STAT605: Data Science Computing Project

Homework 4: Large-scale distributed computing at CHTC

This exercise, an expanded version of Homework 2, is a full-scale search using the CHTC (via learn.chtc.wisc.edu) for an undiscovered, gravitationally lensed, high-redshift Lyman-break galaxy.

Note: This homework requires running about 2500 jobs of a few minutes to an hour each. This seems to take 2-3 hours when the CHTC is normally loaded. It will become heavily loaded near this homework deadline, so that the computation may take half a day or longer. Please plan accordingly. In particular, it may not be realistic to try to finish this homework on the day it is due.

1. Revise your `hw2.R` from Homework 2 to a new `hw4.R` that takes two command-line arguments: a template spectrum for which to search and a data directory in which to find spectra to compare to the template. If called from the command line *without* two arguments, it should display this message:

   ```
   usage: Rscript hw4.R <template spectrum> <data directory>
   ```

   An example usage is `Rscript hw4.R cB58_Lyman_break.fit data`.

   Your `hw4.R` should write an output file whose name is the data directory name followed by `.csv` and whose line format is `distance,spectrumID,i` where

   - `distance`: your measure of the distance from this spectrum to the template
   - `spectrumID`: the spectrum ID, e.g., `spec-1353-53083-0579.fits`
   - `i`: the index in the spectrum at which your alignment with the template begins

   The output file should include one line per spectrum in the data directory and should be sorted by increasing distance. A sample line is `1032,spec-1353-53083-0579.fits,456`, which says "the object in `spec-1353-53083-0579.fits` has a distance `1032` from galaxy `cB58` when red-shifted by `456`."

   (If your `hw2.R` did not find `spec-1353-53083-0579.fits` among its top three of the 100 spectra searched in HW2, revise your `hw4.R` further so that it does find this spectrum from among those 100. Please ask early for help with this if you need it.)

2. Write a `hw4.sh` (modeled on `myscript.sh` from the `http://pages.stat.wisc.edu/~jgillett/605/CHTC/callingR` example) that unpacks R, unpacks the FITSio package, tells bash where R is, tells R where itself and its libraries are, unpacks the current `.tgz` file (like `3586.tgz`), and runs `hw4.R` on that directory (like `3586`).

3. There are about 2.5 million spectra stored in the directory `/home/groups/STAT605/boss/tgz` on the CHTC cluster. This directory contains 2459 `.tgz` files, each around 100 MB. Each `.tgz` file extracts to a directory containing about 1000 spectra. The template `cB58` is stored in `/home/groups/STAT605/boss/cB58_Lyman_break.fit`.

(a) Write an HTCondor submit script `hw4_1job.sub` that runs 1 job to processes the first `.tgz` file, `3586.tgz`. That job should transfer `3586.tgz` to a compute node, extract it to a directory `3586` containing 1000 `.fits` files, run `hw4.R` on the `3586` directory, and return the `3586.csv` file described in (1) above. A few notes:

- Do not copy files from `/home/groups/STAT605/boss/tgz` to your `/home/NetID` directory, as this would unnecessarily blow up our usage of `learn.chtc.wisc.edu` disk space. Instead, refer to `/home/groups/STAT605/boss/tgz` in your `.sub` script and let HTCondor transfer the `.tgz` file to a compute node.
- I suggest that, while coding and debugging, you limit your `hw4.R` to process only about 3 of the 1000 spectra.
- If your computation is slow, so that processing 1000 spectra takes more than an hour, limit yourself to fewer than 1000 spectra per data directory. That is, ignore some of the spectra rather than run a very long job.

(b) After your `hw4_1job.sub` runs correctly, note its `Cpus`, `Disk (KB)`, and `Memory (MB)` use from the bottom of its `.log` file. Include these requirements (after increasing the Disk and Memory by a little) in your `hw4_1job.sub` script. Run it again, being sure to process all 1000 spectra.

4. Write a `hw4_5jobs.sub` script that runs 5 parallel jobs to process the first five `.tgz` files (`3586.tgz`, `3587.tgz`, `3588.tgz`, `3589.tgz`, and `3590.tgz`), one per job.

   - After your `hw4_5jobs.sub` runs correctly, note the `Cpus`, `Disk (KB)`, and `Memory (MB)` use from the bottom of the five `.log` files. Include these requirements (after increasing the `Disk` and `Memory` by a little) in your `hw4_5jobs.sub` script. Run it again, being sure to process all 1000 spectra in each job.
   - Write a `hw4merge.sh` script that merges your five `.csv` files into one sorted by distance and writes the best 100 spectra to `hw4best100.csv`.

5. Write a submit script `hw4.sub` that runs 2459 parallel jobs to process all 2459 `.tgz` files, one per job. Run your `hw4merge.sh` again to merge your 2459 `.csv` files into one, and write the best 100 spectra to `hw4best100.csv`.

   Regarding this big run:

   - Monitor your jobs with `condor_q`. To stop all your jobs, run `condor_rm <NetID>` (for me it's `condor_rm jgillett`).
   - No job should run longer than one hour. Kill any job that runs longer and redesign it. Remember you may limit yourself to fewer than 1000 spectra per job, ignoring some, to get job times under an hour.
   - Do not use more than 2 GB of data on any CHTC computer (i.e., no more than 2 GB per job).
   - Do not launch a large number of jobs via an untested script. Start with 1 job, then 5, and only then 2459, as described above.
   - Ask for help if you have trouble managing your jobs.

6. Revise your `hw2.Rmd` to a new file `hw4.Rmd` (you may do this on the CHTC or on your local machine):
    (a) Include your name and `NetID@wisc.edu` email address.
    (b) Include a leading summary paragraph describing what you did and mentioning any difficulties you encountered.
    (c) Your `hw4.Rmd` should read your `hw4best100.csv` file and make ten graphs, showing `cB58` aligned with each of your top ten spectra from your search in `hw4.sub`. Include a legend with each graph identifying `cB58` and the other spectrum. Reorder your graphs so that the best match (according to your eyes rather than your measure) is at the top.
    (d) Knit your `hw4.Rmd` to make `hw4.html`.

What to submit

Make a directory `NetID/lyman`. Copy only these files there:

1. `hw4.R`
2. `hw4.sh`
3. `hw4_1job.sub` and a corresponding `hw4_1job.log`
4. `hw4_5jobs.sub` and a corresponding `hw4_5jobs.log` (you can make this via `cat *.log > hw4_5jobs.log` in your log file directory)
5. `hw4merge.sh`
6. `hw4.sub` and a corresponding `hw4.log` (you can make this via `cat *.log > hw4jobs.log` in your log file directory)
7. `hw4best100.csv`
8. `hw4.Rmd`
9. `hw4.html`

Include any supporting code files used by your scripts, but do not include the data.

From the parent directory of NetID, run `tar cvf NetID.tar NetID`, Upload `NetID.tar` to Canvas under the HW4 assignment. (I recommend downloading your submitted file from Canvas into a temporary directory and confirming that it is correct.)

Getting Help

Here are ways to get help:

- Ask the instructors or TA in class or office hours.
- Ask your peers questions, but do not share code with peers.
- Check the `HTCondor` manual: `http://research.cs.wisc.edu/htcondor/manual`.
- Ask a question of the CHTC Research Computing Facilitators via email: `https://chtc.cs.wisc.edu/uw-research-computing/get-help` or attend their office hours.