ACM SIGMOD Programming Contest 2023

TEAM X2A3008M

Member: Meng Chen, Advisor: Kai Zhang

DASlab - System Group @ Fudan University

Contact: mengchen22@m.fudan.edu.cn, zhangk@fudan.edu.cn

Task Overview

Task: The objective is to create an approximate K-NN graph using a set of high-dimensional vectors where each vertex is linked to its approximate k nearest neighbors based on the Euclidean distance.

Dataset:

Dataset	# of vectors	dimension
Turing	10,000,000	100

Vectors are Bing queries encoded by Turing AGI v5.

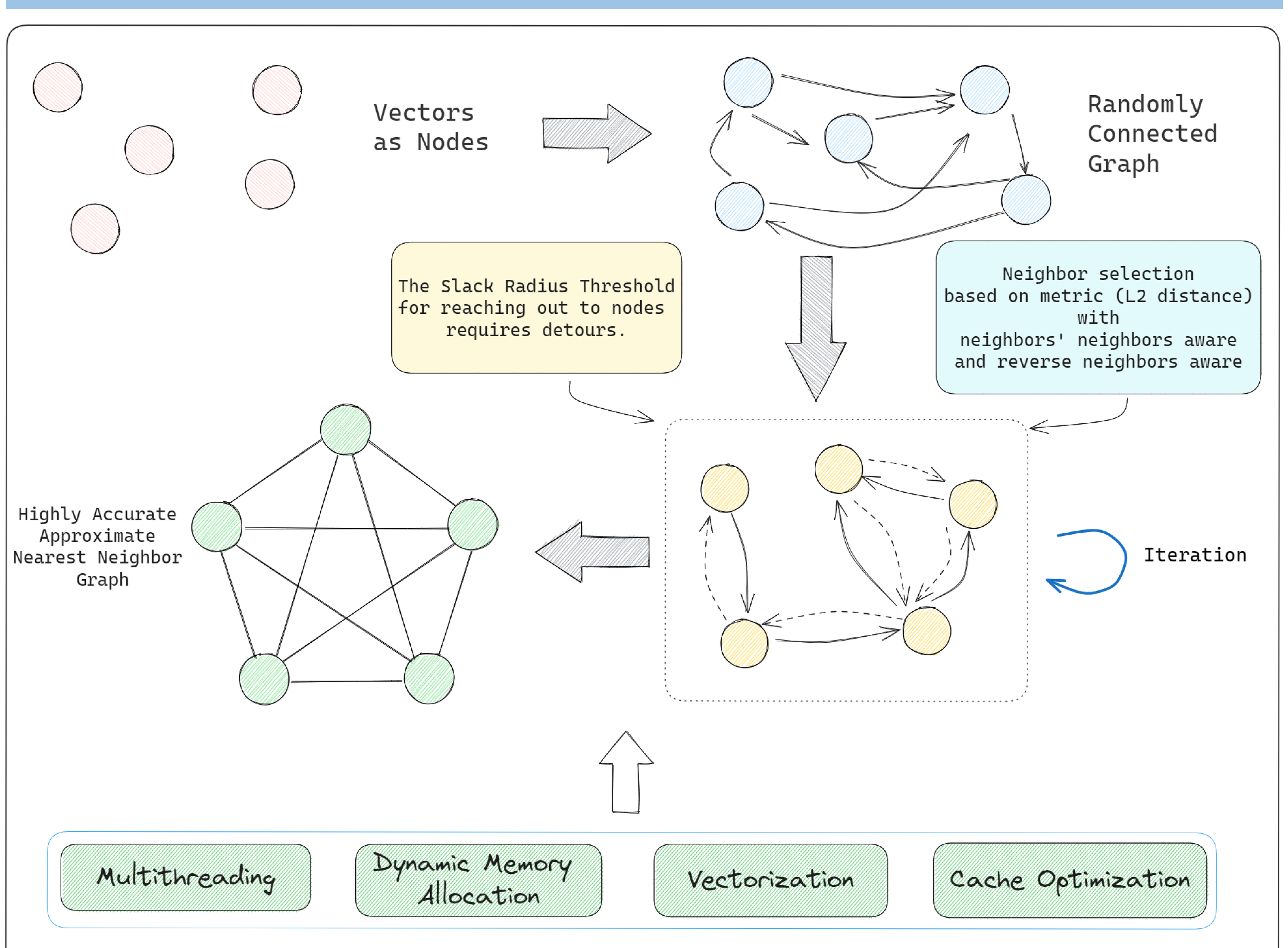
Input: A set of vectors **Output:** Approximate KNN Graph (K=100)

Evaluation Metric: $Recall = \frac{ground\ truth\ (100-NN\ neighbors)}{100}$

Hardware Conditions: Azure Standard F32s_v2, 32C64G

Time Limit: 30 minutes + 60 seconds (reprozip overhead)

Solution Overview



Implementation

Algorithm: Slack-Threshold NN-Descent, which is based on NN-Descent [1] with optimizations and improvements.

Data Structures Required by Each Vector(Node):

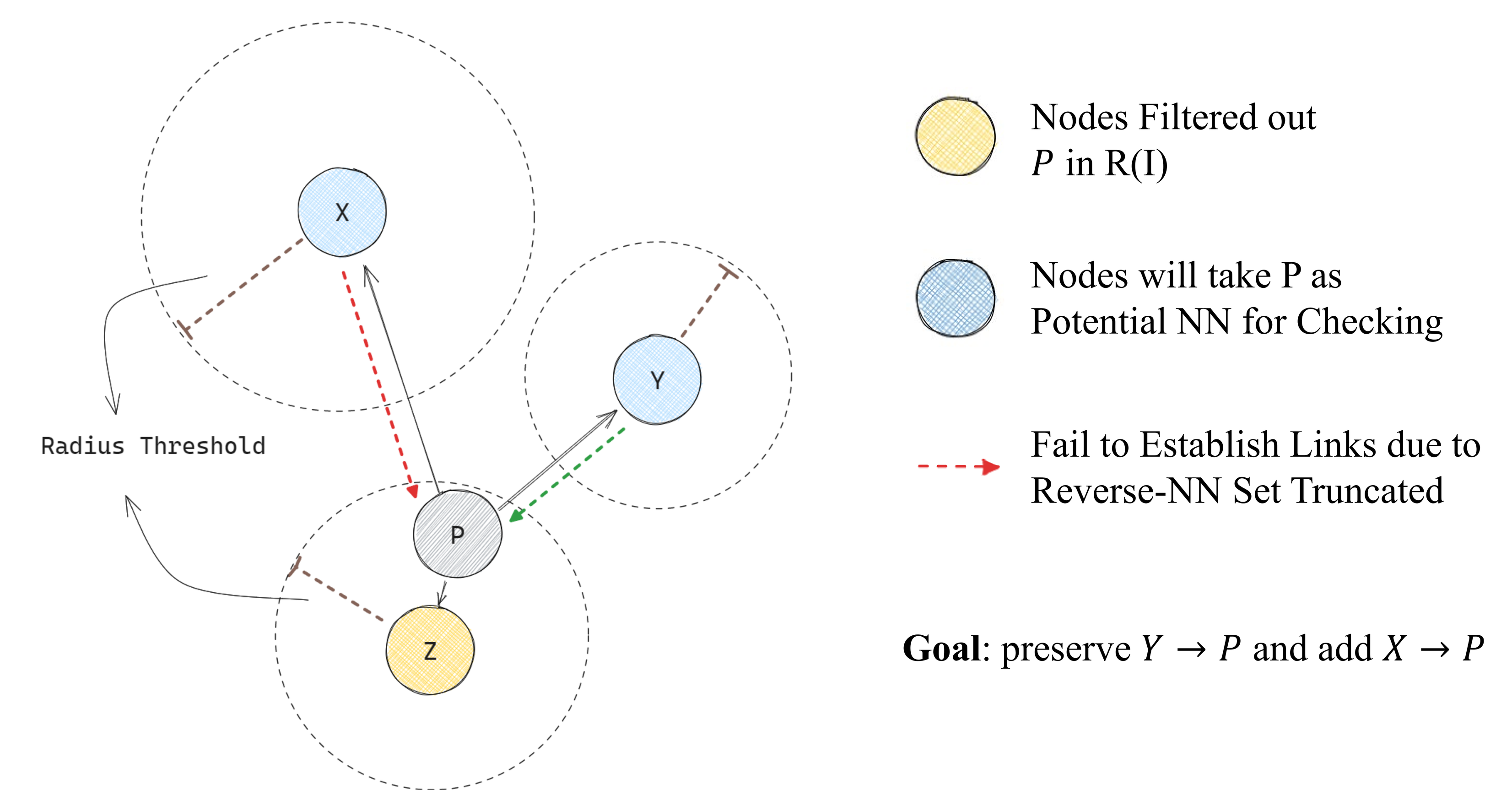
- Candidates Set (C)
- Incoming Nodes (I)
- Reverse-NN Nodes (R)

Overall Procedure:

1. Randomly selecting 100 neighbors for each node.
2. The incoming nodes are recognized as closer NNs in the current round (Random nodes at the beginning). The reverse-NN nodes consist of nodes that have out-degrees to the current one.
3. Pairs generated by Cartesian Product between set $I \cup R(I)$ itself, set $I \cup R(I)$ and set $C \cup R(C)$ for each node are mutually checked whether they constitute the nearest neighbor.

With Tight-Threshold:

The original algorithm utilizes the radius threshold to establish the range for reverse-NN detection and also filter intimate nodes, which are considered well-connected due to their shorter distances.

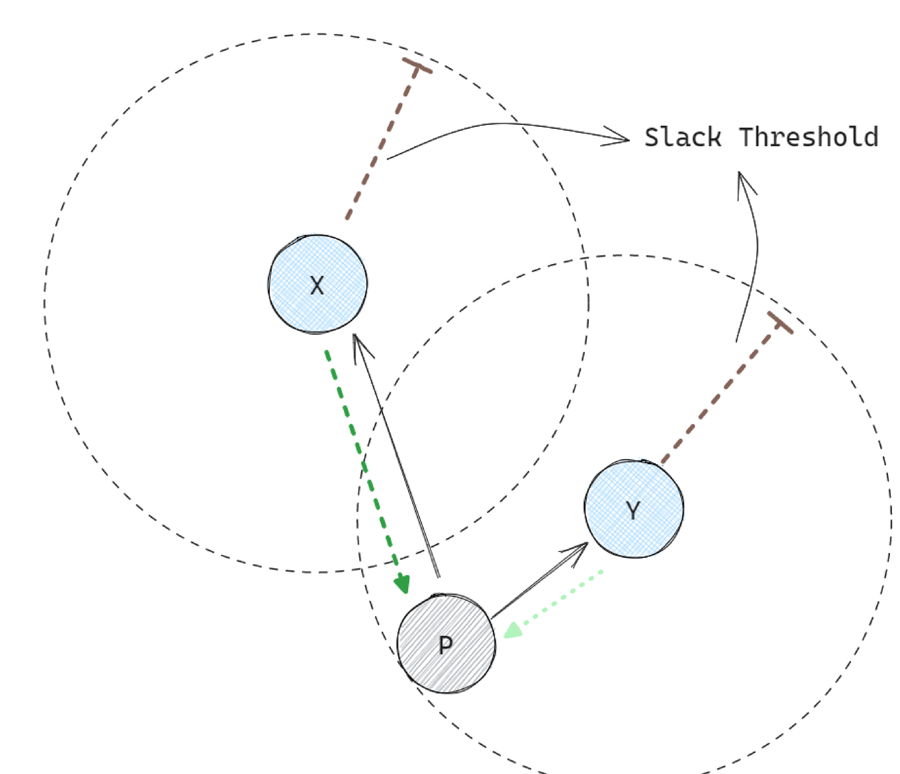


Unfortunately, the curse of dimensionality frequently causes hubs to appear in high-dimensional space, which we also observed in the Turing Dataset during our analysis. This causes the reverse-NN set to be truncated, thereby saving memory budget, but also undermining neighbor discovery through reverse-NN paths.

With Slack-Threshold:

This method is based on two key principles:

- A neighbor of a neighbor is also likely to be a neighbor. (from NN-Descent)
- A closer neighbor could be found by taking a slight detour. (Inspired by the RNG property [2])



With Slack-Threshold, nodes with relatively far distance from P are more likely to include P into its neighbor checking phase.(Node X)

It filters out Node Y. However, by applying local join and relying on principle 1 of NN-Descent, we ensure that local close nodes are detected and well-connected. This approach increases the likelihood of reaching nodes that would otherwise require detours, while also preserving strong local connections.

Low-Level Design:

- The previously mentioned structure will be dynamically maintained to reduce memory footprint.
- SIMD with AVX-512 is used with aligned data for accelerating L2 distance computation.
- Prefetching has been effectively integrated into the computational checking progress.

Results

Results are selected from submissions in the contest.

Dataset	Recall	Time (s)
Turing-10M	**0.981**	**1847**
Turing-10M	0.976	1654
Turing-10M	0.954	1300

References

[1] Dong, Wei, Charikar Moses, and Kai Li. "Efficient k-nearest neighbor graph construction for generic similarity measures." Proceedings of the 20th international conference on World wide web. 2011.

[2] Jaromczyk, Jerzy W., and Godfried T. Toussaint. "Relative neighborhood graphs and their relatives." Proceedings of the IEEE 80.9 (1992): 1502-1517.