Computer and Information Science
www.ccsenet.org/journal.html
Vol. 1, No. 4
November 2008

Adjustment of Performance for the Application of PL/SQL

Erjun Chen, Shaohong Yin
College of Computer and Automatization
Tianjin Polytechnic University
Tianjin 300160, China

Abstract

The oracle database system is the large relation database management system which is applied most broadly at present. The PL/SQL language is a sort of block structured language, and it is the procedural language that the Oracle extends SQL, and it allows encapsulating operation logics, and its advantage is to use inscribed SQL sentence to define and realize 3GL language structure and describe more complex query, trigger, memory process and other objectives. Therefore, to more reasonably use PL/SQL program can exert important function to implement the Oracle database performances. Through the adjustment of database shared pool and the introduction of different technologies of data access, we discussed some technologies to enhance the implementation performance of PL/SQL and optimize the performance of the database system by means of PL/SQL language.

Keywords: PL/SQL language, Performance optimization, Shared pool, Data access

PL/SQL (Procedural Language/SQL) is the extension of SQL, and it absorbs many excellent characters of programming language in recent years, such as data encapsulation, information concealment, reloading, and exception treatment. It allows the data control language and query sentence of SQL contained in the block structure and the code process language, and makes PL/SQL become into a sort of transaction processing language with powerful function. Figure 1 shows the client/server system structure of PL/SQL based on the application of Oracle.

In the process of software design and development, the implementation performance of software is an important problem which we should consider. The quick and highly effective implementation performance of software is the important limited factor to decide whether its output, input, throughput and customer satisfaction are eligible. In the application of Oracle database, this sort of limitation is more important. The size of Oracle database and the user amount using this application are important factors to decide whether the application achieve maximum implementation performance through adjustment. Therefore, we need to adjust the application of PL/SQL and make its performance implement by the optimal mode according to a great lot of user’s throughput and retractility.

Before the application is adjusted, we need to know what reasons induce it operate so slowly, so Oracle offers many database supervision and diagnoses tools to analyze the performance of the application, or we can use the mechanism offered by PL/SQL to get the implementation time of centisecond, and this mechanism can be described as that it uses the function DBMS_UTILITY.GER_TIME, and usually transfer this function twice, and get the difference between two return values, and the difference is the time consumption of program operation, so we can confirm where is the bottleneck of the application.

1. The adjustment to the SGA shared pool

Both the shared memory buffer and the background course are called as an Oracle example. When one Oracle example is started, many Oracle background courses will be started, and every course takes charge the treatment to database for different aspects, and various courses communicate each other through shared memory, and this memory is the whole system SGA. SGA is divided into different areas and one of them is called as the shared pool which contains the text and PL/SQL block of SQL sentence transmitting to the database, and the presentation forms and implementation project after analysis.

Before one memory module is implemented or one memory package is cited, the codes in the coded package must be loaded into SGA. The default method to maintain the package in SGA is to let RDBMS use its arithmetic management codes which are used least recently. When one package is first cited, the coded codes will be loaded into the shared pool, so any user with right of EXECUTE to the package can use it. It will be in the shared pool until other resources based on memory need memory and that package is not used recently, and so the package is washed out from the shared pool. When this package objective is needed next time, the whole package must be loaded into the memory again. Therefore, the shared pool is larger, the probability that the program stays in the memory next time will be higher. However, too large shared pool will waste memory. So we should supervise the shared buffer to ensure it saves all analyzed SQL cursors and PL/SQL code snippets which usually are cited in the application. If too many exchanges occur, we can adjust the parameter of SHARED_POOL in the document of INIT.ORA to add the size of the shared pool (when the physical memory is allowed).

2. The adjustment to data access

2.1 Utilizing package data to implement minimized SQL access

When announcing one variable in a package or package explanation, its function area is not limited in any special process or function. So the function area of package data is the whole Oracle dialog, and the value of that data will be continually effectively in the whole term of dialog, and we can utilize this fact to minimize the times of access from the SQL layer. The structured implementation query aiming at PGA is much quickly than the query to SGA from A to Izzard, so when the application needs to implement the query that the term of multiple re-dialogs is not changed, the above method is the most convenient one. For example, suppose that one program needs to get one exclusive dialog id to avoid folding with other dialogs, one method to realize this aim is to transfer DBMS_LOCK.ALLOCATE_UNIQUE and pick up the exclusive latch appellation.

2.2 Citing the transfer parameter

The citation of transfer parameter is to use the transfer of a finger pointing to the argument to replace the parameter transfer mechanism which transfers the copy of the argument.

The call-back of subprogram parameter is decided by the parameter type and parameter mode. The argument and the parameter in the parameter type are corresponding. For parameter, the parameter modes are respectively IN, OUT and INOUT. Two transfer modes include value transfer and address transfer when the argument is transferred. PL/SQL adopts the mode of address transfer for the parameter of IN mode, and adopts the mode of value transfer for the parameter of OUT and INOUT mode to transfer them. Because the copy of value must be made, so when large-sized data result uses OUT or INOUT parameters, the transfer of value will occupy more implementation time and consume more memory space, especially transferring the result set, the recorded index-by tables or objectives. In order to reduce this sort of consumption, we should adopt the coder clew of NOCOPY. Table 1 describes the behaviors of appointed IN, OUT, INOUT and NOCOPY.

Because of avoiding copying the argument to the parameter, the mode of cited transfer makes the program implement quicker, and when large-sized data is transferred, its meaning is more obvious.

2.3 Using local dynamical SQL and batch binding

The local dynamical SQL means the announced sentence when operating and the SQL sentence implementing dynamic confirmation, which is the interior support of dynamic SQL in PL/SQL explainer, and is implemented through the mode of IMMEDIATE and OPEN FOR without relative API. The implementation performance of local dynamic SQL is relative to the using times of dialog of local dynamic SQL, and if the dynamic SQL sentence is implemented many times in one dialog, so the local dynamic SQL is implemented quickly, because the local dynamic SQL sentence will analyze the SQL sentence when it is implemented every time. Otherwise, because the local dynamic SQL announced in the expression doesn't contain API, so it can minimize the system spending of process transfer, on the other hand, because same SQL sentences are implemented by the binding variables with different values, so one cursor can be shared by many SQL sentences, thus the binding variables in the local dynamical SQL will be implemented quicker.

The batch binding means to use set input or take many rows of records from the table to one set, then only use one order to insert, renovate and delete many rows of records in the table. This process needs few switches of context between PL/SQL and SQL, and this advantage is combined with the quickness of implementation of local dynamical SQL, which can fully enhance the implementation performance of program.

2.4 Using objective type and objective set

The objective type can simulate the entity in the real world, and it and its methods operating on these types integrate data and methods into one structure, which reduces the demands to multiple tables. The introductions of objective REFS and REF line in the database are more convenient than the use of exterior keys, because it avoids complex links. Through reducing the system spending of link treatment, we can save and operate the objectives as one whole to enhance the implementation performance of program.

2.5 Using pipelined table functions (from Oracle9i)

Pipelined table function means the function that uses PIPELINED sentence in the definition of function uses PIPEROW sentence in the interior function to return the set by the pipelined way.

The steps to establish the pipelined table functions include following aspects.

(1) Establishing one objective type or PL/SQL record type.

(2) Establishing the set of this objective type or PL/SQL record type.

(3) Establishing one function to return this set.

(4) Using PIPELINED and defining this function as the pipelined function.

(5) Using PIPE sentence to return one factor of the set.

(6) Compiling one null RETURN sentence which is used to appoint the complete form of the pipelined table function.

When returning large-sized record set, the pipelined table functions can obviously enhance the implementation performance of program, and this function needs not be completely implemented, and relative sets need not be completely initialized in the memory, which can enhance the response time and reduce the memory consumption, and the function can be transferred from one table function to another common table function, which can reduce the demand of memory data in the middle table and enhance the implementation performance.

The performance optimization in the applied system has been the standard to measure the applied system. To effectively apply various optimized measures based on good database design are the base to get higher performance. In this article, we mainly discuss some technologies to enhance the implementation performance of PL/SQL and optimize the performance of the database system. In fact, the optimization of applied system in Oracle should enter into the whole applied development process including system design, database design, application design and database server deployment, and we need studying the system characters and concrete applied environment such as the resource deployment of database layer, the flux control of network layer and the whole design of operation system layer, and based on that, we can utilize the characters of PL/SQL language to implement selections from multiple selectable projects and reasonably optimize the system performance of Oracle database.

References

Bulusu Lakshman. (2004). *Manual of Oracle9iPL/SQL for Developing Personnel.* Beijing: Tsinghua University Press.

Li, Xuedong. (2007). The Technical Characters of PL/SQL and Its Application in the Software Developing. *Cfhi Technology.* No.6.

Ron Hardman & Michael McLaughlin. (2006). *Expert Oracle PL/SQL Design and Develop Advanced PL/SQL Solutions.* Beijing: Tsinghua University Press.

Shah , Nilesh. (2002). *Database Systems Using Oracle: a Simplified Guide to SQL & PL/SQL**. Prentice Hall.

Shi, Wenyuan. (2005). *Research and Implementation of some Key Technologies for PL/SQL Engine.* Changsha: National University of Defense Technology.

Thomas Kyte. (2006). *Programming Art of Oracle9i&10g.* Beijing: People's Post and Telecommunication Publishing House.

Table 1. Behaviors of various parameter modes

IN	OUT	INOUT	INONT NOCOPY, OUT NOCOPY
Arguments are transferred by the mode of citation	Arguments are transferred by the mode of value transfer	Arguments are transferred by the mode of value transfer	Parameters are transferred by the mode of citation
The transfer points to the finger of value	Transferring the copy of value	Transferring in or out the copy of value	Transferring the address of value
N/A	When the un-disposal abnormity occurs, the value transferred is returned	When the un-disposal abnormity occurs, the value transferred is returned	When the un-disposal abnormity occurs, the value transferred is not returned, so it can not correctly predict the value of out

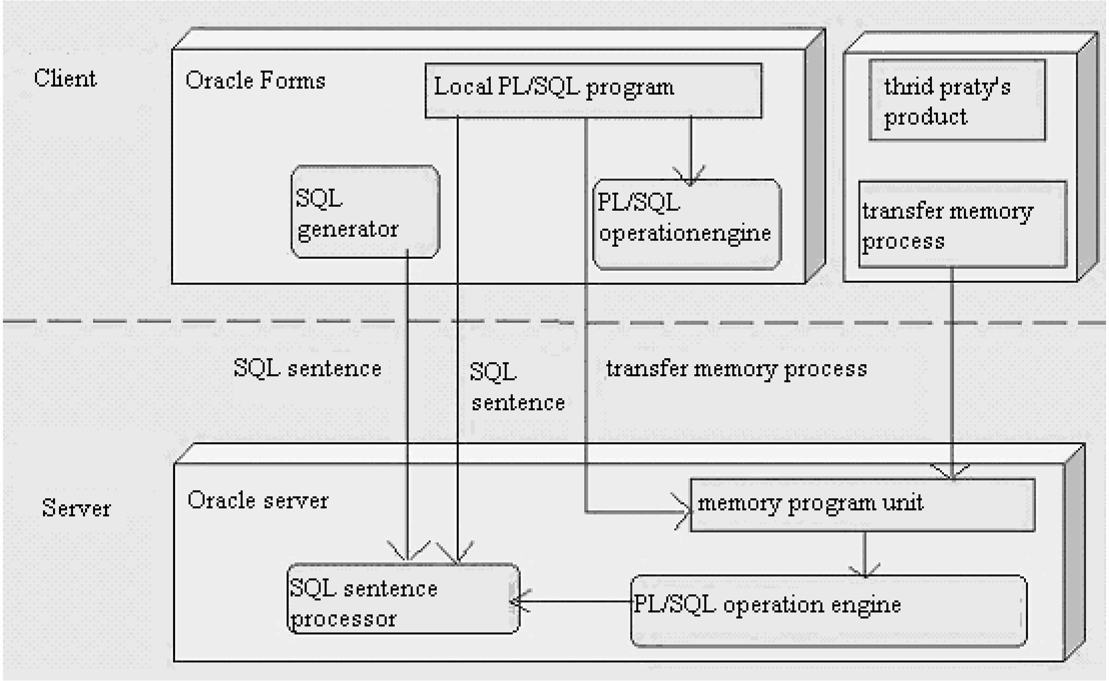

Figure 1. PL/SQL in Oracle Client/Server System Structure