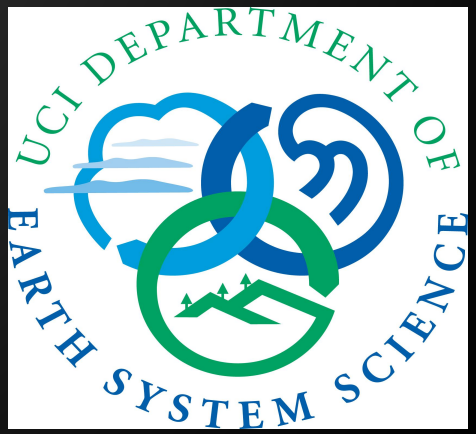
How to Generate Climatologies

Charlie Zender <zender@uci.edu>

Departments of Earth System Science and
Computer Science, UC Irvine

ACME PI Meeting
Albuquerque, NM
November 2-4, 2015

Seminar on Web



Add N numbers, then divide by N

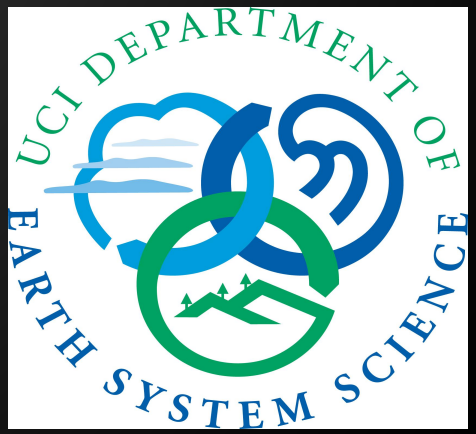
Optimizing Intrinsic Parallelism to generate climatologies with netCDF Operators (NCO)

Charlie Zender <zender@uci.edu>

Departments of Earth System Science and Computer Science, UC Irvine

ACME PI Meeting
Albuquerque, NM
November 2-4, 2015

Seminar on Web



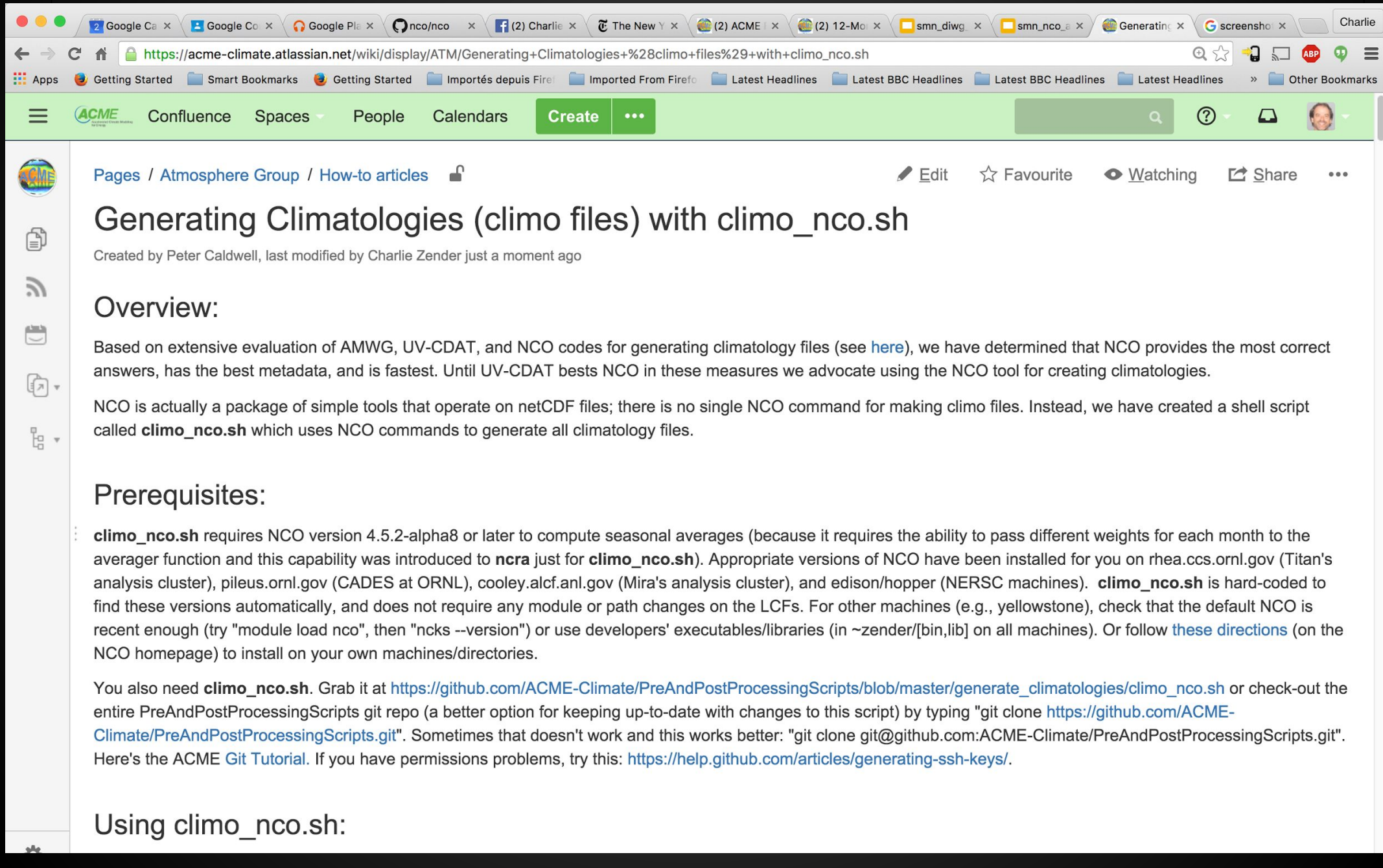
Climatologies

1. Prerequisites
2. Monthly, Seasonal, Annual "Climos"
3. Regridding
4. Parallelism Strategies
5. Examples for Cooley, Edison, Rhea

Find/Obtain/Install "climo_nco.sh"

```
> git clone git@github.com:ACME-Climate/ \
PreAndPostProcessingScripts
> ln -s ~/PreAndPostProcessingScripts/generate_\
climatologies/climo_nco.sh ~/bin/climo_nco.sh
```

Confluence "climo_nco.sh" docs



https://acme-climate.atlassian.net/wiki/display/ATM/Generating+Climatologies+%28climo+files%29+with+climo_nco.sh

Confluence Spaces People Calendars Create

Pages / Atmosphere Group / How-to articles

Edit Favourite Watching Share

Generating Climatologies (climo files) with climo_nco.sh

Created by Peter Caldwell, last modified by Charlie Zender just a moment ago

Overview:

Based on extensive evaluation of AMWG, UV-CDAT, and NCO codes for generating climatology files (see here), we have determined that NCO provides the most correct answers, has the best metadata, and is fastest. Until UV-CDAT bests NCO in these measures we advocate using the NCO tool for creating climatologies.

NCO is actually a package of simple tools that operate on netCDF files; there is no single NCO command for making climo files. Instead, we have created a shell script called **climo_nco.sh** which uses NCO commands to generate all climatology files.

Prerequisites:

climo_nco.sh requires NCO version 4.5.2-alpha8 or later to compute seasonal averages (because it requires the ability to pass different weights for each month to the averager function and this capability was introduced to **ncra** just for **climo_nco.sh**). Appropriate versions of NCO have been installed for you on rhea.ccs.ornl.gov (Titan's analysis cluster), pileus.ornl.gov (CADES at ORNL), cooley.alcf.anl.gov (Mira's analysis cluster), and edison/hopper (NERSC machines). **climo_nco.sh** is hard-coded to find these versions automatically, and does not require any module or path changes on the LCFs. For other machines (e.g., yellowstone), check that the default NCO is recent enough (try "module load nco", then "ncks --version") or use developers' executables/libraries (in ~zender/[bin,lib] on all machines). Or follow these directions (on the NCO homepage) to install on your own machines/directories.

You also need **climo_nco.sh**. Grab it at https://github.com/ACME-Climate/PreAndPostProcessingScripts/blob/master/generate_climatologies/climo_nco.sh or check-out the entire PreAndPostProcessingScripts git repo (a better option for keeping up-to-date with changes to this script) by typing "git clone https://github.com/ACME-Climate/PreAndPostProcessingScripts.git". Sometimes that doesn't work and this works better: "git clone git@github.com:ACME-Climate/PreAndPostProcessingScripts.git". Here's the ACME Git Tutorial. If you have permissions problems, try this: https://help.github.com/articles/generating-ssh-keys/.

Using climo_nco.sh:

Find/Obtain/Install NCO

Cooley, Edison, Pileus, Rhea, Yellowstone:

```
> export PATH='~zender/bin':${PATH}
> export LD_LIBRARY_PATH='~zender/lib': \
${LD_LIBRARY_PATH}
```

Local clusters, workstations, laptops:

```
> sudo aptitude install nco                  # Debian
> sudo dnf install nco                       # Fedora
> sudo port install nco                      # Mac Ports
> git clone git@github.com:nco/nco.git # DIY
> ./configure; sudo make install
```

GitHub NCO homepage

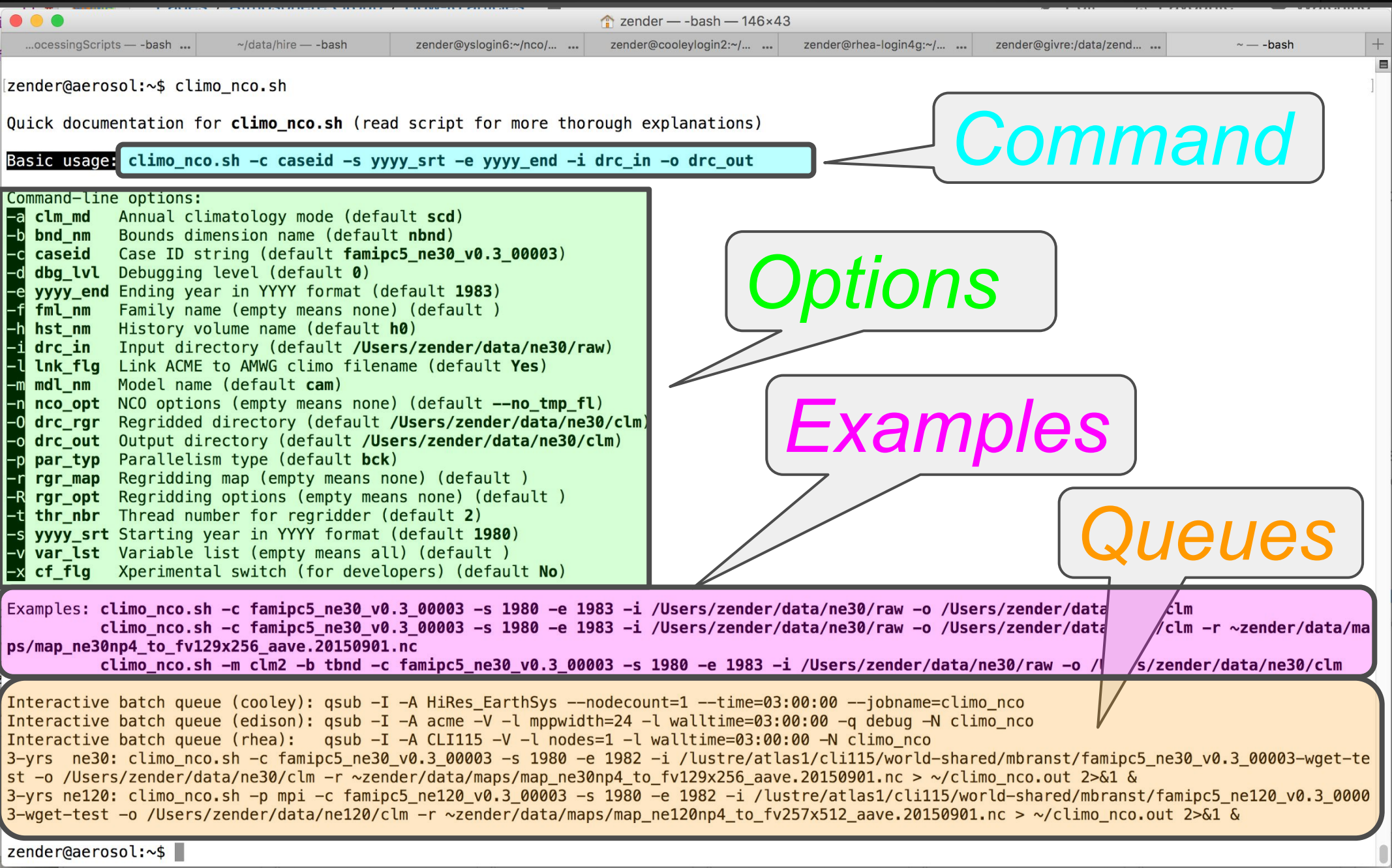
Climatologies

1. Prerequisites
2. Monthly, Seasonal, Annual "Climos"
3. Regridding
4. Parallelism Strategies
5. Examples for Cooley, Edison, Rhea

Run "climo_nco.sh"

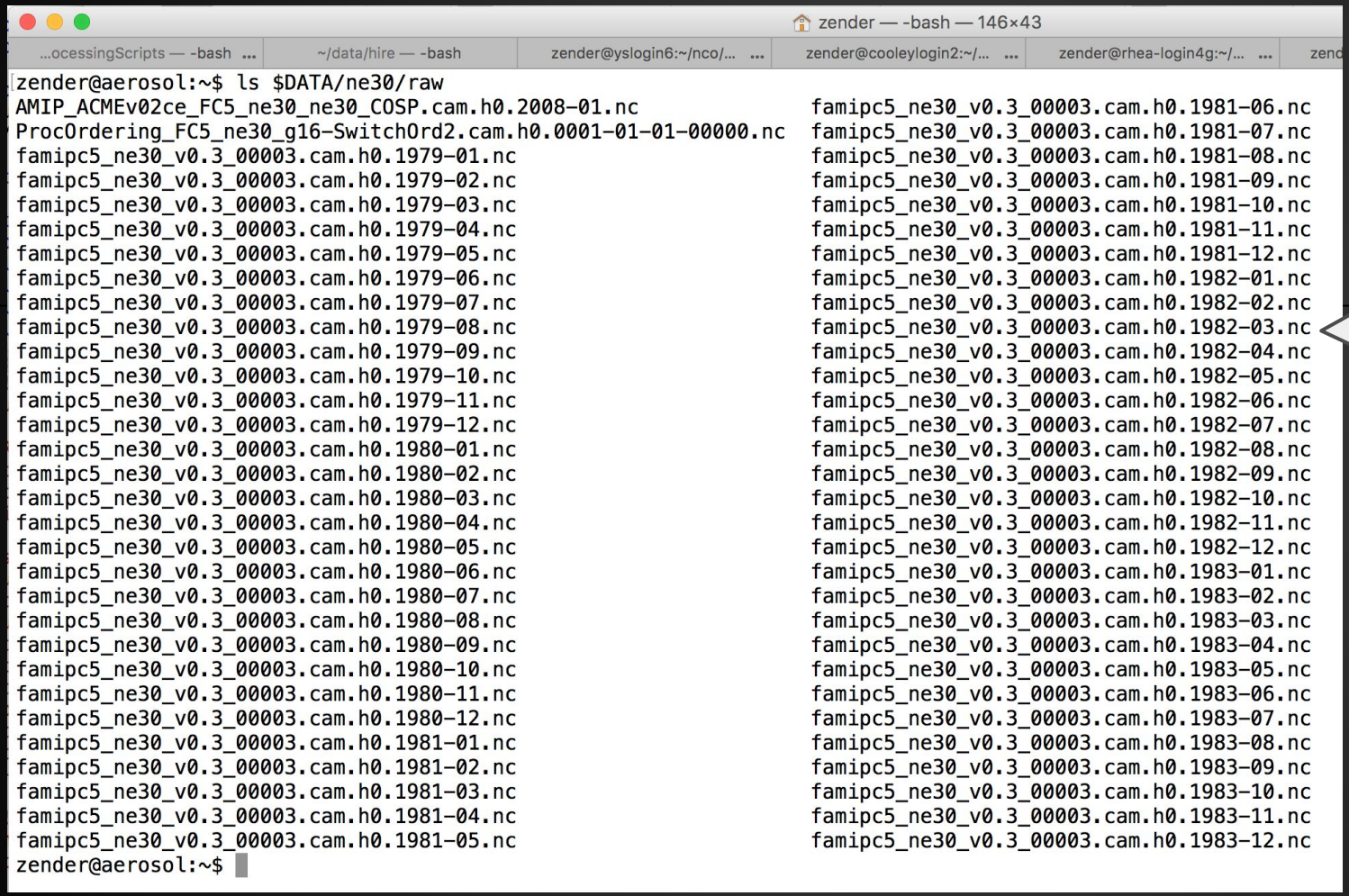
```
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out
> climo_nco.sh  # Prints help
```

climo_nco.sh Help Screen



Generate Climatology

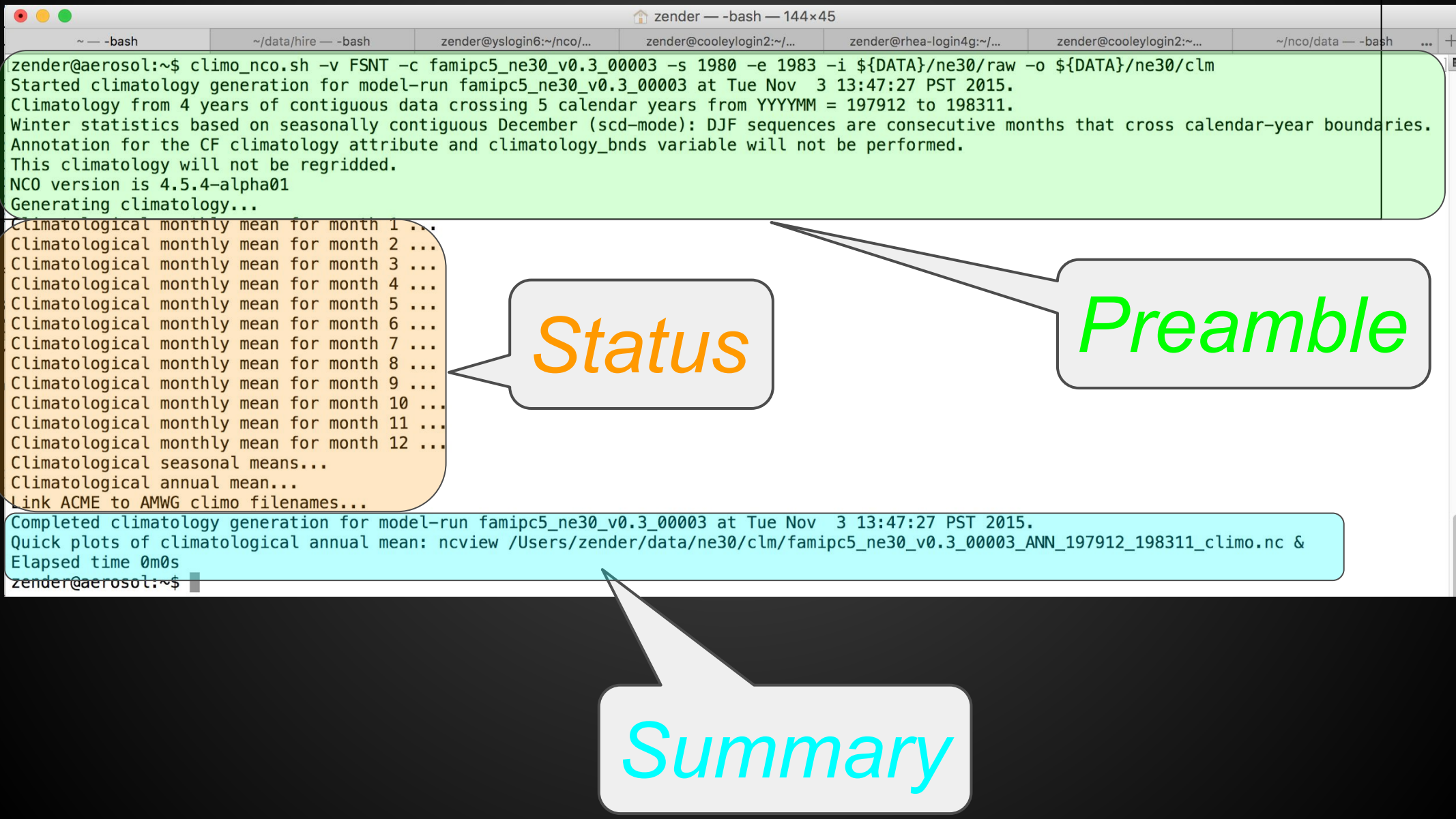
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out



Native grid input

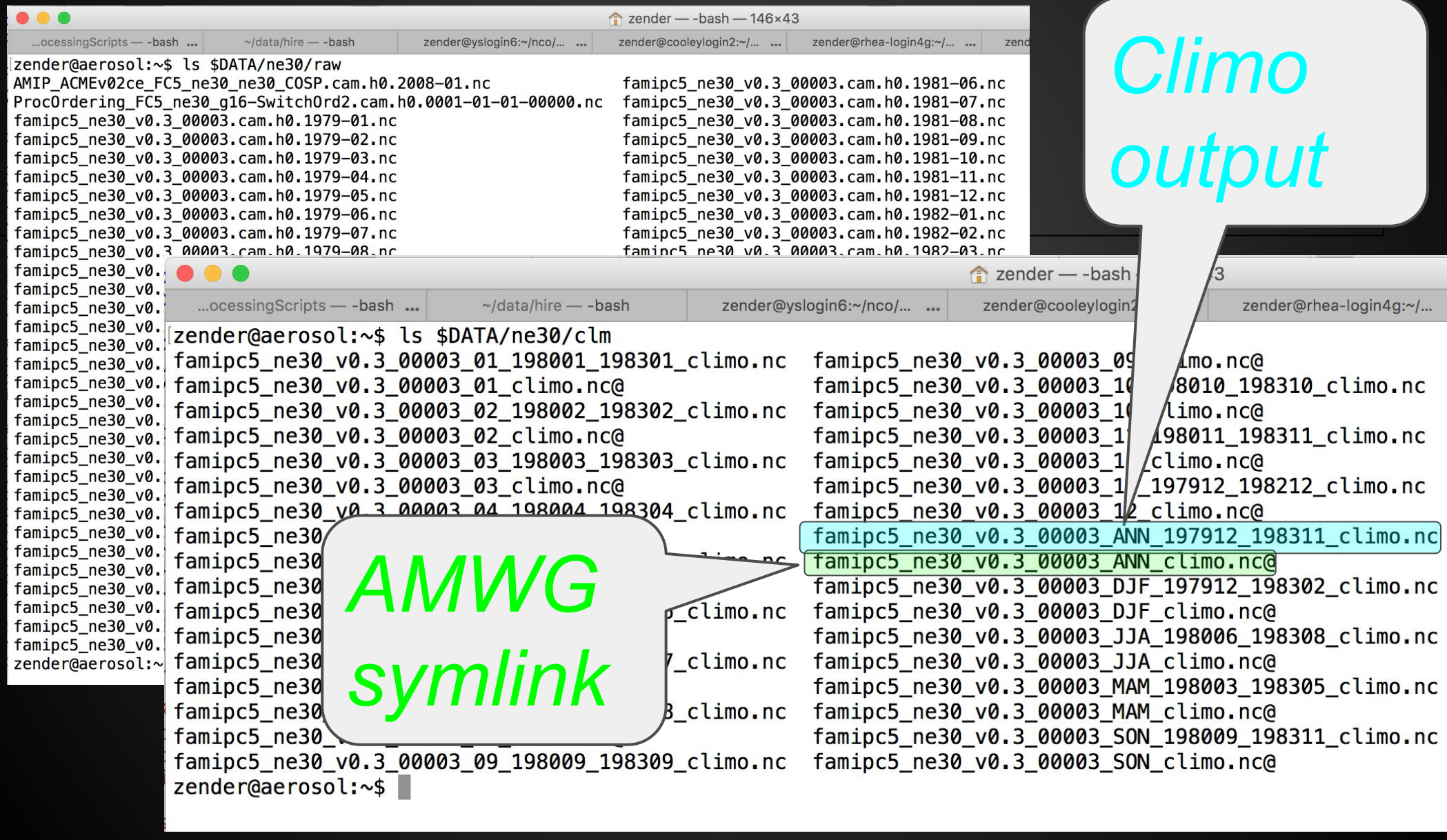
Standard Output

> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out



Climatology Files

> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out



climo_nco.sh Options

```
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out
> climo_nco.sh … -v FSNT,Q …                  # Subset
> climo_nco.sh … > climo_nco.out 2>&1 &       # top
> climo_nco.sh … -m clm2 -b tbnd …            # CLM2
> climo_nco.sh … -m cism -h h …               # CISM
> climo_nco.sh … -m pop -h h …                # POP
> climo_nco.sh … -R 'col_nm=nCells' …         # MPAS-O
> climo_nco.sh … -d 1 …                       # Debug
> climo_nco.sh … -f Experiment …              # Nickname
> climo_nco.sh … -p mpi …                     # Parallel
```

Climatologies

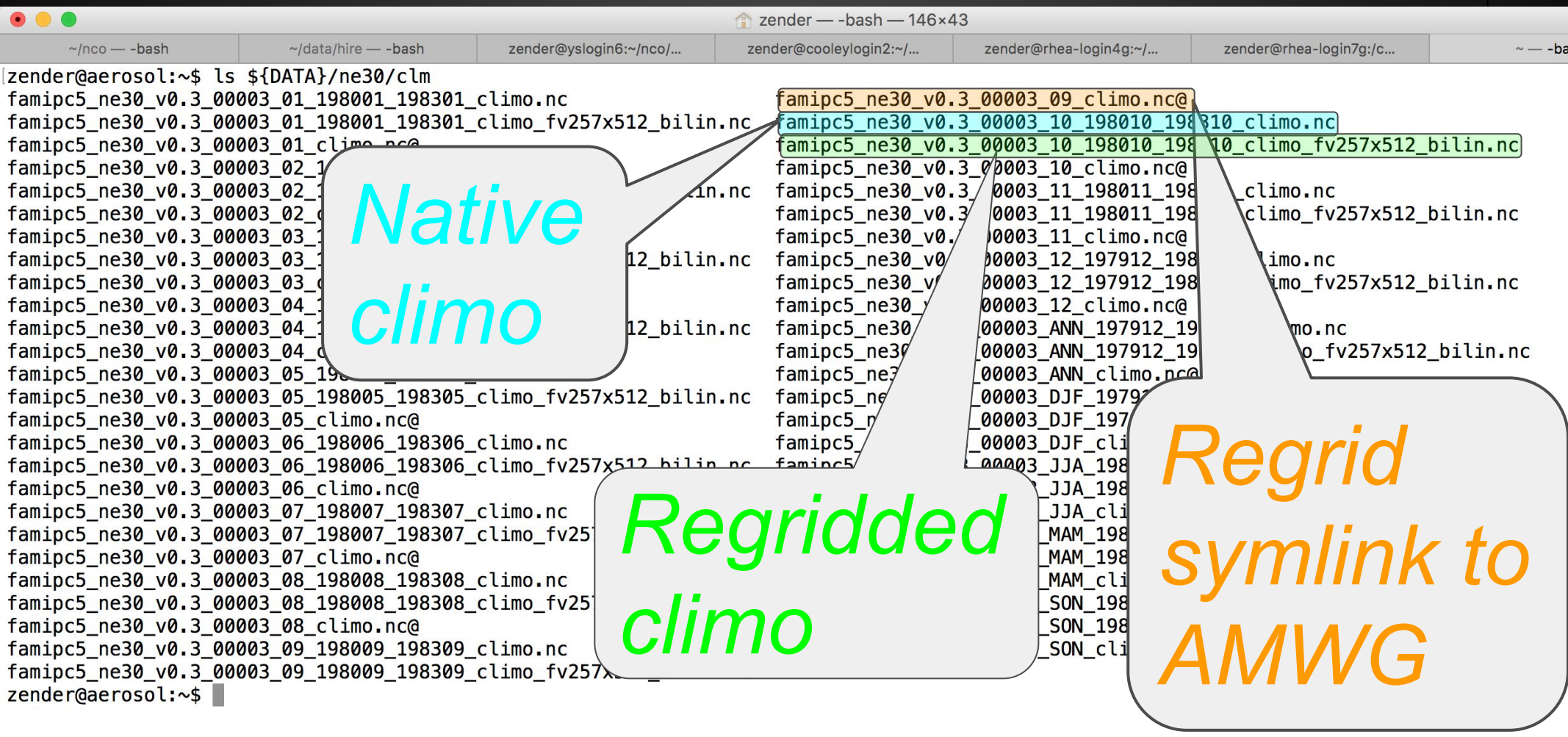
1. Prerequisites
2. Monthly, Seasonal, Annual "Climos"
3. Regridding
4. Parallelism Strategies
5. Examples for Cooley, Edison, Rhea

Simultaneous Regridding & "Loose Coupling"

```
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out
> climo_nco.sh … -r map.nc …                          # Mapfile
> climo_nco.sh … -r ~zender/data/maps/ \
  map_ne30np4_to_fv129x256_aave.20150901.nc
```

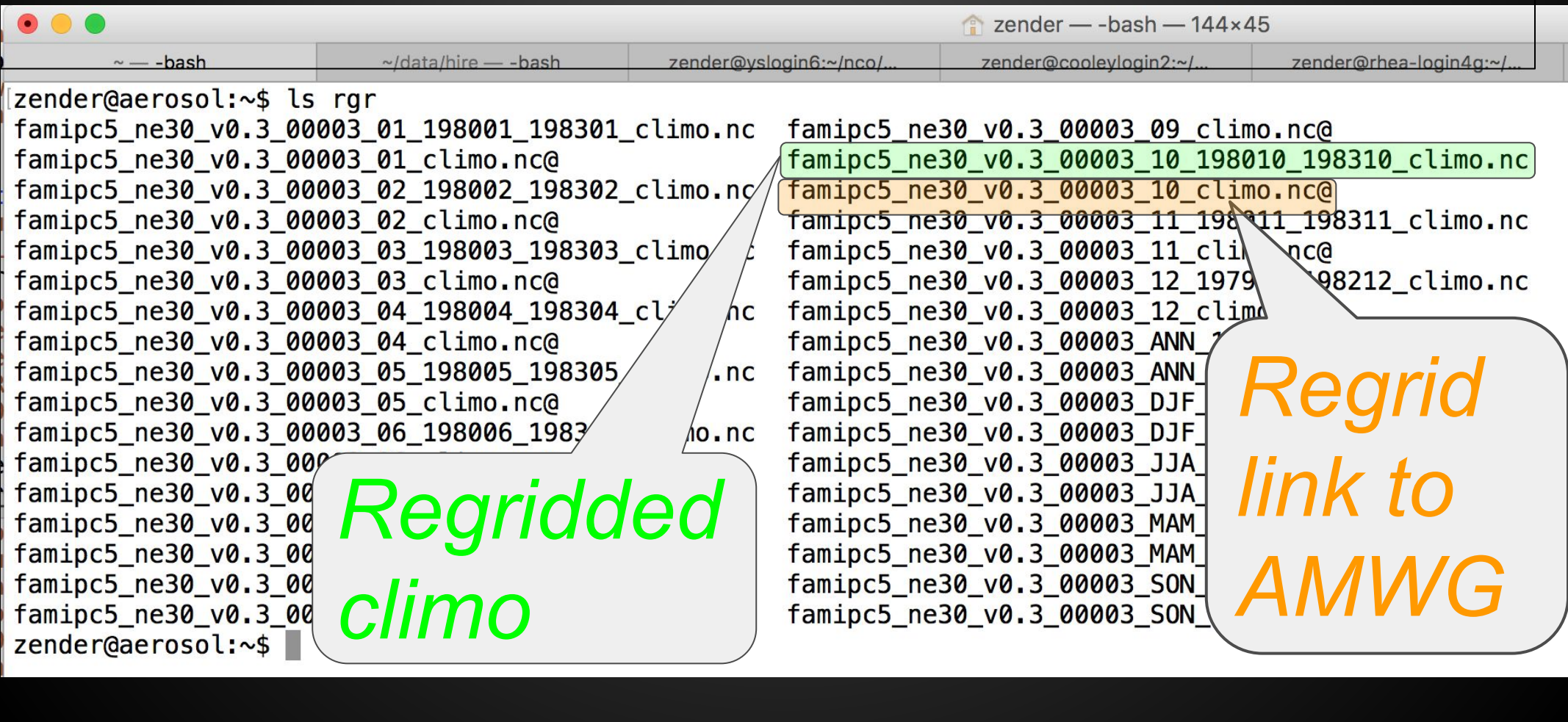
Regridding (1): Combined Destination

```
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out \
  -r map_ne30np4_to_fv257x512_bilin.20150901.nc
```



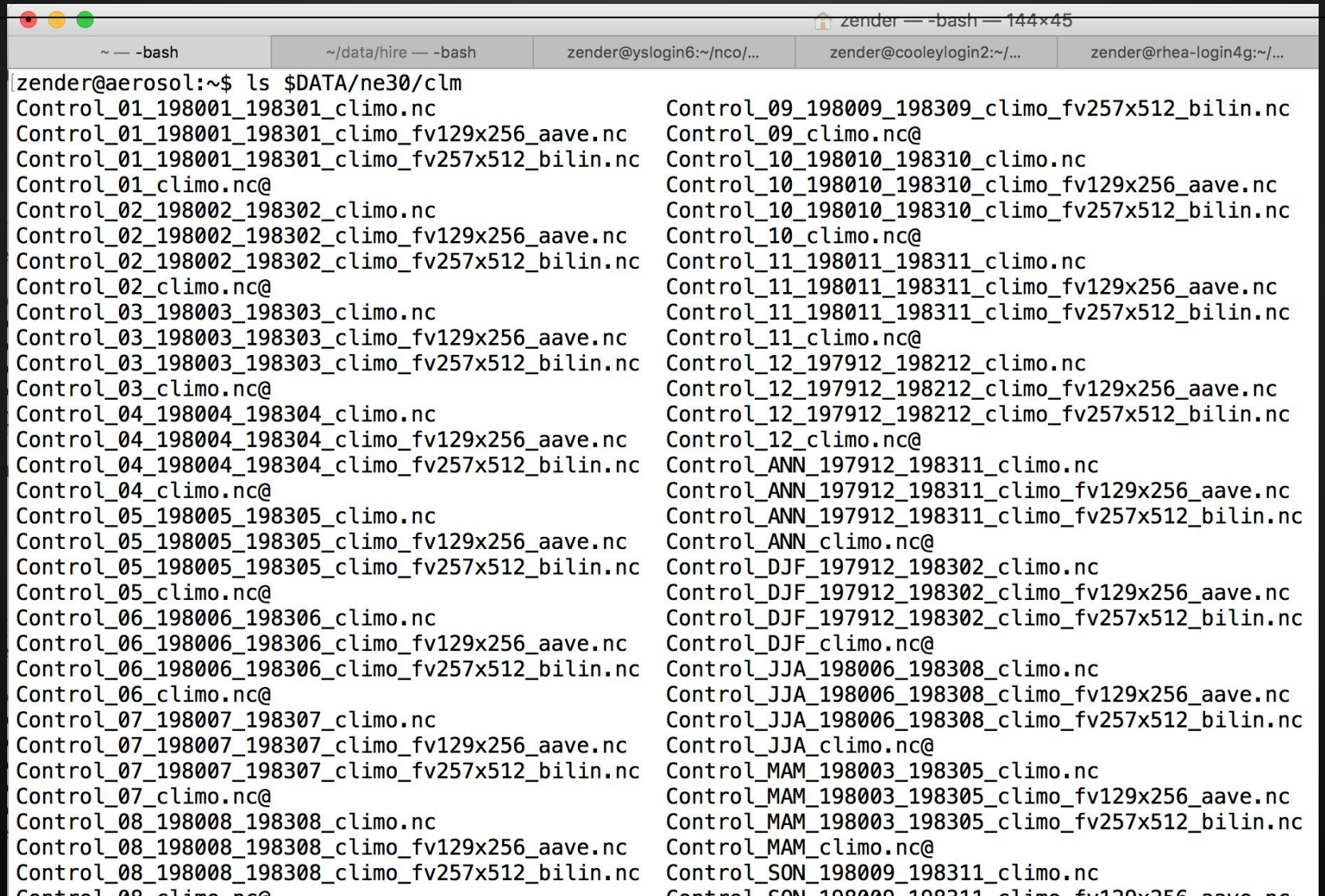
Regridding (2): Separate Destinations

```
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out \
  -r map_ne30np4_to_fv257x512_bilin.20150901.nc \
  -O rgr # rgr = regrid directory
```



Nicknames simplify names

```
> climo_nco.sh -c famipc5_ne30_v0.3_00003 … \
  -f Control # nickname replaces CASEID name
```



Climatologies

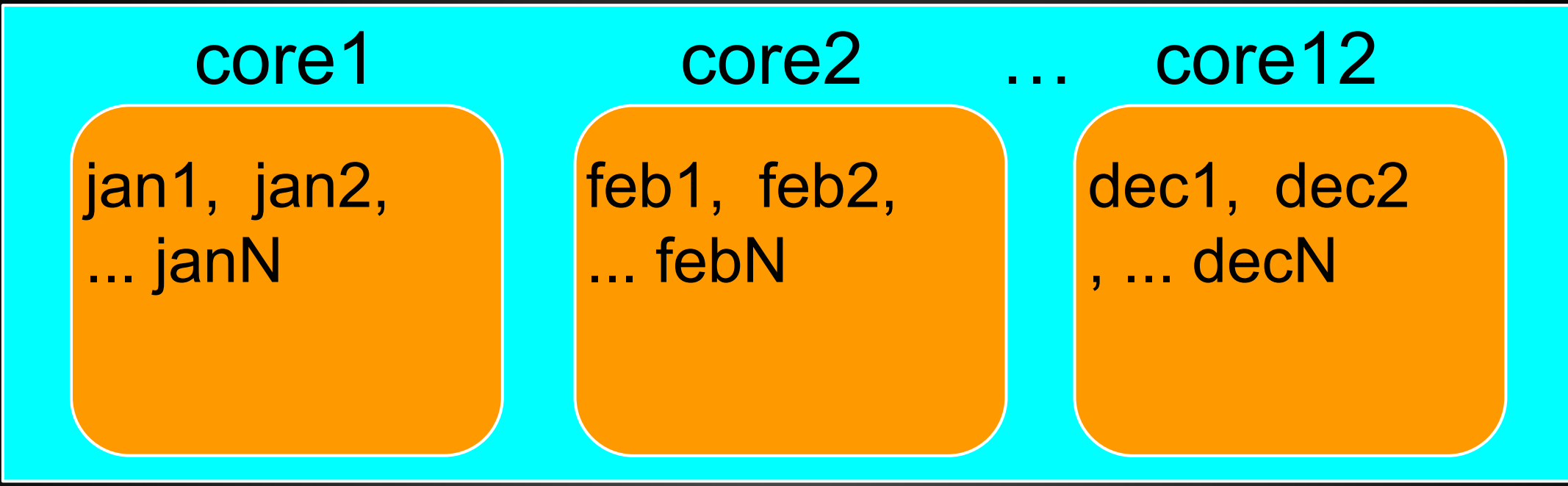
1. Prerequisites
2. Monthly, Seasonal, Annual "Climos"
3. Regridding
4. Parallelism Strategies
5. Examples for Cooley, Edison, Rhea

Three Parallelism Modes

```
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out
> climo_nco.sh … -p nil …      # Serial
> climo_nco.sh … -p bck ...    # Background (default)
> climo_nco.sh … -p mpi …      # MPI
```

- Serial: Execute sequentially on single node
- Background: Fork process for each month, then each season, all on single node. Optimal w/ 12 cores.
- MPI: Spawn MPI process to new node for each month, then each season. Optimal w/ 12 nodes.

Background Parallelism &

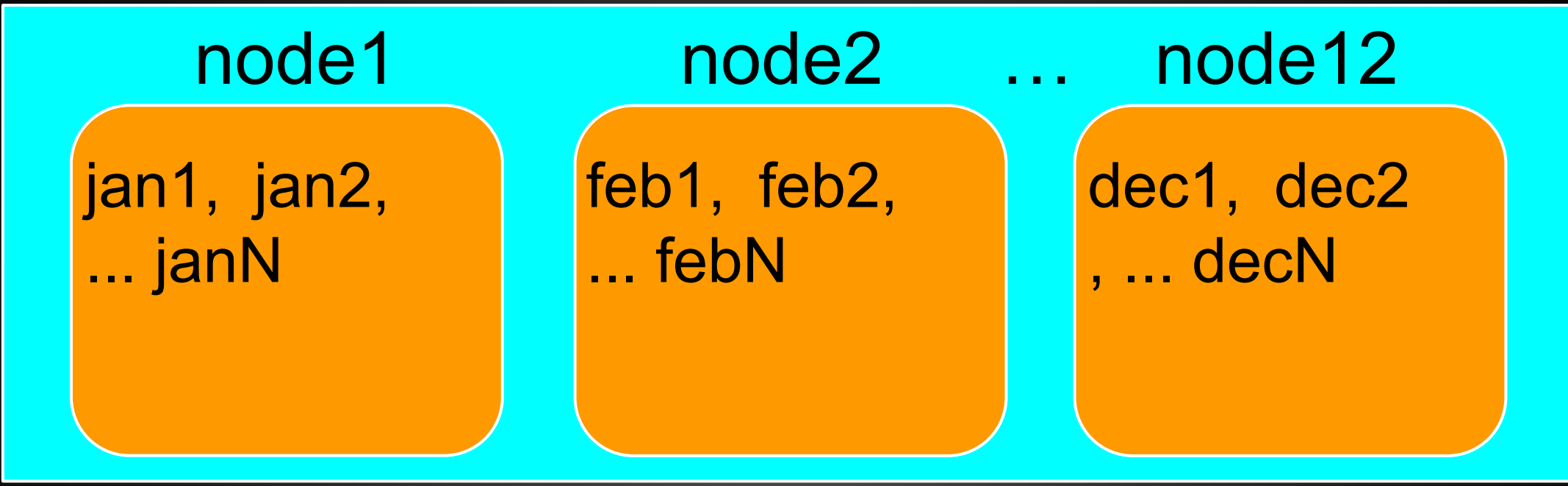


- Stage 1: 1 month per core (12 cores)
- Stage 2: 1 season per core (4 cores)

+ load-balanced regridding (8 cores)

- Stage 3: 1 annual (1 core) + load-balanced regridding (4 cores)

MPI Parallelism

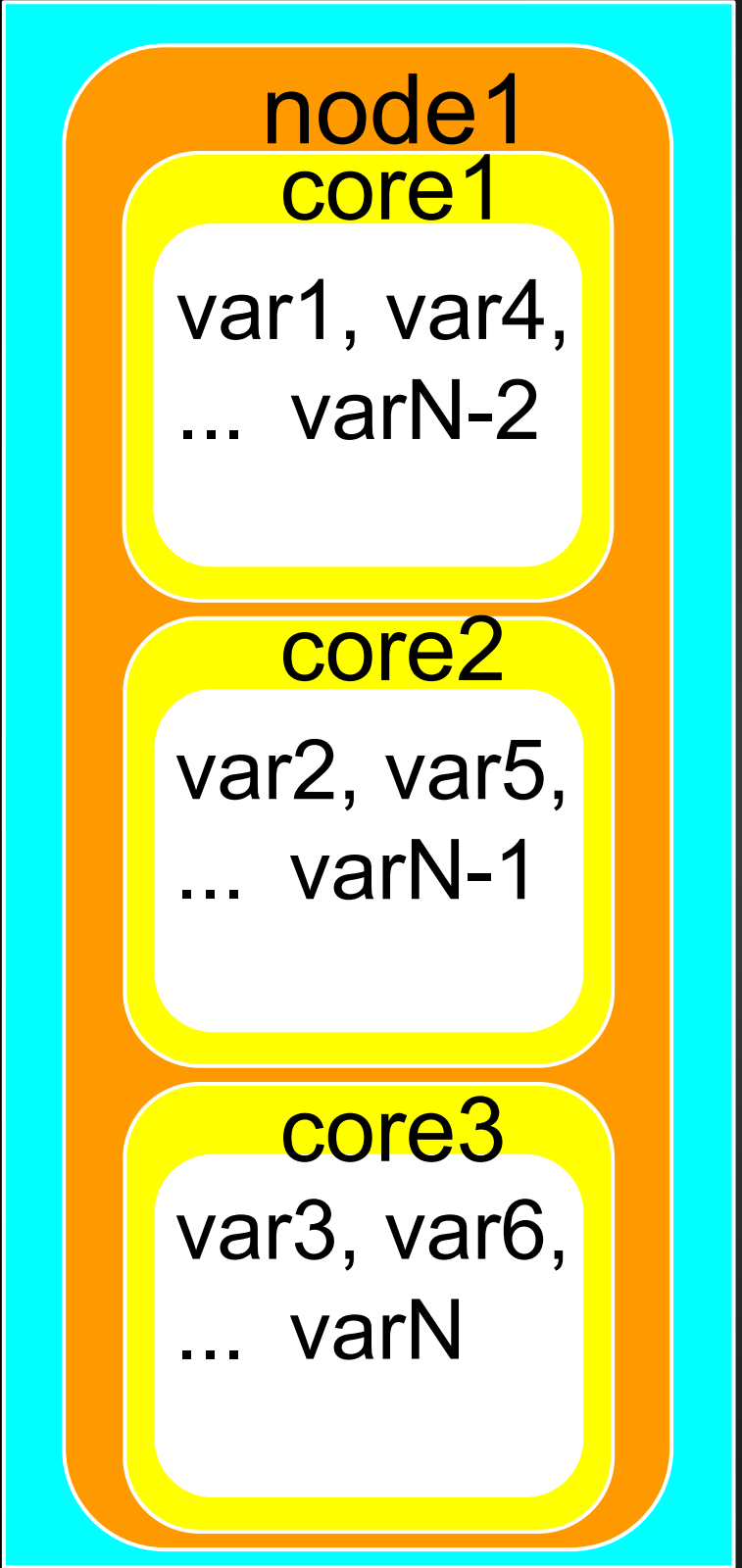


- Stage 1: 1 month per node (12 nodes)
- Stage 2: 1 season per node (4 nodes)

+ load-balanced regridding (8 nodes)

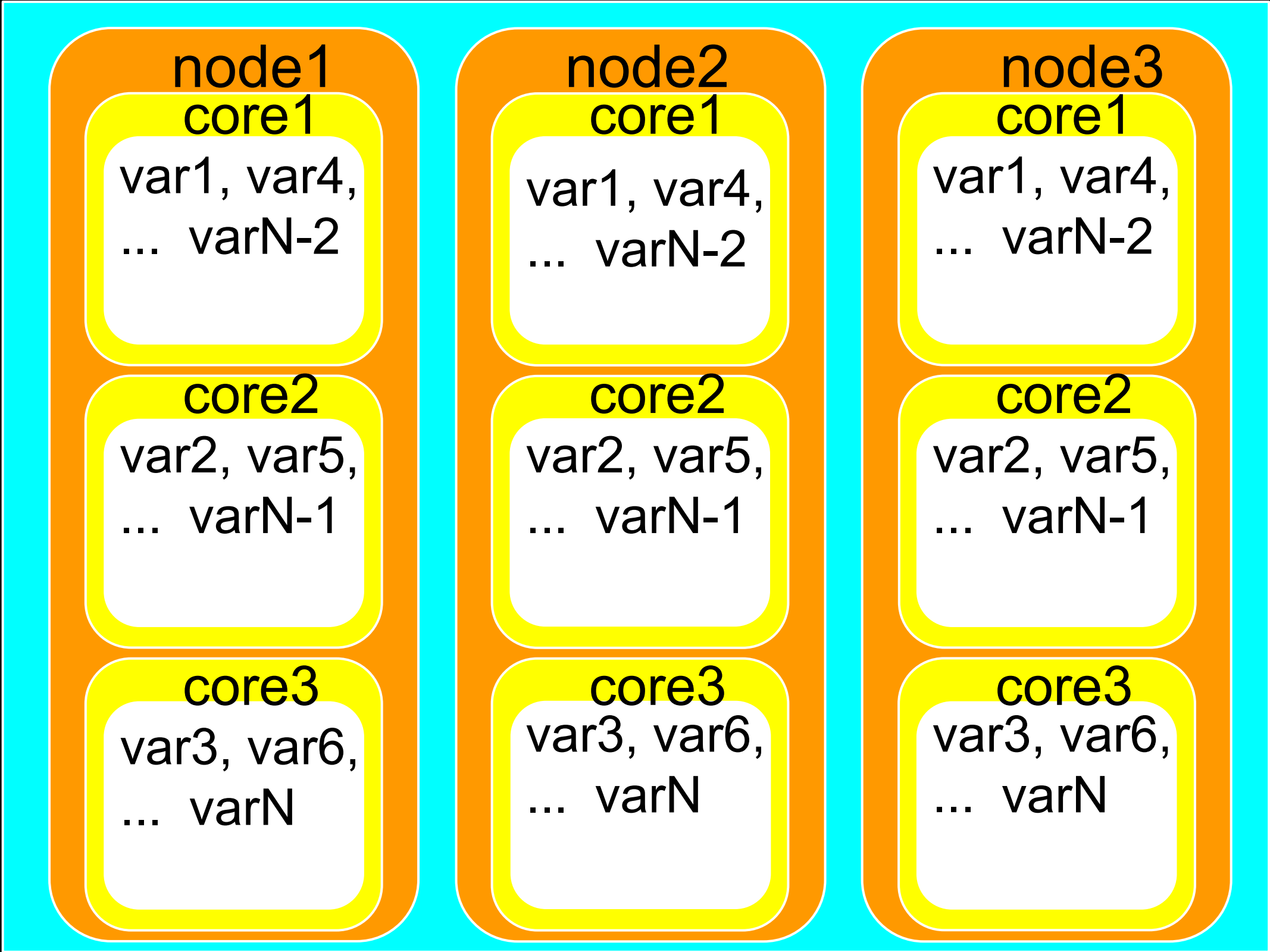
- Stage 3: 1 annual (1 node)

+ load-balanced regridding (4 nodes)



Threaded Regridding

- Regridding automatically invokes 2 (default) to 8 threads
- > climo_nco.sh…-t 8... # Threads


node1
core1
var1, var4,
... varN-2
core2
var2, var5,
... varN-1
core3
var3, var6,
... varN
node2
core1
var1, var4,
... varN-2
core2
var2, var5,
... varN-1
core3
var3, var6,
... varN
node3
core1
var1, var4,
... varN-2
core2
var2, var5,
... varN-1
core3
var3, var6,
... varN

Chooose Parallel Mode to Optimize Throughput

RAM required: ~4x sizeof(monthly file) per process

- Serial: Safest, slowest. Ideal for laptops, desktops. Alternative to MPI-mode for small RAM systems.
- Background: Default mode. Works on login nodes, compute nodes, and beefy (≥12 cores) workstations for ne30. Can work for ne120 on large memory compute nodes (e.g., Cooley = 384 GB/node).
- MPI: Necessary for fast ne120 on small-moderate (64-192 GB RAM/node) systems

Climatologies

1. Prerequisites
2. Monthly, Seasonal, Annual "Climos"
3. Regridding
4. Parallelism Strategies
5. Examples for Rhea, Cooley, Edison, (Yellowstone)

Rhea Example

```
> climo_nco.sh -p nil -v FSNT -c name -s 1980 -e 1985 -i in -o out -r map.nc # Fast initial serial to debug
> qsub -I -A CLI115 -V -l nodes=1 -l walltime=00:10:00 -N climo_nco # ne30, grab one interactive node
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out -r map.nc # ne30 production, background (ne120 dies!)
> qsub -I -A CLI115 -V -l nodes=12 -l walltime=00:30:00 -N climo_nco # ne120, grab twelve nodes
> climo_nco.sh -p mpi -c name -s 1980 -e 1985 -i in -o out -r map.nc # ne120 production, MPI
```

Cooley Example

```
> climo_nco.sh -p nil -v FSNT -c name -s 1980 -e 1985
-i in -o out -r map.nc # Fast initial serial to debug
> qsub -I -A HiRes_EarthSys --nodecount=1 --time= \
00:10:00 --jobname=climo_nco # ne30, one node
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out -r
map.nc # ne30 production, background (ne120 works!)
```

Edison Example

```
> climo_nco.sh -p nil -v FSNT -c name -s 1980 -e 1985
-i in -o out -r map.nc # Fast initial serial to debug
> qsub -I -A acme -V -l mppwidth=24 -l walltime= \ 00:
10:00 -q debug -N climo_nco # ne30, one node
> climo_nco.sh -c name -s 1980 -e 1985 -i in -o out -r
map.nc # ne30 production, background (ne120 dies!)
```

Other

1. Low frequency climos (from, e.g., annual mean output)

2. High frequency (daily/hourly h[1-N]) climos/regridding

3. Regridding unstructured data when horizontal dimension (i.e., cell/column) varies more slowly than others (e.g., vertical, species)

4. Extensive variables (e.g., numbers, counts) climos/regridding

http://dust.ess.uci.edu/hire

Supplementary Slides