Incremental Memory Model for MonetDB

Pavlos Katsogridakis

September 28, 2017

Malcom

Overview

- MAL level query memory estimator
- Groups the MAL instructions into categories
- Uses previous query traces, and base column statistics

MAL example

```
1  select
2    sum(l_extendedprice * l_discount) as revenue
3  from
4    lineitem
5  where
6    l_shipdate >= date '1994-01-01'
7    and l_shipdate < date '1994-01-01'+interval '1' year
8    and l_discount between 0.06 - 0.01 and 0.06 + 0.01
9    and l_quantity < 24;
10
```

MAL example

```
1   X_16:= bind("sys","lineitem","l_quantity")
2   X_33:= bind("sys","lineitem","l_discount")
3   X_40:= bind("sys","lineitem","l_shipdate")
4   C_13:= tid("sys","lineitem")
5   C_51:= select(X_40,C_13,1994-01-01,1995-01-01)
6   C_65:= select(X_33,C_51,5,7)
7   C_68:= thetaselect(X_16,C_65,2400,"<")
8   X_26:= bind("sys","lineitem","l_extendedprice")
9   X_71:= projection(C_68,X_26)
10  X_72:= projection(C_68,X_33)
11  X_78:= *(X_71,X_72)
12  X_80:= sum(X_78)
13
```

What is memory footprint ?

Definition

The maximum memory usage in the query execution timeline

```
1 max_mem  = 0
2 curr_mem = 0
3 for i in ilist:
4   max_mem  = max(max_mem,curr_mem + i.mem_fprint)
5   curr_mem = curr_mem + i.mem_fprint-i.free_size
```

Goal

Take the MAL instruction plan and give an estimation of the memory footprint

- Use only traces from previous executions and basic column statistics

MAL Instruction Grouping

- Arithmetic (+,-,...)
- Aggregate (avg, sum, ...)
- Limit (firstn, sample)
- Grouping (groupdone)
- Selection (select, thetaselect)
- Join
- Set (intersect, merge, diff)

Output Count Prediction

Arithmetic Operators

Output = Input

Aggregate Operators

Output = 1

Limit Operators

Output = min(arg size,N)

Set Operators

Merge(A,B)

Output = size(A)+size(B)

Intersect(A,B)

Output = min(size(A),size(B))

Diff(A,B)

Output = size(A)

Select Operators

Range Select

- Keep a dictionary of all the previous selections
- Find all the instructions having the same column, op
- Run a kNN to find the 5 nearest based on the range
- Extrapolate based on the selectivity and arg size
- extrap(testi, traini) =
  $traini.count * \frac{testi.estim\_arg\_cnt}{traini.arg\_cnt} * \frac{testi.range}{traini.range}$

Point Select

- Find all the instructions having the same column, op
- Run a kNN to find the 5 nearest selects
- Extrapolate based on the arg size

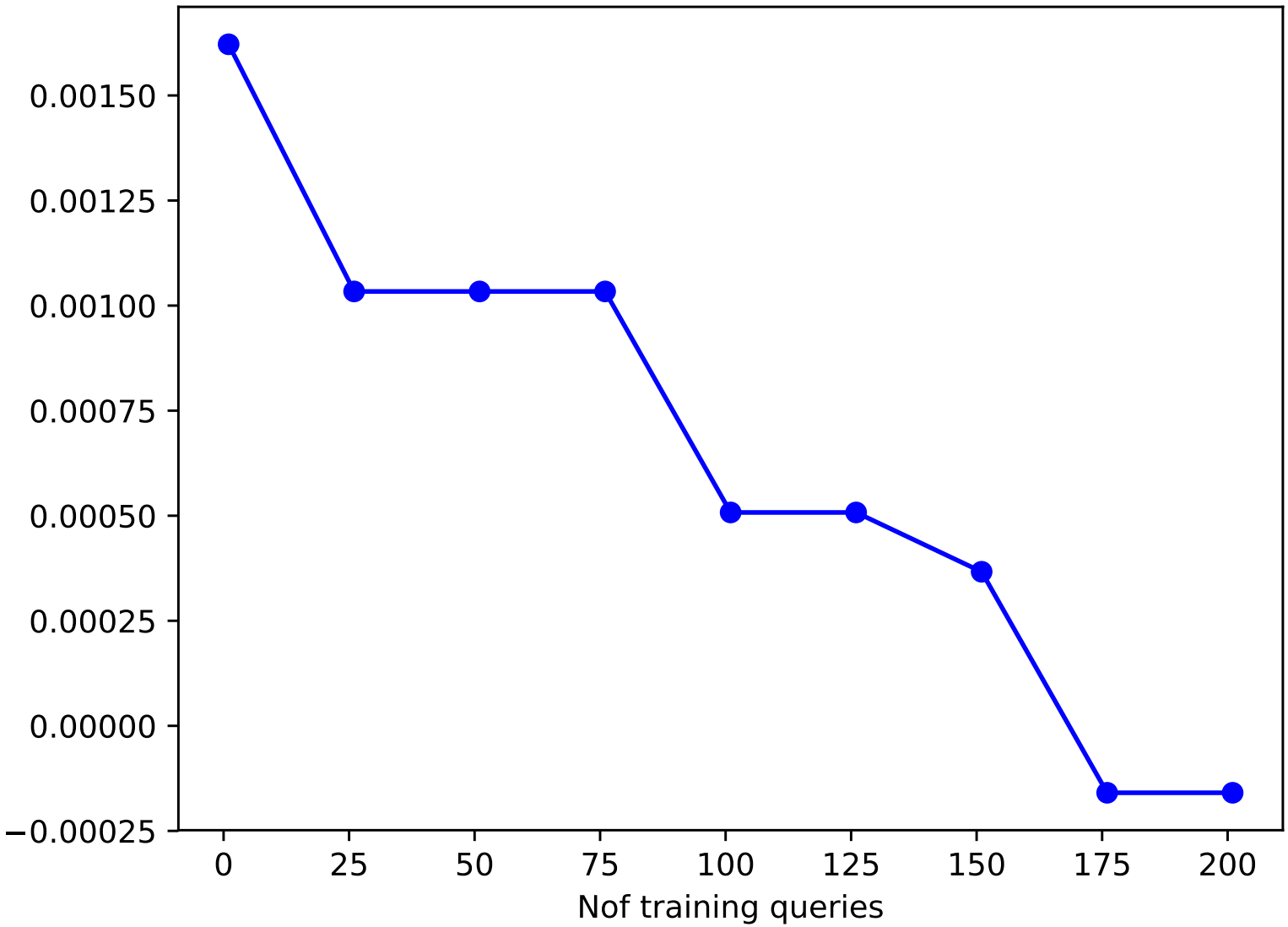
Join Operators

- Find all the instructions on the same columns
- Run a kNN to find the 5 nearest based on the arg size
- Extrapolate based on the arg size

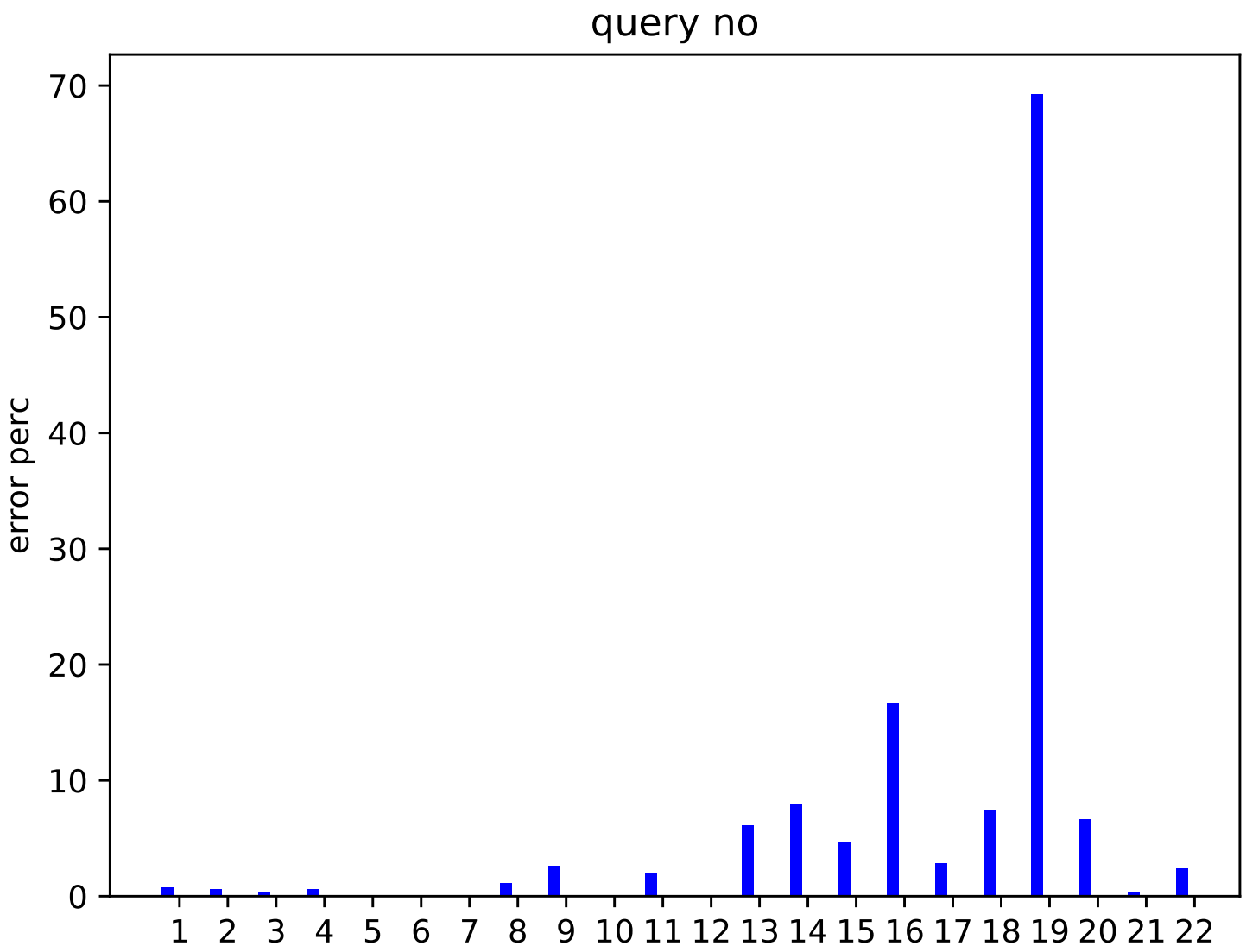
Tpch sf10 Evaluation

- Produce 200 queries randomizing the selection values
- Keep the original query as test set
- Error = 100 * (predict_mem - real_mem) / real_mem

Tpch sf10 Q6

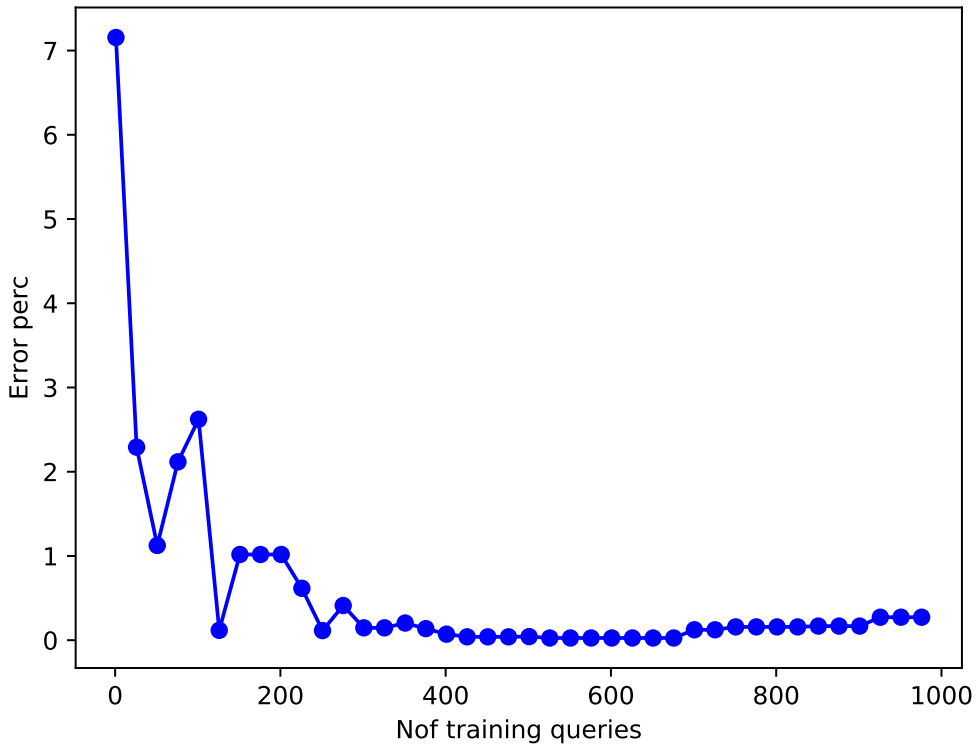

0.00150
0.00125
0.00100
0.00075
0.00050
0.00025
0.00000
−0.00025
0
25
50
75
100
125
150
175
200
Nof training queries

Tpch sf10


query no
error perc
0
10
20
30
40
50
60
70
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

Airtraffic benchmark suite

```
1 SELECT SQL_MIN("Origin", "Dest"),
2        SQL_MAX("Origin", "Dest") AS route,
3        COUNT(*) FROM ontime
4 WHERE 2010-09-11<FlightDate AND FlightDate<2011-09-11
5 GROUP BY route;
```



```
1 SELECT "DayOfWeek", COUNT(*) AS "Flights"
2 FROM ontime
3 WHERE "DepDelay" > 15
4 GROUP BY "DayOfWeek"
5 ORDER BY "DayOfWeek";
```

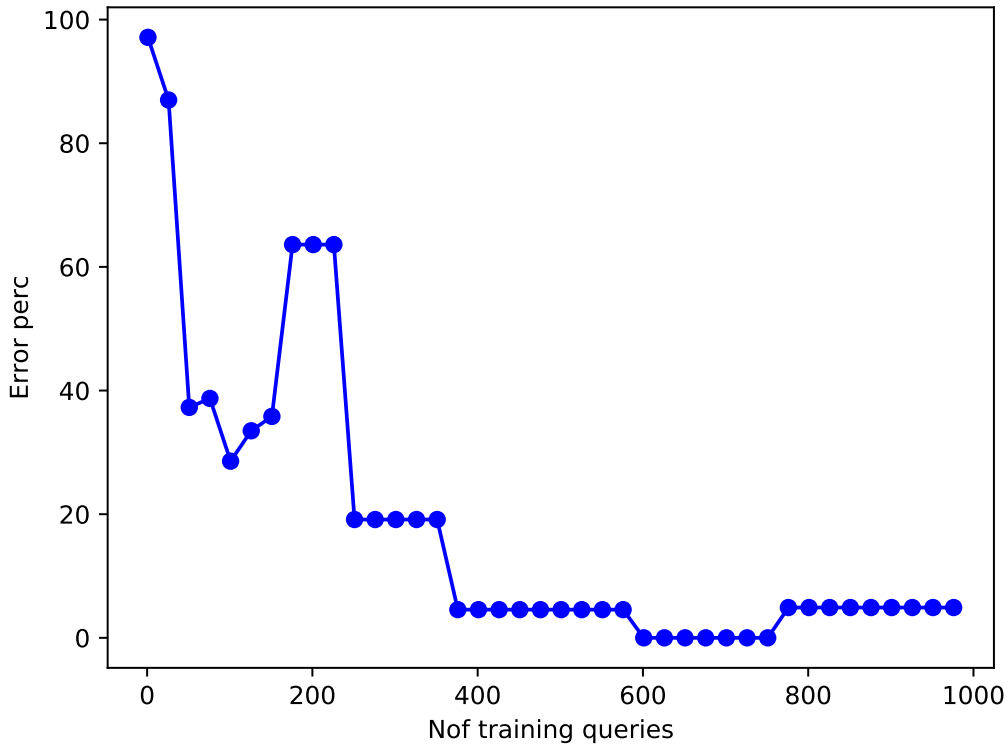


Figure: Q04 memory error

```
1 SELECT "Origin" AS ap, COUNT(*) AS cnt_in
2 FROM ontime
3 WHERE "Dest" = 'ORD'
4 GROUP BY "Origin"
```

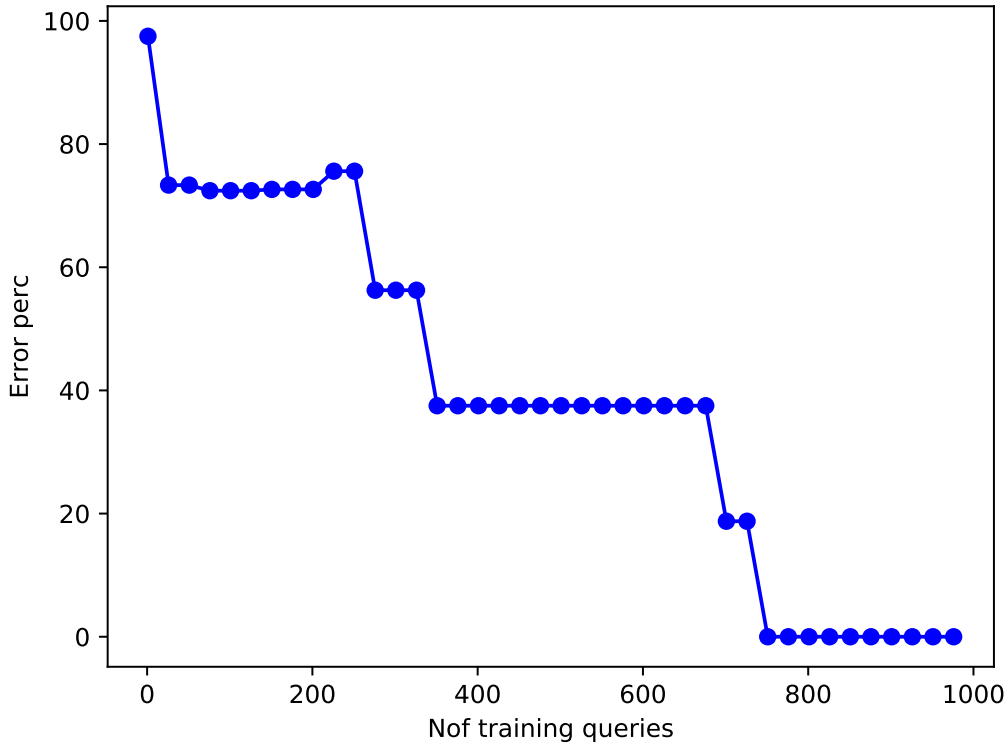


Figure: Q10 select error

```
1 SELECT *
2 FROM ontime
3 WHERE "DepDelay" > 15 AND "ArrDelay" > 15
4 AND "Month" = 3 AND "DayofMonth" = 24 AND "Year" = 2013
```

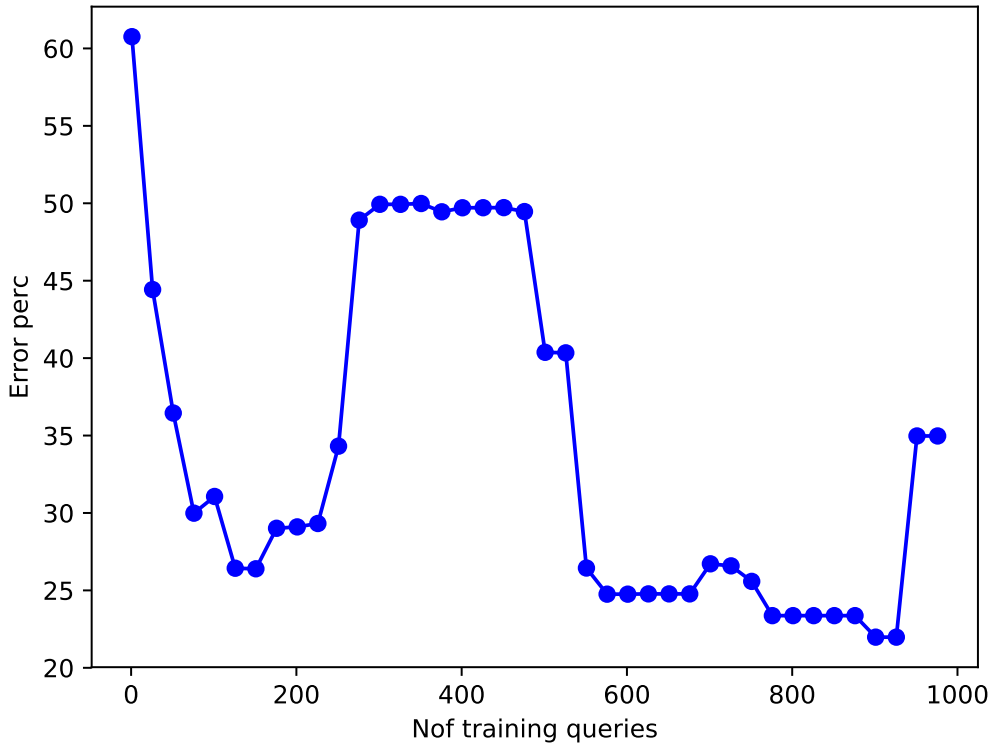


Figure: Q09 select error

Future Work

- Provide min,avg,max policies
- Consider parallel executions
- Handle correlated columns on selections
- Also use histograms