The importance of having an appropriate relational data segmentation in ATLAS

Gancho DIMITROV (CERN)
On behalf of the ATLAS Collaboration

CHEP 2015
Okinawa, Japan

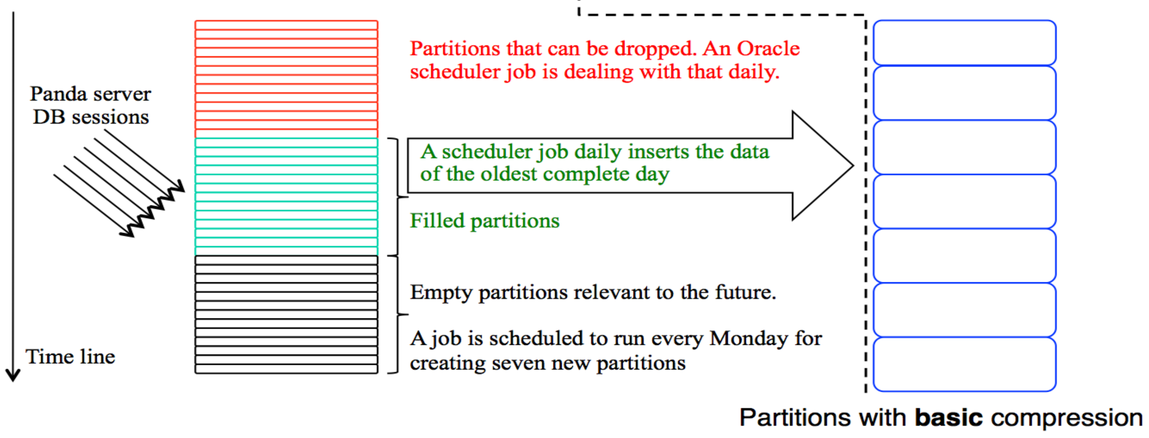
The Oracle Relational Database Management System (RDBMS) has been addressing a majority of the ATLAS database requirements for many years. Much expertise was gained through the years which was used as a good foundation for the next generation applications. Below are presented key applications that use one of the key DB features: data partitioning. Oracle provides **range, list, hash, interval, reference and composite types of partitioning**. In ATLAS we use all of them except the hash option. We use the partitioning mostly because of operations flexibility but also the appropriate data segmentation provides some key advantages in terms of performance.

PanDA

The PanDA system (Production ANd Distributed Analysis) is the ATLAS workload management system for production and user analysis jobs. The PanDA system relies on the Oracle RDBMS since 2008. The DB system has to deal efficiently with two different workloads: transactional (PanDA server) and data warehouse load (PanDA monitor).
Due to the different workloads the PanDA data are logically split into 'operational' (400 GB) and 'archive' (8 TB) types and respectively stored into separate DB schemes.

PanDA data segments organization via range partitioning

'Operational' data	'Archive' data
Tables are range partitioned on a column of type "Date" **Each partition covers a time range of a day.**	Tables are range partitioned on a column of type "Date" **Some tables are with 3 days interval partitions others with monthly partitions**



Key advantages by having range partitioning

- ✓ **Scalability:** the PanDA bulk data copy and deletion is done on table partition level instead on row level
- ✓ **Not IO demanding:** Removing the already copied data is not IO demanding as this is a simple DDL operation over a table segment and its relevant index segments (alter table … drop partition)
- ✓ **Avoids fragmentation** in the PanDA 'operational' table. Much better space utilization and caching in the buffer pool
- ✓ **No need for indices rebuild or coalesce operations** for these partitioned tables.

Applied policies on the PanDA archive data

- ✓ **Table partitions by design are set to reside into dedicated yearly based Oracle tablespaces.**
- ✓ **Activated basic compression** . With this option on, we can fit 50% more rows within a data block of 8KB into the 'jobs' tables and 60% more rows into the "files" table blocks.
- ✓ Sustained data sliding window of 12 months **performing partition exchange operations** from the primary tables to new yearly based tables.
- ✓ The new yearly based tables **can be easily moved to another database cluster** as they are self-contained in dedicated tablespaces.

Rucio

The Rucio project is a new generation Distributed Data Management system for allowing the ATLAS collaboration to manage large volumes of proton-proton collisions, processed or simulated data (currently >150 PB). Data are physically stored in files (files can be grouped in datasets and datasets in containers) on the Grid. In Rucio each file (or copy of it) is assigned to an user, group or ATLAS production activity. **Rucio logical units (scopes) with their metadata for managing ATLAS data on the Grid turned out to fit naturally to the Oracle's data segmentation via the List partitioning feature.**

Composite paritioning in place:

- ✓ Each list partition hosts data of a single Rucio scope, further is sub-partitioned into three pieces 'F'- files, 'D' - datasets, 'C' – containers
- ✓ The number of partitions in the Rucio DB schema depends on the number of registered accounts. Currently there are a bit more than 5000 accounts.
- ✓ As there are defined 5 tables with list partitioning (four are heap organized one of which with sub-partitions and one is Index-organized), the total number of list partitions is 25520, total number of list sub-partitions is 15312 sub-partitions

```
CREATE TABLE DIDS (
    scope VARCHAR2(25),
    did_name VARCHAR2(255),
    account VARCHAR2(25),
    did_type CHAR(1 CHAR),
 ...
    CONSTRAINT "DIDS PK" PRIMARY KEY (scope, name) USING INDEX COMPRESS 1,
    ...
) TABLESPACE &&m_tbs
PARTITION BY LIST(SCOPE)
SUBPARTITION BY LIST(DID_TYPE)
SUBPARTITION TEMPLATE
     (
     SUBPARTITION C VALUES('C'),
     SUBPARTITION D VALUES('D'),
     SUBPARTITION F VALUES('F')
     )
(
PARTITION INITIAL_PARTITION VALUES ('INITIAL_PARTITION')
);
```

PARTITION_NAME	LAST_ANALYZED	NUM_R...	BLOCKS	SAMPLE_SIZE	HIGH_VALUE
MC12_8TEV	22.10.14 19:52:07	203837511	3552784	203837511	'mc12_8TeV'
DATA12_8TEV	22.10.14 18:57:15	85003159	1578011	85003159	'data12_8TeV'
MC12_14TEV	22.10.14 19:34:50	72690471	1275932	72690471	'mc12_14TeV'

efresh: 0

SUBPARTITION_NAME	LAST_ANALYZED	NUM_ROWS	BLOCKS	SAMPLE_SIZE	HIGH_VALUE
MC12_8TEV_C	22.10.14 14:51:38	988899	22354	988899	'C'
MC12_8TEV_D	22.10.14 14:52:10	1443101	50914	1443101	'D'
MC12_8TEV_F	22.10.14 15:08:14	201405511	3479516	201405511	'F'

Key advantages by having list partitioning

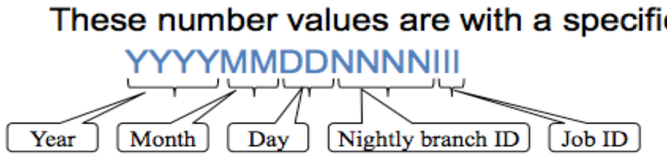
- ✓ Data of each Rucio account is **stored in separate segment** and maintenance operations does not impact the other partitions.
- ✓ Provides advantage on **straightforward MOVE, COALESCE, DROP or REBUILD operations on partition level.**
- ✓ To serve the users free pattern searches, **Oracle reads only the relevant (sub-)partition data chunk.**

ATLAS Nightly Build System

The ATLAS Nightly build system database schema hosts the package tags, nightly jobs timestamps, status information and other important parameters. The data volume is few 10s GB/ year, the application and the availability of the data in the database are considered critical. The retention of the data is agreed to be 12 months. **As most appropriate in many aspects, the reference partitioning technique was chosen. The parent table is configured to use range partitioning.**

A parent table JOBS is created with range partitioning based on a column of type NUMBER (the table's PK, equi-partitioned)

These number values are with a specific format and encoded logic :




```
CREATE TABLE jobs
(
 jid    NUMBER(15,0)    constraint J_JID_NN not null,
 ...
 list of other columns
 ...
 constraint J_PK primary key (jid, nid, relid) USING INDEX LOCAL
) PCTFREE 0
PARTITION BY RANGE (JID)
(
        PARTITION nightly_jobs_Jan2014 VALUES LESS THAN (201402000000000),
        PARTITION nightly_jobs_Feb2014 VALUES LESS THAN (201403000000000),
        PARTITION nightly_jobs_Mar2014 VALUES LESS THAN (201404000000000),
        PARTITION nightly_jobs_Apr2014 VALUES LESS THAN (201405000000000),
        PARTITION nightly_jobs_May2014 VALUES LESS THAN (201406000000000),
        PARTITION nightly_jobs_Jun2014 VALUES LESS THAN (201407000000000),
        PARTITION nightly_jobs_Jul2014 VALUES LESS THAN (201408000000000),
        PARTITION nightly_jobs_Aug2014 VALUES LESS THAN (201409000000000),
        PARTITION nightly_jobs_Sep2014 VALUES LESS THAN (201410000000000),
        PARTITION nightly_jobs_Oct2014 VALUES LESS THAN (201411000000000),
        PARTITION nightly_jobs_Nov2014 VALUES LESS THAN (201412000000000),
        PARTITION nightly_jobs_Dec2014 VALUES LESS THAN (201501000000000),
        PARTITION nightly_jobs_Jan2015 VALUES LESS THAN (201502000000000),
        PARTITION nightly_jobs_Feb2015 VALUES LESS THAN (201503000000000)
```

Each partition covers a time range of a calendar month

	TABLE_NAME	PARTITIONING_TYPE	REF_PTN_CONSTRAINT_NAME	INTERVAL
1	JOBSTAT	RANGE	(null)	(null)
2	TAGS	RANGE	(null)	1000
3	JOBS	RANGE	(null)	(null)
4	TESTRESULTS	REFERENCE	TR_FK	(null)
5	TSTAT	REFERENCE	TS_FK	(null)
6	QARESULTS	REFERENCE	QR_FK	(null)
7	GENRESULTS	REFERENCE	GR_FK	(null)
8	CSTAT	REFERENCE	CS_FK	(null)
9	COMPRESULTS	REFERENCE	CR_FK	(null)

Table with automatic interval partitioning

Parent table named JOBS with its six child tables – all partitioned in uniform way

Partition data segment is created only if the partition is used

TABLE_NAME	PARTITION_NAME	HIGH_VALUE	SEGMENT_CREATED
JOBS	NIGHTLY_JOBS_JUN2014	201407000000000	YES
TSTAT	NIGHTLY_JOBS_JUN2014	(null)	YES
CSTAT	NIGHTLY_JOBS_JUN2014	(null)	YES
QARESULTS	NIGHTLY_JOBS_JUN2014	(null)	NO
GENRESULTS	NIGHTLY_JOBS_JUN2014	(null)	NO
COMPRESULTS	NIGHTLY_JOBS_JUN2014	(null)	YES
TESTRESULTS	NIGHTLY_JOBS_JUN2014	(null)	YES
JOBS	NIGHTLY_JOBS_JUL2014	201408000000000	YES
TSTAT	NIGHTLY_JOBS_JUL2014	(null)	YES
CSTAT	NIGHTLY_JOBS_JUL2014	(null)	YES
QARESULTS	NIGHTLY_JOBS_JUL2014	(null)	NO
GENRESULTS	NIGHTLY_JOBS_JUL2014	(null)	NO
COMPRESULTS	NIGHTLY_JOBS_JUL2014	(null)	YES
TESTRESULTS	NIGHTLY_JOBS_JUL2014	(null)	YES
JOBS	NIGHTLY_JOBS_AUG2014	201409000000000	YES
TSTAT	NIGHTLY_JOBS_AUG2014	(null)	YES
CSTAT	NIGHTLY_JOBS_AUG2014	(null)	YES
QARESULTS	NIGHTLY_JOBS_AUG2014	(null)	NO
GENRESULTS	NIGHTLY_JOBS_AUG2014	(null)	NO
TESTRESULTS	NIGHTLY_JOBS_AUG2014	(null)	YES
COMPRESULTS	NIGHTLY_JOBS_AUG2014	(null)	YES

Advantages of having the reference partitioning:

- ✓ **Tables are partitioned in a uniform way. On event of partition creation in the parent table Oracle automatically creates relevant partitions into the children tables**
- ✓ **Provides flexibility for maintaining any data sliding window. Dropping the parent partition triggers automatic removal of the relevant child partitions.**