A Selection of Data Structure for SMART Alarm System Database

Duk-Hyun Seong*, Gwi-Sook Jang**,
Jong-Hyun Ko*, Young-Rok Sim*, Geun Ok Park**, Jung-Seon Kim*, In-Soo Koo**
*Samchang Enterprise Co.**
*Korea Atomic Energy Research Institute***
sdh1331@samchang.com

1. Introduction

A design goal of SMART Alarm System is providing inteligence alarm information to operator in main control room. To achive this, we should apply advaced alarm process logics and manage alarm data sets for advanced alarm logic.

SMART Alarm System must analyze a lot of alarm by the cycle to determines alarms. For this, performance optimization of database is essential. Especially, high performance of search function is required.

In this paper, we propose most a suitable search method to database by compare several search methods.

2. Data Structures for Database

In this section, we describe the existing data structures used in database, such as B tree, B+ tree, T tree and Hash.

- **B tree** has been used in most database. It keeps all of the actural data in the nodes of the tree.[1]
- **B+ tree** is optimizer than B tree for continuouslly data process. And it keeps all of the actural data in the leef nodes of the tree.[1,2]
- **T tree** has more depth and narrow width than B(+) tree. So it is more suitable in memory based database.[3,4]
- **Hash** is most suitable in static data structures and all functions are excuted in fixed time.[1]

3. Charactistics of Alarm System Database

We are assume that SMART Alarm System has below charateristics.

- Search function is frequently more than performed than others funtions(entity creation, modification and deletions) on on-line. But change of values are frequently happened.
- It should has fast excution time and be satisfied real-time charactistics. So it should use as possible as simpler and faster data structure.

4. Test for Selection of Data Structure

We described data structures and requirement of alarm system in Section 2, 3 and we'll describe test procedure for selection of data structures in this section.

4.1 Test Environment

We used prototype for SMART Alarm System in this test. The prototype is composed of target system with host system.

Target System Environment
CPU: TMS320C40(interner clock: 30 MHz)
Operating System: Not Applicable
Main-Memory: 2MB(local + global)
External-Interface: RS-232

Host System
PentiumIII based General PC
Operating System : windows98
Management S/W: AIS2000

Software Development
Language: Assembly, C, C++
Tool: CodeComposer4, Visual C++6.0

4.2 Software Renewal

We modified exist function or add fuction for this test. Signal process software was added new fuction to described data structres in Section 2. Especially, we had to impliment timer fuction for excution time measuring. Because, Alarm System has had neither operating system nor external timer. For this, we made a interrupt service routine by interrupt function in CPU. It is implemented to change time variable every 1ms. We changed fuction in management software(AIS2000) for result monitoring. Because it hasn't any display for user interface. They communicate via serial cable(RS-232) for data-transfer.

4.3 Criteria of Test

According to Section 3, the most important characteristics are for Alarm System Database as following.

Excution time of search process
It is suit to satisfy real-time charateristics that excution time is short and consistente.[7]

Simplicity of data structure

In generally, simple data structures have fast excution time[1]

Avoidance of worst-case for data handling

Excution time comes fast as avoid worst-case.[1,5]

4.4 Test Description

We used data table for the search which has only index filed for the data search. Real index values were 9-bytes character string and it was determined by random number generator. The tests had been run in the following order.

1. Insert 10,000 elements
   The insert vallue were 9-bytes randomized string. And each insert operation involved a search to ensure that the insert value was not already exist.
2. Search for 10,000 elements
   We used 9-bytes randomized string value to measure the excution time in search process. Because, we could produce the worst case in search process.
3. Time check for the search elements.
   We checked consistency and speed of excution time
4. Evaluate simplicity of data structure
   We checked simplicity of data structure based on number of function. Because as complexity increase, number of function is increased. For example, B tree need comparison, division and mergence for insertion.[1,3]
5. Evaluate avoidance of worst-case
   We checked avoidance of worst-case in search function. The worst-case is to retrieve all data though it does not exist in data set.[1]

4.5 Result of Test

Table 1 show the summary of test result for data structres. According to the table 1, the hash is most suitable data structure for SMART Alarm System Database based on Section 4.3.

All tree structures have same excution time. Because B+ tree is almost same structure with B tree except continouslly data processing. And case T tree, the number of data is so small to show different between B tree and T tree. Of course, it can show difference in high resolution timer(ie use micro second timer). But we have used milli second timer and haven't care more detailed time.

Only hash has real time charatistic. Because others data structure has variable excution time. They need more excution time as increase number of data. But hash takes fixed excution time regardless of number of data.

All data structures except hash have complex sub-functions for main functions(Insertion, deletion and search). But hash doesn't need subfunctions for main fuctions.

And hash doesn't need to retrieve all data to confirm that data doesn't exist in data set. But others data structures need to retrieve all data.

5. Conclusion

In this paper, we proposed to suitable data sturcture for SMART Alarm System Database by performance tests. First, we set charactistic of Alarm System Database. And then, we used SMART Alarm System prototype and modified and added functions for test. Finally, we set criterias for evaluation of test result.

Based on the test result, we determined suitable data structure is Hash.

Table 1. Deviation of Data Structure

O: satisfy, X: not satisfy

	B tree	B+ tree	T tree	Hash
Excution Time(ms)	1~6	1~6	1~6	1~2
Real time charactistic	X	X	X	O
Simplicity of data structure	X	X	X	O
Avoidance of worst case	X	X	X	O

REFERENCES

[1] Tomas Cormen, Charles Leiserson, Ronald Rivest, Introduction to Algorithms, The MIT Press, 2001.

[2] Gruenwald. L, Eich. M. H, Choosing the best storage technique for a main memory database system, IEEE Information Technoloty, Vol. 1, pp. 1~10, 1990

[3] Hongjun Lu, Yuet Yeung Ng, Zengping Tian. T-tree or B-tree:main memory database index structure revisited, Database Conference, pp. 65~73, 2000

[4] Kong-Rim Choi, Kyung-Chang Kim, T*-tree:a main memory database index structure for real time applications, Real-Time Computing Systems and Applications, pp. 81~88, 1996

[5] Garcia-Molian. H, Salem. K, Main memory database systems: an overview, IEEE Knowledge and Data Engineering, Vol. 4, pp. 509~516, 1992

[6] Cui.B, Ooi.B, Tan. K, Main memory indexing: the case for BD-tree, IEEE Knowledge and Data Engineering Vol. 16, pp. 870~874, 2004

[7] Berryman. S. J, Sommerville. I, Modelling real-time constraints, IEEE Software Engineering for Rela-Time Systems, pp.164~169, 1991