

Databases in the Age of AI

North East Database Day, Brandies University,
Jan 9, 2025

Dhruba Borthakur
Data Infrastructure Lead, OpenAI

Introduction

- Data Infra Lead, OpenAI
- Co-founder, Rockset
- Founding Engineer, RocksDB
- Founding Architect, Hadoop FS
- Contributor: HBase, AFS, Condor

The Dawn of AI with ChatGPT

First version of ChatGPT from OpenAI

- released Nov 30, 2022
- a Research Prototype
- understands natural language
- responds in natural language



Model Training

- needs lots of compute for training
- needs lots of real-world data to train
- no significant role for databases in this phase

The Rise of the Reasoning Models

Now is the time for reasoning models

- o1 production released in 2024
- Gemini2 experimental
- o1 thinks really hard before responding



Why are Reasoning Models useful?

- healthcare researchers: annotate cell sequencing
- software developers: build multi-step workflows
- physicists: generate mathematical formulas

<https://today.ucsd.edu/story/how-artificial-intelligence-could-automate-genomics-research>

Simple example of reasoning

Question:

Given that the sum of three consecutive even numbers is 30, find the middle number

ChatGPT says:

Let the three consecutive even numbers be: $x-2$ (smallest), $x+2$ (largest)

The sum of the three numbers is given as 30:

$$(x-2)+x+(x+2)=30$$

Simplify the equation: $3x=30$

Solve for x , $x=10$

The middle number is: 10

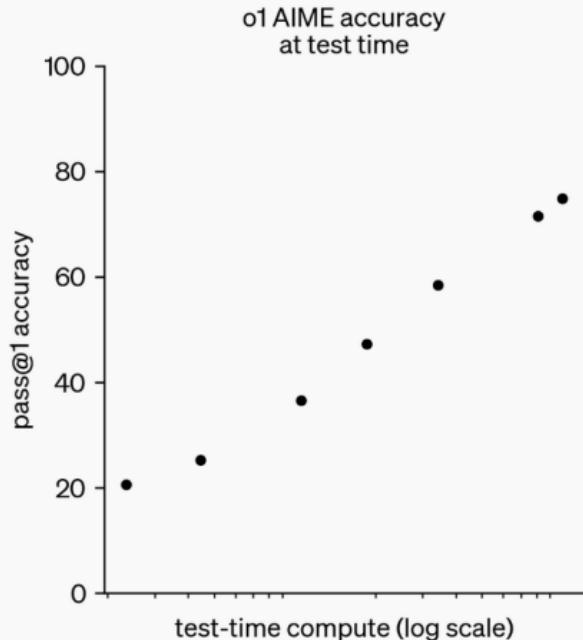
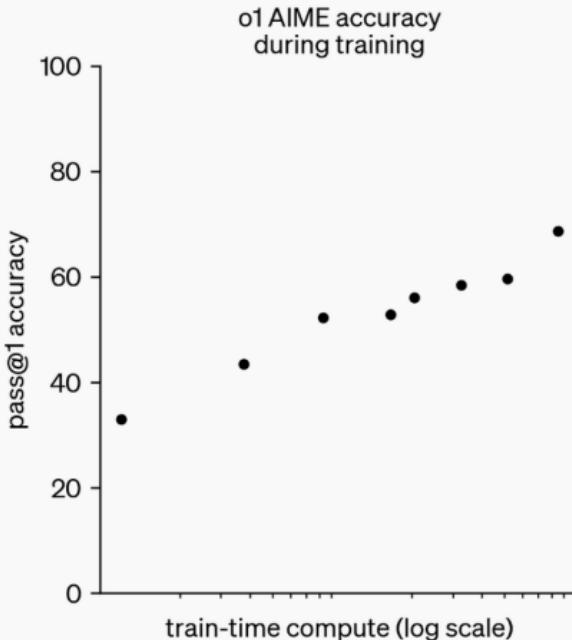
Reasoning needs

- Lots of compute at inference (test-time compute)
- lots of memory for storing intermediate steps (chain of thought)

AI Memory

- "memory" is how AI people refer to context data
- what can we do to increase the size of 'memory'?
- is a database needed for this functionality?
- accessible from model code running on GPUs
- extend RAM with SSD

Training vs Inference



New importance of
Inference Time (test-
time) compute

o1 performance smoothly improves with both train-time and test-time compute

Specialized Models

- chats: 40
- reasoning: o1 and o3 (coming soon)
- speech to text: Whisper
- text to image: DALLE

Pick most appropriate model(s) at inference time!

Combine multiple models at inference time

- Combine multiple models to analyze input, text, speech and images
- Combine multiple models to generate output, Send the same query to multiple models, evaluate their accuracy of predictions and let the best one win

Can be done only at inference time!

Three Trends for 2025

1. training is not the only way to improve model accuracy
 - enterprises using their own data to fine-tune models
2. reasoning models need large memory
 - need fast database access at inference time
3. combine multiple models at inference time

AI focus is moving from training to inference

Enterprises starting to use their own data to improve model accuracy

Enterprise databases & AI

Enterprise data locked in databases

- text to SQL (not very significant)
- associate semantic meaning to database columns and their relationships via prompt engineering (very significant)

Semantic meaning via Prompts

Sample Database: `data_records`

a1	b2	c3	d4	e5	f6
101	JSmith	10/12/2022	\$1200	XYZ Corp	\$200
102	MDoe	11/15/2022	\$1500	ABC Inc	\$300
103	TKing	09/18/2022	\$1000	DEF Ltd	\$150

What is the total sales amount for employees at
XYZ Corp in October 2022?

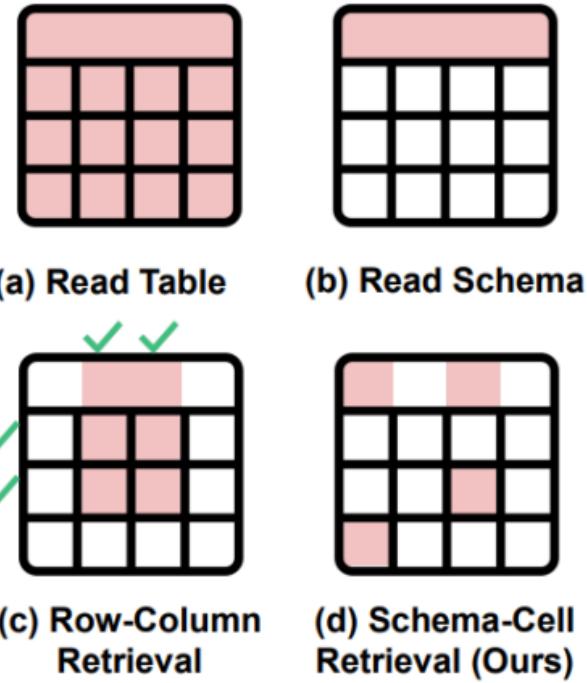
Sample Database: data_records

a1	b2	c3	d4	e5	f6
101	JSmith	10/12/2022	\$1200	XYZ Corp	\$200
102	MDoe	11/15/2022	\$1500	ABC Inc	\$300
103	TKing	09/18/2022	\$1000	DEF Ltd	\$150

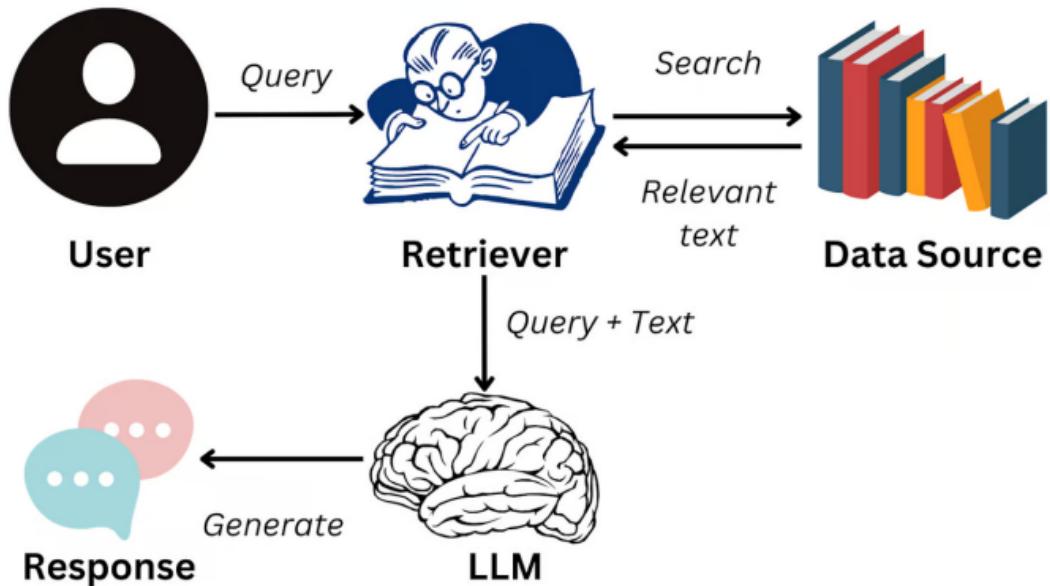
"Given the database where a1 = Employee ID,
b2 = Employee Name, c3 = Transaction Date,
d4 = Total Sales Amount, e5 = Employer Name,
and f6 = Profit Amount, calculate the total sales
for employees at XYZ Corp in October 2022."

Querying databases via LLM

- (a) Read Table: The LM reads the entire table
- (b) Read Schema: The LM reads only the schema
- (c) Row-Column Retrieval: Rows and columns are encoded and then selected based on their similarity to the question. Only the intersection of these rows and columns is presented to the LM.
- (d) Schema-Cell Retrieval: Column names and cells are encoded and retrieved based on their relevance to LM-generated queries about the question. Only the retrieved schema and cells are provided to the LM, enhancing efficiency in both encoding and reasoning



Enterprise data via RAG



source: datacamp.org

Basic Requirement

- need a search database

Challenges:

- chunk size
- context window size
- query rewriting
- distillation

Chunking challenge

- Traditional: gather sentences, with a size limit
- New Agentic approach:
- use the Model itself to chunk each document
- need large context window for large documents
- compute grows exponentially with larger context window

Can we build a database that can intelligently chunk documents?

Memory Challenge for large Context

- Transformers store relationships between all tokens in the window.
- Larger windows mean more relationships to store, which can overload memory

Can we build a database that can provide memory for a Model?

Memory Augmented Models

- Uses an external database to enhance Model memory
- Latency is primary design goal of such a database
- database can store data in SSD storage
- database has to be accessed from Model code running on GPU

Query Rewriting for a large query

- Chunk the query itself
- Summarization: use the model to 'summarize' the query and then use traditional RAG query flow

The Birth of Retrieval Databases

The Key Characteristics of a Retrieval Database:

- Search Optimized
- Realtime: data never stops and queries never stop
- Programmatic queries: high qps and low latency

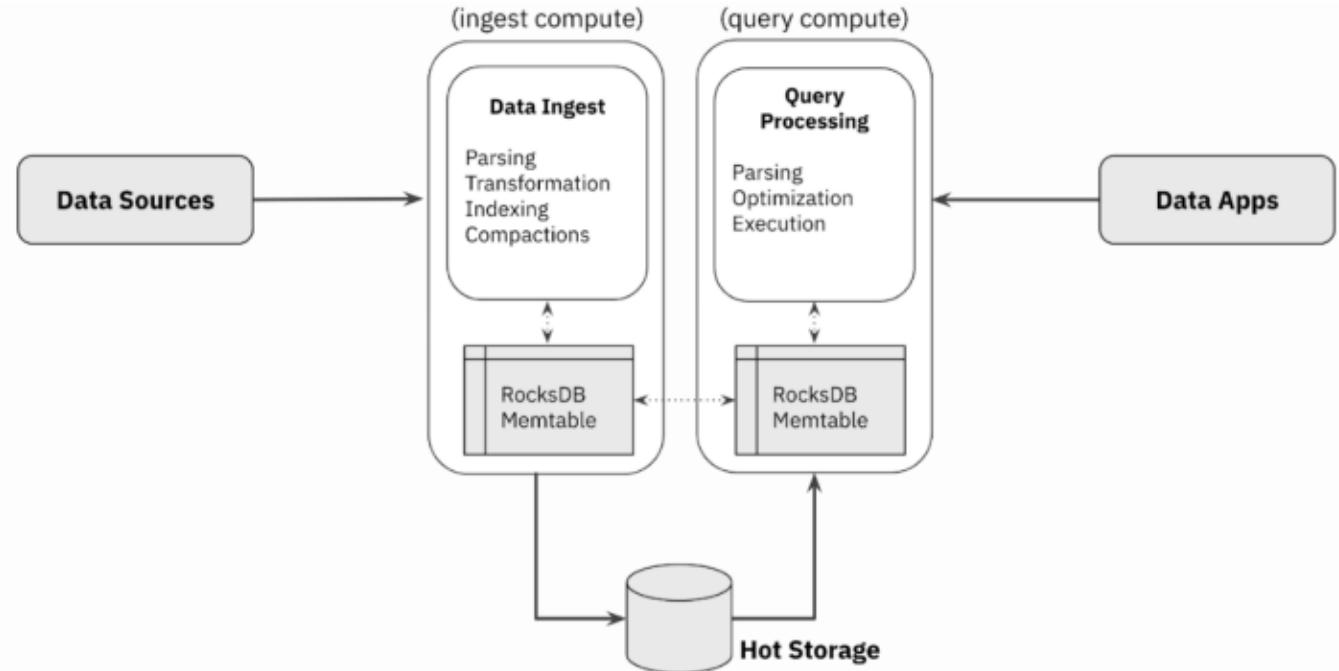


Search Optimized

- index all data types
- flexible schema
- vector search
- tokenization, BM25 search, exact word match, phrase match
- combining scores from vector search & BM25 search using reciprocal rank fusion

Realtime

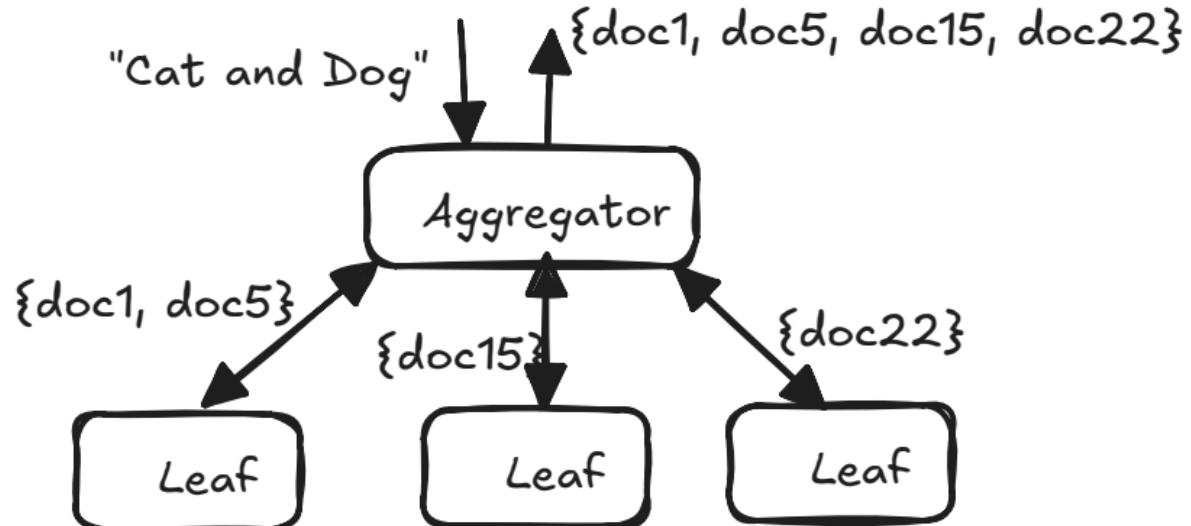
- Not just separation of compute and storage
- But also separation of ingest compute from query compute



[ROCKSET]

Low Latency Queries

- Search architecture : document sharded
- Scatter gather: every query goes to every shard



The Aggregator Leaf Tailer Architecture (ALT)

Retrieval Databases: 2025 and beyond

- AI is moving to Agentic Tool Invocations
- An Agent needs access to your data all the time
- An Agent needs long lasting memory
- Retrieval Database will be the Memory of AI

Distillation

- Transfer knowledge from a larger, more complex model to a smaller, simpler model
- Finetune with custom, enterprise data
- Thereby reducing cost of inference