Bridging the Operational and Analytical Worlds with Lakebase

Matei Zaharia

databricks NEON UC Berkeley Sky Computing Lab

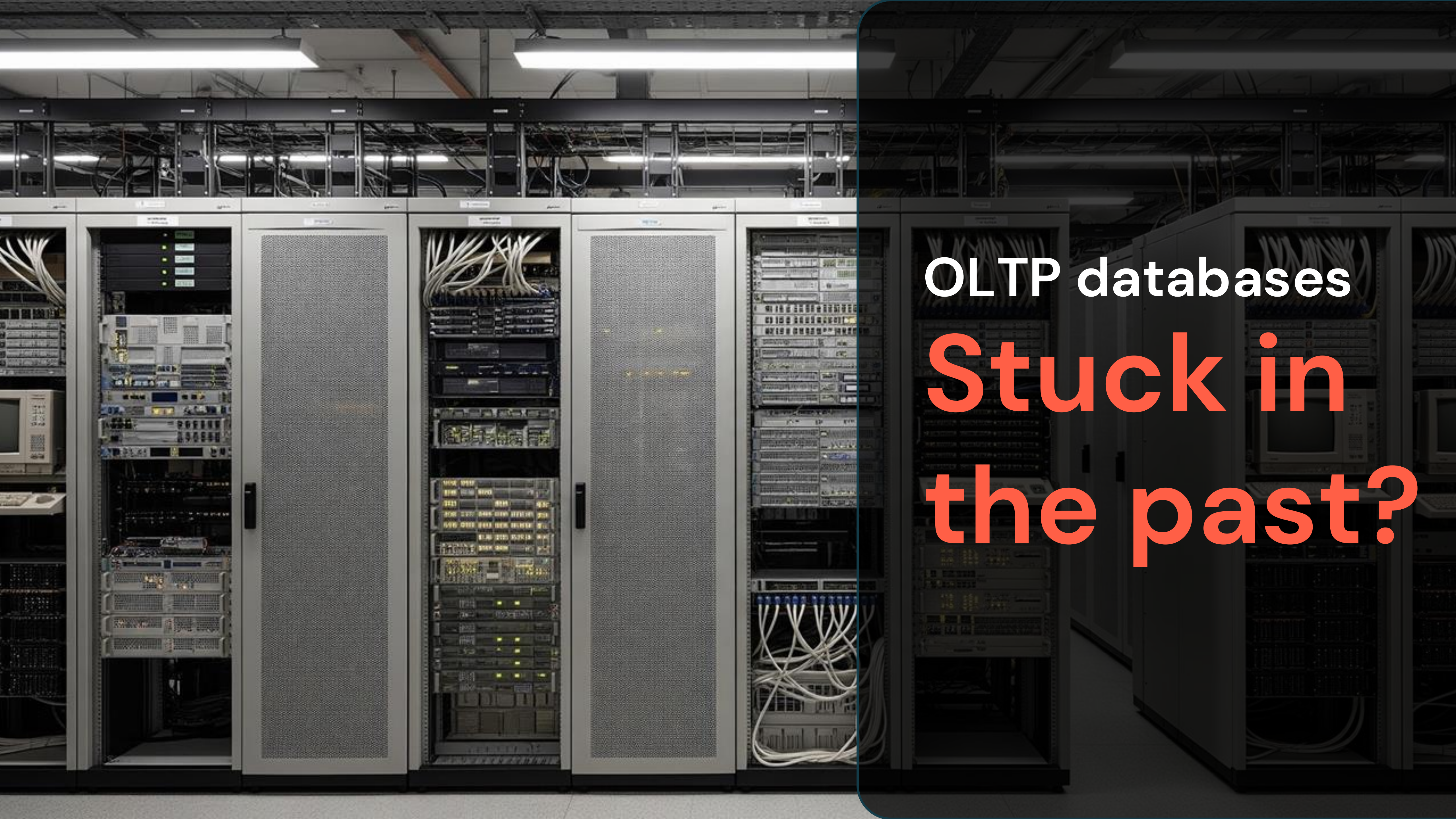
Analytics systems have changed dramatically since the 1990s

Columnar Storage
1995

Vectorized Processing
~2000

Streaming
~2010

Lakehouse Open Formats
~2020


OLTP databases
Stuck in the past?

I Hope to Convince You That…

We're at an inflection point for OLTP systems due to changes in environment and workloads

There's potential for a new, open and scalable OLTP architecture, the Lakebase, extending the Lakehouse

+ Lots of interesting open research questions

Three Motivating Trends

The cloud environment

New demands on analytics platforms

AI’s impact on database systems

Three Motivating Trends

The cloud environment

New demands on analytics platforms

AI's impact on database systems

The Cloud: A Radically New Environment

Big, multitenant datacenters with everyone's apps & data in one place

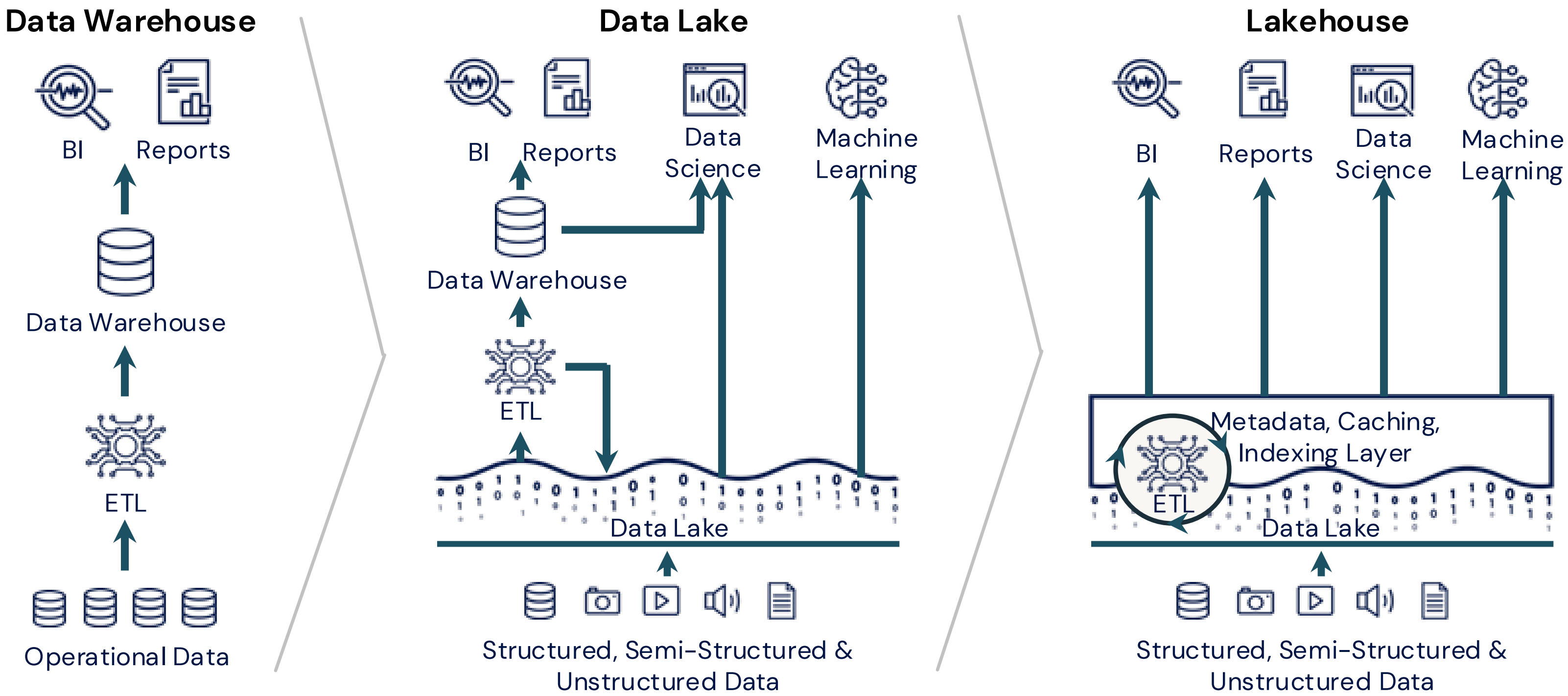
Very fast intra-datacenter networks

Rapid elastic scaling (and user expectations thereof)

The Cloud Transformed Analytical DBs

Separate compute & storage to scale them elastically

Use open formats so any engine can access the storage (Lakehouse!)

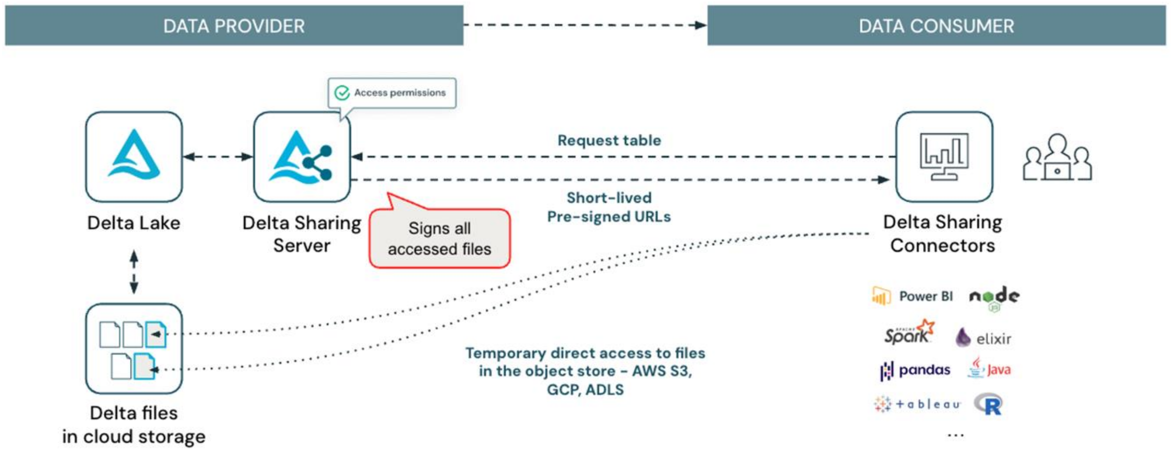


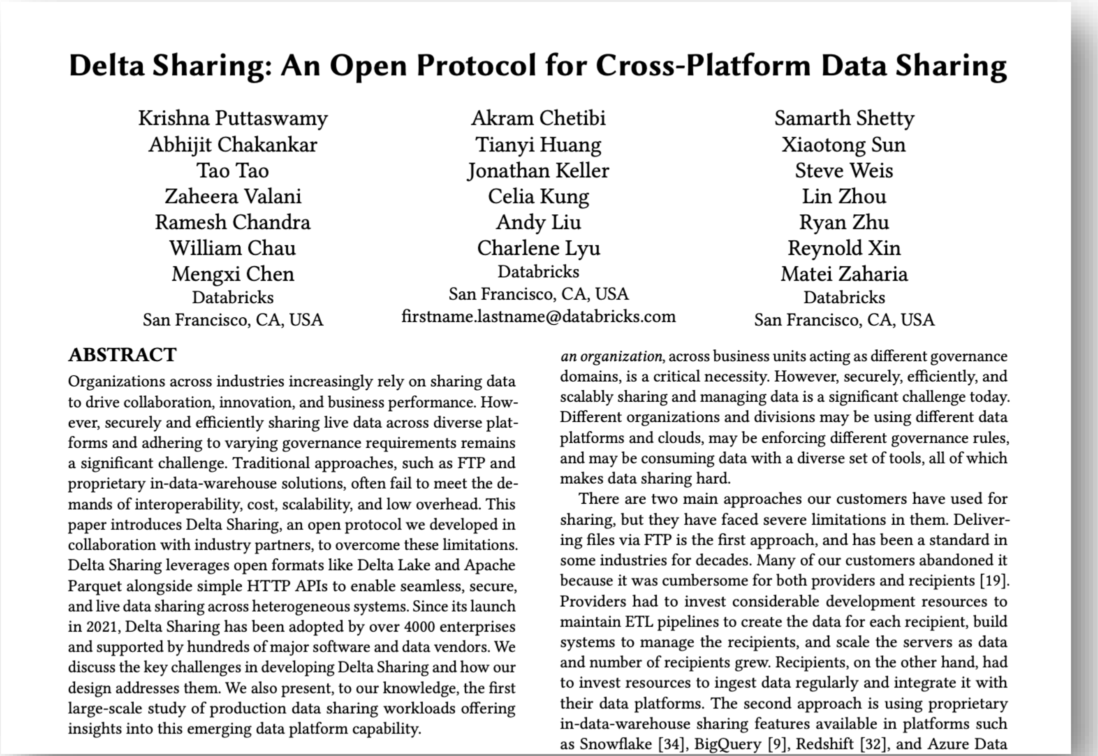
The Cloud Transformed Analytical DBs

Separate compute & storage to scale them elastically

Use open formats so any engine can access the storage (Lakehouse!)

New: open cross-org sharing API (Delta Sharing at this VLDB!)





Delta Sharing: An Open Protocol for Cross-Platform Data Sharing

Krishna Puttaswamy, Abhijit Chakankar, Tao Tao, Zaheera Valani, Ramesh Chandra, William Chau, Mengxi Chen
Databricks
San Francisco, CA, USA

Akram Chetibi, Tianyi Huang, Jonathan Keller, Celia Kung, Andy Liu, Charlene Lyu
Databricks
San Francisco, CA, USA
firstname.lastname@databricks.com

Samarth Shetty, Xiaotong Sun, Steve Weis, Lin Zhou, Ryan Zhu, Reynold Xin, Matei Zaharia
Databricks
San Francisco, CA, USA

ABSTRACT

Organizations across industries increasingly rely on sharing data to drive collaboration, innovation, and business performance. However, securely and efficiently sharing live data across diverse platforms and adhering to varying governance requirements remains a significant challenge. Traditional approaches, such as FTP and proprietary in-data-warehouse solutions, often fail to meet the demands of interoperability, cost, scalability, and low overhead. This paper introduces Delta Sharing, an open protocol we developed in collaboration with industry partners, to overcome these limitations. Delta Sharing leverages open formats like Delta Lake and Apache Parquet alongside simple HTTP APIs to enable seamless, secure, and live data sharing across heterogeneous systems. Since its launch in 2021, Delta Sharing has been adopted by over 4000 enterprises and supported by hundreds of major software and data vendors. We discuss the key challenges in developing Delta Sharing and how our design addresses them. We also present, to our knowledge, the first large-scale study of production data sharing workloads offering insights into this emerging data platform capability.

an organization, across business units acting as different governance domains, is a critical necessity. However, securely, efficiently, and scalably sharing and managing data is a significant challenge today. Different organizations and divisions may be using different data platforms and clouds, may be enforcing different governance rules, and may be consuming data with a diverse set of tools, all of which makes data sharing hard.

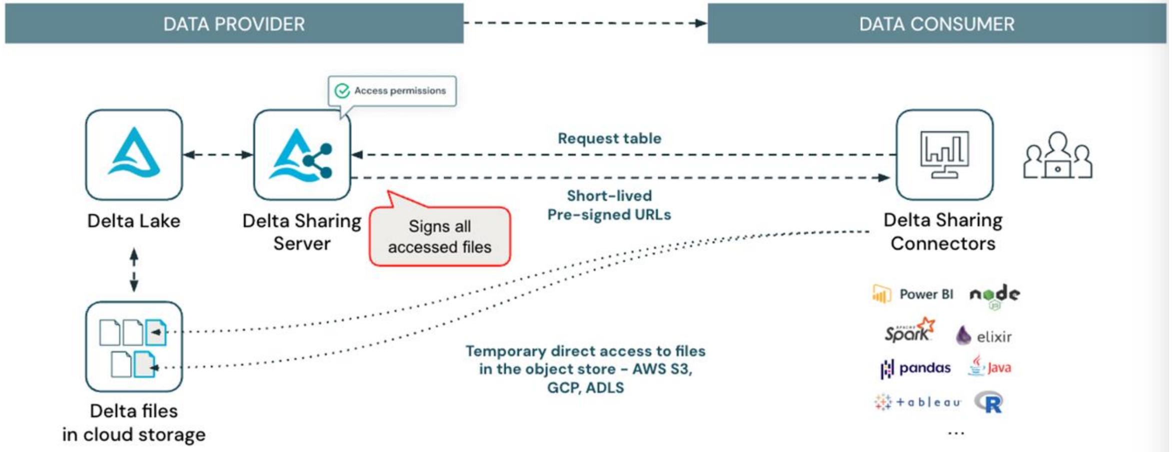
There are two main approaches our customers have used for sharing, but they have faced severe limitations in them. Delivering files via FTP is the first approach, and has been a standard in some industries for decades. Many of our customers abandoned it because it was cumbersome for both providers and recipients [19]. Providers had to invest considerable development resources to maintain ETL pipelines to create the data for each recipient, build systems to manage the recipients, and scale the servers as data and number of recipients grew. Recipients, on the other hand, had to invest resources to ingest data regularly and integrate it with their data platforms. The second approach is using proprietary in-data-warehouse sharing features available in platforms such as Snowflake [34], BigQuery [9], Redshift [32], and Azure Data

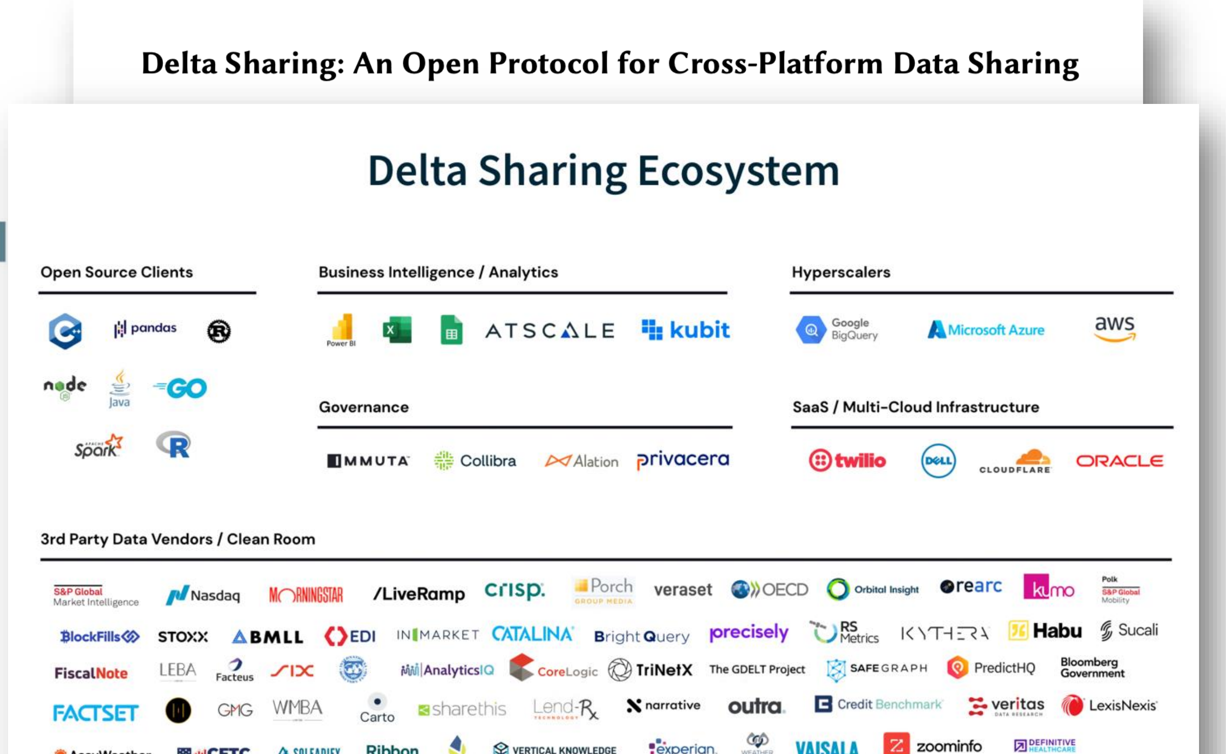
The Cloud Transformed Analytical DBs

Separate compute & storage to scale them elastically

Use open formats so any engine can access the storage (Lakehouse!)

New: open cross-org sharing API (Delta Sharing at this VLDB!)





What Does the Cloud Mean for OLTP?

Traditional OLTP DBs have coupled compute & storage, limited elasticity, and limited support for sharing across engines

But if all the data is on the same disks in the cloud, can we rethink OLTP systems to get some of the same benefits we got in analytics?

NEON Database Properties

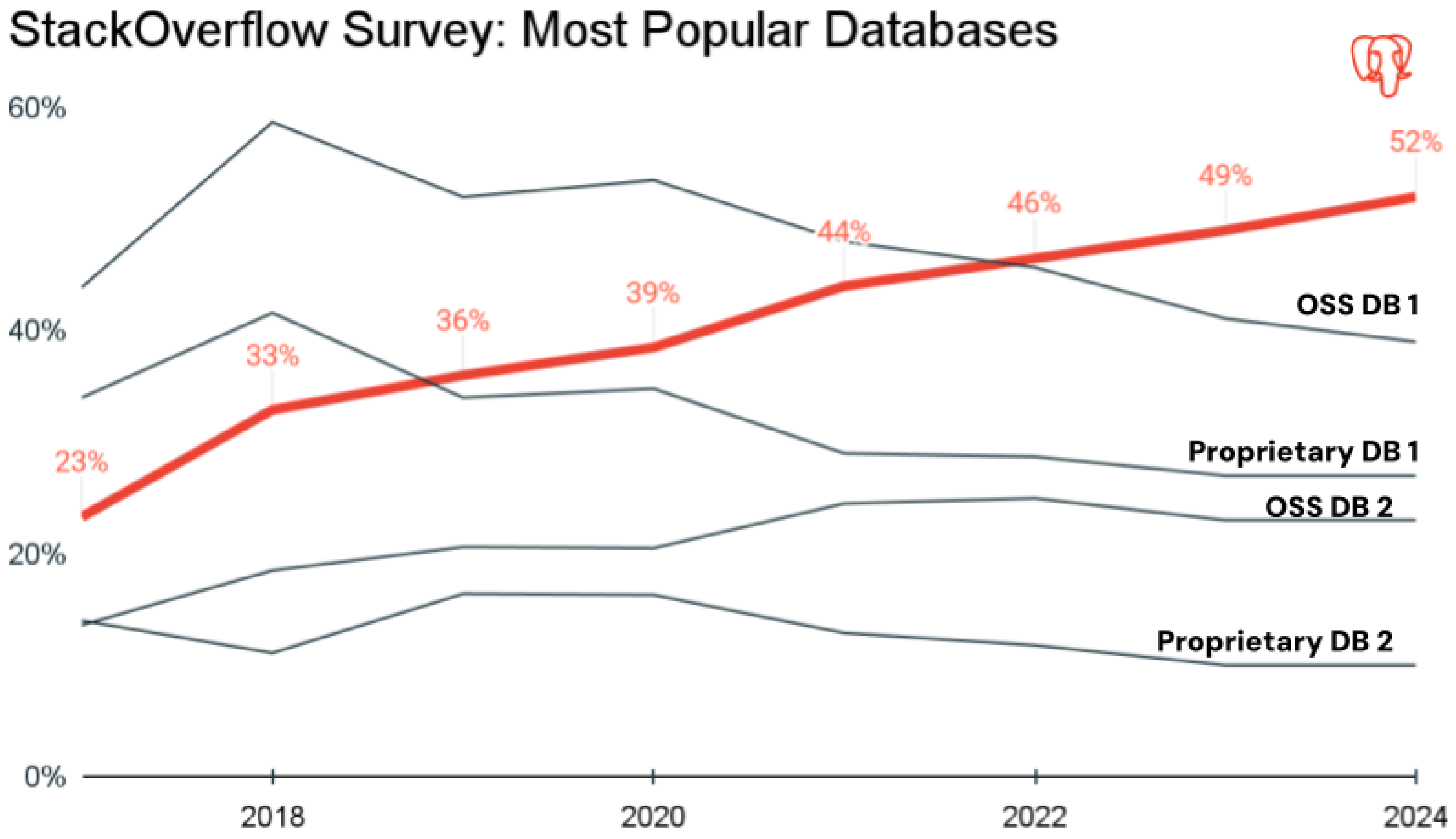
Open API: 100% Postgres

Decoupled compute & storage: cold data is on object stores

Rapid elasticity: scale up in seconds, or down to 0

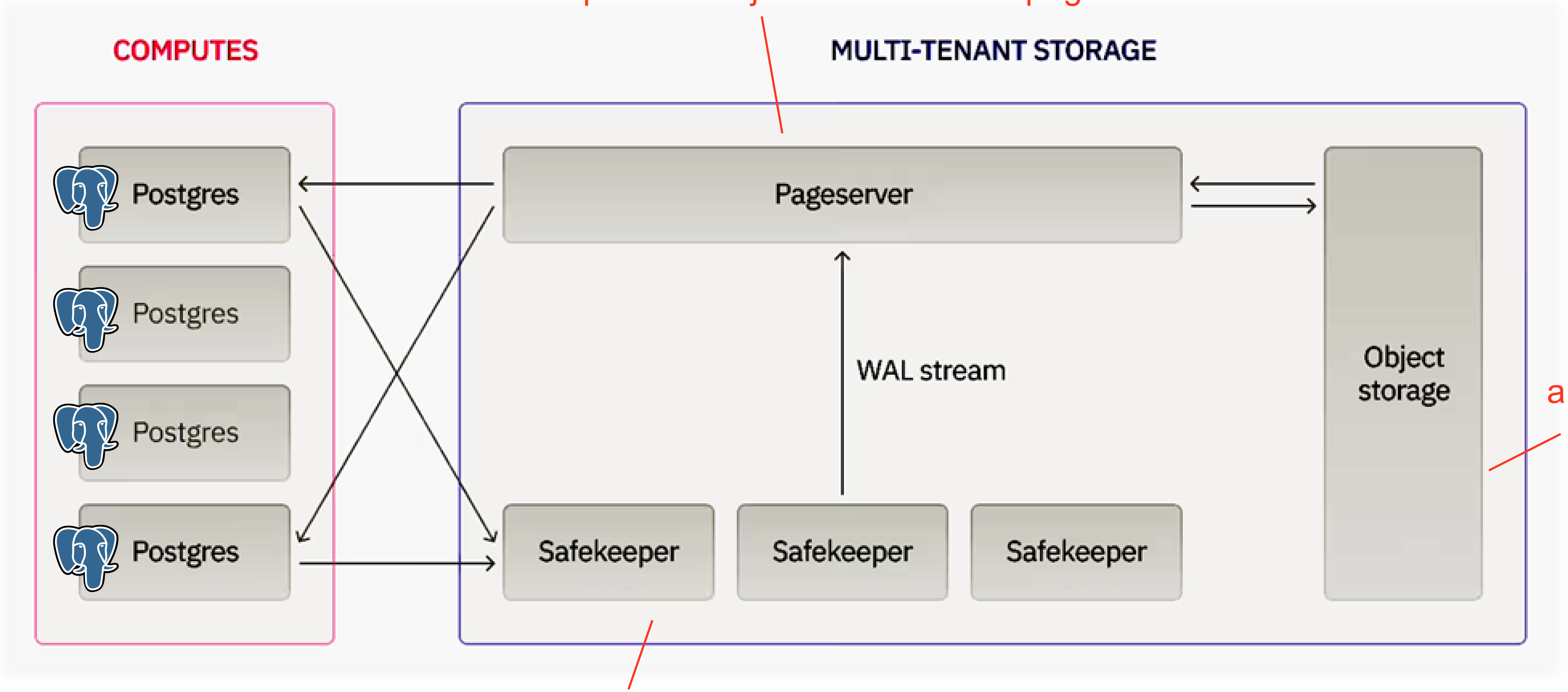
Fast sharing: branch a DB in milliseconds

Why Postgres?



NEON Architecture

Pageservers process updates,
upload to object store & serve pages



Storage format
allows time travel,
is tuned for S3

Safekeepers manage WAL via Paxos

NEON Results

Creating a database is fast! Try it by going to [pg.new](pg.new)

Cost is low, with performance competitive: for example, about 4x cheaper storage than Aurora Serverless v2, and 4x faster scaling

Reads scale out well to 10,000s of connections by default in Neon

Branching is virtually free, changing developer workflows

Postgres extensions just work

30-year-old Postgres behaves cloud-native!

Three Motivating Trends

The cloud environment

New demands on analytics platforms

AI's impact on database systems

A Story from Mike Franklin

Today: Growing Need for Operational Apps on Analytics Data

As more of the world is digitized, more data is available in real-time and more actions can be taken in real-time

As algorithms improve, automated actions can be smarter

New Demands on Analytics Platforms

Ingest and transform data faster

Serve transformed data out to analytical apps

Examples from Databricks

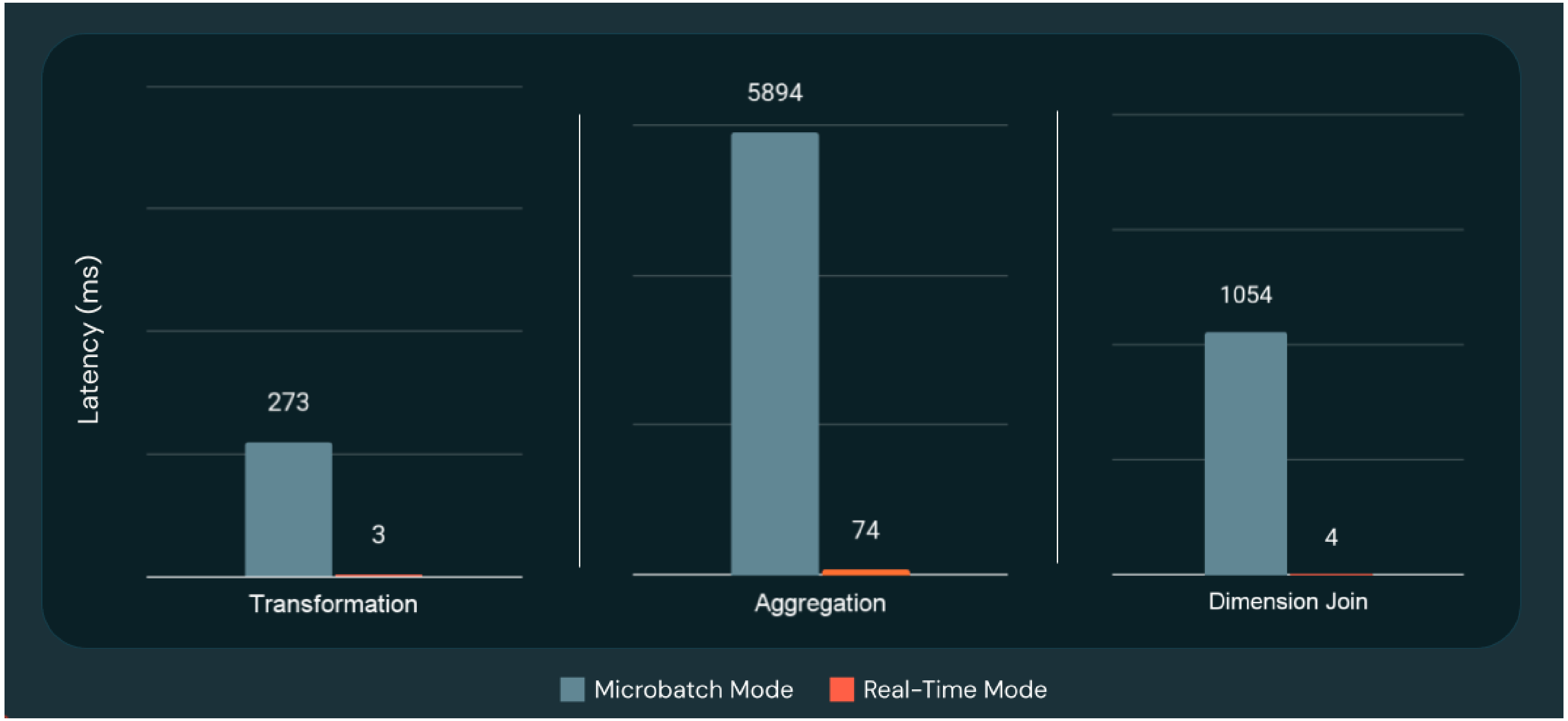
Zerobus: Ingest rows rapidly into any existing Delta/Iceberg table

```
curl endpoint { json }
```

On by default for every table with no setup, no tuning, and fast scaling, supporting petabytes per day at <100ms latency

Examples from Databricks

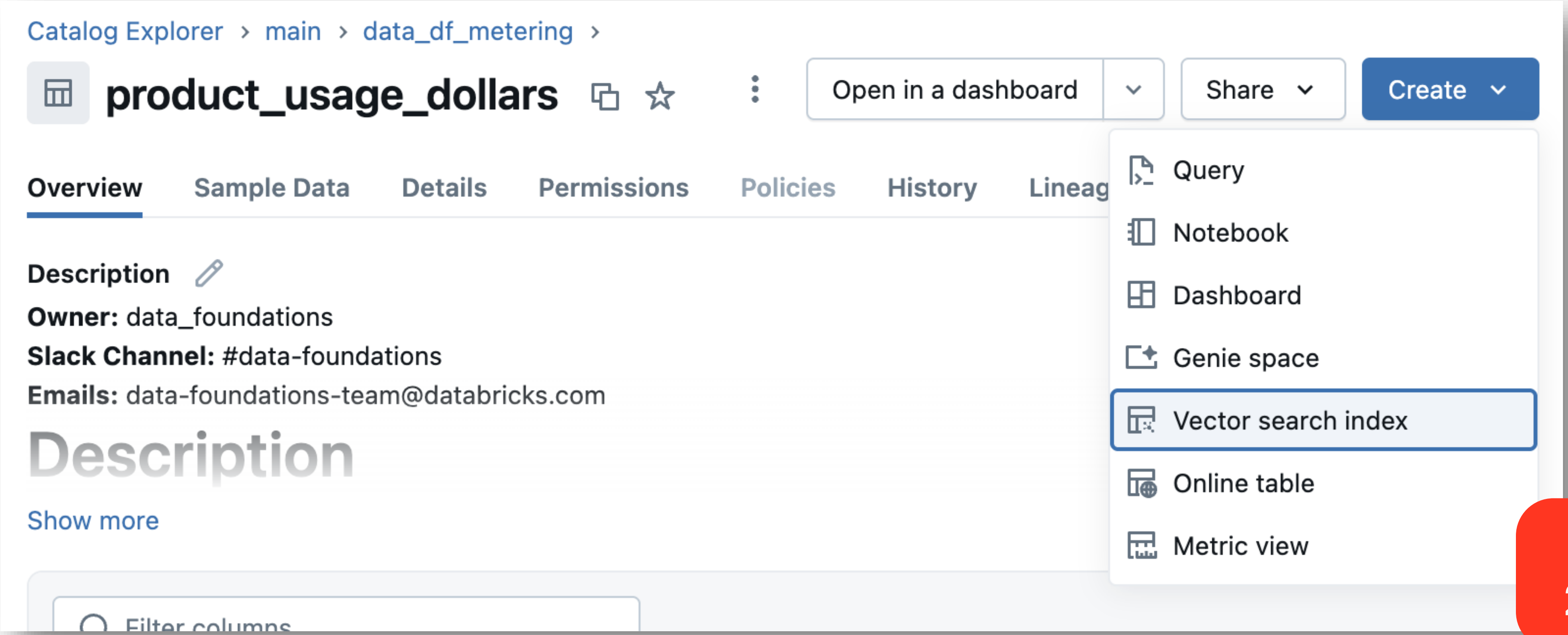
Apache Spark Real-Time Mode: low latency continuous version of all streaming operators, with no changes to job logic



tinyurl.com/spark-rtm

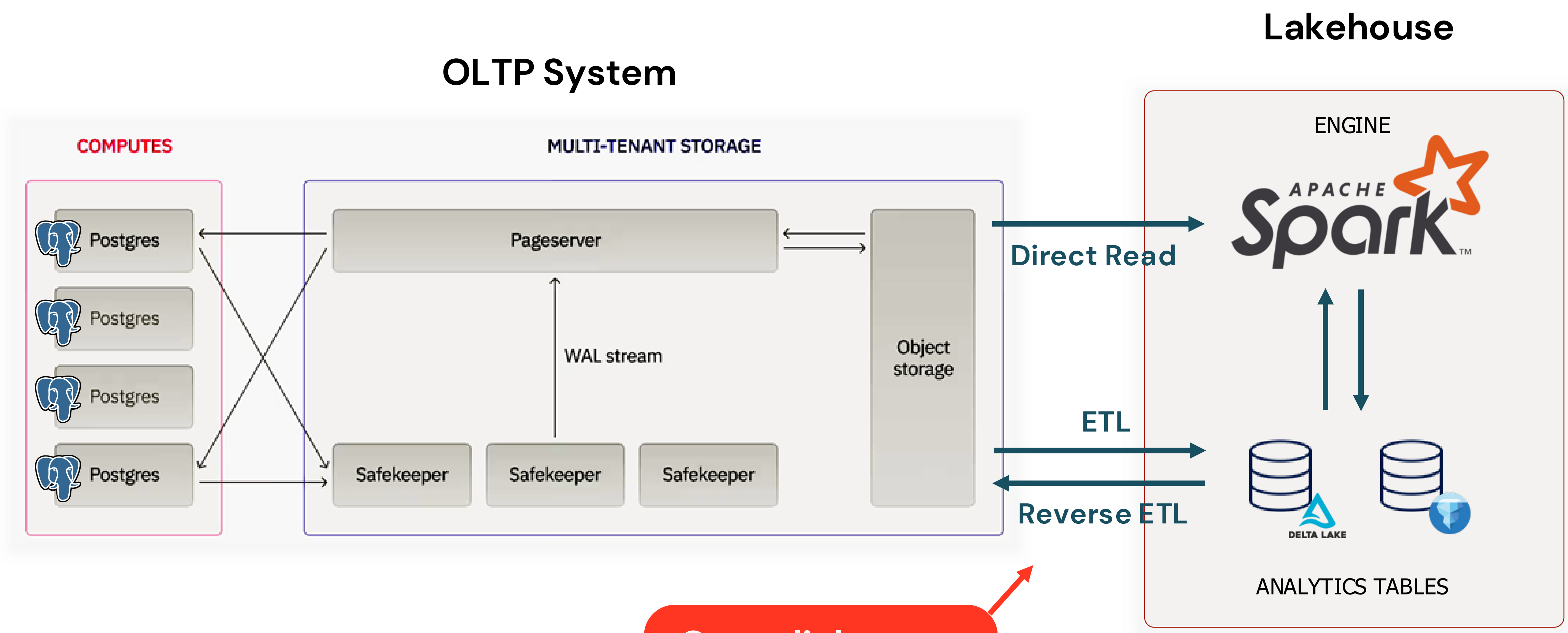
Examples from Databricks

Vector Search: Serve any Delta/Iceberg table for low latency vector or keyword search, with automatic sync on changes to the table



Growing 2.4x YoY!

OLTP + Lakehouse Fits Hand-in-Glove!



One-click setup on Databricks

This is Lakebase

An OLTP architecture built on cloud data lakes, characterized by:

- Separated compute and storage
- Open APIs
- Serverless elastic scaling, sharing and branching
- Easy integration with analytical lakehouse systems & data

How Databricks Customers use Lakebase

Serving analytical data: some customers had their own term, “lakeshore”

Real time features/metrics

Stateful data apps & agents

Adobe AstraZeneca Bridgestone Chevron Phillips Chemical Company LP Dream11
easyJet Ensemble Health Partners Epsilon Erste Group Ford | FordDirect
gainwell Gjensidige Hertz hipages
ibotta ifood Insight Insulet KRAFTON
Milliman novo nordisk Opendoor Quartile RED | VENTURES Rolls-Royce
RWE sse Energy Solutions SUPERCELL Takeda T Mobile TheStandard
WGU A NEW KIND OF U. THE WORLD BANK yipit DATA

But Wait, is this HTAP?

Musings

HTAP is Dead

Zhou Sun · May 4, 2025 · 5 min read

HTAP: Still the Dream, a Decade Later

Dani Palma · Follow · 10 min read · Jun 18, 2025

73 · 2

But Wait, is this HTAP?

Google

is htap dead

AI Overview

+5

No, HTAP isn't strictly dead, but the original vision of a single, consolidated database for hybrid transactional and analytical processing is being replaced by a more complex, composed approach. Recent discussions suggest that while integrated HTAP databases haven't succeeded as envisioned, the core goal of combining real-time transactional and analytical workloads remains necessary, now achieved through streaming architectures, ... query engines.

Why HTAP as a Single Database Failed

Complexity and Practical Limitations:

Nikita Shamgunov

It is certainly HTAP.

Google

is htap alive

AI Mode | All | Images | Videos | News | Short videos | Shopping | More | Tools

AI Overview

Yes, HTAP (Hybrid Transactional/Analytical Processing) is very much alive and thriving in the database world, with major advancements being made by companies like Google and SingleStore. While there's ongoing debate about the ideal implementation, the fundamental concept of running both operational and analytical workloads on the same system is becoming a necessity for businesses seeking real-time insights, driving innovation in traditional database systems and paving the way for a more unified data architecture.

But Wait, is this HTAP?

However you define HTAP, it’s usually useful to architect your operational and analytical tables differently, and if you’re going to do that, then Lakebase can be a great solution on top of open lake storage!

The latencies and interop across engines will only get better

> **Wrapping It Up: HTAP Isn’t Dead, It’s Just Boring Now (That’s a Compliment)**
>
> We’ve spent over a decade chasing the HTAP dream, and we still haven’t built The One Engine. But maybe that’s okay.

Three Motivating Trends

The cloud environment

New demands on analytics platforms

AI's impact on database systems

AI’s Impact on Database Workloads

OLTP: agentic coding creates even more need for fast branching & scaling

OLAP: very different, non-BI access pattern

Both: vector search and semantic operators arxiv.org/pdf/2509.00997v1

Supporting Our AI Overlords: Redesigning Data Systems to be Agent-First

Shu Liu, Soujanya Ponnapalli, Shreya Shankar, Sepanta Zeighami, Alan Zhu
Shubham Agarwal, Ruiqi Chen, Samion Suwito, Shuo Yuan, Ion Stoica, Matei Zaharia
Alvin Cheung, Natacha Crooks, Joseph E. Gonzalez, Aditya G. Parameswaran
UC Berkeley

Abstract

Large Language Model (LLM) agents, acting on their users' behalf to manipulate and analyze data, are likely to become the dominant workload for data systems in the future. When working with data, agents employ a high-throughput process of exploration and solution formulation for the given task, one we call *agentic speculation*. The sheer volume and inefficiencies of agentic speculation can pose challenges for present-day data systems. We argue that data systems need to adapt to more natively support agentic workloads. We take advantage of the characteristics of agentic speculation that we identify, i.e., scale, heterogeneity, redundancy, and steerability—to outline a number of new research opportunities for a new agent-first data systems architecture, ranging from new query interfaces, to new query processing techniques, to new agentic memory stores.

1 Introduction

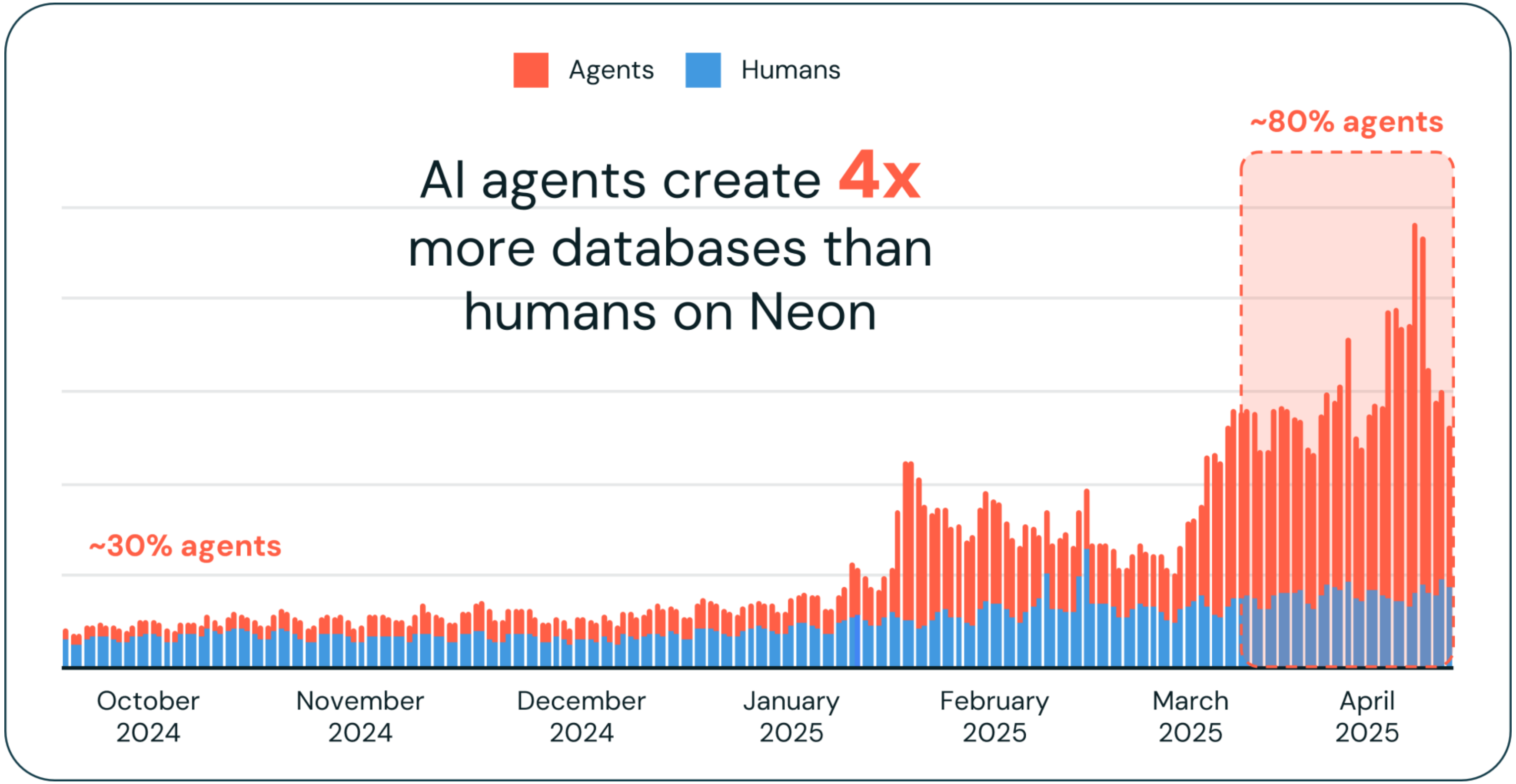
Powered by Large Language Models (LLMs) that can reason, invoke tools, author code, and communicate with each other, we are on the precipice of a new agentic revolution that will transform how data

of LLM agents tasked with finding reasons for why profits in coffee bean sales in Berkeley was low this year relative to last. Since they are not limited by human cognitive bandwidth and response times, an army of agents could employ an enormous volume of queries to data systems, far more than any human could—all for a single task. Many of these queries are likely wasteful, and are simply providing the agents grounding. As another example, if an LLM agent is tasked with identifying a new crew for a delayed flight, it would need to consider various hypothetical transactions to surface to a human decision maker, each with dozens of updates to various databases.[2] For such tasks, agents may explore many alternatives in parallel by forking database state, running speculative updates, and rolling back branches. Overall, as agentic workloads become more and more prevalent, the sheer scale and inefficiencies of agentic speculation will become the bottleneck, and our data systems will need to evolve in response.

So we ask the question: *how can data systems evolve to better support agentic workloads?* In particular, can data systems natively—and efficiently—support agentic speculation, helping LLM agents determine the best course of action? This question—which, as we

AI Impact on OLTP: Want Faster Dev Loops

Agents code fast and try things in parallel, so cheap dev loop matters!



Ideally, you want:

- Cheap branching
- Very fast spin up
- Easy rollback
- Anonymized data in some branches

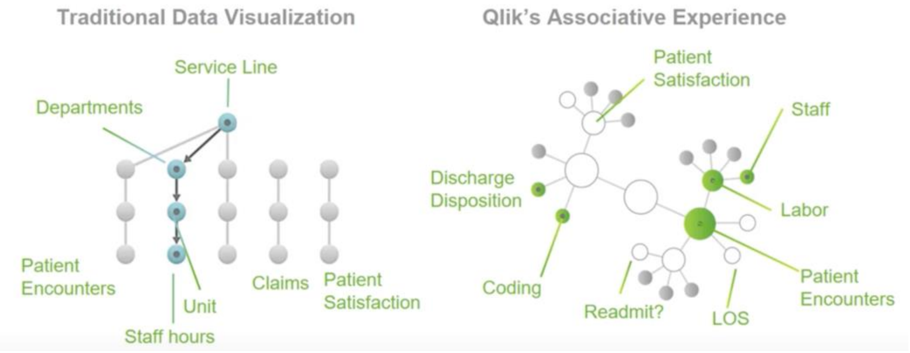
And now, even more people are building apps with AI tools!

replit Lovable Superblocks

AI Impact on OLAP: New Access Pattern

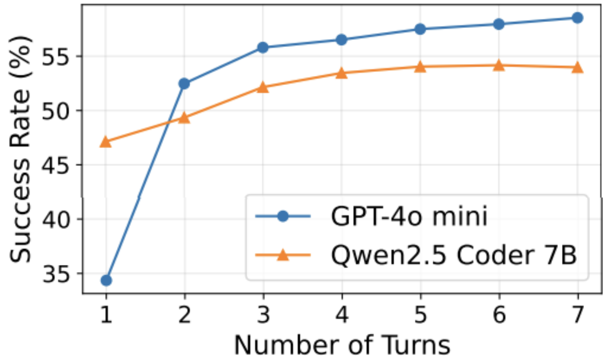
AI agents explore data speculatively to produce rich reports

BI tools typically emit a giant SQL query with many joins, based on associative data model



AI agents tend to *speculatively explore* to get context on the data and answer rich questions

Moreover, agents can explore many similar queries in parallel to increase quality



AI Across OLTP and OLAP: Unstructured Data is Now Queryable

LLMs can now answer bulk queries about unstructured data

- See Liana Patel's paper on LOTUS & Semantic Operators!

Vector and classic search needs to power these new applications

- Huge use of pg_vector, Vector Search
- Could get much cheaper and smarter, e.g. see our LEANN lightweight index

LOTUS

```
papers_df.sem_filter("the {research_paper} has an open source repo")
    .sem_topk("the {research_paper} has the most ground-breaking ideas", K=20)
```

LEANN

Python 3.9 | 3.10 | 3.11 | 3.12 | 3.13 CI passing Platform Ubuntu & Arch & WSL | macOS (ARM64/Intel) License MIT
MCP Native Integration Slack Join WeChat Join

The smallest vector index in the world. RAG Everything with LEANN!

LEANN is an innovative vector database that democratizes personal AI. Transform your laptop into a powerful RAG system that can index and search through millions of documents while using **97% less storage** than traditional solutions **without accuracy loss**.

lotus-data.github.io, github.com/yichuan-w/LEANN

The Future of Lakebase

Tight integration across transactional updates, SQL analytics and AI analytics on the same data

Ability to call all those powerful AI agents and functions in online apps

Agent- and human-friendly development loop

What Does All This Mean for Research?

It’s Very Early Days!

Many open problems around Lakebase and cloud-native OLTP

Very scalable writes (without losing compatibility!)

Cross-org sharing

Integrating streaming with transactional data

Querying across storage types

Beyond AI as a database user: can agents manage your database?

Conclusion

This is a very exciting time to rethink OLTP and operational data broadly

Beyond the shift to cloud, we have a whole new class of database users (agents + new people coding with them) and AI methods

Lakebase is one direction I think systems will go based on the cloud's affinity for open APIs and elasticity, but there's a lot more to explore!