网格监控体系结构的一种可执行性模型*)

刘显明　李师贤
(中山大学计算机科学系　广州 510275)

摘　要　为网格监控体系结构建立可执行性模型有助于网格监控系统的服务质量提升。因为网格环境的动态性和不可靠性,所以对网格监控体系结构建模时要从可用性和性能两方面综合考虑。讨论系统可用性、响应时间分布、事件丢失概率、公平性等问题。用随机 Petri 网建立网格监控体系结构的可执行性模型并讨论了模型的应用。网格监控体系结构的系统模型有一个关键特性:事件发布信息和监控事件数据由不同通道传递,在模型中重点关注这一特性。

关键词　随机 Petri 网,可执行性模型,系统性能评价,网格监控体系结构

A Performability Model for Grid Monitoring Architecture

LIU Xian-Ming　LI Shi-Xian
(Department of Computer Science, Sun Yat-Sen University, Guangzhou 510275)

Abstract　To develop the performability model for grid monitoring architecture is helpful to enhance Quality of Service of grid monitoring system. Because of the dynamic and unreliable characters of grid environment, we must consider availability and performance at the same time when we develop the system model for grid monitoring architecture. We discuss the issues including availability of system, response time distribution, probability of events loss, fairness. Finally we develop the performability model for grid monitoring architecture using stochastic petri nets and discuss the application of this model. There is a key feature in system model of grid monitoring architecture: event publication information and monitoring event data must be transferred in different channels, analysis of performability model focuses on this feature.

Keywords　Stochastic petri nets, Performability model, Performance evaluation, Grid monitoring architecture

1　引言

全球网格论坛(Global Grid Forum, GGF)性能工作组制定的网格监控体系结构(Grid Monitoring Architecture, GMA)[1]目前已经被一些网格监控系统所采用或考虑[2,3],GMA 的目标是为网格监控提供标准并使得现有系统互操作。对于一个实际的网格监控系统来说,GMA 的内容还远远不够,比如:GMA 没有为网格监控系统提供服务质量(Quality of Service, QoS)控制方面的建议。基于以上考虑,综合考虑可用性和性能两方面内容,为 GMA 建立可执行性模型成为一项有意义的工作。对 GMA 可执行性模型进行分析的结果可以对网格监控系统的 QoS 控制起到积极作用。

系统的可执行性是指系统的可用性和性能的组合[4]。如果不考虑网格环境中的系统失效和恢复行为,仅仅从纯性能的角度去分析 GMA 系统模型会显得过于乐观;而如果只考虑网格环境中的可用性[5],对于强调低延迟、高速率的监控信息来说则意义不大。只有把可用性和性能结合起来考虑的可执行性模型适合用来分析 GMA。研究人员在系统的可执行性建模和分析方面已经开展了许多工作[6~10]。其中 S. Ranami 和 K. Trivedi 等人经过对 CORBA 事件服务和通知服务的性能分析[8,9],给出了分布系统消息服务的一种可执行性建模框架[10]。进一步研究了对实时系统响应时间分布建模的技术[11]。本文在文[10]的基础上,考虑了网格环境和监控信息的特点,使用随机 Petri 网开发了 GMA 的一种可执行性模型,并讨论了模型的应用。文[1]中指出,虽然 GMA 的系统模型是参照 CORBA 事件服务的事件通道模型定义的,但是 GMA 的系统模型与 CORBA 事件通道模型存在一个关键差异:CORBA 事件通道模型中事件发布信息(Event Publication Information, EPI)和监控事件数据(Monitoring Event Data, MED)由同一个事件通道传递;但是 GMA 中监控事件数据通过监控事件通道传递,而事件发布信息通过目录服务处理,所以在 GMA 的可执行性模型中必须考虑两条通道的有效协同工作,以及目录服务对系统可执行性的影响。

本文第 2 节对问题进行描述,从可用性和性能两方面分析了 GMA 系统模型的特点,讨论了系统可用性、事件丢失概率、响应时间分布和公平性等问题;第 3 节使用随机 Petri 网(Stochastic Petri Nets, SPN)[12]建立 GMA 的可执行性模型,讨论模型的参数设置和求解;第 4 节是模型的应用,对前面建立的可执行性模型进行瞬态和稳态分析;最后是结论。

2　问题描述

在为 GMA 建立可执行性模型的时候,首先必须理解 GMA 的逻辑结构,分析它的特点。由于网格环境的动态和不可靠性,必须分析系统可用性;由于监控信息具有生命周期短、更新频繁以及随机性的特点,必须分析事件丢失概率、响应时间分布;由于两条通道协同工作,必须分析 QoS 策略的公平性。

2.1　网格监控体系结构

GMA 的逻辑视图如图 1 所示。GMA 将监控数据以带时间戳的事件方式传递,一个事件是一个特定结构的数据集合。

为了把数据发现和数据传输分离,用作事件发布信息的

*)本文工作受到广东省科技计划项目基金资助(No. 2003A1030403)。刘显明　博士生,主要研究领域为 Petri 网,系统性能评价,网络计算;李师贤　教授,博士生导师,主要研究领域为分布式计算,形式语义学。

元数据必须被抽取出来放到一个可统一访问的位置-目录服务。生产者作为监控事件数据的源点，消费者作为监控事件数据的接收点。生产者向目录服务注册它所能提供的监控事件数据，消费者向目录服务注册它所需要的监控事件数据，经过目录服务处理后，在生产者和消费者之间直接传输监控事件数据。很显然，监控事件数据构成了大部分的通信流量，同时事件发布信息丢失造成的损害肯定超过监控事件数据丢失所造成的损害。通过设置适当的 QoS 策略可以保证 EPI 通道和 MED 通道的有效协同。这些都是在对 GMA 构建可执行性模型时需要仔细考虑的问题。

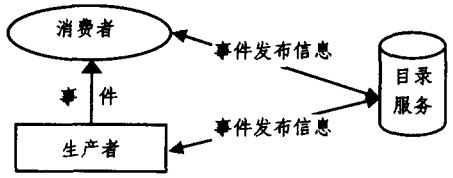

图 1　GMA 的逻辑视图

2.2　系统可用性

在文[10]中只考虑了生产者、消费者和消息服务的可用性，在本文中必须考虑生产者、消费者、监控服务、目录服务以及相应网络资源的可用性。令 N_1 表示生产者，N_2 表示消费者，N_3 表示监控服务，N_4 表示目录服务。C_1，C_2，C_3，C_4 分别表示与 N_1，N_2，N_3，N_4 相连接的网络资源，那么系统整体可用性定义如下：

$$A(GMA)=\prod_{i=1}^{k}A(N_i)A(C_i) \tag{1}$$

其中 A(GMA)为系统整体可用性，$A(N_i)$为 N_i 的可用性，$A(C_i)$为 N_i 相应的网络资源可用性，$k=4$。假设 N_i 的失效率为 γ_{N_i}，修复率为 τ_{N_i}，C_i 的失效率为 γ_{C_i}，修复率为 τ_{C_i}，那么

$$A(GMA)=\prod_{i=1}^{k}\frac{\tau_{N_i}\tau_{C_i}}{(\gamma_{N_i}+\tau_{N_i})(\gamma_{C_i}+\tau_{C_i})} \tag{2}$$

2.3　响应时间分布

监控信息具有时效性很强的特点，信息从生产者传递到消费者花费的时间超过一定的时限之后，就会失去作用。比如，监控信息的处理中经常需要解决的问题是：确定有多大比例的监控事件在 1 秒钟的时限内不能够传送到消费者端？所以在 GMA 的可执行性模型中仅仅考虑平均响应时间是不够的，还要考虑响应时间分布。令 $F_{H\text{-}EPI}(t)$为 EPI 通道的驻留时间分布，$F_{S\text{-}EPI}(t)$为 EPI 通道中事件成功传递的时间分布，$F_{D\text{-}EPI}(t)$为 EPI 通道中事件丢弃的时间分布，则可定义公式如下：

$$F_{H\text{-}EPI}(t)=F_{S\text{-}EPI}(t)+F_{D\text{-}EPI}(t) \tag{3}$$

令 $F_{H\text{-}EPI|S}(t)$为 EPI 通道中事件成功传递的条件驻留时间分布，$F_{H\text{-}EPI|D}(t)$为 EPI 通道中事件丢弃的条件驻留时间分布，$\pi_{S\text{-}EPI}$为 EPI 通道的事件成功传递概率，$\pi_{D\text{-}EPI}$为 EPI 通道的事件丢弃概率，则有：

$$F_{H\text{-}EPI}(t)=\pi_{S\text{-}EPI}F_{H\text{-}EPI|S}(t)+\pi_{D\text{-}EPI}F_{H\text{-}EPI|D}(t) \tag{4}$$

通过求解响应时间分布，可以进一步求解事件在一定时限内未能成功传递的概率，令 $P_{DV\text{-}EPI}$表示 EPI 通道中事件在时限 d 内未成功传递的概率，则有：

$$P_{DV\text{-}EPI}(d)=1-F_{S\text{-}EPI}(d)/F_{S\text{-}EPI}(\infty)$$
$$=1-F_{S\text{-}EPI}(d)/\pi_{S\text{-}EPI} \tag{5}$$

因为 EPI 通道和 MED 通道的各种指标的求解方式是一样的，所以在本文中只写出 EPI 通道各种指标的求解公式，由此可以类推出 MED 通道各种指标的求解公式。例如，从式(5)中可以推得 MED 通道中的事件在时限 d 内未成功传递的概率：

$$P_{DV\text{-}MED}(d)=1-F_{S\text{-}MED}(d)/F_{S\text{-}MED}(\infty)$$
$$=1-F_{S\text{-}MED}(d)/\pi_{S\text{-}MED} \tag{6}$$

2.4　事件丢失率

GMA 的可执行性模型从本质上讲是一种分布式系统的消息传递模型，所以用事件丢失率作为模型的主要性能指标是合适的。令 LP_{EPI}表示 EPI 通道的事件丢失概率，LR_{EPI}表示 EPI 通道的事件丢失速率，PR_{EPI}表示 EPI 通道的事件产生速率，则有：

$$LP_{EPI}=LR_{EPI}/PR_{EPI} \tag{7}$$

求解事件丢失速率的时候，必须考虑系统资源的可用性和事件传递的时限。令 $LR_{EPI|FD}$表示各部分都未失效时 EPI 通道的事件队列溢出所产生的事件丢失速率，$LR_{EPI|PF}$表示生产者失效时 EPI 通道的事件丢失速率，$LR_{EPI|CF}$表示消费者失效时 EPI 通道的事件丢失速率，$LR_{EPI|MSF}$表示监控服务失效时 EPI 通道的事件丢失速率，$LR_{EPI|DSF}$表示目录服务失效时 EPI 通道的事件丢失速率，$LR_{EPI|NF}$表示网络失效时 EPI 通道的事件丢失速率，$LR_{EPI|DV}$表示未在时限内成功传递事件时 EPI 通道的事件丢失速率，则有：

$$LP_{EPI}=LR_{EPI|FD}+LR_{EPI|PF}+LR_{EPI|CF}+LR_{EPI|MSF}+LR_{EPI|DSF}+LR_{EPI|NF}+LR_{EPI|DV} \tag{8}$$

下面具体讨论各种情况下事件丢失率的求解：

$$LR_{EPI|FD}=\prod_{i=1}^{4}A(N_i)A(C_i)\lambda_{EPI}P_{EPI-Full} \tag{9}$$

其中 λ_{EPI}为 EPI 通道的事件产生速率，$P_{EPI\text{-}FULL}$为事件队列溢出的概率。

$$LR_{EPI|PF}=\prod_{i=1}^{4}A(N_i)A(C_i)\gamma_{N_1}\bar{N}_{EPI}+(1-A(N_1))\cdot\prod_{i=2}^{4}A(N_i)\prod_{i=1}^{4}A(C_i)\lambda_{EPI} \tag{10}$$

$$LR_{EPI|CF}=\prod_{i=1}^{4}A(N_i)A(C_i)\gamma_{N_2}\bar{N}_{EPI}+A(N_1)(1-A(N_2))\cdot\prod_{i=3}^{4}A(N_i)\prod_{i=1}^{4}A(C_i)\lambda_{EPI} \tag{11}$$

其中 $\bar{N}_{EPI}$为稳定状态下 EPI 通道事件队列期望长度。

$$LR_{EPI|MSF}=\prod_{i=1}^{4}A(N_i)A(C_i)\gamma_{N_3}\bar{N}_{EPI}+\prod_{i=1}^{2}A(N_i)(1-A(N_3))A(N_4)\prod_{i=1}^{4}A(C_i)\lambda_{EPI} \tag{12}$$

$$LR_{EPI|DSF}=\prod_{i=1}^{4}A(N_i)A(C_i)\gamma_{N_4}\bar{N}_{EPI}+\prod_{i=1}^{3}A(N_i)(1-A(N_4))\prod_{i=1}^{4}A(C_i)\lambda_{EPI} \tag{13}$$

因为网格环境的不可靠性，所以对网络失效时 EPI 通道事件丢失率 $LR_{EPI|NF}$的求解必须分别考虑 C_1，C_2，C_3，C_4 失效的情况，则有：

$$LR_{EPI|NF}=\prod_{i=1}^{4}A(N_i)A(C_i)\sum_{i=1}^{4}\gamma_{C_i}\bar{N}_{EPI}+\prod_{i=1}^{4}A(N_i)(((1-A(C_i))\prod_{i=2}^{4}A(C_i)+A(C_1)(1-A(C_2))\prod_{i=3}^{4}A(C_1)+\prod_{i=1}^{2}A(C_i)(1-A(C_3))A(C_4)+\prod_{i=1}^{3}A(C_i)(1-A(C_4)))\lambda_{EPI} \tag{14}$$

因为监控信息的时效性，所以需要分析未在时限 d 内成功传递事件时 EPI 通道的事件丢失率

$$LP_{EPI|DV}=\prod_{i=1}^{4}A(N_i)A(C_i)(1-F_{S\text{-}EPI}(d)/\pi_{S\text{-}EPI})((1-P_{EPI\text{-}FULL})\lambda_{EPI}+\gamma_{N_1}\min(\bar{N}_{EPI},\mu_{EPI}/\tau_{N_1})) \tag{15}$$

其中 μ_{EPI} 为 EPI 通道的事件处理速率。

将式(9)～(15)代入式(8)可求解 EPI 通道事件丢失速率 LR_{EPI}，再根据式(7)求解 EPI 通道事件丢失概率 LP_{EPI}，同理求解 MED 通道事件丢失概率 LP_{MED}。

2.5 QoS 策略的公平性

为了实现 EPI 和 MED 两条通道的有效协同，必须考虑公平性。基于丢失率的比例公平性原则[13]要求服务类的性能指标(如事件丢失概率)应该正比于相应的区分参数。令 $\bar{l}_i$ 为通道 i 在稳定状态下事件丢失概率，σ_i 为丢失率区分参数，F_i 为公平性指标，则有：

$$F_i=\bar{l}_i/\sigma_i \tag{16}$$

如果所有的 F_i 相等，就说系统的 QoS 策略是公平的。

由于响应时间分布对于监控信息的传递来讲非常重要，因此除了文[13]中的基于丢失率的比例公平性原则之外，本文还使用一种基于响应时间分布的比例公平性原则。令 $F_{S_i}(d)$ 表示通道 i 中的事件在时限 d 内成功提交的概率，则有：

$$F_i=Fs_i(d)/\sigma_i \tag{17}$$

3 GMA 的可执行性模型

本部分采用随机 Petri 网作为建模工具，分别建立了生产者、消费者、监控服务和目录服务的可用性模型，并建立了 EPI 和 MED 两条通道的性能模型，综合得到 GMA 的可执行性模型。为了建模响应时间分布，引入了“标记客户方法”[14]。通过设置相关变迁的可实施谓词和可实施概率，以及相关弧的弧权变量来表示各种 QoS 策略组合。

3.1 随机 Petri 网模型

本文建立的 GMA 可执行性模型如图 2 所示。该模型体现了 GMA 中事件发布信息和监控事件数据并行处理，不走一个通道的特点。EPI 通道和 MED 通道之间的并发控制则通过控制位置来实现。也正是因为 Petri 网建模并发系统的优势，本文才选择随机 Petri 网作为建模工具。

下面具体描述一下图 2 的内容。位置 p_{ai} 表示节点资源 N_i 以及相关的网络资源 C_i 处于可用状态，初始状态包含一个标记。时间变迁 t_{nfi} 表示节点资源 N_i 以平均速率 λ_{nfi} 发生失效，一旦 t_{nfi} 实施，则将一个标记从可用状态位置 p_{ai} 移入节点失效状态位置 p_{nfi}。时间变迁 t_{nri} 表示节点资源 N_i 以平均速率 μ_{nri} 进行修复。同理，时间变迁 t_{cfi} 和 t_{cri} 分别表示网络资源 C_i 的失效和修复，位置 p_{cfi} 为网络资源 C_i 的失效状态位置。

当位置 p_{a1} 中包含一个标记时，说明生产者处于可用状态，时间变迁 t_{e1} 以平均速率 λ_{e1} 产生事件发布信息；当位置 p_{a1} 为空时，说明生产者处于失效状态，则令 t_{e1} 的速率为 0。位置 p_{e1} 中的标记表示 EPI 通道等待队列的队头元素，位置 p_{e1} 的容量为 1。瞬时变迁 t_{e2} 的实施将队头元素放入位置 p_{e2}。位置 p_{e2} 则表示 EPI 通道等待队列的主体，容量为 k_1-2，这里令 k_1 为 EPI 通道等待队列的长度。瞬时变迁 t_{e3} 的实施将位置 p_{e^2} 中的元素放入队尾位置 p_{e3}，位置 p_{e3} 的容量为 1。当位置 p_{a3} 中含有 1 个标记时，说明监控服务可用，瞬时变迁 t_{e4} 的实施表示根据某种 QoS 策略从位置 p_{e3} 中选择一个标记进入位置 p_{e4} 中进行处理。在模型中设置 t_{e4} 和 t_{m4} 的可实施概率可以表示先进先出(First-In-First-Out，FIFO)和基于优先级(Priority，PR)的提交策略。当位置 p_{a3} 为空时，令 t_{e4} 和 t_{m4} 不可实施。设定控制位置 p_{c3} 中只有一个标记可以保证任何时刻两个并发的通道中只能有一个事件正在被处理，这反映了实际系统的物理特性。当位置 p_{a4} 中包含一个标记时，变迁 t_{e5} 表示以速率 μ_{e5} 对事件发布信息进行处理，物理意义是使用目录服务；当位置 p_{a4} 为空时，变迁 t_{e5} 的速率为 0。位置 p_{e5} 中是已经处理完的事件发布信息，通过瞬时变迁 t_{e6} 清除。

瞬时变迁 t_{e7} 表示当一定条件满足时丢弃队头位置 p_{e1} 中的标记，瞬时变迁 t_{e8} 表示当一定条件满足时丢弃队尾位置 p_{e3} 中的标记，瞬时变迁 t_{e9} 表示当一定条件满足时丢弃位置 p_{e4} 中的标记，通过设置 t_{e7}、t_{e8} 和 t_{e9} 的可实施谓词，则可以表示先进先出和后进先出(Last-In-First-Out，LIFO)的丢弃策略。瞬时变迁 t_{e10} 表示当一定条件满足时丢弃位置 p_{e3} 中的标记，而瞬时变迁 t_{e11} 表示当一定条件满足时丢弃位置 p_{e4} 中的标记。结合控制位置 p_{c1} 和 p_{c2} 设置 t_{e10}，t_{e11}，t_{m10}，t_{m11} 的可实施谓词以及可实施概率，同时设置弧权变量 v，可以表示基于优先级的丢弃策略。

最后，位置 p_{e6}、p_{e7} 和变迁 t_{e12} 的作用是建模标记客户的响应时间分布。

3.2 标记客户方法

为了建模响应时间分布，引入“标记客户方法”，位置 p_{e6} 中存放的标记表示一个标记客户，从位置 p_{e1}、p_{e2}、p_{e3}、p_{e4} 到变迁 t_{e12} 的禁止弧保证了将 EPI 通道等待队列中所有的事件和正在处理的事件都处理完以后再处理标记客户。当 t_{e12} 实施以后，一个标记移入位置 p_{e7}，说明这个标记客户被成功处理。必须对模型作暂态分析才能求出标记客户被处理的响应时间分布，进一步解出在时限 d 内标记客户被成功传递的概率。

MED 通道的变迁 t_{m1} 到 t_{m12} 和位置 p_{m1} 到 p_{m7} 的意义可以参照 EPI 通道的变迁 t_{e1} 到 t_{e12} 和位置 p_{e1} 到 p_{e7} 进行理解，限于篇幅，就不再赘述。

3.3 QoS 策略的设置

接下来讨论一下影响 EPI 通道和 MED 通道各种性能指标的 QoS 策略。在本部分主要考虑不同提交策略和丢弃策略对 GMA 可执行性模型的影响。提交策略有 2 种：FIFO、Priority；丢弃策略有 3 种：FIFO、LIFO、Priority；所以模型中共有 6 种 QoS 策略组合。

下面分别讨论相应于这 6 种 QoS 策略组合的各个变迁的可实施谓词和可实施概率以及弧权变量的设置，其中 k_1、k_2 分别表示 EPI 通道和 MED 通道的缓冲区长度，r_1、r_2 分别表示 EPI 通道和 MED 通道的优先级：

(1)FIFO×FIFO：

变迁 t_{e4}、t_{m4} 的可实施概率　$P(t_{e4})=P(t_{m4})=0.5$

变迁 t_{e7}、t_{m7} 不可实施。

变迁 t_{e8} 的可实施谓词

$y_{e8}:(M(P_{e1})+M(P_2)+M(P_{e3})>k_1)\wedge(M(P_{e4})=0)$。

变迁 t_{e9} 的可实施谓词

$y_{e9}:(M(P_{e1})+M(P_{e2})+M(P_{e3})>K_1)\wedge(M(P_{e4})=1)$。

变迁 t_{m8} 的可实施谓词

$y_{m8}:(M(P_{m1})+M(P_{m2})+M(P_{m3})>k_2)\wedge(M(P_{m4})=0)$。

变迁 t_{m9} 的可实施谓词

$y_{m9}:(M(p_{m1})+M(p_{m2})+M(p_{m3})>k_2)\wedge(M(p_{m4})=1)$。

变迁 t_{e10}、t_{e11}、t_{m10}、t_{m11} 不可实施，弧权变量 $v=0$。

(2)FIFO×LIFO：

变迁 t_{e4}、t_{m4} 的可实施概率　$P(t_{e4})=P(t_{m4})=0.5$

变迁 t_{e7} 的可实施谓词

$y_{m7}:(M(P_{e1})+M(P_{e2})+M(P_{e3})>k_1)$。

变迁 t_{m7} 的可实施谓词

$y_{m7}:(M(P_{m1})+M(P_{m2})+M(P_{m3})>k_2)$。

变迁 t_{e8}、t_{e9}、t_{m8}、t_{m9} 不可实施。

变迁 t_{e10}、t_{e11}、t_{m10}、t_{m11} 不可实施，弧权变量 $v=0$。

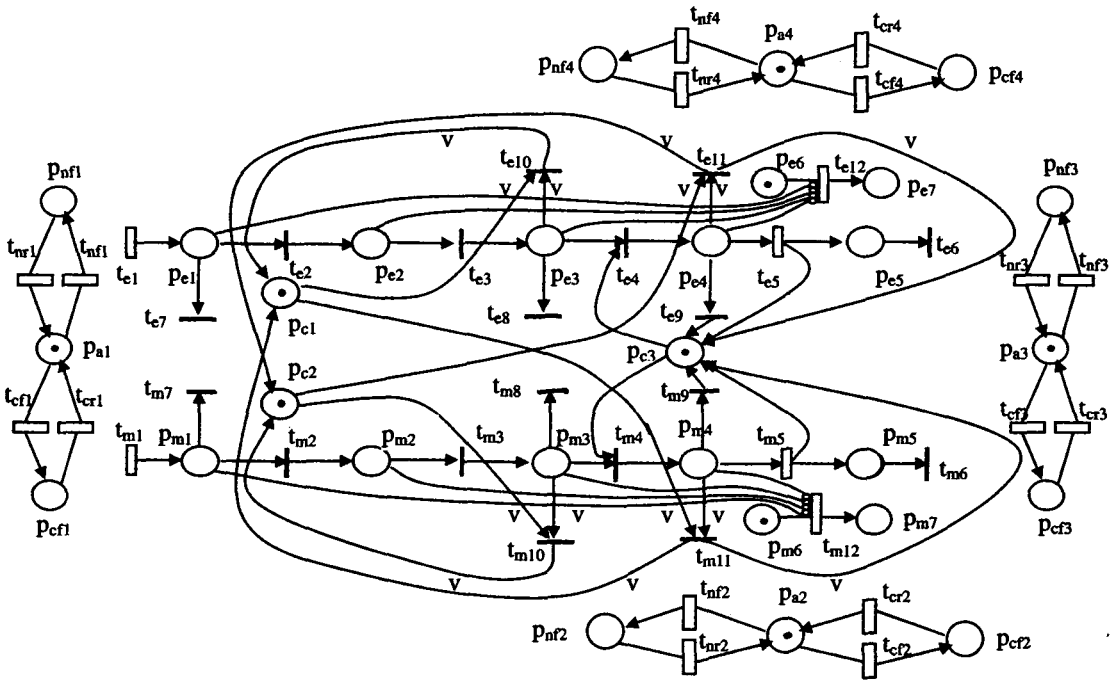

图 2　GMA 的 SPN 模型

(3)FIFO×Priority：

变迁 t_{e4}、t_{m4} 的可实施概率 $P(t_{e4})=P(t_{m4})=0.5$。

变迁 t_{e7}、t_{m7} 不可实施。

变迁 t_{e8} 的可实施谓词

$y_{e8}:\neg(M(P_{m4})=1)\wedge((M(P_{e1})+M(p_{e2})+M(P_{e3}))>k_1)$。

变迁 t_{e9} 不可实施。

变迁 t_{m8} 的可实施谓词

$y_{e8}:\neg(M(P_{e4})=1)\wedge((M(P_{m1})+M(p_{m2})+M(P_{m3}))>k_2)$。

变迁 t_{m9} 不可实施。

变迁 t_{e10} 的可实施谓词

$y_{e10}:(M(p_{m4})=1)\wedge(M(p_{e1})+M(p_{e2})+M(p_{e3})>k_1)$。

变迁 t_{e11} 的可实施谓词

$y_{e11}:(M(p_{e4})=1)\wedge(M(p_{m1})+M(p_{m2})+M(p_{m3})>k_2)$。

变迁 t_{m10} 的可实施谓词

$y_{m10}:(M(p_{e4})=1)\wedge(M(p_{m1})+M(p_{m2})+M(p_{m3})>k_2)$。

变迁 t_{m11} 的可实施谓词

$y_{m11}:(M(p_{m4})=1)\wedge(M(p_{e1})+M(p_{e2})+M(p_{e3})>k_1)$。

弧权变量 $v=1$。

变迁 t_{e10}、t_{m11} 的可实施概率

$P(t_{e10})=r_2/(r_1+r_2)$；$P(t_{m11})=r_1/(r_1+r_2)$。

变迁 t_{e11}、t_{m10} 的可实施概率

$P(t_{e11})=r_2/(r_1+r_2)$；$P(t_{m10})=r_1/(r_1+r_2)$。

(4) Priority×FIFO：

除了变迁 t_{e4}、t_{m4} 的可实施概率为 $P(t_{e4})=r_1/(r_1+r_2)$，$P(t_{m4})=r_2/(r_1+r_2)$ 之外，其余设置同方案 1。

(5)Priority×LIFO：

除了变迁 t_{e4}、t_{m4} 的可实施概率为 $P(t_{e4})=r_1/(r_1+r_2)$，$P(t_{m4})=r_2/(r_1+r_2)$ 之外，其余设置同方案 2。

(6)Priority×Priority：

除了变迁 t_{e4}、t_{m4} 的可实施概率为 $P(t_{e4})=r_1/(r_1+r_2)$，$P(t_{m4})=r_2/(r_1+r_2)$ 之外，其余设置同方案 3。

4　模型的应用

本部分在一个具体的应用环境中使用前面开发的可执行性模型，来求解系统可用性、响应时间分布、事件丢失率、公平性等各种指标。所有实验数据都是使用 SPNP 软件[15]对模型求解而获得的，SPNP 是一个经典的随机 Petri 网软件包，目前在很多研究单位得到应用。根据对有关论文和实验结果[8~9,16]的分析，确定模型的参数如表 1 和表 2。设定变迁的延时服从指数分布。性能模型求解的硬件环境为 PIII 667，512M SDRAM，软件工具为 SPNP 6.0，模型的每次求解过程会产生 966418 个状态。下面讨论对模型进行稳态和瞬态分析所取得的实验结果。

表 1　失效率和恢复率参数的设置

i	$1/\lambda_{nfi}$	$1/\mu_{nri}$	$1/\lambda_{efi}$	$1/\mu_{cri}$
1	10000	100	2000	10
2	10000	100	2000	10
3	50000	200	20000	10
4	50000	200	20000	10

表 2　EPI 通道和 MED 通道参数的设置

i	λ_{i1}	k_i	μ_{i5}	μ_{i12}	Pr_i
e	10～100	25	100	μ_{e5}	3
m	100～1000	50	1000	μ_{m5}	1

4.1 系统可用性分析

这里讨论系统整体可用性，综合考虑式(2)中各种失效和恢复情况，对模型进行求解，稳定状态下得出的实验结果为：

A(GMA)=96.073%

4.2 响应时间分布及公平性分析

首先分析各种 QoS 策略组合对响应时间分布造成的影响。设定 $\lambda_{e1}=100$ 事件/秒，$\lambda_{m1}=1000$ 事件/秒，$\mu_{e5}=100$ 事件/秒，$\mu_{m5}=1000$ 事件/秒。对模型做瞬态分析，以式(3)中 *EPI* 通道事件成功传递的时间分布 $F_{S\text{-}EPI}(t)$ 为例讨论实验结果。从图 3 中可见在系统负载很重的情况下，选择不同的 QoS 策略组合将会对系统的响应时间分布产生较大影响。

接下来分析两条通道的协同工作，设定 $\lambda_{e1}=50$ 事件/秒，$\lambda_{m1}=500$ 事件/秒，$\mu_{e5}=100$ 事件/秒，$\mu_{m5}=1000$ 事件/秒。提交和丢弃策略都是 FIFO。对模型做瞬态分析，以两条通道中事件成功传递的时间分布 $F_S(t)$ 为例讨论实验结果。从图 4 中可见两条通道中事件成功传递的的时间分布差别很大。例如在 0.02 秒时限内，EPI 通道中事件成功传递的概率远小于 MED 通道中事件成功传递的概率，由式(17)可知此时策略组合 FIFO×FIFO 的公平性不好；但当时限 d 取 0.09 秒时，策略组合 FIFO×FIFO 的公平性很好。

从上面的分析可知，根据具体应用场合中对监控事件传递时限的要求，来灵活选择各种 QoS 策略组合将能够有效地提高系统的可执行性。

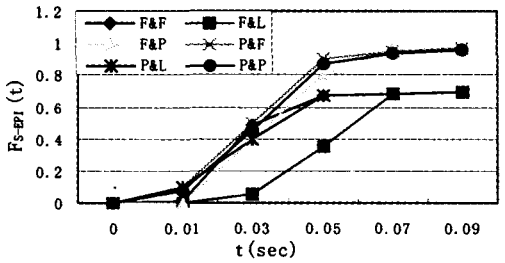

图 3 QoS 策略组合的响应时间分布

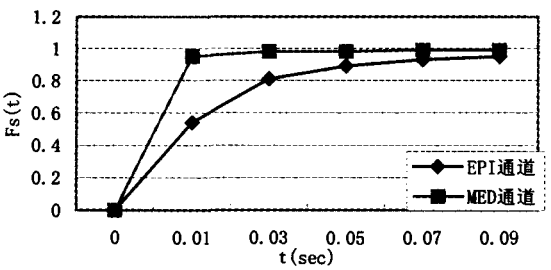

图 4 两条通道的响应时间分布

4.3 事件丢失概率及公平性分析

首先分析当系统负载逐渐增加时，各种 QoS 策略组合对事件丢失率的影响。设定 λ_{e1}、λ_{m1} 为变量，$\mu_{e5}=100$ 事件/秒，$\mu_{m5}=1000$ 事件/秒。将式(8)中的 LR_{EPI} 和同理求得的 LR_{MED} 代入模型求解，由式(7)有 $LP_{EPI}=LR_{EPI}/\lambda_{e1}$，$LP_{MED}=LR_{MED}/\lambda_{m1}$。对模型做稳态分析，以 LP_{MED} 为例讨论实验结果。从图 5 中可见基于优先级丢弃策略的两种 QoS 策略组合可以获得相对较好的可执行性。

接下来分析两条通道的协同工作。设定比例值 $k=\lambda_{e1}/10=\lambda_{m1}/100$ 表示系统负载的变化。对模型做稳态分析，以 QoS 策略组合 FIFO×Priority 的 LP_{EPI} 和 LP_{MED} 为例讨论实验结果。从图 6 中可见在系统负载较轻的时候，LP_{EPI} 和 LP_{MED} 曲线几乎重合，由式(16)可知此时策略组合 FIFO×Priority 的公平性较好；当系统负载逐渐加重，策略组合 FIFO×Priority 的公平性则变得很差。

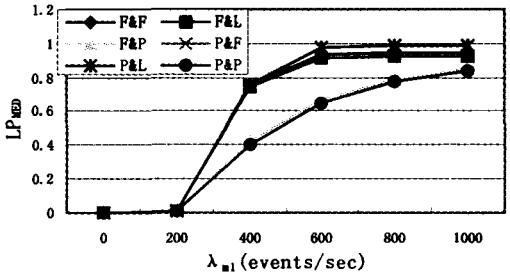

图 5 QoS 策略组合的事件丢失概率

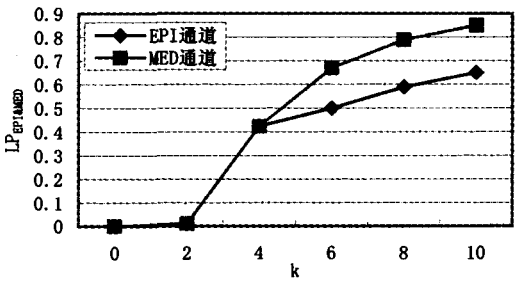

图 6 两条通道的事件丢失概率

结论 本文分析了 GMA 的可用性和性能两方面问题，提出并构造了 GMA 的一种可执行性模型。这一模型综合考虑了生产者、消费者、监控服务、目录服务等因素对系统可执行性的影响。由于网格的不可靠特点，因此需要分析可用性；由于监控事件的低延迟、高速率要求，因此需要分析响应时间分布；在此基础上，最后以事件丢失率来衡量 GMA 的可执行性。为了让 GMA 中的两条通道有效协同工作，分析了公平性问题。

该模型可用于分析某种基于 GMA 的网格监控系统设计方案在应用环境中的性能和可用性特点，也可以作为某种基于 GMA 的网格监控系统实际使用中 QoS 策略选择的判断手段。

参 考 文 献

1 Tierney B, Aydt R, Gunter D, et al. A Grid Monitoring Architecture(2002). GGF Performance Working Group, January 2002. http://www.-didc. lbl. gov/ GGF-PERF/GMA-WG/papers/GWD-GP-16-1. pdf/

2 Baker M A, Smith G. GridRM: An Extensible Resource Monitoring System. In: Proc. of the IEEE Cluster Computing Conf. 2003, Hong Kong, Dec. 2003. 221～232

3 Li C, Xu Z W, Lin G Z, Li W. A Grid monitoring system using LDAP. Journal of Computer Research and Development, 2002, 39(8): 930～936

4 Meyer J F. On evaluating the performability of degradable computer systems. IEEE Transactions on Computers, 1980, 29(8): 720～731

5 Li C J, Li D S, Xiao N, Yang X J. A Messurement Model for the Availability of Applications in Computational Grid. Journal of Computer Research and Development, 2003, 40(12): 1704～1709

6 Haverkort B, Marie R, Rubino G, Trivedi K S. Performability Modeling Tools and Techniques. John Wiley & Sons, Chichester, England, 2001

(下转第 99 页)

表 3 模式树查询(QA 对应 XMark 集,QN 对应 Nasa 集)

	查询模式树(标签后加 * 的结点需要返回)	查询结果数
QA1	site/regions//item/description/text *	7729
QA2	site//item/description/parlist/listitem/text * [bold *][keyword *]/emph *	7450
QA3	site/regions/asia/item * [//mail/text [//bold]//keyword]/description * /text//emph	68
QN1	dataset//para/project//xlink:simple/@role *	5905
QN2	dataset//title//author * [middleName *][lastName *]/affiliation/xlink:simple/@role *	1634
QN3	dataset//title * //footnote * [para/address [homePage *]/street *]//author/lastName *	1210

BLAS[4]的 P-label 对于实验中选用的 XMark 数据集有 84^{12} 种可能的取值,需要 8 个 byte 才能表示一个 XML 结点的 P-label,索引一共需要 8M 的空间。而对于 Nasa 数据集,有 141^{28} 种可能取值,需要 26 个 byte 才能表示,一共需要 21M 空间,接近于原文档的大小。与之相比,JoinGuide 分别只需要 2k 和 51k 的空间。这说明,JoinGuide 需要的空间远小于文[4]中的 P-label,更具有实用性。

表 4 TwigStack, BLAS, merge-match, merge-union, multi-merge 五种算法执行查询的运行时间和读入结点个数

运行时间(ms)/读入结点数	TwigStack	BLAS	merge-match	match-union	multi-merge join
QA1	18500/ 85807	730 / 7730	80 / 7729	80 / 7729	80 / 7729
QA2	4200 / 232043	1340 / 20574	410 / 20573	430 / 20573	340 / 20573
QA3	1320 / 254310	550 / 118543	50 / 4901	210 / 98235	50 / 4901
QN1	1660 / 142714	930 / 58431	90 / 5905	90 / 5905	90 / 5905
QN2	1610 / 216608	900 / 85849	140 / 7846	130 / 7846	100 / 7846
QN3	550 / 61221	380 / 23446	90 / 2548	140 / 10358	40 / 2548

结论 本文对模式树匹配问题提出了一种新的结构索引 JoinGuide,并且提出了使用 JoinGuide 的三种查询匹配算法,实验证实了本文中算法的优越性。

以后,我们打算使用局部相似性来改进 JoinGuide,使它避免对路径信息过于细化,对复杂结构的 XML 也能保持合适的大小,同时能够提高在索引上的查询速度。

参 考 文 献

1 Al-Khalifa S, et al. Structural joins: A primitive for efficient XML query pattern matching. In ICDE, 2002

2 Bruno N, et al. Holistic twig joins: Optimal XML pattern matching. In SIGMOD, 2002

3 Chen Q, et al. D(k)-index: An adaptive structural summary for graph-structured data. In SIGMOD, 2003

4 Chen Y, et al. BLAS: An efficient XPath processing system. In SIGMOD, 2004

5 Goldman R, Widom J. Dataguides: Enabling query formulation and optimization in semistructured databases. In VLDB, 1997

6 IBM XML Generator. http://www.alphaworks.ibm.com/tech/xmlgenerator

7 Jiang H, Wang W, Lu H. Holistic twig joins on indexed XML documents. In VLDB, 2003

8 Jagadish H V, et al. TAX: A Tree algebra for XML. In DBPL, 2001

9 Kaushik R, et al. Exploiting local similarity for efficient indexing of paths in graphstructured data. In ICDE, 2002

10 Milo T, Suciu D. Index structures for path expressions. In ICDT, 1999

11 NASA XML Group. Available at: http://xml.gsfc.nasa.gov/.

12 XMARK. The XML-benchmark project, April 2001. http://monetdb.ewi.nl/xml/index.html

13 Zhang C, et al. On supporting containment queries in relational database management systems. In SIGMOD, 2001

(上接第 67 页)

7 Kuntz M, Siegle M. A stochastic extension of the logic PDL. In: Sixth Int. Workshop on Performability Modeling of Computer and Communication Systems (PMCCS6), Monticello (IL), 2003. 58～61

8 Ramani S, Trivedi K S, Dasarathy B. Performance analysis of the CORBA Event Service using stochastic reward nets. In: Proc. of the 19th IEEE Symposium on Reliable Distributed Systems, Nurnberg, Germany, 2000. 238～247

9 Ramani S, Trivedi K S, Dasarathy B. Performance analysis of the CORBA notification service. In: Proc. 20th IEEE Symposium on Reliable Distributed Systems, New Orleans, USA, 2001. 227～236

10 Ramani S, Go℃seva-Popstojanova K, Trivedi K S. A framework for performability modeling of messaging services in distributed systems. In: the 8th IEEE Intl. Conf. on Engineering of Complex Computer Systems, Greenbelt, MD, 2002. 229～238

11 Trivedi K S, Ramani S, Fricks R. Recent Advances in Modeling Response-Time Distributions in Real-Time Systems. In: Proc. of the IEEE 91(7), 2003. 1023～1037

12 Lin C. Performance Evaluation of Computer Network and Computer System. Press of Tsinghua University, Beijing. 2001, ISBN 7-302-04265-5

13 Lin C. Quality of Service of Computer Networks. Press of Tsinghua University, Beijing, 2004. ISBN 7-302-08076-3

14 Mainkar V, Trivedi K S. Transient analysis of real-time systems using deterministic and stochastic petri nets. In Quality of Communication-Based Systems. Dordrecht, Netherlands: Kluwer, 1995. 69～84

15 Ciardo G, Muppala J, Trivedi K. SPNP: Stochastic Petri Net Package. In: Proc. of Third International Workshop on Petri Nets and Performance Model, Kyoto, Japan, June 1989. 142～151

16 Liu X M, Li S X. Performance Modeling and Analysis of a Grid Monitoring Service. To appear in Journal of Information and Computational Science