Events (particle collisions) in the ATLAS detector @LHC

Particle collisions are called Events
The one-million table partitions challenge in an ATLAS DB application

Gancho Dimitrov (CERN)
The Event WhiteBoard (EWB) project @ Oracle 18.3

• **EWB concept**: logically groups particle collision Events into collections

• **Collection**: events are processed in Event ranges

• **Collection removal**: once processing of a given collection is finished

• **Lifetime of an EWB collection**: from week(s) to month(s)
The EWB collections of data

<table>
<thead>
<tr>
<th>TABLE object</th>
<th>COLL 2014</th>
<th>COLL 2014</th>
<th>COLL 22567</th>
<th>COLL 26104</th>
<th>COLL 27336</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLL 57310</td>
<td>COLL 58783</td>
<td>COLL 59933</td>
<td>COLL 63382</td>
<td>COLL 63388</td>
<td></td>
</tr>
<tr>
<td>COLL 5385</td>
<td>COLL 63820</td>
<td>COLL 64581</td>
<td>COLL 65002</td>
<td>COLL 65531</td>
<td></td>
</tr>
<tr>
<td>COLL 67763</td>
<td>COLL 77833</td>
<td>COLL 77345</td>
<td>COLL 78741</td>
<td>COLL 78803</td>
<td></td>
</tr>
<tr>
<td>COLL 78804</td>
<td>COLL 78818</td>
<td>COLL 79933</td>
<td>COLL 82233</td>
<td>COLL 82235</td>
<td></td>
</tr>
<tr>
<td>COLL 84143</td>
<td>COLL 86673</td>
<td>COLL 86674</td>
<td>COLL 86676</td>
<td>COLL 88337</td>
<td></td>
</tr>
</tbody>
</table>

-> TABLE partition
The EWB sponge in action

<table>
<thead>
<tr>
<th>TABLE object</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COL1 20113</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 54310</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 53783</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 63870</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 62763</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 78804</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 81438</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL1 20114</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 58378</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 59933</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 63382</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 65002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 47491</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 79939</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL1 22567</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 226101</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 65531</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 82233</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 86673</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 86674</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 86676</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2 88332</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What is the maximum allowed number of partitions in a single table in Oracle RDBMS?

1) 100 thousand
2) 500 thousand
3) 1 million
4) More
### COLLECTIONS
- **COLLECTION ID**
- **COLLECTION NAME**
- **COLLECTION TYPE**
- **COLLECTION STATUS**
- **CREATION TIME**
- **MODIFICATION TIME**
- **... more columns ...**
- **COLLECTION METADATA**

| JSON block in VARCHAR2(32K) |

### COLLECTION_OBJECTS
- **COLLECTION ID**
- **STORAGE SERVER ID**
- **EVENT RANGE MIN ID**
- **EVENT RANGE MAX ID**
- **EVENT RANGE STATUS**
- **... more columns ...**
- **EVENT RANGE METADATA**

| JSON block |
How to partition the EWB “collection_objects” child table?
Design approach “1”
List partition per COLL_ID single value

Idea: each table partition contains data of a single EWB collection. **Removal of any EWB collection data would be straightforward.**
List-type partition for each data collection

```
CREATE TABLE COLLECTION_OBJECTS
(
  COLL_ID NUMBER(10,0),
  ...
CONSTRAINT COLLOBJ_PK PRIMARY KEY (...) using index LOCAL
)
PARTITION BY LIST(COLL_ID) -- AUTOMATIC
(  PARTITION COLLOBJ_ZERO VALUES(0) );
```

• Partition removal is easy:
  ALTER TABLE COLLECTION_OBJECTS DROP partition FOR (5276);

• All worked well,
  but does not seem scalable because of the 1048575 partitions limit per table (error ORA-14299)
Design approach “2”

List partition per sequence of COLL_ID values

Idea: Each List-type table partition to host sequence of data collections (e.g. 10, 20, 50 or more collections per table partition).
Automation in List-type partitions creation (COLL_ID set)

A dedicated List partition per set of collections is created by a **BEFORE INSERT** trigger on the parent table which calls an in-house created PLSQL procedure.
Interesting finding:

- **Achieved flexibility** as # sequential collections per partition can be changed by changing a single value in the "before insert" trigger:
  
  Sequence of 10 collections: created 88485 partitions  
  Sequence of 5 collections: created 32745 partitions

- **After creation of 121230 partitions:**
  
  "Error "ORA-14309: Total count of list values exceeds maximum allowed"

- What is the **maximum number** of list values in the Oracle DB?

  **Count on the existing list partition key values showed:**  
  1048575
Design approach “3”

List automatic partitions on virtual column based on COLL_ID

Idea: List partition on virtual column MOD(COLL_ID, nnn). It guarantees maximum “nnn” partitions on the child table (note: ”nnn” must be smaller than 1 million)

Avoids the max partitions per table limit (ORA-14299) and the number of list-key values limit (ORA-14309).
MOD function returns the remainder of COLL_ID divided by 500000.

The table will have max 500K partitions
“List automatic“ partitions on virtual column

• Test: **500000 partitions were automatically created** using “INSERT INTO collection_objects ...” statement.

• It took **about a week time**.
  Over the time, a partition creation was taking more time.

  Upto 30K partitions: rate of 50-60 partitions/second
  After 70K partitions: rate of 3-4 partitions/second
  After 80K partitions: rate of 3 partitions/second
  After 160K partitions: rate of 1-2 partitions/second
  After 180K partitions: rate of 1 partition/second
  After 200K partitions: rate of 1 partition/second
  Within 200K-400K partitions: rate of 1 partition per 1-2 seconds
  Within 400K-500K partitions: rate of 1 partition per 2 seconds
Best approach out of the explored five paths?

Detailed presentation (50min)
@BGOUG conference
7th-9th June
Borovets resort (Bulgaria)