Manual running POP
on Cartesius

8 october 2021

Michael Kliphuis

1. Introduction

This document describes how to run the high resolution (0.1 °) Parallel Ocean Program (POP) model on the national supercomputer 'Cartesius'.

For this specific resolution POP has a horizontal domain of 3600 x 2400 (longitude x latitude) gridpoints and 42 depth levels which adds up to 362,88 million gridpoints.

Chapter 2 describes how to build and run the POP model and check if the output is correct.

Chapter 3 is about the performance of the model. It described how to change the number of cores and the duration of the run. This information is needed for doing benchmark tests. In July 2020 Caspar van Leeuwen of SURFsara used this document to do benchmark tests with POP for the acquisition of the successor of Cartesius. That is why the name of the test run in this document is called `bmrun`.

2. Building and running POP

2.1 Get the code and initial files

1. Copy the tar file

   ```
   /projects/0/prace_imau/prace_2013081679/pop/tar/bmrun_pop.tar
   ```

   in your $HOME directory on Cartesius.

   The file contains all the code and initial files.
   It is quite big, 54GB so make sure you have enough free space.

2. In your $HOME type:

   ```
   tar xvf bmrun_pop.tar
   ```

All the needed POP files are now in $HOME/models/pop.
the three files below contain a lot of pathnames:

```
$HOME/models/pop/scripts/bmrun/build_pop.sc
$HOME/models/pop/scripts/bmrun/pop.slurm
$HOME/models/pop/scripts/bmrun/pop_in
```

They all start with $HOME (or ~). This means that if you untar the tar file in your own $HOME then everything should work.

2.2. Important directories and files

directory with source code	`$HOME/models/pop/code/source`
directory containing makefile `bull.gnu`	`$HOME/models/pop/code/build`
directory from which you start the run	`$HOME/models/pop/scripts/bmrun`
directory with output files	set this with environment variable `$outputdir_base` in `pop.slurm`
file with settings for the run e.g. duration, nr of cores etc	`$HOME/models/pop/scripts/bmrun/pop_in`
file that will be offered to the batch queue with sbatch	`$HOME/models/pop/scripts/bmrun/pop.slurm`
fortran file containing settings that are used for decomposing the model domain across the cores.	`$HOME/models/pop/scripts/bmrun/` `POP_DomainSizeMod.F90`

2.3 Build

Next we will compile the POP code and make an executable.

1. `cd  $HOME/models/pop/scripts/bmrun`

2. make sure you load all needed modules by typing:

```
module purge
module load pre2019
module load surfsara
module load fortran
module load c
module load iimpi/2016b
module load hdf5/serial/intel/1.8.10-patch1
module load netcdf
module load mkl
```

Note that now Intel 15.0.0 fortran and c libraries are loaded, you may want to use newer ones.

3. type:

```
./build_pop.sh
```

This will build the executable `./pop` in the present work directory for a run on 1280 cores. If after building the executable is indeed there then everything went fine. If not then find out what went wrong by checking the file

```
build.log.pop.bmrun.<yymmdd-hhmmss>
```

2.4 Run

1. Open the file `pop.slurm`.

   This is the job script for SLURM. It also creates the output directory via environment variable `outputdir_base.`

   The output files are quite big (many GBs) so it is best to choose a 'project directory' on Cartesius, in my case I set:

   ```
   export outputdir_base=/projects/0/prace_imau/prace_2013081679/pop
   ```

   All output of the run will end up in its underlying directories:

   ```
   restart      (restart files)
   tavg         (monthly means)
   movie        (daily means)
   ```

2. Open the file `pop_in`.

   This is the so called namelist file that contains all the settings for the run i.e. output directories, duration of the run, which initial files to use and many parameters that determine the physics of the run (diffusion, viscosity, mixing etc).

   Somehow it is not possible anymore to pass the environment variable `outputdir_base` to SLURM so in `pop_in` you need to set by hand:

   ```
   restart_outfile = '/projects/0/prace_imau/prace_2013081679/pop/restart/r'
   tavg_outfile    = '/projects/0/prace_imau/prace_2013081679/pop/tavg/t'
   movie_outfile   = '/projects/0/prace_imau/prace_2013081679/pop/movie/m'
   ```

3. In this case the model starts from a restart file which it reads from file `pointer.restart` During the run this pointer file is updated. To make sure we start from the correct one (in this case one of January year 150 of another run) type:

   ```
   cp pointer.restart_1500101 pointer.restart
   ```

4. Finally type:

   ```
   sbatch pop.slurm
   ```

   This starts a POP model job on 1280 (Haswell) cores that runs for 5 modeldays.

   You can check in what day the simulation is by 'tail'-ing or looking in the slurm file.

2.5 Output

The output of a run is written to the directory `$outputdir_base` (see 2.4)

There are three types of output files:

1. `$outputdir_base/restart` contains the restart files.

   In this case the following restartfile is created:

   `r.t0.1_42l_nccs01.end.nc`

2. `$outputdir_base/tavg` contains files with monthly averaged fields.

   You can set the fields you want to output (TEMP,SALT etc) in file:

   `$HOME/models/pop/scripts/bmrun/tavg_contents`

   In this case there are no files in this directory. There will only be files if `stop_count` in file

   `$HOME/models/pop/scripts/bmrun/pop_in`

   has a value > than 31 (= # model days if `stop_option = 'eod'` meaning end of day)

3. `$outputdir_base/movie` contains files with daily averaged values.

   You can set the fields you want to output (TEMP_5m, SALT_5m etc) in file:

   `$HOME/models/pop/scripts/bmrun/movie_contents`

   In this case there are 5 files. You can quickly check them as follows:

Open the last moviefile with application 'ncview'. Do this by going to the movie output directory and type:

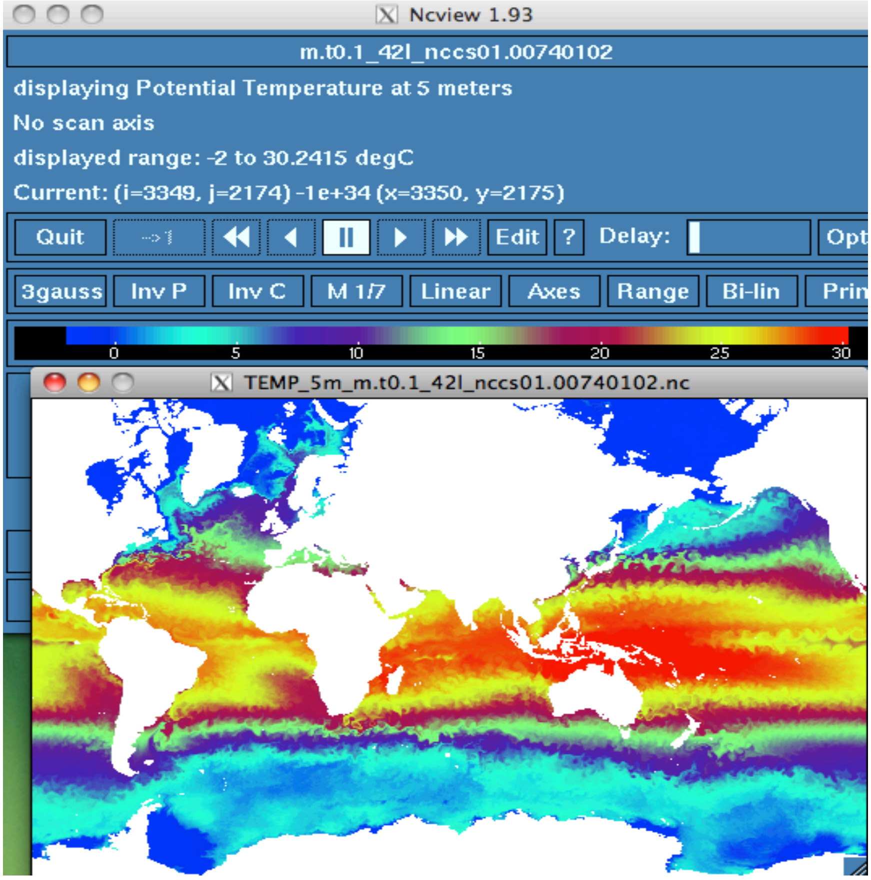
```
module load ncview
ncview  m.t0.1_42l_nccs01.01500106.nc
```

Then check the following fields:

TEMP_5m = Temperature of seawater at 5m depth in ° Celcius
SALT_5m = Salinity of seawater at 5m depth in psu
UVEL_5m = Zonal velocity of seawater at 5m depth in cm/sec

These fields should look similar to the screenshots below

TEMP_5m



Ncview 1.93

m.t0.1_42l_nccs01.00740102

displaying Potential Temperature at 5 meters

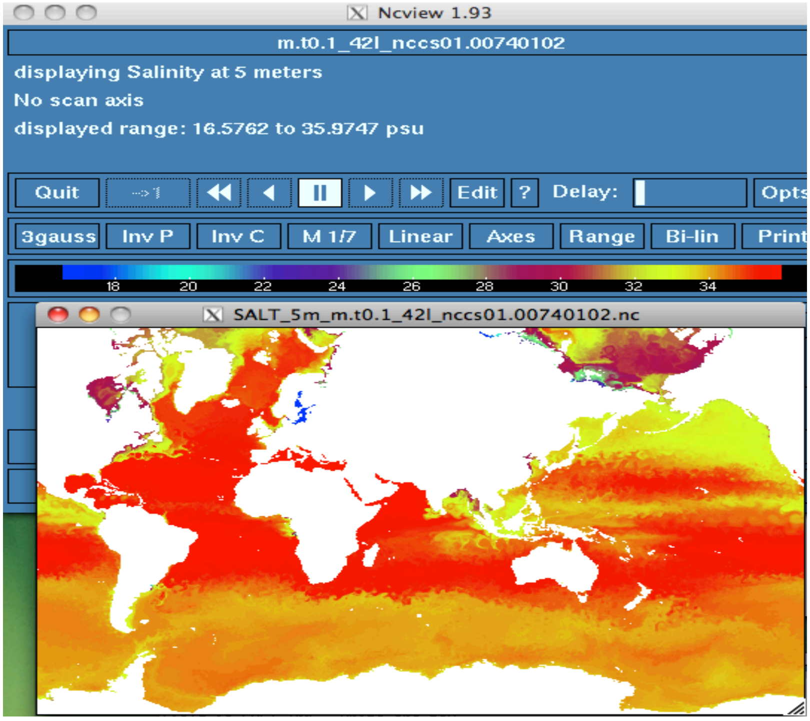
No scan axis

displayed range: -2 to 30.2415 degC

Current: (i=3349, j=2174) -1e+34 (x=3350, y=2175)

Range must be in the order of -2 to 30 degrees Celcius

SALT_5m



Ncview 1.93

m.t0.1_42l_nccs01.00740102

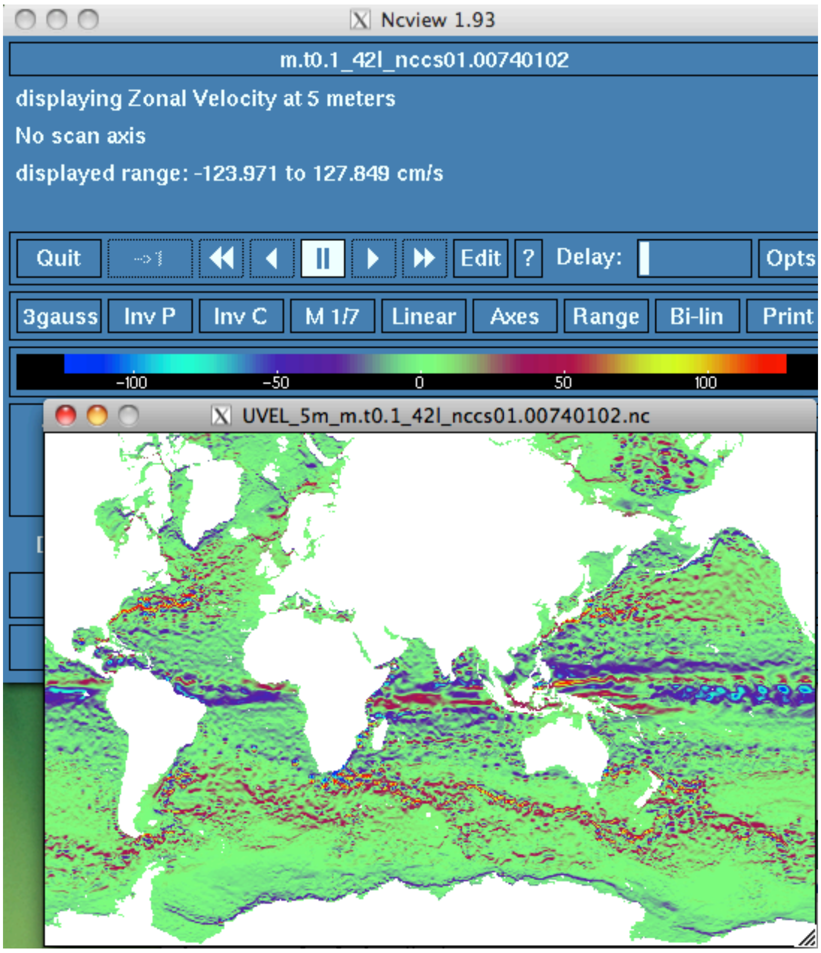
displaying Salinity at 5 meters

No scan axis

displayed range: 16.5762 to 35.9747 psu

Range must be in the order of 16 to 35 psu

UVEL_5m



Ncview 1.93

m.t0.1_42l_nccs01.00740102

displaying Zonal Velocity at 5 meters

No scan axis

displayed range: -123.971 to 127.849 cm/s

UVEL_5m_m.t0.1_42l_nccs01.00740102.nc

Range must be in the order of -123 to 127 cm/s

3. Performance of POP

When the run is finished, the performance of the run can be measured via the job statistics section in the end of the slurm file in the run directory `$HOME/models/pop/scripts/bmrun`

With the current settings i.e. running on 1280 cores* for 5 model days and write a daily output file every model day there is a lot of I/O. The run takes about 7 minutes so the performance is 5/7 * 60 * 24 ≅ 1028 modeldays/24h ≅ 2.8 modelyears/24h.

The performance obviously depends on the number of cores you run on. Section 3.1 describes how to change this.

To get a better picture of the true performance it is better to run for at least 1 month and not output files every day but every month. In our production runs we usually do not output daily fields. How you do this is shown in section 3.2. The performance then increases from 2.8 to approximately 4 modelyears/24h.

* we are actually running on 1296 cores because we need to use full Haswell nodes (in total 54)

3.1 Change the number of cores

The number of cores on which POP runs is now set to 1280 but if needed they can be changed as follows:

Suppose you want to run on 640 cores then:

1. Open file: `$HOME/models/pop/scripts/bmrun/pop_in`

   and set:

   ```
   nprocs_clinic  = 640
   nprocs_tropic  = 640
   ```

2. Open file: $HOME/models/pop/scripts/bmrun/pop.slurm

   and set:

   ```
   #SBATCH –n  640
   ```
   If needed change the wallclocktime

   ```
   #SBATCH --time=30:00:00
   ```

   Usually the needed wallclocktime for running 5 modeldays on 640 cores
   is less than 20 min.

3. Open file: `$HOME/models/pop/scripts/bmrun/POP_DomainSizeMod.F90`

   In this file you set the blocksize i.e. the number of grid points in the horizontal direction and over all 42 depth levels (X x Y x 42) that each core handles.
   The 0.1° POP model is defined on a 3600 x 2400 x 42 gridpoints.
   For running on 640 cores we set

   ```
   POP_blockSizeX = 180
   POP_blockSizeY = 75
   ```

   meaning that each core handles 180x75 (x42) grid points;
   indeed 3600 x 2400 / (180 x 75) = 640.

   To each number of cores (a user wants to run on) belong specific blocksizes.
   More options are given in comments in `POP_DomainSizeMod.F90`

3.2 Change the duration of the run

If you want to run more than 5 model days then change the value of `stop_count` in

```
$HOME/models/pop/scripts/bmrun/pop_in
```

Note that `stop_option     = ‘eod’` meaning end of day.
You can also set it to `‘eom’` or `‘eoy’` for months resp. years.

So setting

```
stop_option = ‘eom’
stop_count  = 1
```

lets the model run for 1 model month, in this case 31 days because it starts in January.