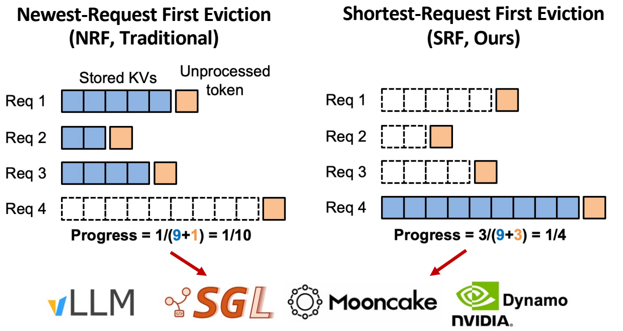
Harmonizing Data Systems, LLMs, and Vectors :)

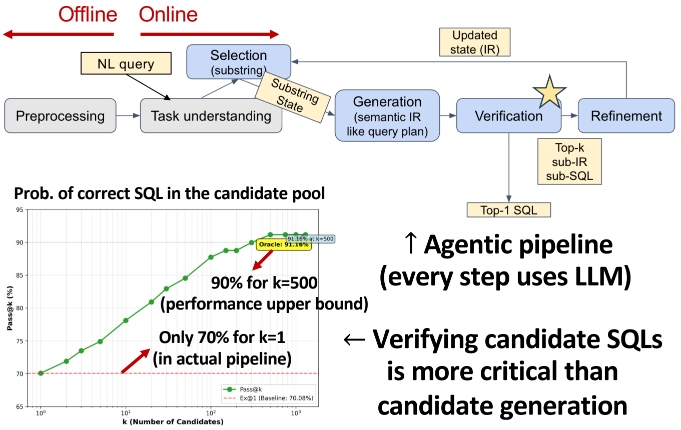
Data-Intensive Applications and Systems Laboratory

LLM Inference System = Data System



Data management is crucial (~20% GPU hour saving in LLM inference by doing better KV cache replacement)

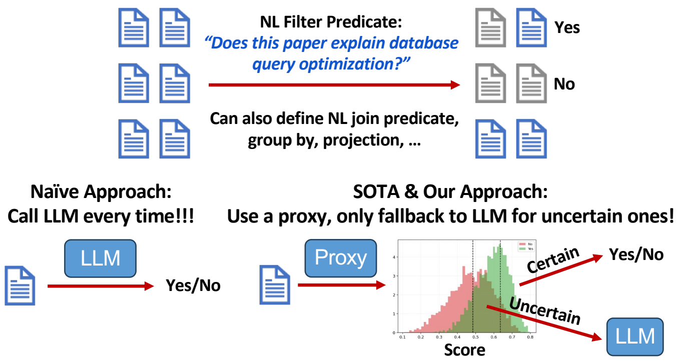
Agentic LLM Pipeline for Translating NL to SQL



↑ Agentic pipeline (every step uses LLM)

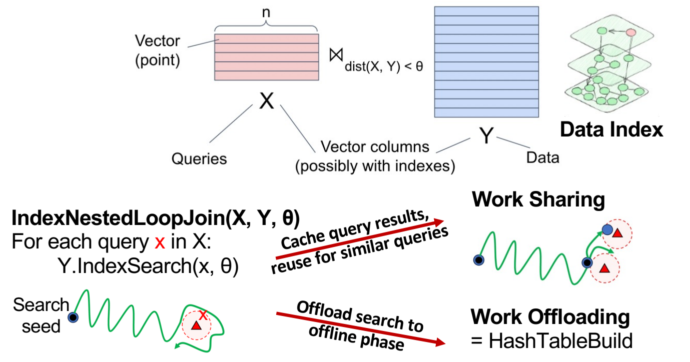
← Verifying candidate SQLs is more critical than candidate generation

Bulk LLM Calls for Semantic Operators


NL Filter Predicate:
"Does this paper explain database query optimization?"
Can also define NL join predicate, group by, projection, ...
Yes
No
Naïve Approach:
Call LLM every time!!!
LLM
Yes/No
SOTA & Our Approach:
Use a proxy, only fallback to LLM for uncertain ones!
Proxy
Certain
Yes/No
Uncertain
LLM
Score

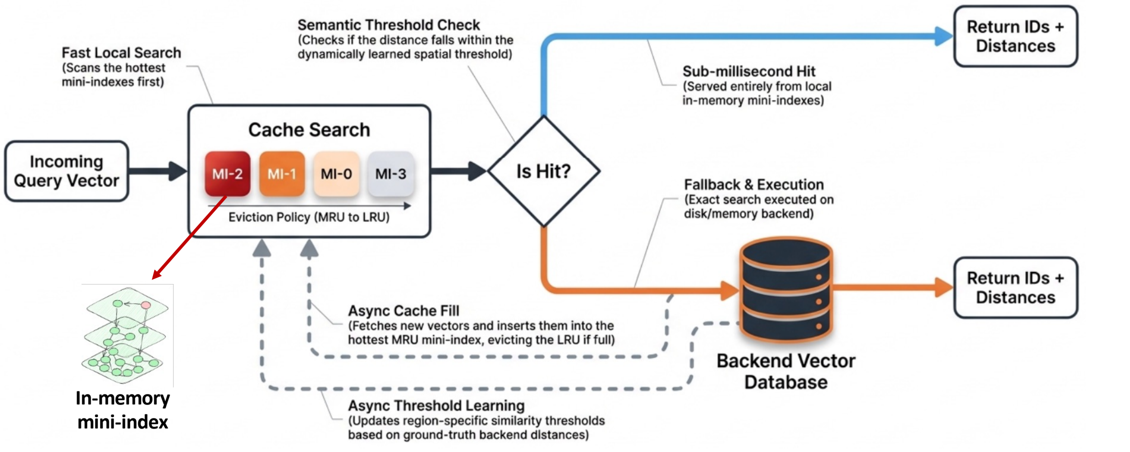
Saving # LLM calls by ~99%, and 50% on average

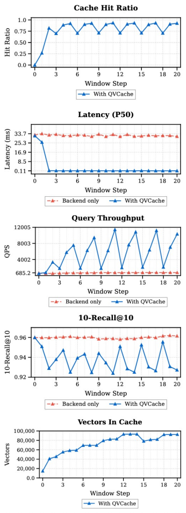
Why Not Vector *Joins* for Batch Query Proc.?



Work sharing/offloading: ~33x speedup, ~43% higher recall

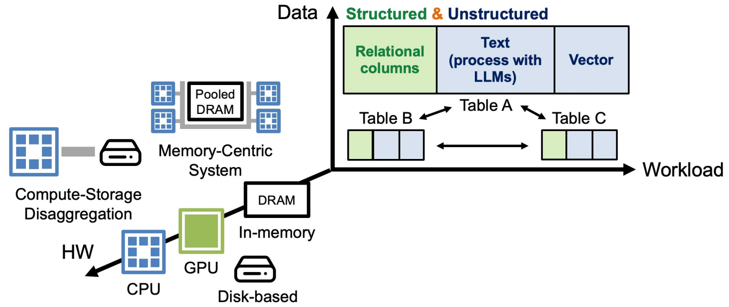
In-memory Similarity Caching for Disk-based Vector Search


Incoming
Query Vector
Fast Local Search
(Scans the hottest
mini-indexes first)
Cache Search
MI-2
MI-1
MI-0
MI-3
Eviction Policy (MRU to LRU)
Semantic Threshold Check
(Checks if the distance falls within the
dynamically learned spatial threshold)
Is Hit?
Sub-millisecond Hit
(Served entirely from local
in-memory mini-indexes)
Return IDs +
Distances
Fallback & Execution
(Exact search executed on
disk/memory backend)
Backend Vector
Database
Return IDs +
Distances
Async Cache Fill
(Fetches new vectors and inserts them into the
hottest MRU mini-index, evicting the LRU if full)
Async Threshold Learning
(Updates region-specific similarity thresholds
based on ground-truth backend distances)
In-memory
mini-index



DB-agnostic caching delivers 40-1000x lower query latencies on cache hits

Our End Goal: HW-Optimized System Supporting Hybrid Relational/Vector/Semantic Queries



Lots of open challenges to solve!!!

- One-size-fits-all solution for filtered vector search
- Declarative control of accuracy-efficiency trade-off
- Query optimization of vector/semantic queries
- LLM for query optimization and direct plan writing

Under large & complex data, queries (100's ops), and HW configurations