COMP0235 Coursework challenge

Introduction

In the field of biochemistry there have recently been large advances in predicting protein structure. A group of biochemist researchers at UCL Medical school would like to make predictions of possible 3D structures of proteins in the Human Genome. The have written a small data analysis pipeline, comprising 4 steps that is capable of making such predictions. A prediction for each protein in their pipeline takes 13 minutes and they have 21,856 proteins they wish to analyse. They calculate that on their computer with just one CPU core, it would take nearly 200 days to make all the predictions they want.

They have approached you to help turn their data analysis program in to a distributed analysis system which can run their predictions in a timely fashion. They can provide you with their python scripts. Their pipeline makes use of two machine learning predictors; S4Pred and HHSearch. To test the system the researchers give you a list of 6,000 proteins they are most interested in and they they would like you build a system that can analyse these 6,000 entries as a proof of concept.

The researcher's pipeline and data

The researchers provide you with two files. One file containing all the proteins in the mouse proteome **UP000000589_10090.fasta.gz** and a second file with their pipeline python code **code.tar.gz**.

The code file includes 3 items. Two python scripts and a text file with 6,000 protein IDs. The python script, **pipeline.script.py** runs the data analysis pipeline they have written. The python script **results_parser.py** is a short piece of code that their pipeline script requires. The **experiment_ids.txt** contains the list of protein IDs for the subset of 6,000 proteins they would like you to run predictions for.

If you open **pipeline.script.py** you can follow the logic of their analysis. At the start of the process the script reads in all the proteins from a fasta file. Then for each file it runs the following 4 steps.

1. Run the s4pred ML tool
2. Rewrite the input sequence to include the s3pred predictions
3. Run the HHSearch ML tool
4. Parse the HHSearch output

The 4th step outputs the results that the researchers want to capture.

Coursework task

In this coursework you are required to build a distributed pipeline across your cloud machines that will run the 4 steps in the `pipeline_script.py`. It should accomplish this in distributed fashion across your mini-cluster of 6 machines (one host and 5 workers). The host machine is a low resource machine and not intended to run calculations. And it must be accomplished using a message passing framework such as RabbitMQ, Celery, Kafka, etc. You must write the code to implement that pipeline yourself. You are free to accomplish this as you see fit. Your solution should include the following features

1. Must use appropriate automated and distributed configuration system (ansible)
2. Make use of an appropriate data storage for the complete mouse proteome data contained in the file provided. The storage mechanism should be able to return appropriate records from a list of arbitrary protein IDs
3. Make use of appropriate monitoring and logging of your mini-cluster and your data analysis pipeline
4. Must collate the results calculated on the client machines in a central location

You need to collate the following information from step 4 for the researchers, preserving these file formats:

1. A csv file that contains a list of proteins and the identity of the best hit calculated by HHSearch. You can find an example such file in **coursework_example_output/example_hits_output.csv**
2. A file containing the mean Standard Deviation and mean Geometric means for all 6,000 HHSearch runs you calculate (i.e. capture the STD and Gmean values for each pipeline run and take the average across 6,000 runs). You can find an example such file in **coursework_example_output/example_profile_output.csv**

You should not write any terraform code to set up your machines. You must use the terraform code provided in the **build_cluster/** directory to create the 6 machines you will use for the coursework. Make sure to complete the variables.tf file with your details. In general we do not want you to change this code. However the one exception is tags (instance labels). You are allowed to add the tags you need to the terraform code we provided to allow access to IP ports, web applications and so forth.

Challenges/hints

1. On your machines we estimate it should take about one to two days to run all the calculations.
2. The host instance is too small to run the calculations. Use this for running ansible, any centralised or control processes
3. You need to be able to understand how to install and run the s4pred and hhsearch programs

4. You need to be able to understand how to fetch the required datasets for s4pred and hhsearch
5. You will need to be able to understand the FASTA data format
6. You should ensure you can successfully run **pipeline.script.py**. This could be on either your own machine or on one of the cloud machines you have access to. In the directory `pipeline_example`. You can find an example input sequence `test.fa`. If you run the script successfully you should produce a number of intermediary files, example of these can be found in the directory. And a final output file **hhr_parse.out**. The files you produce should be equivalent (though some figures may have some minor differences)
7. At runtime the Load Average for a Client machine should generally not exceed 3 (num cpus minus 1)
8. Is your design scalable beyond this one coursework task?
9. What about the future, what if a user wanted to analysis a different set of proteins?

Deliverables

1. You must submit all your code as a git repository in the course gitlab instance, this can be found at https://gitlab.ds4eng.condenser.arc.ucl.ac.uk/
2. Your code repository must include a README with instructions on how to use your code to install AND run your data analysis system.
3. A working version of your system running on the 6 machines. That the marker can log in to and run one example fasta and get results. Each machine must include the lecturer_key.pub contents in the authorized_hosts
4. Assessment is via a short viva (time to be arranged). You will be expected to give a short (no more than 10mins) presentation which explains the design of your solution to the problem and gives a live demo of your system. You will then have to answer some questions to justify your design choices.

Pipeline Dependencies

Executables

1. S4Pred - https://github.com/psipred/s4pred
2. HHSuite - https://github.com/soedinglab/hh-suite
3. pipeline_script.py
4. results_parser.py

Datasets

1. pdb70 protein structure sequence dataset for HHSearch (this may take 4 hours to download) - **https://wwwuser.gwdg.de/~compbiol/data/hhsuite/databases/hhsuite_dbs/pdb70_from_mmcif_latest.tar.gz**
2. uniprot mouse proteome set -

3. **https://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/reference_proteomes/Eukaryota/UP000000589/UP000000589_10090.fasta.gz**
4. List of fasta IDs to analyse - **experiment_ids.txt**

Python dependencies

1. biopython
2. torch
3. numpy
4. scipy

Background Reading

1. https://en.wikipedia.org/wiki/Human_genome
2. https://en.wikipedia.org/wiki/Protein
3. https://en.wikipedia.org/wiki/Proteome
4. https://en.wikipedia.org/wiki/FASTA_format
5. https://en.wikipedia.org/wiki/Protein_Data_Bank_(file_format)