Data Lifecycle Management (DLM) on Exadata

Northern California Oracle Users Group – Summer 2021

Bin Zhang

Sr. MTS, Database Engineering

About this session

- Definition for DLM on Exadata
- Design and Implement
- Observation
- Wrap up: Q & A

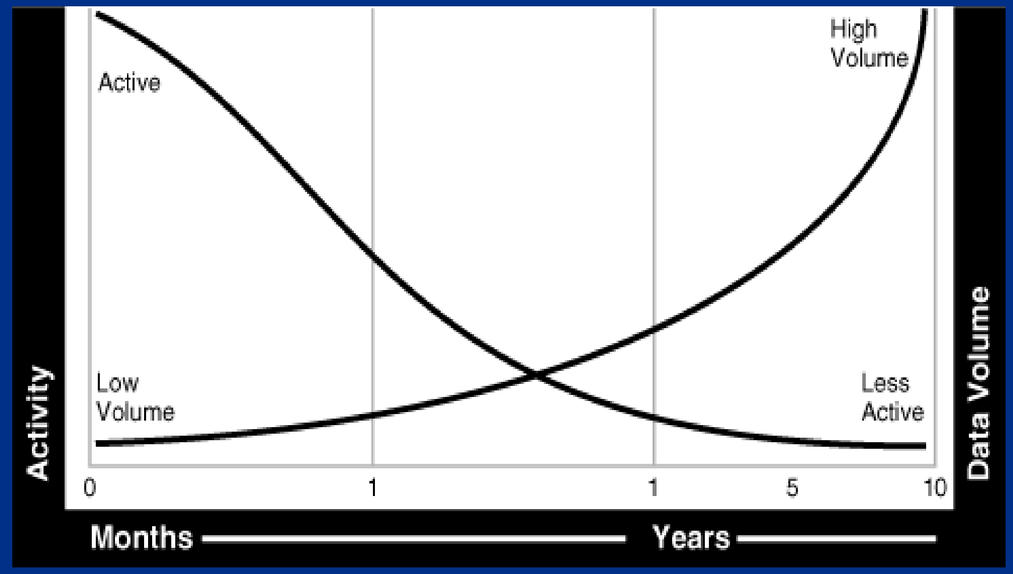
About PayPal

- Providing Simple, Affordable, Secure and Reliable financial services and digital payments
- Mission is to democratize financial services for everyone around the world
- Offers various financial services including new services, Zettle@POS, QR code@POS, BNPL, Crypto trading on PayPal and Venmo platforms
- 400+ million active accounts around 200+ countries and regions
- Growing faster than ever

Definition for Data Lifecycle Management(DLM)

- Most data has lifecycle. Many users are accessing current data while very few users are accessing older data. Data is considered to be : active, less active, historical, or ready to be archived.
- Various regulatory requirements specify how long it must be retained
- Depending on where the data is in its lifecycle, it must be located on the most appropriate storage device.
- Different disk type (Persistent Memory, SSD, HDD) has different price , performance and capacity.
- Cost saving by compress.

1. Active in HOT
2. Less active in DATA with lower compress ratio
3. History in COLD with higher compress ratio



Apply DLM requirements on Exadata

- Exadata flexible storage tier

1. Exadata Extreme Flash (EF) Storage Server, first introduced with Exadata X5, is the foundation of a database optimized all-Flash Exadata Database Machine.
2. High Capacity (HC) Storage Server: Tiered Disk and Flash Deliver Cost of Disk with Performance of Flash. X8-2 server includes twelve 14 TB SAS disk with 168 TB total raw disk capacity and total raw capacity of 25.6 TB of Flash memory. The Flash memory can be used as Flash disks and Flash Cache (Exadata Smart Flash Cache) in front of disk storage
3. Exadata X8-2 Extended (XT) Storage Server. New for Exadata X8, - the Each XT Server includes twelve 14 TB SAS disk drives with 168 TB total raw disk capacity. To achieve a lower cost, Flash is not included, and storage software is optional.
4. HOT on Flash disks ; DATA on SAS disks supported by Smart Flash Cache ; COLD on XT disk.

- Exadata introduce Hybrid Columnar Compression

1. An OLTP application can store historical data in partitions with HCC, while active data remains in partitions with Oracle's OLTP Table Compression. OLTP Table compression typically provides storage savings of 2x - 4x.
2. HCC can continue archiving OLTP compressed table by 2x~8x. (query low & archive high)

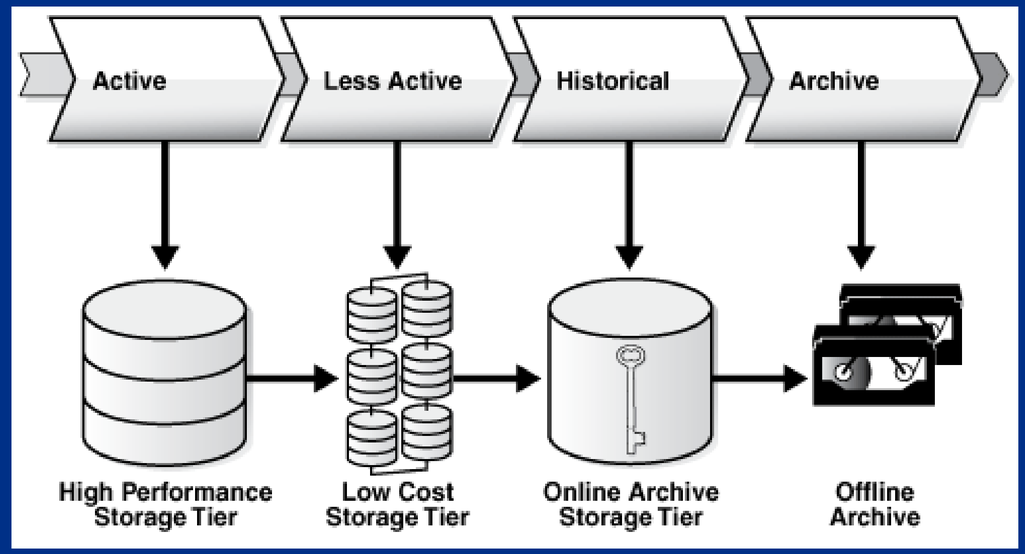
- Online DLM

1. Oracle Database Release 18c and above provides the ability to change the type of compression used by individual table partitions online. 12c already support datafile online move.

Design and Implement

It's just a **Smart** Data Porter

1. Understand data access pattern. Is the partition storing active , inactive or cold data?
2. Assign the data classes (partitions) across the appropriate compress and archive policy and storage.
3. All operations need to be online.
4. Partition is all for VLDB and the basis for DLM
   - range or interval partition
   - Partition pruning from app side
   - no global index



Design and Implement: Understand Data Access Pattern

- Freecon Heat Map

1. Based on v$segstat, we can know how many read/write blocks happened on one segment in a period. stat name is “db block changes”,” logical reads”
2. Giving one table, we can know each partition’s read/write pattern; E.g., cold partition has almost zero block change/read.
3. “Diagnosing Mission-Critical Databases: The Paypal Way” in May 2019 NCOUG

- Data Age

1. Convert partition metadata, include HIGH_VALUE , name of partition to date time range.
2. Query sample data if table includes CREATION_TIME column.

- DBA_tab_modifications

1. Approximate number of inserts/updates since the last time statistics were gathered for each partition/sub-partition.
2. It tells us which partitions have write activity.

Data Access Pattern: Freecon Heatmap

1.How many logical IO Reads and block changes for one partition in past N days

2.Vertical metrics: In past N days, what's percent of read/write happened on each partitions

PARTITION_NAME/POSITION	BC	LR	BC_24_HRS	BC_24_HRS_V	BC_7_DAYS	BC_7_DAYS_V	BC_30_DAYS	BC_90_DAYS	BC_180_DAYS	BC_365_DAYS	LR_24_HRS	LR_24_HRS_V	LR_7_DAYS	LR_7_DAYS_V	LR_30_DAYS	LR_90_DAYS	LR_180_DAYS	LR_365_DAYS
PART_NAME_30 / 30	37280	436769584	0	0	0	0	0	0	0	29.871	.003	.001	.016	.001	.192	.405	2.361	81.886
PART_NAME_31 / 31	37888	229659136	0	0	0	0	0	0	0	41.385	.01	.002	.053	.001	.337	.737	3.604	72.637
PART_NAME_32 / 32	58656	260843152	0	0	0	0	0	0	0	65.112	.018	.003	.182	.005	.493	.918	2.803	74.732
PART_NAME_33 / 33	44400	103211088	0	0	0	0	0	0	0	33.369	.045	.003	.112	.001	.997	2.098	6.937	34.031
PART_NAME_34 / 34	48528	155759376	0	0	0	0	0	0	0	27.662	.026	.003	.053	.001	.555	1.293	11.923	57.507
PART_NAME_35 / 35	50656	138432144	0	0	0	0	0	0	0	30.196	.021	.002	.063	.001	1.133	2.191	6.922	47.034
PART_NAME_36 / 36	132672	138128048	0	0	0	0	0	0	0	10.275	.03	.003	.361	.005	1.261	2.336	9.905	55.099
PART_NAME_37 / 37	50608	127704656	0	0	0	0	0	0	0	27.347	.011	.001	.048	.001	.944	2.124	12.375	51.161
PART_NAME_38 / 38	139120	140524928	0	0	0	0	0	0	0	10.489	.009	.001	.055	.001	1.45	2.333	10.8	40.771
PART_NAME_39 / 39	205936	376295552	0	0	0	0	0	0	0	61.868	.003	.001	.029	.001	.414	.674	2.658	69.76
PART_NAME_40 / 40	57904	147604752	0	0	0	0	0	0	0	31.169	.013	.001	.173	.003	.473	1.474	8.403	38.478
PART_NAME_41 / 41	51808	157943776	0	0	0	0	0	0	0	30.204	.017	.002	.05	.001	.355	1.725	8.845	35.654
PART_NAME_42 / 42	54576	245310608	0	0	0	0	0	0	0	27.47	.015	.003	.044	.001	.262	1.268	6.072	56.038
PART_NAME_43 / 43	55392	543854640	0	0	0	0	0	0	0	29.174	.007	.003	.02	.001	.463	1.023	4.082	84.706
PART_NAME_44 / 44	55296	227064672	0	0	0	0	0	0	0	32.87	.018	.003	.054	.001	1.368	3.063	6.95	60.835
PART_NAME_45 / 45	55408	237193328	0	0	0	0	0	0	0	34.219	.047	.008	.09	.002	1.403	3.337	9.61	59.362
PART_NAME_46 / 46	132976	254754096	0	0	0	0	0	0	0	22.885	.015	.003	.052	.001	2.083	3.735	9.749	63.457
PART_NAME_47 / 47	71264	218259904	0	0	0	0	0	0	0	38.392	.018	.003	.057	.001	1.815	3.429	9.356	54.624
PARTITION_NAME/POSITION	BC	LR	BC_24_HRS	BC_24_HRS_V	BC_7_DAYS	BC_7_DAYS_V	BC_30_DAYS	BC_90_DAYS	BC_180_DAYS	BC_365_DAYS	LR_24_HRS	LR_24_HRS_V	LR_7_DAYS	LR_7_DAYS_V	LR_30_DAYS	LR_90_DAYS	LR_180_DAYS	LR_365_DAYS
PART_NAME_79 / 79	9270976384	2.9601E+10	0	0	0	0	0	0	0	100	.002	.055	.011	.033	.084	1.411	3.826	100
PART_NAME_80 / 80	1752284304	5849210320	0	0	0	0	0	0	0	100	.016	.071	.189	.11	.874	6.791	37.217	100
PART_NAME_81 / 81	8326099168	2.1487E+10	0	0	0	0	0	0	100	100	.006	.101	.03	.065	.182	5.984	100	100
PART_NAME_82 / 82	1.0435E+10	2.7791E+10	0	0	0	0	0	0	100	100	.079	1.619	.695	1.924	1.168	1.368	100	100
PART_NAME_83 / 83	1798299552	5728274368	0	0	0	0	0	0	100	100	.055	.235	.532	.303	.961	4.74	100	100
PART_NAME_84 / 84	7740348112	2.0348E+10	0	0	0	0	0	100	100	100	.261	3.94	3.193	6.471	8.348	100	100	100
PART_NAME_85 / 85	1.1159E+10	3.7322E+10	3.001	100	20.132	100	90.793	100	100	100	3.336	92.364	22.948	85.308	92.73	100	100	100
PART_NAME_86 / 86	0	5424	0	0	0	0	0	0	0	0	15.929	0	48.083	0	81.121	100	100	100
PART_NAME_87 / 87	0	3664	0	0	0	0	0	0	0	0	17.467	0	52.402	0	80.349	100	100	100
PART_NAME_88 / 88	0	5152	0	0	0	0	0	0	0	0	17.702	0	49.689	0	80.124	100	100	100
PART_NAME_90 / 89	0	176	0	0	0	0	0	0	0	0	0	0	0	0	90.909	100	100	100
PART_NAME_91 / 90	0	208	0	0	0	0	0	0	0	0	0	0	0	0	46.154	100	100	100

Data Access Pattern: Data Age for each partition

- Better Partition logic, easier to get data age

```
SQL> select  replace(PARTITION_NAME,'[redacted]','PART_') PARTITION_NAME ,HIGH_VALUE,PARTITION_POSITION,DATA_AGE
  2  from ppdba.DLM_DATA_AGE_TABLE_PART where table_name='[redacted]' and DATA_AGE>sysdate-365 order by PARTITION_POSITION;

PARTITION_NAME                 HIGH_VALUE                     PARTITION_POSITION DATA_AGE
------------------------------ ------------------------------ ------------------ -------------------
PART_M1596652962               1596652962                                    120 2020-08-05 11:42:42
PART_M1599331362               1599331362                                    121 2020-09-05 11:42:42
PART_M1601923362               1601923362                                    122 2020-10-05 11:42:42
PART_M1604605362               1604605362                                    123 2020-11-05 11:42:42
PART_M1607200962               1607200962                                    124 2020-12-05 12:42:42
PART_M1609882962               1609882962                                    125 2021-01-05 13:42:42
PART_M1612564962               1612564962                                    126 2021-02-05 14:42:42
PART_M1614984162               1614984162                                    127 2021-03-05 14:42:42
PART_M1617658962               1617658962                                    128 2021-04-05 14:42:42
PART_M1620247362               1620247362                                    129 2021-05-05 13:42:42
PART_M1622922162               1622922162                                    130 2021-06-05 12:42:42

PARTITION_NAME                 HIGH_VALUE                     PARTITION_POSITION DATA_AGE
------------------------------ ------------------------------ ------------------ -------------------
PART_M1625510562               1625510562                                    131 2021-07-05 11:42:42
PART_M1628188962               1628188962                                    132 2021-08-05 11:42:42
PART_M1630867362               1630867362                                    133 2021-09-05 11:42:42
PART_M1633459362               1633459362                                    134 2021-10-05 11:42:42
PART_M1636141362               1636141362                                    135 2021-11-05 12:42:42
PART_MAX                       MAXVALUE                                      136 2052-01-01 00:00:00

17 rows selected.

SQL> SELECT TO_DATE('19700101','yyyymmdd') + 1596652962/24/3600 from dual;

TO_DATE('19700101',
-------------------
2020-08-05 18:42:42
```

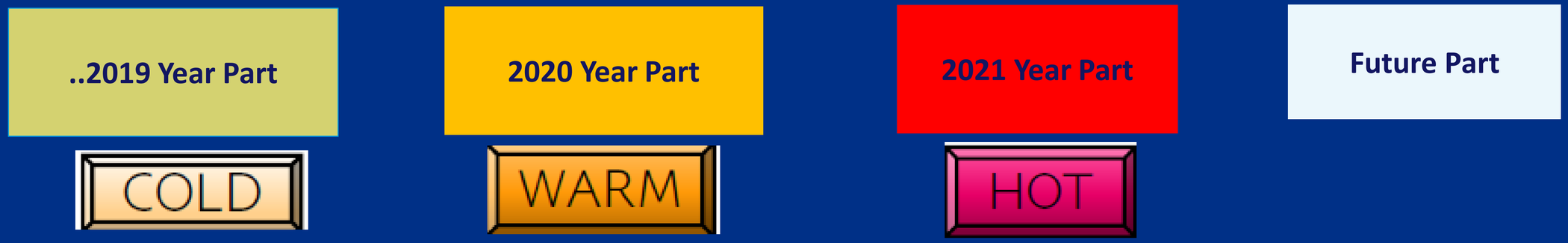
Data Access Pattern: DBA_TAB_Modifications

- Its metrics is real time! It tell which partitions are being written.
- All 3 type of metrics (Heat Map, Data Age) validate each other.

```
17:18:09 SQL> select inserts,updates,deletes,TIMESTAMP from DBA_tab_modifications where table_name='T';

   INSERTS    UPDATES    DELETES TIMESTAMP
---------- ---------- ---------- -------------------
      1000          2          1 2021-07-22 17:18:05

17:18:27 SQL> insert into t select * from t where rownum<=1000;

1000 rows created.

17:18:37 SQL> commit;

Commit complete.

17:18:41 SQL> select inserts,updates,deletes,TIMESTAMP from DBA_tab_modifications where table_name='T';

   INSERTS    UPDATES    DELETES TIMESTAMP
---------- ---------- ---------- -------------------
      2000          2          1 2021-07-22 17:18:05
```

Design and Implement: Assign Data Storage Policy

- Get data access pattern from Heat Map, Data Age



- Assign storage tier and compress policy based on data access pattern

```
1.hcc_archive_high : 'BC_90_DAYS=0 and (LR * LR_30_DAYS/100) < 1000*30 and (LR * LR_7_DAYS/100) <
  1000*7 and (LR * LR_24_HRS/100) < 1000’
2.hcc_query_low     : 'BC_90_DAYS=0 and (LR * LR_30_DAYS/100) < 36000*30 and (LR * LR_7_DAYS/100) <
  36000*7 and (LR * LR_24_HRS/100) < 36000’
3.And Data Age < sysdate – N months
```

- Assign archive destination based on defined policy.

```
SQL> select POLICY_NAME,TABLESPACE_NAME,DG from DLM_TBS_POLICY;

POLICY_NAME                    TABLESPACE_NAME                          DG
------------------------------ -------------------------------- --------
PDB_HCC_DATA                   <PDB>_HCC_DATA                           +DATA
PDB_HCC_DATA_YYYY              <PDB>_HCC_DATA_<YYYY>                    +DATA
PDB_HCC_DATA_Q                 <PDB>_HCC_DATA_<YYYY><Q>                 +DATA
PDB_HCC_DATA_MM                <PDB>_HCC_DATA_<YYYY><MM>                +DATA
```

Design and Implement: Archive Data Online

❑ Online partition/sub-partition move

- Auto-rebuild index online
- There would be high active session on library cache contention if higher QPS
- *Bug 32323277 - TO REDUCE OR CLEAN UP CURSOR INVALIDATION DURING ALTER TABLE MOVE*

❑ Exchange partition/sub-partition

1. CTAS a temporary table to new tablespace with proper archive policy
2. Create index , exp & imp statistics etc.
3. Verify none of DML based on dba_tab_modifications when each DDL finished
4. Cksum temporary table via ora_hash(col) function.
5. Lock partition/sub-partition in shared mode
6. Cksum source partition/sub-partition
7. Exchange

❑ Move datafile online

1. Lock tablespace first. This tablespace will not accept new compressed segments.
2. Move datafiles from High Capacity (HC) Storage Server to Extended (XT) Storage Server.

Observation

1. Log everything for each data archive task. It logs status, timing for each steps, and before/after seg size.
2. Based on task history, we can predicate compress ratio and auto allocate new datafiles.
3. Move table online parallel N will call alter index rebuild parallel N. Index rebuild consumes 2 * N parallel process. Be able to define parallel degree dynamically based on host load and segment size.
4. Small extent size can save more space in some case. Table might have lots of small sub-partitions. Each sub-partition may be just few extents. If archived tablespace extent size is same, compressed segment still need at least 1 extent.
5. During CTAS temporary table, assign an order by clause might save more space. Make sure enough temp tablespace for sort.

Subpartition	Order by	SizeBefore	SizeAfter	Ratio	CTAS Minutes
SYS_SUBP32243		7100M	2720M	38.3%	2.7
SYS_SUBP32524	USER_ID	7100M	1890M	26.6%	2.6
SYS_SUBP32000	TIME_CREATED	7000M	2290M	32.7%	3.21
SYS_SUBP32086	USER_ID ,SOME_FLAG,TIME_ CREATED	7100M	1720M	24.2%	2.3

Observation

1. For exchange partition solution, during CKSUM it requires to block writes to the source partition. It might take a few minutes and might block occasional DML. There's a separate performance monitor module running in background, checking TX/TM lock tree and kill the archive session if necessary.

2. When query sample data to get data age, if table has too many partition/sub-partitions, it may cause too many literal SQL. It could try construct ROWID and query sub-segment by rowid range.

3. Less support for index compression.
   - Don't have option to change index storage option when move table/partition online.
   - Exchange sub-partition require same storage option for index.
   - Rebuild index sub-partition can only change tablespace
   - Modify partition compress will mark sub-partition indexes unusable

Summary

Smart Data Porter

1. Understand data access pattern by freecon heatmap, data age etc.
2. Via predefined policy (heatmap etc.) , assign data class (active, inactive, history) to partition/sub-partition.
3. Apply compress policy to different data class,
4. Apply target disk group to different data class
5. CTAS/move segment online from HOT(FC) to DATA(HC), DATA(HC) to DATA(HC)
6. Move datafiles online from DATA (HC) to COLD (XT)

Thank you.