Research on the Voting Algorithm and its Application in Intrusion Tolerant System

Zhao Yuehua
School of Computer Science and Communication Engineering, Jiangsu University
Zhenjiang, Jiangsu
e-mail: zhaoyh@ujs.edu.cn

Fan Lijuan
School of Computer Science and Communication Engineering, Jiangsu University
Zhenjiang, Jiangsu
e-mail: flj0304@126.com

Abstract—**To solve the problems of Voting algorithm's no output and no high accuracy, a voting algorithm based on testing information and historical record values has been designed. The algorithm can immediately extract services run-time error message from the memory and arithmetic logic unit and improve voting accuracy combined with the copy-computers's recently history record values. The experiments prove that the improved voting algorithm has more accurate output than the effective weight voting which is based on a copy. It can provide continuous service, improve the integrity of the system and is more suitable for intrusion-tolerant system.**

Keywords- *testing information; historical record values;voting; intrusion-tolerant system*

I. INTRODUCTION

The study of intrusion-tolerant[1] has been paid more attention in recenty years. Differently from the intrusion de-tection technologies, the intrusion-tolerant system can still run correctly and reliably even the invasion or fault exists, and it can continue to provide effective services. The system mainly considers the survivability when it is invaded successfully. The main technology of providing the intrusion-tolerant capabilities for the system is the object replication (redundancy) and voting[2]. As a technology of replication control method and shielding defective replications (it means the replications of malicious, arbitrary or Byzantine behavior), voting has played a key role in intrusion-tolerant.

Now the most commonly used voting strategy is Large Number Voting[3]. However, it has shortcomings such as: the rate of failure is high, the voting parameters of the algorithm is not flexible enough and the quantitative assessment of the replications is lack. In order to solve the shrotcomings of the Large Number Voting algorithm, Yu Yan-ping [4] and others has proposed an adaptive Large Number Voting mechanism which uses the Majority Voting algorithm based on historical information. Li Jian[5] and others has proposed a voting mechanism based on the reliability weight of the copy. For each round of voting, the mechanism doesn't need all copies to paiticipate, and the credibility is measured through comparing the response result for the customer's service request of each copy and the final result of the voting algorithm. It's more likely that the high-credibility of the copy can be the final output. Although the above two kinds of algorithms can solve the shortcomings of the Large Number Voting algorithm and reduce the failure rate, the shortcomings still exist: One, when the service copies can not reach consensus or the threshold is inappropriate, the result can not be output with an exception. This is not allowed in some occasions which is the performance of voting failure. The other, the above algorithms don't consider the causes of failures, and the real-time faults are generated more than permanent faults in the system. The above algorithms determine the voting result of the new round according to the voting contribution made by the prior service copies. However, if the service copy make an error which makes the final voting output in the new round of voting, the correct rate will be reduced, and the real-time characteristics of the service process can not be reflected correctly.

To address the above problems, this paper presents a voting algorithm based on detection information and history records and applies it to the intrusion-tolerant systems. It can solve the output problems in the inconsistency situation and supply the correct response for users..

II. A VOTING ALGORITHM BASED ON DETECTION INFORMATION AND HISTORY RECORDS

A. *A voting algorithm based on effective weight*

The thought of the algorithm is to calculate the voting contribution value for each copy. In the new round of voting, the voting machine firstly divide the voting results, then select the set with the largest collection as the voting set. According to the previous contribution value of the copy, select the response result of the copy with the biggest contribution value as the voting output.

B. *The though of the improved algorithm*

As the transient errors are prone to appear in the system which can easily make the copy to get wrong response result, the vote machine will get the wrong result. Therefore, add some detection code to extract the service running situation, and the detection result can be a basis for the response result of the copy. In addition, the recently voting contribution value of the copy can be provided for the voting machine. However, even if the previous response is correct, it doesn't mean the current response is correct, so it doesn't need to take all voting contribution values into account. Thus the voting machine in this paper is based on the recently history records and detection information of this current round.

The threshold of the voting machine is the basis to judge whether the input values of any two copies are consistent in the voting process. For any two copies i and j, set the input values x_i and x_j and the threshold ε. If $|x_i - x_j| \leq \varepsilon$, it means the two copies i and j can meet the consistency, otherwise satisfied. The threshold can be determined before the vote, but it can be changed though the re-allocation strategy of the system.

The actual operation of the voting machine is as follows:

Accumulate the history record for each round of voting (the specific technology and implementation can be found in Literature 7), the voting machine divide the voting results of the current round according to the history records of previous rounds ofand detection information of this current round as well as the set dividing rules of the Large Number Voting algorithm. Select select the set with the largest collection as the voting set. If the collection of the set is greater than（N+1）/2, then select the service response with the biggest history record and the smallest detection information as the output value. If the collection of the set is less than（N+1）/2, then select the service response with the smallest detection information as the output value and output an exception message. Through analyzing the state information and history records of the voting machine, it can diagose the fault copies.

C. The description of the devoting algorithm

For convenient presentation, we call the requested service program as the service code. The typical process of the service code is to read sensor information, calculate according to the control rate and output the control information. The process is called as a service cycle. The copy calculate according to the input and output the service result and the detection result as the input of the voting machine, and the voting machine get the voting output according to the voting algorithm. The process is called a voting cycle. The formal description of the voting algorithm is given as follows:

- Set $A = \{a_1, a_2, \cdots, a_n\}$ as the output of N copies about the voting cycle q, where $a_i = < r_i, s_i >$ and r_i is the output of the service result, s_i is the output of the detection information, $H = \{h_1, h_2, h_3 \cdots h_n\}$ is the history records of the copies till the voting cycle q. Set $V_1, V_2, \cdots, V_p$ as a division of A which makes each V_i can contain as much elements as possible, where for $V_1 \cup V_2 \cup \cdots \cup V_P = A$, $V_i \cap V_j = \varnothing$, i≠j, $\forall a_i, a_j \in V_k$, there is $C(r_i, r_j) \leq \varepsilon$ .C is the voting operations of the space A. If the inequality is correct, it indicates that the copies i and j can meet the consistency, otherwise satisfied. The threshold ε can be determined before the vote, but it can be changed though the re-allocation strategy of the system.
- Set $|V_k|$ as the numbers of V_k, and $V_m : |V_m| \geq |V_k|$, 1≤k≤p. While V_m is not single, catch the smallest value of $\sum_{a_i \in V_m} d(s_i - s)$, where $d(s_i - s)$ means the distance of s_i and s, and s is the standard value of detection information with the usual value 0.
- If $|V_m| \geq (N+1)/2$, select the highest history record value a_i which meets the conditions where $a_i \in V_m$ and $d(s_i - s) \leq d(s_j - s)$, and r_i in $a_j \in V_m$ is as the output of the voting machine. If r_i is not single, select the value with the smallest subscript. Then calculate the voting result of the new round.
- If $|V_m| < (N+1)/2$, select the value a_i which meets the conditions where $a_i \in A$, and $d(s_i - s) \leq d(s_j - s)$, and r_i in $a_j \in A$ is as the output of the voting machine. If r_i is not single, select the value with the smallest subscript and output an exception warning message. Then calculate the voting result of the new round.

D. The acquisition of detection information

In order to capture the error event which occurred in the running process, the appropriate detection codes are inserted into the original process codes to extract the environmental information during the running process. Set the sequence of the original process codes as $\{t_1, t_2, \cdots, t_n\}$ and the sequence of the detection codes as $\{d_1, d_2, \cdots, d_m\}$. Insert $\{d_1, d_2, \cdots, d_m\}$ into $\{t_1, t_2, \cdots, t_n\}$ which can equal to $\{t_1, d_1, t_2, t_3, d_2, \cdots, d_m, t_n\}$. With different insertion methods, the sequences are captured differently. In the system, as the transient errors mainly occurre in the memory and processor[8], the detection sequence is designed to detect the transient errors of RAM and the arithmetic logical unit. The detection sequence is defined as follows:

```
<TestCode>: <Init>; <TestPair>; <Finish>;
  <Init> : TestI=0; TestJ=0;
 <TestPair>: TestI=TestI+1.0;
             TestJ=TestJ+1.0;
 <Finish>:  TestResult=TestI-TestJ.
```

TestPair can be repeated an arbitrary number of times. The typical cycle service can be described as follows after the detection codes are inserted:

```
//the global variables used in the detection codes outside the
function
 while(1)  //cycle service, runs constantly and repeatedly
{
 Init     //detect initialization, detect variables
 t1;  //the beginning of a cycle, such as reading sensor information
          //insert into a detection pair every certain amount of codes
TestPair ;
   t2;
   t3;
TestPair ;
   ⋮
TestPair ;
 tn;           //operation ended, prepare to vote
 Finish             //make the detection results standard
r=vote(r,DetectResult);  //voting with the self-detection
 output(r);         //output the control information
}
```

If no error occurs during the execution, then the final TestResult should be 0; If there is RAM error or ALU error in some service cycle q, then TestResult may not be 0 which indicates that an error occurs in this cycle.

III. THE EXPERIMENTAL RESULTS

In order to verify the performance comparison when there exists the invasion and error about the improved voting algorithm and the Majority Voting algorithm based on copy reliable weight, the intrusion-tolerant system which is bulit in LAN-based client/server (C/S)is used as the experimental test environment. Here the client is a broad one which has the functions such as: send service requests, receive the redundancy response result of the service copy and make voting treatment, that is, the voting algorithm based on self-diagnosis and the Majority Voting algorithm based on history records are implemented in client. In this experiment, the client is PC with PentiumIV2.8GHz, and the operation system is Windows XP.

The server consists of four sets of Web server with different types and platforms. The specific configuration is shown in Table 1 as follows:

TABLE I. THE PLATFORM OF SERVER

	operation system	processor	memory
server 1	Linux 7.0	Athlon2.31 GHz	1G
server 2	Windows xp	XEON2.8 GHz	1G
server 3	Red hat 9.0	XEON2.8 GHz	1G
server 4	Windows 2007	Athlon2.31 GHz	1G
server 5	Ubuntu 9.0	Athlon2.31 GHz	1G

The comparison between the improved voting algorithm and the Majority Voting algorithm based on copy reliable weight is mainly got through testing the voting capability of algorithms, that is the capability of outputing the correct result. In this experiment, the system runs Round=1000 times under the setted error numbers (because there are 5 service copies, while the assumption error is only one per server) and the threshold is 0.5. Record the times with the correct output value, that is the successful voting times. The record of failure times means that the times with the failure output value plus the times which can not meet the consistency. The final statistical result is shown in Table 2:

TABLE II. THE DEVOTING CONDITIONS

Number of errors	Voting algorithms	Successful numbers	Failure numbers
0	the voting algorithm based on copy reliable weight	1000	0
	the voting algorithm in this paper	1000	0
1	the voting algorithm based on copy reliable weight	927	73
	the voting algorithm in this paper	965	35
2	the voting algorithm based on copy reliable weight	853	147
	the voting algorithm in this paper	914	86
3	the voting algorithm based on copy reliable weight	791	209
	the voting algorithm in this paper	889	111

In order to measure relationship between the voting performance and the system error conditions, make definations as follows:

Defination 1 The successful rate S: In multiple voting results, the percentage about the correct output voting results in the total number of output voting results, that is, S=Correct/Round.

Defination 2 The failure rate F: In the system, the percentage about the number of the failure response input value in the total number of the response input values. If there are n service requests about the server copies in the system, there are m copies with errors, then F=m/n.

The relationship between the successful rate S and the failure rate F is shown in figure 1:

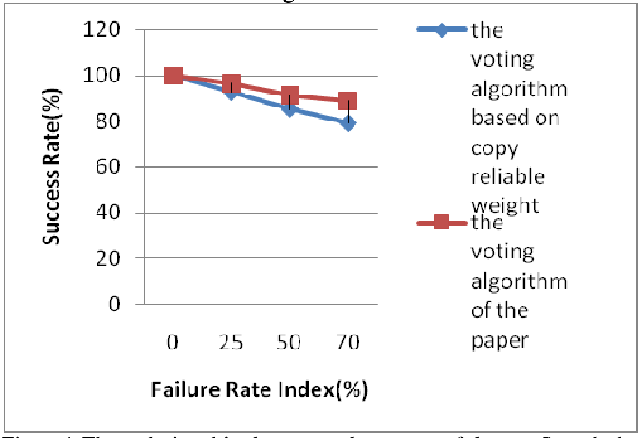

Figure1:The relationship between the successful rate S and the failure rate F

From figure1, we can see that, in the same failure rate F, the voting algorithm in this paper has higher successful rate S than the voting algorithm based on copy reliable weight.

With the failure rate F increases, the successful rate S gradually decreases, but the voting algorithm based on copy reliable weight decreases faster than the the voting algorithm in this paper. At the same time, the voting algorithm in this paper ensures the successful rate at a high level.

IV. CONCLUSION

In this paper, a voting algorithm on detection information is designed based on the voting algorithm on copy reliable weight. This algorithm extracts detection information when the server is running and sets it as one of the basis for the voting machine, so it can solve the output problems when the service copies are not consistency or the threshold is not appropriate. Compared to the voting algorithm on copy reliable weight, the voting algorithm in this paper can output the more accurate results and solve the output problems in intrusion-tolerant system. At the same time, it can effectively identify and delete the failure copy to increase the correctness.

The experimental results show that, compared to the voting algorithm on copy reliable weight, the voting algorithm in this paper can output the more accurate results and solve the output problems in intrusion-tolerant system. At the same time, it can effectively identify and delete the failure copy. Through reconfiguration, the system can get better availability, therefore it is more suitable for the intrusion-tolerant system.

As the selection of the threshold is different in the voting machine, the data in Table 1 may be different. So the further research will be made in the future work in order to make the voting algorithm better which can be better applied to the intrusion-tolerant system.

REFERENCES

[1] Reynolds J, James Just, Ed Lawson, t al. The design and implementation of an intrusion tolerant system[C].Proc. of Int'l Conference on Dependable Systems and Networks, Washington D.C., 2002. 258-290.

[2] Wang F, Uppalli R, Killian C. Analysis of techniques for building intrusion tolerant serversystems[C]. Proc. of Military Communications Conference, Washington DC, 2003. 729-734.

[3] Bass J M, Latif-Shabgahi G, Bennett S. Experimental comparison of voting algorithms in cases of disagreement [A]. Proceedings of the 23rd EUROMICRO Conference on New Frontiers of Information Technology [C]. Washington: IEEE Computer Society, 1997. 516 - 523.

[4] Yu Yanping,Guo Boyuan,Ma Jianfeng. Large numbers on the voting mechanism based on adaptive intrusion-tolerant model [J]. Systems Engineering and Electronic Technology, 2005, 27(6):1098-1101

[5] Li Jian, Service-based intrusion-tolerant replication and voting technology research[D]. Harbin Engineering University Master's degree thesis,2009.

[6] Yang Mengfei. The basic problem of space and Countermeasures of Fault-tolerant computer[J]. Aerospace Control,1994,2: 25 39.

[7] Latif-shabgahig,Bennett S. Adaptive majority voter: a novel voting algorithm for real-timefault-tolerant control systems[C]. //Proc of the 25th EUROMICRO Conference.Washington DC:IEEE Computer Society, 1999: 2113-2120.

[8] Gil D, Martinez R, Busquets J V. Fault injection into VHDL models: Experimental validation of a fault tolerantmicrocomputer system [A]. Proceedings of 3rd European Dependable Computing Conference [C]. Berlin Heidelberg:Spinger-Verlag, 1999. 191-208.