POPPerf

POPperf is a custom high performance large scaling version of POP written by John Dennis.

POPperf requires a Fortran compiler, an MPI implementation and NetCDF.

POPperf can be downloaded fromhttp://web.ncar.teragrid.org/~bmayer/interconnectPOPperf.tar.gz

Build included support libraries:

```
cd pio/mct
export PIOARCH=juropa
vim Makefile.conf
Modify INSTALL, MKINSTALLDIRS, abs_top_builddir, MCTPATH, MPEUPATH, EXAMPLEPATH
vim ../Makefile.juropa
Modify SNETCDF, MPIINC, FC, CC
make
cd ../pio
make clean
make
```

Build POP:

```
cd ../../gx01v2/
export ARCHDIR=intel_mpi
vim intel_mpi.gnu
Modify FC, LD, CC, SNETCDF (both instance), MPILIB, MPIINC
make
```

To run POPperf:

```
mpirun -np <# processors> ./pop
```

To change the number of processors that the program is run on, look in ‘pop_in’ for nprocs_clinic and nprocs_tropic and change the parameters as appropriate.

The program may produce an error saying that the max number of blocks needs to be increased. Change this in the domain_size.F90 file. Modify max_blocks_clinic and max_blocks_tropic as in the below table and recompile POP (Building POP section only).

When choosing which processor counts to start with we tend to start with 541 cores as a base that most systems can handle, then choose several others fairly far apart to see where the scaling curve starts to turn over. With that data in hand we refine the turn over point by running several more test cases near the inflection point.

What are the block sizes that can be used? From Inverse Space-Filling Curve Partitioning of a Global Ocean Model

block_size_x	block_size_y	maxblock	nblocks	# cores (nprocs_clinic, nprocs_tropic)
144	96	1	541	541
120	80	1	764	764
90	60	1	1312	1312
72	48	1	2009	2009
60	40	1	2822	2822
45	30	1	4884	4884
36	24	1	7545	7545

To generate new # of cores use the following formula to update the maxblock and # cores column:

```
# cores = ceil( nblocks / max blocks)
```

For example if we took the above listed row with 2009 processors and set maxblocks to 3 we would want to use 670 cores. The new line in the above table would look like:

block_size_x	block_size_y	maxblock	nblocks	# cores (nprocs_clinic, nprocs_tropic)
72	48	3	2009	670

Reporting