

DBDBD-2016

Incremental evaluation of updates in factorized in-memory databases

Muhammad Idris

Ph.D. Student-(ULB, TUD)

Email: Muhammad.idris@ulb.ac.be

Joint work with: Martin Ugarte, Stijn Vansummeren

Wednesday, October 26, 2016

Incremental view maintenance (IVM)

1- IVM: Re-evaluation of the query, Expensive in live updates

2- HIVM : Materializing auxiliary views in-memory:

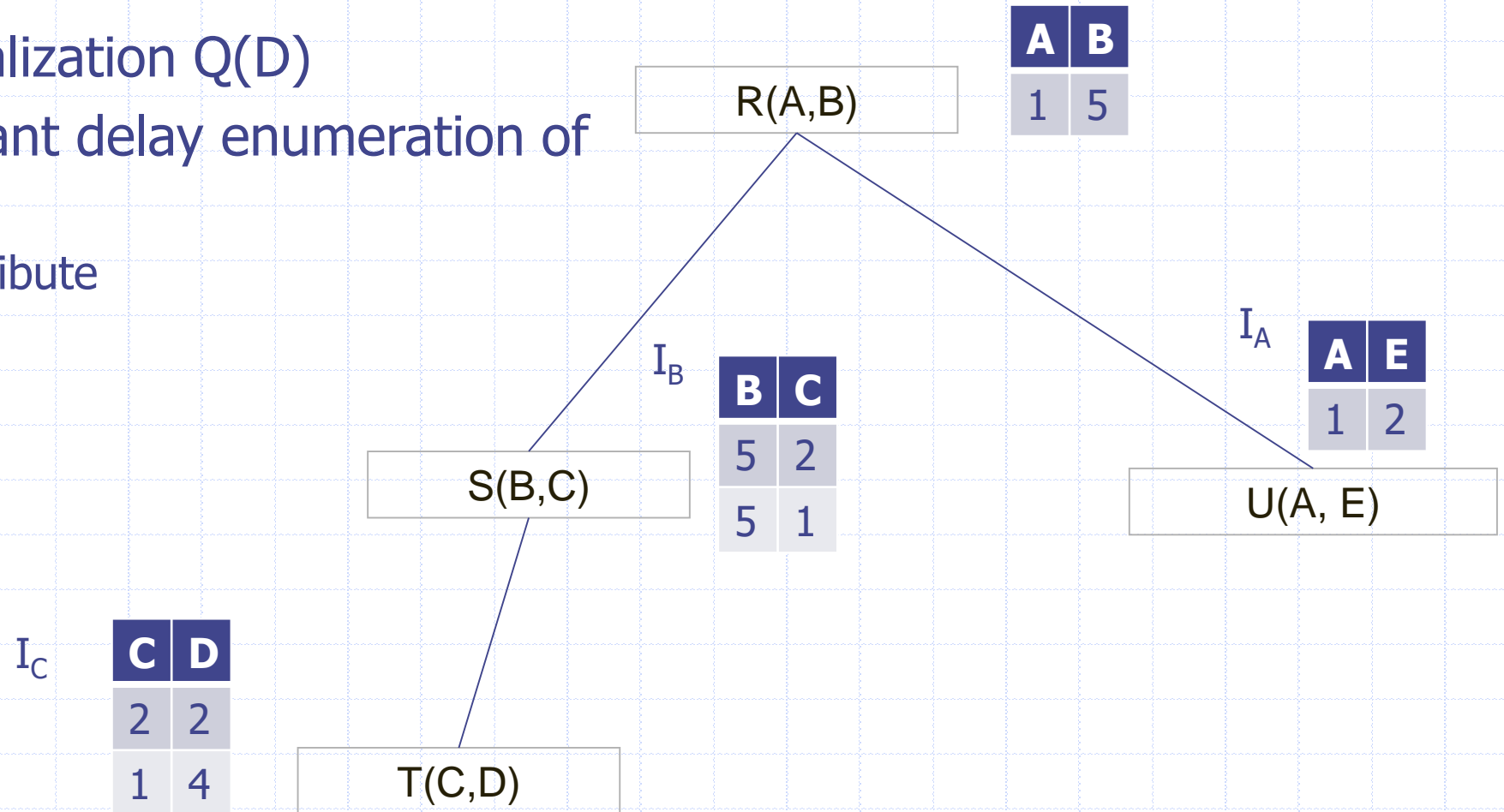
- High memory footprint
- Trades space for time

Can we do better?

- Less memory footprint
- Efficient processing time

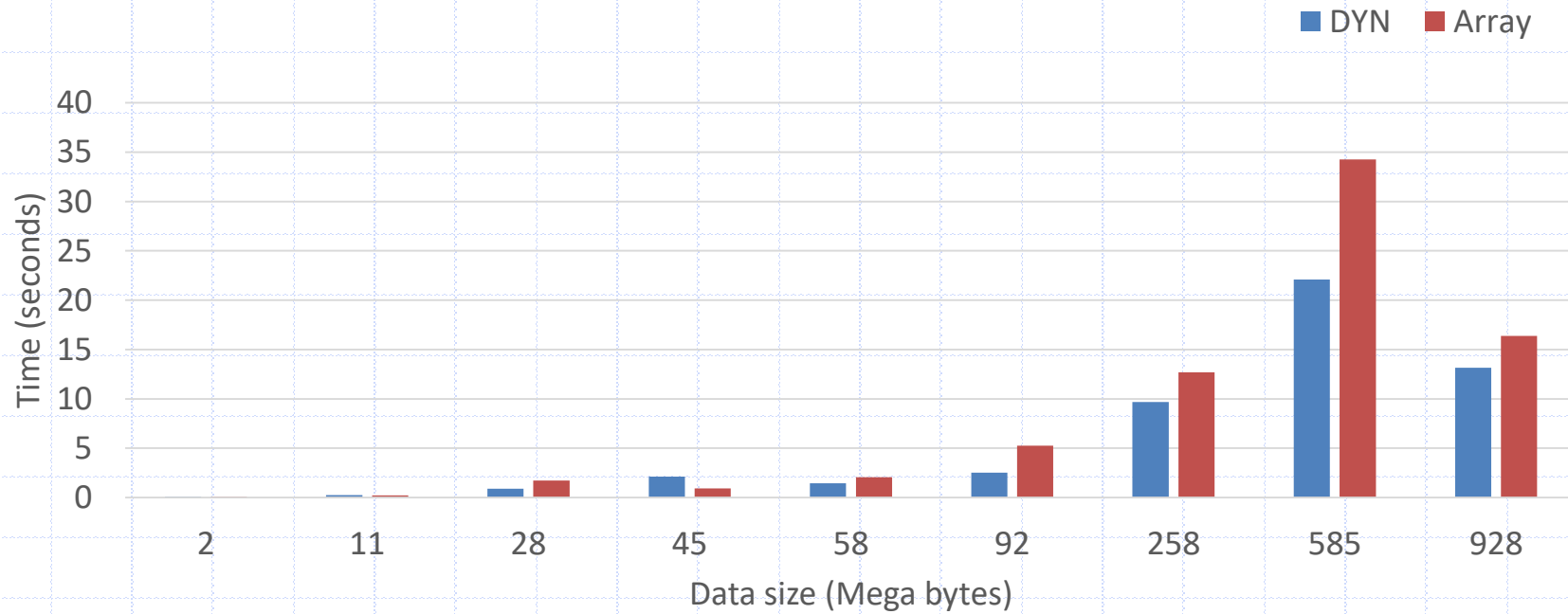
Yes!

- ◆ Give up materialization $Q(D)$
- ◆ Maintain Constant delay enumeration of $Q(D)$
 - All tuples contribute
 - Indexed



Experimentation

◆ Constant delay enumeration



All tuples contribute to the result?

Yannakakis Algorithm

Yannakakis Algorithm

$$Q = R(A, B) \bowtie S(B, C) \bowtie T(C, D)$$

1. Bottom-up semi joins
2. top-down
3. Bottom

Result n
 Static al
 $O(\ln + C)$

A	B	C	D
1	5	2	2
1	5	1	4

A	B	C	D	E
1	5	2	2	2
1	5	1	4	2

A	B
1	5
2	3

R(A,B)

How can we do with dynamic updates?

C	D
2	2
4	3
1	4

T(C,D)

3	2
4	3
5	1

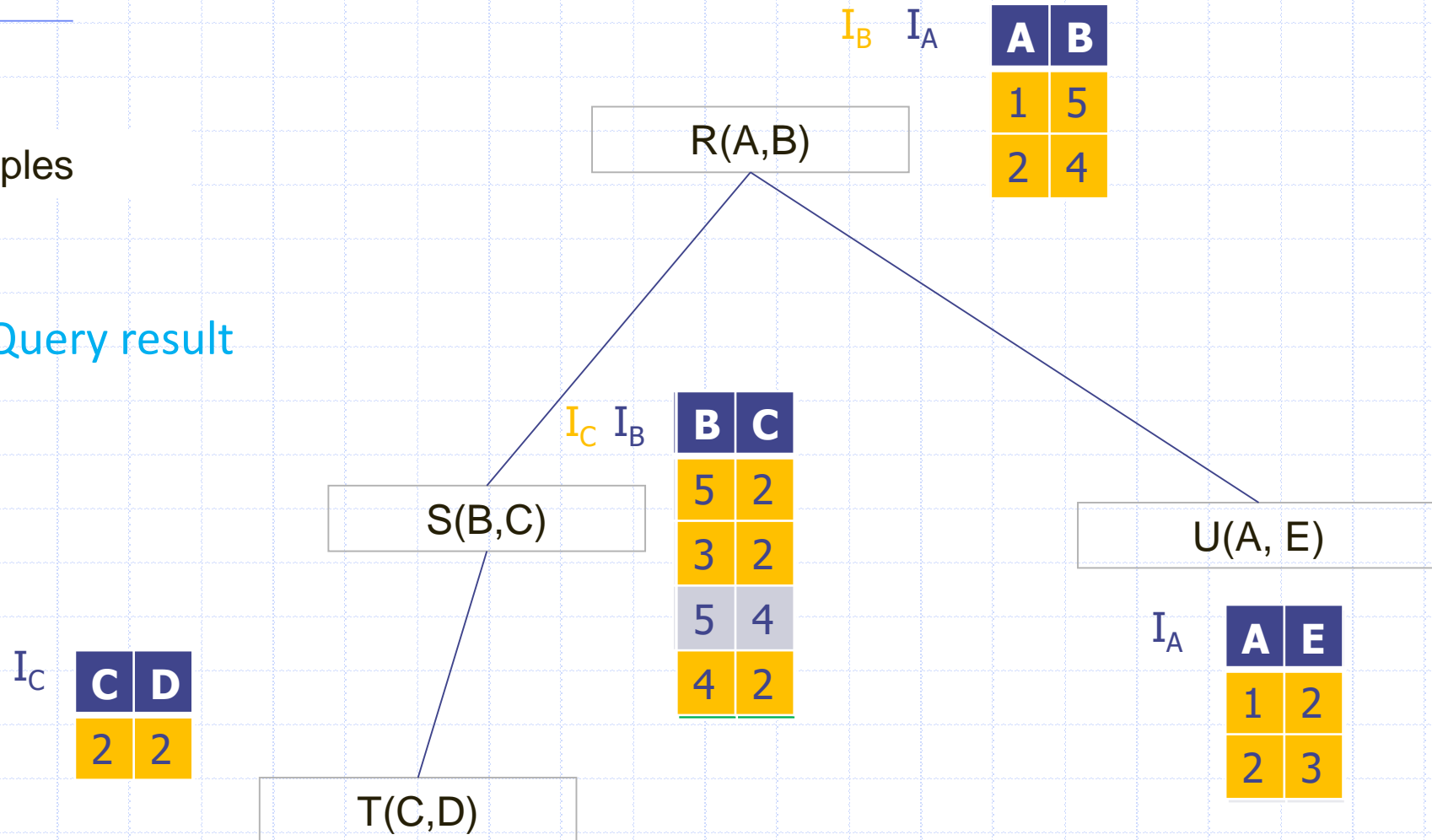
U(A, E)

A	E
1	2
6	3

Dynamic Yannakakis

1. Live tuples

Factorized Query result



Experimentation

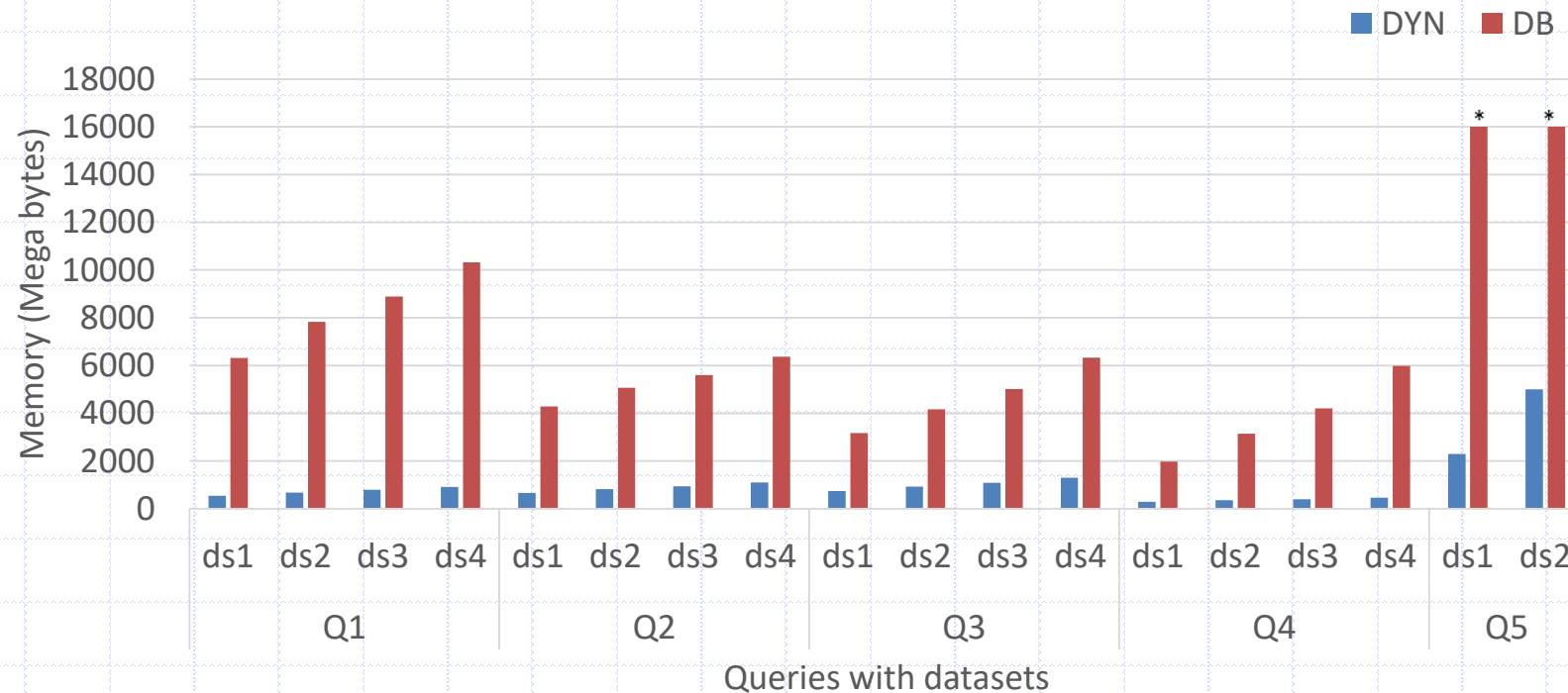
◆ DYN vs DBToaster

◆ Memory Footprint and Update processing time

Column1	ds1	ds2	ds3	ds4	
Q1		1199969	2000497	2999671	6001215
Q3		1529969	2550497	3824671	7651215
Q4		1499969	2500497	3749671	7501215
Q6		1199969	2000497	2999671	6001215
Q9		1701994	2837185	4254696	8511240
Q10		1529994	2550522	3824696	7651240
Q11		162025	270022	405025	810025
Q13		330000	550000	825000	1650000
Q15		1201969	2003830	3004671	6011215
Q16		202000	336663	505000	1010000
Q18		1529969	2550497	3824671	7651215
Q4-full join		757056	973153	1135047	1361969
Q3-full join		940388	1208866	1410047	1691969
Q2-full join		872611	1121725	1308382	1569969
Q1-full join		944833	1214581	1416714	1699969
Q5-full join		4892553	7780920		

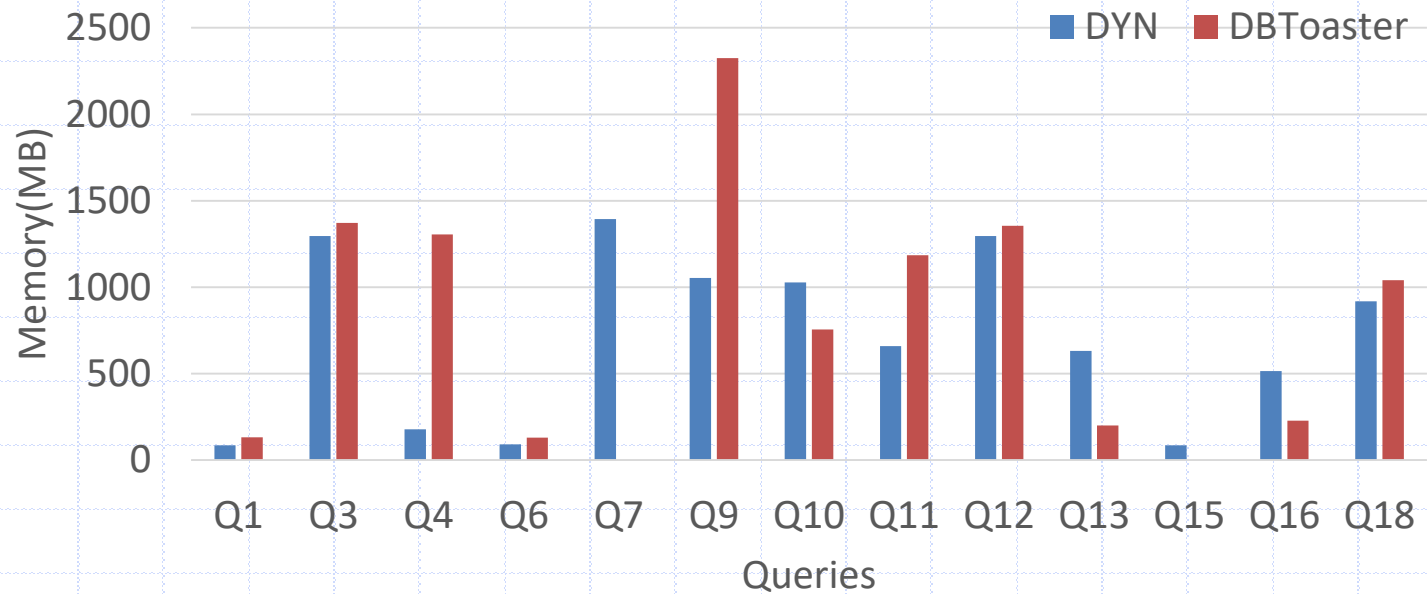
Experimentation

◆ Memory footprint – TPC-H Full Join Queries



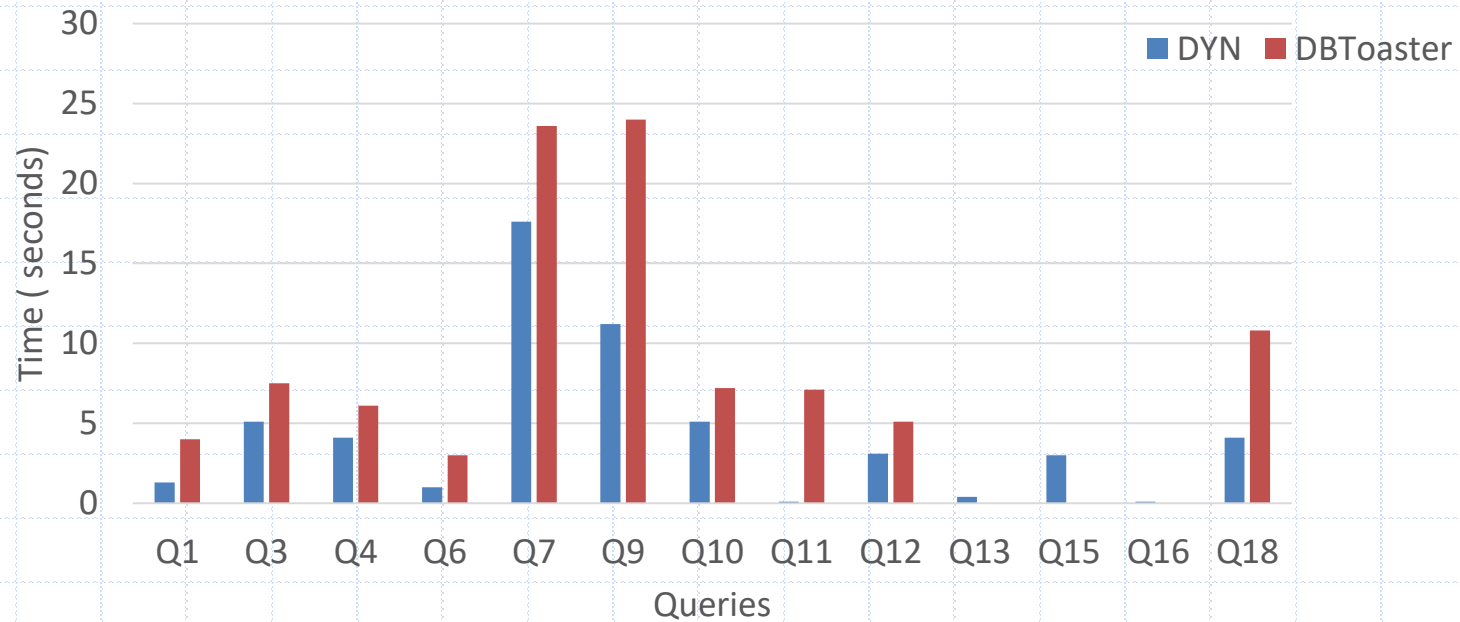
Experimentation

◆ Memory footprint – TPC-H aggregate queries



Experimentation

◆ Processing time— TPC-H aggregate queries





THANK YOU!

Any questions or comments?

