# Parallel Execution Pitfalls and Fallacies

## ITOUG Tech Day, 15 April 2021

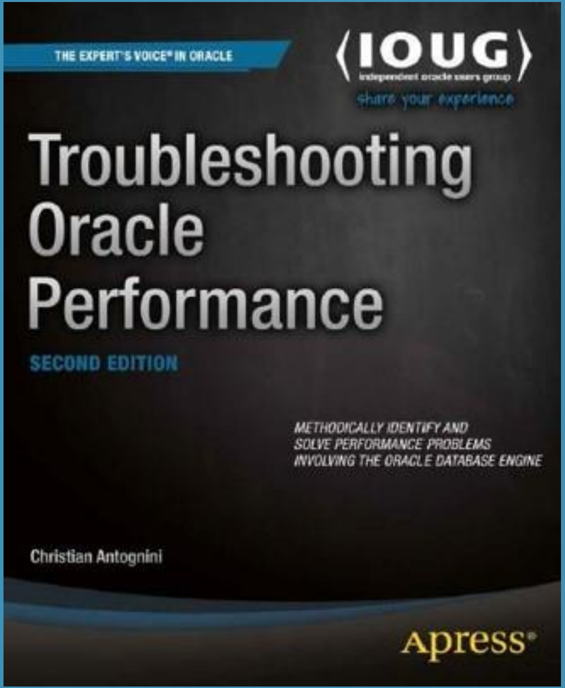
Christian Antognini

BASEL I BERN I BRUGG I BUKAREST I DÜSSELDORF I FRANKFURT A.M. I FREIBURG I.BR. I GENF
HAMBURG I KOPENHAGEN I LAUSANNE I MANNHEIM I MÜNCHEN I STUTTGART I WIEN I ZÜRICH

# Christian Antognini

- Senior principal consultant and partner at Trivadis
- Focus: get the most out of database engines
  - Logical and physical database design
  - Query optimizer
  - Application performance management
- Author of *Troubleshooting Oracle Performance*

@ChrisAntognini antognini.ch


THE EXPERT'S VOICE® IN ORACLE
IOUG independent oracle users group
share your experience
Troubleshooting Oracle Performance
SECOND EDITION
METHODICALLY IDENTIFY AND SOLVE PERFORMANCE PROBLEMS INVOLVING THE ORACLE DATABASE ENGINE
Christian Antognini
Apress®

# Contents

- PARALLEL and PARALLEL_INDEX
- User-Defined Functions
- Parallel DML Statements
- Index Maintenance
- Validation of Constraints

# PARALLEL and PARALLEL_INDEX

# PARALLEL and PARALLEL_INDEX

- They only override the DOP defined at the table and index levels
- They don’t force the utilization of parallel execution
- They don’t enable parallel DML statements

# User-Defined Functions

# User-Defined Functions

- Not all user-defined functions can be evaluated in parallel
- To support parallel execution, a user-defined function must neither write to the database nor read or modify package variables
  - PRAGMA RESTRICT_REFERENCES WNDS, RNPS, WNPS
- User-defined functions that support parallel execution should be decorated with PARALLEL_ENABLE
- In case PARALLEL_ENABLE is incorrectly used, wrong results are expected!

# Parallel DML Statements

# Parallel DML Statements

- The following DML statements can be executed in parallel:
  - DELETE
  - INSERT with a subquery
  - MERGE
  - UPDATE
- INSERT statements with the VALUES clause can’t be parallelized

# Parallel DML Statements – Enabling

- They are disabled by default
- They can be enabled at the session and the SQL statement level
  - ALTER SESSION ENABLE PARALLEL DML
  - ENABLE_PARALLEL_DML
- When executing parallel DML statements, parallel queries should be enabled
  - They are enabled by default

# Parallel DML Statements – LOB Columns

- Parallel INSERT
  - Non-partitioned tables: only SecureFile LOBs are supported
  - Partitioned tables: supported
    - With BasicFile LOBs, only partition granules are supported
- Parallel DELETE, MERGE and UPDATE
  - Table must be partitioned
  - Only partition granules are supported

# Parallel DML Statements – Other Restrictions

- DML statements can’t be executed in parallel when:
  - A trigger is involved
  - Either an ON DELETE CASCADE or deferrable FK is involved
  - The modified table has a FK referencing itself
  - An object column is modified
  - A clustered or temporary table is modified
  - A distributed transaction is involved

# Parallel DML Statements – ORA-12838

- The session executing a parallel DML statement or a direct-path insert can’t access the modified table without committing (or rolling back) the transaction
- SQL statements executed against the modified table before committing (or rolling back) terminate with an ORA-12838
- Because of this limitation, parallel DML statements can’t be used by all batch jobs (a strategy to cope with partial failures is needed)

# Index Maintenance

# Index Maintenance

- It can take place either during the operation that modifies the data or in the INDEX MAINTENANCE row source operation
- It depends on
  - The SQL statement being executed
  - Whether parallel DML is enabled
  - The type of the index

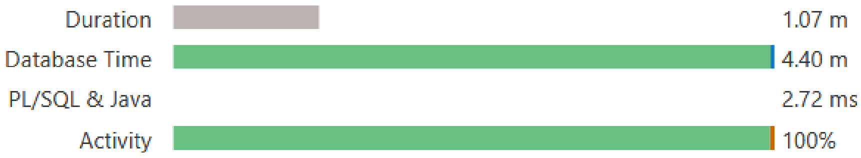
# Index Maintenance – Execution Plan

```
--------------------------------------------------------------------
| Operation                             | Name     |    TQ  |IN-OUT|
--------------------------------------------------------------------
| INSERT STATEMENT                      |          |        |      |
|  PX COORDINATOR                       |          |        |      |
|   PX SEND QC (RANDOM)                 | :TQ10001 |  Q1,01 | P->S |
|    INDEX MAINTENANCE                  | T        |  Q1,01 | PCWP |
|     PX RECEIVE                        |          |  Q1,01 | PCWP |
|      PX SEND RANGE                    | :TQ10000 |  Q1,00 | P->P |
|       LOAD AS SELECT (HYBRID TSM/HWMB)| T        |  Q1,00 | PCWP |
|        OPTIMIZER STATISTICS GATHERING |          |  Q1,00 | PCWP |
|         PX BLOCK ITERATOR             |          |  Q1,00 | PCWC |
|          TABLE ACCESS FULL            | MASTER   |  Q1,00 | PCWP |
--------------------------------------------------------------------
```

# Index Maintenance – Performance Impact

- Index created after the insert (2 min)

```
INSERT INTO t SELECT …
```



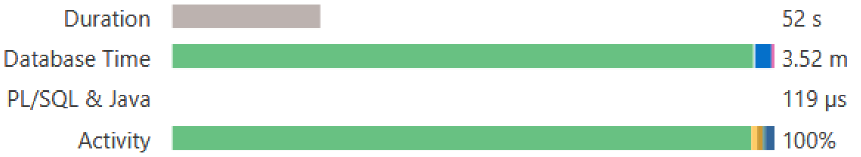
Duration 1.07 m

Database Time 4.40 m

PL/SQL & Java 2.72 ms

Activity 100%

```
CREATE UNIQUE INDEX i ON t (id)
```



Duration 52 s

Database Time 3.52 m

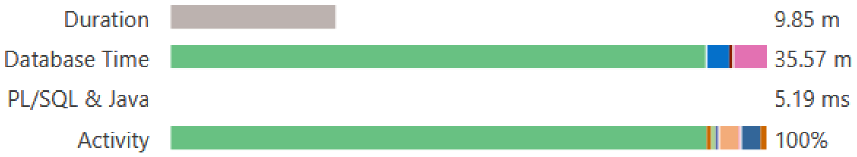
PL/SQL & Java 119 µs

Activity 100%

- Index created before the insert (10 min)

```
CREATE UNIQUE INDEX i ON t (id)
```

```
INSERT INTO t SELECT …
```



Duration 9.85 m

Database Time 35.57 m

PL/SQL & Java 5.19 ms

Activity 100%

# Validation of Constraints

# Primary Key and Unique Constraints

- When they are created, the index supporting them can’t be created in parallel
- To avoid this limitation, create the (unique) index before defining the constraint
- To create an index in parallel without storing the DOP in the data dictionary, use the PARALLEL hint

```
CREATE /*+ parallel */ UNIQUE INDEX i ON t (id)
```

# Foreign Keys and Check Constraints

- When they are created or validated, the data already stored in the table is validated through a recursive query
- The recursive query is executed in parallel when:
  - 12c-18c
    - The table-level DOP is set to a value greater than 1
  - 19c:
    - The table-level DOP is set to a value greater than 1 or the DOP is forced at the session level
    - Parallel queries have to be enabled

# Summary

- Plenty of requirements must be fulfilled to use parallel execution
- To know whether a SQL statement is executed in parallel, verify its execution plan
- Parallel DML statements are disabled by default
- The index maintenance is expensive; if possible, avoid it by creating the indexes after loading the data
- Constraints can be created/validated in parallel

OUR MISSION
TOGETHER
for a smarter
LIFE