

Database Design for Performance Data in Integrated Network Management System

YeonJoo Na, IlSoo Ahn

Network Management Solutions Lab.
Telecommunication Systems Division
Samsung Electronics Co., Korea

e-mail: {yjna, isahn}@samsung.com

Abstract

It is important to integrate different network interworking-protocols and network management data for the development of the integrated network management system. The integration of network management data means the integration of network management database, which is the integration of different database table schemas. Since performance data out of the network management data have different table schemas by equipment kinds, they have resulted in the main factor of troubles for the integration of database. This paper proposes a new table schema to integrate and manage complex performance data and comparatively analyzes data of processing performance on the proposed table.

Keywords: Integrated NMS, Performance Data, Database, Partitioning, XML

Introduction

- Recently, Integrated Network Management System is required as communication service systems are integrated on the basis of IP technologies
- Performance data is the biggest trouble in DB integration
 - Conventional performance data tables have been respectively defined according to measurement types of each equipment
 - Many tables need to be defined
- Propose a new table schema to integrate complex performance data

```

CREATE TABLE org_call_r (
  ME_ID          NUMBER(10)      NOT NULL,
  ATTEMPT       NUMBER(10)      NOT NULL,
  INT_ATT       NUMBER(10)      NOT NULL,
  INT_SZR       NUMBER(10)      NOT NULL,
  INT_ANS       NUMBER(10)      NOT NULL,
  ORO_ATT       NUMBER(10)      NOT NULL,
  ORO_SZR       NUMBER(10)      NOT NULL,
  ORO_ANS       NUMBER(10)      NOT NULL,
  EMR_ATT       NUMBER(10)      NOT NULL,
  EMR_ANS       NUMBER(10)      NOT NULL,
  VMS_ATT       NUMBER(10)      NOT NULL,
  VMS_ANS       NUMBER(10)      NOT NULL,
  CVC_ATT       NUMBER(10)      NOT NULL,
  CVC_ANS       NUMBER(10)      NOT NULL,
  MMS_ATT       NUMBER(10)      NOT NULL,
  MMS_ANS       NUMBER(10)      NOT NULL,
  SMS_CALL     NUMBER(10)      NOT NULL,
  SS_CALL      NUMBER(10)      NOT NULL,
  SS_SUCC      NUMBER(10)      NOT NULL,
  SS_FAIL      NUMBER(10)      NOT NULL,
  MNP_ATT      NUMBER(10)      NOT NULL,
  MNP_ERR      NUMBER(10)      NOT NULL,
  :            :              :
  :            :              :
  F_RATIO      NUMBER(9,2)     NOT NULL,
  A4D_TIME     NUMBER(9,2)     NOT NULL,
  TRAFFIC      NUMBER(9,2)     NOT NULL,
  DATE_TIME    DATE            NOT NULL
)
    
```

[Figure 1] An example of conventional performance data table

1. Introduction

The network management system uses database to manage and store network management data. The network management data are divided into and managed by fault management information, configuration management information, accounting&subscriber management information, performance management information and security management information according to the main functions provided by the network management system. In general, the design and the implementation of the network management database for the network management system have been done according to the network equipment kinds (such as switching, mobile and transferring equipment).

However, the Integrated Network Management System is recently being required as communication service systems are integrated on the basis of IP technologies. It is important to integrate different network interworking-protocols and network management data for the development of the integrated network management system. Out of network management data, performance data are important factor for monitoring the efficiency of network services and extending or maintaining networks. Performance data have the most complicate parameters and structures, and so it has resulted in the biggest trouble in database integration. The design of conventional performance data tables have respectively defined according to measurement types of each equipment, and so many tables have been defined.

The [Figure 1] shows an example of conventional performance data table for the CORE (MSC, MGW, SGSN, GGSN) network system of WCDMA. For the CORE network management system of WCDMA, more than 100 performance data tables by measurement types have been defined and used [8][9]. If performance data tables are defined according to measurement types in the development of integrated network management system as shown in [Figure 1], many tables are required. In addition, whenever new network equipments are added, the network management program should be made up to manage new performance data tables.

This paper proposes a new table schema to integrate and manage complex performance data and comparatively analyzes data of processing performance on the proposed table.

Measurement-type Independent Table Design

- Design goals for Unified Performance Data Table
 - Measurement-type independent table design
 - Standardized DB I/O library(API)
 - Low overhead in hard disk space
 - Good performance in data search
- Performance data need many tables for many measurement types
 - Each parameter has different name and type
 - The number of parameter of each measurement type is different
- Unified performance data table
 - Simplify parameter names like P1, P2 ...
 - Simplify data type as VARCHAR
 - Define a maximum number of columns

```
CREATE TABLE pm_L_base (  
  NE_ID          NUMBER(10)          NOT NULL,  
  LLOC          VARCHAR2(100)        NULL,  
  VERSION       VARCHAR2(20)        NULL,  
  MEAS_TYPE     NUMBER(5)           NOT NULL,  
  EVENT_TIME    DATE                NOT NULL,  
  RESERVE_STR   VARCHAR2(20)        NULL,  
  P1            VARCHAR2(20)        NULL,  
  P2            VARCHAR2(20)        NULL,  
  P3            VARCHAR2(20)        NULL,  
  P4            VARCHAR2(20)        NULL,  
  P5            VARCHAR2(20)        NULL,  
  P6            VARCHAR2(20)        NULL,  
  P7            VARCHAR2(20)        NULL,  
  P8            VARCHAR2(20)        NULL,  
  P9            VARCHAR2(20)        NULL,  
  P10           VARCHAR2(20)        NULL,  
  :            :                   :  
  :            :                   :  
  P95           VARCHAR2(20)        NULL,  
  P96           VARCHAR2(20)        NULL,  
  P97           VARCHAR2(20)        NULL,  
  P98           VARCHAR2(20)        NULL,  
  P99           VARCHAR2(20)        NULL,  
  P100          VARCHAR2(20)        NULL  
)
```

[Figure 2] Unified Performance data table

2. Integration of Performance Data

The integration of performance data aims to design a table independent of measurement types of network equipments. Once the independent table is designed for measurement types of network equipments, the standardized management of performance data and easy operation to add equipments can be available in integrated network management system.

The detailed design goals for unified performance data table are as follows;

- Measurement-type independent table design
- Standardized DB I/O library(API)
- Low overhead in hard disk space
- Good performance in data search

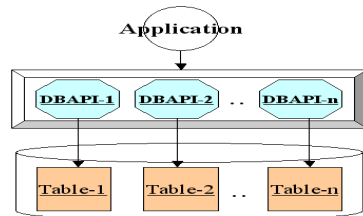
2.1 Measurement-type Independent Table Design

Many tables have been required for performance data by measurement types because each measurement type has different number of parameter, which has different name and different type. However, The unified performance data table that this paper proposes can improve complexity of the conventional performance data tables by simplifying parameter names, simplifying data type as VARCHAR and defining a maximum number of columns. Unified performance data table can be applied to the measurement types of all network equipments. The [Figure 2] shows unified performance data table independent of measurement types. The measurement type in the performance data table is classified with value of MEAS_TYPE(Measurement Type) column.

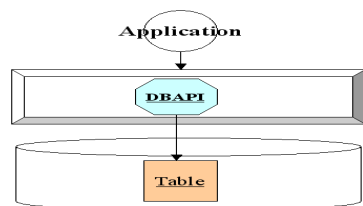
DB I/O Library & Hard Disk Space

Standardized DB I/O Library(API)

- Conventional DB I/O Library



- New DB I/O Library



Low Overhead in Hard Disk Space

- Unified Performance Data Table
 - maximum number of columns to store all measurement types
 - maximum size of columns to store all measurement values
- NUMBER Type (Fixed Length)
 - Disk space is wasted as many of the allocated columns are not used
- VARCHAR Type (Variable Length)
 - Disk space varies according to the number of columns and the size of each value
- No Disk Fragmentation in VARCHAR type
 - Performance data are hardly changed

APNOMS 2005

(4)



2.2 Standardized DB I/O Library

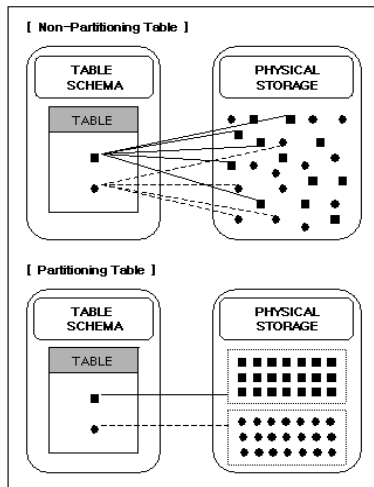
In general, when the network management system is implemented, DB I/O library is used for handling data (INSERT, DELETE, UPDATE, QUERY) to/from DBMS. When tables are defined on the basis of measurement types with conventional way, the different DB APIs (application Programming Interface) are required for each table, and network management application should be updated for the addition or the modification of DB API. However, all performance data can be processed through standardized DB API without modification of network management application, because unified performance data table proposed in this paper can be applicable to all measurement types of performance data.

2.3 Low Overhead in Hard Disk Space

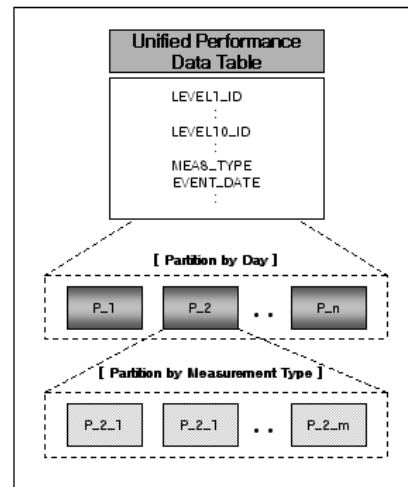
In order to store all measurement types of performance data into an unified table, which should include the maximum numbers of parameters. As a result of this definition, disk storage space may be wasted as many as the numbers of the columns that do not use. For the solution of this problem, the unified performance data table defines its columns as VARCHAR (Variable Length Character) Types. Unlike the NUMBER Types, the VARCHAR Type columns are variably stored in a disk depending on the size of the data to be stored. In other words, the columns that do not use do not occupy the disk storage space.

VARCHAR Type columns can variably use disk storage space according to the size of the data to be stored but may cause the fragmentation of the storage space when the stored value is changed. However, disk fragmentation does not occur for the unified performance data table in hard disk space because the stored performance data are hardly changed.

Good Performance in Data Search



[Figure 3] Comparison of Partitioning Table and Non-partitioning Table



[Figure 4] Applying the Partitioning technique to Unified Performance data table

APNOMS 2005

(5)



2.4 Good Performance in Data Search

The unified table may be estimated to lower the data search performance by storing too many data in single table. To solve this problem, the table partitioning technique is applied. The table partitioning technique clusters data to store the data concerned to a physical space. These grouped data structure can improve the searching speed [1][3].

The [Figure 3] compares a partitioning table and a non-partitioning table. In the non-partitioning table, performance data are randomly written down to a physical storage space, and so the search performance is lowered when many kinds of performance data are stored. On the other hand, in the partitioning tables, data are stored in the successive physical storage according to data groups (by generation time and measurement type of performance data) and the searching speed can be improved [4].

Performance data are collected from network equipments at every collection period (5 min, 10 min. and 15 min.) and stored in DBMS. Generally, the network management system automatically searches old data at every fixed time and removes them from the performance data table to store new performance data to a restricted physical storage space. If a lot of old data are daily or monthly removed by the delete command of SQL, the load of DBMS is significantly increased. On the other hand, the partitioning table can remove data in partition unit, and so it can remove data without affection on the performance of DBMS [3].

The [Figure 4] shows a way to apply the partitioning technique to the unified performance data table that this paper proposes. In this way, performance data are partitioned by generation date and the daily partitioned data are sub-partitioned by measurement types. Since the performance data are partitioned on the basis of generation date and measurement types, the searching speed in the unified performance data table can be improved significantly.

Unified Performance Data Table

- Unified Performance Data Table
 - Apply VARCHAR data type
 - Apply range partition based on the event_time of performance data.
 - Apply sub-partition based on the measurement type of performance data.
 - Apply dynamic partition allocation

```

CREATE TABLE pm_t_base (
  NE_ID          NUMBER(10)          NOT NULL,
  LLOC          VARCHAR2(100)        NULL,
  VERSION       VARCHAR2(20)        NULL,
  MEAS_TYPE     NUMBER(5)           NOT NULL,
  EVENT_TIME    DATE                NOT NULL,
  RESERVE_STR   VARCHAR2(20)        NULL,
  P1            VARCHAR2(20)        NULL,
  P2            VARCHAR2(20)        NULL,
  P3            VARCHAR2(20)        NULL,
  P4            VARCHAR2(20)        NULL,
  P5            VARCHAR2(20)        NULL,
  P6            VARCHAR2(20)        NULL,
  P7            VARCHAR2(20)        NULL,
  P8            VARCHAR2(20)        NULL,
  P9            VARCHAR2(20)        NULL,
  P10           VARCHAR2(20)        NULL,
  :            :                   :
  :            :                   :
  P95           VARCHAR2(20)        NULL,
  P96           VARCHAR2(20)        NULL,
  P97           VARCHAR2(20)        NULL,
  P98           VARCHAR2(20)        NULL,
  P99           VARCHAR2(20)        NULL,
  P100          VARCHAR2(20)        NULL
)
partition by range(EVENT_TIME)
subpartition by hash(MEAS_TYPE) subpartitions 10
(
  partition PBD [current_day] values less than (to_date('next_1day', 'yyyymmdd')),
  partition PBD [next_day] values less than (to_date('next_2day', 'yyyymmdd'))
)
    
```

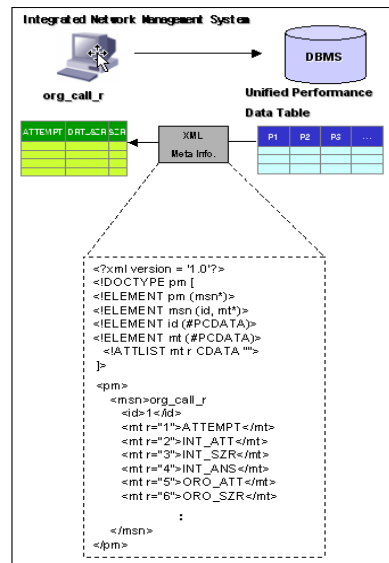
[Figure 5] Unified performance data table that partitioning technique is applied to

2.5 Unified Performance Data Table

The [Figure 5] shows the schema of the unified performance data table that partitioning technique is applied to. On the basis of the generation date of EVENT_TIME column and the measurement type of MEAS_TYPE column, range partition and sub-partition are applied separately. In addition, dynamic partition allocation is applied to improve the efficiency of the limited physical storage space. In the creation of unified table, some numbers of partition are created dynamically according to the configuration of the keeping term for the performance data instead of static number of partition[1]. When the unified performance data table is created, two partitions, i.e. day and next day, are basically created by EVENT_TIME column as a partition key and the partitions are added/removed according to the configured keeping term.

Utilization of XML for Displaying Performance Data

- Operators cannot understand the meaning of columns in unified performance data table
 - Column name is logically defined like P1, P2 ...
- DB Metadata with XML
 - Map logical column names to parameter names for each measurement type
 - Add new measurement type without changing the software



[Figure 6] Utilization of DB metadata by XML in Unified network management system

APNOMS 2005

(7)



3. Utilization of XML for Displaying Performance Data

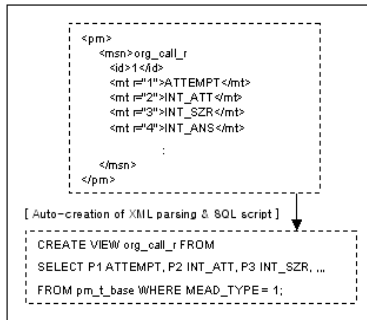
In the unified performance data table, column names are defined as logical names instead of measurement parameter names, and so it can include all measurement types. However, there occurs a problem that the operator cannot understand the contents of parameters from logical names.

The [Figure 6] shows an example that DB metadata with XML is utilized to convert the logical columns of the unified performance data table into parameters by measurement types. XML (eXtensible Markup Language), which was proposed in 1996, is a kind of Meta language that expresses information to make all computers and most people understand[9]. The integrated network management system displays parameter names and parameter types by measurement types through the pre-defined DB metadata with XML. The use of DB metadata with XML for measurement types enables to manage the performance data of new network equipments by only updating DB metadata, not changing the network management system program.

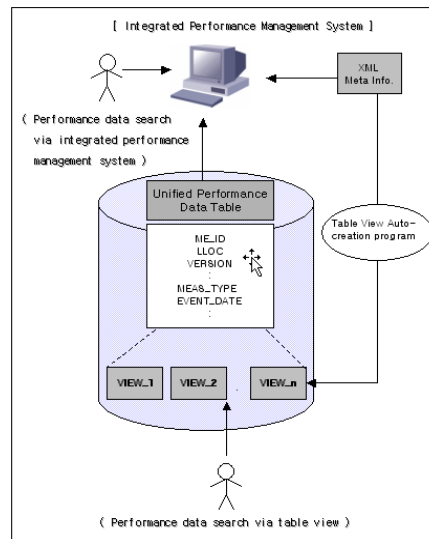
As shown in [Figure 6], the performance data are classified by the `<id>` tag and displayed as the measurement type name of `<msn>` tag, and logical names such as P1 and P2 of each parameter column are displayed as the measurement parameter name of `<mt>` tag in DB metadata with XML.

Supporting ad-hoc Query with Table View

- Ad-hoc Query
 - User defining Query
 - Not fixed Query in Program



[Figure 7] Auto-creation of Unified Performance Data Table View



[Figure 8] Unified Performance Data Management Structure using DB metadata by XML

APNOMS 2005

(8)



4. Supporting ad-hoc Query with Table View

The management of performance data is allowed through the performance management GUI(Graphic User Interface) of integrated network management system using DB metadata, but operator may want to directly access the performance data from table by using ad-hoc query. Ad-hoc query means user-defining query, not fixed query in program. When operator directly accesses the unified performance data table, operator cannot classify measurement type and parameter value of performance data. For this problem, the table View is defined. In DBMS, the View is a virtual table that does not exist in physical situation. By defining the View having real parameter name of performance data, operator can use the View to search performance data from the DBMS.

The table View is automatically created via DB metadata. The measurement types and the measurement parameters defined in DB metadata are mapped as View names and View column names, respectively. The [Figure 7] shows an example that creates the View of org_call_r performance data defined in DB metadata. The performance data table View is automatically created by the View creation program.

The [Figure 8] shows the whole structure of unified performance data management system. Operator can add/remove performance data by simply editing DB metadata. Operator can search performance data via the performance management GUI of the integrated network management system and, also, directly search performance data through performance data table View.

Experiment for Search Performance and Storage Space

- Experiment environment
 - H/W : Sun Blade 1000
[CPU: 900 MHz*2, MEM: 2 G]
 - DBMS : Oracle 9i
- Experiment conditions
 - Number of measurement types : 10 types
 - Number of parameters/ measurement types : 20 ~ 100 parameters
 - Stored records per day for each measurement type : 10,000 records / day
 - Experiment period : 30 days
 - Total stored records for each measurement type : 10 * 10,000 * 30 = 3,000,000 records / 30 days
- Evaluation factors
 - Search performance : measure and compare search time for data per day of one measurement type
 - Storage space : measure and compare the hard disk space required to store 300,000 records with 20 ~ 100 parameters
- Search performance test case
 - Test-1 : measure the search performance of conventional performance data table
[No partition, No Index, Full Scan]
 - Test-2 : measure the search performance of conventional performance data table
[Partition by day, No Index, Full Scan]

SQL for Test-1 , Test2 :

```
select avg(p1) from org_call_r where
event_time = to_date('2005-03-15', 'yyyy-mm-dd');
```

 - Test-3 : measure the Unified performance data table without the partitioning technique
[No partition, No Index, Full Scan]
 - Test-4 : measure the Unified performance data table with the partitioning technique
[Partition by day and measurement type, No Index, Full Scan]

SQL for Test-3 , Test-4

```
select avg(p1) from pm_t_base where
meas_type = 1 and
event_time = to_date('2005-03-15', 'yyyy-mm-dd');
```

APNOMS 2005

(9)

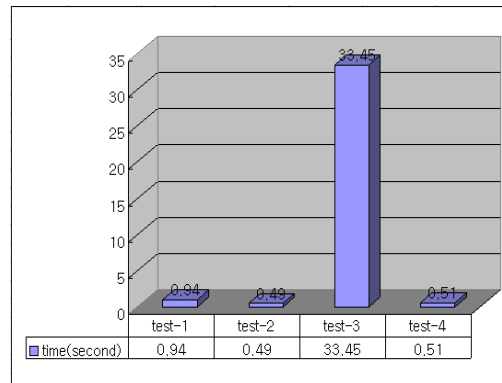


5. Experiment for Search Performance and Used Storage Space

Search performance and used hard disk storage space are evaluated for the unified performance data table and conventional performance data table. Above detail experiment plan is described to evaluate the search performance and hard disk storage space for performance data. With a simulator, overall 3,000,000 records of performance data for 30 days are generated and stored by 10,000 records per day. For conventional performance data table and unified performance data table, the searching time for the performance data of specific measurement type for a day is repeatedly measured and evaluated 10 times. In addition, the occupancy in a hard disk to store 300,000 records to conventional performance data table and unified performance data table with 20~100 parameters separately is measured and evaluated.

Test Result of Search Performance

- Unified performance data table
 - Store 3,000,000 records for 30 days
 - Evaluate search time to query 10,000 records of 1 day from 3,000,000 records
- Conventional performance data table for one measurement type
 - Store 300,000 records for 30 days
 - Evaluate search time to query 10,000 records of 1 day from 300,000 records
- Evaluation Results
 - Test-3 : Unified performance data table without partitioning technique is the worst as expected
 - Test-2 : Unified performance data table with partitioning technique is the best
 - Test-4 : Unified performance data table with partitioning technique is the second but there is no big difference between Test-2 and Test-4



[Figure 9] Test Result of Search Performance

6. Test Result of Search Performance

This test is repeated 10 times under the experimental procedure previously presented and the result is given in [figure 9]. According to the results, the search performance of the conventional performance data table with partitioning technique(test-2 : 0.49 second) shows best result and the unified performance data table with partitioning technique(test-4 : 0.51 second) is the second. The unified performance data table without partitioning technique(test-3 : 33.45 second) is the worst as expected because of the unified performance data table includes 3,000,000 records by 10 times than the conventional performance data table. However the unified performance data table with partitioning technique shows good result as the conventional performance data table with partitioning technique. There is no big difference in search performance between test-2 and test-4, and so we can conclude that the unified performance data table with partitioning technique provides good search performance with standardized management of complex performance data.

Test Result of Storage Space

- No overhead in storage space by unused columns in unified performance data table

| Number of Parameter | 40 VARCHAR Type Columns (Mbytes) - A | 100 VARCHAR Type Columns (Mbytes) - B | B - A |
|---------------------|--------------------------------------|---------------------------------------|-------|
| 40 | 64 | 64 | 0 |

[Figure 10] Test Result of Storage Space Usage -1

- VARCHAR Type may use more storage space than NUMBER Type depending on the saved values
- Overhead to use unified performance table is acceptable
 - Complication of performance data management.
 - Difficulty of standardization.

| Number of Parameter | NUMBER Type Table (Mbytes) - A | VARCHAR Type Table (Mbytes) - B | B - A |
|---------------------|--------------------------------|---------------------------------|-----------|
| 20 | 41 | 46 | 5 (12 %) |
| 40 | 54 | 64 | 10 (18 %) |
| 60 | 80 | 96 | 16 (20 %) |
| 80 | 104 | 128 | 24 (23 %) |
| 100 | 128 | 160 | 32 (25 %) |

[Figure 11] Test Result of Storage Space Usage -2

7. Test Result of Storage Space

This test analyzes two kinds of usage of storage spaces. In the first experiment, hard disk usage is checked whether its storage space is wasted when all VARCHAR Type columns of the unified performance data table are not used. In the second experiment, the usage of the hard disk storage space for the unified performance data table and the conventional performance data table are compared and evaluated.

In the first experiment, the hard disk usages are compared to store 300,000 data to table with 40-VARCHAR type columns and table with 100-VARCHAR type columns. The result is shown in [figure 10]. The test result shows that hard disk storage space is not wasted by unused columns when data are stored in the unified performance data table defined as VARCHAR type of 100 columns.

In the second experiment, the hard disk usages are compared to store 300,000 data to conventional table with NUMBER type columns and the unified performance data table with 100-VARCHAR type columns. The result is shown in [figure 11]. When we analyze the overhead of storage space usage for the unified performance data table on the basis of the storage space usage for conventional performance data table, the biggest overhead (25%, 32Mbyte) and the smallest overhead (12%, 5Mbyte) are generated with the numbers of parameters 20~100 separately. If NUMBER Type column and VARCHAR type column store 6-digit value, the NUMBER type column generally occupies a fixed 4 Byte storage space and the VARCHAR type column variably occupies more than 6 Bytes considering future modification. In other words, more hard disk storage spaces are occupied for the VARCHAR type columns comparing with VARCHAR type columns. According to the test result, we know that the unified performance data table may occupy more hard disk storage space than the conventional performance data table with NUMBER type columns because of the difference of storing method between NUMBER type data and VARCHAR type data.

From the test result, on the basis of 100 columns, the unified performance data table of VARCHAR type may use up to 25% more hard disk storage space than the conventional performance data table. However, it can be acceptable cost if considering the complication of performance data management and the difficulty of standardization.

Conclusions

- Utilizing VARCHAR type to standardize management of complex performance data
- Simple & Single Unified Performance Data Table to store all performance data for all network equipments
- Flexible management structure to easily add/remove performance data by DB metadata with XML
- Good search performance using table partitioning technique

8. Conclusion

This paper presents a structure and a management method that can easily add/remove the performance management data of network equipments by standardizing/simplifying the performance management function that had been complicatedly designed and implemented in the construction of network management system. In addition, good search performance is implemented even for the unified performance management data table by applying table partitioning technique and flexible management structure is proposed, which can easily add/remove performance data by means of DB metadata with XML without any modification of network management programs.

References

- [1] Jeon Eung-sun, Ahn Il-su, "How to Manage High-Capacity History Data via Partitioning Table Technique", KNOM Review, Vol. 6, No. 1, pp. 52-58, Jun, 2003.
- [2] Divakara K. Udupa, Network Management Systems Essentials: 413-513.
- [3] Loney Theriault, Oracle9i DBA Handbook: 681-690.
- [4] Peter Scheuermann, Gerhard Weikum, Peter Zabback: Data Partitioning and Load Balancing in Parallel Disk Systems. VLDB Journal 7(1): 48-66(1988)
- [5] ITU-T Recommendation M.3010, Principles for a telecommunication management network, 1996
- [6] 3GPP TS 32.401, Performance Management Concept and Requirements, 2001
- [7] 3GPP TS 32.403, Performance Measurements UMTS and combined UMTS/GSM, 2001
- [8] 3GPP TS 52.402, Performance Measurements - GSM, 2001
- [9] W3C Recommendation, Extensible Markup Language (XML) 1.0, 2004