DATABASES ON ALL-FLASH ARRAYS

How it changes the life of the DBA

April 2018
Istvan Stahl
Gabor Akots
1. Database refresh challenges
2. Database refresh requirements
3. Available options
4. Traditional Block Storage Arrays vs Pure Storage
5. Benefits of using Pure storage refreshes
6. How it works
7. Lessons learned
8. Q/A
DBA CHALLENGES

• Refresh non-Production environments with Production data fast
• The amount of data to be managed is exploding
• Serve Agile development requirements
DEV/UAT/QA DATABASE REFRESH CHALLENGES

Database refresh speed

<table>
<thead>
<tr>
<th>Database Size (GB)</th>
<th>Time (Minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2010</td>
</tr>
<tr>
<td>200</td>
<td>2013</td>
</tr>
<tr>
<td>300</td>
<td>2016</td>
</tr>
</tbody>
</table>

Technology used
Manual DB refresh using exp/imp

Customer satisfaction
DEV/UAT/QA DATABASE REFRESH CHALLENGES

Database refresh speed

Time (Minutes)

Database Size (GB)

Duration (minutes)

Technology used

Customer satisfaction

Self Service Portal leveraging expdp/impdp

Very Good
DEV/UAT/QA DATABASE REFRESH CHALLENGES

Database refresh speed

<table>
<thead>
<tr>
<th>Database Size (GB)</th>
<th>Time (Minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2010</td>
</tr>
<tr>
<td>300</td>
<td>2013</td>
</tr>
<tr>
<td>500</td>
<td>2016</td>
</tr>
</tbody>
</table>

Technology used

Self Service Portal leveraging expdp/impdp
(getting slower)

Customer satisfaction

![Sad emoji]
DEV/UAT/QA DATABASE REFRESH CHALLENGES

Database refresh speed

<table>
<thead>
<tr>
<th>Time (Minutes)</th>
<th>Database Size (GB)</th>
<th>Duration (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100</td>
<td>2010</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
<td>2013</td>
</tr>
<tr>
<td>300</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>700</td>
<td></td>
</tr>
</tbody>
</table>

Technology used

| Manual DB refresh using expdp/impdp |

Customer satisfaction

**Idea** do not validate the constraints during import

**Result** optimizer behaviour changes
# DEV/UAT/QA DATABASE REFRESH CHALLENGES

## Database refresh speed

<table>
<thead>
<tr>
<th>Database Size (GB)</th>
<th>Time (Minutes)</th>
<th>Database refresh speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100</td>
<td>2010</td>
</tr>
<tr>
<td>300</td>
<td>200</td>
<td>2013</td>
</tr>
<tr>
<td>500</td>
<td>400</td>
<td>2016</td>
</tr>
</tbody>
</table>

### Technology used

<table>
<thead>
<tr>
<th>Technology used</th>
<th>Customer satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self Service Portal leveraging expdp/impdp</td>
<td>(this is very slow)</td>
</tr>
</tbody>
</table>

**ISSUE:** Datapump eliminates row chaining
DEV/UAT/QA DATABASE REFRESH CHALLENGES

Database refresh speed

Time (Minutes)

Database Size (GB)

Technology used

Customer satisfaction

RMAN Restore

Very Good
DEV/UAT/QA DATABASE REFRESH CHALLENGES

Database refresh speed

Technology used | Customer satisfaction
---|---
RMAN restore (things are slow again) | 😞
DEV/UAT/QA DATABASE REFRESH CHALLENGES

Database refresh speed

Time (Minutes)

Database Size (GB)

Technology used

Customer satisfaction

Introducing All-Flash Storage snapshots
DATABASE REFRESH OPTIONS

- **Oracle**
  - Multitenant 12.1 Snapshot Clones (Oracle CloudFS aka ACFS)
  - **12.2 New Features:** Hot PDB Clones, Refreshable PDB
  - dNFS Clonedb
  - Enterprise Manager Snap Clone

- **Third party**

*Do it yourself leveraging***

---

**NETAPP SnapManager**

**DELPHIX**

**XTREMIO**

**COHERENT**

**ACTIFIO**

**RUBRIK**

---

**Do it yourself leveraging**

**PURESTORAGE!!!**
PURE VS TRADITIONAL BLOCK STORAGE ARRAYS

- Intelligent features: Deduplication, compression
- Total capacity of the array
- Complexity, manageability (single lun diskgroups)
- Latency below 1 ms
- Replication: sync replication was not available in Pure when we started
- Upgrade path: Evergreen vs. Buy a new array every 3rd year
- Density
- Power consumption
# BENEFITS OF USING PURE STORAGE REFRESHES

<table>
<thead>
<tr>
<th></th>
<th>Traditional Block Storage</th>
<th>Pure Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of the Primary database</td>
<td>5 TB</td>
<td>1.2 TB</td>
</tr>
<tr>
<td>Number of Dev/QA/UAT environments</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Total space consumption</td>
<td>200 TB</td>
<td>10TB</td>
</tr>
<tr>
<td>Total cost of Storage (Capex &amp; Opex)</td>
<td>$x USD</td>
<td>10%</td>
</tr>
<tr>
<td>Data Reduction</td>
<td>1:1</td>
<td>20:1</td>
</tr>
<tr>
<td>Environment refresh time (one environment)</td>
<td>6 hours</td>
<td>2 minutes</td>
</tr>
<tr>
<td>Annual Business time saving (baseline: Traditional)</td>
<td>0</td>
<td>6000 hours</td>
</tr>
<tr>
<td>Annual DBA Ops time saving (baseline: Traditional)</td>
<td>0</td>
<td>2000 hours</td>
</tr>
<tr>
<td>Annual data transfer over the network</td>
<td>5 PB</td>
<td>0 PB</td>
</tr>
</tbody>
</table>
HOW IT WORKS

**Production site**

- **PROD Database**
  - **Pure CloneDB**
    - Pure Snap attached over FC
  - **Snapshot of the CloneDB**

**DR site / Non-Production**

- **DR Database**
  - **Pure CloneDB**
    - Pure Snap attached over FC
  - **Snapshot of the CloneDB**

DataGuard connections between PROD and DR sites.
HOW IT WORKS

1. Cloning process starts on the target Host
2. Get disk details of the Source Diskgroup (ssh)
3. Take a snapshot of the disks on Pure (https) *
4. Copy the snaps to the new target volumes (https)
5. Connect the volumes to the Target Host (https)
6. Rescan the scsi bus on Target Host
7. Setup proper device permissions for ASM
8. Rename Source DG to TargetDG
9. Mount Target Diskgroup
10. Clone CloneDB to TargetDB (TargetDB will be a Primary database)
11. Post refresh actions (eg.: roll forward, database upgrade)

* - we take consistent snaps of a dataguard replica always
LESSONS LEARNED

• Pure does deduplication in multiple rounds

• Databases prior 12.1 support only 2TB disks, even when the ASM layer supports larger disks (chance for data corruption)

• The Application might not benefit from All-Flash
  – When it uses the default pre-fetch size
  – When all data sits in memory and the processing is not I/O heavy

• TDE Encryption kills storage level Compression, Deduplication still helps a lot

• Pure Storage is really simple (even a DBA can understand 😊), fast and its Rest API rocks
Q / A