**Tiresias**: Enabling Predictive Autonomous Storage and Indexing

**Michael Abebe**, Horatiu Lazu, Khuzaima Daudjee

- **DBMS storage** and **indexing choices** have trade-offs based on the workload. **Tiresias** enables adapting these choices automatically based on the workload.

**Objective**

**OLAP Query Lat.** versus **OLTP Throughput**

Neither indexing choice optimal for both OLTP & OLAP

**Architecture**

- **Predict** upcoming accesses and latency under different storage and indexing choices
- **Tiresias**
  - **Model** workload access history
  - **DBMS**
    - Store and index data
    - **Change** data storage and indexing
- Report query latency and access history

**Tiresias**'s use cases:

- **PostgreSQL**: automatically add/remove secondary indexes
- **OLAP DBMS**: predictively cracking (sorting) of data
- **Proteus**: Adaptation of storage formats for HTAP

**Predicting Transaction Latency**

Transactions composed of **physical operators**.

- **Tiresias** learns the **latency** of operators under storage/indexing choices **parameterized** by workload statistics.
- **Tiresias** combines predicted operators lat. to predict **txn. lat.**

**SELECT** book, SUM(qnt) GROUP BY book

<table>
<thead>
<tr>
<th>No Index</th>
<th>Row scan</th>
<th>Logical Plan</th>
<th>With Index</th>
</tr>
</thead>
</table>

Index changes query execution, captured in lat. prediction

**Making Change Decisions**

For Tiresias to make a storage or indexing change:

- **Expected Benefit** > **Upfront Costs**
  - **Predict** transaction latency under **current** and **proposed** choice
    - E.g. with index: predict OLTP txn. will execute 25 ms faster.
  - **Weighted** by predicted likelihood of transaction occurring in near future, to balance trade-offs.
  - Ensures system does not constantly undo/re-do changes

**Experimental Evaluation**

- **Tiresias** automatically adds and removes indexes in PostgreSQL in a workload with a shifting mix
- **Tiresias** completes workload faster than no indexes (best for OLTP) & with indexes (best for OLAP)

**Tiresias** enables predictive cracking; reduces query latency by 40x versus default cracking in an **OLAP DBMS**.

**Tiresias** allows predictive storage format adaptation in Proteus

Completes mixed workload (CH) faster than static layouts: row (best for OLTP) & columnar (best for OLAP)

- **Tiresias** automatically adds and removes indexes in PostgreSQL in a workload with a shifting mix
- **Tiresias** completes workload faster than no indexes (best for OLTP) & with indexes (best for OLAP)

**Tiresias** enables predictive cracking; reduces query latency by 40x versus default cracking in an **OLAP DBMS**.

**Tiresias** allows predictive storage format adaptation in Proteus

Completes mixed workload (CH) faster than static layouts: row (best for OLTP) & columnar (best for OLAP)