

GIAC

# 大模型时代下向量数据库的设计与应用

msup<sup>®</sup> | ARCHNOTES 架構  
高可用 架构

# 个人简介



**邱培峰**

拓数派向量数据库负责人

**目前在拓数派负责向量数据库PieCloudVector产品，聚焦于大模型与大数据领域。**拥有多年数据库内核研发和配套解决方案架构经验，在加入拓数派前曾就职于开源大数据平台Greenplum团队，担任外部数据源访问框架，对象存储访问扩展，ETL工具等产品模块的研发，并曾参与PostgreSQL多个版本的代码贡献，拥有丰富的存储模块核心开发和性能优化等实践经验。

# 拓数派：大模型数据计算系统先行者

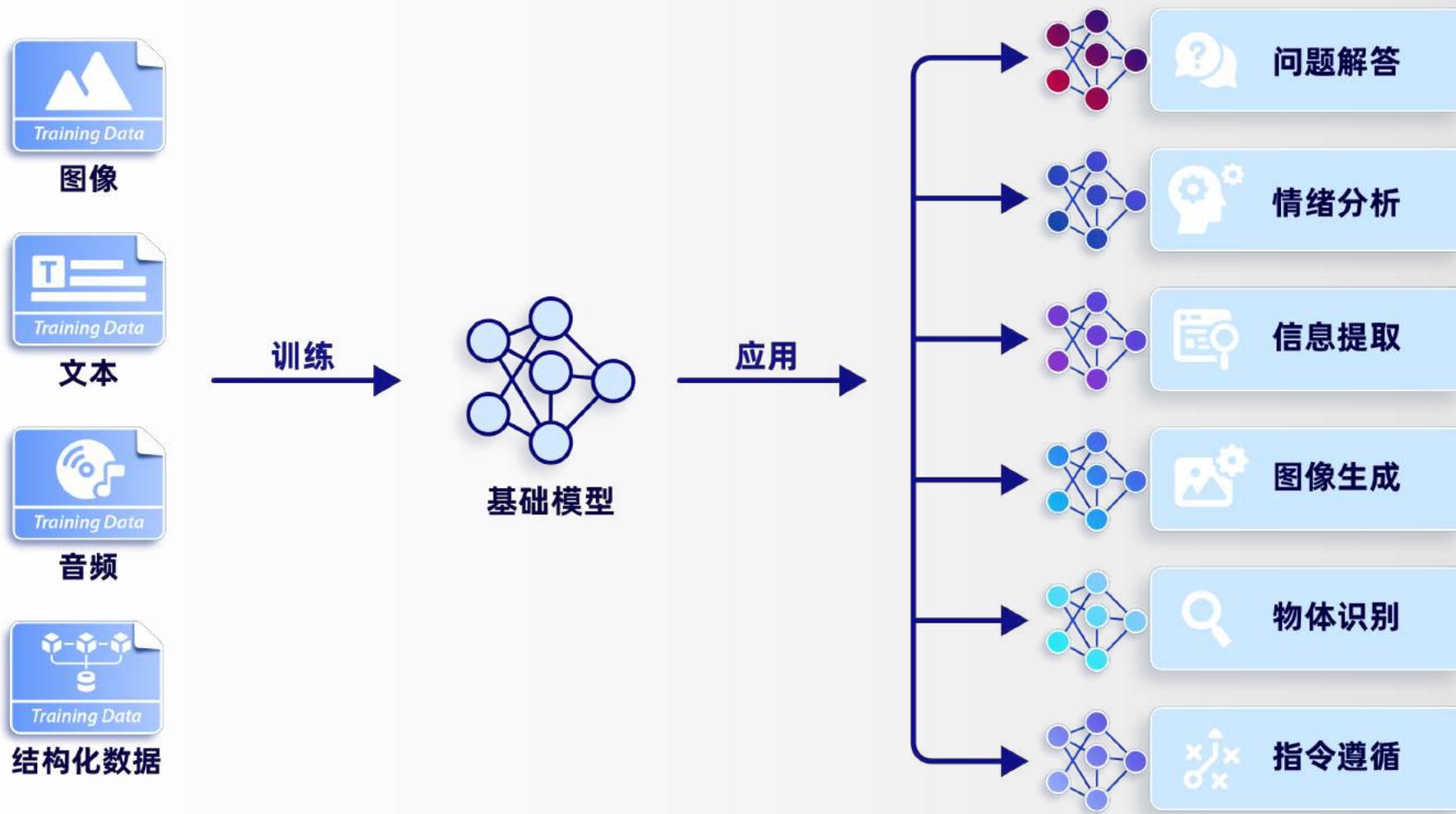
- 拓数派 (OpenPie) 是立足于国内的基础数据计算领域**高科技创新机构**;
- 拥有强大的**数据库内核研发团队**、**数据科学团队**和**数字化转型团队**;
- 国内**虚拟数仓和eMPP技术提出者**，不断在数据计算引擎方向进行创新，**全面拥抱AI技术趋势**。



# 目录

- 大模型应用和RAG
- 向量近似搜索和向量数据库
- PieCloudVector架构设计与挑战
- 案例介绍

# 大模型



## 检索增强生成(RAG)

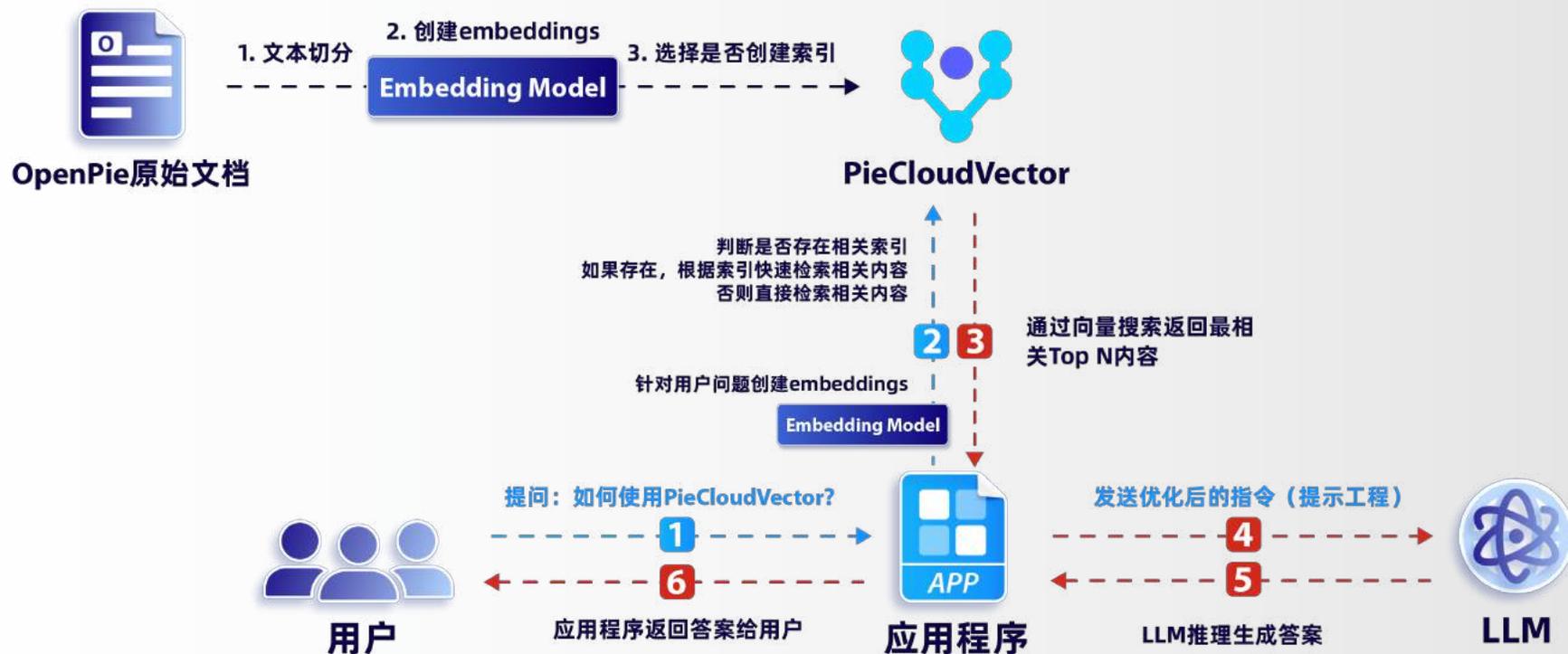
使用大模型可以构造问答，聊天等应用，但同时也存在以下问题

- 数据时效 - LLM训练数据有截止日期，不包含最新信息，无法准确回答相关信息
- 私域数据 - LLM训练数据多来源于公开渠道，无法接触到私域数据，对特定领域的生成任务质量不高。
- 长期记忆 - LLM本身却没有长期记忆能力，对长时间交互的上下文



# 检索增强生成(RAG)

- 将辅助增强数据通过embedding过程转换为向量，加载到向量数据库中并做索引
- 对每个用户输入同样通过embedding过程得到向量，从向量数据库中搜索距离相近数据
- 将这些辅助数据与用户输入同时输入给大模型之后输出



# ▶ 向量数据库

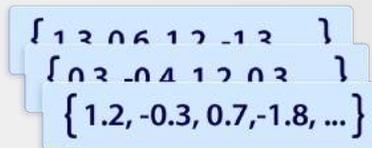
- embedding通过大模型将各种形式的数据转换成向量



视频



视频模型



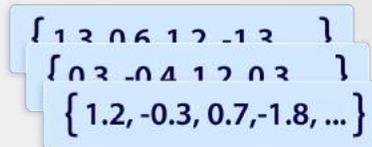
视频向量Embeddings



文本



文本模型



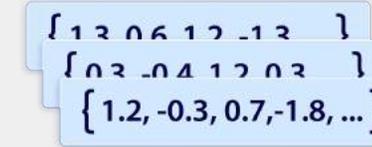
文本向量Embeddings



音频



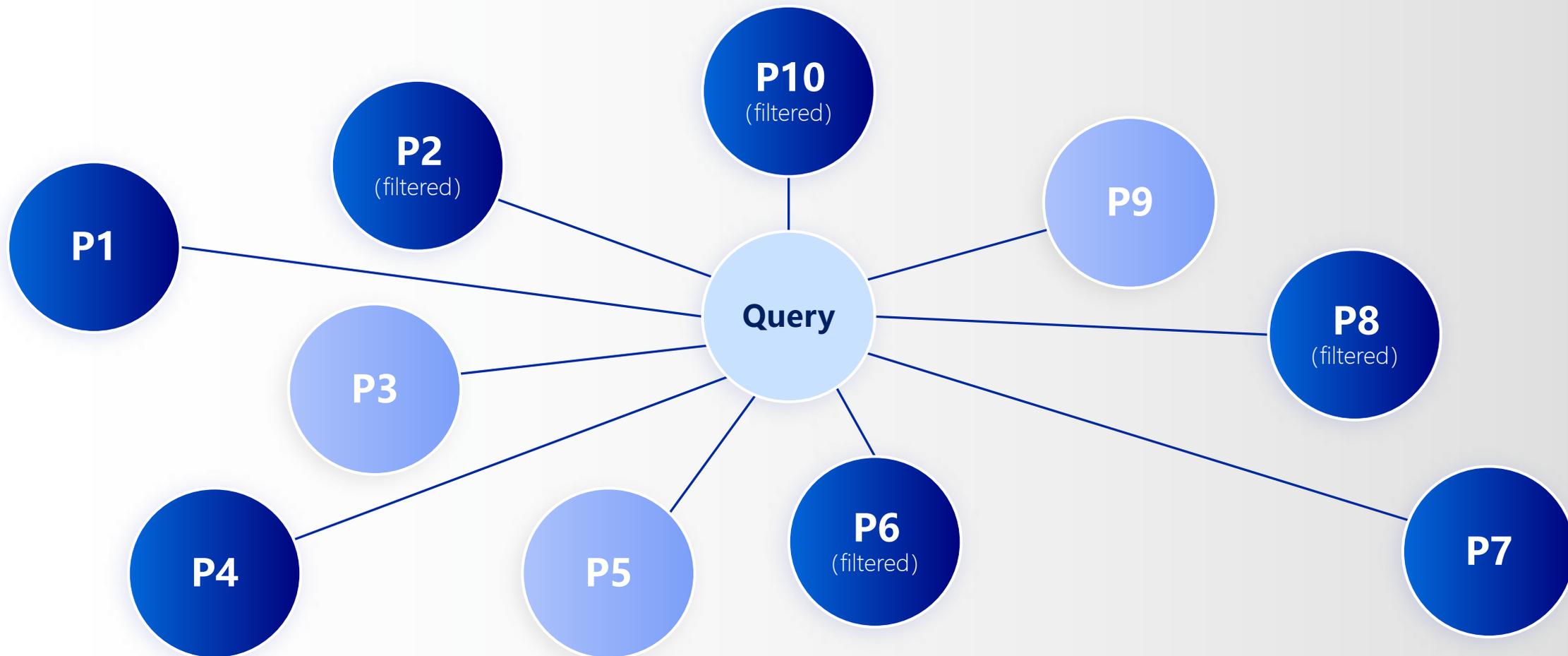
音频模型



音频向量Embeddings

## ▶ 向量数据库

- 两个向量可以计算它们的距离（欧式，余弦/内积，曼哈顿等），距离越近，表示这两个物体越相似
- 向量搜索的基本问题：**K-Nearest Neighbor**
- 在已有的N个向量中找出与给定向量距离最近的K个向量

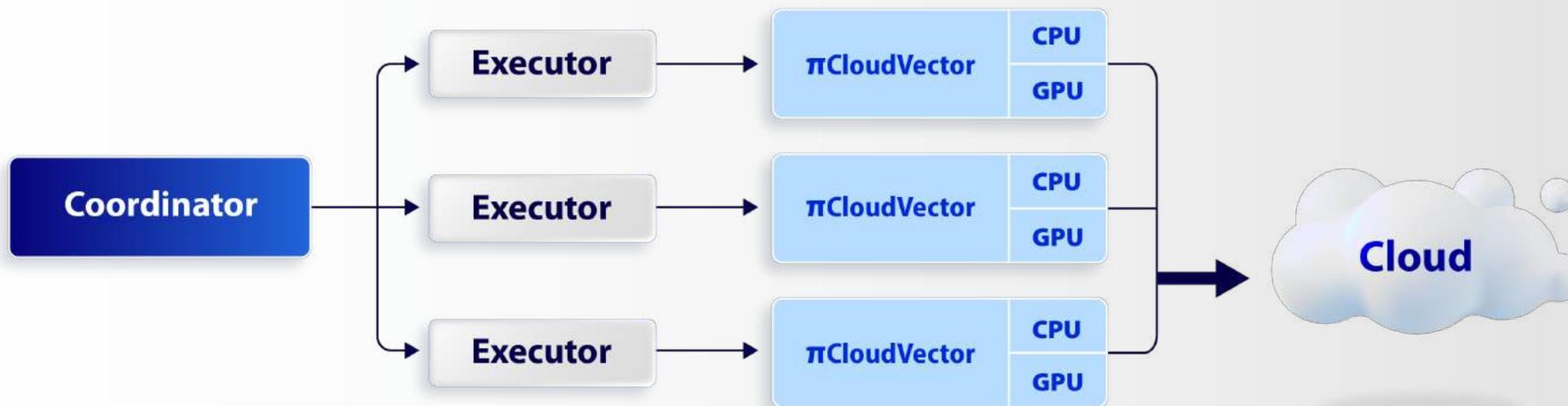


# ▶ 向量数据库

- 存储向量和原始实体(文字/图像/语音)及元信息, 并将它们关联起来
- 对向量数据建立索引, 可以实现高效近似搜索
- 配套调用接口和生态工具
  
- **技术路线**
  - 从**向量搜索**及索引算法实现出发, 为其搭配数据库功能
  - 从**数据存储方案**(关系型数据库/非关系型数据库)出发, 为其开发向量搜索及索引算法

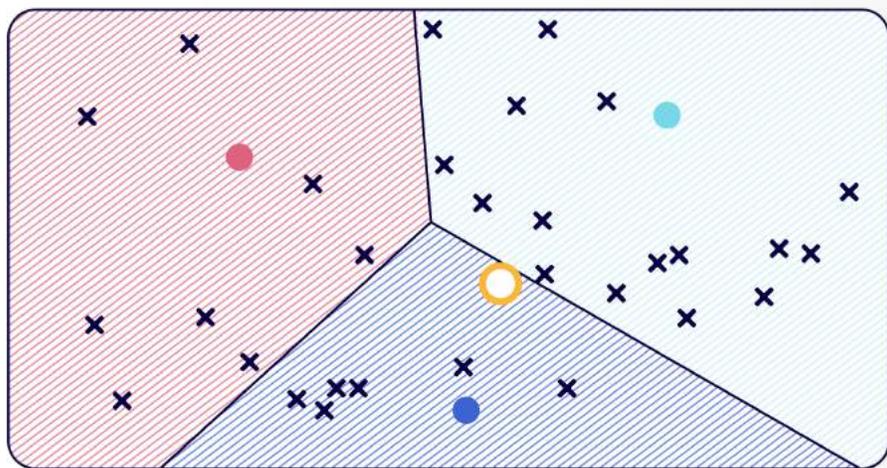
# ▶ PieCloudVector

- 基于postgres打造的数据库内核
  - 单机或分布式部署
  - 支持完整的ACID
  - SQL进行向量搜索
  - 支持向量标量混合查询

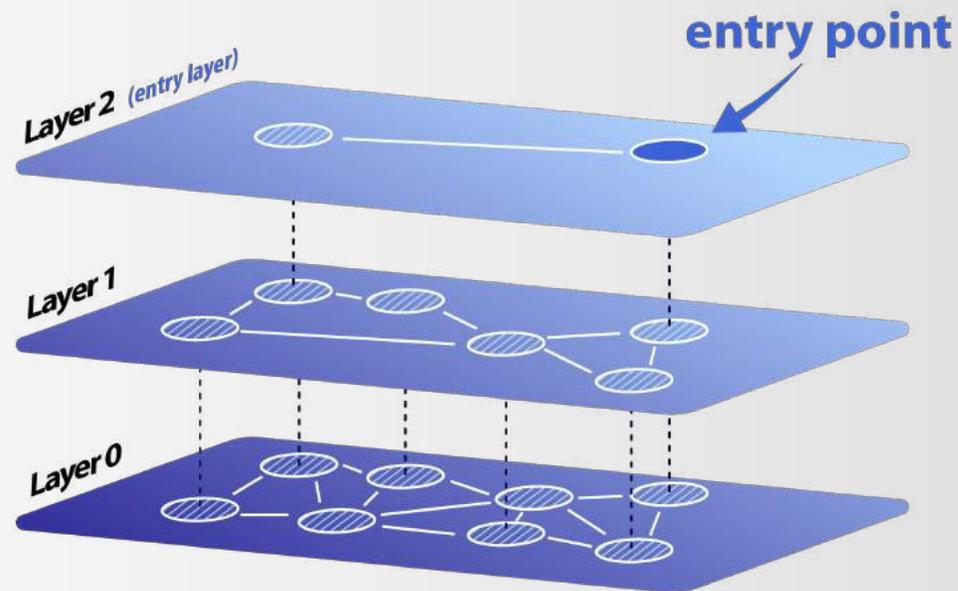


# ▶ PieCloudVector

- 使用faiss开源算法库做为向量搜索引擎
  - 支持主流的ann算法，如ivf和hnsw等



圈 ● ● ● : 区域中心点  
圈 ○ : 要搜索的向量



# ▶ PieCloudVector

- 使用faiss开源算法库做为向量搜索引擎
  - 支持向量编码和压缩如PQ等



# ▶ PieCloudVector

- 使用faiss开源算法库做为向量搜索引擎
  - 支持二进制索引
  - 支持多级索引如HNSW+IVF等
  - CPU多核并行/GPU加速



# ▶ PieCloudVector

- Faiss与postgres内核对接 - 基础接口
  - 增加向量列类型用于基本的加载与卸载
  - 实现向量距离运算符
  - 实现向量近似搜索的索引, 调用faiss

```
nn=# EXPLAIN (COSTS OFF) SELECT id,  
      (embedding <=> (SELECT embedding FROM query LIMIT 1)) AS distance  
      FROM data_collection ORDER BY distance LIMIT 10;  
      QUERY PLAN  
-----  
Limit  
  InitPlan 1 (returns $0)  
    -> Limit  
        -> Seq Scan on query  
    -> Index Scan using vector_index on data_collection  
        Order By: (embedding <=> $0)  
(6 rows)
```

```
nn=# SELECT id,  
      (embedding <=> (SELECT embedding FROM query LIMIT 1)) AS distance  
      FROM data_collection ORDER BY distance LIMIT 10;  
 id | distance  
-----+-----  
322406 | 0  
394441 | 0.12031299615742674  
341913 | 0.12509924530290328  
501332 | 0.12534004714839098  
180224 | 0.1261321029336463  
249700 | 0.12673765682588956  
  1476 | 0.12840686214513697  
240380 | 0.13752303718779268  
532503 | 0.1405180247421598  
202604 | 0.14493797496159655  
(10 rows)
```

# ▶ PieCloudVector

- **Faiss与postgres内核对接 - 数据可见性**
  - faiss索引增加mvcc信息, 与数据库内可见性保持一致

xmin	xmax	id	vector
10		1	[1.0, 1.1, 1.2 ...]
10	30	2	[1.0, 1.1, 1.2 ...]
15		3	[1.0, 1.1, 1.2 ...]

# ▶ PieCloudVector

- Faiss OpenMP线程改造

- LLVM解析源码，找到所有OpenMP指令语句
- 转换为调用自定义线程池和lambda表达式
- 共享变量替换及并发保护

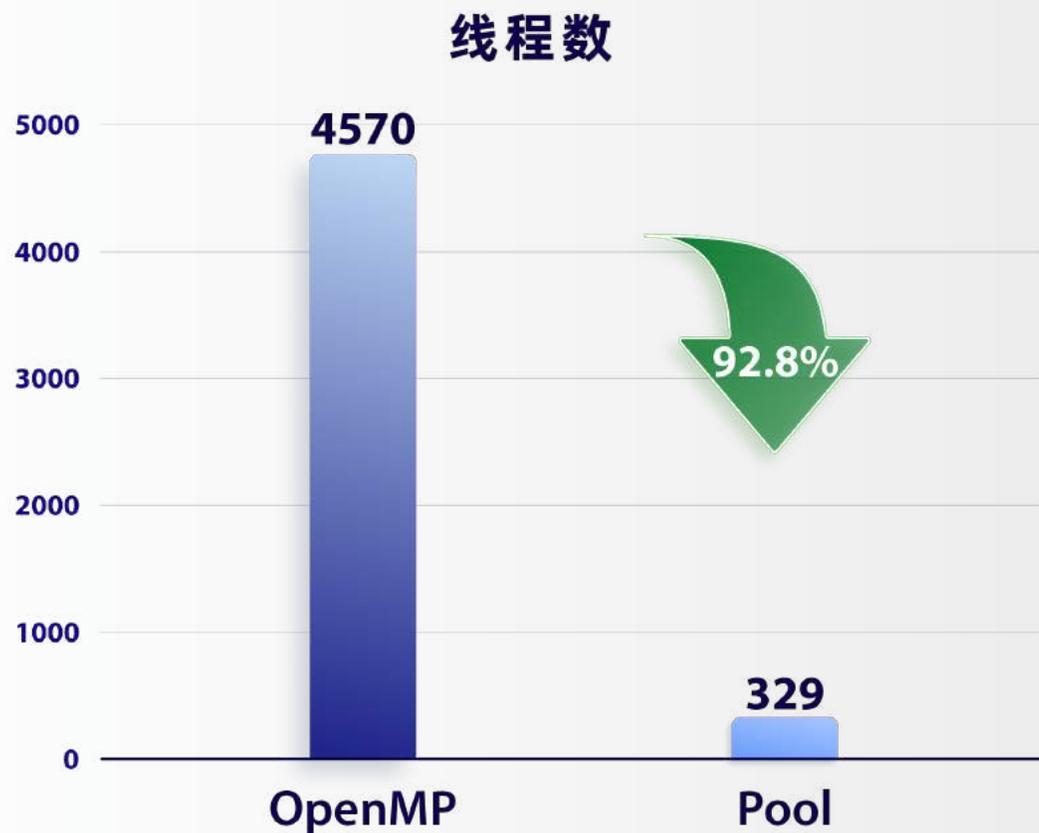
```
318     - #pragma omp parallel
327+    // #pragma omp parallel
328+    {
329+    OmpDispatcher ompd;
330+    std::atomic<idx_t> loopvar0 = i0;
331+    ompd.SpawnEachThread([&](int nt, int rank){
319    332    {
320    333        VisitedTable vt(index->ntotal);
321    334        typename BlockResultHandler::SingleResultHandler res(bres);
322    335
323    336        std::unique_ptr<DistanceComputer> dis(
324    337            storage_distance_computer(index->storage));
325    338
326    - #pragma omp for reduction(+ : n1, n2, n3, ndis, nreorder) schedule(guided)
327    -     for (idx_t i = i0; i < i1; i++) {
339+    // #pragma omp for reduction(+ : n1, n2, n3, ndis, nreorder) schedule(guided)
340+    for (idx_t i = loopvar0.fetch_add(1); i < i1; i = loopvar0.fetch_add(1)) {
341+    size_t _n1 = 0;
342+    size_t _n2 = 0;
343+    size_t _n3 = 0;
344+    size_t _ndis = 0;
345+    size_t _nreorder = 0;
```

# ➤ PieCloudVector

- **Faiss OpenMP线程改造**
  - 控制全局线程数
  - 降低线程锁冲突
  - 降低内存使用

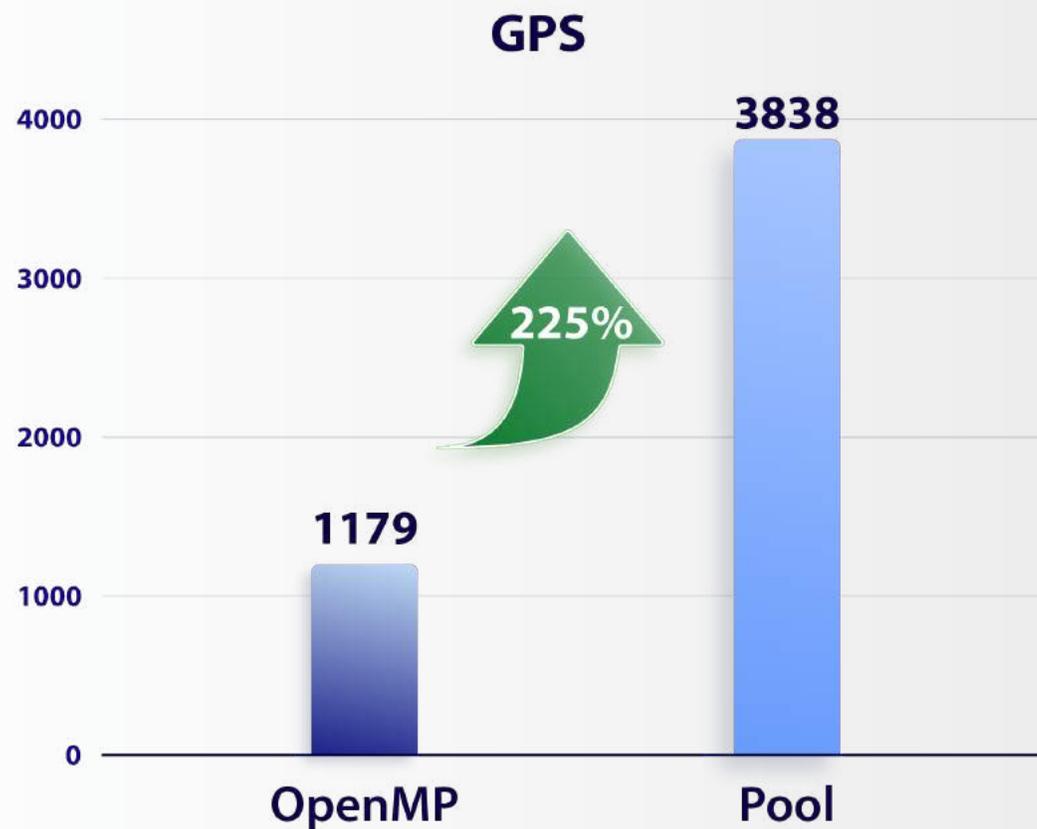
# ▶ PieCloudVector

- Faiss OpenMP线程改造
  - 避免无效线程



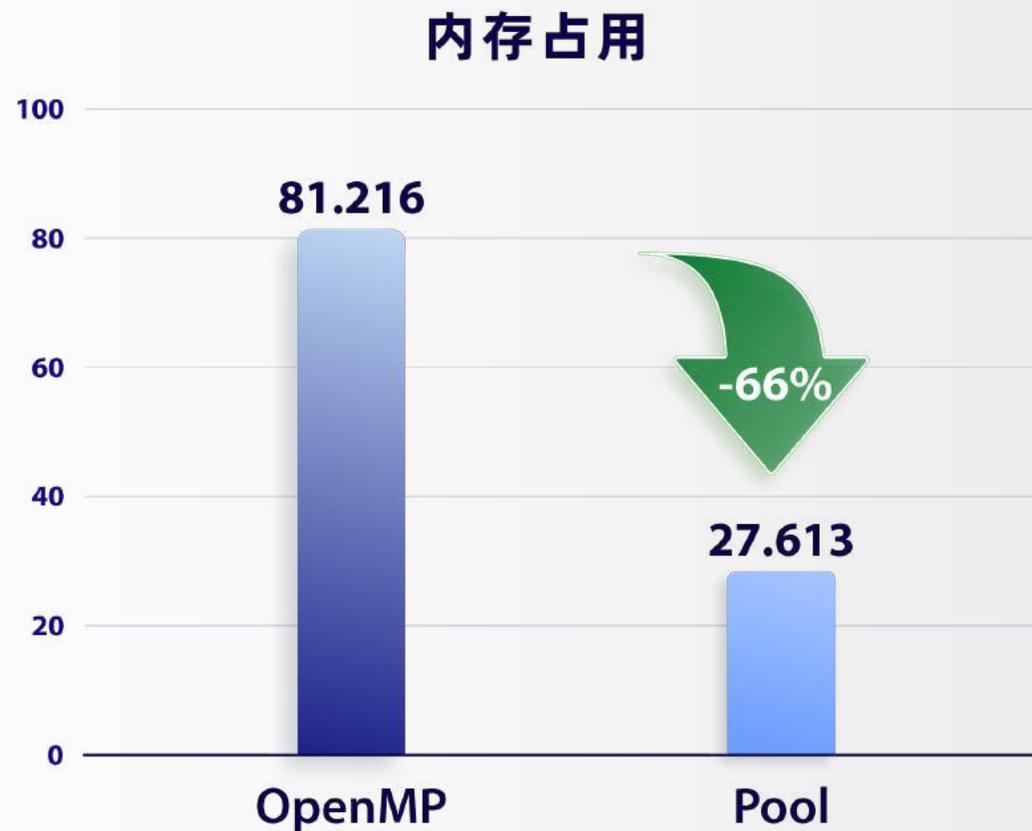
# ▶ PieCloudVector

- Faiss OpenMP线程改造
  - QPS大幅提升



# ▶ PieCloudVector

- Faiss OpenMP线程改造
  - 内存占用大幅降低



# ▶ PieCloudVector

- **Faiss与postgres内核对接** - gpu搜索的特殊路径
  - 避免并发调用gpu
  - 查询请求按批单线程提交

# ▶ PieCloudVector

- 兼容国产硬件和操作系统



# ▶ PieCloudVector

- 通过信通院测试

## Congrats! 拓数派PieCloudVector完成中国信通院向量数据库产品测试

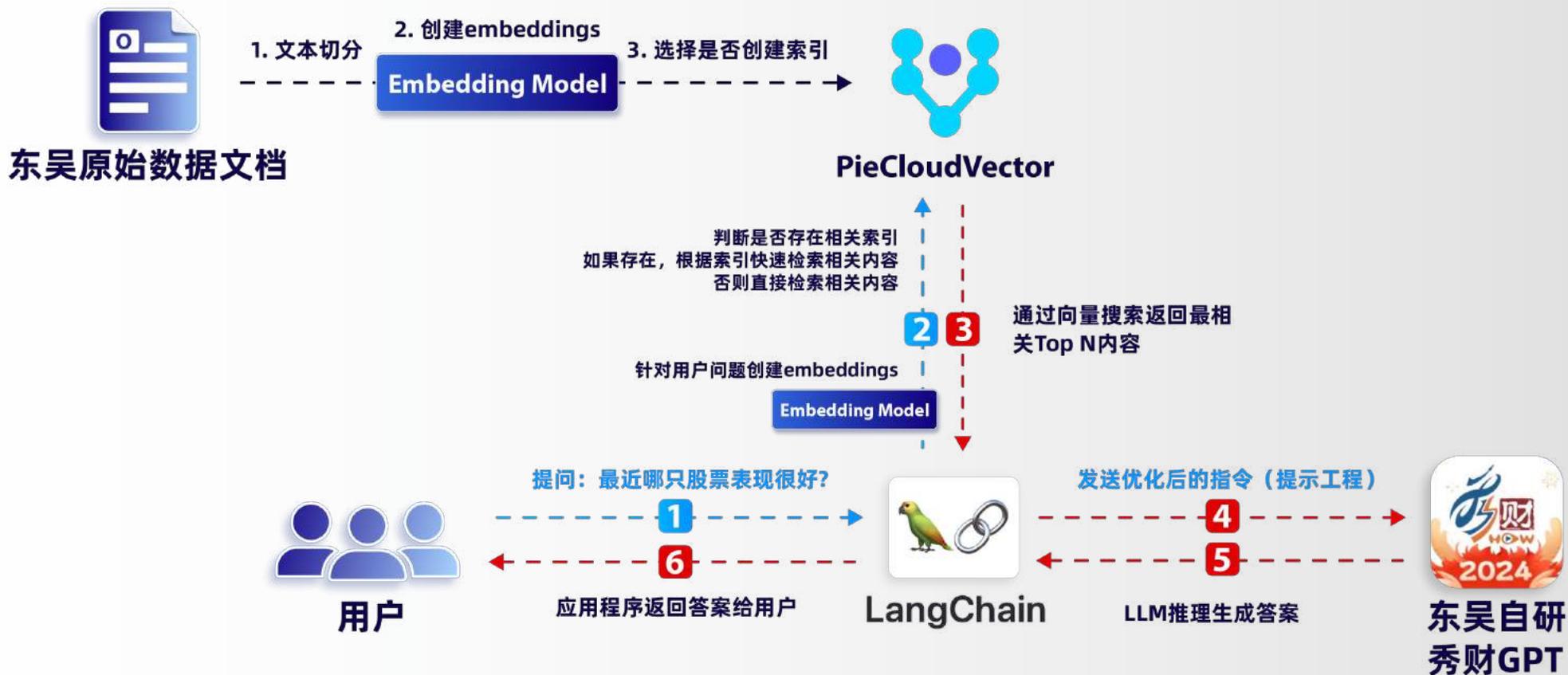
大数据技术标准推进委员会 2024-04-12 10:55 北京

文本、图像、音视频等海量的非结构化数据占数据总量不断上升，预计2025年，将达到八成以上。这些数据由于结构不统一，所以在计算机中通常以向量形式进行特征提取后，再进行存储、计算、应用。为了解决海量数据场景下，向量的存储、计算问题，向量数据库应运而生，其通过数据算法和计算机工程，将向量特征进行分组和索引，以实现高效的相似性搜索。《数据库发展研究报告（2023年）》显示，向量数据库作为12个数据库技术发展趋势之一，其使用场景十分广泛，包括但不限于支撑文本翻译、产品推荐、语义搜索、图像搜索、视频推荐、异常检测、欺诈检测、人脸识别等应用。

近日，在中国信通院“可信数据库”向量数据库产品测试中，杭州拓数派科技发展有限公司（简称：拓数派）PieCloudVector数据库产品顺利完成向量数据库产品基础能力测试。测试过程依据标准在基本功能、运维管理、安全性、兼容性、扩展性、高可用以及工具生态七大能力域的相关要求。测试结果表明，该产品在向量数据库的功能完备性、易用性、通用性等方面均符合标准要求。

# 案例分析 - 东吴证券秀财gpt

- 采用自研大模型东吴秀财GPT + LangChain开发框架 + PieCloudVector向量数据库构建了AIGC应用平台，接入了交易应用的结构化数据和非结构化数据，其中非结构化数据主要是文本类数据



## ▶ 下一步展望

- GraphRAG

▶ 欢迎关注我们!



**关注OpenPie公众号**

解锁更多新鲜资讯



**加入PieCloudDB技术群**

获得更多技术干货

# GIAC



麦思博(msup)有限公司是一家面向技术型企业的培训咨询机构, 携手2000余位中外客座导师, 服务于技术团队的能力提升、软件工程效能和产品创新迭代, 超过3000余家企业续约学习, 是科技领域占有率第1的客座导师品牌, msup以整合全球领先经验实践为己任, 为中国产业快速发展提供智库。



高可用架构主要关注互联网架构及高可用、可扩展及高性能领域的知识传播。订阅用户覆盖主流互联网及软件领域系统架构技术从业人员。高可用架构系列社群是一个社区组织, 其精神是“分享+交流”, 提倡社区的人人参与, 同时从社区获得高质量的内容。

msup® | ARCHNOTES  
高可用架构