第35卷 第17期 计 算 机 工 程 2009年9月
Vol.35 No.17 Computer Engineering September 2009
·软件技术与数据库· 文章编号：1000—3428(2009)17—0094—03 文献标识码：A 中图分类号：N945

面向Web服务的层次型动态失效检测体系

陈宁江
(广西大学计算机与电子信息学院，南宁 530004)

摘 要：根据Web服务的特点，考虑服务依赖关联和检测的可靠性需求，提出一种基于服务依赖关联和可靠性需求的层次型失效检测体系构造算法，研究失效检测的动态适应机制，包括反馈式和预防式策略。仿真实验结果表明，该算法可以提高失效检测的准确性和适应性，避免了“伪失效”现象。

关键词：Web服务；可靠性；失效检测；适应性

Hierarchical and Dynamical Failure Detection Architecture for Web Service

CHEN Ning-jiang
(College of Computer and Electronic Information, Guangxi University, Nanning 530004)

【Abstract】According to the character of Web service and in view of service dependent correlation and reliability requirement, a hierarchical failure detection architecture construction algorithm based on service dependent correlation and reliability requirement is presented. The dynamical adaptation mechanisms including reactive strategy and proactive strategy are studied. Simulation experimental results show this algorithm can improve the accuracy and adaptation of failure detection and avoid fake failure case.

【Key words】Web service; reliability; failure detection; adaptation

1 概述

目前，Web服务研究方向逐步转向了服务质量(Quality of Service, QoS)保障问题，如可靠性。人们期望Web服务具有容错能力，在运行期间出现异常或错误时能检测到并进行恢复。失效检测(failure detection)[1]是容错处理的一种关键技术，与容错的效果密切相关。如果出现检测误判、漏测或者检测冗余较高等情况，则会给容错处理带来不利影响。Web服务具有的一些特点，如运行环境的动态多变性、服务运行长期性、复合服务分布性等，要求在实现服务容错时需进行额外的考虑。本文提出一种面向Web服务的失效检测体系，目标是获得检测的完整性、准确性和效率的有效折中，并具有对动态环境的适应性。

2 研究背景

目前Web服务容错研究在可靠消息传递和容错服务架构等方面较多。WS-RM, WS-ReliablityMessaging等规范为服务消息可靠传递提供了全局唯一标识、消息重发等机制[2]，但是它们仅处理了消息层面的问题，还要有上层的机制(如检测服务)来满足应用的容错需求。一些Web服务中间件对Web服务的容错架构及其支持机制进行研究，如文献[3]在消息层提供动态绑定和恢复备份策略；文献[4]为服务组合提供一定的故障检测和恢复操作功能；有学者研究了服务状态复制、消息日志等技术，从进程和内核实现方面探讨了如何缩短失效检测时间。它们的失效检测只是采用了较简单的“心跳”机制，对于面向Web服务特点的检测体系、检测质量改善等方面，还有待进一步研究。

Web服务是种松耦合的软件系统，若干Web服务分布于网络环境中，按照特定的业务逻辑合成应用，如图1所示。

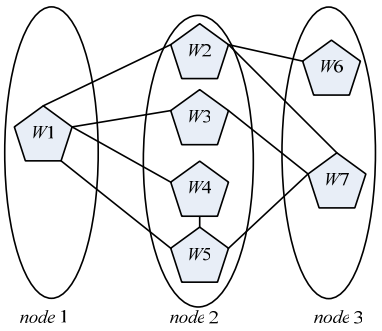

图1 Web服务应用

为保障复合应用的可靠性，需要监测和判断各个Web服务的运行状态(正常或失效)，即失效检测任务。在Web服务的情境下，需要着重考虑3个方面问题：(1)除了节点级失效外，应加强细粒度的服务级失效检测；(2)一个服务的失效可能会影响到服务复合而构成的整个应用，由于重启服务的代价较大，尤其需要避免单个服务“伪失效”而引起无谓的级联失效传播，因此应关注失效检测对环境的适应性，降低误判率；(3)在检测的完整性和效率之间取得适当折中，既要保障每个服务都受到监测，也要尽可能降低检测开销。

3 失效检测体系的构造

本文基于如下假设：服务节点上部署有失效检测器(Failure Detector, FD)与Web服务相对独立；FD基于底层可

基金项目：广西教育厅科研基金资助项目(桂教科研(2006)26号)
作者简介：陈宁江(1975－)，男，副教授，主研方向：网络分布计算，软件工程，中间件技术
收稿日期：2009-03-05 **E-mail：**chnj@gxu.edu.cn

靠消息协议，按照给定的发送时间间隔 T_I 向受检服务发送检测消息，根据该消息在给定时间内的响应情况来判断服务是否失效。针对 Web 服务的分布式和松耦合特点，设计了失效检测器的层次型结构，如图 2 所示。

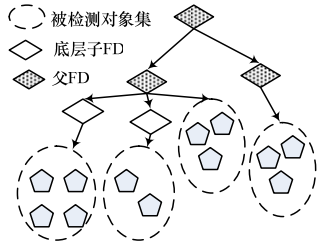

图 2　层次型检测体系

在图 2 中，每个 FD 负责检测一定数量的服务；FD 之间可形成父子关系，即子 FD 的检测对象集 S_s 是父 FD 的检测对象集 S_p 的子集，但是父 FD 是通过子 FD 传递来的检测结果来获知 S_s 中服务的状态。这种结构是集中式和分布式结构的折中，好处主要在于：由于一个 FD 只检测有限范围内的服务，因此可减少检测消息传递的开销；提高检测体系自身的可靠度，某些 FD 的失效不会使某个服务失去监控；分层结构易于扩展，既避免集中式结构的检测中心点的瓶颈问题，也可改善分布式结构的协作效率问题。此时，如何组织 FD 形成检测体系是首先要考虑的问题。

Web 服务应用对失效检测服务质量的期望通过失效检测策略描述。失效检测器的度量指标体系比较复杂[1]。考虑到上述对 Web 服务失效检测的关注点，本文主要选取 3 个指标作为失效检测策略的构成。

定义 1(失效检测策略) $FD\_Strategy=(T_I, Dd, Rn)$，其中，$T_I$ 为检测消息发送的时间间隔；Dd 为检测深度，表示在按照检测父子关系构成的树中的节点层次数；Rn 为备份数目，表示一个 FD 的检测结果向 Rn 个 FD 备份。

定义 2(服务失效依赖树) $FDTree=(S, E)$，其中，S 是 Web 服务节点集；$E=<e1, e2>$为失效依赖关联。若 $serviceDepend(x, y)$表示服务 x 逻辑依赖于 y，即 x 发送业务消息驱使 y 执行，则有 $serviceDepend(e2, e1) \Rightarrow <e1, e2>$。

服务失效依赖反映一个 Web 服务的失效对其他服务的传递影响，可以用它来为失效检测提供一定支持。根据应用的失效检测策略和服务失效依赖关联，设计一种分层服务失效检测体系构造(Hierarchical Service Failure Detector Architecting, HSFDA)算法。设 Web 服务集合为 S，失效检测器 f 所检测的服务集 $W_f=\{w_1, w_2, \cdots, w_p\}$，检测矩阵 $\boldsymbol{FM}=\{fm_{ij}\}$：若 f_i 监测 s_j，则 $fm_{ij}=1$；反之，$fm_{ij}=0$。

HSFDA 算法的步骤如下：

输入 $S; T, FD\_Strategy(T_I, Dd, Rn)$，其中，$S$ 是待检测的服务集合；T 是已知的 $FDTree$ 集；$FD\_Strategy$ 是失效检测策略

输出 $FS, \boldsymbol{FM}$，其中，FS 是 FD 集合；$\boldsymbol{FM}$ 为检测矩阵

BEGIN

(1)初始化阶段：

(1.1)FS←Φ ；获取权重向量 WD =<wd_1, wd_2, …, wd_n>的初值；//wd_i 是树 td_i ∈T 中的服务数

(1.2)FOR each $s_j \in S$ DO {

FS←FS ∪{ f_i }；　//构建初始的 FD 集合，初始时一个节点的//服务选择由位于相邻节点的一个检测器 f_i 检测

S_{fi} ← { $s_1, s_2, \cdots, s_p$ } // S_{fi} 为 f_i 所监测的服务集合

IF　($s_j \in S_{fi}$) THEN fm_{ij}=1；//构建初始的 FM 矩阵，f_i 监测 s_j

(1.3)FOR (k =1 ; k≤|S|; k++) DO

{　IF (fm_{ik}==1) THEN

//若 s_k 无依赖，则选择一个距离最近的检测器 f_n 对其监测

IF (s_k belongs to $td_m \in$T AND wd_m== 1) THEN { fm_{nk} = 1;　S_{fn} = S∪{s_k};　}

ELSE { FOR all s_j where AND wd_m > 1)　DO　// 若 s_j 依赖于 s_k，则将 s_j 纳入 f_i 的检测集

{IF (<s_j , s_k > ∈ td_m)　THEN { fm_{ij} = 1;　S_{fi} = S∪{s_j}; }}

}

(1.4) FOR (k =1 ; k≤|FS|; k++)

//若仅有一个检测器 f_k 对 s_j 检测，则增加一个对 s_j 的检测器

{　获取一个距 f_k 最近的检测器 f_l ;

FS←FS ∪{ f_l };　fm_{lj} = 1 ;}}

(2)构建 FD 的层次

(2.1)将 FS 分为 (Rn+1)个子集

(2.2)FOR (k=1; k<Dd; k++) DO

{FOR each FS^k DO {构造 $FS^{k=1}$}

DO{使用 round-robin 方法选择 Rn 个 FS^k 的成员，作为 m 的备份目标；}}

END

可以验证，HSFDA 算法生成的检测矩阵 $\boldsymbol{FM}$ 满足：$\forall i, \forall j, \sum_{i=1}^{m} fm_{ij} > 1, \sum_{j=1}^{n} fm_{ij} > 1$，因此，在构造的失效检测体系中，任一 FD 的失效，不会导致它所检测服务的状态无法被获知。除此之外，算法的特点还有：(1)对于服务依赖树中的$<s_j, s_k>$，如果 f 检测 s_k，那么 f 对于存在依赖关联的多个服务都能检测，并可根据依赖推导受失效影响的服务，这有助于减少 FD 的冗余度；(2)若服务 s 失效后，f 根据依赖关联标识与 s 相关的服务进入“怀疑”状态，然后根据其他结果判断是否真正失效，这有助于减少误判；(3)算法建立 FD 的父子关系和备份关系，并使每组的检测集尽量均衡，在一定程度上实现可靠性和效率的折中。

4　适应性策略

动态的网络环境会影响失效检测消息的传递，由于消息丢失或者延迟过大而导致误判失效的可能性是存在的。若一个服务被误判为失效，可能会引起整个复合服务运行中断或者重启，带来不必要的开销。其中，检测相关的时间值是重要影响因素。在 HSFDA 构造的失效检测体系的基础上，研究如何增强对失效检测参数的动态调整，主要有反馈式和预防式 2 种策略：

(1)反馈式调整策略。监测和统计检测消息发送和响应所需的实际时间，根据消息延迟情况来判断网络通信状态，在统计基础上计算调整检测超时时间值 Tr(即 FD 经过 Tr 时间后，如未有检测消息的响应，则判断可能发生失效)。

$$Tr=\begin{cases}\max(T_{lower}, Tr+\alpha stat^K(\text{probe}(T_D))), & \text{if } stat^K(\text{probe}(T_D))<0\\ \min(Tr+\alpha stat^K(\text{probe}(T_D)), T_{upper}), & \text{if } stat^K(\text{probe}(T_D))\geqslant 0\end{cases} \quad (1)$$

其中，T_{lower}, T_{upper} 分别为 Tr 的允许下界和上界；probe(T_D) 为对检测时间 T_D 的探测函数；$stat$ 为特定的统计函数，$stat^K$(probe(T_D))表示 K 次探测的统计变化值，若其值≥0 表示延迟增加，反之则延迟减少；$\alpha \geqslant 1$，为用于调节变化幅度的系数。

根据实时监测的数据所反映的网络延迟，动态调整 Tr 值，使失效检测对网络环境的负载变化具有一定的适应能力，特别是避免由于网络忙而导致误判失效。

(2)预防式调整策略。正常运行时间长的服务可靠性比较高，出于性能的考虑，可适当降低检测强度。因此，根据失

效检测的结果统计，对于出错较少的服务，可增加检测消息发送时间 T_I；反之，则通过减少 T_I 来加强检测。式(2)给出了 T_I 的动态调整函数：

$$T_I^{'}=\begin{cases}\max(T_{I\_lower},T_I-(1/2)^{success(n)}T_S),\text{if 第}n\text{次检验后状态正常}\\ \min(T_{I\_upper},T_I+(1/2)^{fail(n)}T_S),\text{if 第}n\text{次检验后状态失效}\end{cases}\quad(2)$$

其中，T_I 和 T_I' 分别为调整前后的检测消息发送时间；T_{I_lower}，T_{I_upper} 分别为 T_I 的允许下界和上界；$success(n)$和 $fail(n)$分别为连续 n 次检测成功次数和检测到失效次数的统计函数；T_S 为递增或递减的时间步长。以累积检测成功或者失效次数为参考依据，使检测消息发送时间动态修改，通过调整检测消息数目和频度，有助于实现失效检测在可靠性和效率之间的权衡。

5 实验模拟

对 HSFDA 检测体系的检测效果及其适应性策略的效果进行模拟实验。实现一组 Web 服务(规模为 30~40)，其功能是简单地收发 SOAP 消息，预先定义服务之间的业务关联，由此定义失效依赖树。它们分布于局域网中的 4 台 PC 机，以 JBoss4.0 应用服务器(http://www.jboss.org)作为 Web 服务的实现平台，失效检测器模块作为可动态插拔的 JMX 服务插入 JBoss 服务器，通过监测 Web 服务的消息收发进行失效检测。当被测服务在等待给定时间无响应后则进入“怀疑”状态，再等待一段时间若无响应则判断为“失效”。在实验中，采取停止服务的方式模拟失效；使用给定的软件延时程序来模拟环境负载的增减变化；此实验中对于式(1)中的 $stat$ 函数，取求平均值函数。实验主要考察 2 个失效检测的质量指标[1]：用于衡量准确性的检测错误率 λ_M 以及用于衡量快速性的检测时间 T_D。在不同的检测消息发送时间 T_I 取值下，分别进行有无适应性策略的多组测试，统计 2 个指标的平均值。图 3、图 4 为实验结果。

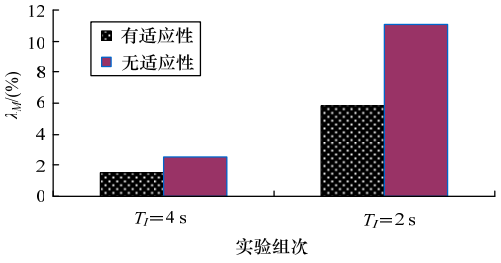

图 3 λ_M 的测试结果比较

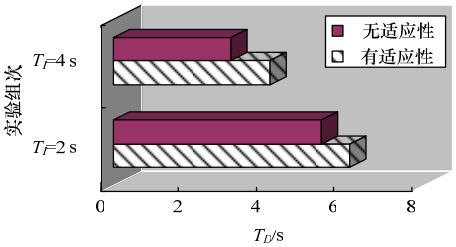

图 4 T_D 的测试结果比较

从图 3、图 4 可以看出，HSFDA 失效检测体系能够较好地检测到失效，并具有对运行环境变化的适应能力。在有适应性策略的情况下的 λ_M 值比无适应性情况大为减小。另一方面，因为考虑了检测的适应性处理和备份处理，带来额外的延迟，所以从 T_D 的值上有适应性策略的情况显得快速性不够。但由于 Web 服务的长运行特点，并且检测误判的代价较高，因此改善 λ_M 比 T_D 更重要。

6 结束语

本文对面向 Web 服务的失效检测技术进行研究，提出一种层次型失效检测体系构造算法，考虑服务依赖关联和检测的可靠性需求，增强对检测的可伸缩性和完整性的满足能力。设计失效检测的动态调整策略，以提高失效检测的准确性，降低“伪失效”情况。下一步工作是深化适应性策略研究，考虑更丰富的检测上下文信息，进一步探讨失效检测器在网络环境中的位置分布问题，以更好地满足检测的 QoS 需求。

参考文献

[1] Toueg S. On the Quality of Service of Failure Detectors[J]. IEEE Trans. on Computers, 2002, 51(5): 561-580.

[2] 刘 影, 何克清, 梁 鹏, 等. Web 服务中可靠性消息规范的比较研究[J]. 计算机应用研究, 2006, 23(12): 101-103.

[3] Erradi A. QoS-aware Middleware for Reliable Web Services Interactions[C]//Proc. of IEEE Int'l Conf. on e-Technology, e-Commerce and e-Service. Washington D. C., USA: IEEE Computer Society, 2005.

[4] Eric W. A Service-oriented Middleware for Runtime Web Services Interoperability[C]//Proc. of the IEEE Int'l Conf. on Web Services. [S. l.]: IEEE Computer Society, 2006.

编辑 陈 文

（上接第 93 页）

算法的填充准确率和分类准确率都下降，因为缺失比率越大，可用信息越少，所以准确率越低。而 CII 算法的填充准确率和分类准确率始终最高。

参考文献

[1] Quinlan J. C4.5: Programs for Machine Learning[M]. San Mateo, Brazil: Morgan Kaufmann, 1993.

[2] Zhang Shichao, Zhu Xiaofeng, Zhang Jilian, et al. Efficient Imputation Method for Missing Values[C]//Proc. of PAKDD'07. Nanjin, China: [s. n.], 2007: 1080-1087.

[3] Numao M. Ordered Estimation of Missing Values[C]//Proceedings of PAKDD'99. Beijing, China: [s. n.], 1999: 499-503.

[4] Zhu Xiaofeng, Zhang Shichao, Zhang Jilian, et al. Cost-sensitive Imputing Missing Values with Ordering[C]//Proc. of the 22nd AAAI Conference on Artificial Intelligence. Vancouver, Canada: [s. n.], 2007: 1922-1923.

[5] Breiman L, Friedman J, Olshen R, et al. Classification and Regression Trees[M]. Monterey, Canada: Wadsworth & Brooks, 1984.

编辑 陈 晖