Efficiently Executing UDFs via Batching

Kai Franz
Batching UDFs

- Naive UDF execution: process one row at a time
- Batching: apply each statement to all rows at the same time
- DBMS is good at set-oriented processing
Parameters & Local Variables

- Create temp table with a column for each variable

```sql
CREATE OR REPLACE FUNCTION maxPurchaseChannel(ckey INT, fromDateSk INT, toDateSk INT)
RETURNS VARCHAR(50)
LANGUAGE plpgsql
AS $$
DECLARE
  numSalesFromStore INT;
  numSalesFromCatalog INT;
  numSalesFromWeb INT;
  maxChannel VARCHAR(50);
...```

```sql
CREATE TABLE #temp
(
  ckey INTEGER,
  fromDateSk INTEGER,
  toDateSk INTEGER,
  numSalesFromStore INTEGER,
  numSalesFromCatalog INTEGER,
  numSalesFromWeb INTEGER,
  maxChannel VARCHAR(50),
  maxPurchaseChannel VARCHAR(50),
);```
Writing to Local Variables

- SET, SELECT INTO, etc. become UPDATE
Branching

- Froid-style boolean column for each branch predicate
- Statements inside branch guarded with WHERE

```sql
CREATE TABLE #temp
(
    ...
    p1 BIT,
    p2 BIT,
    p3 BIT,
    ...
)
);

UPDATE #temp
    SET p1 = CASE WHEN numSalesFromStore > numSalesFromCatalog THEN 1 ELSE 0 END;

UPDATE #temp
    SET maxChannel = 'Store'
WHERE p1 = 1;
```
Return Statements

- Boolean column tracks if row has returned yet

```sql
IF netProfit > 0 THEN
    RETURN 1;
ELSE
    RETURN 0;
END IF;
```

```sql
UPDATE #temp
    SET profitablemanager = 1,
        returned = 1
    WHERE p1 = 1
        AND returned = 0;

UPDATE #temp
    SET profitablemanager = 0,
        returned = 1
    WHERE p1 = 0
        AND returned = 0;
```
Experiments

- ProcBench benchmark, scale factor 1
- Compared against Froid
Experiments

- Postgres not good at decorrelation
- Batching works well with SQL->SQL decorrelation but unsure if this is a fair comparison

PostgreSQL 15

![Bar chart showing execution time for different queries. The chart includes bars for Froid, Batched (Correlated), Batched (Decorrelated), and Hand-written. The x-axis represents queries (Q1 to Q13) and the y-axis represents execution time in seconds.](chart.png)
Experiments

- CockroachDB is pretty good at handling Froid queries
- Temp tables still an experimental feature

CockroachDB v22.2.8

![Execution time chart for Q1 to Q13 with Froid and Batched execution times对比](image-url)
Very inconsistent with Froid queries
Batched queries have less variance
Future Work

- Test on MySQL, DuckDB
- Optimizing procedural code using large language models

Jo Kristian Bergum
@jobergum

Tensor and vector databases will replace most legacy databases in this decade. A disruption fueled by natural language interfaces and deep neural representations. In other words:

Natural query languages (NQL) replace the Istructured query language (SQL).
Thank you!