Stanford @ NorCalDB Day Outline

- Accelerating New DB Workloads
- Applications of Foundation Models
- Applications of Transactional DBs
- Graph Neural Networks
Stanford @ NorCalDB Day Outline

- Accelerating New DB Workloads
- Applications of Foundation Models
- Applications of Transactional DBs
- Graph Neural Networks
General-Purpose Video Analytics

VIVA: An End-to-End System for Interactive Video Analytics

**What is it?** Spark-based system that optimizes video queries using declarative hints about models relationships

**Results:** 16.6x speedup without sacrificing accuracy

ZELDA: Video Analytics using Vision-Language Models (VLMs)

**What is it?** Top-K natural language video analytics system using vision-language models

**Results:** Relevant and diverse results with interactive latency

[Links to VIVA and ZELDA papers]

---

Francisco Romero, Johann Hauswald, Caleb Winston

---

<table>
<thead>
<tr>
<th>Current Systems</th>
<th>ZELDA</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Query</code></td>
<td><code>VLM = &quot;cars during daytime at traffic intersections&quot;</code></td>
</tr>
<tr>
<td><code>Modell = &quot;car&quot; AND Model2 = &quot;daytime&quot; AND camID LIKE &quot;intersection%&quot;</code></td>
<td></td>
</tr>
</tbody>
</table>
Quokka  fault-tolerant SQL query engine

How it started: SparkSQL’s stage-wise execution affords it robust fault tolerance. Can we have similar fault tolerant execution in a pipelined engine?

What is it now: fault-tolerant distributed pipelined query engine with a dataframe interface.

Where it’s going: TB-scale time series analytics, such as asof joins, range joins and match recognize on TAQ and observability data.

https://github.com/marsupialtail/quokka
Hybrid Search in Vector Databases

Key Ideas:
Prior ANN engines provide limited support for hybrid data models, which include vector embeddings from unstructured data and structured data.

We built a custom index to support i) hybrid data and ii) ANN search queries with metadata filtering.

Results: 10x faster hybrid search queries

More to come!
Stanford @ NorCalDB Day Outline

- Accelerating New DB Workloads
- Applications of Foundation Models
- Applications of Transactional DBs
- Graph Neural Networks
Meerkat for Model Analysis and Dataset Exploration

Key idea: An open-source Python library that teams can use to interactively and reliably wrangle their unstructured data with FMs.

Results: We’re excited to see Meerkat enriching and speeding up the data analysis workflows of a large community of cross-disciplinary users. Join our Discord!

https://github.com/HazyResearch/meerkat
Evaporate for Generating Structured Views of Heterogeneous Data Stores

**Key idea:** In extracting structured fields from unstructured documents, we change the task’s logical plan to avoid processing every document with costly FMs.

**Results:** With no human-effort, evaporate achieves SoTA quality with merely a sublinear pass over the data store (110x reduction in inference cost at 10k documents per data store)!

https://github.com/HazyResearch/evaporate

Simran Arora, Brandon Yang
Stanford @ NorCalDB Day Outline

- Accelerating New DB Workloads
- Applications of Foundation Models
- Applications of Transactional DBs
- Graph Neural Networks
Apiary: A DBMS-Integrated Transactional Function-as-a-Service Framework

**Challenge:** Existing FaaS separates app logic and data, making it hard to build efficient and reliable data-centric apps

**Insight:** Leverage distributed database engine as a unified runtime, co-locating compute and data

**Apiary:** A DBMS-integrated transactional FaaS framework
- Functions run transactionally inside the database
- Fault-tolerant workflows
- Trace application-DB interactions for advanced observability
- Achieve $2 - 68x$ higher performance by co-location

Qian Li, Peter Kraft

https://github.com/DBOS-project/apiary
**Challenge**: Distributed apps are hard to debug because they share data across many concurrent requests

**Insight**: Transactional data access pattern makes "time travel" uniquely practical

**R³**: A practical time-travel debugging tool for DB-backed apps

- Always-on low-overhead recording at transaction level granularity
- Faithfully replay past requests
- Retroactively execute modified code on past events
- Require at least snapshot-isolation, supported by most production DBs
**Epoxy: ACID Transactions Across Diverse Data Stores**

**Challenge:** Hard to provide transactions across multiple data stores without modifying their internals.

**Insight:** Provide transactions through MVCC in a bolt-on layer, storing transaction information in record metadata.

---

**Diagram Description:**

- **Clients** receive queries and updates from applications.
- **Epoxy Shim** with secondary stores like MongoDB and CouchDB is used to interface with the primary database.
- **Coordinator** handlesBegin/Commit transaction requests and queries/updates.
- **Epoxy Shim** processes transactions on the secondary stores, including Solr and elasticsearch.

---

**Diagram Flow:**

1. **Client** begins transaction
2. **Coordinator** receives Begin/Commit transaction request
3. **Coordinator** sends to secondary store
4. **Secondary Store** (e.g., Solr, elasticsearch) processes transaction
5. **Coordinator** validates transaction
6. **Coordinator** makes Commit/Abort decision
7. **Coordinator** sends back results
8. **Client** receives results
Stanford @ NorCalDB Day Outline

- Accelerating New DB Workloads
- Applications of Foundation Models
- Applications of Transactional DBs
- Graph Neural Networks
Complex Reasoning Over Knowledge Graphs with Neural Graph Reasoning

**Problem:**

\[ C_7, \exists P : \text{ASSOC}(d_1, P) \land \text{ASSOC}(d_2, P) \land \text{TARGET}(P, C_7) \]

“Predict drugs \( C_7 \) that might target proteins that are associated with the given disease nodes \( d_1 \) and \( d_2 \).”

**Solution:** Reason in the embedding space

---

![Diagram of knowledge graph and reasoning process](image-url)
PyG allows for ML over relational data with no feature engineering! Lots of cool research.

- **Fully open-sourced**
- ~2.3k research papers
- ~350 contributors
- ~17k GitHub stars
- ~100k monthly downloads

**Hot research area:**

ICLR 2022 Submission Top 50 Keywords

```
conda install pyg -c pyg
```
Talk to us!