

Vertica Assessment

Query Execution Logs Extraction and Source Code

Prerequisites

Contents

- 1. Introduction3
- 2. Assessment Process.....3
 - 2.1 Logs.....3
 - 2.2 Day wise Total IO Usage5
 - 2.3 Hardware Config for Total CPU Calculation5
 - 2.4 Database Object Count5
 - 2.5 Database Volume6
 - 2.6 High Data Volume Tables6
 - 2.7 Total Number of Databases, Users and Schemas7
 - 2.8 Total CPU Per Hour.....7
- 3. Source Code Assessment.....8

1. Introduction

LeapLogic Assessment profiles existing inventory, identify complexity, lineage and provides comprehensive recommendations for migration to modern data platform.

Assessment checklist

Information	Format	Availability?
Vertica Query Logs	File Export in text format	Choose an item.
Day-wise Total I/O Usage	File Export in text format	Choose an item.
Hardware Config for Total CPU Calculation	Inline in the document	Choose an item.
Database Object Count	File Export in text format	Choose an item.
Database Volume	File Export in text format	Choose an item.
High Data Volume Tables	File Export in text format	Choose an item.
Total Number of Databases, Users, Schemas	File Export in text format	Choose an item.
Total CPU hour-wise	File Export in text format	Choose an item.
Source Code	File Export	Choose an item.

Follow the steps in section 2, 3 to collect the required information for assessment.

2. Assessment Process

All the environment specific variables are highlighted in the document in **Green**. The start and end date should be consistent for all the queries.

2.1 Logs

This requires query execution logs in a CSV format. The CSV file must have the following columns (separated file) retrieved from query_profiles and related system tables.

```
FILLER~~USERNAME~~appid~~CLIENTID~~STARTTIME~~ampcputime~~TotalIOCount~~ParserCPUtime~~firstresptime~~firststeptime~~procid~~queryid~~maxampcputime~~maxampio~~totalcpu~~totalio~~exec_time~~objectdatabasename~~SCHEMANAME~~StatementType~~QUERYTEXT
```

This query is used to get columns in a specific order as required for the assessment. This enables the tool to capture the most used tables.

```
Select
'IDWWM' || '~~' ||
User_name || '~~' ||
'DEFAULT_APPLICATION' || '~~' ||
```

```

'DEFAULT_client_id' || '~' ||
start_timestamp || '~' ||
memory_acquired_mb || '~' ||
'0' || '~' ||
'0.0' || '~' ||
end_timestamp || '~' ||
'00:00:00' || '~' ||
Request_id || '~' ||
transaction_id || '~' ||
max(memory_acquired_mb) over(order by null) || '~' ||
'0' || '~' ||
'0.0' || '~' ||
'0' || '~' ||
(end_timestamp-start_timestamp ) || '~' ||
" || '~' ||
substr(search_path, 0, instr(search_path, ',')) || '~' ||
" || '~' ||
Request
FROM query_requests
where start_timestamp between '2021-05-15 00:04:11' and '2021-05-22 00:04:11'
and success=true;

```

Note: Change the date range according to the assessment time period.

- Second query will be used to perform the assessment manually. This query will find the longest running queries.

```

SELECT
(query_duration_us/1000000)::NUMERIC(10,3) duration_sec,
session_id,
transaction_id,
statement_id,
node_name,
query
FROM
query_profiles
WHERE
query_start BETWEEN '2021-05-22 19:00:0' AND '2021-05-26 22:00:00'
ORDER BY
duration_sec DESC;

```

Note: Change the date range as per the assessment time period.

- Third query will be used to find the list of users with longest running aggregated queries.

```
SELECT
user_name,
sum(query_duration_us/1000000)::NUMERIC(10,3) duration_sec
FROM
query_profiles
WHERE
query_start BETWEEN 2021-05-22 19:00:00 AND 2021-05-26 22:00:00
group by 1
ORDER BY
duration_sec DESC;
```

2.2 Day wise Total IO Usage

Save the results of this query in a CSV file.

```
select to_date(cast(start_time as varchar(30)), 'yyyy-mm-dd'), sum(total_reads_peak_delta+total_writes_peak_delta)
from dc_io_info_BY_DAY
where start_time between 2021-05-16 and 2021-05-18
group by 1
```

Replace 2021-05-16 and 2021-05-18 **with** actual export assessment 'start date' and 'end date +1'. For e.g. in the current example, it would give I/O for **two dates** 2021-05-16 and 2021-05-17.

2.3 Hardware Config for Total CPU Calculation

Inline in the document.

- Vertica processors, Memory per physical core
- Count of Vertica nodes

2.4 Database Object Count

Save the results of this query in a CSV file.

```
select schema_name, TABLE_TYPE, count(table_name)
from all_tables
group by 1,2
order by 1
```

2.5 Database Volume

Save the results of this query in a CSV file.

```
SELECT /*+ label(estimated_raw_size)*/
  pj.anchor_table_schema,
  pj.used_compressed_gb
FROM (SELECT ps.anchor_table_schema,
  SUM(used_bytes) AS used_compressed_gb
FROM v_catalog.projections p
  JOIN v_monitor.projection_storage ps
  ON ps.projection_id = p.projection_id
WHERE p.is_super_projection = 't'
GROUP BY ps.anchor_table_schema) pj
order BY pj.used_compressed_gb DESC;
```

2.6 High Data Volume Tables

Save the results of this query in a CSV file.

```
SELECT
  anchor_table_schema ,
  anchor_table_name ,
  SUM (used_bytes) / (1024^3) AS used_gb
FROM
  v_monitor.column_storage
GROUP BY
  anchor_table_schema ,
  anchor_table_name
having SUM (used_bytes) / (1024^3) >10
ORDER BY
  SUM (used_bytes) DESC;
```

This query will collect database tables with volume above 10 GB

Note: The yellow highlighted filter condition can be changed for collecting lower volumes.

2.7 Total Number of Databases, Users and Schemas

Save the results of these queries in a CSV file.

```
select database_name from databases;  
select USER_NAME from users;  
select SCHEMA_NAME from SCHEMATA;
```

2.8 Total CPU Per Hour

Save the results of these queries in a CSV file.

```
Select  
time,  
node_name,  
start_time,  
end_time,  
processor_id,  
number_of_processors,  
cpu_min_frequency_mhz_sample_count,  
cpu_min_frequency_mhz_sample_sum,  
cpu_max_frequency_mhz_sample_count,  
cpu_max_frequency_mhz_sample_sum,  
user_microseconds_peak_delta,  
nice_microseconds_peak_delta,  
system_microseconds_peak_delta,  
idle_microseconds_peak_delta,  
io_wait_microseconds_peak_delta,  
irq_microseconds_peak_delta,  
soft_irq_microseconds_peak_delta,  
steal_microseconds_peak_delta,  
guest_microseconds_peak_delta  
Where start_time BETWEEN '2021-05-22 19:00:0' AND '2021-05-26 22:00:00';
```

3. Source Code Assessment

Provide the below active or in-scope Vertica source code artifacts as applicable in the migration scope.

Sl. No.	Code Artifact	Criticality	Remarks
1	Orchestration Scripts (Control-M / Autosys / Cron etc.)	Must	To identify interdependencies across scheduler scripts / jobs, queries, and dependent workloads
2	Procedures / Functions*	Must	To identify workload complexity, query patterns, query count etc. for effort estimation and technical debt
3	Views	Must	To identify view complexity, patterns and effort estimations
4	Shell Scripts*	Must	To identify count, dependencies, SQL queries and PL/SQL, logic (example: email, notification etc.) and effort estimations
5	DDL	Must	To identify column usage, and provide recommendation on column level lineage, and query optimization on the target system
6	DML / SQL Files*	Must	To identify count, dependencies, SQL queries and effort estimations

Note:

Limit: Assuming the orchestration script is a trigger point for every single use case execution in the existing setup. If customer is not comfortable sharing all the workloads, then share those workloads which are referred or executed through the orchestration scripts. However, in such scenarios the scope and effort estimates will be based on the given workloads.