Journal of Computer Science (Special Issue): 104-105, 2005
ISSN 1549-3636

Fault-tolerant Distributed Systems with Diagnostics Algorithms

[1]Oleg Viktorov and [2]Afif Mghawish
[1]Department of Software Engineering, Al-Zaytoonah University, Jordan
[2] Department of Computer Science, Al-Zaytoonah University, Jordan

Abstract: To provide consistent actions in distributed systems with faulty nodes the Byzantine agreement protocol (algorithm) is widely used. In case of using message exchange scheme without authentication the Byzantine agreement algorithm leads to agreement if the number of nodes doesn't exceed 1/3 of the total number. The proposed algorithms based on diagnostics procedures are used to reach an agreement in distributed models with $2n+2$ nodes and fewer than k failed nodes. The hierarchical diagnostic procedures give the possibility to vary the complexity of hardware and software overhead according to required level of fault-tolerance.

Key words: Distributed systems, nodes, Byzantine agreement, protocol, fail-free system, multistage voting scheme, consensus actions, authentication, general, lieutenant, tester, traitor, malicious node behavior

INTRODUCTION

The Byzantine Generals problem involves reaching an agreement among n nodes of distributed system, some of which may be faulty. It can be stated as follows:

Given a set of n nodes which are sending messages to one another, to find an algorithm by which one of the nodes – general can transmit the message a to all other nodes such that:

a. If General is nonfaulty, then any nonfaulty nodes get the same message.
b. If nodes i and j are nonfaulty, then both get the same message.

Nonfaulty nodes are assumed to correctly follow their algorithm, but faulty nodes may do anything.

To provide consistent actions in distributed systems with faulty nodes the Byzantine agreement protocol (algorithm) is widely used. In case of using message exchange scheme without authentication the Byzantine agreement algorithm leads to agreement if the number of faulty nodes doesn't exceed 1/3 of the total number. Byzantine agreement is an efficient method, since the failure model considered in this problem is most general and if we can handle satisfactorily, we can be sure that most types of failures can be masked. It is interesting problem to be solved - to reach an agreement if the number of faulty nodes exceed one third.

DISTRIBUTED MODEL

Let us consider a distributed system of $2k + 2$ nodes, where there are not more than k faulty nodes. Let the node i be the **general**. Node i sends messages to each of other nodes (**lieutenants**). Then lieutenants exchange the messages received from general. After analysis performed by each lieutenant, the nodes which do not meet the loyalty condition (contradictory messages are sent to the various lieutenants) are defined as **traitors.** The rest of lieutenants are loyal.

DIAGNOSTICS ALGORITHMS

The diagnostics of the nodes performed by each loyal lieutenant leads to the only solution concerning to the traitors an agreement between general and loyal lieutenants can be achieved. The algorithm 1 should be performed by each node of the distributed system.

Algorithm 1

1. **General** sends the message a ($a \in \{1,0,\emptyset\}$) to all rest nodes.
2. Each of the n **lieutenants** sends the message received from the **general** to all other **lieutenants.** Each lieutenant p ($p = 1,2,..., n$) forms a vector V_p ($V_p = a_1, a_2, ..., a_n$) from the messages received.
3. Each of the n **lieutenants** p ($p = 1,2,..., n$) sends the vector V_p received from other **lieutenants.** Each lieutenant p ($p = 1,2,..., n$) forms a matrix M_p from vectors V_p $(p = 1,...,n)$ from messages received.
4. For each lieutenant p apply the majority function on each matrix column to get vector MAJ_p .
5. Mark the corresponding lieutenant T_p ($p = 1,..., n - 1$) with asterisks if $MAJ_p(r) \neq M_p(r,s)$
6. Set up the expression $cond\_sub\_M_p = \overset{m}{\underset{k=1}{\wedge}} (r_k \vee s_k)$
7. Form from $cond\_sub\_M_p$ a set of conjunctions sub_M_p, that contains not more than k nodes.

Corresponding Author: Oleg Viktorov, Department of Software Engineering, Al-Zaytoonah University

8. Define variable $alg1\_result_p$

a. $alg1\_result_p = 0$ if the set of conjunctions contains one element $|sub\_M_p| = 1$.

b. $alg1\_result_p = 1$ if $|sub\_M_p| > 1$

It can be shown easily that the application of algorithm 1 can results with a few solutions and there is a problem to choose the right solution. The following theorem has been proved:

Theorem 1: The right solution is always present among the solution given by $cond\_sub\_M_p$.

Corollary: If $cond\_sub\_M_p$ contains the only solution, it is correct.

If the class of faults is more extend, including malicious node behavior, then algorithm 2 can be applied. Algorithm 2 consists of several stages. Each stage includes several phases. We assume that there is the mechanism for assigning the node that provides the distributed system diagnostics. Such node is referred as **tester**. A different **tester** has been assigned on different stage.

Algorithm 2

1. The **tester** *i* sends the messages to other nodes:

a. "I've defined the traitor"

b. Perform the $test_i^r$ (r = 1,2, ... , f), where f is the number of phases) in the course of time $t = delay\_t_i^r(j)$ node *i* sends " send $test_i^{r+1}$" to the node *j*.

2. The **tester** *i* processes the received messages:

a. if in the course of time $t = delay\_T_i^r(j)+(n-1)*delay\_message(i,j)$ no one messages has been received from *j* then **tester** *i* considers the node *j* to be faulty.

b. if in the course of the time *t* a message $result_i^r(j)$ is obtained the following facts are checked:
   - the correctness of the test result;
   - the time of answer receiving.

   If one of these conditions is not met the node is considered to be faulty.

3. If the number of phases of the current stage having been completed, node *i* sends the message "*I've completed the stage*" or if node *i* comes to the unambiguous solution of the diagnostics it sends the message "*I've defined the traitors*".

RESULTS

As we have mention above in case of using message exchange scheme without authentication the Byzantine agreement algorithm leads to agreement if the number of faulty nodes doesn't exceed 1/3 of the total number. We are able to overcome this limitation if the diagnostics of faulty nodes is carried out and then their messages are excluded from consideration to achieve an agreement. The implementation of suggested algorithms leads us to an agreement in the distributed systems with $2k + 2$ nodes, where the number of faulty nodes does not exceeds k.

CONCLUSION

The multilevel diagnostics algorithms of distributed systems are suggested. The most attractive feature of the diagnostic is adaptive increasing of diagnostics procedures power that depends on the class of faults. The implementation of algorithms suggested in the paper lead us to an agreement in the distributed systems with $2k + 2$ nodes, where the number of faulty nodes does not exceeds k. At the beginning algorithm 1 is to be applied. It seems to be rather effective in the case of hardware faults. If the class of faults is more extend including malicious node behavior, the algorithm 2 is used.

REFERENCES

1. Pease, M., R. Shostak and L. Lamport, 1980. Reaching agreement in the presence faults. J. ACM, 27: 228-234.
2. Lamport, L., R. Shostak and M. Pease, 1982. The byzantine generals problem. ASM Trans. on Programming Languages and Systems, pp: 382-401.
3. Dolev, D., M. Fisher, R. Fowler, N. Linch and H. Strong, 1982. Efficient byzantine agreement without authentication. Information and Control, 52: 257-274.
4. Lamport, L., 1983. The weak general problem. J. ACM, 30: 668-676.
5 Turpin, R., 1984. Extending binary byzantine agreement to multivalued byzantine agreement. Information Process. Lett., 18: 73-46.
6. Perry, K., 1985. Randomized byzantine agreement. IEEE Trans. on Software Engineering, SE-11: 539-546.
7. Garcia-Molina, H. and D. Barbara, 1985. How to assign votes in a distributed system. J. ACM, 32: 841-860.
8. Perry, K. and S. Tueng, 1986. Distributed agreement in the presence of processor and communication faults. IEEE Trans. on Software Engineering, SE-12: 477-482.
9. Pankaj, J., 1994. Fault Tolerance in Distributed Systems. PTR Prentice Hall.
10. Lorenzo, A., D. Malkhia, E. Pierce and M. Reiter, 2001. Fault detection for byzantine quorum systems. IEEE Trans. on Parallel and Distributed Systems, 12: 996-1007.
11. Yin, J., J. Martin, A. Venkataramani, L. Alvisi and M. Dahlin, 2003. Separating agreement from execution for byzantine fault-tolerant services. Proc. 19th ACM Symp. on Operating Systems Principles, NY, pp: 15-28.