hotsos

Optimizing RAC
Performance & Architecture with Method R & more

Cache Fusion, Parallel Query, Oracle Clusterware, GCS, Interconnect,

hotsos

Hotsos the company…

- Thought leadership
  - *Optimizing Oracle Performance*
  - *www.hotsos.com* Library
  - Method R

- Software Solutions
  - Hotsos Profiler
  - Laredo
  - HAWCS

hotsos p5

- Consulting Services
  - Rapid performance assessment
  - On-site consulting
  - Remote consulting
- Education
  - Oracle performance curriculum
  - Public and private events
  - Hotsos Symposium

hotsos symposium 2006 march 5-9 dallas

hotsos

Overview

- Undo & Cache Fusion
- Parallel Query Executions – DOP & IIP
- iDLM
- Lock Elements
- CRS, GES & GCS
- CLUVFY
- Clusterware File Mirroring
- Interconnect Architecture
- Network Myth
- Network Configurations
- Network Interface Cards – NICs

hotsos

According to Webster….

efficient: **ef·fi·cient**
Pronunciation: i-'fi-sh&nt
Function: *adjective*

1 : being or involving the immediate agent in producing an effect <the *efficient* action of heat in changing water to steam>

2 : productive of desired effects; *especially* **:** productive without waste

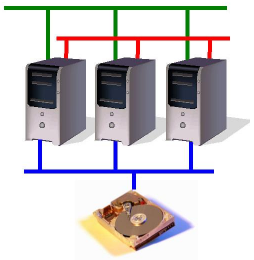
hotsos

RAC Physical Architecture

Public

Interconnect

SAN/NAS



hotsos

CLUster VeriFY Utility (CLUVFY or CVU)

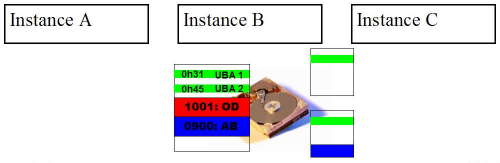
- Verifies CRS & RAC installs are done correctly
- 10gR2 Utility however will work 10gR1 with "–r 10gR1 "
- Available on Metalink & on 10gR2 DVD
- CVU FAQ Note : 316817.1
- CRS Installed:
  - cd $ORACLE_HOME/bin
  - script cluvfy.log ### run the following as oracle software owners id
  - cluvfy stage -post crsinst -n all -verbose
- CRS Not Installed
  - cd /Disk1/cluvfy
  - ./runcluvfy stage -pre crsinst -n <node1>,<node2>
- Consider command line "–n all" to find all nodes in a cluster vs. using individual node names
- May error with EMC Powerpath
- RedHat Linux requires installation of /Disk1/cluvfy/*cvuqdisk-1.0.1-1.rpm*

Clusterware File Mirroring

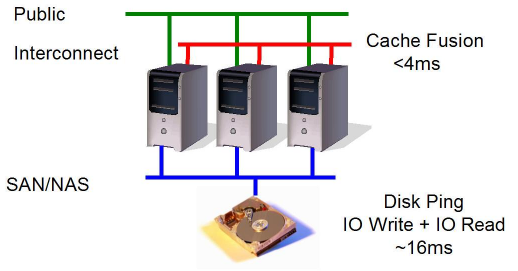
- Prior to 10gR2, two little known Single Point Of Failures for CRS
  - OCR (formally aka SRVM)
  - Voting Disk (aka CSS file)
  - Both of which were SPOFs for RAC unless SAN mirroring (RAID 1) was used to protect them both
  - Lose of either will cause the cluster to fail
- 10gR2 New Feature
  - OCR can now be mirrored, 2 copies max
  - Voting Disk can be mirrored, 3 copies max
- This was an Achilles heel to Oracle CRS vs. some 3rd party
- crsd backups OCR file
  - ocrconfig -showbackup
  - also recommend backing up of voting with dd using 4k block

Consistent Reads

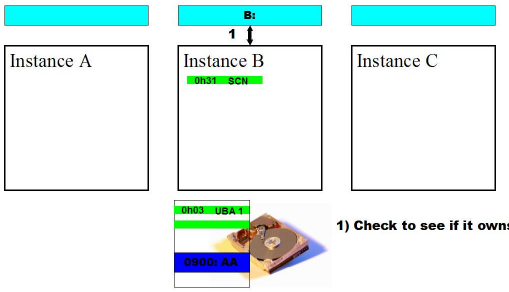
- Provides for transactional consistency across RAC instances
- Numerous CR blocks can exist for a single on db block
  - Two different users are connect to Instance A & C both update same block different rows
  - User on Instance B reading rows via index range scan!



OPS Hard/Soft Ping vs. Cache Fusion



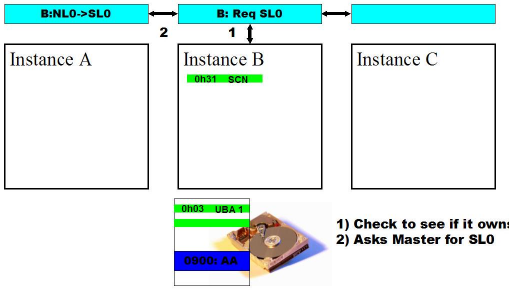
Cache Fusion Block Read – Step 1



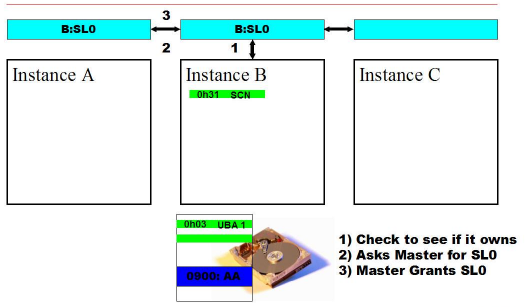
Cache Fusion Introduction

- Resource Locking Mechanism
- Resource Messaging
- Resource Transfer Protocol
- Before DB Block Read
  - Server process check to see if that block resource lock is already owned by the local instance
  - Resource Lock Modes
    - NL0=Null Local (N)
    - SL0=Shared Local (S)
    - XL0=eXclusive Local (X)
    - SG0 = Shared Global, instance owns current block image
    - XG0 = eXclusive Global, instance owns current block image
    - NG1 = Null Global, instance owns past block image 1
    - SG1 = Shared Global, instance owns past block image 1
    - XG1 = eXclusive Global, instance owns past block image 1
    - Past Images Indication can be 0, 1 or 2

Cache Fusion Block Read – Step 2

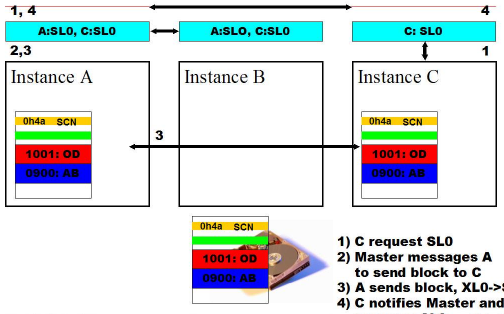


Cache Fusion Block Read – Step 3



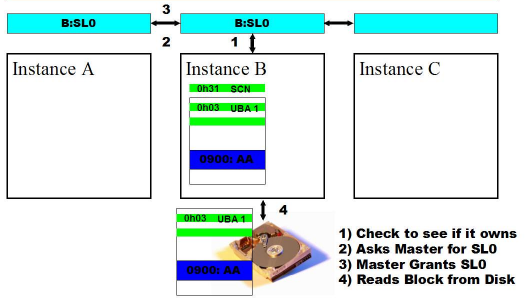
1) Check to see if it owns
2) Asks Master for SL0
3) Master Grants SL0

Answer



1) C request SL0
2) Master messages A to send block to C
3) A sends block, XL0->SL0
4) C notifies Master and assumes SL0

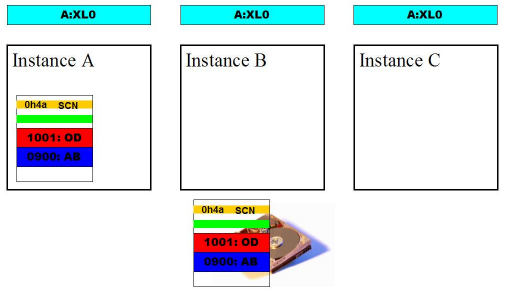
Cache Fusion Block Read - Basic steps



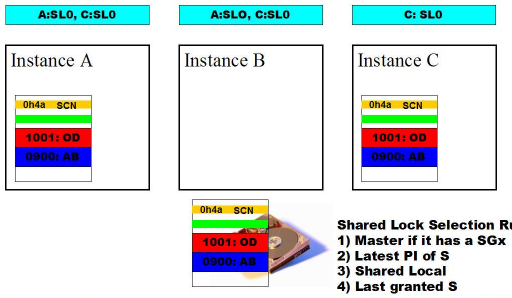
1) Check to see if it owns
2) Asks Master for SL0
3) Master Grants SL0
4) Reads Block from Disk

Question: What happens if B wants to read block?

Question: What happens if C wants to read block?

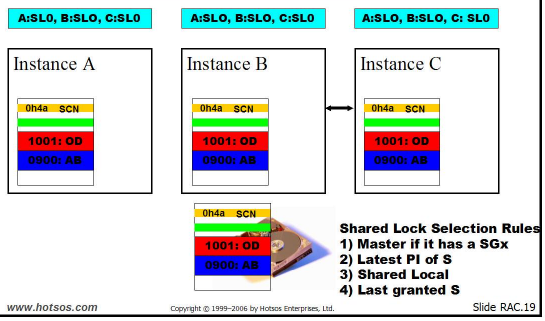


Question: What happens if B wants to read block? Hint



Shared Lock Selection Rules
1) Master if it has a SGx
2) Latest PI of S
3) Shared Local
4) Last granted S

Answer: C sends the block to B at the Masters request



Network Myths

- As a DBA, I cannot see network issues.
- DBA's cannot do anything to address network issues
- Network Engineer's don't help
- The network is fine, that network segment is only 5% utilized
- Your running on a Gig-E & No one needs more than 100mbps
- The server is on the backbone, you cannot get better than that.
- All the cards are Gig-E by the same manufacture, one NIC cannot be causing an issue
- What do you mean, Erwin wants you to go buy three $100 switches @ Best Buy for RAC to replace our corporate $130,000 switch?
- Transparent Application Failover

Lock Elements, Resources & Enqueues

- v$lock_type
- Resources
  - v$ges_resource
  - v$dlm_ress
- Enqueues

Global Cache Services (Cluster Ready Services)

- GCS Resource = x$kjbr & x$kjbrfx
- GCS Enqueues = x$kjbl & x$kjblfx
- Dump GCS with:

```
alter session set events 'immediate trace name gc_elements level 1'
```


Oradebug Investigation of Lock Mastering

- oradebug lkdebug
  - sysdba privilege required

```
SQL> ORADEBUG LKDEBUG HELP
Usage:lkdebug [options]
    -l [r|p] <enqueue pointer>   Enqueue Object
    -r <resource pointer>        Resource Object
    -b <gcs shadow pointer>      GCS shadow Object
    -p <process id>              client pid
    -P <process pointer>         Process Object
    -O <lt> <l2> <types>         Oracle Format resname
    -a <res/lock/proc>           all <res/lock/proc> pointer
    -A <res/lock/proc>           all <res/lock/proc> contexts
    -a <res> [<type>]            all <res> pointers by an optional type
    -a convlock                  all converting enqueue (pointers)
    -A convlock                  all converting enqueue contexts
    -a convres                   all res ptr with converting enqueues
    -A convres                   all res contexts with converting enqueues
    -a name                      list all resource names
    -a hashcount                 list all resource hash bucket counts
    -t                           Traffic controller info
    -s                           summary of all enqueue types
    -k                           GES SGA summary info
    -m pkey <objectno>           request for remastering this object at current instance
    -m dpkey <objectno>          request for dissolving remastering of this object at
                                 current instance
```


Dynamic Lock Remastering

- v$gcspfmaster_info
- Instances start with 0
- Initially empty
- To force remastering of objects to local instance
  - **ORADEBUG LKDEBUG -m pkey 12345 <- object #**
- To force remastering of objects to be distributed instance
  - **ORADEBUG LKDEBUG -m dpkey 12345 <- object #**
- v$rowcache
  - dlm_requests
  - dlm_conflicts
  - dlm_releases

hotsos

So what can cause lock master issues?

- Hot Blocks
- Non-Cached Sequences
- Index Blocks
- Lock Conversion Times
  - Interconnect Contention
  - Interconnect Traffic

hotsos

Profiler Output

	Oracle timed event	Duration (seconds)		Cumulative duration (seconds)		Event count	Duration per event (seconds) mean	min	skew	max	Drill-down
1.	**CPU service, EXEC calls**	**1,133.680**	**45.2%**	1,133.680	45.2%	**89,060**	0.012616	0.000000		6.500000	hits
2.	**db file sequential read**	**432.030**	**17.2%**	1,565.710	62.4%	62,495	**0.006913**	0.000000		1.790000	hits
3.	**CPU service, FETCH calls**	**390.720**	**15.6%**	1,956.430	77.9%	**63,925**	0.006112	0.000000		4.190000	hits
4.	unaccounted-for	203.550	8.1%	2,160.980	86.1%	1	209.550000	209.550000		209.550000	hits
5.	global cache lock s to x	99.870	4.0%	2,260.850	90.1%	3,434	0.029083	0.000000		0.780000	hits
6.	global cache lock open s	85.930	3.4%	2,351.780	93.7%	3,567	0.024501	0.000000		0.830000	hits
7.	global cache lock open x	57.880	2.3%	2,409.660	96.0%	1,930	0.029990	0.000000		0.960000	hits
8.	latch free	24.770	1.1%	2,434.430	97.0%	1,010	0.026505	0.000000		0.300000	hits
9.	SQL*Net message from client [IPC/BEQ driver]	19.110	0.8%	2,453.540	97.8%	6,714	0.002846	0.000000		0.460000	hits
10.	write complete waits	11.130	0.4%	2,464.670	98.3%	155	0.071806	0.000000		0.370000	hits
11.	row cache lock	11.100	0.4%	2,477.770	98.7%	405	0.022667	0.000000		0.670000	hits
12.	enqueue	11.090	0.4%	2,488.860	99.1%	300	0.033606	0.000000		0.440000	hits
13.	log file switch completion	7.310	0.3%	2,496.170	99.4%	15	0.487333	0.000000		1.010000	hits
14.	log file sync	3.210	0.1%	2,499.400	99.6%	39	0.084872	0.000000		0.430000	hits
15.	CPU service, PARSE calls	3.110	0.1%	2,502.590	99.7%	4,472	0.000695	0.000000		0.060000	hits
16.	wait for DLM latch	2.950	0.1%	2,505.540	99.8%	91	0.032418	0.010000		0.400000	hits
17.	global cache lock busy	1.530	0.1%	2,507.070	99.9%	11	0.139091	0.020000		0.280000	hits
18.	DFS lock handle	1.400	0.1%	2,508.470	99.9%	43	0.032558	0.000000		0.490000	hits
19.	global cache lock null to x	0.900	0.0%	2,509.370	100.0%	8	0.112500	0.040000		0.310000	hits
20.	rdbms ipc reply	0.570	0.0%	2,509.940	100.0%	7	0.081429	0.010000		0.110000	hits
21.	global cache lock null to s	0.420	0.0%	2,510.360	100.0%	7	0.060000	0.050000		0.100000	hits
22.	library cache pin	0.110	0.0%	2,510.470	100.0%	7	0.015714	0.010000		0.020000	hits
23.	SQL*Net message to client	0.020	0.0%	2,510.490	100.0%	6,714	0.000003	0.000000		0.010000	hits
24.	file open	0.000	0.0%	2,510.490	100.0%	13	0.000000	0.000000		0.000000	hits
25.	SQL*Net more data from client	0.000	0.0%	2,510.490	100.0%	2	0.000000	0.000000		0.000000	hits
26.	Total measured response time	2,510.490	100.0%								

hotsos

Interconnect Architectures & Realities

```
WAIT #1: nam='gc buffer busy' ela= 44459 p1=9 p2=1380 p3=65537
WAIT #1: nam='gc buffer busy' ela= 32022 p1=9 p2=1380 p3=65537
WAIT #1: nam='gc buffer busy' ela= 99622 p1=9 p2=1364 p3=65537
WAIT #1: nam='gc buffer busy' ela= 99622 p1=9 p2=1356 p3=65537
WAIT #1: nam='gc buffer busy' ela= 973216 p1=9 p2=1381 p3=65537
WAIT #1: nam='gc buffer busy' ela= 5893 p1=9 p2=1364 p3=65537
WAIT #1: nam='gc buffer busy' ela= 99713 p1=9 p2=1364 p3=65537
```

- ela is 32ms to 99.7ms
- Is this really a GC Buffer Busy or is it GC transfer time?

hotsos

Interconnect Architectures & Realities - cont

- GigE NICs for Interconnect
  - Realistic 670mbps per nic
  - MTU1500
  - 8192 Block Size
  - RAC Over Head - Send
  - RAC Over Head – Ack
  - ~1500 blocks per second
- What happens when the auto 10/100/1000 NIC chooses 100mbps?
- Jumbo Frame impact?
- Moving from TCP/IP to either LLT or UDP?
  - Let's do the math, shall we.

hotsos

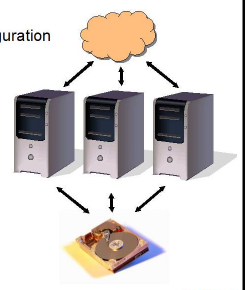
Cache Fusion Notes

- Master (iDLM) knows SCN of all versions of blocks
- Past Images are referred to with SCN
- Undo still works just like normal, in addition to DLM
- Trace Options
  - 10432
    - level 1 : debugging DLM
    - level 4 : cache fusion messages
    - level 8 : traces all GCS calls
  - 10704 level 10 : ksq - Kernel enQueue Service
  - 10706 level 10 : ksi – Kernel Service Instance locks
  - 10708 level 1 : GCS lock elements
  - 10254 level 10 : trace calls across all instances
  - 29700 to 29722 : in orau.msg - Use at your own peril !

hotsos

Network Configuration Overview

- 8 CPU Servers
- Small Enterprise 3 Node RAC Configuration
- 300Gb on a EMC DMX
- Oracle ASM
- Backup strategy via NetBackup



How many HW network connections are in this Best Practice configuration?

a) 6
b) 18
c) 24
d) 48

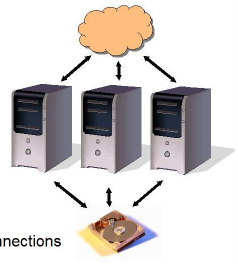
Network Configuration Overview – cont.

48

- HBAs going to DMX = 2
- Private Interconnect for RAC = 3
- Private backup network = 1
- Public Network = 2

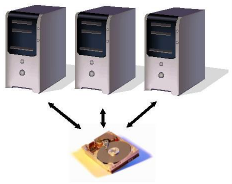
((2+3+1+2)*3)=24

Each NIC/HBA has two physical HW connections



Host Based Adapter - HBAs

- Modern FDDI
- GigE or 2 GigE
- Typically 2 per node

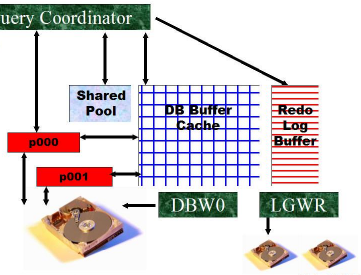


Cache Fusion Interconnect Configuration

- Typically default IP packet frame has a MTU: 1500 bytes
  - Small IP TCP & UDP Messaging
  - MTU:1500 was selected for SMTP
    - (i.e. simple ASCII text emails)
- Oracle DB Blocks are 2048 or larger… most likely 8192
- Small MTU requires more IP communication transfers between servers
- An MTU of 1500 requires at least 6 UDP messages to handle a single 8192 byte DB block transfer
- Use Jumbo IP frames
  - MTU:9000
  - Ensure all network path HW supports the increased MTU

Overview of Parallel Query – PQ

- Parallelization of a SQL command on a instance
- Default = 1
- Takes advantage of SMP
- CPU Concurrency
  - HyperThreading
  - Dual Core
- Messaging via IPC
- Table Queues (TQ)

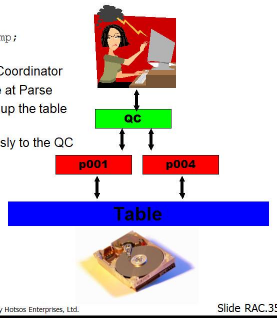


Parallel Query Architecture

- Degree of Parallelization (DoP)

```
select /+ parallel (emp,2) +/ from emp;
```

- Users server process become a Query Coordinator
- PARSE: If 2 PQ processes are available at Parse
- EXEC: Then 2 PQ processes will divide up the table and do a parallel execution
- FETCH: returning the data asynchronously to the QC
- Messaging via IPC
- Table Queues (TQ) used in sorting



Parallel Query Parameters

NAME	TYPE	VALUE
parallel_adaptive_multi_user	boolean	TRUE
parallel_automatic_tuning	boolean	FALSE
parallel_execution_message_size	integer	2148
parallel_instance_group	string	
parallel_max_servers	integer	0
parallel_min_percent	integer	0
parallel_min_servers	integer	0
parallel_server	boolean	FALSE
parallel_server_instances	integer	1
parallel_threads_per_cpu	integer	2

Parallel Query Execution Parameters

- parallel_adaptive_multi_user boolean TRUE
  - As the workload increases, Oracle will reduce DOP (9i default was FALSE)
- parallel_automatic_tuning boolean FALSE
  - Automagically tunes PQ (9i default was TRUE)
- parallel_execution_message_size integer 2148
  - IPC transfer size in bytes
- parallel_max_servers integer 0
  - Maximum amount of PQ server to allow to start
- parallel_min_percent integer 0
  - Minimum percent of servers to obtain, if available PX server are below this at PARSE, statement will be serial
- parallel_min_servers integer 0
  - Minimum number of servers to start at instance startup
- parallel_threads_per_cpu integer 2
  - Just as name states, used with p_a_m_u & p_a_t

Parallel Query Execution Plan

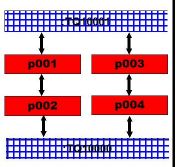
```
select sum(salary) from emp group by dep_id;
```

```
| Id | Operation                | Name     | Rows | Bytes | Cost (%CPU) | TQ     |IN-OUT| PQ Distrib |
| 0 | SELECT STATEMENT         |          |  107 |  2782 | 3 (34)      |        |      |            |
| 1 |  PX COORDINATOR          |          |      |       |             |        |      |            |
| 2 |   PX SEND QC (RANDOM)    | :TQ10001 |  107 |  2782 | 3 (34)      | Q1,01  | P->S | QC (RAND)  |
| 3 |    HASH GROUP BY         |          |  107 |  2782 | 3 (34)      | Q1,01  | PCWP |            |
| 4 |     PX RECEIVE           |          |  107 |  2782 | 3 (34)      | Q1,01  | PCWP |            |
| 5 |      PX SEND HASH        | :TQ10000 |  107 |  2782 | 3 (34)      | Q1,00  | P->P | HASH       |
| 6 |       HASH GROUP BY      |          |  107 |  2782 | 3 (34)      | Q1,00  | PCWP |            |
| 7 |        PX BLOCK ITERATOR |          |  107 |  2782 | 2 (0)       | Q1,00  | PCWP |            |
| 8 |         TABLE ACCESS FULL| EMP      |  107 |  2782 | 2 (0)       | Q1,00  | PCWP |            |
```

- Table is set to Parallel 2
- PCWP = Parallel Child With Parent
- P->P = Parallel to Parallel
- P->S = Parallel to Serial

Parallel Query Execution Plan – cont.

```
| Id | Operation                | Name     | TQ     |IN-OUT| PQ Distrib |
| 0 | SELECT STATEMENT         |          |        |      |            |
| 1 |  PX COORDINATOR          |          |        |      |            |
| 2 |   PX SEND QC (RANDOM)    | :TQ10001 | Q1,01  | P->S | QC (RAND)  |
| 3 |    HASH GROUP BY         |          | Q1,01  | PCWP |            |
| 4 |     PX RECEIVE           |          | Q1,01  | PCWP |            |
| 5 |      PX SEND HASH        | :TQ10000 | Q1,00  | P->P | HASH       |
| 6 |       HASH GROUP BY      |          | Q1,00  | PCWP |            |
| 7 |        PX BLOCK ITERATOR |          | Q1,00  | PCWP |            |
| 8 |         TABLE ACCESS FULL| EMP      | Q1,00  | PCWP |            |
```

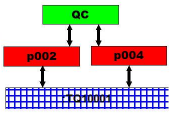


- EMP is scanned in Parallel (p002 & p004)
- Based on the PX Block Iterator split (remember 9/13th)
- P002 & p004 pass the row source to p001 & p003 respectively
- P001 & p002 do the HASH GROUP BY aggregating into Table Queue :TQ10000

Parallel Query Execution Plan – cont.

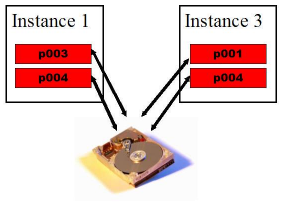
```
| Id | Operation                | Name     | TQ     |IN-OUT| PQ Distrib |
| 0 | SELECT STATEMENT         |          |        |      |            |
| 1 |  PX COORDINATOR          |          |        |      |            |
| 2 |   PX SEND QC (RANDOM)    | :TQ10001 | Q1,01  | P->S | QC (RAND)  |
| 3 |    HASH GROUP BY         |          | Q1,01  | PCWP |            |
| 4 |     PX RECEIVE           |          | Q1,01  | PCWP |            |
| 5 |      PX SEND HASH        | :TQ10000 | Q1,00  | P->P | HASH       |
| 6 |       HASH GROUP BY      |          | Q1,00  | PCWP |            |
| 7 |        PX BLOCK ITERATOR |          | Q1,00  | PCWP |            |
| 8 |         TABLE ACCESS FULL| EMP      | Q1,00  | PCWP |            |
```

- Slaves again work in parallel (P->S) sending ROW in hashed :TQ1,01 to a single Query Coordinator
- PX Coordinator is most likely users dedicated server process
- PX Coordinator sends results back to user client serially



Inter-instance Parallelism vs. Query Parallelization

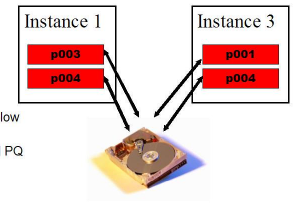
- Inter-Instance Parallelism (IIP)
- Determined @ Execution plan build
- At least 1 PQ server processes on 2 instances
  - across multiple instances
- Degree of Parallelization
- PARALLEL 1 DEGREE 4
- PARALLEL 4 DEGREE 4
- PARALLEL 4 DEGREE 20
- PARALLEL 2 DEGREE 2



hotsos
www.hotsos.com

Typical Parallelism & Parallelization Issues

- PARALLEL_MAX_SERVERS set to large
- On non-MT CPU systems, consider no more than 6 to 8 per CPU
  - See "Magic of "2" by C. Milsap
- On MT CPU systems, maybe 8 or 10
- Trying to get non parallel
  - Hint
  - Set on TABLE
  - Set on Table Space
  - Set Default
- If high PX used, occasionally someone will report a query as slow due to lack of available slaves
- Metalink Note 201799.1 review all PQ parameters & settings



hotsos
www.hotsos.com

ORACLE-RAC SIG Mailing List & Forum Information

- ORACLE-RAC@yahoogroups.com
- http://groups.yahoo.com/group/ORACLE-RAC

- Post message:ORACLE-RAC@yahoogroups.com
- Subscribe:ORACLE-RAC-subscribe@yahoogroups.com
- Unsubscribe:ORACLE-RAC-unsubscribe@yahoogroups.com
- List owner:ORACLE-RAC-owner@yahoogroups.com