Jasper: ANNS Quantized for Speed, Built for Change on GPU

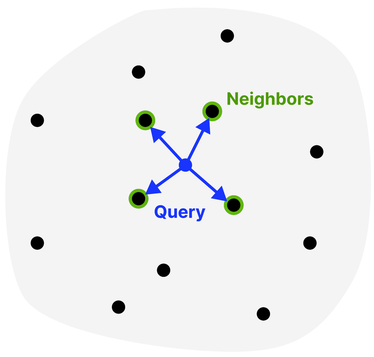
Hunter McCoy, Zikun Wang, Prashant Pandey

{mccoy.hu, wang.zikun, p.pandey}@northeastern.edu

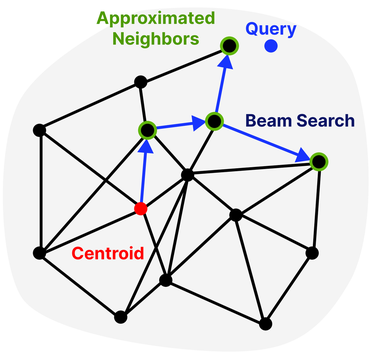
Current state of GPU-accelerated ANN search

Current GPU-accelerated ANNS indices face three key **limitations**:

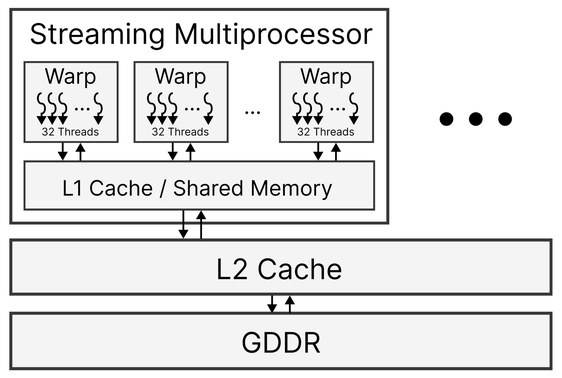
1. Lack the capacity for **incremental updates** as new data arrives.
2. Lack of efficient vector quantization method that addresses memory constraint while **avoid random memory access**.
3. Reduced performance due to **data-dependent memory accesses** during beam search.



Nearest Neighbor Problem



Graph-based Index for Approximate Nearest Neighbor



GPU Architecture

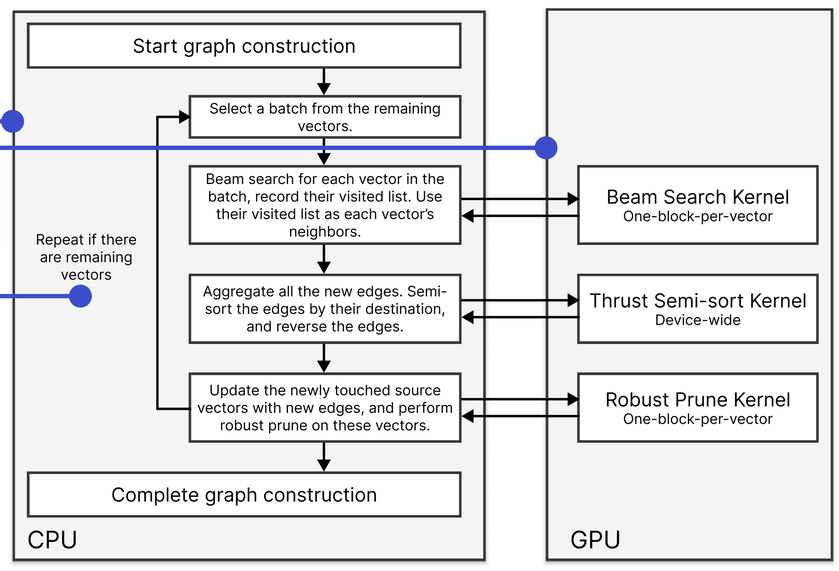
Building high performance ANNS index on GPU with full incremental updates

Jasper implements a **GPU graph index** based on DiskANN construction and query operations, fully utilizing modern GPU performance.

Jasper uses a **lock-free batch-parallel construction algorithm** optimized for block-level parallelism in GPUs, achieving best-in-class construction performance.

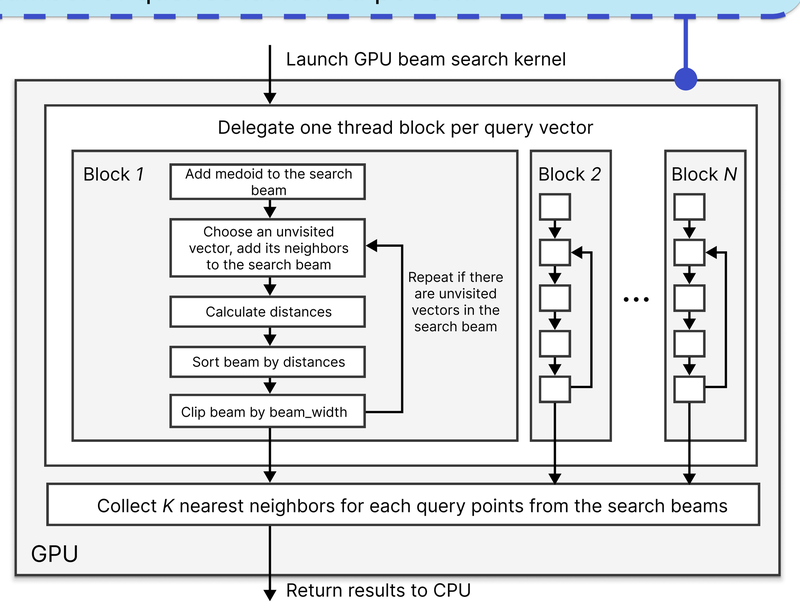
Through **kernel fusion**, Jasper minimizes the overhead of repeated kernel launch, and we can accelerate memory accesses by keeping all the information in shared memory.

Jasper supports **efficient incremental updates**. Each batch iteration performs a bounded calculation over a small subset of vertices, minimizing the memory overhead during construction.



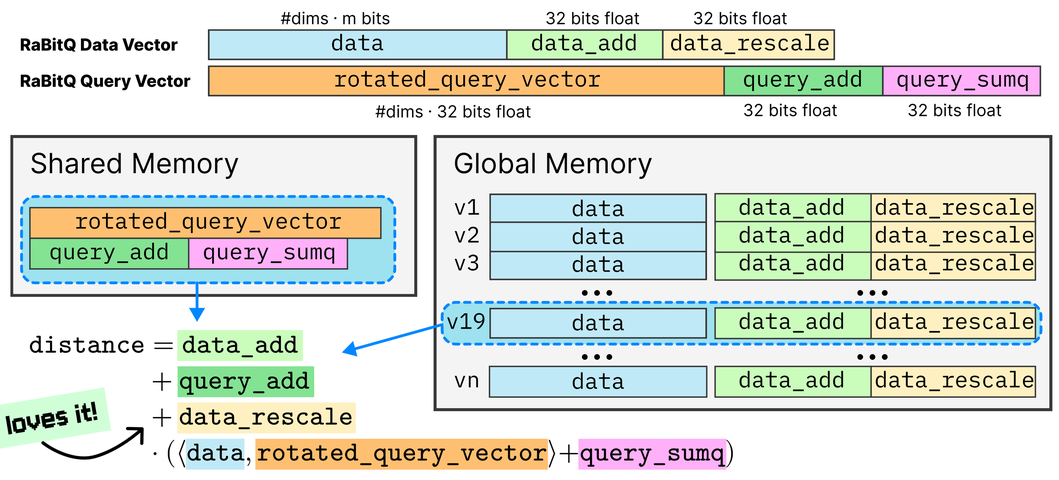
Batched index construction

Jasper offers a **lightweight beam search kernel** on GPU that strips all unnecessary components to maximize the number of queries launched per SM.

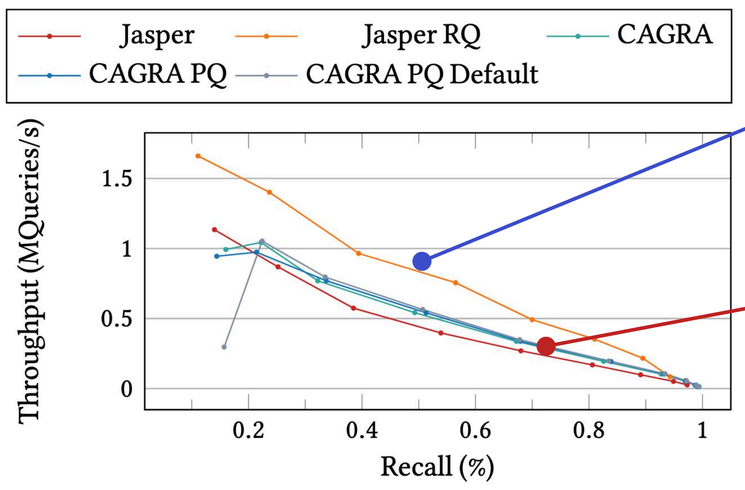


Beam search kernel

Jasper uses **RaBitQ** for vector quantization that fully utilizes GPU hardware advantages, with no random memory lookup during distance estimation, achieving both high accuracy and performance.



Efficient distance estimation using RaBitQ

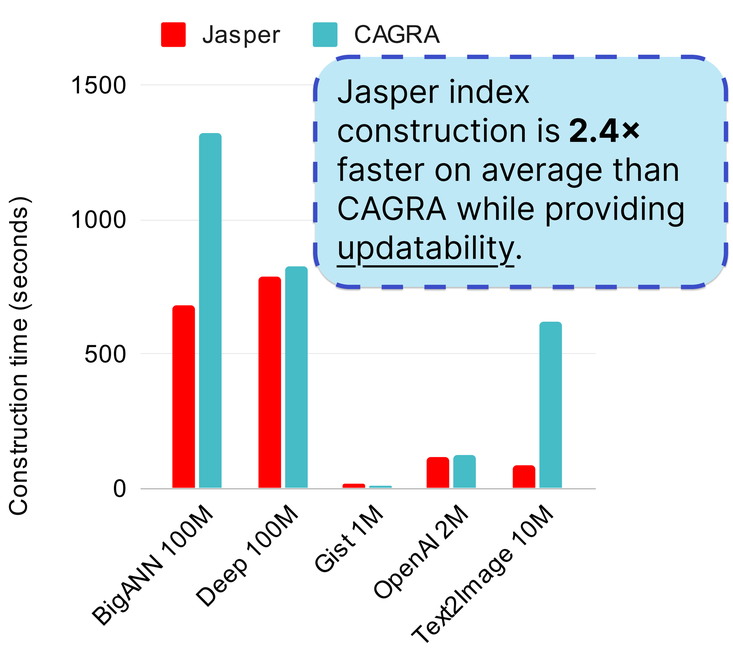


RaBitQ saves 8 times memory usage (4 bits per dimension instead of 32 bits) while achieving higher performance.

Even through **PQ** saves Cagra memory usage, it suffers performance losses due to memory indirections.

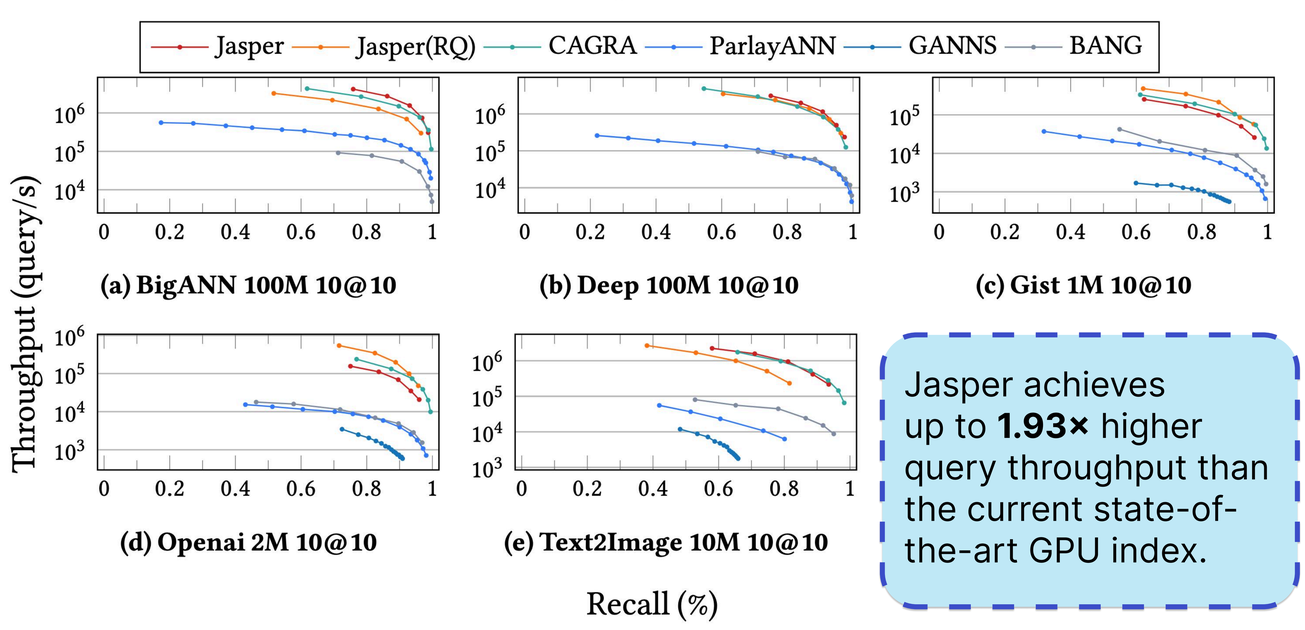
Comparison between RaBitQ and Product Quantization
Dataset: Gist 1M 1@1

Fast index construction and query throughput



Jasper index construction is **2.4×** faster on average than CAGRA while providing updatability.

Index construction speed



Jasper achieves up to **1.93×** higher query throughput than the current state-of-the-art GPU index.

Query recall/throughput curves

Github Repository

All experiments are done on machine with Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz (14 cores) CPU and a NVIDIA A100 GPU with 80GB of HBM memory using CUDA 12.9.