ACM SIGMOD Programming Contest 2023

SUSTech DBGroup · Finalist

Yanqi CHEN, Jiarui LUO, Long XIANG, Shimin LUO, Hongxun DING

Advisor: Professor Xiao Yan, Professor Bo Tang

DBGroup@SUSTech: https://dbgroup.sustech.edu.cn

Task Overview

Task: Build an **approximate K-NN Graph** for a set of vectors.

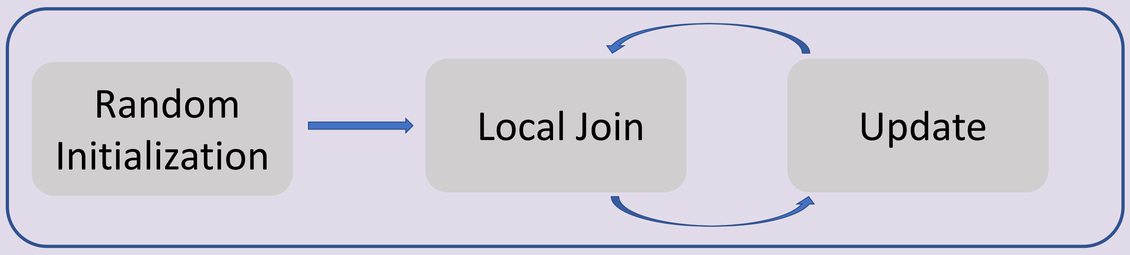
Input/Output:

- **Inputs:** dataset contains 10M 100 dimension vector data.
- **Output:** 100-nearest neighbors for each vector in given dataset.

Performance Metric:

$$Recall = \frac{\text{number of true top 100 nearest neighbors}}{100}$$

Solution Overview



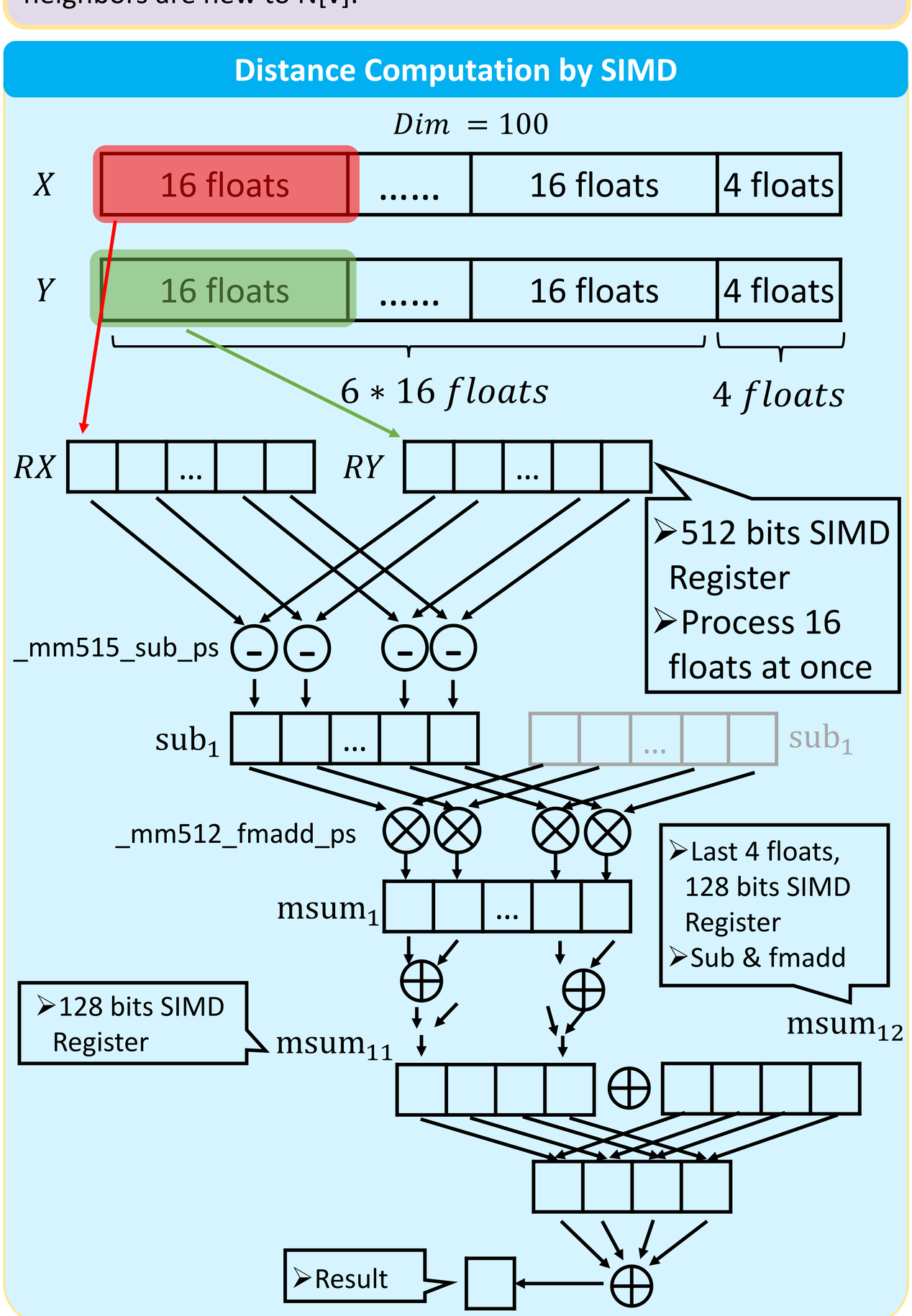
NN-Descent Framework

Random Initialization: randomly initialize the neighbor lists N[v] of each node v.

Local Join: for each node v, and p, q ∈ N[v], update N[p] and N[q] based on the similarity between p and q if one of them is new to the N[v].

Update: update the information of N[v] to determine which neighbors are new to N[v].

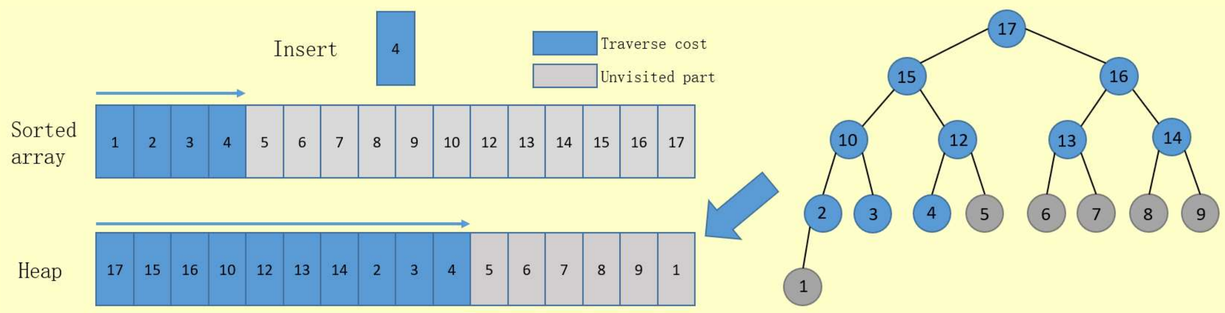
Distance Computation by SIMD



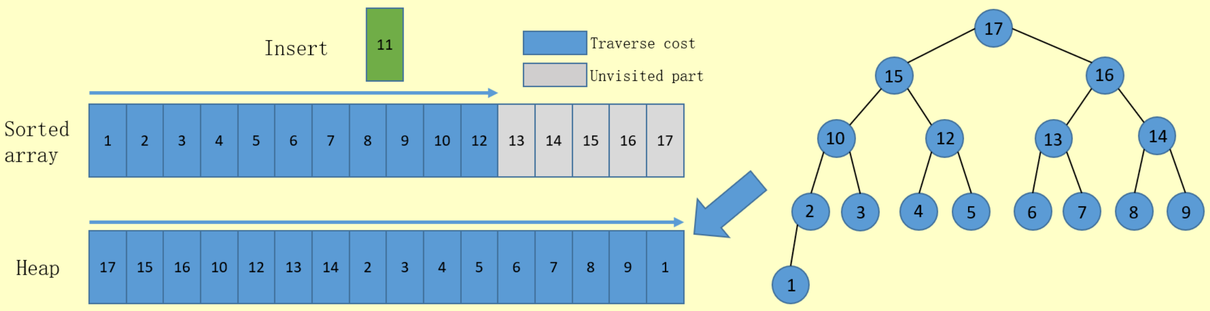
Heap v.s. Sorted Array

Observation:

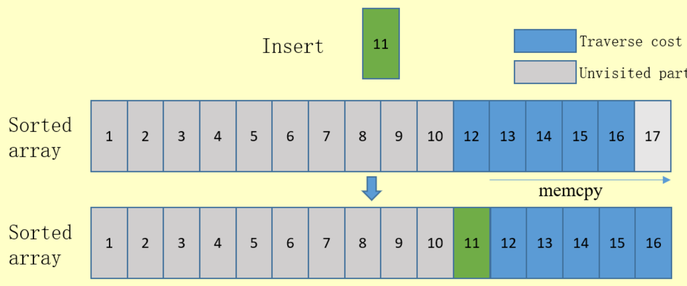
- In the first few iterations, heap is a nice data structure to represent neighbor lists due to frequent update operations.
- After several iterations, the update of neighbor lists becomes less frequent. On the contrary, many update attempts fail since there they have been in the neighbor list.
- In that case, using sorted array can be more efficient than heap due to less traverse cost as well as low insertion cost.



(a) Less traverse cost for duplicate value insertion



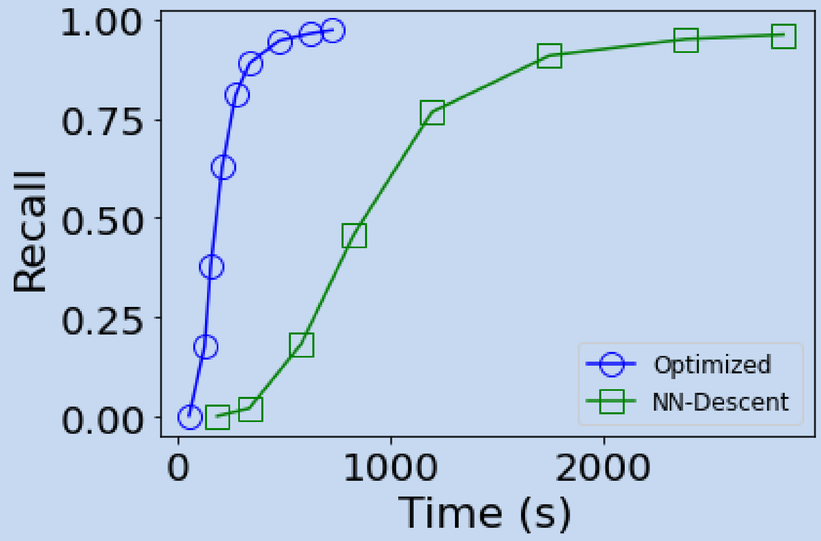
(b) Less traverse cost for new value insertion



(c) Low Insertion memory copy cost

Experimental Evaluation

- Experiment Environment: Intel(R) Xeon(R) Gold 5318Y CPU @ 2.10GHz and 512GB memory.
- The experiment is conducted on the released dataset of the contest, which consists of 10M float vectors of 100 dimensions.
- Optimized NN-Descent is **3** times faster than original one.



References

[1] Dong W, Moses C, Li K. Efficient k-nearest neighbor graph construction for generic similarity measures[C]//Proceedings of the 20th international conference on World wide web. 2011: 577-586.

[2] Fu C, Cai D. Efanna: An extremely fast approximate nearest neighbor search algorithm based on knn graph[J]. arXiv preprint arXiv:1609.07228, 2016.