Declarative Partitioning Has Arrived!

Ashutosh Bapat (EnterpriseDB)

Amit Langote (NTT OSS center)

@PGConf.ASIA 2017

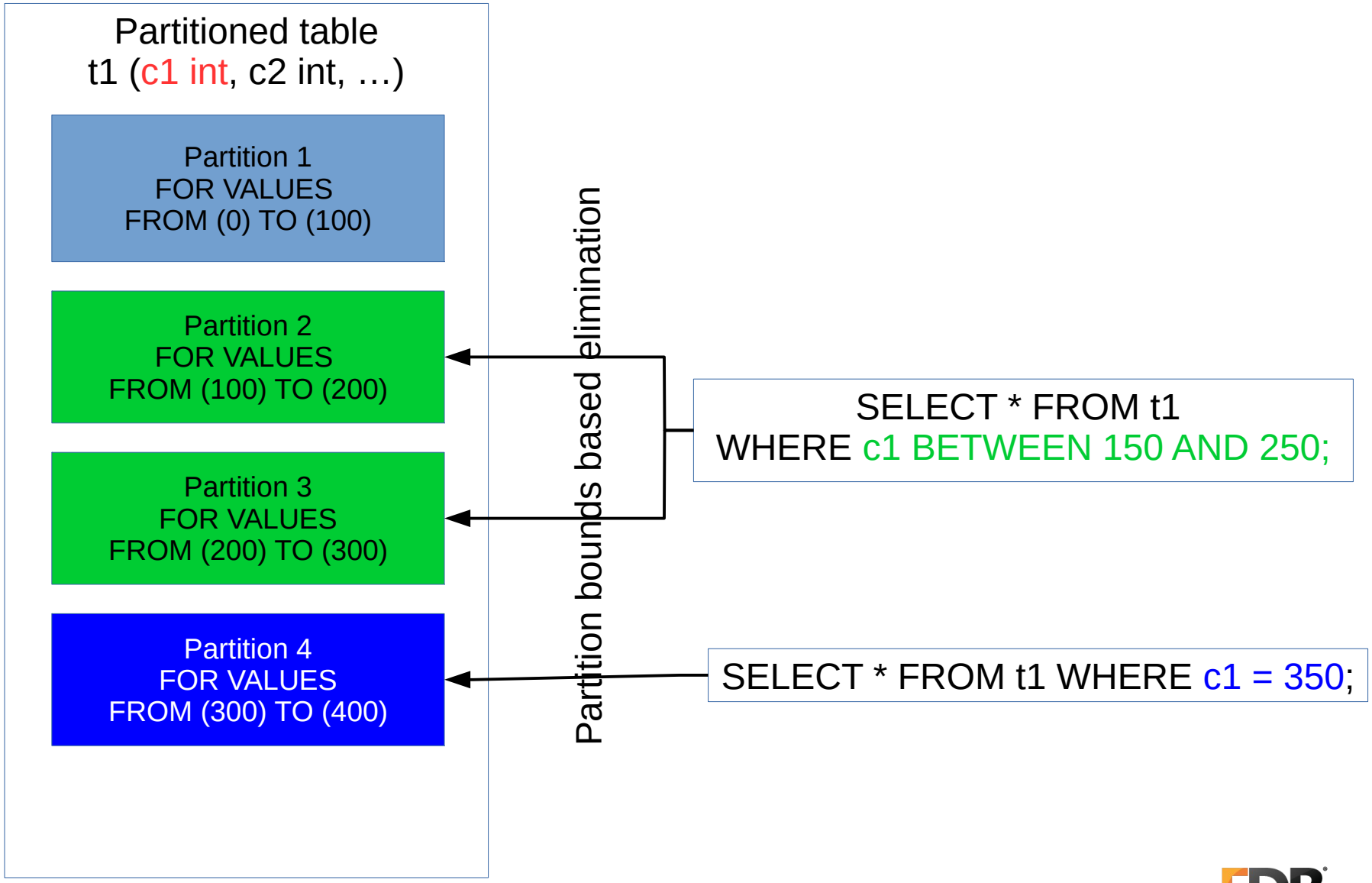
Query Optimization Techniques

- Partition pruning
- Run-time partition pruning
- Partition-wise join
- Partition-wise aggregation
- Partition-wise sorting/ordering

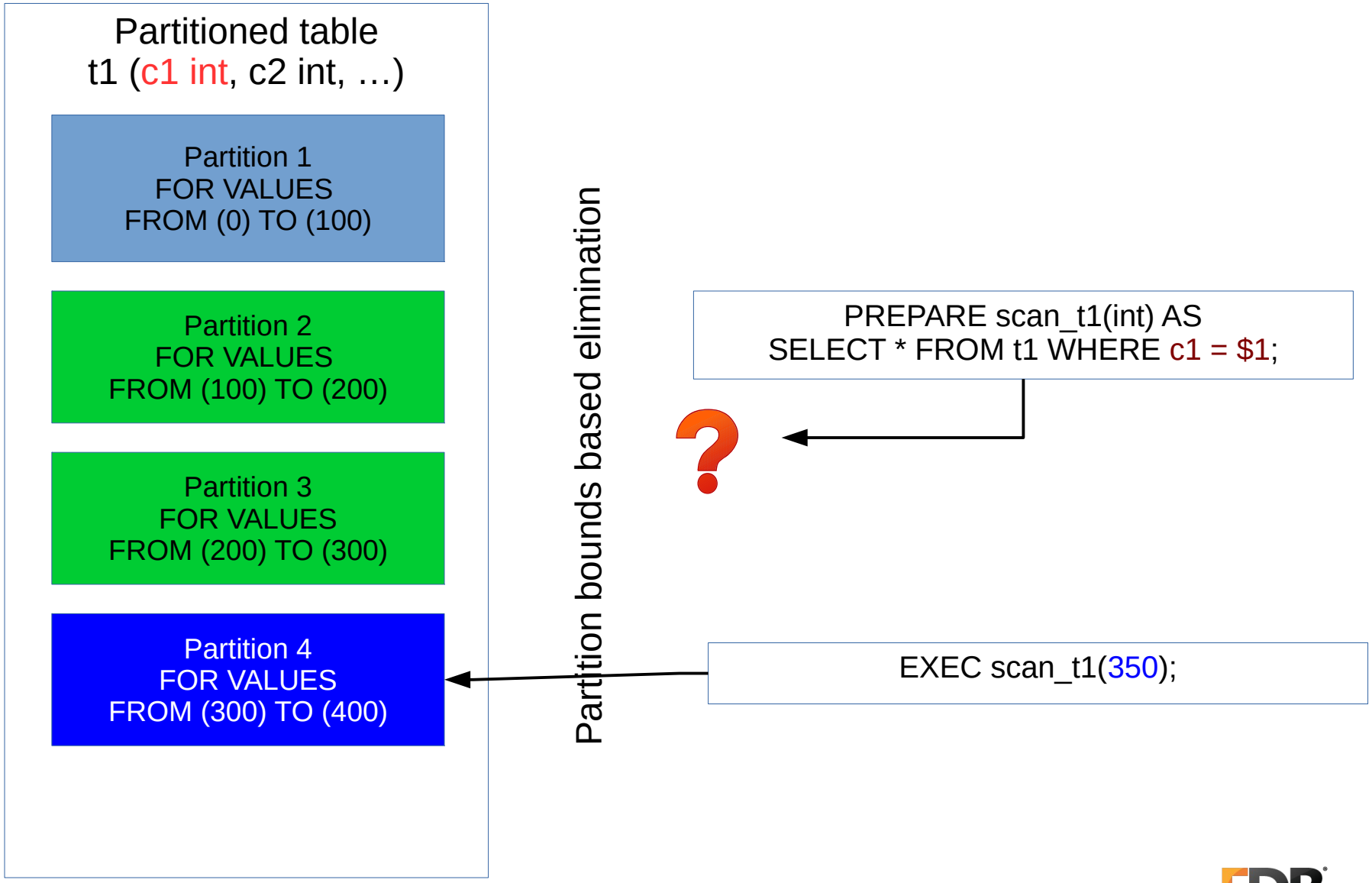
Partition-wise Operations

- Push operations down to partitions
- Improve performance by exploiting properties of partitions
  - Indexes, constraints on partitions
- Faster algorithms working on smaller data
  - Smaller hash tables
  - Faster in-memory sorting
- Parallel query: one worker per partition
- FDW push-down for foreign partitions
- Eliminate data from pruned partitions

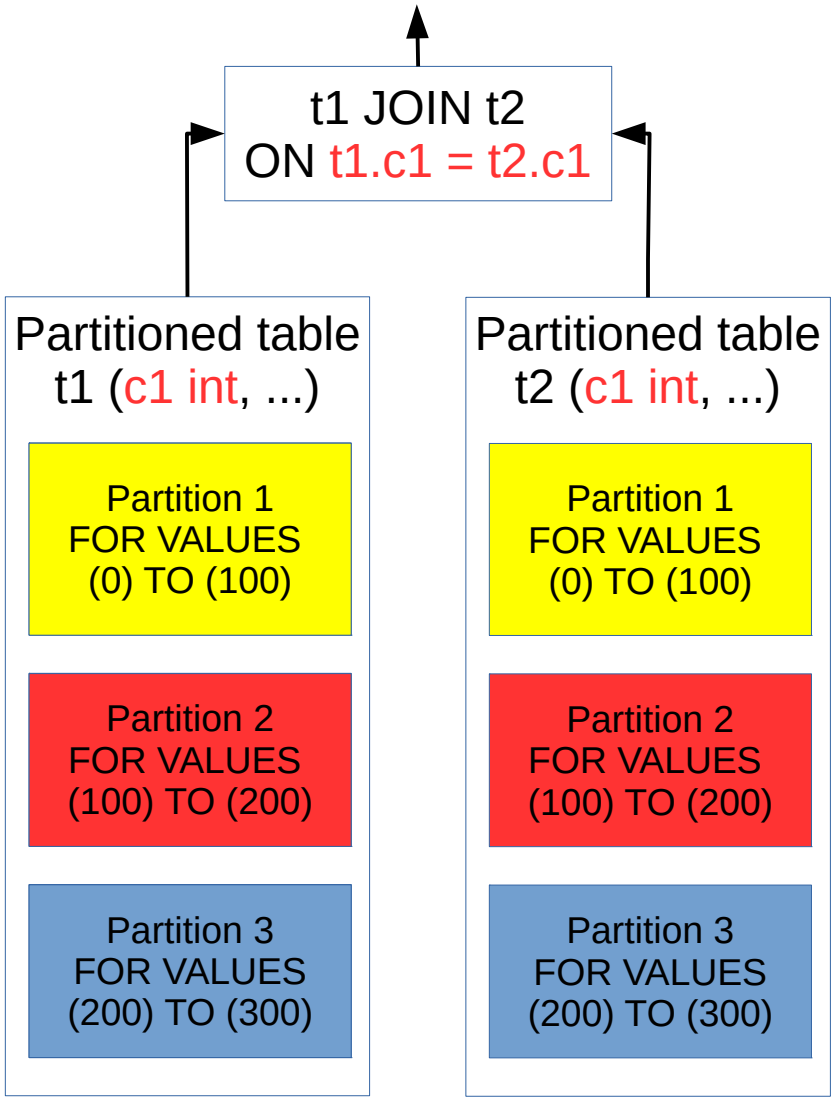
Partition Pruning

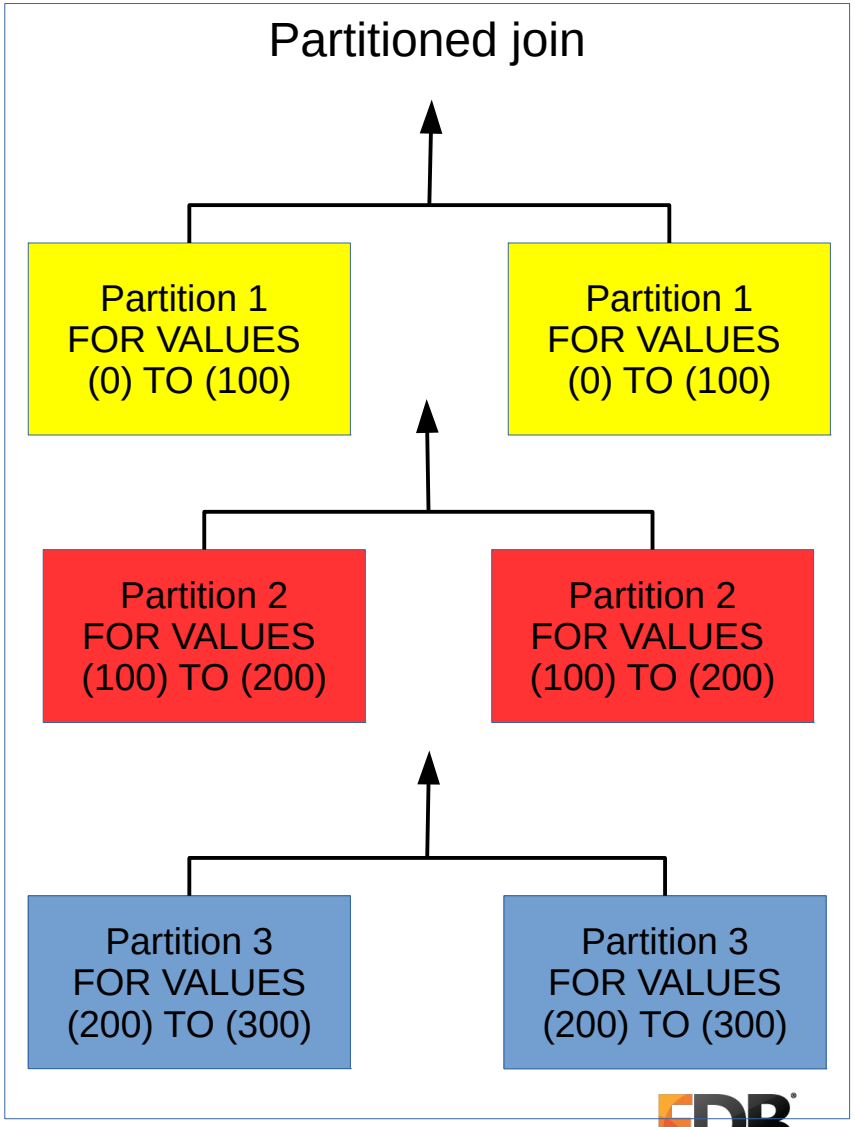


Run-time Partition Pruning



Partition-wise Join





Partition-wise Join Performance

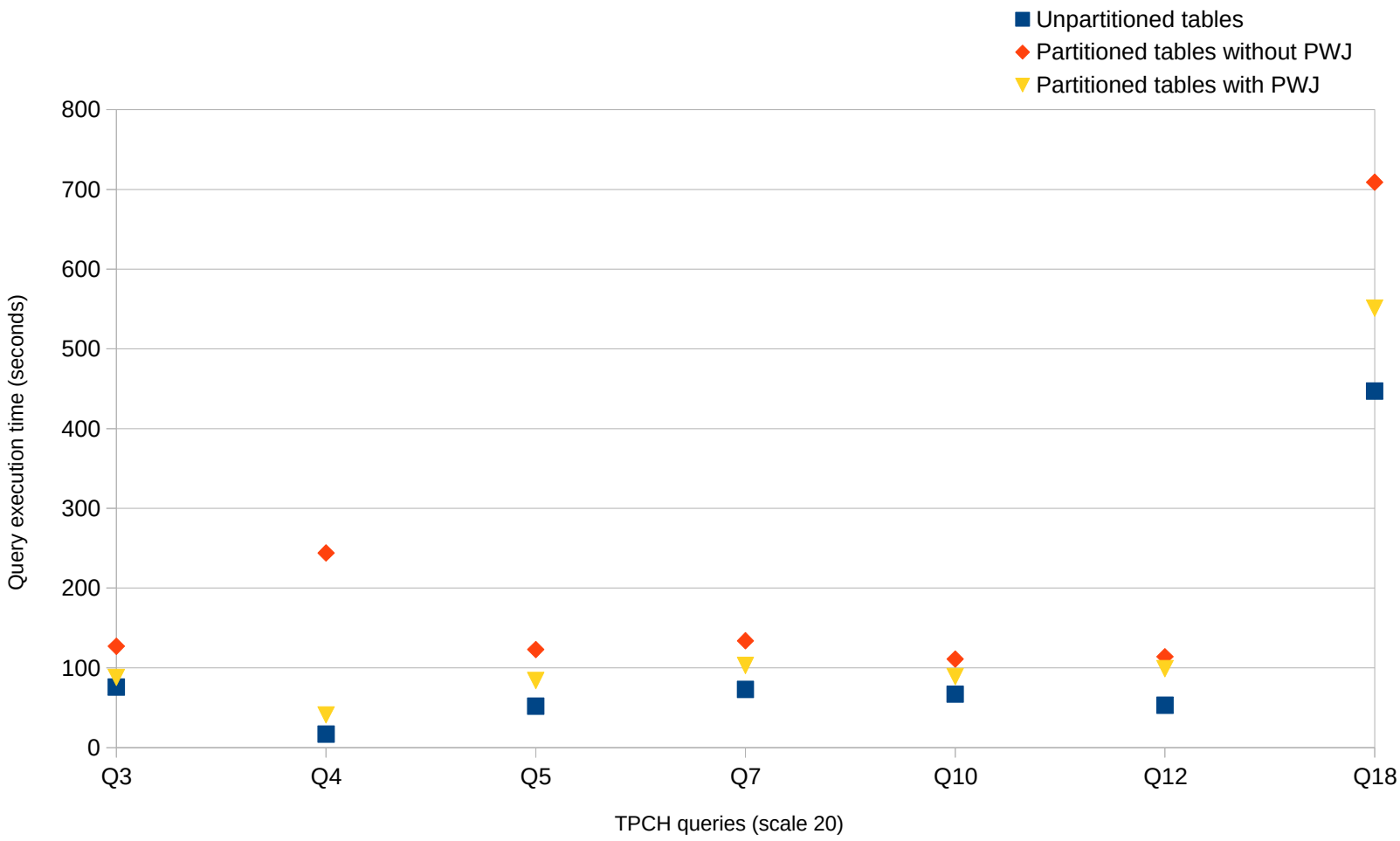
- Different join strategy for each child join
  - Based on properties of partitions like indexes, constraints, statistics, sizes etc.
- Cheaper strategy for smaller data instead of expensive strategy for large data
  - hash join instead of merge join
  - parameterized nested loop join instead of hash/merge join
- Each child-join may be executed in parallel
- Child-join pushed to the foreign server
  - Partitions being joined reside on the same foreign server

TPCH Vs. Partition-wise Join

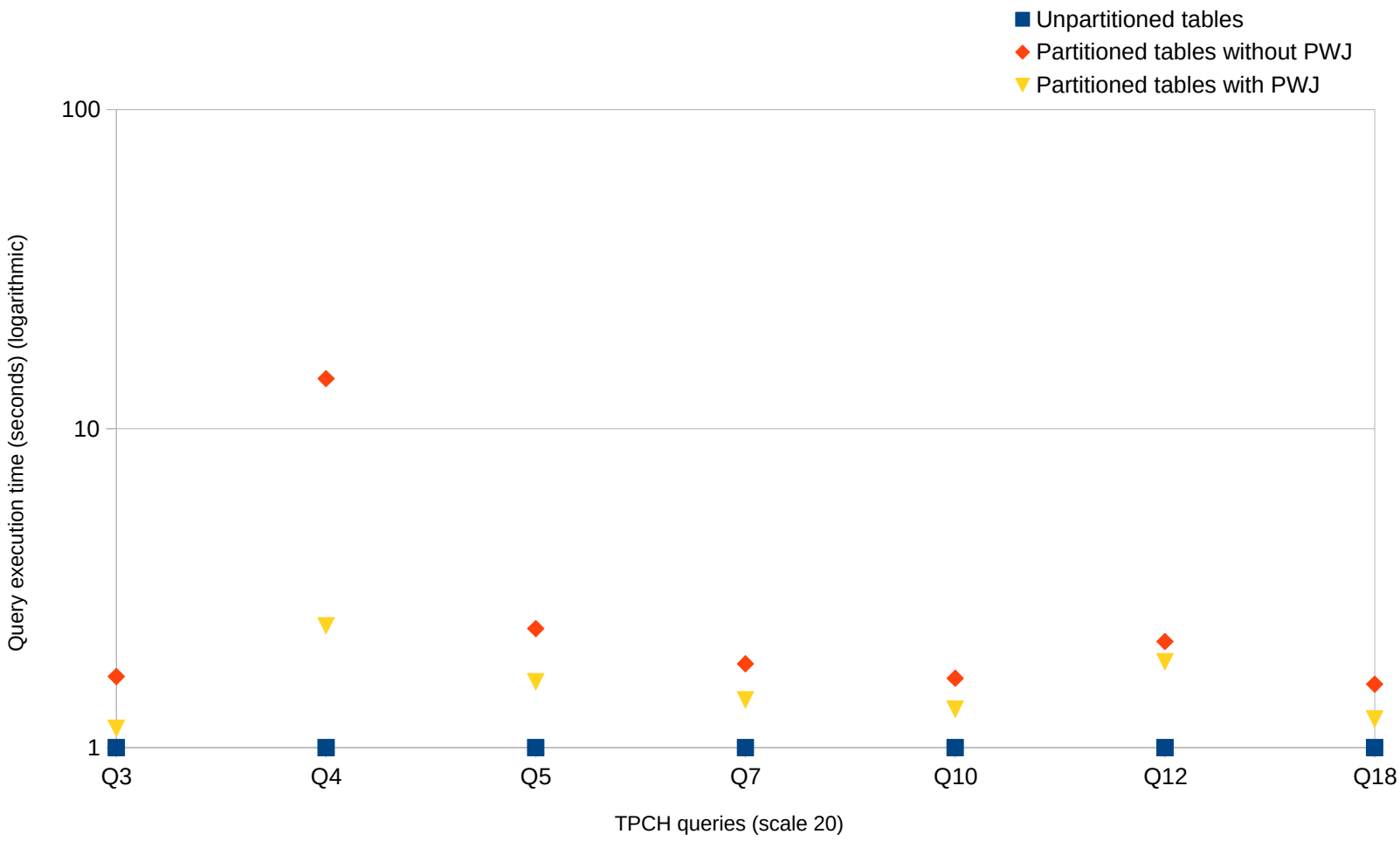
Reported by Rafia Sabih

- Scale 20
- Schema changes
  - lineitems PARTITION BY RANGE(l_orderkey)
  - orders PARTITION BY RANGE(o_orderkey)
  - Each with 17 partitions
- GUCs
  - work_mem - 1GB
  - effective_cache_size - 8GB
  - shared_buffers - 8GB
  - enable_partition_wise_join = on

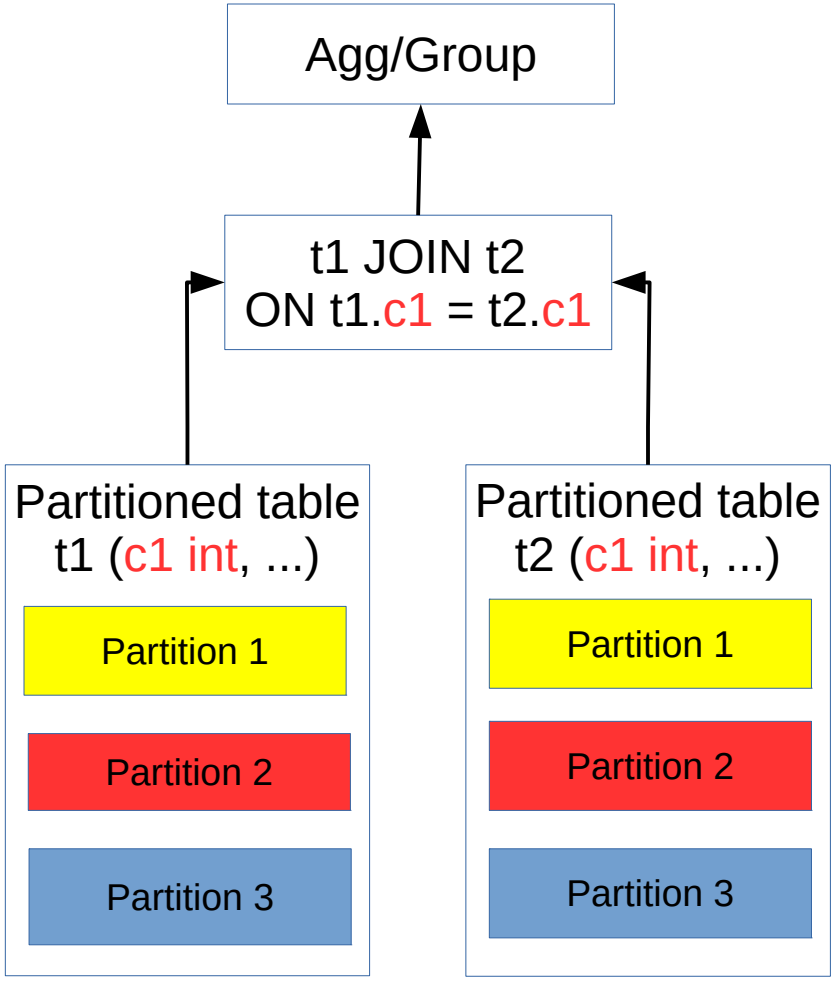
TPCH Vs. Partition-wise join

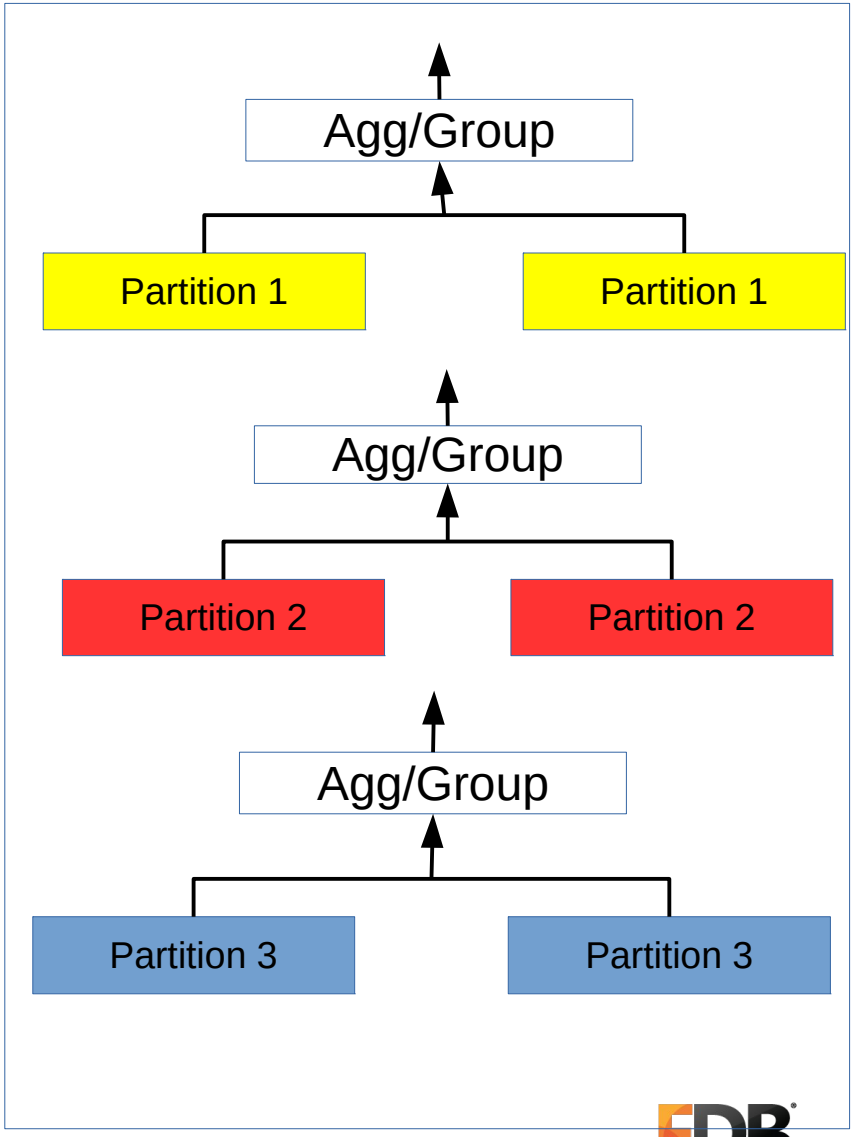


TPCH Vs. Partition-wise join



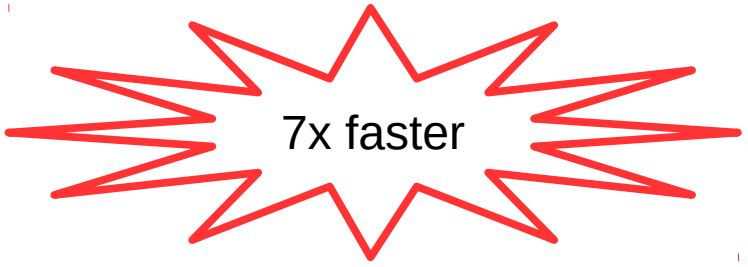
Partition-wise aggregation



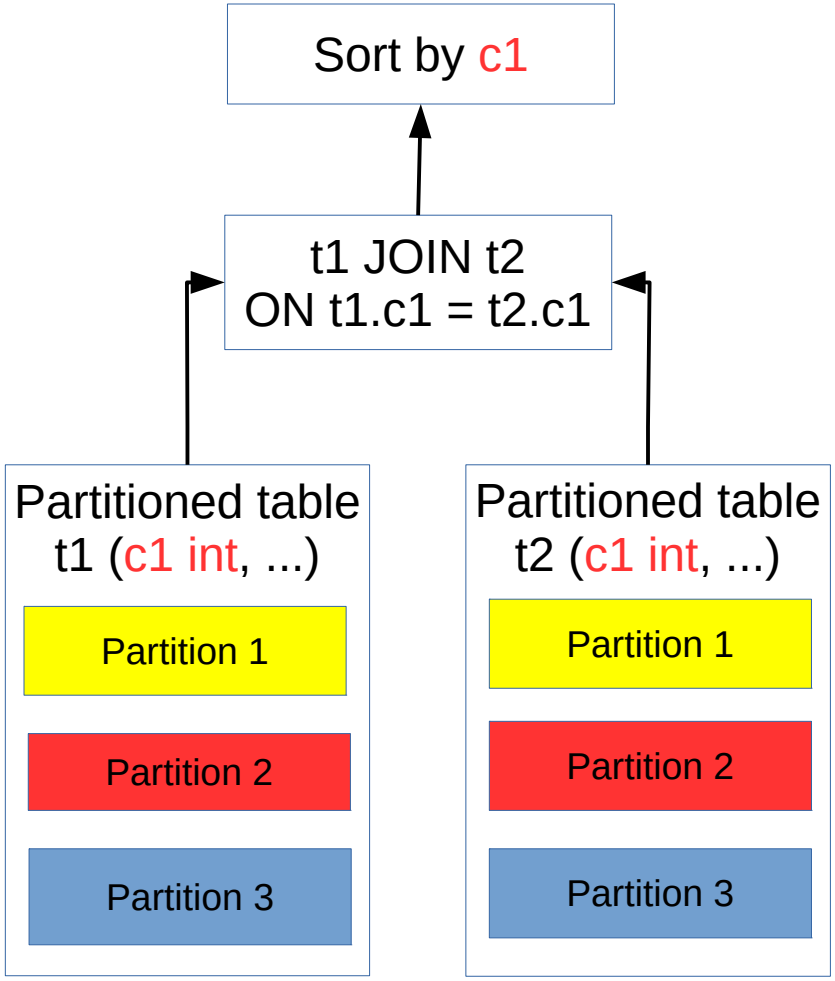


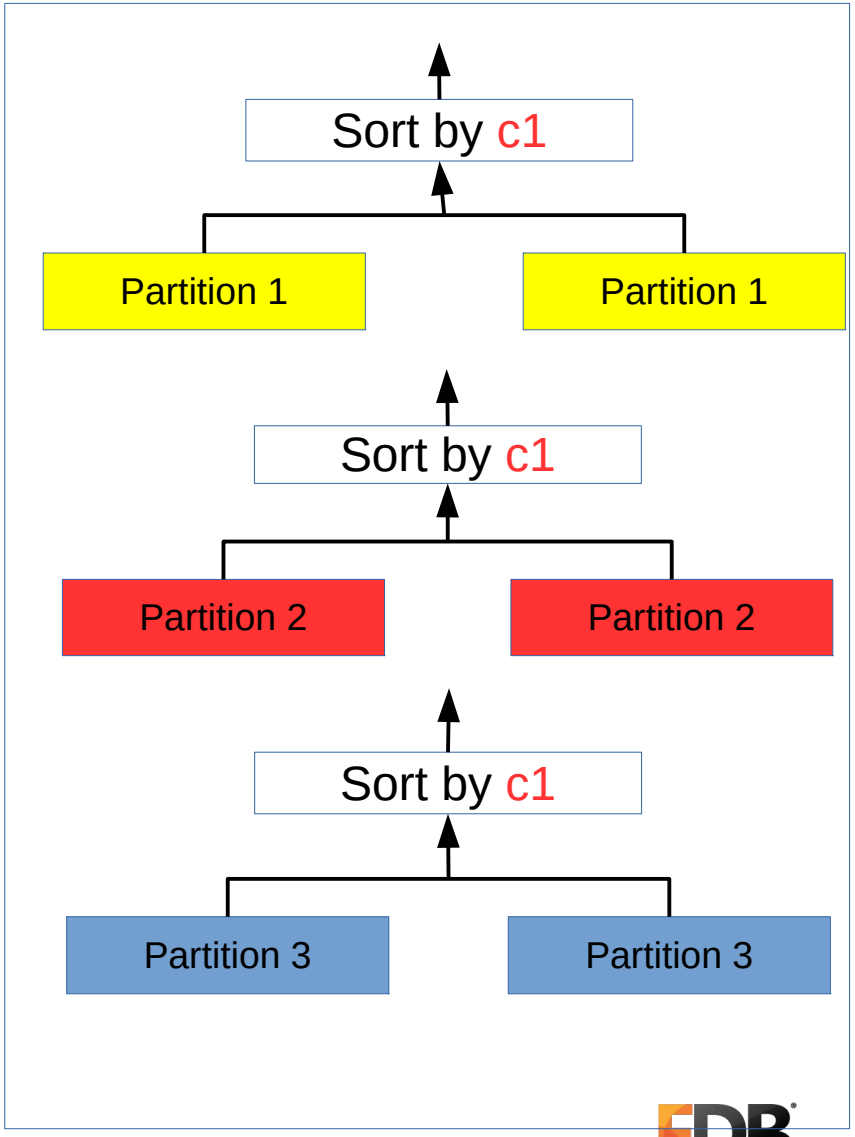
Example

```
Source: Jeevan Chalke's partition-wise aggregate proposal
Query: SELECT a, count(*) FROM plt1 GROUP BY a;
plt1: partitioned table with 3 foreign partitions, each with 1M rows

Query returns 30 rows, 10 rows per partition
enable_partition_wise_agg to false
QUERY PLAN
------------------------------------
 HashAggregate
   Group Key: plt1.a
   ->  Append
         ->  Foreign Scan on fplt1_p1
         ->  Foreign Scan on fplt1_p2
         ->  Foreign Scan on fplt1_p3
 Planning time: 0.251 ms
 Execution time: 6499.018ms ~ 6.5s

enable_partition_wise_agg to true
QUERY PLAN
-----------------------------------------------------
 Append
   ->  Foreign Scan: Aggregate on (public.fplt1_p1 plt1)
   ->  Foreign Scan: Aggregate on (public.fplt1_p2 plt1)
   ->  Foreign Scan: Aggregate on (public.fplt1_p3 plt1)
 Planning time: 0.370ms
 Execution time: 945.384ms ~ .9s
```



Partition-wise sorting





Query Optimization Techniques - patches

- Committed patches
  - Basic partition-wise join - Ashutosh Bapat - EDB
- Patches submitted on hackers and being reviewed
  - Partition pruning – Amit Langote - NTT
  - Run-time partition pruning – Beena Emerson - EDB
  - Partition-wise aggregation – Jeevan Chalke - EDB
  - Partition-wise sorting/ordering – Ronan Dunklau, Julien Rouhaud - Dalibo


THANK YOU
grazie
merci
spasiba
kam ouen
tak
gratzias
manana
mahalo
kitos
cheers
hvala
welalin
toda
gracias
grassie
thank you
danki
mahalo
danki
gracias
merci
thanks
na gode
mesi
dankon
modupe
talofa
miigwetch
thanks
takk
domo arrigato
danke
kitos
takk
dziekuje
grattitude
takk