

Parallel Execution Plans

A critical sound-bite

Jonathan Lewis

jonathanlewis.wordpress.com

www.jlcomp.demon.co.uk

Sample Query

```
select
    count(t1.small_vc),    count(t2.small_vc),
    count(t3.small_vc),    count(t4.small_vc)
from
    t4,                    -- big table
    t1,                    -- small table
    t2,                    --      "
    t3                     --      "
where
    t1.id = t4.id1
and   t2.id = t4.id2
and   t3.id = t4.id3
and   t1.small_vc in (1,2,3)    -- 3 rows out of 70
and   t2.small_vc in (1,2,3)    --      "
and   t3.small_vc in (1,2,3)    --      "
;
```

Sample Query (serial plan)

select * from table(dbms_xplan.display_cursor);

Id	Operation	Name	Starts	E-Rows	A-Rows
0	SELECT STATEMENT		1		1
1	SORT AGGREGATE		1	1	1
* 2	HASH JOIN		1	26	27
* 3	TABLE ACCESS FULL	T3	1	3	3
* 4	HASH JOIN		1	612	630
* 5	TABLE ACCESS FULL	T2	1	3	3
* 6	HASH JOIN		1	14491	14700
* 7	TABLE ACCESS FULL	T1	1	3	3
8	TABLE ACCESS FULL	T4	1	343K	343K

Jonathan Lewis
© 2011 - 2014

PX Plans
3 / 10

Parallel plan (*hash / hash*)

Id	Operation	Name	Rows	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT						
1	SORT AGGREGATE		1				
2	PX COORDINATOR						
3	PX SEND QC (RANDOM)	:TQ10006	1		Q1,06	P->S	QC (RAND)
4	SORT AGGREGATE		1		Q1,06	PCWP	
* 5	HASH JOIN		26	00:00:29	Q1,06	PCWP	
6	JOIN FILTER CREATE	:BF0000	3	00:00:01	Q1,06	PCWP	
7	PX RECEIVE		3	00:00:01	Q1,06	PCWP	
8	PX SEND HASH	:TQ10004	3	00:00:01	Q1,04	P->P	HASH
9	PX BLOCK ITERATOR		3	00:00:01	Q1,04	PCWC	
* 10	TABLE ACCESS FULL	T3	3	00:00:01	Q1,04	PCWP	
11	PX RECEIVE		612	00:00:29	Q1,06	PCWP	
12	PX SEND HASH	:TQ10005	612	00:00:29	Q1,05	P->P	HASH
13	JOIN FILTER USE	:BF0000	612	00:00:29	Q1,05	PCWP	
* 14	HASH JOIN BUFFERED		612	00:00:29	Q1,05	PCWP	
15	JOIN FILTER CREATE	:BF0001	3	00:00:01	Q1,05	PCWP	
16	PX RECEIVE		3	00:00:01	Q1,05	PCWP	
17	PX SEND HASH	:TQ10002	3	00:00:01	Q1,02	P->P	HASH
18	PX BLOCK ITERATOR		3	00:00:01	Q1,02	PCWC	
* 19	TABLE ACCESS FULL	T2	3	00:00:01	Q1,02	PCWP	
20	PX RECEIVE		14491	00:00:29	Q1,05	PCWP	
21	PX SEND HASH	:TQ10003	14491	00:00:29	Q1,03	P->P	HASH
22	JOIN FILTER USE	:BF0001	14491	00:00:29	Q1,03	PCWP	
* 23	HASH JOIN BUFFERED		14491	00:00:29	Q1,03	PCWP	
24	JOIN FILTER CREATE	:BF0002	3	00:00:01	Q1,03	PCWP	
25	PX RECEIVE		3	00:00:01	Q1,03	PCWP	
26	PX SEND HASH	:TQ10000	3	00:00:01	Q1,00	P->P	HASH
27	PX BLOCK ITERATOR		3	00:00:01	Q1,00	PCWC	
* 28	TABLE ACCESS FULL	T1	3	00:00:01	Q1,00	PCWP	
29	PX RECEIVE		343K	00:00:29	Q1,03	PCWP	
30	PX SEND HASH	:TQ10001	343K	00:00:29	Q1,01	P->P	HASH
31	JOIN FILTER USE	:BF0002	343K	00:00:29	Q1,01	PCWP	
32	PX BLOCK ITERATOR		343K	00:00:29	Q1,01	PCWC	
* 33	TABLE ACCESS FULL	T4	343K	00:00:29	Q1,01	PCWP	

Jonathan Lewis
© 2011 - 2014

PX Plans
4 / 10

Oracle Enterprise Manager (SYS) - SQL Details: 96f267bcnjqqf - Mozilla Firefox

https://localhost:1158/em/console/database/instance/sqlDetail?leftEdge=&rightEdge=&event=switchDisplay&direction=&selectedBand=&iconIndex=&switchID=&refreshChoice=RT...

Operation	Object	Object Node	Order	Rows	Bytes	Cost	CPU (%)	Time TQ	IN-OUT	PQ Distribution
SELECT STATEMENT			34	0	0	464	100	0:0.0		
SORT AGGREGATE			33	1	38	0		0:0.0		
PX COORDINATOR			32	0	0	0		0:0.0		
PX SEND QC (RANDOM)	SYS.TQ10006	Q1006	31	1	38	0		0:0.0	Q1006	P->S
SORT AGGREGATE		Q1006	30	1	38	0		0:0.0	Q1006	PCWP
HASH JOIN		Q1006	29	26	988	464	1	0:0.6	Q1006	PCWP
JOIN FILTER CREATE	SYS.BF0000	Q1006	5	3	18	2	0	0:0.1	Q1006	PCWP
PX RECEIVE		Q1006	4	3	18	2	0	0:0.1	Q1006	PCWP
PX SEND HASH	SYS.TQ10004	Q1004	3	3	18	2	0	0:0.1	Q1004	P->P
PX BLOCK ITERATOR		Q1004	2	3	18	2	0	0:0.1	Q1004	PCWC
TABLE ACCESS FULL	TEST_USER_T3	Q1004	1	3	18	2	0	0:0.1	Q1004	PCWP
PX RECEIVE		Q1004	38	612	19.125K	461	1	0:0.6	Q1006	PCWP
PX SEND HASH	SYS.TQ10005	Q1005	27	612	19.125K	461	1	0:0.6	Q1005	P->P
JOIN FILTER USE	SYS.BF0000	Q1005	26	612	19.125K	461	1	0:0.6	Q1005	PCWP
HASH JOIN BUFFERED		Q1005	25	612	19.125K	461	1	0:0.6	Q1005	PCWP
JOIN FILTER CREATE	SYS.BF0001	Q1005	10	3	18	2	0	0:0.1	Q1005	PCWP
PX RECEIVE		Q1005	9	3	18	2	0	0:0.1	Q1005	PCWP
PX SEND HASH	SYS.TQ10002	Q1002	8	3	18	2	0	0:0.1	Q1002	P->P
PX BLOCK ITERATOR		Q1002	7	3	18	2	0	0:0.1	Q1002	PCWC
TABLE ACCESS FULL	TEST_USER_T2	Q1002	6	3	18	2	0	0:0.1	Q1002	PCWP
PX RECEIVE		Q1005	24	14.491	367.936K	459	1	0:0.6	Q1005	PCWP
PX SEND HASH	SYS.TQ10003	Q1003	23	14.491	367.936K	459	1	0:0.6	Q1003	P->P
JOIN FILTER USE	SYS.BF0001	Q1003	22	14.491	367.936K	459	1	0:0.6	Q1003	PCWP
HASH JOIN BUFFERED		Q1003	21	14.491	367.936K	459	1	0:0.6	Q1003	PCWP
JOIN FILTER CREATE	SYS.BF0002	Q1003	15	3	18	2	0	0:0.1	Q1003	PCWP
PX RECEIVE		Q1003	14	3	18	2	0	0:0.1	Q1003	PCWP
PX SEND HASH	SYS.TQ10000	Q1000	13	3	18	2	0	0:0.1	Q1000	P->P
PX BLOCK ITERATOR		Q1000	12	3	18	2	0	0:0.1	Q1000	PCWC
TABLE ACCESS FULL	TEST_USER_T1	Q1000	11	3	18	2	0	0:0.1	Q1000	PCWP
PX RECEIVE		Q1003	20	343.000	6.542M	455	0	0:0.6	Q1003	PCWP
PX SEND HASH	SYS.TQ10001	Q1001	19	343.000	6.542M	455	0	0:0.6	Q1001	P->P
JOIN FILTER USE	SYS.BF0002	Q1001	18	343.000	6.542M	455	0	0:0.6	Q1001	PCWP
PX BLOCK ITERATOR		Q1001	17	343.000	6.542M	455	0	0:0.6	Q1001	PCWC
TABLE ACCESS FULL	TEST_USER_T4	Q1001	16	343.000	6.542M	455	0	0:0.6	Q1001	PCWP

Show Explain Rewrite

OEM has got the order of operation **completely** wrong. It's using the basic "first child, recursive descent" algorithm.

Parallel Images

Parallel execution servers for ORDER BY operation

Parallel execution servers for full table scan

User Process

Parallel Execution Coordinator

A - G

H - M

N - S

T - Z

employees Table

SELECT *
from employees
ORDER BY last_name;

Intra-Operation parallelism

Inter-Operation parallelism

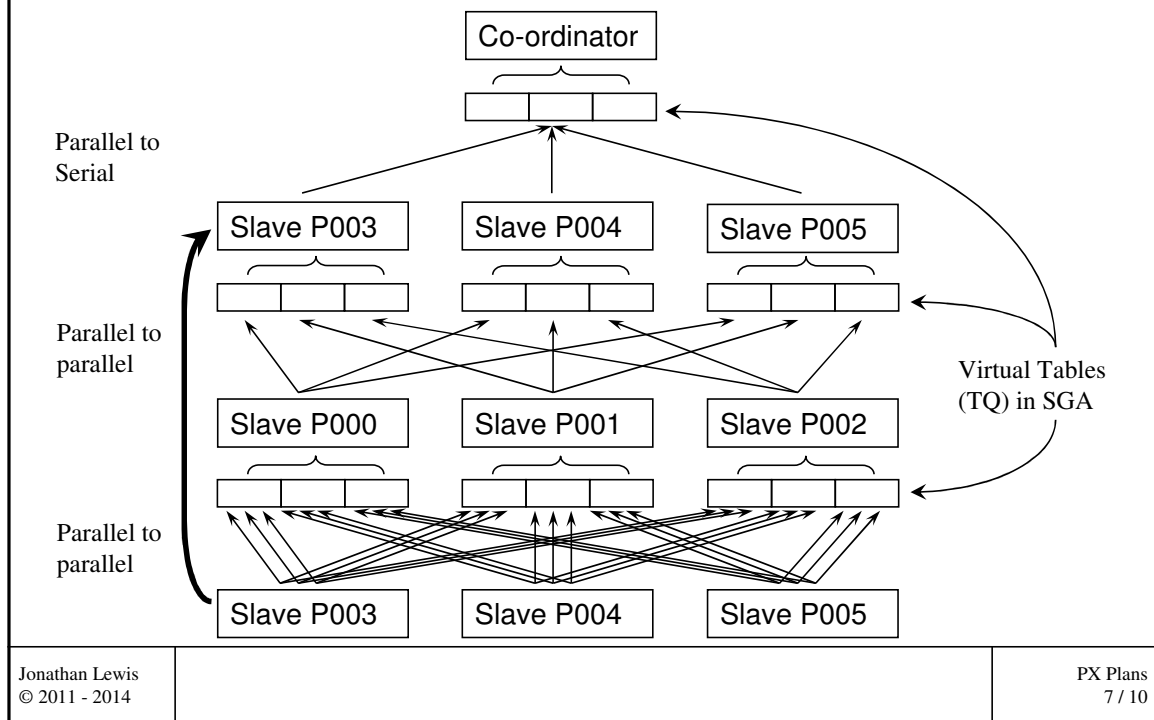
Intra-Operation parallelism

Oracle® Database VLDB and Partitioning Guide Ch. 8

Jonathan Lewis
© 2011 - 2014

PX Plans
6 / 10

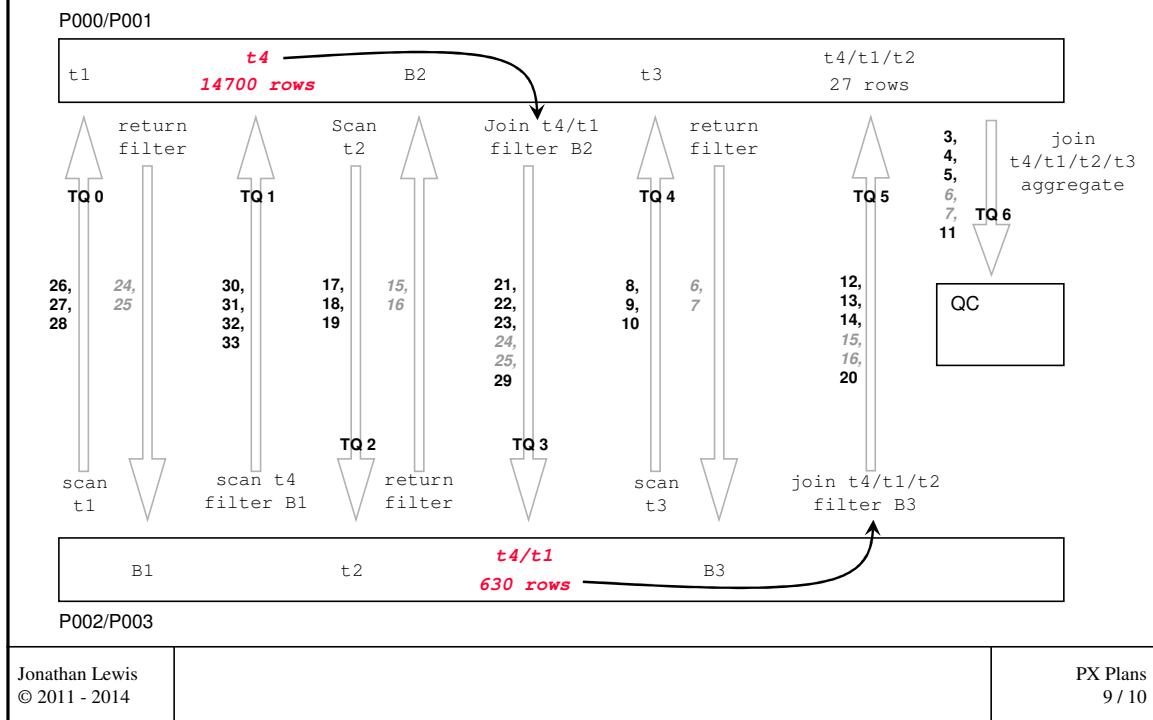
Parallel Execution - visual



Execution plan (hash / hash)

Id	Operation	Name	Rows	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT						
1	SORT AGGREGATE		1				
2	PX COORDINATOR						
3	PX SEND QC (RANDOM)	:TQ10006	1		Q1,06	P->S	QC (RAND)
4	SORT AGGREGATE		1		Q1,06	PCWP	
* 5	HASH JOIN		26	00:00:29	Q1,06	PCWP	
6	JOIN FILTER CREATE	:BF0000	3	00:00:01	Q1,06	PCWP	
7	PX RECEIVE		3	00:00:01	Q1,06	PCWP	
8	PX SEND HASH	:TQ10004	3	00:00:01	Q1,04	P->P	HASH
9	PX BLOCK ITERATOR		3	00:00:01	Q1,04	PCWP	
* 10	TABLE ACCESS FULL	T3	3	00:00:01	Q1,04	PCWP	
11	PX RECEIVE		612	00:00:29	Q1,06	PCWP	
12	PX SEND HASH	:TQ10005	612	00:00:29	Q1,05	P->P	HASH
13	JOIN FILTER USE	:BF0000	612	00:00:29	Q1,05	PCWP	
* 14	HASH JOIN BUFFERED		612	00:00:29	Q1,05	PCWP	
15	JOIN FILTER CREATE	:BF0001	3	00:00:01	Q1,05	PCWP	
16	PX RECEIVE		3	00:00:01	Q1,05	PCWP	
17	PX SEND HASH	:TQ10002	3	00:00:01	Q1,02	P->P	HASH
18	PX BLOCK ITERATOR		3	00:00:01	Q1,02	PCWP	
* 19	TABLE ACCESS FULL	T2	3	00:00:01	Q1,02	PCWP	
20	PX RECEIVE		14491	00:00:29	Q1,05	PCWP	
21	PX SEND HASH	:TQ10003	14491	00:00:29	Q1,03	P->P	HASH
22	JOIN FILTER USE	:BF0001	14491	00:00:29	Q1,03	PCWP	
* 23	HASH JOIN BUFFERED		14491	00:00:29	Q1,03	PCWP	
24	JOIN FILTER CREATE	:BF0002	3	00:00:01	Q1,03	PCWP	
25	PX RECEIVE		3	00:00:01	Q1,03	PCWP	
26	PX SEND HASH	:TQ10000	3	00:00:01	Q1,00	P->P	HASH
27	PX BLOCK ITERATOR		3	00:00:01	Q1,00	PCWP	
* 28	TABLE ACCESS FULL	T1	3	00:00:01	Q1,00	PCWP	
29	PX RECEIVE		343K	00:00:29	Q1,03	PCWP	
30	PX SEND HASH	:TQ10001	343K	00:00:29	Q1,01	P->P	HASH
31	JOIN FILTER USE	:BF0002	343K	00:00:29	Q1,01	PCWP	
32	PX BLOCK ITERATOR		343K	00:00:29	Q1,01	PCWP	
* 33	TABLE ACCESS FULL	T4	343K	00:00:29	Q1,01	PCWP	

Graphic (hash / hash)



Sound-bite

Follow the TQxyyyy name order

"Name" = :TQxyyyy and "TQ" = Qxx,yyyy

"Buffered" is a threat

Broadcast plans don't need the buffering

Broadcast plans use Bloom filters sooner

Broadcast plans can be read using "first child first"