Oracle GoldenGate

Cache Memory Usage

```
./ggsci
dblogin USERIDALIAS ggadmin
SEND EXTRACT ext_mr CACHEMGR CACHESTATS
SEND REPLICAT rep_file CACHEMGR CACHESTATS
```

Statistics

```
./ggsci
STATS REPLICAT rep_mr, TOTALSONLY *.*, REPORTRATE MIN
STATS EXTRACT ext_mr, TOTALSONLY *.*, REPORTRATE MIN
```

Process Info

```
INFO REPLICAT *, ALLPROCESSES
INFO REPLICAT *, TASKS
INFO EXTRACT *, TASKS
```

Network Stats

```
send extract dp_mr gettcpstats
send extract dp_mr resettcpstats
```

Stats from within DB

```
/* Extract side */
SELECT capture_name, 86400 *(available_message_create_time -
capture_message_create_time) latency_in_seconds FROM GV$GOLDENGATE_CAPTURE;

/* Replicat side */
SELECT r.apply_name, 86400 *(r.dequeue_time - c.lwm_message_create_time)
latency_in_seconds FROM GV$GG_APPLY_READER r, GV$GG_APPLY_COORDINATOR c
WHERE r.apply# = c.apply# AND r.apply_name = c.apply_name;
```

Heartbeat Table

Show heartbeat for all extracts and replicats

```
SELECT * FROM ggadmin.gg_heartbeat;
```

Show lag history

```
SET pagesize 200 linesize 200
col heartbeat_received_ts format a30
col incoming_path format a24
col incoming_lag format 999,999.999
ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';
SELECT heartbeat_received_ts, incoming_path, incoming_lag FROM
ggadmin.gg_lag_history;
```

Tuning STREAMS pool

Set the STREAMS_POOL_SIZE initialization parameter for the database to the following value:
(MAX_SGA_SIZE * # of integrated Extracts) + 25% head room

```
SELECT current_size FROM v$sga_dynamic_components WHERE component = 'streams
pool';

/* Streams pool advice */
COLUMN STREAMS_POOL_SIZE_FOR_ESTIMATE  HEADING 'Streams Pool|Size for
Estimate'
  FORMAT 999999999999
COLUMN STREAMS_POOL_SIZE_FACTOR HEADING 'Streams Pool|Size|Factor' FORMAT
99.9
COLUMN ESTD_SPILL_COUNT HEADING 'Estimated|Spill|Count' FORMAT 99999999
COLUMN ESTD_SPILL_TIME HEADING 'Estimated|Spill|Time' FORMAT 99999999.99
COLUMN ESTD_UNSPILL_COUNT HEADING 'Estimated|Unspill|Count' FORMAT 99999999
COLUMN ESTD_UNSPILL_TIME HEADING 'Estimated|Unspill|Time' FORMAT 99999999.99

SELECT STREAMS_POOL_SIZE_FOR_ESTIMATE,
       STREAMS_POOL_SIZE_FACTOR,
       ESTD_SPILL_COUNT,
       ESTD_SPILL_TIME,
       ESTD_UNSPILL_COUNT,
       ESTD_UNSPILL_TIME
  FROM V$STREAMS_POOL_ADVICE;
```

Monitor Dependency Waits

If there are many wait dependencies when using integrated Replicat, try reducing the value of BATCHTRANSOPS.

```
SELECT APPLY_NAME, STATE, ACTIVE_SERVER_COUNT, TOTAL_WAIT_DEPS,
TOTAL_WAIT_COMMITS FROM V$GG_APPLY_COORDINATOR;
```

Process Memory Usage

gg_memory_usage.sh

```
#!/bin/bash

################################
# determine the OS type
################################
OSNAME=`uname`

case "$OSNAME" in
  "SunOS")
    echo "OSNAME = $OSNAME"
    ;;
  "Linux")
    echo "OSNAME = $OSNAME"
    ;;
  "*")
    echo "This script has not been verified on $OSNAME"
    exit 1
    ;;
esac

################################
# set the temp file
################################
TMPFILE=/tmp/pmem.tmp
if [ -f $TMPFILE ]
then
  rm -f $TMPFILE
fi

################################
# loop over the gg process types
################################
PROCESSES="extract replicat"

for PROCESS in $PROCESSES
do
  FLAG=""
  FLAG=`ps -ef | grep $PROCESS | grep -v grep`
  if [ -z "$FLAG" ]
  then
    echo "No $PROCESS processes found"
  else
    echo
    echo "#####################################"
    echo "# Individual $PROCESS Process Usage #"
    echo "#####################################"
```

```
    case "$OSNAME" in
      "Linux")
        ps -C $PROCESS -O rss > $TMPFILE
        cat $TMPFILE | grep $PROCESS | awk '{print $2/1024, "MB", $12}'
| sort -k 2
        ;;
      "SunOS")
        ps -efo vsz,uid,pid,ppid,pcpu,args | grep -v grep | grep
$PROCESS > $TMPFILE
        cat $TMPFILE | grep $PROCESS | awk '{print $1/1024, "MB", $8}'
| sort -k 2
        ;;
      "*")
        echo "This script has not been verified on $OSNAME"
        exit 1
        ;;
    esac
    rm -f $TMPFILE

    echo
    echo "#######################################"
    echo "#   Total $PROCESS Process Usage    #"
    echo "#######################################"
    case "$OSNAME" in
      "Linux")
        ps -C $PROCESS -O rss > $TMPFILE
        cat $TMPFILE | grep $PROCESS | awk '{count ++; sum=sum+$2; }
END \
          { print "Number of processes      =",count; \
          print "AVG Memory usage/process =",sum/1024/count, "MB"; \
          print "Total memory usage       =", sum/1024,  " MB"}'
        ;;
      "SunOS")
        ps -efo vsz,uid,pid,ppid,pcpu,comm | grep -v grep | grep
$PROCESS > $TMPFILE
        cat $TMPFILE | awk '{count ++; sum=sum+$1; } END \
          { print "Number of processes      =",count; \
          print "AVG Memory usage/process =",sum/1024/count, "MB"; \
          print "Total memory usage       =", sum/1024,  " MB"}'
        ;;
      "*")
        echo "This script has not been verified on $OSNAME"
        exit 1
        ;;
    esac
    rm -f $TMPFILE
  fi
done

exit 0
```

Output

```
OSNAME = Linux

#####################################
# Individual extract Process Usage #
#####################################
41.7109 MB DP_CLI
31.9805 MB DP_FMS
29.9297 MB DP_MR
37.3672 MB DP_MR_O
37.8438 MB DP_SDE
29.1562 MB DP_SERV
1516.69 MB EXT_CLI
74.8516 MB EXT_FMS
556.352 MB EXT_MR
246.926 MB EXT_MR_O
68.7773 MB EXT_SDE
91.5234 MB EXT_SERV

#####################################
#   Total extract Process Usage    #
#####################################
Number of processes      = 12
AVG Memory usage/process = 230.259 MB
Total memory usage       = 2763.11  MB
No replicat processes found
```

Apply Server Details

```
SELECT APPLY_NAME, APPLY#, SERVER_ID, STATE, TOTAL_MESSAGES_APPLIED FROM
V$GG_APPLY_SERVER ORDER BY APPLY_NAME, SERVER_ID;
```

Apply Coordinator Detail

```
SELECT APPLY_NAME, STATE, TOTAL_APPLIED, TOTAL_ASSIGNED, TOTAL_ROLLBACKS,
TOTAL_ERRORS, ACTIVE_SERVER_COUNT, TOTAL_WAIT_DEPS, TOTAL_WAIT_COMMITS FROM
V$GG_APPLY_COORDINATOR;
```

Queue Status

```
SELECT DST_QUEUE_SCHEMA, DST_QUEUE_NAME, TOTAL_MSGS, SPID, STATE FROM
V$PROPAGATION_RECEIVER;
```

Apply Latency

```
SELECT r.apply_name, 86400 *(r.dequeue_time -c.lwm_message_create_time)
latency_in_seconds FROM GV$GG_APPLY_READER r, GV$GG_APPLY_COORDINATOR c
WHERE r.apply# = c.apply# AND r.apply_name = c.apply_name;
```

Trace Monitoring

under OGG installation home directory, create an xml file named gglog-<EXT_OR_REP_NAME>.xml
gglog-<EXT_OR_REP_NAME>.xml contents:

```
<?xml version="1.0"?>
<configuration reset="true">
<appender name="myTraceFile" class="FileAppender">
<param name="File" value="myTrace-%I.log"/>
<param name="Append" value="true"/>
<layout class="PatternLayout"/>
</appender>
<logger name="ggstd.ptrace">
<level value="all"/>
</logger>
</configuration>
```