

文章编号:1001-9081(2008)05-1313-03

基于时间 Petri 网的事件关联检测机制研究

曹海

(四川理工学院 计算机科学系,四川 自贡 643000)

(caohai@suse.edu.cn)

摘要:网络事件关联检测是网络管理急需解决的一个关键问题,基于 Petri 网的事件关联检测机制,提出基于时间 Petri 网的事件关联检测机制,该机制充分考虑了事件关联时间窗口的起始和大小,有效地提高了事件关联检测的准确率。

关键词:网络管理;事件关联;时间 Petri 网

中图分类号:TP393.07 **文献标志码:**A

Study on event correlation detecting mechanism based on time Petri nets

CAO Hai

(Department of Computer Science, Sichuan University of Science & Engineering, Zigong Sichuan 643000, China)

Abstract: Network events correlation detecting is a key issue of network management to solve. The event correlation detecting mechanism based on Time Petri nets was brought forward on the basis of the event correlation detecting mechanism based on Petri nets in this paper. This mechanism considers the beginning and the size of the time window in the event correlation sufficiently and improves the accuracy of the correlated events.

Key words: network management; event correlation; time Petri nets

0 引言

基于 Petri 网的事件关联检测机制考虑到了事件检测中的时间因素,实现了网络事件属性相关的细粒度并行检测。但该机制虽然考虑到了时间因素,但对时间窗口的起始和大小没有进行设置^[1],容易引起关联失效和关联丢失等问题的发生。

鉴于该机制存在的以上不足点,本文在文献[2]基础上,提出基于时间 Petri 网的事件关联检测机制。该机制对时间窗口的起始和大小进行设置,即以某个关键告警事件出现时间来设定时间窗口的起始,并利用时间 Petri 网中时间变迁的时间变量来表示事件关联时间窗口大小,对于时间窗口大小的取值,可根据历史数据的统计和网络的实际运行情况动态综合后得出,从而有效地提高事件关联检测的准确率。

1 网络事件

事件也称告警事件,网络管理领域中的事件通常定义为有关网络中正在发生的情况的信息,包括被管对象状态异常的消息^[3]。网络环境中受管理设备上的硬件和软件故障、安全侵害、性能下降、环境参数变动等都可能通过事件表现出来。其具体表现形式一般为软硬件系统日志、性能参数的测量、各种网络管理协议所定义的事件等可供观察、收集的信息和数据。

2 事件关联概念及操作类型

2.1 事件关联规则

在网络管理中,管理站接收到的事件为原始事件,这些事件是由被管的网络对象产生的。在通常情况下,由于网络中几个失效可能引起许多原始事件,因此很多网络事件是相互

关联的。文献[2]定义了五种基本的事件关联关系,它们的含义如下:

1) $CE = Event1 \wedge Event2$

操作符“ \wedge ”表示“与”,含义是两种事件 Event1 和 Event2 都发生。

2) $CE = Event1 \vee Event2$

操作符“ \vee ”表示“或”,含义是两种事件 Event1 和 Event2 至少有一个发生。

3) $CE = Event1 \rightarrow Event2$

操作符“ \rightarrow ”表示“先”,含义是 Event1 事件发生在 Event2 事件之前。

4) $CE = Event1 ! Event2$

操作符“ $!$ ”表示“非”,含义是 Event1 事件发生而 Event2 事件不发生。

5) $CE = (Event, n)$

操作符“ n ”表示“循环”,含义是 Event 事件实例发生 n 次。

2.2 事件关联操作类型

Jakobson 和 Weissman 根据对事件进行操作的方式,将事件关联分为以下几种类型^[4]:

1) 压缩:将相同事件的多次重现减少为一个事件;

2) 过滤:抑制参数符合特定条件的事件;

3) 抑制:在特定的上下文中抑制事件;

4) 计数:对重复出现的同一事件的数量进行计数,并设立数量阈值,当超过阈值时,对事件进行抑制;

5) 升级:在特定的上下文中,对特定的事件参数赋予更高的值(例如,严重程度);

6) 一般化:将事件替换为它的某一超类事件;

7) 特殊化:将事件替换为它的某一子类事件;

收稿日期:2007-11-14;修回日期:2008-01-14。

作者简介:曹海(1975-),男,四川自贡人,讲师,主要研究方向:计算机网络通信。

8)时序关系:用事件的顺序和发生时刻来关联事件。

3 基于时间 Petri 网的事件关联检测机制

事件关联检测过程实际是将网络子事件与事件关联规则相匹配的过程,因此可以用状态跃迁技术(如时间 Petri 网)来实现。

3.1 对时间 Petri 网的扩充

为了使时间 Petri 网满足事件关联检测的要求,本文在文献[2]的基础上对时间 Petri 网模型进行了扩充,主要表现在以下几个方面:

1)库所

在应用时间 Petri 网模型中,库所分为主库所和辅助库所两种类型。每个主库所与一种类型的事件相对应,而辅助库所只描述状态,没有实际意义。

2)令牌

模型相应地也定义了两种令牌:主令牌和辅助令牌。相对于普通的时间 Petri 网模型,主令牌包含更多的信息,令牌所处的位置决定了令牌的内容。辅助令牌与辅助位置相对应,不包含任何信息,在事件检测中起到一种类似并发概念中的信号量的作用。

3)流

模型中输入弧 $arc \in P * T$ 可以包含变量和常量,变量作为令牌的声明,包含令牌的信息;常量定义了该输入弧上一次移动的令牌数量(又称为弧权),即事件实例。输出弧 $arc \in T * P$ 上的函数表示对输入弧上的变量执行的操作。

4)变迁

模型中变迁 T 上的谓词(guard)限定令牌的内容。guard 是一个逻辑表达式,其参数是该变迁 T 输入 token 所代表的事件的属性,以对事件的内容进行限制,从而实现细粒度的网络事件的关联检测,提高事件关联的准确性;如果变迁 T 不包含令牌,则表示对令牌没有限制。并且瞬时变迁激发是不需要时间的,但是时间变迁的激发是需要一段时间的,激发是瞬时的^[5]。

3.2 基于时间 Petri 网的事件关联检测机制

对于上述的事件关联关系,相应地建立时间 Petri 网模型。模型中事件 Event1 和事件 Event2 的发生时间分别为 t_1 和 t_2 , $[Ta, Ta]$ 和 $[Tb, Tb]$ 代表事件关联时间窗口大小。

图 1 为一个简单事件时间 Petri 网模型。该关联规则表示在瞬时变迁上必须满足 guard,限定了 Event 这个库所产生令牌的方式。对于事件集合 Event 可以采用复杂的逻辑表达式来表示,这种时间 Petri 网模型非常简单,而且从形式上来讲和普通的时间 Petri 网模型一样,在这里不再赘述。

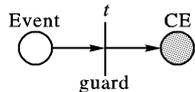


图 1 简单事件时间 Petri 网模型

图 2 为关联规则 $Event1 \wedge Event2$ 的时间 Petri 网模型。该模型表示事件 Event1 和 Event2 在关联时间窗口内都发生。此时间 Petri 网模型的所有可能情况:

- 1)事件 Event1 先发生后,在它所限定的关联时间窗口 $[t_1, t_1 + Ta]$ 内,事件 Event2 也发生;
- 2)事件 Event2 先发生后,在它所限定的关联时间窗口 $[t_2, t_2 + Tb]$ 内,事件 Event1 也发生;
- 3)事件 Event1 先发生后,在它所限定的关联时间窗口

$[t_1, t_1 + Ta]$ 内,事件 Event2 没有发生;

4)事件 Event2 先发生后,在它所限定的关联时间窗口 $[t_2, t_2 + Tb]$ 内,事件 Event1 没有发生。

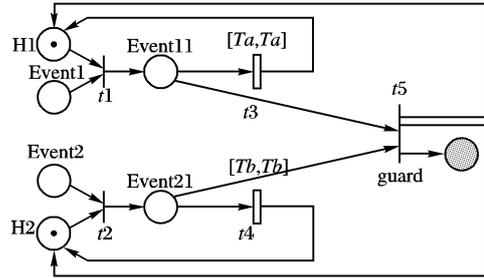


图 2 $Event1 \wedge Event2$ 的时间 Petri 网模型

图 3 为关联规则 $Event1 \vee Event2$ 的时间 Petri 网模型。该模型表示事件 Event1 和 Event2 在关联时间窗口内至少一个发生。由于模型形式与普通的时间 Petri 网的模型一样,并且模型比较简单。

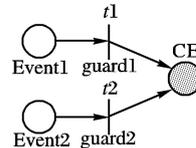


图 3 $Event1 \vee Event2$ 的时间 Petri 网模型

图 4 为关联规则 $Event1 \rightarrow Event2$ 的时间 Petri 网模型。该模型表示事件 Event1 先发生,在事件 Event1 先发生后的关联时间窗口内,事件 Event2 也发生。此时间 Petri 网模型的所有可能情况:

- 1)事件 Event1 先发生,在它所限定的关联时间窗口 $[t_1, t_1 + Ta]$ 内,事件 Event2 也发生;
- 2)事件 Event1 先发生,在它所限定的关联时间窗口 $[t_1, t_1 + Ta]$ 内,事件 Event2 没有发生;
- 3)事件 Event2 先发生。

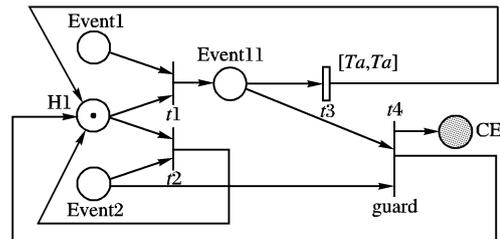


图 4 $Event1 \rightarrow Event2$ 的时间 Petri 网模型

图 5 为关联规则 $Event1 ! Event2$ 的时间 Petri 网模型。该模型表示事件 Event1 先发生,在事件 Event1 发生后的关联时间窗口内,事件 Event2 不能发生。此时间 Petri 网模型的所有可能情况:

- 1)事件 Event1 先发生,在它所限定的关联时间窗口 $[t_1, t_1 + Ta]$ 内,事件 Event2 没有发生;
- 2)事件 Event1 先发生,在它所限定的关联时间窗口 $[t_1,$

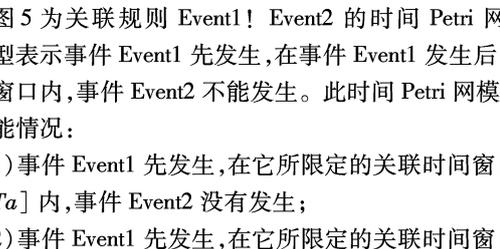


图 5 $Event1 ! Event2$ 的时间 Petri 网模型

$t1 + Ta]$ 内,事件 Event2 发生;

3)事件 Event2 先发生。

图 6 为关联规则循环(Event, n)的时间 Petri 网模型。该模型表示在事件 Event 第一次发生后,在规定的时段内,将事件 Event 的 n 次发生压缩为一次发生。此时间 Petri 网模型的所有可能情况:

- 1)事件 Event 发生后,在它所限定的关联时间窗口 $[t1, t1 + Ta]$ 内,恰有 n 个事件 Event 发生;
- 2)事件 Event 发生后,在它所限定的关联时间窗口 $[t1, t1 + Ta]$ 内,事件 Event 发生不到 n 次。

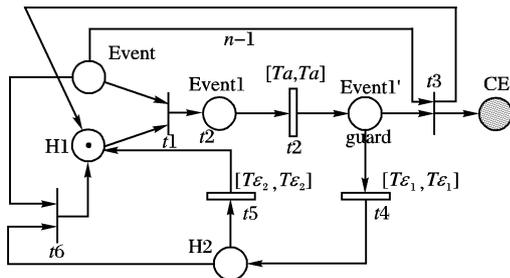


图 6 (Event, n)的时间 Petri 网模型

3.3 利用时间 Petri 网的事件关联检测机制处理告警事件

根据上节中建立的 6 种时间 Petri 网模型,使用其中一种或多个模型的相互组合,可完成事件关联操作类型的处理。

1)使用简单事件时间 Petri 网模型,可将 $P(Event) \in U$ 作为谓词 guard,如果事件 Event 发生,但不满足谓词要求,则过滤掉事件 Event,从而完成事件过滤的关联处理。

2)使用简单事件时间 Petri 网模型,并且在谓词 guard 上设定权值大小, $Q(Event) > C$,当事件 Event 发生,但是 $Q(Event) \leq C$,则不发生事件 Event,从而完成事件升级的关联处理。

3)使用 $Event1 \vee Event2$ 的时间 Petri 网模型,可将 $Event1 \in Event2$ 作为谓词 guard,如果事件 Event1 发生或 Event2 发生,且 $Event1 \in Event2$,则产生事件 Event2,从而完成事件泛化的关联处理^[6]。

4)使用 $Event1 \vee Event2$ 的时间 Petri 网模型,可将 $Event2 \in Event1$ 作为谓词 guard,如果事件 Event1 发生或 Event2 发生,且 $Event2 \in Event1$,则产生事件 Event2,从而完成事件特化的关联处理。

5)使用 $Event1 \wedge Event2$ 的时间 Petri 网模型,如果事件 Event1 和 Event2 同时发生,将事件关联的结果映射为高优先级事件 Event1,从而抑制了低优先级事件 Event2,从而完成事件抑制的关联处理^[6]。

6)使用 $Event1 \rightarrow Event2$ 的时间 Petri 网模型,如果事件 Event1 和 Event2 先后发生,可将事件关联的结果 $Event1 \rightarrow Event2$ 映射为事件 Event3,从而完成事件时序关系的关联处理。

7)使用 $(Event, n)$ 的时间 Petri 网模型,可将发生的多个事件压缩到一个事件中,如果 Event 多次发生,可将其映射为一个事件,从而完成事件压缩的关联处理。

8)使用 $(Event, n)$ 的时间 Petri 网模型,可在谓词上设定事件发生次数,如果事件发生满足设定的阈值,则映射为一个事件,从而完成事件计数的关联处理。

此外,根据这 6 种时间 Petri 网模型或模型间的相互结合,可以很容易地构造更为复杂的事件关联模型,从而完成各种复杂告警事件的关联处理。例如:事件关联 $(Event1 \rightarrow$

$Event2) ! Event3$ 可以分解为两个基本的事件关联: $CE1 = Event1 \rightarrow Event2$ 和 $CE2 = CE1 ! Event3$ 。因此,可以通过这两个基本事件关联模型的组合,可以构造出 $(Event1 \rightarrow Event2) ! Event3$ 的时间 Petri 网模型。

4 实例分析

假设在一个网络环境中,网络管理员定义了以下事件关联规则:

1)如果收到一个链路断开(Link down)事件,并且在接下来的 5 秒内又收到该链路的链路恢复(Link up)事件,则产生一个链路波动(Link bounce)事件,否则将此 Link down 事件发送给网络管理员;

2)如果在 10 秒内收到 5 个 Link bounce 事件,则产生一个链路信号不良(Link Quality low)事件发送给网络管理员。

设事件关联处理器在 $t1$ 时刻收到一个 Link down 事件,随后在 $t2$ 时刻又收到同一位置发出的 Link up 事件。根据事先定义的关联规则 Link down \rightarrow Link up 的 Petri 网模型,可以知道事件关联处理器收到的这两个事件满足规则模型的要求,因此产生一个 Link bounce 事件给网络管理员。

由于该规则模型中没有对时间窗口的大小进行限制,那么事件关联处理器只要收到同一个位置发生的 Link down 事件和 Link up 事件后就会产生一个 Link bounce 事件。而事实上可能会出现以下两种情况:

1)时刻 $t2$ 与 $t1$ 发生的时间间隔刚好在 5 秒内,那么使用关联规则 Link down \rightarrow Link up 的 Petri 网模型可以得到正确的关联结果。

2)时刻 $t2$ 与 $t1$ 发生的时间间隔已经超过了 5 秒,由于关联规则 Link down \rightarrow Link up 的 Petri 网模型中没有对时间窗口大小进行设置,它只是限制了事件发生时间的先后顺序,同样可以得到一个 Link bounce 事件。但是实际上,正确的事件关联结果应该是网络管理员收到 Link down 事件。

采用时间 Petri 网模型时,对于出现的 2 种情况,模型可进行如下处理:

1)可以使用事件关联规则先的时间 Petri 模型,它利用其自身时间变迁的时间变量表示出事件关联时间窗口大小。

设事件关联处理器在 $t1$ 时刻收到一个 Link down 事件,在 $t2$ 时刻又收到同一位置的告警 Link up 事件,并且时刻 $t2$ 与时刻 $t1$ 相差小于 5 秒。根据定义关联规则 Link down \rightarrow Link up 的时间 Petri 网模型,可知事件关联处理器收到的这两个事件满足规则模型要求,即两个事件的产生位置相同,且 Link up 事件须在 Link down 事件发生后的 5 秒内发生。事件关联处理器产生一个 Link bounce 事件给网络管理员。

由此可看出,此模型能正确表示 Link down 事件必须在 Link up 事件之前发生,并且利用其自身时间变迁的时间变量限制了两个事件的发生时间间隔必须在 5 秒内,这样才能得到一个 Link bounce 事件。

2)还可以运用事件关联规则非的时间 Petri 模型,利用其自身时间变迁的时间变量表示出事件关联时间窗口大小。

设事件关联处理器在 $t1$ 时刻收到一个 Link down 事件,在随后的 5 秒内事件关联处理器没有收到来自同一位置的 Link up 事件。根据事先定义的关联规则 Link down ! Link up 的时间 Petri 网模型,可以知道事件关联处理器收到的事件满足规则模型的要求,即收到 Linkdown 事件后的 5 秒内,同一

软件的运行都可以看成单个或多个线程的执行,因此信息管理器为每个线程都动态创建捕获信息存储队列。传感器以线程 ID 为索引把信息投放在某一线程的存储队列中,同一队列内部以到达时间排序。

本文提供两种信息呈现方式:任务视图和时序视图。任务视图的呈现方式中,每个线程看作一个任务,在软件运行的主时间轴下,每个线程在视图中创建各自的运行轨迹,线程自身的运行轨迹由本线程信息存储队列中的捕获信息依次排列而成。时序视图的呈现方式中,软件的主时间轴下,信息呈现引擎按时间顺序从不同的存储队列中取捕获信息,由于存储队列内部是局部排序的,所以呈现引擎仅比较每个队列的首信息。取出首信息后,呈现引擎根据“调用者名称”、“函数执行起始时间”和“函数执行结束时间”在运行轨迹中查找调用函数的运行轨迹(分支),并在其上动态创建新的运行轨迹(分支)示意图。

3 系统运行轨迹分析工具

本文基于的运行路径捕获技术,实现了系统运行轨迹分析工具 SRT(Software Runtime Tracer)。该工具由监控仪表, AOP 织入工具 XWeaver, 方面代码生成器共同组成。用户首先输入观测点声明,然后由代码生成器自动生成路径捕获传感器。传感器通过 XWeaver 与原系统代码结合并生成可运行 EXE。程序运行后, SRT 监控仪表接收系统内部传感器捕获的信息,经综合处理后呈现出软件的运行轨迹,如图 4 所示。

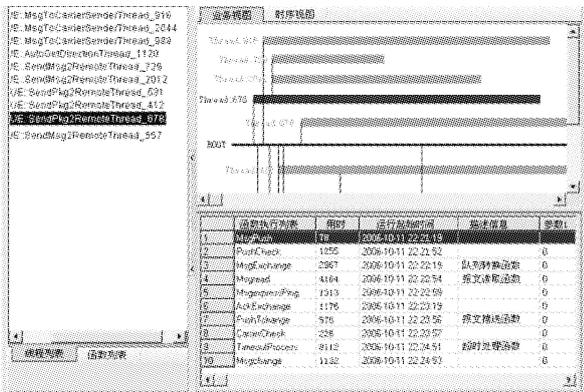


图 4 SRT 的监控仪表

图4左侧树形列表可层次化的显示线程的父子关系及函

数的调用关系,界面的右上方用于显示软件运行的业务视图和时序视图,图中展示了线程的运行轨迹,每条轨迹的长度代表该线程的生命周期,纵线指示了线程的创建关系。当选一中某一线程,视图下方会按时间顺序列出当前线程正在运行和已经运行的函数信息。

借助 SRT,用户可以在原系统之外实现软件的运行轨迹捕获关注点,从而在一定程度上避免了代码混乱和代码纠结,使软件具有更好的模块性、可维护性与可扩展性。如图 4 所示,目前 SRT 已实际应用在一个高可靠网络报文传输服务中,使用结果表明 SRT 能够在服务的运行中提供有效地运行轨迹监测与运行行为监控功能,在故障诊断、性能评估、运行维护中发挥了重要作用,提升了软件的可信性,取得了良好的实用效果。

4 结语

面向方面作为一种全新的软件开发方法,提供了有效的手段来区分、描述和实现软件的非功能方面。从应用层面来看,当今软件开发正朝着以正面功能为核心兼顾非功能方面的方向发展。软件的运行轨迹监测作为一种关键的非功能方面,其重要程度已是人所共识。利用面向方面的软件开发方法,能够使其以一种更加灵活、松散的方式融入系统,软件也因此具有了更好的模块性和扩展性。本文首先研究了基于 AOP 的软件运行轨迹捕获技术,并在此之上实现了通用的系统运行轨迹分析工具 SRT,为软件运行轨迹监测提供了更为灵活、高效的实现手段,在实践中取得了良好效果。

参考文献:

- [1] 杨美清,梅宏,吕建,等. 浅论软件技术发展[J]. 电子学报, 2002, 12(30): 1901 - 1906. .
- [2] 高海洋,陈平. AOP 综述[J]. 计算机科学, 2002, 29(10): 133 - 135.
- [3] MAHRENHOLZ D, SPINCZYK O, SCHRODER-PREIKSCHAT W. Program instrumentation for debugging and monitoring with AspectC++ [C]// Proceedings of the 5th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2002). Washington, DC: IEEE Computer Society, 2002: 249 - 256.
- [4] EDELSTEIN D V. Report on the IEEE STD 1219-1993 standard for software maintenance [J]. ACM SIGSOFT Software Engineering Notes, 1993, 18(4): 94 - 95.

(上接第 1315 页)

置上并没有出现 Link up 事件。最后事件关联处理器将此 Link down 事件发送给网络管理员。

由此可看出,此模型能正确表示 Link down 事件必须在 Link up 事件之前发生,并且利用其自身时间变迁的时间变量限定了两个事件的发生时间间隔必须超过 5 秒,最终才能得到这个 Link down 事件。

5 结语

事件关联技术作为重要的故障定位策略,长期以来一直是研究的热点。基于时间 Petri 网的事件关联检测机制充分考虑了事件关联时间窗口的起始和大小,这样基于时间 Petri 网的事件关联检测机制不仅保留了基于 Petri 网的事件检测机制的优势,同时能很好地弥补基于 Petri 网的事件关联检测机制中出现的关联失效和关联丢失问题,从而能有效地提高

事件关联检测的准确率。

参考文献:

- [1] 唐勇,张欣,周明天. 一种基于 FSM 的告警事件关联方法[J]. 计算机应用研究, 2006, 23(9): 243 - 246.
- [2] 王平,李莉,赵宏. 网络管理中事件关联检测机制的研究[J]. 通信学报, 2004, 25(3): 73 - 81.
- [3] 杨洪涛,王继龙. 网络事件管理系统中关联技术的选择及实现[J]. 计算机工程, 2006, 32(4): 197 - 199.
- [4] 张文雯. 基于事件关联技术的互连网故障诊断研究[D]. 南京: 南京理工大学, 2004.
- [5] WANG JIA-CUN, DENG YI, ZHOU MEN-CHUN. Compositional time petri nets and reduction rules[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2000, 30(4): 562 - 572.
- [6] 张欣. IP 网络监控系统设计及告警事件关联研究[D]. 成都: 电子科技大学, 2005.