
ORACLE®

Your. Open. World.

Performance Analysis of Oracle Streams Configurations: MAA Best Practices

Patricia McElroy
Principal Product Manager
Oracle Streams

Darryl Presley
Consulting Member of Technical Staff
MAA / HA Group, Server Technologies

ORACLE

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- Streams Architecture
- Streams Configuration
- Streams Performance Data Collection
- Performance Tuning Methodology & Examples

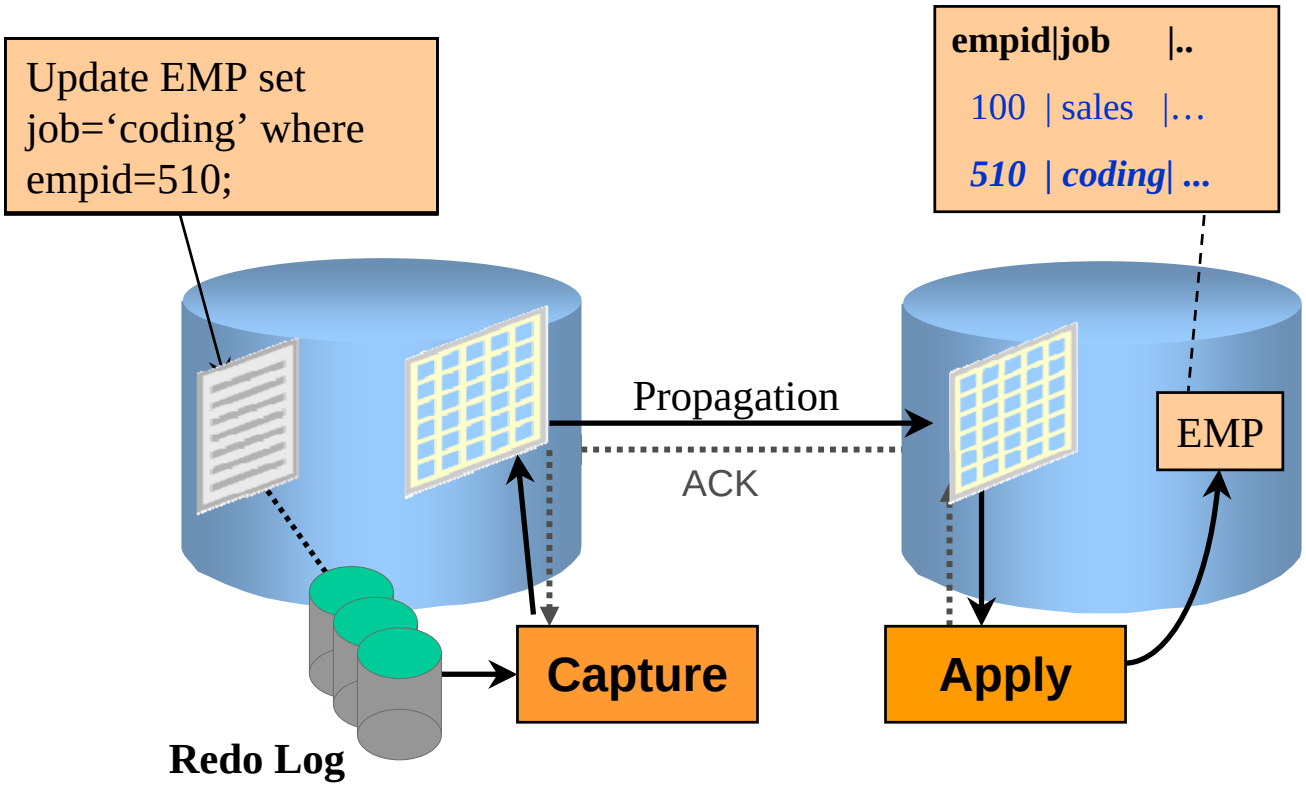
Streams Architecture

Oracle Streams

- Simple solution for information sharing
- Provides
  - Uniquely flexible information sharing
  - Active/Active replication
  - Real-time ETL for data warehouses
  - Availability during database migration
  - Availability during application upgrade
  - Message queuing
  - Event management and notification

Streams Capture and Apply

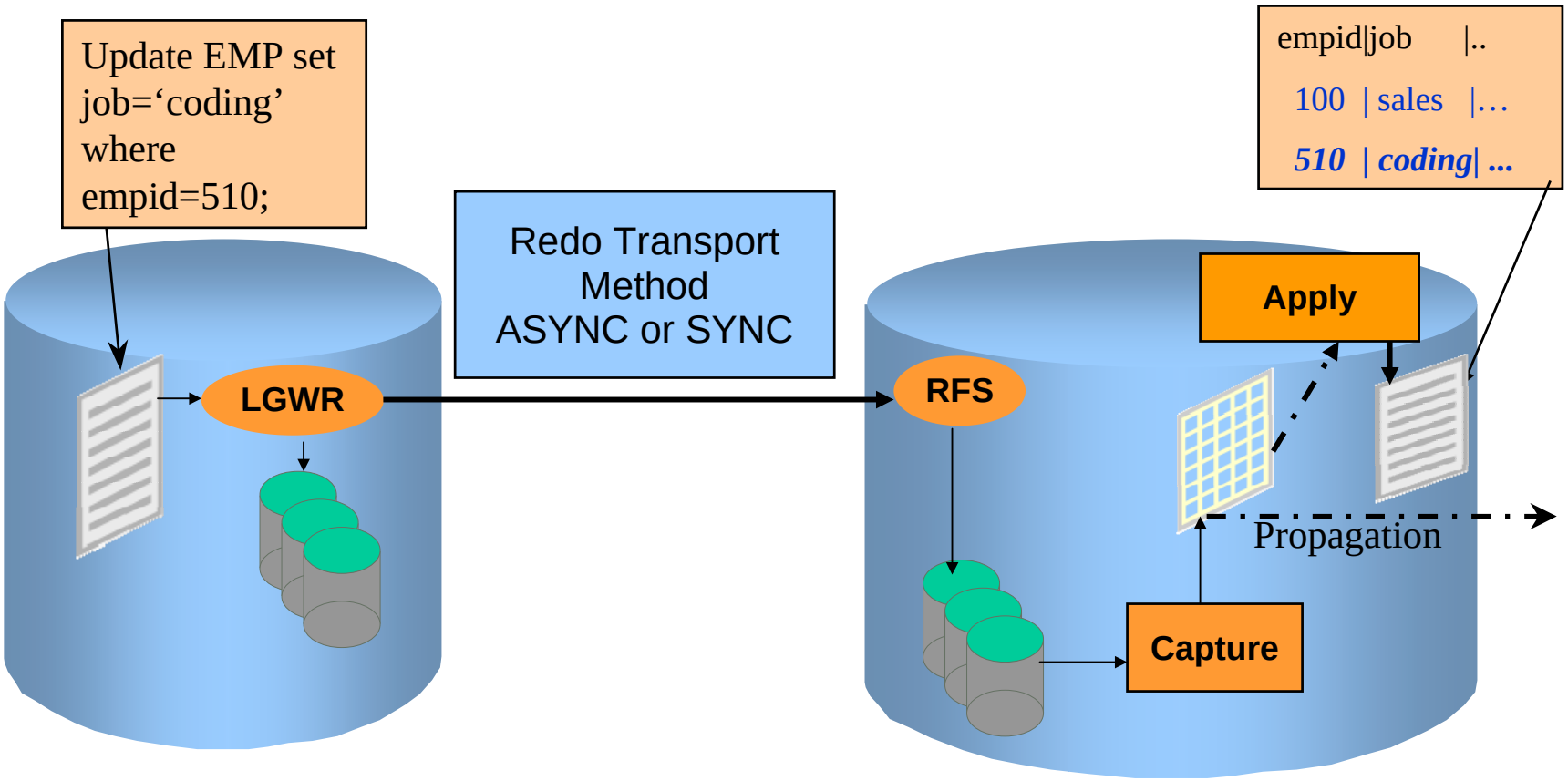
Replication Configuration



Local (Upstream) Capture

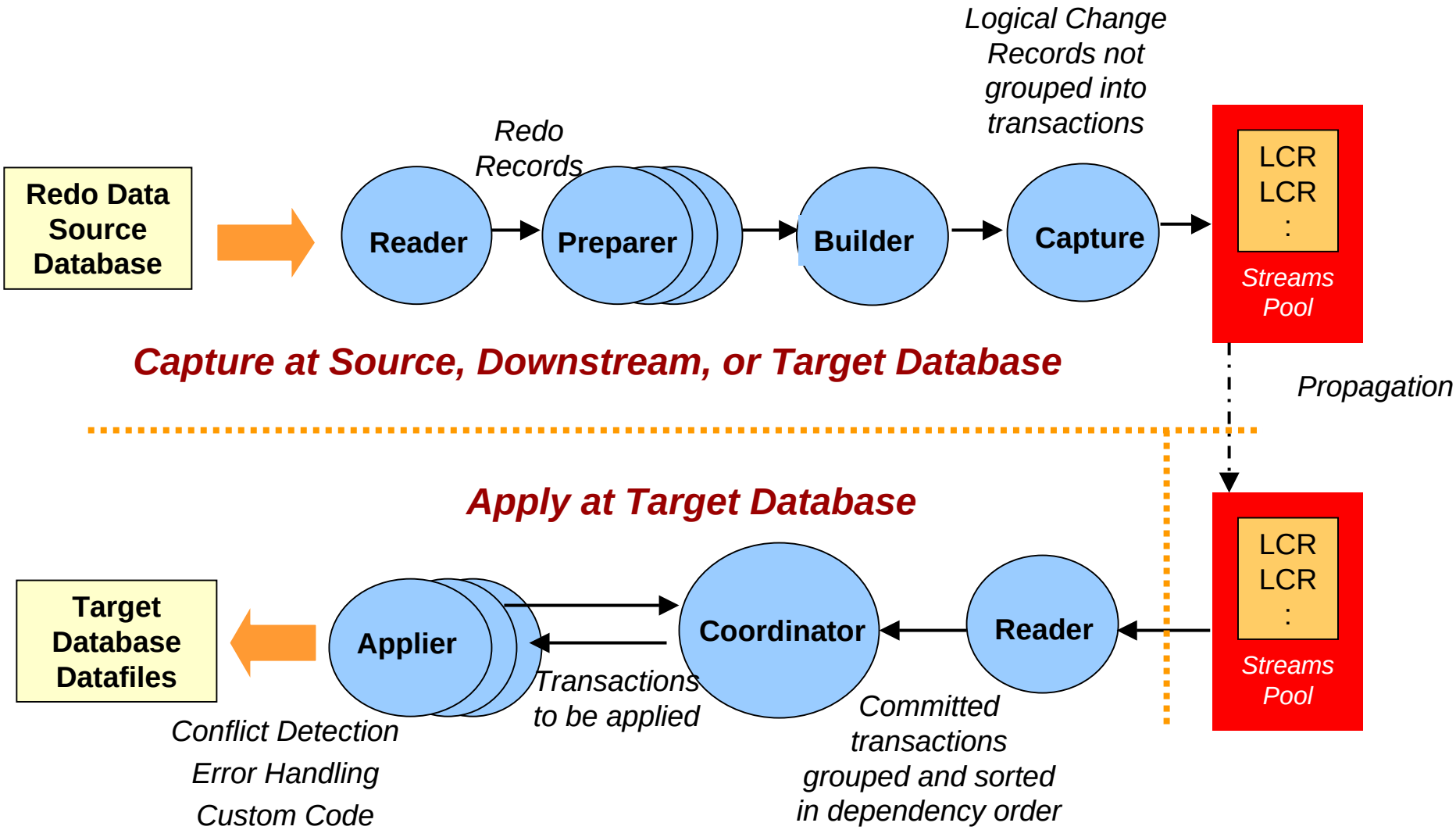
If subset of data, network bandwidth requirements reduced significantly

Downstream Capture



Offload capture from production database

Streams Process Architecture



Streams Configuration

Streams Configuration

- Setup Wizards for Streams in Enterprise Manager
- DBMS_STREAMS_ADM.MAINTAIN_* package procedures
  - Sets up Streams with proven best practices configurations
- Follow MAA best practices in *Oracle Streams Configuration Best Practices*
  - Simplifies configuration and provides better performance base line
  - Avoid most configuration and run-time problems during Streams set up

Streams Performance Data Collection

Performance Data Collection

- Runtime Views
- Streams HealthCheck
- Performance Data Collection Tools
  - STRMMON (9*i*/10*g*)
  - UTL_SPADV (11*g*)
  - AWR / ASH

Streams Views

- Configuration and management views specific to Capture, Propagation, and Apply
  - DBA_CAPTURE, DBA_PROPAGATION, DBA_APPLY, etc.
- Key runtime views show statistics and state information for Streams processes
  - V$STREAMS_TRANSACTION, …
  - V$STREAMS_CAPTURE, V$LOGMNR_STATS
  - V$PROPAGATION_SENDER, V$PROPAGATION_RECEIVER, DBA_QUEUE_SCHEDULES
  - V$STREAMS_APPLY_READER, V$STREAMS_APPLY_COORDINATOR, V$STREAMS_APPLY_SERVER
- For RAC, use GV$ format to collect statistics from any instance

Streams HealthCheck Script

- Configuration
  - Generic database information
  - Streams process configuration details
- Analysis
  - Rules correlated with other configuration to identify discrepancies
  - Detail view information for diagnostic purposes
- Statistics
  - Runtime state of Streams processing
- Report
  - Advice and Warnings of potential issues with configuration

Navigation Aids

STREAMS Health Check (V4.0.06) for HRTSE.WORLD on Instance=HRTSE3 generated: 15:30:31 08/05/08

Configuration: Queue Capture Propagation Apply Database
Analysis: History Rules Notifications Configuration Performance Wait Analysis Topology
Statistics: Streams Statistics Queue Capture Propagation Apply Apply_Errors

==========

++ DATABASE INFORMATION ++

Database Identifier	Name	CREATED	MIN_LOG	PK_LOG	UI_LOG	FK_LOG	ALL_LO	FORCE_LOG	RESETLO
4012323324	HRTSE	12:35:45 07/07/08	YES	NO	NO	NO	NO	NO	17:20:20 07

==========

Summary Report

```
++ Notifications ++

+ WARNING: There are 925 archived logs ready to be purged from disk.
+ Use the following select to identify unneeded logfiles:
+ select name from dba_registered_archived_log where purgeable = "YES"
PL/SQL procedure successfully completed.


++
++ SYS Checks
++
+
PL/SQL procedure successfully completed.
++
++ init.ora checks ++
++
+ ERROR: The parameter 'aq_tm_processes' should not be explicitly set to 0!
+ Queue monitoring is disabled for all queues.
+ To resolve this problem, set the value to 1 using: ALTER SYSTEM SET AQ_TM_PROCESSES=1;
++
PL/SQL procedure successfully completed.


++
++ Configuration checks ++
```

Wait Analysis

++ STREAMS Process Wait Analysis ++

Analysis of last 30 minutes of Streams processes

Note: When computing the busiest component, be sure to subtract the percentage where BUSY = 'NO'
Note: 'no rows selected' means that the process was performing no busy work, or that no such process exists on the system.
Note: A null Wait Event implies running - either on the cpu or waiting for cpu

++ PROPAGATION RECEIVER PROCESSES ++

PROPAGATION_RECEIVER_NAME	EVENT_COUNT	TOTAL_COUNT	PERCENTAGE	BUSY	Wait Event
"STRMADMIN"."HRTSE_PAY_CAP_Q @HRTSE.WORLD"=>HWTSA.WORLD	4	759	.5	YES	latch: shared pool
	24	759	3.2	YES	SQL*Net more data fro client
	82	759	10.8	YES	
	156	759	20.6	YES	Streams: flow control
**********			----------		
TOTAL			35.0		
"STRMADMIN"."HRTSE_TL_CAP_Q@ HRTSE.WORLD"=>HWTSA.WORLD	2	759	.3	YES	latch: shared pool
**********			----------		
TOTAL			.3		

Healthcheck Tips Summary

- Start with Wait Analysis
  - Identify busiest component in last 30 min
- Look at the Report section
  - Identify any ERRORs or WARNINGs that might be applicable to problem
- Look at Statistics section
  - Confirm problem statement from statistics
  - Determine if messages are flowing from source to target
- Check Configuration and Analysis sections
  - Look for indicators of problem symptoms in analysis sections

Performance Data Collection Tools

- Install Application Workload Repository (AWR)
  - Active Session History (ASH) is configured with AWR
  - AWR snapshots should be taken at least once an hour
  - Requires Oracle Diagnostic Pack
- Install and Configure STRMMON (10*g*R2)
  - Available from Metalink: Article 290605.1
- Install UTL_SPADV (11*g*R1)
- Use AWR, STRMMON/UTL_SPADV during performance period of interest
- Run Streams Healthcheck at each database: before, during , and after performance period of interest
  - Available from Metalink: Article 273674.1

STRMMON

- Install and Configure STRMMON (10*g*R2)
  - Download from Metalink Article 290605.1
  - Ensure version is 2.6 or higher
- Collect STRMMON information regularly
  - At least once every hour to align with AWR
  - Use more granular collection for trend analysis
- For specific performance investigation window,
  - Take manual AWR snapshots at the start and end of this window
  - Run STRMMON to collect every 30 seconds

STRMMON: Connect to 2 databases

STRMMON Command line to collect 30 minutes of output from source and target databases in single line

- 30 seconds X 60 Iterations= 1800 seconds or 30 minutes

```
strmmon -interval 30 -count 60 -user sys  -passw  <sys_password>
-dbname <TNS alias to source database>   -sysdba \

 -user sys -passw <sys_password>  -dbname <TNS alias to target
database>   -sysdba >  strmmon.log
```

STRMMON Example

```
2008-02-15 14:52:21
|| STREAMS1->
| LOG 961K
| NET 11K 454K
| C001 455 452 4sec
| PS01 472 467671 456K
|| STREAMS2->
| LOG 539K
| NET 571K 0
| PR01 550 <100%I 0%F ->
| Q51526 552 0
| - A001 1002 2 - AR:<0%I 0%F ->  AS(8) <226%I 0%F ->
| MEM 6 % 208M
```

Streams performance advisor

ORACLE DATABASE 11g

- 11*g* Feature for Streams
  - Discovers Streams topology of database(s) via dbLinks
  - Populates topology views
  - Collects statistics (via dbLinks)
  - Analyzes current interval performance end-to-end
  - Detects most active component (bottleneck)
- UTL_SPADV package
  - Optional installation
  - Wrapper over Streams Performance Advisor
  - Run from 11*g* database with dblinks to Streams databases
  - End-to-end analysis
  - Replacement for STRMMON
  - Future version will analyze 10.2 databases , too.

How to Run UTL_SPADV

- Load UTL_SPADV package into schema with DBLinks to target databases
  - Typically Streams Administrator schema.

```
Sql> @?/rdbms/admin/utlspadv.sql
```

- Run collect_stats
  - Procedure will wait until interval and count satisfied
  - Default interval= 60 sec; Default count=10 iterations

- Monitor statistics with show_stats
  - Use another session to monitor in real-time
  - Connect to same schema
  - Enable serveroutput in session
  - Spool output to file

UTLSPADV Example

11.1.0.7

```
PATH 7 RUN_ID 7 RUN_TIME 2008-AUG-05 15:36:15 CCA Y
|<C> HRTSE_TL_CAP 573 378 4 LMR 54.8% 0% 29% "CPU +
Wait for CPU"
LMP (1)  95.2% 0% 3.2% "" LMB 98.4% 0% 1.6% "" CAP
95.2% 0% 4.8% ""
|<Q> "STRMADMIN"."HRTSE_TL_CAP_Q" 378 0 0
|<PS> =>HWTSA.WORLD
|<PR> HRTSE.WORLD=>
|<Q> "STRMADMIN"."HRTSE_TL_APP_Q" 98 0 0
|<A> HRTSE_TL_APP 384 2 4
APR 95.2% 4.8% 0% "" APC 100% 0% 0% "" APS(8) 775.8%
0% 19.4% "Streams apply: waiting for dependency"
|<B> NO BOTTLENECK IDENTIFIED
```

Streams Performance Tuning Methodology

Performance Tuning Methodology

- Best Practice for simplifying tuning process
- Methodology applies to both 10*g*R2 and 11*g*R1
- Relies on HealthCheck and STRMMON or UTL_SPADV output
- Performance Tuning Steps

  Collect Data

  Identify Trend from STRMMON / UTL_SPADV

  Choose case scenario

  Drill down further with Healthchecks and AWR/ASH information

  Tune performance
- Repeat …

Getting Started: Collect Data

- Streams Healthcheck
  - Multiple samples: before, during, and end of collection period
- STRMMON or UTL_SPADV
  - Collect for entire range of collection period from source and target
  - Use small interval (no more than 60 seconds between iterations)
  - Set count (of iterations) to large number
- AWR Snapshots
  - Multiple snapshots: Start, during, and end of collection period

Simple Checks

- All Processes Running
  - No aborted or disabled processes
  - No failed propagation jobs
- Streams Pool Size
  - Estimated Spill counts are low for 24-hour production day
  - Check for queue spills in HealthCheck
- Adequate System Resources
  - CPU, I/O, Memory

Two Cases for methodology

- Use latency trends of either capture or apply process as a good starting point
- Methodology simplified to detect bottlenecks or weak points in Streams flows

- Case 1: Apply latency increasing and reduced apply rates
  - Start review with Apply and work back towards Capture

- Case 2: Capture latency increasing and reduced capture rates
  - Focuses on Capture Process

CASE 1: Apply Latency Is Increasing

Performance Tuning Methodology Case 1: Apply Latency is Increasing

- Start with Apply
  - Check for Flow Control <F> or Bottleneck <B> indicators in front of Apply (A00x)
  - Check percent flow control, percent in wait events
    - Find last component in flow control (left to right)
    - Identify first component to the right not in flow control
    - Determine if that component is experiencing performance issues i.e., wait events.

Performance Tuning Methodology Case 1: Apply Latency is Increasing

- If last component in flow control is:
  - Propagation Receiver, look at Apply process
  - Propagation Sender, look at:
    - Wait events
    - Streams pool utilization on target database
    - Network bandwidth
  - Capture, look at:
    - Propagation processes
    - Streams pool size on source database

Performance Tuning Methodology Case 1: Apply Latency is Increasing

STRMMON Example

```
2008-03-17 15:50:50 || STRM10g5-> | LOG 3M | NET 2M 518K | <F> C001 123 84
11min <0%I 8%F 67%"Streams AQ: enqueue blocked on low memory">  |
Q53088 84 12 | PS01 134 0 0 <0%I 97%F ->  | MEM 67 % 256M || STRM10g6-> |
LOG 2M | NET 530K 49 | PR01 118 <0%I 60%F 18%"events in waitclass Other">  |
Q53478 118 0 | Q51358 0 0 | -  A001 390 1 12min AR: <91%I 0%F>  AS(8) <361%I
0%F 254%"log file switch completion">  | MEM 69 % 256M
```

- Last component (viewed left to right) showing flow control is propagation receiver (60%F)
- Apply reader is 91% idle
- All 8 Apply Servers are spending approximately 31.7% on wait event “log file switch completion”
- Look at top 5 wait events in AWR and frequency of occurrence in ASH reports
- Address this wait event by tuning redo log I/O performance

Apply Spill

- Apply spill is used by Apply process to free Streams pool memory for more LCRs
- LCRs for spilled transactions are stored in a spill table
- Apply Spill is triggered on
  - Large transactions
  - Long-running transactions
- Apply Spill can be inferred from STRMMON and UTL_SPADV
  - Apply Reader very busy
  - LCR and transaction apply rates are zero
  - Apply process may have the <B> indicator for bottleneck
- Healthcheck shows information about apply spill
  - TOTAL_MESSAGES_SPILLED column in V$STREAMS_APPLY_READER will indicate apply spill

Identifying Large/Long Transactions

- Large Transactions
  - APPLY: Number of LCRs exceeds TXN_LCR_SPILL_THRESHOLD parameter
  - CAPTURE: message written to alert.log indicating a large transaction (>10000 LCRs) exists
- Long running transactions
  - Fewer row changes but takes longer than 15 minutes
  - Capture process writes message to alert.log of open transaction longer than 20 minutes
- Detecting Long and Large Transactions
  - V$STREAMS_TRANSACTION view identifies open transactions and number of LCRs
  - DBA_APPLY_SPILL_TXN view identifies apply spilled transactions

Queue Spill

- Queue spill protects Streams pool memory by writing to a spill table, AQ$_*queueTableName*_P
- Queue spill triggered on
  - Insufficient streams pool memory
    - Utilization above 70% of Streams pool
  - Slow throughput rates of streams components
    - LCRs spilled after 5 minutes with no browse or dequeue operations
- Not the same as Apply spill
- STRMMON and UTL_SPADV show buffer queue spill rates

Recommendations for Apply Tuning

- Increase Apply parameter PARALLELISM to support more simultaneous transactions
  - Start with 4, then increment by 2 after monitoring
- Set COMMIT_SERIALIZATION to NONE
  - Transaction dependencies are still maintained
  - Performance can improve significantly
- For tables with LOB columns
  - Use Apply Error Handler to Improve Apply Performance
  - Configure apply parameters ASSEMBLE_LOBS to TRUE and ERROR_HANDLER to TRUE
  - See example in MAA 10*g*R2 Streams Performance white paper and download examples.

CASE 2: Capture Latency Is Increasing

Performance Tuning Methodology Case 2: Capture Latency is Increasing

- Characteristics
  - All other Streams components (propagation sender, receiver, apply) are moderately idle, waiting for more work
  - Propagation and Apply rates fluctuate with Capture

- Common causes
  - Capture with moderate to high percent idle
    - Waiting for Redo
    - Large DDL or PDML activity
  - Capture too busy

IMPORTANT: Capture is typically not in flow control in this scenario

Waiting for Redo

- Characteristics
  - LCR capture and enqueue rates drop to zero
  - Capture will be near 100% idle
  - Capture is waiting for more redo (from archive logs or from redo transport) to arrive
  - The STATE column in V$STREAMS_CAPTURE will indicate "WAITING FOR REDO, LAST SCN MINED <scn#>"
- Recommendations
  - Make sure all logfiles needed by capture are available
  - For Downstream Capture
    - Configure redo transport services and standby redo logs
    - Set the capture parameter DOWNSTREAM_REAL_TIME_MINE to 'Y' using the DBMS_CAPTURE_ADM.SET_PARAMETER() procedure

Large DDL or DML Activity

- Characteristics
  - LCR capture and enqueue rates drop to zero with some LCRs being captured periodically
  - Capture will be near 100% idle
  - Large DDL, Parallel DDL and Parallel DML with thousands or millions of changes being executed on the source database
    - PDDL Example: CREATE TABLE … AS SELECT PARALLEL;
    - PDML Example: INSERT INTO TABLE … (SELECT …) PARALLEL;
  - Capture must process through all redo, independent of interesting tables
- Recommendations
  - Let Capture run past all DDL, PDDL, PDML changes

Capture Process 100% Busy

- Characteristics
  - STRMMON / UTL_SPADV may show capture as a bottleneck <B>
  - Capture process not in flow control
  - Capture processing cpu-bound
- Recommendations
  - Distribute replicated tables across multiple Streams paths (Capture->Propagation->Apply)

Performance Tuning Methodology
Case 2: Capture Latency is Increasing

Example – Single capture process at capacity (cpu-bound)

2008-03-17 11:11:30 || STRM10g5-> | LOG 9M | NET 5M 3M | - **C001 1037 704 1min <2%I 0%F ->** | Q53088 704 0 | PS01 703 0 0 <17%I 61%F -> | MEM 67 % 256M || STRM10g6-> | LOG 4M | NET 3M 49 | PR01 707 <43%I 0%F -> | Q53478 707 0 | Q51358 0 0 | Q51211 0 0 | Q51070 0 0 | - A001 728 2 1min AR: <62%I 0%F -> AS(8) <570%I 0%F -> | MEM 33 % 256M

2008-03-17 11:12:30 || STRM10g5-> | LOG 9M | NET 6M 2M | - **C001 884 600 2min <1%I 0%F ->** | Q53088 601 0 | PS01 597 0 0 <21%I 54%F -> | MEM 70 % 256M || STRM10g6-> | LOG 4M | NET 2M 50 | PR01 609 <58%I 0%F -> | Q53478 609 0 | Q51358 0 0 | Q51211 0 0 | Q51070 0 0 | - A001 599 1 2min AR: <73%I 0%F -> AS(8) <649%I 0%F -> | MEM 35 % 256M

- Capture is near zero percent idle
- Capture is not in flow control
- Consider spreading tables being replicated across multiple Streams paths

Performance Tuning Methodology

- Best Practice for simplifying tuning process
- Methodology applies to both 10*g*R2 and 11*g*R1
- Relies on HealthCheck and STRMMON or UTL_SPADV output
- Performance Tuning Steps

  Collect Data

  Identify Trend from STRMMON / UTL_SPADV

  Choose case scenario

  Drill down further with Healthchecks and AWR/ASH information

  Tune performance
- Repeat …
- Refer to MAA *Oracle Streams Performance Tuning Best Practice* paper – *Coming soon!*

HA Sessions, Labs, Demos From Oracle Development

Mon, Sep 22

- 2:30 pm - Database 11*g*: Next-Gen HA, Moscone South 103

Tue, Sep 23

- 9:00 am - Active-Active Data Centers, Moscone South 103
- 11:30 am - Sharding with Oracle, Moscone South 302
- 11:30 am - HA with Oracle VM, Moscone West 3024
- 1:00 pm - Active Data Guard, Moscone South 104

Wed, Sep 24

- 9:00 am - Fusion Middleware Grid HA, Marriott Nob Hill AB
- 11:30 am - RMAN Best Practices, Moscone South 103
- 5:00 pm - Data Guard & Real Application Testing, Moscone 102
- 5:00 pm - EM in Secure MAA, Moscone West 2001

Wed, Sep 24 (contd.)

- 5:00 pm - E-Business Suite HA, Moscone West 2002/04

Thu, Sep 25

- 9:00 am - Oracle Secure Backup, Moscone South 102
- 10:30 am - Streams Replication, Moscone South 102
- 12:00 pm - Rolling Database Upgrades, Moscone South 103
- 1:30 pm - Streams Performance, Moscone South 102
- 3:00 pm - Oracle Grid Computing, Moscone South 303
- 3:00 pm - E-Business Suite R12 MAA, Moscone West 2007
- 3:00 pm - Siebel MAA, Moscone South 308
- 3:00 pm - Fusion SOA HA & Scalability, Marriott Salon 14/15

Hands On Labs - Thu, Sep 25

- 10:30 - 11:30 am, 12:00 - 1:00 pm - Active Data Guard, Marriott Golden Gate A3

DEMOgrounds, Mon-Thu

- Active Data Guard, Streams, Oracle Secure Backup, RMAN/Flashback, MAA

For More Information

search.oracle.com

or

oracle.com


ORACLE®