VecDB @VLDB2026

Plasma: A Layout-Aware Benchmark Reveals Memory Layout Matters for Graph-based ANNS on GPU

Yutaro Oguri [1] **Mai Nishimura** [2] **Yusuke Matsui** [1]

[1] The University of Tokyo [2] OMRON SINIC X Corporation

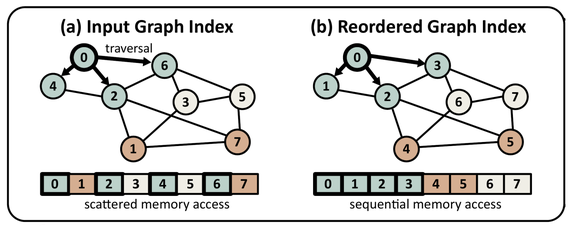
Up to 80% Faster, JUST BY CHANGING Memory Layout

Project Page

1 Graph-based ANNS on GPU

- Core primitive of modern vector search app
- GPU is promising, like CAGRA [Ootomo+, ICDE 2024]
- Bandwidth-bound far more than CPU
- Our scope: **Graph-based x GPU**

2 Memory layout matters



GPU-optimized ANNS speed with enhanced memory utilization

- Effective on CPU execution [Coleman+, NeurIPS 2022]
- Reordering gives co-accessed vertices near IDs
- Scattered access → coalesced
- Proven on CPU, but **still unexplored on GPU**

3 Issue: No fair evaluation platform

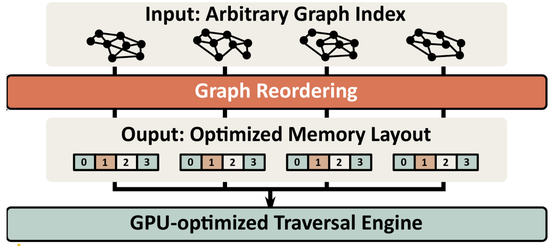
- Each index ships its own search impl. (kernel)
- Topology and implementation are **entangled**
- Existing: focus on, not aware of memory layout

	ANN Benchmark	BigANN	VDB Bench	VIBE	PLASMA
Platform	CPU	CPU	CPU	CPU+GPU	**GPU**
Modern embedding	–	partial	partial	✓	✓
HW profiling	–	–	–	–	✓
Layout-aware	–	–	–	–	✓
Reproduction	docker	docker	docker	apptainer	`pixi`

4 Contributions

- **PLASMA:** Topology decoupled from layout under one unified GPU kernel for the fair evaluation
- **First systematic study:** 4 indices × 12 datasets × 5 orderings
- **Surprise:** Graph Topology < Memory Layout reordering alone up to **+80% QPS**
- **Attribution:** HW profiling reveals DRAM bandwidth, not L1/L2 hit rate, explains well
- **Open release:** `pixi` build for easy reproduction

5 Our proposal: PLASMA



Fair cross-index comparison without implementation bias

- **a** **Graph adapter:** Input adjacency list format
  Enable NSG and Vamana on GPU for the first time
- **b** **Unified kernel:** cuVS-style search kernel
  For fair evaluation independent from impl. hack
- **c** **Layout arrangement:** Reorder vertex ids

6 Experiments

Indices	CAGRA, NSG, Vamana, NN-Descent
Datasets	Modern 12 datasets, dim=96-1536
Reordering	Degree Sort, Hub Sort, GOrder, RCM
Hardware	NVIDIA A100 (80GB), A6000 (48GB)
Metrics	Recall@10, QPS

HEADLINE RESULTS

Up to 80%
QPS gain at equal recall

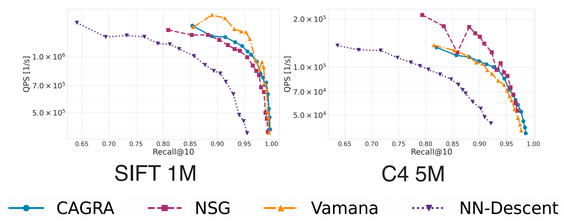
10–30%
typical gain per index–dataset pair

1.2×
CPU-oriented NSG can outperform GPU-native CAGRA

7 Findings

(1) Topology is less significant than expected

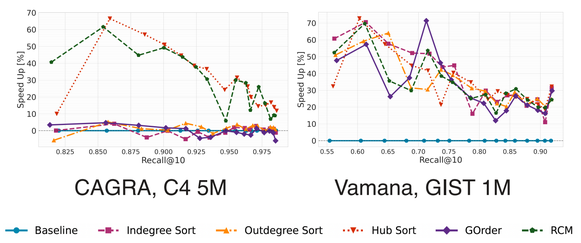
- NSG and Vamana can beat CAGRA
  *CPU-oriented indices running on GPU!
- NSG accelerated by 1.2× on C4 5M
- Gaps widen on high-dimensional embeddings
- Each indices have largely different topologies



Recall@10 - QPS, without reordering. Upper-right is better.

(2) Reordering improves QPS substantially

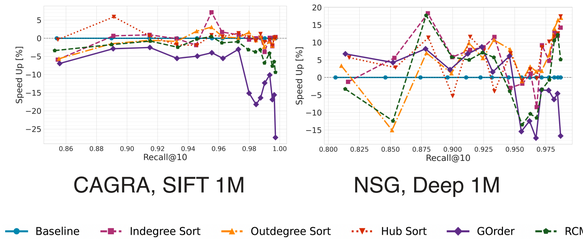
- QPS improvement: Up to +80%
- No one single winner of reordering methods
- Effectiveness depends on search parameters



Speed-up [%] vs. non-reordered baseline with various params

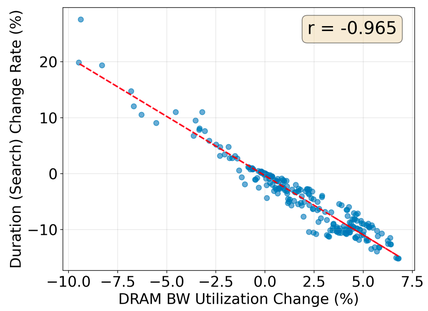
Some failure cases

- QPS improvement is unstable
- No one single winner of reordering methods



(3) DRAM bandwidth explains the speed-up

- L1 hit ≈ 0% (0.007 - 0.025%), L2 hit at 8 - 28%
- Cache hit does not explain well
- DRAM bandwidth tracks the speed-up effectively



One point per (index, dataset, reordering) at a certain parameter

- GPU with higher memory bandwidth benefits more

TAKEAWAY

On GPU ANNS, the memory layout of graph matters more than how nodes are connected

- Extend to latest H100 and B100 architectures
- Can lead to memory layout-aware graph index
- Can extend to online reordering scenario

Visualized results, Code, and Docs: **omron-sinicx.github.io/plasma**

We thank Prof. Hiroaki Shiokawa for valuable comments. Supported by JST, Japan (AIP Acceleration Research JPMJCR23U2, PRESTO JPMJPR2518).