

云计算环境下大规模 Web 服务故障诊断技术研究

赵雪琴¹, 付媛媛²

(1. 郑州大学西亚斯国际学院, 河南 新郑 451150; 2. 河南财经政法大学 现代教育技术中心, 郑州 450002)

摘要: 随着 Web 服务规模的不断增大, 系统中往往会伴随着大量的噪音数据; 为了提高 Web 服务诊断在噪音影响下的健壮性, 文章研究了大规模 Web 服务下的故障诊断; 首先, 提出了一种 Web 服务自动诊断框架; 其次, 分别定义了转移矩阵和发射矩阵, 并通过这两种矩阵来表达 Web 服务的状态转换过程; 最后, 提出了一种基于 Viterbi 方法的故障诊断算法; 实验表明, 文章提出的算法与其它相关的 Web 服务故障诊断算法相比准确性高, 并且对噪音数据有着良好的健壮性。

关键词: 云计算; Web 服务; 故障诊断; 转移矩阵

Research on Fault Diagnosis for Massive Web Services in Cloud Computing

Zhao Xueqin¹, Fu Yuanyuan²

(1. Sias international university of Zhengzhou Colleage, Xinzheng 451150, China;

2. Henan University of Economics and Law Modern Educational Technology Center, Zhengzhou 450002, China)

Abstract: As the scale of Web services is becoming larger, there are always noise data in the system. In order to improve the robustness of fault diagnosis of the Web services, this paper studies the problem of fault diagnosis for Web services. Firstly, we proposed an automatic diagnosis framework for Web service. Secondly, we defined the transition and emission matrixes, respectively, and used these matrixes to represent the state transition of Web services. Finally, we proposed a Viterbi based fault diagnosis algorithm. The experiments show that, the proposed algorithm has higher accuracy than related works, and has good robustness for noise data in Web services.

Keywords: cloud computing; web services; fault diagnosis; transition matrix

0 引言

随着云计算技术的发展, 互联网上出现越来越多的 Web 服务。为了提供持续的高质量的 Web 服务, 云服务平台往往需要进行系统故障的自动识别和诊断^[1]。Web 服务中的故障诊断有着广泛的研究, 采用的方法主要有基于模型的诊断方法和基于进程历史数据的诊断方法。

基于模型的故障诊断方法^[2]假设 Web 服务故障可以预先定义, 故障的整个进程信息是已知的。然而对于大多数 Web 服务来说, 由于系统的复杂性以及大规模的用户交互, 故障模型很难预先定义。基于进程历史数据的故障诊断方法利用进程运行时产生的数据记录作为先验知识, 通过对先验知识的分析来诊断 Web 服务中的故障。这种方法应用 Web 服务的历史数据构造故障模型, 从而在故障模型未知的情况下进行故障的诊断。然而, 这种方法却不能对故障的形成原因进行合理的解释。此外, Kopp^[3]等人将 Web 服务分为多个层次, 在故障定位和故障理解时采用由上至下的方式逐步细化。

本文提出了一种基于路径匹配的故障诊断方法。应用 Web 服务运行时产生的日志来构建转移矩阵和发射矩阵, 并根据构建的矩阵和例外执行路径计算出进程正确运行时的路径, 最后将正确运行的路径与故障的路径进行对比分析故障的原因。应用本文提出的故障诊断方法不仅可以对 Web 服务的故障进行诊断, 还可以对诊断的结果进行合理的解释。

1 相关工作

为了加强复杂 Web 服务中的故障管理, 并在服务发生整体故障时进行合理的解释, Ardissono 等人^[4-6]提出了一种用于 Web 服务的故障诊断框架。在该方法中, 每个 Web 服务作为一个组件, 采用灰盒模型对组件进行分析和测试, 通过局部诊断和全局诊断的协作来进行故障的解释。Yan 等人^[7]通过研究发现, 如果大多数 Web 服务都是粗粒度的, 并且它们的输出依赖于输入, 那么 Ardissono 等人提出的方法几乎需要对所有的 Web 服务进行诊断。

Li 等人^[8]对每个 Web 服务建立局部诊断模型, 并将局部故障转化为有色 Petri 网模型。在全局诊断中, 通过递归地进行局部诊断, 并最终使整个网络达到一致性状态。该方法的诊断正确性依赖于模型的完整性, 因此当故障不可预测时, 该方法不能进行正确的诊断。

Yan 等人^[7]提出一种自动的故障诊断模型。该模型根据进程模型和系统的监测数据重构服务的执行路径, 并根据重构的执行路径进行故障的诊断。该方法的假设前提是, 当重构的执行路径与预先定义的进程模型不一致时, 系统发生故障。然而实际上 Web 服务的进程模型往往不能预先定义, 因此该方法不能保证故障诊断结果的正确性。

Friedrich 等人^[9]提出一种基于有限状态机的故障诊断方法对 Web 服务接口的行为进行建模。通过计算两个接口的相似性, 该方法可以诊断任意两个服务接口的行为不一致性。然而, 该方法只适用于服务接口的诊断, 并不能有效地应用于 Web 服务的逻辑故障诊断。

Liu 等人^[10]通过定义活动之间的数据依赖性和控制依赖性来对 Web 服务过程进行建模。在诊断出故障后, 该方法可以

收稿日期:2014-05-03; 修回日期:2013-06-09。

基金项目:河南省自然科学基金项目(20072633)。

作者简介:赵雪琴(1972-),女,河北藁城人,硕士,讲师,主要从事计算机技术方向的研究。

根据服务之间的依赖关系进行故障的修复, 然而却不能对故障发生的位置进行具体的描述。

2 Web 服务自动诊断框架

本节提出一种 Web 服务自动诊断框架, 该框架在 Web 服务发生异常时能及时准确地检测故障行为, 对故障的原因进行识别, 从而进行有效的故障恢复策略。

在 Web 服务运行时, 系统中各个组件之间的状态转换可以用转移矩阵来表示, 本文应用 Web 服务的运行日志来生成转移矩阵。Web 服务运行日志是 Web 服务正常运行下产生的路径集合, 即 $SL = \{et_i \mid 1 \leq i \leq n\}$; 并且每个执行路径都是一个活动构成的序列, 即 $et_i = ((a_1, in, a_1, a_1.out), \dots, (a_m.in, a_m, a_m.out))$, 其中 m 为 et_i 中活动的个数。

本文提出一种 Web 服务故障诊断框架, 其结构如图 1 所示。该诊断框架主要包括以下几部分:

- 1) Web 服务日志 SL ;
- 2) 异常处理器 (Exception Handler) 用于捕获服务异常并触发全局故障诊断服务 GDS;
- 3) 预处理器 (Preprocessor) 从日志 SL 中构建相关的矩阵;
- 4) 全局故障诊断服务 GDS 接收诊断需求, 触发局部诊断服务 DS, 并记录诊断结果;
- 5) 局部诊断服务 DS 负责对相应的 Web 服务进行诊断, 并将诊断结果传递到 GDS。

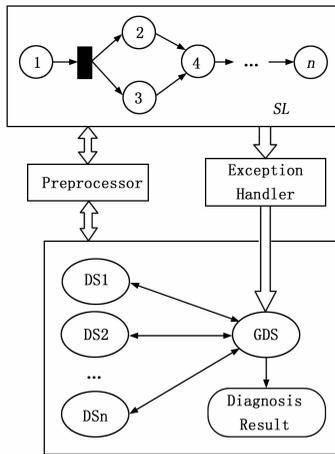


图 1 Web 服务故障诊断框架结构图

在该诊断框架中, 可以在任意时刻独立地加入、移除和替换局部诊断服务, 这极大地简化了故障的诊断过程。每个局部诊断服务可以依据局部 Web 服务定制相应的诊断策略, 这种方法可以提高整个诊断系统的诊断质量。此外, 在双方没有预先定义的情况下, 局部诊断服务可以与全局诊断服务进行单方向通信。

3 转移和发射矩阵

本文引入了活动转移矩阵、消息转移矩阵和从活动到消息的发射矩阵。在 Web 服务中, 部分活动可能并发地或者重复执行, 因此在构建转移矩阵时不仅要考虑活动或者消息在执行路径中的位置, 还要考虑这种特殊关系。

为了识别上述两种关系, 本文应用文献 [11] 提出的工作

流挖掘方法构建上述矩阵。如果活动 a_1 和 a_2 在任何执行路径中都相互伴随, 那么可以认为它们是并发的, 并用 $\{a_1, a_2\}$ 表示。于是, 包含并发活动 a_1 和 a_2 的执行路径可以表示为 $(\dots, a, \{a_1, a_2\}, a', \dots)$ 。在计算转移矩阵时, 要同时考虑 $a \rightarrow a_1, a \rightarrow a_2, a_1 \rightarrow a'$ 和 $a_2 \rightarrow a'$ 4 种情况。当一个子序列在执行路径中连续且重复出现时, 认为该子序列是一个自环, 并用 $[a_1, a_2]$ 表示。于是, 包含自环 $[a_1, a_2]$ 的执行路径可表示为 $(\dots, a, [a_1, a_2], [a_1, a_2], [a_1, a_2], a', \dots)$ 。此时, 计算转移矩阵仅需考虑 $a \rightarrow a_1$ 和 $a_2 \rightarrow a'$ 两种情况。

活动转移矩阵 $AM_{m \times m}$ 是一个二维矩阵, 其中第 k 列第 j 行的值 am_{kj} 表示活动 a_k 到活动 a_j 的转移概率, 其值为 $a_k \rightarrow a_j$ 的个数与 a_k 的个数的比值, 即

$$am_{kj} = \frac{|(a_k, a_j)|}{|a_k|} \quad (1)$$

其中: $|x|$ 为 x 在状态转移图中出现的次数。

消息转移矩阵 $MM_{r \times r}$ 中第 k 列第 j 行的值 mm_{kj} 表示消息 m_k 到活动 m_j 的转移概率, 其值为 $m_k \rightarrow m_j$ 的个数与 m_k 的个数的比值, 即

$$mm_{kj} = \frac{|(m_k, m_j)|}{|m_k|} \quad (2)$$

其中: m_k 和 m_j 分别为活动 a 的输入和输出, 即 $m_k = a.in, m_j = a.out$ 。

活动到消息的发射矩阵 $EM_{m \times r}$ 的第 k 列第 j 行值 em_{kj} 表示活动 a_k 到消息 m_j 的转移概率, 其值为 $a_k \rightarrow m_j$ 的个数与 a_k 个数的比值, 即

$$em_{kj} = \frac{|(a_k, m_j)|}{|a_k|} \quad (3)$$

4 诊断算法

为了简化分析, 本文假设故障的运行路径是可以获得的, 并且转移矩阵和发射矩阵独立于程序的运行时间。

首先, 利用消息转移矩阵 MM 和故障执行路径的活动序列 FAS 来构造正确的消息对序列。如果 MM 中消息对的值小于预先定义的 Δ , 那么认为该消息对是一个错误的消息对, 并赋予该活动一个正确的消息对。算法的细节见算法 1。通过该算法, 可以得到故障执行路径的相应的正确的消息对序列 PS 。

算法 1

输入: MM, EM, FAS

输出: PS

- 1) For $i = 1$ to $FAS.length$
- 2) $a_i = FAS(i).activity$;
- 3) $mp_i = (a_i.in, a_i.out)$;
- 4) If $MM(mp_i) \leq \Delta$
- 5) If $MM(a_i.in, \cdot) \leq \Delta$
- 6) $PS(i) = \max(EM(a_i.in, \cdot))$;
- 7) Else
- 8) $PS(i) = \max(MM(a_i.in, \cdot))$;
- 9) Else
- 10) $PS(i) = mp_i$;
- 11) Return PS ;

在计算出 PS 后, 将 AM, EM 和 PS 作为输入数据, 并应用 Viterbi 算法计算得出故障的正确活动序列。Viterbi 算法的

详细信息请参考文献 [12], 算法 2 为故障正确活动序列的 Viterbi 算法描述。

算法 2

输入: $S = AM \cup EM \cup PS$

输出: AS

初始化:

$$\delta_1(s_i) = \pi_i b_i(O_1), \varphi_1(s_i) = 0, 1 \leq i \leq n;$$

迭代:

$$\delta_t(s_j) = \{ \max[\delta_{t-1}(s_i) a_{ij}] b_j(O_t) \mid 1 \leq i \leq n \},$$

$$\varphi_t(s_j) = \{ \operatorname{argmax}[\delta_{t-1}(s_i) a_{ij} \mid 1 \leq i \leq n \},$$

其中 $2 \leq t \leq T, 1 \leq j \leq n;$

停止:

$$P^* = \{ \max[\delta_T(s_i)] \mid 1 \leq i \leq n \},$$

$$q_T^* = \{ \operatorname{argmax}[\delta_T(s_i)] \mid 1 \leq i \leq n \};$$

结果序列 AS:

$$q_t^* = \varphi_{t+1}(q_{t+1}^*), t = T-1, T-2, \dots, 1$$

在得到了故障的消息对序列 PS 和正确活动序列 AS 后, 将这两个序列与 Web 服务中故障的运行路径相对比就可以对故障进行诊断和识别。

5 实验结果与分析

为了对提出的方法进行有效性验证, 本文应用文献 [13] 收集的 2 507 个真实 Web 服务来构成复杂的服务, 建立了一个 Web 服务模拟环境。该模拟实验环境的结构如图 2 所示。在从 workflow 生成文件中得到节点个数 m 和控制节点数 cn 后, 先生成一些服务池, 并将这些 Web 服务分配到服务池中。随后, 从工作池中随机选择一个服务作为工作流的开始节点。根据开始节点的输出端口选择下一个节点, 并随机插入 cn 个控制节点直到节点数为 mn 。最终, 可以得到一个包含指定节点数的服务执行路径。

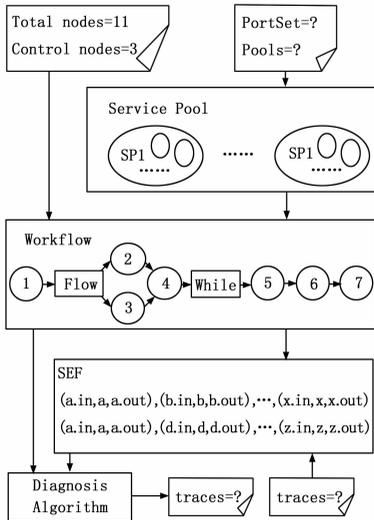


图 2 Web 服务模拟环境结构图

实验共生成 5 个工作流组, 每个分组包括 10 个包含相同活动数目的工作流。在这 5 个分组中, 包含的活动个数为别为 20, 40, 60, 80 和 100。在每个工作流中随机插入 10% 的结构化活动并生成 100 个服务执行路径。其中执行路径中包含不同数量的噪音, 实验选取 10 个故障执行路径进行诊断。

实验采用正确诊断次数与故障路径个数的比值来度量算法准确性。

表 1 为每个工作流注入 2% 的噪音时的准确性测量结果。每一行表示一个分组的测量结果, 每一列表示每个故障执行路径的诊断结果, 最后一列为每个分组的故障诊断准确性。从该表可以看出, 本文提出的方法对不同故障进行诊断的准确性比较稳定, 其值都在 0.8 左右。

表 1 分组实验故障诊断结果

分组	1	2	3	4	5	6	7	8	9	10	准确性
a ₂₀	10	7	9	8	8	7	8	9	9	4	0.79
a ₄₀	6	7	9	10	9	9	8	9	8	3	0.78
a ₆₀	7	8	8	8	8	10	8	8	6	9	0.8
a ₈₀	6	9	8	8	7	10	8	7	8	7	0.78
a ₁₀₀	8	9	8	9	7	7	9	7	9	8	0.81

随后, 通过对工作流注入不同程度的噪音来观察算法的准确性, 实验结果如图 3 所示。图 3 中 x 轴表示噪音占工作流的比例, y 轴为算法的准确性。从该图可以看出, 随着噪音比例的增大, 算法在 5 组数据中的准确性都逐渐降低。然而, 即使噪音的比例达到 10%, 算法的准确性仍然在 70% 以上。

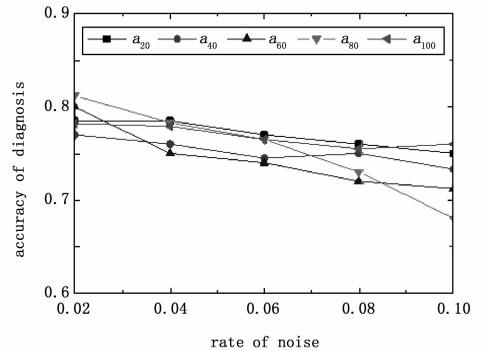


图 3 算法的准确性随噪音的变化

最后, 实验将本文提出的方法与 Fingerprints 算法^[14] 和 AFD^[15] 算法的准确性进行了对比。该实验分别测试了 3 种算法在已知故障类型和未知故障类型两种情况下故障预测的准确性, 实验结果如图 4 所示。从该图可以看出, 本文提出方法在故障类型已知和未知两种情况下算法准确性都高于其它两种算法。

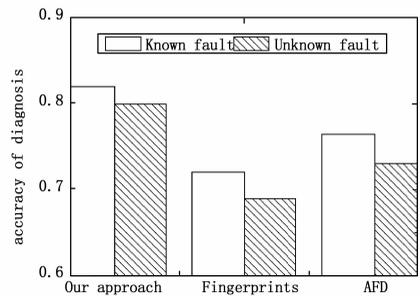


图 4 算法的准确性对比

6 结论

本文研究了云计算环境下大规模 Web 服务的故障诊断和 (下转第 2766 页)

惯性大, 传感器和采集系统的延时作用, 实际控制效果存在微小的滞后。

5 结束语

本文搭建了电动助力转向系统的半实物仿真试验台, 该平台以 dSPACE 为核心, 借助模块化建模工具 Matlab/Simulink 与 dSPACE 系统的无缝链接以及实时测试工具 ControlDesk, 采用模糊和遗传离线优化 PID 算法进行建模和控制算法分析, 并进行在线分析、调试和验证。实验结果验证了控制算法的良好控制性能和该实验平台的可靠性, 具有一定的应用价值。

参考文献:

- [1] Chen X, Yang T B, Chen X Q. A Generic Mode-Based Advanced Control of Electric Power-Assisted Steering Systems [J]. IEEE Transaction on Control Systems Technology, 2008, 16 (6): 1298-1299.
- [2] 熊亮, 刘和平, 彭东林. 无转矩传感器汽车电动助力转向系统的控制策略 [J]. 汽车工程, 2013, (8): 711-715.

- [3] 赵树恩, 刘文文. 基于 NCD/自适应模糊 PID 的汽车 EPS 系统控制特性研究 [J]. 计算机测量与控制, 2013, (11): 2963-2965.
- [4] 邱亚琴, 张向文. 电动助力转向系统的滑模控制算法研究 [J]. 计算机仿真, 2013, (8): 159-163.
- [5] 谭光兴, 宋 畅, 吴列军, 等. 基于整车动力学模型的 EPS 预测控制策略仿真研究 [J]. 计算机测量与控制, 2012, (7): 1796-1799.
- [6] 向铁明, 易际明, 方 遵. 电动助力转向系统特性曲线的设计 [J]. 西华大学学报 (自然科学版), 2009, (3): 35-38.
- [7] 向 丹, 迟永滨, 李武波, 等. 电动助力转向系统控制策略及其仿真研究 [J]. 控制工程, 2013, (2): 254-258.
- [8] 刘金琨. 先进 PID 控制及其 MATLAB 仿真 [M]. 北京: 北京航空航天大学, 2003.
- [9] 谢 刚, 殷国富, 等. 基于 PID 参数自整定控制的电动助力转向系统跟随性研究 [J]. 机床与液压, 2007, 7 (7): 15-17.
- [10] 国 珍. 改进遗传算法在无刷直流电机调速系统 PID 参数优化中的应用 [J]. 组合机床与自动化加工技术, 2013, (7): 90-91.

(上接第 2762 页)

识别技术。随着 Web 服务规模的增大, 传统的监控技术无法实时地诊断并定位发生异常的 Web 服务。本文提出了一种 Web 服务故障自动诊断框架, 并提出了一种基于 Viterbi 方法的故障诊断算法。实验表明, 该算法与其它相关的 Web 服务故障诊断算法相比准确性高, 并且对噪音数据有着很好的健壮性。

参考文献:

- [1] 朱大奇, 刘永安. 故障诊断的信息融合方法 [J]. 控制与决策, 2007, 22 (12): 1321-1328.
- [2] 范贵生, 虞慧群, 陈丽琼, 等. 基于 Petri 网的服务组合故障诊断与处理 [J]. 软件学报, 2010, 21 (2): 231-247.
- [3] Kopp O, Leymann F, Wutke D. Fault handling in the web service stack [M]. Service-Oriented Computing. Springer Berlin Heidelberg, 2010: 303-317.
- [4] Ardissono L, Bocconi S, Console L, et al. Enhancing Web service composition by means of diagnosis [A]. Business Process Management Workshops [C]. Springer Berlin Heidelberg, 2009: 468-479.
- [5] Ardissono L, Console L, Goy A, et al. Cooperative model-based diagnosis of web services [A]. 16th International Workshop on Principles of Diagnosis (DX 05) [C]. 2005: 125-130.
- [6] Ardissono L, Furnari R, Goy A, et al. Fault tolerant web service orchestration by means of diagnosis [M]. Software Architecture. Springer Berlin Heidelberg, 2006: 2-16.
- [7] Yan Y, Dague P, Pencole Y, et al. A model-based approach for diagnosing fault in web service processes [J]. The International Journal of Web Services Research (JWSR), [6] 87-110, 2009.

- [8] Li Y, Ye L, Dague P, et al. A decentralized model-based diagnosis for BPEL services [A]. Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on. IEEE [C]. 2009: 609-616.
- [9] Fugini M G, Mussi E. Recovery of faulty Web applications through service discovery [A]. Proceedings of the 1st SMR-VLDB Workshop, Matchmaking and Approximate Semantic-based Retrieval: Issues and Perspectives. 32nd International Conference on Very Large Databases [C]. 2006: 67-80.
- [10] Liu A, Li Q, Huang L, et al. Facts: A framework for fault-tolerant composition of transactional web services [J]. Services Computing, IEEE Transactions on, 2010, 3 (1): 46-59.
- [11] Van der Aalst W, Weijters T, Maruster L. Workflow mining: Discovering process models from event logs [J]. Knowledge and Data Engineering, IEEE Transactions on, 2004, 16 (9): 1128-1142.
- [12] Forney Jr G D. The viterbi algorithm [J]. Proceedings of the IEEE, 1973, 61 (3): 268-278.
- [13] Al-Masri E, Mahmoud Q H. Investigating web services on the world wide web [A]. Proceedings of the 17th international conference on World Wide Web. ACM [C]. 2008: 795-804.
- [14] Bodik P, Goldszmidt M, Fox A, et al. Fingerprinting the data-center: automated classification of performance crises [A]. Proceedings of the 5th European conference on Computer systems [C]. ACM, 2010: 111-124.
- [15] Zhu Q, Tung T, Xie Q. Automatic Fault Diagnosis in Cloud Infrastructure [A]. Cloud Computing Technology and Science (Cloud-Com), 2013 IEEE 5th International Conference on. IEEE [C]. 2013, 1: 467-474.