仿真网格下的节点选择

宋长峰 李伯虎 柴旭东

(北京航空航天大学 自动化科学与电气工程学院, 北京 100191) (北京仿真中心, 北京 100854)

摘 要: 仿真网格是一个将网格技术和建模与仿真技术相结合的网络化仿真平台, 节点选择是提高仿真网格下仿真执行性能的重要手段. 根据仿真网格下资源异构性的特点, 提出一套形式化的节点性能评价标准, 以此标准为依据建立节点选择的数学模型, 并给出选择最优节点集合的节点选择算法. 算法实验表明, 该算法以节点的计算能力为优先考虑, 同时顾及节点之间的通信性能, 并能够回避通信性能参差不齐时好的通信性能掩盖住差的通信性能的情况, 能够起到提高仿真执行效率目的.

关 键 词: 仿真网格; 高层体系结构 (HLA); 节点选择

中图分类号: TP 301.6

文献标识码: A 文章编号: 1001-5965(2009)01-0056-05

Node selection in simulation grid

Song Changfeng Li Bohu

(School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Chai Xudong

(Beijing Simulation Center, Beijing 100854, China)

Abstract Simulation grid is a networked simulation platform by combining modeling and simulation (M&S) technology and grid technology. Node selection is an important method to increase the performance of simulation execution in simulation grid. Based on the feature of resource heterogeneousness in simulation grid, a formula appraising criterion for node selection was proposed, a mathematic model for node selection was built on the appraising criterion, and finally a node selection algorithm was designed. Experiment result shows that the algorithm puts emphasis on selecting nodes with good computation capacity, communication performance among nodes is considered at the same time, and the situation where good communication performance overlaps poor communication performance is avoided in this algorithm. So the node selection method is able to achieve the purpose of raising simulation execution efficiency.

Key words simulation grid; high level architecture (HLA); node selection

仿真网格 (simulation grid)[1]是一个将建模与仿真技术 M&S(Modeling and Simulation)和网格[2]技术相结合的网络化仿真平台. 仿真网格利用共享的仿真模型服务为用户构建基于 HLA (High Level Architecture)规范[3]的仿真系统, 同时为仿真系统分配计算资源, 对仿真执行进行管理和维护. 性能是仿真网格关心的问题, HLA 仿真系统是一种分布式交互仿真, 而仿真网格是一个异构的计算环境, 因此节点选择成为提高仿真系统性能的一个重要手段.

以往的分布式应用中, 节点选择问题通常是分成几种典型的类型研究[4], 由于 HLA 仿真系统中任何一个联邦成员[3]都可能与其它的联邦成员交互, 因此本文研究的节点选择归为 All-to-All 类型. 目前, All-to-All 类型的节点选择一般采用图论的方法[5], 通过反复选取子图, 不断排除性能最差的节点和通信, 直到形成一种全局平衡状态. 但是图论方法中的子图选择比较复杂, 且在一

收稿日期: 2008-01-31

基金项目: 国家 863计划信息领域专项资助项目 (2004AA104330)

作者简介: 宋长峰 (1977-), 男, 上海人, 博士生, biscf@21cn.com.

次循环选择中满足条件的子图可能存在多个, 而其中最优子图在后续选择中获得的子图不一定能保证是全局最优的; 另外, 对节点的计算能力和节点间通信性能的考虑比较独立, 会造成顾此失彼. 因此, 本文综合考虑计算和通信性能, 为节点选择建立数学模型, 并提出相应算法实现.

1　仿真网格下的节点选择模型

设可用节点数量 (即候选节点数量)为 N, 仿真应用需要的节点数量为 M ($N \geqslant M$), 节点选择就是从 N 个候选节点中选择整体性能最佳的 M 个节点.

1.1　节点性能描述

根据仿真网格下仿真应用的需要, 节点的性能从计算能力和节点间通信延迟两方面进行评判. 假设节点 n_k 上有 m 个任务 $\{T_i \mid i = 1, 2, \cdots, m\}$, 那么新任务 T 能够被分配到的计算能力 $c(T, n_k)$ 可用下式描述:

$$c(T, n_k) = \frac{p}{p + \sum_{i=1}^{m} p_i} \times c_{max} \qquad (1)$$

式中, p_i 为任务 T_i 的计算优先权级别; p 为任务 T 的计算优先权级别; c_{max} 为节点最大计算能力.

某个节点的通信延迟是对该节点与其所在节点集合中的所有其它节点之间的通信延迟的综合考虑. 节点 n_k ($1 \leqslant k \leqslant M$)的通信性能 $t(n_k)$ 用下式描述:

$$t(n_k) = \{d(k, i) \mid i = 1, 2, \cdots, M; i \neq k\} \qquad (2)$$

式中, $d(k, i)$ 为节点 n_k 到 n_i 的通信延迟.

1.2　节点性能评判标准

节点选择要求提供尽可能大的 $c(T, n_k)$ 和尽可能小的 $t(n_k)$, 但这是一个抽象的衡量指标, 本文为节点选择的性能比较提供了一个数学评判标准. 先将节点的计算能力和通信性能映射到同一量纲下描述. 计算能力的映射如下:

$$c_k = \frac{c(T, n_k)}{\max\{c(T, n_i) \mid i = 0, 1, \cdots, N\}} \qquad (3)$$

式 (3)的含义是节点 n_k 的计算能力相对于候选节点中最好的计算能力的程度, 即 c_k 表示相对计算能力 (值越大表示计算能力越强), 且 $0 < c_k \leqslant 1$.

对于通信性能的映射, 首先令

$$t'_{k,l} = \frac{d(k, l)}{\min\{d(i, j) \mid 1 \leqslant i, j \leqslant N; i \neq j\}} \qquad (4)$$

式 (4)表示节点 n_k 到 n_l 的通信延迟相对于候选节点间最小通信延迟的程度 (值越大表示通信性能越差), 且 $t'_{k,l} \geqslant 1$.

c_k 和 $t'_{k,l}$ 都是相对于各自最优状态的比例值, 因此将计算能力和通信性能统一在相同的量纲下. 但是 c_k 的值越大表示性能越好, 而 $t'_{k,l}$ 的值越小表示性能越好. 对 $t'_{k,l}$ 取倒数, 那么将取得与 c_k 相同的比较方向, 而且值的描述范围也与 c_k 相同, 即对节点 n_k 到 n_l 的通信性能的映射描述为

$$t_{k,l} = \frac{\min\{d(i, j) \mid 1 \leqslant i, j \leqslant N; i \neq j\}}{d(k, l)} \qquad (5)$$

以式 (3)作为计算能力的评判指标, 式 (5)作为通信性能的评判指标, 它们具有相同的量纲和比较方向, 下面使用这两个评判指标建立节点选择模型.

1.3　节点选择模型

令 $w_{k,k}$ 为节点 n_k ($1 \leqslant k \leqslant N$)的计算能力权重, $w_{k,l}$ ($1 \leqslant l \leqslant N$, $l \neq k$)为节点 n_k 到 n_l 的通信性能权重, 则 n_k 在所选择的 M 个节点中提供的性能用下式表示:

$$p_k = \begin{cases} 0 & x_k = 0 \\ w_{k,k} \times c_k + \sum_{i=1, i \neq k}^{N} w_{k,i} \times t_{k,i} \times x_i & x_k = 1 \end{cases} \qquad (6)$$

式中, 如果第 i 个节点被选中, 则 $x_i = 1$, 否则 $x_i = 0$, 且 $\sum_{i=1}^{N} x_i = M$.

被选中的 M 个节点的整体性能用下式表示:

$$p(\boldsymbol{X}) = \sum_{k=1}^{N} p_k = \boldsymbol{X} \times \boldsymbol{F} \times \boldsymbol{X}^{\mathrm{T}} \qquad (7)$$

式中, $\boldsymbol{X}$ 为 N 维的 0, 1行向量 $[x_1, x_2, \cdots, x_N]$, $x_i = 1$表示第 i 个节点被选中, 否则表示未被选中, $\sum_{i=1}^{N} x_i = M$, 随着 $\boldsymbol{X}$ 取值的不同, 方程有不同返回值; $\boldsymbol{F}$ 为一个 $N \times N$ 的性能加权矩阵:

$$\boldsymbol{F} = \begin{bmatrix} w_{1,1} \times c_1 & w_{1,2} \times t_{1,2} & \cdots & w_{1,k} \times t_{1,k} & \cdots & w_{1,N} \times t_{1,N} \\ w_{2,1} \times t_{2,1} & w_{2,2} \times c_2 & \cdots & w_{2,k} \times t_{2,k} & \cdots & w_{2,N} \times t_{2,N} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ w_{k,1} \times t_{k,1} & w_{k,2} \times t_{k,2} & \cdots & w_{k,k} \times c_k & \cdots & w_{k,N} \times t_{k,N} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ w_{N,1} \times t_{N,1} & w_{N,2} \times t_{N,2} & \cdots & w_{N,k} \times t_{N,k} & \cdots & w_{N,N} \times c_N \end{bmatrix}$$

以整体性能最好为目标, $\boldsymbol{X}$ 的取值为

$$\boldsymbol{X} = \{\boldsymbol{X} \mid p(\boldsymbol{X}) \geqslant p(\boldsymbol{X}_k), \boldsymbol{X}_k \text{为任意允许取值}\} \tag{8}$$

如果没有特殊说明, 认为所有候选节点都是平等的, 因此不同的节点具有相同的计算能力权重, 而且节点之间的通信具有相同的权重. 由于节点选择的目标是选出 M 个节点, 这 M 个节点中的任意一个节点都会与其它 $M-1$ 个节点进行通信, 所以节点之间的通信权重为 $\frac{1}{M-1}$. 此外, 节点的性能是由计算能力和通信性能组成, 设计算能力的权重为 w_c, 通信性能的权重 w_t ($w_c + w_t = 1$). 仿真应用中, 一般来说节点提供的计算作用较通信作用重要, 故取 $w_c \geqslant w_t$. 因此式 (8) 可变为

$$p(\boldsymbol{X}) = \boldsymbol{X} \times \left(w_c \times \boldsymbol{F}_c + w_t \times \frac{1}{M-1} \times \boldsymbol{F}_t\right) \times \boldsymbol{X}^{\mathrm{T}} \tag{9}$$

式中, $\boldsymbol{F}_c$ 为一个 N 维的对角矩阵, 描述节点计算能力:

$$\boldsymbol{F}_c = \begin{bmatrix} c_1 & & & & \\ & \ddots & & \boldsymbol{0} & \\ & & c_k & & \\ & \boldsymbol{0} & & \ddots & \\ & & & & c_N \end{bmatrix}$$

$\boldsymbol{F}_t$ 为一个 $N \times N$ 维矩阵, 描述节点的通信性能:

$$\boldsymbol{F}_t = \begin{bmatrix} 0 & t_{1,2} & \cdots & t_{1,k} & \cdots & t_{1,N} \\ t_{2,1} & 0 & \cdots & t_{2,k} & \cdots & t_{2,N} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ t_{k,1} & t_{k,2} & \cdots & 0 & \cdots & t_{k,N} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ t_{N,1} & t_{N,2} & \cdots & t_{N,k} & \cdots & 0 \end{bmatrix}$$

进一步考虑通信性能对节点选择时性能评估的影响. 例如有 4个节点, 第 1个节点到另外 3个节点的通信性能分别为 0.95, 0.85和 0.2, 通过累加获得整体通信性能为 2.0; 将后面 3个节点换成另外 3个节点, 这时第 1个节点到新的 3个节点的通信性能分别为 0.65, 0.8和 0.55, 通过累加同样获得整体通信性能为 2.0. 但是, 显然第 1组的通信性能不如第 2组好, 第 1组的通信性能值 0.2是一个性能瓶颈, 会造成性能降低. 可见, 式 (9) 中没有考虑节点通信性能差异较大时好的通信性能会掩盖住差的通信性能, 所以有必要对式 (9) 进行改进. 通常, 人们使用均方差来衡量数值的分散程度, 本文引入一个均方差向量 $\boldsymbol{D}$ 对式 (9) 实施改进, 改进后的数学表达为

$$p(\boldsymbol{X}) = \boldsymbol{X} \times \left(w_c \times \boldsymbol{F}_c + w_t \times \frac{1}{M-1} \times \boldsymbol{F}_t\right) \times \boldsymbol{X}^{\mathrm{T}} - \boldsymbol{D}^{\mathrm{T}} \tag{10}$$

式中, w_c, w_t 和 $\boldsymbol{F}_c$, $\boldsymbol{F}_t$ 与式 (9) 中的表示相同; $\boldsymbol{D}$ 为一个 N 维的向量 $[d_1, d_2, \cdots, d_N]$; 令 $F_t(i, j)$ 为 $\boldsymbol{F}_t$ 的第 i 行第 j 列的元素, 那么 $\boldsymbol{D}$ 中的任意一个元素 d_k 表达为

$$d_k = \begin{cases} \left(\dfrac{\sum_{i=1, i\neq k}^{N} (x_i \times (F_t(k, i) - E_k))^2}{M-2}\right)^{\frac{1}{2}} & x_k = 1 \\ 0 & x_k = 0 \end{cases} \tag{11}$$

式中, E_k 为均值, 表达式为

$$E_k = \frac{1}{M-1} \times \sum_{i=1}^{N} F_t(k, i) \times x_i \tag{12}$$

从式 (11) 和式 (12) 可以看出, $\boldsymbol{D}$ 也是一个关于 $\boldsymbol{X}$ 的函数.

式 (10) 和式 (8) 构成了节点选择模型.

2 节点选择算法描述

基于以上的节点选择模型, 本节给出节点选择算法步骤如下.

1) 初始化操作

初始化式 (10) 中的 w_c, w_t 和 $\boldsymbol{F}_c$, $\boldsymbol{F}_t$. 这些值为常量, 在运算中不会改变; 初始化一个 N 维的 0, 1行向量 $\boldsymbol{X}'$, $\boldsymbol{X}'$ 的前 $M-1$ 位元素为 1, 其余的元素为 0, $\boldsymbol{X}'$ 作为算法中生成自变量 $\boldsymbol{X}$ 的基础; 初始化变量 $q = M$, q 记录调整 $\boldsymbol{X}'$ 中的某一位元素值 (置 1操作) 产生自变量时被调整元素的位置; 初始化 N 维行向量 $\boldsymbol{S} = [0, 0, \cdots, 0]$ 和变量 $m = 0$, 其中 m 用于保存当前最大性能评估值, $\boldsymbol{S}$ 保存最大评估值所对应的自变量 $\boldsymbol{X}$.

2) 产生新的自变量 $\boldsymbol{X}$

对 $\boldsymbol{X}'$ 的第 q 位元素值进行置 1操作构成自变量 $\boldsymbol{X}$, 此时可保证 $\boldsymbol{X}$ 中为 1的元素数量为 M.

3) 计算整体节点性能

将 $\boldsymbol{X}$ 中的各个元素代入式 (12) 中计算, 获得 E_k ($1 \leqslant k \leqslant N$); 再将 E_k 和 $\boldsymbol{X}$ 中的各个元素代入式 (11) 中计算, 获得 d_k ($1 \leqslant k \leqslant N$); 由 d_k 组成式 (11) 中的 $\boldsymbol{D}$; 将 $\boldsymbol{X}$ 代入式 (10) 得到目标函数值 p; 如果 $p > m$, 则 $m = p$, $\boldsymbol{S} = \boldsymbol{X}$.

4) 对 $\boldsymbol{X}'$ 的第 q 位元素值置 0操作

令 $q = q + 1$, 如果 $q > N$, 转移至 5), 否则转到步骤 2);

5) 从后向前扫描值为 1的元素

根据逻辑关系此时 X'的最后一位元素必定为 0(步骤 4)执行的置 0操作). 扫描 X'自倒数第 2位元素向前连续为 1的元素的个数 k; 如果 $k = M - 1$, 则转移至步骤 6); 否则继续扫描 X中自第 $N - k - 1$位元素 (倒数第 $k + 1$位元素)向前第 1次出现 1的元素位置 e, 将 X的第 e位元素值置 0操作, 第 $e + 1$至 $e + 1 + k$位元素值置 1操作, 第 $e + k + 1$至 $N - 1$位元素置 0操作; 令 $q = e + k + 1$, 转至 2);

6) 算法结束

S即为性能评估最大时对应的自变量 X, 变量 m 为性能评估最大值.

3　节点选择实验

3 1　实验条件

提供 10个候选节点, 从中选取 6个整体性能最佳的节点. 利用网格资源状态查询工具 NWS (Network Weather Service) [6]等, 获得相关状态数据代入式 (3)和式 (5), 由计算值构成如下 F_c和 F_t矩阵. 由于在目前的实验的网络条件下, 节点之间的双向通信延迟是相同的, 所以 F_t是一个对称矩阵. 取权重 $w_c = 0.6$, $w_t = 0.4$

$$F_c = \begin{bmatrix} 0.365 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.934 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.781 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.684 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.763 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.400 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.599 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.824 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.866 \end{bmatrix}$$

$$F_t = \begin{bmatrix} 0 & 0.672 & 0.311 & 0.687 & 0.913 & 0.829 & 0.892 & 0.971 & 0.384 & 0.756 \\ 0.672 & 0 & 0.266 & 0.860 & 0.860 & 0.784 & 0.866 & 0.774 & 0.780 & 0.934 \\ 0.311 & 0.266 & 0 & 0.303 & 0.287 & 0.284 & 0.294 & 0.257 & 0.333 & 0.285 \\ 0.687 & 0.860 & 0.303 & 0 & 0.780 & 0.915 & 0.944 & 0.734 & 0.946 & 0.873 \\ 0.913 & 0.860 & 0.287 & 0.780 & 0 & 0.934 & 0.830 & 0.876 & 0.636 & 0.964 \\ 0.829 & 0.784 & 0.284 & 0.915 & 0.934 & 0 & 0.865 & 0.783 & 1.000 & 0.745 \\ 0.892 & 0.866 & 0.294 & 0.944 & 0.830 & 0.865 & 0 & 0.884 & 0.873 & 0.800 \\ 0.971 & 0.774 & 0.257 & 0.734 & 0.876 & 0.783 & 0.884 & 0 & 0.538 & 0.946 \\ 0.384 & 0.780 & 0.333 & 0.946 & 0.636 & 1.000 & 0.873 & 0.538 & 0 & 0.414 \\ 0.756 & 0.934 & 0.285 & 0.873 & 0.964 & 0.745 & 0.800 & 0.946 & 0.414 & 0 \end{bmatrix}$$

3 2　实验结果

在以上实验条件下进行算法实验, 实验结果见表 1.实验中共有 210种互异的节点组合方式, 从以上 3个性能值的比较可看出, 不同的组合方式下性能值差异较大, 说明有必要进行节点选择.

表 1　节点选择算法实验结果

实验记录项	实验结果
性能计算次数	210
最差性能值	2.29874
平均性能值	3.19265
最优性能值	4.48028
节点选择结果	2, 4, 5, 6, 8, 10

3 3　实验结果分析

首先, 通过对 F_c的分析可以看出, 10个候选节点单纯从计算性能的角度来比较, 可以分成性能好 (节点 2、节点 4), 性能较好 (节点 3、节点 6、节点 9、节点 10), 性能中等 (节点 5、节点 8)和性能差 (节点 1、节点 7)4个类别.

从实验结果来看, 属于“性能好”的节点全部被选中, 而属于“性能差”的节点都没有被选中. 尤其是节点 7, 从 F_t中可看出尽管与其它节点间的通信性能较优秀而且性能分布比较平均, 但由于其计算性能差而没有被选中. 同时, 节点 3属于“性能较好”类别, 在同类别中也不是排名最后的节点, 但却不存在于最后选中的节点中. 通过分析 F_t中节点 3对应的通信性能, 发现其通信性能非常差, 远远落后于其它节点的通信性能, 故未被选中. 另外, 属于“性能较好”的节点 9也没有被选中. 节点 9与其它节点之间的通信性能不像节点 3那样比较差, 其通信性能从整体角度而言也是不错的, 但节点 9与其它节点之间的通信性能的

差异比较大(有很好的通信性能,也有很差的通信性能),容易造成通信瓶颈,故也未被选中.

可见,算法以计算性能考虑为重但也不排除通信性能的影响因素,并能避免某一节点与其它节点之间较好的通信性能掩盖住较差的通信性能的情况.

4 结束语

本文以提高仿真网格下仿真系统的性能为目标,针对仿真网格下资源的异构性,提出了一套形式化的节点选择评价标准,以此标准为依据建立了节点选择的数学模型,同时给出了选择最优节点集合的节点选择算法,算法结果表明本文设计的节点选择模型和算法是有效的.

参考文献 (References)

[1] Li Bohu, Chai Xudong, Zhu Wenhai, et al. Research on service oriented simulation grid[C]//Qian Qingquan. The 7th International Symposium on Autonomous Decentralized Systems (ISADS 2005). USA: IEEE Computer Society Press, 2005: 7-14

[2] Foster I. The grid: a new infrastructure for 21st century science [J]. Physics Today, 2002, 32(1): 42-47

[3] Kilby M. Overview of the DOD high level architecture[EB/OL]. US Department of Defense, 2000 [2005-12-20]. http://www.dmso.mil

[4] Weismann J B. Metascheduling: A scheduling model for metacomputing systems[C]//Salim Hariri. The Seventh IEEE International Symposium on High Performance Distributed Computing. USA: IEEE Conputer Society Press, 1998: 348-349

[5] Reiher P L, Jefferson D. Virtual time based dynamic load management in the time warp operation system[C]//Proceedings of the 1990 SCS Multiconference on Distributed Simulation. US: Trans of the Soc for Computer Simulation, 1990: 103-111

[6] Wolski R, Spring N, Hayes J. The network weather service: a distributed resource performance forecasting service for metacomputing[J]. Future Generation Computer Systems, 1999, 15(5-6): 757-768

(责任编辑:刘登敏)

(上接第 55页)

5 结束语

传统可靠性理论认为 $C(k, n: G)$系统只要有连续 k个部件处于工作状态,系统就工作.而基于模糊状态的可靠性理论则认为系统中只要有故障部件,系统就只是模糊成功,并非完全正常,只有系统中所有部件都工作时,系统才完全正常.因此模糊可靠性指标的估计比传统可靠度指标偏保守.但它综合反映了可修 $C(k, n: G)$系统的状况,不仅考虑故障的出现概率,而且考虑了各类故障对系统工作的影响程度.本文得到了计算 $C(k, n: G)$系统的模糊可靠性指标的公式.由于复杂大系统的可靠性模型一般可以化为部件的串联和部件并联模型的复合形式.而当 $k=1$时, $C(k, n: G)$系统为并联系统;当 $k=n$时, $C(k, n: G)$系统为串联系统.因此串并联系统是 $C(k, n: G)$系统的特殊情况.本文提出的概念和方法为复杂大系统的可靠性分析提供了一套新的思路.

参考文献 (References)

[1] Kontoleon J M. Reliability determination of a r-successive-out-of-n: F system[J]. IEEE Transactions on Reliability, 1980, 29(5): 290-294

[2] Kuo Way, Zhang Weixiang, Zuo Mingjian. A consecutive-k-out-of-n: G system: the mirror image of a consecutive-k-out-of-n: F system[J]. IEEE Transactions on Reliability, 1990, 39(2): 244-253

[3] Zhang Yuanlin, Lam Yeh. Reliability of consecutive-k-out-of-n: G repairable system[J]. International Journal of Syetem Science, 1998, 29(12): 1375-1379

[4] Cheng K, Zhang Y L. Analysis for a consecutive-k-out-of-n: F repairable system with priority in repair[J]. International Journal of Systems Science, 2001, 32(5): 591-598

[5] Yam Richard C M, Zuo M J, Zhang Y L. A method for evaluation of reliability indices for repairable circular consecutive-k-out-of-n: F system[J]. Reliability Engineering and System Safety, 2003, 79(1): 1-9

[6] Zhang Y L, Zuo M J, Yam Richard C M. Reliability analysis for a circular consecutive-2-out-of-n: F repairable system with priority in repair[J]. Reliability Engineering and System Safety, 2000, 68(2): 113-120

[7] Wu Y Q, Guan J C. Repairable consecutive-k-out-of-n: G systems with r repairmen[J]. IEEE Transactions on Reliability, 2005, 54(2): 328-337

[8] Guan J C, Wu Y Q. Repairable consecutive-k-out-of-n: F system with fuzzy state[J]. Fuzzy Sets and Systems, 2006, 157(1): 121-142

[9] Cai K Y, Wen C Y, Zhang M L. Survivability index for CCNs: a measure of fuzzy reliability[J]. Reliability Engineering and System Safety, 1991, 33(1): 71-99

[10] Das C R, Bhuyan L N. Reliability simulation of multiprocessor systems[C]//Proceedings of the International Conference on Parallel Processing. New York: IEEE, 1985: 591-598

[11] Cai K Y. Introduction to fuzzy reliability[M]. London: Kluwer Academic Publishers, 1996

(责任编辑:文丽芳)