NCAR
UCAR

**Ben Foster <foster@ucar.edu>**

# Having problems with running several instancesof TIEGCM in parallel

6 messages

**Victor Solea** <VXS492@student.bham.ac.uk> Thu, Mar 5, 2015 at 2:14 PM
To: "foster@ucar.edu" <foster@ucar.edu>

Dear Mr. Foster,

I am a first-year PhD student at the University of Birmingham, working under the supervision of prof. Matthew Angling. As part of my work, I have set up and compiled (wthout MPI) TIE-GCM. It is all running properly. I use a Python script which launches in parallel (it runs with MPI) several TIEGCM instances with, each one with different input files. Most of the times, it works very well: it is starting up and behaving properly.

Occasionally, however, some of the instances mess up part of the data they write to the primary output, and the next invocation crashes. When reading the primary output file of the previous run, tey always fail at the zpmid values. The relevant section of the output file is:

```
Acquired source history file /home/victor/tiegcm-enkf/temp/nc_files/007/ensemble_run_
priout_2002-03-21_00:15:00_007.nc
  (disk file is /home/victor/tiegcm-enkf/temp/nc_files/007/ensemble_run_priout_2002-03-21_00:15:00_007.nc)
Reading source history from diskfile /home/victor/tiegcm-enkf/temp/nc_files/007/ensemble_run_
priout_2002-03-21_00:15:00_007.nc:
nc_rdhist: seeking   80   0  15 found ****   0  60  n=  1
nc_rdhist: seeking   80   0  15 found   80   0  15  n=  2

>>> nc_rdhist: zpmid_rd (as read from new history) is not equal to model zpmid
nlevp1= 29 zpmid_rd=
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
nlevp1= 29 zpmid   =
 -6.750 -6.250 -5.750 -5.250 -4.750 -4.250 -3.750 -3.250 -2.750 -2.250
 -1.750 -1.250 -0.750 -0.250  0.250  0.750  1.250  1.750  2.250  2.750
  3.250  3.750  4.250  4.750  5.250  5.750  6.250  6.750  7.250

>>>>>>>>>>>>>>>>>>>>>>>>>>>> MODEL SHUTDOWN <<<<<<<<<<<<<<<<<<<<<<<<<<<<
Shutdown: stop message: zpmid
```

(or zpmid_rd can be any nonsense value - it just happened to be zero in this example).

In other cases TIEGCM expands and opens the wrong startup file, such as in the example below, from another output file:

```
 LABEL = 'ensemble_run'
 START_YEAR = 2002
 START_DAY = 80
 CALENDAR_ADVANCE = 1
 SOURCE = '/home/victor/tiegcm-enkf/temp/nc_files/002/ensemble_run_priout_2002-03-21_00:15:00
_002abc.nc'
 SOURCE_START = 80,00,15
```

```
 START = 80,00,15
 STOP = 80,00,30
 STEP  = 60
 HIST = 0,0,15
 OUTPUT = '/home/victor/tiegcm-enkf/temp/nc_files/002/ensemble_run_priout_2002-03-21_00:30:00
_002abc.nc'
 MXHIST_PRIM = 10
 SECSTART = 80,00,30
 SECSTOP = 80,00,30
 SECHIST = 0,0,15
 SECOUT = '/home/victor/tiegcm-enkf/temp/nc_files/002/ensemble_run_secout_2002-03-21_00:30:00
_002abc.nc'
 MXHIST_SECH = 24
 SECFLDS = 'DEN','ZG','TEC','NE'
 TIDE = 0.,0.,0.,0.,0.,0.,0.,0.,0.
 TIDE2 = 0.,0.
 GSWM_MI_DI_NCFILE  = '$TGCMDATA/gswm_diurn_5.0d_99km.nc'
 GSWM_MI_SDI_NCFILE = '$TGCMDATA/gswm_semi_5.0d_99km.nc'
 POTENTIAL_MODEL = 'HEELIS'
 KP = 2
 F107 = 179.357595437
 F107A = 140
 BXIMF   = 0.
 BYIMF   = 0.
 BZIMF   = -5.
 AURORA = 1
 COLFAC = 1.5
/

 Reading input data...

 Completed successful read of namelist inputs.

 Will use the Heelis potential model
 Input: Expanded secout file /home/victor/tiegcm-enkf/temp/nc_files/002abc/ensemble_run_
secout_2002-03-21_00:30:00_009.nc to
   /home/victor/tiegcm-enkf/temp/nc_files/002abc/ensemble_run_secout_2002-03-21_00:30:00_009.nc
```

What I find most confusing is that these failures happen completely at random.

While I admit that my way of running TIE-GCM might be an unorthodox one, I would want to ask if it is a known issue, or whether my scenario has been tested at all (running several TIEGCM processes in parallel).

If you might be aware of a workaround or a solution to my problem, I would be very grateful.

Thank you very much.
Sincerely,
Victor Șolea

---

**Ben Foster** <foster@ucar.edu> Fri, Mar 6, 2015 at 10:57 AM
To: Victor Solea <VXS492@student.bham.ac.uk>, Joe McInerney <joemci@ucar.edu>

Victor,

I'm a bit confused. When you say "When reading the primary output file of the previous run", this implies to me that the current run is dependent on reading
the output of the previous run, i.e., a continuation. If this is the case, I don't see how they can be run in parallel, i.e., simultaneous executions, unless the python script is watching for output to appear before starting the next invocation.

I assume that all runs are writing output to unique file paths/names, although they should be able to read from the same SOURCE at the same time. And you said they are reading different input file names. Are you executing the same executable for each run? Are you making all the runs in one directory? If possible, it might be safer to make each run in a separate execdir with a unique job script. Of course you can still submit everything from your working dir. I know this is awkward, and may be difficult if you are chaining runs, where they have to read output from the previous run.

Looks like you are writing output history files to your home directory. If so, make sure you have enough disk space to handle all the files being written. It will be safer If you set execdir in the job script to a large temporary directory, then you can set OUTPUT,SECOUT to simple file names, and it will build, execute, and write output all in the big disk space. (leave your working directory, source code, etc. under your home).

Keep me in touch, although I may be late replying. I'm copying Joe McInerney. Are you running on your own local machine, and why do you need to run without MPI? Is this tiegcm1.95, or some other? Do you have a login on hao or other ncar machines? Also, I would be interested in seeing your python script, just for my own education ;-)

--Ben
[Quoted text hidden]
--
Ben Foster
National Center for Atmospheric Research (NCAR)
High Altitude Observatory (HAO)
303-497-1595

---

**Victor Solea** <VXS492@student.bham.ac.uk> Mon, Mar 9, 2015 at 5:56 AM
To: Ben Foster <foster@ucar.edu>

Hello,

Thank you very much for your quick answer!

Yes, I am doing continuation runs, and I am running TIEGCM1.95.

I have compiled without MPI and I am not using parallelization in TIEGCM because calling an MPI program from another MPI program is not recommended, it can lead to highly unstable and unpredictable behaviour. My Python script is parallel with MPI. The basic structure for each process is currently:

a. Create the namelist input file

b. Run TIEGCM (invocation of the job script with os.call() – this means that the script will continue only when the called program finishes)

c. Check the sanity of the output

d. Make some plots from the netCDF files.

The subfolder inside the home directory, where I am writing, is a symbolic link to a folder hierarchy on an external hard drive, so disk space is not an issue. I made that link there just for my convenience.

Our idea is to write our own ensemble Kalman filter data assimilation tool for ionospheric modelling. We are aware of the existence of DART and the work of J Anderson, T Matsuo and A Chartier, but my thesis advisor feels that it is better if we have our own: it offers more flexibility for us and we might later want to extend it and use it for other models as well. Right now it is still in very incipient stages, I am just building up and testing the software (haven't done any real science with it yet). We also have a cluster with 38 cores and 100 GB RAM. I am not yet at the stage where interprocess communication is relevant (i.e the actual ensemble tools). It is still a huge mess, because it is a work in progress, so I do not know how instructive it would be. There are still things which I need to optimize, add, document properly, etc. However, if you wish to see it the way it is at the current stage, I would be happy to send it to you. We intend to assimilate data from ionosondes and GNSS receiving stations.

The naming convention of the netCDF files at the moment is label – date and time of the run – process number, so there is no naming conflict. Each .inp and .out file is also suffixed with the process number, so there is again no naming conflict. All names are unique. Basically what I want to have right now is to be able to independently run several instances of TIEGCM, because I have not yet implemented the actual ensemble analysis.

If I understand correctly, you suggest that we should have, besides this, a different exec folder (with all of its contents) for each individual process? Also my supervisor suggested this when we were speculating what might be done.

Sincerely,

Victor.

**From:** Ben Foster [mailto:foster@ucar.edu]
**Sent:** 06 March 2015 17:57
**To:** Victor Solea; Joe McInerney

**Subject:** Re: Having problems with running several instancesof TIEGCM in parallel

[Quoted text hidden]

---

**Ben Foster** <foster@ucar.edu> Mon, Mar 9, 2015 at 9:45 AM
To: Victor Solea <VXS492@student.bham.ac.uk>, Joe McInerney <joemci@ucar.edu>

Victor,

I think you should start w/ the simplest most basic step and move on from there. That could mean starting with a non-MPI python script (or possible an MPI python script, but using only the root task), and make a single invocation in a single execdir.. Sounds like this will already work. Then add the ability to do a continuation upon successful completion of the first run, using the last history of the first run. Get this working consistently for multiple continuation runs using a single processor in a single execdir. I would even start with premade job and namelist scripts for each invocation. Later, you could make all job and namelist files from python before doing the first execution.

When the above is working, you can try adding a run using a new processor from the python script, a new execdir, and new job and namelist scripts, i.e., completely independent of the other run in the other execdir. Allow only one run (with continuations) in each execdir. Use conditionals in the python script to insure only one processor is working only a single run, w/ unique job and namelist files.
This is when you may see problems, but if the runs are completely independent, then the problem would be isolated to the python script, where the process/mpi tasks would be inadvertently confusing each other when making scripts and/or executions.

You say you have a cluster w/ 38 nodes. At first you might want to insure each run is executing on a different node, to eliminate potential memory conflict errors. Is there a batch queuing system? On the ncar machine yellowstone, we use Load Sharing Facility (LSF) for queuing batch jobs, and executing on dedicated nodes. This allows for multiple simultaneous jobs that are guaranteed independent. There is also an easy way to queue continuation jobs that depend on successful completion of the previous run. If you want, we could get you a login on the ncar machine so you could try this. That would help me help you, because I could see your setup, etc. You would also have help from our consulting team.
But I also understand you probably want to use your own machine for convenience in a long-term project, so this is just a suggestion.

Good that you are aware of Alex, Tomoko, et.al. work. There is also It-e Lee, who set up parallel runs of TIEGCM with DART. Do you know him? I think he was doing similar to what you need, wrt multiple parallel runs of the model. His email is ikkyu.itlee@gmail.com. If you write to him, you can say I referred you. Anyway, good luck, and keep me informed.

--Ben

[Quoted text hidden]

---

**Victor Solea** <VXS492@student.bham.ac.uk> Mon, Mar 9, 2015 at 10:17 AM

To: Ben Foster <foster@ucar.edu>, Joe McInerney <joemci@ucar.edu>

Hello Ben,

Thank you again for your time and patience. In the end, I guess my problem was that I had only one single execdir. And all TIEGCM instances that I was starting, were started from there. I would have never imagined that each instance needs to have its own execdir, and that not having it would be the cause of so much trouble.

That was the root of the problems, I guess, the corrupted nc files I encountered and all else. Now I have several independent execdirs, as you previously suggested, (named tiegcm-linux_000, tiegcm-linux_001, etc.) and at the moment it is running wonderfully. I will want to make some more runs, some more tests, to make sure all is working, and then I will feel ready to send you my source code in its current state – most likely in the next few days. As I wrote in my previous e-mail, it is still incipient and perhaps a bit chaotic, just the backbone of my project. I am however already making the namelist files in Python.

Our cluster runs on Rocks Cluster (Red Hat-based Linux) and we have the Sun Grid system installed, and we use their commands to request cores and submit and manage job queuing (qsub, qdel, etc). That way we ensure that the work load is balanced among all cores and managing tasks is easy. Of course, 38 cores is much smaller than your machine, but we are only a small research group and we're the only users of that cluster, so scheduling conflicts are much less likely to occur.

I did read one of Lee's papers on the topic, and I used it as a reference in the first progress report of my PhD, however I never communicated with him.

Sincerely,

Victor.

**From:** Ben Foster [mailto:foster@ucar.edu]
**Sent:** 09 March 2015 15:45

[Quoted text hidden]

[Quoted text hidden]

---

**Ben Foster** <foster@ucar.edu> Mon, Mar 9, 2015 at 10:29 AM
To: Victor Solea <VXS492@student.bham.ac.uk>
Cc: Joe McInerney <joemci@ucar.edu>

Oh good, I'm glad its working. Now that you have independent execdirs, I'll bet you could run TIEGCM w/ MPI, which would speed

up your runs significantly. If you try this, make sure the runs are executing on only a single node, independent from the other nodes, i.e., if you have 8-way nodes, set nproc <= 8 mpi tasks, and that you are executing on a unique dedicated node. Also make sure your python script is on its own node as well. But this is probably for later -- don't want to confuse things further now that its working!

--Ben

[Quoted text hidden]