SLMOS 데이터베이스의 시스템 構成 및 分散運営 設計

Design of the Architecture & Distribution Management for Database of SLMOS

李暻休 / 網運營研究室

<Abstract>

The role of the database system of the Subscriber Line Maintenance Operations System (SLMOS) is to maintain up-to-date information about the subscriber's telephone service and trouble history to facilitate subscriber trouble repair. The discussion covers the architecture and the distribution management for the database system.

〈요 약〉

가입자시설 집중보전 시스템(SLMOS)의 데이터베이스 역할은 가입자 시설에 대한 모든 정보를 유지하며 최근의 정보를 신속히 변경, 추가해야 한다. 또한 수시로 발생하는 고장 정보에 대한 유지가 필요하다.

이에 대한 데이터베이스 시스템은 SLMOS의 구조와 UNIX 시스템 구조와의 양립성을 파악하여 이 기반위에 데이터베이스 시스템을 구성함으로써 목적을 달성할 수 있으며, 시스템의 확장과 운용상의 효율성에 대비한 분산처리 방식을 결합구성함으로써 SLMOS에 적합한 데이터베이스 시스템 구성을 보인다.

I. Introduction

The database system of SLMOS had to accommodates the following major repair functions:

1) Taking trouble reports
   The operational objective was to display, in five seconds or less, information about a subscriber's telephone service when the subscriber contacted the Centralized Repair Service Answering Bureau.
2) Tracking open troubles
   The system had to provide a capability for accepting new trouble reports and maintaining status information about the troubles until closed out.
3) Maintaining trouble history
   The most recent forty days of closed trouble information had to be maintained in the database for summary and review purposes.
   Afterwards, the trouble history data could be transferred to magnetic tape storage.
4) Maintaining up-to-date line record data
   Changes made to the subscriber's telephone service had to be reflected (typically with in 24 hours) in the SLMOS databases.

5) Maintaining up-to-date cable information
Change made to each office's cable information. This is used to process the management in the unit of pair in cable.

By the above function, in SLMOS, there are five types information in the database system:

1) subscriber line information (CLR)
2) cable information (CAB)
3) trouble report information (TR)
4) abbreviated trouble report information (ATH)
5) trouble report analysis information (TRT)

The above information are divided to two groups, the one (1~4) is the operation information, the other (5) is the statistic information.

The operation information requires the database system to provide rapid access to data and to manage volatile trouble report information while the trouble is being processed.

The statistic information is to provide the trouble report analysis in period of daily, weekly, monthly and yearly.

So, in SLMOS database system, there are two different database systems. Figure 1 gives the SLMOS database system operation architecture.

In the next main subject, it describes about the design considerations, the external view, the internal view, data structure, and software structure about the database system design.

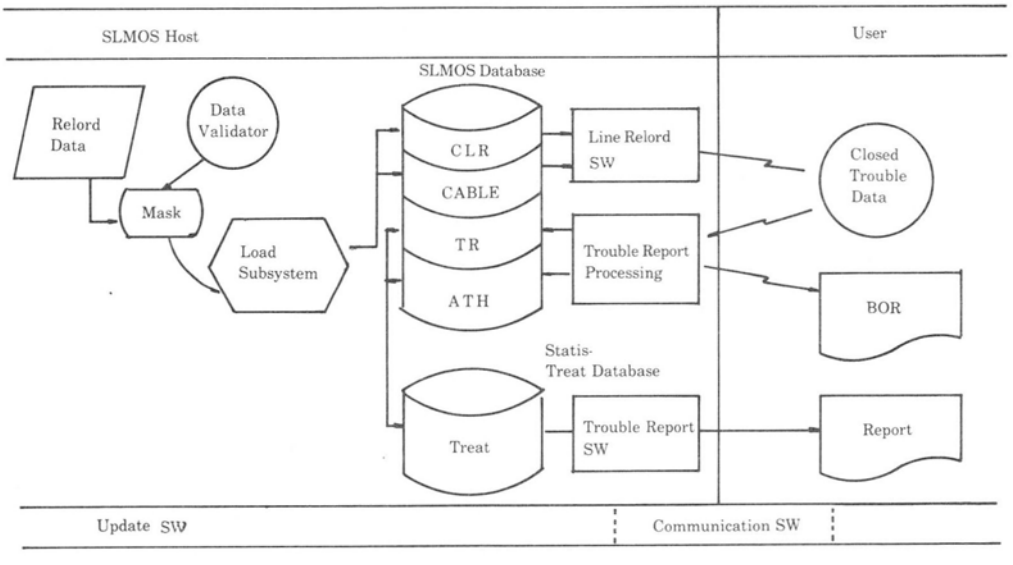
II. Subject

1. System Organization and Design Consideration

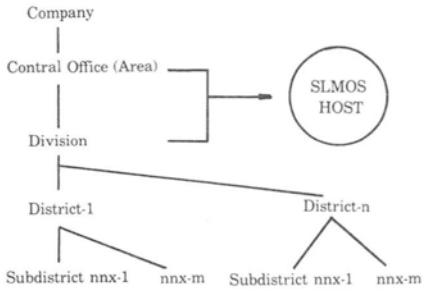
Figure 2 illustrates the SLMOS installation organization. The root exists superfically, and means the control management in the all around system in Korea Telephone Company. SLMOS is operation on the centralized office which covers all districts and has the host computer system.

SLMOS host database in the central office must have all informations for all districts and maintain information provided by trouble report processing. Therefore, the SLMOS host database stores in a large "SLMOS database" complete information on the subscriber's telephone service including premise equipment data; class of service; network facilities assigned to the subscriber's circuit; and trouble history.

The purpose of the design SLMOS database system must be done by the efficient control about



<Fig. 1> SLMOS Database System Operation Architecture



<Fig. 2> Administration Hierarchical

the administration hierarchical. In this aspect, it is necessary to be processed by the distribution management and design.

This administration hierarchical is similar to the UNIX file system. Therefore, the design of SLMOS database system on the UNIX file system is done by some compatibilities.

2. The UNIX File System

A file system is the abstraction of a disk provided by the operating system, although the space used by a file system can be done only part of a disk, or several disks. To the user, a file system consists of a tree of directories, each directory containing the names of other directories and names of files.

To the operating system, a file system has three parts, a super block at the beginning containing descriptive information, a contiguous collection of blocks containing inodes, and the bulk of the file system containing all the file and directory blocks, and some other blocks.

A file is an indefinitely extensible vector of bytes with operations read, write, and seek. A directory is a file, except that users can not write in one, and the operating system interprets the contents as an array of (inode, file-name) pairs. These pairs provide the mapping from file names to inode numbers. The inode number says which inode in the array of inodes in the file system describes the file. The inode for a file contains information about the file's type and accessibility, its length, and where on the disk the data are.

The first ten entries in the array are the addresses of the first ten data blocks of the file. The next is the address of an indirect block, which is filled with addresses of data blocks. The next is the address of a doubly indirect block, which is filled with addresses of indirect blocks, and the last is the address of a triple indirect block, which is filled with addresses of doubly indirect blocks. The addresses of the free blocks in the file system are kept in some of the free blocks, which are chained together in a list.

The operating system maintains the list of free blocks approximately as a stack.

3. Architecture Overview

In figure 3, there are five types in the view, the view is divided into a set of the external view and the internal view. The database system level in the view is classified by user, application, Database Administrator (DBA), physical level.

The user can process all information on the SLMOS mask independent of the database structure.

The application is provided data to user by packet buffer provided by DBA. The physical level contains contents of the UNIX file system. The structure of physical level in the UNIX file system has one tree structure which is equal to the following diagram (Figure 4)

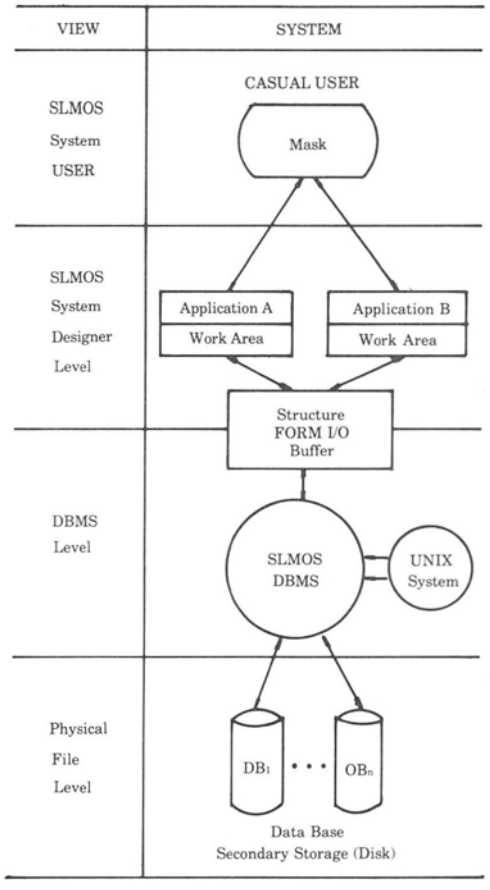
This distribution control are implemented as distribution key in the Data Manipulation Language (DML).

The administrative hierarchical system, database hierarchical, and the UNIX file system have the characteristics of the compatibility.

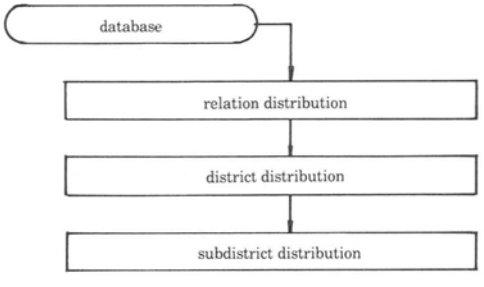
In the next step, the detail design illustrates as the following step.

4. Data Manipulation Language (DML)

A Data Manipulation Language (DML) is provided to the application programmers as a pro-



<Fig. 3> SLMOS Database System Level Architecture



<Fig. 4> The Hierarchical Architecture in Physical Level

gram interface to the database.

The host language for the DML is the programming language C.

The DML is a set of C function for control (retrieval, append, replace, and deletion, etc) of records. Most of the DML commands have the following format:

```
dbopen (database-id, accessing-mode, distribution-key);
setid=dbms (database-id, command-mode, accessing-key);
dbclose (database-id);
```

These three DML commands are used in the application program which controls the operation for the trouble processing.

The dbopen command is the initialization course for the using database. Database-id is one of the operation information, CLR, CAB, TR, ATH. The accessing-mode is one of the read, write, and read & write. The distribution-key is one of the distribution navigation identifier in the figure. The command-mode is a set of commands for retrieval, append, replace, and deletion of records in the each distribution database. The accessing-key is defined by key for searching telephone number in the CLR, TR, ATH, and cable identifier in CAB database. The setid is the identifier returned by the each command-mode that is operating.

The application programmer can see the status of command-mode operation. If the setid is 0, this command-mode operation is done normally. The dbclose command is the end of database manipulation.

The following DML calls may be used:

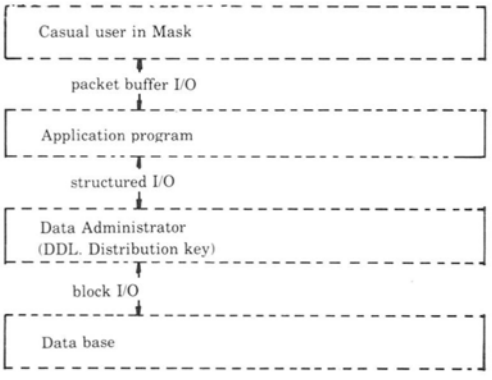
```
dboupen (S-CLR, 2, "822");
if (dbms (S-CLR, REPLACE, "8221234") == 0)
prinft ("%s %s\n", clr. tel, clr. name);
dbclose (S-CLR);
```

The above example is a DML command for getting the line record information about tel# 8221234. The statistic DML for the trouble report analysis is the following form:

```
Topen (District, mode, day);
Tappend ( );
Tret ( );
Tclose ( );
```

5. Data Definition Language (DDL)

The Data Definition Language (DDL) is provided for the Database Administrator (DBA) to interface with the database. The interface in SLMOS Database system is shown the following figure 5.



<Fig. 5> Interface in SLMOS Database System

The following items are the parameters of DDL.

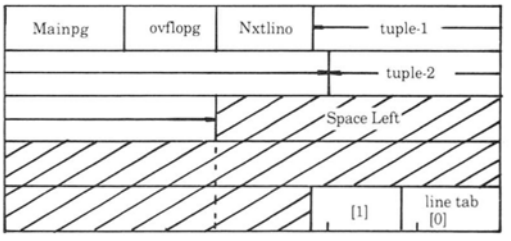
Relation-name #Distribution-name Domain-name Domain-type Domain-size

There are consisted of the structure table in the head of the database software module. This structure tables are shown as the following:

```
#external definition
  #relation-name #domain-name
      #domain-length
      #domain-type
  #struct-size
#internal definition
  #relation-id #max-domain
      #domain-length
      #domain-type
      #domain-offset
      #relation-name #key-size #tuple-length
```

6. Internal View and Data Structure

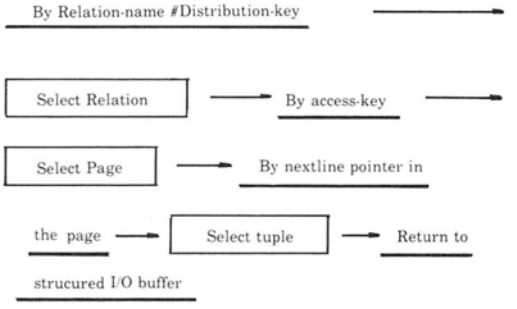
The internal data structures include the following items: main page status, overflow page status, next line numbers, first tuple, line tab, this page and buffer status. The data structure for the above items are the following form (Fig. 6)



<Fig. 6> Data Structure in Each Page

The tuple is the fixed length records and the other information is variable in one I/O page. The internal data structures are designed for high performance and conservation of disk space. The variable record format also provides a mechanism for achieving flexibility.

The mapping algorithm has a hash structure. First, it is selected relation file including internal data structure by the distribution key. Second, the accessing-key in the first domain of tuple is transferred to hash algorithm routine. The hash algorithm routine returns the value of page which is placed tuple to be selected or to be updated. The following diagram illustrates the course for searching any tuple (Fig. 7)



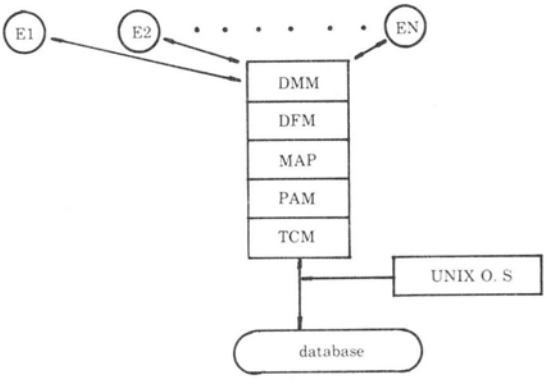
<Fig. 7> Mapping Cource

7. Software Architechture

The SLMOS DBMS software has a modular design that minimizes dependencies among the software modules. The DBMS software is functionally divided into modules: DDL Manipulation Manager (DMM), Data Format Manager (DFM), Mapping Function (MAP), Page Access Manager (PAM), Tuple Access Manager (TAM), and other routines.

The interface between two modules is simple. An interface is a set of C function calls specifying the operations for particular module to perform.

The graphical representation of the software architecture for DBMS and the surrounding processes is shown in figure 8, where E1, ... , En represent the external view in the application processes.



<Fig. 8> Software Architecture

These are the combination program between the SLMOS operation control program and Database Manipulation Language.

The interface between the external view and internal view is consisted of the each common structured I/O form.

The DMM handles the demand of the application program, and retrieved the necessary parameters from the application program. Also, after the below functions are done, the information provided by the block I/O buffer is converted to structured I/O form by the DFM.

The MAP handles the path of distribution management, and the mapping by using the accessing-key in the internal data structure. The mapping function performs the translations that have been described previously. The information in the page selected by the MAP is handled by PAM which is performed by reading the blocks and searching for the optimal page. This information includes the demanding tuple.

The accessed tuple is controled by TCM which decides the contents of the domain in this tuple.

8. Performance

In general, a database management system has overhead to an application system in the three parts:

(a) processing time
(b) mapping algorithm
(c) data structure

The most times for SLMOS database operation are dependent on the processing time which is charcterized by SLMOS user process. Because the UNIX system is the timesharing system. But this SLMOS DBMS is performed SLMOS operation without any overhead. The average time of ten thousands subscribers line data manipulation in the sequence of telephone number is about 90 seconds.

III. Conclusion

The SLMOS DBMS provides a flexible production application. In spite of the new relation increasement, application program can be used without reprogramming and changing the database management software.

Application programs are easier to write because of the encapsulation of internal structures.

The many factors of SLMOS DBMS is designed by the modularity independent of each factor. But the physical level design is dependent on the UNIX file system.

The idea of distribution management is contributed to the performance in the administrative hierarchical management system likes as SLMOS. The importance of data models, modular architecture, performance, and manageable software components is considered on the SLMOS DBMS design.

The future database systems will continue the evolution toward the complete concurrency and crash management.

<References>

1. "Automated Repair Service Bureau", B.S.T.J., 1982.
2. "Database Systems", B.S.T.J., 1982.
3. SLMOS Manual, 1984.
4. The UNIX System Manual, Vol. 1, 2A, 2C.
5. Kernighan, B.W., and D.M. Ritchie, The C Programming Language, Prentice Hall, 1978.
6. Ritchie, D.M., and K. Thompson, "The UNIX Time Sharing System", B.S.T.J., 1982.