



# Monitoring Cache Transfers in Real Application Clusters

By Murali Vallath

**A** cash transfer from the United States to India can take days before a confirmation is received due to currency conversion and the necessary additional verifications. Within the information technology world, cache transfer refers to movement of contents in memory from one system to another. The cache transfer between two Oracle instances in a Real Application Cluster (RAC) implementation should not take days or even minutes. Monitoring this activity is critical.

In a RAC database cache transfer depends on few factors, such as interconnect bandwidth, the latency, storage sub-system configuration, user behavior and distribution. The primary factor of importance is the cluster interconnect because Oracle relies on the cluster interconnect to transfer cache between instances. The data usage pattern and the data distribution on disk such as data partitioning are the next most important.

## Cluster Interconnect

This is of primary importance as interconnects that are slow in speed probably cause high network latency and therefore slower cache transfer. Latencies on the cluster interconnect could be caused by:

- Large number of processes in the run queues waiting for CPU or scheduling delays
- Platform specific O/S parameter settings that affect IPC buffering or process scheduling
- Slow, busy or faulty interconnects

This means even if the interconnect is of a high speed such as a 1G Ethernet there could be other limitations defined by the protocol. For example, on Sun Solaris that uses the UDP protocol there is an O/S limitation, which is 64K buffer size. This means for transferring 300K blocks it would take five round trips. If the buffer size were set to 8K, then the number of round trips would be even more. The buffer size could be set to a much higher value on other hardware platforms such as HP-UX and HP-Tru64

Cluster interconnect protocol information on certain hardware platforms could be verified using the oradebug utility from SQLPLUS.

```
SQL>ORADEBUG SETMYPID
SQL>ORADEBUG IPC
SQL>EXIT
```

The following is an extract from the trace file pertaining to the interconnect protocol. The output confirms that the cluster interconnect is being used for instance-to-instance message transfer.

```
SSKGXPT 0x3671e28 flags SSKGXPT_READPENDING info for network 0
socket no 9 IP 142.23.153.1 UDP 59084
sflags SSKGXPT_WRITESSKGXPT_UP
info for network 1
socket no 0 IP 0.0.0.0 UDP 0
sflags SSKGXPT_DOWN
context timestamp 0x4402d
no ports
```

The above output is from a Sun 4800 and indicates the IP address and that the protocol used is UDP. On certain operating systems such as Tru64 the trace output does not reveal the Cluster interconnect information.

The number of packets or blocks transferred across the interconnect is also dependent on the initialization parameter `DB_BLOCK_MUTLIBLOCK_READ_COUNT`. The higher the value of this parameter, the more the blocks that are read from the instance cache during a cache transfer operation.

The STATSPACK report is also a good source of information to determine the interconnect latency. For example the following extract indicates timeouts during 'gcs remote message' transfer.

*continued on page 38*

Event	Waits	Timeouts	Total Wait Time (s)	wait (ms)	Avg Waits /txn
gcs remote message	391,629	377,590	7,017	18	2,163.7
ges remote message	77,311	74,159	3,488	45	427.1
gcs remote message	209,188	186,286	3,505	17	817.1
ges remote message	77,257	73,797	3,489	45	301.8
gcs remote message	599,108	323,177	6,943	12	303.5
ges remote message	81,552	72,571	3,485	43	41.3

The output above indicates high timeouts between remote message transfers. Also, the number of waits and the waits per transaction is significantly high.

### Monitoring Cache Transfer

Oracle provides various views to measure the cache transfer activity across the cluster interconnect, the view of primary importance is GV\$CACHE\_TRANSFER.

This view displays types and classes of blocks that Oracle has transferred over the cluster interconnect at least once. It contains information from the block header of each block in the SGA of the current instance. For example, it represents a block in the buffer cache of the current instance. These can be used to help identify which blocks are being pinged between instances using the XNC column, which shows the number of lock conversions from exclusive to NULL status. These conversions represent potential pings. This view only shows buffers with a nonzero XNC count.

If the NAME column is blank it indicates that the buffer is associated with a temporary segment. The data in the XNC column is important in this view as it maintains the count on the block level locks. Each block starts with an XNC value of zero when it first enters the buffer cache. This value is incremented each time the instance releases the lock covering that block.

Cache transfer activity across the instances could be observed using the following set of queries. The query below will help identify the objects that have high cache transfer activities between instances. It displays contention statistics of buffers that are currently in the buffer cache of the corresponding instance. This could be identified with objects that have a high number of exclusive to NULL conversions.

```
SELECT
  INST_ID,
  NAME,
  FILE#,
  CLASS#,
  MAX(XNC)
FROM GV$CACHE_TRANSFER
GROUP BY INST_ID,
  NAME,
  FILE#,
  CLASS#
/
```

From the output below, its clear that the object COMPANY is the possible source for high cache transfer activity.

INST_ID	NAME	FILE#	CLASS#	MAX(XNC)
1	IDL_UB2\$	1	4	231
1	PK_USPRL	4	1	47
1	PK_COMP	4	1	39
1	COMPANY	171	1	2849

Using the query below on the GV\$CACHE\_TRANSFER will help identify the frequency of lock conversions and the block related information in the COMPANY table.

```
SELECT FILE#,
  BLOCK#,
  CLASS#,
  STATUS,
  XNC
FROM GV$CACHE_TRANSFER
WHERE NAME = 'COMPANY'
AND FILE# = 171
/
```

The output below, displays the frequency of lock conversions for the object found in the previous query namely the COMPANY table.

FILE#	BLOCK#	CLASS#	STAT	XNC
171	898	1	XCUR	1321
171	1945	1	XCUR	27
171	1976	1	XCUR	19
171	2039	1	XCUR	849

Drilling down further will help identify the actual rows that are being transferred frequently across the cluster interconnect, the query below is used to display the rows in the block found in the previous query. The DBMS\_ROWID package is used to extract the block number from the ROWID pseudocolumn. Monitoring the activity or the data being transferred also helps identify values that may be partitionable to avoid future contention. If the data cannot be partitioned and the data in these identified tables is updated frequently, then another alternative to reduce contention will be to reduce the number of rows per block to spread out the I/O activity across different blocks.

```
SELECT COMP_ID,
  COMP_NAME
FROM COMPANY
WHERE DBMS_ROWID.ROWID_BLOCK_NUMBER(ROWID) = 898
/
```

The output below displays the rows contained in block 898.

COMP_ID	NAME
3949	SUMMERSKY DB CONSULTANTS
3952	CATAMARAN INC.
3957	PRIYAR BROTHERS INC.
3961	DIGITAL BROADCASTING INC.

## Conclusion

RAC relies on the cluster interconnect speed to provide good performance. The smaller the interconnect buffer size parameter then the fewer the number of rows transferred across the cluster interconnect. This means more round trips to complete the transaction resulting in hindered performance.

Oracle provides dynamic performance views to track and monitor cache transfer activity across the cluster interconnect. These views help analyze user access patterns across the instances, which in turn helps determine the best data distribution strategy.



## About the Author

**Murali Vallath** is an Oracle certified database administrator with more than 16 years of experience designing and developing databases. He currently works as a senior database architect at Elogex Inc. ([www.elogex.com](http://www.elogex.com)) in Charlotte, N.C., USA. His primary focus is designing and performance tuning of Oracle databases and his specialty is OPS/RAC. He has presented at IOUG and various other national and international conferences and has published papers both externally and internally to his employers.

He is the author of an upcoming book titled *Oracle Real Application Clusters* (Publisher: Digital Press ([www.digitalpressbooks.com](http://www.digitalpressbooks.com))). Murali can be reached at [mvallath@elogex.com](mailto:mvallath@elogex.com) or at [muralivallath@hotmail.com](mailto:muralivallath@hotmail.com).