

Running the Isca framework on the cluster

Isca is a framework for the idealized modelling of the global circulation of planetary atmospheres at varying levels of complexity and realism. The framework is an outgrowth of models from GFDL designed for Earth's atmosphere, but it may readily be extended into other planetary regimes.

Installation

First of all define a folder `${WORK}` on the `/work` or the `/scratch` filesystem (somewhere where you have write permissions):

```
export WORK=/work/FAC/...  
mkdir -p ${WORK}
```

Load the following relevant modules and create a python virtual environment:

```
dcsrsoft use arolle  
  
module load gcc/10.4.0  
module load mvapich2/2.3.7  
module load netcdf-c/4.8.1-mpi  
module load netcdf-fortran/4.5.4  
module load python/3.9.13  
  
python -m venv ${WORK}/isca_venv
```

Install the required python modules:

```
${WORK}/isca_venv/bin/pip install dask f90nml ipykernel Jinja2 numpy pandas pytest sh==1.14.3  
tqdm xarray
```

Download and install the Isca framework:

```
cd ${WORK}
git clone https://github.com/ExeClim/Isca
cd Isca/src/extra/python
${WORK}/isca_venv/bin/pip install -e .
```

Patch the Isca makefile:

```
sed -i 's/-fdefault-double-8$/-fdefault-double-8 \\n          -fallow-invalid-boz -fallow-argument-mismatch/' ${WORK}/Isca/src/extra/python/isca/templates/mkmf.template.gfort
```

Create the environment file for curnagl:

```
cat << EOF > ${WORK}/Isca/src/extra/env/curnagl-gfortran
echo Loading basic gfortran environment

# this defaults to ia64, but we will use gfortran, not ifort
export GFDL_MKMF_TEMPLATE=gfort

export F90=mpifort
export CC=mpicc
EOF
```

Compiling and running the Held-Suarez dynamical core test case

Compilation takes place automatically at runtime. After logging in to the cluster, create a SLURM script file `start.sbatch` with the following contents:

```
#!/bin/bash -l

#SBATCH --account ACCOUNT_NAME
#SBATCH --mail-type ALL
#SBATCH --mail-user <first.lastname>@unil.ch

#SBATCH --chdir ${WORK}
#SBATCH --job-name isca_held-suarez
#SBATCH --output=isca_held-suarez.job.%j

#SBATCH --partition cpu
```

```

#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 16
#SBATCH --mem 8G
#SBATCH --time 00:29:59
#SBATCH --export ALL

dcsrsoft use arolle

module load gcc/10.4.0
module load mvapich2/2.3.7
module load netcdf-c/4.8.1-mpi
module load netcdf-fortran/4.5.4

WORK=$(pwd)

export GFDL_BASE=${WORK}/Isca
export GFDL_ENV=curnagl-gfortran
export GFDL_WORK=${WORK}/isca_work
export GFDL_DATA=${WORK}/isca_gfdl_data

export C_INCLUDE_PATH=${NETCDF_C_ROOT}/include
export LIBRARY_PATH=${NETCDF_C_ROOT}/lib

sed -i "s/^NCORES =.*$/NCORES = $(echo ${SLURM_CPUS_PER_TASK:-1})/"
${GFDL_BASE}/exp/test_cases/held_suarez/held_suarez_test_case.py

${WORK}/isca_venv/bin/python $GFDL_BASE/exp/test_cases/held_suarez/held_suarez_test_case.py

```

You need to carefully replace, at the beginning of the file, the following elements:

- On line 3: **ACCOUNT_NAME** with the project id that was attributed to your PI for the given project
- On line 5: **<first.lastname>@unil.ch** with your e-mail address (or double-comment that line with an additional '#' if you don't wish to receive e-mail notifications about the status of the job)
- On line 7: **\${WORK}** must be replaced with the **absolute path** (ex. */work/FAC/.../isca*) to the chosen folder you created on the installation steps
- On line 15-17: you can adjust the number of CPUs, the memory and the time for the job (the present values are appropriate for the default Held-Suarez example)

Then you can simply start the job:

```
sbatch start.sbatch
```

Révision #10

Créé 6 mars 2023 13:49:27 par Flavio Calvo

Mis à jour 22 octobre 2024 08:40:26 par Flavio Calvo