conda module overview

Last updated: 11/2/21

CISL-provided conda (and mamba) install

Located: /glade/u/apps/opt/conda
Provided by: module load conda/latest

Will override which conda binary (and therefore system .condarc) takes precedent in the user environment *while module is loaded only* via setting CONDA_EXE.

tcsh init script can be provided by alias from module to enable activation without initialization; custom bash script is written to enable similar functionality there.
Thus, with module users can activate envs; no need for shell init.

Philosophy of least impact and customizations

Only impose changes on the user environment that are additive, not overrides

- module adds init-bypass only if user has not already initialized conda
- auto_activate_base is set to false so we don’t impose unknown binaries
- Settings made in $CONDA_ROOT/.condarc so they can be overridden by user’s ~/.condarc and environment variables
  - Set default environment location to /glade/work/$USER/conda-envs to preserve $HOME
  - Set default pkg cache to /glade/scratch/$USER/.conda/pkgs to preserve $HOME
  - Use conda-forge and “ncar” (for GeoCAT) as channels

“NPL” environments and conda

Planning to maintain NPL-like environment for convenience

- Default library available on command-line and JupyterHub
- Refresh once every month, with three versions available
  - Default version linked to previous month’s version

Seeking community input on these questions:

- Possibly offer machine learning version?
  - Cutting edge versions generally installed via pip, not conda (makes environments fragile)
- Preference for MPI libraries to be machine specific with an NCAR channel or use common MPICH offerings via conda-forge?
  - Currently do the latter to allow for single environment serving both systems
- How best to support NCAR projects that use our Python libraries?