

An Efficient Method to Search Real-Time Bulk Data for an Information Processing System

Seong Jin Kim,^a Jong myung Kim,^a Yong Suk Suh,^b Jong yong Keum,^b Heui Youn Park,^b
a Control Tech. Research Inst., SEC Co., Ltd., 974-1 Goyeon-ri Woongchon-myon, Ulju-gun, Ulsan, Korea, 689-871,
ksj1330@samchang.com
b I&C and HF Div., KAERI, 150 Dukjin-dong, Yuseong-gu, Daejeon, Korea, 305-353

1. Introduction

The Man Machine Interface System (MMIS) of System-integrated Modular Advanced Reactor (SMART) is designed with fully digitalized features. The Information Processing System (IPS) of the MMIS acquires and processes plant data from other systems. In addition, the IPS provides plant operation information to operators in the control room. The IPS is required to process bulky data in a real-time. So, it is necessary to consider a special processing method with regards to flexibility and performance because more than a few thousands of Plant Information converges on the IPS. Among other things, the processing time for searching for data from the bulk data consumes much more than other the processing times. Thus, this paper explores an efficient method for the search and examines its feasibility.

2. Method and Result

2.1 Data Processing Concept of IPS Server

The IPS consists of two systems: a data processing and storing system (server) and a display system (client). Its main functions are data acquisitions via a network, data-input processing, alarm processing, operational information processing, data-output processing, and a self-diagnostic processing. The structure of the processing modules in the server is shown in figure 1.

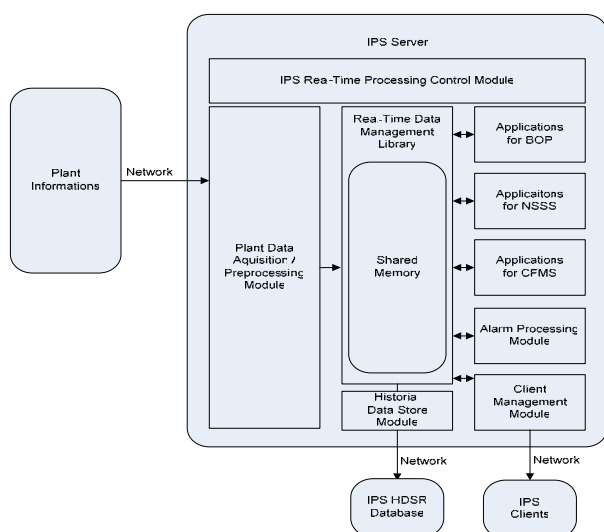


Figure 1. Data Processing Concept in SMART IPS Server

Basic concepts of the data processing feature:

- 1) A data acquisition module acquires plant data, and an input-processing module preprocesses the plant operational data into the useable data formats for applications. Then, it stores the data in the shared memory through the real-time data management library.
- 2) Each application processes the data, and its results are stored in a shared memory using the real-time management library.
- 3) After the processing finishes, the historical data storage module stores the data and operators can see the data from clients.
- 4) The real-time processing control module will determine the timing and sequence of the processing modules.

In these concepts, all the applications need a consistent software internal interface in order to access the plant information data and a flexible access method adapting the changes of the plant information. For this, the following features can be considered:

- 1) Plant data should be stored in a particular place within a shared memory, and this place should be determined by a Point Identification (PID).
- 2) Applications, which perform the processing, identify the place of the data in the shared memory through a search function with a PID parameter, and have an access to the shared memory.

By doing that, all the applications will be provided with a consistent software internal interface of the plant data. This can prevent interface errors in a software development. It also provides a flexibility in terms of a maintenance such as plant data additions, deletions, or changes. In order to achieve the consistent and flexible accessibility to the data in the shared memory, the following search requirements should be met:

- 1) Despite a data increase the search performance should not be degraded.
- 2) The search performance should be excellent in order to process data at the speed of the real-time.

We choose the Hash algorithm as a candidate search function. The Hash algorithm shows a better performance than other algorithms even if the data size is increasing [1].

2.2 Data Search Algorithm for IPS

The searching time of the Hash Algorithm is constant, that is $O(1)$. This means that every data can be searched in a constant time regardless of the amount of data. The constant time is determined at the time of a system initialization, which is the time of creating the Hash table. From our experiment, the algorithm should implement the chaining method in order to prevent any conflicts of the key values. We use an indirect address table method in order to separate the Hash table and the plant data table. The Hash table has its own address index and the plant data table keeps the list structure. The structure of the table is shown in figure 2.

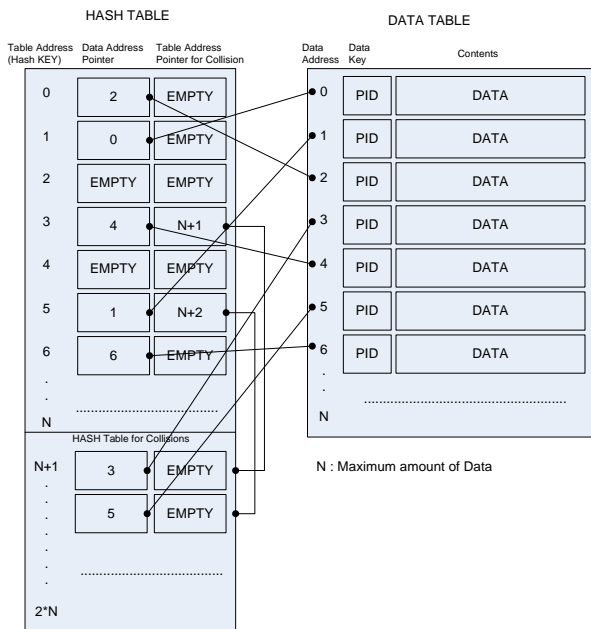


Figure 2. The structure of Hash table and data table

After constructing the tables at a system initialization time, the search algorithm, which receives a PID as a parameter and returns a data address, will work as follows:

- 1) Convert the PID into Hash Key within a proper range (0~N) by using the Hash function.
- 2) Read Data Address Pointer from Table Address (Hash Key) of the Hash Table.
- 3) Compare the PID in Data Address Pointer with the PID to be searched.
- 4) Return the Data Address pointer and Finish the search if the PID is identical.
- 5) Return the failure of the result and Finish the search if the PID is empty.

- 6) Read 'Table Address Pointer for Collision' for new Hash Key and Repeat 3) if the PID is different.

2.3 Experiment and its Result

The Hash function and search algorithm are performed in the HP Visualization Workstation B2000. Input keys are temporarily set up from PID-00001 to PID-16363 and the size of the Hash table N is 16362. The algorithm of Hash function is as follows:

```
for (i=0;*string;string++) i=131*i+*string;
return (i% HASH_TABLE_SIZE);
```

The search algorithm is performed with priority 31 of the POSIX scheduling policy which is the highest priority in the Operating System (HP-UX) [2]. As a result, the mean of the searching time is 1.088 μ s/PID. For comparisons, another experiment set up by the B-Tree search algorithm is conducted. The mean of the searching time in the B-Tree is 5.633 μ s/PID. The Hash algorithm shows a 5 times better performance than the B-Tree algorithm.

3. Conclusion

The IPS has a responsibility for processing real-time bulk data. The data searching time occupies the biggest part of the processing time in the IPS. It is required to find an efficient method to search a data in a real-time. For this, the Hash algorithm method is chosen because the search algorithm is constant. So, it is appropriate to the feature of the real-time monitoring information. The algorithm can search about 920,000 data per second. It satisfies the IPS requirements of searching about 10,000 data per 10 milliseconds.

For a future work, we need to research how to optimize the performance and how to resolve the problem of a synchronization in a multi-processing system. This will be conducted in the process of a integration for other applications and systems.

REFERENCES

[1] Tobin.J. Lehman, A Study of Index Structures for Main Memory Database Management System., Computer Sciences Department University of Wisconsin Madison

[2] IEEE Std 1003.1™, Information technology – Portable Operating System Interface(POSIX®) Part2: System Interface, 2003.8