Partitioning

Beyond the Basics

Arup Nanda
Starwood Hotels & Resorts
White Plains, NY
Dynamic Bread Artist
What’s It

• Divide and Conquer
• The Best Thing Since Sliced Bread
Partition

- Partition Elimination
- Parallel Execution
- Partition Exchange
- Deletion
- Archiving
Partition Views

View Created as

SELECT some fields
FROM Table1
UNION ALL
SELECT same fields
FROM Table2
Problem

While INSERTING

Row1 → View

{ Table A
  Table B }
Real Partitions

- Came Out with Oracle 8.0
- Range Partitioned Only
- Partition Key Column
  - VALUES LESS THAN (CONSTANT)
Hash Partitions

- Where Range is Not Practical
- Performance
Composite Partitioning

• Partitions are subdivided
• Range-Hash
• Range-List
More Techie

In V$SESSION_WAIT: ‘latch wait’
Latch: “cache buffer chain latch”
More Accesses to a Segment = Need More Latches
Partitioning ➔ More Segments per Object
➔ Less Accesses per Segment
➔ Less “cache buffer chain latch” contention
List Partitioning

Similar to Range but Values are Discrete

PARTITION BY LIST (STATE_CODE)
(
  PARTITION P1 VALUES (‘CT’, ’NY’),
  PARTITION P2 VALUES (’NJ’),
  PARTITION PM VALUES (DEFAULT)
)

*Multi-column not supported*
Local Index

- Partitioning Schemes of Index and Table are Same
- Easy to Administer
- REBUILD part by part
Global Indexes

- Indexes That Span Across All Rows of All Partitions
- Typically Used to Enforce Primary Keys

Oracle 10g: Global Index can be Hash Partitioned
Equi-Partitioning

If a Table and another Table are partitioned in exactly the same way with the same Partitioning Key.

E.g.
SALES on ORDER_DATE
ORDERS on ORDER_DATE
SALES
Subpartitioning

Select from a partition by

```sql
SELECT ... FROM TAB1 PARTITION (P1);
```

Select from a subpartition by

```sql
SELECT ... FROM TAB1 SUBPARTITION (SP1);
```

Works for Insert/Update/Delete, SQL*Loader

```sql
INTO TABLE TAB1 SUBPARTITION (SP1)
```

And Export, too

```sql
TABLE=MYTAB1:SP1
```
Subpartitioning ...

```
DBA_SEGMENTS

SELECT * FROM DBA_SEGMENTS
WHERE TABLE_NAME = 'TAB1'
AND PARTITION_NAME = 'SP1'
/

SEGMENT_TYPE
PARTITION_ID
```
Subpartitions

Subpartition Templates

PARTITION BY RANGE (COL1)
SUBPARTITION BY HASH (COL2)
SUBPARTITION TEMPLATE
   (SUBPARTITION SP1 TABLESPACE T1,
    SUBPARTITION SP2 TABLESPACE T2
   )
   (PARTITION P1 VALUES LESS THAN (101),
    PARTITION P2 VALUES LESS THAN (201),
    ...
    P1_SP1, P1_SP2, P2_SP1, P2_SP2

Can’t Specify Storage (8i)
Can specify Storage (9i) DBA_SUBPARTITION_TEMPLATES
Partition Pruning

Partition Based on Date – 1 partition per quarter

SELECT ... FROM SALES
WHERE ORDER_DATE = '1/1/2003'

Will search only the 2003 Q1 partition
Plan_Table Revisited

Relevant Columns

PARTITION_START
PARTITION_STOP
PARTITION_ID

*The step id that decided the partition start and stop*

FILTER_PREDICATES

*The exact condition used to evaluate partitions*
select * from table ( dbms_xplan.display ( /* plan table name */ 'plan_table', /* statement id */ NULL, /* format */ 'TYPICAL', 'ALL', 'SERIAL' ) )
Partition Wise Joins

Equi-partitioned tables

SALES partitioned on SALES_DATE
REVENUE partitioned on BOOKED_DATE

SELECT ... FROM SALES S, REVENUE R
WHERE S.SALES_DATE = R.BOOKED_DATE

SALES

Partition1
Partition2
Partition3

REVENUE

Partition1
Partition2
Partition3

\*Ptest3a ptest3b
Partitionwise Join

- Hash Partitioned Tables
- Elimination will occur in Equality only, not in range.
Character Values in Range

CREATE TABLE EMPLOYEE (...........)
PARTITION BY RANGE (LAST_NAME)
(
PARTITION P1 VALUES LESS THAN ('D%'),
PARTITION P2 VALUES LESS THAN ('M%'),
PARTITION P3 VALUES LESS THAN ('T%'),
PARTITION PM VALUES LESS THAN (MAXVALUE)
)
partition by range (col1, col2)
(partition p1 values less than (101, 101),
partition p2 values less than (201, 201),
partition pm values less than (maxvalue, maxvalue))
Multi-Column Decision

Consider 1st Column

1st Partition

2nd Partition

Consider 2nd Column

3rd Partition
Deciding Partition Key

- Performance – Elimination/Join, etc.
- Backup
- Other Manageability
Converting to Partitioning

Oracle Recommended (MetaLink Note 1070693.6)

• Create the partitioned table and Insert
• Create Table As Select (CTAS)
• Create Small Tables and Exchange Partitions
• **PROBLEM:** Space Requirement and Time
Alternatives

- Oracle 9i Online Redefinition
  - Small Downtime
  - Space Needed
  - Oracle 9i

- Materialized View Method
  - CREATE MATERIALIZED VIEW MV1 ON PREBUILT TABLE ...
  - DROP MATERIALIZED VIEW
Split-Split Method

- Create partitioned table exactly same as source table with one partition with MAXVALUE
- Exchange source table with this partition
- Split this partition at the lowest boundary
- Split the maximum partition at the next boundary
- Repeat till all partitions are created
Table NOPART

<table>
<thead>
<tr>
<th>COL1</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL2</td>
<td>VARCHAR2(10)</td>
</tr>
<tr>
<td>COL3</td>
<td>CHAR(2)</td>
</tr>
</tbody>
</table>

Index IN_NOPART (COL2)

Constraint CK_NOPART (COL3 IS NOT NULL)

Partitioned BY RANGE (COL1)

| P1 LESS THAN (101) |
| P2 LESS THAN (201) |
| P3 LESS THAN (301) |
| P4 LESS THAN (401) |
| PM … (MAXVALUE) |

Table PART

Table, Index, Constraint Name Change
Summary

• Create table part …
• Exchange partition pMAX with table NOPART
• Split partition pMAX repeatedly
• Drop original table nopart
• Rename part to no part
• Rename index to in_nopart
• Rename constraint to ck_nopart
Pros & Cons

- Space Needs – Less
- Redo Log Generation – Minimized
- Time – Less
- Concurrency – Higher
- Time – Still High
- Concurrency – Still Low
Exchanging Partitions

Main Table

Partition p1
Subpartition sp1
Subpartition sp2
Subpartition sp3

),

...

Source Table

partition p1
partition p2
partition p3
Subpartition Statistics

- **DBMS_STATS.GATHER_TABLE_STATS**
  - tabname => ‘MYTABLE’
  - partname => ‘P1’
  - granularity =>
    - DEFAULT
    - GLOBAL
    - PARTITION
    - SUBPARTITION
    - ALL
# Statistics Collection

<table>
<thead>
<tr>
<th>Granularity</th>
<th>Table Global</th>
<th>Partition Global</th>
<th>Partition Stats</th>
<th>Subpart Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>PARTITION</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>SUBPARTITION</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>ALL</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>
Parallel Index Rebuilding

- One Query Server per Partition
- Slows down processing
- Can’t exploit the Parallel Processing Capabilities
Parallel Index Rebuilding

DBMS_PCLXUTIL.BUILD_PART_INDEX

jobs_per_batch  NUMBER  DEFAULT 1
procs_per_job    NUMBER  DEFAULT 1
tab_name         VARCHAR2
idx_name         VARCHAR2
force_opt        BOOLEAN  DEFAULT FALSE
The Rule Based Optimizer

- Invokes CBO
- Makes Up Statistics
- Don’t Use Partitioning if RBO is Used
Coalesce –vs- Merge

• Coalesce – Hash Partitions
• Merge – Range and List

![Diagram showing the comparison between coalesce and merge](image-url)
Oracle 10g Enhancements

• Partitioned Index Organized Tables (IOT)
  – List Partitioning Possible
  – Bitmap Index Support
  – Global Index Maintenance

• Hash Partitioning of Indexes
  – On Partitioned Tables
  – On Index Organized Tables
  – On Regular Tables
Thank You

Questions?

More Resources:
www.proligence.com