**Slow Performance of Saving Scheduled, or Final for Post, Inventories on a Ship troubleshooting recommendations**

**The Purpose of this Document is to:**

* Provide guidance for what to do when there is a report of slow performance saving Inventory for a Ship.
* Provide a course of action for our Customers to take if any of 3 specific Queries ( SQL IDs) are being used as Crunchtime has identified a degradation in performance due to Oracle not choosing the best execution plan for these SQL IDs. Using our extensive pool of over 200 Databases, we are able recommend a solution that will force Oracle to use the best choices by adding optimal Baselines for the SQL IDs to your Ship Database.

**In General:**

* Self Hosted Customers must monitor and troubleshoot their own Database Performance.
* Self Hosted Customers must collect database statistics on their Databases regularly and perform basic maintenance to Oracle standards.
* Although Crunchtime can make recommendations from time to time as a courtesy, we are not able to provide ongoing or regular advice or guidance on how to monitor, troubleshoot, or administer a self hosted Database.

**In the specific case of slow Inventory performance, a Known Issue that has been reported to us by multiple customers after upgrade to Version 8.0+, we are providing specific troubleshooting steps and a recommended solution:**

Background on the Known Issue:

The root cause of the slow performance is not a defect of the Crunchtime Code/Product.

In some Environments, Oracle is not choosing the optimal execution plan for certain Queries. We have identified 3 specific Queries ( SQL IDs) and in this document we are providing instructions for our Customer’s DBAs to follow to implement in their Ship Databases Baselines that force Oracle to choose the optimal execution plan. The source of these baselines are from Crunchtime Hosted Customers in who’s Databases those same SQL IDs are performing well.

**Part 1, Monitoring:**

**Please note that once the Inventory Save Process is running, even if Net-Chef times out, the Inventory is still Saving in the Database. Advise Users that they should not be attempting to Save again.**

* When Slow Performance is reported, the Customer’s DBA will need to monitor the session. They will need to be looking for slowly running queries and, if identified, record those queries (SQL IDs, Plan Hash Values, Elapsed Execution Time including the Execution Plan).
* If the SQL IDs are any of the following 2syc28xvgzvr1, cr6gkqwytnbpg , and 7w7kzr32xfunf  please proceed to **Part 2, Deploy Baselines**.
* If the SQL IDs are not any of the aforementioned, please record the Queries and collect execution statistics on the Queries by running the following and providing it in the Support Ticket along with the SQL IDs. Crunctime support will continue to engage with you after receiving the required information.

*You will need to enter in the SQL IDs in the statement below:*

set echo on

set pagesize 2000

col name for a20

col sql\_plan\_baseline for a30

col parsing\_schema\_name for a30

select inst\_id, name, s.parsing\_schema\_name, sql\_id, child\_number, plan\_hash\_value, SQL\_PLAN\_BASELINE, executions exe, round(cpu\_time/1000000/executions) cpu, round(elapsed\_time/1000000/executions,3) avg\_sec, round(elapsed\_time/1000000) total\_time

from gv$sql s join v$pdbs p on s.con\_id = p.con\_id

where sql\_id in ('&QuerySQLID') and executions > 0

order by sql\_id, plan\_hash\_value, name;

-- Put SQL ID and child number displayed by the query above.

SELECT \* FROM table(DBMS\_XPLAN.DISPLAY\_CURSOR('&QuerySQLID', &child\_number));

select sql\_fulltext from gv$sql where sql\_id = '&QuerySQLID' and rownum = 1;

**Part 2, Deploy Baselines:**

**Available baselines:**

Crunchtime has baselines available for queries with the following SQL IDs: 2syc28xvgzvr1, cr6gkqwytnbpg, 7w7kzr32xfunf

**Example of the tasks for Customer’s DBA to complete prior to Closing of the Voyage:**

* Copy SQLID\_7W7KZR32XFUNF\_3633281728.dmp.gz, SQLID\_CR6GKQWYTNBPG.dmp.gz and SQLID\_2SYC28XVGZVR1.dmp.gz files to your DB server and uncompress them.
* Import SQLID\_7W7KZR32XFUNF\_3633281728, SQLID\_CR6GKQWYTNBPG and SQLID\_2SYC28XVGZVR1 tables into SYSTEM schema.

impdp system/password@DB\_hostname/DB\_service directory=CTDUMP dumpfile=SQLID\_7W7KZR32XFUNF\_3633281728.dmp logfile=SQLID\_7W7KZR32XFUNF\_3633281728.imp

impdp system/password@DB\_hostname/DB\_service directory=CTDUMP dumpfile=SQLID\_CR6GKQWYTNBPG.dmp logfile=SQLID\_CR6GKQWYTNBPG.imp

impdp system/password@DB\_hostname/DB\_service directory=CTDUMP dumpfile=SQLID\_2SYC28XVGZVR1.dmp logfile=SQLID\_2SYC28XVGZVR1.imp

* Load baseline into the Database (Successful completion of the following code block will display  v\_plan\_cnt: 1. If you see v\_plan\_cnt: 0 then something went wrong and baseline did not get loaded.)

set echo on

set serveroutput on

DECLARE

v\_plan\_cnt NUMBER;

BEGIN

v\_plan\_cnt := DBMS\_SPM.UNPACK\_STGTAB\_BASELINE (table\_owner => 'SYSTEM', table\_name => 'SQLID\_7W7KZR32XFUNF\_3633281728');

dbms\_output.put\_line('v\_plan\_cnt: ' || v\_plan\_cnt);

END;

/

DECLARE

v\_plan\_cnt NUMBER;

BEGIN

v\_plan\_cnt := DBMS\_SPM.UNPACK\_STGTAB\_BASELINE (table\_owner => 'SYSTEM', table\_name => 'SQLID\_CR6GKQWYTNBPG');

dbms\_output.put\_line('v\_plan\_cnt: ' || v\_plan\_cnt);

END;

/

DECLARE

v\_plan\_cnt NUMBER;

BEGIN

v\_plan\_cnt := DBMS\_SPM.UNPACK\_STGTAB\_BASELINE (table\_owner => 'SYSTEM', table\_name => 'SQLID\_2SYC28XVGZVR1');

dbms\_output.put\_line('v\_plan\_cnt: ' || v\_plan\_cnt);

END;

/

* Verify that baseline has been successfully loaded (The following query should return 1 row).

col PLAN\_NAME for a40

SELECT SQL\_HANDLE, SQL\_TEXT, PLAN\_NAME, ORIGIN, ENABLED, ACCEPTED FROM DBA\_SQL\_PLAN\_BASELINES WHERE PLAN\_NAME in ('SQL\_PLAN\_2h10800r5qgfj3617a647','SQL\_PLAN\_akc767y6150v140bb3975','SQL\_PLAN\_a27wutw8y56fu05ea41a3');

**Ship Voyage Closing (Coordinate DBA/Ship resources to monitor DURING close):**

* Test the Scheduled of Final for Post review/save close - DBA must monitor the Database while Inventory is being Saved, if poor performance is identified then verify the statistics as seen below (provide them to Crunchtime along with the SQL identified). This must be done immediately following the completion of the save.

* Verify SQL execution statistics, execute the following query right after completion of Cruise Closing to avoid SQL aging out of the Database SQL area:

set echo on

set pagesize 2000

col name for a20

col sql\_plan\_baseline for a30

col parsing\_schema\_name for a30

select inst\_id, name, s.parsing\_schema\_name, sql\_id, child\_number, plan\_hash\_value, SQL\_PLAN\_BASELINE, executions exe, round(cpu\_time/1000000/executions) cpu, round(elapsed\_time/1000000/executions,3) avg\_sec, round(elapsed\_time/1000000) total\_time

from gv$sql s join v$pdbs p on s.con\_id = p.con\_id

where sql\_id in ('7w7kzr32xfunf','cr6gkqwytnbpg','2syc28xvgzvr1') and executions > 0

order by sql\_id, plan\_hash\_value, name;