COL733: Fundamentals of Cloud Computing
Semester I, 2023-2024

Lab-1: Batch processing
3 August 2023

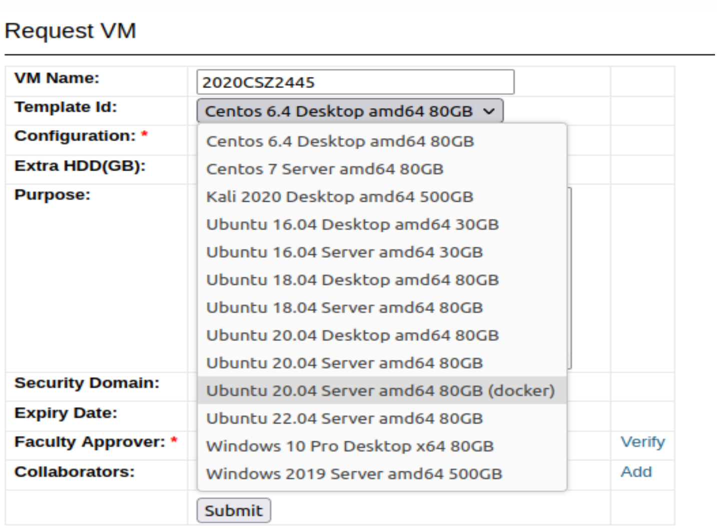
Submission Instructions

1. You can **only** use Python and Redis for this Lab. **Use of any other libraries** will lead to zero marks in the Lab.
2. You will submit the source code in **zip** format to Moodle (Lab 1). The naming convention of the zip file should be <Entry_Number>_<First_Name>.zip. Additionally, you will need to later submit a **pdf** for analysis questions on Gradescope.
3. The Lab would be **auto-graded**. Therefore, **follow** the same naming conventions described in the Deliverables section. Failing to adhere to these conventions will lead to zero marks in the Lab.
4. You should write the code **without** taking help from your peers or referring to online resources except for documentation. The results reported in the report should be **generated from Baadal-VM**. Not doing any of these will be considered a breach of the honor code, and the consequences would range from zero marks in the Lab to a disciplinary committee action.
5. You can use **Piazza** for any queries related to the Lab.

Setup Instructions

How to get your Virtual Machine?

- Go to BaadalVM website to request a VM https://baadal.iitd.ac.in/user/request_vm
- Use your entry number as the VM name. Choose `Ubuntu 20.04 Server amd64 80GB (docker)` in Template ID.

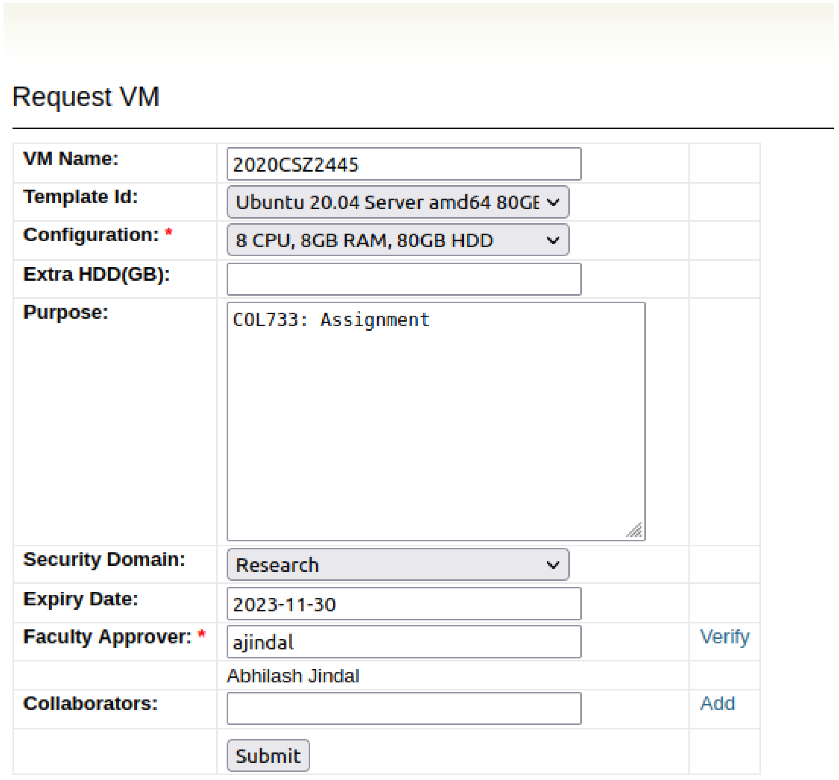


Request VM

VM Name:	2020CSZ2445	
Template Id:	Centos 6.4 Desktop amd64 80GB	
Configuration: *		
Extra HDD(GB):		
Purpose:		
Security Domain:		
Expiry Date:		
Faculty Approver: *		Verify
Collaborators:		Add
	Submit	

Centos 6.4 Desktop amd64 80GB
Centos 7 Server amd64 80GB
Kali 2020 Desktop amd64 500GB
Ubuntu 16.04 Desktop amd64 30GB
Ubuntu 16.04 Server amd64 30GB
Ubuntu 18.04 Desktop amd64 80GB
Ubuntu 18.04 Server amd64 80GB
Ubuntu 20.04 Desktop amd64 80GB
Ubuntu 20.04 Server amd64 80GB
Ubuntu 20.04 Server amd64 80GB (docker)
Ubuntu 22.04 Server amd64 80GB
Windows 10 Pro Desktop x64 80GB
Windows 2019 Server amd64 500GB

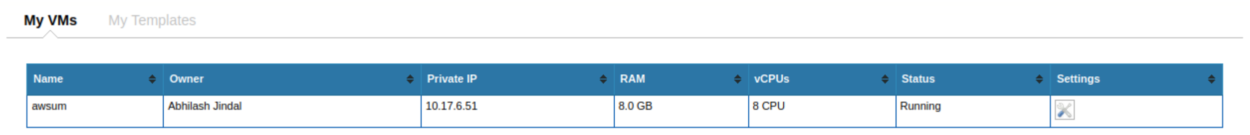
- Choose 8 CPU, 8GB RAM, 80GB HDD in configuration and 30 November 2023 as the VM expiry date. Add `ajindal` as faculty approver and submit.



Request VM

VM Name:	2020CSZ2445	
Template Id:	Ubuntu 20.04 Server amd64 80GE	
Configuration: *	8 CPU, 8GB RAM, 80GB HDD	
Extra HDD(GB):		
Purpose:	COL733: Assignment	
Security Domain:	Research	
Expiry Date:	2023-11-30	
Faculty Approver: *	ajindal	Verify
	Abhilash Jindal	
Collaborators:		Add
	Submit	

- Once the VM is created, you will be able to check the VM's Private IP by clicking `My VMs`.



My VMs | My Templates

Name	Owner	Private IP	RAM	vCPUs	Status	Settings
awsum	Abhilash Jindal	10.17.6.51	8.0 GB	8 CPU	Running	

Using your Virtual Machine

- If you're outside the IITD campus, you will first need to get VPN access. See here: VPN instructions.
- After verifying that you're able to ssh into a CSC machine and after receiving your VM IP. You may receive an email with a default username (baadalvm) and password (baadal) during VM creation.

```
$ ssh baadalvm@<YOUR_PRIVATE_IP>
```

- You can change your password after your first login by running:

```
$ passwd baadalvm
```

Note: Remember to note this password. If you forget your password, there may not be any way to recover it.

Redis Installation:

- Execute the following commands to install redis on your VM.

```
$ sudo apt-get update
$ sudo apt install redis
$ redis-cli --version
$ redis-server <PATH_TO_REDIS_CONFIG>
```

- To verify the successful installation of Redis. Check the service status using the following command.

```
$ sudo systemctl status redis
```

Dataset Description

The dataset is available at the link [1]. Each CSV file contains 7 attributes, following are a brief description of each attribute:

- ***tweet_id:*** A unique, anonymized ID for the Tweet. Referenced by response_tweet_id and in_response_to_tweet_id.
- ***author_id:*** A unique, anonymized user ID. @s in the dataset have been replaced with their associated anonymized user ID.
- ***inbound:*** Whether the tweet is "inbound" to a company doing customer support on Twitter. This feature is useful when re-organizing data for training conversational models.
- ***created_at:*** Date and time when the tweet was sent.
- ***text:*** Tweet content. Sensitive information like phone numbers and email addresses are replaced with mask values like __email__.
- ***response_tweet_id:*** IDs of tweets that are responses to this tweet, comma-separated.
- ***in_response_to_tweet_id:*** ID of the tweet this tweet is in response to, if any.

Problem Statement

The Hogwarts School of Witchcraft and Wizardry is hosting a challenge to count the number of words in magical "tweet" messages from their student support dataset [1]. Your task is to create an application that can handle the large amount of data, which is estimated to be in the range of GBs. Since a serial word count won't be sufficient, you need to design a scalable word count application that can handle the size of the dataset.

You are provided with the starter code for the challenge.

Deliverables

- ***Source code***: You need to provide the source code for the word counting application implemented using the python ***multiprocessing*** library. The source code should be in a .zip format and should be uploaded to moodle. A sample source code folder structure is shown below:

```
directory:  2020CSZ2445_Abhisek
            2020CSZ2445_Abhisek/client.py
            2020CSZ2445_Abhisek/base.py
            2020CSZ2445_Abhisek/constants.py
            2020CSZ2445_Abhisek/mrds.py
            2020CSZ2445_Abhisek/worker.py
            2020CSZ2445_Abhisek/__init__.py
            2020CSZ2445_Abhisek/requirements.txt
```

  When we unzip the submission then we should see the above files in the aforementioned structure.
  - Your word-count application should be named ***client.py*** and runnable by the following command. *Note:* All the relevant information necessary for the word count application is available in ***constants.py***

```
python3 client.py
```

  - ***We will change the constants.py file with appropriate values during evaluation. Therefore, do not change constants.py file.***

- ***Analysis:*** Answer the following questions on Gradescope (Lab 1-3: Analysis):
  - What is the best speedup achieved over a serial implementation (a regular Python program that counts words without using Redis)?
  - Given a fixed input size, measure how the efficiency of the word-count application varies with an increase in workers (in the range of [1, 32]) allocated to the application. Justify.
  - Given a fixed worker processes (= 8) allocated to the application, measure how the efficiency of the word-count application varies with input size. Justify.
  - Argue about the iso-efficiency of your implementation. Is the designed solution scalable? Justify.

Rubrics (25 marks)

1. 2 marks: Correctness of word-count application with single worker process.
2. 3 marks: Correctness of word-count application with multiple worker processes.
3. 10 marks: This has relative grading. The faster programs on multiple workers will receive higher marks.
4. 10 marks: Justifications and analysis as requested in the deliverables.

References

[1]: https://www.kaggle.com/thoughtvector/customer-support-on-twitter