

## $k$ 元 $n$ 立方网络的 $k$ 圈排除问题的递归算法

杨玉星\*, 王世英

(河南师范大学 数学与信息科学学院, 河南 新乡 453007)

(\* 通信作者电子邮箱 yxyang@163.com)

**摘要:**为了度量以 $k$ 元 $n$ 立方网络为底层网络拓扑的并行计算机系统的容错能力,通过构造 $k$ 元 $n$ 立方网络中使得所有的 $k$ 元1立方子网都发生故障的最小节点集的方法,提出求解其 $k$ 元1立方子网排除点割集的一种递归算法;证明了要使 $k$ 元 $n$ 立方网络中所有 $k$ 元1立方子网都发生故障至少需要破坏掉 $k^{n-1}$ 个节点。结果表明,在不超过 $k^{n-1}-1$ 个节点被破坏的情况下,以 $k$ 元 $n$ 立方网络为底层拓扑构建的并行计算机系统中依然存在无故障的 $k$ 元1立方子网。

**关键词:**并行计算机系统;互连网络;容错; $k$ 元 $n$ 立方;节点故障;可靠性

**中图分类号:**TP393.022;TP301.6 **文献标志码:**A

### Recursive algorithm for $k$ -cycle preclusion problem in $k$ -ary $n$ -cubes

YANG Yuxing\*, WANG Shiyang

(School of Mathematics and Information Science, Henan Normal University, Xinxiang Henan 453007, China)

**Abstract:** In order to measure the fault tolerance ability of the parallel computers which take the  $k$ -ary  $n$ -cube as underlying topology, by constructing the minimum node set whose removal will lead to every  $k$ -ary 1-cube in the  $k$ -ary  $n$ -cube faulty, a recursive algorithm for finding the  $k$ -ary 1-cube subnet preclusion node cut of the  $k$ -ary  $n$ -cube was proposed. It is proved that at least  $k^{n-1}$  nodes need to be damaged if a rival wants to destroy all  $k$ -ary 1-cubes in the  $k$ -ary  $n$ -cube. The result indicates that there are still undamaged  $k$ -ary 1-cubes in the parallel computers which take the  $k$ -ary  $n$ -cube as underlying topology if the number of the faulty nodes does not exceed  $k^{n-1}-1$ .

**Key words:** parallel computer system; interconnection network; fault tolerance;  $k$ -ary  $n$ -cube; node failure; reliability

## 0 引言

并行计算机系统中元件之间的连接模式称为该系统的互连网络(Interconnection network),简称为网络(Network)。从拓扑上讲,一个多处理机系统的互连网络拓扑逻辑上确定了该系统中所有元件之间的连接方式。一个网络常用图 $G=(V, E)$ 来表示,其中,一个节点对应一个处理机,一条边对应两个节点之间的一条直接通信线路。

在大型并行计算机系统中,元件故障是在所难免的。因此,网络的容错性成为一个重要的话题并得到了广泛的研究<sup>[1-6]</sup>。网络的容错性通常以在一定数量的元件发生故障时互连网络性质的保持程度来度量。显然,有故障发生时,原网络将遭到一定程度的破坏。在此背景下,Latifi<sup>[1]</sup>于2007年提出了一个有意义的问题:至少多少个元件发生故障时,网络中将不再有给定规模的子网(和原有网络具有相同拓扑结构,但节点数较少的网络)?为便于描述,本文称该问题为“子网排除”问题。2008—2010年,Walker等<sup>[2-3]</sup>对Star网络的子网排除问题进行了较为深入的研究。Wang等<sup>[4]</sup>研究了bubble-sort互连网络的子网排除问题,他们使用构造映射的方法得到了bubble-sort互连网络排除一定规模的子网所需要的最小节点集,从而得到了这些子网排除数的确切值;同时,还到了某些规模的子网排除数的上、下界。目前,大多数对于子网排除问题的研究还仅局限于子网排除数的上、下界的确定问题上,未对最小子网排除集的求解进行深入的研究。

$k$ 元 $n$ 立方网络( $k$ -ary  $n$ -cube)是并行计算机系统最为常用的底层拓扑之一,它具有易实现、低延迟、高带宽等优秀性质<sup>[7-9]</sup>。许多以 $k$ 元 $n$ 立方网络的多处理机系统已经问世。譬如, $i$ Warp、Cray T3D、Cray T3E、IBM Blue Gene<sup>[10]</sup>等。当 $k \geq 3$ 时, $k$ 元 $n$ 立方网络中的 $k$ 元1立方子网是一个长为 $k$ 的圈(简称 $k$ 圈)。在破坏对手的以 $k$ 元 $n$ 立方网络为底层拓扑结构的系统时,如果能够破坏掉该系统中的所有 $k$ 圈,那么该系统将出现通信故障,甚至瘫痪。若要破坏 $k$ 元 $n$ 立方网络中的所有 $k$ 圈,最重要的是选取需要破坏的节点。显然,在保证破坏掉所有 $k$ 圈的前提下,需要打击的节点数目越少越好。

目前,有关 $k$ 元 $n$ 立方网络的子网排除数的研究仅见文献<sup>[11]</sup>,在该文中得到了当 $k$ 为不小于3的奇数时 $k$ 元 $n$ 立方网络的 $k$ 元 $n-m$ 立方网络排除数的一个较紧的上界。本文研究 $k$ 元 $n$ 立方网络的 $k$ 圈排除问题,给出一种选择“最少的节点以破坏 $k$ 元 $n$ 立方网络中所有 $k$ 圈”的递归算法,并计算出了破坏 $k$ 元 $n$ 立方网络中所有 $k$ 圈所需要破坏的最少节点的数目。下文中没有定义而直接使用的图论术语参见文献<sup>[12]</sup>。

## 1 问题描述

### 1.1 $k$ 元 $n$ 立方网络

$k$ 元 $n$ 立方 $Q_n^k(k \geq 1, n \geq 1)$ 是一个具有 $k^n$ 个节点的无

收稿日期:2013-03-12;修回日期:2013-05-06。 基金项目:国家自然科学基金资助项目(61070229);教育部高等学校博士点专项基金资助项目(20111401110005);河南师范大学博士科研启动费支持项目(qd12105)。

作者简介:杨玉星(1981-),男,河南商丘人,讲师,博士,CCF会员,主要研究方向:图与网络优化、网络容错;王世英(1961-),男,山西左权人,教授,博士生导师,博士,主要研究方向:图论。

向图。每个节点可以标记为  $u = \sigma_{n-1}\sigma_{n-2}\cdots\sigma_0$ , 其中,  $\sigma_i \in \{0, 1, \dots, k-1\}$ 。两个节点  $u = \sigma_{n-1}\sigma_{n-2}\cdots\sigma_0$  和  $v = \delta_{n-1}\delta_{n-2}\cdots\delta_0$  由一条  $j$  维边相连当且仅当存在一个整数  $j \in \{0, 1, \dots, n-1\}$ , 使得  $\sigma_j = \delta_j \pm 1$  (模  $k$ ) 且对每个  $i \in \{0, 1, \dots, n-1\} \setminus \{j\}$  均有  $\sigma_i = \delta_i$  成立。当  $k$  为偶数时,  $k$  元  $n$  立方是二部的。 $k$  元  $n$  立方网络具有正则性、点传递性、边传递性和递归性等拓扑性质。其中, 递归性是  $k$  元  $n$  立方的一个重要性质。更具体地讲, 给定任意一个整数  $d \in \{1, 2, \dots, n\}$ , 可以删除  $Q_n^k$  的所有  $d$  维边, 从而将  $Q_n^k$  沿第  $d$  维划分成  $k$  个不交的  $Q_{n-1}^k$ , 即:  $Q_{n-1}^k[0], Q_{n-1}^k[1], \dots, Q_{n-1}^k[k-1]$ 。3 元 2 立方网络如图 1 所示。

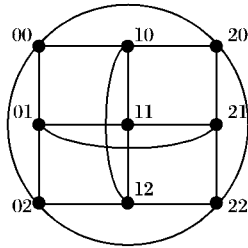


图 1 3 元 2 立方网络

令  $N_0 = \emptyset$ ;  $j \geq 1$  时,  $N_j = \{0, 1, \dots, j-1\}$ 。给定一个整数  $k \geq 1$ , 设  $A_k = \{0, 1, \dots, k-1, X\}$ , 其中,  $X$  表示  $\{0, 1, \dots, k-1\}$  中某个符号。 $Q_n^k$  中每个子立方  $Q_{n-m}^k$  可以由  $A_n$  的符号组成的、只有  $X$  可以重复的符号串表示。更为确切地, 在表示  $Q_n^k$  中的一个  $Q_{n-m}^k$  的  $n$  位符号串中,  $X$  恰出现  $n-m$  次。例如, 符号串  $0X1XX2$  表示  $Q_3^k$  中由节点子集  $\{0x_11x_2x_32 \mid x_1, x_2, x_3 \in N_k\}$  导出的  $Q_3^k$ 。

1.2  $k$  圈排除

正如 1.1 节所述,  $Q_n^k$  中的任意一个  $Q_{n-m}^k$  可以用一个  $n$  位的符号串表示, 其中符号  $X$  出现  $n-m$  次。注意, 当  $k \geq 3$  时,  $Q_1^k$  恰为  $Q_n^k$  中的一个  $k$  圈; 当  $k \geq 3$  且为奇数时,  $Q_n^k$  中的每个  $k$  圈恰为一个  $Q_1^k$ 。更为确切地, 对于任意的整数  $0 \leq i \leq n-1$ , 给定  $N_k$  中的  $n-1$  个符号  $a_1, a_2, \dots, a_{n-1}$ , 令  $M_i = \{a_1a_2\cdots a_i b a_{i+1} a_{i+2} \cdots a_{n-1} \mid b \in N_k\}$ , 其中, 当  $i=0$  时  $a_1a_2\cdots a_i$  是空串; 当  $i=n-1$  时  $a_{i+1}a_{i+2}\cdots a_{n-1}$  是空串。则  $Q_n^k$  中由  $M_i$  导出的子图是一个  $k$  圈。为便于表述, 对于任意的整数  $0 \leq i \leq n-1$ , 本文将  $Q_n^k$  中由  $M_i$  导出的子图表示成一个  $n$  位的字符串:

$$a_1 a_2 \cdots a_i X a_{i+1} a_{i+2} \cdots a_{n-1}$$

注意, 当  $i=0$  时,  $a_1 a_2 \cdots a_i X a_{i+1} a_{i+2} \cdots a_{n-1}$  恰为  $X a_1 a_2 \cdots a_{n-1}$ ; 当  $i=k$  时,  $a_1 a_2 \cdots a_i X a_{i+1} a_{i+2} \cdots a_{n-1}$  恰为  $a_1 a_2 \cdots a_{n-1} X$ 。即, 每个  $k$  圈恰能表示成一个  $n$  位的字符串。

定义 1 设  $F$  是  $Q_n^k$  的一个节点子集, 若  $Q_n^k - F$  不含  $Q_{n-m}^k$ , 则称  $F$  是  $Q_n^k$  的一个  $Q_{n-m}^k$  排除集。

定义 2  $Q_n^k$  的最小  $Q_{n-m}^k$  排除集的元素个数叫作该  $Q_n^k$  的  $Q_{n-m}^k$  排除数。

特别地,  $Q_1^k$  排除集也叫  $k$  圈排除集,  $Q_1^k$  排除数也叫  $k$  圈排除数。

2  $k$  圈排除问题的递归算法

2.1 重要引理

引理 1 给定 3 个整数  $n \geq 2, k \geq 2$  及  $0 \leq m \leq n-1$ , 设  $F(n, m)$  是  $Q_n^k$  的  $Q_{n-m}^k$  排除数, 则

$$k^m \leq F(n, m) \leq k^m C_n^m$$

证明 首先用归纳法证明  $Q_n^k$  中含有  $k^m$  个互不相交的  $Q_{n-m}^k$ 。对  $m$  进行归纳。当  $m=0$  时, 显然,  $Q_n^k$  中恰有  $k^0 = 1$  个  $Q_n^k$ 。假设对于任意的整数  $1 \leq m \leq n-2$ ,  $Q_n^k$  中恰有  $k^m$  个互不相交的  $Q_{n-m}^k$ 。对于上述  $k^m$  个互不相交的  $Q_{n-m}^k$  中的任意一个, 本文可以将其沿着某一维划分成  $k$  个互不相交的  $Q_{n-(m+1)}^k$ 。因此,  $Q_n^k$  中共有  $k \times k^m = k^{m+1}$  个互不相交的  $Q_{n-(m+1)}^k$ 。综上可得,  $Q_n^k$  中含有  $k^m$  个互不相交的  $Q_{n-m}^k$ 。欲破坏这些  $Q_{n-m}^k$ , 每个  $Q_{n-m}^k$  至少需要一个故障节点。于是,  $F(n, m) \geq k^m$ 。

接下来, 用归纳法证明  $Q_n^k$  中含有  $k^m C_n^m$  个互不相同的  $Q_{n-m}^k$ 。对  $m$  进行归纳。当  $m=0$  时, 显然,  $Q_n^k$  中恰有  $k^0 = 1$  个  $Q_n^k$ 。假设对于任意的整数  $1 \leq m \leq n-2$ ,  $Q_n^k$  中恰有  $k^m C_n^m$  个互不相同的  $Q_{n-m}^k$ 。对于上述  $k^m C_n^m$  个不同的  $Q_{n-m}^k$  中的任意一个, 可以将其沿某一维划分成  $k$  个互不相同的  $Q_{n-(m+1)}^k$ , 这样的划分方式共有  $n-m$  种 (在这样的所有划分中, 会有  $m+1$  倍的重复), 即: 沿着  $Q_{n-m}^k$  的每一维进行划分。因此,  $Q_n^k$  中共有

$$\frac{(n-m)k \times k^m C_n^m}{m+1} = (n-m)k^{m+1} C_n^m = k^{m+1} C_n^{m+1}$$

个互不相同的  $Q_{n-(m+1)}^k$ 。综上可得,  $Q_n^k$  中含有  $k^m C_n^m$  个互不相同的  $Q_{n-m}^k$ 。欲破坏这些  $Q_{n-m}^k$ , 只需每个  $Q_{n-m}^k$  中有一个故障节点即可, 于是,  $F(n, m) \leq k^m C_n^m$ 。

综上可得,  $k^m \leq F(n, m) \leq k^m C_n^m$ 。证毕

当  $m=n-1$  时, 有如下推论:

推论 1 对于两个数  $n$  和  $m$ , 其中  $n \geq 2$  为整数,  $k \geq 3$  为奇数。设  $F(n, 1)$  是  $Q_n^k$  的  $k$  圈排除数, 则  $k^{n-1} \leq F(n, 1) \leq nk^{n-1}$ 。

2.2 算法描述

当  $k \geq 3$  为奇数时, 本文给出一个求解  $Q_n^k$  的一个最小  $k$  圈排除集的算法。

算法 1 求解  $Q_3^k$  的一个最小  $k$  圈排除集的算法。

输入 一个不小于 3 的奇数  $k$ 。

输出  $Q_3^k$  的一个  $k$  圈排除集  $F_3$ 。

1) 令  $V_i = \{000, 011, \dots, 0(k-1)(k-1)\}$ ;

2) For ( $i = 1; i \leq k-1; i++$ )

3)  $V_i = \{((\sigma_2 + i) \bmod k)\sigma_1((\sigma_0 + 1) \bmod k) \mid \sigma_2 \sigma_1 \sigma_0 \in V_0\}$ ;

4) End For

5)  $F_3 = \bigcup_{j=0}^{k-1} V_j$ ;

当  $k \geq 3$  为奇数,  $n \geq 4$  为整数时, 设  $F_{n-1}$  是  $Q_{n-1}^k$  的一个最小  $k$  圈排除集。对于任意的  $0 \leq i \leq k-1$ , 令

$$F_{n-1}^i = \{u_i = \sigma_{n-2}\sigma_{n-3}\cdots((\sigma_0 + i) \bmod k) \mid \sigma_{n-2}\sigma_{n-3}\cdots\sigma_0 \in F_{n-1}\}$$

算法 2 求解  $Q_n^k$  的一个最小  $k$  圈排除集的算法。

输入 不小于 4 的整数  $n$  和不小于 3 的奇数  $k$ 。

输出  $Q_n^k$  的一个  $k$  圈排除集  $F_n$ 。

1) For ( $i = 0; i \leq k-1; i++$ )

2) For ( $t = 1; t \leq \lceil F_{n-1}^i \rceil; t++$ )

3)  $u_i^* = \text{concat}(i, u_t)$ ;

4) End For

5)  $V_i = \bigcup_{t=1}^{\lceil F_{n-1}^i \rceil} u_t^*$ ;

6) End For

$$7) F_n = \bigcup_{j=0}^{k-1} V_j;$$

2.3 算法分析

首先证明算法的正确性。

**定理1** 算法1的输出为  $Q_3^k$  的一个最小的  $k$  圈排除集,  $Q_3^k$  的  $k$  圈排除数  $F(3,1) = k^2$ 。

证明 首先证明算法1输出的集合  $F_3$  为  $Q_3^k$  的一个  $k$  圈排除集。当  $k \geq 3$  且为奇数时,  $Q_3^k$  中的每个  $k$  圈恰能表示成一个长为3的符号串,其中,符号  $X$  恰出现一次。注意,  $Q_3^k$  中的节点  $\sigma_2\sigma_1\sigma_0$  可以破坏  $Q_3^k$  中3个  $k$  圈,即:  $X\sigma_1\sigma_0, \sigma_2X\sigma_0, \sigma_2\sigma_1X$ 。若能找到  $Q_3^k$  的一个节点子集  $F$ ,使得该节点子集节点标号的任意2位的排列均遍历集合  $S_0 = \{x_1x_2 \mid x_1, x_2 \in N_k\}$ ,则  $F$  是  $Q_3^k$  的一个  $k$  圈排除集。算法1的第1步所得的节点集合  $V_0$  中节点的前两位分别为  $00, 01, \dots, 0(k-1)$ ; 在第2)~3)步的循环中,所得节点集合  $V_i$  中节点的前两位分别为  $i0, i1, \dots, i(k-1)$ ,其中,  $i \in \{1, 2, \dots, k-1\}$ 。所以第4)步中  $F_3 = \bigcup_{j=0}^{k-1} V_j$  所得节点集合的前两位遍历  $S_0$ ; 类似可证  $F_3$  中节点的第1位和第3位的排列以及第2位和第3位的排列均遍历  $S_0$ 。因此,  $F_3$  是  $Q_3^k$  的一个  $k$  圈排除集。

下面,证明算法1输出的  $k$  圈排除集是最小的。由  $V_i (i \in \{1, 2, \dots, k-1\})$  中节点的选取方法知  $F_3$  中没有重复的节点,所以  $F(3,1) \leq |F_3| = k^2$ 。由推论1知,  $F(3,1) \geq k^2$ 。因此,  $F(3,1) = k^2$ 。进而可知,  $F_3$  是  $Q_3^k$  的一个最小  $k$  圈排除集。

**定理2** 算法2的输出为  $Q_n^k$  的一个最小的  $k$  圈排除集,  $Q_n^k$  的  $k$  圈排除数  $F(n,1) = k^{n-1}$ ,其中  $n \geq 4$  为整数。

证明 首先证明算法2输出的集合  $F_n$  为  $Q_n^k$  的一个  $k$  圈排除集。当  $k \geq 3$  且为奇数时,  $Q_n^k$  中的每个  $k$  圈恰能表示成一个长为  $n$  的符号串,其中,符号  $X$  恰出现一次。注意,  $Q_n^k$  中的节点  $\sigma_{n-1}\sigma_{n-2}\dots\sigma_0$  可以破坏  $Q_n^k$  中  $n$  个  $k$  圈,即:  $X\sigma_{n-2}\dots\sigma_0, \sigma_{n-1}X\sigma_{n-3}\sigma_{n-4}\dots\sigma_0, \sigma_{n-1}\sigma_{n-2}X\sigma_{n-4}\dots\sigma_0, \dots, \sigma_{n-1}\sigma_{n-2}\dots X$ 。不难看出,若能找到  $Q_n^k$  的一个节点子集  $F$ ,使得该节点子集节点标号的任意  $n-1$  位的排列均遍历集合  $S_1 = \{x_1x_2\dots x_{n-1} \mid x_1, x_2, \dots, x_{n-1} \in N_k\}$ ,则  $F$  是  $Q_n^k$  的一个  $k$  圈排除集。由定理1知,  $F_3$  是  $Q_3^k$  的一个  $k$  圈排除集。作为归纳假设,设  $F_{n-1}$  的节点标号的任意  $n-2$  位均遍历集合  $S_2 = \{x_1x_2\dots x_{n-2} \mid x_1, x_2, \dots, x_{n-2} \in N_k\}$ 。算法2的输入为  $n$  和  $k$  时,其输出  $F_n$  的节点标号的  $n-1$  位中必定包含第1位或第  $n$  位,本文考虑以下情形。

情形1 这  $n-1$  位不包含第1位(最左端)。

在此情形下,这  $n-1$  位为  $F_n$  的后  $n-1$  位。故只需证

$\bigcup_{i=0}^{k-1} F_{n-1}^i$  遍历  $S_1$  即可。由归纳假设  $F_{n-1}$  的前  $n-1$  位遍历  $S_2$ 。结合  $F_{n-1}^i = \{u_i = \sigma_{n-2}\sigma_{n-3}\dots((\sigma_0 + i) \bmod k) \mid \sigma_{n-2}\sigma_{n-3}\dots\sigma_0 \in F_{n-1}\}$  可知  $\bigcup_{i=0}^{k-1} F_{n-1}^i$  遍历  $S_1$ 。

情形2 这  $n-1$  位不包含第  $n$  位(最右端)。

在此情形下,这  $n-1$  位为  $F_n$  的前  $n-1$  位。由归纳假设  $F_{n-1}$  的前  $n-1$  位遍历  $S_2$ ,所以  $F_{n-1}^i$  的前  $n-1$  位遍历  $S_2$ 。又

因为  $u_i^* = \text{concat}(i, u_i)$ ,其中,  $u_i \in F_{n-1}^i$ 。于是,  $V_i = \bigcup_{i=1}^{|F_{n-1}^i|} u_i^*$

的前  $n-1$  位遍历  $S_2$ 。结合  $F_n = \bigcup_{j=0}^{k-1} V_j$  知  $F_n$  的前  $n-1$  位遍历  $S_1$ 。

情形3 这  $n-1$  位同时包含第1位和第  $n$  位。

此时,这  $n-1$  位不包含从第2位到第  $n-1$  位中的某一位,不妨设不包含第2位。只需证明  $F_n$  中除第2位之外的那  $n-1$  位的排列遍历  $S_1$ 。即:  $T = \bigcup_{i=0}^{k-1} \{\text{concat}(i, \sigma_{n-3}\dots((\sigma_0 + i) \bmod k)) \mid \sigma_{n-2}\sigma_{n-3}\dots\sigma_0 \in F_{n-1}\}$  遍历  $S_1$ 。其中,  $\text{concat}$  函数表示两个字符串的拼接。由归纳假设知  $\bigcup_{i=0}^{k-1} \{\sigma_{n-3}\sigma_{n-4}\dots\sigma_0 \mid \sigma_{n-2}\sigma_{n-3}\dots\sigma_0 \in F_{n-1}\}$  遍历  $S_2$ 。所以有  $\bigcup_{i=0}^{k-1} \{\sigma_{n-3}\sigma_{n-4}\dots((\sigma_0 + i) \bmod k) \mid \sigma_{n-2}\sigma_{n-3}\dots\sigma_0 \in F_{n-1}\}$  遍历  $S_2$ 。进而可知  $T$  遍历  $S_1$ 。

综上可知,  $F_n$  的节点标号的任意  $n-1$  位均遍历  $S_1$ 。因此,  $F_n$  是  $Q_n^k$  的一个  $k$  圈排除集。

下面,证明算法2输出的  $k$  圈排除集是最小的。

由  $F_n$  的构造方法易得  $|F_n| = k^{n-1}$ ,所以  $F(n,1) \leq |F| = k^{n-1}$ 。由推论1得  $F(n,1) \geq k^{n-1}$ 。故  $F(n,1) = k^{n-1}$ 。进而可知  $F_n$  是  $Q_n^k$  的一个最小  $k$  圈排除集。证毕。

由上述两个定理及算法1和2,不难看出,本文提出的求解  $k$  元  $n$  立方网络的  $k$  圈排除问题的递归算法的时间复杂度为  $O(k^{n-1})$ ,复杂度较高;然而,作为并行计算机系统的  $k$  元  $n$  立方网络的  $k$  和  $n$  的取值比较小(例如,以3元15立方网络为底层拓扑结构可以构建拥有一千四百多万个处理器的超级并行计算机),因此,该算法仍不失为一种实用的算法。

3 结语

目前,随着高性能并行计算机技术的发展,人们越来越关注具有优良性能的互连网络。按照具有优秀性能的互连网络为底层拓扑结构设计并行计算机系统或大规模集成电路,可以减少路由时间,增强电路板的可移植性,提高系统的性能。网络容错能力是设计和选择互连网络时需要考虑的一个主要问题。在一定程度上,子网排除问题不仅可以反映网络的容错能力,在信息科学中有着重要的应用背景;而且在化学、分子生物学等学科也有着一定的应用价值。

本文研究了  $k$  为奇数时  $k$  元  $n$  立方网络中1元  $n$  立方子网(即  $k$  圈)的排除问题。提出了  $k$  元  $n$  立方网络中  $k$  圈排除问题的一种递归算法,并计算出当  $k \geq 3$  为奇数、 $n \geq 3$  为整数时  $k$  元  $n$  立方网络的  $k$  圈排除数  $F(n,1) = k^{n-1}$ 。本文算法虽然具有较高的时间复杂度,然而,在实际的应用中,以  $k$  元  $n$  立方网络作为底层拓扑构建并行计算机系统时,  $k$  和  $n$  的取值均很小,因此,本文算法依然是非常有效的。  $k$  元  $n$  立方网络中较大规模的子网排除问题以及其他著名互连网络的子网排除问题是值得进一步研究的问题。

参考文献:

[1] LATIFI S. A study of fault tolerance in star graph [J]. Information Processing Letters, 2007, 102(5): 196-200.  
 [2] LATIFI S, SABERINIA E, WU X L. Robustness of star graph network under link failure [J]. Information Sciences, 2008, 178 (3): 802-806.  
 [3] WALKER D, LATIFI S. Improving bounds on link failure tolerance of the star graph [J]. Information Sciences, 2010, 180(13): 2571-2575.

可以接受的范围内。本文下一步的工作将主要集中在建立功耗与性能的关系模型,设计出更优的低功耗算法方案。

#### 参考文献:

- [1] GONZALEZ R, HOROWITZ M. Energy dissipation in general purpose microprocessors [J]. IEEE Journal of Solid State Circuits, 1996, 31(9): 1277 - 1284.
- [2] VIJAYKRISHMAN N, KANDEMIR M, IRWIN M J, *et al.* Energy-driven integrated hardware-software optimizations using simple-power [C]// ISCA-27: Proceedings of the 27th Annual International Symposium on Computer Architecture. New York: ACM Press, 2000: 95 - 10.
- [3] CALDER B, GRUNWALD D, EMER J. Predictive sequential associative cache [C]// Proceedings of the 2nd International Symposium on High-Performance Computer Architecture. New York: ACM Press, 1996: 244 - 253.
- [4] CHEN H C, CHIANG J S. Low-power way-predicting cache using valid-bit predecision for parallel architectures [C]// Proceedings of the 19th International Conference on Advanced Information Networking and Applications. Washington, DC: IEEE Computer Society, 2005: 203 - 206.
- [5] ZHANG C, ZHANG X, YAN Y. Two fast and high-associativity cache schemes [J]. IEEE Micro, 1997, 17(5): 40 - 49.
- [6] SUH G E, RUDOLPH L, DEBADAS S. Dynamic partitioning of shared cache memory [J]. Journal of Supercomputing, 2004, 28(1): 7 - 26.
- [7] PWELL M D, AGARWAL A, VIJAYKUMAR T N, *et al.* Reducing set-associative cache energy via way-prediction and selective direct-mapping [C]// Proceedings of the 34th Proceedings ACM/IEEE International Symposium on Micro-Architecture. Washington, DC: IEEE Computer Society, 2001: 54 - 65.
- [8] CHAUDHURI M. Page NUCA: selected policies for page grain locality management in large shared chip multiprocessor caches [C]// Proceedings of HPCA-15. Washington, DC: IEEE Computer Society, 2009: 227 - 238.
- [9] AWASTHI M, SUDAN K, BALASUBRAMONIAN R, *et al.* Dynamic hardware - assisted software controlled page placement to manage capacity allocation and sharing within large caches [C]// Proceedings of HPCA-15. Washington, DC: IEEE Computer Society, 2009: 250 - 261.
- [10] CHANG J. Cooperative caching for chip multi-processors [D]. Madison, Wisconsin, USA: University of Wisconsin at Madison, 2007.
- [11] LIU C, SIYASUBRAMANIAM A, KANDEMIR M. Organizing the last line of defense before hitting the memory wall for CMPs [C]// Proceedings of the 10th International Symposium on High Performance Computer Architecture. Washington, DC: IEEE Computer Society, 2004: 176 - 185.
- [12] LIN J, LU Q, DING X, *et al.* Gaining insights into multi-core cache partitioning: bridging the gap between simulation and real systems [C]// HPCA 2008: Proceedings of the 14th IEEE International Symposium on High Performance Computer Architecture. Piscataway, NJ: IEEE Press, 2008: 367 - 378.
- [13] SANCHEZ D, KOZYRAKIS C. Vantage: scalable and efficient fine-grain cache partitioning [J]. Computer Architecture News, 2011, 39(3): 57 - 68.
- [14] HASENPLAUGH W, AHUJA P S, JALEEL A, *et al.* The gradient-based cache partitioning algorithm [J]. ACM Transactions on Architecture and Code Optimization, 2012, 8(4): 1 - 20.
- [15] MATICK R E, HELLER T J, IGNATOWSKI M. Analytical analysis of finite cache penalty and cycles per instruction of a multiprocessor memory hierarchy using miss rates and queuing theory [J]. IBM Journal of Research and Development, 2001, 45(6): 819 - 843.
- [16] 所光, 杨学军. 面向多线程多道程序加权共享 Cache 划分 [J]. 计算机学报, 2008, 33(11): 1938 - 1946.
- [17] QURESHI M K, PATTT Y N. Utility based cache partitioning: a low overhead, high performance, runtime mechanism to partition shared caches [C]// MICRO 39: Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture. Washington, DC: IEEE Computer Society, 2006: 423 - 432.
- [18] 隋秀峰, 吴俊敏, 陈国良. ARP: 同时多线程处理器中共享 Cache 自适应运行时划分机制 [J]. 计算机研究与发展, 2008, 45(7): 1269 - 1277.
- [19] IYER R. CQoS: a framework for enabling QoS in shared caches of CMP platforms [C]// ICS 2004: Proceedings of the 18th Annual International Conference on Supercomputing. New York: ACM Press, 2004: 257 - 266.
- [20] PAUL M, PETROV P. Dynamically adaptive I-cache partitioning for energy-efficient embedded multitasking [J]. IEEE Transactions on Very Large Scale Integration Systems, 2011, 19(11): 2067 - 2080.
- [21] INOUE K, ISHIHARA T, MURAKAMI K. Way-predicting set-associative cache for high performance and low energy consumption [C]// ISLPED 1999: Proceedings of the 1999 International Symposium on Low Power Electronics and Design. New York: ACM Press, 1999: 273 - 275.
- [22] MAGNUSSON P S, CHRISTENSSON M, ESKILSON J, *et al.* Simics: a full system simulation platform [J]. Computer, 2002, 35(2): 50 - 58.

(上接第 2403 页)

- [4] WANG S, YANG Y. Fault tolerance in bubble-sort graph networks [J]. Theoretical Computer Science, 2012, 421: 62 - 69.
- [5] ZHANG Z, XIONG W, YANG W. A kind of conditional fault tolerance of alternating group graphs [J]. Information Processing Letters, 2010, 110(22): 998 - 1002.
- [6] YANG Y, WANG S. Conditional connectivity of star graph networks under embedding restriction [J]. Information Sciences, 2012, 199: 187 - 192.
- [7] BOSE B, BROEG B, KWON Y, *et al.* Lee distance and topological properties of  $k$ -ary  $n$ -cubes [J]. IEEE Transactions on Computers, 1995, 44(8): 1021 - 1030.
- [8] GHOZATI S A, WASSERMAN H C. The  $k$ -ary  $n$ -cube network: modeling, topological properties and routing strategies [J]. Computers and Electrical Engineering, 1999, 25(3): 155 - 168.
- [9] STEWART I A, XIANG Y. Bipanconnectivity and bipancyclicity in  $k$ -ary  $n$ -cubes [J]. IEEE Transactions on Parallel and Distributed Systems, 2009, 20(1): 25 - 33.
- [10] ADIGA N R, BLUMRICH M A, CHEN D, *et al.* Blue Gene/L torus interconnection network [J]. IBM Journal of Research and Development, 2005, 49(2): 265 - 276.
- [11] WANG S, ZHANG G, FENG K. Fault tolerance in  $k$ -ary  $n$ -cube networks [J]. Theoretical Computer Science, 2012, 460: 34 - 41.
- [12] BONDY J A, MURTY U S R. Graph theory [M]. Berlin: Springer, 2008.