Hello All!  
  
Parallel databases give you the potential to both speedup and scale up your application. This is given to us from the concept of parallel processing. This is when a large task is divided into a smaller tasks and work is divided among other nodes. These nodes can be CPU’s in the same device or essentially CPU’s in another device. It is good to note that that not all process can be divided and that parallel programming in some instances cannot be resourceful. An Example of a bottleneck that would cause this is as follows:

“Let a single line to order food now become two lines with two registers to order food. The cook however can only cook one meal at a time till completion.”

The line to order food goes twice as fast because there are now two nodes processing orders but the length to get your food remains the same because the orders are till being cooked one node at time. In real life scenarios in code I feel as if Java multithreading issues and JavaScript closures issues are good examples that are relatable to these concepts of parallel processing. So that is how and why parallel databases utilize parallel processing to speed up operations and scale their applications. Another caveat to be discussed when choosing to use a parallel database or not is the concept of synchronization. Sometimes business logic will have to be applied to synchronize nodes to process the information correctly in the order that you want. If your resources used to process this synchronization is greater than the resources saved with utilizing a parallel database, then it is not worth it to the parallel route. Parallel databases are all about utilizing resources efficiently and if they can not do that do to synchronization complexity then it is not worth it.

When it comes to how Oracle databases utilize memory, they take traditional row level storage and it extend it by adding a new column level storage on top of it. One popular take away from this is allowing both traditional business queries at a row level and analytical queries that need to take place at a column level for big data. Some other popular features that extend these new capabilities are:

Vector Processing - Using vector instructions These instructions can process multiple values in one instruction.

In memory storage indexes –These indexes stored in memory can conserve processing power by allowing an entire index to be skipped during a query if it is known that no matching value will be found within that index.

In-Memory Optimized Joins and Reporting – This optimizes the bloom filter to perform joins faster by using the scan of the outer table, generating a compact bloom filter which can be used to reduce resources used by the join from the scan of the inner table.

Thanks again!

-ERIC WEBB

https://www.csee.umbc.edu/portal/help/oracle8/server.815/a67778/ch1\_unde.htm#2804

https://www.oracle.com/technetwork/database/in-memory/overview/twp-dbim-usage-2441076.html