基于流体随机 Petri 网的服务器集群再生模型

杜小智[1] 齐 勇[1] 侯 迪[1] 刘 亮[2]
(西安交通大学电子与信息工程学院 西安 710049)[1] (IBM 中国研究院 北京 100094)[2]

摘 要 服务器集群是提高系统 QoS 和可用性的一种方法,但由于需要长期连续运行提供服务,集群系统仍然会存在软件老化现象,从而造成系统的处理能力随时间的增加而降低、失效率随时间的增加而增加。现同时考虑软件老化对系统处理能力与失效率的影响,采用流体随机 Petri 网对集群系统的工作模式进行建模,给出系统各个状态概率的数值分析方法,选择系统的可用性和吞吐量作为评价集群系统性能的指标,并进行仿真试验。结果表明该方法能很好地描述系统且具有可扩展性,分析并求得系统的最佳再生周期。

关键词 软件再生,软件老化,流体随机 Petri 网,集群
中图法分类号 TP391 **文献标识码** A

Rejuvenation Model of Server Cluster with Fluid Stochastic Petri Net

DU Xiao-zhi[1] QI Yong[1] HOU Di[1] LIU Liang[2]
(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)[1]
(IBM China Research Lab, Beijing 100094, China)[2]

Abstract Server clustering is an usual method to improve QoS and availability of system. But due to the need for long-term continuous operation to provide services, there exists software aging phenomenon in cluster system, which makes the system service rate is reduced and the failure rate is increased with the time increases. In this paper, the effect of software aging and the effect of failure were all considered, and fluid stochastic Petri net was adopted to model the working pattern of cluster system. Then the analysis methods of probability of each system state were given, and the formulas of system availability and throughout were presented. The availability and throughout were taken as the performance metrics of the cluster system. And some numeric experiments were done. The results show that the method introduced in this paper has the ability to describe system and is extendable. By using this method, the optimal rejuvenation period is gotten.

Keywords Software rejuvenation, Software aging, Fluid stochastic petri net, Cluster

1 引言

随着基于互联网的各种企业级应用的大量出现,有关系统的高可靠性和运行时服务质量(QoS)保证问题受到人们的普遍重视。然而,近年来对于软件可靠性的研究发现很多大型软件系统在持续平稳运行一段时间之后,往往就会出现系统资源大量消耗,服务性能和质量下降,甚至出现挂起或停机,称为"软件老化"(software aging)现象[5,6]。在操作系统[3]、Java 虚拟机[2]、Web 服务器[1]、SOA 服务器[4]等系统中都普遍发现软件老化现象。为了消除或者减缓系统软件老化现象,文献[6]提出了"软件再生"(software rejuvenation)的思想。软件再生是一种"前摄式"(proactive)的容错技术,主要通过适时地暂停软件的运行,清除持续运行系统的内部状态,重新启动并恢复到干净的初始或中间状态,从而达到预先防止将来可能发生更严重的故障的目的。

服务器集群是提高系统 QoS 和可用性的一种方法,同单一的服务器主机相比,服务器集群系统具有更高的性价比,更好的可扩展性、可靠性和容错性。但由于需要长期连续运行提供服务,集群系统仍然会存在软件老化现象,甚至宕机从而造成较大的经济损失。因此,需要研究集群系统的软件老化现象和再生方法。文献[9]研究了再生在集群系统中的应用,采用随机回报网分别建模基于时间与基于预测再生策略的集群系统,以系统可用性和成本为度量指标决定最佳再生间隔。文献[10]考虑了负载对系统性能的影响,将负载分为峰值与非峰值两种状态,提出了延迟再生策略,并采用随机回报网对集群系统进行建模,以吞吐量和可用性作为度量标准。文献[11]采用半马尔可夫方法对 active/standby 集群系统建立再生模型,以系统的可用性和停工成本为度量指标,求解最佳再生周期。文献[12]将再生作为带攻击的集群系统的恢复措施,分别采用随机和马尔可夫决策过程方法进行建模,估计系统在未知攻击下的可生存概率。文献[13] 对分发-工作(dispatcher-worker)集群系统进行建模和分析,该集群在分发池

到稿日期:2009-01-16 返修日期:2009-02-24 本文受国家 863 项目(2006AA01Z101),IBM 合作项目资助。
杜小智(1979—),男,博士生,主要研究方向为软件可靠性等,E-mail:smartzhi@yahoo.cn;齐 勇(1957—),男,教授,博士生导师,主要研究方向为容错计算、分布式系统、移动计算等;侯 迪(1960—),男,副教授,主要研究方向为分布式计算、软件形式化等。

和工作池采用基于预测的再生方法，对整个系统采用随机回报网建模。文献[14]将再生应用到两节点集群系统，采用连续时间马尔可夫链(CTMC)，研究比较两种不同的集群技术(冷备用、热备用系统)的再生特征。虽然上述文献对集群系统进行了研究，并取得一些成果，但它们着重于研究再生在集群中的应用，并没有考虑软件老化对集群系统的影响。

系统的软件老化过程与系统正常运行的时间有密切的关系，系统运行时间越长，系统的老化程度越高，其服务能力、处理能力越差，系统越容易发生故障。流体随机 Petri 网(FSPN)[7,8]作为随机 Petri 网的扩展，将位置中的标记数目从离散域扩展到连续域，用来描述结合连续决定性和离散随机性的混合系统。FSPN 可以对连续量进行描述与建模，而时间是一个连续量，为了对集群系统的行为特征进行描述，本文采用 FSPN 对集群系统的工作模式进行建模，选择系统的可用性和吞吐量作为评价集群系统性能的指标，并进一步分析不同参数下的最佳再生策略。

2 集群系统工作模型

为叙述方便且不失一般性，本文考虑两节点集群系统。采用 FSPN 对两节点集群系统进行描述，其运行模型如图 1 所示，模型中各个模块的详细定义如表 1、表 2 所列，其中表 1 为模型中变迁的定义，表 2 为模型中位置的定义。系统整体上分为 3 个部分：系统缓冲区、服务器 1 状态和服务器 2 状态，各部分的详细说明如下。

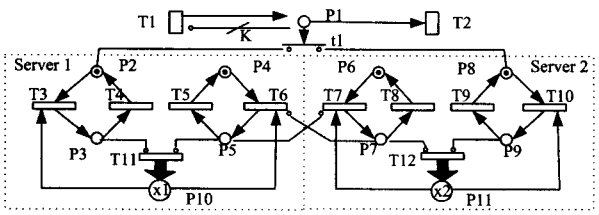

图 1 集群系统 FSPN 工作模型

(1)系统缓冲区(P1,T1,T2,t1)：T1 产生服务请求，标记进入位置 P1，T2 处理请求，请求处理完成后标记离开 P1。当两个服务器都不在工作态时，瞬时迁移 t1 触发，将 P1 中等待的任务移除。服务处理率与系统的老化程度有关(即与 P10，P11 有关)，系统的容量为 K。

(2)服务器 1 状态，可细分为 3 个模块。1)服务器 1 工作状态(P2,P3,T3,T4)：token 位于 P2 内，此时服务器 1 处于工作状态；服务器 1 以一定的速率失效，触发 T3，标记进入 P3，此时服务器 1 不可用，系统修复后，触发 T4，标记回到 P2。失效率随系统的老化程度(即运行时间 x_1)的增加而增大，修复率服从指数分布。2)服务器 1 再生(P4,P5,T5,T6)：服务器 1 每隔一定周期执行一次再生，触发 T6，此时 token 位于 P5，服务器 1 处于再生状态；当再生操作完成后，服务器 1 的状态进行更新，触发 T5 标记回到 P4。假定 T5 的转移率服从指数分布，T6 的转移率为运行时间 x_1 的函数。若系统采用基于时间的再生策略，则服务器 1 每隔一段时间 Tr_1 就执行再生操作。3)服务器 1 连续运行时间(P10,T11)：累计进入位置 P10 的标记数表示服务器 1 的连续运行时间，T11 的转移率为常数。当服务器 1 发生失效或执行再生时，即 T3 或 T6 实施时，服务器 1 的连续运行中断，P10 清零。

表 1 FSPN 模型中变迁的定义

变迁	事件	转移率/函数
T1	服务请求到达	λ
T2	任务完成	(#P1)*F2(x1,x2)
T3	Sever 1 失效	$F_3(x1)$
T4	Sever 1 恢复	μ_{1f}
T5	Sever 1 再生结束	μ_{1r}
T6	Sever 1 再生开始	$F_6(x1)$
T7	Sever 2 再生开始	$F_7(x2)$
T8	Sever 2 再生结束	μ_{2r}
T9	Sever 2 恢复	μ_{2f}
T10	Sever 2 失效	$F_{10}(x2)$
T11	Sever 1 连续运行	r_{11}
T12	Sever 2 连续运行	r_{12}
t1	清空任务队列	G2=(#P2)+(#P8)<1

表 2 FSPN 模型中位置的定义

位置	含义	初始标记数
P1	系统中的任务数	0
P2	Sever 1 正常工作	1
P3	Sever 1 失效	0
P4	Sever 1 未执行再生	1
P5	Sever 1 执行再生	0
P6	Sever2 未执行再生	1
P7	Sever 2 执行再生	0
P8	Sever 2 正常工作	1
P9	Sever 2 失效	0
P10	Sever 1 连续运行时间	0
P11	Sever 2 连续运行时间	0

(3)服务器 2 状态，可细分为 3 个模块。1)服务器 2 工作状态(P8,P9,T9,T10)：token 位于 P8 内，此时服务器 2 处于工作状态；服务器 2 以一定的速率失效，触发 T10，标记进入 P9，此时服务器 2 不可用，系统修复后，触发 T9，标记回到 P8。失效率随系统的老化程度(即运行时间 x_2)的增加而增大，修复率服从指数分布。2)服务器 2 再生(P6,P7,T7,T8)：服务器 2 每隔一定周期执行一次再生，触发 T7，此时服务器 2 处于再生状态，token 位于 P7；当再生操作完成后，服务器 2 的状态进行更新，触发 T8 标记回到 P6。假定 T8 的转移率服从指数分布，T7 的转移率为运行时间 x_2 的函数。若系统采用基于时间的再生策略，则服务器 2 每隔一段时间 Tr_2 就执行再生操作。3)服务器 2 连续运行时间(P11,T12)：累计进入位置 P11 的标记数表示服务器 2 的连续运行时间，T12 的转移率为常数。当服务器 2 发生失效或执行再生时，即 T10 或 T7 实施时，服务器 2 的连续运行中断，P11 清零。

3 数值分析

基于上述 FSPN 模型，根据各个子系统的状态，系统可以处于 8 个状态：两个服务器都正常工作 m_0={P2,P4,P6,P8}，服务器 1 工作、服务器 2 再生 m_1={P2,P4,P7,P8}，服务器 1 再生、服务器 2 工作 m_2={P2,P5,P6,P8}，服务器 1 工作、服务器 2 失效 m_3={P2,P4,P6,P9}，服务器 1 失效、服务器 2 工作 m_4={P3,P4,P6,P8}，服务器 1 再生、服务器 2 失效 m_5={P2,P5,P6,P9}，服务器 1 失效、服务器 2 再生 m_6={P3,P4,P7,P8}，两个服务器都失效 m_7={P3,P4,P6,P9}。由于不能人为地将两个系统都处于不可用状态，因此不存在服务器 1 和服务器 2 都处于再生的状态。设系统处于这 8 个状态的概率分别为：$\pi_0(t,x_1,x_2)$，$\pi_1(t,x_1)$，$\pi_2(t,x_2)$，$\pi_3(t,x_1)$，$\pi_4(t,x_2)$，$\pi_5(t)$，$\pi_6(t)$，$\pi_7(t)$。

基于上面符号，采用下面的偏微分方程可以描述系统的

演化过程。

$$\frac{\partial \pi_0(t,x_1,x_2)}{\partial t}+r_{11}\frac{\partial \pi_0(t,x_1,x_2)}{\partial x_1}+r_{12}\frac{\partial \pi_0(t,x_1,x_2)}{\partial x_2}$$
$$=-\pi_0(t,x_1,x_2)[F_6(x_1)+F_3(x_1)+F_7(x_2)+F_{10}(x_2)+F(x_1,x_2)] \quad (1)$$

$$\frac{\partial \pi_1(t,x_1)}{\partial t}+r_{11}\frac{\partial \pi_1(t,x_1)}{\partial x_1}$$
$$=-\pi_1(t,x_1)[F_3(x_1)+\mu_{2r}]+\int_0^{\infty}\pi_0(t,x_1,x_2)F_7(x_2)\,dx_2 \quad (2)$$

$$\frac{\partial \pi_2(t,x_2)}{\partial t}+r_{12}\frac{\partial \pi_2(t,x_2)}{\partial x_2}$$
$$=-\pi_2(t,x_2)[F_{10}(x_2)+\mu_{1r}]+\int_0^{\infty}\pi_0(t,x_1,x_2)F_6(x_1)\,dx_1 \quad (3)$$

$$\frac{\partial \pi_3(t,x_1)}{\partial t}+r_{11}\frac{\partial \pi_3(t,x_1)}{\partial x_1}$$
$$=-\pi_3(t,x_1)[F_3(x_1)+\mu_{2f}]+\int_0^{\infty}\pi_0(t,x_1,x_2)F_{10}(x_2)\,dx_2 \quad (4)$$

$$\frac{\partial \pi_4(t,x_2)}{\partial t}+r_{12}\frac{\partial \pi_4(t,x_2)}{\partial x_2}$$
$$=-\pi_4(t,x_2)[F_{10}(x_2)+\mu_{1f}]+\int_0^{\infty}\pi_0(t,x_1,x_2)F_3(x_1)\,dx_1 \quad (5)$$

$$\frac{\partial \pi_5(t)}{\partial t}=-\pi_5(t)(\mu_{2f}+\mu_{1r})+\int_0^{\infty}\pi_2(t,x_2)F_{10}(x_2)\,dx_2 \quad (6)$$

$$\frac{\partial \pi_6(t)}{\partial t}=-\pi_6(t)(\mu_{1f}+\mu_{2r})+\int_0^{\infty}\pi_1(t,x_1)F_3(x_1)dx_1 \quad (7)$$

$$\frac{\partial \pi_7(t)}{\partial t}=-\pi_7(t)(\mu_{1f}+\mu_{2f})+\int_0^{\infty}\pi_3(t,x_1)F_3(x_1)dx_1+\int_0^{\infty}\pi_4(t,x_2)F_{10}(x_2)dx_2+\int_0^{\infty}\int_0^{\infty}\pi_0(t,x_1,x_2)F(x_1,x_2)dx_1dx_2 \quad (8)$$

边界条件：

$$\pi_0(t,0,0)=\mu_{2r}\int_0^{\infty}\pi_1(t,x_1)dx_1+\mu_{1r}\int_0^{\infty}\pi_2(t,x_2)dx_2+\mu_{2f}\int_0^{\infty}\pi_3(t,x_1)dx_1+\mu_{1f}\int_0^{\infty}\pi_4(t,x_2)dx_2 \quad (9)$$

$$\pi_1(t,0)=\mu_{1f}\pi_6(t) \quad (10)$$

$$\pi_2(t,0)=\mu_{2f}\pi_5(t) \quad (11)$$

$$\pi_3(t,0)=\mu_{1r}\pi_5(t)+\mu_{1f}\pi_7(t) \quad (12)$$

$$\pi_4(t,0)=\mu_{2r}\pi_6(t)+\mu_{2f}\pi_7(t) \quad (13)$$

其中，$F(x1,x2)$为两个系统同时失效的概率函数，可以由 $F_3(x1)$，$F_{10}(x2)$得到。

对上述偏微分方程求解，可以得到各状态的概率函数 π_i $(i=0,1,\cdots,7)$。

根据集群系统的状态关系描述，当系统处于状态 $m_0\sim m_4$ 时可以提供服务，即系统的可用性 P_{avail} 为：

$$P_{avail}=\pi_0+\pi_1+\pi_2+\pi_3+\pi_4 \quad (14)$$

服务器 1 和服务器 2 只有在工作状态下才能提供服务，由于软件老化的影响，其处理能力随时间的增加而逐步减小，即其工作状态下的吞吐量 $h_1(t)$，$h_2(t)$为单调非增函数。由于系统在运行过程中会执行再生操作和失效后的恢复操作，而 x_1，x_2 分别表示服务器 1，2 更新后的运行时间，则在某时刻 t，系统的处理能力 $g(t)$为：

$$g(t)=\begin{cases}h_1(x_1)+h_2(x_2) & \text{状态 0}\\ h_1(x_1) & \text{状态 1,3}\\ h_2(x_2) & \text{状态 2,4}\\ 0 & \text{状态 5,6,7}\end{cases} \quad (15)$$

若假设服务请求以固定速率 λ 到达，则系统到 t 时刻的吞吐量 $Th(t)$：

$$Th(t)=\frac{\int_0^t\min(g(\tau),\lambda)d\tau}{t} \quad (16)$$

4 仿真实验与分析

为了分析与验证所提出的软件再生模型是否有效，进行仿真实验。本文假定再生策略是基于时间的，即系统每隔一段时间就执行再生操作，则 $F_6(x_1)=\delta(x_1-Tr_1)$，$F_7(x_2)=\delta(x_2-Tr_2)$，其中 Tr_1 为服务器 1 的再生周期，Tr_2 为服务器 2 的再生周期。分别对 Tr_1，Tr_2 取不同的值，以系统可用性和吞吐量为度量指标选择最佳的再生周期。软件老化对系统的影响还表现在失效率随运行时间的增加，假设各个服务器的失效率为其运行时间的线性函数，即 $F_3(x_1)=\eta_1*\alpha_1*x_1$，$F_{10}(x_2)=\eta_2*\alpha_2*x_2$。

设服务器 1，2 在工作状态下的服务率与时间成线性关系，即初始时刻服务器 1 具有最大的服务率 $\mu_{1\max}$，随着时间的增加 μ_1 逐渐减小，到 τ_1 时刻，μ_1 达到最小 $\mu_{1\min}$，此后 μ_1 一直维持为 $\mu_{1\min}$。相应地，服务器 2 对应的服务率分别为 $\mu_{2\max}$，τ_2，$\mu_{2\min}$。

实验中所采用的参数为：$\mu_{1f}=0.2$，$\mu_{1r}=0.5$，$\mu_{2f}=0.2$，$\mu_{2r}=0.5$，$r_{11}=1$，$r_{12}=1$，$\eta_1=2$，$\eta_2=2$，$\alpha_1=2*10e\text{-}4$，$\alpha_2=2*10e\text{-}3$，$\lambda=5$，$\mu_{1\max}=4$，$\mu_{1\min}=2$，$\tau_1=30$，$\mu_{2\max}=4$，$\mu_{2\min}=2$，$\tau_1=40$。

图 2 为 $Tr_2=10$，Tr_1 分别为 19，23，43 时，系统的可用性随时间的变化关系。从图中可以看出：(1)对于任意的 Tr_1 取值，在经过一段时间后，系统的可用性基本上维持在某个范围内，即系统具有稳态可用性；(2)随着 Tr_1 的增大，系统可用性波动越小。

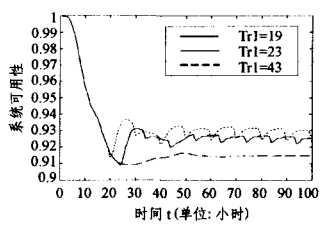

图 2 系统可用性随时间的关系图

取系统可用性在时间段[60,90]上的均值作为稳态可用性。表 3 为在 $Tr_2=10$ 时，各个 Tr_1 取值所对应的系统稳态可用性。从表中可以看出，随着 Tr_1 的增大，系统的可用性增加，当 Tr_1 达到某个值时，继续增加 Tr_1，系统的可用性下降。

因此，存在最佳的再生周期，使系统的可用性达到最大，即 Tr_1 为 9 时，系统可用性最大为 0.9362。

表 3　系统可用性与 Tr_1 关系

Tr_1	3	6	9	13
Avail	0.9307	0.9346	0.9362	0.9352
Tr_1	16	19	23	26
Avail	0.9325	0.9295	0.9257	0.9230
Tr_1	29	33	36	39
Avail	0.9205	0.9179	0.9164	0.9154

图 3 为 $Tr_2=20$，Tr_1 分别为 16，24，32 时，集群系统的吞吐量随时间的变化情况，从图中可以看出：(1)经过一段时间后，系统的吞吐量趋于稳定；(2)随着 Tr_1 的增加，稳态吞吐量逐步减小，但差值也逐步减小。

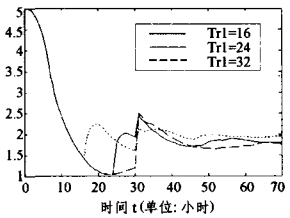

图 3　吞吐量随时间的关系图

结束语　服务器集群是提高系统 QoS 和可用性的一种方法，在现实中应用得越来越广。但由于软件老化的影响，使得系统的可用性与 QoS 不能得到保障。由于软件老化过程与时间有关，针对两节点集群系统，采用流体随机 Petri 网对系统的工作模式进行建模，分析系统各个状态概率，给出系统的可用性和吞吐量计算公式，并进行仿真试验。结果表明本文的方法能很好地描述系统且具有可扩展性，可以分析求得系统的最佳再生周期。

软件老化过程除了与时间有关外，还与已处理的任务数、系统当前的排队等待处理的任务数有关。因此，该研究点下一步的工作是在用 FSPN 对系统工作行为建模时，考虑已处理任务数和排队等待任务数对系统的影响；并考虑其它再生策略。

参 考 文 献

[1] Grottke M, Li L, Vaidyanathan K, et al. Analysis of Software Aging in a Web Server[J]. IEEE Transactions on Reliability, 2006, 55(3): 411-420

[2] Cotroneo D, Orlando S, Russo S. Characterizing Aging Phenomena of the Java Virtual Machine[C]// 26th IEEE International Symposium on Reliable Distributed Systems(SRDS'07). 2007: 127-136

[3] Vaidyanathan K, Trivedi K S. A Measurement-Based Model for Estimation of Software Aging in Operational Software Systems [C]// Intl. Symp. On Software Reliability Engineering. ISSRE 1999, Nov. 1999

[4] Silva L, Madeira H, Silva J G. Software aging and rejuvenation in a soap-based server[C]// IEEE-NCA: Network Computer and A-pplications. July 2006: 56-65

[5] Avritzer A, Weyuker J. Monitoring Smoothly Degrading Sys-tems for Increased Dependability [J]. Empirical Software Eng. Journal, 1997, 12(1): 59-77

[6] Huang Y, Kintala C, Kolettis N, et al. Software Rejuvenation: Analysis, Module and Applications [C]// Proc. 25th Symp. Fault Tolerant Computing. Pasadena, USA. IEEE Press, 1995: 381-390

[7] Gribaudo M, Sereno M, Horvath A, et al. Fluid Stochastic Petri nets augmented with flush-out arcs: modeling and analysis [J]. Discrete Event Dynamic Systems, 2001, 11(1): 97-117

[8] Horton G, Kulkarni V G, Nicol D M, et al. Fluid stochastic Petri nets: Theory, application, and solution techniques[J]. European Journal of Operation Research, 1998, 105(1): 184-201

[9] Vaidyanathan K, Harper R E, Hunter S W, et al. Analysis and Implementation of Software Rejuvenation in Cluster Systems [C]// Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems. 2001: 62-71

[10] Xie W, Hong Y, Trivedi K S. Software rejuvenation policies for cluster systems under varying workload[C]// Proceedings of 10th IEEE Pacific Rim International Symposium on Dependable Computing. March 2004: 122-129

[11] Kiejin Park, Sungsoo Kim. Availability analysis and improve-ment of active/standby cluster systems using software rejuve-nation [J]. Journal of Systems and Software, 2002, 61(2): 121-128

[12] Aung K M M, Park K, Park J S. A rejuvenation methodology of cluster recovery[C]// IEEE International Symposium on Cluster Computing and the Grid. May 2005: 90-95

[13] Fan Xin-yuan, Xu Guo-zhi, Eng Ren-dong, et al. Performance analysis of software rejuvenation on dispatcher-worker based cluster system[C]// Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies. 2003: 562- 566

[14] Koutras V P, Platis A N, Applying Software Rejuvenation in a Two Node Cluster System for High Availability[C]// Proceedings of the International Conference on Dependability of Computer Systems. 2006: 175-182

(上接第 130 页)

[4] Kuo D, Fekete A, Greenfield P, et al. Expressing and Reasoning about Service Contracts in Service-oiented Computing[C]// IEEE International Conference on Web Services(ICWS'06). 2006: 915-918

[5] Skonnard A. Contract-First Service Development[EB/OL]. http: // msdn. microsoft. com/msdn mag/issues/05/05 /Service Station/, 2005-05-23

[6] SParastatidis A, Woodman S, Webber J, et al. Asynchronous Messaging between Web Services Using SSDL[J]. IEEE Internet Computing, January 2007: 26-39

[7] Bhuiyan J, Nepal S, Zic J. Checking Conformance between Business Processes and Web Service Contract in Service Oriented Applications[C]// Australian Software Engineering Conference (ASWEC'06). 2006: 80-89