

大规模向量存储系统研究综述



Survey on Large-Scale Vector Storage Systems

郭昊/Guo Hao¹, 谢旻晖/Xie Minhui², 陆游游/Lu Youyou¹

(1. 清华大学, 中国 北京 100084;

2. 南京大学, 中国 南京 210023)

(1. Tsinghua University, Beijing 100084, China;

2. Nanjing University, Nanjing 210023, China)

DOI: 10.12142/ZTETJ.202604007

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260828.1138.002>

网络出版日期: 2026-08-28

收稿日期: 2026-06-05

摘要: 大规模向量存储系统作为面向海量非结构化数据的高维向量表示载体, 是提供高效存储与检索能力的核心基础设施。针对数据规模激增所带来的内存容量瓶颈与成本压力, 聚焦于大规模向量索引与检索系统的演进路径。首先, 概述内存图索引与聚类索引的基本机制; 随后, 重点分析十亿级乃至更大规模场景下的向量量化压缩、基于固态硬盘(SSD)的索引架构以及异构存储硬件加速技术, 并从精度、延迟、吞吐量、构建更新效率及资源开销等维度对比各方案的差异; 进一步, 探讨向量检索与检索增强生成(RAG)的端到端协同优化; 最后, 总结不同技术路线的适用场景、主要瓶颈与未来研究方向。

关键词: 大规模向量存储; 近似最近邻检索; SSD; RAG

Abstract: Large-scale vector storage systems serve as the core infrastructure for efficiently storing and retrieving high-dimensional representations of massive unstructured datasets. To address the memory capacity bottlenecks and cost pressures arising from the explosive growth of data volumes, the evolution of large-scale vector indexing and retrieval systems is reviewed. In-memory graph-based and clustering-based indexing methods are first outlined. Then, vector quantization, solid-state drive (SSD)-based index architectures, and heterogeneous storage-computing hardware acceleration for billion-scale and even larger datasets are examined. These approaches are compared in terms of accuracy, latency, throughput, construction/update costs, and resource overhead. End-to-end collaborative optimization between vector retrieval and Retrieval-Augmented Generation (RAG) is further discussed. Finally, applicable workloads, major bottlenecks, and future research directions are summarized.

Keywords: large-scale vector storage; approximate nearest neighbor search; SSD; RAG

引用格式: 郭昊, 谢旻晖, 陆游游. 大规模向量存储系统研究综述 [J]. 中兴通讯技术, 2026, 32(4): 41-46. DOI: 10.12142/ZTETJ.202604007

Citation: Guo H, Xie M H, Lu Y Y. Survey on large-scale vector storage systems [J]. ZTE technology journal, 2026, 32(4): 41-46. DOI: 10.12142/ZTETJ.202604007

随着人工智能大模型的持续发展, 向量正逐步成为一种新兴的数据表示范式。文本、图像、视频等非结构化数据经嵌入模型处理后, 可转化为数百至数千维的高维向量。语义相近的数据在向量空间中呈现相近的几何距离, 这为基于语义的向量检索带来了新的可能性。在此过程中, 向量存储系统需根据查询所生成的嵌入向量, 在海量数据集中检索并返回若干个最近邻向量, 同时还需支持高吞吐的向量更新等操作。

当数据规模扩展至百万级或更高时, 精确向量检索的计

算开销呈指数增长, 查询延迟往往高达秒级以上。传统的 K-D 树等精确检索方法在低维场景下表现良好, 但在高维场景中易受“维度灾难”影响, 导致空间划分失效, 搜索效率退化至与暴力扫描相当。为此, 学术界与工业界普遍采用近似最近邻检索 (ANNS)。该方法以极小的召回精度损失为代价, 实现检索效率的数量级提升, 将单次查询延迟降低至毫秒级甚至更短。

1 向量存储的主流索引算法

随着数据规模的增长, 百万级别的向量通常占据百 MB 至 GB 级存储空间, 仍可完全驻留于内存。因此, 传统向量索引大多基于内存构建, 主要分为基于图的索引与基于聚类的索引两类。

基金项目: 国家重点研发计划项目 (2024YFE0203300); 国家自然科学基金项目 (62332011, 62502521); 北京市自然科学基金项目 (L242016, 4254083); 北京市科学技术协会青年人才托举工程项目 (20250982); 霍英东教育基金会高等院校青年教师基金项目; CCF-腾讯犀牛鸟科研基金项目

1.1 基于图的向量索引

图索引结构将所有向量组织为一张有向图，每个向量作为图节点，节点间通过边相连。其优势在于支持细粒度的图上游走，能够在保证高召回率的同时实现低检索延迟。为保证连通性与检索效率，图索引既需包含距离相近节点间的短边，以便在逼近目标时实现精确收敛，也需包含跨度较大的长边，以便在搜索初期快速拉近与查询向量的距离。

分层可导航小世界图 (HNSW)^[1]、导航扩展图 (NSG)^[2]及 DiskANN^[3]等主流图索引采用了不同的选边策略。基础策略通常依赖于三角不等式，即对最长边进行裁剪。例如，对于即将插入的向量 A ，若 A 到 B 的边长大于 A 到 C 与 B 到 C 的边长之和，则该边会被移出 A 的邻居集合。HNSW 整体采用该策略，并额外引入多层级结构以增加长边连接：最底层包含全量向量，每向上一层，节点规模按固定比例（如 1%）指数级递减。检索时自顶向下逐层贪心路由，直至底层局部最优。相比之下，NSG 和 DiskANN 采用单层图结构，其中 DiskANN 进一步放宽了三角不等式的限制，允许长度小于其他边 1.2 倍的长边保留，以增强图的连通性。

尽管图拓扑结构各异，其查询过程大体遵循启发式贪心遍历策略：从入口节点出发，将邻居按与查询的距离加入候选池，反复扩展当前最优的未访问节点，直至候选集收敛。传统方法通常需要对被访问节点的绝大多数邻居执行精确距离计算；分区极值次序统计量 (PEOs)^[4]则利用空间分区与随机投影构造带概率保证的路由判定，仅对更有可能改善搜索结果的邻居计算精确距离。原文实验表明，将 PEOs 接入 HNSW 和 NSG 后，在不同召回率条件下可将查询吞吐量提升 1.6 ~ 2.5 倍，体现了其相较启发式邻居剪枝方法更具可分析性与稳定性。

图索引虽检索性能优异，但其构建与更新受节点间依赖关系影响，开销较大。HNSW 与新版本 DiskANN 采用逐点插入方式：每加入一个新节点，需在现有图中搜索候选邻居，并按度数上限 R 进行裁剪；NSG 与初代 DiskANN 则以 k 近邻 (kNN) 图或随机连通图为起点，通过全局重构优化邻居连接。上述两类方法均需反复执行邻居搜索与选边操作，构建成本随数据规模增长而快速上升。

1.2 基于聚类的向量索引

聚类索引通过 K-Means 等方法将向量空间划分为若干簇，并为每个簇提取中心向量。查询时，先定位距离最近的若干中心，再扫描相应簇内的候选向量；典型工作包括倒排文件索引 (IVF)^[5]与 IVF-分层 HNSW^[6]，后者采用图索引管理聚类中心，以降低路由开销并支持更细粒度的空间划分。

传统 IVF 通常将每个向量分配至单一簇，边界向量易被漏检；基于正交增强残差的向量副本方法 (SOAR)^[7]则利用正交放大残差损失为向量选择冗余分区，使次级分区专门补偿主分区失效的查询方向。与简单复制到第二近簇或独立训练多个索引相比，SOAR 侧重于降低不同分区失败事件之间的相关性，在较低额外存储开销下改善了召回率与扫描效率。

聚类索引的构建主要依赖 K-Means 迭代过程：聚类数较少时可通过采样快速训练，超大规模聚类则需借助层次化或平衡聚类来降低多轮全量扫描的开销。与图索引相比，聚类之间的数据依赖较弱，更便于并行构建与局部更新，但其检索质量更易受到聚类边界效应的影响。

2 大规模向量存储系统

在真实生产环境中（如在线大模型服务的全网数据检索），向量规模通常高达十亿乃至千亿级别。将如此海量的高维数据全量驻留于单机内存，不仅成本高昂，亦受限于物理容量。当前主流的扩展思路有两种：一是采用精度有损的向量量化压缩技术；二是将数据下沉至闪存固态硬盘 (SSD) 等大容量非易失性存储介质中。

2.1 向量量化

向量量化是一种有损压缩技术，不仅能成倍缩减存储占用，还能显著降低单次距离计算的算力开销。量化技术与上述图/聚类检索方法相互正交，既可独立应用于每个向量，也可嵌入索引遍历过程中以加速距离测算。常用量化技术主要分为标量量化 (SQ) 与乘积量化 (PQ) 两类。

SQ 将向量每一维度的浮点值线性映射至固定区间。例如，8-bit 量化将极值区间划分为 256 段，用整型值替代原始浮点数。RaBitQ^[8]提出一种改进型 SQ 量化方法，通过随机正交变换与残差量化降低精度损失，并利用单指令多数据流 (SIMD) 指令集加速扫描过程。总体而言，标量量化计算速度极快，但固定步长的离散化方式会引入一定程度的精度误差。

PQ 则将高维向量沿维度切分为若干子段，在每个子段内独立进行 K-Means 聚类（如设定 256 个中心），随后将向量编码为对应子段聚类中心的索引号。检索时，通过预先计算查询向量各子段到所有聚类中心的距离，并构建非对称距离表，即可通过查表求和快速估算近似距离。相较于标量量化，PQ 在同等压缩率下具有更高的保真度，但查表操作破坏了内存连续访问模式，使得单步计算开销略高于基于 SIMD 的并行计算方法。ScaNN^[9]进一步将空间分区与各向异

性向量化相结合：其不再对所有重构误差进行等权处理，而是对与原始向量方向平行的残差施加更大惩罚，从而减少最大内积搜索中的排序误差。与传统 PQ 主要致力于最小化全局重构误差不同，ScaNN 直接围绕检索目标优化量化损失，通过分区缩小候选集，并借助量化加速候选向量的打分过程。

2.2 盘上向量存储——基于图索引

除了对向量进行压缩之外，部分研究还通过引入 SSD 等非易失性存储介质来扩展容量。相较于内存，SSD 具有更大的存储容量和更低的单位成本，能够支持十亿乃至百亿级别的向量数据。然而，SSD 的读写吞吐量显著低于内存，且读写延迟高出数个数量级。现有基于 SSD 的向量存储系统主要围绕这两方面进行优化。

在降低输入/输出（I/O）读写量与提升吞吐方面，DiskANN^[9]采用了内存与 SSD 协同的设计：在内存中仅缓存所有向量的压缩量化版本（如 PQ 码），而将完整的图索引拓扑结构与全量高维向量存储于 SSD 上。检索时，针对邻居向量的距离比较与图遍历可直接在内存中的量化向量上完成，系统仅在读取节点的邻居编号及进行向量重排时访问 SSD，无需读取大量邻居向量的原始数据，从而显著降低了整体 I/O 读取量。Starling^[10]在此基础上进一步优化了磁盘数据布局，通过图节点重排技术，将图结构中彼此相邻、极易被共同访问的节点聚集存放于 SSD 的同一数据页面或相邻块中。这种设计大幅提升了单次 I/O 的数据命中率，减少了磁盘带宽的浪费。

在缓解高 I/O 延迟方面，DiskANN 摒弃了传统的逐节点贪心搜索机制，转而采用集束搜索。该方法在每一轮迭代中并发探索多个候选节点，向 SSD 一次性发出多个并行 I/O 请求，充分利用 SSD 内部并发度来均摊单次 I/O 延迟。PipeANN^[11]则进一步放宽了贪心搜索的约束，仅探索当前局部最优节点，允许每次探索时存在尚未完成的节点读取操作，而不必等待这些读取完成。通过这种方式，PipeANN 消除了步间依赖关系，使得 SSD I/O 与计算能够重叠并流水线执行，从而有效掩盖 I/O 延迟。

除了向量检索外，向量存储还需支持数据动态更新，以适应不断变化的向量数据集。针对图向量索引的动态更新，现有方法主要分为两类：增量索引合并与直接插入。

增量索引合并的代表系统为 FreshDiskANN^[12]。其在内存中开辟独立空间，维护一个动态增量图以临时容纳新插入的向量。当内存缓冲区达到容量上限后，系统通过后台操作将增量图批量合并至 SSD 的主图索引中。该方式能有效减少写

放大，但合并过程中产生的突发 I/O 可能干扰前台检索请求。直接插入的代表系统为 OdinANN^[13]。其直接将新向量插入磁盘上的全量图索引中，将插入过程的 I/O 开销均匀摊薄至持续操作流中，使得前台查询延迟在持续插入场景下更为稳定，但代价是写放大较高。

2.3 盘上向量存储——基于聚类索引

将传统聚类索引迁移至 SSD 是一种直观的扩展思路。该系统在内存中维护聚类中心表，并将各聚类内的向量存储于 SSD。检索时，通过内存中的聚类中心定位到距离最近的若干个聚类，再从 SSD 中并行读取相应聚类内的向量进行比较。该方式虽然具备可扩展性，但仍面临两方面的性能瓶颈：一方面，传统聚类中心数量较少，即使单次检索只查询少数聚类，仍需读取大量聚类内向量，导致 I/O 开销较大；另一方面，聚类算法本身效率较低，K-Means 聚类需流式读取全量数据。当数据规模达到十亿级时，聚类复杂度显著上升，若内存容量有限，聚类迭代过程需进行数遍乃至数十遍全量数据扫盘，计算与 I/O 开销均十分庞大。

针对上述问题，SPANN（微软的系统名）^[14]提出了一种海量聚类方法，每个聚类仅包含数十个向量（远小于传统倒排文件索引 IVF 中数千至数万个向量的规模），并在内存中维护一个规模约为总向量数 1/10 的聚类中心导航图，以快速定位至目标聚类中心。通过该架构，SPANN 对单个聚类进行遍历的 I/O 开销显著降低，使其能够在单次查询中访问多个精细聚类以提升搜索精度，同时保留聚类索引并行读取的特性，从而保持较低的查询延迟。为缓解海量聚类收敛慢的问题，SPANN 采用层次化平衡聚类策略，通过层次聚类加速收敛过程，并确保各聚类长度相对均衡。针对聚类索引的边界效应，SPANN 引入副本机制，将边界向量复制并存储于多个相邻倒排列表中，从而提升系统召回率。在此基础上，SPFresh^[15]进一步设计了聚类分裂与合并机制，以支持索引的动态更新。

2.4 基于异构存算硬件的向量存储

除基于中央处理器（CPU）和 SSD 的架构外，另有研究探索在混合内存、图形处理器（GPU）等场景下进行大规模向量检索，以突破 CPU 的算力瓶颈和 SSD 的带宽限制。

图向量索引方面，基于异构内存的近似最近邻搜索算法（HM-ANN）^[16]针对异构内存的图索引进行了探索。HM-ANN 利用了持久性内存大容量与动态随机存取存储器（DRAM）高速度的物理特性，设计了一种层次化的图索引放置策略。HM-ANN 将图索引的上层导航节点存放在

DRAM中以保证高速的初期路由，而将底层全量图拓扑及未压缩的原始高维向量持久化于非易失内存中。这种设计不仅规避了量化带来的精度损失，还能在单机架构下无损支撑十亿级向量检索。GustANN^[17]则针对GPU-SSD结构的向量检索，提出了一种以GPU为中心、CPU辅助的架构。与传统的基于GPU直访的I/O不同，GustANN利用CPU进行I/O操作，通过仅将需要的部分传输到GPU中以缓解外围组件互连高速总线（PCIe）带宽瓶颈。与此同时，GustANN设计了一种GPU高效的检索内核，以减少搜索过程的分支。

聚类向量索引方面，内存处理近似最近邻搜索系统（PIMANN）^[18-19]首次将存内计算技术应用于向量检索以应对带宽瓶颈。传统聚类检索在扫描海量倒排列表时极易耗尽内存带宽，而PIMANN通过将距离比较逻辑直接卸载至靠近存储单元的存内处理（PIM）协处理器，利用其提供的极高内部聚合内存带宽，大幅减少了向量数据在内存总线与CPU之间的搬移开销，从而加速了聚类内部的距离扫描过程。FusionANNS^[20]则提出了一种面向CPU、GPU与SSD的三级异构协同架构。FusionANNS在SSD上持久化存储全量原始向量，在GPU的高带宽内存中缓存PQ量化向量，同时在CPU主机内存中仅维护轻量级的聚类导航图与倒排链表ID。检索时，CPU首先在内存中快速定位至最近的若干聚类，随后仅将向量ID传输至GPU；GPU利用其庞大的并发线程与高带宽存储器（HBM）带宽完成大规模近似距离计算与过滤；最后，系统根据GPU的筛选结果，从SSD中读取极少量原始向量进行精确重排。该协同机制充分利用了GPU的算力以及CPU在分支处理上的优势，以较低的硬件成本兼顾了高吞吐、低延迟与高精度。

表1从精度、查询性能、构建更新和资源开销4个方面对代表性技术路线进行了比较，其中“高”“中”“低”表示在相近召回率与硬件预算条件下的相对水平。

因此，内存方案优先服务于容量可控且追求极低延迟与高吞吐的负载；盘上图索引更适用于查询密集、更新较少的数据集；盘上聚类索引则更适合持续写入与批量更新的场景；异构方案以更复杂的数据放置与调度策略为代价，换取更高的查询吞吐量。

3 向量存储与RAG的协同优化

从端到端的视角来看，检索增强生成（RAG）系统包含离线或增量的数据处理链路以及在线查询链路。以用户上传PDF为例，系统需先完成解析或光学字符识别（OCR）、清洗与分块，再由嵌入模型生成向量并构建或增量更新索引；在线阶段则依次执行查询编码、近似最近邻检索、候选重排序与上下文组装，最后将检索结果送入生成模型。向量检索位于两条链路的交汇点，其召回率、更新时效和尾延迟会直接影响后续重排质量、上下文长度及端到端响应时间。

上述环节之间存在明显的耦合关系：图索引、聚类索引、SSD布局与量化精度共同决定候选质量与检索时延；热点文档块、聚类中心及查询结果可跨请求缓存；CPU侧的解析检索与GPU侧的嵌入生成需统一调度，以减少数据搬移并实现各阶段执行的重叠。因此，RAG系统的协同优化不应仅追求向量索引的独立吞吐量，而应在质量约束下，联合优化数据摄取、索引更新、在线检索、重排序、缓存与模型推理的端到端尾延迟及资源成本^[21-22]。

检索增强生成优化器（RAGO）^[21]从全局资源调度与配置的角度进行协同优化。实际部署的RAG系统包含多样的配置（如不同的数据库规模、是否包含文档编码器、查询重写器和结果重排器等）。为此，RAGO提出了一种结构化的负载抽象框架RAGSchema，用来统一描述这些复杂的应用流水线与特征。基于这种抽象，RAGO能够在系统资源受限的前提下，自动探索并搜索出最优的调度策略，这包括：决

表1 大规模向量索引与检索技术路线比较

技术路线	精度	查询延迟与吞吐	构建与更新	内存/存储开销	适用负载与主要瓶颈
内存索引及量化(HNSW、NSG、ScANN + PQ、RaBitQ)	中—高;受量化率影响	延迟低、吞吐高	量化快;图构建偏慢,更新依基础索引	DRAM高;量化可压缩	适合极低延迟、极高吞吐的负载;瓶颈为内存容量
盘上图索引(DiskANN、PipeANN)	高	延迟较低、吞吐受限于盘随机访问IOPS	构建和增量更新开销高	DRAM低;SSD容量开销高	适合查询密集的大规模负载;瓶颈是随机IOPS
盘上聚类索引(SPANN、SP-Fresh)	中	延迟较低,I/O并发好;吞吐低	构建和增量更新相对友好	DRAM低;SSD存储开销高	适合更新密集的大规模负载;瓶颈是扫描带宽与边界效应
异构硬件加速(GustANN、PIMANN、FusionANNS)	取决于底层索引	吞吐高、延迟较高(受PCIe传输影响)	数据布局和预处理复杂,动态更新较难	需HBM、持久性内存或PIM等专用资源	适合稳定高吞吐负载;瓶颈是PCIe/内存带宽与数据搬移
ANN: 近似最近邻 DRAM: 动态随机存取存储器 HBM: 高带宽存储器 HNSW: 分层可导航小世界图		IOPS: 每秒输入/输出操作次数 NSG: 导航扩展图 PCIe: 外围组件互连高速总线 PIM: 存内计算		PIMANN: 存内计算近似最近邻搜索 PQ: 乘积量化 ScANN: 可扩展最近邻搜索 SPANN: 微软的系统名	
				SSD: 固态硬盘	

定各个模型组件是分离部署还是混合部署任务放置、各阶段分配多少计算资源,以及采用多大的批处理等。

HedraRAG^[22]针对细粒度流水线进行协同优化。面对多跳推理、迭代检索等高度异构且复杂的多阶段RAG workflow, HedraRAG将整个处理过程抽象为图结构。为避免检索与生成相互阻塞, HedraRAG通过动态图变换(如节点拆分)将大任务切分为耗时相近的微阶段,使CPU端的检索与GPU端的生成形成高效的并行流水线。同时,利用复杂请求在多个阶段间的语义相似性进行推测执行:在完整检索结果返回之前,大模型可借助早期获得的部分检索结果提前启动生成过程;若最终检索结果一致,则直接采纳,从而有效掩盖检索延迟。此外, HedraRAG还通过缓存热门聚类来提高跨请求的检索效率。

4 未来趋势展望

预计在未来较长一段时期内,图索引与聚类索引仍将保持共存。图索引的优势在于能够通过精细的图结构实现高精度与低延迟,但其构建过程算力开销大、节点依赖性强,且拓扑结构复杂,导致扩展困难、更新成本高。相比之下,聚类索引结构简单,聚类间无数据依赖,构建与更新易于实现,但检索精度相对较低。预期未来研究将围绕这两类索引分别展开:一方面,打破图索引的强依赖关系,进一步提升其构建性能与水平扩展能力;另一方面,优化聚类划分策略,在保持高扩展性的同时提高聚类索引的检索精度。

随着大模型应用进一步发展,向量存储系统也面临新的挑战,未来主要呈现以下3个技术趋势:

1) 超大规模检索。随着嵌入模型的演进,对单个数据的表征正从传统的单向量表示转向多向量表征。加之数据规模的持续增长,向量检索的规模预计将从十亿级扩展至千亿级乃至万亿级。以全网搜索等场景为例,其总数据规模已达千亿级别。面向十亿级规模的单机SSD检索方案难以满足千亿级规模下PB级索引的存储需求。如何利用新型网络与存储硬件实现分布式构建与检索,在大规模条件下兼顾精度与性能,是亟待解决的新挑战。

2) 混合检索。RAG对底层存储的需求已不再局限于纯粹的语义向量相似度计算,实际业务场景往往需要复合过滤条件,例如“仅从权威来源中筛选”或“优先检索最新数据”。这要求未来的向量存储系统具备高效的混合检索能力。如何在保证向量检索召回率的前提下,将倒排索引、列式标量存储等技术与向量检索有效结合,以支持多维度的标量筛选,是提升RAG系统准确性与合规性的关键所在。

3) 向量存储-应用协同优化。以“边想边搜”及 Deep

Research 长程推理等为代表的新型AI负载,呈现出高度动态、多轮迭代和多跳检索的语义特征,使得检索与大模型生成之间不再是简单的串行执行关系。向量存储系统需针对此类负载进行定制化设计,例如:感知多轮对话中的语义变化以支持索引预取,针对多跳推理的中间状态进行缓存优化等。通过存储层与应用层的协同设计,可有效掩盖I/O与检索延迟,从而提升端到端的响应速度。

5 结束语

向量存储作为人工智能的关键支撑技术,正经历从单一算法优化向全栈系统协同乃至应用协同设计的演进。从内存态的图与聚类算法,到基于大容量固态盘的盘上架构,再到异构软硬件加速,向量存储系统已能够以较低成本实现十亿级别的高效检索。展望未来,向量存储仍面临超大规模检索、混合检索以及向量存储-应用协同优化等关键挑战。通过攻克这些挑战,向量存储将为下一代通用人工智能提供更为坚实、智能的数据底座。

参考文献

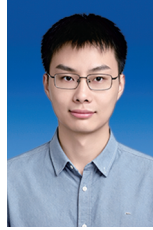
- [1] Malkov Y A, Yashunin D A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs [J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 42(4): 824-836. DOI: 10.1109/TPAMI.2018.2889473
- [2] Fu C, Xiang C, Wang C X, et al. Fast approximate nearest neighbor search with the navigating spreading-out graph [J]. Proceedings of the VLDB endowment, 2019, 12(5): 461-474. DOI: 10.14778/3303753.3303754
- [3] Jayaram Subramanya S, Devvrit F, Simhadri H V, et al. DiskANN: fast accurate billion-point nearest neighbor search on a single node [EB/OL]. (2019-12-08)[2026-06-12]. <https://dl.acm.org/doi/abs/10.5555/3454287.3455520>
- [4] Lu K, Xiao C, Ishikawa Y. Probabilistic routing for graph-based approximate nearest neighbor search [PP/OL]. arXiv(2024-02-17)[2026-06-15]. <https://arxiv.org/abs/2402.11354>
- [5] Babenko A, Lempitsky V. The inverted multi-index [J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(6): 1247-1260
- [6] Baranchuk D, Babenko A, Malkov Y. Revisiting the inverted indices for billion-scale approximate nearest neighbors [PP/OL]. arXiv(2018-07-23) [2026-06-15]. <https://arxiv.org/abs/1802.02422>
- [7] Sun P, Simcha D, Dopson D, et al. SOAR: improved indexing for approximate nearest neighbor search [PP/OL]. arXiv(2024-05-31)[2026-06-14]. <https://arxiv.org/abs/2404.00774>
- [8] Gao J Y, Long C. RaBitQ: quantizing high-dimensional vectors with a theoretical error bound for approximate nearest neighbor search [J]. Proceedings of the ACM on management of data, 2024, 2(3): 1-27. DOI: 10.1145/3654970
- [9] Guo R, Sun P, Lindgren E, et al. Accelerating large-scale inference with anisotropic vector quantization [PP/OL]. arXiv (2019-08-27)[2020-12-04]. <https://arxiv.org/abs/1908.10396>
- [10] Wang M, Xu W, Yi X, et al. Starling: an I/O-efficient disk-

- resident graph index framework for high-dimensional vector similarity search on data segment [J]. Proceedings of the ACM on management of data, 2024, 2(1): 1–27
- [11] Guo H, Lu Y. Achieving low-latency graph-based vector search via aligning best-first search algorithm with SSD [EB/OL]. (2025-07-07) [2026-06-14]. <https://www.usenix.org/system/files/osdi25-guo.pdf>
- [12] Singh A, Subramanya S J, Krishnaswamy R, et al. FreshDiskANN: a fast and accurate graph-based ANN index for streaming similarity search [PP/OL]. arXiv(2021-05-20) [2026-06-18]. <https://arxiv.org/abs/2105.09613>
- [13] Guo H, Lu Y. OdinANN: direct insert for consistently stable performance in billion-scale graph-based vector search [EB/OL]. [2026-06-14]. <https://storage.cs.tsinghua.edu.cn/papers/fast26-odinann.pdf>
- [14] Chen Q, Zhao B, Wang H, et al. SPANN: highly-efficient billion-scale approximate nearest neighborhood search [PP/OL]. arXiv (2021-11-05)[2026-06-11]. <https://arxiv.org/abs/2111.08566>
- [15] Xu Y M, Liang H Y, Li J, et al. SPFresh: incremental in-place update for billion-scale vector search [C]//Proceedings of the 29th Symposium on Operating Systems Principles. ACM, 2023: 545–561. DOI: 10.1145/3600006.3613166
- [16] Somboonmark W, Chumnuat J. Enhancing AQI forecasting accuracy: integrating ARIMA, ANN, and regression techniques with the development of HM4AQI web application [J]. Journal of big data, 2026, 13(1): 10672–10684. DOI: 10.1186/s40537-026-01410-y
- [17] Jiang H D, Guo H, Xie M H, et al. High-throughput, cost-effective billion-scale vector search with a single GPU [J]. Proceedings of the ACM on management of data, 2025, 3(6): 1–27. DOI: 10.1145/3769799
- [18] Wu P Q, Xie M H, Zhao E R, et al. Turbocharge ANNS on real processing-in-memory by enabling fine-grained per-PIM-core scheduling [C]//Proceedings of the 2025 USENIX Conference on Usenix Annual Technical Conference. ACM, 2025: 1223–1241. DOI: 10.5555/3768039.3768111
- [19] Wu P Q, Xie M H, Zhao E R, et al. Unlocking PIM-core capabilities for efficient ANNS on commodity processing-in-memory hardware [J]. ACM transactions on storage, 2026, 22 (3): 1–39. DOI: 10.1145/3806055
- [20] Tian B, Liu H K, Tang Y H, et al. Towards high-throughput and low-latency billion-scale vector search via CPU/GPU collaborative filtering and re-ranking [C]//Proceedings of the 23rd USENIX Conference on File and Storage Technologies. ACM, 2025: 171–185. DOI: 10.5555/3724648.3724659
- [21] Jiang W Q, Subramanian S, Graves C, et al. RAGO: systematic performance optimization for retrieval-augmented generation serving [C]//Proceedings of the 52nd Annual International Symposium on Computer Architecture. ACM, 2025: 974–989. DOI: 10.1145/3695053.3731093
- [22] Hu Z D, Murthy V, Pan Z F, et al. HedraRAG: co-optimizing generation and retrieval for heterogeneous RAG workflows [C]// Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles. ACM, 2025: 623–638. DOI: 10.1145/3731569.3764806

作者简介



郭昊，清华大学计算机系在读博士研究生；主要研究领域为文件系统和向量存储系统。



谢旻晖，南京大学讲师；主要研究领域为存储系统和人工智能系统；主持国家自然科学基金青年基金项目、北京市自然科学基金项目等；入选北京市青年托举人才工程，获得ACM中国优秀博士学位论文奖等；发表论文20余篇。



陆游游，清华大学计算机系长聘副教授、博士生导师，清华大学（计算机系）-软通计算机人工智能系统联合研究中心主任，青年北京学者，FAST、SOSP、OSDI、USENIX ATC、EuroSys等国际会议组织委员会或程序委员会委员；主要研究方向为存储与人工智能系统；曾获国家优秀青年科学基金项目资助，作为项目负责人主持了新一代人工智能国家科技重大专项项目、国家重点研发计划项目、国家自然科学基金重点项目等；6次获得最佳国际论文奖项，参与研制的高性能文件系统SuperFS部署于鹏城云脑II，2023年—2025年该系统多次蝉联世界超算存储IO500榜单第一名；发表论文80余篇。