

文章编号: 1005-3026(2002)07-0613-04

一种基于事件检测的分布式 网络管理系统模型

李 莉¹, 王 平², 栾贵兴^{1,3}

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110004; 2. 东北大学 软件中心, 辽宁 沈阳 110004;
3. 中国科学院 沈阳计算技术研究所, 辽宁 沈阳 110003)

摘 要: 为了有效地处理大规模网络事件, 克服由于网络事件大量性、多样性和相关性造成的困难, 提出了一种基于事件检测的分布式网络管理方案。在对网络事件分类的基础上, 采用基于策略的层次型事件处理机制, 提高了事件检测效率, 平衡了系统负载, 降低了网络资源的占用率。基于动态时间窗的事件合成方法, 保证了事件检测的及时性和准确性, 为网络的可靠运行提供了保障。

关 键 词: 分布式网络管理; 事件检测; 复合事件; 管理策略; 事件合成; 动态时间窗
中图分类号: TP 393 **文献标识码:** A

网络实体在运行过程中会产生各种各样的网络事件, 这些事件潜在地展示了网络实体的运行状态和行为, 有效地监视这些网络事件是实现网络管理的重要保障。然而在分布式网络环境中, 一个原始事件可以影响许多设备和子系统的操作, 进而引起许多来自于设备和子系统的事件。对大量的事件如果没有相应的控制功能, 管理者将被系统产生的事件风暴淹没, 导致诊断时间过长, 或者忽略一些关键事件, 做出错误判断, 给企业带来很大损失。同时, 网络传输延时造成的事件到达次序颠倒问题大大增加了事件检测的难度。因此, 如何有效地检测网络事件是大规模网络管理面对的一个挑战^[1, 2]。

人们提出了多种分布式管理的解决方案^[3~8], 为大规模网络管理提供了经验。然而, 这些方案都没有有效的大规模网络事件处理机制。为了有效地处理大规模网络事件, 克服由于网络事件大量性、多样性和相关性造成的困难, 本文提出了基于事件检测的分布式网络管理系统模型, 事件检测通过实时地过滤和关联原始事件的方式, 减少了网络事件个数; 事件检测模块分布在系统的各个管理域内, 减少了网络交通, 实现自动网管的功能。本文首先分类网络事件, 然后采用基于策略的层次型事件处理机制, 提高了事件的检测

效率, 减少了管理数据占用的网络资源, 防止了网络事件在整个网络中的扩散。基于动态时间窗的事件合成方法, 克服了事件在网络传输中造成的延时和无序的问题, 保证了相关事件合成的准确性, 为系统的决策提供了可靠的保障。

1 系统管理模型

1.1 系统模型

为了适应网络管理的层次化要求和网络规模可变性和弹性的特点, 本系统采用了层次型结构, 该结构提供了分布式环境下的系统管理方法, 支持多个管理域的分布式网络结构。系统包括 3 种类型的核心组件^[9]: 被管对象管理代理 MA、中间层管理者 M-M 和上层管理系统 MS。

系统采用基于策略的事件处理模式, 每个管理组件包含三个主要模块(如图 1 所示): 策略解释模块、事件处理模块和控制模块。策略解释模块从上层接收管理策略, 由解释器进行处理, 策略解释器一方面将管理策略翻译成控制命令, 传送给控制模块执行; 另一方面, 将策略进一步分解为细粒度的策略, 分配给下层管理者执行。事件处理模块接收下层发送来的事件, 在控制模块的“监督”下, 对事件进行过滤、合成等处理, 并根据控制命

收稿日期: 2002-01-31
基金项目: 国家“九五”重点科技攻关项目(97-769-03-01)。
作者简介: 李 莉(1975—), 女, 山东齐河人, 东北大学博士研究生; 栾贵兴(1937—), 男, 辽宁大连人, 研究员, 博士生导师。
?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

令产生相应的动作。

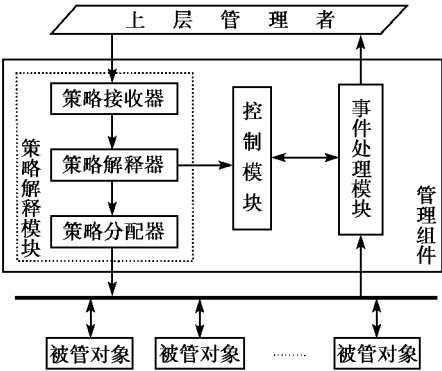


图 1 管理模型
Fig. 1 Management model

系统以策略的形式定义管理任务, 管理员可以通过改变系统执行的策略实现不同的管理功能。策略与系统组件具有相对独立性, 可以被不同的组件重复使用, 实现管理任务的动态配置, 提高网络管理的灵活性。

1.2 网络事件及管理策略

为了克服网络事件的多样性造成的事件检测的困难, 本文根据网络事件的特点对其进行分类, 将网络事件定义成两种类型: 原始事件和复合事件。

定义 1 原始事件: 单一消息产生的事件, 包含 5 个属性:

- Primitive-Event := $\langle \text{event-id} \rangle; \langle \text{source} \rangle;$
 $\langle \text{time-stamp} \rangle; \langle \text{status-value} \rangle; \langle \text{event-pri} \rangle$
- event-id $\in N^+$ — 系统中该事件区别于其他事件的唯一标识;
 - source — 产生此事件的设备信息;
 - time-stamp — 事件产生时间;
 - status-value — 引起该事件的原因;
 - event-pri — 优先级。系统定义了 Immediate 和 Delayed 两个等级, 分别用“0”和“1”表示, Immediate 表示该事件需要立即处理, 其优先级高于 Delayed 型事件。

定义 2 复合事件: 多个原始事件组合成的事件:

- Composite-Event :=
 $\langle \text{primitive-event} \rangle \langle \text{event-op} \rangle \langle \text{primitive-event} \rangle |$
 $\langle \text{primitive-event} \rangle \langle \text{event-op} \rangle \langle \text{composite-event} \rangle$
- Event-Op := $\wedge | \vee | ! | ;$

本文定义了四种具有代表性的“原子复合事件”, 这四种原子复合事件可以任意组合构成复杂的复合事件。描述如下:

$e1 \wedge e2$ — $e1$ 和 $e2$ 都发生

- $e1 \vee e2$ — $e1$ 或者 $e2$ 中至少一个发生
- $e1; e2$ — $e1$ 发生在 $e2$ 之前
- $(e1; e2) ! e3$ — $e1$ 发生在 $e2$ 之前, 期间 $e3$ 未发生

定义 3 管理策略: 是一组描述管理任务和规则的集合, 包含以下属性:

- Policy := $\langle \text{policy-id} \rangle \langle \text{subject} \rangle \langle \text{target} \rangle$
 $\{ \langle \text{trigger} \rangle \} \langle \text{rule-list} \rangle \{ \langle \text{constraint} \rangle \}$
 $\{ \langle \text{exception} \rangle \}$
- policy-id — 策略标识, 在系统中区别其他策略的唯一标识;
 - subject — 策略主体, 是管理策略的执行者, 在网络管理系统中为管理者;
 - target — 策略客体, 是管理策略施加的对象, 在网络管理系统中为被管网络对象;
 - rule-list — 策略规则集, 包含一组规则, 定义了策略的动作行为;
 - trigger — 触发器, 策略被激活的条件, 在系统中有事件和时间两种类型的触发器;
 - constraint — 策略执行的约束条件, 只有当约束条件满足时, 管理策略才会执行;
 - exception — 异常, 定义了管理策略未能顺利实施时的补救措施, 其中“{ }”表示可选项。

2 层次型事件处理机制

网络事件的检测和处理是系统的核心工作, 由各个管理组件中的事件处理模块合作完成。如图 2 所示, 每个事件接收器与一个不同的信息源相对应, 接收该信息源产生的事件, 并把它们存储在事件队列中。事件分配器从各个队列中按照优先级提取网络事件, 传递给相应的事件过滤器进行处理。过滤规则是通过逻辑操作符连接构成的一个逻辑表达式, 如果网络事件满足整个逻辑表

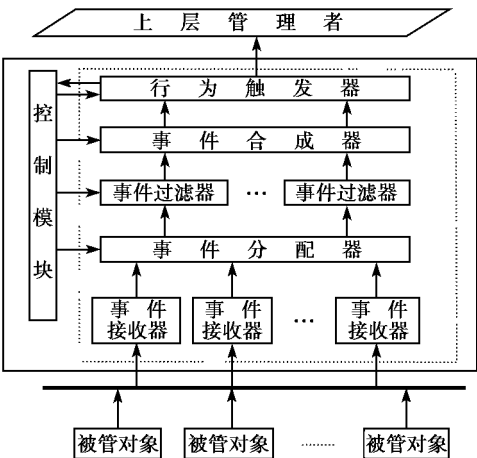


图 2 事件处理模块
Fig. 2 Event process module

达式,则该事件被上传给事件合成器进一步处理,否则系统认为是无效事件,将其删除。事件合成器根据策略定义的时间关系和属性关系将网络事件合成,并触发相应的动作。

系统的每个管理组件都包含一个事件处理模块,这些事件处理模块相对独立的并发运行,不同管理组件的事件处理模块之间具有层次关系。下层的事件处理模块处理原始的网络事件,产生的结果作为上层事件处理模块的输入,依此类推,因此上层处理的网络事件具有更丰富的语义,更能反映全局性的网络运行情况。这种层次关系使系统能够控制管理任务的粒度,实现细粒度的任务配置,提高网络管理的灵活性。

3 网络事件合成

由于网络事件是由地理上分散的不同网络对象产生的,网络传输的不确定性会导致事件延时和到达次序混乱,给网络事件合成带来困难。例如对应复合事件“(e1; e2) ! e3”,事件的产生顺序是: e1, e2, e3, 而接收到的事件序列为 e1, e3, e2, 如果在事件合成时不考虑传输延时,就会产生错误。

解决传输延时问题的一种方法是假设网络传输可能造成的最大延时为 Dtime, 对于每一个接收到的事件在被检测之前先等待一定时延以达到 Dtime, 这样可以在事件检测之前按照事件发生的顺序排列^[9]。对每个事件都增加时延的方法虽然保证了事件排序的正确性,但进一步增加了检测系统的延时,降低了系统的效率。

本系统采用“动态时间窗”的事件合成方法, 实现在事件的检测和合成过程中动态排序。“时间窗”是系统指定的一个时间范围,即此类型网络事件可以容忍的最大延时。在此时间区间内发生的事件对复合事件的合成有效,超出该区间则认为事件失效,将被丢弃。“动态”是指时间窗的绝对大小是固定的,但是时间窗的起始时间和终止时间是根据事件发生的实际情况变化的,即其相对位置是变化的。在描述动态时间窗合成复合事件的算法之前,作以下定义:

CE: 复合事件表达式,用树表示,存储结构采用孩子兄弟表示法(二叉链表表示法);

Toccr(): 此函数返回事件的产生时间;

First(): 此函数返回最先发生事件的产生时间;

Last(): 此函数返回各事件中最后发生事件的产生时间;

TWB: 时间窗的起始时间。

算法如下:

(1) 将复杂的复合事件分解成本系统定义的四种原子复合事件;

(2) 当复合事件的子事件被检测到时,按照系统定义的四种基本类型事件的合成规则和复合事件的树形结构由底到上进行合成运算。合成规则定义如下:

① $CE=e1 \wedge e2$

$Toccr(CE)=First(Toccr(e1), Toccr(e2));$

② $CE=e1 \vee e2$

$Toccr(CE)=Last(Toccr(e1), Toccr(e2));$

③ $CE=e1; e2$

$Toccr(CE)=Toccr(e1);$

④ $CE=\{e1; e2\} ! e3$

$Toccr(CE)=Toccr(e1)。$

⑤ $TWB=First(Toccr(所有基本复合事件))。$

(3) 如果在时间窗范围内检测到该复合事件的相同子事件时,与该子事件相关的树形结构的分枝要按照合成规则重新运算,以重新确定时间窗的起始时间,时间窗的开始时间 = First(Toccr(相同子事件))。

(4) 时间窗的终止时间 TWE= 时间窗的起始时间 TWB+时间窗大小,当当前时间 ≥ TWE 时,如果在该时间窗内复合事件成立则相关的动作被触发,否则相关事件被认为无效,将被丢弃。

下面以事件 CEvent=(A ∧ B); ((C; D) ! E) 为例(时间窗大小= 10)说明动态时间窗合成复合事件的方法:

首先,按照步骤 1 将复合事件 CE 分解成 F=A ∧ B, G=(C; D) ! E, CEvent=F; G。

假设事件的序列为 C(1, 3), E(2, 8), A(3, 9), B(4, 7), D(5, 6), A(6, 11), C(8, 12), D(9, 10), 其中第一个数字代表该事件的产生时间,第二个数字代表该事件被检测到的时间(如图 3 所示)。按照事件被检测到的顺序,系统的工作过程如下(如表 1 所示)。

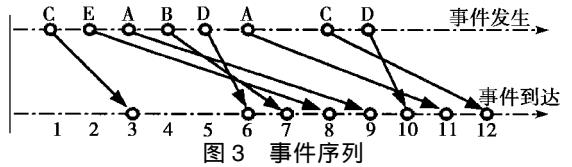


Fig. 3 Event sequence

从复合事件的合成实例可以看到,采用动态时间窗方法合成网络事件可以有效地消除由于网络通信延时造成的次序混乱,正确反映网络事件产生的实际情况,保证了事件合成的正确性。此外,采用该方法,事件排序和合成同时进行,保证

了事件检测的及时性。

表 1 事件合成过程
Table 1 Event composition process

检测时间	到达事件	使用规则	产生事件	TW B	TWE
3	C(1, 3)	④⑤	/	1	11
6	D(5, 6)	④⑤	产生 G	1	11
7	B(4, 7)	①⑤	/	1	11
8	E(2, 8)	④⑤	删除 G	4	14
9	A(3, 9)	①⑤	产生 F	3	13
10	D(9, 10)	④⑤	/	3	13
11	A(6, 11)	①⑤	/	3	13
12	C(8, 12)	①③④⑤	产生 GC	3	13

4 结 束 语

本文提出的基于事件检测的分布式网络管理方案, 具有如下特点:

- (1) 基于策略的管理任务配置机制, 提高了网络管理的灵活性。
- (2) 层次型事件处理机制, 提高了事件的检测效率, 平衡了系统负载, 减少了网络资源占用率。
- (3) 基于动态时间窗的事件合成方法, 实现了事件检测的及时性和准确性。

参考文献:

[1] Martin Flatin J P. A survey of distributed enterprise network

and systems management [J] . *Journal of Network and Systems Management*, 1999, 7(1): 9— 26.

[2] Erfani S, Lawrence V B, Malek M, et al. Network management: emerging trends and challenges[J] . *Bell Labs Technical Journal*, 1999, 4(4): 3— 22.

[3] Goldszmidt G, Yemini S, Yemini Y. Distributed management by delegation[A] . *Proceeding of the CAS Conference*[C] . Anchorage, LCDCS Press, 1991. 347— 359.

[4] Goldszmidt G. On distributed system management [A] . *Proceeding of the CAS Conference*[C] . Toronto: CSREA Press, 1993. 637— 647.

[5] Hong J W K, Kong J Y. Web-based intranet services and network management[J] . *IEEE Communication Magazine*, 1999, 35(10): 100— 110.

[6] Kahani M. Decentralized approaches for network management [J] . *ACM Computer Communiation Review*, 2000, 27(3): 36— 47.

[7] Brunner M, Stadler R. Service management in multiparty active networks[J] . *IEEE Communications Magazine*, 2000, March: 144— 151.

[8] Han S J, Lee J H. A mobility management using dynamic updates of domain name in mobile computing environment [A] . *International Conference on Internet Computing*[C] . Las Vegas: CSREA Press, 2001. 444— 450.

[9] 王平, 李莉, 赵宏. 一种基于多级 Manager/ Agent 结构的智能网络管理系统[J] . *东北大学学报(自然科学版)*, 2001, 22(6): 604— 607.

(Wang P, Li L, Zhao H. A Multi-management/ agent based intelligent network management system[J] . *Journal of Northeastern University (Natural Science)*, 2001, 22(6): 604— 607.)

Event Monitoring Based Management Model for Distributed Network

LI Li¹, WANG Ping², LUAN Gui-xing^{1,3}

(1. School of Information Science & Engineering, Northeastern University, Shenyang 110004, China; 2. Software Center, Northeastern University, Shenyang 110004, China; 3. Shenyang Institute of Computing Technology, CAS, Shenyang 110003, China. Correspondent: LI Li, E-mail: roselily @sict.ac.cn)

Abstract To efficiently handle large-scale network events and solve the problems caused by events' largeness, events' variety and events' relevance, a event-monitoring-based model was presented for managing large scale distributed networks. Two types of events primitive event and composite event were defined to monitor network efficiently. Policy-based hierarchical event-processing mechanism can efficiently distribute the loads of the system and improve the reliability of the system. Event composition approach named Dynamic Time Windows can deal with out-of-order sequences of event arrivals due to communication delays in an efficient manner. This significantly improves the reliability of the system.

Key words distributed network management; event monitor; composition event; management policy; event composition; dynamic time window

(Received January 1, 2002)