Cairo University
Faculty of Engineering

Department of Computer
Engineering

Phase 2

ADB

Submitted to:

Eng / Abdelrahman Kaseb

Name	ID
Ahmed Sabry	9202119
Eman Mohamed	9202351
Zeinab Moawad Fayez Hassan	9202611
Menatalh Hossamalden	9203998

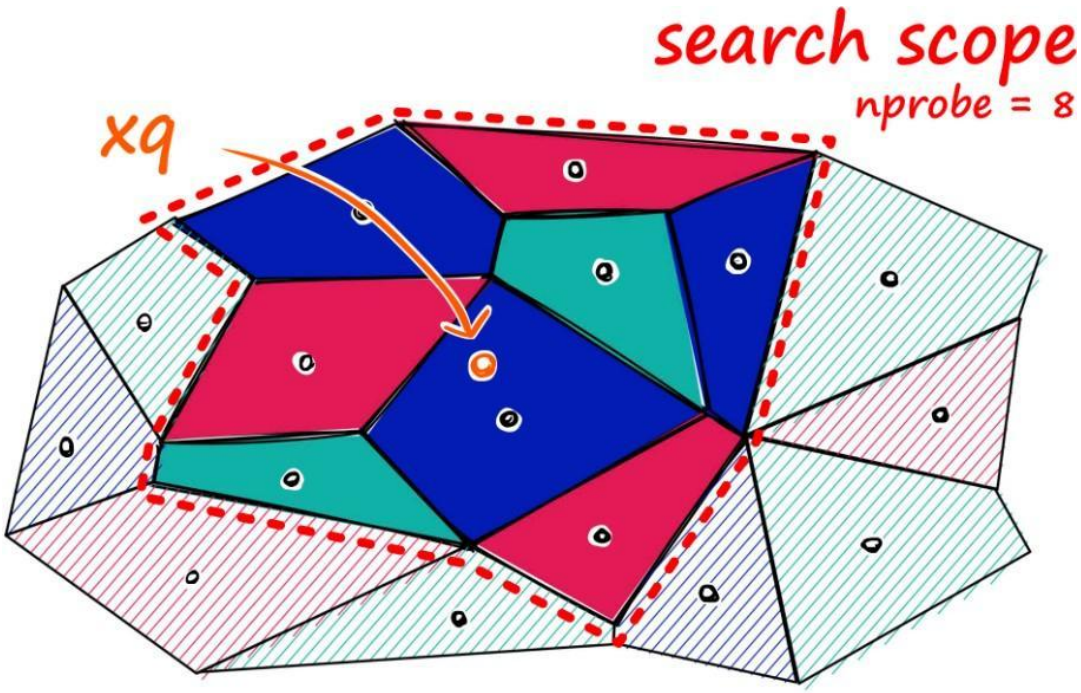
Indexer at size 10K - 1M: IVF

Algorithm:

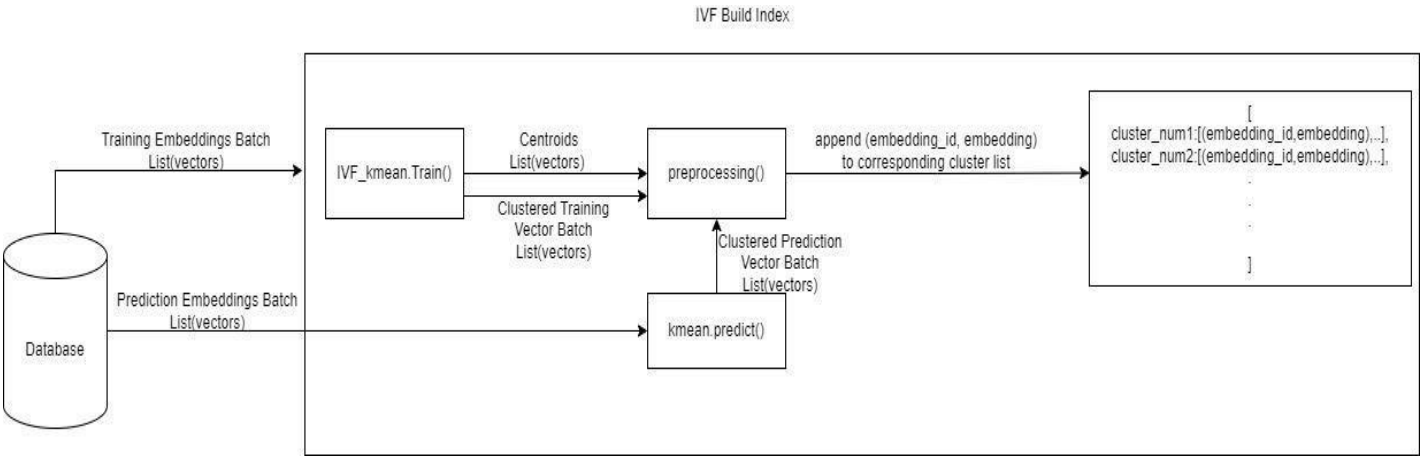
Inverted File Indexer:

- Training: convert space into subspaces using clustering methods (kmeans) And assign each subspace representative “centroids”
- Searching: get nearest nprobe centroids to query vector and apply flat index in each subspace to get top_k nearest vectors to query submitted.

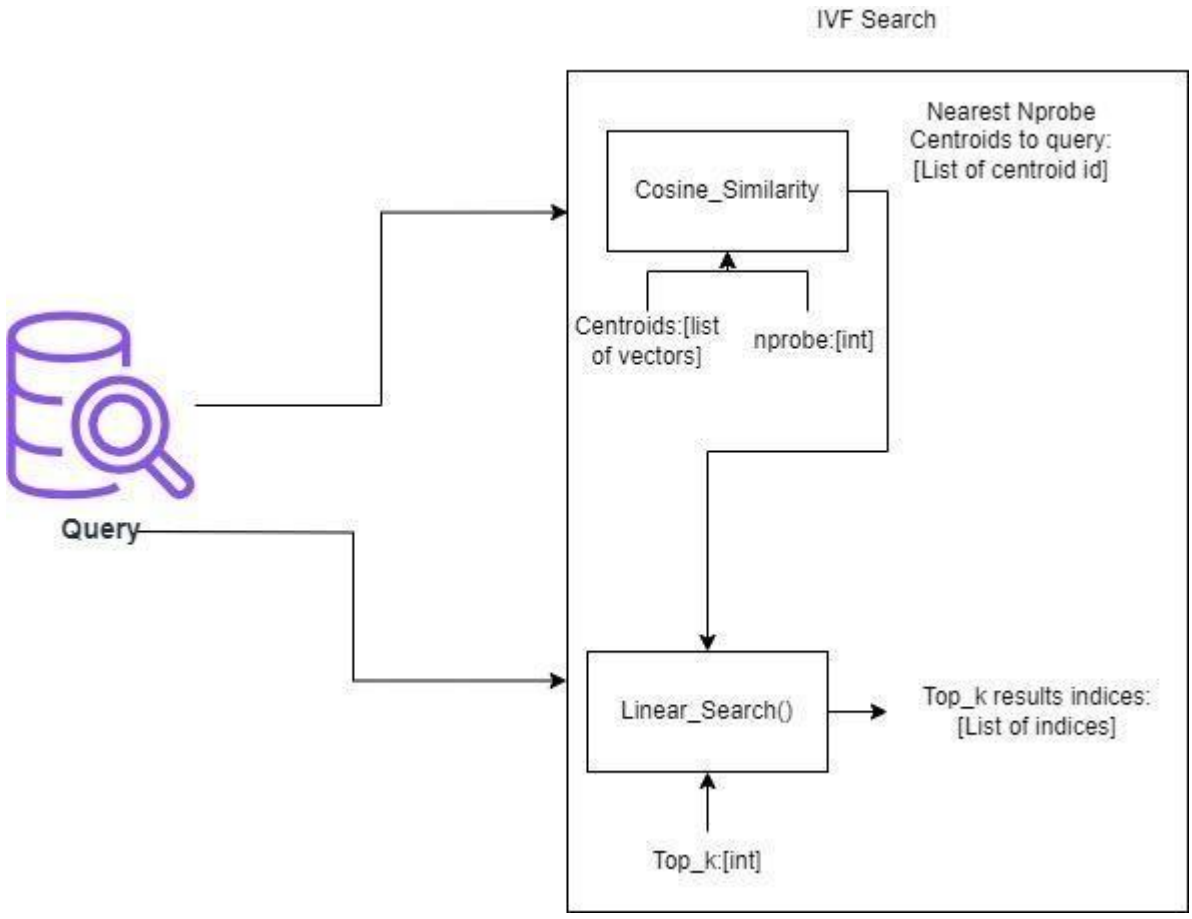
Architecture:



Indexing System:



Search System:



Results of our implementation:

algorithm	Parameters	Size	Score	Time in s	RAM MB
IVF	Nprobe = 3 , centroids = 16, iter = 32	10K	0.0	0.12	0.41
IVF	Nprobe = 16 , centroids = 128, iter = 32	100K	0.0	0.68	0.80
IVF	Nprobe = 32, centroids = 256, iter = 32	1M	0.0	3.93	11.19
IVF	Nprobe = 128 , centroids = 2048, iter = 32	10M	0.0	40.55	204.98
IVF	Nprobe =64 , centroids = 1024, iter = 32	10M	0.0	42.82	187.29

Indexer at size 1M - 20M : **IVF-ADC**

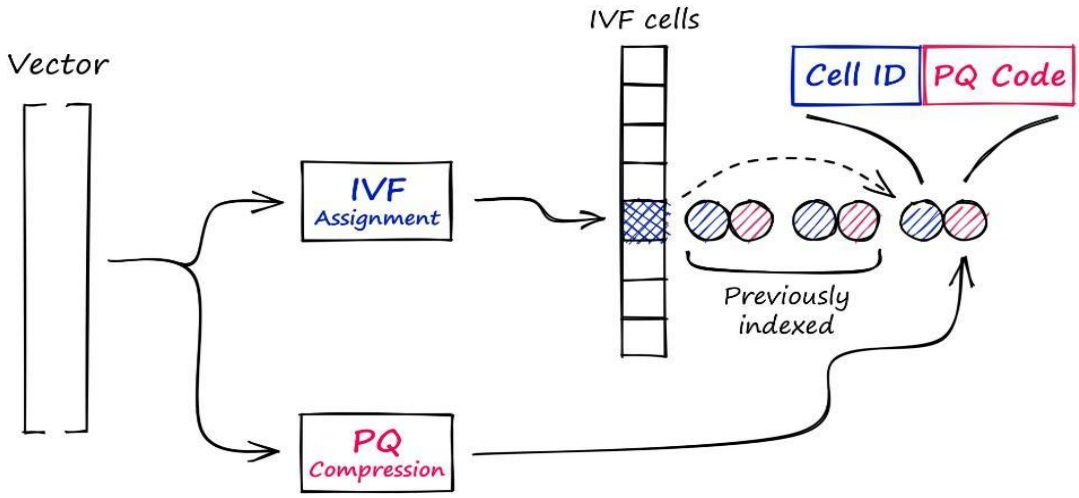
Algorithm:

Inverted File Asymmetric Data Computation:

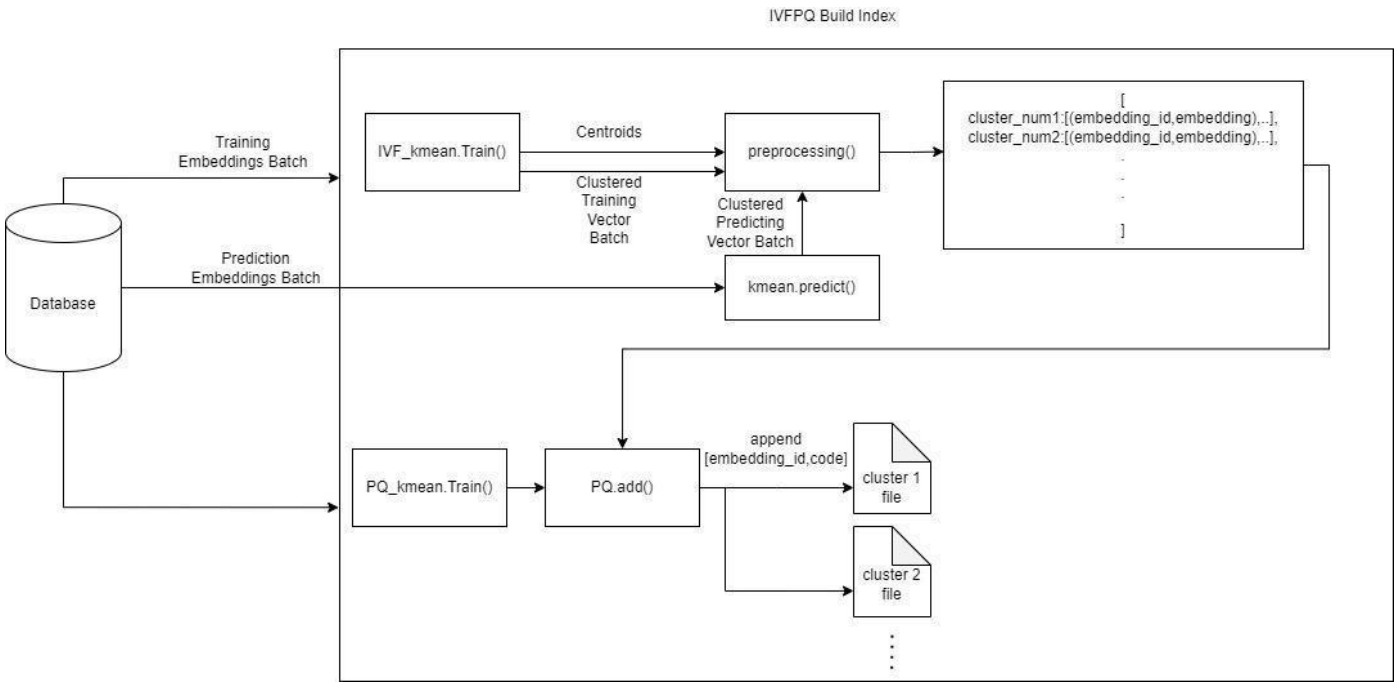
- Training
  - IVF: convert space into subspaces using clustering methods (kmeans) And assign each subspace representative “centroids”
  - PQ: divide vector to m subvector , for each subvector apply kmeans clustering to get 2**nbits centroids “IDs” that used to quantize vector to m-dimension vector
- Searching

  get nearest nprobe centroids to query vector and apply flat index on quantized vectors using ADC in each subspace to get 2*top_k nearest vectors to query submitted, then apply cosine similarity using original vectors on 2*top_k to get top_k.

Architecture:



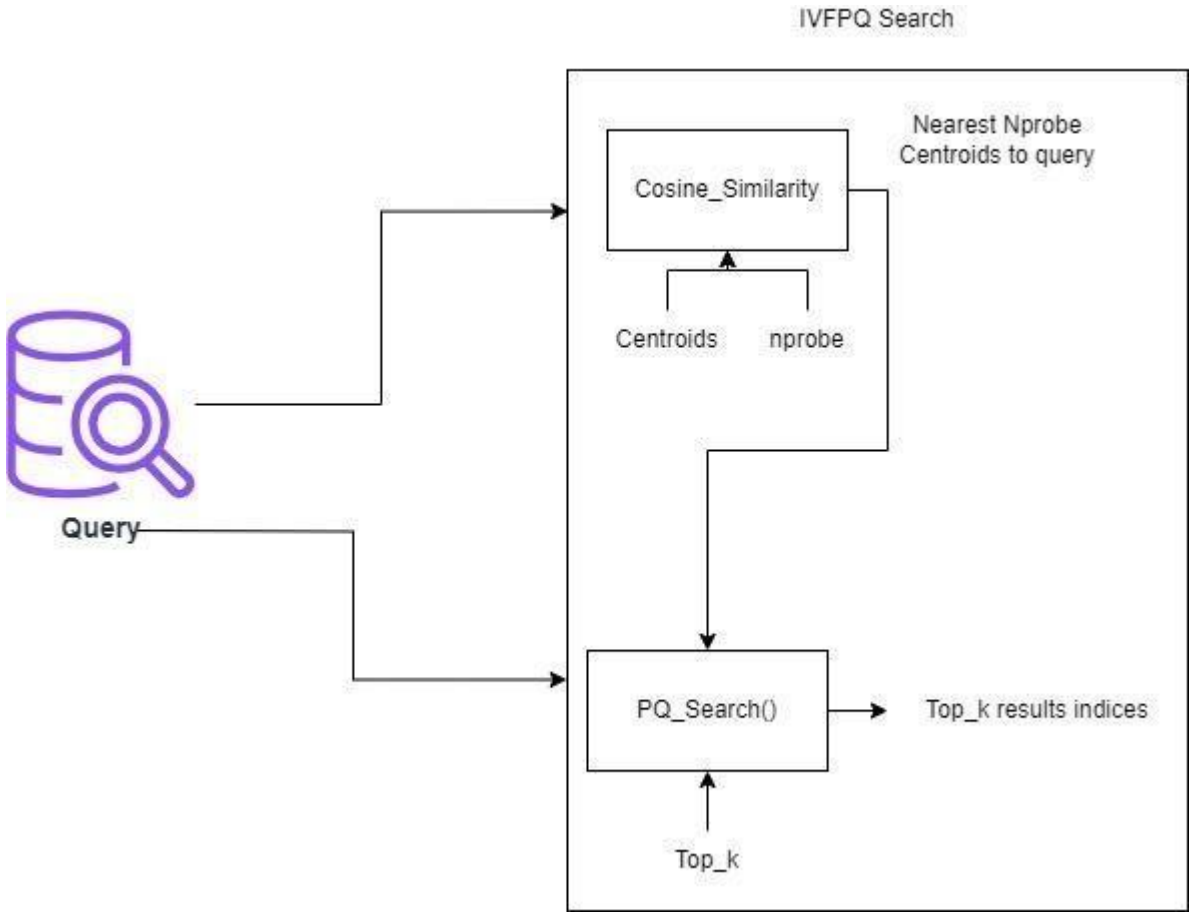
Indexing System:



PQ.add():

1. Designed to encode input data using a set of pre-trained KMeans estimators. It divides the input data into segments and applies Vector Quantization (VQ) using the KMeans estimators to each segment. The resulting output organizes the encoded data with associated identifiers.
2. Return data is added to the bin file according to the centroid it belongs to(codes_0.bin, codes_1.bin, ...).

Search System:



PQ_Search():

1. For each of these closest centroids, it accesses the corresponding binary file containing stored database codes.
2. It computes the distances between the query points and the stored codes from each binary file, resulting in a list of distances
3. From each set of distances, the function selects the top k*2 closest distances.
4. It aggregates these top k*2 distances from various centroids into a list.
5. Finally, it sorts and selects the top k*2 closest database codes from this aggregated list.
6. Get the dequantized vector of each code in the list and sort them according to cosine similarity between query
7. The function returns the IDs associated with the similar top k database vectors to the query points.

Results of our implementation:

algorithm	Parameters	Size	Score	Time sec	RAM MB
IVF-PQ	Nprobe = 64 , centroids = 1024, iter = 32 IVF training batch = 100K, predict batch=100K d = 70, m = 10, nbits = 7 PQ training batch = 100K, predict batch = 1K	1M	0.0	0.17	0.79
IVF-PQ	Nprobe = 64 , centroids = 1024, iter = 32 IVF training batch = 100K, predict batch=100K d = 70, m = 14, nbits = 8 PQ training batch = 100K, predict batch = 1K	5M	0.0	0.27	0.4
IVF-PQ	Nprobe = 64 , centroids = 1024, iter = 32 IVF training batch = 100K, predict batch=100K d = 70, m = 14, nbits = 8 PQ training batch = 100K, predict batch = 1K	10M	0.0	0.66	0.06
IVF-PQ	Nprobe = 64 , centroids = 2048, iter = 100 IVF training batch = 100K, predict batch=100K d = 70, m = 14, nbits = 8 PQ training batch = 100K, predict batch = 1K	15M	0.0	0.32	2.62
IVF-PQ	Nprobe = 64 , centroids = 1024, iter = 100 IVF training batch = 500K, predict batch=500K d = 70, m = 14, nbits = 8 PQ training batch = 500K, predict batch = 1K	20M	0.0	0.67	-152.55