

ORACLE®



ORACLE®

How Adaptive Plans Work in 12c?

Balavignesh Arumugam
Global Customer Services

ORACLE®

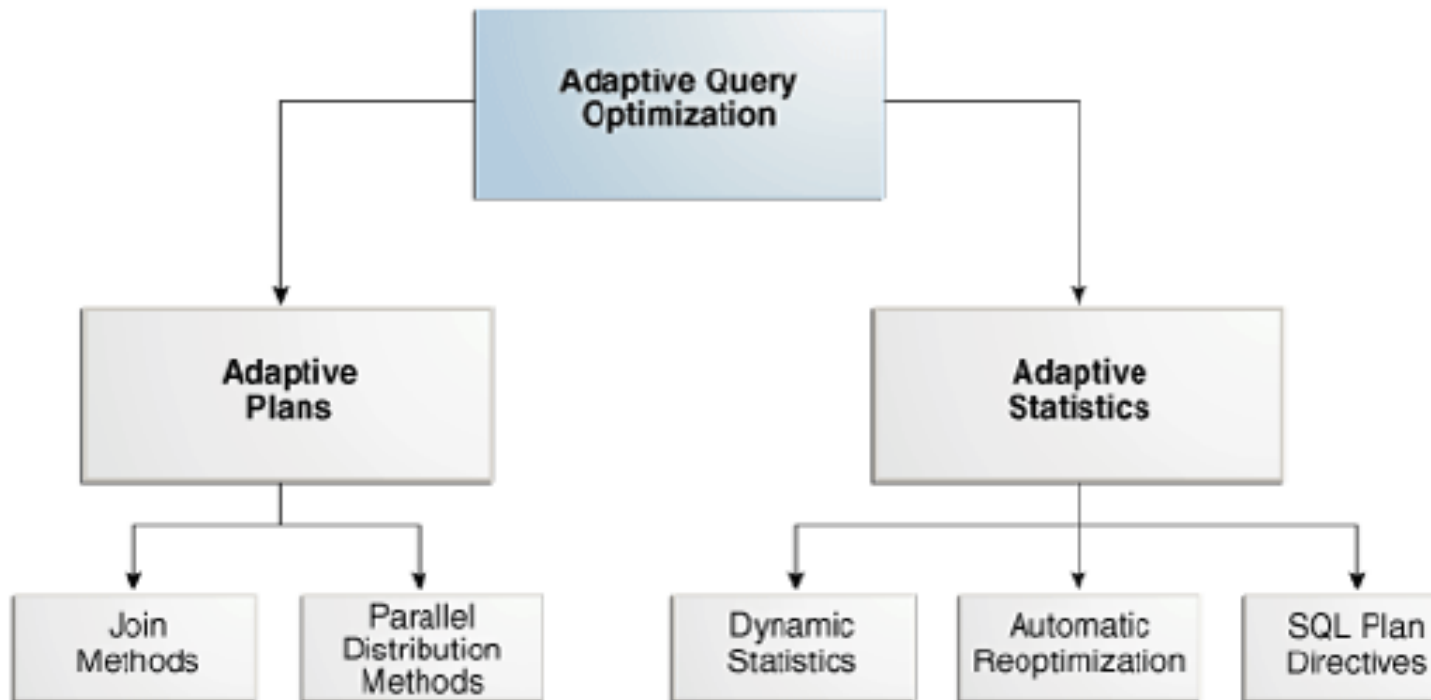
Agenda

- About Adaptive Query Optimization
- Adaptive Plans – What is it?
- Factors Influencing Adaptive plans
- Mechanism: How it works
- Displaying the operations in an adaptive plan
- Sanity checks
- Parameters controlling adaptive plans
- Quick Live Demo
- References

About Adaptive Query Optimization

- A new feature in 12C
- A set of capabilities that enables the optimizer to make run-time adjustments to execution plans
- Discovers more information that leads to better statistics
- Helpful when existing statistics are not sufficient to arrive at an optimal plan

Contd..



Agenda

- About Adaptive Query Optimization
- **Adaptive Plans – What is it?**
- Factors Influencing Adaptive plans
- Mechanism: How it works
- Displaying the operations in an adaptive plan
- Sanity checks
- Parameters controlling adaptive plans
- Quick Live Demo
- References

Adaptive Plans – What is it?

- Adaptive plans enable the optimizer to defer the final plan decision for a statement, until execution time.
- The optimizer instruments its chosen plan (the default plan), with statistics collectors to adapt to a new plan.
- The plan or a portion of it can be automatically adapted to avoid suboptimal performance on the first execution of a SQL statement.

Agenda

- About Adaptive Query Optimization
- Adaptive Plans – What is it?
- **Factors Influencing Adaptive plans**
- Mechanism: How it works
- Displaying the operations in an adaptive plan
- Sanity checks
- Parameters controlling adaptive plans
- Quick Live Demo
- References

Factors Influencing Adaptive plans

- Following components of an execution plan can be adapted automatically:-
 - Join Methods
 - Parallel Distribution Methods

Agenda

- About Adaptive Query Optimization
- Adaptive Plans – What is it?
- Factors Influencing Adaptive plans
- **Mechanism: How it works**
- Displaying the operations in an adaptive plan
- Sanity checks
- Parameters controlling adaptive plans
- Quick Live Demo
- References

Mechanism – How it works?

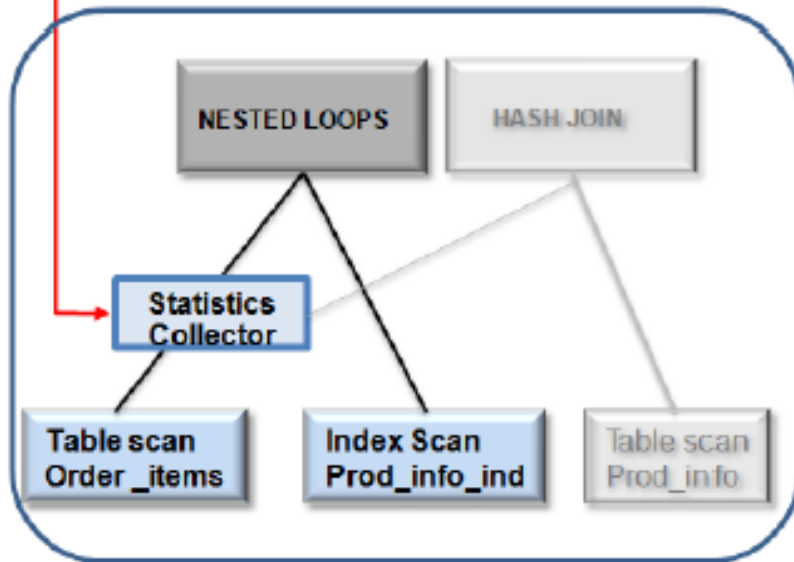
- An adaptive plan contains multiple predetermined **subplans**, and an optimizer statistics collector.
- An **optimizer statistics collector** is a row source inserted into a plan at key points to collect run-time statistics.
- These statistics help the optimizer make a final decision between multiple subplans.
- The optimizer is able to adapt join methods on the fly by predetermining multiple subplans for portions of the plan.

Mechanism – Contd.

- During execution, the **statistics collector** gathers information about the execution and buffers a portion of rows coming into the subplan.
- Based on the information seen in the statistics collector, the optimizer will make the final decision about which subplan to use.
- After the optimizer chooses the final plan, the statistics collector stops collecting statistics.
- Currently the optimizer can switch from a **nested loops join** to a **hash join** and vice versa.

Mechanism – Example

Rows coming out of `order_items` table are buffered up to a point. If row count is less than the threshold use nested Loops otherwise switch to hash join

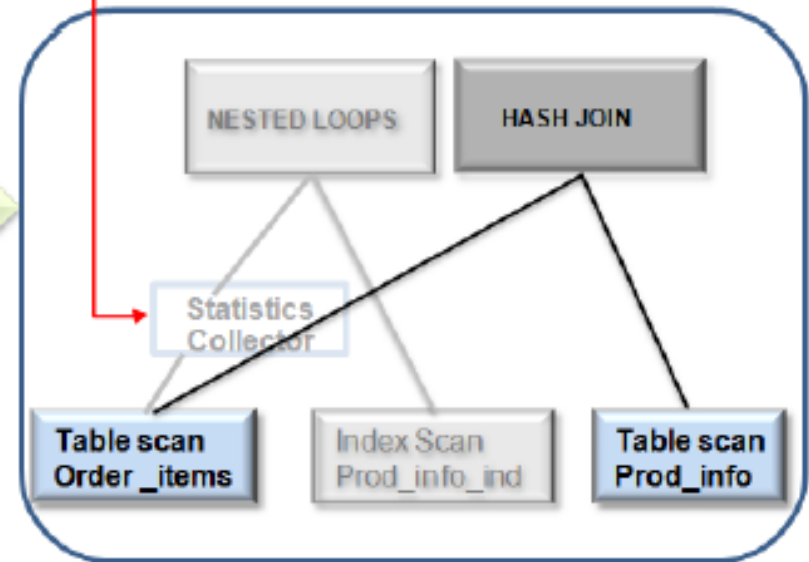


Default Plan is a nested loops join



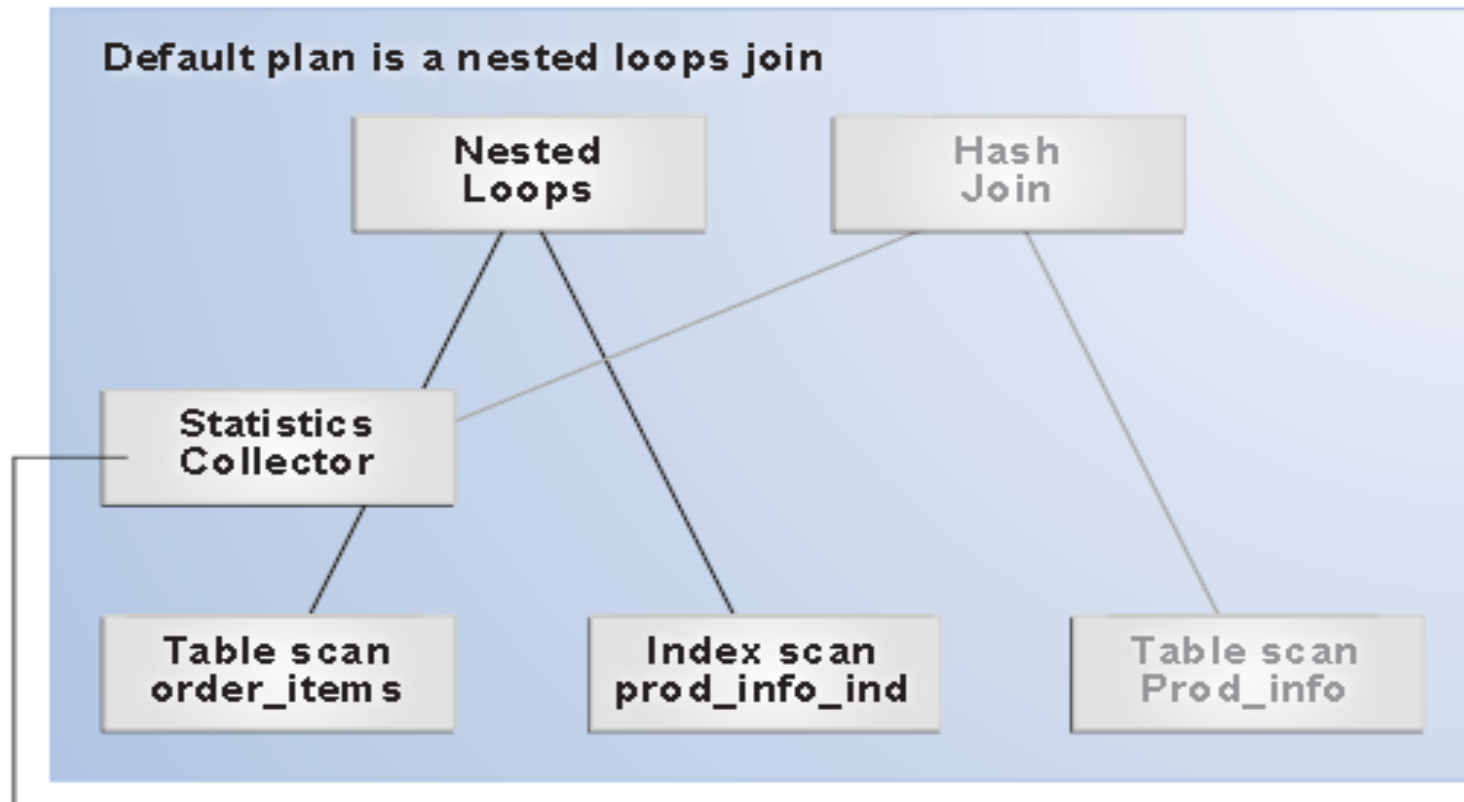
Threshold exceeded
subplan switches

Statistics collector disabled after decision is made and becomes a pass through operation



Final Plan is a hash join

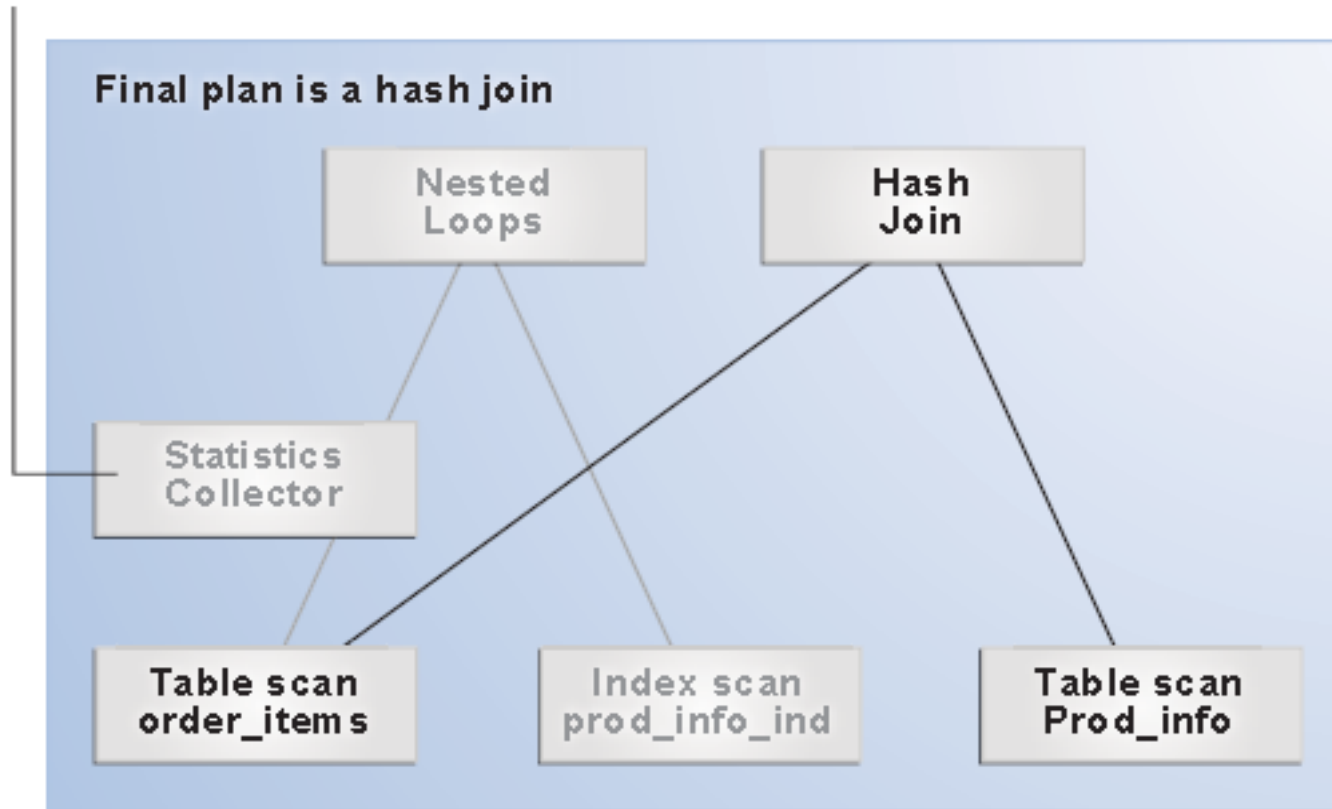
Mechanism – Default plan



The optimizer buffers rows coming from the `order_items` table up to a point. If the row count is less than the threshold, then use a nested loops join. Otherwise, switch to a hash join.

Mechanism – Subplan switch

The optimizer disables the statistics collector after making the decision, and lets the rows pass through.



Agenda

- About Adaptive Query Optimization
- Adaptive Plans – What is it?
- Factors Influencing Adaptive plans
- Mechanism: How it works
- **Displaying the operations in an adaptive plan**
- Sanity checks
- Parameters controlling adaptive plans
- Quick Live Demo
- References

Displaying adaptive plans

- By default, the **explain plan** command will show only the **initial or default plan** chosen by the optimizer.
- The `DBMS_XPLAN.DISPLAY_CURSOR` function displays only the final plan used by the query.

Displaying adaptive plans – Example 1

```
SQL> explain plan for
 2 select /*+ gather_plan_statistics*/ p.product_name
 3 from order_items2 o, product_information p
 4 where o.unit_price = 15
 5    and o.quantity > 1
 6    and p.product_id = o.product_id;

Explained.

SQL>
SQL> select * from table(dbms_xplan.display());

PLAN_TABLE_OUTPUT
-----
Plan hash value: 983807676
-----
| Id | Operation | Name |
-----|-----|-----|
| 0 | SELECT STATEMENT | |
| 1 | NESTED LOOPS | |
| 2 | NESTED LOOPS | |
|* 3 | TABLE ACCESS FULL | ORDER_ITEMS2 |
|* 4 | INDEX UNIQUE SCAN | PRODUCT_INFORMATION_PK |
| 5 | TABLE ACCESS BY INDEX ROWID | PRODUCT_INFORMATION |
-----

Predicate Information (identified by operation id):
-----
 3 - filter("O"."UNIT_PRICE"=15 AND "O"."QUANTITY">1)
 4 - access("P"."PRODUCT_ID"="O"."PRODUCT_ID")

Note
-----
- this is an adaptive plan
```

Displaying adaptive plans – Example 2

```
SQL> select * from table(dbms_xplan.display_cursor());
```

PLAN_TABLE_OUTPUT

```
SQL_ID d3mzkmxzn264d, child number 0
```

```
select /*+ gather_plan_statistics */ p.product_name from order_items2
o, product_information p where o.unit_price = 15 and o.quantity > 1
and p.product_id = o.product_id
```

Plan hash value: 2886494722

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT				7 (100)
1	HASH JOIN		4	128	7 (0)
2	TABLE ACCESS FULL	ORDER_ITEMS2	4	48	3 (0)
3	TABLE ACCESS FULL	PRODUCT_INFORMATION	1	20	1 (0)

Predicate Information (identified by operation id):

```
1 - access("P"."PRODUCT_ID"="0"."PRODUCT_ID")
2 - filter(("0"."UNIT_PRICE"=15 AND "0"."QUANTITY">1))
```

Note

```
- this is an adaptive plan
```

Displaying adaptive plans – Example 3

```
SQL> select * from table(dbms_xplan.display_cursor(format=>'adaptive'));
```

```
PLAN_TABLE_OUTPUT
```

```
SQL_ID d3mzkmzxn264d, child number 0
```

```
select /*+ gather_plan_statistics */ p.product_name from order_items2  
o, product_information p where o.unit_price = 15 and o.quantity > 1  
and p.product_id = o.product_id
```

```
Plan hash value: 2886494722
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		4	128	7 (100)
* 1	HASH JOIN		4	128	7 (0)
- 2	NESTED LOOPS				
- 3	NESTED LOOPS		4	128	7 (0)
- 4	STATISTICS COLLECTOR				
* 5	TABLE ACCESS FULL	ORDER_ITEMS2	4	48	3 (0)
- * 6	INDEX UNIQUE SCAN	PRODUCT_INFORMATION_PK	1		0 (0)
- 7	TABLE ACCESS BY INDEX ROWID	PRODUCT_INFORMATION	1	20	1 (0)
8	TABLE ACCESS FULL	PRODUCT_INFORMATION	1	20	1 (0)

```
Predicate Information (identified by operation id):
```

- 1 - access("P"."PRODUCT_ID"="0"."PRODUCT_ID")
- 5 - filter(("O"."UNIT_PRICE"=15 AND "O"."QUANTITY">1))
- 6 - access("P"."PRODUCT_ID"="0"."PRODUCT_ID")

```
Note
```

```
- this is an adaptive plan (rows marked '-' are inactive)
```

Agenda

- About Adaptive Query Optimization
- Adaptive Plans – What is it?
- Factors Influencing Adaptive plans
- Mechanism: How it works
- Displaying the operations in an adaptive plan
- **Sanity checks**
- Parameters controlling adaptive plans
- Quick Live Demo
- References

Sanity checks on adaptive plan

- A new column has also been added to **V\$SQL.IS_RESOLVED_ADAPTIVE_PLAN** to indicate if a SQL statement has an adaptive plan.
- If **IS_RESOLVED_ADAPTIVE_PLAN** is set to 'Y', it means that the plan was not only adaptive, but the final plan has been selected.
- If **IS_RESOLVED_ADAPTIVE_PLAN** is set to 'N', it indicates the plan selected is adaptive but the final plan has not yet been decided on.
- This column is set to NULL for non-adaptive plans.

Agenda

- About Adaptive Query Optimization
- Adaptive Plans – What is it?
- Factors Influencing Adaptive plans
- Mechanism: How it works
- Displaying the operations in an adaptive plan
- Sanity checks
- **Parameters controlling adaptive plans**
- Quick Live Demo
- References

Parameters controlling Adaptive Plans

- By default adaptive plans are enabled if OFE \geq 12.1.0.1.
- OPTIMIZER_ADAPTIVE_FEATURES = TRUE (DEFAULT)
- _OPTIMIZER_ADAPTIVE_PLANS = TRUE (DEFAULT)
- Both the above parameters have to be enabled for adaptive plans to work.
- If any one of these two are disabled then the feature is disabled.



ORACLE®

D E M O N S T R A T I O N

ORACLE®

Summary – Take Away

- Overview of Adaptive Query Optimization
- How Adaptive Plans Work
- How to get the Adaptive Execution Plan
- New Column Additions to V\$SQL for Adaptive Plans
- Parameters controlling the Adaptive Plans Feature

References

- **Oracle documentation 12c:**
http://docs.oracle.com/database/121/TGSQL/tgsql_optcn_cpt.htm#BABGDDJF
- **Adaptive Execution Plans (Doc ID 1409636.1)**
- <https://www.youtube.com/watch?v=9o9iuxNBciQ>
- <http://www.oracle.com/technetwork/database/bi-datawarehousing/twp-optimizer-with-oracledb-12c-1963236.pdf>

The image features the Oracle logo, which consists of the letters 'O', 'R', and 'A' in a grey, serif font. A large, vibrant red ampersand (&) is superimposed over the 'R' and 'A', partially overlapping the 'O' as well. The ampersand is rendered in a thick, stylized font. In the top-left corner of the slide, there is a solid red square.