TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.3, March 2014, pp. 1676 ~ 1686
DOI: http://dx.doi.org/10.11591/telkomnika.v12i3.4455

Table Partitioning Technology Based on Massive Data

Peng Wang, Ai-xue Tian*, Chang Guo
ChangChun University of Science and Technology, JiLin Province, China
*Corresponding author, e-mail: 631920575@qq.com

Abstract

In the Logistics Distribution System, in order to enhance the availability of massive data table, simplify the maintenance operations of data table, balance disk I/O and improve query performance, this system uses the range partitioning technology, interval partitioning technology, reference partitioning technology and list partitioning technology supported by Oracle 11g. Compared to retrieve the non-partitioned table and the partition table has been used time, execution plan and statistical information. Through analysis and comparison examples show that mass data table using partitioning technologies, can improve the SQL retrieval performance, and reduce execution costs. Therefore, the contribution of table partitioning technology on massive data tables is to improve the manageability and availability of data tables, and optimizing SQL retrieval performance.

***Keywords**: massive data table, oracle 11g, table partitioning technology, retrieval performance*

1. Introduction

From the point of view of the database administrator, table partitioning is according to certain principle can divided the massive data table into common or separate management of multiple fragments, which makes the management of massive data tables more convenient and flexible [1]. From the application user's perspective, the partition table and the non-partitioned table is the same, the data table has not changed [2]. Table partitioning technology can store the partitions in different disks or table spaces, balance the I/O system, and optimize system performance.

This paper mainly studies the range partition technology and interval partition technology used on the orders information table of the Logistics Distribution System, and the reference partition technology used on the distribution list information table, and the distribution phase information table, and the list partitioning technology applied to the goods information table. Among them, the order information table records the each order and its distribution information; the distribution list information table and the distribution phase information table are the relationship between parent and child, which are used to manage the distribution route of each distribution list. Goods information table records the goods' type and other goods basic information in each goods list. The key data tables of the system are mass data tables that data are growing rapidly, through examples to analysis and compare the difference between the partition table and the non-partitioned table, to illustrate this system database design using table partitioning technology is necessary, as well as the table partitioning technology's contribution to the massive data processing.

2. The Order Information Table

The database system of the Logistics Distribution System is the OLTP system based on relational database, the operations of the system's order information table that has massive data include: logistics company retrieve the order information, customers submit orders, and logistics company process historical orders.

Now creating the order information table called Orderinfo01 with the sentence:

```
SQL>Create Table OrderInfo01
   (OrderId number(15) primary key, DisListId number(15), Province varchar2(20),
    City varchar2(20),District varchar2(20),Address varchar2(100),
    GoodsInfo varchar2(50),CustomerId varchar2(10),OrderDate Date);
```

Received August 25, 2013; Revised September 24, 2013; Accepted October 4, 2013

and insert 12,052 records of order information into the table. Logistics company to statistical the number of the processed orders in March 2007, use the SQL sentence:

```
SQL>select count(*) from OrderInfo01
   where OrderDate between to_date ('01/03/2007','dd/mm/yyyy')
   and to_date ('01/04/2007','dd/mm/yyyy');
```

The elapsed time, execution plan and statistics information of the operation as shown in Table 1.

Table 1. Retrieve Order Information Table SQL Statement's Relevant Information

```
Elapsed Time     00: 00: 00.01
Execution Plan
| Id| Operation                | Name              | Rows | Bytes | Cost (%CPU) |   Time    |
---------------------------------------------------------------------------------------------
| 0|SELECT STATEMENT     |                       |   1    |    9   |   55   (0)       | 00:00:01|
| 1|  SORT AGGREGATE     |                       |   1    |    9   |                     |              |
|*2|   TABLE ACCESS FULL |  ORDERINFO01 |  156   |  1404 |   55   (0)       | 00:00:01|
Statistics Information
    4  recursive calls                        415  bytes received via SQL*Net from client
    0  db block gets                             2  SQL*Net roundtrips to/from client
  273  consistent gets                           0  sorts (memory)
    0  physical reads                            0  sorts (disk)
    0  redo size                                 1  rows processed
  423  bytes sent via SQL*Net to client
```

Table 1 shows that the whole search process takes time is 0.01 seconds, retrieve order information table the most expensive step in execution plan is "TABLE ACCESS FULL ORDERINFO01", so the data scanning mode is the full table scan during the retrieval process, the cost of the plan of the recursive call number is 4, in a state with cache consistency to read blocks is 273. As a result, the order information table that has massive data needs technical methods to be able to improve the retrieval speed and reduce implementation costs.

Logistics company's handling of historical orders can be deleted file historical data, in the form of data files archiving or historical data migration. If using the delete statement deletes historical data, during the execution process SQL statements need to scan the whole order information table or index found eligible data records, and then delete the data. This process can only delete the data on the designated period of time, and consume large amounts of CPU resources and generate large amounts of disk I/O, and can not free up the space. Even if use the drop and truncate operations also cannot avoid the costly step that scans the full table. At this time, need a suitable data table structure which is convenient to process historical order information data.

Based on the above requirements and goals, Oracle provides table partitioning technology, and uses the table partitioning technology for order information table to achieve performance optimization on massive data tables.

3. Range Partitioning Technology Based on Order Information Table

In order to reduce costs retrieve order information table, to improve the retrieval performance and convenience historical data processing, because order information table has a time data type column can be used to partition scope, so the order information table using range partitioning technology.

3.1. Technology Realization

Range partitioning, as the name implies, data table data distribution is implemented, according to the scope of the partitioning key value of the data distribution, and it is a continuous process, a complete data table can be divided into multiple partitions. In practice, range partitioning is usually a time column as the partitioning key [3]. Because of the order date column of order information table is the data type of date, and the number of orders every month basic balanced, so in order information (OrderInfo) table under the Order date (OrderDate) column each month in accordance with the way a partition to create range partitions. Add create range partitioning statements after the above SQL statements that create the order information table OrderInfo01:

```
partition by range(OrderDate)
( partition part_01 values less than (to_date('01/01/2007','dd/mm/yyyy')),
  partition part_02 values less than (to_date('01/02/2007','dd/mm/yyyy')),
  partition part_03 values less than (to_date('01/03/2007','dd/mm/yyyy')),
  partition part_04 values less than (to_date('01/04/2007','dd/mm/yyyy'));   );
```

The order information table OrderInfo based on Range partitioning successfully created, during the process of importing 12052 order information to the table, the system will be according to each record order date (OrderDate) column value belongs range, and insert it into the corresponding partition.

3.2. Technology Application

1) Optimize data retrieval performance

Count the number of orders in March 2007, because records of this period are stored in the partition PART_04, so no longer need to scan the entire order information table, so as long as use the SQL statement SQL> select count(*) from OrderInfo partition(PART_04); to statistical the number of records in partition part_04. The elapsed time, execution plan and statistics information of the operation are as shown in Table 2.

Table 2. The Relevant Information of Searching Range Partition Table OrderInfo

```
Elapsed Time  00: 00: 00.00
Execution Plan
| Id | Operation                  |      Name      |Rows|Cost (%CPU) |   Time    |Pstart| Pstop|
----------------------------------------------------------------------------------------------
| 0 | SELECT STATEMENT     |                   |   1  |     2     (0)     | 00:00:01 |        |        |
| 1 |  SORT AGGREGATE       |                   |   1  |                     |              |        |        |
| 2 |   PARTITION RANGE SINGLE |               |  50 |     2     (0)     | 00:00:01 |   4    |    4   |
| 3 |    TABLE ACCESS FULL   | ORDERINFO |  50 |     2     (0)     | 00:00:01 |   4    |    4   |
Statistics Information
   5  recursive calls                         415  bytes received via SQL*Net from client
   0  db block gets                             2  SQL*Net roundtrips to/from client
   8  consistent gets                           0  sorts (memory)
   0  physical reads                            0  sorts (disk)
   0  redo size                                 1  rows processed
423  bytes sent via SQL*Net to client
```

Comparison the information in Table 1 and Table 2 which are producing by searching the tables created in two different ways, the most costly step of the execution plan of searching range partitioning order information table is “PARTITION RANGE SINGLE”, because the Pstart and Pstop both is four, so it is only need to scan a PART_04 range partition, it makes SQL statement execution cost greatly reduced, the elapsed time is also reduced, read consistency condition with cache blocks is reduced to eight, this is strong evidence to improve the performance of data retrieval. Because the number of records to each partition is a small part of the whole data table, so, to retrieve a partition, is bound to improve data retrieval performance [4]. As a result, the Oracle range partitioning technology can optimize the huge amounts of data table query performance.

2) Historical Data Processing

Delete the partition. Since the order information table data is growing very fast, and the data takes up large system memory space, only retains data in a specific length of time, in order to reduce the system memory pressure and optimize the performance of the system, the partitions stored outdated data can be deleted [5]. If to delete the records in the order information table OrderInfo before 2007/01/01, just need sentence SQL> alter table OrderInfo drop partition (PART_01); to delete the records in partition PART_01, in this process not only delete the partition, data of the partition has been deleted, and release the space occupies by the partition. Since the order information table is partitioned by month, so in order to delete the partition of the previous month information in the beginning of each month.

Backup. Order information table backup in two ways, one is if the order data is loaded into the database with the mode of data file, load data of the order and at the same time the data file archive to the target storage medium, completes the data backup; Another way is to export orders data with the mode of table or table space transmission, and then filing, however, this way need to consume more database resources [6].

Data migration. Migrate the table space occupied by partition stored order information's historical data to an offline database, so that if the order information table of the historical data with data mining or statement analysis needs, these actions can switch to the offline database. In this way the efficiency of dealing with stale data is the highest, because they do not involve the use of table data processing, just need to directly copy table space files [7]. In every month 1st migrates the partition that last month's order information data stored to the offline database, and realizes the historical data processing efficiency.

4. Interval Partitioning Technology Based on Order Information Table

For range partitioning order information (OrderInfo) table, because the amount of data is increasing, the partition number is also increasing, and the maximum number of partitions can not be expected, so, for the data table to create all possible partitions are impossible, so the database administrator creates the next month of partition manually at the end of each month. In order to automatically create a new partition, Oracle 11g introduced new features-interval partitioning technology [8]. If you use the interval partitioning techniques, when there is record that does not belong to the created partition inserts into the order information table, the database will automatically create a new partition instead of creating by administrator manually.

Interval partitioning is based on range partitioning, is a partitioning method that defines the interval of the partitioning key value. Interval partitioning improves the manageability of the partitioned table. After the SQL statements that create the order information table OrderInfo01 adding statements create interval partitions, and realizing the interval partition order information (OrderInfo04) table creation:

```
partition by range(OrderDate)interval(numtoyminterval(1,'month'))
(partition intpart_01 values less than(to_date('01/01/2007','dd/mm/yyyy')) );
```

By the creation process of the interval partition table found that in order to achieve interval partition, it should create at least one range partition. After the success of creating Interval partition table, three records are imported to the table, through the SQL statements:

```
SQL > select Partition_name, high_value from user_tab_partitions
   where table_name = 'ORDERINFO04' order by partition_name;
```

to check the partition settings of the order information table OrderInfo04, the results as shown in Table 3.

Table 3. Interval Partition Order Information (OrderInfo04) Table Partition Settings

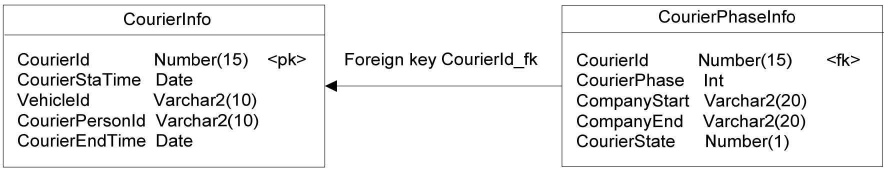
PARTITION_NAME	HIGH_VALUE
INTPART_01	TO_DATE(' 2007-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIA
SYS_P21	TO_DATE(' 2007-03-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIA
SYS_P22	TO_DATE(' 2007-02-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIA

Partition information of Table 3 show that partition SYS_P21 ceiling is on March 1, 2007, stores the order information in February 2007, partition SYS_P22 ceiling is on February 1, 2007, and storage is the order information in January 2007. The order date of the second record inserted to the interval partitioned order information table is bigger than the third record's, and in the different months, therefore, in the process of inserting the third data record does not belong to an existing partition split the SYS_P21 partition to create the SYS_P22 partition the third record belongs.

Actually, the interval partition is created dynamically range partitioning, convenient for the management which data type is date or data, because do not need to manually create a new partition, effectively improve the manageability of the partition table and the interval partitioning technology improve the performance of an SQL query as the same with range partitioning techniques. The name of the latest created partition through interval partitioning method is unknown, therefore, according to the partitioning key to know partition name first, then do query operations to this partition, also make the SQL execution price by scanning the entire data table into scanning several partitions, query operation efficiency greatly increased. In Oracle create job to modify the partition's name, each time a new partition is created, the system automatically calls the Job, the system will automatically modify the automatic partition name to the predetermined name.

5. Distribution List and Distribution Phase Information Table

The distribution list information table and the distribution phase information table is a pair of parent and child relationship table in the Logistics Distribution System, the parent table distribution list information table's primary key distribution list number is the foreign key of the distribution phase information table, a single distribution list has multiple records in the distribution phase information table, the parent table and the table is a one-to-many relationship, the relationship between the two tables is shown in Figure 1.



```
CourierInfo
CourierId         Number(15)    <pk>
CourierStaTime    Date
VehicleId         Varchar2(10)
CourierPersonId   Varchar2(10)
CourierEndTime    Date
```

```
CourierPhaseInfo
CourierId       Number(15)     <fk>
CourierPhase    Int
CompanyStart    Varchar2(20)
CompanyEnd      Varchar2(20)
CourierState    Number(1)
```

Figure 1. The Relationship between Distribution List and Distribution Phase information table

Statistics all the distribution route lines of the distribution lists in January 2007, first make two tables in accordance with the primary key and foreign key relationships linking conditions, then with conditions of the query to count again. As shown in Figure 1 of table structure. Use the SQL sentence:

```
SQL> create table CourierInfo01( CourierId number(15) primary key,
   CourierStaTime date,VehicleId varchar2(10),CourierPersonId varchar2(10),
      CourierEndTime date );
```

to create the distribution list information table called CourierInfo01, and use the SQL sentence:

```
SQL> create table CourierPhaseInfo01( CourierId number(15) not null,
   CourierPhase int not null,CompanyStart varchar2(20),CompanyEnd varchar2(20),
      CourierState number(1),constraint CourierId_fk01 foreign key (CourierId)
   references CourierInfo(CourierId) );
```

To create the distribution phase information table called CourierPhaseInfo01. Query the join table with conditions, the execution plan and statistics information as shown in Table 4.

Table 4. Statistics All Distribution Route Lines of Distribution Lists in January 2007

```
Elapsed Time   00: 00: 00.06
Execution Plan
| Id | Operation              |      Name              | Rows  | Bytes  | Cost (%CPU) |  Time      |
-----------------------------------------------------------------------------------------------------
|  0 | SELECT STATEMENT       |                        | 101K  | 7160K |   331   (4)      |  00:00:04  |
|  1 |  SORT GROUP BY         |                        | 101K  | 7160K |   331   (4)      |  00:00:04  |
|*2 |   HASH JOIN             |                        | 101K  | 7160K |   326   (2)      |  00:00:04  |
|*3 |    TABLE ACCESS FULL   | COURIERINFO01          | 1636  | 35992  |    64    (2)     |  00:00:01  |
|  4 |    TABLE ACCESS FULL |  COURIERPHASEINFO01|  222K |   10M   |   261  (2)       |   00:00:04  |
Statistics Information
       0  recursive calls                           3891  bytes received via SQL*Net    from client
       0  db block gets                              318  SQL*Net roundtrips to/from client
    1181  consistent gets                             1  sorts (memory)
       0  physical reads                              0  sorts (disk)
       0  redo size                                4750  rows processed
163268  bytes sent via SQL*Net to client
```

By the information in Table 4, execution cost of scanning two data tables is too big, the number of the called cache blocks is 1181 in the consistent read state of the statistical information. According to the experience that retrieve the order information table of the system, the two tables should be partitioned, to retrieve one or several partitions to reduce the SQL execution costs and improve the SQL query performance [9].

Under the existing technical conditions, first created range partitions for the distribution list information table on the distribution end date column, but the child table distribution phase information does not has the distribution end date column, evenly does not has a column with the date data type, if you want to achieve the scope of range partitioning this table, only do non-standardized operation for the child table. First, the distribution end date of the parent table columns should be copied to the child table, and then use range partitioning technology to the distribution end date column of child table.

The parent-child table to achieve compliance with the rules of the same partition as the process is too cumbersome, if it can achieve the child table automatically inherits the parent table partitioning strategy instead of non-standardized operation, not only can achieve the optimization of SQL query performance and convenient data table records management purposes, can also simplify the child table partitioning process and saving system resources.

6. The Reference Partitioning Technology Based on Parent-child Table

In order to achieve the parent-child table the distribution list information table and the distribution phase information table to the same partition strategy goal in the most simplified the process, Oracle 11g reference partition technology was introduced. Reference partition technology, the child table references the parent table partition method, and to make the child table has a one-to-one relationship or a many-to-one relationship of parent-child table inheriting the parent table partitioning strategy, to achieve compliance with the child and parent tables the same partitioning rules [10], and simplify the child table partition process.

6.1 Technology Realization

In the Logistics Distribution System, the first step to make reference partitioning come true, the parent table distribution list information (CourierInfo) table use the distribute end date (CourierEndTime) column as a partitioning key to create range partitioning. SQL statements are as follows:

```
SQL> create table CourierInfo( CourierId number(15) primary key,CourierStaTime date,
           VehicleId varchar2(10),CourierPersonId varchar2(10),CourierEndTime date )
           partition by range(CourierEndTime)
   (  partition part_01 values less than ( to_date ('01/01/2007','dd/mm/yyyy')),
              partition part_02 values less than ( to_date ('01/02/2007','dd/mm/yyyy')),
              partition part_03 values less than ( to_date ('01/03/2007','dd/mm/yyyy')),
              partition part_04 values less than ( to_date ('01/04/2007','dd/mm/yyyy')),
              partition part_05 values less than(maxvalue));
```

Second step, distribution list information (CourierInfo) table's child table distribution phase information (CourierPhaseInfo) table does not include the distribute end date (CourierEndTime) column, the distribution phase information (CourierPhaseInfo) table uses the reference partition technology, through reference the distribution list information (CourierInfo) table's partition method, and use the following SQL statements:

```
SQL>create table CourierPhaseInfo( CourierId number(15) not null,
           CourierPhase int not null,CompanyStart varchar2(20),
   CompanyEnd varchar2(20),CourierState number(1),
           constraint CourierId_fk foreign key(CourierId)references CourierInfo(CourierId))
           partition by reference(CourierId_fk);
```

To achieve the parent-child tables use the same partitioning method.

The distribution list informations' distribution end date (CourierEndTime) before 2007/04/01 in this table, its child table distribution phase information (CourierPhaseInfo) corresponding record in the table will be stored in the same partition, the SQL statements used to check the SQL partition information are as follows:

```
SQL> select table_name, partition_name, tablespace_name, high_value
   from user_tab_partitions
   where table_name in('COURIERINFO','COURIERPHASEINFO')
      order by partition_position,table_name;
```

The partition information as shown in Table 5.

The partition information in the Table 5 shows that child table distribution phase information (CourierPhaseInfo) table partitions' high_value are empty, which means that the child table partitions' boundaries are derived from the parent table, the child table and the parent table have the same partitions' boundary values.

Table 5. The Parent Table Courier Info and Child Table Courier Phase Info Partition Information

TABLE_NAME	PARTITION _NAME	TABLESPACE _NAME	HIGH_VALUE
COURIERINFO	PART_01	SYSTEM	TO_DATE('2007-01-0100:00:00','SYYYY-MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIA
COURIERPHASEINFO	PART_01	SYSTEM	
COURIERINFO	PART_02	SYSTEM	TO_DATE('2007-02-0100:00:00','SYYYY-MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIA
COURIERPHASEINFO	PART_02	SYSTEM	
COURIERINFO	PART_03	SYSTEM	TO_DATE('2007-03-0100:00:00','SYYYY-MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIA
COURIERPHASEINFO	PART_03	SYSTEM	
COURIERINFO	PART_04	SYSTEM	TO_DATE('2007-04-0100:00:00','SYYYY-MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIA
COURIERPHASEINFO	PART_04	SYSTEM	

6.2 Optimization Based on the Reference Partition

When to do SQL operations for the parent-child joined table that has used reference partition technology, the reference partition can realize intelligent connection of parent and child table, so as to improve the performance of the join operation [11]. Retrieval all distribution routes in January 2007, the elapsed time, execution plan and statistics information as shown in Table 6.

Table 6. Elapsed Time, Execution Plan and Statistics Information

```
Elapsed Time   00: 00: 00.04

Execution Plan
| Id | Operation                      |   Name                    | Rows  |  Bytes  | Cost (%CPU)    |    Time   | Pstart | Pstop |
---------------------------------------------------------------------------------------------------------------------------------
|  0 | SELECT STATEMENT   |                               | 62920  | 4424K |   270   (3)       | 00:00:04 |        |        |
|  1 |  SORT GROUP BY        |                               | 62920  | 4424K |   270   (3)       | 00:00:04 |        |        |
|  2 |   PARTITION RANGE ITERATOR  |                | 62920  | 4424K |   267   (2)       | 00:00:04 |   2    |    3   |
|*3 |    HASH JOIN               |                               | 62920  | 4424K |   267   (2)       | 00:00:04 |        |        |
|*4 |     TABLE ACCESS FULL| COURIERINFO          | 950      | 20900 |     3   (0)       | 00:00:01 |   2    |    3   |
|  5 |     TABLE ACCESS FULL| COURIERPHASEINFO| 219K   |  10M   |   262   (2)       | 00:00:04 |   2    |    3   |

Statistics Information
          0  recursive calls                                3891  bytes received via SQL*Net from client
          0  db block gets                                   318  SQL*Net roundtrips to/from client
         36  consistent gets                                   1  sorts (memory)
          0  physical reads                                    0  sorts (disk)
          0  redo size                                      4750  rows processed
     163268  bytes sent via SQL*Net to client
```

Comparison Table 4 and Table 6 statistics information, use the parent-child table reference partitioning technology, SQL query statement's execution cost of the execution plan reduced, read consistency condition with cache blocks also reduced from 1181 to 36, and retrieval elapsed time shortened 0.02 seconds, so the reference partition is very effective to optimize parent-child join table retrieval performance, and simplify the management of the parent table corresponding to the child table records.

7. The Goods Information Table

The goods information table is used to manage the information of each goods on the list in the Logistics Distribution System, including types of goods, weight, volume level and other basic information. Logistics Company in distribution of goods in the process, the same type of goods can transport under the same environmental conditions, such as, food goods need to protect from light and heat. In a device of vehicle transportation, liquid class goods need with shockproof device transport vehicles. Therefore, the role of this table is the statistics of certain type of goods quantity of the goods, convenient vehicle dispatching and scheduling. The SQL statements of creating goods information table:

```
SQL> create table GoodsInfo01
          ( GoodsId Number(10) constraint PK_GoodsId01 primary key,
          GoodsKind varchar2(10),Remark varchar2(100),Weight Number(5,2),
           CapLevel Number(5,2) );
```

The goods information table GoodsInfo01 created successfully. Import 51,000 records into this table. Statistical the number of goods that goods type is Clothes of the goods information table GoodsInfo01, the SQL statement, elapsed time, execution plan and statistics information, as shown in Table 7.

Table 7. The Information of Statistical the Number of Goods that Goods Type is 'Clothes'

SQL Statement	SQL> select count(*) from GoodsInfo01 where GoodsKind='Clothes';
Elapsed Time	00: 00: 00.00

```
Execution Plan
| Id | Operation              |  Name          | Rows | Bytes | Cost (%CPU) |  Time    |
-----------------------------------------------------------------------------------------
| 0 | SELECT STATEMENT        |                |   1  |    7  |   73  (2)   | 00:00:01 |
| 1 |  SORT AGGREGATE         |                |   1  |    7  |             |          |
|*2 |   TABLE ACCESS FULL     | GOODSINFO01    | 6001 | 42007 |   73  (2)   | 00:00:01 |
```

```
Statistics Information
    0  recursive calls                 415  bytes received via SQL*Net from client
    0  db block gets                     2  SQL*Net roundtrips to/from client
  264  consistent gets                   0  sorts (memory)
    0  physical reads                    0  sorts (disk)
    0  redo size                         1   rows processed
  423  bytes sent via SQL*Net to client
```

The statistical results of the SQL statement is 5000 lines, by the execution plan, the SQL is executed to scan the entire data table, so for the execution cost of massive data tables must not small, requires the table structure that can obtain the results after scanning a small amount of data records, and can reduce SQL execution cost, due to the table partitioning key goods type column GoodsKind is neither a value type nor date type, so the existing range partitioning technology can't achieve ideal goal.

8. List Partitioning Technology Based on Goods Information Table

As the type of goods are enumerated values that number is limited and rate of repeat is high due to the goods information table, therefore, need a partition technology that can be convenient management of the same type of goods list. Basing on these characteristics and functions of goods information table, list partitioning technology can achieve the same type or several types of goods in the same partition, the partitions can also like other partition method are stored in a different table space or the disk space.

8.1. Technology Realization

List partitioning is based on the value of the data fields in the table partitioning to partition, to facilitate a particular record management. Usually the status column or region column of the data table as the partitioning keys.The goods type column (GoodsKind) of the goods information (GoosInfo) table can be as a partitioning key to create list partitions, and add statements of creating partitions behind the create GoosInfo table statements:

```
partition by list(GoodsKind)
(    partition part_01 values('Clothes'), partition part_02 values ('Foods'),
partition part_03 values ('Dangerous'),partition part_04 values('Liquid', 'Paste'),
partition part_05 values(default)   );
```

List partitioning makes a particular column value of records in the uniform list stored in the same partition. List partitioning is implemented by the view, list partitioning is suitable for the specific record management, so the records that have special operations can be stored in a certain partition, convenient recording backup, query, and delete operations. Goods information table GoodsInfo is a list of partitions, convenient to goods type column GoodsKind's value is Clothes, Foods, hazardous materials, Liquid and Paste body of records management. Statistics the number of goods information that goods type is 'Clothes', due to the records' goods type is 'Clothes' are stored in the partition PART_01, so just scan the partition PART_01, the SQL statement, elapsed time, the execution plan and statistics information as shown in Table 8.

Compare execution plans in Table 7 and Table 8, statistical number of records to the list partitioning table, as long as the scan one partition, Cost (% CPU) reduce significantly, at the same time, the consistency state with cache blocks read from a full table scan of 264 reduced to 23, optimize the retrieval performance, especially the specific records retrieval operations.

Table 8. The Information of Statistical the Number of Goods That Goods Type is ‘Clothes'

```
SQL Statement    SQL> select count(*) from GoodsInfo partition(PART_01);

Elapsed Time     00: 00: 00.00

Execution Plan
| Id | Operation                  | Name          |  Rows | Cost (%CPU) | Time      |Pstart| Pstop|
-------------------------------------------------------------------------------------
| 0 | SELECT STATEMENT        |                |    1   |    7   (0)    | 00:00:01|        |        |
| 1 |  SORT AGGREGATE         |                |    1   |                 |           |        |        |
| 2 |   PARTITION LIST SINGLE |                |  5000 |    7   (0)    | 00:00:01|  KEY | KEY |
| 3 |    TABLE ACCESS FULL    |GOODSINFO|  5000 |    7   (0)    | 00:00:01|   1    |   1    |

Statistics Information
   0  recursive calls                       415  bytes received via SQL*Net from client
   0  db block gets                           2  SQL*Net roundtrips to/from client
  23  consistent gets                        0  sorts (memory)
   0  physical reads                         0  sorts (disk)
   0  redo size                              1  rows processed
 423  bytes sent via SQL*Net to client
```

8.2. Technology Application

Partition part_05 is used to store the partitioning key value is not in the former four partitions’ lists in the records, so, is to store the goods that goods' types do not belong to the first four partitions, after a default partition can't add the new partition. If you want to add the new partition, first use sentence SQL > alter table GoodsInfo drop partition (PART_05); Remove the default partition, then the sentence is SQL > alter table GoodsInfo add partition PART_06 values ('Pets'); Adding a new partition part_06 stores the goods that kind is ‘Pets’, last, use the sentence SQL> alter table GoodsInfo add partition PART_07 values (default); to add the default partition.

List partitioning and range partitioning are Oracle's basic partitioning method. Range partitioning is very useful for periodic data management, list partitioning emphasizes the rationalization management of the data, sometimes can be equated with range partitioning. Comparison the execution plan of Table 2 and Table 8, SQL execution plans of range partitioning and list partitioning are basically the same. So for range partition table and list partition table, if the data records are the same, the execution plans are basically the same, in performance does not have the too big difference, just list partitioning to solve the problem of specific data distribution, beneficial to manage data according to certain ways.

9. The Benefits of Partitioning Technology

Optimize SQL retrieval performance. To query the partition table, can only retrieve a certain or some partitions, it improves the search speed, and reduced the execution cost, the retrieve operations based on the few pieces of the partition tables also fully illustrate this point.

Enhance the data table availability. The physical storage of partition table is in partitioning forms, which make the independence of the partitions on the storage area, if the data table's one partition can't use, the other partition fragments still can use, as long as the Oracle optimizer does not involve the partition of the data have been damaged, so the SQL query plans don't refer to damaged partitions, SQL operations can be executed successfully, it may not cause any impact to the operation of users, also it makes the database administrator to restore data table the effort required is to restore the complete mass data tables to reduce a lot of [12].

Simplify data table maintenance. If the order information table that has large amounts of data is to be migrated position, the role of partition technology cannot be ignored. Assuming that the size of the order information table is 200GB, the migration process may require up to 20 hours, and would not rule out the situation occurred during this downtime, needs to be re-migrated. If it’s partitioned into 50 small tables, each partition migration can be done separately, per unit time is shortened, and reduced the pressure on the system, and the migration operation can be completed in stages, so fully embodies the partition technology superiority in simplified data table maintenance [13].

10. Conclusion

Partition technology to a certain extent, optimize the performance of the system, to increase the parallelism of the system, but for different systems, partition technique can play its advantages is not sure. To focus on application effect of huge amounts of data table data warehouse, partitioning technology can play a role in optimizing system performance [14]. For some systems, the partition technique only in massive data table when a full table scan plays to the role of the performance optimization, but not for data update. So, comprehensively consider the contributions of partitions, and then make a decision whether to use partition technology or not [15].

This paper studies the basic and extended partitioning technologies supported by Oracle 11g apply in the Logistics Distribution System of the database design. Through research, range partitioning is suitable for the partitioning key is time or data columns of data table, in particular need of expired for data processing or load data by time of the data table; Because interval partitioning and reference partitioning are extended based on the basic partition, so the extended partitions and the basic partitions based on have similar applying conditions and characteristic. List partitioning applies to a specific record management, and the number of partitioning key value of the data table is limited to use the list of partition technology. Table partitioning technology can improve the data tables' manageability, for maintenance operations can focus on a certain partition; because the query operations can be specified partitions directly, optimize the efficiency of the query; Mutual independence between the partitions is made when a partition is unavailable, the table in the query and transaction operations can still continue [16].

Acknowledgement

This work is the Jilin province science and technology development plan project, and is the key project of science and technology support plan

References

[1] Thomas Kyte. Effective Oracle by Design. Oracle Press. 2003: 178.
[2] Shihui Wang, Zhengjun Li. Oracle partitioning technology research and implementation. *Science and technology square. China.* 2011; (9).
[3] Miao Chen, WeiBin Chen, Xin Hong. Research Based on Table Partitioning train ticketing Optimization of massive data. Expert Dr BBS. China.
[4] Jonathan Lewis. Cost-Based Oracle Fundamentals. APress. 2005: 34.
[5] Yong Jiang. The ORACLE database partition technology and its application. Cutting edge of science and technology. China. 2011; (29).
[6] Xian Lou. Massive data storage and query optimization strategy. Dalian maritime university master's degree thesis. China. 2012; 05.
[7] Huaiyuan Tan. Let Oracle run faster 2 - based on the database design and optimization of massive data. China: Publishing House of Electronics Industry. 2012: 17.
[8] Kavin Loney. Oracle Database 11g: The Complete Reference. Oracle Press. 2009: 323.
[9] Wei Zhang. By technical transformation method to explore and practice of the Oracle database partitions. *Fujian computer.* China. 2011; (11).
[10] Thomas Kyte. Expert Oracle Database Architecture Oracle Database 9i, 10g and 11g Programming Techniques and Solutions. APress. 2010.

[11] Immanuel Chan, Oracle Database Performance Tuning Guide, 10g Release 2 (10.2). Oracle Press. 19-10.

[12] Thomas Kyte. Expert one-on-one Oracle. APress. 2003: 777.

[13] WenPing Hou. The Art of Database Performance Optimizatio. China Machine Press. 2012: 235.

[14] Hongbo Zhao. Real-time data warehouse environment partitioning technology research and application. Northeastern university master's degree thesis. China. 2008; 06.

[15] Angli Liu. Theoretical Analysis for Scale-down-Aware Service Allocation in Cloud Storage Systems. *International Journal of Electrical and Computer Engineering (IJECE).* 2013; 3(1): 21-29.

[16] Hong Sun, Shi-ping Chen, Li-ping Xu, Ying-ying Chen. Base-on Cloud Computing A new type of distributed application server system design. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(7): 1800-1807.