JASMIN Workshop: Exercise 05: Batch run a script on LOTUS

Scenario

Having established (in exercise 4) that I can extract the total cloud cover ("TCC") variable from a single ERA-Interim file I now wish to extract that data from an entire month.

Objectives

Write script(s) to batch up separate processes that run CDO to extract the "TCC" variable from a series of ERA-Interim files. Each run of the script will loop through 4 x 6-hourly files for one day. I will run it 30 times, once for each day in September 2018. Each run will be sent to the LOTUS cluster.

JASMIN resources

- LOTUS batch processing cluster
- Space to store the output file: `/group_workspaces/jasmin2/workshop/users/$USER/ex05`
- Access to the CDO (Climate Data Operators) tool
- Read-access to the ERA-Interim data set in the CEDA archive - requires a CEDA account

Local resources

- SSH client (to login to JASMIN)

Instructions

1. Start ssh-agent session and add JASMIN private key
2. SSH to a scientific analysis server
3. Write an "`extract-era-data.sh`" wrapper script that calls the CDO extraction command
4. Write a script, called "`submit-all.sh`", to loop over dates from **01**/09/2018 to **02**/09/2018 and submit the "`extract-era-data.sh`" script to LOTUS for each day
5. Run the "`submit-all.sh`" script
6. Examine which jobs are in the queue
7. Examine the standard output and standard error files
8. Modify "`submit-all.sh`" so that it will run for all 30 days in September 2018
9. Re-run the "`submit-all.sh`" script
10. Examine which jobs are in the queue
11. Kill one of the jobs - just to see how it is done

Review

This exercise demonstrates how to:

- Create a script that takes an argument to process a single component (day) of an overall task.
- Create a wrapper script that loops through all the components that need to be processed.
- Submit each component as a LOTUS job using the "`bsub`" command.
- Define the command-line arguments for the "`bsub`" command.
- Use other LSF commands, such as "`bjobs`" (to monitor progress) and "`bkill`" (to kill jobs).

Alternative approaches and best practice

- Build up in stages before running your full workflow on LOTUS
- Write the output to a "scratch" directory
- Specify the memory requirements for your job
- *Have any files been accidentally left on the system? (E.g. in `/tmp/`)*

Cheat sheet for Exercise 05: Batch run a script on LOTUS

1. Start ssh-agent session and add JASMIN private key (skip if already done)

```
eval $(ssh-agent -s)
ssh-add ~/.ssh/id_rsa_jasmin
```

2. SSH to a scientific analysis server

```
ssh -A <username>@jasmin-login1.ceda.ac.uk
ssh jasmin-sci5 # Could use sci[123456]
```

3. Write an "`extract-era-data.sh`" wrapper script that calls the CDO extraction command, that:
    a. Takes a date string ("`YYYYMMDD`") as a command-line argument
    b. Locates the 4 x 6-hourly input file paths for the date provided
    c. Activates environment containing the CDO tool
    d. For each 6-hourly file:
        i. Defines the output file path
        ii. Run the CDO tool to extract the "`TCC`" variable from the input file to the output file
    e. If you are stuck, you can use the script located at:
    `/group_workspaces/jasmin2/workshop/exercises/ex05/code/extract-era-data.sh`
    [Source:
    https://github.com/cedadev/jasmin-workshop/blob/master/exercises/ex05/code/extract-era-data.sh]

4. Write a script, called "`submit-all.sh`", to loop over dates from 01/09/2018 to 02/09/2018 and submit the "`extract-era-data.sh`" script to LOTUS for each day:

    a. You should define the following LOTUS directives:
        i. Standard output file - please ensure this is unique to each job by including the "`%J`" variable in the file name.
        ii. Standard error file - please ensure this is unique to each job by including the "`%J`" variable in the file name.
        iii. Queue name:
            1. We will use the main queue for quick serial jobs: "`short-serial`"
        iv. Job duration - to allocate a maximum run-time to the job, e.g.: "`00:05`" (5 mins)
        v. Estimated duration - to hint the actual run-time of the job, e.g.: "`00:01`" (1 min)
            1. Setting a low estimate will increase the likelihood of the job being scheduled to run quickly.

    b. The Help page on submitting LOTUS jobs is here:
    https://help.jasmin.ac.uk/article/113-submit-jobs

    c. And use the "`bsub`" command to submit each job.

    d. If you need some advice you can use the script at:
    `/group_workspaces/jasmin2/workshop/exercises/ex05/code/submit-all.sh`
    [Source:
    https://github.com/cedadev/jasmin-workshop/blob/master/exercises/ex05/code/submit-all.sh]

5. Run the "`submit-all.sh`" script
6. Examine which jobs are in the queue
    a. Type "`bjobs`" to review any running jobs.

7. Examine the standard output and standard error files.

8. If you are happy that the job is doing the right thing, now modify "`submit-all.sh`" so that it will run for all 30 days in September 2018.

9. Re-run the "`submit-all.sh`" script.

10. Examine which jobs are in the queue

11. Kill one of the jobs whilst it is still running - just to see how it is done:
    a. Use the "`bkill`" command:

```
bkill <job_id>
```

Alternative approaches and best practice

- Build up in stages before running your full workflow on LOTUS:
    - This is really good practice!
        - 1. Check your code - is it *really* doing what you think it is doing?
        - 2. Run locally (on a "sci" server) for one iteration.
        - 3. Run for one or two iterations on LOTUS.
        - 4. Check everything ran correctly on LOTUS.
        - 5. Submit your full batch of jobs to LOTUS.

- Write the output to a "`scratch`" directory.
    - There are two main scenarios in which you might write the output to a scratch directory:
        - 1. You only need to store the output file for temporary use (such as intermediate files in your workflow).
        - 2. You want to write outputs to scratch before moving them to a GWS.

    - The Help page (https://help.jasmin.ac.uk/article/176-storage#diskmount) tells us that there are two types of scratch space:
        - `/work/scratch` – supports parallel writes
        - `/work/scratch-nompiio` – does NOT support parallel writes
    - Since we do not need parallel write capability, we can use the "`nonmpiio`" version.

    - You need to set up a directory under "`/work/scratch-nompiio`" as your username:

```
MYSCRATCH=/work/scratch-nompiio/$USER
mkdir -p $MYSCRATCH
```

    - Then you would write output files/directories under your scratch space, e.g.:

```
OUTPUT_FILE=$MYSCRATCH/output.nc
...some_process... > $OUTPUT_FILE
```

    - When you have finished with the file, tidy up (good practice).

```
rm $OUTPUT_FILE
```

    - Do not leave data on the "scratch" areas when you have finished your workflow.
        - Please remove any temporary files/directories that you have created.
        - You cannot rely on the data persisting in the "scratch" areas.

- Specify the memory requirements of your job:
    - If your job has a significant memory footprint:
        - Run a single iteration on LOTUS and review the standard output file to examine the memory usage.
        - You can then reserve a memory allocation when you submit your subsequent jobs.
        - See help pages:
          https://help.jasmin.ac.uk/article/115-how-to-estimate-job-resources
          https://help.jasmin.ac.uk/article/112-how-to-allocate-resources#memcontrol

- *Have any files been accidentally left on the system? (E.g. in `/tmp/`)*
    - It is important to clean up any temporary files that you no longer need.
    - Please check whether the tools you use have left any files in "`/tmp/`".