ORCA and PostgreSQL

Venkatesh Raghavan

South Bay PostgreSQL Meetup, San Jose

Pivotal

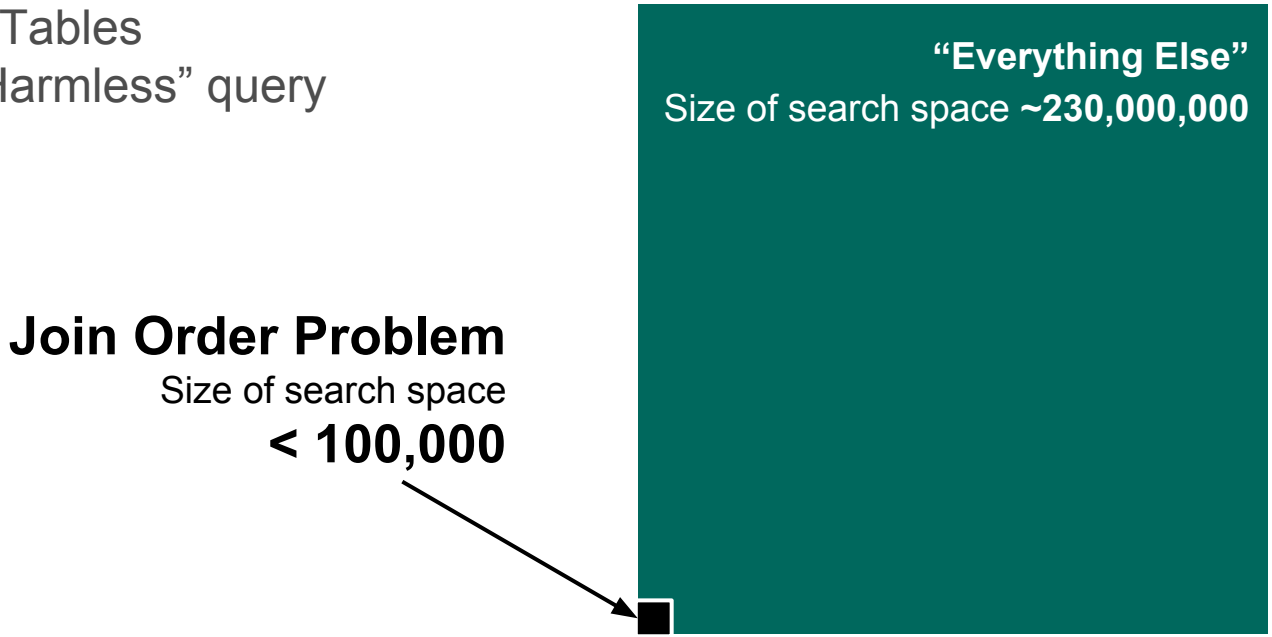
PostgreSQL Planner

- Addresses join re-ordering
- Treats everything else as add-on (grouping, with clause, etc.)
- Imposes order on specific optimization steps
- Recursively descends into subqueries
  - Sometimes planner is unable to unnest complex correlated subqueries
- High code complexity
  - Maximum: 102 (Orca 8.5)
  - Minimum: 6.4 (Orca 1.5)

Pivotal

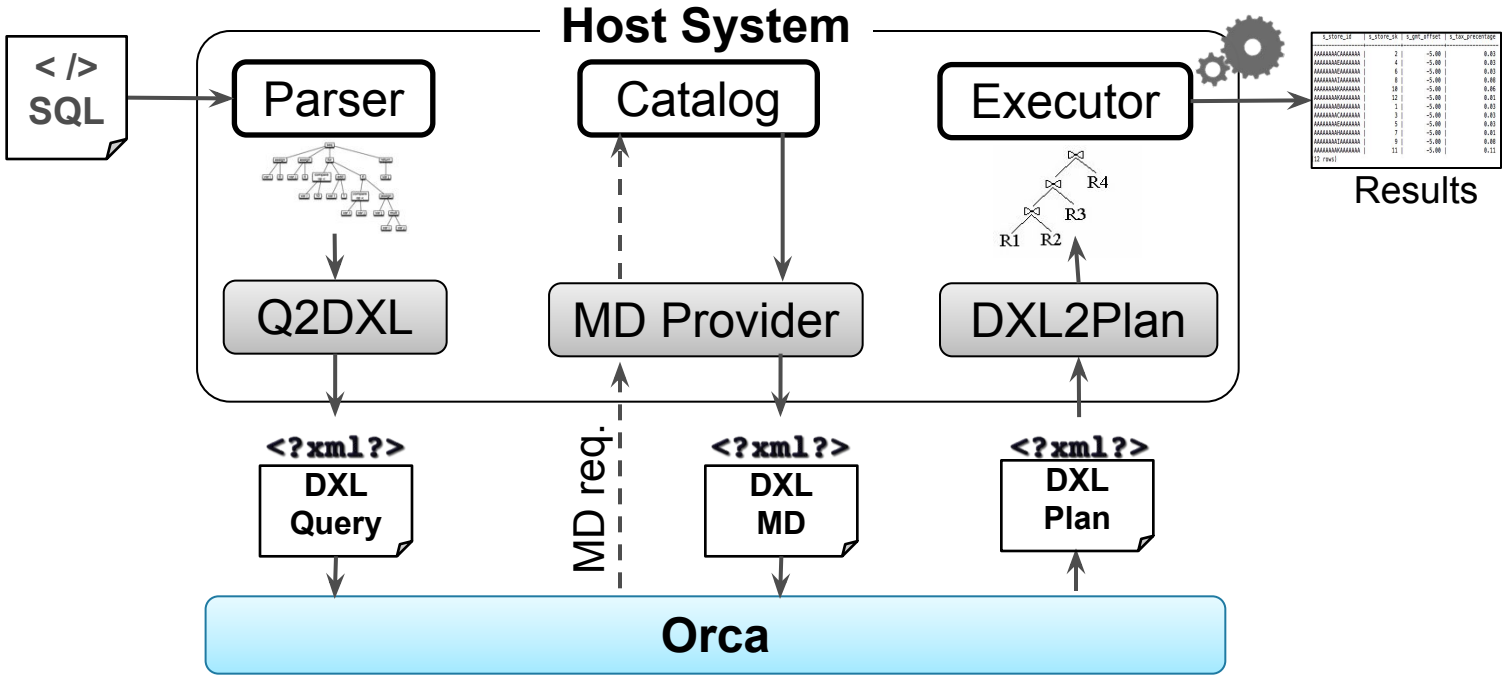
Join Ordering vs. “Everything Else”

- TPC-H Query 5
  - 6 Tables
  - “Harmless” query



Orca Architecture

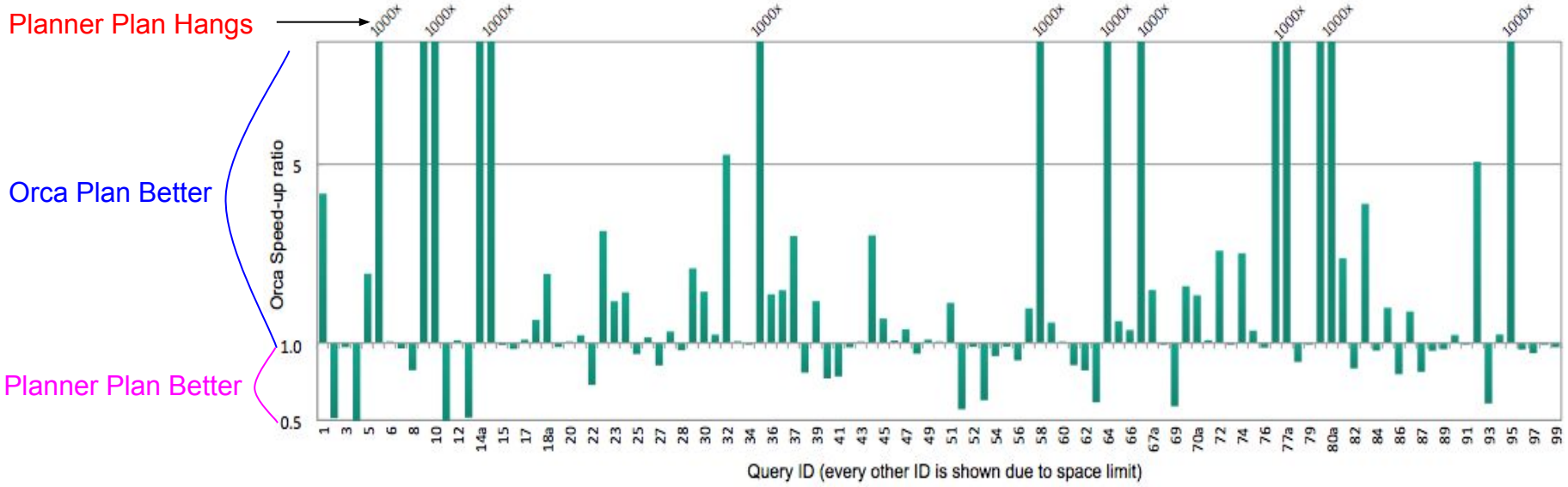
- Orca is not baked into one host system



Key Features

- Smarter partition elimination
- Subquery unnesting
- Optimizing common table expressions (CTE)
- Additional functionality
    - Improved join ordering
    - Join-aggregate reordering
    - Sort order optimization

TPC-DS Orca vs. Planner on HAWQ



TPC-DS 10TB, 16 nodes, 48 GB/node

Pivotal

Subquery Unnesting

Pivotal

Subqueries: Impact

- Heavily used in many workloads
  - BI/reporting tools generate substantial number of subqueries
  - TPC-H workload: 40% of the 22 queries
  - TPC-DS workload: 20% of the 111 queries
- Inefficient plans mean query takes a long time or does not terminate
- Optimizations
  - De-correlation
  - Conversion of subqueries to joins

Subquery Handling: Orca vs. Planner

CSQ Class	Planner	Orca
CSQ in select list	Correlated Execution	Join
CSQ in disjunctive filter	Correlated Execution	Join
Multi-Level CSQ	**No Plan**	Join
CSQ with group by and inequality	Correlated Execution	Join
CSQ must return one row	Correlated Execution	Join
CSQ with correlation in select list	Correlated Execution	Correlated Execution

Pivotal

How to enable Orca on Postgres?

Pivotal

Get Your Hands On It!

- GPORCA: https://github.com/greenplum-db/gporca
- White Paper: http://bit.ly/1ntrE8v
- Pivotal Tracker: http://bit.ly/1m1WGDn

Pivotal

Getting Orca on Postgres

- Step 1: Lift and shift translators already implemented in GPDB
- Step 2: Functions in **`lsyscache.*`** and **`gucs*`** need to moved
- Step 3: Update makefiles (most of this already done in GPDB)

Publications

- Optimization of Common Table Expressions in MPP Database Systems, **VLDB 2015**
  - Amr El-Helw, Venkatesh Raghavan, Mohamed A. Soliman, George C. Caragea, Zhongxian Gu, Michalis Petropoulos
- Orca: A Modular Query Optimizer Architecture for Big Data, **SIGMOD 2014**
  - Mohamed A. Soliman, Lyublena Antova, Venkatesh Raghavan, Amr El-Helw, Zhongxian Gu, Entong Shen, George C. Caragea, Carlos Garcia-Alvarado, Foyzur Rahman, Michalis Petropoulos, Florian Waas, Sivaramakrishnan Narayanan, Konstantinos Krikellas, Rhonda Baldwin
- Optimizing Queries over Partitioned Tables in MPP Systems, **SIGMOD 2014**
  - Lyublena Antova, Amr El-Helw, Mohamed Soliman, Zhongxian Gu, Michalis Petropoulos, Florian Waas
- Reversing Statistics for Scalable Test Databases Generation, **DBTest 2013**
  - Entong Shen, Lyublena Antova
- Total Operator State Recall - Cost-Effective Reuse of Results in Greenplum Database, **ICDE Workshops 2013**
  - George C. Caragea, Carlos Garcia-Alvarado, Michalis Petropoulos, Florian M. Waas
- Testing the Accuracy of Query Optimizers, **DBTest 2012**
  - Zhongxian Gu, Mohamed A. Soliman, Florian M. Waas
- Automatic Capture of Minimal, Portable, and Executable Bug Repros using AMPERe, **DBTest 2012**
  - Lyublena Antova, Konstantinos Krikellas, Florian M. Waas
- Automatic Data Placement in MPP Databases, **ICDE Workshops 2012**
  - Carlos Garcia-Alvarado, Venkatesh Raghavan, Sivaramakrishnan Narayanan, Florian M. Waas

Pivotal

Backup

Pivotal

What Is GP-Orca?

- State-of-the-art query optimization framework designed from scratch to:
  - **Improve** – performance, ease-of-use
  - **Enable** – foundation for future research and development
  - **Connect** – applies to multiple host systems (GPDB, HAWQ, Postgres)

Subqueries: Definition

- A query that is nested inside an outer query block
- Correlated Subquery (CSQ) is a subquery that uses values from the outer query

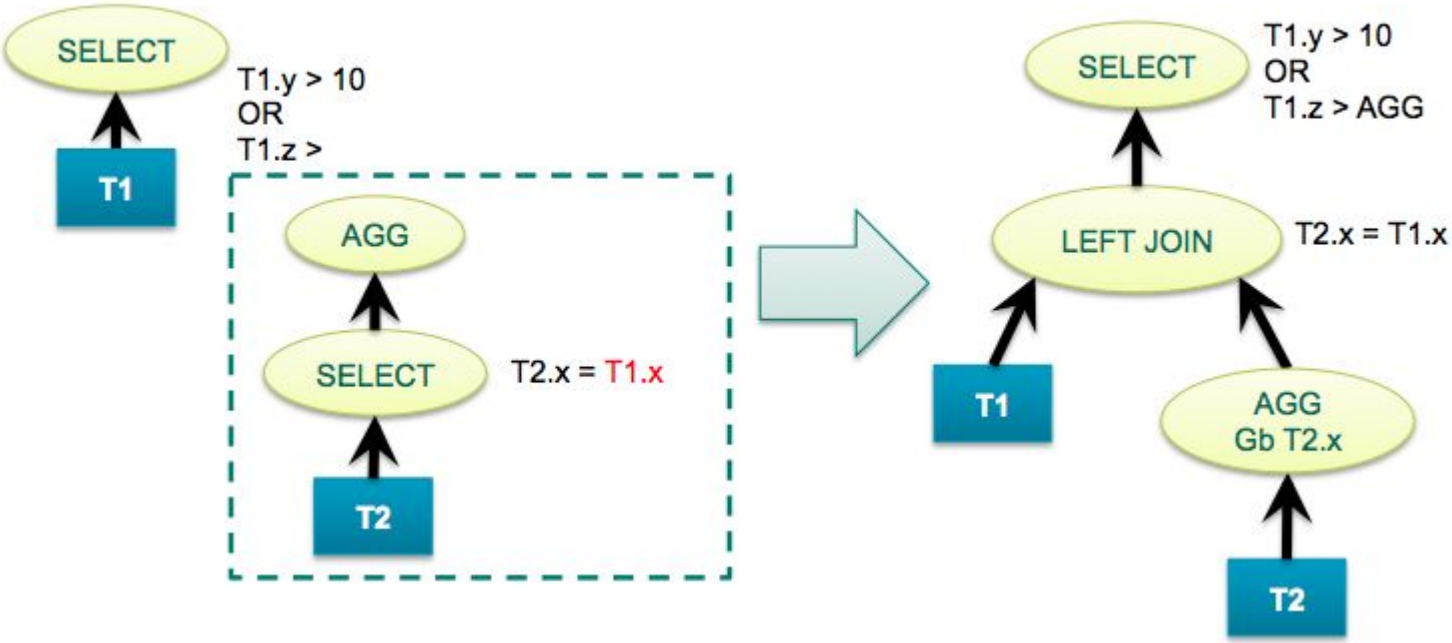
```
SELECT * FROM part p1
WHERE price >
(SELECT avg(price) FROM part p2  WHERE  p2.brand = p1.brand)
```

Subqueries in Disjunctive Filters

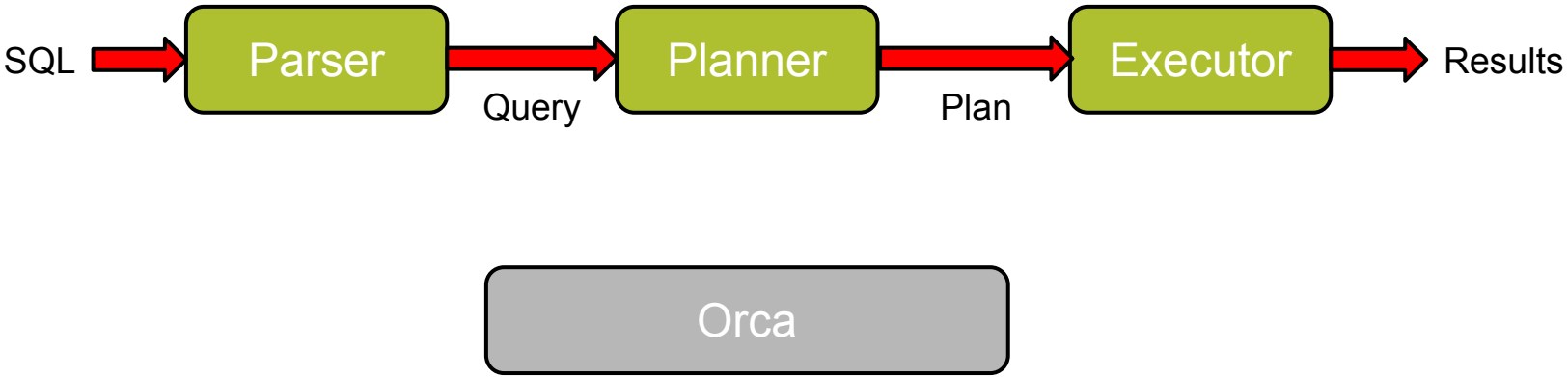
- Find parts with: *size > 40 OR price > the average brand price*

```
SELECT *
FROM part p1
WHERE p_size > 40 OR
p_retailprice >(SELECT avg(p_retailprice)
        FROM part p2
        WHERE p2.p_brand = p1.p_brand)
```

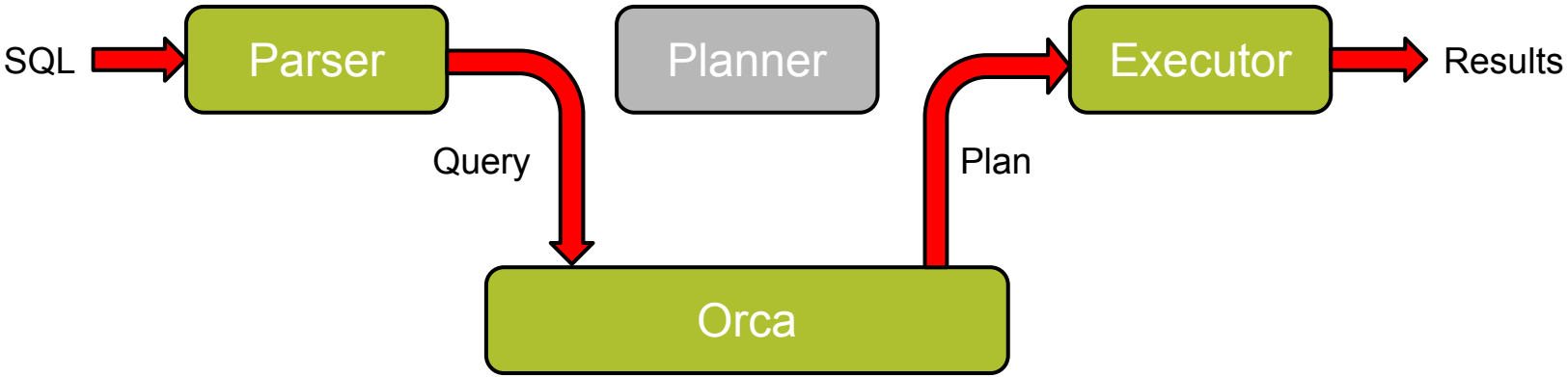
Subqueries in Disjunctive Filters



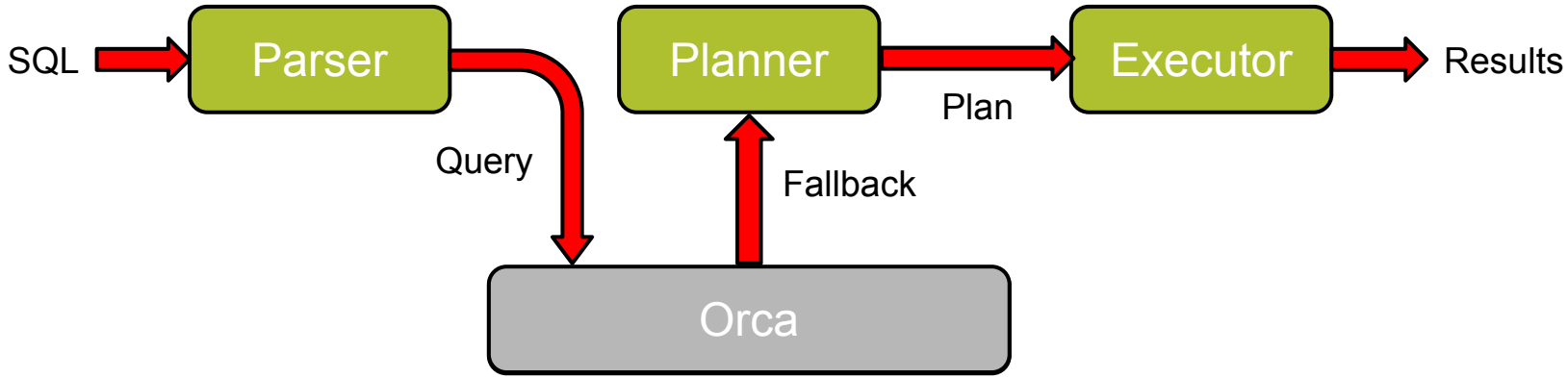
Currently in Apache HAWQ



When Orca is exercised



When Orca fallbacks



Orca will automatically fallback to the legacy optimizer for unsupported features