WarpX/HPC/Linux Notes and Flows

# Vocabulary – WarpX

* Macroparticle (or super-particle): a computational particle that represents many real particles (Wikipedia – particle-in-cell: <https://en.wikipedia.org/wiki/Particle-in-cell>)

# Notes

* Stability conditions for PIC include grid size and time step:
  + , where is the Debye length
  + , where is the plasma frequency.

# Linux Commands

* Find – find file/directory
  + -name (optional flag to search by name)
  + E.g. >find -name run\_warpx.sh<
* List files in directory:
  + ls 🡺 “list short” 🡺 file/directory names only
    - To count # of directories in a directory: >ls -l . | grep -c ^d<
      * No idea how that works atm
  + ll 🡺 “list long” 🡺 file/directory names, details, history, what they ate for breakfast, etc.

# Modules

* nano – text editor
  + Used to edit text files, commonly GNUmakefile, input “decks”, etc.
  + Needs to be loaded on every WebShell login for Onyx
    - Does not need to be loaded on Narwhal – maybe comes preloaded
    - “module load nano”

# Vocabulary - Linux

* Dot >.< means “here”
  + E.g. >ls .< means “list here”, which lists all files in current directory
* Less, i.e. >less GNUmakefile< will show a document in a view-only reader, press >q< to quit, arrows to navigate, spacebar to navigate but more
* Tilde >~< means “default (?) directory”
  + E.g. >cd ~< means “change directory default”, which changes current directory to the starting point of the user.
  + For Narwhal, ~ dir is /p/home/[user].
    - This specific command can also be accomplished by >cd< by itself. I assume ~ is the default for the cd command.
* Directories to know:
  + Temporary directory: >cd /tmp<
  + Work directory: >cd /p/work1/[username]<
  + CENTER directory: >cd /p/cwfs/[user]<

# Flow: How to get WarpX up and running on a new HPC system

1. This method may be outdated, so check WarpX documentation: <https://warpx.readthedocs.io/en/latest/install/users.html>. This method will cover a legacy installation method with GNUmake.
2. Create a directory with a strong name and clone the WarpX repository of choice. Make directories for each import:
   1. >mkdir [warpx]<
   2. >cd [warpx]<
   3. >git clone <https://github.com/ECP-WarpX/WarpX.git> ./WarpX<
   4. >git clone <https://github.com/ECP-WarpX/picsar.git> ./picsar<
   5. >git clone <https://github.com/ECP-WarpX/warpx-data.git> ./warpx-data<
   6. >git clone <https://github.com/AMReX-Codes/amrex.git> ./amrex<
3. Modify AMReX make.unknown file:
   1. >cd [warpx]/amrex/Tools/GNUMake/sites<
   2. >nano make.unknown<
   3. change CC/cc/ftn/ftn90 to their corresponding values in the system’s user guide – look for Available Compilers and find your system’s compiler commands
4. Modify GNUmakefile:
   1. >cd [warpx]/WarpX<
   2. >nano GNUmakefile<
   3. At the top of the document, type NO\_MPI\_CHECKING = TRUE
   4. Make sure DIM is set to what you need for your sim (1,2,3)
   5. Set COMP to your machine’s environment (cray, intel, gnu, etc.)
   6. Set everything that has a Boolean input to FALSE except WARN\_ALL and TINY\_PROFILE.
   7. Output file appears in “Bin” directory, titled something like “main2d.cray.broadwell.TPROF.MTMPI.ex”
      1. Breakdown:
         1. main2d: main file(?) 2D simulation
         2. cray: uses cray compiler
            1. Onyx and Narwhal are cray machines
         3. Broadwell – has to do with architecture – Narwhal says x86-intel
         4. MTMPI: idk
         5. TPROF: Tiny PROFile
         6. ex: file is an executable
5. Compile WarpX:
   1. In [warpx]/WarpX:, >make clean<
   2. Run a short test compile to make sure nothing is terribly wrong: >make<
   3. If everything seems to be running, cancel and run with more cores:
      1. >ctrl+C<
      2. >make -j 20<
   4. Look for a SUCCESS message; if errors come up, troubleshooting is needed.

# Flow – How to Run a Sim

1. Copy an input file from an Examples folder to the directory that runs will be executed in
2. Copy compiled executable to directory with input file
3. Modify PBS (Portable Batch Script)/run\_warpx.sh file to spec
   1. >nano run\_warpx.sh<
      1. Note: .sh indicates a “shell” that supercomputer works on (?)
   2. For PBS Parameters, see your system’s [user guide](https://www.navydsrc.hpc.mil/docs/narwhalUserGuide.html#pbsCom).
      1. Required PBS Parameters:
      2. -l select=1:ncpus=128:mpiprocs=128 & -l walltime=00:30:00
         1. Select cores, # cpus, something else, computation time (guess)
      3. -q debug
         1. Queue on which to put the selected job.
         2. “debug” is the debug queue for jobs < 1hr,
         3. “standard\_sm” is the standard queue for small jobs
         4. More options available on user guide (search “Queue informat”):
      4. -A [a bunch of capital letters and numbers]
         1. User’s Subproject # on the currect system. Make sure this input matches your subproject #.
         2. Run >show\_usage< to show Subproject # and other maybe useful info.
      5. Optional PBS Parameters:
      6. -N WarpXrun
         1. Name for job on supercomputer queue
      7. -V
         1. Export all variables; -v will export specific vars.
      8. -j oe
         1. “Merge stderr and stdout into stdout”. 🡺 merge output files into one file (?)
      9. -m be and -M [email address]
         1. Send email when the job **b**egins and **e**nds.
   3. Execution block: Run the code you need to run:
      1. cd $PBS\_O\_WORKDIR
         1. This will change directory to the dir that you ran the .sh file from.
      2. aprun -n 128 [GNUmakefile .ex file name] [input file name] > warpx.out
         1. Make sure -n # matches the -l PBS args. There are ways to split jobs between different cores, I know none of them as of yet.
4. Submit the job
   1. >qsub run\_warpx.sh<
   2. After, you can use >qstat [job #]< or >qstat -u [username]<to check in on its progress
   3. If the job breaks, can use >qdel [job #]< to stop the job from continuing.
   4. After the job runs, you should see either Backtrace files and a bunch of other junk or a diag directory. Backtrace means the job failed somewhere and you need to troubleshoot. A diag directory means that the job was successful. You may need to wait a couple minutes for the job to fully finish to all diag files can be written to the diag directory.

# Flow: How to visualize output diag files with VisIt

Start: a “diag” directory from a WarpX run is present somewhere. You want to visualize the results of this run in a single figure using this directory.

1. Give your diag directory a unique name so it’s easy to find and for documentation purposes.
2. Move your diag directory to the $CENTER directory. There are two methods
   1. Command line: >mv [diag directory] /p/cwfs/[user]<
   2. Through the Enhanced File Manager: Find the file manager in HPC home. Navigate to the directory that your diag dir is located and copy/paste or drag/drop it to CENTER. Either the portal’s CENTER or the supercomputer’s CENTER will work; they are one and the same.
3. Wait a while. Contemplate life, or something. Then, Go back to HPC home page. Search “visi” and click on the VisIt application.
4. Once the app opens, go to file>open… and type /p/cwfs/[user] in Path.
5. Click a Header file of a diag dir you are interested in visualizing and click OK.
6. On the VisIt toolbar, click “add” and click the quantity you’re interested in.
   1. Recommend pseudocolor.
7. Click “draw”. A picture should appear.
8. To save, File>save settings>fill out settings (name, directory)>save.

Start: a “diag” directory from a WarpX run is present somewhere. You want to visualize the results of this run in a movie using this directory.

1. Give your diag directory a unique name so it’s easy to find and for documentation purposes.
2. Create the file movie.visit: navigate to your diag directory with all the diag1\* files in them. Run the command **>ls -1 diag1\*/Header | tee movie.visit<** to collate all diag header paths into a single file.
3. Copy your diag directory to the $CENTER directory. There are two methods:
   1. Command line: **>cp [diag directory] /p/cwfs/[user]/[new name if desired]<**
   2. Through the Enhanced File Manager: Find the file manager in HPC home. Navigate to the directory that your diag dir is located and copy/paste or drag/drop it to CENTER. Either the portal’s CENTER or the supercomputer’s CENTER will work; they are one and the same.
4. Open the VisIt app from the HPC home page.
5. File>open movie.visit.
6. Add>[recommend pseudocolor]>[some quantity you’re interested in].
7. Draw>move the slider/press play
8. To save, File>save movie…>adjust settings>save.

# Flow: New conda WarpX installation

1. Module load python/3
2. Conda create -n warpx -c conda-forge warpx
3. Conda activate warpx
   1. If that doesn't work, start a new shell: conda init bash
   2. Exec bash
   3. Conda activate warpx
4. I don’t know how to change the executable or copy it to a run directory yet.