CMMAP Data Services Update

Karen Schuchardt

Ft Collins

August 2011

Pacific Northwest
NATIONAL LABORATORY

Outline

- IO with ZGrd
- Visualization (Visit) Update
- Pagoda Update
- Geodesic Data Model Update
- IO Agent

Pacific Northwest
NATIONAL LABORATORY

IO Reminder Slide

- PNetCDF
  - Stable 1.2 release for quite some time
  - Only PNetCDF can process large files generated by PNetCDF
  - Relies strictly on collective IO
- NetCDF4
  - Current Version 4.1.3 not as stable or stress tested
  - Designed to sit on top of HDF5(more complicated to build)
  - still has large variable problem (fortran)
  - Can use collective or independent IO

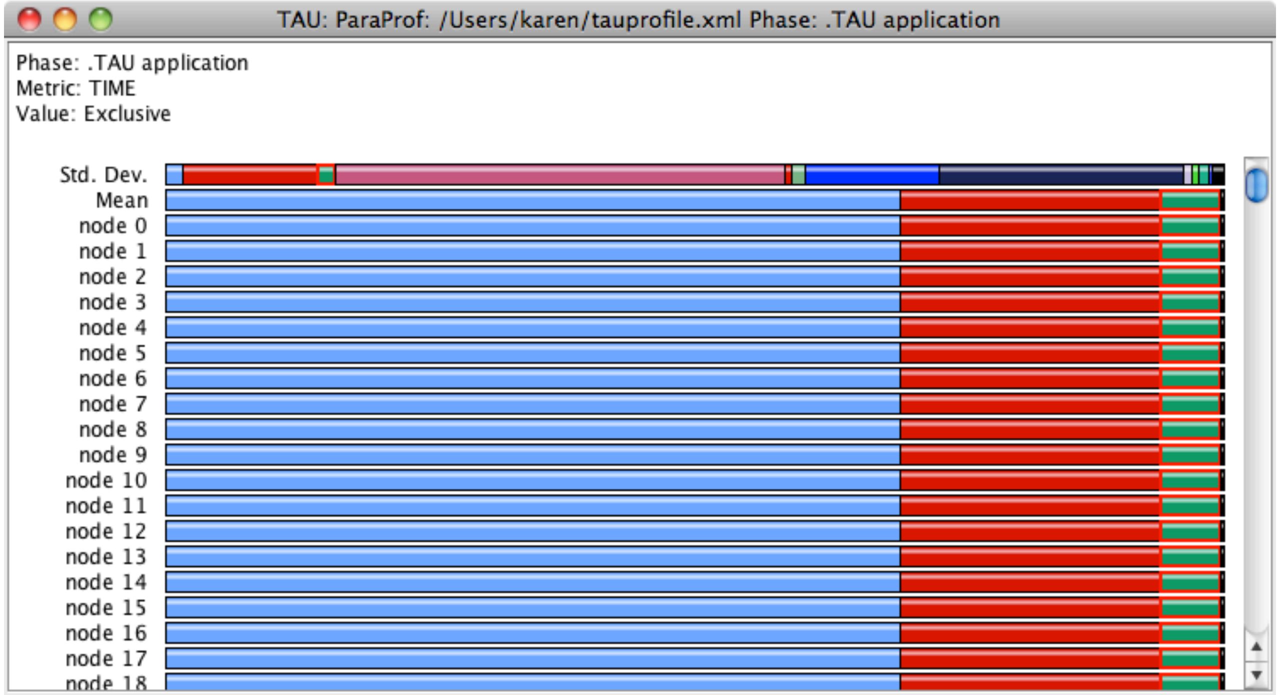
- BUT NetCDF4 build can specify to use PNetCDF underneath

IO in ZGrd

- Supports PNetCDF, NetCDF4
- Restart is working with physics
- Supports instantaneous and averaged data
- Supports collective IO, Application Aggregation, and Psuedo non-blocking
- Supports per variable time series files or all variables in one file
- Testing on hopper, franklin; Intrepid (BG/P planned)
  - Up to 20,000 processors, primarily 10,000
  - Primarily very short runs for profiling
- History Output
  - 4km, 100 vertical (4 Billion cells, 8 B corners, 12 B edges)
  - 3 surface vars, ~ 12 center vars, ~ 3 corner vars (and growing)
  - ~400 GB per timestep

ZGrd IO Performance Profile - PNetCDF

- IO Configuration
  - Hopper 80 OSTs, collective IO (multiple options)
  - 100 model timesteps/write (every 2 minutes with 12 sec timestep)
  - 8% of time in IO including averaging



TAU: ParaProf: /Users/karen/tauprofile.xml Phase: .TAU application

Phase: .TAU application
Metric: TIME
Value: Exclusive

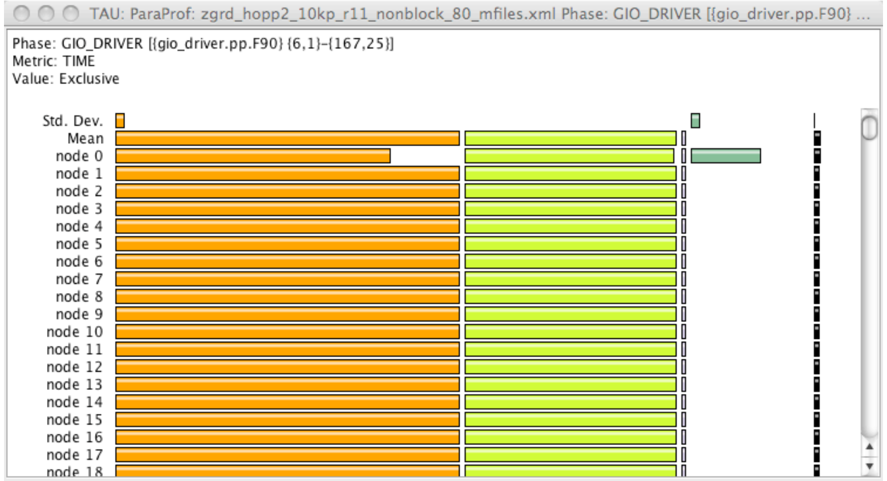
🟦 ZGrid

🟥 Init (bisect)

🟩 IO

IO Library and File System Matter

- Cray MPI-IO based on open source ROMIO (ANL)
- Very poor scaling of MPI_File_Open on hopper
  - Cray is looking at this with possible fix in Sept.

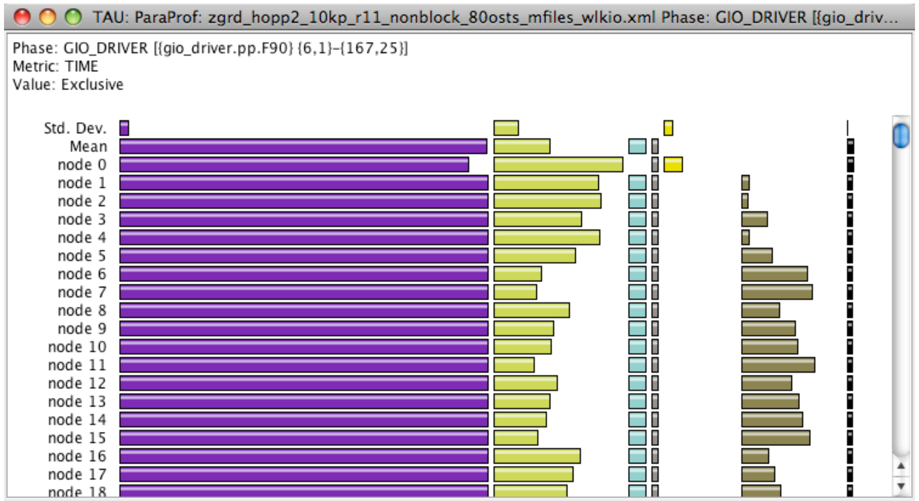


TAU: ParaProf: zgrd_hopp2_10kp_r11_nonblock_80_mfiles.xml Phase: GIO_DRIVER [{gio_driver.pp.F90} ...

Phase: GIO_DRIVER [{gio_driver.pp.F90} {6,1}-{167,25}]
Metric: TIME
Value: Exclusive

Cray IO

Write_all,open,avg,write



TAU: ParaProf: zgrd_hopp2_10kp_r11_nonblock_80osts_mfiles_wlkio.xml Phase: GIO_DRIVER [{gio_driv...

Phase: GIO_DRIVER [{gio_driver.pp.F90} {6,1}-{167,25}]
Metric: TIME
Value: Exclusive

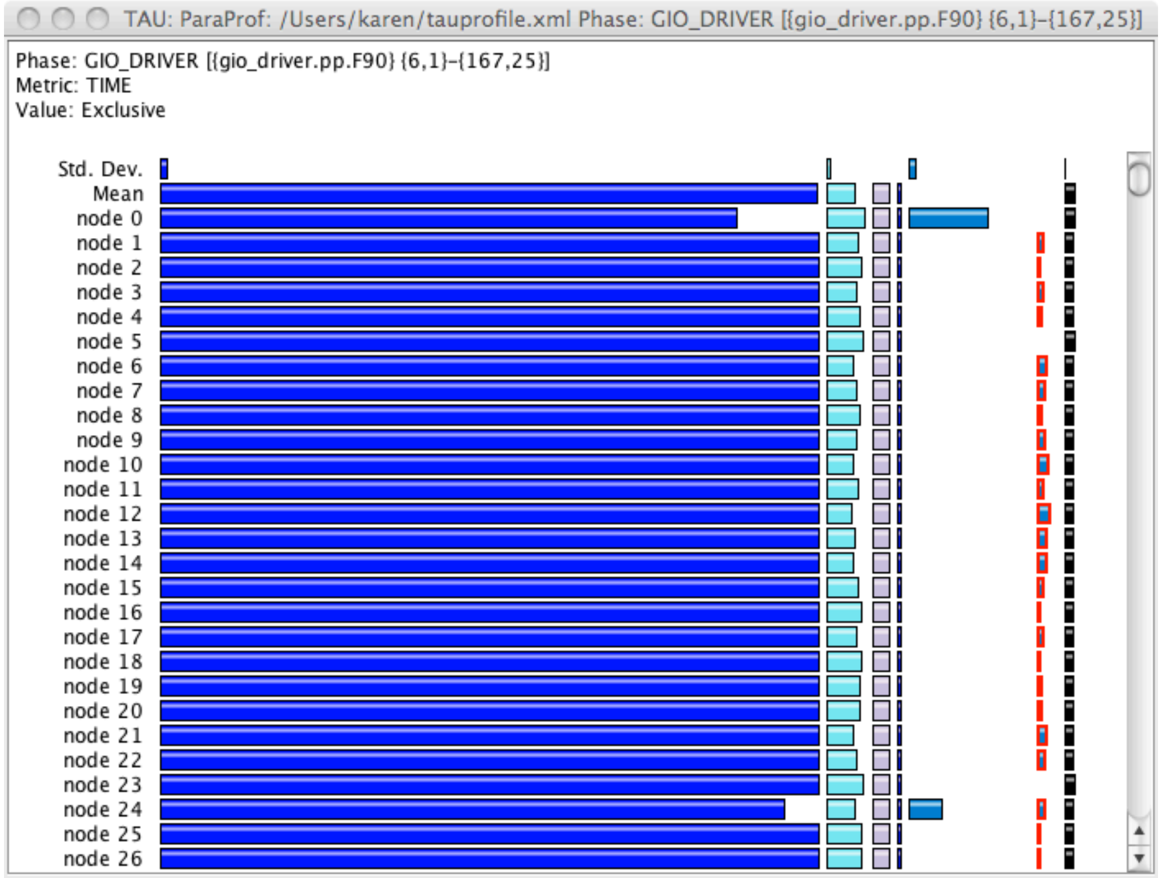
ROMIO

Write_all, mipopen bcast,, avg, write, open

Pacific Northwest
NATIONAL LABORATORY

Time Series Files vs One File

- All variables to one file (reduced open overhead)
- Same parameters as previous slide (2 minute data)
- Best performance (for now); very large files



TAU: ParaProf: /Users/karen/tauprofile.xml Phase: GIO_DRIVER [{gio_driver.pp.F90} {6,1}-{167,25}]

Phase: GIO_DRIVER [{gio_driver.pp.F90} {6,1}-{167,25}]
Metric: TIME
Value: Exclusive

Write all, mpiopen, avg, write open

Pacific Northwest
NATIONAL LABORATORY

Some Important Settings

- LUSTRE - Reduce overhead of file open calls with collective IO
  - Reduce cost of open by only opening on aggregator nodes
    - export MPICH_MPIIO_HINTS=l/data48/*.nc:romio_no_indep_rw=true
  - Avoid costly call to statfs in open call
    - Specify filenames to open like “lustre:filename”

- Other
  - The vendor MPI-IO library may not be the best choice
  - Pay attention to lustre striping settings
    - Lfs getstripe <dir/file>
    - Lfs setstripe <dir/file>

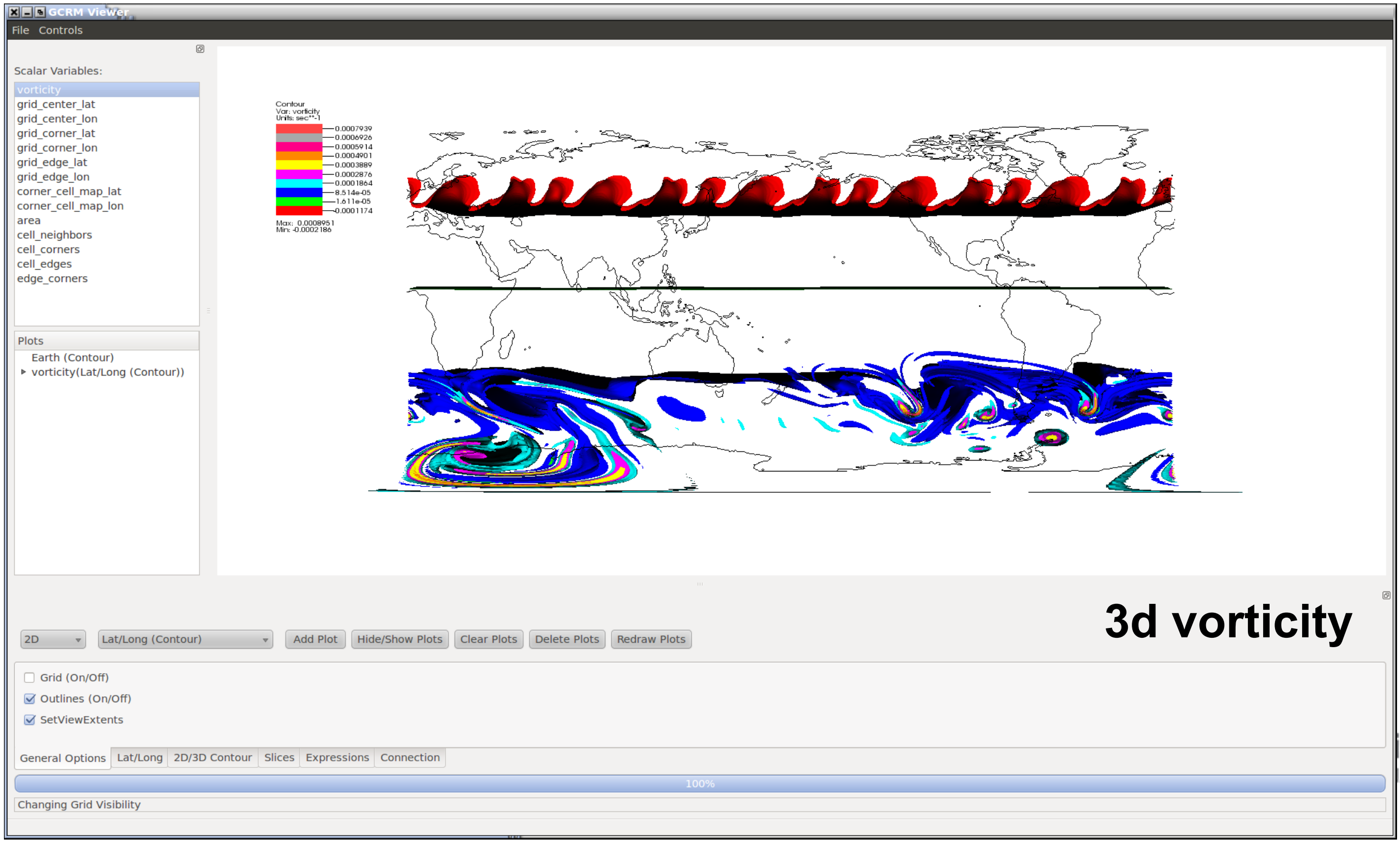
NetCDF4

- 4.1.3 recently released
- Proper chunking is critical to write performance
- Currently upgrading to this release and working with HDF5 group to profile and improve performance
- Current performance < 1GB/s
- I have C routines that bypass the 32 bit indexing problem if anybody wants them

- CESM focusing on use of PNetCDF though their IO layer (PIO) supports both

Visit – Climate Skin Prototype

- Climate Skin – a climate-centric UI
  - 2d contour plots, 3d surfaces plots
  - continental outlines for 2d and 3d
  - Zonal plots
  - 3d slices
  - Remote data access
  - sub regions at full resolution
  - Variable expressions
  - wind vectors
  - Images,movies
  - Suggestions, Priorities??



□ Grid (On/Off)

☑ Outlines (On/Off)

☑ SetViewExtents

Changing Grid Visibility

3d vorticity

Visit – Core Tasks

- NCML definition and script for processing collections of files
- Pending
  - Parallel support for edge variables
  - Performance profiling
  - Support for reading ncml dataset summary

Pagoda

NCO	Pagoda
ncks	pgsub
ncra	pgra
ncea	pgea
ncwa	(soon, v0.7)
ncbo	pgbo
ncflint	pgflint
ncrcat	N/A*
ncecat	N/A*
ncrename	
ncatted	NA**
ncpdq	
ncap/ncap2	

*Don’t concatenate, aggregate

** Not a parallel operation

- Fully data parallel
- Current version is 0.6
- Output verified against NCO
  - Tested GCRM data
    - 8km resolution
    - 2km (in progress)
  - Tested against ANL data
    - 1/8 degree CAM HOMME
    - 19 8.5GB files (15 variables each)
    - 19 2.5GB files (4 variables each)
  - Assumes NCO infallible
- Scriptable (but not as simple)
- Plan to incorporate ESMF parallel gridder
- Working on schemes to improve parallel reads

Semi-Structured Grid Standards

- Tiger Team to develop proposed standard for semi-structured (and maybe unstructured) grids
  - NOAA, USGS, ASA, Deltares, PNNL
  - Hope to solidify 2 ½ D and push this out to larger community this fall
  - ZGrd is updated to the latest ideas from this group but some things may change
  - http://public.deltares.nl/display/NETCDF/Deltares+CF+proposal+for+Unstructured+Grid+data+model

IO Agent

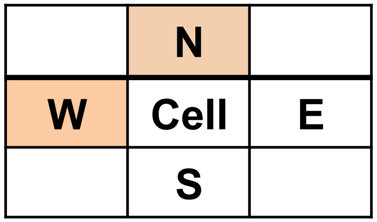
- Problem:
  - We want to save history data for later analysis on very short time scales – 5 minutes or less.
  - ZGrd, r11, 100 interfaces currently generates ~400 GB per snapshot. ZGrd will add more data overtime.
  - Data requires lots of storage:
    - A one hour simulation 3.5 TB
    - One day simulation 86 TB
  - IO overhead takes time away from simulation
  - Simulations are too expensive to just rerun to collect targeted data

IO Agent Approach

- Identify interesting data
  - targeting tropical cyclones
  - Initially using very simple thresholding cutoff of surface diagnostic data to determine "interesting" on a per cell basis
  - Mon. Weather Rev., 128, 377-384 (2000), Wea. Forcasting 17, 1152-1162 (2002), Geo. Res. Let. 38:L04809 (2011).
- Perform cluster detection
  - within processors
  - across processors
- Modify IO to write only the clusters
- Assess performance of algorithm and IO

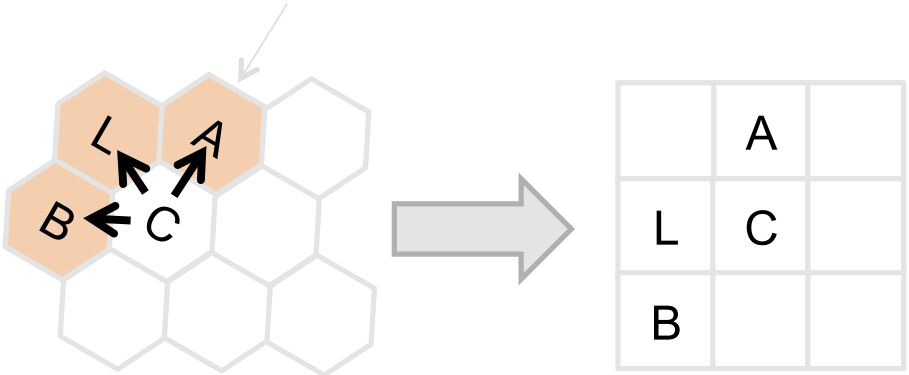
Cluster Detection Algorithm

- Hoshen-Kopelman Algorithm
  - Parallelization of the Hoshen-Kopelman Algorithm using a Finite State Machine. *Berry, Contantin, Vander Zanden 1995 **
    - Two cells belong to the same cluster if they have been identified as interesting and they are neighbors
    - Single pass using NEWS (North East West South) neighborhood rule
    - Linear scaling of large data sets (serially)



	N	
W	Cell	E
	S	

Pixels Clustering



	A	
L	C	
B		

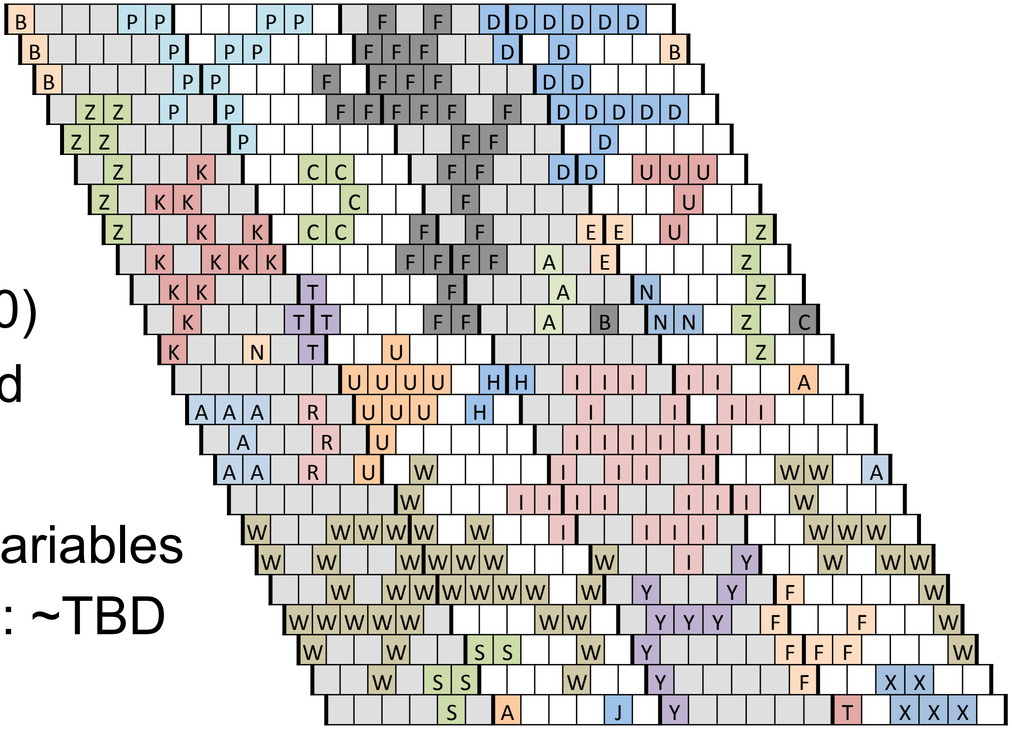
Geodesic Grid Clustering

*Original Article: Phys. Rev. B. 1(14):3438-3445 (1976), implementation from: Int. J. Supercomp. App. High Perf. Comp. 11(1):34-48 (1997)

IO Agent

- Challenges
  - Efficient parallelization in particular the cross-processor clustering
  - Dynamic generation of data sets
    - Array compaction
    - Re-indexing
  - Efficient IO
    - Rewrite grid each time since it will change
    - New file each time
  - Tools to process the data
    - Will be unstructured (by 2 ½ D) grid

Status and Next Steps



- Status:
  - Implemented and tested in independent MPI code (both C and F90)
  - Partially inserted into ZGrd
- Costs:
  - Memory overhead: ~1.5 variables
  - Performance cost/benefit : ~TBD
- Next Steps:
  - Complete insertion into ZGrd
  - I/O portion of project
  - Performance evaluation
  - Add input file parameters to control variables and threshold

Pacific Northwest
NATIONAL LABORATORY