Partitioning

HANAExpImp

HANAExpImp can help with export and import of a table to be partitioned according to SAP Note 3568889

- It is a python script to be downloaded from https://github.com/chriselswede/hanaexpimp
- It is intended to be executed as <sid>adm on your SAP HANA Server
- It connects via host, port and DB user, provided in hdbuserstore
- The views must be created before running this script
  - It is assumed that the are named as <view_name>_1, <view_name>_2, ..., <view_name>_<number_views>
- That DB user needs proper privileges
  - For export (-exp true) it needs
    - The system privilege EXPORT
    - The object privilege SELECT on the views
  - For import (-exp false) it needs
    - The system privilege IMPORT
    - The object privilege INSERT on the table

HANAExpImp – input flags

The execution of HANAExpImp is controlled by some input flags

Flag	Unit	Details	Explanation	Default
-k		DB user key	this one has to be maintained in hdbuserstore	'' (must be provided)
-os	true/false	output sql	prints all crucial tasks	false
-op	true/false	output path	full literal path of the folder for the output logs (will be created if not there)	'' (not used)
-es	true/false	execute sql	execute all crucial tasks (useful to turn off for investigation with -os=true a.k.a. chicken mode :)	true
-st	seconds	sleep time	time to sleep between each export and each import	false
-ts		table schema	the schema where the table is located	'' (must be provided)
-tn		table name	the name of the table to be imported back into	'' (must be provided)
-vs		view schema	the schema where the views to be exported are located	'' (must be provided)
-vn		view name	first part of the name of views to be exported, the end of the names must be _1, ..., _<number views>	'' (must be provided)
-vp		view path	the full path to where the views will be exported	'' (must be provided)
-nv		number of views	the number of views to be exported	10
-sv		start view number	the number of the first view to be exported	1
-exp	true/false	export	true --> export, false --> import	true

Partitioning

HANAExpImp – Example (1/3)

While following SAP Note 3568889 to repartition with the export/import method the export could run cause an out of memory issue
Solution: Split the export with multiple views

1. Create a table and check that it has 100 million rows

```
CREATE COLUMN TABLE "PLAYGROUND"."MASS_DATA_PART_DEMO" LIKE "PLAYGROUND"."MASS_DATA" WITH DATA;
select count(*) from "PLAYGROUND"."MASS_DATA_PART_DEMO";
```

100000000

2. Create two views with ~half the rows of the table each:

```
CREATE VIEW VIEWMASS_1 AS SELECT * FROM "PLAYGROUND"."MASS_DATA_PART_DEMO" where COL1 < 20171214;
CREATE VIEW VIEWMASS_2 AS SELECT * FROM "PLAYGROUND"."MASS_DATA_PART_DEMO" where COL1 >= 20171214;
select count(*) from "VIEWMASS_1";
select count(*) from "VIEWMASS_2";
```

54300100
45699900

3. Export both views into .csv files with HANAExpImp

```
chpadm@atgvmls7040:/tmp/HANAExpImp> python hanaexpimp.py -k SYSTEMKEYT1 -ts PLAYGROUND
 -tn MASS_DATA_PART_DEMO -vs SYSTEM -vn VIEWMASS -vp /tmp/HANAExpImp/ -nv 2 -exp true
-st 10 -op /tmp/HANAExpImp/log
Will now export VIEWMASS_1 to /tmp/HANAExpImp/exported_VIEWMASS_1.csv
Number of rows in /tmp/HANAExpImp/exported_VIEWMASS_1.csv is now 54300100
Will now sleep for 10 seconds before exporting VIEWMASS_2
Will now export VIEWMASS_2 to /tmp/HANAExpImp/exported_VIEWMASS_2.csv
Number of rows in /tmp/HANAExpImp/exported_VIEWMASS_2.csv is now 45699900
Total number of exported rows from all views is 100000000
```

HANAExpImp – Example (2/3)

4. Truncate the table and check that it now has 0 rows, also 0 raw rows in all partitions

```
truncate table "PLAYGROUND"."MASS_DATA_PART_DEMO";
select count(*) from "PLAYGROUND"."MASS_DATA_PART_DEMO";
select PART_ID, RECORD_COUNT, RAW_RECORD_COUNT_IN_MAIN, RAW_RECORD_COUNT_IN_DELTA
from SYS.M_CS_TABLES where TABLE_NAME = 'MASS_DATA_PART_DEMO';
```

PART_ID	RECORD_COUNT	RAW_RECORD_COUNT_IN_MAIN	RAW_RECORD_COUNT_IN_DELTA
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0

5. We want to change the partition setup up from HASH on KEY to HASH on QUARTER. Therefore, make sure QUARTER is part of the primary key:

```
ALTER TABLE "PLAYGROUND"."MASS_DATA_PART_DEMO" DROP PRIMARY KEY;
ALTER TABLE "PLAYGROUND"."MASS_DATA_PART_DEMO"
ADD CONSTRAINT PK_MASS_DATA PRIMARY KEY ("KEY", "QUARTER");
```

6. Repartition the (empty) table, and check that all (new) partitions has still 0 rows

```
ALTER TABLE "PLAYGROUND"."MASS_DATA_PART_DEMO" PARTITION BY RANGE (QUARTER) (
  PARTITION VALUE = 1,
  PARTITION VALUE = 2,
  PARTITION VALUE = 3,
  PARTITION VALUE = 4,
  PARTITION OTHERS);
```

```
select PART_ID, RECORD_COUNT, RAW_RECORD_COUNT_IN_MAIN, RAW_RECORD_COUNT_IN_DELTA
from SYS.M_CS_TABLES where TABLE_NAME = 'MASS_DATA_PART_DEMO';
```

PART_ID	RECORD_COUNT	RAW_RECORD_COUNT_IN_MAIN	RAW_RECORD_COUNT_IN_DELTA
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0

Partitioning

HANAExpImp – Example (3/3)

7. Import into the table from the .csv files, one after each other

```
chpadm@atgvmls7040:/tmp/HANAExpImp> python hanaexpimp.py -k SYSTEMKEYT1 -ts PLAYGROUND
 -tn MASS_DATA_PART_DEMO -vs SYSTEM -vn VIEWMASS -vp /tmp/HANAExpImp/ -nv 2 -exp false
 -st 10 -op /tmp/HANAExpImp/log
Will now import all data from VIEWMASS_1 into "PLAYGROUND"."MASS_DATA_PART_DEMO"
Number of rows in "PLAYGROUND"."MASS_DATA_PART_DEMO" is now 54300100
Will now sleep for 10 seconds before importing data from VIEWMASS_2
Will now import all data from VIEWMASS_2 into "PLAYGROUND"."MASS_DATA_PART_DEMO"
Number of rows in "PLAYGROUND"."MASS_DATA_PART_DEMO" is now 100000000
```

8. Check the table

```
select p.table_name, p.part_id, p.range, t.record_count, t.loaded, t.load_unit,
LPAD(TO_DECIMAL((T.MEMORY_SIZE_IN_TOTAL + T.PERSISTENT_MEMORY_SIZE_IN_TOTAL)/1024/1024, 10, 2), 11) MEM_MB
from M_CS_PARTITIONS p inner join M_CS_TABLES t
on p.schema_name = t.schema_name and p.table_name = t.table_name and p.part_id = t.part_id
where p.table_name = 'MASS_DATA_PART_DEMO';
```

TABLE_NAME	PART_ID	RANGE	RECORD_COUNT	LOADED	LOAD_UNIT	MEM_MB
MASS_DATA_PART_DEMO	1	1	25402045	FULL	COLUMN	1034.70
MASS_DATA_PART_DEMO	2	2	19400171	FULL	COLUMN	791.91
MASS_DATA_PART_DEMO	3	3	27599157	FULL	COLUMN	1115.96
MASS_DATA_PART_DEMO	4	4	27598627	FULL	COLUMN	1115.77
MASS_DATA_PART_DEMO	5		0	FULL	COLUMN	0.06

HANAExpImp – Prerequisites

Note: Depending on where the .csv file is located you might have to set following parameter to false before the import

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','SYSTEM')
SET ('import_export','enable_csv_import_path_filter')='false'
WITH RECONFIGURE COMMENT 'to be allowed to import from a .csv
file located in any folder';
```

Note: The HANA user that executes the import needs

- the system privilege IMPORT, and
- the object privilege INSERT on the table