Advanced Performance Optimization Techniques

Introduction

Performance tuning focuses on improving the execution time of a program without changing the overall functionality. This page presents some advanced performance improvement techniques.

Performance Tuning Techniques

a) **Use of Hash Tables**

Hash tables are useful when there is a requirement of READ operations on a large DATASET with full table key. The response time for a key access is constant in case of Hash tables and is independent of the number of the entries in the table.

Pre Requisites: The entries in the table should be unique and the key is to be defined for the table during the declaration of the internal table.

b) **Buffered Reading (Using special Function Modules)**

SAP recommends use of Buffered Reading (SAP Note - 332856) in cases where a program contains several accesses to a Master table with a fully specified primary key. SAP has provided special Function Modules which store the result of the last database request in the memory. You can use this stored value again without needing to access the database.

Note 332856 - Reading buffered master data for cust.-spec. enhancements

c) **Using Oracle hints**

In some scenarios, the Oracle Optimizer is not able to choose the correct access path (index) for retrieval of the data from the database for your query. In such cases, we can force the Optimizer to use the required index, by providing the details in the query itself, using "Oracle Hints". (SAP Note - 772497)

Note 772497 - FAQ: Oracle Hints

d) **Avoiding high buffer reads**

In order to find the inefficient SQL Statements using excessive CPU or doing excessive I/O, use the transaction ST04. A query is inefficient if both Buffer Gets/Row and Buffer Gets/Exec are high. A Buffer Get is when Oracle references a page of Memory in the Oracle database memory. A high ratio means that oracle is searching a large amount of data to find the results. (SAP Note - 766349)

Note 766349 - FAQ: Oracle SQL optimization

e) **Activating Buffering for tables**

In case of small master data tables which are accessed frequently and updated rarely, it is advisable to activate Buffering.

Fully Buffered - Very small tables with many different accesses
Single Record Buffer - For tables where there are frequent single-record accesses (with SELECT SINGLE ...).

f) **Complete Specification of the Primary Key Fields**
If we take an example of some of the Tables such as VBFA or BSEG or MSEG, then there are more than 1 Primary Key fields present in the respective Tables. In order for the SELECT Query to Run faster, it is important to include all the Primary Key Fields (As many as Possible - Recommended to Use All the Fields) in the WHERE Clause so that the Query runs Faster and thus improving the Performance.

Also, adhering to the ground Rule of Specify the Field List after the SELECT Statement in the Order that they are existing in the Actual Table would be much faster than just throwing a Zigsaw Puzzle in the SELECT Query with the Fields not in the Order.