Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp405-409)

Research of Heartbeat Detection Protocol Based on Multiple Master-nodes in Multi-machines Environment

Dong Jian , Liu Hongwei, Zuo Decheng, Yang Xiaozong
School of Computer Science and Technology
Harbin Institute of Technology
No.92, West Da-Zhi Street, Harbin, Heilongjiang, 150001
CHINA

Abstract: Heartbeat is the most important method of fault detection in the distributed systems. A heartbeat protocol allows two nodes to detect the states of each other by exchanging messages periodically. But it remains two problems in the heartbeat detection of multi-machines. That are, the disagreement of detection results and the over costs of the master-nodes. This paper promotes a heartbeat protocol basing on multiple master-nodes (HPMM). HPMM solves the problem of the disagreement in detection results by voting and electing among master-nodes, and also improves the continuous work time as well as the availability of the system. In addition, the detection costs can be reduced by distributing workload into multiple master-nodes.

Key-Words: distributed; fault detection; heartbeat detection; voting

1 Introduction

With the development of Internet, providing highly available service becomes increasingly important. Fault detection is one of critical technologies of designing highly available system. Heartbeat mechanism is one of the most common ways of fault detection in the distributed systems. A heartbeat protocol allows two nodes to detect the states of each other by exchanging messages periodically. As long as a node *p* keeps receiving right beat messages from a node *q* , *p* recognizes that *q* is up; if p does not receive any beat messages from *q* for a long time, then *p* recognizes that *q* has terminated or failed. The heartbeat mechanism is simple and easy to be realized. At the present time, heartbeat protocols have been applied in many fields. For example, they are used in system diagnosis [1], network protocols [2], reaching agreement [3], and mobile computing [4].

Heartbeat detections of multi-machines often adopt Master/Slave structure with single master-node, it chooses a master-node from *N* nodes, and the master-node executes a binary heartbeat protocol with every other *N-1* nodes. For example, the Expanding Heartbeat Protocol (EPH) [5], Heartbeat Detection Protocol Basing on Election in Multi- machine Environment (BEHP) [6]. The former adopts the static master-node structure. In order to prevent the master-node from becoming a bottleneck of the system, the latter uses the dynamic master-node structure with ability to elect. However, as most of the current heartbeat protocols, they have several common problems as follows.

(1) After one node *p* of the heartbeat detection becomes overtime, it is impossible to distinguish whether the node *q* has been failed (Crashing) or the communication medium from p to q has been failed (Link Failure), so the system can not make sure the type of the fault. Under such conditions, in order to avoid the disagreement of detection results, common detection algorithms will make the whole system terminate, and carry out a out-line diagnosis. Although the great mass of the nodes can also work normally at that time. Ref. [5] defined it as a basic principle of heartbeat protocol. Thus these problems induce the interruption of the service, and reduce the availability of the system.

(2) The over costs of detection of the single master-nodes will accelerate its failure. Even in the BEHP, the failure of master-node can produce frequent election, which also would lower the efficiency of the system.

Aiming at the above problems, this paper promotes a heartbeat protocol basing on multiple master-nodes (HPMM). HPMM solves the problem of the disagreement in detection results by voting and electing among multiple master-nodes, and also improves the continuous work time as well as the availability of the system. In addition, the detection costs can be reduced by multiple master-nodes network structure.

2 Structure of HPMM

In order to simplify descriptions, it is supposed that there is a discrete global timer, and the scope of the timer *T* is defined to the natural muster. The timer is a fictional one, and can not be visited by the nodes.

Assume that the whole network system is *G=(V, E)*. Among them, *V* represents the *N* nodes which constitute the system, namely *V*={*1,2,3,…,N*}, while *E* represents the links between those nodes. It is supposed that the system is fully- connected, namely $E = \{(p,q) \mid \forall p, q \in V, p \neq q\}$. Fig.1 shows the heartbeat structure of HPMM.

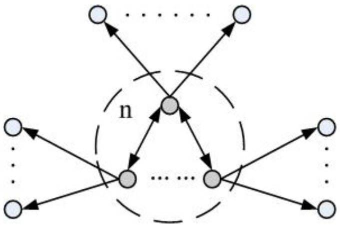


Fig. 1: HPMM Heartbeat Protocol Structure

As shown in the figure, n (*2 < n < N*) nodes constitute the set of master-node, and the other N-n nodes are defined as slave-nodes. Any two master-nodes commit the binary heartbeat detection with each other. Every master-node takes charge of several slave-nodes, and performs heartbeat detection periodically. Consider this two-layer heartbeat detection structure, and we can define the HPMM as a 5-tuple *H*.

$$H = (M, \Sigma, F, D_M, D_{M-S})$$

$$D_{M,} : T \rightarrow 2^{M \cup E_M}$$

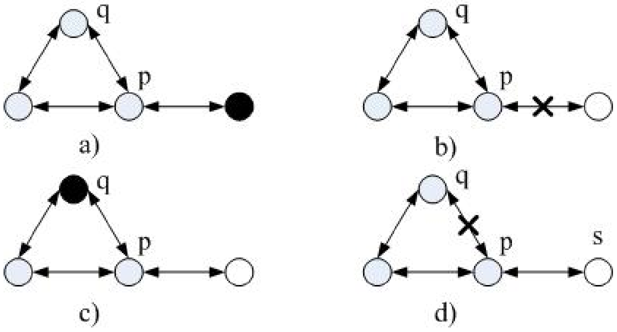
$$D_{M-S} : T \rightarrow 2^{M \cup E_{M-S}}$$

M is the set of master-nodes, $\Sigma = \{S_p \mid p \in M\}$, S_p is the set of slave-nodes which are managed by the master-node p. For every couple of master-node, e.g. *p* and *q*, it must be true that $S_p \cap S_q = \phi$. F is a set of fault-models which can be detected by *H*, and *F*={Crashing, Link-failure}. D_M is the first layer—heartbeat detections among master-nodes, $E_M = \{(p,q) \mid p, q \in M, p \neq q\}$; and $D_{M\text{-}S}$ is the second layer — heartbeat detections between master-node and slave-node , $E_{M-S} = \{(p,q) \mid p \in M, q \in S_p\}$. *D(t)* is the result of fault detection at t $(t \in T)$, $D(t) = D_M(t) \cup D_{M-s}(t)$.

3 Working Principle of HPMM

The fault-models can be divided into four classes by the structure of HPMM, as shown in the Fig.2.

Among D_M, every master-node executes binary heartbeat detection with another master-node in terms of heartbeat period T_M. When c) or d) takes place, the node can not do a judgment to the current system states any more, and now, the master-node *p* sends out its voting requests to other master-nodes for finding the position where faults take place. After receiving voting requests, the other master-nodes send their detection results of *q* to *p*, and then node *p* makes its decision basing on the other master-nodes' results of heartbeat detection. If other nodes' results are consistent with *p*, they judge *q* a failed node, at the same time, *p* tells other master-nodes to turn into electing state. All the remainders elect a new master-node from the slave-nodes to replace the failed node *q*, in order to maintain the integrality of the first-layer heartbeat. If any master-node believes the node *q* is right, it is believable that the link between *p* and *q* has been failed. A master-node has to be abandoned according to the PRI of the nodes *p* and *q*, and then an election is carried on by those master-nodes. The voting algorithm is shown in the Fig. 3.



a) slave-node fail b) link between master-node and slave-node fail
c) master-node fail d) link between master-nodes fail
Fig. 2: Classification of Fault-Models of HPMM

When it needs to elect new master-node, all the nodes in *M* choose a node *r* whose PRI is the highest to inquire its state by the information in Σ (every correct master-node keeps a Σ , which keeps consistence by the message of heartbeat). If *r* can respond accurately, it will be joined into *M*; otherwise, choose the node of the second highest PRI to elect again and again, until they elect a new master-node. If all the elections fail, the scale of the master-nodes degrades to *n-1*, and the system keeps on running. The election algorithm is shown in Fig.4.

The support to the electoral ability of master-node makes the set of master nodes form a standby system, which can make use of the system resource efficiently, as well as improve the continuous work time, and the availability of the system.

```
Procedure Vote (node $q$)
 || Task1:
     For all $k \in M \wedge q \neq k$ do send$_{p,k}$(VOTE)
     While((receive$_{p,k}$(result)!=true from any k)$\wedge$!TimeOut($T_M/2$))
     if(result==true)
         Report(Link( $p \to s$ ) Fail)
         if ($p$ is prior to $q$) M=M-{ $q$ }, Election($S$)
         else $p$.stop()  //the node whose PRI is lower turn into failed state
     else M=M-{ $q$ }
           Election($S$)
 || Task2:  //for any $k \in M \wedge q \neq k$
      Upon receive$_{k,p}$(VOTE) do
        send$_{k,p}$( $R(k \to q)$)   // $R(k \to q)$is the latest result of $k$ to $q$
```

Fig. 3: Voting algorithm of master-node

```
For every node $p \in M$
   Procedure Election($S$)
    || Task1：
     if（|$S$|==0） return
       send$_{p,r}$(Change) //$r$ is the node whose PRI is highest in $S$
       while(!TimeOut($T_M/2$))
         if(reveive$_{p,r}$(C-ACK))
                for (all $q \in M$ ) do send$_{p,q}$(true)
               Return
      for (all $q \in M$ ) do send$_{p,q}$(false)

    || Task2:
      while(!TimeOut($T_M$))
           Upon receive$_{p,q}$(result), $q \in M$
           if (result==true) $r$[ $q$]=true else $r$[ $q$]=false
        if (any $i$, $\exists$ ($r$ [$i$]==false))
          Election($S$-$r$)
        else send$_{p,r}$(Master-Ack)//send master-node acknowledgement
           $M=M+\{r\}$
```

Fig. 4: Electing algorithm of master-node

Among $D_{M\text{-}S}$, every master-node sends heartbeat message to slave-nodes which it takes charge of in terms of heartbeat period T_S（$0< T_M < T_S$）, and the slave-node s responds passively to the heartbeat detection. When a) or b) takes place, node p requests other master-nodes to vote for node s, and the voting process is similar to the algorithm in Fig. 3, except that every master-node must make separate heartbeat detection in terms of heartbeat period $T_S/2$. If more than half of master-nodes' voting results in accordance with those of p, p can makes a verdict that the slave node has been failed. Node p will delete s from S_p and inform other master-nodes, besides, all of the master-nodes will modify their Σ . If other master-nodes believe s is right, they consider the link between p and s had been failed. Node s will record the failed link, and inform other master-nodes, at the same time, it is assigned to another master-node to take charge of *s*.

4 Analyses and Emulation

(1) Fault Detection Delay

Assume that λ_n is the failure rate of nodes and λ_l is the failure rate of links, both of them accord with exponential distribution. The time when a failure occurs in a heartbeat interval [0，T] is:

$$l_T = \int_0^T t \cdot \frac{\lambda e^{-\lambda t}}{1-e^{-\lambda T}} dt$$

$$\lambda = \lambda_n + \lambda_l$$

So the average fault detection delay in DM is:

$$T_{D_M} = (T_M^{'} - l_{T_M}) + T_M^{'}/2$$

$$= \frac{3}{2}T_M^{'} - \frac{\frac{1}{\lambda} - e^{-\lambda T_M}(T_M + \frac{1}{\lambda})}{1-e^{-\lambda T_M}}$$

$T_M^{'} = T_M + T_h$， T_h is the cost of heartbeat itself. $T_M^{'}/2$ is the voting time, which is caused by the need of voting period to make sure fault types. We can compute $T_{D_{M-S}}$ as well, just by changing T_M to T_S.

(2)System Costs and the Value of *n*

Firstly, considering the system costs of binary heartbeat protocol, we get data in Table1 by executing binary heartbeat protocol between two PC connected by 100M Ethernet.

Table 1: Costs of Binary Heartbeat Detection (Unit： sec.）

Time costs of transmitting heartbeat messages	Time costs of taking up CPU（heartbeat sender + receiver）
$\theta(10^{-4})$	0.0012

As shown in Table 1, time spent in transmitting heartbeat messages is far smaller than the time spent in dealing with heartbeat messages. So the transmitting costs can be ignored while computing costs of heartbeat protocol. In this case, the system costs lie on the number of heartbeat message produced by nodes in unit time.

Suppose that $T_S=2T_M$， and consider a slave-node heartbeat period, e.g. [0, T$_S$]. A master-node’ cost can be represented by $f_p(n)$, the number of messages processed by node p. Assume that the master-nodes are assigned slave-nodes equally, we can compute $f_p(n)$ as followings.

$$f_p(n) = 2(2(n-1)) + 2\frac{N-n}{n}$$

$$f_p^{'}(n) = 4 - 2\frac{N}{n^2} = 0$$

As a result, when $n = \lfloor \sqrt{N/2} \rfloor$ or $n = \lfloor \sqrt{N/2} \rfloor + 1$， the value of $f_p(n)$ is minimum. In this case, the costs of master-nodes are least, although the costs of the whole system increase. It accelerates the response of master-nodes and minimizes the possibility of failure. The result of emulation experiment can be seen in Fig. 5.

The emulation environment is based on Windows OS, implemented by Visual C++ (Version 6.0), and simulates nodes of practical system by Threads. Software runs in a PC server basing on SMP structure in emulation process.

During emulation, the number of the nodes in the system *N*=200, and *Ts* =2s .The values of the number of master-nodes *n* are (1，5，10，…,200), as shown in Figure6.

We carry on separate emulation experiments for each value of *n*, and spend *T*=100 hours on experiment. The value of axis Y is the average number of messages which all the nodes have deal with in T_s. As shown in Figure 5, the testing values can not be consistent with the theoretic values exactly, because we ignore the processing time of heartbeat itself while computing. As a result, the data gained from experiment is less than the theoretic values, and T_h will increase as the costs of system increase.

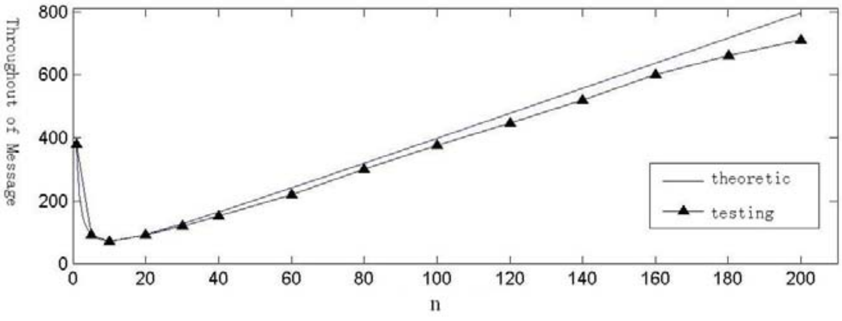


Fig.5: Costs of Master-Nodes

Table 2: Comparison among multi-machine heartbeat protocols

	detection types		detection delay	System costs of master-nodes	Electing ability of master-nodes	availability
	Crashing	Link Failure				
HPMM	support	support	longer*	small	yes	high
BEHP	support		short	big	no	middle
EHP	support		middle	big	yes	low

* Detection delay of HPMM contain voting period

So, in the case the value of *n* is greater, the difference between theoretic values and testing values would be greater.

(3) Qualitative Comparison with other Heartbeat Protocols of Multi-Machine

Compare HPMM with the Expanding Heartbeat Protocol (EPH) and Heartbeat Detection Protocol Basing on Election in Multi-machine Environment (BEHP) qualitatively, the result is shown in Table 2. The fault detection delay of HPMM is longer than those of EHP and BEHP, because it supports to determine fault types. But preferable fault detection ability and voting ability of master-nodes make the system adopting HPMM more available.

5 Conclusion

High availability distributed system plays an important part in current network application. Heartbeat detection is one of the important methods which design highly available system. This paper analyzes existing multi-machine heartbeat protocols, and brings forward a heartbeat protocol basing on multiple master-nodes (HPMM). HPMM makes judgment of node detection and link detection effectively and solves inherent problems existing in heartbeat detection by adopting voting mechanism of multiple master-nodes. Besides, master-node is able to elect new master-node, and make use of system resource effectively. The combination between them improves the continuous work time as well as the availability of the system. Distributing workload into multiple master-nodes minimizes the costs of system detection effectively and improves response speed of system. We will develop correlative application basing on HPMM, for instance, fault tolerance group-membership protocol, fault tolerance routing algorithm and so on.

References:

[1] Barborak, M., M. Malek, A. Dahbura. The Consensus Problem in Fault-Tolerance Computing. *ACM Computing Surveys*, Vol.6, No.25,1998, pp.171-220

[2] Braden, R., editor. Requirements for Internet hosts-Communication Layers. *RFC 1122*, 1989

[3] Cristian, F. Reaching Agreement on Processor-group Membership. *Distributing Computing*, No.4 1991,pp.175-187

[4] D. K. Pradhan, P. Krishna, Nitin H. Vaidya. Recoverable Mobile Environment: Design and Trade-off Analysis. *Technical Report, Texas A & M University* , 2001

[5] M. G. Gouda, T. M. McGuire. Accelerated Heartbeat Protocols. *Proceedings of the The 18th International Conference on Distributed Computing Systems, ICDS*, Washington DC., USA, 1998, pp.202-210

[6] Zonghao Hou , Yongxiang Huang, Shouqi Zheng, Xiaoshe Dong. Design and Implementation of Heartbeat in Multi-machine Environment. *17th International Conference on Advanced Information Networking and Applications, AINA'03*, Xi'an, China, 2003, pp.583-586