CVTK_FAST suite on DATARMOR

Author: G. Cambon, LOPS/IRD, gildas.cambon@ird.fr
Version: V1.1
Date: 2307 March 2019

1. Introduction and general principles

CVTK_FAST is a suite of scripts used to test the parallel reproducibility of the CROCO ocean model. It has been developed to be operational on the Ifremer supercomputer DATARMOR but the scripts can be adapted to other computers.

The general concept is to launch a series of test in parallel covering large choices of options in the code to test i) the compilation, ii) the execution and iii) the parallel reproducibility.

What means the parallel reproducibility? It is when we get exactly the same results using either sequential either parallel computing with either domain tiling (1X2 or 2x1, or 2x2 or whatever).

For this purpose a dedicated cppkey has been coded in CROCO, ***#define RVTK_DEBUG***.

- In sequential mode, it writes a binary file (called ***check_file***) where defined array are saved at every time step.
- In parallel mode, this file is read, if the array value is different from the array read, the code stops with an error message. The message is explicit mentioning the array that differ and the location (i,j) where the difference is observed.

It is a great capability used for a few years to detect parallel reproducibility error.

2. Architecture of the CVTK_FAST suite

The CVTK_FAST suite is divided in 3 mains family of tests:

- The various **analytical test-cases**, referred as KTEST
- The **regional test-case**, based on the BENGUELA_VHR configuration, using various set of cppkey referred as REG (for regional)
- The **semi analytical vortex test-case**, referred as VORT, dedicated to test the online AGRIF nesting procedure. The reproducibility without AGRIF nesting, then with AGRIF, one-way and 2 way).

All the scripts are located in the GIT repository of CROCO, in the directory **`CVTK/test_repro/CVTK_DEBUG_FAST_src`**

You will find several types of files:

- Some configuration files that will be sourced in the scripts:
  - **CONFIGURE_ANA** (ANA suffix is for analytical test-case)
  - **CONFIGURE_REG** (REG suffix is for regional test-cases)
  - **CONFIGURE_VORT (**VORT suffix is for vortex test-case)

- Some directories dedicated to scripts for specific family [KTEST, VORT, REG]:
  - **Scripts_reg/**
  - **Scripts_vort/**
  - **Scripts_ana/**

- **TEST_CASES_CVTK/:** contain the input files (netcdf, .in, AGRIF_FixedGrid.in, etc…) needed for all the various test cases.

- **src_global :** contain the encapsulated bash scripts used to launch the full CVTK_FAST test suite.

3. Procedure to launch the CVTK testing

A. Step-by-step procedure

- Define the CVTK_FAST source directory
  - **setenv CVTKHOME $HOME/croco/CVTK/test_repro/CVTK_DEBUG_FAST_src**

- Create the work directory for the tests
  - **cd $DATAWORK**
  - **mkdir -p TESTROOT/VORT TESTROOT/KTEST TESTROOT/REG**
  - **setenv CVTKWORK $DATAWORK/TESTROOT**

- Edit the jobcomp_rvtk.bash and the various CONFIGURE_[GLOBAL,ANA,REG,VORT] files
  - **cd $CVTKHOME**
  - **gedit jobcomp_rvtk.bash**
  - **gedit CONFIGURE_GLOBAL**
  - **gedit CONFIGURE_[ANA,REG,VORT]**

- Create the KTEST (for analytical), REG and VORT tests environment
  - **cd $CVTKHOME**
  - **cd Scripts_ana ; ./create_link_master_ana.sh ; cd -**
  - **cd ./Scripts_vort ; ./create_link_master_vort.sh ; cd -**
  - **cd ./Scripts_ana ; ./create_link_master_reg.sh ; cd -**

- For KTEST family

cd $DATAWORK/TESTROOT/KTEST

- Launch tests suite
  - **./mk_TESTALL_ana.bash**
- Check the right compilation:
  - **./mk_CHECKALL.bash**
    - DONE=> test passed
    - BUGBIN detection=> test failed
- Extract and gather the results in a text file
  - **./gather_recap.bash KTEST**
  - **creation of the log file: KTEST_gather_recap_*yyyymmdd*_git*revnumber***
- Clean the tests suite :
  - **./mk_CLEANALL.bash**

- For VORT family

  cd $DATAWORK/TESTROOT/VORT
  - Launch tests suite
    - **./mk_TESTALL_vort.bash**
  - Check the right compilation:
    - **./mk_CHECKALL.bash**
      - Done=> test passed
      - BUGBIN=> detection: test failed
  - Extract and gather the results in a text file
    - **./gather_recap.bash VORT**
    - **creation of the log file: VORT_gather_recap_*yyyymmdd*_git*revnumber***
  - Clean the tests suite :
    - **./mk_CLEANALL.bash**

- For REG family

  cd $DATAWORK/TESTROOT/REG
  - Launch tests suite:
    - **./mk_TESTALL_reg.bash**
  - Check the right compilation:
    - **./mk_CHECKALL.bash**
      - Done: test passed
      - BUGBIN detection: test failed
  - Extract and gather the results in a text file:
    - **./gather_recap.bash VORT**
    - **=> creation of the log file: REG_gather_recap_*yyyymmdd*_git*revnumber***
  - Clean the tests suite :
    - **./mk_CLEANALL.bash**

The log file produced by the CVTK_FAST test suite are copy here
/home/datawork-croco/public/ftp/CVTK_FAST/Log_Summary

They are available online here :
ftp://ftp.ifremer.fr/ifremer/croco/CVTK_FAST/Log_Summary/

B. Automatic procedure (to be used for example in a crontab)

You can launch all the procedure automatically using the scripts in **$CVTKHOME_repro/CVTK_DEBUG_FAST_src/MAIN_launch_cvtk_fast.bash**

- **./MAIN_launch_cvtk_fast.bash**
  It will launch the entire CVTK_FAST tests suite.
- **qsub gather_all.bash.pbs**
  It creates and gather the log file and store them in Log_Summary (a symbolic link to /home/datawork-croco/public/ftp/CVTK_FAST/Log_Summary)

4. How to add a test

For example I want to add a test in the REG family, using the cppkey CLIMATOLOGY cpp-key, with a1x4 domain tiling and no AGRIF nesting, how do I proceed?

- Go in $CVTKHOME/ Scripts_reg/Configure_Test_reg
- Create the file CLIM14 dedicated to this particular test
  - Have a special attention to the keyword
    - `TEST_NAME='CLIM14'` (**have to be the same as the file name**)
    - `LIST_KEY_PHYS='REGIONAL CLIMATOLOGY'`
    - `NBPROCS_X=1`
    - `NBPROCS_Y=4`
    - `LIST_KEY_NEST=''`
- Verify that you have all the input files necessary in:
  - **$CVTKWORK/../CROCO_FILES_BCK** for REG family
  - **[$CVTKHOME/TEST_CASES_CVTK** for VORT and KTEST families]
- Launch the test: **./mk_TestDIR.bash CLIM14**
- Check the compilation: **./mk_CHECKALL.bash**

You should see 3 lines by test (6 in case of AGRIF nesting with 2 grids)

- Get the log file: **./gather_recap.bash CLIM14**
- Check the log file: **vi Recap_CLIM14_*yyyymmdd*.git*revison_number***

5. Test files

You will see below an example of CLIM22 bash file needed to create a new test using CLIMATOLOGY but a 2x2 domain tiling.

```
#!/bin/bash

LIST_KEY0='BENGUELA_LR PSOURCE PSOURCE_NCFILE FRC_BRY CLIMATOLOGY TIDES
AGRIF AGRIF_2WAY BULK_FLUX MPI OPENMP'
# => Keys that will put to undef at the beginning of the rvtk scripts

TEST_NAME='CLIM22'
CONFIG_NAME='BENGUELA_VHR'
LIST_KEY_PHYS='REGIONAL CLIMATOLOGY'
LIST_KEY_PAR='OPENMP MPI'
FLAG_OPENMP=1 ; FLAG_MPI=1
NBPROCS_X=2
NBPROCS_Y=2
NBPROCS=$(( $NBPROCS_X * $NBPROCS_Y ))
LIST_KEY_NEST=''
KEY_DEBUG='RVTK_DEBUG'
CROCOIN='croco.in'
```