

Sort Performance Improvements in Oracle Database 10g Release 2

An Oracle White Paper
June 2005

Sort Performance Improvements in Oracle Database 10g Release 2

Executive Overview	3
Feature Description	3
Performance Characteristics	4
Conclusion.....	5
Appendix A	6
700 MB sort area size	6
10 MB sort area size	7

Sort Performance Improvements in Oracle Database 10g Release 2

EXECUTIVE OVERVIEW

Working with larger data sets always requires one of the probably most fundamental data processing operations: sorting. Whether you want to retrieve a list of your employees in an alphabetical order, to see all customer related events for the last month in a chronological order, or whether you want to analyze revenue numbers of your sales regions in a descending order – data has to be sorted. Other operations, for example the creation of indexes, also heavily rely on sorting.

Oracle Database 10g Release 2 introduces a new, more efficient sort algorithm that improves the performance of database sorts significantly. Because sorting is an internal database operation invoked by SQL operations, every application will transparently take advantage of the new sort algorithm when migrated to Oracle Database 10g Release 2.

The improvements of the new sort implementation are significant: improved sort-performance of up to 5 times has been measured in lab conditions. Traditionally, file-based sort engines running outside a database engine were often chosen when the database performance was not considered sufficient. External sort engines might have shown better performance than the database in certain situations. These days are over. The new sort algorithm in Oracle Database 10g Release 2 makes Oracle's in-database sort performance comparable with – or even more performant than – these external sort engines. Oracle Database 10g Release 2 eliminates the need for sort outside the database.

FEATURE DESCRIPTION

Oracle Database 10g Release 2 introduces a completely new sort algorithm. This new algorithm is automatically and transparently leveraged whenever using sort operations in Oracle Database 10g Release 2.

One of the big advantages of the new sort algorithm is that it is far more efficient in using a large memory pool. Unlike the new sort, the old sort algorithm would not improve – sometimes even degrade – when adding more memory after a certain threshold. Spilling the sort to disk (the temp tablespace) would be more efficient in these cases! The new sort algorithm leverages all possible memory it can get. Consequently, the memory allocation for the new algorithm will be more aggressive

when the memory allocation is controlled by the database, using the automatic runtime memory management (controlled by PGA_AGGREGATE_TARGET).

The new sort algorithm also benefits from faster CPUs. You will see more performance improvements on a system with fast CPUs than you will on a system with slower CPUs.

However, the algorithm cannot be applied to every data type, such as complex data types or XML types. Data types that cannot use the new algorithm continue to use the old algorithm. The most common data types, including VARCHAR2, NUMBER and DATE, are compatible with the new sort algorithm and will benefit automatically as discussed in this paper.

PERFORMANCE CHARACTERISTICS

For performance tests comparing the old sort with the new sort implementation, the same SQL statement was run on identical systems (CPU, Memory, IO) against the same data set, using Oracle Database 10g Release 1 and Oracle Database 10g Release 2 respectively.

In-database sort is usually combined with other operations, including any kind of calculations, joins, aggregations, or data display. To evaluate the performance improvements of the new sort algorithm, we use a SQL statement eliminating as many of these dependencies as possible. The following SQL statement was chosen for the performance evaluation:

```
select count(*)
from (select /*+ NO_MERGE */ <column(s)>
      from <table>
      order by <column(s)>
      )
```

Comparing the sort performance with the old algorithm versus the new algorithm shows significant performance improvements (see Figure 1).

Sort Performance Improvement, 700 MB sort area

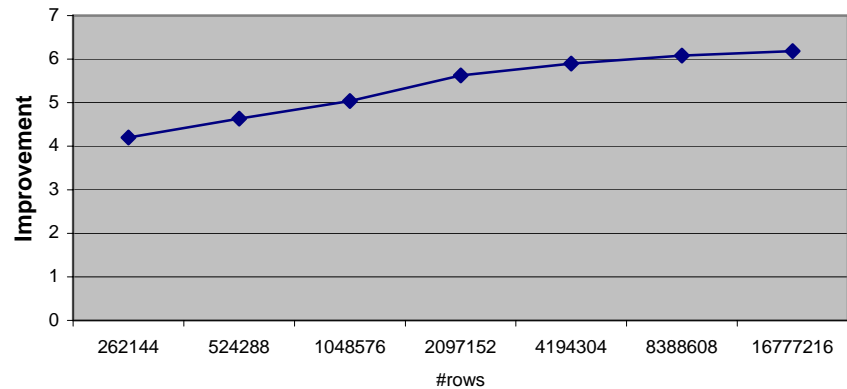


Figure 1. Sort performance improvements

A closer analysis of the performance improvement ratios shows they vary depending on the characteristics of the data and the sort operation. Our findings show that the ratio of improvement depends on:

- The amount of memory that is available for the sort, and as a result whether the sort has to spill to disk. **The new sort algorithm shows more performance improvement for in-memory sorts.**
- The cardinality (relative number of unique values) of the column or columns in the sort. **The performance improvement is better for higher cardinality sorts.**
- The number of columns in the select list. **Queries with fewer columns in the select list will show more improvements.**

Two specific test cases are discussed in more detail in Appendix A.

CONCLUSION

Sort is a very common operation in a database. Many statements either implicitly or explicitly use sorts. Oracle Database 10g Release 2 introduces a new sort algorithm that significantly improves the performance of in-database sorts. Database applications that migrate to Oracle Database 10g Release 2 transparently take advantage of the new sort algorithm, showing improvements of up to a factor of five, eliminating the need for specific sort engines outside the database for heavy sort operations.

APPENDIX A

This appendix describes two test cases that compare the old database sort algorithm with the new sort algorithm.

Both test cases use the following common test parameters:

- 28 byte rows (8 byte key column, 20 byte non-key column)
- Each byte of the key has one of 25 values (this is comparable to uppercase alphabetic data)
- There are few duplicate key values

The tests have been performed on a 3 GHz Pentium 4 server with 2 GB of RAM, running Redhat Enterprise Linux 3. The results below were run in serial. Parallel queries show comparable performance improvements.

The following statement was used to perform the tests:

```
select count(*)
from (select /*+ NO_MERGE */ <column>
      from <table>
      order by <column>
      )
```

700 MB sort area size

The first set of results used a 700 MB sort area size. With 28 byte rows, this means that a sort with more than 26,214,400 records will spill to disk.

Table 1. Sort performance results for 700 MB sort area size

# rows	old sort (seconds)	new sort (seconds)	improvement ratio
262,144	0.84	0.20	4.2
524,288	1.90	0.41	4.6
1,048,576	4.28	0.85	5.0
2,097,152	9.79	1.74	5.6
4,194,304	22.11	3.75	5.9
8,388,608	50.75	8.35	6.1
16,777,216	115.81	18.73	6.2
33,554,432*	305.01	116.54	2.6*

* Sort spills to disk

10 MB sort area size

The second set of results used a 10 MB sort area size. With 28 byte rows, this means that a sort with more than 374,491 records will spill to disk.

Table 2. Sort performance results for 10 MB sort area size

# rows	old sort (seconds)	new sort (seconds)	improvement ratio
262,144*	0.84	0.22	3.8*
524,288	2.06	0.79	2.6
1,048,576	4.15	1.44	2.9
2,097,152	8.39	2.91	2.9
4,194,304	16.85	6.03	2.8
8,388,608	33.93	12.73	2.7
16,777,216	69.09	26.03	2.7
33,554,432	143.51	63.85	2.2

* In-memory sort



Sort Performance Improvements in Oracle Database 10g Release 2
June 2005

Author: Mark Van de Wiel
Contributing Authors: Hermann Baer

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2005, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.