**COORDINATOR OF SERIAL FORCED PX**

<https://www.eehelp.com/question/coordinator-of-serial-forced-px/>

So I have lots and lots of parallel slaves available in one of my databases. However, Monday, one of the support guys came to me and said "this query used to take three hours, but now it takes more than 24 hours before having to kill him".  
  
Unfortunately, the last time it was run was last month and I don't explain stats of plans/AWR going back that far, so can not see what he was doing.  
  
However, when I look at, the request includes several PARALLELS, but reference tables valued DEGREE PARALLEL. It should be parallel. However, it is not.  
  
When I see the plan to explain it, I see the following right at the top:  
  
**INSERT STATEMENT  
LOAD SELECT AS  
COUNTY STOPKEY  
COORDINATOR OF SERIAL FORCED PX  
PX SEND QC (RANDOM)  
COUNTY STOPKEY  
OUTER HASH JOIN  
PX RECEIVE**Blah, blah, blah  
  
What is the purpose of the FORCED SERIAL COORDINATOR PX? The stats of the objects look OK and last month, the thin application as expected. They were no changes in the structure of the underlying table. I suspect that it is forcing the query to run in series, which is at the origin of the performance problem.  
  
We never saw it? It comes to 10.2.0.4

I found a few interesting article on this topic:

<http://www.oaktable.NET/category/tags/PX-Coordinator-forced-serial>

<http://Oracle-Randolf.blogspot.com/2011/03/PX-Coordinator-forced-serial-operation.html>

<http://oracledoug.com/Serendipity/index.php?/archives/788-PX-issues-continued.html>

You have all the user functions in your actual SQL? If so, this parallel has been activated?  
A database logon trigger? If so, it changed in the last n days?

HTH - Mark D Powell.

PS - If special occurs then the next the RTO thread may be of interest

Parallel query of 'CONTAINS' and 'SERIAL FORCE' exec plan

<https://community.oracle.com/thread/687119>

Edited by: Mark D Powell on August 10, 2011 08:20

**I was looking at some presentations today, when I ran across the URL below, it was very informative and covered much of the information I already knew but one of the things that was interesting, was that these BUFFERING operations are a consequence of the producer/consumer architecture that oracle created for parallel operations and tried to fix in 12cR1 but wasn't able to completely.**

**Some things like suppressing multiple DFO trees in one plan, which made things very confusing they fixed. These BUFFERING operations are meant stop PX Send and PX Receive operations from happening at the same time.**

**The BUFFERING means that the results of a send are saved to until the second receive data operation is done. This save could be to memory in the PGA or to TEMP disk and this creates more problems. They are now trying to fix this in 12cR2.**

**I will be installing 12cR2 on my Linux Ubuntu this week.**

**from Randolf video**

**PARTITION WISE OPERATIONS**

**http://www.oaktable.net/category/tags/parallel-execution**

**EACH DIFFERENT DFO TREE IS OPTIMIZED BY THE OPTIMIZER INDIVIDUALLY**

**execution plans with multiple DFO trees might run with different degrees of parallizum.**

**PX Trace, Parallel groups**

**NEW IN 12C "PX SEND 1 SLAVE" NEW OPERATION MULTIPLE TREES NOW IN ONE**

**HTTPS://WWW.youtube.com/watch?v=6BAiW6AwhfQ**