> Tom,
>
> I have some queries that need to access large amounts of data to return the desired result. When trying to tune these queries I often see the optimizer is using buffer sort.
>
> I've read these posts about the subject which suggest that buffer sort is not really doing a sort at all:
> http://jonathanlewis.wordpress.com/2006/12/17/buffer-sorts/
> http://toddlerdba.blogspot.com/2010/09/one-of-our-developer-complained-that-he.html
>
> I still do not fully comprehend what exactly buffer sort means, when can a query benefit from using buffer sort and when should it be avoided? (when possible)
> I think it's often used in combination with Cartesian Joins but that is not the case in my queries.
>
> My experience up till now is that if I avoid the buffer sort the performance will improve. The queries are executed in parallel by the way.
>
> Please explain

and we said...

> It is used in a wide variety of plans, you can see them with most any plan - as Jonathan demonstrated.
>
> It is a buffering technique - it may or may not actually involve sorting.
>
> In the one case above, the parallel one, it involved changing the degree of parallelism (from 1 to something more than 1). If that is how you've been avoiding them - then I would say the performance difference is not due to the lack of a buffer sort, but rather the slightly higher degree of parallelism. That is, the plan changed in a major way - but not because of a lack of buffer sort, but rather because of a parallel change.
>
> If we did not buffer sort, then we would have to hit the table over and over again - so when we use it, it is better than not using it (given the same plan). But if you change the query in a large way (by influencing the degree of parallelism), you really cannot compare it to the buffer sort plan anymore. They are very different - resource wise - at that point.