ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.17, No.11, November 2006, pp.2362−2372
DOI: 10.1360/jos172362

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 一种基于 QoS 的自适应网格失效检测器*

董 剑+, 左德承, 刘宏伟, 杨孝宗

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

## An Adaptive Failure Detector for Grid Based on QoS

DONG Jian+, ZUO De-Cheng, LIU Hong-Wei, YANG Xiao-Zong

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

+ Corresponding author: Phn: +86-451-86413555 ext 804, Fax: +86-451-86414093, E-mail: dj@ftcl.hit.edu.cn, http://www.hit.edu.cn

**Dong J, Zuo DC, Liu HW, Yang XZ. An adaptive failure detector for grid based on QoS. *Journal of Software*, 2006,17(11):2362−2372.** http://www.jos.org.cn/1000-9825/17/2362.htm

**Abstract**: Failure detector is one of the fundamental building blocks to build reliable grid environments. There are large numbers of distributed applications with different QoS of failure detection in grids. Thus, in order to keep its efficiency and scalability in grid environments, a failure detector should not only provide QoS of accurate failure detection for applications, but also avoid redundant loads of designing multiple detectors for different QoS. Therefore, a new failure detector GA-FD (adaptive failure detector for grid) is presented, which adopts heartbeat detection strategy based on PULL mode. GA-FD can provide QoS of failure detection for multi-applications according to quantitative QoS metrics $(T_D^U, T_{MR}^L, T_M^U)$, and does not need any hypothesis about message behavior and clock synchronization. In addition, it proves that GA-FD implements a failure detector that belongs to $\diamond P$ in the partially synchronous model, and the experimental results are given in the end.

**Key words**: grid; failure detector; QoS; adaptive; heartbeat

摘 要: 失效检测器是构建可靠的网格计算环境所必需的基础组件之一.由于网格中存在大量对失效检测有着不同 QoS 需求的分布式应用,对于一个网格失效检测器来说,为保持其有效性和可扩展性,应该既能够准确提供应用程序所需的失效检测 QoS,又能够避免为满足不同 QoS 而设计多套失效检测器所产生的多余负载.基于 QoS 基本评价指标 $(T_D^U, T_{MR}^L, T_M^U)$,采用 PULL 模式主动检测策略实现了一种新的失效检测器——GA-FD (adaptive failure detector for grid),可以同时支持多个应用程序定量描述的 QoS 需求,不需要关于消息行为和时钟同步的任何假设.同时,证明了 GA-FD 在部分同步模型下可实现一个 $\diamond P$ 类的失效检测器,并给出了相应的实验及数据.

关键词: 网格;失效检测器;QoS;自适应;心跳
中图法分类号: TP393 文献标识码: A

* Supported by the Defense Pre-Research Project of the 'Tenth Five-Year-Plan' of China under Grant No.41312.1.2 (国家"十五"国防预研项目); the Defense Pre-Research Foundation of China under Grant No.514160401HT0151 (国防预研基金项目)
Received 2006-06-06; Accepted 2006-08-07

失效检测器是构建可靠的网格计算环境所必需的基础组件之一[1,2].众所周知,在存在失效进程的异步分布式系统中,由于无法正确区分失效进程和反应很慢的进程,导致很多重要的基础性问题(例如一致性问题)无法解决[3].失效检测器作为一种增强异步系统计算模型的有效途径,最早由 Chandra 和 Toueg[4]提出,并给出了形式化定义,目前已经广泛应用到了网格计算[1,2,5]、集群管理[6]、通信协议[7]等多个相关领域之中.

在网格这样的大规模分布式系统中,结构复杂,节点较多且分布较广,失效检测器作为基础组件,必须充分考虑其有效性(efficiency)和可扩展性(scalability)[2,8].Chen[9]提出了一个定量的 QoS 评价体系,可用来衡量失效检测器的有效性(检测速度和准确性).而可扩展性要求检测器在节点数量较大且不断变化的情况下能够保持稳定的低负载.但是,目前针对大规模分布式系统提出的失效检测算法,大部分都只是通过改善传统的检测结构,采用分层[10,11]或流言(gossip)[8,12]等方法来降低节点数目对检测负载的影响,很少有算法考虑应用程序对失效检测服务的 QoS 需求.另一方面,基于 QoS 提出的一些优秀的自适应失效检测算法,如 Chen[9],Bertier[13]及 Chen[14]等人提出的一些算法,可以按照某一特定 QoS,根据网络状况调节检测器的参数,使检测结果能够满足要求.但是,在网格环境下存在大量不同 QoS 需求的应用程序,考虑到负载,不可能为每一个应用程序定制单独的失效检测器.因此,如何将失效检测器设计成一个通用的网格组件,使其可以同时满足多个应用程序的 QoS 需求,是十分重要的.

Hayashibara 等人在这一方面作出了一些有益的尝试,提出了一种$\varphi$-失效检测器[15,16],将失效监测权与解释权相分离,输出一个随时间变化的$\varphi$值,代表对被检测进程的怀疑程度.应用程序可以根据自己的 QoS 需求对其作出不同的解释,这样,不同应用程序可以从同一个检测模块中获得自己需要的不同检测结果.但是,$\varphi$-检测器的实现,只能使应用程序获得一个定性的 QoS 描述,例如检测速度较快、准确性较低等.而我们在设计应用程序时,更加希望能够使用一种定量的度量体系来设定其对失效检测服务的 QoS 需求.为此,本文提出了一种新的失效检测器——GA-FD(adaptive failure detector for grid),它可以自适应于网格多变的网络条件,按照 Chen 提出的 QoS 基本评价指标 ($T_D^U$,$T_{MR}^L$,$T_M^U$) 同时支持多个应用程序定量描述的 QoS 需求.在实现中,根据网格计算环境的特点,采用了基于 PULL 模式的主动检测策略,并且不需要对消息行为和时钟同步进行任何假设.此外,我们证明了在部分同步模型下,GA-FD 实现了一个$\diamond P$类的失效检测器[4],满足解决一致性(及原子广播)等基础问题所需要的完整性和准确性要求.

## 1 相关工作

### 1.1 失效检测器模型

Chandra 和 Toueg 最早提出了失效检测器的形式化定义[4].假设一个由 $n$ 个进程组成的系统,$\Pi=\{p_1,p_2,\ldots,p_n\}$.假设系统存在一个独立的全局时钟,根据时钟报时信号得到的值域 $T$ 作为自然记数.将失效检测器定义为一个失效检测模块的集合,每一个模块对应于一个进程,其输出为一系列此进程所怀疑发生失效的进程集合.一个失效检测器的输出历史 $H$ 定义为 $\Pi\times T\mapsto 2^{\Pi}$,$H(p,t)$为进程 $p$ 的失效检测模块在 $t$ 时刻的输出,即应用程序在 $t$ 时刻“查询操作”所得到的答案.目前,大部分失效检测器的实现都依照这一模型.为了衡量检测器解决问题的能力,Chandra 和 Toueg 按照完整性和正确性将其分为 8 类,这里给出最终的最优失效检测器的集合(eventually perfect failure detector)$\diamond P$ 的定义.一个$\diamond P$类的失效检测器应满足:

强完整性(strong completeness):在任意一次运行中,每一个发生失效的进程最终都会被所有正确的进程永远判定为失效.

最终强准确性(eventual strong accuracy):在任意一次运行中,在某个时刻 $t$ 之后,每个正确的进程都不会被认为发生失效.

### 1.2 失效检测器的**QoS**

在实际系统中,应用程序主要从以下两方面评价一个失效检测器所提供的 QoS:

1) 失效检测器对真实发生的失效的检测速度,即一个失效检测器需要多少时间发现失效进程;

2) 失效检测器在一定的检测速度下发生错误输出的概率,即对一个正确进程产生怀疑的概率.

为了准确评价失效检测器的 QoS,Chen 等人提出了定量的 QoS 度量体系[9].我们用 $T$ 表示检测器认为进程处于正常状态,$S$ 表示检测器认为进程处于失效状态,则 $T$-transition 表示检测器的输出由 $S$ 变为 $T$,$S$-transition 表示检测器的输出由 $T$ 变为 $S$.下面 3 个基本评价指标可以定量地描绘一个失效检测器的 QoS.

- 检测时间($T_D$):是指从进程发生失效的时刻到它开始被怀疑的时间,即到发生 $S$-transition 之间的时间.这一指标保障失效检测器输出的完整性要求.
- 错误间隔时间($T_{MR}$):两次发生错误输出之间的时间间隔的度量,即连续两个 $S$-transition 之间的时间间隔,描述失效检测的准确度.
- 错误持续时间($T_M$):检测器修正一次错误怀疑所需要的时间,即从 $S$-transition 到下一个 $T$-transition 之间的时间间隔,描述失效检测的准确度.

以这 3 个基本指标为基础,可以唯一地确定其他 QoS 评价指标,如平均错误发生概率($\lambda_M$)、查询准确率($P_A$)等.

### 1.3 自适应失效检测器

作为一个服务组件,应用程序的 QoS 需求是设计失效检测器的主要依据.目前已有的检测器大部分采用了基于超时的心跳检测策略[15,16],它通过周期性地互相发送检测消息来检测对方的状态.在这种机制下,心跳发送周期$\Delta t_i$和检测超时值$\Delta t_o$可以决定检测器的行为,但是随着网络状况(如流量、传输延迟等)的不断变化,却无法确保应用程序得到的 QoS 能够一直满足要求[9,14].因此,为满足应用程序的需要,提出了自适应失效检测器[9,13−20],它通过自动调节参数$\Delta t_i$和$\Delta t_o$来适应不断变化的网络状况.最初,这类调节都是依据对心跳检测消息到达时刻的预测对超时值$\Delta t_o$进行调整[17],在检测延迟和准确性之间作一个简单的权衡.Falai[21]等人研究了 LAST、MEAN、线性序列等常用预测方法在检测器中的应用,并给出了各种方法的性能比较.Chen[9]在提出 QoS 的度量体系之后,基于概率网络模型提出了一系列自适应算法,实现了 QoS 对检测器参数调整的定量控制,在不同的网络状况下,可以用最小的检测负载得到所需要的 QoS.Bertier[13]结合 Jacobson 预测算法对 Chen 的算法作了进一步的改进,使参数的选择更加准确.此外,还产生了一些借鉴其他领域知识的算法,例如 Tian[18]基于模糊分类系统设计的一个自适应检测器.

上述算法从不同途径获得了自适应于网络变化的失效检测器.但值得注意的是,这些算法都是针对某一具体的 QoS 进行其自适应的调节,而在网格这样的大规模分布式系统中,同时运行着大量不同的分布式应用,很显然,它们不可能对失效检测服务具有同样的 QoS 需求.无论采用上述的哪一种算法,都需要为每一种 QoS 需求单独配置一套失效检测器.作为网格系统中使用非常频繁的基础组件,数目如此众多的失效检测器将产生大量的检测负载.为此,一个自适应网格失效检测器在能够适应不断变化的网络环境的同时,还应该能够适应不同应用程序的 QoS 需求.Hayashibara 等人提出的$\varphi$-检测器[15,19]在这方面作出了有益的尝试.

### 1.4 $\varphi$-检测器

Hayashibara 提出的$\varphi$-检测器对其检测的每一个进程输出一个相关的值$\varphi$,用来表示对该进程是否发生失效的可信度,而不再像传统的检测器那样简单地输出怀疑还是信任这样的二值性的结果.$\varphi$-检测器的实现采用了基于 PUSH 模型的心跳检测策略.被检测进程 $p$ 按照一个时间间隔 $\Delta_i$ 周期性地向它的检测进程 $q$ 发送检测消息.令 $T_{last}$ 为最近收到的消息的到达时间.系统假设消息到达间隔的分布概率 $P(\Delta<t)$符合正态分布,这样,假设当前时间为 $t_{now}$,则对应的$\varphi(t_{now})=-\log_{10}(P(\Delta<(t_{now}-t_{last})))$.应用程序设定一个相应的可信度阈值$\Phi$,如果$\varphi\geq\Phi$,则 $q$ 开始怀疑 $p$.这样,一个较低的阈值就表示应用程序对检测速度要求较高,而一个较高的阈值就表示对检测准确性要求较高.由此,一个$\varphi$-检测器可以为多个应用程序提供检测服务.但是,$\varphi$-检测器仍存在一些较为严重的缺点:

1) 在$\varphi$-检测器的实现中,应用程序设定的可信度阈值$\Phi$只能定性地刻画其 QoS 要求,如设定一个稍小的$\Phi$,只能表示其对检测速度要求较高、对准确率要求较低等定性的表述.但是,实际应用中的大多数分布式应用程序都存在一些较为严格的时间上的约束[9],因此,更加需要失效检测器可以按照 QoS 评价

指标支持准确的定量的 QoS 需求.

2) 在基于心跳的失效检测器中,一个检测消息 $m_i$ 对失效检测结果的影响仅与其自身检测延迟时间相关[9].而$\varphi$-检测器利用检测消息的到达时间间隔$\Delta$作为计算$\varphi$值的主要依据,这样虽然不需要同步时钟,但是根据 $m_i$ 的延迟计算出的$\varphi$值却依赖于 $m_{i-1}$ 的到达时间 $t_{last}$.这使得检测结果变得不可靠;

3) $\varphi$-检测器的实现,需要假设消息行为符合正态分布.在网格这样复杂的大规模分布式系统中,存在更高级别的异步性、较长的传输延时、较高的消息丢失率,而且其基础结构是动态可配置的,大量的组件及其分布也是动态的[20].在这样的环境下,消息行为不可能一直符合某种特定分布的特性.因此,作为一个通用组件来设计的失效检测器不应该对此作出任何假设.在本文最后的实验中验证了这一问题.

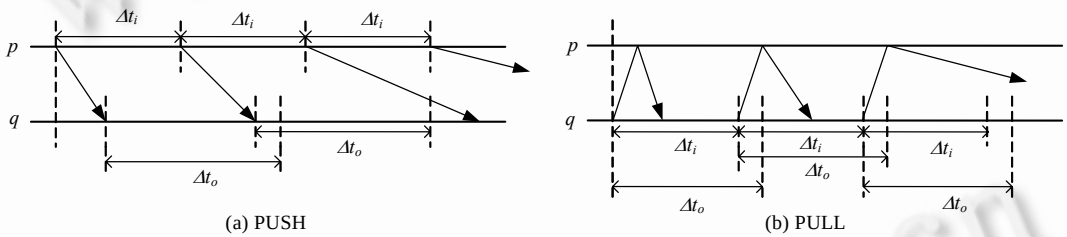
针对上述问题,本文提出的 GA-FD 失效检测器提供了一个很好的解决方案.

## 2 GA-FD 失效检测器

为了简化描述,我们先考虑一个只包含两个进程的系统,进程 $q$ 检测进程 $p$.故障模型与 Chandra 一样采用 fail-stop 模型[4].进程之间的链路为 fair-lossy 链路[22],即链路只允许丢失有限个消息,假设消息的丢失率为 $p_L$.进程之间不存在同步时钟.

### 2.1 心跳策略

心跳是失效检测器最为常用的实现技术,按照其实现方式的不同,可以分为 PUSH 和 PULL 两种方式[16],如图 1 所示.




(a) PUSH (b) PULL

Fig.1 The strategy of heartbeat detection

图 1 心跳检测策略

二者都是通过周期性地发送检测消息来检测对方的状态(是否已经发生失效).所不同的是:PUSH 方式是被检测者主动向它的检测者周期性地发送“I am alive!”消息;而 PULL 方式则是由检测者向被检测者主动发送查询信息“Are you alive?”,而被检测者收到查询后,被动地发回应答消息“I am alive!”.对于传统的基于超时机制的失效检测器来讲,一般需要设置一个相应的超时值$\Delta t_o$,当超过此时间仍未收到预计的消息时,就认为被检测进程已经发生失效,检测器输出为失效状态 $S$.很显然,PULL 方式与 PUSH 方式相比,得到同样的性能需要两倍数量的消息,但这并不影响其可扩展性(增加的倍数为常数,与节点数目 $n$ 无关).然而,PULL 方式是一种主动检测方式,可以只在需要的时候才发起检测,这更适合网格计算的特点,需要经常地改变检测关系,比如一个任务完成,或者网络结构重新配置后.而且,使用 PULL 方式可以保证每个检测消息及其延时对检测结果的影响是独立的.因此,在 GA-FD 中采用基于 PULL 模式的检测策略.

### 2.2 GA-FD失效检测器

在应用程序查询时,GA-FD 根据当前的网络状态,为每一个被检测进程输出一个 0~1 之间的实数$\rho$.$\rho$的计算方法如下:

$$\rho=\begin{cases}1-\dfrac{\delta^2}{(t_{current}-sn_{current}\cdot\Delta t_i-u)^2+\delta^2}, & \text{if} \quad (t_{current}-sn_{current}\cdot\Delta t_i)>u,\\ 0, & \text{else}\end{cases}$$

其中,$t_{current}$为应用程序查询时的当前时刻,$sn_{current}$为当前未收到应答的消息的最小序号.$u$和$\delta^2$为检测时间$D$(即从检测消息发出到收到应答所消耗的时间)的分布期望和方差.$u$和$\delta^2$可用检测时间$D$的样本期望$E(D)$和方差$V(D)$来获得.为此,需要建立一个固定尺寸($w$)的滑动窗口$W_D$,保存最近收到应答的$w$个查询消息的检测时间,为计算$E(D)$和$V(D)$建立样本.完整的GA-FD失效检测器算法所下:

```
1.  For process q:
2.  detector_module:
3.      for all i>0, at time i·Δt_i, send mq_i to p;
4.      upon receive ma_i from p do
5.          add (t_current−i·Δt_i) to W_D;
6.          sn_current←i+1;
7.  query_module:
8.      T_e←t_current−sn_current·Δt_i;
9.      If T_e>E(D) do
10.         ρ ← 1 − V(D) / ((T_e − E(D))^2 + V(D));
11.     else do
12.         ρ←0;
13. For process p:
14.     upon receive mq_i from q do
15.         send ma_i to q;
```

GA-FD可分为检测模块和查询模块两部分.检测模块负责按照发送间隔$\Delta t_i$周期性地向被检测进程$p$发送检测消息$mq_i$($i$为检测消息的序号),$p$在收到该消息后发回应答消息$ma_i$,以表明自己处于正常状态.进程$q$收到应答之后,首先将消息$mq_i$的检测时间($t_{current}-i\cdot\Delta t_i$)保存到滑动窗口$W_D$中,并同时删除$W_D$中序号最小的消息的对应项.在应用程序对某一被检测进程进行查询时,查询模块根据当前的网络状态($E(D)$和$V(D)$)计算出$\rho$值,返回给应用程序.应用程序会根据自己对QoS的需求设定一个阈值$P$,当$\rho>P$时,则应用程序会认为进程$p$已经失效.由此可见,GA-FD既可适应不同QoS需求的应用程序,又可自适应于网络状况的变化.

### 2.3 计算阈值$P$

阈值$P$的选择直接影响到检测结果是否满足应用程序的需求,其值由QoS中的准确性的基本度量指标$T_M$和$T_{MR}$来确定.下面通过定理1给出其计算方法.

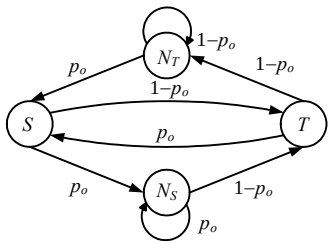
**定理1**. 对于准确性评价指标为$(T_{MR}^L, T_M^U)$的一个应用程序,令$P=\mathrm{Max}\left(\frac{1+\sqrt{1-4\Delta t_i/T_{MR}^L}}{2(1-p_L)}, \frac{\Delta t_i}{T_M^U(1-p_L)}\right)$($p_L$为消息丢失率),则按照条件$\rho>P$查询GA-FD所得到的检测结果满足其准确性指标要求.

$T_{MR}^L$表示平均错误间隔的下界,$T_M^U$代表错误平均持续时间的上界.令$E(T_{MR})$和$E(T_M)$分别为$T_{MR}$和$T_M$的期望,则有$E(T_{MR})\geq T_{MR}^L$和$E(T_M)\leq T_M^U$成立.对于如图1(b)所示的PULL方式的失效检测策略,假设$p$一直保持正确.采用超时机制,对于检测消息$mq_i$,在($i\cdot\Delta t_i+\Delta t_o$)时刻,$q$没有收到$p$的应答消息$ma_i$的概率为$p_o=(1-p_L)P(D>\Delta t_o)+p_L$.为了证明定理1,首先给出下面两个引理.

引理**1**. $E(T_{MR})=\Delta t_i/p_S$,其中,$p_S=(1-p_L)P(D<\Delta t_o)\cdot[(1-p_L)P(D>\Delta t_o)+p_L]$.

证明:设$\{X_i, i>0\}$为一个随机序列,其中,随机变量$X_i$表示($i\cdot\Delta t_i+\Delta t_o$)时刻检测器的输出状态.对于PULL型检测器,状态空间可定义为$G=\{S,T,N_S,N_T\}$,即$S$-transition,$T$-transition,保持$S$状态,保持$T$状态.从图1(b)中可以看出:变量$X_i(i>0)$的取值是由消息$ma_i$与$X_{i-1}$的值共同决定的,并且与$X_j(0<j<i-1)$无关,即$P(X_i|X_{i-1})=P(X_i|X_{i-1},X_{i-2},\ldots,X_1)$.由此可知,随机序列$\{X_i, i>0\}$是一条有限状态马尔可夫链,其状态转移图及转移矩阵如图2所示.

由于 $0<p_o<1$,从转移矩阵 $P^{(n)}$可知,$\{X_i,i\geq 0\}$是一个常返的马尔可夫链,且状态 $S$ 是遍历的.对于状态 $S$,存在 $\lim_{n\to\infty} p_{ii}^{(n)} = p_o(1-p_o)>0$,则从状态 $S$ 出发再回到 $S$ 的平均回转时间 $u_S=1/p_o(1-p_o)$,考虑到检测器系统中发生一次状态转移所需时间为$\Delta t_i$,则有 $E(T_{MR})=\Delta t_i/p_S$. □




$$P^{(n)}=\begin{pmatrix} p_o(1-p_o) & p_o(1-p_o) & p_o^2 & (1-p_o)^2 \\ p_o(1-p_o) & p_o(1-p_o) & p_o^2 & (1-p_o)^2 \\ p_o(1-p_o) & p_o(1-p_o) & p_o^2 & (1-p_o)^2 \\ p_o(1-p_o) & p_o(1-p_o) & p_o^2 & (1-p_o)^2 \end{pmatrix}$$

Fig.2 The transition diagram of state and its transition probability matrix ($n$>2)

图 2 状态转移图及其转移概率矩阵($n$>2)

引理 **2**. $E(T_M)\leq \dfrac{\Delta t_i}{(1-p_L)P(D<\Delta t_o)}$.

证明:为了计算 $E(T_M)$,需要先计算查询正确率 $P_A$,即在任意时刻 $t$,失效检测结果保持正确的概率,有

$$P_A=1-E(T_M)/E(T_{MR}).$$

假设 $t\in[(i-1)\Delta t_i+\Delta t_o, i\Delta t_i+\Delta t_o]$,符合均匀分布.此区间上的查询正确率为

$$P_A^{(i)}=1-\int_{(i-1)\Delta t_i+\Delta t_o}^{i\Delta t_i+\Delta t_o}\frac{1}{\Delta t_i}(p_L+(1-p_L)P(D>\Delta t_o+x-((i-1)\Delta t_i+\Delta t_o))\mathrm{d}x$$

$$=1-\frac{1}{\Delta t_i}\int_0^{\Delta t_i}(p_L+(1-p_L)P(D>\Delta t_o+x))\mathrm{d}x.$$

可以看出,上式中, $P_A^{(i)}$ 与区间无关.因此,有 $P_A=P_A^{(i)}$.

令 $p(x)=p_L+(1-p_L)P(D>\Delta t_o+x)$,则有

$$E(T_M)=\frac{\int_0^{\Delta t_i}p(x)\mathrm{d}x}{p_S}\leq\frac{\int_0^{\Delta t_i}p(0)\mathrm{d}x}{p_S}=\frac{\Delta t_i}{(1-p_L)P(D<\Delta t_o)}.$$ □

下面来完成对定理 1 的证明.

证明:根据条件 $E(T_{MR})\geq T_{MR}^L$ 和 $E(T_M)\leq T_M^U$,由引理 1 和引理 2 可以得到:

$$\begin{cases}\dfrac{\Delta t_i}{(1-p_L)P(D<\Delta t_o)\cdot[(1-p_L)P(D>\Delta t_o)+p_L]}\geq T_{MR}^L \\ \dfrac{\Delta t_i}{(1-p_L)P(D<\Delta t_o)}\leq T_M^U\end{cases}.$$

求解,可得:

$$\begin{cases}P(D<\Delta t_o)\geq\dfrac{1+\sqrt{1-4\Delta t_i/T_{MR}^L}}{2(1-p_L)} \\ P(D<\Delta t_o)\geq\dfrac{\Delta t_i}{T_M^U(1-p_L)}\end{cases}.$$

一般认为,$1-4\Delta t_i/T_{MR}^L\geq 0$ 是成立的.若不成立,则每 4 个检测周期就要发生一次错误,失效检测器错误发生率太高,这对实际应用程序来讲已经没有意义.

再来看$\rho$的计算.根据单边切比雪夫(one-sided Chebyshev)不等式,

对$\forall t>E(D)$,存在 $P(D<t)\geq 1-\frac{V(D)}{(t-E(D))^2+V(D)}$.

在某一任意查询时刻 $t_{current}$,令 $t_{iqu}=t_{current}-sn_{current}\cdot\Delta t_i$.根据 GA-FD 的算法,有$\rho\leq P(D<t_{iqu})$.对于

$$P=\mathrm{Max}\left(\frac{1+\sqrt{1-4\Delta t_i/T_{MR}^L}}{2(1-p_L)},\ \frac{\Delta t_i}{T_M^U(1-p_L)}\right),$$

当$\rho>P$ 时,判断被检测对象发生失效(此时$\Delta t_o=t_{iqu}$),可使上面的方程组成立.即按照条件$\rho>P$ 进行检测,失效检测器输出结果可满足准确性评价指标 $(T_{MR}^L,T_M^U)$ 的要求. □

### 2.4 关于 $T_D^U$

GA-FD 的输出,最终需要应用程序按照其 QoS 基本评价指标来解释,即 $(T_D^U,T_{MR}^L,T_M^U)$.其中,$T_D^U$ 是检测时间的上界,限定了检测速度,保证了失效检测的完整性级别,是一个十分重要的评价指标.因此,一个应用程序在某一时刻 $t_{current}$ 判定 $q$ 发生失效的条件应该为 $(\rho>P)\vee(T_e+\Delta t_i>T_D^U)$.

$\rho>P$ 可保证检测结果满足 $T_{MR}^L$ 和 $T_M^U$ 限定的准确性要求,$T_e$ 记录了序号为 $sn_{current}$ 的检测消息从发送到当前查询时刻所花费的时间,可提供给应用程序关于检测所消耗时间的信息.完整性和准确性是两个互相矛盾的属性.因此,对于一组给定的 QoS 需求,在一定的网络状态下,可能存在无法同时满足的情况.例如,在某一时刻,$\rho\leq P$,而 $T_e+\Delta t_i$ 已经超过了 $T_D^U$,此时的输出将无法满足要求.在 Chen 和 Bertier 等人提出的自适应检测算法中,通过根据网络状况的变化动态地调整检测间隔$\Delta t_i$ 来满足 QoS 需求.但在其算法中,也同样不可能在任何网络条件下满足所有 QoS 需求.而且,文献[23]指出:失效检测器能取得的 QoS 受$\Delta t_i$ 的影响并不大,而主要是由网络系统的基本特性(如消息平均传送时间$\Delta t_{tr}$)决定的.因此,在实际系统中,GA-FD 的$\Delta t_i$ 可以根据检测双方之间的网络特性来选定一个值,不需要根据应用程序的不同 QoS 而采用多个$\Delta t_i$,设计多个检测器.在第 4 节,我们将通过实验对两种做法进行比较.

## 3 算法证明

在这一节,按照 Chandra 和 Toueg 对失效检测器的分类,我们将证明在部分同步模型下,GA-FD 为应用程序所提供的失效检测服务可以等价于一个$\diamond P$ 类失效检测器.

Chandra 和 Toueg 所描述的部分同步模型[4]是一种常用的、与实际系统更为相似的分布式系统模型.在这个模型中,假设在到达某个未知的时间,即全局稳定时间(global stabilization time,简称 GST)之后,进程的处理速度和消息的传输时间都是有界的,但界限是未知的.

为了证明上述结论,首先引入几个引理.

引理 **3(strong completeness)**. 每一个发生失效的进程最终都会被所有正确的应用程序永远判定为失效.

证明:假设在一个正确的进程 $q$ 使用 GA-FD 来检测进程 $p$.需要证明:如果 $p$ 发生失效,那么存在某一时刻 $t$,对于$\forall t_{iqu}>t$,一个应用程序 $A$ 查询 GA-FD 的结果都将判定 $p$ 失效.

在 $p$ 失效之后,将不会对任何检测消息作出应答,也就是说,进程 $q$ 此后将收不到任何消息.因此,$E(D)$和 $V(D)$将不会发生变化,从图 2 中的算法来看(9~12 行),随着 $T_e$ 的增长,$\rho$是非减的,且当 $T_e>E(D)$时,$\rho$将保持严格递增.假设 $ma_i$ 是 $q$ 在时刻 $t_{last}$ 从 $p$ 收到的最后一个应答消息,则 $sn_{current}$ 将会永远等于 $i+1$.此后,在任何查询时刻 $t_{current}>t_{last}$,都有 $T_e=t_{current}-(i+1)\cdot\Delta t_i$,随着 $t_{current}$ 的增长,$T_e$ 将严格单调递增.假设应用程序 $A$ 的检测阈值为 $P_A$,一定可以找到一个时刻 $t>t_{last}$,使得$\rho=P_A$.由前面的分析可知,此时$\rho$是递增的,所以,对于任意时刻 $t_{current}>t$,$\rho>P_A$ 永远成立.因此,对于应用程序 $A$,在任意时刻 $t_{current}>t$ 查询 GA-FD,都会将 $p$ 判定为失效进程. □

引理 **4(eventual strong accuracy)**. 在某个时刻 $t$ 之后,每个正确的进程都不会被错误地认为发生失效.

证明:按照引理 3 的证明进行同样的假设,需要证明:在某个时刻 $t$ 之后,应用程序 $A$ 将不会错误地认为 $p$ 发生失效.因为此属性具有最终(eventual)性,在证明中,可不考虑 $T_D^U$ 的限制.

假设 $mq_i$ 是 GST 之后 $q$ 发出的第 1 个检测消息,发送时刻为 $t_s$.在 GST 之后,进程的处理消息时间和消息的传递时间都有上界,分别记为 $\overline{\Delta t_p}$ 和 $\overline{\Delta t_{tr}}$.因此,对于任意检测消息 $ma_j$ ($j\geq i$),其检测时间 $T_{D_j}\leq \overline{\Delta t_p}+2\overline{\Delta t_{tr}}$.那么,对于任意的查询时刻 $t_{current}>t_s$,$T_e$ 存在一个上界.按照第 2.2 节中的算法,$\rho$是递增的,因此,$\rho$值也将存在一个上界,假设这个上界为$\rho_{\max}$($0<\rho_{\max}<1$).对于一个 QoS 需求为 $(T_D^U, T_{MR}^L, T_M^U)$ 的应用程序 $A$ 来讲,只要 $T_D^U > T_{D_j}$,那么,对于任意的查询时刻 $t_{current} > t_s+\overline{\Delta t_p}+2\overline{\Delta t_{tr}}$,$A$ 都不会怀疑进程 $p$.我们分两种情况来讨论:

1) $\rho_{\max}\leq P_A$

此时,对于 $\forall t_{current} > t_s+\overline{\Delta t_p}+2\overline{\Delta t_{tr}}$,$\rho\leq\rho_{\max}\leq P_A$ 成立,故 $A$ 不会怀疑 $p$;

2) $\rho_{\max}>P_A$

此时,修改 $A$ 的检测阈值 $P_A$ 为 $P_A'$,令其等于$\rho_{\max}$,则有 $P_A'>P_A$.我们知道,检测阈值越高,检测结果的准确性就越高,故按照 $P_A'$ 所得到的解释结果仍可满足原来的 QoS 准确性要求.而对于 $\forall t_{current} > t_s+\overline{\Delta t_p}+2\overline{\Delta t_{tr}}$,$\rho\leq\rho_{\max}= P_A'$ 成立,$A$ 不会怀疑 $p$.

综上所述,引理 4 得证. □

定理 **2**. 在部分同步系统中,使用 GA-FD 的应用程序可以得到等价于$\diamond P$ 类失效检测器的输出结果.

证明:根据引理 3 和引理 4,定理 2 得证. □

## 4 实验结果及分析

我们通过实验对 GA-FD 检测器的性能进行了验证.实验在位于北京和哈尔滨的两台计算机上进行.其中,位于北京的被测机是一台 Web 服务器,24 小时对外提供服务.检测周期$\Delta t_i$ 设定为 1 000ms.根据不同的需要,分别进行了以下实验:

1) 准确性指标 $(T_{MR}^L, T_M^U)$

这一实验主要是为了验证GA-FD的检测结果是否可以满足应用程序的准确性指标 $(T_{MR}^L, T_M^U)$ 的要求.因此,在实验中未考虑 $T_D^U$ 对检测结果的影响,令 $T_D^U$ =5000ms.对于实验中的每一组 $(T_{MR}^L, T_M^U)$ 的组合,均进行了上万次测试,其均值形成了如图 3 所示的数据.

从图中可以看出,GA-FD 的检测结果严格地符合准确性指标 $(T_{MR}^L, T_M^U)$ 的要求.同时可以看到,随着准确性要求的不断提高,两项指标交替地受到 $T_{MR}^L$ 和 $T_M^U$ 的影响.随着对 $T_{MR}^L$ 要求的不断提高,检测结果的 $T_{MR}$ 随之不断增大,而 $T_M$ 快速减小,且逐渐不再受 $T_M^U$ 的影响,检测准确率不断提高.同样可以看出,如果对 $T_M^U$ 的要求比较严格,即使 $T_{MR}^L$ 的要求较低,GA-FD 同样也会取得较高的 $T_{MR}$ 和较低的 $T_M$,从而保证较高的检测准确率.

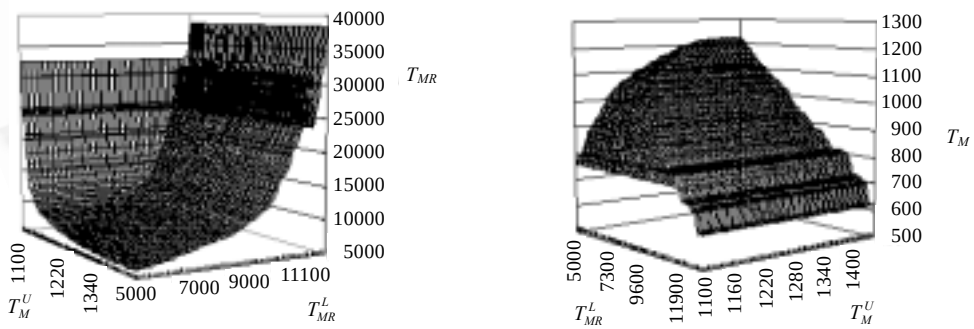



Fig.3 Accuracy metrics ($T_{MR}$,$T_M$) validation (ms)

图 3 准确性指标($T_{MR}$,$T_M$)验证(ms)

2) 对检测结果的影响

评价指标 $T_D^U$ 规定了检测器的检测延迟的上界,同时保障了检测结果的完整性要求.因此, $T_D^U$ 的取值会影响到检测结果的准确性.图 4 显示了 $T_D^U$ 的不同取值对检测结果的 $T_{MR}$ 和 $T_M$ 的影响,图中的每一个点均为同样条件下多组实验数据的平均值.可以看出,对于同样的准确性要求 ($T_{MR}^L$ =10000ms,$T_M^U$ =1100ms),$T_D^U$ 导致检测结果的准确性出现了较大的差别,当 $T_D^U$ 取值较小时,检测器需要在较短时间内作出判断,发生错误的概率会较大,这甚至可能使得结果可能无法符合 QoS 的准确性要求.图 4(b)给出了同样实验数据下的查询准确率变化趋势,可以更加直观地看出它对准确性的影响.然而,如图 4 所示,随着 $T_D^U$ 的不断增长,其对检测结果准确性的影响越来越小,存在某个点,在此之后,GA-FD 的检测结果将完全由 ($T_{MR}^L$,$T_M^U$) 来决定.

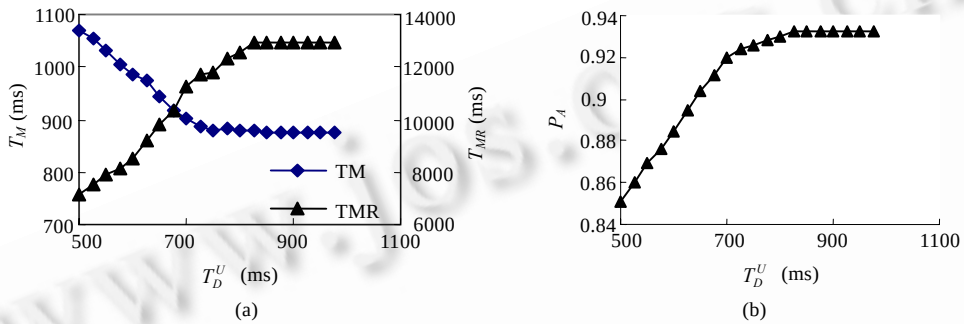



Fig.4 $T_D^U$ influence on accuracy metrics

图 4 $T_D^U$ 对准确性指标的影响

3) 与其他检测器的比较

为了检验 GA-FD 实际运行的性能,将其与另外一种有代表性的基于 QoS 的失效检测器(Chen 的 NFD-E 检测器)进行了实验对比.NFD-E 检测算法是 Chen 提出的一系列算法中性能最好的,而且与 GA-FD 的系统假设最为相近(没有对同步时钟和延迟分布的任何假设).对于$\varphi$-检测器,由于其$\varphi$值的选取完全是定性的,无法与其他检测器在同样的 QoS 配置下进行比较.但是,我们实现了一个基于正态分布假设的 GA-FD 版本,对关于检测延迟分布假设的有效性进行了比较.

为了提高实验的真实性,在上面的实验环境之外,增加了对一台位于美国的服务器的检测实验,结果如图 5 所示.通过考察检测时间和查询准确率 $P_A$ 的关系来比较上述 3 种算法的检测能力.

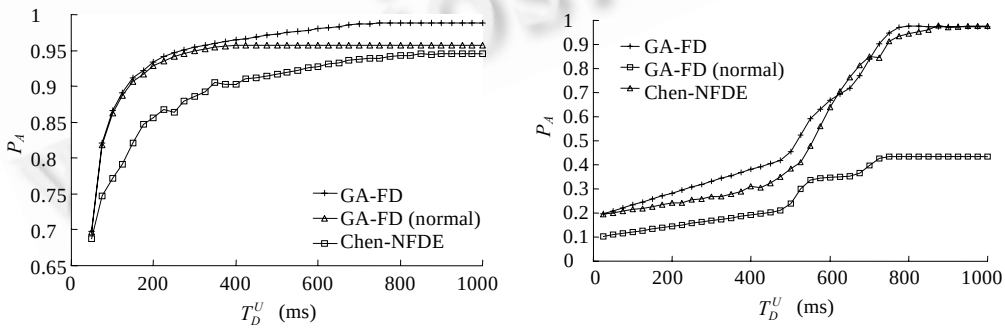



Fig.5 Comparison of GA-FD and others detectors

图 5 GA-FD 与其他检测器的比较

从图 5 中可以看出:GA-FD 虽然采用了静态检测周期,但是并未对性能产生较大的影响,检测速度和准确性

与 NFD-E 比较接近,且大部分情况下略优于对方; $T_D^U$ 增大到某个值之后,准确性逐渐不再受 $T_D^U$ 的影响,而能够符合 QoS 配置的准确性要求.图 5(a)和图 5(b)也同时表明,GA-FD 同样可以在不同的网络条件下,通过自身的调节实现同样的检测准确性.此外,从图中还可以看出,对于假设延迟分布符合正态分布的 GA-FD(normal)来讲,其性能受网络环境的影响较大.在图 5(a)中,实验涉及的网络环境较为简单,检测延迟比较小($E(D)$=86ms),消息丢失率非常低($p_L$=0.06%),其分布比较接近正态分布的假设.因此可以看到:与 GA-FD 相比,二者的检测结果相差很小,而且 GA-FD(normal)受 $T_D^U$ 的影响要小一些;但是,在网络情况较为复杂的情况下(如图 5(b)所示),检测延迟较大($E(D)$=630ms),且信息丢失率较高($p_L$=1.03%),此时可以明显看到,与其他两种算法相比,GA-FD(normal)检测结果的准确性要低很多.由此可知,对检测延迟分布进行假设并不适合于网格这样复杂、多变的网络环境.

## 5 结束语

网格这样复杂的大规模分布式系统为失效检测器的研究提出了新的挑战.本文提出的 GA-FD 网格失效检测器实现了一个部分同步模型下的 $\diamond P$ 类的失效检测器,可以作为网格系统的一个通用组件来使用.它可以自适应于网格多变的网络条件,可以按照 QoS 基本评价指标 ($T_D^U$, $T_{MR}^L$, $T_M^U$) 同时为多个分布式应用提供定量描述的失效检测 QoS.我们设计了相应的实验方案,对 GA-FD 的性能进行了验证和对比,结果表明,GA-FD 不仅符合设计要求,而且在同样的条件下具有比其他检测器更为优秀的检测能力.

在实际应用中,GA-FD 可以结合一些基于新的检测框架的可扩展性较好的算法来使用,例如,Stellin 的分层结构[11]或是 Horita 基于有向 Gossip 的算法[2],这样可以在节点之间通过某种方式共享检测结果,从而避免 all-to-all 的检测框架带来的指数级负载.此外,GA-FD 作为网格的一个基础组件,不仅可以为失效检测服务,其输出还可以支持任务调度、成员协议等常见分布式应用,这也是我们下一步的主要研究内容.

**References**:


[1] Jin H, Zou DQ, Chen HH, Sun JH, Wu S. Fault-Tolerant grid architecture and practice. Journal of Computer Science and Technology, 2003,18(4):423−433.

[2] Horita Y, Taura K, Chikayama T. A scalable and efficient self-organizing failure detector for grid applications. In: Katz DS, ed. Proc. of the 6th IEEE/ACM Int'l Workshop on Grid Computing. Washington: IEEE CS Press, 2005. 202−210.

[3] Fischer MJ, Lynch NA, Paterson MS. Impossibility of distributed consensus with one faulty process. Journal of the ACM, 1985, 32(2):374−382.

[4] Chandra TD, Toueg S. Unreliable failure detectors for reliable distributed systems. Journal of the ACM, 1996,43(2):225−267.

[5] Jain A, Shyamasundar RK. Failure detection and membership management in grid environments. In: Buyya R, ed. Proc. of the 5th IEEE/ACM Int'l Workshop on Grid Computing. Pittsburgh: IEEE Computer Society Press, 2004. 44−52.

[6] Zhang YH, Wang DS, Zheng WM. The automatic reconfiguration of COW. Acta Electronica Sinica, 2000,28(5):13−16 (in Chinese with English abstract).

[7] Braden R, ed. Requirements for internet hosts-communication layers. RFC 1122, 1989. http://www.ietf.org/rfc/rfc1122.txt

[8] Gupta I, Chandra TD, Goldszmidt GS. On scalable and efficient distributed failure detectors. In: Kshemkalyani A, Shavit N, eds. Proc. of the 20th Symp. on Principles of Distributed Computing (PODC 2001). New York: ACM Press, 2001. 170−179.

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

[10] Foster I, Kesselman C, Tuecke S. The anatomy of the grid. Int'l Journal of High Performance Computing Applications, 2001,15(3): 200−222.

[11] Stelling P, Foster I, Kesselman C, Lee C, von Laszewski G. A fault detection service for wide area distributed computations. In: Schmidt D, ed. Proc. of the 7th IEEE Symp. on High Performance Distributed Computing. Chicago: IEEE Computer Society Press, 1998. 268−278.

[12] van Renesse R, Minsky Y, Hayden M. A gossip-style failure detection service. In: Davies N, Raymond K, Seitz J, eds. Proc. of the IFIP Int'l Conf. on Distributed Systems Platforms and Open Distributed Processing Middleware. New York: Springer-Verlag, 1998. 55−70.

[13] Bertier M, Marin O, Sens P. Implementation and performance evaluation of an adaptable failure detector. In: Martin DC, ed. Proc. of the 15th Int'l Conf. on Dependable Systems and Networks. Bethesda: IEEE CS Press, 2002. 354−363.

[14] Chen NJ, Wei J, Yang B, Huang T. Adaptive failure detection in Web application server. Journal of Software, 2005,16(11): 1929−1938 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/1929.htm

[15] Hayashibara N, Défago X, Yared R, Katayama T. The $\varphi$-accrual failure detector. In: Titsworth FM, ed. IEEE Int'l Symp. on Reliable Distributed Systems (SRDS 2004). Florianopolis: IEEE Computer Society Press, 2004. 66−78.

[16] Sotoma I, Madeira ERM. Adaptation—Algorithms to adaptive fault monitoring and their implementation on CORBA. In: Blair G, Schmidt D, Tari Z, eds. Int'l Symp. on Distributed-objects and Applications (DOA 2001). Rome: IEEE Computer Society Press, 2001. 219−228.

[17] Fetaer C, Raynal M, Tronel F. An adaptive failure detection protocol. In: Williams AD, ed. Proc. of the 2001 Pacific Rim Int'l Symp. on Dependable Computing. Seoul: IEEE Computer Society Press, 2001. 146−153.

[18] Tian D, Chen SY, Li J. Novel adaptive failure detector for distributed systems. Journal of Harbin Institute of Technology, 2006, 38(Suppl.):374−377 (in Chinese with English abstract).

[19] Hayashibara N, Défago X, Katayama T. Two-Ways adaptive failure detection with the $\varphi$-failure detector. In: Fich FE, ed. Proc. of the Workshop on Adaptive Distributed Systems (WADiS). Sorrento: Springer-Verlag, 2003. 22−27.

[20] Shi XH, Jin H, Han ZF, Qiang WZ, Wu S, Zou DQ. ALTER: Adaptive failure detection services for grid. In: Cantarella JD, ed. Proc. of the IEEE Int'l Conf. on Services Computing. Orlando: IEEE CS Press, 2005. 355−358.

[21] Falai L, Bonvadalli A. Experimental evaluation of the QoS of failure detectors on wide area network. In: Tsuchiya T, ed. Proc. of the Int'l Conf. on Dependable Systems and Networks (DSN 2005). Yokohama: IEEE CS Press, 2005. 624−633.

[22] Aguilera MK, Delporte-Gallet C, Fauconnier H, Toueg S. On implementing omega with weak reliability and synchrony assumptions. In: Borowsky E, ed. Proc. of the 22nd ACM Symp. on Principles of Distributed Computing. Boston: ACM Press, 2003. 306−314.

[23] Muller M. Performance evaluation of a failure detector using SNMP. Technical Report, LSR-REPORT-2004-034, Lausanne: École Polytechnique Fédérale de Lausanne, 2004. 12−24.

附中文参考文献:

[6] 张悠慧,汪东升,郑纬民.工作站机群系统自动重构机制.电子学报,2000,28(5):13−16.

[14] 陈宁江,魏峻,杨波,黄涛.Web 应用服务器的适应性失效检测.软件学报,2005,16(11):1929−1938. http://www.jos.org.cn/1000-9825/16/1929.htm

[18] 田东,陈蜀宇,李静.一种新的分布式系统自适应故障检测器.哈尔滨工业大学学报,2006,38(增刊):374−377.

董剑(1978－),男,山东章丘人,博士生,讲师,主要研究领域为分布式计算,容错计算技术.

刘宏伟(1971－),男,博士,副教授,CCF 高级会员,主要研究领域为软件测试,软件可靠性评估.

左德承(1972－),男,博士,副教授,CCF 高级会员,主要研究领域为移动计算,可穿戴计算技术.

杨孝宗(1939－),男,教授,博士生导师,CCF 高级会员,主要研究领域为容错计算,移动计算,网格计算技术.