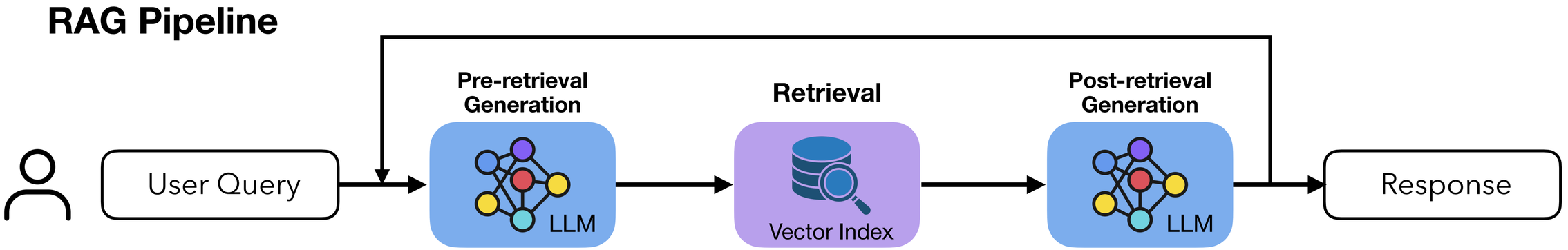
🔭 **TeleRAG**: Efficient Retrieval-Augmented Generation Inference with Lookahead Retrieval

Chien-Yu Lin* ∞, Keisuke Kamahori*, Yiyu Liu*, Xiaoxiang Shi, Madhav Kashyap, Yile Gu, Rulin Shao, Zihao Ye, Kan Zhu, Rohan Kadekodi, Stephanie Wang, Arvind Krishnamurthy, Luis Ceze, Baris Kasikci

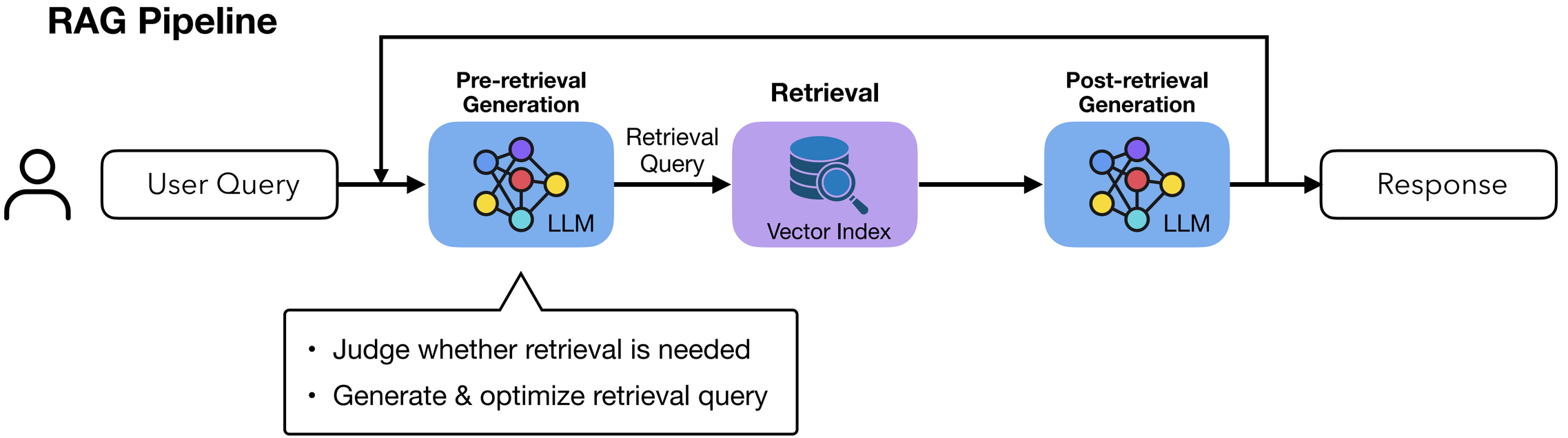
University of Washington, Harvard University, Shanghai Jiao Tong University


*equal contributions, ∞currently at Meta ☝

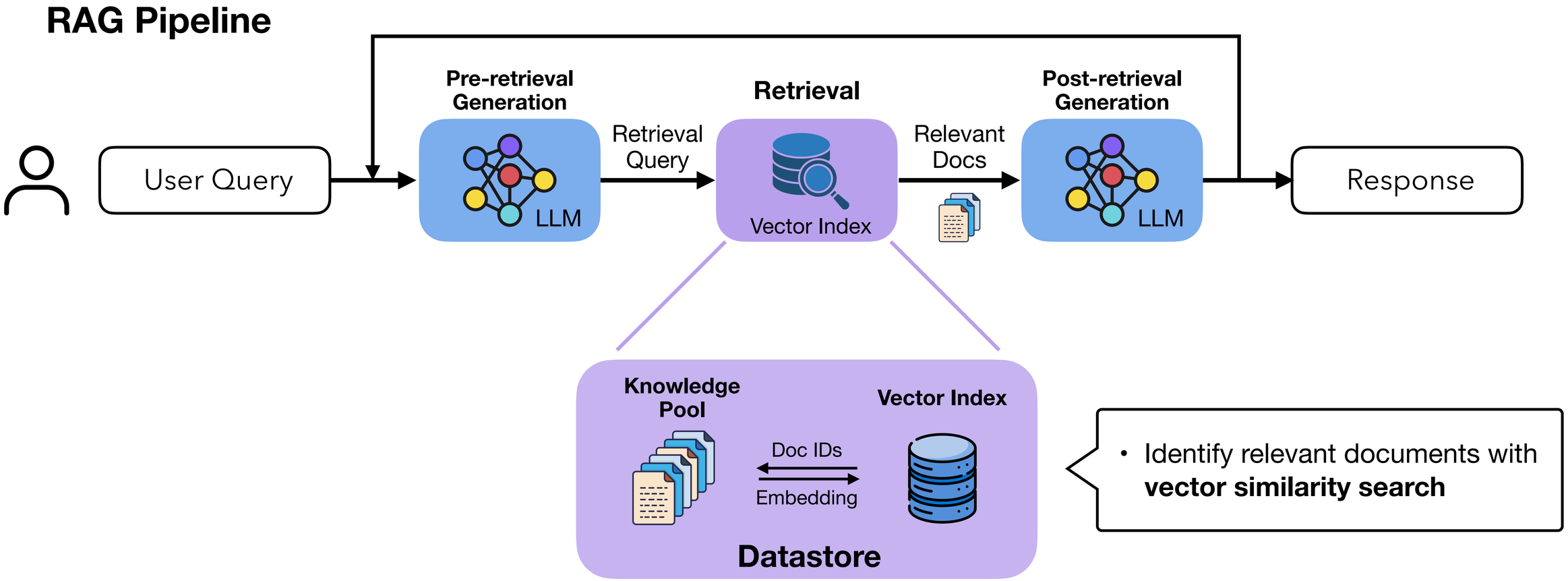
Retrieval-Augmented Generation (RAG)



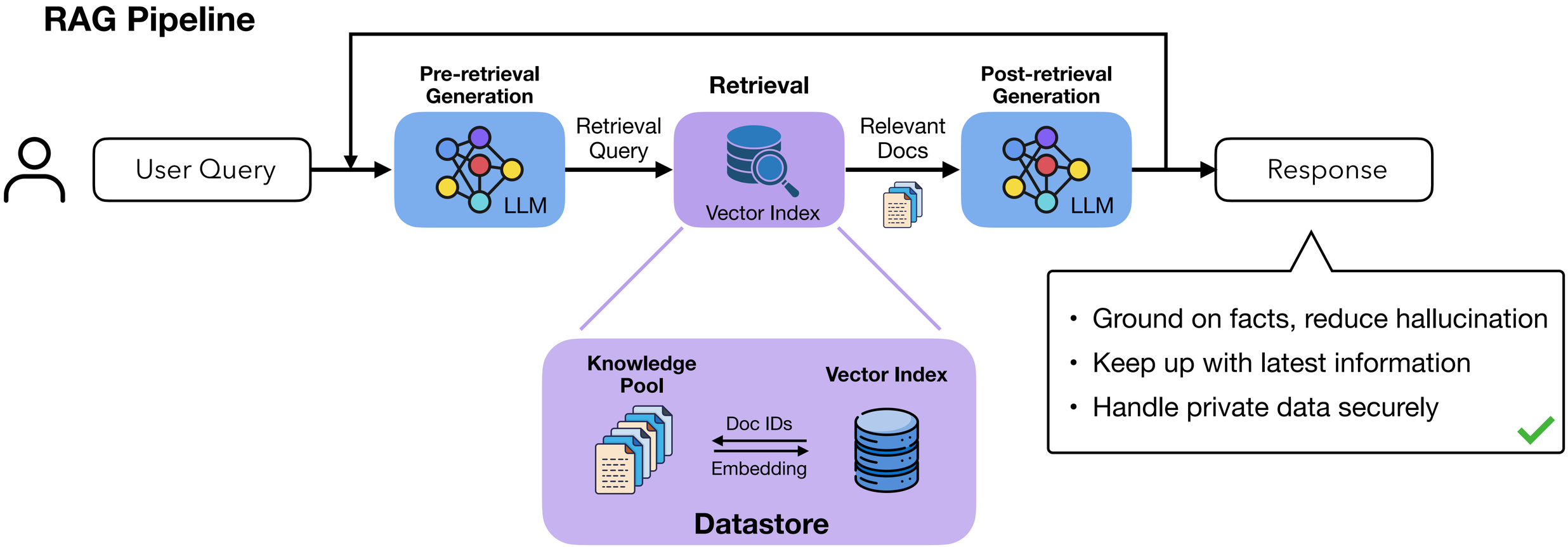
Retrieval-Augmented Generation (RAG)



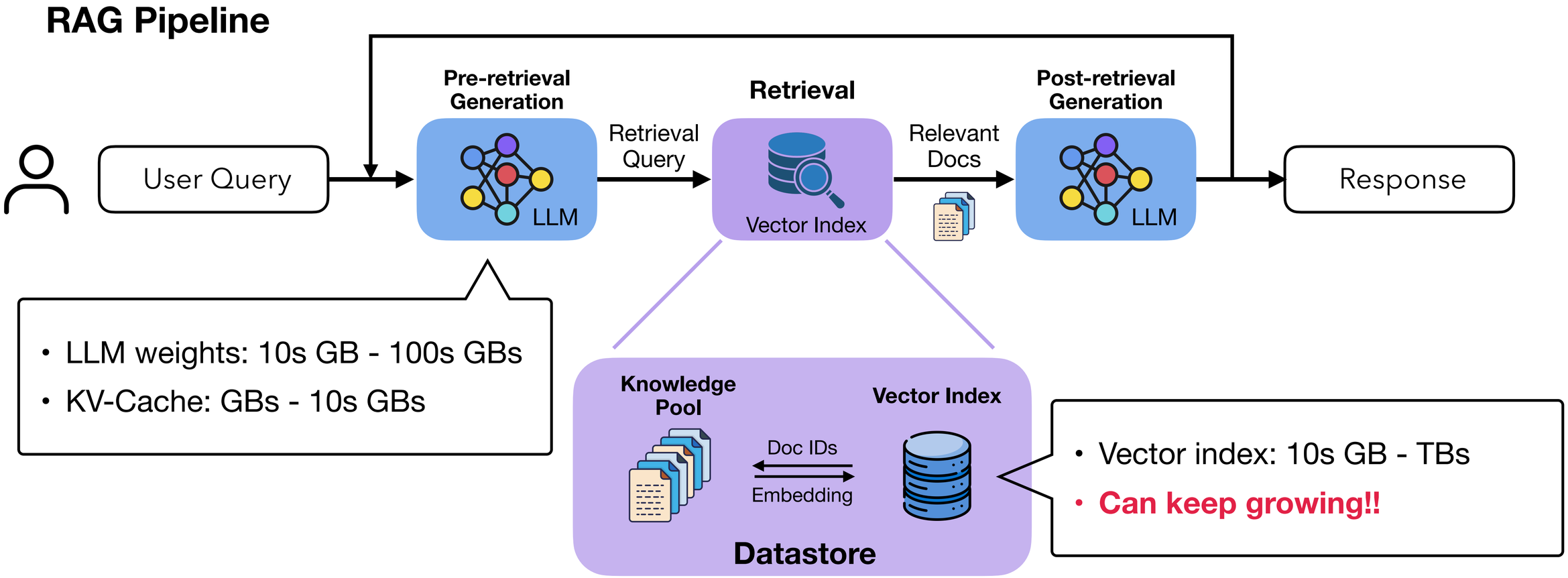
Retrieval-Augmented Generation (RAG)



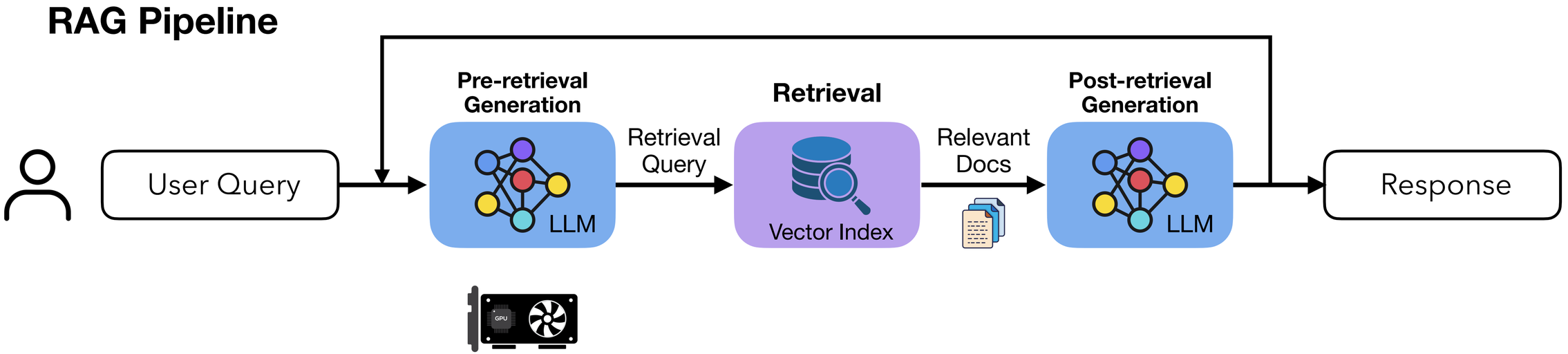
Retrieval-Augmented Generation (RAG)



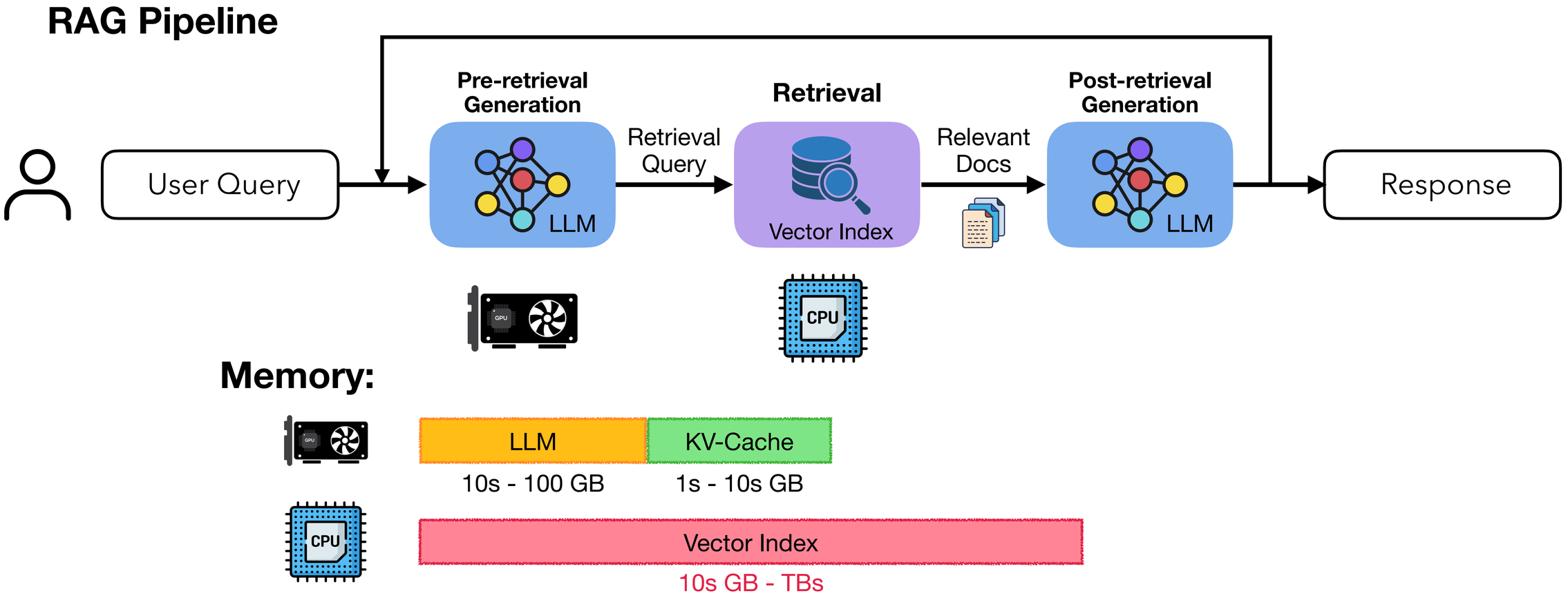
Challenges of RAG: Large Memory



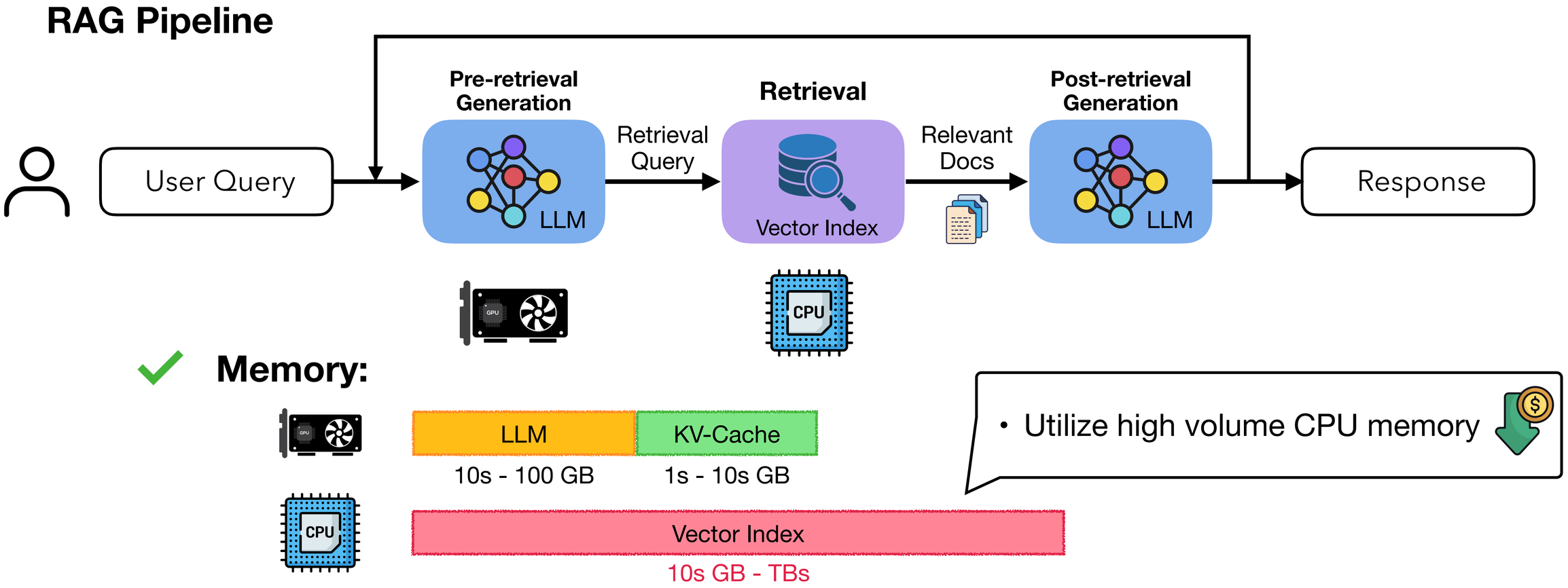
Common Deployment: CPU Off-loading Vector Index



Common Deployment: CPU Off-loading Vector Index

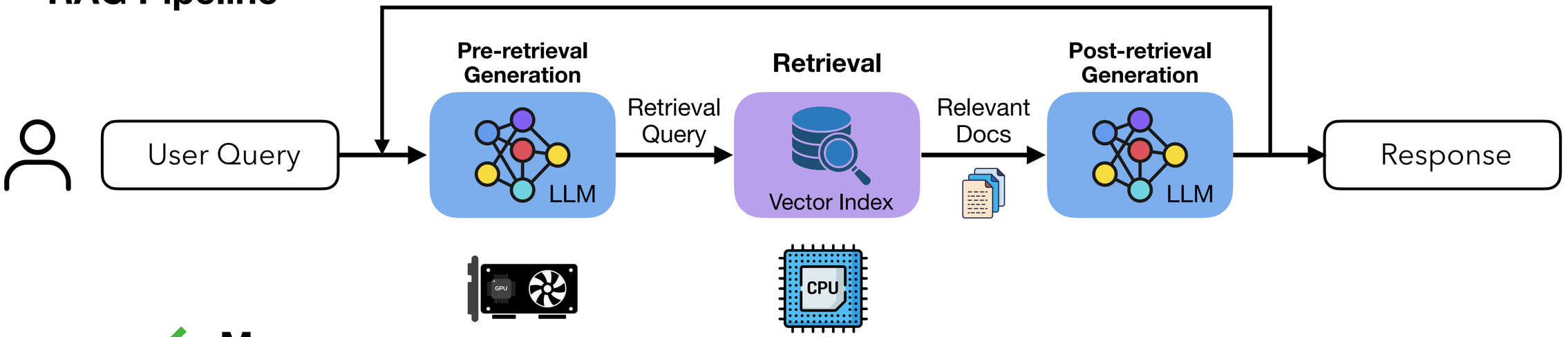


Common Deployment: CPU Off-loading Vector Index

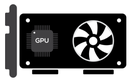


Common Deployment: CPU Off-loading Vector Index

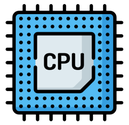
RAG Pipeline



✓ **Memory:**



LLM	KV-Cache
10s - 100 GB	1s - 10s GB



Vector Index
10s GB - TBs

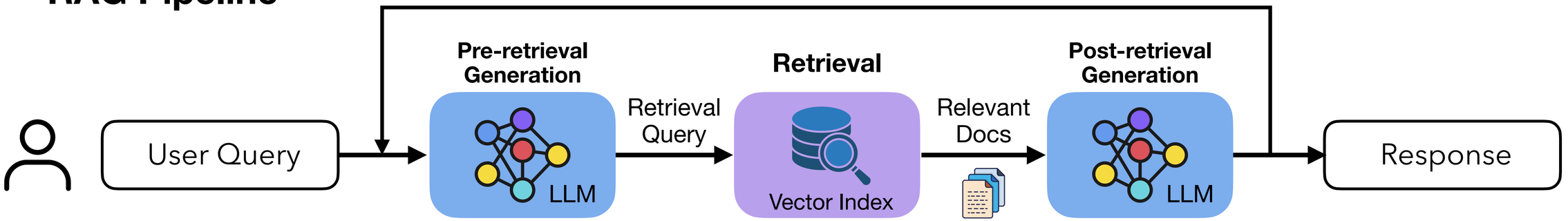
- Suffer from slow CPU retrieval

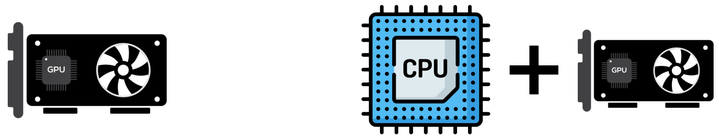
✗ **Latency:**

LLM Gen.	CPU Retrieval	LLM Gen.
200 - 500ms	300ms - 1s	200 - 500ms

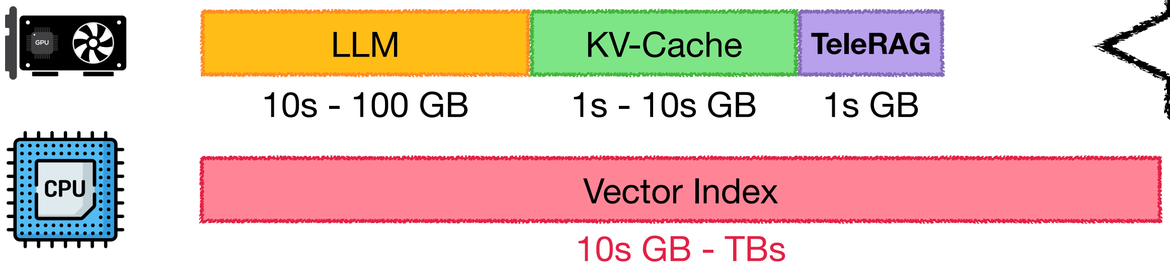
TeleRAG Goal: Fast Retrieval with Small GPU Memory

RAG Pipeline



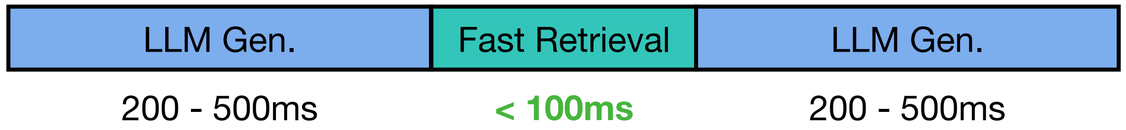


✓ **Memory:**



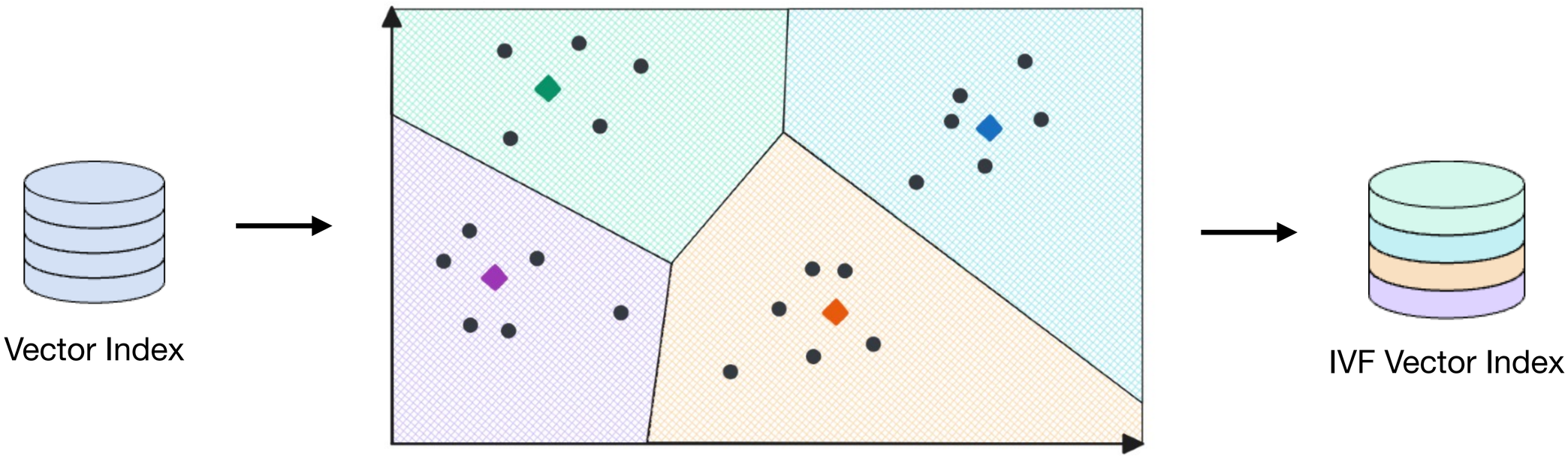
TeleRAG:
- GPU-like retrieval speed
- Minimal GPU memory requirement

✓ **Latency:**



Retrieval: Inverted File Index (IVF)

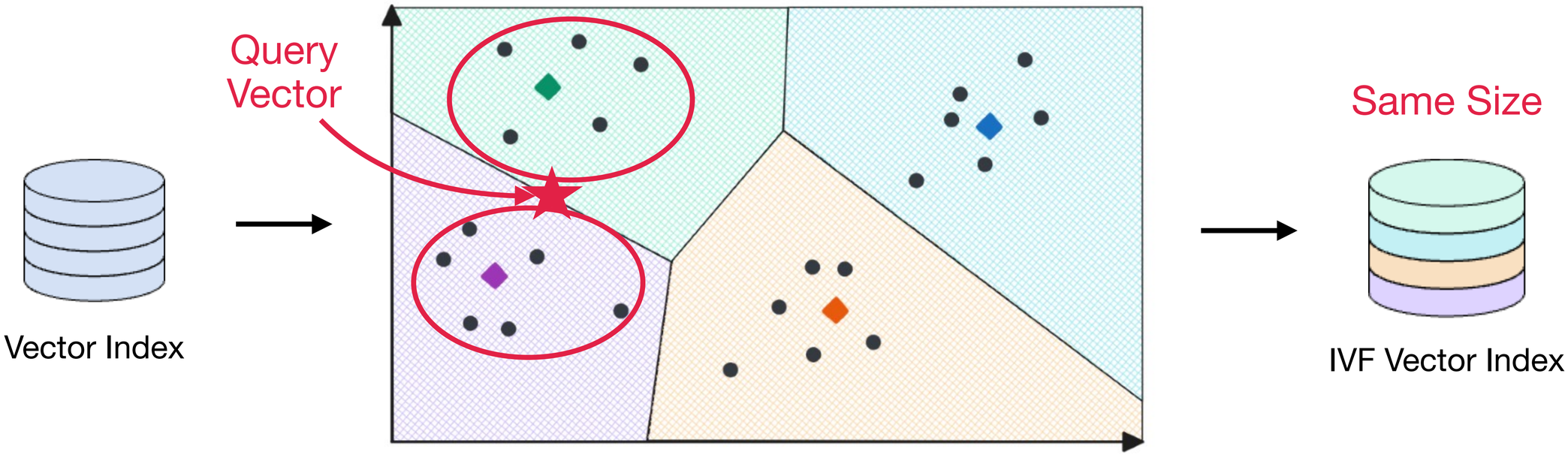
- Partition the datastore into multiple clusters based on vector similarity
- Two-step search: (1) **coarse-grained** on centroids (2) **fine-grained** in selected clusters
- **Speed up** the search process with high recall, but ***do not*** save memory



IVF: Partition vector datastore into clusters

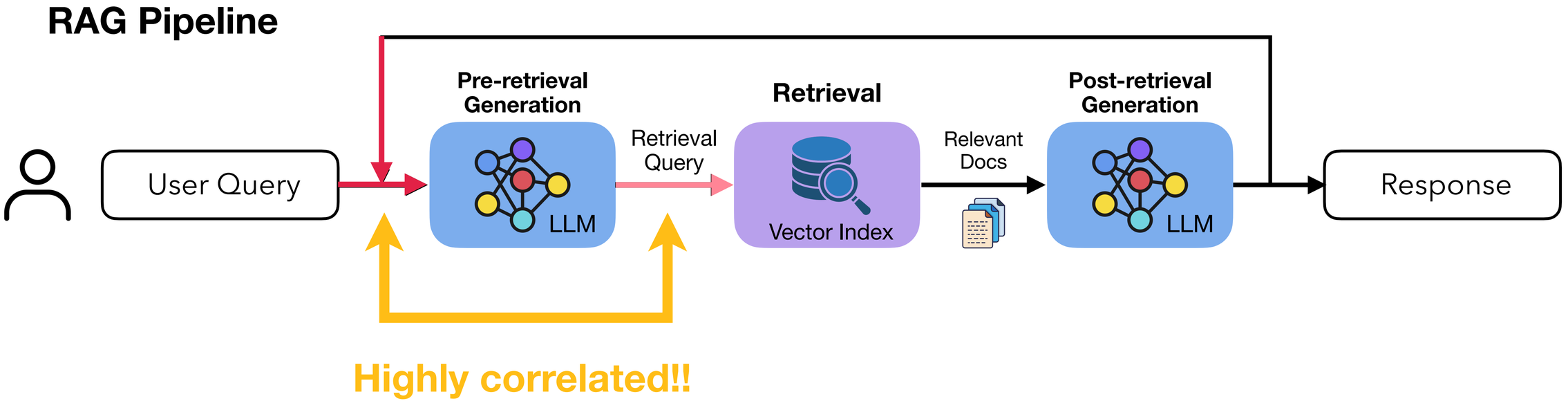
Retrieval: Inverted File Index (IVF)

- Partition the datastore into multiple clusters based on vector similarity
- Two-step search: (1) **coarse-grained** on centroids (2) **fine-grained** in selected clusters
- **Speed up** the search process with high recall, but ***do not*** save memory

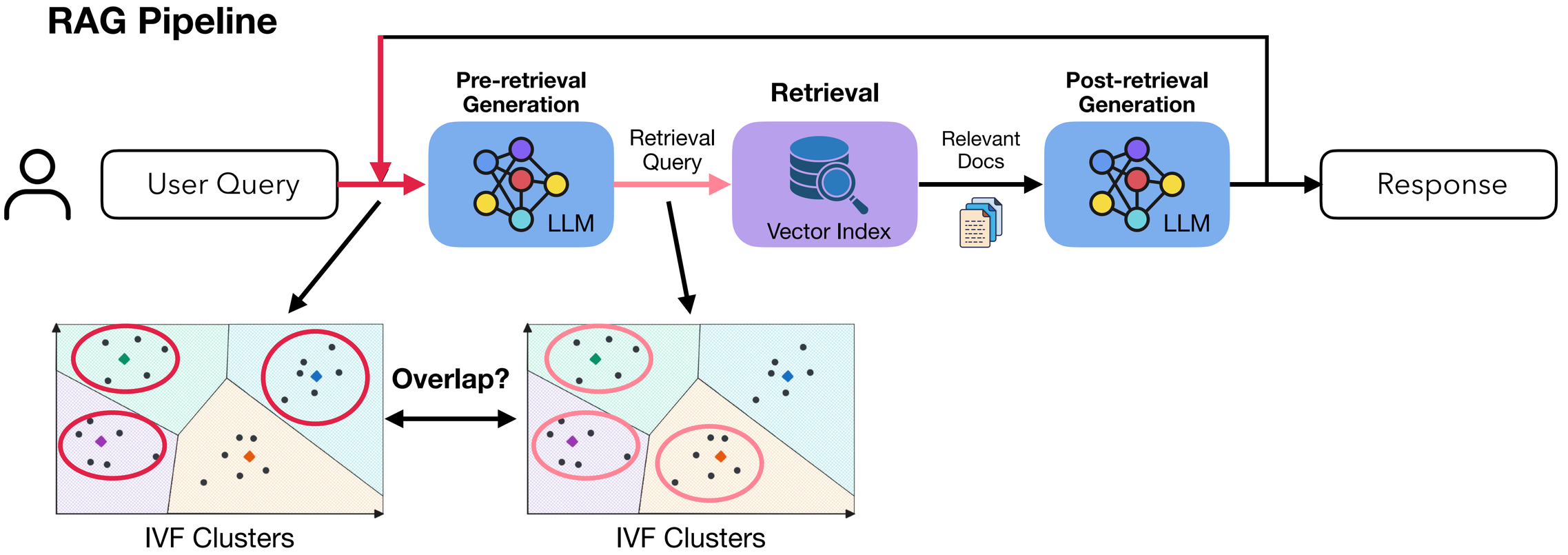


IVF: Partition vector datastore into clusters

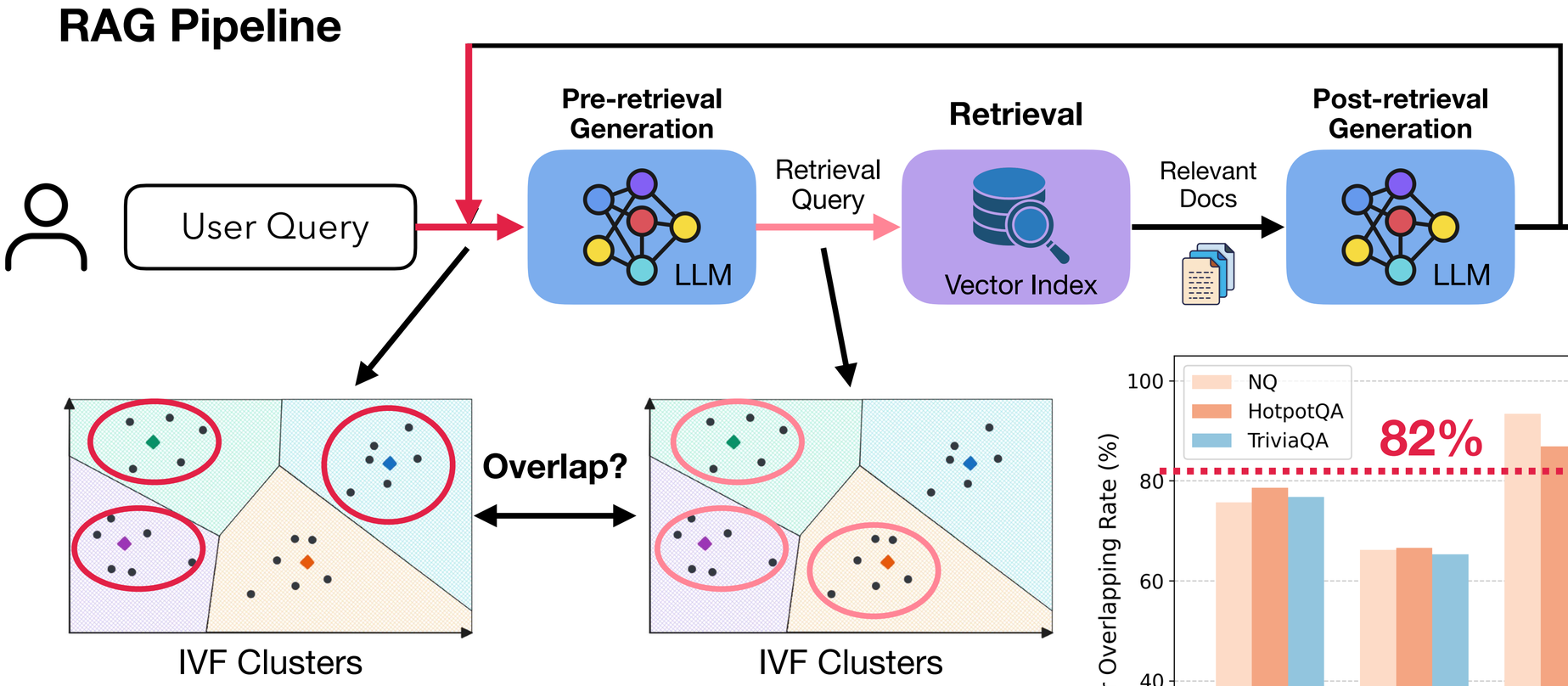
Insight: Query Correlation Between Stages

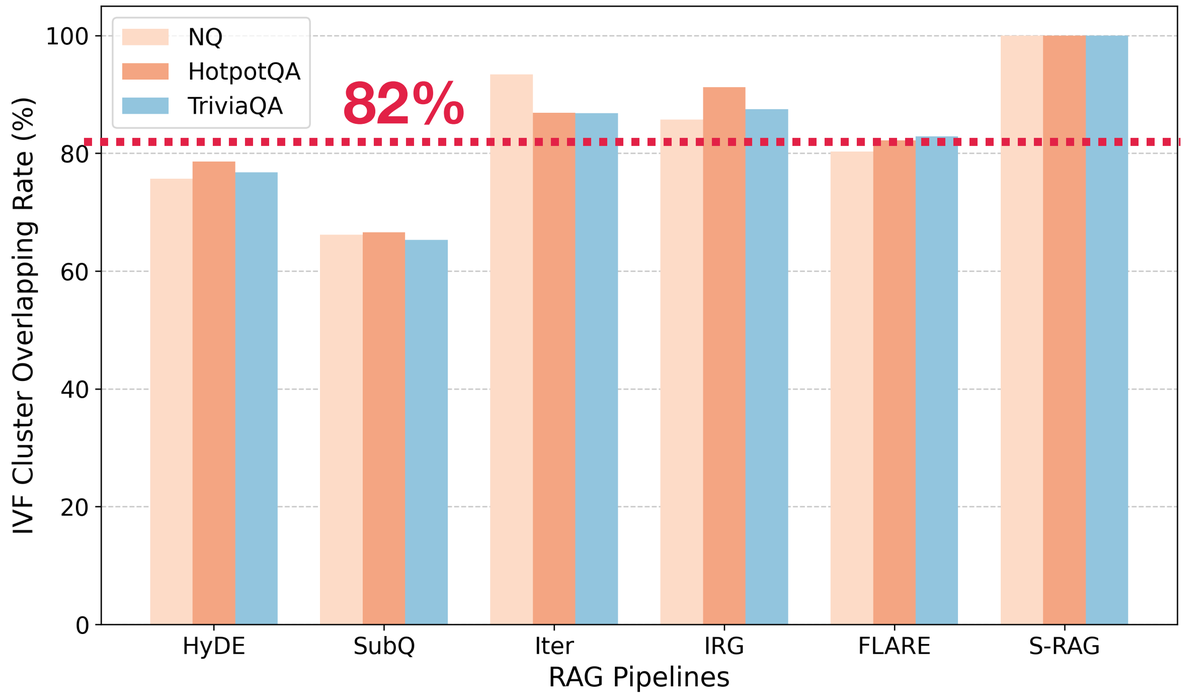


IVF Cluster Overlapping Analysis



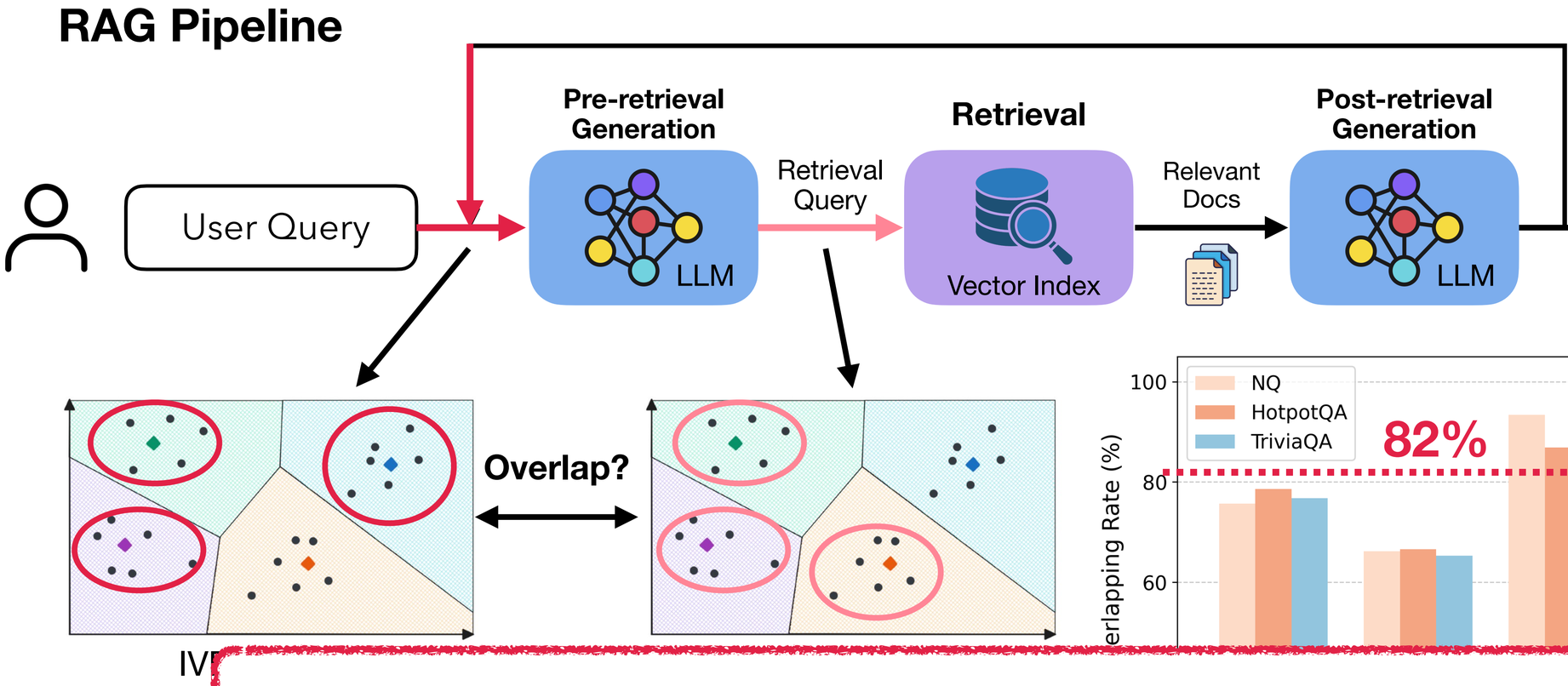
IVF Cluster Overlapping Analysis

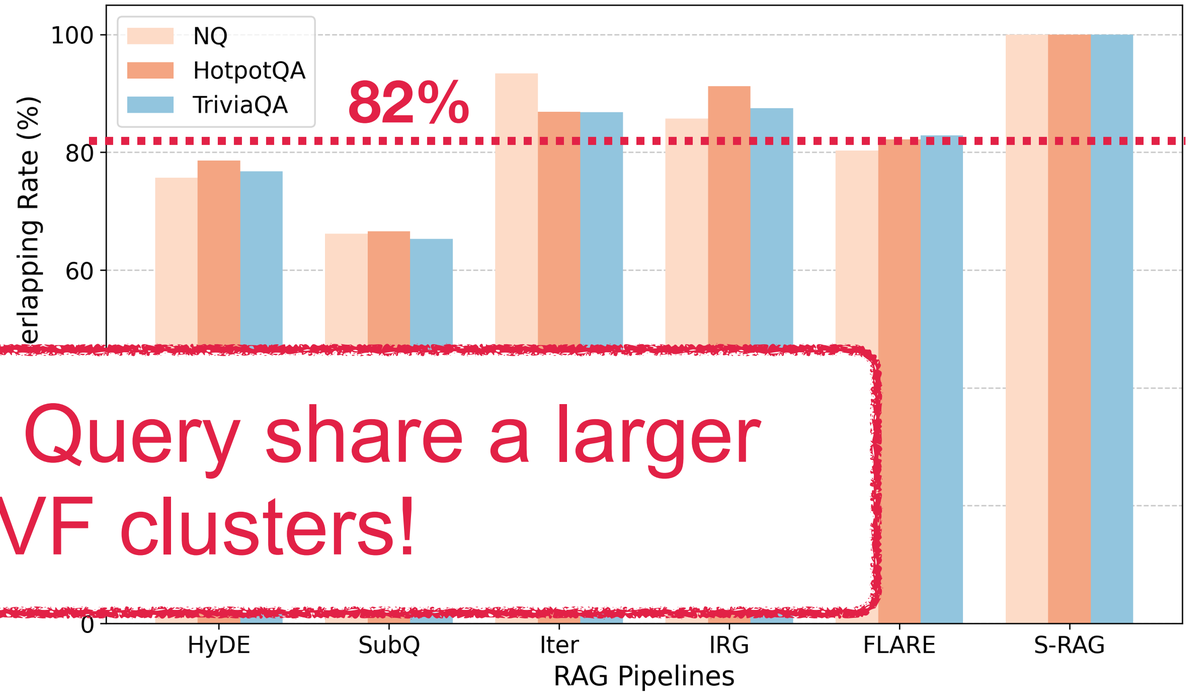




* Test on a IVF index with 4096 cluster.

IVF Cluster Overlapping Analysis

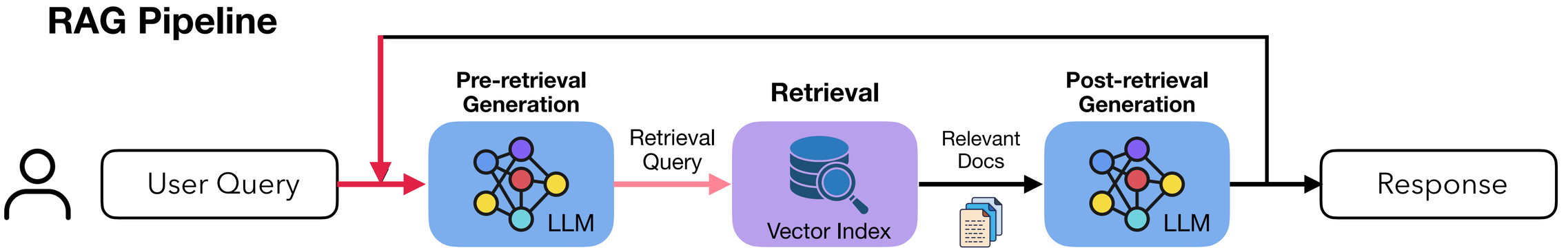




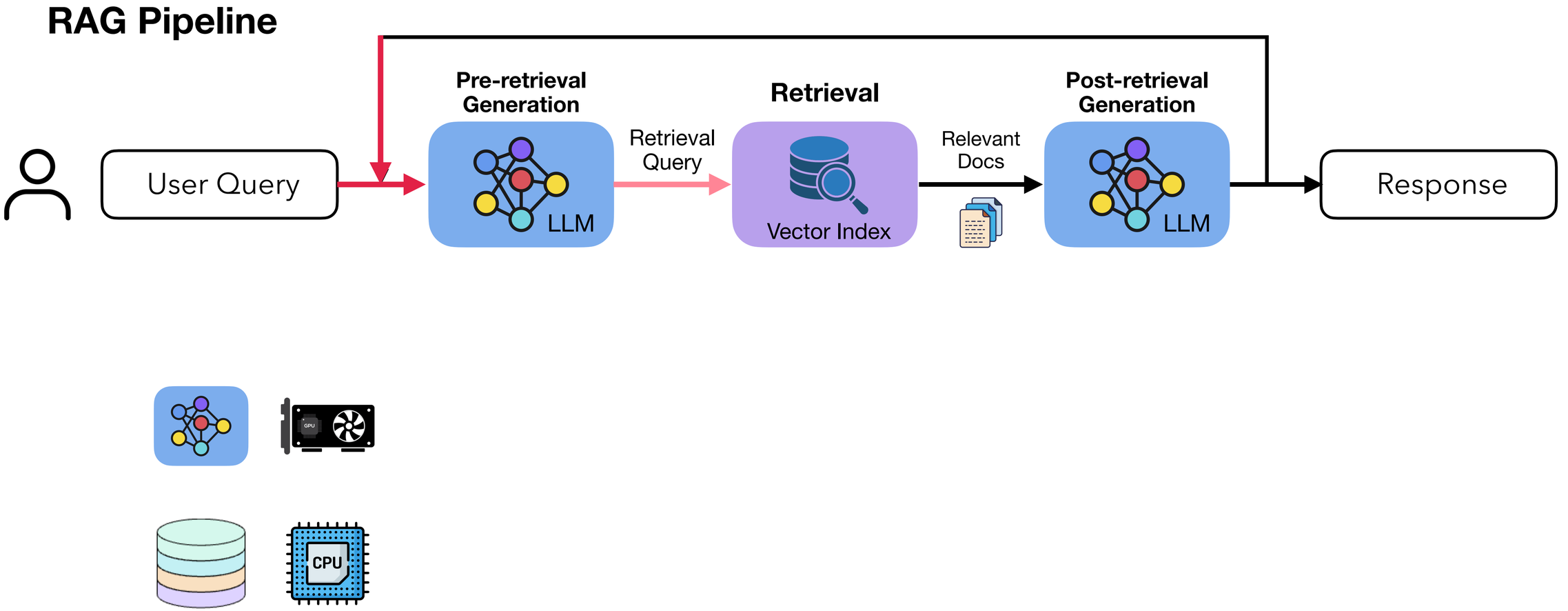
User Query and Retrieval Query share a larger portion of their IVF clusters!

* Test on a IVF index with 4096 cluster.

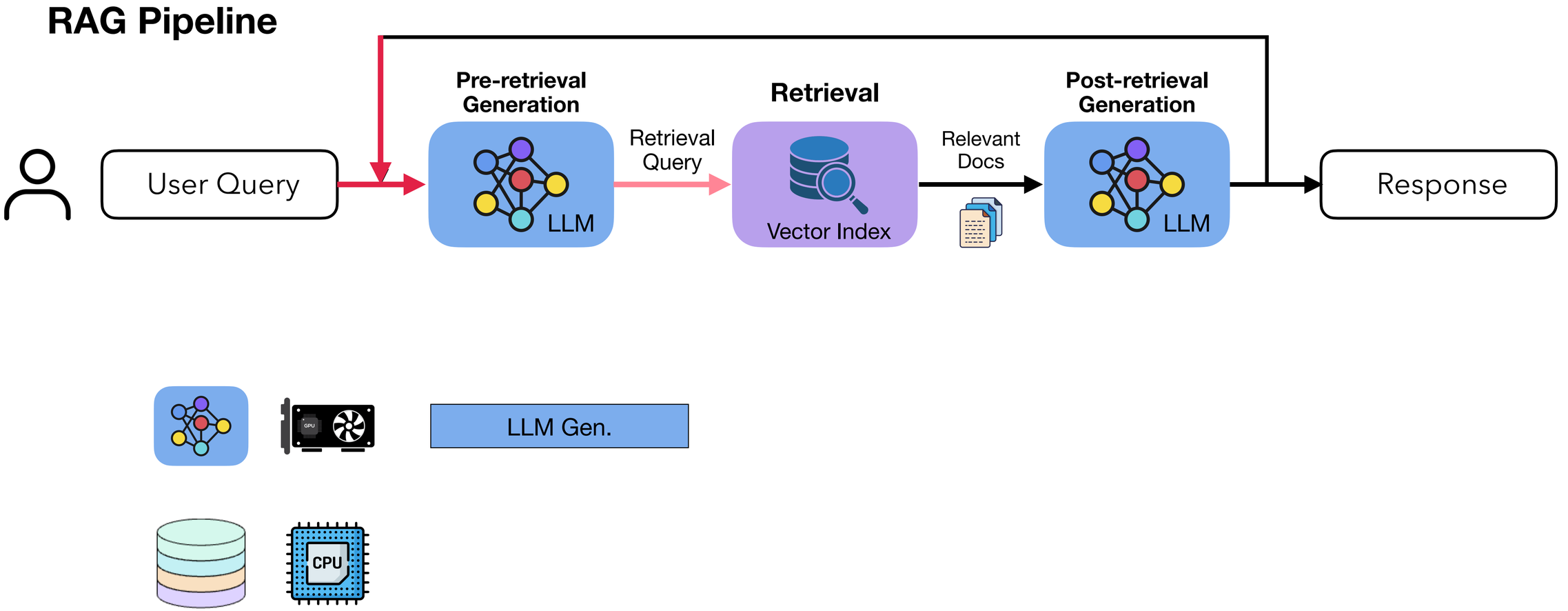
TeleRAG: Lookahead Retrieval



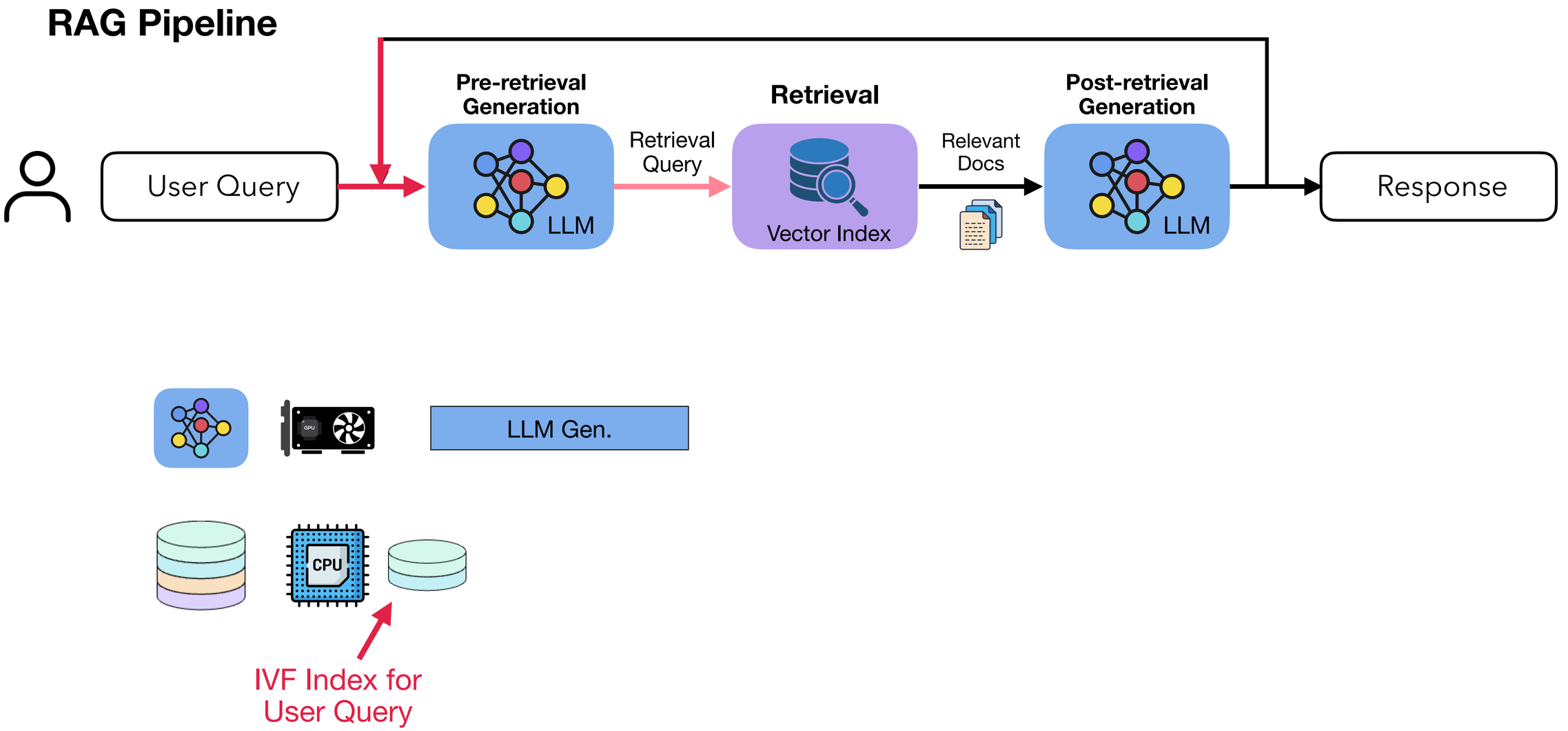
TeleRAG: Lookahead Retrieval



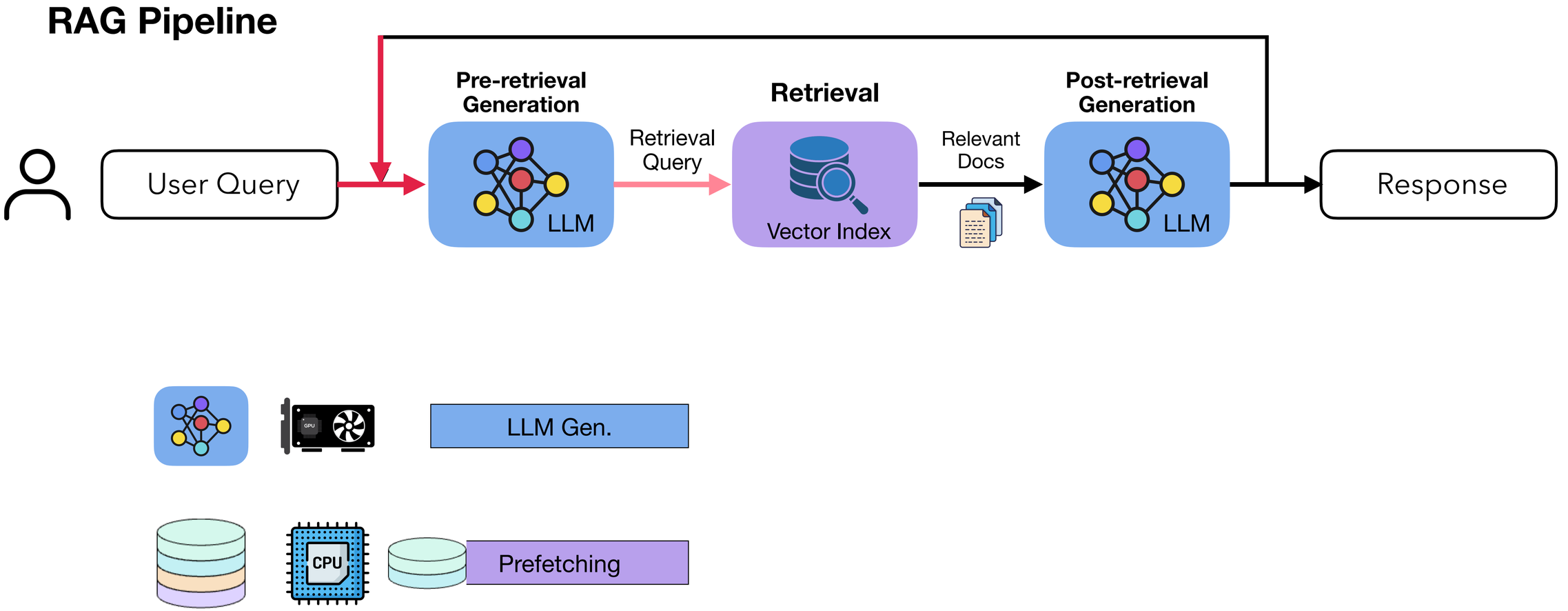
TeleRAG: Lookahead Retrieval



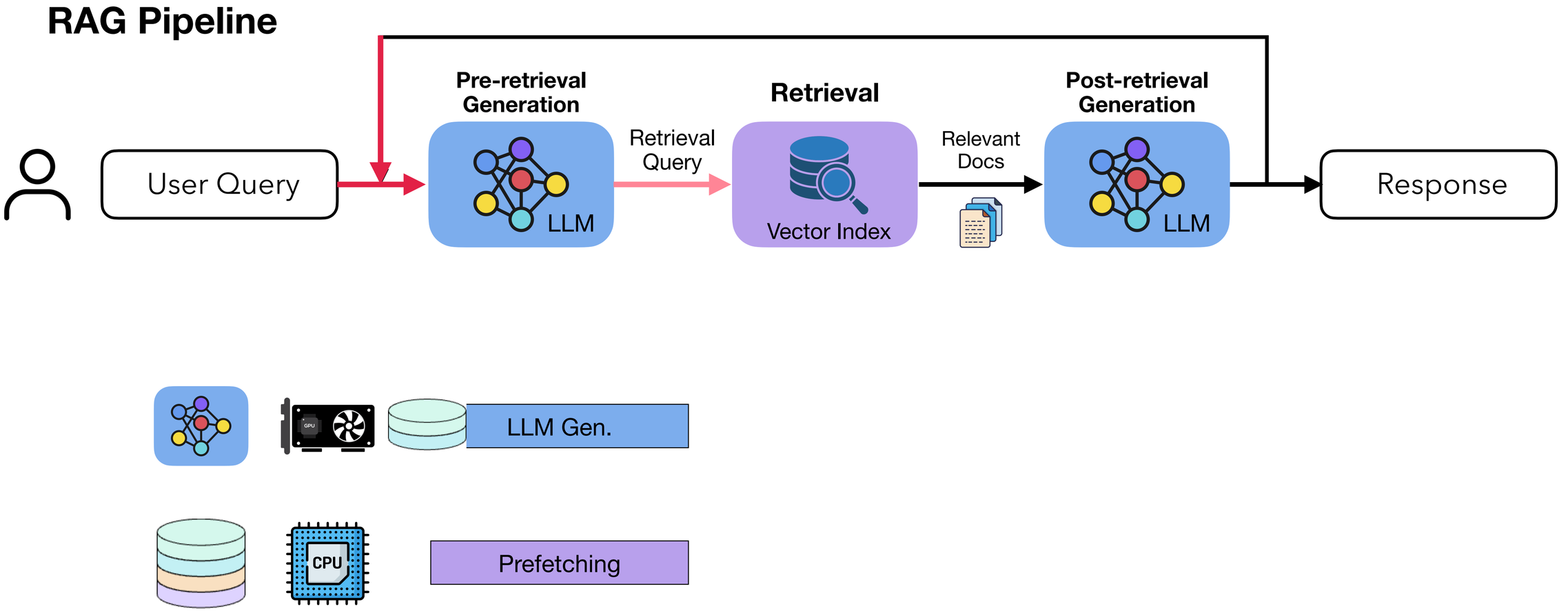
TeleRAG: Lookahead Retrieval



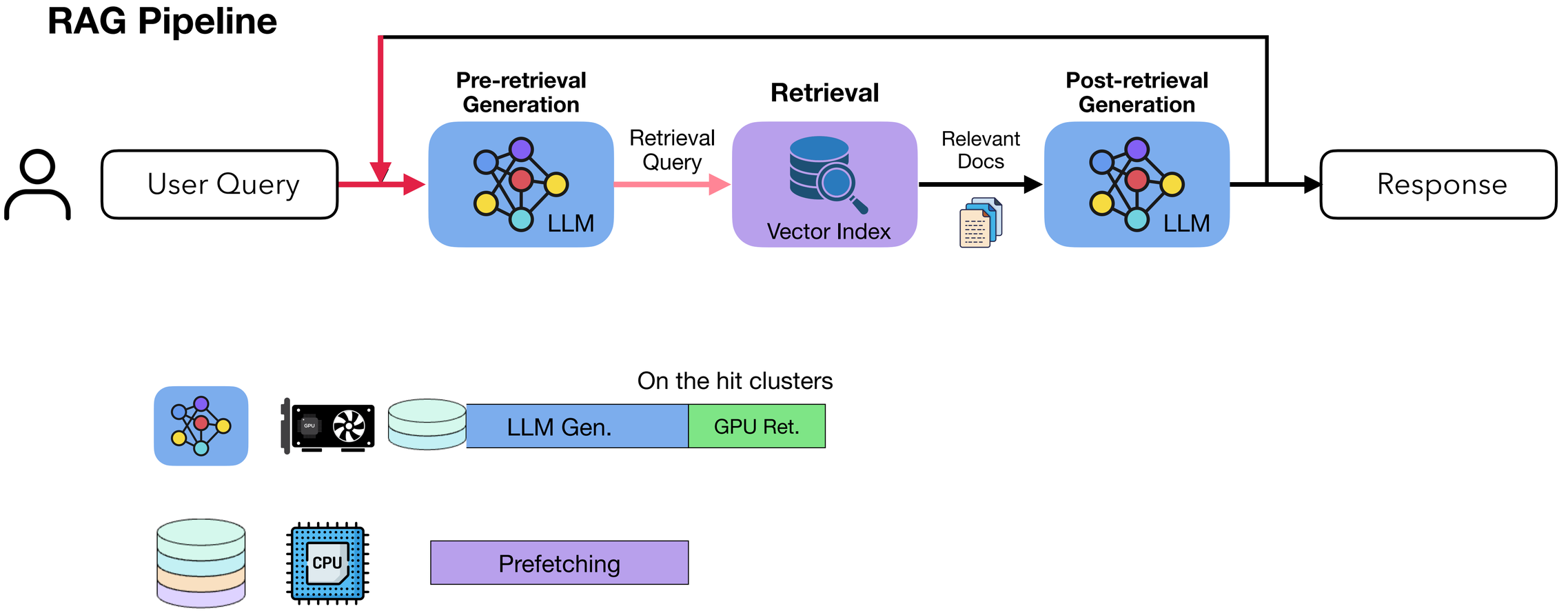
TeleRAG: Lookahead Retrieval



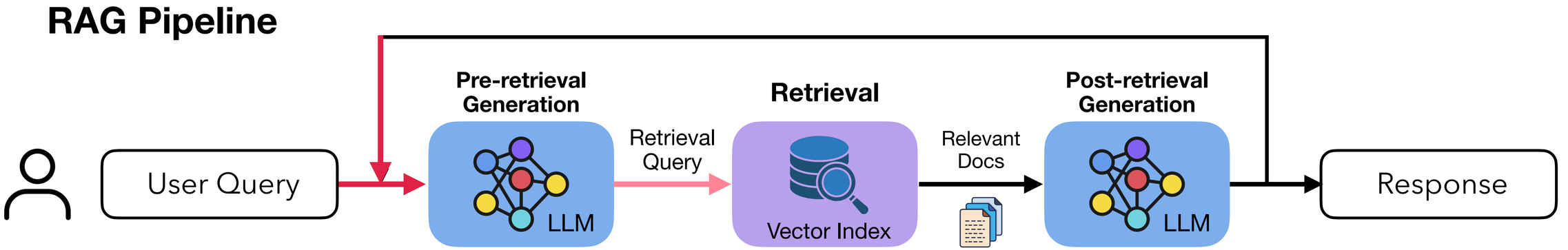
TeleRAG: Lookahead Retrieval

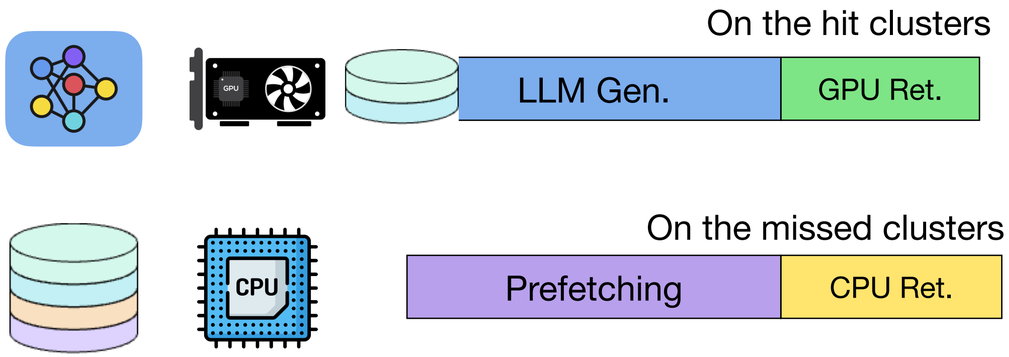


TeleRAG: Lookahead Retrieval

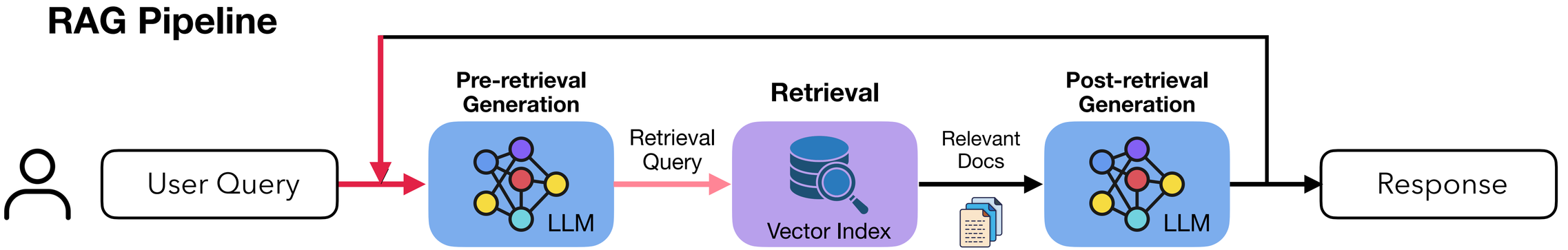


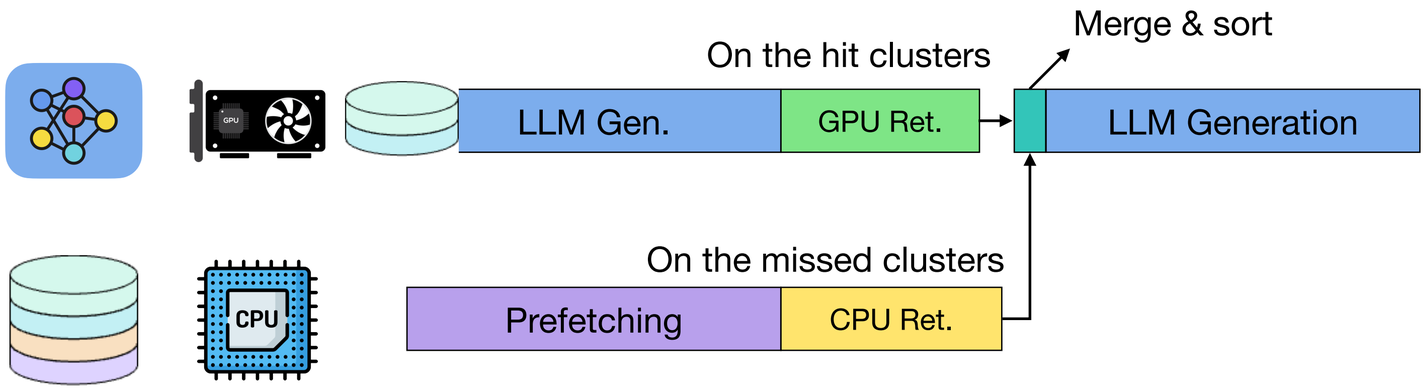
TeleRAG: Lookahead Retrieval



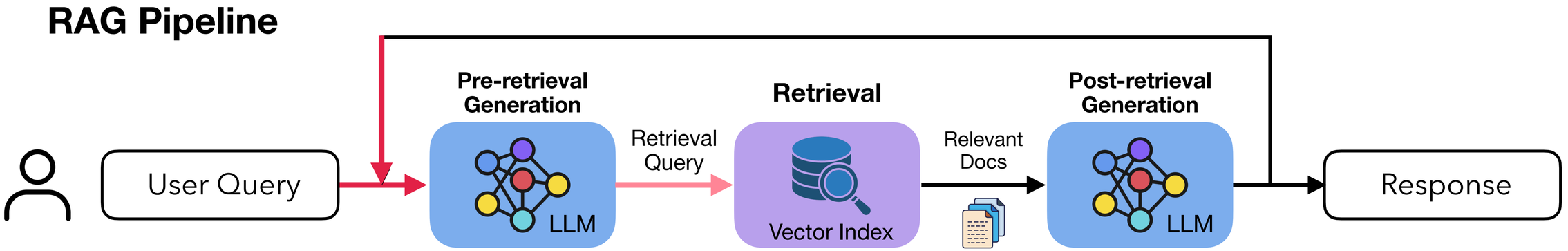


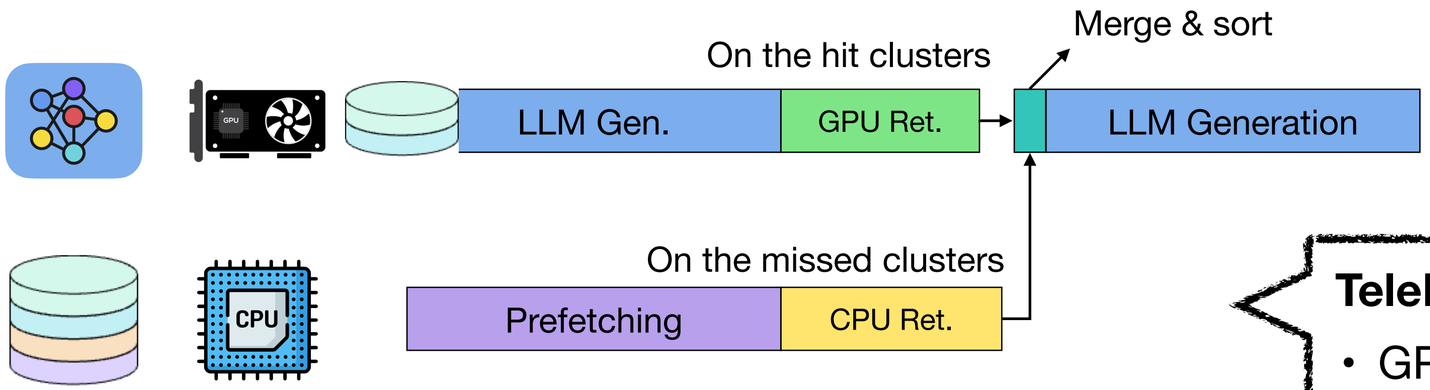
TeleRAG: Lookahead Retrieval





TeleRAG: Lookahead Retrieval

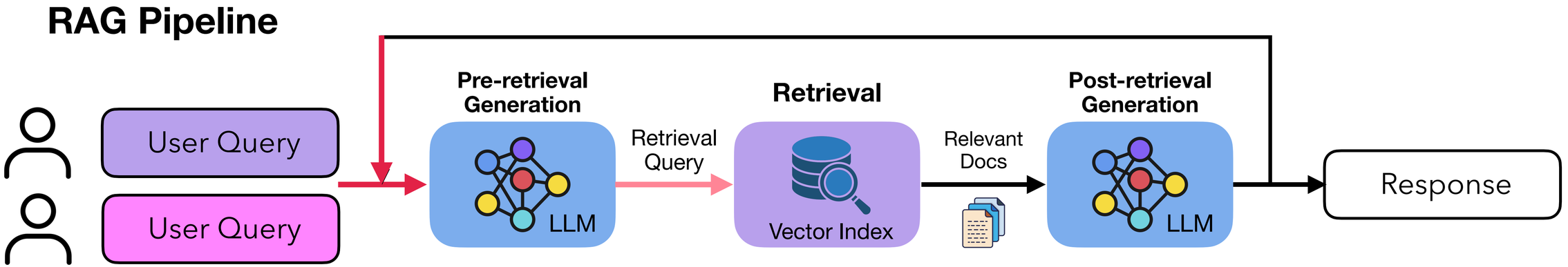


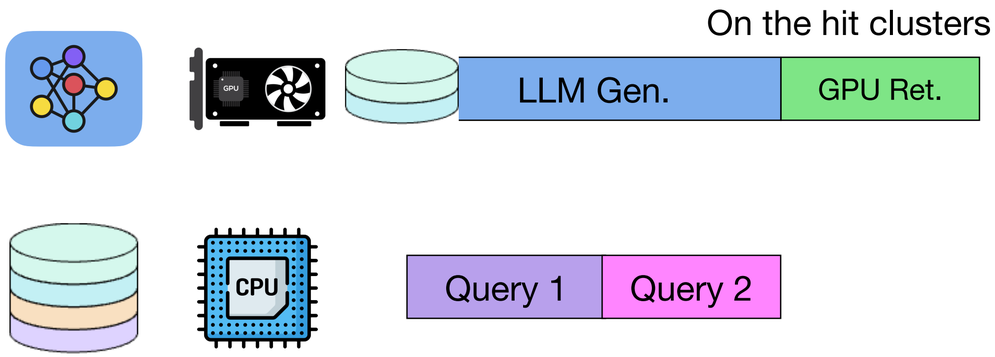


TeleRAG:

- GPU-like retrieval speed
- Minimal GPU memory requirement

Challenge of TeleRAG: Prefetching for Multi-Query

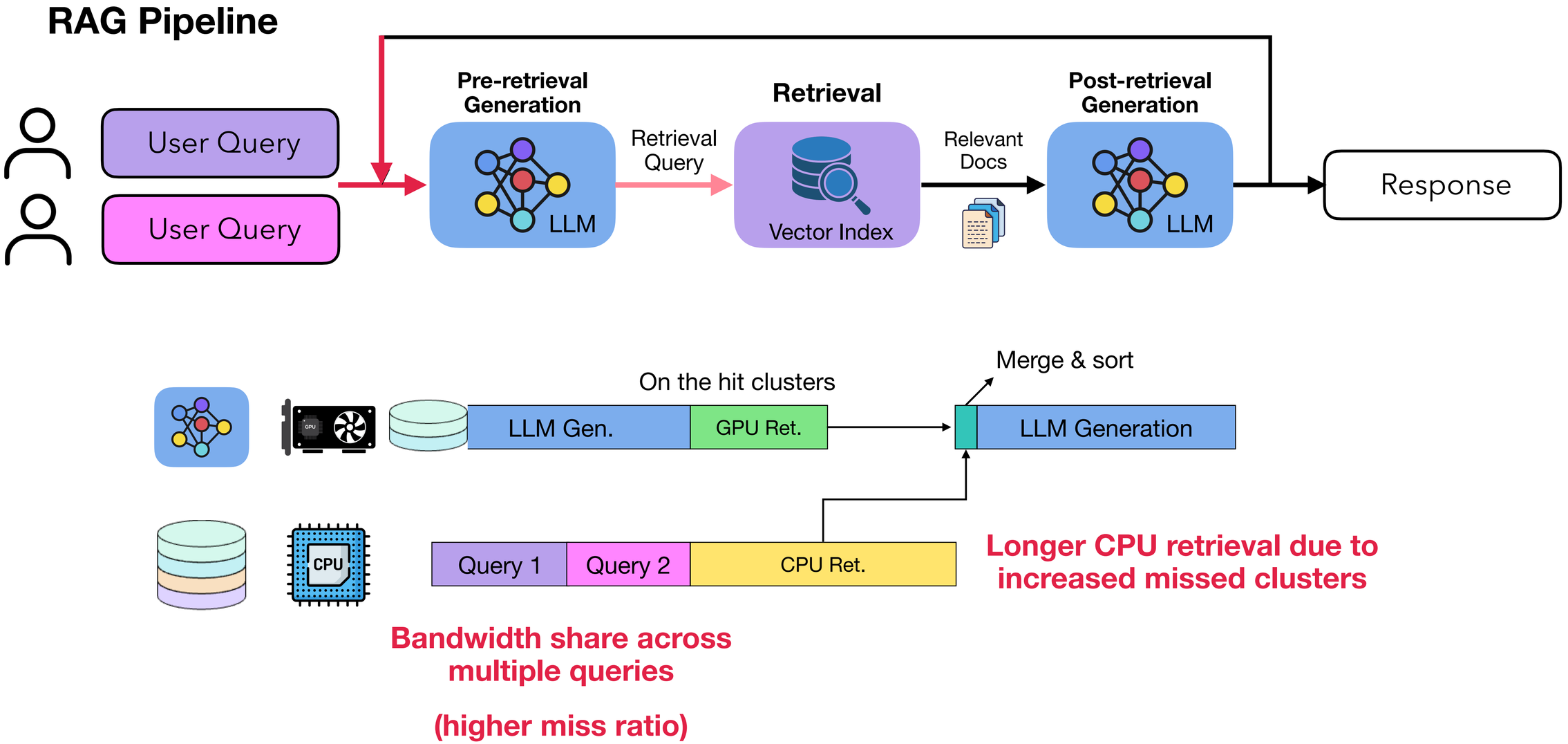




Bandwidth share across multiple queries

(higher miss ratio)

Challenge of TeleRAG: Prefetching for Multi-Query



TeleRAG: Batching (Prefetching Scheduler)

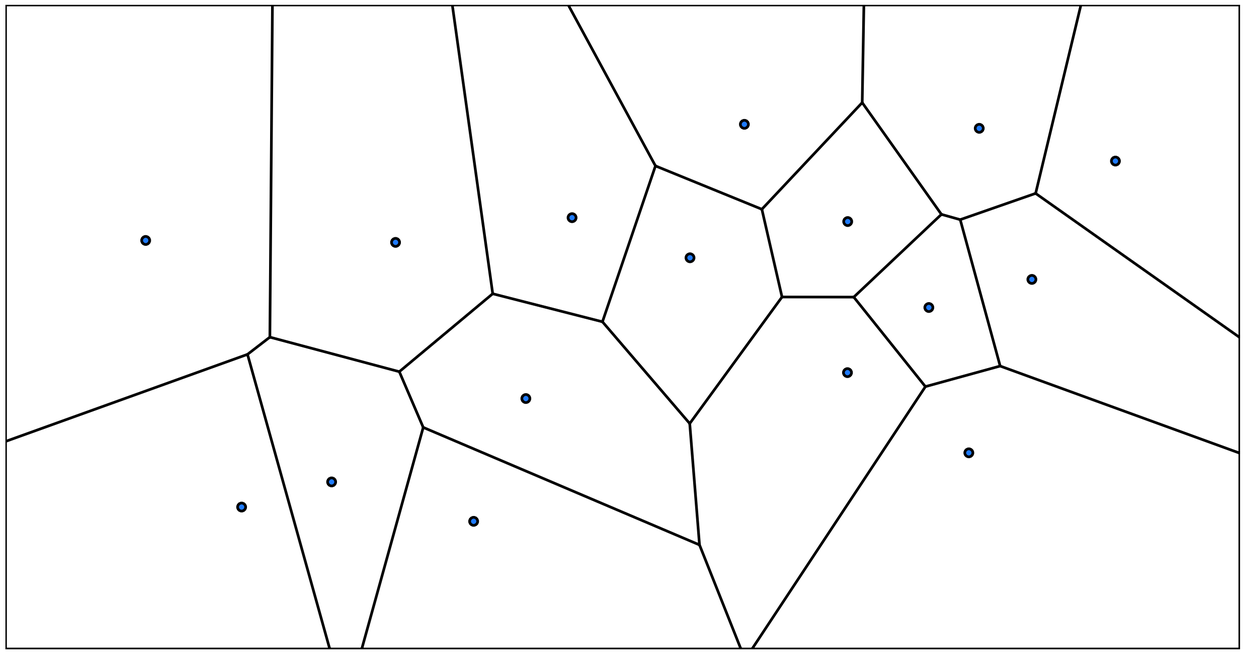
- Different queries can have diverse cluster sets

Query A

Query B

Query C

Query D



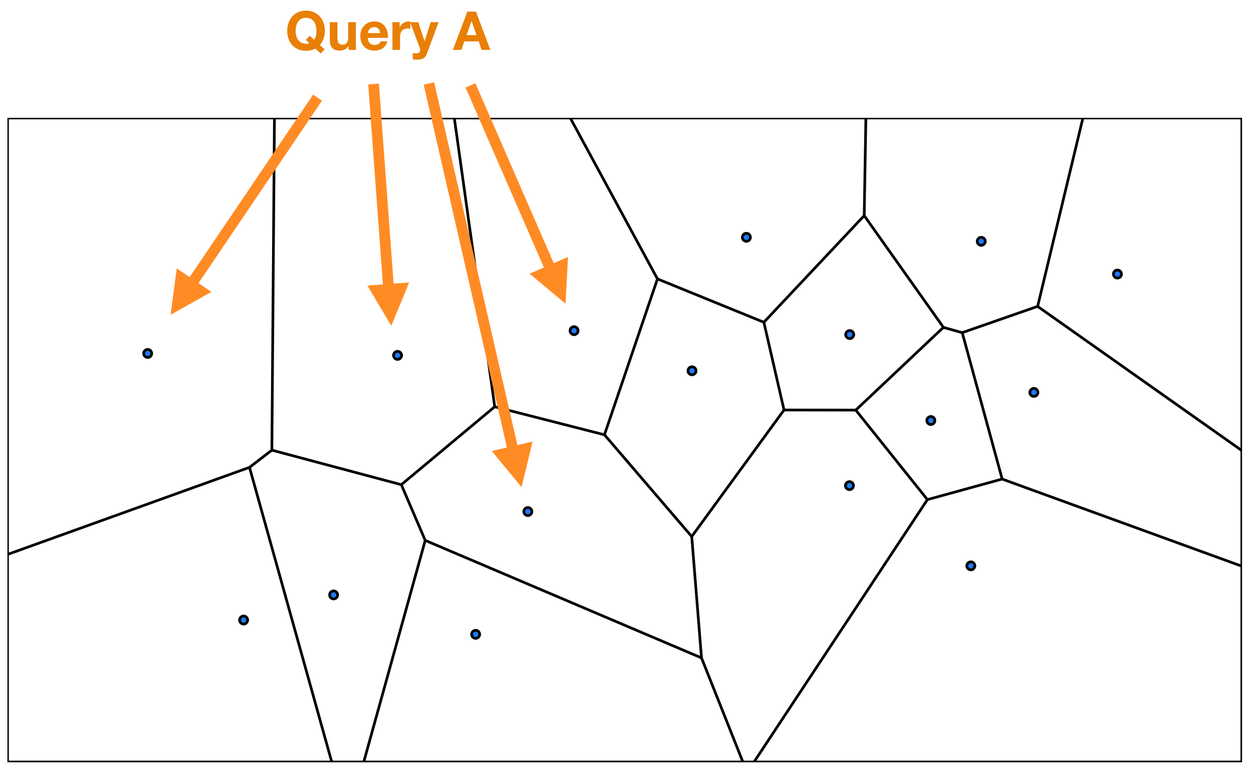
TeleRAG: Batching (Prefetching Scheduler)

- Different queries can have diverse cluster sets

Query B

Query C

Query D

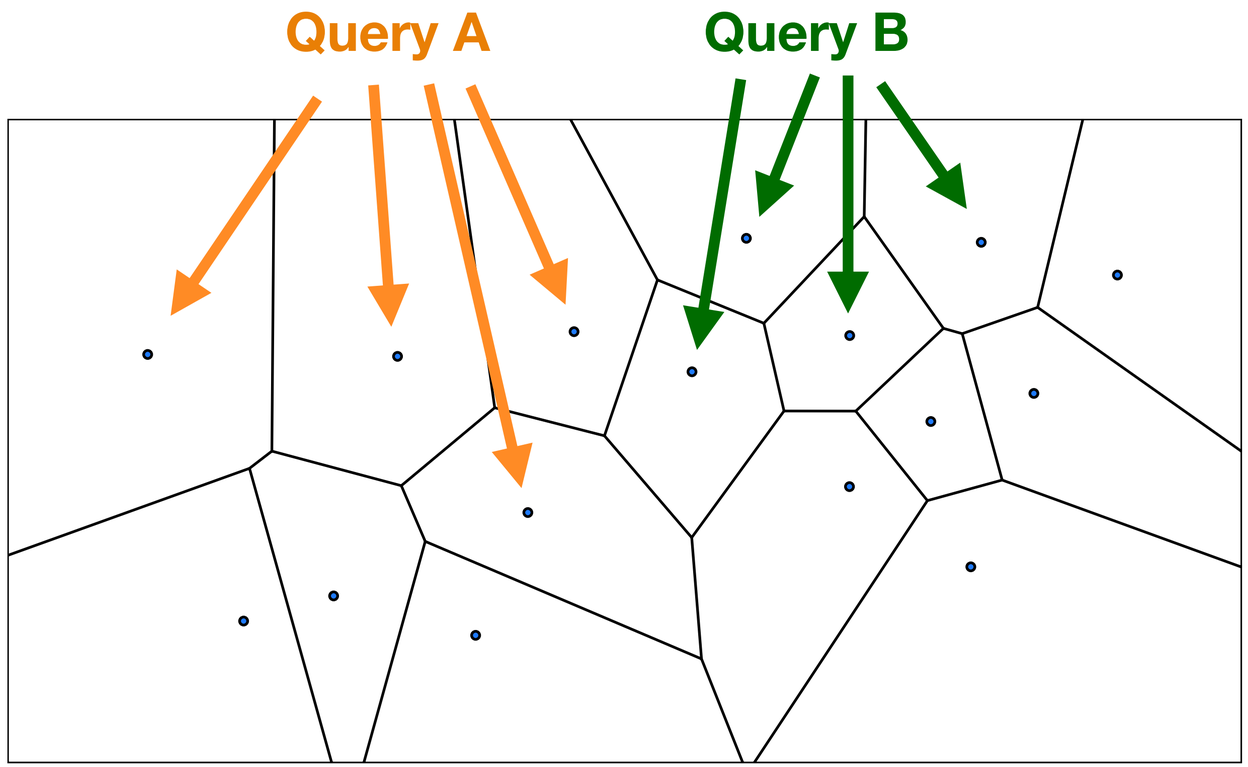


TeleRAG: Batching (Prefetching Scheduler)

- Different queries can have diverse cluster sets

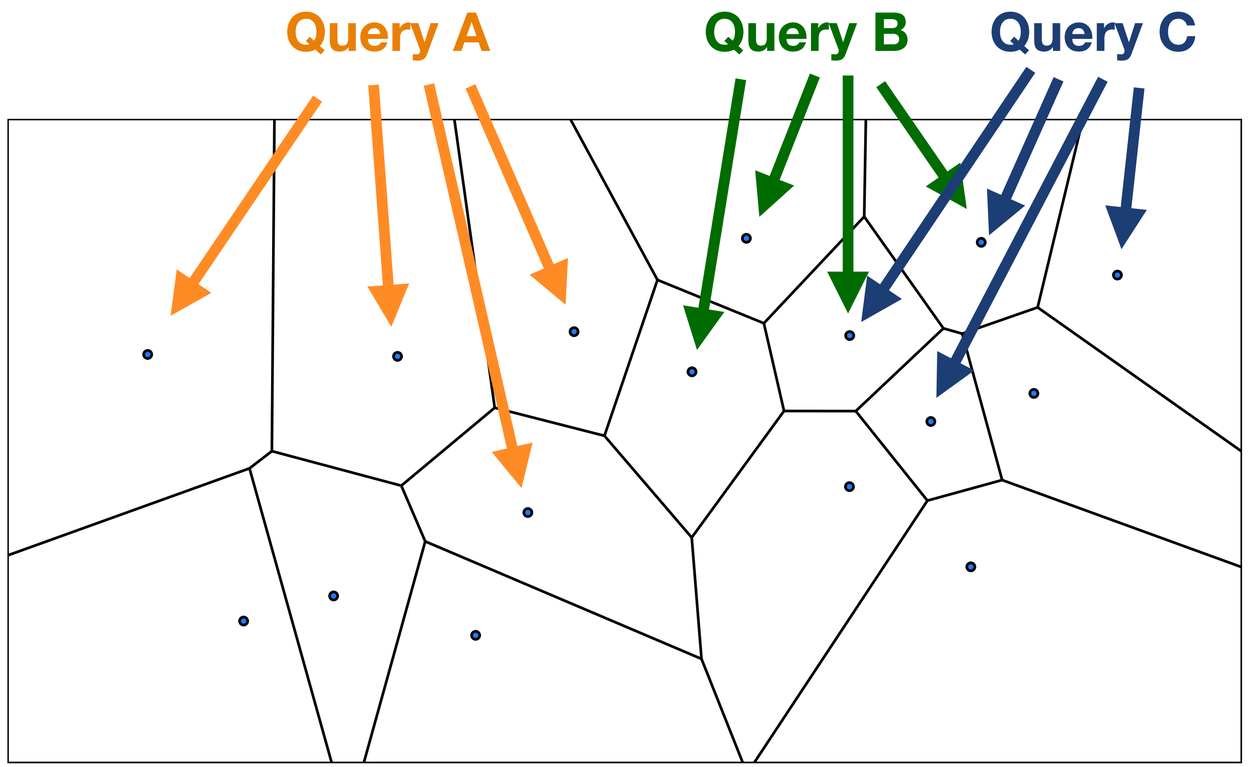
Query C

Query D



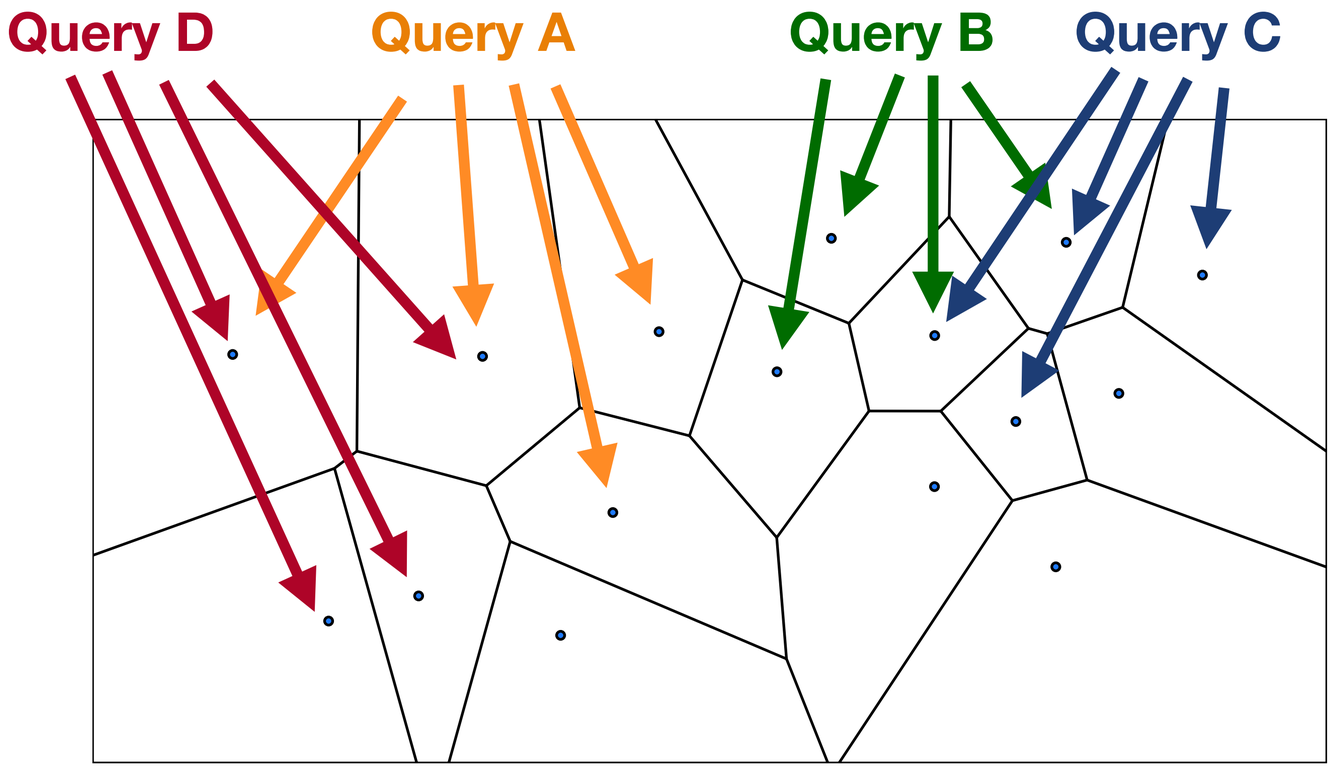
TeleRAG: Batching (Prefetching Scheduler)

- Different queries can have diverse cluster sets



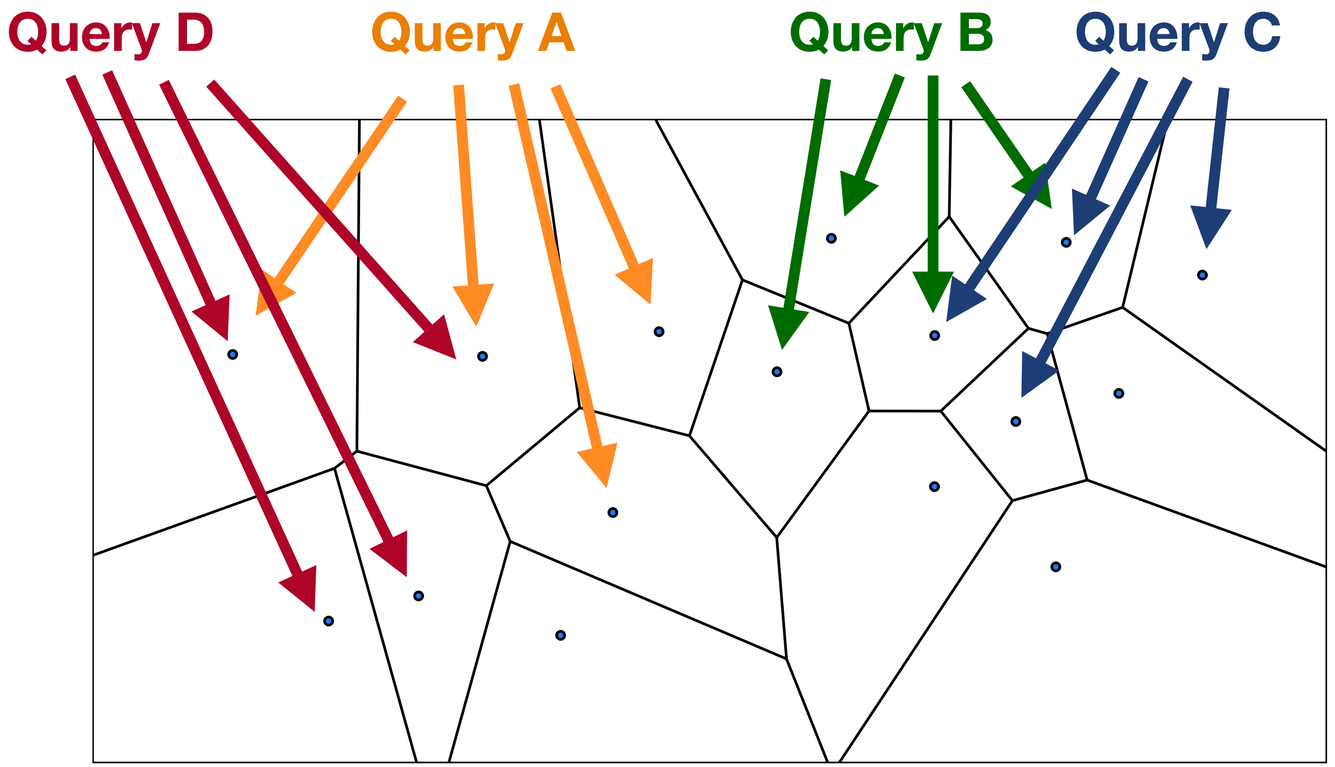
TeleRAG: Batching (Prefetching Scheduler)

- Different queries can have diverse cluster sets



TeleRAG: Batching (Prefetching Scheduler)

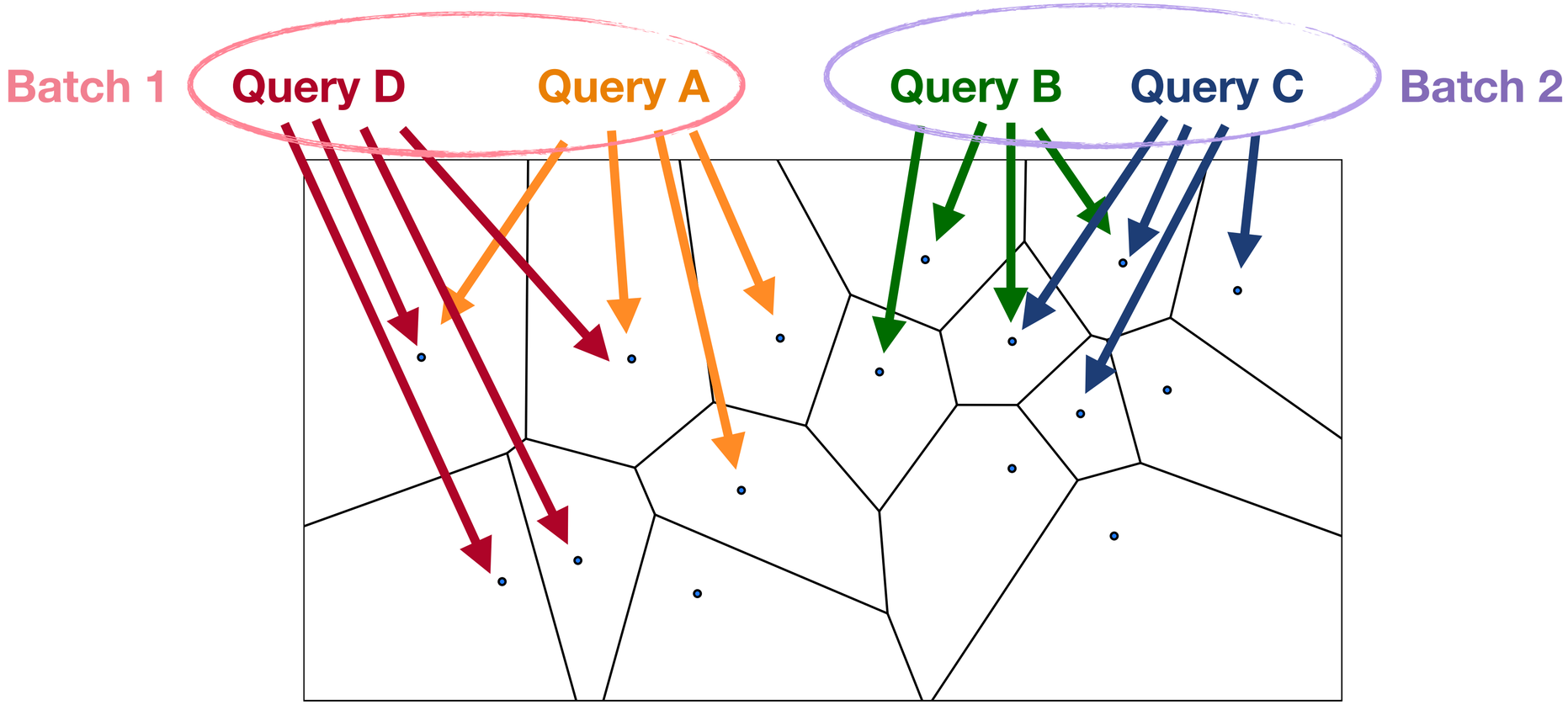
- Different queries can have diverse cluster sets



Ineffective batching strategies (e.g., pairing Query A with B, and Query C with D) can severely degrade prefetching performance.

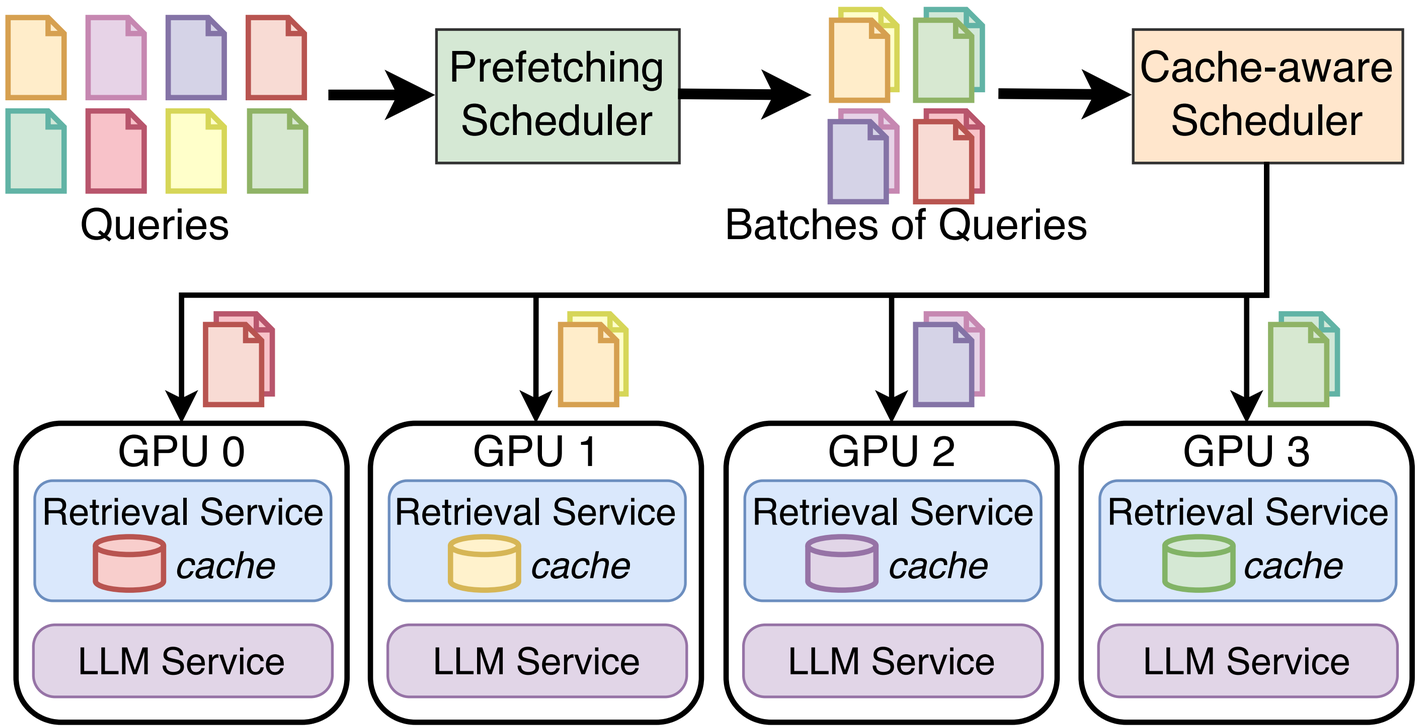
TeleRAG: Batching (Prefetching Scheduler)

- Different queries can have diverse cluster sets
- Solution: group queries with **higher semantic similarity**



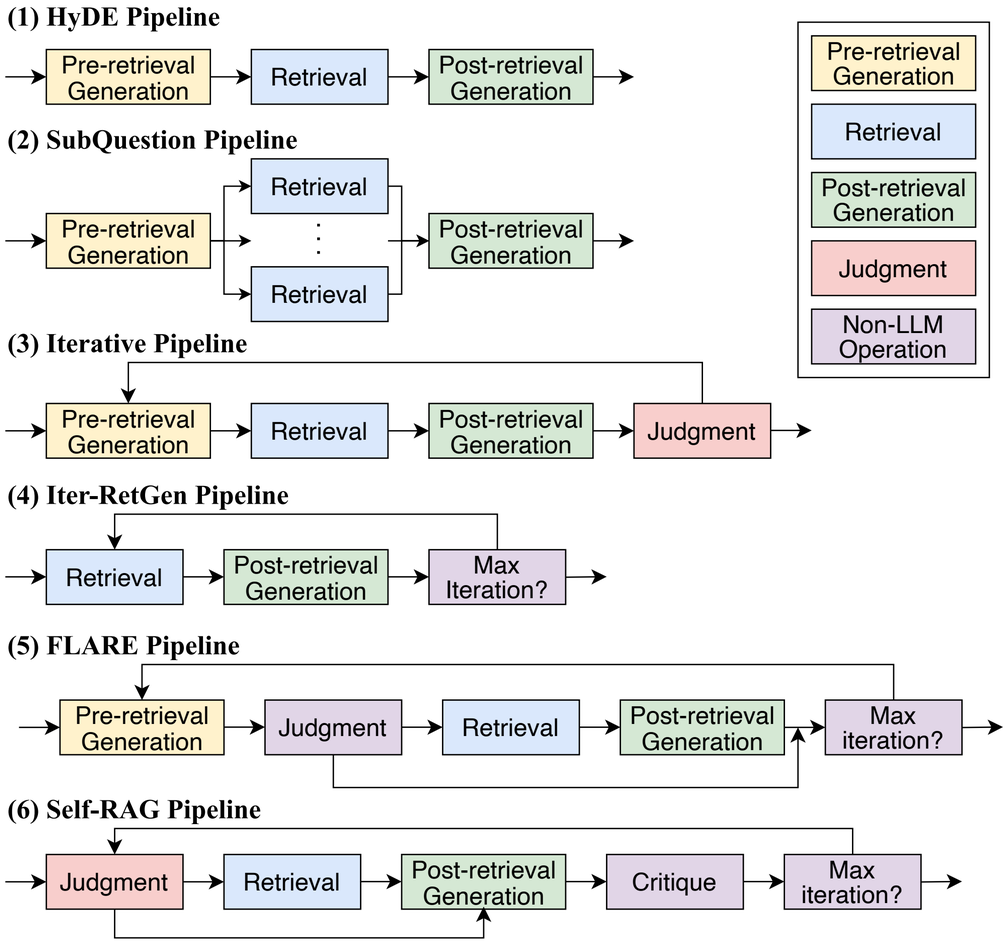
TeleRAG: Caching & Cache-aware Scheduling

- **Caching**: keep most frequently used IVF clusters in GPU memory
- **Cache-aware scheduling**: assign micro-batches based on cache locality



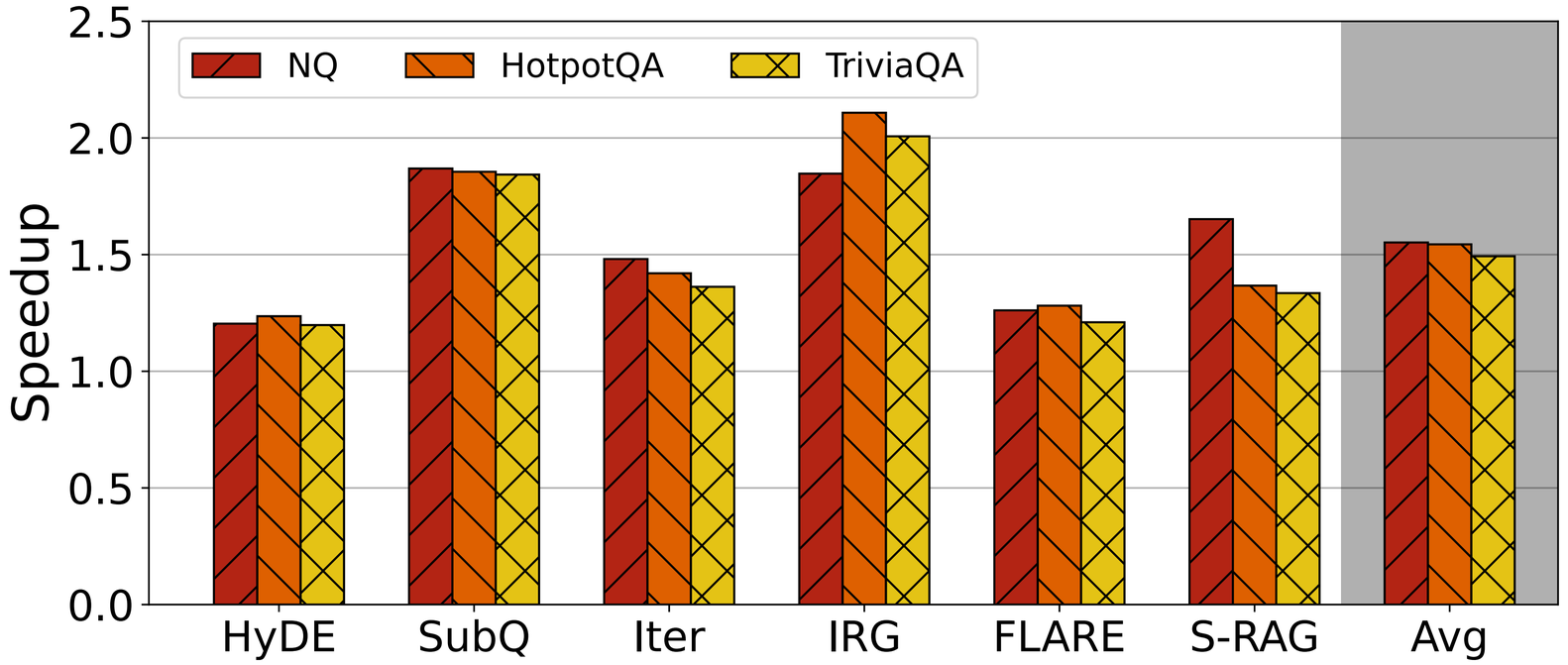
Evaluation Setup

- Evaluated on six representative RAG pipelines
- Retrieval index: 61GB IVF index with 4096 clusters
- Tested on three popular QA datasets:
  - NQ, TriviaQA and HotpotQA
- Baseline: Faiss-CPU + SGLang for LLM inference



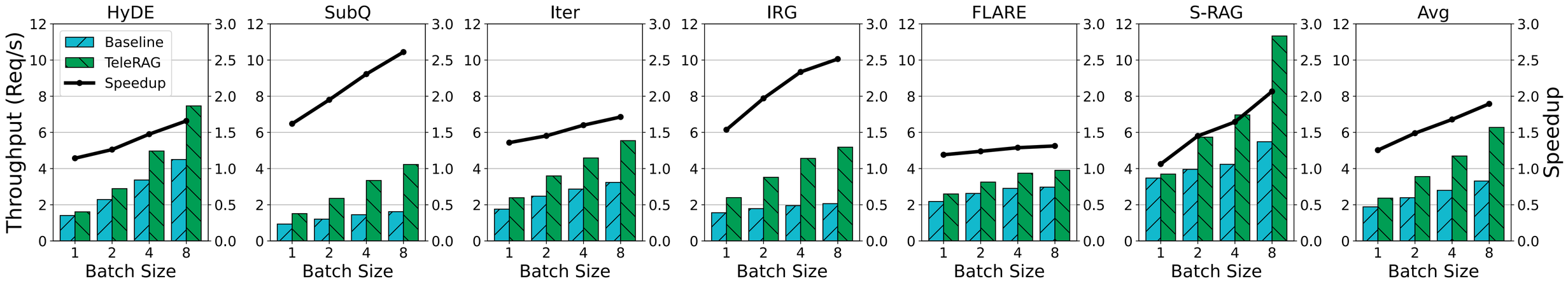
End-to-End Latency on RTX 4090 (24GB)

- With Llama3.2-3B LLM
- Over **1.5x** latency speedup on average



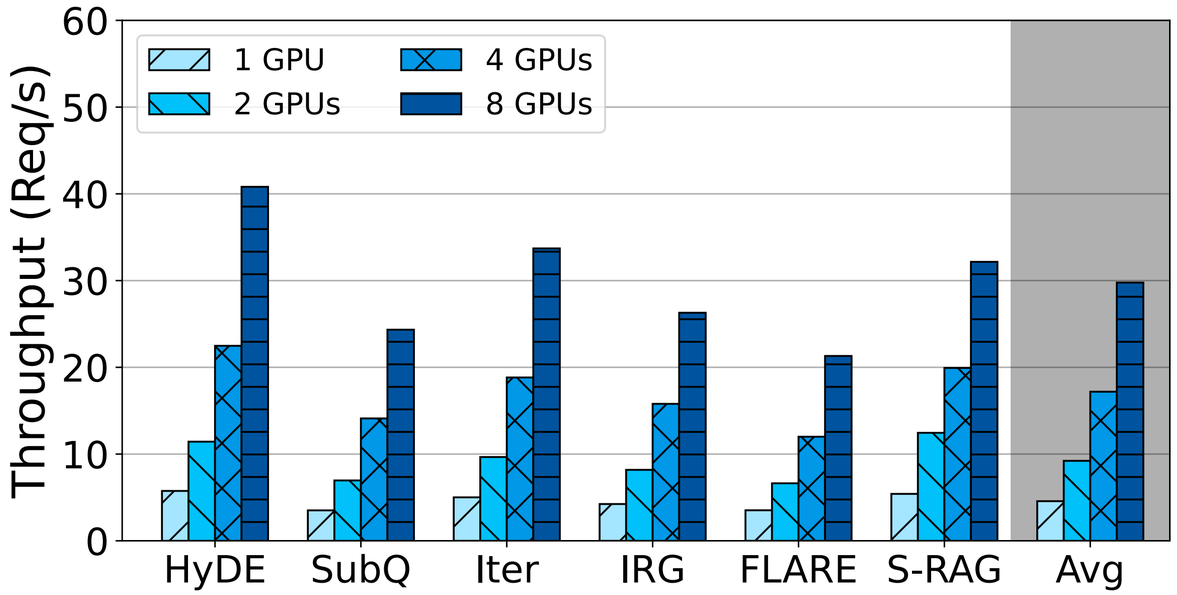
Throughput Enhancement on H100

- With Llama3-8B LLM on NQ dataset
- Increased throughput with a larger batch size
- Achieve 1.9x throughput increase at batch size 8 on average



Multi-GPU Scaling

- With Llama3-8B LLM on NQ dataset
- Evaluated on a 8-H200 node
- Strong scaling thanks to the prefetching and cache-aware schedulers



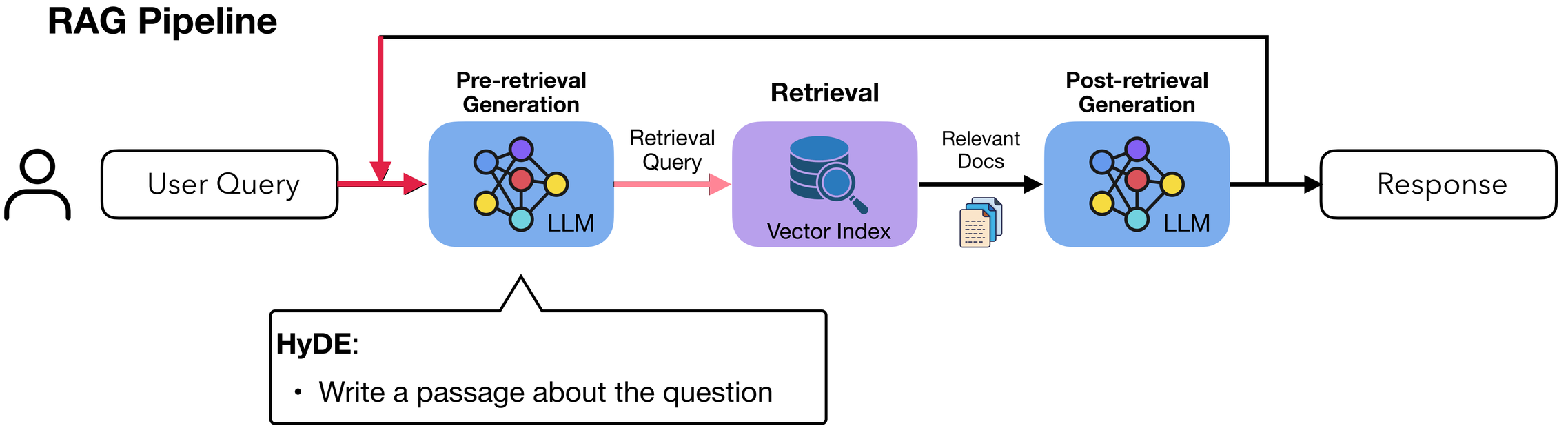
Conclusion

- We found **high similarity** between queries of **different stages** in RAG
- We proposed **lookahead retrieval** which prefetches IVF index data to GPU memory during LLM generation stage
- Alongside the lookahead retrieval, we designed optimizations for **batching, caching, and multi-GPU scheduling**
- With these, **TeleRAG** achieves high RAG performance from single to multi-GPU scenarios with reduced GPU memory usage

Q & A

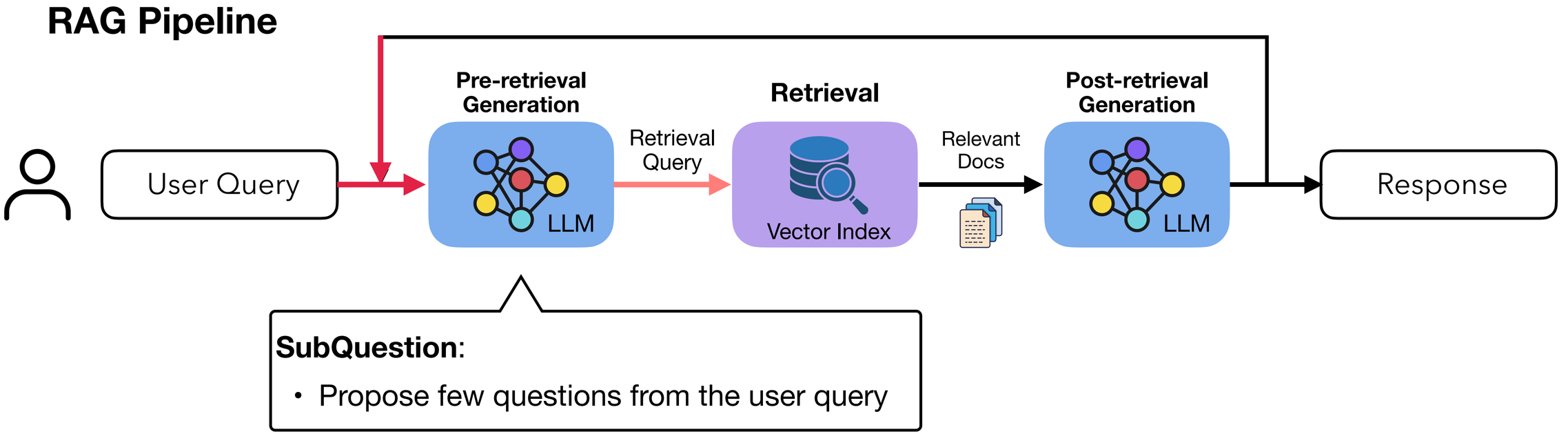
Appendix

Insight: Query Correlation Between Stages



Precise Zero-Shot Dense Retrieval without Relevance Labels, ACL 2023

Insight: Query Correlation Between Stages



https://docs.llamaindex.ai/en/stable/examples/query_engine/sub_question_query_engine/