Performance Diagnosis

Sachin Jain.
APD Team Lead, Netezza

Performance Issues

Step 1: Initial Information Gathering

- Understand nature of performance slowness.

- Following are some sample questions to ask:
  - Is the system slow all the time or slow at specific hours?
  - Any specific queries / jobs?
  - If system is show, what % degradation you see?

- This step generally does not take that long, but don't rush through it. The biggest mistake in the initial phase of your investigation is go charging off in the wrong direction.
  - Example: Starting our investigation for hardware problem that is really a software or database issue or vice versa.

<u>Step 2</u>: Ruling out basic checks

nzhealthckeck

- Any warning related to disk?
- Any warning related to topology issues?

Note:

- Above is only applicable to **Netezza version 7.2.x**
- Presence of above may be indication of some issue, but not always necessarily causing / contributing to performance problem. It is therefore important to continue debugging further unless we have strong evidences that these problems are causing current problem. One can leverage other tools (like nz_responders) to confirm if performance is related to slow disk or something else.

nz_catalog_size

- Does customer have a huge catalog, and that it needs manual vacuum?
- Every query have to go through catalog operation. If catalog on the host is very huge or if it has not been manual vacuumed for some time, probably its a good point to bring it to customer’s attention
- Running nz_catalog_size will print out details of current catalog size and if vacuuming will help reducing the catalog size
- One can perform nz_online_vacuum while database is still online.

Note: It may not be cause of current performance problem, but it is one of the important aspect to keep a check on.

nz_responders

- Any slow dataslices or slow SPU?
- If periodic run of nz_responders shows consistent slow dataslices or set of dataslices, then that may be indication of some problem - either hardware or skewed processing.
- Hardware can be ruled out by checking disk smart data, any topology issues, etc
- Skew / intermediate skew can be ruled out by using nz_spu_swap_space or nz_skew scripts.

Note: It may be that query is processing intermediate data skew, thus doing more work as compared to other dataslices. Because of this, it will cause delays for other queries as well on those dataslices. One can use nz_spu_swap_space to confirm presence of skewed processing or nz_skew to identify skewed tables (discussed in next slides).

nz_skew

- Any skewed tables present?

- If customer have skewed tables, it should be notified to customer and asked to be taken care. It may not be causing current performance problem. Even if customer argures that these tables are not being used, unknowingly they do get accessed because of following:
    - Many times, customers groom on all tables in all databases
    - Many times, customers run genstats on all tables in all databases
    - Backup / restore jobs in another example

- All above are example when skewed tables gets accessed and causes un-necessarly delays in execution.

Note: Small skew are ok to be present..but anything that has big skew ratio.. That should be addressed.

nz_spu_swap_space

- Any intermediate data skew processing ?

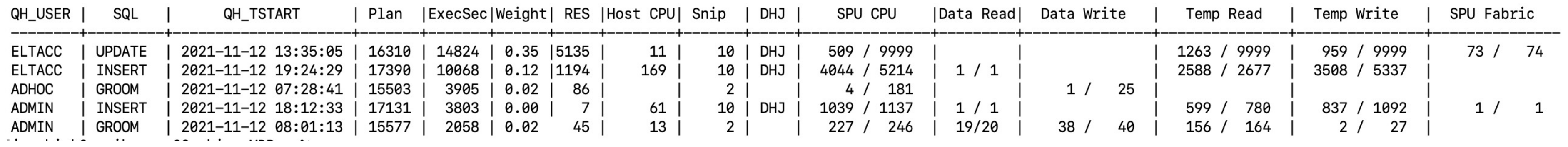
- If there are queries that are generating huge transient data, and that is skewed, that one query can cause some slowness on the system. It is good to check for any significantly long running query that executed only on few dataslices most of the time.

- One can also review nz_query_history (if enabled) to determine resource intensive queries

nz_query

NEW

- If nz_query_history is enabled, one can use nz_query utility to analyze long and resource intensive queries.
- Sample commands and usage
  - `nz_query HISTORY_DATABASE -limit 20 -date '2021-09-01'`
  - `nz_query HISTORY_DATABASE -limit 20 -range '2021-09-01' '2021-10-01'`
  - `nz_query HISTORY_DATABASE -limit 20 -range '2021-09-01' '2021-10-01' -sql "UPDATE"`
  - `nz_query HISTORY_DATABASE -limit 20 -range '2021-09-01' '2021-10-01' -desc "Exec Secs”`



```
QH_USER |   SQL    |      QH_TSTART        | Plan  |ExecSec|Weight| RES |Host CPU| Snip  | DHJ |    SPU CPU     |Data Read|  Data Write    |   Temp Read    |   Temp Write   |  SPU Fabric
--------+----------+-----------------------+-------+-------+------+-----+--------+-------+-----+----------------+---------+----------------+----------------+----------------+----------------
ELTACC  | UPDATE   | 2021-11-12 13:35:05 | 16310 | 14824 | 0.35 |5135 |      11 |     10 | DHJ |   509 / 9999   |         |                |  1263 / 9999   |   959 / 9999   |    73 /    74
ELTACC  | INSERT   | 2021-11-12 19:24:29 | 17390 | 10068 | 0.12 |1194 |     169 |     10 | DHJ |  4044 / 5214   |  1 / 1  |                |  2588 / 2677   |  3508 / 5337   |
ADHOC   | GROOM    | 2021-11-12 07:28:41 | 15503 |  3905 | 0.02 |   86 |         |      2 |     |     4 /  181   |         |     1 /    25  |                |                |
ADMIN   | INSERT   | 2021-11-12 18:12:33 | 17131 |  3803 | 0.00 |    7 |      61 |     10 | DHJ |  1039 / 1137   |  1 / 1  |                |   599 /  780   |   837 / 1092   |     1 /     1
ADMIN   | GROOM    | 2021-11-12 08:01:13 | 15577 |  2058 | 0.02    45 |      13 |      2 |     |   227 /  246   |  19/20  |    38 /    40  |   156 /  164   |     2 /    27  |
```

```
                                        Exec Sec : Query Execution Seconds
                                          Weight : Resource % used by query
                                             RES : Total Resources used by query
                                        Host CPU : Total Host CPU
SPU CPU / READ / WRITE / TEMP READ / TEMP WRITE : Two numbers, AVG / MAX across the spu
```

Note: As a best practice, one should perform this exercise at regular intervals. This helps to understand how your system is performing and what actions needs to be taken for resource intensive queries, if any.

Long running queries

- Are there any long running queries ?

- Are these part of regular jobs but took longer to execute?

- Any queries (other than load/unload jobs) that have been running for a long time (over 1 hour for example) means that it must have utilized lot of resources on the system, causing others to slow down.

- One should try to understand why these queries are long? Few things to look at are:
    - Statistics on the tables involved.
    - Skewed table.
    - Transient / Intermediate data skew.
    - Versioned table, etc.

- If query is new, it may need some tuning by user to run it efficiently.

- One should utilize nz_query_stats to get daily statistics from query history database. This usually helps in understanding what type of workload / queries run on system and if anything is different than previous day(s). For example:

    - `nz_query_stats <hist_db_name> -interval "1 day" -periods 7-startdate 2016-01-01`

nz_transactions

- Is there any big difference between stable txid and current txid?

```
/nz/support/bin/nz_transactions

General transaction information
==========================================================
      Stable TXid : 3224629    (0x313435)    All transactions lower than this are visible (i.e., committed/rolled back)
        Last TXid : 3224630    (0x313436)    The last transaction ID that was assigned (probably to this transaction)
       Difference : 0                        Number of transactions that separated these two values
    TX Array Size : 2                        Maximum is system.maxTransactions=131596
     Stable PGxid : 47231367
       Last PGxid : 47231367                 Maximum is 4B
       Difference : 0
```

- If there is a big difference,
    - it can cause performance issues on hosts
    - Storage does not free’s up as part of truncate

- May of our internal operations (such as groom, backups/restores) are based on stable transaction ID and at times, we have seen customer have open transaction for days breaking these day-to-day process, causing performance problems.

- If replication is enabled, subordinate node will start lagging because that too is depended on stable txid on master.

nz_altered_table

- Any tables that were altered and not groomed ?
- NPS allows up to 4 versions of a table, but these are internally treated as views.
- Thus, when user queries such altered tables, optimizer will generate plans against views, and depending on what operation, we may end up generating bad plans. It is therefore advised to periodically check on presence of versioned tables and groom them using following command
- Groom table <tableName> versions; OR nz_altered_tables -groom

Note: It may not be causing current performance problem, but its always a good practice to keep a check on this.

General Comments

- Have customer being following backups/groom/generate statistics best practices?
- Have any of these been missed in last few days/weeks because of any unrelated issues?
- Any other outages, and Netezza just may be victim leading to performance issues ?
- All these information are important factors that can contribute to overall degradation of system performance. Thus, these should not be ignored.

Performance Issues

<u>Step 3</u>: Further Troubleshooting

- Once you have followed through step 1 and step 2, you should know which direction your investigation should go.

- Following are major classification of performance issues:
    - **Hardware**: Identify / Eliminate hardware issues by running system health check
    - **Software**: This can go in different directions (including but not limited to)
        - Is the problem happening right now or already happened?
        - Any long running jobs/queries when slow performance was observed?

- What were the WLM resource allocations at the time of slow performance?

- **Temporary Hung**: At times we would see that system was not responding, and after some time, queries started running.

- For such issues, we should try identify if it was fcomm related issue or workload related issue.

Ruling Out Hardware Issues

- If we suspect we need to rule out hardware issues on the system, following tests can be performed.

- Perform disk scan speeds test

```
nz_check_disk_scan_speeds
 -- while monitoring it via --
nz_responders -sleep 1
```

- A "loop-back" test using nz_migrate to unload + reload a dummy table. This test helps to eliminate problems associated with internal fabric (among other things)

```
nzsql -c "create database ibm_test_db"
nz_migrate -shost `hostname` -thost `hostname` -sdb system -tdb ibm_test_db -t
nz_check_disk_scan_speeds -format ascii -threads 12 -cksum none -createtargettable true -
truncatetargettable true -status 10 -sourcewhereclause "the_extent <= 100”
nzsql -c "drop database ibm_test_db"
```

- Note: System needs to be idle when performing these tests

Ruling Out Software Issues

Working with monitor.sh

- Monitor.sh script with “-type performance” and “-archive_plans yes” helps to collect the required information all through one script.

- This information is very important for our investigation from software standpoint.

- We can leave script running and whenever customer reports performance issue, we can look for logs for that window, including plan files.

- When working with IBM Netezza support team, make sure to capture all the information detailed here.

Performance Issues… (continued)

- Monitor.sh script with -type performance -archive_plans yes
- collects performance related information, for example

```
*/2 * * * * /nzscratch/monitor.sh -type performance -archive_plans yes -pmr TS123456789
```

- Always ask customer for the window when the performance problem was observed. that makes our investigation to the window.
- Make sure to collect following logs:

```
postgres logs     for period of job/query execution
dbos logs         for period of job/query execution
sysmgr logs       for period of job/query execution

Logs from monitoring script (for specific date / time when the issue occurred)


tar cvzf monitor.tar.gz /nzscratch/monitor/log/ TS123456789/DD-MM-YYYY/
tar cvzf plans.tar.gz /nzscratch/monitor/log/plansarchive
```

Thank You!!