基于二次检测的失效检测算法的改进①

范媛媛，唐 蔚

(同济大学 计算机科学与技术系，上海 201804)

摘 要：为了使失效检测满足分布式网络环境的要求，针对分布式网络环境中存在的网络延迟和丢包两种情形，提出了基于失效检测点和二次检测模式的失效检测改进算法。该算法根据误判次数动态计算二次检测时间，在失效检测的准确性和效率之间作了较好的权衡。理论分析和测试结果均表明该算法能够适应网络环境的变化并有效地减小网络丢包对失效检测算法的影响。

关键字：失效检测；检测点；二次检测；丢包率；网络延迟

Improvement of Failure Detection Algorithm Based on Second Detection

FAN Yuan-Yuan, TANG Wei

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

Abstract: In order to satisfy requirements of distributed network environment, where network latency and message loss exists, an improved failure detection algorithm based on the concept of fresh point and second detection mode is proposed. This algorithm calculates the second detection time dynamically, and gives a good tradeoff between accuracy and efficiency. Both theoretical analysis and testing results show that the algorithm can adapt to changes of network environment and effectively reduce the impact of message loss.

Key words: failure detection; fresh point; second detection; network packet loss rate; network latency

失效检测是分布式容错系统的基础和关键技术之一[1]。目前失效检测主要采用超时机制，根据被检测实体在规定时间内有无应答来判断该实体是否失效。失效检测的具体实现可分为轮询和心跳两种模式，由于心跳模式中消息单向传输，能有效地减少网络中传输的消息数量，因而在实际系统中被广泛应用。

Chandra 等人[2]提出了失效检测的完整性和准确性两个基本属性，完整性体现了失效检测器具有最终怀疑每一个失效进程的能力，准确性体现了失效检测器不会将正确的进程误判为失效的能力。

实际分布式环境中网络状况复杂多变，存在消息延迟和丢失等多种情况。传统的基于心跳模型的失效检测算法未考虑网络环境等因素的影响，因而在完整性和准确性上均达不到分布式系统应用的要求。Bertier[1] 、Chen[3]、Fetzer[4]等人在传统的失效检测算法研究的基础上分别提出了各自的自适应失效检测算法和相应改进。

本文首先建立了异步分布式网络环境的模型，并在上述研究的基础上针对网络丢包和延迟两种情形对失效检测时间的估计值做了相应改进，使算法在有效降低失效检测错误率的同时较好地控制检测时间。

1 相关研究

传统基于心跳模型的失效检测算法为：进程 p 以固定的时间间隔向检测进程 q 发送心跳消息，q 收到消息则信任 p，并以固定的超时时间 T 启动计时，若在 T 结束时仍没有收到 p 的心跳消息更新则开始怀疑 p，直到收到新的心跳消息为止。由于在实际的系统中存在网络延迟、时钟漂移和系统负载等现象，该算法存在过早超时、检测时间过长等问题[3]。

① 收稿时间:2011-05-15;收到修改稿时间:2011-06-19

Chen 等人提出了基于失效检测点的自适应失效检测算法，根据历史心跳消息的平均网络延迟和消息的发送时间预测下一个心跳消息的到达时间。在 $t\in[\tau_i, \tau_i+1)$时刻进程 q 相信进程 p 当且仅当 q 收到心跳消息 m_j $(j\geq i)$，其中 τ_i 为由心跳消息的估计值 EAi 和固定的网络偏移量 α 相加得到的检测点。

该算法为下一个消息到达的时间提供了较好的估计值，但是固定的偏移量不符合动态变化的实际网络环境，且在实际应用中当网络丢包率较大时，单次检测有可能导致每个进程所维护的怀疑队列频繁地改变。

Bertier 等人在 Chen 算法的基础上，结合 Jacobson[5]动态修正网络偏移量的方法改进了失效检测点的估计值。此外利用二次检测模式构造失效检测算法的第二层，延长判断被检测进程失效的时间。然而该算法没有考虑网络丢包的影响，同时利用误判次数作为二次检测延长时间，检测时间会随着误判次数的增加而线性增加，导致当被检测节点真正失效时，不能迅速地被检测出来。

2　失效检测算法估计值

针对上述算法中失效检测点估计值存在的不足，本文首先先给出了分布式网络环境模型，并在此基础上综合考虑失效检测效率、准确性等因素，给出了新的失效检测点的估计值。

2.1 网络环境模型

分布式系统∑由 n 个节点组成，每个节点上都运行一个失效检测进程，这 n 个进程$\prod=\{p_1, p_2, \cdots, p_n\}$构成了分布式失效检测系统。

本文假设这 n 个节点分布在不可靠广域网中，节点间的链路不会产生或复制消息，但可以延迟或丢弃消息，且心跳消息丢失和消息延迟符合随机概率事件。

2.2 失效检测参数估计

消息丢失概率 PL：如果被检测进程向检测进程发送了 M 条心跳消息，检测进程只接收到了其中的 N 条心跳消息，则 PL 表示为：(M−N)/M。

错误率 Pe：在一次运行中，一个正确进程被错误地判断为失效的频度。在分布式系统中，错误率是由心跳消息延迟超过预设值和心跳消息丢失共同造成的。

假设两个检测点间的时间间隔为Δ_i，网络延迟为 D，进程 q 第 i 条消息 $m_i(q)$的发送时间为 $S_i(q)$，p 估计接收到 $m_i(q)$的时间为 $EA_i(q)$，实际接收到 mi 的时间为 $A_i(q)$。

心跳消息的延迟和消息丢失这两个事件相互独立[7]。当 $D\geq\Delta_i$ 时，检测端判断被检测对象失效，即 $Pe=P(D\geq\Delta_i)$；当 $D<\Delta_i$ 时，导致误判的因素是消息丢失，假设有连续有 k 条消息丢失导致误判，此时 P_e 正比于$(PL)^k$：

$$Pe=P(D<\Delta_i)(PL)^k+P(D\geq\Delta_i) \quad (1)$$

本文采用 Jacobson 提出的方法动态计算网络偏移量 α，并在 Chen 计算检测点的基础上考虑到实际应用的计算量和计算效率加入缓存机制，设缓存数为 n，当到达的消息数达到 n 时使用的 EA 估计值为：

$$EA_{(k+1)}(q)\leftarrow EA_k(q)+(Ak(q)-A_{(k-n)}(q))/n \quad (2)$$

此外，由于上述估计没有考虑心跳消息丢失对失效检测的影响，本文引入 Bertier 算法中二次检测的概念，并改进其二次检测的时间，以减少由心跳消息丢失造成的失效检测误判，同时提高失效检测效率。

当进程 p 收到 $m_i(q)$时，根据前面接收到的 i 条消息的平均延迟 E(D)预测第 i+1 条心跳消息到达的时间为：

$$EA_{(i+1)}(q)=E(D)(q)+S_{(i+1)}(q) \quad (3)$$

假设从 i+1 条心跳消息开始连续有 k 条消息丢失，p 收到 q 的第(i+k+1)条消息的时间为：

$$A_{(i+k+1)}(q)=S_{(i+k+1)}(q)+D_{(i+k+1)}(q) \quad (4)$$

p 在 $\tau_{(i+1)}$时间点没有收到消息，则启动二次检测等待时间$\Delta_p(q)$，若二次检测有效则有：

$$A_{(i+k+1)}(q)\leq EA_{(i+1)}(q)+\Delta_p(q) \quad (5)$$

综合 3、4、5 三式可得：$\Delta_p(q)\geq k\Delta$。

考虑到失效检测的效率，本文中 k 的值取 1 即 $\Delta_p(q)\geq\Delta$。根据实际情况，进程 p 错误判断进程 q 的次数越多，二次检测的时间应该越长，因此本文拟用错误率来定义二次检测时间即得到：$\Delta_p(q)=(1+P_e)\Delta$。

3　算法描述

每一个进程 p 周期性的向所有其它进程$\prod-\{p\}$发送心跳消息。当 p 在 t 时刻收到进程 q 发送的心跳消息时， p 利用上述分析的估计值计算 q 下一个心跳消息期望到达的时间 EA(q)和安全值 α，并由此确定进

程 q 的失效检测点 τ (q)。如果进程 p 当前怀疑 q，则将 q 从其怀疑的列表中移除，并调整错误率 Pe 和二次检测时间$\Delta_p(q)$。当前时间已经到达 q 的失效检测点 τ (q)，仍没有收到来自 q 的心跳消息更新，此时再等待二次检测时间$\Delta_p(q)$，如果仍没有收到心跳消息则怀疑进程 q。

具体算法描述如下：

对每一个进程 $p\in\Pi$ 执行：

Initialization:

$SUSPECTED_p\leftarrow\emptyset$

对于所有进程 $q\in\Pi-\{p\}$

$K=0$ {当前收到的最大消息序号}

$\tau_0(q)=\alpha_0(q)=var_0(q)=error_0(q)=0$

$delay_0(q)$=初始值

$n=0$ {缓存的消息数}

$s=0$ {错误判断的次数}

$P_e=0$ {错误率初始化为 0}

$\Delta_p(q)=0$ {二次检测时间}

Task 1:

在时刻 $i*\Delta$，发送 $m_i(p)$到 $\Pi-\{p\}$

Task 2:

若在时刻 t 收到从 q 发送的消息 $m_j(q)$

if ($j>K(q)$) **then** {收到消息更新}

$K(q)\leftarrow j$

if($K\le n$)then {更新 EA 的值}

$EA_{k+1}(q)\leftarrow EA_k(q)+(A_k(q)-A_0(q))/k$

else

$EA_{k+1}(q)\leftarrow EA_k(q)+(A_k(q)-A_{(k-n)}(q))/n$

{计算安全值 α}

$error_{(k)}(q)\leftarrow t-EA_k(q)-delay_{(k)}(q)$

$delay_{(k+1)}(q)\leftarrow delay_{(k)}(q)+\gamma\cdot error_{(k)}(q)$

$var_{(k+1)}(q)\leftarrow var_{(k)}(q)+\gamma\cdot(|error_{(k)}|-var_{(k)}(q))$

$\alpha_{(k+1)}(q)\leftarrow\beta\cdot delay_{(k+1)}(q)+\varphi\cdot var_{(k+1)}(q)$

{设置下一个消息检测点}

$\tau_{(k+1)}(q)\leftarrow EA_{k+1}(q)+\alpha_{(k+1)}(q)$

if ($q\in SUSPECTED_p$) **then**

$SUSPECTED_p\leftarrow SUSPECTED_p-\{q\}$

$s=s+1$ {增加误判次数}

$P_e=(s/k)+1)$

$\Delta_p(q)=(P_e+1)\Delta$ {二次检测时间}

Task3:

$\tau_{k+1}(q)$=当前时间

{到检测点没有收到 q 的消息更新}

等待$\Delta_p(q)$如果仍然没有收到 q 发送的消息:

$SUSPECTED_p\leftarrow SUSPECTED_p\cup\{q\}$

{怀疑进程 q}

4 算法满足的可靠性完整性

根据失效检测器能够同时满足的完整性和准确性级别可以将失效检测器分为八类。本文的算法满足强完整性和最终强准确性[4]。

性质 1. 算法满足强完整性，即每个失效的进程 q 最终会被所有正确的进程永久怀疑。$\exists t_0$: $\forall t>t_0$, $\forall_p\in correct(t), \forall_q\in crashed, q\in suspectedp(q)$

假设进程 q 发送第 i 条消息 $m_i(q)$后在 tcrash 时刻失效，t_{si}为第 i 条消息的发送时间，$t_{crash}>t_{si}$且$\nexists m_k(q)|t_{sk}>t_{crash}$。

任意一个正确的进程 p 在 $t_{rp}(q)$时刻收到最后一个由 q 发送的消息，并计算 q 下一条消息的失效检测点 $\tau_p(q)$和二次检测时间$\Delta_P(q)$，设 $T_p=\tau_p(q)+\Delta_P(q)$，已知 q 失效后不会向 p 发送心跳消息，所以 p 在 T_p 时刻不会收到 q 的心跳消息更新，即在时刻 T_p 进程 p 认为进程 q 失效，算法中 $\tau_p(q)$和$\Delta_P(q)$都是有界的，因此 T_p 也是有界的。

同理，所有其它正确的进程也会产生这样一个时刻 T_{fail}。对于所有正确的进程 T_{fail} 有界，则必然存在最大值 T_{fmax},即在 T_{fmax} 时刻后，所有正确的进程都认为 q 已经失效了，即：$\exists T_{fmax}:\forall t\ge T_{fmax},\forall p\in correct(t),q\in suspected_p(q)$。

性质 2. 算法满足最终强准确性，存在一个时刻，在这个时刻后任意一个正确的进程不会被其他正确的进程怀疑。即：$\exists t_{bound}$, $\forall t\ge t_{bound}$, $\forall p,q\in correct(t)$, $q\notin suspect_p(t)$。

由算法可知，假设 q 的心跳消息超时到达，那么所有正确的进程 $p\in\Pi-\{q\}$将动态增加检测时间，以适应与进程 q 之间的网络变化。随着时间的推移，检测时间不断增加，直到某个时刻 t 之后，p 会在检测时间内收到进程 q 发送的心跳消息，因此，$\exists t_{bound}|\forall t\ge t_{bound}$, $\Delta_{msg}\le\Delta_p(q)$。其中$\Delta_{msg}$为实际消息的延迟，即：

$\exists t_{bound}|\forall t\ge t_{bound}, \tau_k(q)+\Delta_p(q)>t_{sk}(q)+\Delta_{msg}(q)$

τ_k和 $t_{sk}(q)$分别为进程 q 第 k 条消息的检测时间点和发送时间。

5 实验及分析

本文主要针对网络延迟和丢包两种情形，对基于失效检测点的失效检测第二层检测时间作了相应的改进，因此测试主要侧重于验证本文引入的二次检测时间能否有效的降低错误率同时保证失效检测的效率。

整个测试的框架使用 Java 实现，分别对 Chen、Bertier 和本文提出的算法在相同网络丢包的情况下计算各算法的错误率。同时统计当被检测进程真正失效时，比较后两种算法的检测时间。

算法的预设参数为：心跳消息发送时间间隔Δ=1000ms，丢包率 P_L 从 0.05 变化到 0.1，网络延迟 D=200ms，α=10ms，γ=0.1，β=1，φ=2。果如图 1 所示：

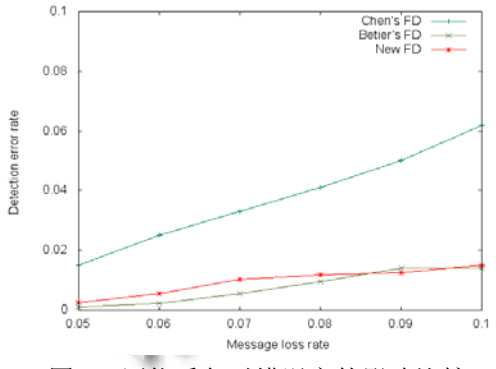

图 1　网络丢包对错误率的影响比较

由图可以看出 Bertier 的算法和本文提出的算法的错误率都明显小于 0.02，且随着丢包率的增长错误率的增长幅度较小，而 Chen 的算法由于没有采用二次检测当丢包率为 0.1 时错误率达到 0.06 以上，且随着丢包率的增长错误率的增长幅度较大。可见增加二次检测时间可以有效地降低错误率，同时也证明了本文所设置的二次检测时间符合实际的情况。

为比较 Bertier 和本文的算法在节点真正失效时的检测时间，每整百条消息后人为设置某节点失效，统计心跳停发的时间点到接收端怀疑该进程的时间间隔，由此作为失效算法的检测反应时间。试验结果如表 1 所示：

表 1　失效检测的反应时间

消息数 / 时间(ms)	100	200	300	400
Bertier's FD	3224	4219	5213	6228
Our FD	2250	2235	2242	2237

由表 1 可以看出，随着发送消息数量的增加和错误次数的增多，Bertier 算法的检测时间明显大于本文提出的算法，且随着消息数量的增多和误判次数的增加有明显增长趋势(每误判一次增加 1000ms 的二次等待时间)，也就是说当被检测节点真正失效时，Bertier 的算法夸大了延迟时间，导致在节点真正失效时不能及时的发现。而本文提出的算法检测时间较短且长时间内检测时间增长趋势平稳，因而在检测效率上优于 Bertier 的失效检测算法。

综上实验可以得出结论，本文提出的失效检测算法在错误率和检测时间上作了较好的权衡，既有效地减少了由于网络延迟和网络丢包所造成的错误判断次数，又较好地控制了失效检测时间。

6　结语

本文在有关自适应失效检测研究基础上，针对分布式网络环境中网络延迟和网络丢包两种情形，给出了基于失效检测点的二次检测时间估计值的改进。在计算失效检测点时引入缓存机制减少计算量，同时将错误率作为二次检测的等待时间的参数，使二次等待的时间更合理。理论分析和实验结果都证实了该等待时间引入有效地降低了失效检测算法的错误率，并且较好地控制了失效检测的时间。

参考文献

1 Bertier M, Marin O, Sens P. Implementation and performance evaluation of an adaptable failure detector. IEEE Conference on Dependable Systems and Networks, 2002:354–363.

2 Chandra TD, Toueg S, Hadzilacos V. Unreliable failure detectors for reliable distributed system. Journal of the ACM, 1996,(4):685–722.

3 Chen W, Toueg S, Aguilera MK. On the quality of service of failure detectors. IEEE Trans. on Computers, 2002,51(5):561–580.

4 Fetzer C. Perfect failure detection in timed asynchronous systems. IEEE Trans. on Computers, 2003,52(2):99–112.

5 Jacobson V, Karels MJ. Congestion Avoidance and Control. Proc. of the ACM Symposium on Communications, Architectures and Protocols, 1988: 314–329.

6 Zhou JL, Guang Y, Dong LJ, Gang L. Implementation and performance evaluation of an adaptable failure detector for distributed system. International Conference on Computational Intelligence and Security, 2007: 266–270.

7 石磊,侯垚.基于消息延迟预测的自适应失效检测模型.计算机应用,2010,(5):1312–1315.

8 云晓春,余翔湛.基于确认度失效检测算法的研究与设计.北京邮电大学学报,2005,(3):10–13.