并行不悖——Oracle数据库并行的是是非非

杨廷琨

个人介绍

ORACLE ACE Director

ACOUG
All China Oracle User Group
中国 Oracle 用户组

- 杨廷琨(yangtingkun)
  - Oracle ACE Director
  - ITPUB数据库管理区版主
  - ACOUG副总裁
  - 参与编写《Oracle数据库性能优化》、《Oracle DBA手记》、《Oracle DBA手记3》和《Oracle性能优化与诊断案例精选》
  - 十九年的DBA经验
  - 个人BLOG中积累了2500篇原创技术文章
  - 云和恩墨CTO

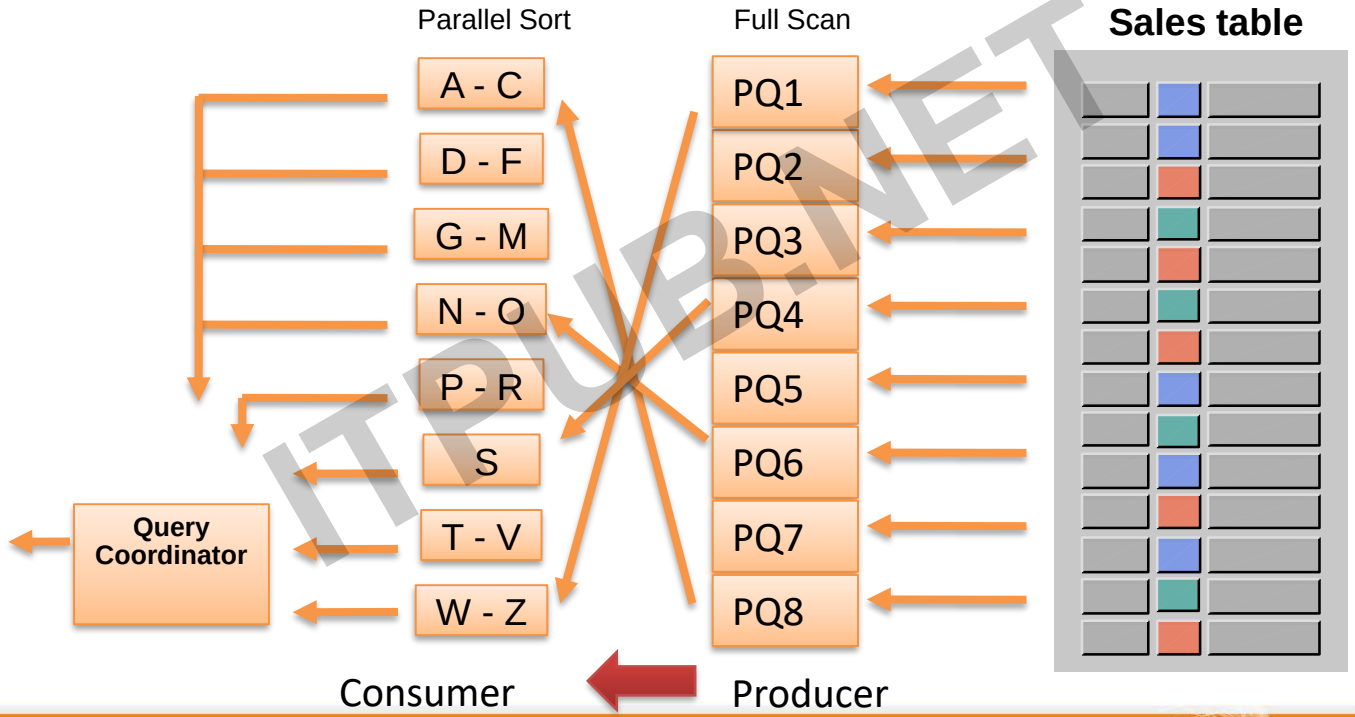
- 并行的基础概念
- 并行的发展演进
- 并行的不只是SQL
- 并行的常见问题

并行概述

- 单进程操作分拆由多个进程同时运行
- 充分利用主机CPU、IO能力
- 适用于OLAP系统
- 适用于OLTP的后台批处理
- 优化的最后手段

并行概述

```
SELECT * FROM sales s ORDER BY name;
```



并行概述

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT		99772	9840K	26 (4)	00:00:01			
1	PX COORDINATOR								
2	PX SEND QC (ORDER)	:TQ10001	99772	9840K	26 (4)	00:00:01	Q1,01	P->S	QC (ORDER)
3	SORT ORDER BY		99772	9840K	26 (4)	00:00:01	Q1,01	PCWP	
4	PX RECEIVE		99772	9840K	25 (0)	00:00:01	Q1,01	PCWP	
5	PX SEND RANGE	:TQ10000	99772	9840K	25 (0)	00:00:01	Q1,00	P->P	RANGE
6	PX BLOCK ITERATOR		99772	9840K	25 (0)	00:00:01	Q1,00	PCWC	
7	TABLE ACCESS FULL	SALES	99772	9840K	25 (0)	00:00:01	Q1,00	PCWP	

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		99772	9840K	181 (3)	00:00:01
1	SORT ORDER BY		99772	9840K	181 (3)	00:00:01
2	TABLE ACCESS FULL	SALES	99772	9840K	178 (1)	00:00:01

并行概述

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT		99772	9840K	26 (4)	00:00:01			
1	PX COORDINATOR								
2	PX SEND QC (ORDER)	:TQ10001	99772	9840K	26 (4)	00:00:01	Q1,01	P->S	QC (ORDER)
3	SORT ORDER BY		99772	9840K	26 (4)	00:00:01	Q1,01	PCWP	
4	PX RECEIVE		99772	9840K	25 (0)	00:00:01	Q1,01	PCWP	
5	PX SEND RANGE	:TQ10000	99772	9840K	25 (0)	00:00:01	Q1,00	P->P	RANGE
6	PX BLOCK ITERATOR		99772	9840K	25 (0)	00:00:01	Q1,00	PCWC	
7	TABLE ACCESS FULL	SALES	99772	9840K	25 (0)	00:00:01	Q1,00	PCWP	

Q1: Parallel Slave Group 1
00: Parallel Slave Sets 1
01: Parallel Slave Sets 2

并行概述

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT		99772	9840K	26 (4)	00:00:01			
1	PX COORDINATOR								
2	PX SEND QC (ORDER)	:TQ10001	99772	9840K	26 (4)	00:00:01	Q1,01	P->S	QC (ORDER)
3	SORT ORDER BY		99772	9840K	26 (4)	00:00:01	Q1,01	PCWP	
4	PX RECEIVE		99772	9840K	25 (0)	00:00:01	Q1,01	PCWP	
5	PX SEND RANGE	:TQ10000	99772	9840K	25 (0)	00:00:01	Q1,00	P->P	RANGE
6	PX BLOCK ITERATOR		99772	9840K	25 (0)	00:00:01	Q1,00	PCWC	
7	TABLE ACCESS FULL	SALES	99772	9840K	25 (0)	00:00:01	Q1,00	PCWP	

P->S: Paralle to Serial
P->P: Parallel to Parallel
S->P: Serial to Parallel
PCWP: Parallel Combined With Parent
PCWC: Parallel Combined With Child
SCWP: Serial Combined With Parent
SCWC: Serial Combined With Child
R->S: Remote to Serial

并行概述

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT		99772	9840K	26 (4)	00:00:01			
1	PX COORDINATOR								
2	PX SEND QC (ORDER)	:TQ10001	99772	9840K	26 (4)	00:00:01	Q1,01	P->S	QC (ORDER)
3	SORT ORDER BY		99772	9840K	26 (4)	00:00:01	Q1,01	PCWP	
4	PX RECEIVE		99772	9840K	25 (0)	00:00:01	Q1,01	PCWP	
5	PX SEND RANGE	:TQ10000	99772	9840K	25 (0)	00:00:01	Q1,00	P->P	RANGE
6	PX BLOCK ITERATOR		99772	9840K	25 (0)	00:00:01	Q1,00	PCWC	
7	TABLE ACCESS FULL	SALES	99772	9840K	25 (0)	00:00:01	Q1,00	PCWP	

并行概述

- HASH
  - 利用HASH算法进行数据打散
- BROADCAST
  - 将数据集广播方式发给所有消费者
- RANGE
  - 对数据范围打散配合GROUP BY和ORDER BY操作
- KEY
  - 针对键值的逻辑聚集
- ROUND ROBIN
  - 将数据循环发送给所有消费者

- 并行的基础概念
- 并行的发展演进
- 并行的不只是SQL
- 并行的常见问题

跨节点并行

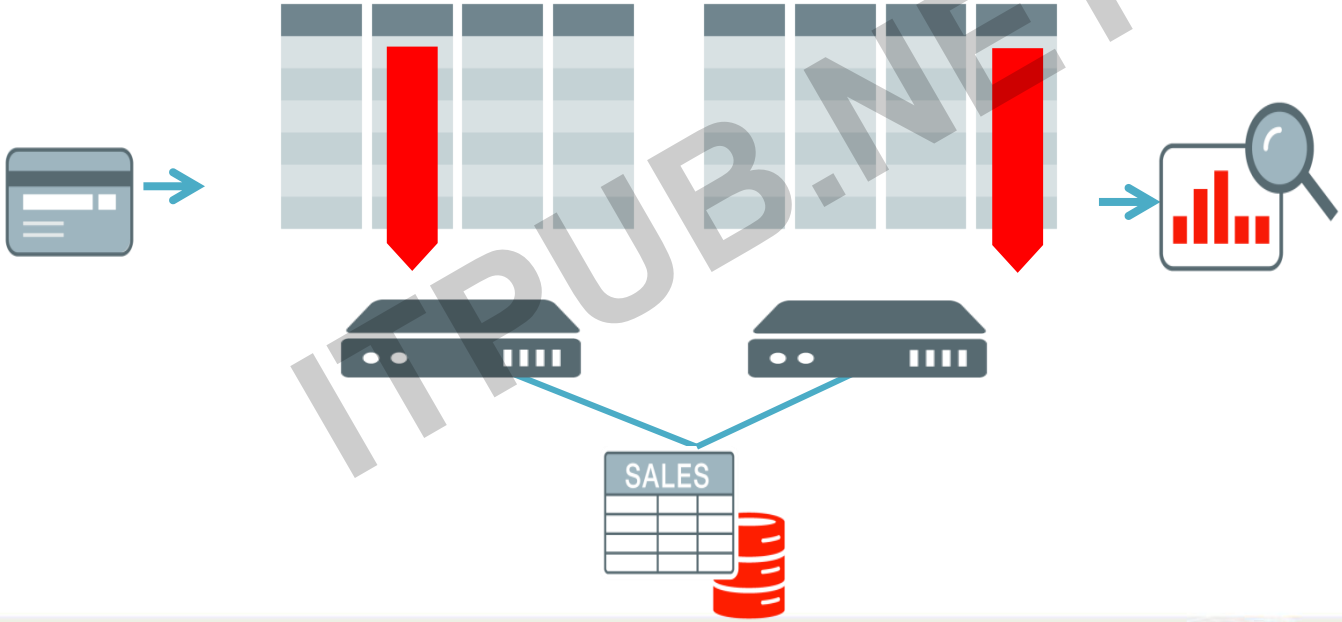
- RAC环境下，并行自动在多个节点同时执行
- 充分利用多个主机的CPU内存资源
- 通过参数parallel_force_local来避免跨节点并行

并行队列

- 传统方式:
  - 并行进程数量不足导致并行操作串行执行
  - 并行进程数量太多导致主机资源耗尽
- 11g特性：
  - 并行进程不足时进入队列

并行与IN MEMORY

RAC下的INMEMORY采用SHARE NOTHING架构



并行与IN MEMORY

- 表数据分布方式
  - 分区

```
SQL> alter table t_part inmemory distribute by partition;

Table altered.
```

  - 范围

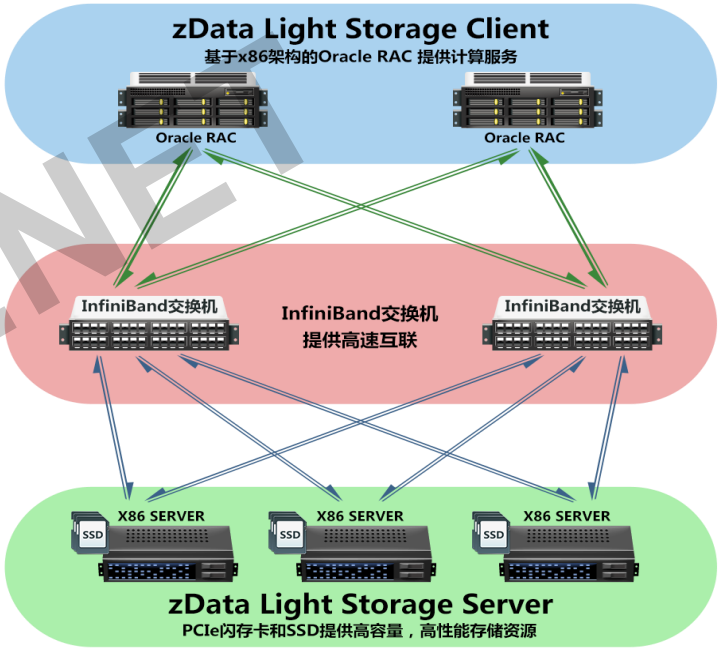
```
SQL> alter table t inmemory distribute by rowid range;

Table altered.
```

  - 自动
- 启用INMEMORY的前提条件
  - alter system set parallel_degree_policy = AUTO
  - alter system set parallel_force_local = TRUE

一体机对并行的支撑

- PCI Flash提供高速低延迟IO能力
- 存储节点提供IO动态扩展能力
- InfiniBand提供高速心跳网络带宽
- RDMA协议进一步降低心跳网络延迟
- zData一体机解决了并行扩展能力的瓶颈



- 并行的基础概念
- 并行的发展演进
- 并行的不只是SQL
- 并行的常见问题

统计信息并行收集

```
SQL> exec dbms_stats.delete_database_stats

PL/SQL procedure successfully completed.

SQL> exec dbms_stats.gather_database_stats

PL/SQL procedure successfully completed.

Elapsed: 00:47:41.54

SQL> exec dbms_stats.delete_database_stats

PL/SQL procedure successfully completed.

SQL> exec dbms_stats.gather_database_stats(degree => 8)

PL/SQL procedure successfully completed.

Elapsed: 00:39:07.87
```

统计信息并行收集

```
SQL> exec dbms_stats.delete_database_stats

PL/SQL procedure successfully completed.

SQL> alter system set parallel_adaptive_multi_user = false;

System altered.

SQL> exec dbms_stats.set_global_prefs('CONCURRENT', 'TRUE')

PL/SQL procedure successfully completed.

SQL> exec dbms_stats.gather_database_stats(degree => 8)

PL/SQL procedure successfully completed.

Elapsed: 00:36:09.94
```

数据泵并行

```
[test@DEVDB ~]$ expdp yangtk directory=d_temp dumpfile=user20190425.dp

Export: Release 11.2.0.4.0 - Production on Thu Apr 25 14:41:58 2019

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.
Password:

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Starting "YANGTK"."SYS_EXPORT_SCHEMA_01":  yangtk/******** directory=d_temp dumpfile=user20190425.dp
Estimate in progress using BLOCKS method...
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 3.682 GB
......
Master table "YANGTK"."SYS_EXPORT_SCHEMA_01" successfully loaded/unloaded
******************************************************************************
Dump file set for YANGTK.SYS_EXPORT_SCHEMA_01 is:
  /tmp/user20190425.dp
Job "YANGTK"."SYS_EXPORT_SCHEMA_01" successfully completed at Thu Apr 25 14:45:23 2019 elapsed 0 00:03:15
```

数据泵并行

```
[test@DEVDB ~]$ expdp yangtk directory=d_temp dumpfile=user20190425_para%U.dp parallel=4

Export: Release 11.2.0.4.0 - Production on Thu Apr 25 14:46:02 2019

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.
Password:

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Starting "YANGTK"."SYS_EXPORT_SCHEMA_01":  yangtk/******** directory=d_temp dumpfile=user20190425_para%U.dp parallel=4
Estimate in progress using BLOCKS method...
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 3.682 GB
......
Dump file set for YANGTK.SYS_EXPORT_SCHEMA_01 is:
  /tmp/user20190425_para01.dp
  /tmp/user20190425_para02.dp
  /tmp/user20190425_para03.dp
  /tmp/user20190425_para04.dp
Job "YANGTK"."SYS_EXPORT_SCHEMA_01" successfully completed at Thu Apr 25 14:48:12 2019 elapsed 0 00:01:59
```

RMAN并行

```
RMAN> run
2> {
3> allocate channel c1 device type disk format '/u02/rman/%U';
4> allocate channel c2 device type disk format '/u02/rman/%U';
5> backup tablespace system;
6> backup tablespace sysaux;
7> backup tablespace undotbs1;
8> backup tablespace users;
9> }
```

RMAN并行

```
.....
input datafile file number=00001 name=/u01/app/oracle/oradata/DB18C/datafile/o1_mf_system_f9ovorc1_.dbf
channel c1: starting piece 1 at 2019-04-25 15:41:12
channel c1: finished piece 1 at 2019-04-25 15:41:27
piece handle=/u02/rman/19tvsrgo_1_1 tag=TAG20190425T154112 comment=NONE
channel c1: backup set complete, elapsed time: 00:00:15
Finished backup at 2019-04-25 15:41:27
.....
input datafile file number=00003 name=/u01/app/oracle/oradata/DB18C/datafile/o1_mf_sysaux_f9ovq5gy_.dbf
channel c1: starting piece 1 at 2019-04-25 15:41:28
channel c1: finished piece 1 at 2019-04-25 15:42:13
piece handle=/u02/rman/1atvsrh8_1_1 tag=TAG20190425T154128 comment=NONE
.....
input datafile file number=00004 name=/u01/app/oracle/oradata/DB18C/datafile/o1_mf_undotbs1_f9ovqyl3_.dbf
channel c1: starting piece 1 at 2019-04-25 15:42:13
channel c1: finished piece 1 at 2019-04-25 15:42:14
piece handle=/u02/rman/1btvsril_1_1 tag=TAG20190425T154213 comment=NONE
.....
input datafile file number=00007 name=/u01/app/oracle/oradata/DB18C/datafile/o1_mf_users_f9ovqznq_.dbf
channel c1: starting piece 1 at 2019-04-25 15:42:14
channel c1: finished piece 1 at 2019-04-25 15:43:19
.....
```

RMAN并行

```
RMAN> run
2> {
3> allocate channel c1 device type disk format '/u02/rman/%U';
4> allocate channel c2 device type disk format '/u02/rman/%U';
5> backup tablespace system, sysaux, undotbs1, users;
6> }
```

RMAN并行

```
channel c1: SID=290 device type=DISK
Starting backup at 2019-04-25 15:30:54
channel c1: starting full datafile backup set
channel c1: specifying datafile(s) in backup set
input datafile file number=00007 name=/u01/app/oracle/oradata/DB18C/datafile/o1_mf_users_f9ovqznq_.dbf
input datafile file number=00004 name=/u01/app/oracle/oradata/DB18C/datafile/o1_mf_undotbs1_f9ovqyl3_.dbf
channel c1: starting piece 1 at 2019-04-25 15:30:55
channel c2: starting full datafile backup set
channel c2: specifying datafile(s) in backup set
input datafile file number=00003 name=/u01/app/oracle/oradata/DB18C/datafile/o1_mf_sysaux_f9ovq5gy_.dbf
input datafile file number=00001 name=/u01/app/oracle/oradata/DB18C/datafile/o1_mf_system_f9ovorc1_.dbf
channel c2: starting piece 1 at 2019-04-25 15:30:55
channel c1: finished piece 1 at 2019-04-25 15:32:30
piece handle=/u02/rman/16tvsqtf_1_1 tag=TAG20190425T153054 comment=NONE
channel c1: backup set complete, elapsed time: 00:01:36
channel c2: finished piece 1 at 2019-04-25 15:32:31
piece handle=/u02/rman/17tvsqtf_1_1 tag=TAG20190425T153054 comment=NONE
channel c2: backup set complete, elapsed time: 00:01:36
Finished backup at 2019-04-25 15:32:31
released channel: c1
released channel: c2
```

SQLLDR并行

```
$ sqlldr yangtk/yangtk control=control1.ctl direct=true parallel=true

SQL*Loader: Release 11.2.0.4.0 - Production on Wed May 1 22:35:47 2019

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.


Load completed - logical record count 1000000.

$ sqlldr yangtk/yangtk control=control2.ctl direct=true parallel=true

SQL*Loader: Release 11.2.0.4.0 - Production on Wed May 1 22:35:48 2019

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.


Load completed - logical record count 1000000.
```

升级并行

```
[oracle@DEVDB admin]$ $ORACLE_HOME/perl/bin/perl catctl.pl -n 8 -l /home/oracle catupgrd.sql
......
Number of Cpus        = 4
Database Name         = DEVDB_N
DataBase Version      = 11.2.0.4.0
Parallel SQL Process Count            = 8

------------------------------------------------------
Phases [0-108]         Start Time:[2019_05_06 16:11:08]
------------------------------------------------------
**********   Executing Change Scripts   **********
Serial   Phase #:0    [DEVDB_N] Files:1     Time: 870s
***************   Catalog Core SQL   ***************
Serial   Phase #:1    [DEVDB_N] Files:5     Time: 51s
Restart  Phase #:2    [DEVDB_N] Files:1     Time: 1s
***********   Catalog Tables and Views   ***********
Parallel Phase #:3    [DEVDB_N] Files:19    Time: 10s
Restart  Phase #:4    [DEVDB_N] Files:1     Time: 0s
......
Serial   Phase #:107  [DEVDB_N] Files:1     Time: 0s
Serial   Phase #:108  [DEVDB_N] Files:1     Time: 29s
------------------------------------------------------
Phases [0-108]         End Time:[2019_05_06 17:10:12]
------------------------------------------------------
Grand Total Upgrade Time:    [0d:0h:59m:7s]
```

升级并行

```
[test@DEVDB db18c]$ ps -ef|grep 4629
oracle    4629 31265  0 16:11 pts/0    00:00:00 /u02/db18c/perl/bin/perl catctl.pl -n 8 -l /home/oracle catupgrd.sql
oracle    4712  4629  0 16:11 pts/0    00:00:00 /u02/db18c/bin/sqlplus
oracle    4713  4629  0 16:11 pts/0    00:00:00 /u02/db18c/bin/sqlplus
oracle    4714  4629  0 16:11 pts/0    00:00:00 /u02/db18c/bin/sqlplus
oracle    4715  4629  0 16:11 pts/0    00:00:00 /u02/db18c/bin/sqlplus
oracle    4716  4629  0 16:11 pts/0    00:00:00 /u02/db18c/bin/sqlplus
oracle    4720  4629  0 16:11 pts/0    00:00:00 /u02/db18c/bin/sqlplus
oracle    4733  4629  0 16:11 pts/0    00:00:00 /u02/db18c/bin/sqlplus
oracle    4739  4629  0 16:11 pts/0    00:00:00 /u02/db18c/bin/sqlplus
test      5446  1361  0 16:12 pts/4    00:00:00 grep 4629
```

人工并行

- 数据泵大表导出
  expdp query对大表同时分片导出
- SQLLOADER导入
  启动多个SQLLDR命令同时导入表中，利用PARTITION EXCHANGE交换到目标表
- 多分区/子分区DDL操作
  多会话同时对多个分区进行DDL操作
- 大批量更新或删除
  使用DBMS_PARALLEL_EXECUTE包

人工并行

```
SQL> DECLARE
  2     V_SQL VARCHAR2(4000);
  3     V_STATUS NUMBER;
  4  BEGIN
  5     DBMS_PARALLEL_EXECUTE.CREATE_TASK('T_PARALLEL_UPDATE');
  6     DBMS_PARALLEL_EXECUTE.CREATE_CHUNKS_BY_ROWID('T_PARALLEL_UPDATE', 'YANGTK', 'T_RECORD', TRUE, 32);
  7     V_SQL := 'UPDATE /*+ ROWID(A) */ T_RECORD A SET NAME = LOWER(NAME) WHERE ROWID BETWEEN :START_ID AND :END_ID';
  8     DBMS_PARALLEL_EXECUTE.RUN_TASK('T_PARALLEL_UPDATE', V_SQL, DBMS_SQL.NATIVE, PARALLEL_LEVEL => 8);
  9        V_STATUS := DBMS_PARALLEL_EXECUTE.TASK_STATUS('T_PARALLEL_UPDATE');
 10        WHILE (V_STATUS != DBMS_PARALLEL_EXECUTE.FINISHED) LOOP
 11              DBMS_OUTPUT.PUT_LINE('PARALLEL EXECUTE STATUS:' || V_STATUS);
 12              DBMS_LOCK.SLEEP(5);
 13              V_STATUS := DBMS_PARALLEL_EXECUTE.TASK_STATUS('T_PARALLEL_UPDATE');
 14        END LOOP;
 15     DBMS_PARALLEL_EXECUTE.DROP_TASK('T_PARALLEL_UPDATE');
 16  EXCEPTION
 17     WHEN OTHERS THEN
 18              RAISE;
 19  END;
 20  /

PL/SQL procedure successfully completed.

Elapsed: 00:04:41.50
```

- 并行的基础概念
- 并行的发展演进
- 并行的不只是SQL
- 并行的常见问题

资源耗尽

- 并行本质是资源换时间
- 并行不会降低系统资源消耗
- 并行是不具备伸缩性的
- 前台应用调用并行是灾难

资源耗尽

- CPU耗尽
- IO耗尽
- 内存耗尽
- 心跳网络耗尽
- Process进程耗尽

资源耗尽

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
DSS	197558254	dss1	1	29-Feb-16 21:02	11.2.0.4.0	YES

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
nbasdb1	AIX-Based Systems (64-bit)	96	24		250.25

	Snap Id	Snap Time	Sessions	Cursors/Session	Instances
Begin Snap:	4847	04-Mar-16 11:00:29	971	2.6	2
End Snap:	4848	04-Mar-16 12:00:04	474	1.9	2
Elapsed:		59.59 (mins)			
DB Time:		175,811.41 (mins)			

资源耗尽

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
library cache: mutex X	1,031,161	6287K	6097	59.6	Concurrency
library cache load lock	7,427	3717.1K	500484	35.2	Concurrency
cursor: pin S wait on X	3,999	212.4K	53103	2.0	Concurrency
latch: parallel query alloc buffer	72,378	182.3K	2519	1.7	Other
row cache lock	105,451	50.1K	475	.5	Concurrency
DB CPU		33.1K		.3	
latch: shared pool	52,094	30K	575	.3	Concurrency
db file sequential read	1,507,859	10.4K	7	.1	User I/O
db file scattered read	748,994	10.2K	14	.1	User I/O
library cache lock	8,220	4535.2	552	.0	Concurrency

资源耗尽

```
INSERT INTO MID_PH_W_USER_FEE NOLOGGING
SELECT /*+PARALLEL(A, 8)*/ NVL(B.S_USER_NO, A.USER_NO) USER_NO, NVL(B.AREA_NO, A.AREA_NO) AREA_NO,
  NVL(B.CITY_NO, A.CITY_NO) CITY_NO, NVL(B.SVC_ID, A.SVC_ID) SVC_ID, NVL(B.BRAND_ID, A.BRAND_ID) BRAND_ID,
  A.DEVICE_NO, SUM(A.ALL_FEE) ALL_FEE, SUM(A.RENT_FEE) RENT_FEE, SUM(A.LOCAL_FEE) LOCAL_FEE,
  SUM(A.LONG_FEE) LONG_FEE, SUM(A.ROAM_FEE) ROAM_FEE, SUM(A.ICR_FEE) ICR_FEE, SUM(A.OTHER_FEE) OTHER_FEE,
  SUM(SF_ALL_FEE - ALL_FEE) FAVOR_FEE, '0', '0' IS_CHANGE, '0', '0', NVL(B.OFFICE_NO, A.OFFICE_NO) OFFICE_NO
FROM MID_M_C_USER_4G A,
  (SELECT /*+PARALLEL(4, 8)*/ S_USER_NO, C_USER_NO, AREA_NO, CITY_NO, SVC_ID, BRAND_ID, DEVELOP_CHANNEL OFFICE_NO
  FROM MID_M_B_C_USER A
  WHERE ACCT_MONTH = :B1
  AND CHANGE_MONTH >= SUBSTR(:B1 , 1, 4) || '01') B
WHERE ACCT_MONTH = :B1
AND A.USER_NO = B.C_USER_NO(+)
GROUP BY NVL(B.S_USER_NO, A.USER_NO), NVL(B.AREA_NO, A.AREA_NO), NVL(B.CITY_NO, A.CITY_NO), NVL(B.SV C_ID,
A.SVC_ID), NVL(B.BRAND_ID, A.BRAND_ID), A.DEVICE_NO, NVL(B.OFFICE_NO, A.OFFICE_NO)
```

资源耗尽

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	116,110,264		120.35
CPU Time (ms)	248,540		0.72
Executions	0		
Buffer Gets	220,723		0.01
Disk Reads	31,392		0.03
Parse Calls	382		0.08
Rows	0		
User I/O Wait Time (ms)	70,079		
Cluster Wait Time (ms)	21,996		
Application Wait Time (ms)	0		
Concurrency Wait Time (ms)	7,326,492		
Invalidations	1		
Version Count	2		
Sharable Mem(KB)	69		

资源耗尽

Pool	Name	Begin MB	End MB	% Diff
java	free memory	896.00	896.00	0.00
shared	ASH buffers	192.00	192.00	0.00
shared	ASM extent pointer array	117.81	117.81	0.00
shared	Checkpoint queue	750.05	750.05	0.00
shared	FileOpenBlock	121.60	121.60	0.00
shared	KGLH0	941.07	145.29	-84.56
shared	KGLHD	180.64		-100.00
shared	KQR L PO	377.22	186.09	-50.67
shared	SQLA	1,358.16	214.32	-84.22
shared	db_block_hash_buckets	178.00	178.00	0.00
shared	dbktb: trace buffer	125.00	125.00	0.00
shared	event statistics per sess	151.72	151.72	0.00
shared	free memory	251.92	3,635.50	1343.13
shared	gc name table	120.00	120.00	0.00
shared	gcs resources	921.44	921.44	0.00
shared	gcs shadows	637.92	637.92	0.00
shared	ges big msg buffers	121.60	121.60	0.00
shared	ges enqueues	221.03	221.03	0.00
shared	ges resource	477.66	437.62	-8.38
shared	ksunfy : SSO free list	144.33	144.33	0.00
shared	state objects	218.45	218.45	0.00
	buffer_cache	30,976.00	30,080.00	-2.89
	fixed_sga	2.15	2.15	0.00
	log_buffer	199.55	199.55	0.00

```
SQL> select * from v$px_process_sysstat where statistic like 'Buffers%';

STATISTIC                                                        VALUE
---------------------------------------------------------------- ----------
Buffers Allocated                                               1991710
Buffers Freed                                                   1991693
Buffers Current                                                      17
Buffers HWM                                                     114212
SQL> show parameter parallel_execution_message

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
parallel_execution_message_size      integer     16384
SQL> select 114212 * 16384 /1024/1024 from dual;

114212*16384/1024/1024
----------------------
                1784.5625
ALTER SYSTEM SET "_PX_use_large_pool" = TRUE SCOPE = SPFILE;
```

未启用并行

- 未设置会话参数
- 并行代价高
- 并行进程资源不足
- 无法并行执行

未启用并行

```
SQL> UPDATE /*+ PARALLEL(8) */ T_RECORD SET NAME = UPPER(NAME);

24832512 rows updated.

Elapsed: 00:06:23.17

Execution Plan
----------------------------------------------------------
Plan hash value: 2336775929
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	TQ	IN-OUT	PQ Distrib
0	UPDATE STATEMENT		24M	568M	6550 (1)	00:00:03			
1	UPDATE	T_RECORD							
2	PX COORDINATOR								
3	PX SEND QC (RANDOM)	:TQ10000	24M	568M	6550 (1)	00:00:03	Q1,00	P->S	QC (RAND)
4	PX BLOCK ITERATOR		24M	568M	6550 (1)	00:00:03	Q1,00	PCWC	
5	TABLE ACCESS FULL	T_RECORD	24M	568M	6550 (1)	00:00:03	Q1,00	PCWP	

未启用并行

```
SQL> ALTER SESSION FORCE PARALLEL DML PARALLEL 8;

Session altered.

Elapsed: 00:00:00.00

SQL> UPDATE /*+ PARALLEL(8) */ T_RECORD SET NAME = UPPER(NAME);

24832512 rows updated.

Elapsed: 00:04:16.67

SQL> COMMIT;

Commit complete.

Elapsed: 00:00:00.08
```

未启用并行

```
SQL> EXPLAIN PLAN FOR UPDATE /*+ PARALLEL(8) */ T_RECORD SET NAME = UPPER(NAME);

Explained.

SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------
Plan hash value: 2898626402

--------------------------------------------------------------------------------------------------------------------------
| Id  | Operation              | Name     | Rows  | Bytes | Cost (%CPU)| Time     |    TQ  |IN-OUT| PQ Distrib |
--------------------------------------------------------------------------------------------------------------------------
|   0 | UPDATE STATEMENT       |          |   24M|   568M|  6550   (1)| 00:00:03 |        |      |            |
|   1 |  PX COORDINATOR        |          |      |       |            |          |        |      |            |
|   2 |   PX SEND QC (RANDOM)  | :TQ10000 |   24M|   568M|  6550   (1)| 00:00:03 | Q1,00  | P->S | QC (RAND)  |
|   3 |    UPDATE              | T_RECORD |      |       |            |          | Q1,00  | PCWP |            |
|   4 |     PX BLOCK ITERATOR  |          |   24M|   568M|  6550   (1)| 00:00:03 | Q1,00  | PCWC |            |
|   5 |      TABLE ACCESS FULL | T_RECORD |   24M|   568M|  6550   (1)| 00:00:03 | Q1,00  | PCWP |            |
--------------------------------------------------------------------------------------------------------------------------
```

未启用并行

```
SQL> select * from v$version;

BANNER
--------------------------------------------------------------------------------
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
PL/SQL Release 11.2.0.4.0 - Production
CORE    11.2.0.4.0      Production
TNS for Linux: Version 11.2.0.4.0 - Production
NLSRTL Version 11.2.0.4.0 - Production

SQL> select /*+ index(t) parallel(2) */ count(name) from t_para_ind t where created >= to_date('201701', 'yyyymm');

-------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name              | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                   |     1 |    34 | 63096   (1)| 00:00:03 |
|   1 |  SORT AGGREGATE              |                   |     1 |    34 |            |          |
|   2 |   TABLE ACCESS BY INDEX ROWID| T_PARA_IND        |  3365K|   109M| 63096   (1)| 00:00:03 |
|*  3 |    INDEX RANGE SCAN          | IND_PARA_CREATED  |  3672K|       |  9850   (2)| 00:00:01 |
-------------------------------------------------------------------------------------------------
```

未启用并行

```
SQL> select banner from v$version;

BANNER
--------------------------------------------------------------------------------
Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production

SQL> select /*+ index(t) parallel(2) */ count(name) from t_para_ind t where created >= to_date('201701', 'yyyymm');
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT		1	26	54292 (1)	00:00:03			
1	SORT AGGREGATE		1	26					
2	PX COORDINATOR								
3	PX SEND QC (RANDOM)	:TQ10001	1	26			Q1,01	P->S	QC (RAND)
4	SORT AGGREGATE		1	26			Q1,01	PCWP	
5	TABLE ACCESS BY INDEX ROWID BATCHED	T_PARA_IND	3073K	76M	54292 (1)	00:00:03	Q1,01	PCWP	
6	PX RECEIVE		3073K		9913 (2)	00:00:01	Q1,01	PCWP	
7	PX SEND HASH (BLOCK ADDRESS)	:TQ10000	3073K		9913 (2)	00:00:01	Q1,00	S->P	HASH (BLOCK
8	PX SELECTOR						Q1,00	SCWC	
* 9	INDEX RANGE SCAN	IND_PARA_CREATED	3073K		9913 (2)	00:00:01	Q1,00	SCWP