PayPal

Global Platform & Infrastructure

Faster Database Upgrade

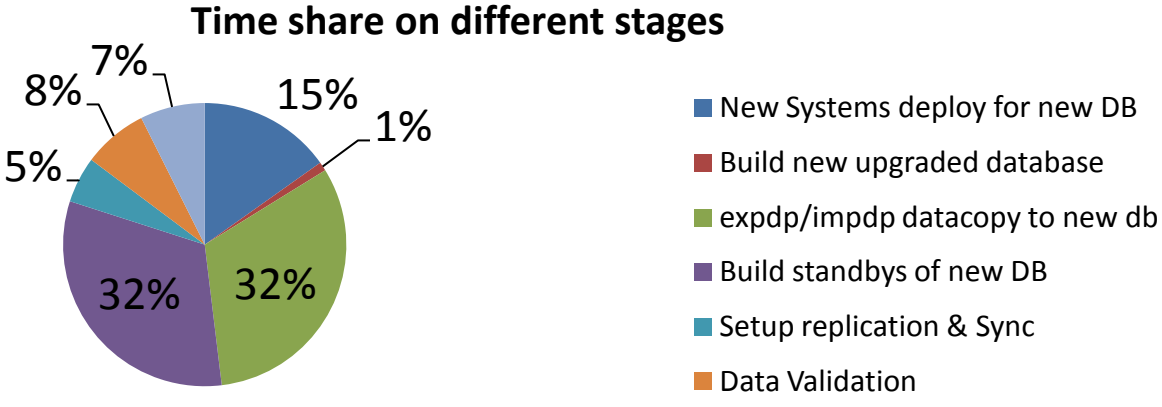
Samrat Roy and Paresh Patel

NoCOUG • November 20, 2015

Agenda

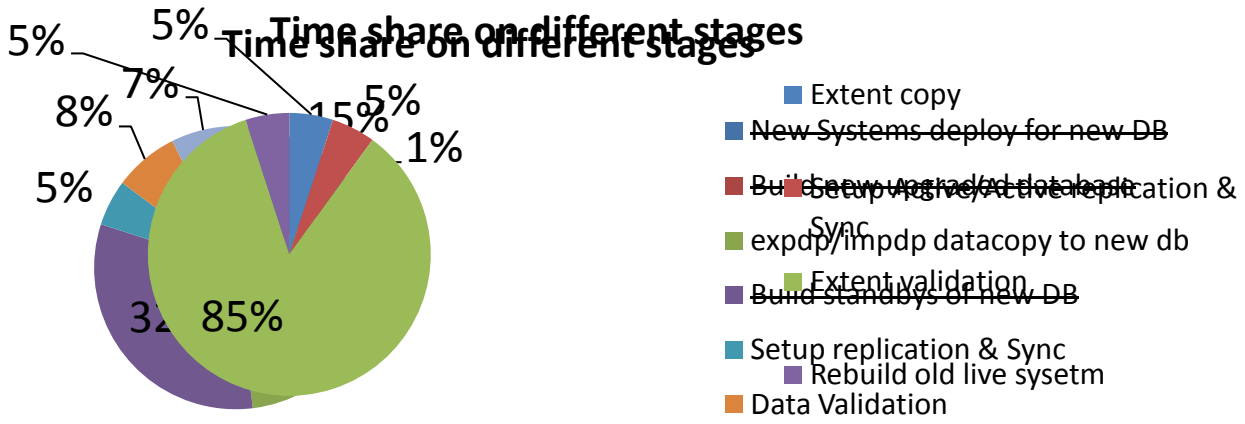
- Traditional DB upgrade Lifecycle
- Faster DB upgrade Lifecycle
  - Step#1 Build target databases
  - Step#2A Configure Goldengate Basics
  - Step#2B Configure Goldengate corner cases
  - Step#2C setup Goldengate
  - Step#3 Data Validation
  - Step#4 Data copy
  - Step#5 Cutover
  - Step#6 Post activities
- Summary
- Q&A

Traditional DB Upgrade Lifecycle



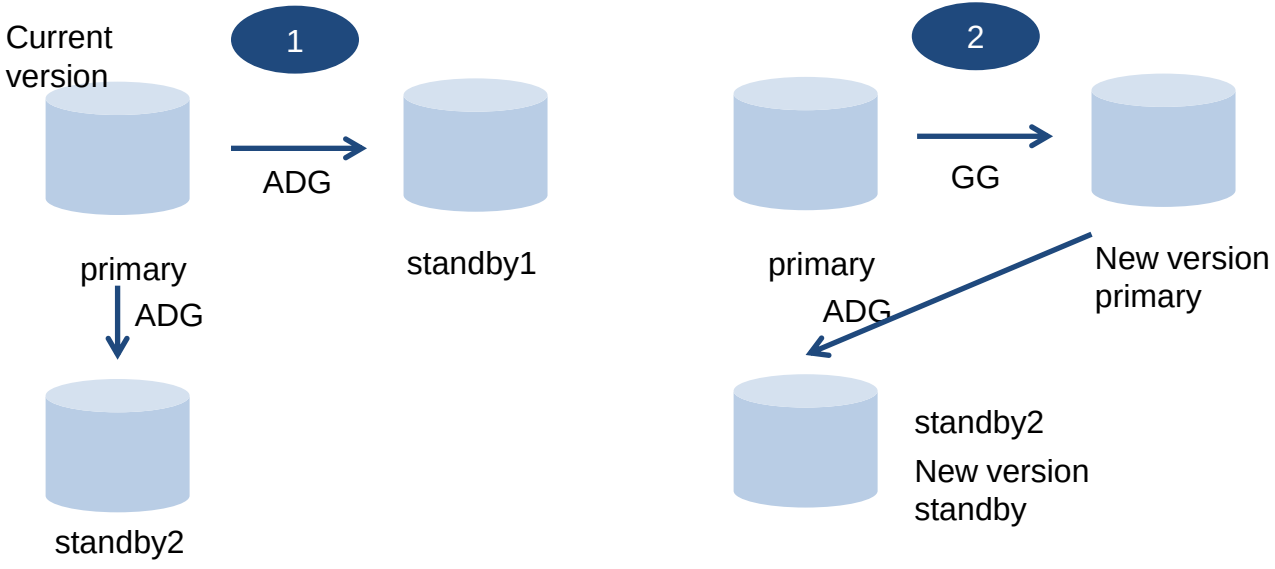
- 60-90 days exercise depending on size of database, cost (systems, storage, DC power/space, backup and replica count, etc.)
- Operational and cost overhead to maintain 2x footprint (Current and To-Be)
- Traditional data copy and validation (of data copied) has limitations for VLDB and LOB segments

Faster DB Upgrade Lifecycle



- No new systems, storage or database deployment required
- 7 days from Start to Cutover and 7 more days to rebuild old live post-cutover
- Independent of database size, reduced dependency on separate teams (network, systems)
- In-house database extent level copy and extent level data validation tool
- Replaces traditional copy tools helps to shorten lifecycle by 90%

Step#1 Build target databases



Cutoff ADG replication between primary and standby1

Reconfigure log shipping for standby2 from standby1

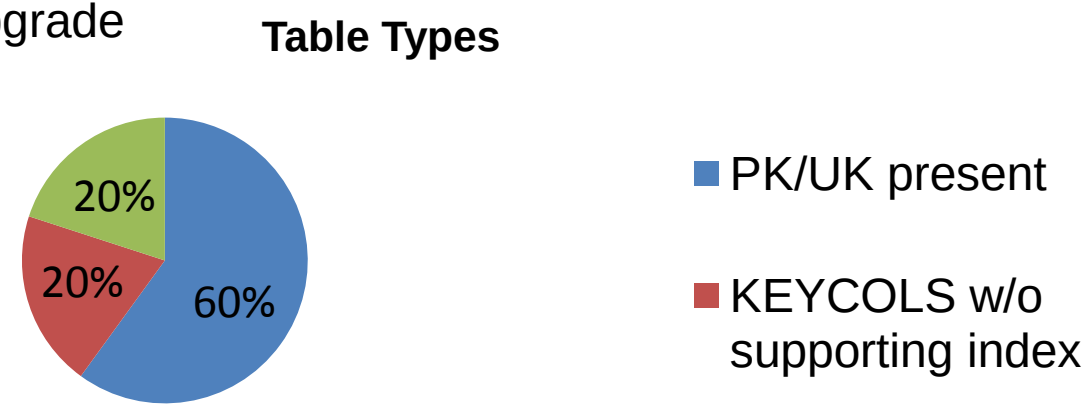
Restart in mount mode standby2 from new oracle home

Activate and upgrade standby1

Open standby2 after upgrade redo apply

Step#2A Configure Goldengate - Basics

- Supplemental logging ALWAYS on UK/PK and KEYCOLS
- Supporting index on KEYCOLS
- Use FILTER column to split high DML tables' replicat into multiple parallel apply
- Use fetch metric from extract report to identify high DML LOB tables. Create dedicated extract in addition to main extract
- Use FLUSHCSECS instead of FLUSHSECS, EOFDELAYCSECS instead of EOFDELAY. Reduce CHECKPOINTSECS to speed redo capture
- Use latest goldengate version to avoid bugs/issues. If needed use separate GG home exclusive for upgrade



Step#2B Configure Goldengate – Corner cases

Problem Statement : “Insert-only” table on source but BLOB column getting written by update on target using KEYCOLS. Table does not have supporting index for KEYCOLS
Workaround : On target ignore INSERTs. Only write UPDATEs after converting them into INSERTs

Problem Statement : Update-able table having functional index only
Workaround : Redirect UPDATEs as INSERT to different table with BEFOREINSERT trigger on it. Trigger merges the data back to original table using existing functional index

Problem Statement : LOB fetches slowing extract
Workaround : Separate LOB tables in separate extract and configure extract to send data to separate stream of trails to target. Configure replicats on target to read from correct trails.

Step#2B Configure Goldengate – Corner cases Continued…

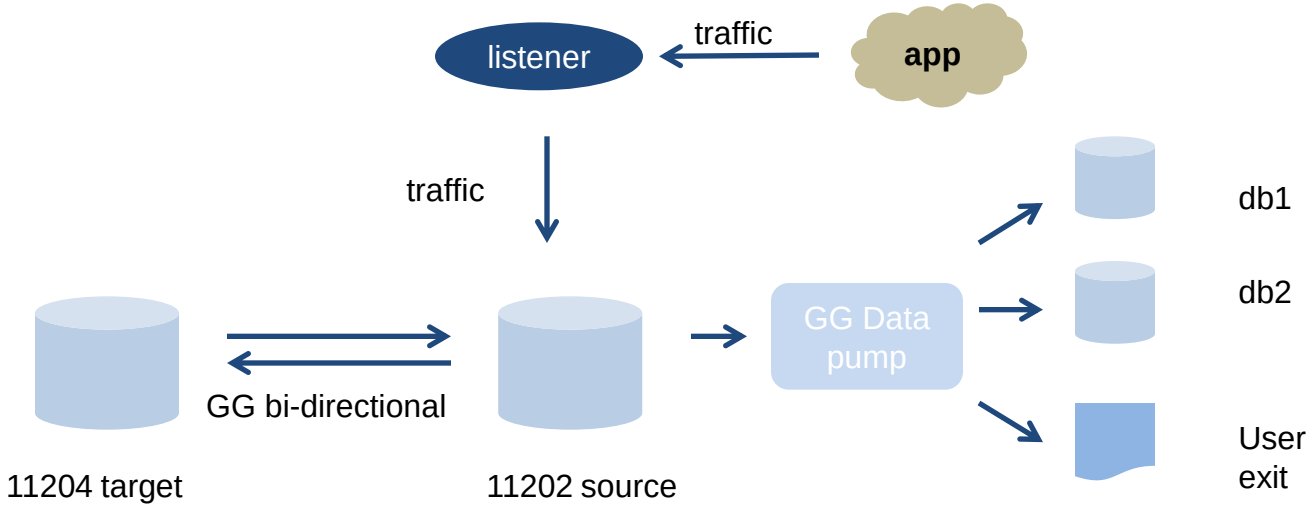
Problem Statement : Filter column in replicate missing supplemental log
Workaround :

- Use other filter column where supplemental logging exists
- Use @coltest and EVENTACTIONS to abend the replicat when the RANGE columns is missing.
- Oracle ER: Request option which will abend replicats if it encounters NULL in filtering column.

Problem Statement : INSERT followed by TRUNCATE of same table/partition
Workaround :

- Use Trigger to capture DDL details in a metadata table on source. On Target use AFTERINSERT Trigger on metadata table to perform TRUNCATE
- Use DDL replication

Step#2C Setup Goldengate



- To avoid conflicts with existing goldengate user, create exclusive namespace by creating new GG user, say “GGC”
- On both source and target, exclude changes made by user “GGC” to be extracted to avoid replication cycle
- At any time, between source and target db only one is active and other is passive

Step#3 Data Validation

- Bring one of Source DB's physical standby and target database to the same logical consistent point to perform custom data validation
- Use ORA_HASH function on non-LOB columns to compare and detect any data corruption issues

```
select /*+ full(c1) parallel(c1, 20) */ 'TESTDBA.VALIDATION_TABLE' lbl, 'CNT:'||count(*) cnt,
nvl(sum(ora_hash(col1)),0)+
nvl(sum(ora_hash(col2)),0)+
nvl(sum(ora_hash(col3)),0)+
nvl(sum(ora_hash(col3)),0) hashsum
from TESTDBA.VALIDATION_TABLE  c1 where 1=1;
```

- Use Oracle supplied DBMS_LOB or DBMS_CRYPTO package to detect any LOB corruption issues

```
select /*+ full(c1) parallel(c1, 20) */ count(1), 'TESTDBA. LOB_TABLE' tbl,
nvl(sum(ora_hash(dbms_crypto.hash(nvl(' || c1.lob_column_1 ||', ''a''), 1))),0)
from TESTDBA.LOB_TABLE c1;
select /*+ full(c1) parallel(c1, 20) */ count(1), 'TESTDBA. LOB_TABLE' tbl,
nvl(sum(ora_hash(dbms_lob.getlength(' || c1.column_name ||' ))),0)
from TESTDBA.LOB_TABLE c1;
```

- Perform at least 3 rounds of data validation

Step#4 Data copy

- Root cause and fix data corruption for any issues found during Data validation
  - Check replicates discard files
  - Check supplemental logging on KEYCOLS and FILTER column(s)
  - Check supplemental logging on tables
  - Check Partition maintenance if present
  - Use replicat/extract stats to check operations on table
- Data reseeding required if corruption found
  - Backup statistics of related objects and truncate table(s)
  - Drop indexes to speed up data copy
  - Create scratch tablespace with NOLOGGING to avoid generating excessive amount of redo
  - Stop replicates related to corrupted table(s)
  - Cancel recovery on source physical database to copy data

Step#4 Data copy

Continue…

- Populate extent mapping with begin and end rowid range to copy one extent at a time and merge

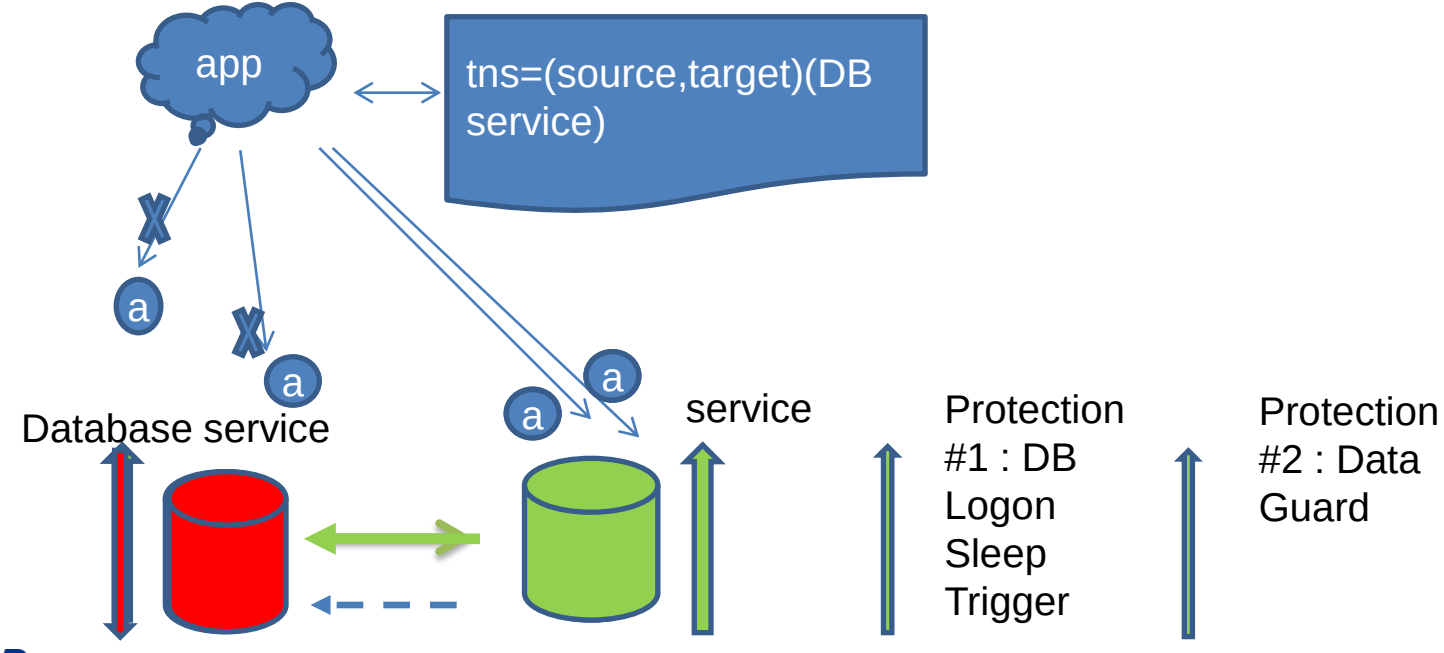
```
select extent_id, owner, segment_name table_name, partition_name,
segment_type, block_id, blocks, bytes/1024/1024 mbytes, relative_fno, data_object_id,
dbms_rowid.ROWID_CREATE (
/* rowid_type */       1,
/* object_number*/     data_object_id,
/* relative_fno*/      relative_fno,
/* block_number*/      block_id,
/* row_number*/        0) start_rowid,
dbms_rowid.ROWID_CREATE (
/* rowid_type */       1,
/* object_number*/     data_object_id,
/* relative_fno*/      relative_fno,
/* block_number*/      block_id+blocks-1,
/* row_number*/        10000) end_rowid
from dba_extents e, dba_objects o
e.owner=o.owner and segment_name=object_name and
object_type = 'TABLE';
```

- Rebuild indexes and fix statistics
- Start replicates from source physical standby database SCN
- Drop scratch tablespace

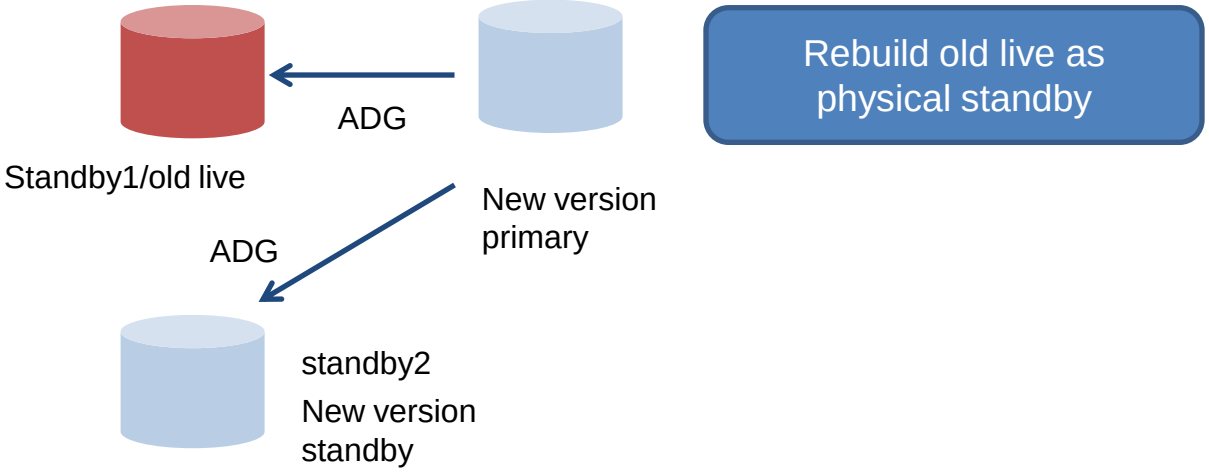
Step#5 Cutover

Continue…

a) New approach to site traffic management
b) Leverage useful features of newer version of Oracle Goldengate replication software like faster replication speed and bi-directional Active/Passive Setup



Step#6 Rebuild old live



Q & A