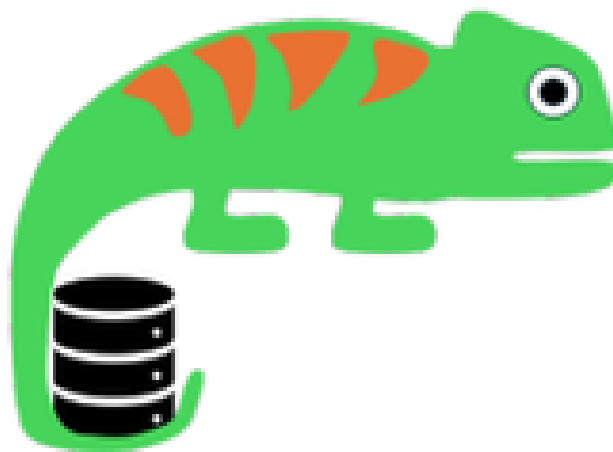


Acknowledgments

The research project is implemented in the framework of H.F.R.I call “3rd Call for H.F.R.I.” Research Projects to Support Faculty Members & Researchers” (H.F.R.I. Project Number: 25146).





Updates for Largescale Vector Search

Petros Ramos

Problem definition

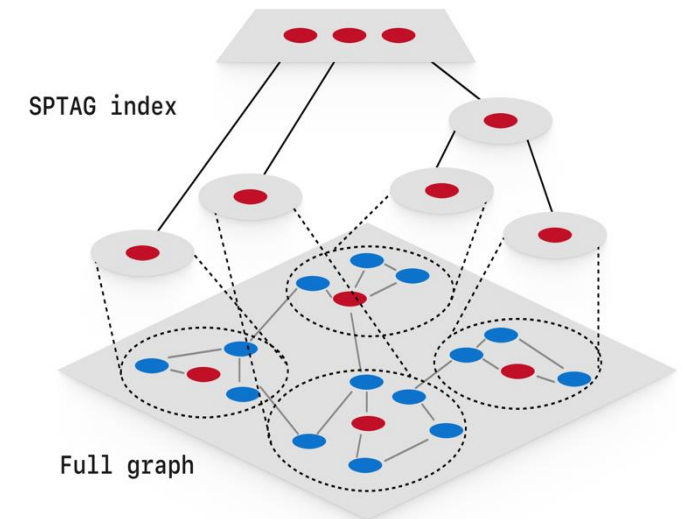
- Vector search indexes continuously accumulate new data during their lifetime.

New
vectors

- Incorporated quickly. Low computational and memory overhead.
- Updates avoid costly full index rebuilds.

Index
maintain

- High search accuracy
- Low query latency



SPFresh



 Yuming Xu, Hengyu Liang, Jin Li, Shuotao Xu, Qi Chen, Qianxi Zhang, Cheng Li, Ziyue Yang, Fan Yang, Yuqing Yang, Peng Cheng, and Mao Yang, “SPFresh: Incremental In-Place Update for Billion-Scale Vector Search”. In Proceedings of the 29th Symposium on Operating Systems Principles (SOSP '23). Association for Computing Machinery, New York, NY, USA, 545–561. <https://doi.org/10.1145/3600006.3613166>

Core Idea

- **Inserts directly**
- **Updates** are performed **in place**
- **Local reorganization**
- Maintain **balanced sizes**
 - Oversized postings are **split**
 - Underutilized postings are **merged**
- Continue **searching during updates**

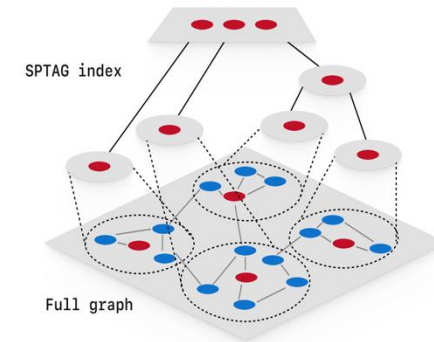
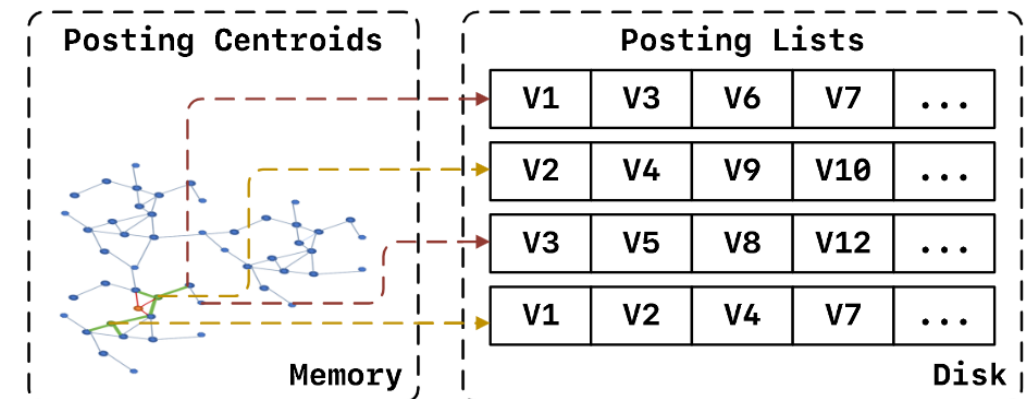
Base index (SPAN)

Structure

- **Cluster-based** index
- Vectors divided into **postings/clusters**
- Each **posting** is represented by a **centroid**
- **In-memory graph** index (SPTAG)

Search process

- Search **on index**
- **Load postings** from disk
- Search **on posting list**



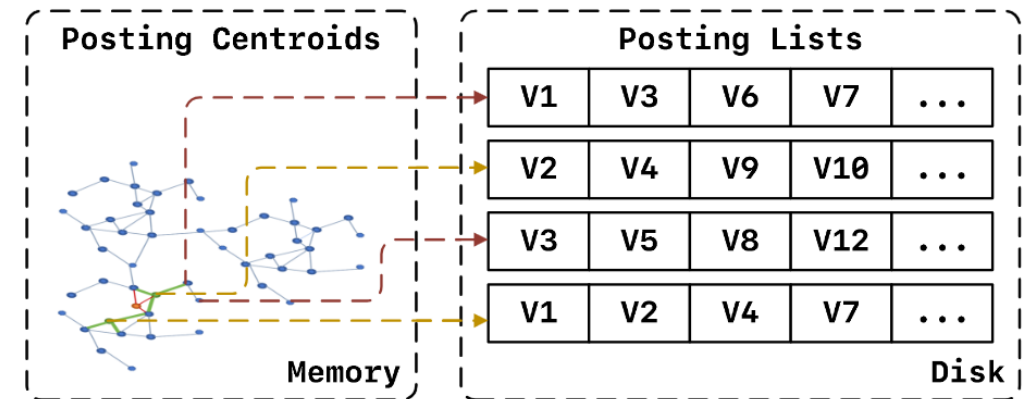
Base index (SPAN)

Index construction

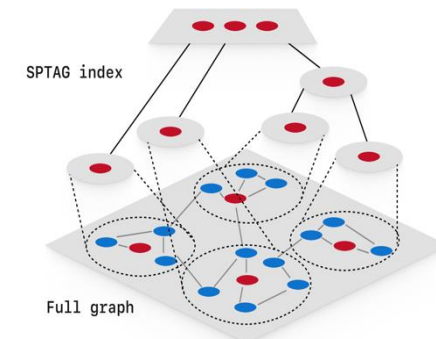
- Balanced **k-means** for **clustering**
- **Vectors**, stored in lists **on the disk** (posting lists)
- Centroids will be organized with **SPTAG** in **memory**

Hybrid flat structure

- K-means or k-d tree used as a **space-partitioning map**
- Then routes queries through a **flat Relative Neighborhood Graph (RNG)**



In memory centroids directing to their posting lists



SPTAG in memory index

LIRE Protocol

Lightweight Incremental Rebalancing

NPA Preserve

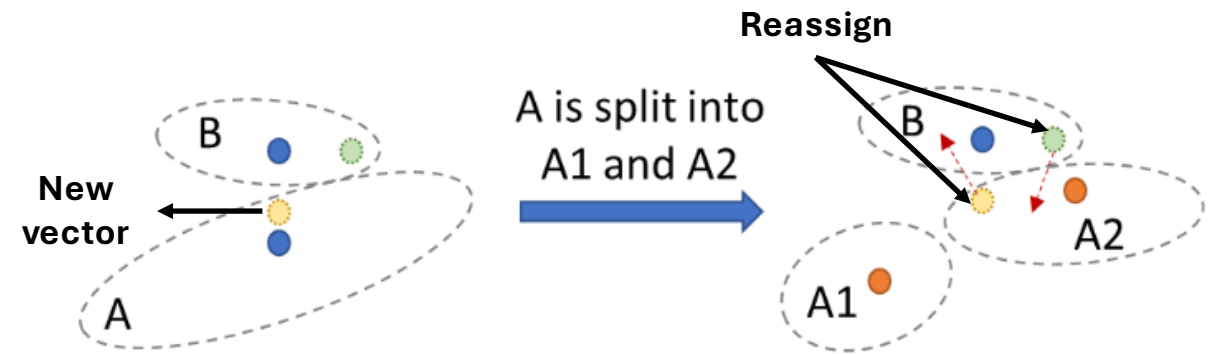
- **Nearest Partition Assignment**
- Each vector $\rightarrow$ nearest partition

Primary operations

- **Insert & Delete**

Internal operations

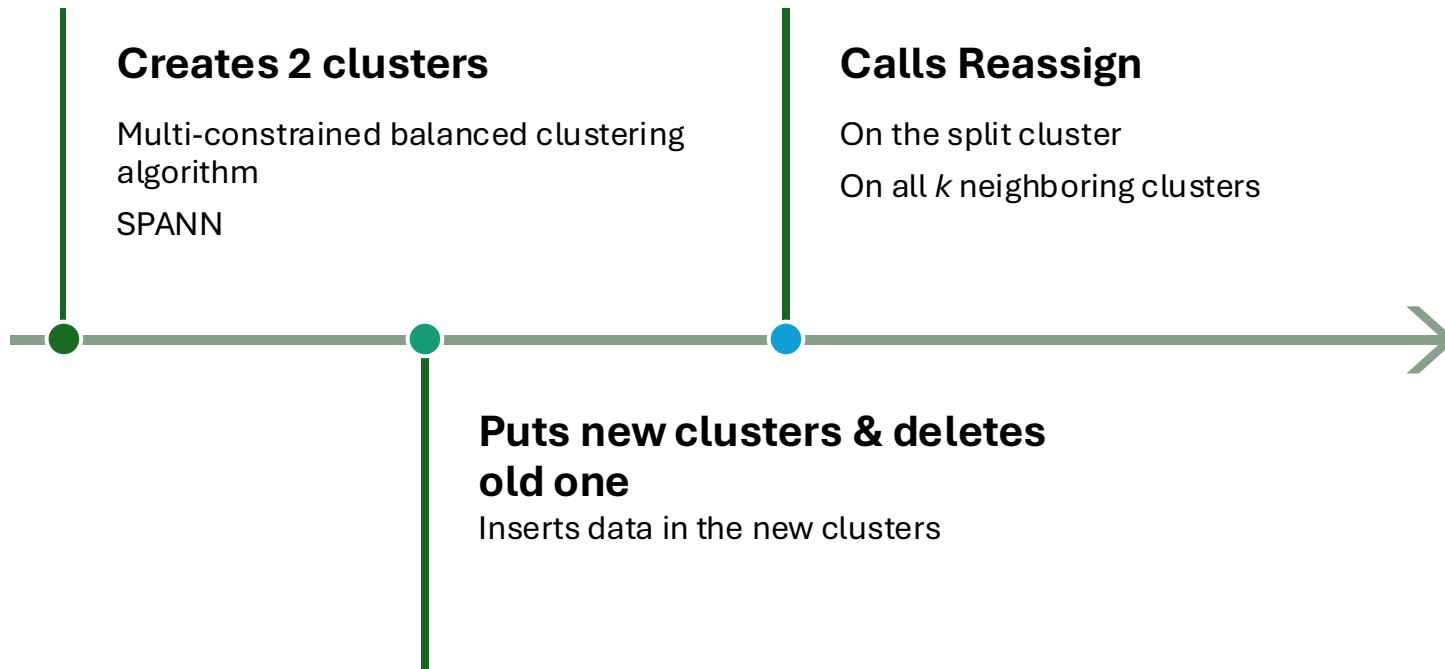
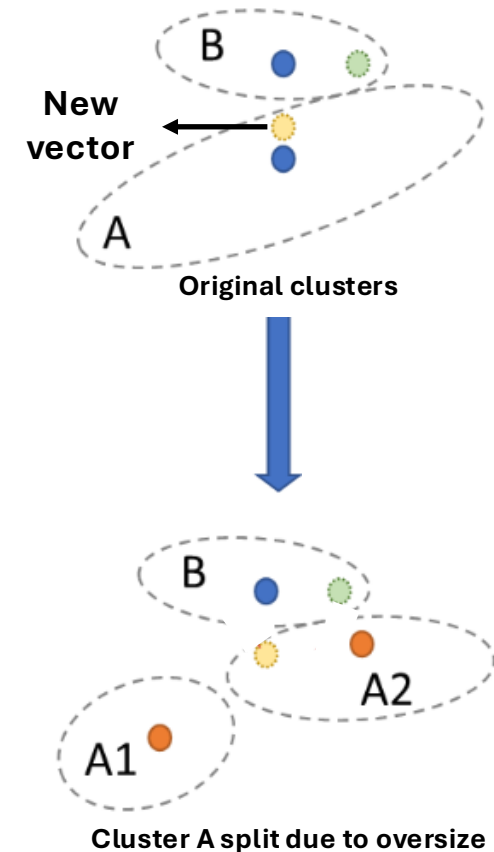
- **Reassign, Split & Merge**



NPA property preservation during split

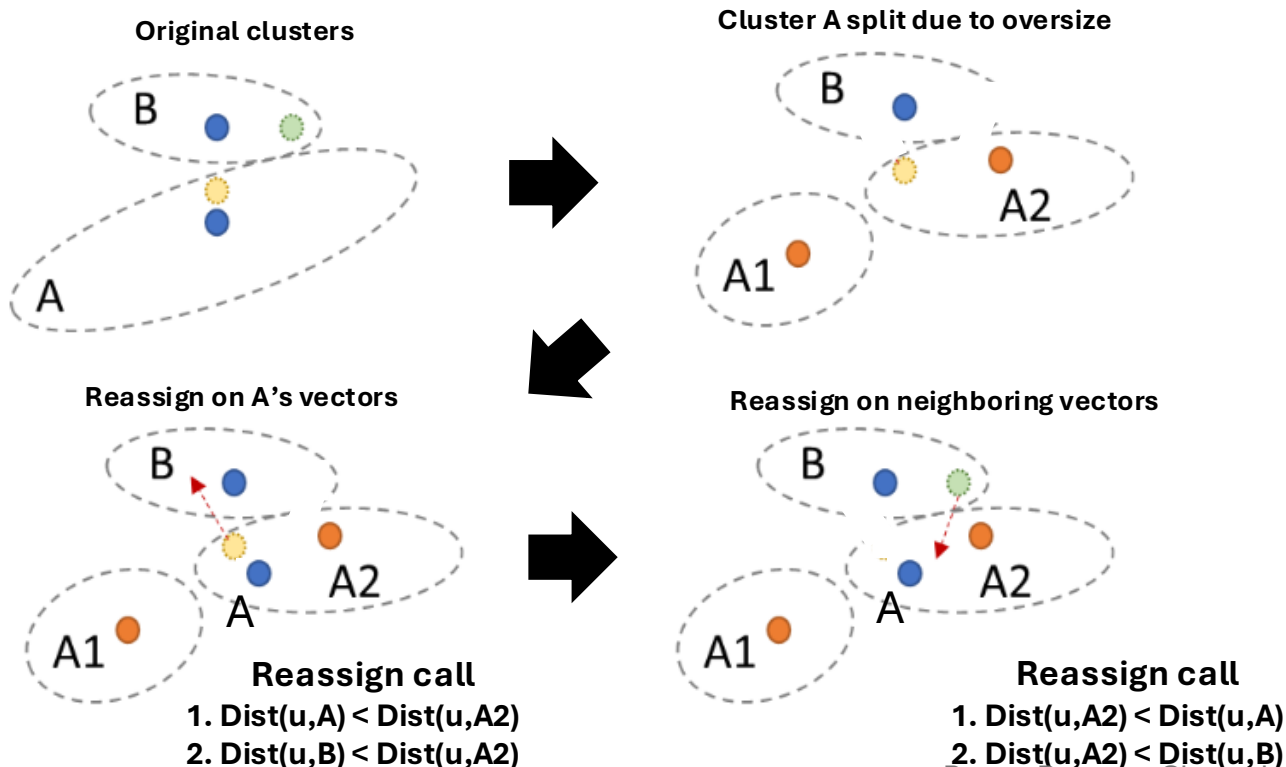
Split

Triggered when Posting **size** > **max**



Reassign on Split

- **Checks** vectors from **original cluster**
- **Checks neighboring clusters** with the new ones



Vectors from the **original cluster**

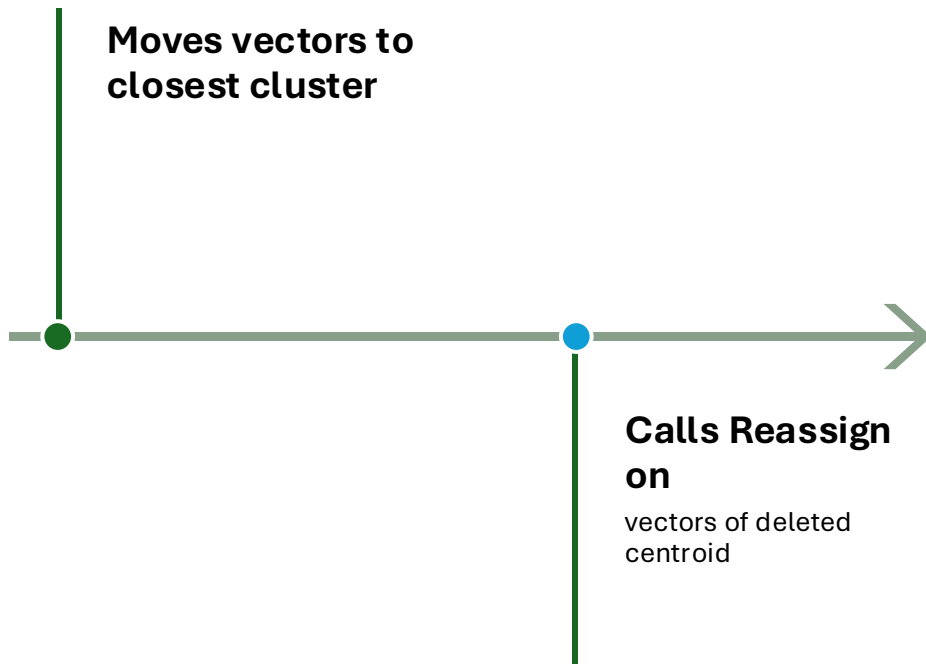
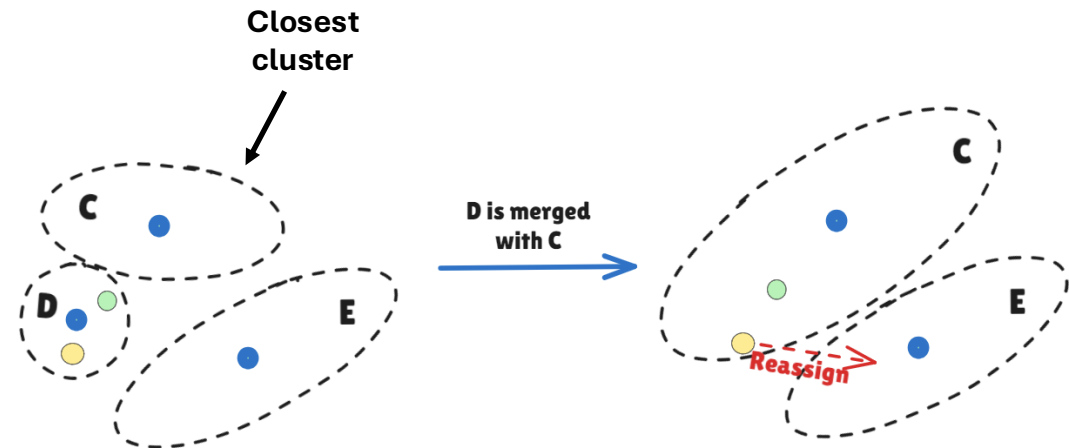
- If $\text{Dist}(u, A0) \leq [\text{Dist}(u, A1) \text{ or } \text{Dist}(u, A2)]$:
 - **Check nearby** neighboring centroids
- **Else:**
 - Assign u to the closer of **A1 or A2**

Vectors from the k **neighboring clusters**

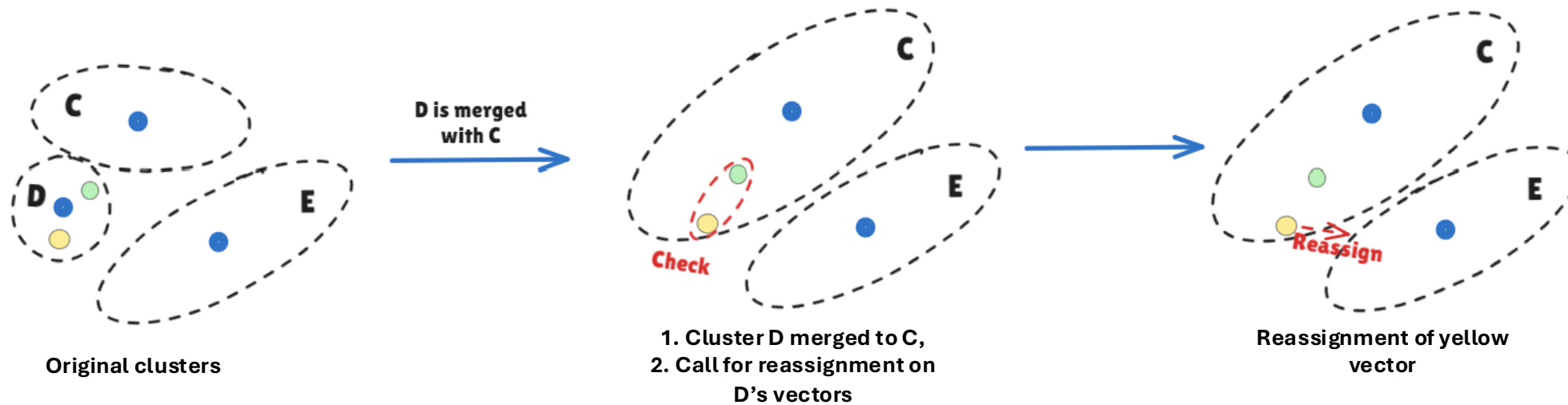
- If $[\text{Dist}(u, A1) \text{ or } \text{Dist}(u, A2)] \leq \text{Dist}(u, A0)$:
 - **Compare** distances to **A1 and A2**
- **Else:**
 - Keep the vector in its **current posting**

Merge

Triggered when Posting **size** < **min**



Reassign on Merge



Shortcomings

Split & merge thresholds

- No info on selecting Min/Max posting sizes
- Optimal values may depend on dataset and workload

Neighbor range

- SPFresh uses the nearest **64 postings**
- 64 is selected from their experiments
- May not generalize to other datasets or drift patterns

Single-node evaluation

- Demonstrates strong billion-scale performance on one node
- A distributed version is explicitly left as future work.

DeDrift



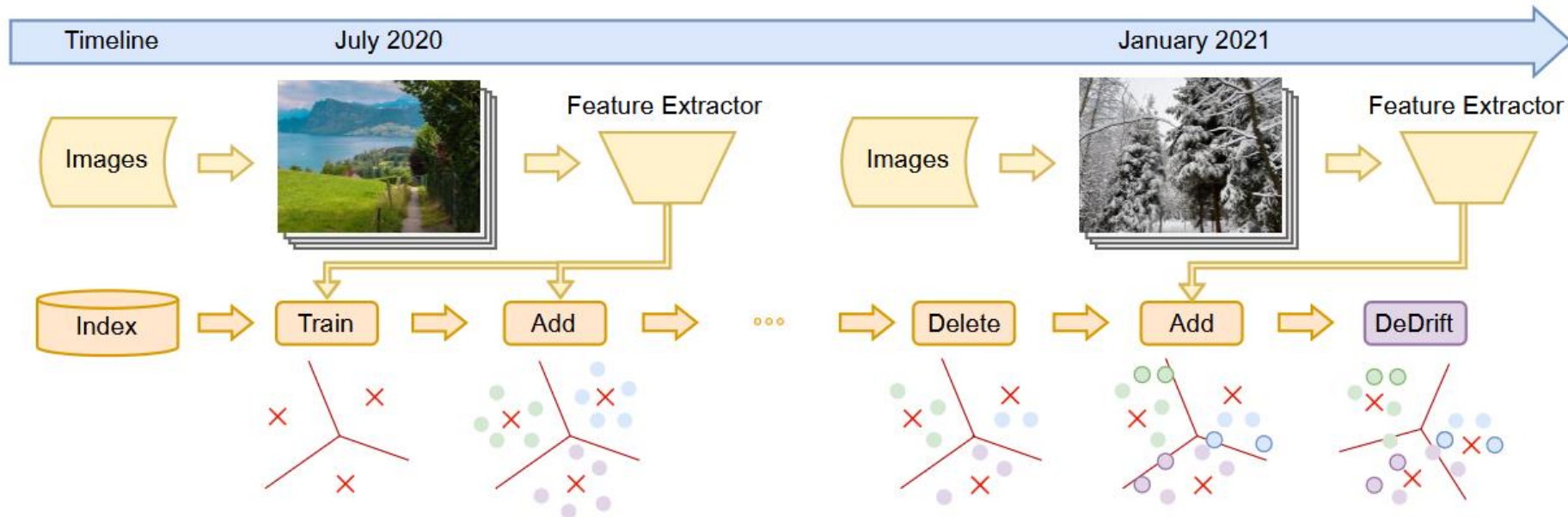
Baranchuk, Dmitry, Matthijs Douze, Yash Upadhyay and Ismet Zeki Yalniz. “DeDrift: Robust Similarity Search under Content Drift.” *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023): 10992-11001.

Core Idea

- **IVF** based indexes
- Content **distribution changes over time**
- **Adapt the index** to the current distribution
- Handle **database drift**

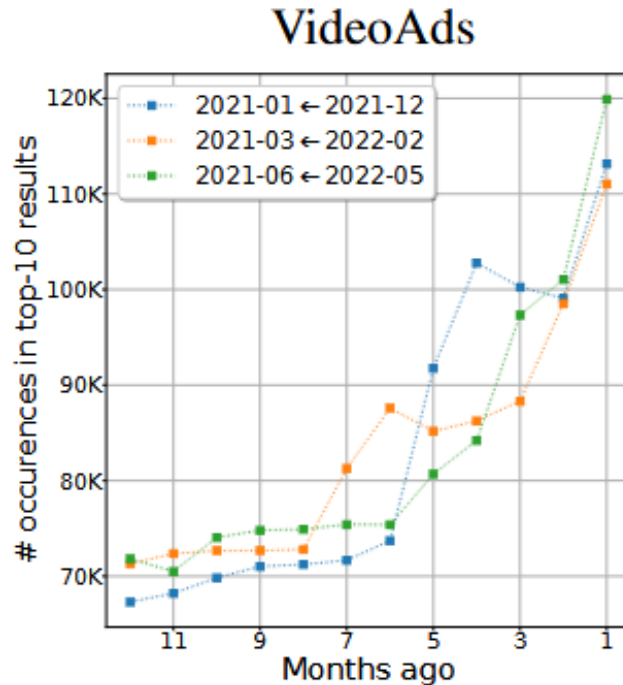
Content Drift

- Content distribution **changes over time**
- Drift can be **seasonal, sudden, or gradual**
- **Search accuracy decreases**



Seasonal change in pictures taken in July and January

Content Drift



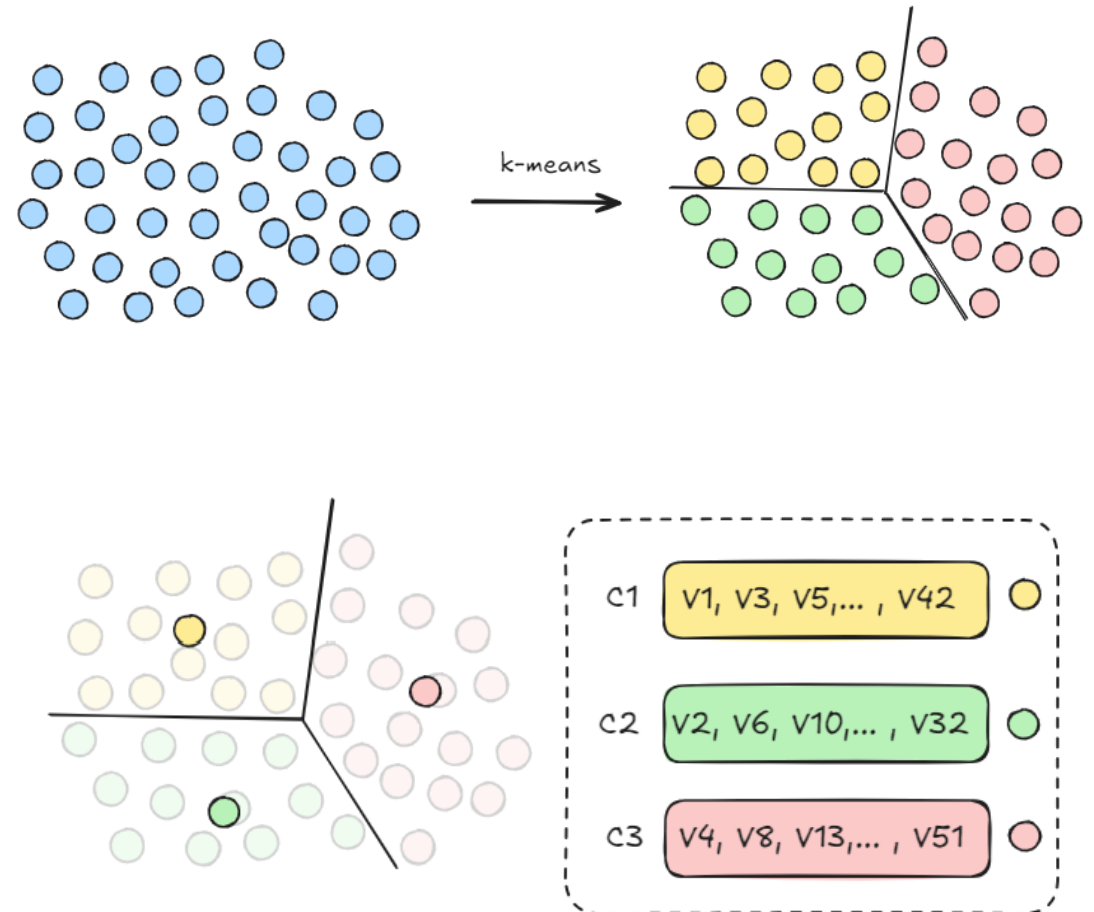
Top-10 vectors for queries
from a given month.

- **Video Ads dataset** (Oct. 2020 – Oct. 2022)
- For each month M_i use queries from that month
- Retrieve the **top-10 nearest neighbors**
- Find months from nearest neighbors
- A shift in the neighbors month distribution indicates **content drift**

Index construction

FAISS computational library

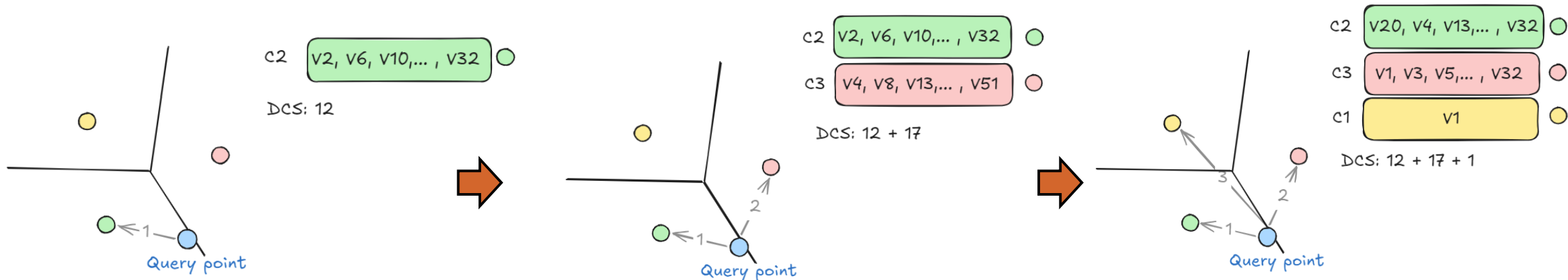
- Similarity based search on top-k
- Based on FAISS computational library
- Uses k-means clustering
- Data stored on the disk
- Index on ram



Search

Methodology

- Finds closest centroids
- Calculates distances on the cluster list
- Repeats process for other centroids
- Stops when distance computation limit (dcs) hits



Updating Algorithms

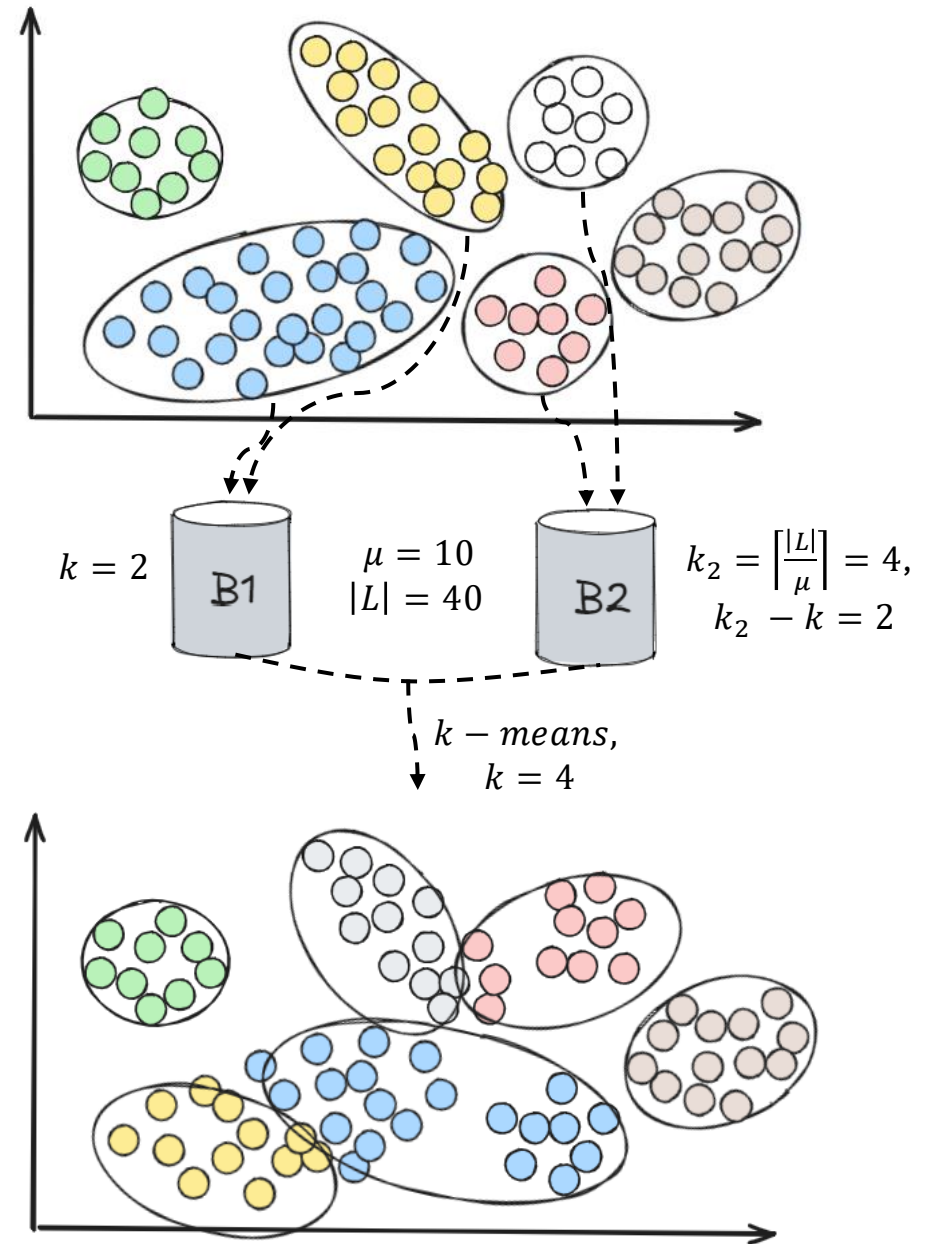
- Dedrift-**Split**:
 - Repartitions the k largest clusters
- Dedrift-**Lazy**:
 - Updates all centroids
 - Vectors are not reassigned
- Dedrift-**Hybrid**:
 - Dedrift-Split + Dedrift-Lazy
 - Updates centroids, then splits k largest clusters
- Dedrift in **compressed** domain
 - Product quantization compressing

Dedrift – Split

The Goal: Break large clusters to restore balance, without changing the total number of clusters.

Methodology

- **Bucket B1:** Stores the k largest, clusters
- From k largest calculate $k_2 = \left\lceil \frac{|L|}{\mu} \right\rceil$
 - $|L|$ = number of embeddings
 - μ = mean cluster size
- **Bucket B2:** Stores the k₂ – k smallest, clusters
- From **B1** and **B2** calculate **K-means**, with **k₂** centroids



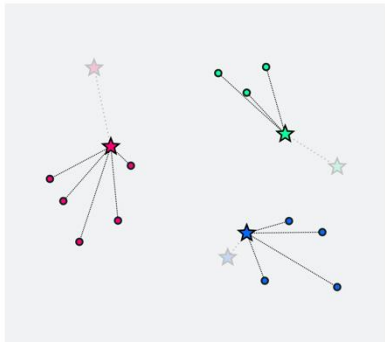
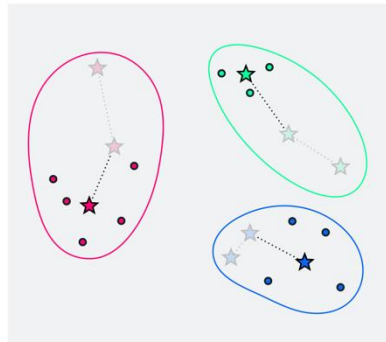
Dedrift – Lazy & Hybrid

Dedrift Lazy:

- Updates the centroids to new mean
- Vectors don't move

Dedrift Hybrid:

- Updates first the centroids of all clusters (like Lazy)
- Then Splits the clusters first (like Split)

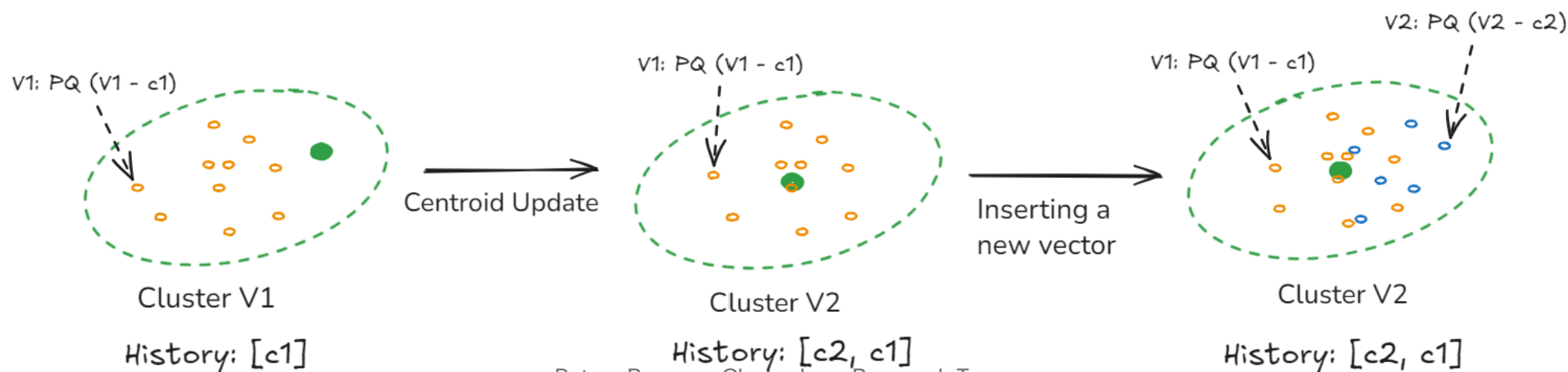


Dedrift with compression

Lazy in compressed domain:

- **DeDrift-Lazy** for compressed IVF
- **PQ compression** is used
- Encoded relative to the centroids position

- **Lazy updates centroid positions over time**
- Retains a **history of centroid versions**
- **PQ lookup table** for each relevant centroid version
- Tracks which centroid version was used to encode each group of vectors



Shortcomings

Top k parameter

- **In Split** the top k clusters might differ across datasets

Periodic adaptation need

- Is not automatic
- Update schedule needs to be chosen

Compressed-domain

- Need for version history
- Need for extra lookup tables

Acknowledgments

The research project is implemented in the framework of H.F.R.I call “3rd Call for H.F.R.I.” Research Projects to Support Faculty Members & Researchers” (H.F.R.I. Project Number: 25146).

