LLAMA-2 Hands-on Assignment :

LLAMA-2 (7b or 13b model) concurrency problem in serving more than one request simultaneously.

Response time could be worse in concurrency mode compared to the sequential mode.

Challenge :

Create a prototype to demonstrate a solution to solve above concurrency problem with LLAMA-2

Specifications/Setup : (Candidate should run the prototype in this runtime setup)

Python based Techstack

LLAMA-2 from Hugging Face. Download local model to the server it is running in . No API to be used

LLAMA-2 13B (Use this model)

AWS EC2 : g5.2xlarge (1 GPU nvidia machine) running Ubuntu OS . Use this machine specifically

AWS Region : US-east virginia preferred.

Candidate should use their own personal AWS Account and buy EC2 runtime on above GPU machine

(Approx cost : $1.35 per hour for above EC2 instance. Tips : EC2 can be turned off when not in runtime mode to save cost.

Candidate is solely responsible for the expense associated with this assignment)

No Need : No need of Vector DB . Focus is on LLAMA-2 input/output and its performance

Hardcode the input text (i.e read from a text file or hardcode in python as a string)

Hardcode the input question (or take the question from command-line from the user)

This program/code should be run in stand-alone mode with input supplied through text file or input hardcoded.

Output should be printed on stdout or into a log file.

Integration : Only interaction of this program should be with Hugging face (for model download) and to other py libraries.

User interface is strictly command-line within the EC2 shell.

Expectations from this assignment :

Single Request Mode :

Input to LLAMA : Block of text holding 2K to 3K token count (about 2000+ word count)

Instruction/Prompt/Question to LLAMA : Ask a question that has an answer in the input text.

It could be an instruction to summarize the text provided in 50 words. Or it could be a specific question.

Response time from LLAMA should be within 5 to 10secs.

Define this module as a callable method/task for any thread to invoke

Muli-thread prototype

Create a simple multi thread prototype with input “N” to fork N no. of threads

Each thread should be assigned with the above task of interacting with LLAMA-2 .

Each thread will be performing the exact same task.

N=1 : Ensure output and response time are in-line with the above “Single Request Mode”

N=2 : Capture the output and response time

N=3 : Capture the output and response time.

success criteria of this assignment.

Candidate should tweak the logic/memory parameters to python and GPU/ parameters to LLAMA and achieve N=3 performance comparable to N=1

N=3 should be having a response time not more than **1.5x of N=1** response time. Python process running this program should not crash

Above performance outcome should be satisfying any length of input text *(within LLAMA-2 limits of input context window)* and any question supplied to LLAMA-2