第36卷 第20期 计 算 机 工 程 2010年10月
Vol.36 No.20 Computer Engineering October 2010
·开发研究与设计技术· 文章编号：1000—3428(2010)20—00262— 文献标识码：A 中图分类号：TP393

多因素集成的分布式应用故障诊断方法

尹 殷，李 巍，李云春
(北京航空航天大学计算机学院网络技术北京市重点实验室，北京 100191)

摘 要：采用主动探测的方法对分布式应用进行故障管理。针对分布式应用的特点，在主动探测的故障检测和故障诊断阶段中，分别提出基于成本效益平衡的检测集选择算法和诊断集中考虑探测成本的最小贪婪搜索算法。实验结果表明，算法在探针数量、探测时间、探测流量上都有明显的改进。

关键词：主动探测；贝叶斯网；分布式应用；故障诊断

Multi-factor Integrated Fault Diagnosis Method in Distributed Applications

YIN Yin, LI Wei, LI Yun-chun
(Key Lab of Beijing Network Technology, School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

【Abstract】 This paper uses active prode for fault management of distributed applications. According to the features of distributed applications, probe set selection algorithm based on cost-benefit balance is proposed in the fault detection phase, and diagnosis probe selection algorithm based on minimum greedy search and probe cost is proposed in the fault diagnosis phase. Experimental results show that the number of probes, detection time and detection traffic have significant improvement.

【Key words】 active probe; Bayesian network; distributed applications; fault diagnosis

1 概述

随着网络规模的不断扩大以及分布式应用技术的日益发展，故障诊断已经成为一项重要的网络任务。针对现今的网络特点，一个成功的故障诊断技术应该具备以下特点：诊断正确率高，诊断速度快，诊断所需的管理流量小，开发成本低等。

在分布式系统管理中常用的故障诊断方法是事件关联[1]，被管设备在状态改变时向外发出警报，中央管理器通过收集这些警报找出故障原因。但这种方法对每个设备有额外的要求，并且很难保证警报正常发送。基于端到端的探测技术[2]可以避免这些问题。探针是执行在特定的机器(探针站)上的程序，通过发送命令或请求到服务器或网络组件并测量其反应，例如 ping 和 traceroute 命令可以用来检测网络的可用性。其他探测工具，如 IBM 的端到端探测平台(EPP)技术[2]，提供了更先进的应用程序级探针，如测试电子邮件、数据库查询等。

2 相关工作

近几年，随着分布式应用故障管理研究的发展，有一种将传统网络中的故障检测方法扩展到分布式应用服务中的趋势。因此，故障节点的概念从物理链路扩大到服务、应用、服务器等组件。

分布式应用的故障管理主要包括故障发现、故障定位和故障恢复 3 个阶段。文献[3]对计算机网络中的故障检测技术进行了综述，总结了多种检测技术，包括基于人工智能的方法、基于模型的方法、基于案例的方法等；应用神经网络、决策树[4]、贝叶斯网络等技术进行实现，指出故障传播模型是近几年的研究重点。

Steinder 和 Sethi 使用 2 层的网络结构对网络服务故障管理中的依赖关系进行建模。以端到端的网络路径服务为例，把物理链路作为故障并抽象为父节点，把对端到端的网络路径服务的观察抽象子节点(症状)、父节点和子节点之间的影响程度用条件概率表示，提出了基于事件驱动的 Incremental Hypothesis Updating(IHU)算法[5]用于故障检测。

文献[2]提出一种基于智能探针的方法对分布式应用进行故障检测，探针的结果用子节点表示(症状)，父节点表示应用、服务器、网络等组件的状态(故障)，使用依赖矩阵表示故障和症状之间的依赖关系，利用集合论和信息论的方法进行探针选择，并根据探针的结果进行故障发现和故障定位。

先前的研究只考虑了探针的检测能力，没有考虑探针的检测成本，只是简单地把探针数量作为衡量算法优劣的标准。这不符合探针检测的实际情况。例如，一次 ping 的检测的成本明显小于一次数据库的连接测试。本文针对分布式应用的特点，采用主动探测的故障检测方法，提出对探针效益和成本中的多个因素进行综合评价，并设计了基于成本效益平衡的检测集选择算法和诊断集中考虑检测成本的最小贪婪算法。

3 故障检测和诊断中的探针选择算法

基于主动探测的故障管理分为 2 个阶段：故障检测和故障诊断。在故障检测阶段，使用一个较小故障检测探针集周期性对整个网络进行检测。该检测集的目标是判断网络中是

基金项目：北京教育委员会共建项目建设计划基金资助项目(JD100060630)；航空基金资助项目(2008ZC51050)
作者简介：尹 殷(1984－)，男，硕士研究生，主研方向：分布式应用故障管理；李 巍、李云春，副教授、博士
收稿日期：2010-05-07 **E-mail：**yinyin2010@gmail.com

否出现了故障，但是无法对故障进行准确的定位。一旦有探针报告错误，说明网络中有节点发生了故障，便会进入故障诊断阶段，此时为了获得更多的信息，会选择更多的探针进行探测，以便对故障的根本原因进行定位。

3.1 基于成本效益平衡的故障检测方法

设故障探针依赖图为 $G(F, P, E)$，其中，F 和 P 中的元素都是顶点，分别代表故障集和探针集；E 中的元素为边，反映了故障和探针的依赖关系。如果存在边 $e=\{f_i, p_j\}$时，就称探针 p_j 能检测故障 f_i。

用于故障检测的探针集必须满足以下条件：如果网络中有节点或服务出现了故障，那么该检测集中至少有一个探针会探测失败。该问题是 NP 问题[2]，并且可能有多个解。针对该问题，文献[6]采用最小搜索和最大搜索，文献[2]采用了基于信息熵算法，它们都是以检测集的元素个数作为衡量标准，选出的解是近似的，并且只给出一个检测集。

但是，在实际的环境中不同探针的成本和效益是不同的。它们在探测时间、所需流量、探针的复杂度等方面有着明显的差别。例如，数据库的连接测试比路由探测的成本高，节点的性能测试比 ping 的诊断效果好，本文的目标是寻找成本小、效益高的检测集。

给定故障探针依赖图 $G(F, P, E)$，以及探针的成本型属性和效益型属性，选择一个探针集 $Pdet \subseteq P$，使得网络中所有的故障节点都能被 $Pdet$ 中的某一探针检测，$Pdet$ 的中任一元素不可或缺，称 $Pdet$ 为故障检测集。同时，选择成本低、效益高的检测集。算法伪代码如下：

算法 PSFD 故障检测集探针选择算法

输入 $G(F, P, E)$：故障探针依赖图；$Pattr$：探针的成本和效益型属性

输出 按优劣排序的故障检测集的集合 PSS

1. 初始 $PSS_0=\{\Phi\}$
2. 对每个在 F 中出现的故障 f_i
3. $PSS_i=\{\Phi\}$
4. 对每个 $p_l \in P$, 使得 $u(p_l)=|P|$
5. 对每个 $PS_j \in PSS_{i-1}$
6. 对每个 $p_l \in PS_j$ 并且 $p_l \in P_{fi}$
7. $u(p_l)=\min(u(p_l),| PS_j|)$
8. 把 PS_j 加入 PSS_i
9. 对每个 $PS_j \in PSS_{i-1}-PSS_i$
10. 对每个 $p_l \in P \cap PTF_{fi}$ 且 $u(p_l)>|PS_j|$
11. 把 $PS_j \cup \{p_l\}$加入 PSS_i
12. 对 $PSS_{|F|}$中的每个检测集进行模糊评价
13. 返回 $PSS_{|F|}$

$PSS_{|F|}$中的每个元素都是一个故障检测集，并利用客观赋权的模糊评价方法对 $PSS_{|F|}$中的元素进行评价，从中选出一个“最优”的故障检测集。

本文的模糊评价方法采用了变异系数法[7]，其输入包括待评的方案及其属性，其中属性分为成本型属性和效益型属性；输出为方案的优劣排序。其思想是：如果某项指标的数值能明确区分各个被评价的方案，说明该指标的分辨信息丰富，就赋予较大的权重；反之，如果区分度不高，就赋予较小的权重。根据属性权重计算各个方案的得分，并按其得分对方案的优劣进行排序。

检测集的成本型属性包括：检测集元素个数，探测总时间，探测总流量等；效益型属性包括：探针检测范围，探针总检测能力等。

算法迭代处理每个可能出现的故障，产生当前已遍历故障的检测集。迭代过程中第 i 个故障 f_i 产生检测集的集合为 PSS_i，该检测集可由 PSS_{i-1} 及故障探针依赖图 $G(F, P, E)$得出。每个 $PS_j \in PSS_i$ 都是已遍历故障$\{F_1, F_2, \cdots, F_i\}$的检测集。在第 i 次迭代中，分析每一个 $PS_j \in PSS_i$，如果 PS_j 能够探测 f_i，则将 PS_j 加入 PSS_i；如果不能探测故障 f_i，需要通过添加 PTF_{fi} 中的探针进行扩展，PTF_{fi} 为所有能探测故障 f_i 的探针集。为了防止计算复杂度以指数级增长，采用启发式算法，若 $P_1 \in P_{fi}$，$PS_j \in PSS_i$，只有当 PS_j 的大小$|PS_j|$小于 PSS_{i-1} 中每一个包含 P_1 且能探测 f_i 时，探针 P_1 才可以被加入 PS_j。

该算法进行了$|F|$次迭代，每次迭代执行了 2 次 for 循环，均需要 $O(\max(|PSS_i|)|P|)$个步骤。在极个别的情况下，PSS 的大小会呈指数型增长，为了避免这个问题，本文限制了$|PSS|$的大小，限制$|PSS|$的大小为 $2|P|$，最后得到算法的复杂度为 $O(|F||P|^2)$。

3.2 多因素考虑的故障诊断方法

当检测探针集中有探针报告错误时，说明系统中出现了故障，此时进入故障诊断阶段。故障诊断的任务是根据当前探针的结果和故障探针依赖图 G 选择合适的探针对系统进行探测，然后根据这些探针的结果推断系统的状态，直至对系统的故障进行了定位。

给定当前已探测探针的结果 $Pdet\_ret$，故障探针依赖图 G，选择一个合适的探针序 $Ploc$，在保证故障检出率的同时，尽量减少 $Ploc$ 造成的各种成本开销。本文提出对探针的多种属性进行考虑，并利用一种基于客观赋权的模糊评价方法对探针进行评价。算法伪代码如下所示：

算法 PSFL 故障诊断探针选择算法

输入 $G(F, P, E)$：故障探针依赖图；$Pdet\_ret$：检测探针集的结果；$Ffail$：故障节点

输出 用于故障定位的探针集 $Ploc$

1. 把成功探针路径上的节点加入 Fpass
2. 初始可疑故障集合 Fsusp，加入在失败探针路径上，但不在 Fpass 内的故障节点
3. 构建可用探针集 Paval(探测路径包括 Fsusp 但是不包括 Ffail 的所有探针)
4. Ploc <- Null,.Ftarg <- Fsusp, Pspac <-Paval
5. while |Ftarg|!=Null do
6. 构建 Pspac 集的属性矩阵 Pattr
7. Porder<-模糊评价算法(Pattr,Pflag)
8. Pnext <- Porder 中最优的探针
9. 把 Pnext 加入 Ploc
10. 从 Pspac 中删除 Pnext
11. 从 Ftarg 中删除 Pnext 能检测的节点
12. end while
13. 返回 Ploc

探针的成本型属性包括探测时间、探测流量、探针的复杂度；探针的效益型属性包括、探针的检测质量、探针的信息熵[6]。优先考虑对可疑故障覆盖较少的探针，当该探针失效时，便能极大地缩小可疑故障的空间。

计算正常节点集 $Fpass$ 和可疑节点集 $Fsusp$；计算出所有能检测可疑节点的可用探针集 $Paval$；计算可用探针的各种属性，利用 3.1 节提到的模糊综合评价对可用探针的优劣进行排序，从中选出最合适的探针 $Pnext$；把 $Pnext$ 加入到 $Ploc$，

更新可用探针搜索空间 *Pspac* 和目标节点 *Ftarg*，如此循环，直到 *Ftarg* 为空。该算法最多需要|*Ftarg*|次循环，循环中的模糊评价算法的复杂度为 $O(|Pspac|^2)$，所以算法的复杂度为 $O(|F||P|^2)$。

4 实验验证

一个典型的分布式应用管理的场景如图 1 所示。一个分布式应用通常包括很多组件，由分散在网络上的逻辑和物理节点组成，各节点实现不同的服务功能，可以把它映射到故障探针依赖图 $G(F, P, E)$中。

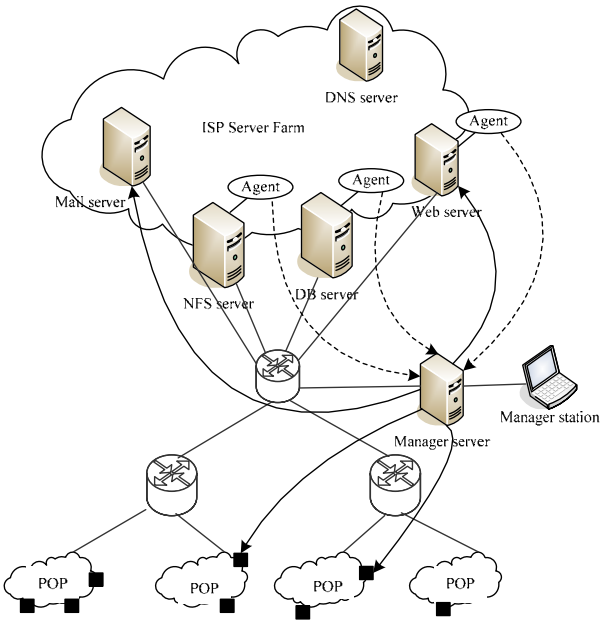

图 1　典型的分布式应用场景

在实验开始时，随机产生 *n* 个节点的有向无环图，建立故障探针依赖关系的贝叶斯网。故障的先验故障概率服从[0.001, 0.01]上的均匀分布，节点间的条件概率服从[0.8, 1]上的均匀分布。本文通过对大量探针实际运行情况的分析，假设每个探针的检测时间服从[5, 2 000]上的均匀分布，单位为 ms，探测流量服从[5, 50]上的均匀分布，单位为 KB。

在故障检测的实验中，网络节点数为 10~90，在每个节点数下会生成 100 个图并测试，实验结果取均值。本文对比了文献[2]中贪婪算法和本文的基于成本效益平衡的检测集选择算法。2 个算法选出的检测集的节点数与探针数见图 2，检测集的探测时间如图 3 所示，在检测流量等方面的比较有类似的效果。本文的算法不仅在检测集探针的数目上，在检测时间和其他标准上都要比最大贪婪算法的效果好，但是计算量略大。

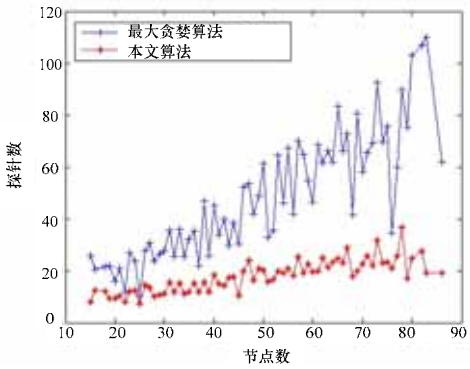

图 2　故障检测集的探针数与节点数

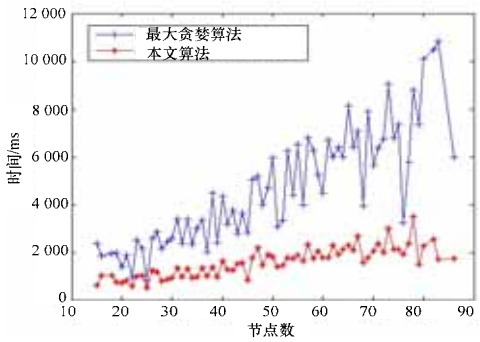

图 3　故障检测集的检测时间对比

故障诊断推理属于贝叶斯网的 MPE 问题，探针的结果为推理提供了证据。一般采用以下 2 个指标评估故障诊断算法：检出率(*DR*)，假阳性率(*FPR*)，其定义如下：$DR=|F_d \cap F_c|/|F_c|$，$FPR=|F_d-F_c|/|F_c|$。其中，F_c 是真实的故障集；F_d 是诊断出来的故障集。检出率表示的是诊断出来的故障占所有故障的百分比。

本文对比了文献[6]提出的最大贪婪算法和本文提出的考虑检测成本的最小贪婪搜索算法。网络节点数为 10~70，在每个节点数下生成 10 个图，每个图的测试样本有 1 000 条，实验结果取平均值。图 4、图 5 分别对比了当 DR 大于阈值 0.95 时，2 个算法所使用的探测节点数和探测流量。在探测时间等方面有类似的结果。可以看出，本文算法与最大贪婪算法相比，探针数没有增加，检测成本有明显减少。

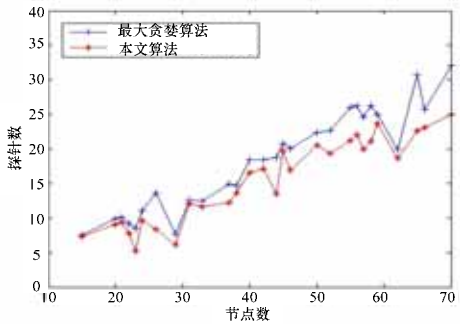

图 4　故障诊断探针数对比

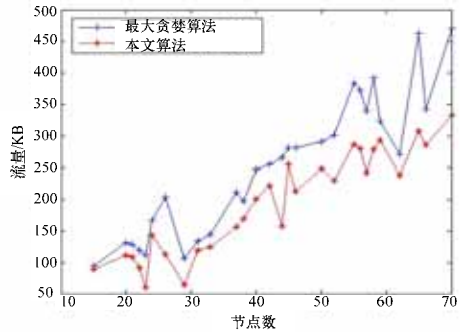

图 5　故障诊断探测流量对比

5 结束语

基于主动探测的故障管理分为故障检测和故障诊断 2 个阶段，本文提出在这 2 个阶段中考虑探针成本的问题，提出的探针选择算法弥补了先前研究的不足。仿真实验结果证明了算法的有效性。但本文只是对考虑成本的探针选择进行了初步尝试，还有一些未曾考虑的问题。例如，在非确定环境下的检测探针集的的选择，应该加入对故障检测质量的评价等，这些是下一步的工作重点。 (下转第 267 页)

参考文献

[1] Gruschke B. Integrated Event Management: Event Correlation Using Dependency Graphs[C]//Proc. of DSOM'98. Samos Island, Greece: [s. n.], 1998.

[2] Rish I, Brodie M, Ma Sheng, et al. Adaptive Diagnosis in Distributed Systems[J]. IEEE Trans. on Neural Networks, 2005, 16(5): 1088- 1109.

[3] Steinder M L, Sethi A S. A Survey of Fault Localization Techniques in Computer Networks[J]. Science of Computer Programming, 2004, 53(2): 165-194.

[4] 曲朝阳, 高宇峰, 聂　欣. 基于决策树的网络故障诊断专家系统模型[J]. 计算机工程, 2008, 34(22): 215-217.

[5] Steinder M, Sethi A S. Non-deterministic Diagnosis of End-to-end Service Failures in a Multi-layer Communication System[C]//Proc. of the 10th International Conference on Computer Communications and Networks. Scottsdale, Arizona, USA: [s. n.], 2001.

[6] Natu M, Sethi A, Lloyd E. Efficient Probe Selection Algorithms for Fault Diagnosis[J]. Telecommunication Systems, 2008, 37(1): 109-125.

[7] Asai K. Fuzzy Systems for Management[M]. Amsterdam, Holland: IOS Press, 1995.

编辑　顾姣健