Adaptive Failure Detection Algorithm for Grid Systems

Dong Tian[1,2], Tai-ping Mao[2], and Jun Xie[2]

[1] College of Software Engineering, Chongqing University, Chongqing, P.R. China
tiandong@cqu.edu.cn
[2] Guizhou Electronic Computer Software Development Center, Guizhou, P.R. China
{mtp,xiejun}@mail.gzst.gov.cn

Abstract. Aimed at the grid system being more in failure and existing failure detection algorithms being not able to satisfy the unique requirement of grids, it was presented to a kind of adaptive failure detection algorithm in this paper. According to the characteristics of grids and the small world theory, the authors established a sort of small world based grid system model and a sort of failure detection model. By means of combining unreliable fault detection method with heartbeat strategy and grey prediction model, it was designed to dynamic heartbeat mechanism, and presented to the adaptive failure detection algorithm for grid systems further. Experimental result demonstrates that it is valid and effective in method, and it can be used for fault detection under grid environment.

Keywords: Grid, Small-world, Grey prediction, Heartbeat strategy, Fault detection.

1 Introduction

Nowadays, in order to solve the model realistic problems accurately, large applications are designed to run for days, weeks, or longer until the task being completed. With emerging of grid technology, it is possible to construct such large-scale applications under grid environment. However, due to the dynamic and heterogeneous characteristics of grid, the developing, deploying, and executing of such applications is a great challenge. The common complaining from the grid users is that large jobs find it very difficult to make any forward progress because of failures. This situation will be exacerbated as the system gets bigger and applications become larger [1]. Hence, the fault tolerance is a key requirement for grid systems. Failure detection is a well-known as fundamental building blocks for fault-tolerant grid systems, and failure detection as a special kind of systems management services has received much attention in the literature and many protocols. Until the present day, though there are some failure detection methods for grid systems, and that can ensure the reliability of grid systems in some extent, but they still have two kinds of default, the one is that failure detectors are organized into layered architecture [2-4] or Gossip-like architecture [5], which can not satisfied with the scalability and flexibility requirement of

B.-y. Cao, C.-y. Zhang, and T.-f. Li (Eds.): Fuzzy Info. and Engineering, ASC 54, pp. 288–296.
springerlink.com

grid. The other is that it is almost based on the static heartbeat mechanism to implement failure detection, the send time and arrival time of heartbeat message are fixed, which can not meet the dynamic requirement of grid systems. In this paper, we presents a sort of adaptive failure detection algorithm, which addresses the unique requirements for failure detection in grids. Based on small world theory [6]. We created the grid system model and adaptive failure detection model. Furthermore, combining unreliable failure detection method [7] with heartbeat strategy and "small samples" grey prediction model [8], we implemented a sort of adaptive heartbeat mechanism, and presented the adaptive failure detection algorithm for grid systems. At last, we demonstrated the correctness and effectiveness of the algorithm by simulation experiments.

2 Failure Detection Model

In the above, we pointed out that existing failure detection algorithms which based on layered architecture or gossip-like architecture can not meet the failure detection requirements in grid environment. After deeply research, we found that small world model has more superiority than hierarchical/gossip-like architectures in dynamics and scalability areas, the reasons are that (1) the nodes in small world model have very small mean distance, and need not consider the network topology, and (2) it has large known coefficient between nodes, and (3) Kleinberg [9] has theoretically proved it is the truth that small world has more superiority than hierarchical/gossip-like architectures and it can lower system cost effectively. So we designed the grid system model and failure detection model based on small world model.

2.1 Grid System Model

Assume considering a grid system as an assemble of limited number of processes, denoted by $G = \{p_1, p_2, ..., p_n\}$, communication and synchronization between processes is by means of sending and receiving messages, and the main failure type of system is the process crash, every two processes are connected by network, and arbitrary two processes can directly communicate by the network. Based on Shan Eerfan's small work construction method [10], we construct the grid system model, depicted as Fig.1. There are two kinds of processes in every virtual organization (VO), they are management process and normal process. The management process is responsible to manage the join/leave activity of normal processes, and maintain some long-link between local VO and other VO.

2.2 Failure Detection Model

According to the definition of failure detection [7] and above grid system model, considering a failure detection system as an assemble of limited number of failure detection modules, denoted by $FDS = \{M_1, M_2, ..., M_n\}$. Furthermore, we assume that failure detection modules will be failure only if process is in failure. Every failure detection module M_i attaches to one process p_i, and has a dynamic

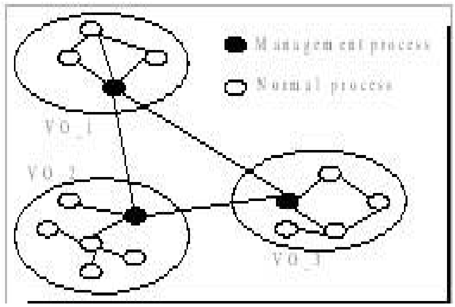

Fig. 1. Grid system model based on small world

failure suspect assemble. That is, according to p_i's failure detection module Mi suspect process p_j failure or not, suspected dynamically add/delete p_j. Thep_j randomly select k processes as it monitoring processes. When process p_i begin to suspect $p_j$has failed, it send confirm request to other k-1 monitoring processes at once.

3 Adaptive Failure Detection Algorithms

Based on above models, we firstly combine grey prediction with heartbeat mechanism to design an adaptive prediction mechanism of heartbeat message arrival time, which meets the dynamic requirement of grid. Then, we present the adaptive failure detection algorithm based on unreliable failure detection method and the heartbeat mechanism.

3.1 Adaptive Heartbeat Mechanism

3.1.1 Basic Motivation

$\forall p_i, p_j \in G$, p_i has failure detection module M_i, p_j has failure detection module M_j, if M_i periodically send "I am alive" heartbeat message to M_j, then we call p_j is p_i's monitoring process, and p_i is a monitored process. According to the recent past K times of heartbeat message arrival time and the real-time prediction strategy, we construct GM (1, 1) grey prediction model to adaptively predict the K+1th heartbeat message arrival time.

3.1.2 GM (1, 1) Based Prediction Mechanism of Adaptive Heartbeat Message

M_j collect the recent past K times of heartbeat message arrival time, and look them as the original sequence to construct GM (1, 1) grey prediction model, and to adaptively predict the K+1th heartbeat message arrival time further, the detail procedure is as follows.

Step 1. Get current time sequence. Collecting the recent passed K times of heartbeat arrival time as prediction samples to form original sequence, which denoted by:

$$t^{(0)} = (t^{(0)}(1), t^{(0)}(2), t^{(0)}(3), \cdots, t^{(0)}(K))$$

where K is the number of sample.

Step 2. Do accumulated generating operation (1-AGO) formation of$t^{(0)}$.

Defined $t^{(1)}$ as:

$$t^{(1)} = (t^{(1)}(1), t^{(1)}(2), t^{(1)}(3), \cdots, t^{(1)}(K))$$

where $t^{(1)}(1) = t^{(0)}(1)$,and $t^{(1)}(K) = \sum_{m=1}^{k} t^{(0)}(m)\ k = 2, 3, \cdots, K$.

Step 3. Form GM (1, 1) model.

From the AGO sequence of $t^{(1)}$, we can form a GM (1, 1) model, which corresponds to the following first-order difference equation:

$$dt^{(1)}(K)/dK + at^{(1)}(K) = b(1)$$

Therefore, the solution of Eq.(4) can be obtained using the least square method. That is,

$$\hat{t}^{(1)}(K) = (t^{(0)}(1) - \tfrac{\hat{b}}{\hat{a}}) \times e^{-\hat{a}(K-1)} + \tfrac{\hat{b}}{\hat{a}}(2)$$

where $[\hat{a}, \hat{b}]^T = (B^T B)^{-1} B^T T_n$

$$\text{and } B = \begin{bmatrix} -0.5(t^{(1)}(1) + t^{(1)}(2)),\ 1 \\ -0.5(t^{(1)}(2) + t^{(1)}(3)),\ 1 \\ \cdots\cdots \\ -0.5(t^{(1)}(K-1) + t^{(1)}(K)),\ 1 \end{bmatrix}$$

$$T_K = [t^{(0)}(2), t^{(0)}(3), t^{(0)}(4), \cdots, t^{(0)}(K)]^T$$

We obtained $t^{(1)}$ from Eq.(2). Let $t^{(0)}$ be the fitted and predicted series,

$$\hat{t}^{(0)} = (\hat{t}^{(0)}(1), \hat{t}^{(0)}(2), \hat{t}^{(0)}(3), \cdots, \hat{t}^{(0)}(K) \cdots),$$

where $\hat{t}^{(0)}(1) = t^{(0)}(1)$, $\hat{t}^{(0)}(K) = (\hat{t}^{(1)}(K) - \hat{t}^{(1)}(K-1)$

Step 4. Predict the next heartbeat arrival time.

Applying the inverse accumulated generating operation (IAGO), we then have

$$\hat{t}^{(0)}(K) = (t^{(0)}(1) - \frac{\hat{b}}{\hat{a}}) \times (1 - e^{\hat{a}}) \times e^{-\hat{a}(K-1)}(3)$$

Where $t^{(0)}$(K+1) is the next heartbeat arrival time.

Step 5. Form new prediction model.

Upon receiving the (K+1)th heartbeat, the monitoring process pj reads the process clock and stores the heartbeat rank and arrival time into a sliding window (thus discarding the oldest heartbeat), and form new prediction model as follows.

$$t^{(0)}_{new} = \{t^{(0)}(2), t^{(0)}(3), \cdots, t^{(0)}(K), t^{(0)}(K+1)\}$$

Then, repeat steps 2- 4 to predict the (K+2)th heartbeat arrival time, and so on.

3.2 Adaptive Failure Detection Algorithm

According to above models and dynamic heartbeat mechanism, combining with unreliable failure detection method, we design the adaptive failure detection algorithm as depicted in algorithms 1.

Algorithms 1.

Step 1. Constructing small world model.

According to the general number of grid processes, constructing small world model depicted as Fig 1, the number and the size of cluster are determined by formulation (4) and (5)

$$M = 2 \log S_C N_{Total} (4)$$

$$S_C = \lambda \sqrt{N_{Total}} \log N_{Total} (5)$$

Where M is the number of clusters, S_c is the size of cluster, N_{Total} is the total number of grid processes,is a reference for computing size of clusters.

Step 2. Constructing failure monitoring relation among processes.

$\forall p_i \in G, i = 1, 2, \cdots, n$,according to the model in step 1, it will belong to at least one cluster.

The p_i randomly select k processes from the other S_c -1 member processes, that is, p_i's failure detection module M_i periodically send "I am alive" message to the k failure detection modules which attach to those processes.

(1) If p_i is a normal process of the cluster, then it randomly select k processes in the same cluster as its monitoring processes.

(2) If p_i is a management process of the cluster, then it randomly select $|k/2|$ normal processes as well as $|k/2|$ long-linked processes to serve as its monitoring

(3) If pi knows that a new process p_j has joined in its cluster, p_i will invite p_j to serve as its monitoring process on the probability of k/S_C . When the number of monitoring processes more than k at that time, it will randomly require a monitoring process to cancel monitoring relation. processes.

(4) If p_i knows that one or more of its monitoring processes have failed, it will randomly add one or more processes to serve as its monitoring processes according to (1)-(3).

Step 3. Suspecting failure.

For every monitoring process which monitors p_i.

(1) Ranking the heartbeat messages arrival time of p_i in sort, and adds them into in original sequence. When the arrival number of heartbeat messages equal to K, trigger the GM (1, 1) model to predict the arrival time of (K+1)th heartbeat message.

In order to ensure the real-time and dynamic characteristics of grid systems, when the (K+1)th heartbeat message arrived, adds the real arrival time into the end of $t^{(0)}$, deletes the first arrival time of $t^{(0)}$, and constructs new original sequence $t_1^{(0)}$ to predict the arrival time of (K+2)th heartbeat message.

(2) If not receiving the "I am alive" message of p_i , then begins to suspect the failure of p_i .

Step 4. Confirming failure.

For every monitoring process which monitors p_i, if it begins to suspect the failure of p_i, then it sends confirming failure request to other k-1 monitoring processes.

(1) If one or more other monitoring processes return messages that p_i has not failed, then it stops suspecting.

(2) If it receives none not fail message, it confirms the failure of p_i, and broadcasts the failure message in the whole grid system.

4 Experimental Results

Analog to reference [4], we establish a real grid testing environment to test the performance of algorithm 1.

4.1 Experimental Setup

The experimental environment is made up of three resource sites on Chinese Education and Research Network (CERNET), they are one 20-node cluster in the national linux technology lab (LinuxCenter) of Chongqing University, one six-node cluster in Netmobilab of Chongqing University, and two PCs in Guizhou electronic computer software development center. Every node of LinuxCenter cluster is equipped with Pentium IV processor at 2.4 GHz and the memory is 512MB, the operating system is also Red Hat Linux 9 (kernel 2.4.20), and the nodes are connected by 100 M Ethernet. Every node of Netmobilab clster is equipped with Pentium IV processor at 2.4 GHz and the memory is 512MB, the operating system is also Red Hat Linux 9(kernel 2.4.20), and the nodes are connected by 100 M Ethernet. Every PC is equipped with Pentium III processor at 766 MHz and the memory is 256MB, the operating system is also Red Hat Linux 7.2 (kernel 2.4.9).

4.2 Evaluation Criteria

In order to evaluate the QoS of failure detection algorithms, Chen [11] presented a series of criteria, and the typical criteria are as follows.

1. Failure detection time (T_D), the time periods between $p_i$failed and monitoring process p_j begins to suspect the failure.

2. Mean mistake rate (λ_M), mistake rate of the failure detection algorithm made.]

4.3 Experimental Results

Experiment 1, determining the size of prediction sample space.

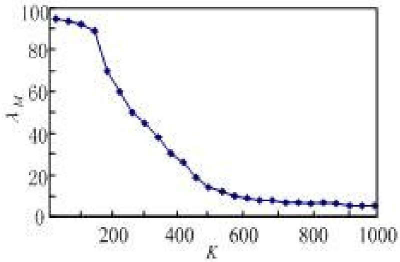

Fig. 2. K-value of algorithm 1

In order to test the effectiveness of algorithm 1, we must know the prediction sample space of the adaptive heartbeat message mechanism, i.e., we firstly must determine K.

The experiment involves two computers, one node computer from LinuxCenter serves as monitored process pi, the other node computer from Netmobilab serves as monitoring process p_j. All messages are transmitted with UDP protocol. Neither machine failed during the experiment.

The experiment lasts for 48 hours, during which heartbeat message is generated every 200ms. In experimental periods, the mean time interval of receiving message is 228.7ms, where the minimum delay is 210.4ms, and the maximum delay is 479.6ms, the number of sent messages is 835,102, the number of received messages is 817,204.(message loss rate is 2.14%).

By changing K-value from 5 to 1000, we compute λ_M of algorithm 1, the results is depicts in Fig.2.

As shown in Fig.2, the experiment confirms that the mistake rate of algorithm 1 improves as K increases. But the curve seems to flatten slightly when the size is more than 630, meaning that increasing it further yields only little improvement.

Experiment 2, determining k of monitoring processes.

Looking processes which running on every node computer as grid processes, and randomly terminating one or more processes to simulate failure. In different k-value condition, the change between processes number and λ_M is depicted in Fig. 3.

As shown in Fig.3, the larger of k-value, the lower ofM. When total number of processes is 1600, if k>10, then λ_M <10%.

However experimental results shows that when k>10, the system load increasing quickly (if k=12, system load is about 90%), and the system performance is decreased accordingly. So, in real world grid systems, we suggest that k should be in [4, 6].

Experiment 3, Comparison between algorithm 1 and HBM .

Given K=400, k=4, comparing algorithm 1 with reference [2], which employed hierarchical and static heartbeat mechanism (HBM). The result of 5hours experiment is depicted as in Figure 4.

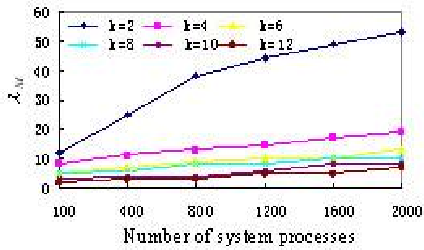

Fig. 3. K-value of algorithm 1

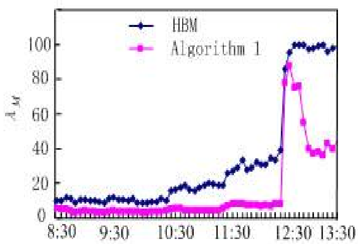

Fig. 4. Comparison between algorithm and ALTER

As shown in Fig. 4, algorithm 1 is much more lower than HBM, especial in peak period, when HBM is out of work, the algorithm is still work well. Experiment 4, Comparison between algorithm 1 and ALTER.

Given K=100, k=4 and the total number of system processes is 400, comparing algorithm 1 with reference [4], which employed hierarchical and dynamic heartbeat mechanism (ALTER).

The computing results show that the mean failure detection time (T_D) of ALTER is 278.6ms, and algorithm 1 is a bit longer, 336.2ms. We think the reason is that ALTER only employs one process to serve as monitoring process, but algorithm 1 employs multi-process to serve as monitoring process.

5 Conclusions and Future Work

Failure detection is a fundamental building block for ensuring fault tolerance in grid systems. In this paper, based on small world theory, we constructed the grid system model and adaptive failure detection model. Furthermore, combining unreliable failure detection method with heartbeat strategy and grey prediction model, we implemented an adaptive heartbeat mechanism, and presented the adaptive failure detection algorithm for grid systems. Moreover, experimental results show that under condition of the experiment determined K and k, compared with the static heartbeat mechanism based grid failure detection algorithm, the algorithm presented by authors has much more lower mistake rate.

Compared with the ones which employ dynamic heartbeat mechanism, the algorithm presented by authors has higher accuracy. In the near future, we will implement a failure detection middleware based on the algorithms presented by this paper.

Acknowledgments

This work is supported by GuiZhou Science and Technology Fund, under contract No. Qiankehe J Zhi 2007(2232). We are grateful to Gang Yang for providing experimental programs, as well as Changze Wu, Xiaobo Ji for many discussions.

References

1. Li, Y., Lan, Z.: Using Adaptive Fault Tolerance to Improve Application Robustness on the TeraGrid. In: Proc. of TeraGrid 2007 (2007)
2. Hwang, S., Kesselman, C.: A Flexible Framework for Fault Tolerance in the Grid. Journal of Grid Computing 1(3), 251–272 (2003)
3. Abawajy, J.H.: Fault Detection Service Architecture for Grid Computing Systems. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004, vol. 3044, pp. 107–115. Springer, Heidelberg (2004)
4. Shi, X., Jin, H., Han, Z., et al.: ALTER: Adaptive failuredetection services for Grids. In: Proc. of the 2005 IEEE International Conference on Services Computing (SCC 2005), pp. 355–358 (2005)
5. Jain, A., Shyamasundar, R.K.: Failure Detection and Membership in Grid Environments. In: Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (GRID 2004), pp. 44–52 (2004)
6. Kleinberg, J.: The Small-World Phenomenon: An algorithmic perspective. ACM Synpon Theory of Computing (2000)
7. Chandra, T.D., Toueg, S.: Unreliable failure detectors for reliable distributed systems. Journal of ACM 43(2), 225–267 (1996)
8. Deng, J.: Control problems of grey system. Systems Control Lett. 5, 288–294 (1982)
9. Kleinberg, J.: Complex networks and decentralized search algorithms. In: Proc. of the International Congress of Mathematicians (ICM) (2006)
10. Erfan, S., Zhuhui, D.: Efficient grid service location mechanism based on virtual organization and the small-world theory. Jounal of Computer Research and Development 40(12), 1743–1748 (2003)
11. Chen, W., Toueg, S., Aguilera, M.K.: On the quality of service of failure detectors. IEEE Transactions on Computers 51(2), 13–32 (2002)