Sensors & Transducers, Vol. 178, Issue 9, September 2014, pp. 292-297

Sensors & Transducers

http://www.sensorsportal.com

Advanced Red-Black Algorithm for Real-Time Data Addressing of Control System

An Sicheng, Wu Kehe, Zhou Huan, Li Yi
Control and Computer Engineer School
North China Electrical Power University, 102206, China
Tel.: 13811568275
E-mail: ansicheng@163.com

Received: 26 May 2014 /Accepted: 29 August 2014 /Published: 30 September 2014

Abstract: The development of scale of control system lead to the increasing amount of the control unit for a single main control station. The organizational structure of hierarchical model, to some extend, relieves the efficiency problem raised by the updating and searching operation of huge amount data point in macroscopic view. But in microscopic level, once there are considerable data points as children of a same point horizontally, the soaring of the access time appears inevitably as a result of the polling mode of traditional method with the widespread use in present control system. A RBA algorithm is proposed by this article to organize and search horizontal data point based on the Advanced Red-Black data structures. It could largely reduce the time consumption of searching and updating operation taking place frequently. Experiments and engineering applications prove that RBA algorithm improve the efficiency of addressing of horizontally data point searching. Further more, the problem of synchronization cycle caused by latency of addressing time could also be overcome. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Real-time data, Addressing, Search, Red-black tree, Control system.

1. Introduction

With the development of computer, communication and RTU&PLC technology, the scale of modern control system become increasing larger. Huge amount of control unit connected to a control system is a significant characteristic of development tendency. In such case, the load ratio of a single primary station sends quickly. The functions of primary station are about two aspects. On the one hand, it receives various of real-time data from all of the large numbers of RTUs and PLCs, and update the corresponding data point in the real-time database, and on the other hand, it provides services such as searching, addressing, accessing of data for the advanced intelligent application supported by real-time monitoring data. Therefore, higher demands such as high throughput, rapid response and low latency of searching are proposed to the real-time database on the primary station.

To meet these demands and adapt to the real device distribution, the real-time database on the primary station usually adopt hierarchical structure to describe and organize real-time data. All data points are stored and divided into levels as a tree [1]. By such organization modality, the chaotic and massive data points realize structurization clearly which improve the efficiency of updating and searching operations.

However, in the practical process of organization and application of system, such an issue that there are considerable amount of data points exists as children of one specific data point often take place [2-4]. For instance, a SCADA control system of power system may have a large transformer substation or a transmission line with very long distance, and thus several thousands even tens of thousands transducers or RTUs may in a some level. Actually, the benefits of hierarchical structure become negligible. For the fact that traditional real-time database just polls the data points in same level to search or update data, a single operation of searching or updating may have lots of polling and judging instruction of processor in extreme cases [5]. However, such operation is the most basic operation of the control system, and would be expected to complete in miner than 20 ms. So, such many instructions in one operation almost could not be accepted in practical. Furthermore, if the time consumption of the operation exceeds the period of a data cycle, the timeliness disruption problem will arise [6, 7].

To deal with this circumstance, we propose a design of advanced red-black tree data structure and RBA algorithm which form lots of horizontal data points to a nearly balance red-black searching tree without interrupting the original hierarchical organization structure. Meanwhile, a sortable nonoverlapping interval represented in 2-turple is formulated as the searching key of red-black tree data point according to the actual situation. With this way, the times of comparing and judging in searching or update operation could be controllable to a predictable max addressing times to ensure the total addressing time. By experiment and engineering application, it is be proved that RBA algorithm can limited the addressing time into the range with constant upper limitation [8-12].

2. RBA Algorithm

2.1. Applicable Conditions Analysis of RBA

Based on the BST (Binary searching tree) data structure, red-black tree can limit the relationship of total nodes number and max depth of the BST by dyeing every node with red or black color. Thus the BST could be change into approximately balancing tree. And according to the feathers of practical control system, the single value key is improved to a sortable nonoverlapping interval represented in 2-turple [13].

As described above, the reason that RBA is applicable to the circumstance that there are many horizontal data points under one parent data point are as follow: 1. red-black tree algorithm can lead the response time of searching and updating operation to such a low level, and almost stable despite the increment of the number of data point in the real-time database system. 2. It is a fact that the construction and destruction process of red-black tree is a little complex and consume more time. But the actual situation of most control system is more searching and updates operation and less or little inserting and deleting operation [14-16]. Thus its disadvantage is just avoided and benefits can fully play an essential role.

Based on the consideration above, the condition that applies the RBA algorithm to the circumstance that massive horizontal data points in control system is ripe, and the superiority of RBA algorithm can be fully took.

2.2. Basic Description of RBA Algorithm

The feather of BST is that every node in the tree including root has two child nodes of right and left, and the key of every node in the left subtree of the node is miner than the key itself, while the key of every node in the right subtree is bigger. Through this rules it is easily realized to searching node [17, 18]. And there is an autologous deficiency in BST that the selected root not changing after it constructs lead to possible severe unbalance in the tree and may, to some extent, lose the performance benefits of BST.

RBA algorithm can overcome the internal disadvantage of BST by dyeing the every node of tree with red and black color and establish a series of rules on the construction process of red-black tree. Thus the red-black tree adhering to these rules formulate an almost balanced tree and the internal fault of BST is conquered.

The constraint rule of red-black tree can be summarized into 5 points as follow:

- Every node in the tree is either black or red.
- The root node is black.
- All the leaf node(empty node) is black
- If a node is red, its two child node must be both black.
- Every path form a specific node to its every descendant node has the same numbers of black nodes.

Within the five constraints of red-black tree, the position and color may be need to change while inserting and updating tree to maintain the property of red-black tree. And by doing such adjusting, the tree in this mechanism can be revealing an almost balancing status.

The position adjusting includes two forms of left-rotation and right-rotation. The detailed steps of left-rotation are shown in graph 1. Left-rotation on X is to rotate left over the connection line of X and Y as axis, and Y becomes the root of the new subtree. After that, X becomes the left child of Y, and the left child of Y becomes the right child of X. Right rotation is inversed. After rotation, the relationship of size of the left and right child of every node is immutable.

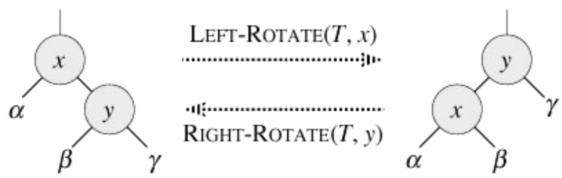

Fig. 1. Left and right rotation process of red-black tree.

By the adjust of position and color, red-black tree can maintain the 5 rules after inserting and deleting node to and from it.

While inserting a node into red-black tree, making the new node red color, continuously iterate from the root to the leaf to find an appropriate position according to the key value of the new node. Then putting the node to the position may lead to break some rules. Thus adjust of position and color is necessary.

The process of adjust is as follow:

Firstly, we assume that the parent node of the new node z is left child of its grandparent c, and its parent c is red (if its parent is black, it is not necessary to adjust). There are three different circumstance need to be dealt with:

I. The uncle node (the brother of its parent) of z is red. In this case, we dye the parent node and uncle node of z with black, and dye the grandparent of z with red.

Analysis: for the red color of the parent and uncle node of z, its grandparent node must be black. Dyeding its parent and uncle node with black solve the problem that z and its parent are both red. And dyeing its grandparent with red solve the interruption of rule 5.

II. The uncle node y of z is black and z is the right child of its parent. In this case, we do the left-rotation on the parent z, and set the left child of z after left rotation to z. Then it changes to the circumstance III.

III. The uncle node y of z is black and z is the left child of its parent. In this case, we dye its parent with black and its grandparent with red, and then do right rotation on its grandparent.

Analysis: circumstance II interrupt rule 4, and even interrupt rule 4 after changing to circumstance III. Dyeing it's parent with black and its grandparent with red solve the interruption of rule 4, but also interrupt rule 5. The following right rotation solve rule 5.

For the condition that the three circumstance is under the hypothesis the parent node of z is the left child of its grandparent, it just need changing the left and right if the parent node of z is the right child of its grandparent.

The whole process of inserting is shown as Fig. 2:

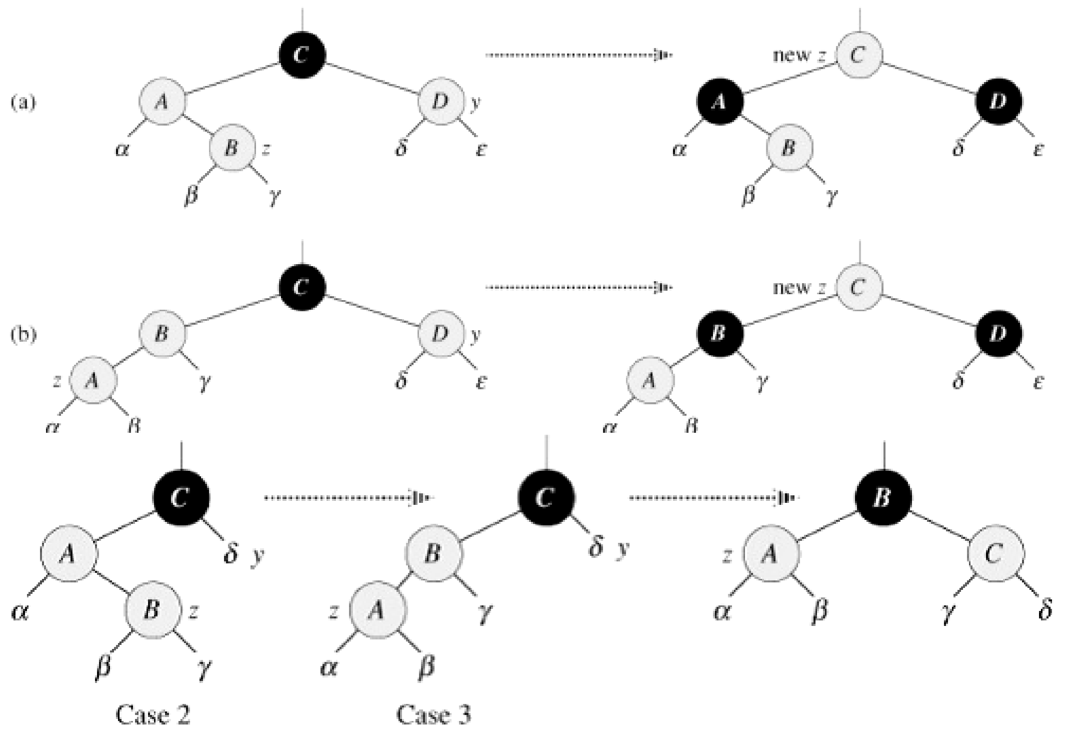

Fig. 2. The process of adjust after inserting red-black tree node.

While deleting nodes of red-black tree, for the limitation that only the leaf node could be delete, it need to something more to delete non-leaf node. Thus we can set the directly descendant node y of the non-leaf node z to z, and then delete y. And there is also a more complex adjust process for this operation than inserting node. The process is described as follow:

While the node x to be deleted is black and is left child of its parent, the different circumstances to adjust can be concluded as the following 4 type:

I. The brother node w of x is red. In this case, we dye the brother node of x with black and dye its parent node with red, and then do left rotation on its parent. Thus the new brother of x is the child of w before rotation, and it changes to the circumstance II, III, IV.

II. The brother node w of x is black and the two children nodes of w are all black. In this case, we dye w with red and treat the parent node of x as x.

Analysis: Because the two children of w is all black and w itself is also black, it is needed to remove a black node between x and w. So we dye w with red color and treat the parent node of x as x to solve the issue.

III. The brother node w of x is black and the left child of w is red and the right is black. In this case, we dye the left child of w with black and dye w with red, then do right rotation on w. After that, we treat the new left child node of w as w. Thus the circumstance changes to circumstance IV.

Analysis: By exchanging the color of w and its left child, the rules 4 is remained. At this time, the new brother w of x is a black node having red right child node.

IV. The brother node w of x is black and the right child of w is red or the two child of w is all red. In this case, we dye w with the color of the parent of x and dye the parent of x and the right child of w with black, and then do left rotation on the parent node of x and treat x as root.

Analysis: Color adjusts and left rotation result in the fact that the extra black of x is shared by its parent after rotation and all rules are not interrupted.

Such circumstances assume the condition that x is the left child of its parent. If x is the right child of its parent, we just exchange right and left.

The whole process of deleting is shown as Fig. 3:

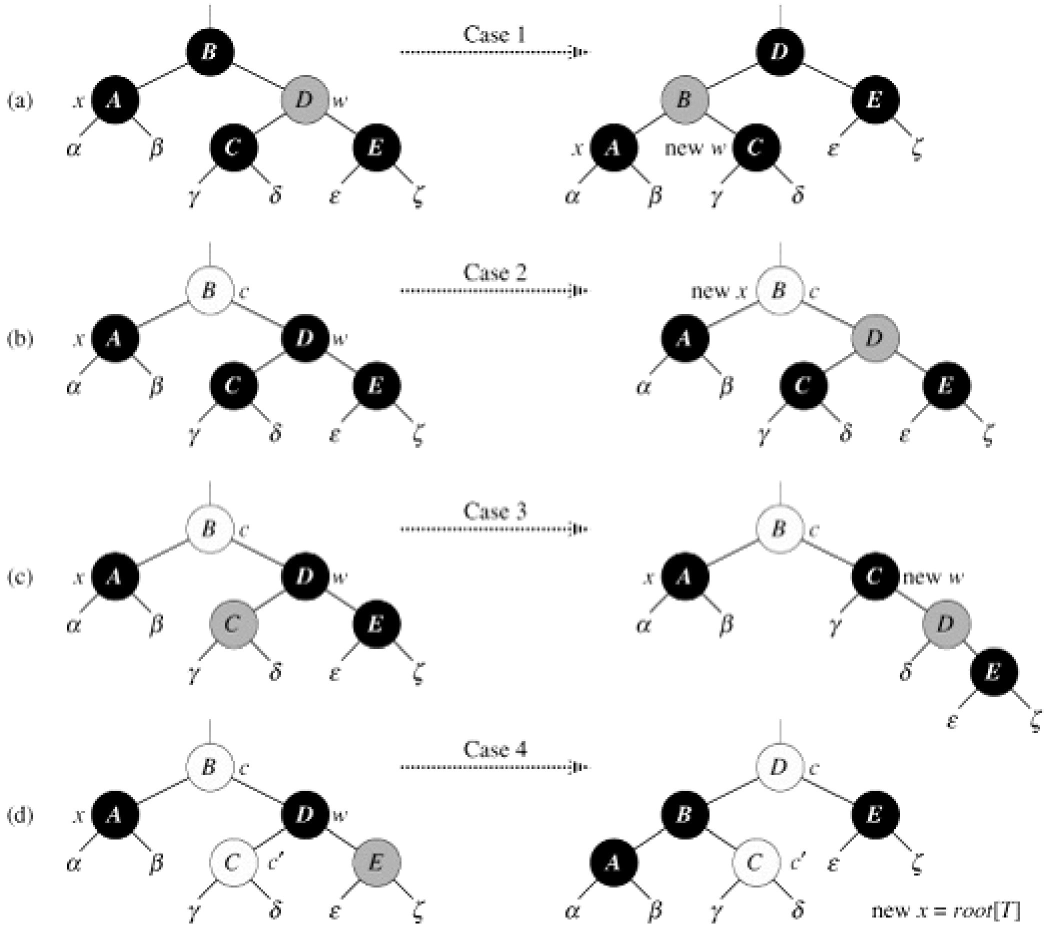

Fig. 3. Process of adjustment of RB tree after deleting.

2.3. Analysis and Proof of Complexity of RBA

Red-Black tree data structure is based on the BST data structure and restricts it by 5 rules. In such constraints, the height of the tree is limited in a constant range. The relationship between the height of the tree h and the number of nodes n is analyzed as follow:

Set the black height of x is bh(x), then the internal node of the tree that root is x is at least $2^{bh(x)}-1$.

I. If the black height of x is zero, then the number of internal nodes is 0(x is leaf).

II. If the child of x is red, its black height is equal to x. If it is black, its black height is bh(x) – 1. So the black height of the child of x whatever its color is at least bh(x)-1. According to inductive assumption, the number of nodes of two subtree of x is at least $(2^{bh(x)-1}-1)\times 2=2^{bh(x)}-2$. And accounting for x itself, the number of nodes of the tree that root is x is $2^{bh(x)}-1$.

According to rule 4, the number of black nodes is always not miner than red in any path of the tree from any node. Then the number of black nodes in the path from root is at least h/2. The follow condition can be summarized:

$$n \geq 2^{bh(x)} - 1 \geq 2^{h/2} - 1$$

So $h \leq 2 \times \lg(n+1)$, and the height of red-black tree is $\circ(\lg n)$.

The time complexity of searching is $\circ(\lg n)$. Because inserting operation is also one iteration form root to leaf, its time complexity is also $\circ(h) = \circ(\lg n)$.

It is can be concluded that By restricting 5 rules on BST structure, the height of BST can be controlled in the logarithm level of nodes number n. Furthermore, the range of height of red-black tree is clear and the tree structure is almost balanced. Thus the inserting, deleting and searching time complexity decreased form $\circ(n)$ to $\circ(\lg n)$, improving efficiency.

3. Analysis of Experiments of RBA

3.1. Analysis of Characteristics of Addressing Request in Practical real-time Database

In the practical process of running real-time database, the addressing request sequence reveals some regularity characteristics. Especially, the addressing request in horizontal nodes of one specific level influences the extent of improvement of RBA. So we research and analyze these requests at first [19, 20].

Firstly, we do statics of the request sequence of addressing. The case we selected is a control with medium scale which has totally 5053 data points in a specific level to be monitored. By monitoring for 15 days in 24-hour mode, the data set of the status of points in the control system has been collected. There are about 40 percent data in the data set is addressing request which the request key r is in the rang (0, 320100).

Fig. 4 presents the probability density of r within 1328603 samples.

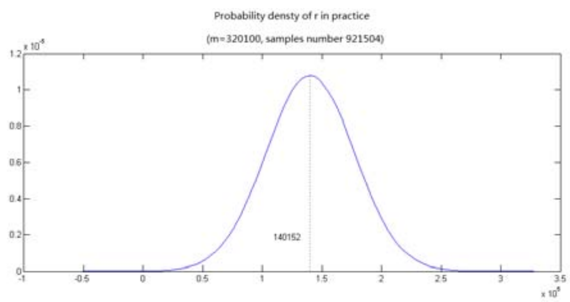

Fig. 4. Probability density distribution of r by sampling observation.

According to the sample, it is can be conclude that r fits normal distribution. And the mathematical expression is E(r)≈1.40152×105, and MSE is D(r)≈0.41815×105. So it is considered that:

r~N(1.40152×105 , 0.41815×1010)

Based on the conclusion above, we generate a testing data sequence that fit the regulation.

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, IFSA, ac, dc, ms, etc. do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

3.2. Testing of Efficiency of addressing of RBA

For the fact that the addressing request is independent, the time distribution can be neglected. And we use the same test sequence on traditional and RBA algorithm. The point number of test sequence is 10, 20, 30, 50, 100, 150, 200 thousand times.

The result is shown in Table 1:

Table 1. Efficiency comparison between traditional and RBA algorithm.

	1	2	3	5	10	15	20
Traditional	83	158	234	395	771	1164	1551
RBA	1	2	3	5	11	16	21

It is can be seen that the request time and addressing efficiency is largely improved in the same sequence of testing. Furthermore, the efficiency does not fluctuate with the increasing of times of addressing requests and is almost linear. For the requests times scale which is about 100 thousands a day in current scale of control system, the updating and searching time is about controlled to 20 ms with RBA algorithm. It is can be seen that the efficiency of addressing operation is largely improved with RBA algorithm.

4. Conclusion and Further Work

For the problem that efficiency of addressing operation declining arised by traditional process method which poll the data points in same level and the fact that the number of addressing request in the level is frequent, we propose a method that construct red-black tree data structure on the data point in same level to improve the efficiency but not interrupt the original data structures.

The main feather of the method can be summarized as 4 points:

I. Constructing red-black tree structure on origin data structure lead to efficient and simple searching operation.

II. By restricting the 5 rules on BST, the structure of BST is more optimizing. Thus the structure of the tree is nearly balancing, and the addressing time of single searching and updating operation is shortened.

III. Construct the method of inserting and deleting of red-black tree to ensure the 5 rules.

IV. By the simple and direct iteration of searching and updating operation, the time consumption is controlled to a constant.

The series of work in this article prove that RBA algorithm can tremendously improve the efficiency of the running control system. And the further work to complete in future is as follow 2 points:

I. If the key of the addressing operation is multiple, how to design the correct red-black data structure to ensure high efficiency addressing operation simultaneously.

II. How to improve the stored red-black tree related data structure to minimize the memory space used to store it.

References

[1]. Woochul Kang, S. H. Song, J. A. Stankovic, Evaluation of a QoS-aware real-time embedded database, *IEEE Transactions on Computers*, Vol. 61, Issue 1, 2012, pp. 45-59.

[2]. S. Chakrabarti, E. Kyriakides, T. Bi, et al, Measurements get together, *IEEE Power and Energy Magazine*, Vol. 7, Issue 1, 2009, pp. 41-49.

[3]. Zhang Zhizhe, Li Xingyuan, Cheng Shijie, Structures, functions and implementation of united information system for smart grid, *Proceedings of the CSEE*, Vol. 30-34, 2010, pp. 1-7 (in Chinese).

[4]. M. Amirijoo, Specification and management of QoS in real-time databases supporting imprecise computations, *IEEE Transactions on Computers,* Vol. 55, Issue 3, 2006, pp. 306-319.

[5]. Ming Xiong, Biyu Liang, Quality of service guarantee for temporal consistency of real-time transactions, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, Issue 8, 2006, pp. 1097-1100.

[6]. H. Garcia-Molina, Main memory database systems: an overview, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, Issue 6, 1992, pp. 509-516.

[7]. N. Soparkar, Ann Arbor, H. F. Korth, Abraham Silberschatz, Databases with deadline and contingency constraints, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, Issue 4, 1993, pp. 552-565.

[8]. A. Meixnei, D. J. Sorin, Dynamic verification of memory consistency in cache-coherent multithreaded computer architectures, *IEEE Dependable and Secure Computing*, Vol. 6, Issue 1, 2009, pp. 18-31.

[9]. Keqin Li, Fast and scalable parallel matrix computations on distributed memory systems, in *Proceedings of 19th IEEE International Conference on Parallel and Distributed Processing*, 2005, 8b.

[10]. Inseon Lee, Yeom Hy, A single phase distributed commit protocol for main memory database systems, in *Proceedings of International Conference on Parallel and Distributed Processing Symposium*, 2002, pp. 14-21.

[11]. Zhang Haiyang, Liang Qiaoyu, Red-black tree used for arranging virtual memory area of Linux, in *Proceedings of the International Conference on Management and Service Science (MASS)*, 2010, pp. 1-3.

[12]. Jianwei Li, Yubin Xu, Hong Guo, Memory organization in a real-time database based on red-black tree structure, in *Proceedings of the Fifth World Congress on Intelligent Control and Automation*, 2004, pp. 3971-3974.

[13]. H. Blaar, T. Lange, R. Winter, M. Karnstedt, Possibilities to solve the clique problem by thread parallelism using task pools, in *Proceedings of 19th IEEE International Conference on Parallel and Distributed Processing*, 2005, pp. 345-351.

[14]. Chang-Tien Lu, Jing Dai, Ying Jin, J. Mathuria, GLIP: A concurrency control protocol for clipping indexing, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, Issue 5, 2009, pp. 714-728.

[15]. A. A. Aravind, Prince George, Yet another simple solution for the concurrent programming control problem, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, Issue 6, 2011, pp. 1056-1063.

[16]. L. Tomas, C. Carrion, B. Caminero, A. Caminero, Meta-scheduling in advance using red-black trees in heterogeneous Grids, in *Proceedings of the IEEE International Symposium on Parallel & Distributed Processing, Workshops and PhD Forum (IPDPSW)*, 2010, pp. 1-8.

[17]. Li Wei, Sheng Deren, Chen Jianhong, et al, Selecting the scheme of real-time database for sis in a power plant, *Proceedings of the CSEE*, Vol. 23, Issue 12, 2003, pp. 221-224 (in Chinese).

[18]. Liu Qing, Huang Chao, Tan Zhenyu, et al, Design and development of a deductive database for power system, *Proceedings of the CSEE,* Vol. 18, Issue 6, 1998, pp. 30-32 (in Chinese).

[19]. Wu Wenchuan, Zhang Boming, Xu Chunhui, Study and implementation of real time database management system (rtdbms) model applied in ems, *Power System Technology*, Vol. 25, Issue 9, 2001, pp. 28-32 (in Chinese).

[20]. A. G. Phadke, H. Volskis, R. M. de Moraes, et al, The wide world of wide area measurement, *IEEE Power and Energy Magazine*, Vol. 6, Issue 5, 2008, pp. 52-65.