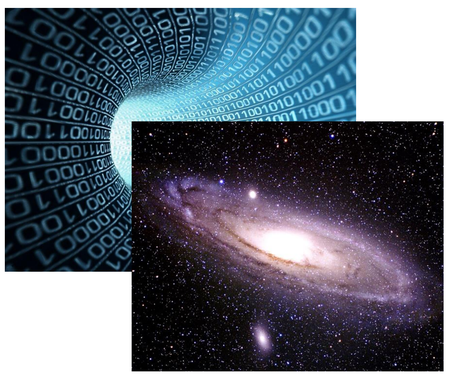
NoDB in Action: Adaptive Query Processing on Raw Data

Ioannis Alagiannis, Renata Borovica, Miguel Branco, Stratos Idreos, Anastasia Ailamaki

In the Era of Data Deluge



Data collections become larger and larger

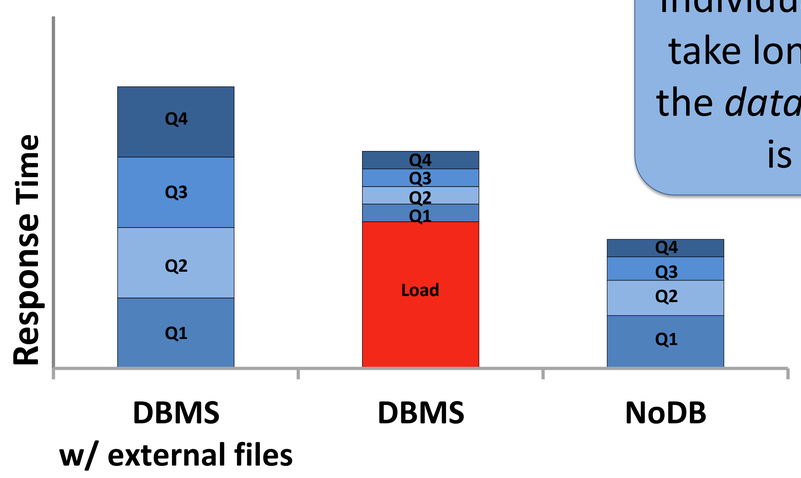
Many applications avoid using DBMS (e.g. social networks, scientific data analysis)

Problem: DBMS startup cost

- **Why wait for loading and tuning?**
- **Why load the entire dataset?**
- **Column vs. row-store?**
- **Hire DB expert?**

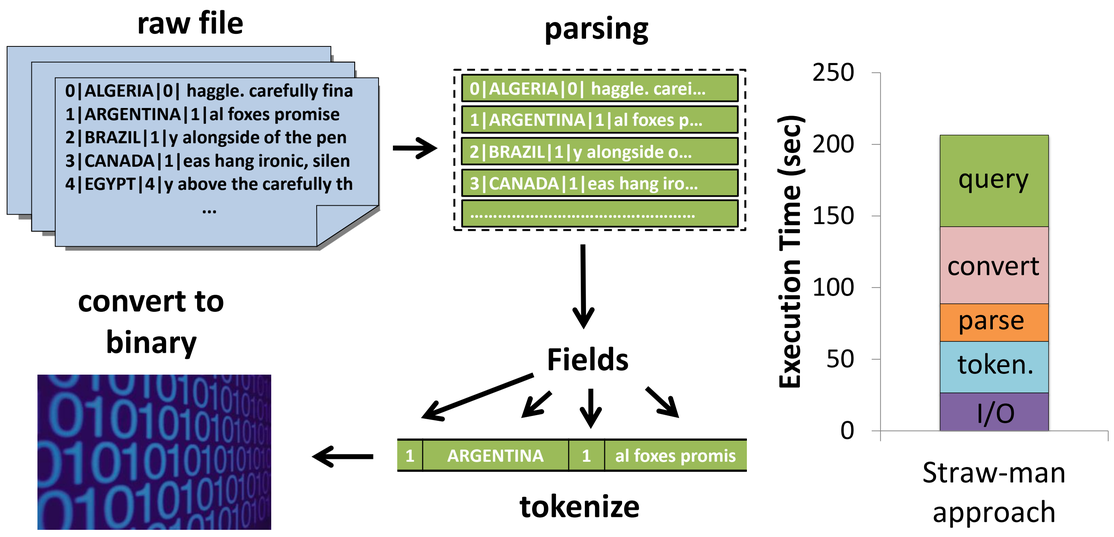
The NoDB "Philosophy"

- **No loading of entire dataset and values**
- **Query processing *in situ***
- **Minimize data-to-query time**
- **Raw data files as a first-class citizen**



Individual queries may take longer; however, the *data-to-query* time is reduced

Querying Raw Data Files

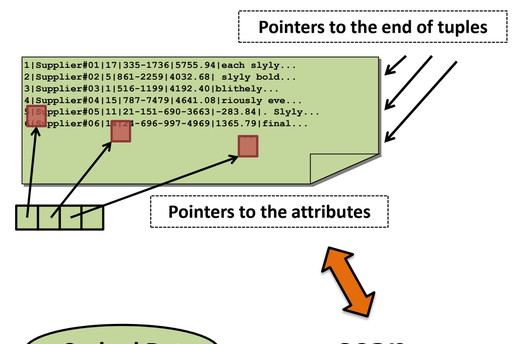


Accessing raw data files is expensive

- **Parsing & tokenizing**
- **Data type conversion**

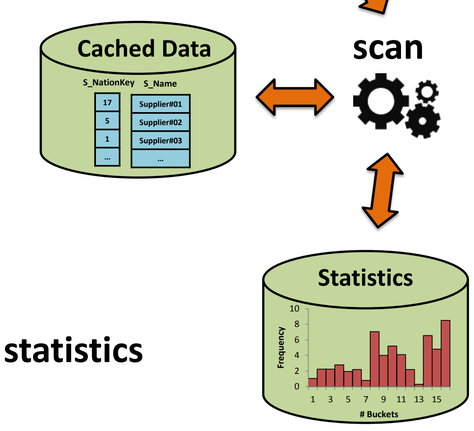
Positional Maps

- **Reduce parsing and tokenizing**
- **Maintain low level metadata**
- **Created on-the-fly**
- **Adapt to queries**



Caching

- **Complementary to positional map**
- **Avoid raw file accesses**
- **Populated during query execution**



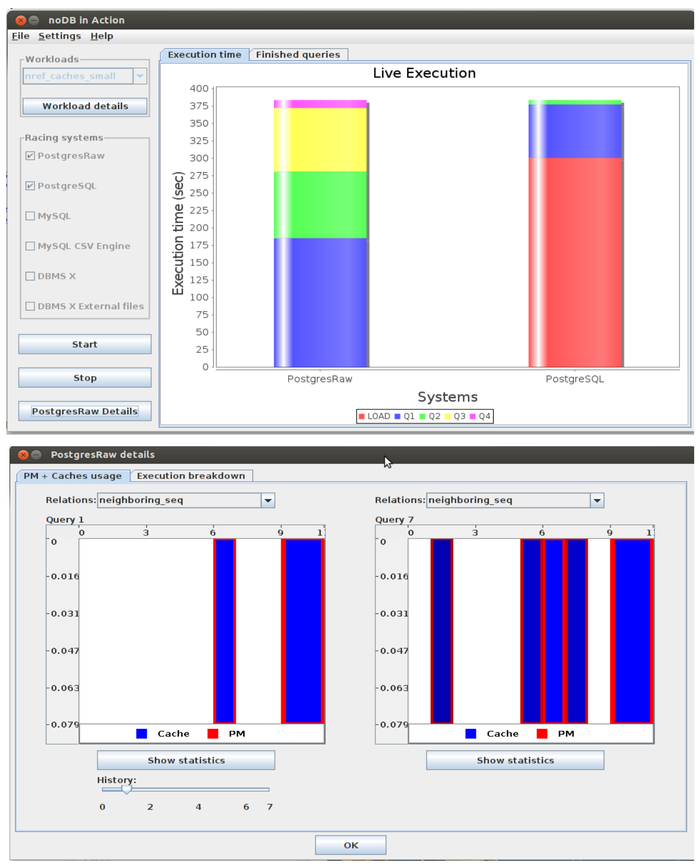
Statistics

- **Extend the scan operator to create statistics**
- **Generate in an adaptive way**

PostgresRaw in Action

PostgreSQL + **NoDB "philosophy"** = **PostgresRaw**

1. **Setup:**
   - **6 Intel Xeon E5-2660 @ 2.20GHz, 128 GB RAM**
   - **Every machine running a different DBMS**
   - **Raw data files replicated in each machine**
2. **Compare PostgresRaw LIVE with MySQL, PostgreSQL, DBMS X**
   - **Startup cost, i.e. loading + creating indexes**
   - **Query execution times**
3. **How does PostgresRaw …**
   - **Adapt to changes in the workload?**
   - **Use positional maps and caching?**



ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
CWI
DATA-INTENSIVE APPLICATIONS AND SYSTEMS