Homework 5: Large-scale distributed computing at CHTC

Due November 13, 11:59pm

This homework will give you a chance to practice using the CHTC Cluster to run large-scale distributed computing jobs, similar to those you ran on the Statistics Department cluster previously.

Note: This is probably the most challenging homework assignment of the semester, and there are several things that can go wrong when you are first starting out. **Please start early** so that we can identify and fix any issues (e.g., user permissions) well ahead of time! Furthermore, the CHTC is a shared resource, and if everyone in the class is trying to finish their homework on the due date, everyone's computations will be slower. That's just another reason to start early!

1 Warm-up: Counting Words on CHTC

In preparation for a large-scale version of our Lyman-break galaxy search from HW2 (see problem 2 below), this problem will walk you through how to submit jobs using HTCondor in a simple setting. We will count how many times each word appears in the works of Shakespeare.

Two things to note:

- The following design is not required. You may use a different design, provided you include a parallel step.
- We are using a small data set in this problem. In fact, the data set we are using is small enough that the parallel computation actually slows us down![1] Working in parallel will really shine when we are in the situation where we have more data than fit in memory on one machine. We're starting out with this small data set to keep things simple.

Write an HTCondor `DAGMan` script, `words.dag`, to make a sorted list of the words and their counts in the works of Shakespeare using the following steps:

[1] Look up *Amdahl's Law* or read this short article from Lawrence Livermore National Labs to learn more: `https://hpc.llnl.gov/tutorials/introduction-parallel-computing/limits-and-costs-parallel-programming`

1. Write a PRE `.sh` script that does the following:
   (a) Download `shakespeare.tar` by calling
   `wget http://pages.stat.wisc.edu/~jgillett/605/CHTC/wordCounting/shakespeare.tar`
   and extract `shakespeare.tar` (it will extract as a directory called `shakespeare`).
   (b) Write all the plays to one large file. **Hint:** use `cat` with a few appropriately-chosen wildcard characters.
   (c) Break the large file into 5 smaller files. **Hint:** try the program `split`. See `man split`, or refer to the standard deviation program from lecture for an example.

2. Write an HTCondor `.sub` script to process the 5 files in parallel by calling a `.sh` script on each file:
   (a) Your script should reformat the current file so it has one word per line. **Hint:** try using `tr`. See `man tr` to find how to replace one or more spaces with a newline. Alternatively, you can use `sed`. To replace every instance of `regexp` with `replacement`, use

   ```
   sed -e 's/regexp/replacement/g'
   ```

   (b) Sort the file of words (see `man sort`).

3. Write a POST `.sh` script:
   (a) Merge the sorted small files into one large sorted file. **Hint:** use `sort` again, but if you just run `sort`, you'll have wasted your earlier parallel sorting of the small files. Instead, read the sort manual page to see how to merge several files that have already been sorted[2] **Note:** sort can, under some circumstances, produce output that looks unsorted because it relies on your computers locale settings. You can ensure familiar English sort order by running `export LC_ALL=C` in your script before running `sort`.
   (b) Convert the large sorted file to one called `countsOfWords.txt`, whose lines have the form `count word`. **Hint:** have a look at the program `uniq`.

2 Lyman-Break Galaxies, Revisited

This exercise is an expanded version of Homework 2. We will perform a full-scale search of a much larger collection of galaxies 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, each of which takes a few minutes to an hour to run. This seems to take 2-3 hours when the CHTC is normally loaded. It will become heavily loaded near this homework deadline, so that running the

[2] For more detail, refer to the Wikipedia page for k-way merge: `https://en.wikipedia.org/wiki/K-way_merge_algorithm`.

computation may take closer to half a day or even 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 `hw5.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 hw5.R <template spectrum> <data directory>
   ```

   An example actual usage is `Rscript hw5.R cB58_Lyman_break.fit data`.

   Your `hw5.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

   Your 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 `hw5.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. 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 `hw5_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 `hw5.R` on the `3586` directory, and return the `3586.csv` file described in step (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`. Instead, refer to `/home/groups/STAT605/boss/tgz` in your `.sub` script and let HTCondor transfer the `.tgz` file to a compute node for you.
   - We suggest that, while coding and debugging, you should limit your `hw5.R` to process only about 3 of the 1000 spectra to save time.
   - 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 at CHTC.

(b) After your hw5_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, just to be safe) in your hw5_1job.sub script. Run it again, being sure to process all 1000 spectra.

3. Write a submission script hw5_5jobs.sub 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 hw5_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 hw5_5jobs.sub script. Run it again, being sure to process all 1000 spectra in each job.
   - Write a hw5merge.sh script that merges your five .csv files into one sorted by distance and writes the best 100 spectra to hw5best100.csv.

4. Write a submit script hw5.sub that runs 2459 parallel jobs to process all 2459 .tgz files, one per job. Run your hw5merge.sh again to merge your 2459 .csv files into one, and write the best 100 spectra to hw5best100.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 about an hour. Kill any job that runs longer and try to figure out what is slowing it down. 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 1 GB of data on any CHTC computer (i.e., no more than 1GB 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.

5. Revise your hw2.Rmd to a new file hw5.Rmd (you may do this on the CHTC or on your local machine, whichever you prefer):

   (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 hw5.Rmd should read your hw5best100.csv file and make ten graphs, showing cB58 aligned with each of your top ten spectra from your search in hw5.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 hw5.Rmd to make hw5.html.

What to submit

On your VM, create a directory `NetID_hw4`, where `NetID` is your NetID. Place in that directory a plain-text file `README` whose first line is of the form `NetID,FamilyName,GivenName`, where `NetID` is your NetID, `FamilyName` is your family name, and so on. If you discussed any problems with any other students, please list them below your information in this file. So, for example, if you worked with George Box (NetID: `gepbox`, you would have a second line in your `README` reading `gepbox,Box,George`.

Create a subdirectory `NetID/shakespeare`, where `NetID` is your NetID. Copy the following files into the `shakespeare` subdirectory:

- your `words.dag` file and any other supporting code you wrote
- your `.sh` files
- your `countsOfWords.txt` file
- Please do not include the input data (i.e., the Shakespeare text files extracted from `shakespeare.tar`).

Create a subdirectory `NetID/lyman`, where `NetID` is your NetID. Copy only these files there:

1. `hw5_1job.sub` and a corresponding `hw5_1job.log`
2. `hw5_5jobs.sub` and a corresponding `hw5_5jobs.log`
3. `hw5merge.sh`
4. `hw5.sub` and a corresponding `hw5.log` (please include any supporting code files used by your scripts, but please do not include the data.
5. `hw5best100.csv`
6. `hw5.Rmd`
7. `hw5.html`

From the parent directory of NetID, run `tar cvf NetID.tar NetID`, where once again `NetID` is your NetID. Upload `NetID.tar` to Canvas under the HW5 assignment. Refer to previous homework handouts for instructions on how to verify that your uploaded `.tar` file is correct.

Getting Help

As mentioned above, this is an especially challenging homework. If you run into trouble, there are a few ways to get help:

- Ask the instructors or TA in office hours or via the Canvas discussion board.
- Ask your peers questions (but please remember that you are not permitted to share code with your classmates).
- Check the `HTCondor` manual: `http://research.cs.wisc.edu/htcondor/manual`.
- Ask a question of the CHTC Research Computing Facilitators via email: `http://chtc.cs.wisc.edu/get-help.shtml` or attend their office hours.
- Make an appointment to meet with an instructor or your TA, if you are unable to attend discussion section or office hours.