Cartridge speed: Partitioned vs Non-Partitioned

P. Hindmarch

9 Aug 2008

Here are some speed comparisons between a portioned and non-partitioned installation. The dataset was 350k reactions taken from a cut of the Array ELN database. I also confirmed that the number of hits was the same in each case.

In most cases, the non-partitioned search is slower, but not by much. We would expect the differences to increase at the database gets bigger. The MW and MF searches are the same speed.

One other thing to note. These are pure Cartridge searches. When doing a search in an application there is always an additional overhead due to preparation of the recordset for display – so the apparent speed difference may not be as visible as the actual differences in the structure search. This may explain why some customers are reporting that Partitioning is not giving the performance benefit that they were expecting. As mentioned below, it would be nice to get some metrics on performance via the ELN interface.

Results are in seconds averaged over three runs.

|  |  |  |  |
| --- | --- | --- | --- |
| Test # | Query | Partitioning | No Partitioning |
| 1 |  | 13.96 | 14.95 |
| 2 | FULL=YES | 7.12 | 8.45 |
| 3 |  | 1.90 | 1.92 |
| 4 |  | 1.80 | 2.14 |
| 5 |  | 0.23 | 0.30 |
| 6 |  | 0.23 | 0.23 |
| 7 |  | 0.25 | 0.25 |
| 8 | Formula = CH5-6 O2-3 | 0.2 | 0.2 |
| 9 | MW = 120 to 130 | 0.11 | 0.11 |

For the test #6, the spped is the same. This is because there are 0 hits. You would expect there to be no – or very little difference – when there are no hits found by screening and ABAS is not called. This also explains why the MF and MW searches were the same speed.

## Further notes:

There are a few issues to consider when considering the performance of our applications with respect to partitioning.

1)      The speed of the Cartridge is only part of the problem. The individual applications (ELN, Reg etc) also need to be optimized properly to fully utilize partitioning for their non-chemical data (DocStore, experiment data etc). This needs to be separately tested and metrics produced.

2)      Installing the Partitioning option doesn’t automatically make a database application faster. It can make things worse if not done properly. There are huge documents out there on how schemas should be designed and optimized for partitioning.

3)      When the Cartridge detects Partitioning is present, it will automatically create partitioned tables. There is no way to override this on a per-index basis – which does make testing more complicated than it could be.

## Testing:

In order to test partitioning against non-partitioning configuration on the same hardware, a workaround was used to simulate a non-partitioned Oracle instance

1. Install Oracle Cartridge (“CSCartridge”) using all of the default options on an Oracle 10R2 system with Partitioning enabled
2. Install a second Crtridge instance along side the first with a different name, e.g CsCartridgeNoP
3. Edit the PL/SQL function CSCARTRIDGENOP.IsPartitioning to always return false. This fools the Cartridge into thinking that Partitioning is not enabled

CREATE OR REPLACE FUNCTION CSCARTRIDGENOP.IsPartitioning RETURN NUMBER AS

partitioning VARCHAR2(100);

BEGIN

EXECUTE IMMEDIATE 'SELECT VALUE FROM SYS.V\_$OPTION WHERE PARAMETER = ' || AddQuotes('Partitioning') INTO partitioning;

if partitioning = 'TRUE' then

return 0;

else

return 0;

end if;

END;

/

1. The SQL partition\_speed\_test.sql executes a series of quieries using both Cartridge instances and reports the time taken for the searches.