Summary of the Neurips 2023 BigANN Challenge - Practical vector search

Neurips workshop, 2023-12-16

Summary of…

- The other submissions
- A few details for other tracks
- Open discussion

Filtered search track

Matthijs Douze

Puck team (Baidu)

- multi-level index structure that has four levels by default.
  - first two levels: trained using vector quantization, which constructs K centroids using k-means at each level.
  - last two levels, product quantization
- Bag of words on the centroids:
  - collection of points' labels in this centroid.
  - At query time: centroids that do not meet the requirements are filtered out
- First pq level:
  - points that do not meet the requirements are filtered (via a callback similar to the baseline implementation)
  - Keep the top-M most similar samples as the candidate set.
- the second pq level:
  - Re-rank samples in the candidate set and return top-K.

HWTL_SDU-ANNS-filter

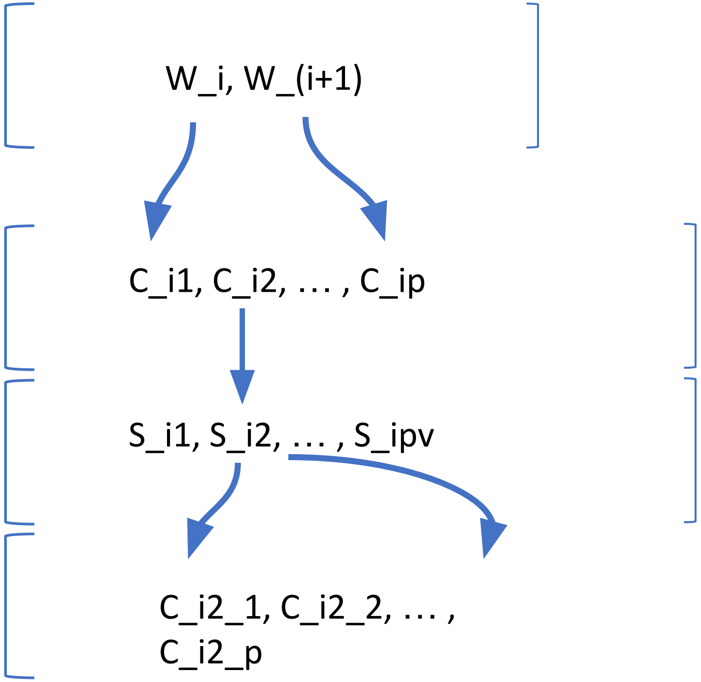
- Authors:
  - HWTL: Yu Gao ,Weipeng Jiang, Weixi Zhang, Chengda Wu, Chaorui Zhang, Yibin Ding, Wei Han, Bo Bai
  - SDU：Zhiwei Chen, Kun Wang, Zixu Li, Rui Wang, Qianyun Yang, Yupeng Hu, Liqiang Nie
- Binary-only submission
- We create a separate index for each tag.
  - used for queries containing the corresponding tag (including single and double tag queries).
- Use binary signatures
  - We optimize the probability for 1s in a random signature
  - minimizing the probability Pr[the signature of a vector with 10 tags other than t covers the signature of t] for a fixed tag t.

wm_filter

- Authors:
  - Ashwani Rajan, SIjie Chen, Hongji Ye, and Zongjun Tan
- Works on top of an IVF ANN index
- Index creation
  - First the cluster of the coarse quantizers are computed
  - Filter data structure is created based on clusters and filters (next slide)
- Search time
  - Coarse quantizer is searched
  - For the top clusters C_i :
  - Scan the filter array to find the indices in the cluster array that belongs to the filter
  - If 2 words are given, intersect the indices
  - Traverse the cluster only using the indices of the filter

Filter data structure

Given the word W_i and a cluster C_i2 allows to find all the indices of the cluster that belong to the word W_i



First array has has many entries as words in the filter. Points to a second array that contains the clusters that have an intersection with the word W_i

Second array contains in a range all the cluster that intersect W_i. Apply bisection to find given cluster

Third array has same dimension of the second one and points to a forth array with the list of indices of the corresponding cluster

Contains all the indices in the cluster for W_i and C_i2

DHQ – Dynamic Hybrid Query

- first constructs an inverted label index for the original vectors in the dataset,
  - filters the labels according to the number of original vectors included.
  - For the labels below the threshold, the IVF index is constructed, and the IVF algorithm is used to filter and search.
- For labels that exceed the threshold,
  - we build a KNNG for each label and randomly add entry nodes to increase the randomness of the search process,
  - product quantization to speed up the vector similarity calculation
  - optimized KNNG's memory to ensure that it meets track memory requirements.
  - in the process of graph index routing, we weighted the distance of vectors that exceeded the label filtering conditions to ensure that their neighbors would not be filtered in advance.

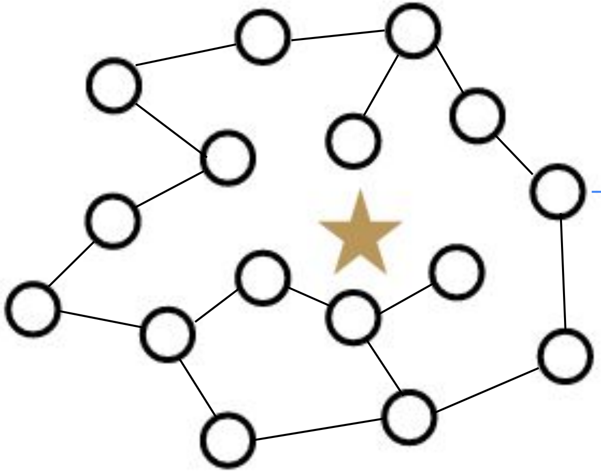
Sparse track

Amir Ingber

Sparse track submissions

- **PyANNs** [Zihao Wang]
  - graph-based modification of hnsw
- **GrassRMA**: GRAph-based Sparse Vector Search with Reducing Memory Accesses (aka **shnsw**) [Meng Chen et al.]
  - See next slide
- **NLE** [Naver labs europe]
  - Inverted-index based, modify a BM-25 index (based on **pisa**) to take in arbitrary values
- **Cufe** [Ibrahim et al.]
  - modification of the baseline
- **Linscan** (baseline): exact scan of an inverted index with early stopping

GrassRMA: GRAph-based Sparse Vector Search with Reducing Memory Accesses (aka **shnsw**)



50x faster than the baseline

- **Storage (data layout)** Conventional CSR ✗

Combine the indices and the data into ONE array ☑

Fewer Random Accesses

Dimension indices	3	10	7
Values	0.4	0.6	0.1

⇩

Combined	3;0.4	10;0.6	7;0.1

- **Similarity Calculation**

Store the lower bound and upper bound of base vectors.

Only calculate the intersected set.

Query	10	59	78		
Base data	54	60	76	89	98

34	57	(Fast return)			
69	77	78	82	99	

Public vs private query set

	Public query set (QPS)	Private query set (QPS)
Linscan	93	95 (+2%)
cufe	105	98 (-7%)
NLE	2,359	1,313 **(-44%)**
shnsw	7,137	5,078 (-28%)
pyanns	8,732	6,500 (-25%)

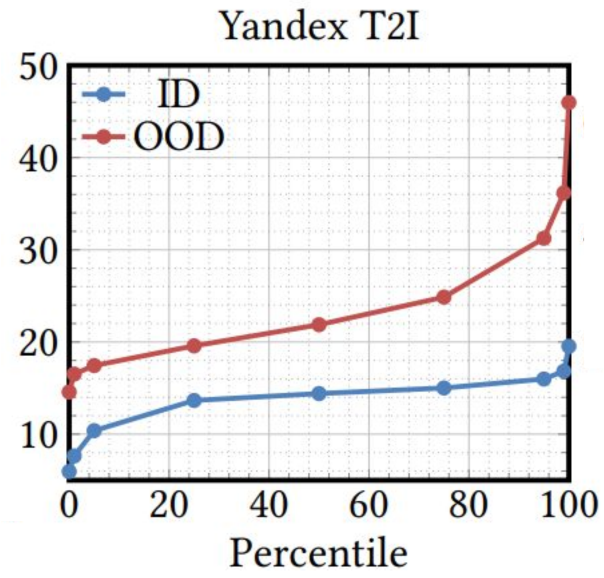
Values are QPS for Recall@10 at least 0.9

Out Of Distribution track

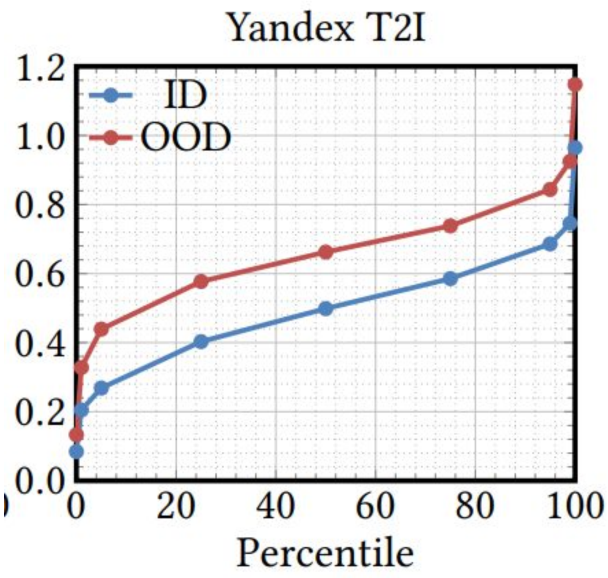
Entries

Entry	QPS with >90% recall	Authors
Mysteryann, mysteryann-dif	22555.248017	Meng Chen, Yue Chen, Rui Ma, Kai Zhang, Yuzheng Cai, Jiayang Shi, Yizhuo Chen, Weiguo Zheng. (Fudan University)
pyanns	22295.584534	Zihao Wang (Shanghai Jiao Tong University)
sustech-ood	13772.370641	Long Xiang, Yuxiang Yang, Xiao Yan, Yanqi Chen, Bo Tang (Southern University of Science and Technology)
puck	8699.573200	Jie Yin, Ben Huang (Baidu)
vamana	6753.344080	Magdalen Dobson, Guy Blelloch (CMU)
ngt	6373.934425	Masajiro Iwasaki (Yahoo Japan)
epsearch	5876.982706	Yusuke Matsui, Yutaro Oguri (The University of Tokyo)
diskann	4132.829728	Baseline
cufe	3561.416286	Michael Ibrahim, Farah Abdelfattah, Abdelrahman Ezzat, Ziad Abdelhameed, Ali Hashish (Cairo University)

Analysis of dataset [credit: Shikhar Jaiswal]



Histogram of Mahalonobis distances between in-distribution (image-image) and out-of-distribution (image-text) pairs

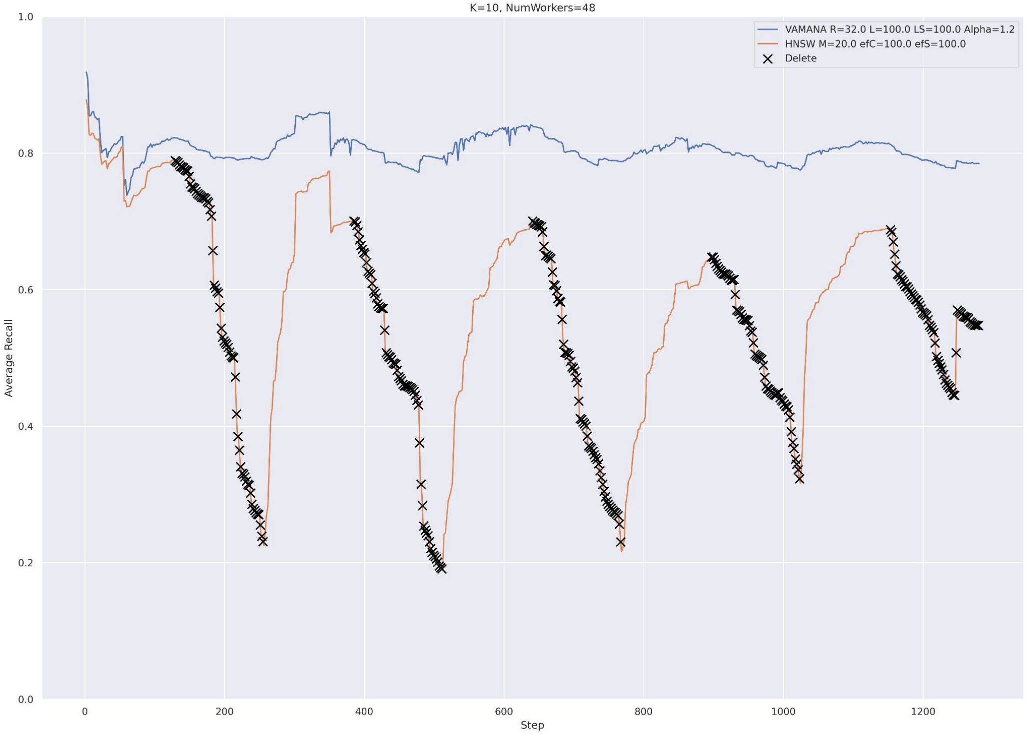


Cluster radii of top-10 NNs of index point (ID) and query (OOD) sample sets

Streaming track

Entry	Recall@10	Authors
Puck	0.9849	Jie Yin, Ben Huang (Baidu)
Hwtl_sdu_anns_stream	0.9675	○ HWTL: Yu Gao ,Weipeng Jiang, Weixi Zhang, Chengda Wu, Chaorui Zhang, Yibin Ding, Wei Han, Bo Bai ○ SDU：Zhiwei Chen, Kun Wang, Zixu Li, Rui Wang, Qianyun Yang, Yupeng Hu, Liqiang Nie
PyANNS	0.9597	Zihao Wang (Shanghai Jiao Tong University)
cufe	0.819	Michael Ibrahim, Farah Abdelfattah, Abdelrahman Ezzat, Ziad Abdelhameed, Ali Hashish (Cairo University)
Baseline	0.883	

Recall across steps



Sequence of inserts, deletes and search

MSTuring-30M-clustered

- Cluster with k-means, k=64
- 5 rounds of
    - insert a sample from one of 64 clusters
    - Search
    - Delete a sample from one of 64 clusters
    - search

Organizer’s solutions (pinecone)

Amir Ingber

Solutions from the pinecone research team

- **In parallel to formal applications, solutions from pinecone:**
  - Many more details to come!
  - Blog post, papers
- **Filter track:** 69k QPS (!) [vs 35kQPS of best challenge submission]
  - Rearranged IVF + optimal recall allocation + AVX512 + …
- **Sparse track:** 7,800QPS (!)
  - Sparse IVF + graph expansion + …
- **OOD:** 19,500QPS
  - DiskANN + IVF + graph expansion + …
- **Streaming:** R@10: 0.99 at 57min
  - Modified DiskANN + reranking

Next steps

Call for datasets

Auto-generate leaderboard

Separate track for billion-scale datasets