基于对象的存储系统对象迁移策略*)

谭支鹏 冯 丹
(华中科技大学教育部外存储国家专业实验室 武汉 430074)

摘 要 对象存储系统是下一代网络存储重要组织模式,对象管理是对象存储系统的关键技术之一。本文对对象存储系统中的对象迁移策略进行了系统的研究,提出了可变阈值和阈长的动态反馈调整模型,以此来确定存储对象迁移的时机和目标对象存储设备的选取。另外,本文还利用 Petri网 工具对对象存储系统存储对象的迁移进行建模分析,给出了存储对象迁移控制的 Petri网模型。但是随着存储系统存储节点的无限增加,Petri网模型将会无限复杂和庞大。为了减少 Petri网模型控制模型的复杂度,又引入有色 Petri网 理论,实现了对存储对象迁移控制 Petri网模型的简化。该模型同样很好地实现了对象存储系统中对象迁移控制的建模分析。这些研究对我们在建立对象存储系统时起到了很好的帮助作用。

关键词 对象存储系统,动态反馈调整模型,对象迁移,Petri网

Strategies of Object Migration for Object-based Storage System

TAN Zhi-Peng FENG Dan
(National Information Storage Sysetm Lab, Huazhong University of Science & Technology, Wuhan 430074)

Abstract Object-based storage(OBS) is an important storage model for storage systems in the next generation Internet, where object management is one of the key components. In this paper we study an object migration strategy for object-based storage systems that dynamically adjusts the variable-threshold through a feedback model to determine the best OBS device for object migration that minimizes migration times. A Petri-net model is developed to analyze the performance of the proposed strategy. In this analytical model, a migration-controllable model is employed to reduce the complexity of the original Petri-net model. This study provides useful insight into the design of OBS systems and can help to make sensible design decisions.

Keywords Object-based storage system, Dynamic-feedback adjusted model, Object migration, Petri net

1 前言

随着计算机信息技术的不断发展,信息量正在呈几何级数增长,信息存储越来越成为人们所研究和关注的焦点问题。目前存在的3种主要存储结构为直接附加存储(DAS)、网络附加存储(NAS)、存储区域网(SAN)[2,3],这3种存储技术是信息存储技术发展的结果,它们在一定程度上缓解了信息存储给计算机技术带来的压力。

DAS是传统的基本存储模式,由单个或多个存储设备构成,并直接连接到主机系统的扩展接口上。它的优点是连接简单、性能高、易于使用,但是难以扩展容量,适合有限共享的场合使用,并不适合现在海量信息的存储。

NAS是一个专用的文件服务器,通过IP网络协议给连接到网络上的用户提供基于文件的数据共享服务。它易于实现跨平台的文件共享,支持不同平台的用户通过标准的接口实现共享服务器上的存储文件,但是它不适合块级数据的应用,而且服务器很容易成为性能的瓶颈。

SAN是数据存储为中心,采用可伸缩的交换网络结构代替传统的总线结构,所有主机系统和存储设备之间都是通过高速网络相连,提供内部任意节点之间的多路可选择的数据交换。它具有可伸缩、高带宽、高可靠性、高容错能力的优点,适合性能、对存储设备可伸缩性要求高的应用,但是SAN只能提供设备共享,不能提供数据共享。

正是由于这3种存储技术存在的这些局限性,使得它们在一定程度上还不能完全满足信息存储技术的要求,于是计算机界提出了基于对象的存储技术,当前对象存储技术正是信息存储领域研究的热点。国际信息技术标准委员会(INCITS)下属的T10技术委员会专门成立了一个技术工作小组,负责OSD的技术标准制定工作[2,3];Gibson教授在Ganegis Mellon大学组织NASD项目;Scott Brandt教授领导的一个研究小组在California大学存储系统研究中心正致力于基于OSD(object storage device)的大规模可伸缩存储系统研究等等。一些国际大IT企业,如IBN、HP、ENC,也启动了相关对象存储系统的研究。

对象存储系统的基本组成单位是对象,对象管理的好坏直接关系到对象存储系统的性能的优劣,也是对象存储系统成功建立的基础。对象迁移对实现存储系统的负载平衡、提高系统的性能有着很重要的作用[4~6]。本文对对象存储系统的对象迁移策略进行了系统研究,建立了对象迁移目标地址的基于可变阈值和阈长的动态反馈调整模型,并用 Petri 网和

*)本文受国家重点基础研究发展计划项目973资助,项目名称:下一代互联网信息存储的组织模式和核心技术研究,项目编号:2004CB318201。
谭支鹏 博士研究生,主要研究兴趣为信息存储、信息安全、数据库技术等;冯 丹 教授、博士导师,主要研究兴趣为信息存储、计算机系统结构、磁盘阵列等。

有色 Petri 网工具对对象的迁移控制进行了模拟,对对象管理策略的研究和分析起到明显作用。

2 存储对象的迁移策略

对象迁移是系统的重组、故障下重构、负载均衡的需要,尤其是在网络存储系统中,存储对象事先并不能准确地知道它存储在哪个 OSD(object storage device,存储对象设备)上系统的性能最优,因而在网络对象存储系统中存储对象的迁移应该是经常的。对象存储系统中的对象迁移是由对象本身和 OSD 本身的存储能力所决定的。一些存储节点随存储对象的使用、存储节点出现故障的暂时脱离系统、故障恢复后的重新连上系统以及技术的更新、系统存储能力的加强等等,都将会导致系统中的存储对象发生迁移。利用存储对象的迁移可以灵活地调整系统的规模,优化系统的结构,提高系统的性能。那么,究竟存储对象如何迁移才能使系统的性能得到加强、系统的结构得到优化呢?下面先给出存储对象迁移的理论模型。

2.1 可变阈值和阈长的动态反馈调整模型

定义 1 在一个对象存储系统中,对于每一个 OSD,可以用一个正整数 C 来抽象地描述其存储能力,称 C 为该对象的存储能力值;存储能力 C 的最小阈值用正整数 $T(T<C)$ 来描述,而阈长用正整数 a 来描述。于是,当前 OSD 的实际负载 L 和存储能力阈值 T 与阈长 a 之间存在如下函数关系:

$$F(L,T,a)=\begin{cases}0, & \text{当 } L<T-a \text{ 时,表示当前对象为空闲对象;}\\ 1, & \text{当 } L\in[T-a,T+a] \text{ 时,表示当前对象为负载适中对象;}\\ 2, & \text{当 } L>T+a \text{ 时,表示当前对象为超载对象。}\end{cases}$$

如图 1 所示,OSD 的当前状态处于图中所描述的 3 个区域之一。当它的负载 L 小于 $T-a$ 时,处于空闲区域,表示该 OSD 负载较少,可以成为其他对象的接受者,接受新的存储对象或者来自其他 OSD 的迁移对象;当 OSD 处于负载适中区域时,表示该 OSD 的负载适中,该 OSD 可以接受新的存储对象,但是它不适合接受其他 OSD 迁移而来的存储对象,而且没必要把存储对象迁移到其他 OSD 上;当 OSD 负载超负荷时,表示该 OSD 负载很重,需要其它 OSD 来分担存储,需要通过存储对象向其他 OSD 的迁移,来减轻负载提高存储系统的存储效率,达到存储系统结构的优化。

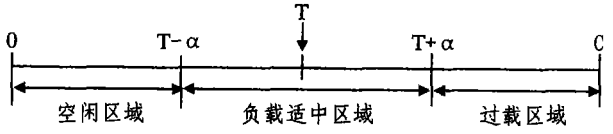

图 1 阈值和阈长的基本原理

定义 2 在具有 n 个 OSD 的对象存储系统中,定义系统整体负载平衡指标为:

$$V=\sum_{i\in N}(r_i-r_0)^2 \qquad (1)$$

其中 $r_i=\dfrac{L_i}{C_i}$, $r_0=\dfrac{\sum_{i\in N}L_i}{\sum_{i\in N}C_i}$

将上式中的 r_i 和 r_0 分别称为 OSD 的负载率和存储系统的整体负载率。很明显,r_0 表示具有 N 个存储节点的整体负载水平。r_0 如果很大,则表明系统整体负载能力强;r_0 如果很小,则表明系统整体负载能力弱。它是一个将系统视为整体的负载率指标。我们可以通过对 V 值的计算来评估存储系统的负载平衡状况。

2.2 对象迁移目标 OSD 的选择

所谓存储对象迁移实质上是指存储对象的复制。存储对象的复制,可以是部分复制,也可以是整个对象完全复制。设对象存储设备 i 上的当前负载为 L_i,将要复制到对象存储设备 i 上存储对象为 P_k,那么设该存储对象将要消耗的存储资源为 $POSL_k$,对象存储设备 i 已消耗的存储资源为 $POPL_k$。对象存储设备 i 能不能接受迁移存储对象的条件是:

$$L_i+(POSL_k+POPL_k)>T_i+a_i \qquad (2)$$

一旦对象存储设备 i 决定接受迁移存储对象,那么存储对象迁移就必须执行,以达到存储系统中各个对象存储设备负载的平衡。根据定义 2,应选择能使 V 值达到最小的对象存储设备作为存储对象迁移的目标 OSD,为此我们提出如下确定对象迁移目标 OSD 的条件:

$$\mathrm{Min}\sum\left(\frac{WS_i*Q+L_i}{C_i}-T_o\right)^2 \qquad (3)$$

$$s.t.\quad \sum WS_i=1$$

$$WS_i*Q+L_i\leqslant C_i \qquad \text{其中 } Q=POSL_k,\ WS_i=0,1$$

很明显,对象迁移将带来 OSD 负载的重新分配。我们通过表达式(3)来考察经过这一次对象迁移后存储对象系统负载平衡所带来的影响,来确定迁移的目标地址。这是一个典型的 0-1 规划问题,可以用贪心法或分支限界法等方法求解。

3 存储对象迁移的 Petri 网分析

3.1 存储对象迁移控制的 Petri 网模型

在对象存储系统中,通过元数据服务器对存储对象迁移实现协作控制,能使系统效率得到提高,存储资源利用率得到加强,同时可以缩短服务的相应时间。网络存储系统中,大量的 I/O 往往是并发、实时的,因此这些并发实时的 I/O 肯定存在冲突,合理的存储对象迁移是降低冲突、提高系统效率和资源利用率的有效途径。前面我们已经讨论了如何正确地选择一个目标 OSD 来供存储对象迁移,但是如何控制存储对象的迁移又成了一个摆在我们面前的问题。鉴于 Petri 网理论在描述系统并发性、实现系统建模等方面的优越性,我们利用 Petri 理论来分析研究对象存储系统中的对象迁移,实现存储对象迁移的有效控制。

定义 3 基本 Petri 网定义为四元组,$N=\{S,T;F,M_0\}$,其中 $S=\{s_1,s_2,\cdots,s_n\}$ 为事件(状态)集;$T=\{t_1,t_2,\cdots,t_n\}$ 为变迁集;$F\subseteq(S\times E)\cup(E\times S)$ 为网的流关系;$M_0:S\rightarrow\{0,1\}$ 为初始标记。记 $^{\cdot}t=\{s|(s,t)\in F\}$ 为变迁的前置集,$t^{\cdot}=\{s|(t,s)\in F\}$ 为变迁的后置集。Petri 网的位置中可能含有托肯(token),变迁在一定的条件下可以被激活。设 t 在标识 M 下是授权的,则 t 可以激活。变迁激活后产生新的标识 M',可以记作 $M[t>M'$。根据基本 Petri 网定义,下面给出存储对象迁移的 Petri 网模型定义。

定义 4 具有 n 个存储节点的存储对象迁移控制 Petri 网模型定义为:$CPN(n)=(P,T,F,M_0)$,其中库所集合 $P=\{P_a,P_b,P_c,P_d,P_{ij}\}$,即库所集中包括这几类库所;变迁集合 $T=\{T_a,T_b,T_{ij}\}$,即变迁集中包括这几类变迁。其中 $i=1,2,\cdots,n$,表示 n 个存储节点,$j=0,1,2,3$,表示存储节点的 4 种不同状态或存储节点的 4 种不同操作请求。F 表示网的流关系;M_0 为初始标记。这些类型的库所和变迁的含义如下:

P_a:对象迁移控制中心;

P_b:对象迁移调度执行策略;

P_c: 存储对象迁移结果;

P_d: 目标存储节点, 没有托肯, 表示目标存储节点还没选定;

P_{i0}: 存储节点 i 的初始状态;

P_{i1}: 存储节点 i 上存储对象需要迁移, 向元数据服务器提交的请求;

P_{i2}: 存储节点 i 的存储对象迁移的相关准备工作;

P_{i3}: 存储节点 i 的存储对象迁移获得元数据服务器认可。

T_a: 元数据服务器对成员请求的调度;

T_b: 对象迁移完成后, 根据存储对象的存储能力修改目标 OSD 的相关信息;

T_{i0}: 存储节点 i 上存储对象产生对象迁移请求;

T_{i1}: 存储节点 i 向元数据服务器提交操作请求;

T_{i2}: 存储节点 i 存储对象迁移目标 OSD 的选取;

T_{i3}: 由存储对象迁移的目标 OSD 和调度策略执行有效的对象迁移。

P_{i0}中都有一个托肯, 表示任意存储节点均可向元数据服务器发送请求。初始状态下 P_d 没有托肯, 表示初始状态下, 还没存储对象迁移的请求, 目标 OSD 还没有选定。

存储对象迁移控制的 Petri 网模型参考图 2。这里我们仅以两个存储节点, 外加一个元数据服务器为例来加以说明, 模型图中 P_a 表示存储系统元数据服务器, P_{1j}、P_{2j} 分别表示两个存储节点, 其中 j= 1、2、3。

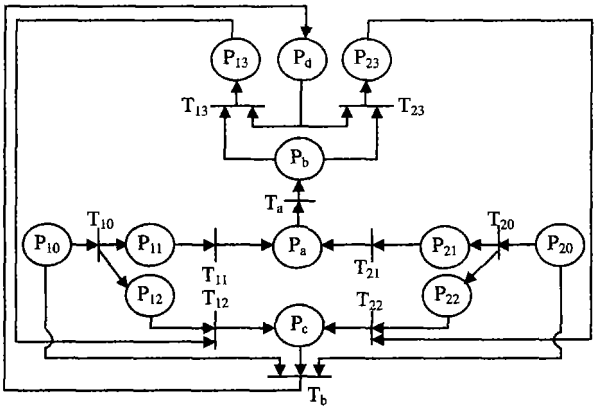

图 2　存储对象迁移控制的 Petri 网模型

下面针对存储对象迁移控制的 Petri 网模型, 说明对象存储系统中存储对象迁移的实现过程。存储节点 P_{i0}上的存储对象需要迁移时, 通过变迁 T_{i0}, 产生存储对象迁移请求 P_{i1}, 同时做好存储对象迁移的必要准备 P_{i2}。对象迁移请求经过变迁 T_{i1}, 传到元数据服务器上, 在元数据服务器库所 P_a 上建立存储节点的请求队列, 保证所有的对象迁移请求都能够被接受。然后选择相应的调度算法, 由变迁 T_2 对这些请求实现调度, 并将调度结果存放于库所 P_b。所有存储节点根据自身的负载情况以及存储对象迁移后的状况, 通过可变阈值和阈长的动态反馈调整模型, 促使变迁 T_3 发生, 确定目标 OSD。根据库所 P_b 的调度策略, 并结合表示目标存储节点的库所 P_d 的目标 OSD 地址, 发生变迁 T_{i3}, 确定对象的迁移路径。在对象迁移条件库所 P_{i2}成熟后, 发生变迁 T_{i2}, 实现对象的真正迁移。

很显然, 存储对象迁移控制的 Petri 网模型是动态的。通常, 对存储系统中的任何存储节点来说, 关于存储对象的迁移都有两个基本的操作, 即对象迁移请求和对象迁移的执行。在执行这些操作时, 通常都会出现下面这些问题:

1) 各个存储节点同时向元数据服务器提出迁移请求造成的请求冲突。

2) 各个存储节点在执行存储对象迁移时的执行冲突, 即一个存储对象同时要向多个存储节点上迁移。

3) 存储节点上的扰动现象, 即一个存储对象刚刚被迁移走又被迁移回来。

4) 一个存储对象的频繁迁移, 即一个存储对象从一个节点迁移到另外一个存储节点后, 马上又向下一个节点进行迁移。

这些问题在存储系统中都是不允许的。在存储对象迁移控制的 Petri 网模型中, 这些问题都可以得到有效的控制和解决。出现的第一个问题可以由库所 P_a 的功能来解决; 后面 3 个问题可以根据可变阈值和阈长的动态反馈调整模型, 由变迁 T3 的发生得到有效的控制。

对象迁移控制的 Petri 网模型有这样一些特点:

1) 在元数据服务器的作用下, 存储系统中的存储节点增减不会影响存储对象在其他节点上的迁移策略。

2) 可直观地掌握存储对象迁移的动态执行以及迁移的效率。

3) 保证每次迁移都能够正常、正确地执行。

3.2　存储对象迁移控制的有色 Petri 网模型

存储对象迁移控制的 Petri 网模型对存储节点不多的存储系统来说, 实现系统的建模和分析是非常有效的。但是, 随着系统中存储节点的不断增多, 图 2 的 Petri 模型中库所和变迁的数量将大量增加, 模型将会变得非常的复杂和庞大, 反而不利于系统的分析和建模。因此, 通过着色对图 2 的 Petri 模型进行改造, 使之成为有色 Petri 模型, 如图 3 所示。

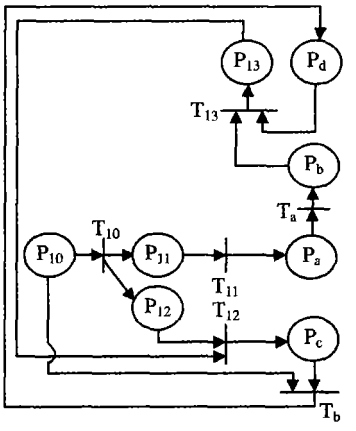

图 3　存储对象迁移控制的有色 Petri 网模型

在有色 Petri 模型中, 我们用一个库所 P_{10}来表示存储系统中的所有存储节点, 然后对库所 P_{10}的托肯实施着色, 这些不同颜色的托肯表示不同的存储节点。下面给出存储对象迁移控制的有色 Petri 网模型定义。

定义 5　存储对象迁移控制的有色 Petri 网模型定义为: $CCPN = (P, T, C, F, M_0)$, P、T、F、M_0 定义同定义 3, 颜色集 $C = \{C_0, C_1, C_2, \cdots, C_n\}$ 表示对象存储系统中的存储节点。

库所 P_d 初始状态下没有托肯, 表示对象迁移的目标 OSD 还没选定; 库所 P_{10}存放表示存储节点相关信息的有色托肯。在初始状态 M_0 情况下, 库所 P_{10}有 n 个托肯, 它们分

(下转第 68 页)

测率,前者在训练和检测时间上有所缩短,对于有的攻击类型(U2R、R2L)的检测,检测速度提高得比较显著。注意到U2R和R2L攻击记录在整个数据集中数量较少,为了提高v-SVM异常检测对U2R和R2L攻击检测的性能,在构造训练和测试集时,我们对U2R和R2L攻击记录做了多次复制,以混进正常样本集,这样可以得到比较均衡的数据集。

结论　在基于SVM的异常入侵检测中,提高检测速度对于实时入侵检测是十分重要的。提高检测速度有几种途径:一是对硬件设备进行升级,如增大存储器的容量、提高CPU的处理速度;一是研究二次规划的快速算法;再一个是对网络连接记录这种高维数据进行特征选择和降维,以减小数据处理的规模。本文着眼于第三种提高检测速度的方法,实现了一种粗糙集属性约简和支持向量机分类相结合的网络异常入侵检测方法。首先利用粗糙集属性约简的方法压缩数据空间,然后采用v-SVM两分类方法处理约简和正规化后的数据。实验结果表明,与基于全部属性的v-SVM两分类方法相比,该方法具有与之相当的分类精度,但有效地缩短了分类时间,减少了存储空间。

参考文献

1 Lee W, Stolfo S J. Data mining approaches for intrusion detection [A]. In: Proceedings, Seventh USENIX Security Symposium, San Antonio, TX, 1998

2 Jha S, Tan K, Maxion R. Markov chains, classifiers and intrusion detection [A]. In: The 14th IEEE Computer Security Foundations Workshop, Canada, 2001. Proceedings, Seventh USENIX Security Symposium, San Antonio, TX, 1998

3 Balajinath B, Raghavan S. Intrusion detection through learning behavior model [J]. Computer Communications, 2001, 24(12): 1202~1212

4 Forrest S, Hofmeyr S A. Computer Immunology [J]. Communications of the ACM, 1997, 40(10): 88~96

5 Vapnik V N. The nature of statistical learning theory [M]. New York: Springer, 1995

6 Kim D S, Park J S. Network-based intrusion detection with support vector machines [A]. In: KahngH-K. Ed. ICOIN 2003, LNCS 2662, 2003. 747~756

7 Sohn T, Seo J T, Moon J S. A study on the covert channel detection of TCP/IP header using support vector machines [A]. In: Qing S, Gollmann D, Zhou J. Eds. ICOIN 2003, LNCS 2836, 2003. 313~324

8 Hu W J, Liao Y H, Vemuri V R. Robust anomaly detection using support vector machines [J]. IEEE Transactrions on Pattern Analysis and Machine Intelligence. (in press)

9 王国胤. Rough集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001

10 Schölkopf B, Smola A, Williamson R C, et al. New support vector algorithms. Neural Computation [J]. 2000, 12(5): 1207~1245

11 Lee W, Stolfo S. A framework for constructing features and models for intrusion detection systems [J]. ACM Transactions on Information and System Security, 2000(3): 227~261

12 http://rosetta.lcb.uu.se/general/

13 Sung A H, Mukkamala S. Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks [A]. In: Proceedings of 2003 Symposium on Application and the Internet(SAINT'03), 2003

14 Wilson D R, Martinez T R. Improved heterogeneous distance functions. Journal of Artificial Intelligence Research [J], 1997, 6(1): 1~34

15 李元诚, 方廷健. 一种基于粗糙集理论的SVM短期负荷预测方法[J]. 系统工程与电子技术, 2004, 26(2): 187~190

16 Chang Chih-Chung, Lin Chih-Jen. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

(上接第63页)

别着不同的颜色。当某个存储节点上有存储对象要迁移时,其相应的有色托肯将激活变迁T_{10},请求元数据服务器判别如何调度,同时根据当前所有存储节点的对象负载情况,确定目标OSD,并实现对象的迁移。

结束语　对象存储系统是下一代互联网存储模式的关键技术,而对象存储系统中对象管理,其成功与否在一定的程度上关系着整个存储系统的成功建立与否。网络对象存储系统中存储节点的动态变化是一项很重要的特点,这样的特点决定了对象存储系统中对象的迁移是经常性的,而且也是必须的。因为存储对象原来的节点暂时不存在了,但是整个系统不能因为存储节点的暂时不在而不能正常运行。为了解决这样的问题,对象迁移是一种非常好的解决办法。本文对对象迁移的策略进行了理论上的分析,通过可变阈值和阈长的动态反馈调整模型确定对象存储系统中目标OSD的选择依据,并引入面向Petri网的相关技术,对对象迁移进行建模和分析,对存储对象的迁移实现了有效的控制。

参考文献

1 Kangasharju J, Roberts J, Ross K W. Object replication strategies in content distribution networks. Computer Communications, 2002, 25(4): 376~383

2 http://www.intel.com/labs/storage/osd/tech.htm, 2003-03

3 Braam P J. The Lustre Storage architecture [EB/OL]. http://www.lustre.org/docs/lustre.pdf, 2003-03

4 Laoutaris N, Zissimopoulos V, Stavrakakis I. Joint object placement and node dimensioning for internet content distribution. Information Processing Letters, 2004, 89(6): 273~279

5 Krishnan P, Raz D, Shavit Y. The cache location problem. IEEE/ACM Transactions on Networking, 2000, 8(5): 568~581

6 Ranganathan K, Foster I. Identifying dynamic replication strategies for a high performance data grid. Proceedings of the International Workshop on Grid Computing, Denver, Colorado. 2001

7 Bowden F D J. Modeling time in PetriNets. The Second Australia, Japan Workshop on Stochastic Models, Gold Coast, 1996

8 Zhou Long-xiang. C2POREL22: A distribute drelational database management system on microcomputer network. Science, Series A, 1986, XXIX(1): 78~91

9 Tlin J. A Petrinet based integrated control and scheduling scheme for flexible manufacturing cells [J]. Computer integrated Manufacturing System, 1997, 10(2): 109~122

10 Wu C H, Lee S J. Enhanced high-level Petri nets with multiple colors and knowledge verification/validation of rule-based expert system [J]. IEEE Trans on System, Man, and Cybernetics, 1997, 27(5): 73~85

11 Montague R M, Denegri S L, Curlin T H, et al. System Area Networks: Next Generation of Scale in the Data Center [Z]. Dain Rauscher Incorporated, 2001

12 袁崇义. Petri网原理. 北京: 电子工业出版社, 1998

13 杜兴, 谢立, 孙中秀. 一种基于对象的分布式系统描述求精方法. 计算机学报, 1994, 17(7): 562~567

14 丁志军, 蒋昌俊. 并发程序验证的时序Petri网方法[J]. 计算机学报, 2002, 25(5): 62~69