

A GPU-Accelerated RAG-Based Telegram Assistant

for Supporting **Parallel Processing** (Any) Students

Guy Tel-Zur
Ben-Gurion University of the Negev
gtelzur@bgu.ac.il

November 16, 2025

The need

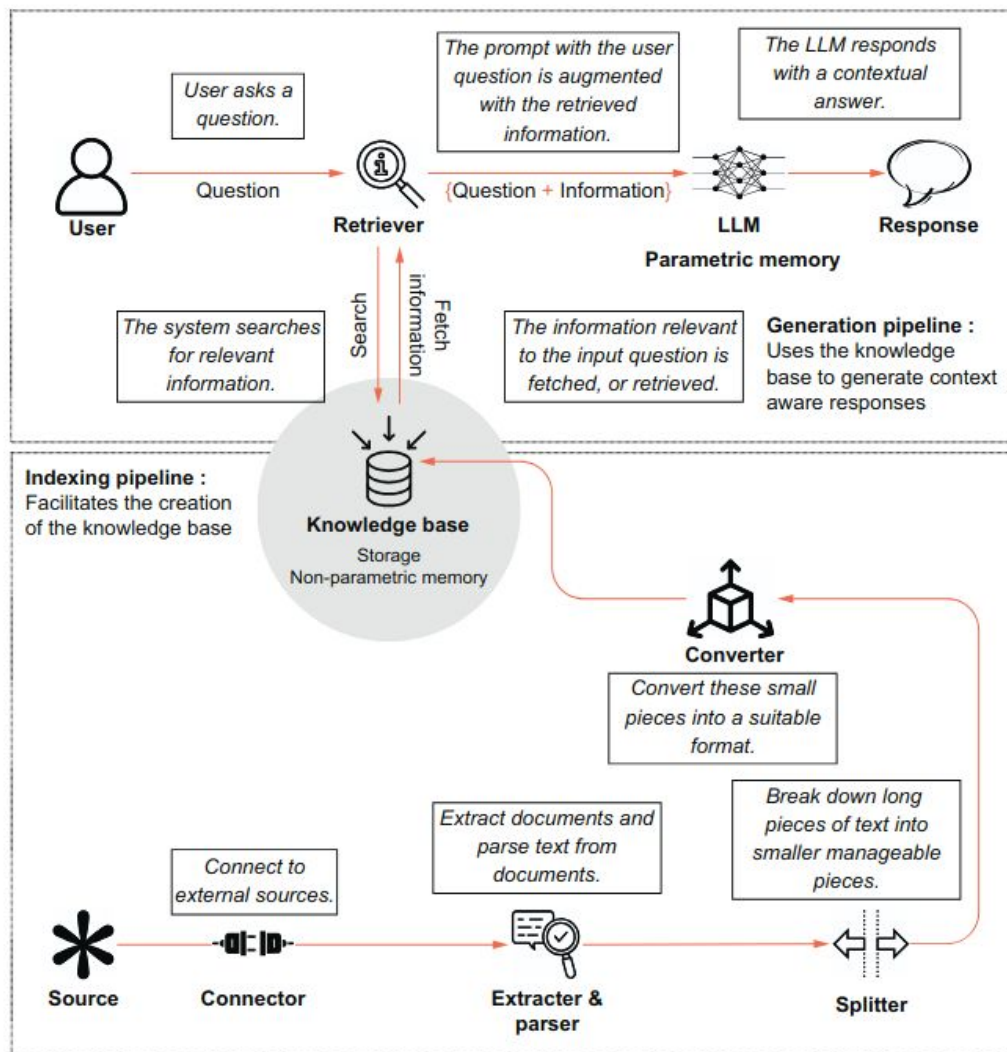
Introduction to Parallel Processing (BGU-ECE)



- Limited office hours: 1 hour per week
- This system availability: 24x7x365
- Smart agent trained on the specific course materials as opposed to commercial tools
- Privacy is kept if that is an issue
- Open source (no licensing, no fees), can operate from a departmental server
- Can be generalized to any kind of material (PDF, PPT, DOC, TXT, MD, ...)
- Can serve any course

What is RAG?

Credit: “A Simple Guide to Retrieval Augmented Generation” by Abhinav Kimothi
Manning Publications 2025



The solution

Telegram:

@ParallelProcessing



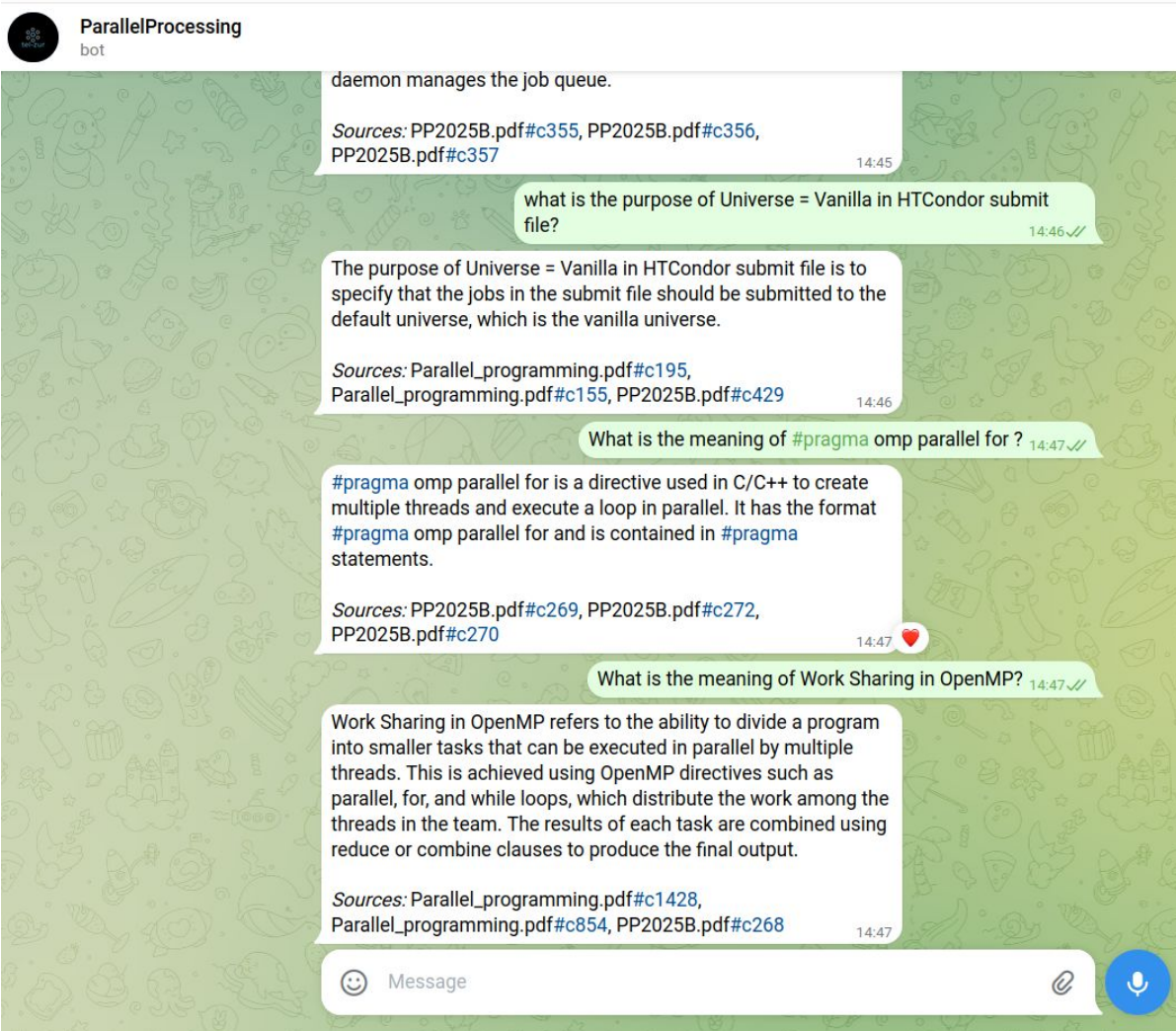
Bot Info



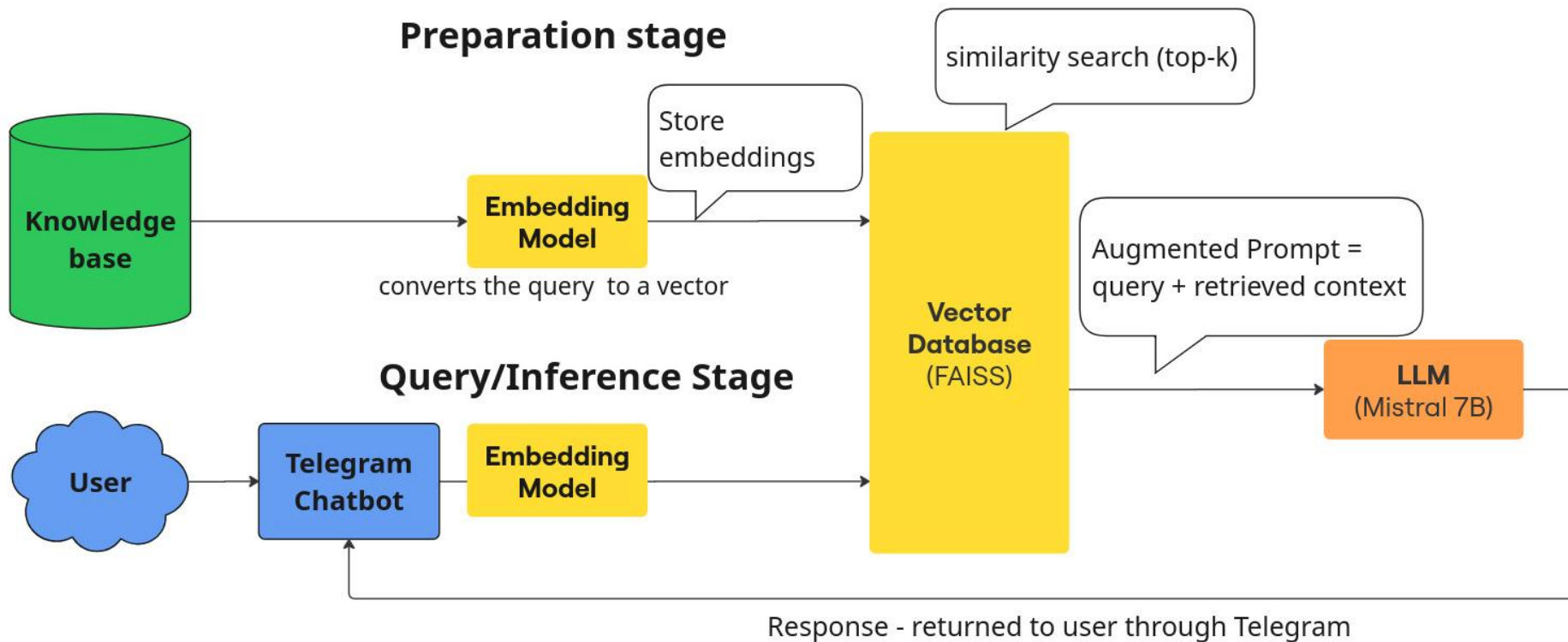
ParallelProcessing
bot



ParallelProcessingBot
Username



System Architecture



Performance

H/W

Asus TUF F17

- I9 CPU, 32GiB
- Nvidia, RTX 4060 GPU

S/W

- Ubuntu 24.04,
- Docker & docker compose
- Nvidia CUDA container toolkit
- mistral-7b-instruct-v0.1.Q4_K_M:
4-bit K-Quantization, Medium variant

Benchmark

Tokens per Second (TPS): ~16

TBFB: ~0.1s

- A good response time.
- Further optimization steps discussed in the paper.

Future Improvements

- Fine tune parameters
- Try [Docling](#)
- Try [RAGflow](#)



Reproducibility

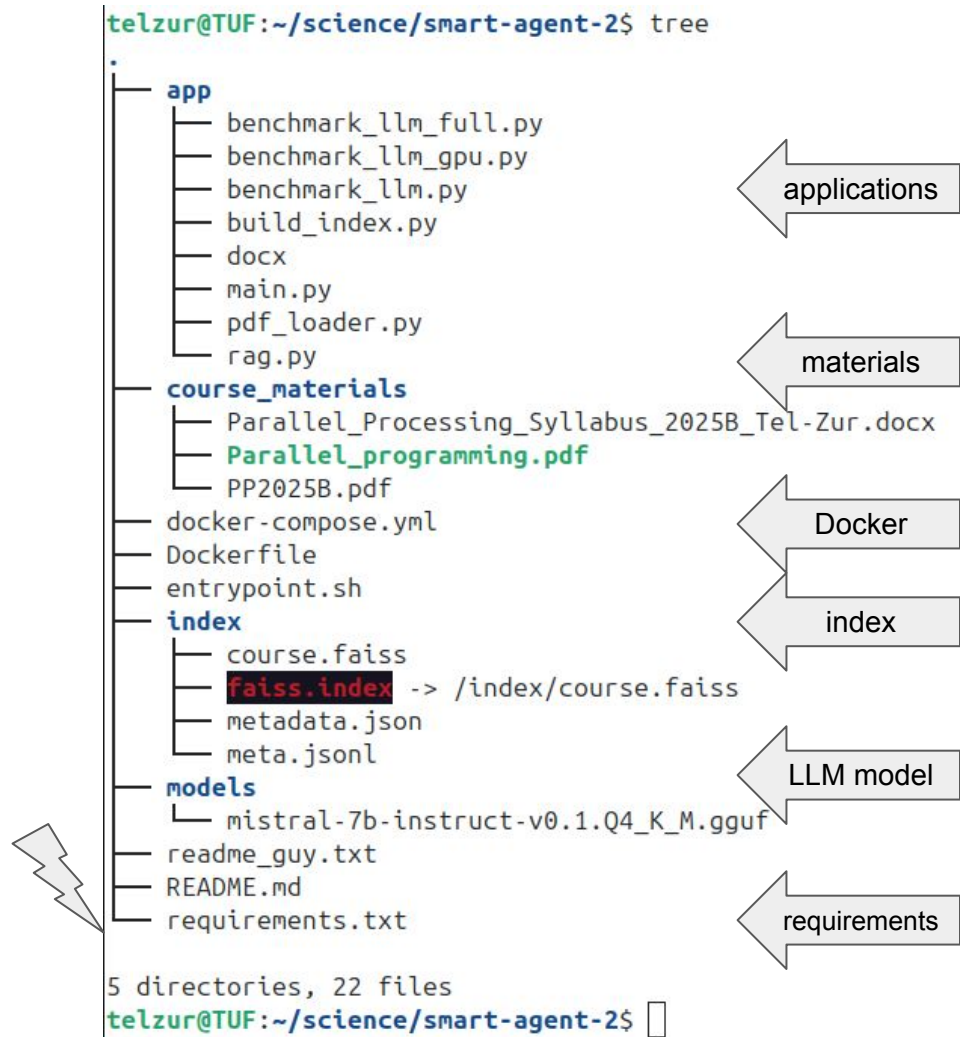
Currently there are 2 versions:

- A local implementation.
- A Container implementation:

Based on Docker → Portable

This implementation is available at:

<https://tel-zur.net/papers/EduHPC25>



The paper

The paper is available at: <https://arxiv.org/abs/2509.11947>

A GPU-Accelerated RAG-Based Telegram Assistant for Supporting Parallel Processing Students

Guy Tel-Zur
Ben-Gurion University of the Negev
Beer Sheva, Israel

Abstract

This project addresses a critical pedagogical need: offering students continuous, on-demand academic assistance beyond conventional reception hours. I present a domain-specific Retrieval-Augmented Generation (RAG) system powered by a quantized Mistral-7B Instruct model [4] and deployed as a Telegram bot [9]. The assistant enhances learning by delivering real-time, personalized responses aligned with the "Introduction to Parallel Processing" course materials [1]. GPU acceleration significantly improves inference latency, enabling practical deployment on consumer hardware. This approach demonstrates how consumer GPUs can enable affordable, private, and effective AI tutoring for HPC education.

ACM Reference Format:

Guy Tel-Zur. A GPU-Accelerated RAG-Based Telegram Assistant for Supporting Parallel Processing Students. In *Proceedings of Workshop on Education for High-Performance Computing (EduHPC25)*. ACM, New York, NY, USA, 9 pages.

1 Introduction

Large Language Models (LLMs) have revolutionized human-computer interaction. However, deploying these systems in a privacy-preserving and cost-effective way remains a challenge. This paper describes the development of a local Retrieval-Augmented Generation (RAG) assistant using a quantized version of Mistral-7B model. The system is deployed as a Telegram bot to support students enrolled in the "Introduction to Parallel Processing" course [8]. It runs entirely on a local machine with a consumer GPU, ensuring both privacy and responsiveness. The term RAG was coined by Lewis, P. et al [3]. In the words of Vinton Cerf, "RAG systems connect LLMs to external, verifiable knowledge bases, allowing them to ground their responses in current, curated information rather than relying solely on their training data. This dramatically reduces the production of factual errors and hallucinations. Some research indicates improvements of 42% – 68% and even higher in specific domains, such as medical, AI when paired with trusted sources" [1].

2 Project description

I have been teaching Parallel Processing for more than 20 years. In 2014 I described the course called "An Introduction to Parallel Processing" in the EduHPC 2014 workshop [7]. The current course

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise or to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
EduHPC25, St Louis, MO
© Copyright held by the owner(s). Publication rights licensed to ACM.

web site is available at [8]. Every lecturer is committed conduct reception hours, on a weekly basis, for answering students questions, sometimes this time slot isn't enough especially toward the final examinations dates where the students are more focused on learning toward the exams and have more questions to ponder. In addition, some of the student may feel shy and will avoid asking questions during these hours. Today when AI is becoming so advanced it is possible to provide a solution to these needs where a smart agent can be available continuously $24 \times 7 \times 365$. When this project idea triggered my imagination, a few months ago, developing a smart agent was still a complicated task. However, with the accelerating pace of AI it is now becoming a very popular topic and there are many alternative ways to implement smart agents. However, the project that I describe here still has a few unique features:

- This project is built using only open-source tool. This means that there are no licensing issues, no payments, and everything is running on a stand alone computer so that privacy is kept if that is important to the users.
- This smart agent uses a Telegram interface which is available from any platform (desktop, mobile phone, tablet, etc').
- The smart bot can run on a commodity computer preferably with a Graphics Processing Unit (GPU). In this project I use an ASUS TUF F17 laptop with 32GB RAM and an Nvidia GeForce RTX 4060 GPU, and the response time was found to be reasonable. In addition, the smart agent project simplicity allows it to be implemented as an educational assignment in AI related courses in addition to the main goal of helping students.

3 Deployment

The course slides were merged into a single PDF file and together with the electronic version of the course textbook [10] served as the knowledge base for the smart-agent. A document preparation pipeline is next in order to build a searchable knowledge base for the RAG system.

A schematic chart showing all the project building blocks is shown in Figure 1. A screen capture of the Telegram window showing a dialog between the user and the agent is shown in Figure 3.

The Embeddings Generator converts each chunk of the course documents into numerical vector embeddings. This stage uses the open-source *all-MiniLM-L6-v2* embedding model locally (via sentence-transformers), ensuring privacy and low cost. The Vector Database (FAISS) stores the embeddings and their metadata for fast similarity search. It enables retrieval of the most relevant chunks based on semantic similarity to user queries. The Chunking and metadata indexing splits documents into manageable pieces (e.g. 512–1024 tokens) with overlap and keeps source references for later citation. The Retrieval-Augmented Generation (RAG) pipeline



A demo video

<https://youtu.be/AqBvKRingoQ>



Discussion & Collaboration

I will be happy to collaborate!

Email: gtelzur@bgu.ac.il

Website: <https://tel-zur.net>

Linktr.ee: <https://linktr.ee/telzur>

YouTube channel: [@tel-zur_computing](#)

Thank you!

