY Shupeng inputs → SEE BELOW
- FOR POST PROCESSING – TRY NCL (HPC has ncl installed), see shupeng sample scripts and these links: **https://www.ncl.ucar.edu/get_started.shtml#RunningNCL**

After successful completion of benchmark, attempting to use Shupeng's inputs for CONUS, Summer 2011

- Copied WRF inputs for a few days in July from Shupeng's external to HPC
    - made new “inputs” directory
- Created link to Shupeng's CMAQ emissions inputs (see email he sent)
    - also in this inputs directory
- Make a backup of twoway_model_run_script before changing
- Started editing this script:
    - Copied Shupeng's namelist.input into the twoway run script wrf namelist section (just replaced everything in the original benchmark simulation) (the namelist.input is from the external HD with the WRF inputs)
        - note that this hard codes a lot of stuff instead of using variables defined in the run script
            - (tried to update some of these accordingly, but most are not even used anymore since the values are hard-coded in the namelist)
        - His does not use the “dda” files for nudging that were used in the twoway benchmark
        - I had to DISABLE (remove) the adaptive time step parameters from the &domains section of the namelist (everything from use_adaptive_time_step to the end of the section)
            - Then hard-coded the value of time_step in this section to be 35 (from test runs it seems that it was in the 40 range, so selected this to test).
            - Then I had to edit the wr_cgrid.F file to use a time step of -000255 (5*35)
                - Then recompile the model (no need to clean, just run ./compile em_real)
        - Various other edits to start date, end date, etc, based on my needs
    - HEAVILY modified the CMAQ parts of the script by looking at the run.cctm script in /dfs2/dabdub/shupengz/Models/CMAQ-5.2Gamma/run/US/2011-sum-cb6
    - basically used MOST of what was in his, **but left some stuff disabled like lightning NOX**
        - see #??? and ### JRH parts
    - hard coded some input files for a single day – will need to be changed later for multi-day simulations
        - NOTE that there are different paths for some files depending on if its the 1st day of simulation in CMAQ or of its 2nd day and onward
    - Had to change CMAQ col_dim and row_dim based on Shupeng's MCIP file
        - /dfs2/dabdub/shupengz/Models/CMAQv5.1/data/run.mcip
    - Also commented out the setenv CONC_BLEV_ELEV line to print all layers
        - may not be required, but the original coupled model run script had it set to all layers based on NZ → not sure what NZ is so just printing all layers incase coupling needs this
- After editing the run script based on Shupeng's wrf namelist.input and the run.cctm script, and recompiling the model after modifying wr_cgrid.F, submitted a run still using 8 cores
    - crashed on drydep file → seems to be because 35sec tstep does not line up neatly with 1h output file time step
- CHANGED wrf time step to be 30 sec in run script, also edited wr_cgrid.F to have -000150 time step, recompiled
- tried to submit with 24 cores but just stalled after startup... no idea why
- changed to 16 cores and submitted run with 30 sec tstep
    - ran for about 3hours of simulation time or so, but then crashed with VPPM error

        - http://www-personal.umich.edu/~sillman/CMAQ_corrections_2010.htm
        - https://www.cmascenter.org/cmaq/documentation/4.7.1/Operational_Guidance_Document.pdf
    - Shupeng said he normally fixes these by changing the CMAQ domain decomposition for NROW/NCOL, but I can't control this in the coupled model? Can I see it in the rsl files??
    - ended up modifying the cmaq source code (in “cmaq” directory):
        - changed MAXITER to 50 in zadvppmwrf.F
        - changed MAXITER to 100 in vppm.F
- Recompiled after those 2 changes, resubmitted with 16 cores
    - NOTE: tried 18, 20, 22, 24 and would not run with any of those... (hang on startup)
        - Shupeng said no odd # but it worked on benchmark partially with odd ##??
    - It crashed with vppm error in exactly the same spot, on exactly the same node (11)
- Changed to 15 cores, resubmitted to air
    - if this does not work, try using workaround that prevent catastrphic error from the link above (basically use/integrate those 2 fortran codes into my cmaq source codes)
    - crashed after about 1.5 hours of model time... for NO apparent reason
- Tried to modify vppm.F based on sillman code
    - for now the main thing that I did was comment out the M3EXIT command in the vppm at 77 if statement so that it would not crash at this non-convergence area
    - also added the “zero protect” to the two main IF statements in there
    - this could cause infinite loop, will need to watch
    - THIS DID NOT WORK – basically did cause infinite loop and velocity kept increasing and printed into rsl.out file for that node, making that file get bigger and bigger. So velocity was clearly diverging for that code
- Tried the same thing as above but recompiled with wrf time step set to 60 sec, made corresponding changes in run script, submitted with 16 cores
    - **SHUPENG GETTING SAME VPPM ERROR IN RCP 8.5 CALI simulations**
- added a GOTO to try and skip a cell if convergence was failing in vppm.F instead of the M3 exit, no idea what effect this is going to have... GO TO 55 and 55 CONTINUE
    - This did not work. Tried to run with 16 cores on air and it crashed after 031500, which is very slightly further than VPPM error, might have crashed from my GOTO beacause there is no clear backtrack for the error/abort
    - SHOULD PROBABLY RECOMPILE WITH UNMODIFIED VPPM unless I am going to try full workaround
- Submitted with 14 cores (still with the vppm.F changes described above, 60 sec tstep, air node), ran for 9:55:50 and then crashed with NaN in Vd for m3dry error
    - saved this output, not sure what to do besides try different number of cores...
    - tried googling this error but not much documentation
    - **<u>LOTS OF VERTICAL PROBLEMS... TRY wrf DAMPING OR SOMETHING?</u>**
- Tried to submit to free 40i with 14 cores → would not run, so **IT SEEMS THAT MAX # of CORES DEPENDS ON THE SPECIFIC QUEUE SUBMITTED TO**
- Changed to 12 cores and submitted to free40i → runs and seems to do about 2:1 speed
    - still 60 sec tstep, same vppm.F mods, etc.
    - crashed with floating point exception for CFRAC_3D → lots of NAN after about 3:15 or so
        - (similar time to when vppm errors were occuring previous, could be due to my modifications to the code → **maybe try this again after recompiling with normal**

code)

- Recompiled with normal vppm code
- Tried on bigmemory with 24 cores, it started up OK but crashed with vppm errors after only like 7 minutes into simulations
- tried on bigmemory with 36 cores, would not startup
- submitted on bigmemory with 28 cores and it started up OK
    - crashed with vppm error 77 after 43:45 (with unmodified vppm)
- to test my vppm modifications, recompiled with my modified vppm code and submitted exact same run to bigmemory with 28 cores
    - this run seemed to go just a little further and then crash with cfl errors
- tried with 29 cores on bigmemory (30 would not run)
    - got abort after 5 mins, deleted output
- <u>**TRIED IMPLEMTNING FULL SILLMAN FIX HERE**</u>
    - *tried a whole bunch of runs shown below... was either not working or crashed immediately when starting to do vertical advection. I think it messed something up in the z*vppm.F code because it seemed to be printing those error messages... then I accidently deleted my modified version of these codes because I did mv instead of cp*
        - tried running on epyc with 32, 24, etc etc etc all the way down to 16 cores and would not run to matter what, even when I recompiled with unmodified vppm code
        - tried like 50 times to submit jobs to free40i, free88i, epyc, etc, with different number of cores and they would NOT start up for some reason... even things that worked before like free40i with 12 cores.
        - It finally ran on free88i with 8 cores but crashed right at the beginning of wrf...
        - submitted to bigmemory with 28 cores, 60 sec step, all my sillman vppm fixes implemented...
            - crashed right away
            - tried 24 cores → crashed right away
            - tried 20 cores → crashed right away
            - same with 14....
- recompiled with normal vppm/z*vppm → submitted exact same run as before with 14 on bigmemory
    - crashed right away (seems to being during first CMAQ calc after first 5 wrf steps)
- even with runs that worked before, with unmodified vppm compiled, submitting with 8 or 12 cores on free40i would NOT run at all
- submitted to bigmemory with 8 cores, this is completely unmodified version recompiled with 60tstep (only mod is more iterations allowed in parameters)
    - would not run...
- copied unmodified vppm.F and z*.F from CCTM src folder, recompiled.
- Submitted to free40i with 10 cores → crashed with vppm 77 error at 3:26:15

<u>**Started Trying AUG inputs**</u>

- Copied 20110804 wrf inputs from Shupeng external
    - note this is actually a starting day of 0803
- verified namelist was the same for wrf
- modified the twoway run script for this new simulation time
    - had to hard code LOTS of values in there to be “8” instead of “08” (error in csh script

    thinking 08 is a hex number or something)
  - probably should have changed the number up top to 8 and then added the 0 where it was needed instead
  - changed all the expected stuff like CMAQ files
- tried to submit to free40i with 12 cores but would not run
- submitted to bigmemory with 20 cores
  - crashed with vppm 66 error after like 2:30 or so
- **modified damp_opt in wrf namelist to be 3 and use damping**
- submitted to free40i with 12 and it started (more cores were available)
  - crashed with unknown segmentation fault after only like 20 mins in
- submitted to air with 16 cores and it started (rest of node unused)
  - running at about 2:1 speed (slightly less), OK so far at 5:00 it seems
  - CRASHED at 5:15.. (after WRF did 5:20)
    - drydep error on 2 cores
    - there are also lots of cfl errors in wrf on various cores, but some occurred much earlier before the crash and it kept going a while longer
- Changed:
  - **damp_opt to 1**
  - **dampcoef to 0.1**
  - **wrftstep to 45 sec** → recompiled model with same change in wr_cgrid.F
- tried submitting to air again with 16 cores... wont run... 14 cores... wont run... no idea why
- submitted to free40i with 12 cores after making changes and recompiling as described above
  - took SS of where it was before leaving for the weekend, left 2 terminals open with tail -f
  - **THIS RUN COMPLETED SUCCESSFULLY while I was gone. Got successful completion of wrf message at end of 20110803. The problem is that it did not create a wrf restart file because the restart_interval from shupeng's namelist was way too long.**
- Made the following changes in the twoway run script
  - run_days = 1
  - restart_interval = 1440
  - also changed path for ocean file since Shupeng moved to /a2 directory
- submitted new run to free32i with 18 cores
  - run crashed with segmentation fault 01:28:30
    - not sure why, some cfl errors but nothing else obvious
  - also seems like run_days=1 did not prevent it from starting the next day?
- resubmitted same run with but 20 cores on free32i
  - crashed during wrf at 03:47:15, again for no reason...
- resubmitted same run on free40i, 12 cores → same configuration that worked before
  - SUPER slow so I killed it
- rebutmitted same thing on free32i, 12 cores → running almost 3x speed for first hour
  - crashed at 11:54:00 in the middle of wrf iterations it seems.... nothing obvious
- resubmitted same run on free40i, 12 cores → same configuration that worked before
  - the rest of the node was open when I started so it was running decently fast at about 3x
  - after 3h seems to slow down to a crawl.... barely moving
  - it was going way too slow so I killed it, could have been because someone started running a MATLAB job on the node

- resubmitted same thing on air, 12 cores
    - seems to go about ½ the speed as on an intel node with same # of cores
    - crashed at 115230 for no reason... seems to be inside wrf so going to try smaller tstep
- recompiled model with 30 sec tstep parameter in wr_cgrid
- resubmitted to air, 12 cores
    - forgot to change tstep in run script so it crashed....
- resubmitted same run on free40i, 12 cores, after fixing timestep in run script
    - after submitting running, QRSH'd into free40i with 28 cores in a different terminal to occupy the rest of the free nodes
    - was at 8:50 before leaving
    - **THIS RUN FINISHED SUCCESSFULLY –> got WRFRST file needed**
        - **see below after SIDE TEST for next steps**
    - saved (renamed) output and did HPC backup to tarrega EPRI external
-
- **–---------------SIDE TEST:**
    - David suggested trying to run WRFv3.8 by itself to see if it would run with more cores
        - got archive from coupled model directory
        - unzipped → configured (sourced LOGIN INTEL) → check configure.wrf and make changes consistent with coupled model (reduced optomization, openmp flags) → compiled with same coupled model command → used twoway runscript GOOD to actually run the model → would not run with 32 cores, same hang on startup. Later ran with 12 but crashed at runtime. Did not investigate further.
- ---------------
-

<u>NOW TRYING TO RUN 1 week (6 following days) using wrfrst produced from 1st day...</u>

- significantly modified runscript to run from 0804 to 0806 (length of wrf bc file)
    - saved old run script as AUG03_1day_twoway_model_run_script
    - basically used built-in loop to give names of files that change with day
    - had to change WRF_RSTFLAG and also restart in wrf namelist
    - manually created links to wrfrst file and SCGRID file from GOOD output directory of 1st day simulation (these are then used as IC for wrf/cmaq)
    - ***would need to edit METPATH and probably at IF statement to get it to loop across different wrf input folders for longer simulation***
    - ***maybe try more cores now that it is smaller time step and different days (and not 1st day)***
- submitted to air, 12 cores, seems to be running but at less than 1:1 speed, mostly because CMAQ calculations seem to take a long time
    - crashed at 042730 in CMAQ - Could not determine Courant-condition safe sync step for model step: 010000 HHMMSS in layer: 12
        - (Max vel)/(dX) =   NaN
- tried running on 88i with 24, 20, 18, 16, cores but would not start up...
- submitted to 32i with 16 cores, when qrsh into 32i to take the remaining 16 cores and watch with htop

  - started up fine, running pretty quickly (note that 3:00 step always takes a long time)
  - crashed right away at 00:32:30 with segmentation fault...
- resubmitted same run with 12 cores on free 32i, then qrsh with 20 cores to take the rest
  - **This run was going GOOD – SUCCESSFULLY finished the first day, 0804, just fine, but there was a problem with the way I set up wrf namelist** → this caused it to use 0804 wrf input when running 0805 with cmaq (dates clearly mismatched in rsl.* files)
    - The start day on wrf namelist needs to be updated every day, so that the start day is actually the current day that is running (does not run sequentially)
      - modified runscript so that that wrf namelist uses $cur_day from the loop through the simulation period
      - also set end day = $cur_day and set run_days=1 so hopefully it will know to only run 1 day at a time and use the restart to start the next day
    - also had to modify the ln -s link for the wrfrst and SCGRID to start first day
    - and edit the actual start day to 05
- killed previous run at about hour 14 in 0805 → made changes above
- resubmitted with 0805 start date (2 days through 0806) to 32i with 12 cores, started running on a different node with 12 cores that someone already had a job on
  - crashed at 03:25:00 with *** ERROR ABORT in subroutine CAL_CFRAC_3D on PE 009 Floating point exception for CFRAC_3D
  - (even though it ran past this point with wrong wrf met date)
- resubmitted to free32i with 16 cores
  - other people running 12 core job/ 1 core qrsh, but was able to qrsh and get last 3 cores on compute-2-9
  - something weird is making this go extremely slow, over 1h in and not even done with 3mins
  - KILLED this
- tried to do 16i with 16 cores → failed with some “internal” MPI or infiniband error
- resubmitted to free32i with 12 cores
  - crashed at exactly the same spot 03:25:00 with floating point exception
    - this was actually the exact same configuration that did this before... so obviously 32i with 12 cores is not going to work
    - BUT THIS WAS THE ONE THAT WAS WORKING WITH WRONG wrf met THAT I KILLED?
- resubmitted to 88i with 16 cores (someone running 64 core job on the node as well)
  - seems to get stuck after the 2:30 step on wrf at 127 HGIIGAS 1.0000
- killed this and resubmitted with 14 cores on free88i
  - would not run on the node it went to.... stalled at startup
- resubmitted again with 14 cores on 88i after exiting qrsh to see if it would use different node
  - this started up fine, made it past the 03:25:00 crash that was happening on 32i with 12 cores
  - there was someone else running with 64 cores and I qrsh'd with a few cores to watch with top as well
  - crashed with segmentation fault at 04:20:00 because of DRY DEP error on node 0009
- resubmitted to free40i with 8 cores (someone else using the rest)
  - this crashed at 21:07:30 after almost 24h of running....
- Changed restart interval in wrf namelist to be every 6 hours
- resubmitted to 32i with 16 cores (20 would not start up)
  - seemed to freeze at "MET_CRO_3D" opened as OLD:READ-WRITE for more than 20

    minutes
    - killed job
- resubmited to 32i with 18 cores, qrsh with the other 14 to watch
    - crashed at 05:26:30 something in wrf.... no clear error
- resubmitted to 88i with 32 cores → STARTED UP OKAY when the rest of the node was empty
    - was even able to qrsh and get the rest of the 56 cores and watch with htop
    - drydep error at 012807
- resubmitted to 88i with 30 cores → STARTED UP OKAY when the rest of the node was empty
    - qrsh for the other 58 cores
    - crashed in wrf at 021730 for no reason...
- tried twice with 28 cores on 88i → would not start up (even though rest of node seemed empty)
- 24 would not work either... (but did later see below)
- resubmitted with 26 and it finally worked (on 88i)
    - qrsh'd for the other 62 cores
    - crashed during CMAQ startup after 00:02:30 in WRF
- resubmitted with 36 and that started up ok.... (88i)
    - qrsh'd for the other 52
    - dry dep error at 3:18:07 for O3, Vd NAN
- tried with 40 cores multiple times and would hang on startup
- resubmitted with 24 and that started up ok this time.... (88i)
    - qrsh'd for the other 64 cores
    - crashed with floating point exception ERROR ABORT in subroutine CAL_CFRAC_3D
        - at 1:13:45
- tried 28 again, would not work on 88i
- resubmitted with 20 cores on 88i
    - qrsh'd with the other 68 cores
    - Floating point exception for CFRAC_3D at 2:11:52
- submitted with 16 cores on free32i → qrsh with the other 16
    - (this is running about 2x as fast as it was with 16 cores on free88i when someone else was running 4x 1 core jobs on the same node, with the rest of the node empty)
    - crashed with floating point exception in CFRAC_3D at 5:06:52
- submitted to air64 with 22 cores (one of the few #'s I havent tried on intel) → and qrsh'd with the rest of the 42 cores to watch and kick free64 people off the node (or else it would not start up with others running)
    - was going super slow and crashed in wrf at 00:16:30
- submitted to free24i with 10 cores, qrsh'd for the other 14
    - this is running 2-3x faster than 22 cores on air64
    - **SUCCESSFULLY COMPLETED BOTH DAYS 05 and 06**
        - **renamed/saved output, did backup to external**
- saved this runscript
- edited runscript to run for Aug 07-09
- submitted to free24i, 10 cores
    - **Looks like it successfully completed Aug07 but crashed opening restart on next day?**
    - **For some reason only created rst at 06:00:00 on Aug 07, not at end of day 000000 on Aug08**

- Changed restart file interval back to 360 in wrf namelist...
- resubmitted to free24i, 10 cores, basically exact same run with different wrfrst interval...
    - **This run seems to have completed Aug 07 and 08 already successfully**
    - **SUCCESSFULLY completed Aug07-09 it seems**
- made a copy of runscript then edited it to re-run Aug04 (only 14 tsteps in that wrfout)
- submitted to 32i with 12 cores, qrsh for 12 more cores, someone else qrsh with 8 but not really running anything
    - crashed with CAL_CRAF_3d errror at 1:37:30
- submitted to 32i with 10 cores → qrsh for some but someone else also running
    - crashed at 18:19:30 for no clear reason...
- submitted to free24i with 12 cores → qrsh with other 12 taking entire node, running quick
    - **THIS RUN COMPLETED SUCCESSFULLY**
        - actually ended up running both 04 and 05 but only needed 04 I believe
- ***I now have successfully ran the "base case" (no NH3 uptake) for Aug03 through Aug09. Aug03 uses WRFinput and CMAQ default ICs, each other day continues as a restart from the previous.***
-
- *NEXT → NEED TO TRY AND COMPILE WITH UPTAKE AND RERUN*
-
-

<u>Modifying for NH3 Uptake, Recompiling, Trying coupled Model NH3 uptake Run:</u>

- ***SHOULD HAVE PROBABLY STARTED ALL THESE SIMULATIONS EARLIER AND DONE SPIN UP RUN – CMAQ ICs can take several days, and so does NH3 uptake processes***
- Talked to Shupeng about files required to implement NH3 uptake:
    - in aero6/
        - aero_subs.F
        - SOA_DEFN.F
- Because my version of CMAQ is different than this (differences in unodified versions of these files), I had to manually implement his code into my source code files, rather than just using his.
- Implemented his NH3 uptake mechanism by looking at diff between those files in his NH3 version compared with ORIG version → inserted code into my source code files in the cmaq folder
    - made a copy of ORIG_cmaq first
    - set uptake coef to 1.0E-3
- recompiled (did not clean) and can see that it rebuilt some of the aero files in cmaq/ and produced a new wrf.exe
- reverted back to AUG03 1st day twoway model runscript
- submitted run to free24i with 12 cores → qrsh with the other 12 cores
    - **THIS RUN FINISHED SUCCESSFULLY, just first day Aug03**
- modified runscript to run Aug 04-06 (the rest of this wrfbdy file)
- submitted this 3-day run to free24i with 12 cores → qrsh for the rest
    - **got SUCCESSFUL COMPELTE WRF for final day (Aug 06)**

- renamed output folder, copied runscript for Aug 07-09, linked to the new NH3 uptake output directory for restart in runscript
    - WOULD NOT RUN on free24i compute-1-14 for some reason
        - kept crashing with seg fault 4 – illegal instructions
        - googled and seemed to be some processor issue?
- Changed and submitted to free48i (a new queue?) with 12 cores, started up fine
    - crashed at 022000 ERROR ABORT in subroutine ADVSTEP Could not determine Courant-condition safe sync step for model step: 010000 HHMMSS in layer: 23 (Max vel)/(dX) = 1.309E+01
- resubmitted to free48i with 10 cores (qrsh for the rest)
    - crashed at 4:58:45 with max iterations exceeded in vppm at 77
- resubmitted to free48i with 14 cores (qrsh for the rest)
    - crashed in wrf at 07:36:30 for no clear reason
- resubmitted to free48i with 8 cores (qrsh for the rest)
    - crashed with vppm error at 05:25:00
- resubmitted to free48i with 16 cores, qrsh the rest
    - crashed with negative DRYDEP error at 01:50:00
- resubmitted free48i with 6 cores, qrsh the rest
    - crashed at 10:47:30 with vppm error
- resubmitted free48i with 11 cores (FIRST ODD #), qrsh the rest
    - **SUCCESSFULLY completed Aug 07 and 08 BUT**
        - **crashed 08-09_04:17:30** with NaN drydep
- modified runscript to only run Aug 09 and submitted to free48i with 10 cores
    - **SUCCESSFULLY completed Aug 09, final day of 1 week simulation with uptake**
- ***Post-processed some outputs using VERDI (cmaq) and ncl script (WRF Temperature)***
    - ***sent email to EPA Group July 12, 2018***
-
- Submitted a run of Aug04-06 to test profilemyjobs for HPC, see emails

Appendix A

IOAPI installation

```
>> export IOAPIDIR=${ROOTDIR}/IOAPI
>> cd ${IOAPIDIR}
>> wget --no-check-certificate https://www.cmascenter.org/ioapi/download/ioapi-3.2.tar.gz
>> tar xfvz ioapi-3.2.tar.gz
>> cp Makefile.template Makefile
```

Edit Makefile, in lines 133-135 this way:

```
#BIN        = Linux2_x86_64
BASEDIR    = ${ROOTDIR}/IOAPI
INSTALL    = $(BASEDIR)
```

```
>> cd ioapi/
>> cp Makefile.nocpl Makefile
```

Edit ioapi/Makefile in line 81 this way:

```
BASEDIR = ${ROOTDIR}/IOAPI
```

```
>> cd m3tools/
>> cp Makefile.nocpl Makefile
```

Edit ioapi/Makefile in line 38 this way:

```
BASEDIR = ${ROOTDIR}/IOAPI
```

Line 61:

```
LIBS  = -L${OBJDIR} -lioapi -L${NCDIR}/lib -lnetcdff -lnetcdf $(OMPLIBS) $(ARCHLIB) $
(ARCHLIBS)
```

```
>> cd ..
>> make all
```

Appendix B – Links and references

WRF-ARW and WPS:
http://www2.mmm.ucar.edu/wrf/users/download/get_sources.html
https://github.com/NCAR/WRFV3/releases
http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3/users_guide_chap5.htm#Namelist
Installation:
https://www.enviroware.com/installing-and-running-wrf-3-8-on-linux-ubuntu-lts-16-04-with-intel-i7-8-core-cpu/
https://www.youtube.com/watch?v=hkLrdlQnKTw
http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php#STEP8
http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Compile/arw_compile2.htm
http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Compile/arw_compile3.htm
Namelist:
https://esrl.noaa.gov/gsd/wrfportal/namelist_input_options.html
Data Download:
http://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog.html

Problems:

- http://www2.mmm.ucar.edu/wrf/users/FAQ_files/FAQ_wrf_runtime.html
- http://www2.mmm.ucar.edu/wrf/users/wrfv3.8/known-prob-3.8.html
- http://www2.mmm.ucar.edu/wrf/users/wrfv3.8/known-prob-3.8.1.html
- Physics options:
  - http://forum.wrfforum.com/viewtopic.php?f=28&t=328
  - http://www2.mmm.ucar.edu/wrf/users/phys_references.html
  -
- About CFL:
  - http://mailman.ucar.edu/pipermail/wrf-users/2012/002934.html
  - http://www.nusculus.com/wrf-errors
- Would go off top and CFL:
  - http://mailman.ucar.edu/pipermail/wrf-users/2011.txt
- Vertical velocity and damping options:
  - http://mailman.ucar.edu/pipermail/wrf-users/2016/004162.html
- CFL and epssm:
  - http://forum.wrfforum.com/viewtopic.php?f=6&t=8939
- Compilation:
  - http://forum.wrfforum.com/viewtopic.php?f=5&t=3660
  - With parallel NETCDF
    - http://mailman.ucar.edu/pipermail/wrf-users/2015/004007.html
- RIBX never exceeds RIC
  - http://forum.wrfforum.com/viewtopic.php?f=9&t=5894
  - http://mailman.ucar.edu/pipermail/wrf-users/2016/004316.html
  -

- timestep
  - http://forum.wrfforum.com/viewtopic.php?f=8&t=357
  - http://forum.wrfforum.com/viewtopic.php?f=6&t=5920
- Circular Buffer files (S_CGRID problems):
  - https://www.cmascenter.org/ioapi/documentation/all_versions/html/DATATYPES.html

CMAQ:
https://www.airqualitymodeling.org/index.php/CMAQ_version_5.2_(June_2017_release)_Technical_Documentation
https://github.com/USEPA/CMAQ/tree/5.2
https://github.com/USEPA/CMAQ/blob/5.2/DOCS/README.md
https://github.com/USEPA/CMAQ/blob/5.2/DOCS/User_Manual/README.md
https://github.com/USEPA/CMAQ/blob/5.2/DOCS/Tutorials/README.md
https://github.com/USEPA/CMAQ/blob/5.2/DOCS/User_Manual/CMAQ_OGD_quick_start.md
https://github.com/USEPA/CMAQ/blob/5.2.1/DOCS/Tutorials/CMAQ_Benchmark.md
https://www.epa.gov/cmaq/cmaq-documentation#tutorials

Compilation flags:
https://stackoverflow.com/questions/9685377/undefined-reference-to-omp-get-max-threads

Coupled Model:
https://www.epa.gov/cmaq/wrf-cmaq-two-way-coupled-model
ftp://newftp.epa.gov/exposure/CMAQ/V5_2/WRF-CMAQ_Coupled_Model/instructions.txt
https://github.com/USEPA/CMAQ/blob/5.2/CCTM/docs/Release_Notes/Two_Way_Coupled_WRF-CMAQ.md
https://github.com/USEPA/CMAQ/issues/78
https://www.epa.gov/cmaq/wrf-cmaq-two-way-coupled-model
https://www.epa.gov/cmaq/access-cmaq-source-code
https://www.epa.gov/cmaq/cmaq-models-0#self
https://drive.google.com/drive/folders/0B2kjxCwKICxUZWp2ZVFGZkx6YzA

Old Version:
https://www.airqualitymodeling.org/index.php/CMAQv5.1_Two-way_model_release_notes

Issues:

IOAPI:
https://www.cmascenter.org/download/software/ioapi/ioapi_3-2.cfm?DB=TRUE
https://github.com/cjcoats/ioapi-3.2/blob/master/README.txt

NETCDF (-lnetcdff goes before -lnetcdf on -L link)
-L$(NETCDF)/lib -I$(NETCDF)/include -lnetcdff -lnetcdf:
http://forum.wrfforum.com/viewtopic.php?f=5&t=9267
http://forum.wrfforum.com/viewtopic.php?f=30&t=8842

HDF5 (and parallel NETCDF with it)
https://www.unidata.ucar.edu/software/netcdf/docs/getting_and_building_netcdf.html#build_parallel
https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.10/hdf5-1.10.2/src/unpacked/release_docs/INSTALL_parallel
https://support.hdfgroup.org/HDF5/faq/parallel.html

JASPER:
http://www.ece.uvic.ca/~frodo/jasper/#download