分布式锁的Petri网建模及死锁检测

金红琳 刘 波

(华南师范大学计算机学院 广州 510631)

摘 要 分布式锁管理DLM细化了锁模式的粒度,使得分布式系统具有更高的并发性,但死锁检测等锁的管理过程却更加复杂了,Petri网的应用能很好地解决该问题。为分布式锁建立Petri网模型,通过化简和合成建立系统的Petri网模型,借助Petri网的可达标识图实时检测出分布式系统的死锁状态,并查找死锁进程。

关键词 分布式系统,锁管理,DLM,Petri网,死锁检测

中图法分类号 TP393 **文献标识码** A

Petri Net Model and Deadlock Detection of Distributed Lock

JIN Hong-lin LIU Bo

(School of Computer, South China Normal University, Guangzhou 510631, China)

Abstract DLM uses 6 kinds of locks, which makes not only more concurrency in distributed systems, but also more difficult management of locks. Then Petri net works. The Petri net model of a distributed system is compounded from the simplified Petri net of its subsystems. With the reachable marking graph of the system's Petri net, deadlocks can be detected in real time, which processes in the deadlock chain will also be known.

Keywords Distributed system, Lock management, DLM, Petri net, Deadlock detection

在分布式系统中,可以通过多个进程的并发执行来提高系统运行效率和资源利用率,这些并发的进程竞争使用共享资源,在进程申请资源次序不当的情况下则可能发生死锁。死锁就是指一组进程中的某些成员由于无法访问组中其他成员已占用的资源而被无限期阻塞的现象。死锁的传统图论模型有等待图和资源分配图[1],Petri网相比于它们,具有动态、异步、并发等特点,更适于对分布式系统加锁进行建模。Petri网可以模拟各并发进程在推进过程中锁状态的变化。

1 分布式锁

当多个进程并发访问同一资源时,必须同步才能得到正确的结果,这一同步过程就是通过加锁来实现的。为了提高进程的并发性,不同的进程根据访问共享资源的方式不同而加不同的锁。因此,分布式锁管理(DLM)提供了多种锁模式[2,4]。在申请加锁或者锁转换的时候,要依据锁模式的兼容性来决定是否授权。

1.1 锁的模式及其兼容性

锁模式决定进程对资源的操作范围及其与其它进程的兼容性。DLM提供了6种锁模式[2,4]:

(1)空(NL):不允许访问资源,但允许其它进程访问资源。当没有其它锁存在时,空锁作为占位符,用于以后的锁转换或保存资源及其上下文的方法。

(2)并发读(CR):允许读访问资源,且允许其他进程对资源进行读/写访问。

(3)并发写(CW):允许写访问资源,且允许其他进程进行读/写访问。

(4)保护读(PR):允许读访问资源,其它进程可以同时读访问共享资源,但不可以写访问共享资源。

(5)保护写(PW):允许写访问资源,其它进程可以同时读访问共享资源,但不可以写访问共享资源。

(6)独占(EX):允许对资源进行读/写访问,其它所有进程不能同时访问该资源。

从以上定义可知DLM的6种锁模式的兼容性如表1所列。

表1 6种锁模式的兼容性

	NL	CR	CW	PR	PW	EX
NL	是	是	是	是	是	是
CR	是	是	是	是	是	否
CW	是	是	是	否	否	否
PR	是	是	否	是	否	否
PW	是	是	否	否	否	否
EX	是	否	否	否	否	否

定义1 NL,CR,CW,PR,PW和EX锁的层次记为$L(NL)$,$L(CR)$,$L(CW)$,$L(PR)$,$L(PW)$和$L(EX)$,其值分别为6,5,4,3,2,1。层次越低的锁越严格,兼容性越差。

定义2 当共有n个进程$P_1,P_2,\cdots,P_n$对资源r加了锁时,这n个锁的层次分别为$L_{P1},L_{P2},\cdots,L_{Pn}$,将表$LLT(r)=$

到稿日期:2010-12-31 返修日期:2011-04-01 本文受广东省自然科学基金项目(9451063101002238),广东省科技计划项目(2010B010600032)资助。

金红琳(1987-),女,硕士生,主要研究方向为分布式计算,E-mail:honglinking@163.com;刘 波(1968-),男,博士,副教授,主要研究方向为分布式计算、网格计算。

$(L_{P1},L_{P2},\cdots,L_{Pn},L_{NULL})$称为资源 r 的锁层次表，$L_{NULL}=7$ 表示未加锁，这样设置是便于实现加锁解锁。

定义 3 $L(r)=\min(L_{P1},L_{P2},\cdots,L_{Pn},L_{NULL})$称为资源 r 的锁层次。

1.2 锁状态

分布式系统中的锁共有 3 种状态[3]：

(1)请求态：新申请的加锁先被放到等待队列中等待被授权。

(2)授权态：在不冲突的条件下，加锁请求被授权，相应的进程可以访问共享资源。

(3)转换态：进程在运行中请求转换成其它模式的锁，若请求与当前已授权的其它锁不兼容，那么锁转换请求进入转换队列，等待其它锁降低限制或解锁。

2 Petri 网模型

加权 Petri 网的可达性、有界性及活性很好地模拟了动态加锁解锁的过程，其可达标识图描述了加锁解锁过程中可能发生的各种状态。最后通过简化 Petri 网，减少要分析的库所及变迁，使得分布式锁的状态更明了。

2.1 加权 Petri 网定义

定义 4 五元组 $\Sigma=(S,T,F,W,M)$称为一个加权 Petri 网，其中

(1)(S,T,F)是一个网；$W:F\to\{\cdots,-1,0,1,\cdots\}$称为权函数；$M:S\to\{0,1,2,\cdots\}$是$\Sigma$的一个标识。

(2)$\forall s\in{}^{\cdot}t:M(s)\geqslant W(s,t)\to M[t>$，则在 M 下发生变迁 t 得到一个新的标识 M'，记为 $M[t>M'$，且 $\forall s\in S$，

$$M'(s)=\begin{cases}M(s)-W(s,t), & \text{若 } s\in{}^{\cdot}t-t^{\cdot}\\ M(s)+W(t,s), & \text{若 } s\in t^{\cdot}-{}^{\cdot}t\\ M(s)+W(t,s)-W(s,t), & \text{若 } s\in{}^{\cdot}t\cap t^{\cdot}\\ M(s), & \text{其它}\end{cases}$$

2.2 Petri 网的可达性、有界性和活性

定义 5 设 $\Sigma=(S,T,F,M)$是一个 Petri 网，若存在 $t\in T$ 使 $M[t>M'$，则称 M' 为从 M 直接可达的；若存在 $t_1,t_2,\cdots,t_k$ 和标识 $M_1,M_2,\cdots,M_k$ 使得 $M[t_1>M_1[t_2>M_2\cdots[t_k>M_k$，则称 M_k 是从 M 可达的。从 M 可达的一切标识 $M_1,M_2,\cdots,M_k$ 的集合为 $R(M)$，约定 $M\in R(M)$。若变迁序列 $t_1,t_2,\cdots,t_k$ 记为 σ，则 $M[\sigma>M_k$。

定义 6 设 $\Sigma=(S,T,F,M_0)$是一个 Petri 网，$s\in S$，若存在正整数 B 使 $\forall M\in R(M_0):M(s)\leqslant B$，则称库所 s 是有界的，并称满足此条件的 B 为库所 s 的界，记为 $B(s)$。即 $B(s)=\min\{B\mid\forall M\in R(M_0):M(s)\leqslant B\}$。如果每个 $s\in S$ 都是有界的，则称 Σ 为有界 Petri 网。有界网的界为 $B(\Sigma)=\max\{B(s)\mid s\in S\}$。

定义 7 设 $\Sigma=(S,T,F,M_0)$是一个 Petri 网：(1)如果 $M\in R(M_0)$使得 $\forall t\in T:\neg M[t>$，则称 M 为 Σ 的一个死标识。如果 Σ 中不存在死标识，则称 Σ 为不死的或弱活的。(2)对于 $t\in T$，$\forall M\in R(M_0):\neg M[t>$，则称变迁 t 是死的。

2.3 可达标识图

对于有界 Petri 网，其可达标识集也是一个有限集，因此，可以以 $R(M_0)$为顶点集，以标识间的直接可达关系为弧集构成一个有界的有向图，这种图称为 Petri 网的可达标识图(Reachable Marking G Graph)。通过 Petri 网的可达标识图可以很直观地分析网系统的状态变化和变迁的发生序列。

定义 8 设 $\Sigma=(S,T,F,M_0)$是一个有界 Petri 网。Σ 的可达标识图定义为一个三元组 $RG(\Sigma)=(R(M_0),E,P)$，其中 $E=\{(M_i,M_j)\mid M_i,M_j\in R(M_0),\exists t_k\in T:M_i[t_k>M_j\}$，$P:E\to T$，$P(M_i,M_j)=t_k$ 当且仅当 $M_i[t_k>M_j$。称 $R(M_0)$为 $RG(\Sigma)$的顶点集，E 为 $RG(\Sigma)$的弧集，若 $P(M_i,M_j)=t_k$，则称 t_k 为弧(M_i,M_j)的旁标。

2.4 Petri 网的合成

Petri 网的合成是指把多个网合成为一个网。在对一个复杂的系统构建 Petri 网模型时，我们一般把系统划分为若干个子系统，先对子系统分布建模，然后通过 Petri 网的合成运算得到整个系统的 Petri 网模型。

定义 9 设 $N_i=(S_i,T_i,F_i,M_{0i}),i=1,2$，则称 $N=(S_1\cup S_2,T_1\cup T_2,F_1\cup F_2,M_0)$为 N_1 和 N_2 的合成网。其中：

(1)若 $S_1\cap S_2=\varnothing,T_1\cap T_2\neq\varnothing$，则 N 为同步合成网，记为 $N=N_1C_TN_2$。且 $M_0(S)=M_{0i}(s)(s\in S_i,i=1,2)$；

(2)若 $S_1\cap S_2\neq\varnothing,T_1\cap T_2=\varnothing$，则 N 为共享合成网，记为 $N=N_1C_SN_2$。且

$$M_0(S)=\begin{cases}M_{0i}(s), & \text{若 } s\in(S_1\cup S_2)-(S_1\cap S_2)\\ M_{01}(s)+M_{02}(s), & \text{若 } s\in(S_1\cap S_2)\end{cases}$$

(3)若 $S_1\cap S_2\neq\varnothing,T_1\cap T_2\neq\varnothing$，则 N 为同步共享合成网。

3 分布式锁的 Petri 网建模

在建模过程中，我们省略了进程访问资源的细节，而只考虑加锁和解锁。状态集 S 代表资源 r 和进程 P 的状态；迁移集 T 代表加锁、解锁的过程。

3.1 分布式锁的 Petri 网模型意义

定义 10 假设分布式系统中有 n 个进程请求对 m 个资源加锁，那么分布式锁的 Petri 网模型对应一五元组 $\Sigma=(S,T,F,W,M_0)$，其意义是：

(1)S 包括了进程 P 的请求状态和授权状态、资源 r 的状态以及一些中间状态 S_f 等子集，即 $S=\{P,P_l,R,S_f\}$，其中，$P=\{p_1,p_2,\cdots,p_n\}$代表 n 个进程的加锁请求状态，$P_l=\{p_{1l},p_{2l},\cdots,p_{nl}\}$代表锁授权状态；$R=\{r_1,r_2,\cdots,r_m\}$代表 m 个资源所处的锁层次；$S_f=\{s_1,s_2,\cdots,s_i\}$代表 i 个中间辅助控制状态。

(2)T 包括进程加锁、解锁的过程以及一些辅助的中间变迁过程，即 $T=\{T_l,P_{unl},T_f\}$，其中，$T_l=\{t_{1l},t_{2l},\cdots,t_{nl}\}$中 t_{kl} $(k\in\{1,2,\cdots,i\})$代表进程 p_i 的加锁过程；$T_{unl}=\{t_{1unl},t_{2unl},\cdots,t_{munl}\}$中 t_{kunl} $(k\in\{1,2,\cdots,i\})$代表 p_i 的解锁过程；$T_f=\{t_1,\cdots,t_i\}$代表辅助的中间变迁过程。

(3)M_0 给出了加锁请求的初始状态：$M_0=M_0(S)=\{M_0(P),M_0(P_l),M_0(R),M_0(S_f)\}$，其中 $M_0(P)\subseteq N^+$ 是进程 p_i 的加锁请求个数；$M_{0r}=\{M_{0r}(r_1),M_{0r}(r_2),\cdots,M_{0r}(r_m)\}$中 $M_0(R)=7$ 表示尚没有进程对其加锁；$M_0(P_l)=0$ 表示所有加锁未得到授权；$M_0(S_f)=0$ 表示中间变迁尚未发生。

我们从一个进程对一个资源申请加一项锁的简单 Petri 网模型入手，分布式系统的全局 Petri 网模型由简单 Petri 网经过多次合成得到。

3.2 进程加锁的 Petri 网模型

当被请求的锁 l 与资源上已有的锁都兼容时，l 的层次 $L(r)\geqslant 7-L(l)$。加锁成功后，将 l 的层次 $L(l)$加入资源的锁层

次表 $LLT(r)$中得到新的锁层次表 $LLT'(r)$。若 $L(l)<L(r)$，则资源的锁层次 $L'(r)=L(l)$。

NL,CR,CW,PW 和 EX 锁的加锁 Petri 网模型如图 1 所示，图中，对于 NL,CR,CW,PW 和 EX 锁，x 的值分别是 1,2,4,5,6。r 代表共享资源，P_r 代表进程请求锁的状态，P_a 表示加锁请求是否得到授权。图中，$M(P_r)=1$ 表示进程请求了一项加锁；$M(P_a)=0$ 表示加锁请求尚未被授权；$M(r)=L(r)$表示资源的锁层次为 $L(r)$，当 $M(r)\geqslant 7-L(l)$时，加锁被授权，变迁 t_l 发生，$M(P_a)=1$。同时，若 $L(l)<L(r)$，资源的锁层次应变为 $L(l)$，通过“减 1”功能的辅助变迁 t_r 的连续执行来实现。

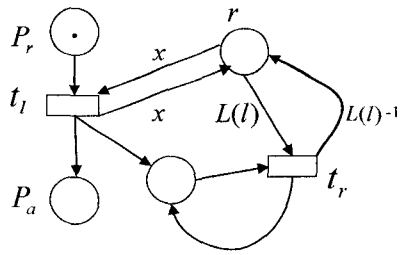

图 1　加锁的 Petri 网模型

PR 锁的 Petri 网模型与其他锁稍微不同，PR 锁与 NL,CR,PR 锁兼容，与 CW,PW,EX 锁不兼容。所以，当 $M(r)=3,5,6,7$ 时，t_{PRl} 发生。加锁后 $M(r)=L(PR)$。PR 锁 Petri 模型如图 2 所示。

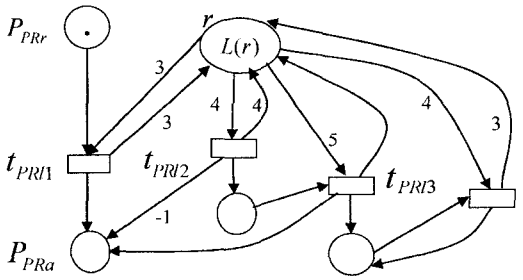

图 2　加 PR 锁的 Petri 网模型

3.3　进程解锁的 Petri 网模型

在对系统建模时，忽略进程的执行过程，当进程加锁成功时，进程才可以解锁。设被解锁的层次是 n，进程解锁的过程是：

(1)从资源 r 的锁层次表 $LLT(r)$中删除值为 n 的一项，得到 r 的新锁层次表 $LLT'(r)$；

(2)解锁；

(3)若资源的锁层次等于被解锁的层次，则找出 $LLT'(r)$中最小的一个值 m，令 $L(r)=m$，否则资源的锁层次不变。

图 3 给出了分布式系统中进程解锁的 Petri 网模型。r 代表共享资源，P_a 代表进程是否已经加锁的状态，t_{unl} 代表解锁，t_f 是修改资源锁层次的变迁过程。

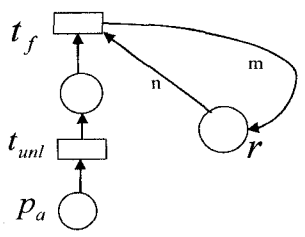

图 3　分布式锁解锁的 Petri 网模型

3.4　分布式系统中并发进程 Petri 网的化简

分布系统中的进程可能包含多个子进程，它们分别需要不同站点上的资源。在单个站点上运行的子进程，只要获得该站点上的所需数据即可执行；而只有所有的子进程全部获得所需资源时，进程才可以运行并解锁、释放资源。

死锁检测考察的是各并发进程状态间的关系，即，在站点局部范围内考察并发进程间的锁是否冲突，在系统全局范围内分析进程等待的资源能否被释放。因此，对于站点需要建立加锁的 Petri 网模型，对于系统需要建立解锁的 Petri 网模型，然后组合多个站点的 Petri 网，并进行化简。

定义 11　系统的 m 个进程 $P_1,P_2,\cdots,P_m$ 有 n 个子进程 $p_1,p_2,\cdots,p_n$ 在站点上，站点的 Petri 网 $\Sigma=(S,T,F,W,M_0)$ 经化简后得到四元组 Petri 网 $\Sigma'=(S',T',F',M_0')$。其中，$S'=\{P_1,P_2,\cdots,P_n\}$是子进程 $p_1,p_2,\cdots,p_n$ 的状态集；$T'=\{t_{1unl},t_{2unl},\cdots,t_{m\text{-}unl}\}$是进程 $P_1,P_2,\cdots,P_m$ 解锁的变迁集；$M_0'=\{M_0'(P_1),M_0'(P_2),\cdots,M_0'(P_n)\}$是各站点完成加锁后各子进程所处的状态集。

例如，进程 P_1 和 P_2 各有 2 个子进程 p_{11}，p_{12}和 p_{21}，p_{22}，p_{11}请求站点上的资源 r 的 EX 锁，p_{21}请求资源 r 的 CR 锁，那么，子进程 p_{11}请求加锁和解锁的合成 Petri 网模型可由图 1、图 3 合成得到，如图 4 所示；同理可得子进程 p_{21}请求加锁和解锁的合成 Petri 网模型(图略)。站点的合成 Petri 网模型如图 5 所示，从该图分析可知，进程 p_{21}能够获取资源(加锁)并释放资源(解锁)的前提是进程 p_{11}能够释放资源(解锁)，因此，可以对复杂的图 5 进行化简，得到图 6。

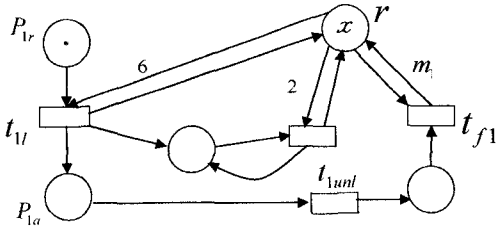

图 4　进程 P_{11}的加锁解锁 Petri 网模型

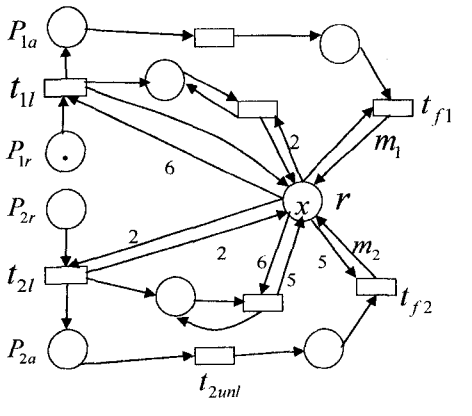

图 5　P_{11}，P_{12}同时加锁的合成 Petri 网模型

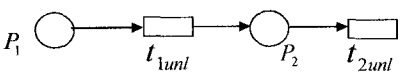

图 6　图 5 的化简 Petri 网

4　死锁检测

分布式系统中的资源存在于多个站点上，而且有多个并

发进程竞争资源。因此，系统进行死锁检测时，先检测各站点局部是否出现死锁，若局部存在死锁，则整个系统肯定处于死锁状态，否则继续在全局范围内检测死锁。

4.1 死锁检测过程

(1)在单站点范围内，建立每个进程加锁请求的 Petri 网模型 $\Sigma_1=(S_1,T_1,F_1,W_1,M_{01})$；

(2)以资源作为连接点，构造站点的局部共享合成 Petri 网模型 $\Sigma_2=(S_2,T_2,F_2,W_2,M_{02})$；

(3)根据并发进程的请求给出 Petri 网的始态 M_0，画出可达标识图 $RG(\Sigma_2)=(R(M_{0P}),E,P)$；

(4)根据 $RG(\Sigma_2)$ 化简 $\Sigma_2=(S_2,T_2,F_2,W_2,M_{02})$ 得 $\Sigma'=(S',T',F',M_0')$；

(5)利用资源解锁的变迁作为连接点，构造分布式系统的全局 Petri 网模型 $\Sigma=(S,T,F,M_0)$；

(6)画出 $\Sigma=(S,T,F,M_0)$ 的可达标识图；

(7)找出死锁的进程，杀死其中一个进程；

(8)从 $\Sigma=(S,T,F,M_0)$ 中删除被杀死的进程的状态及相关变迁，重复(6)、(7)直至系统中不存在死锁。

4.2 死锁检测原理

分布式系统的全局 Petri 网模型 $\Sigma=(S,T,F,M_0)$ 在经过一系列的变迁之后，将得到一个最终状态 $M_T=\{M_T(P_1),M_T(P_2),\cdots,M_T(P_m)\}$，即 $M_0[\sigma>M_T,\sigma\in T$，且$\neg\exists t\in T$ 使 $M_T[t>$。系统死锁的充分必要条件是 $M_T(P_1),M_T(P_2),\cdots,M_T(P_m)$ 中至少有两项大于 0。

证明：充分性：若 $M_T(P_1),M_T(P_2),\cdots,M_T(P_m)$ 中至少有两项不等于 0，则说明至少有两个进程的加锁请求没有得到授权，且 M_T 是终态，这些加锁请求永远不会得到授权，那么可以肯定这些进程在相互等待对方解锁、释放资源。因此系统存在死锁。

必要性：若系统出现死锁，则系统中的每个进程的状态都将稳定不变，且有多于一个的进程 $P_i,\cdots,P_j$ 在相互等待其它进程释放资源，$P_i,\cdots,P_j$ 的加锁请求不能被授权，那么 $M(P_i),\cdots,M(P_j)>0$，即 $M_T(P_1),M_T(P_2),\cdots,M_T(P_m)$ 中至少有两项大于 0。

可以通过杀死进程的方法来解决死锁。设杀死的进程 P_k 有 n 个子进程 $p_1,p_2,\cdots,p_n$，记

$$M_0(P)=\sum_{i=1}^{n}M_0(P_i)$$

为减少重复加锁，P_k 应满足条件 $M_0(P_k)-M_T(P_k)=\max\{M_0(P_j)-M_T(P_j)\mid j\in\{1,2,\cdots,m\}$。

结束语 本文给出了分布式锁管理 DLM 中的 6 种锁的 Petri 网模型，并通过共享合成、同步合成及 Petri 网化简的方法建立分布式系统锁的 Petri 网模型，借助 Petri 网来检测系统的死锁状态。应用等待图或资源分配图检测死锁时，系统只会在进程迟迟不能得到资源的情况下发起死锁检测；基于 Petri 网的分布式死锁检测具有动态模拟性的优点，它能跟踪进程申请资源的动态变化过程，实时发现死锁，并且能选择杀死最合适的进程。

参 考 文 献

[1] 徐高潮，胡亮，鞠九滨. 分布计算系统[M]. 北京：高等教育出版社，2004

[2] Oracle. A42522_1：Oracle7 Parallel Server Concepts and Administration，Release 7. 3 [EB]. http://download. oracle. com/docs/pdf/A42522_1. pdfsfd

[3] 吴哲辉. Petri 网导论[M]. 北京：机械工业出版社，2004

[4] 向学哲. 分布式锁管理在集群系统中的应用研究[J]. MODERN COMPUTER，2007(8)：20-22

[5] 钱迎进，肖侬，金士尧. Lustre 分布式锁管理器的分析与改进[J]. 计算机工程与科学，2009，31(A1)：146-149

[6] 李春强，杨邃，陈春玲. 分布式系统死锁的 Petri 网分析[J]. 微机发展，2003，13(5)：30-35

[7] 刘荣峰，李引珍，吴桂华. 基于 Petri 网的分布式系统并发进程死锁检测[J]. 计算机工程与设计，2007，28(22)：5353-5355

[8] 韩耀军，蒋昌俊，罗雪梅. 基于 Petri 网合成与化简的分布式数据库系统并发控制的死锁检测[J]. 小型微型计算机系统，2004，25(5)：821-826

(上接第 44 页)

[7] Crucitti P，Latora V，Porta S. Centrality in networks of urban streets[J]. Chaos：an interdisciplinary journal of nonlinear science，2006，16(1)：015113

[8] Borgatti S P. Centrality and network flow[J]. Social Networks，2005，27(1)：55-71

[9] Freeman L C. Centrality in social networks conceptual clarification[J]. Social Networks，1979，1(3)：215-239

[10] Freeman L C，Borgatti S P，White D R. Centrality in valued graphs：a measure of betweenness based on network flow[J]. Social Networks，1991，13(2)：141-154

[11] Nieminen J. On the centrality in a graph[J]. Scandinavian Journal of Psychology，1974，15(1)：332-336

[12] Latora V，Marchiori M. A measure of centrality based on network efficiency[J]. New Journal of Physics，2007，9(6)：188-198

[13] Newman M E J. Finding community structure in networks using the eigenvectors of matrices[J]. Physical Review E，2006，74(3)：036104

[14] Wang Gao-xia，Shen Yi，Luan Ee. A measure of centrality based on modularity matrix[J]. Progress in Natural Science，2008，18：1043-1047

[15] Qing Ou，Jin Ying-di，Zhou Tao，et al. Power-law strength-degree correlation from a resource-allocation dynamics on weighted networks[J]. Physical Review E，2007，75(2)：021102

[16] Zhou Tao，Lü Lin-yuan，Zhang Yi-chen. Predicting missing links via local information[J]. The European Physical Journal B，2009，71(4)：623-630

[17] Borgatti S P，Everett M G，Freeman L C. UCINET，Version1. 0 [Z]. Analytic Technologies，Columbia，SC，1992

[18] Everett M G，Borgatti S P. The centrality of groups and classed [J]. Journal of Mathematical Sociology，1999，23(3)：181-201I