

DOI:10.13718/j.cnki.xsxb.2018.09.009

交错立方体上限制容错单播算法的研究^①王 喜^{1,2}, 何福男¹, 张书奎²

1. 苏州工业职业技术学院 软件与服务外包学院, 江苏 苏州 215004;

2. 苏州大学 计算机科学与技术学院, 江苏 苏州 215006

摘要: 在交错立方体中引入限制故障顶点集的概念, 证明了当 $n \geq 3$ 时, 交错立方体中基于限制故障顶点集的限制连通度为 $2n - 2$, 这一结果几乎是交错立方体上传统连通度的两倍; 然后提出了基于该情形下的时间复杂度为 $O(\lceil \log |F| \rceil n^3)$ 的容错单播算法, 并证明了在最坏情形下, 该算法构造出的无故障路径的最长路径长度的上界为 $5m + n - 1$, 其中 $m = \lceil \log |F| \rceil$; 进一步利用上述算法进行仿真.

关键词: 交错立方体; 并行系统; 连通度; 限制连通度

中图分类号: TP393

文献标志码: A

文章编号: 1000-5471(2018)09-0051-09

多处理器互连网络, 简称互连网络, 是指由多个处理器按照一定规则相互连接而构成的网络, 目前越来越广泛地出现在计算机技术的相关研究与应用领域中, 例如并行系统^[1].

近年来, 随着互连网络的规模不断扩大, 互连网络的容错性研究已经成为一个重要的研究课题. 连通度是度量互连网络容错性的一个重要参数, 连通度越大, 则互连网络的容错性能就相对越高. 然而, 互连网络的连通度的定义中有一个明显的假设即所有的处理器发生故障的可能性是一样的, 但实际上往往并非如此, 比如一个处理器的邻居全部发生故障的可能性是很低的. 因此, 使用连通度来评估互连网络的容错性会低估互连网络的容错性能. 为了克服这个缺陷, 文献[2]引入了条件连通度的概念. 进一步, 文献[3]对发生故障的顶点集做出限制, 提出限制连通集概念. 文献[4]证明了条件连通问题是 NP-完全的. 因此, 存在大量特定网络拓扑上的限制连通性问题的研究成果^[3-8].

交错立方体(Cross-cube)^[9]作为超立方体网络的一类重要变型, 与超立方体相比具有低直径、哈密顿连通性等优越性质. 研究交错立方体的限制连通性, 能够更加精确地度量该网络的容错性. 本文将讨论交错立方体在限制连通顶点集情形下的限制连通度, 并设计和分析了一种容错单播算法.

1 预备知识

一个互连网络的拓扑结构可用一个图来表示, 其中处理器与处理器之间的链路分别用顶点集和边集表示. 令 $G = (V(G), E(G))$ 表示一个图, 其中 $V(G)$ 表示顶点集, $E(G)$ 表示边集. 给定图 G 中任意两个顶点 u 和 v , 若 $(u, v) \in E(G)$, 则 u 和 v 是邻居. 顶点 u 在图 G 的邻居集合可表示为 $N(G, u) = \{v \mid (u, v) \in E(G)\}$.

① 收稿日期: 2017-09-30

基金项目: 国家自然科学基金项目(61702351); 中国博士后科学基金资助项目(17KJB520036); 江苏省高校自然科学基金项目(17KJB520036); 江苏省博士后科研资助项目(1701172B); 江苏省“六大人才高峰”项目(2014-WLW-010); 苏州市融合通信重点实验室项目(SKLC2013XX); 苏州工业职业技术学院院级课题(SGYKJ201706).

作者简介: 王 喜(1988-), 男, 博士, 讲师, 主要从事互连网络、并行与分布式系统研究.

$E(G)$ }. 图 G 的连通度可表示为 $\kappa(G)$.

图 G 中顶点 u 与 v 之间的路径 P 可定义为一个顶点序列 $P = (u_0 = u, u_1, \dots, u_k = v)$, P 中除了起始顶点 u 和结束顶点 v 以外的任意两个顶点都是不同的. P 上从 u_i 到 u_j 的子路径可表示为 $\text{Path}(P, u_i, u_j)$, 其中 $0 \leq i < j \leq k$. 路径 P 的反转 $(u_k = v, \dots, u_1, u_0 = u)$ 可表示为 P^{-1} . P 中所有顶点(边)的集合可表示为 $V(P)(E(P))$. 如果 $V' \subseteq V(G)$, 可用 $G[V']$ 表示图 G 的顶点导出子图. 进一步, 使用 $G - V$ 来表示 $G[V(G) - V]$. 对于任意的顶点 $v \in V(G)$, 如果 $(u, v) \in E(G)$, 则 u 是 v 的一个邻居或 u 与 v 相邻. 定义 $N(G, V') = \{x \in V(G) \mid \text{存在顶点 } y \in V' \text{ 满足 } (x, y) \in E(G)\}$. 顶点 v 的邻居集合 $\Gamma(G, v) = \{u \in V(G) \mid (v, u) \in E(G)\}$ 表示 v 的所有邻居. 进一步, 对于一个顶点集合 $V' \subseteq V(G)$, 定义 V' 的邻居集合为

$$\Gamma(G, V') = \bigcup_{x \in V'} \Gamma(G, x) / V'$$

明显地,

$$\Gamma(G, V') = N(G, V') / V'$$

定义 1 给定图 G , 令 $F \subset V(G)$ 是 G 中一个故障顶点集合, 如果 $u \in F$, u 是 G 中一个故障顶点; 否则, u 是 G 中一个无故障顶点. 若 G 中每个顶点至少有一个无故障邻居, 则基于该条件的 G 的限制连通度可表示为 $\kappa'(G)$. 另外, 基于任意故障顶点的 G 的连通度可表示为 $\kappa(G)$.

一个 n 维交错立方体是一个具有 2^n 个顶点和 $(n+1)2^{n-1}$ 条边的 $(n+1)$ -正则图^[9-10], 记为 C_n , 其中 $n \geq 2$. 它的每个顶点用 0 到 $2^n - 1$ 之间的一个长度为 n 的二进制串 $x_{n-1}x_{n-2}\dots x_0$ 表示, 其中 x_{n-1} 为最高有效位, x_0 为最低有效位, 二进制串 x_i 的补 $\bar{x}_i = 1 - x_i$. 所有顶点的二进制串互不相同. 如果两个顶点的二进制串序列只有一位不同, 则构成一条边, 且二进制串的最后两位看作一位. 令 $u = x_{n-1}x_{n-2}\dots x_0$ 表示 C_n 中任意顶点, 用 $u[i:j] = x_i x_{i-1} \dots x_j$ 表示 $x_{n-1}x_{n-2}\dots x_0$ 的子二进制串, 其中 $n-1 \geq i \geq j \geq 0$.

C_2 是一个包含 4 个顶点的完全图, 其中每个顶点的度数为 3, 称其为一个基本组成块. 当 $n \geq 3$ 时, 交错立方体的递归定义如定义 2 所示^[9-10].

定义 2 一个 n 维的交错立方体 C_n , 由两个 $n-1$ 维的交错立方体相互连接而成, 分别记为 C_{n-1}^0 和 C_{n-1}^1 , 其中 C_{n-1}^0 和 C_{n-1}^1 中各顶点的最高有效位分别是 0 和 1. 设 $u = u_{n-1}u_{n-2}\dots u_0 \in V(C_{n-1}^0)$, $v = v_{n-1}v_{n-2}\dots v_0 \in V(C_{n-1}^1)$, 则 $(u, v) \in E(C_n)$ 当且仅当二进制串的最高有效位不同.

一个 4 维的交错立方体如图 1 所示.

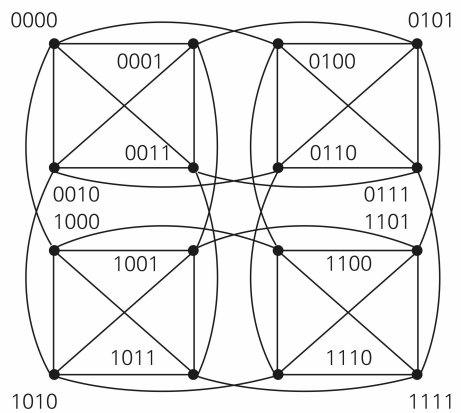


图 1 4 维交错立方体 C_4

2 交错立方体的限制连通度

本节将证明 n 维交错立方体的限制连通度为 $2n - 2$, 这一结果几乎是传统连通度的两倍. 因此, 研究交错立方体的限制连通度的精确值, 能够更加精确地度量该网络的容错性.

定理 1^[9] 若 $n \geq 3$, 则 $\kappa(C_n) = n + 1$.

引理 1 当 $n \geq 3$ 时, 对于任意的故障顶点集合 $F \subset V(C_n)$ 且 $|F| = n + 1$, 若 $C_n - F$ 是不连通的, 则存在一个顶点 $u \in V(C_n)$ 满足 $F = \Gamma(C_n, u)$ 且 $C_n - F$ 有且仅有两个连通分支, 其中一个连通分支是 $C_n[\{u\}]$, 另一个连通分支是 $C_n - (F \cup \{u\})$.

证 该引理通过对整数 n 进行归纳来证明. 容易验证 $n = 3$ 时该引理成立. 假设该引理在 $n = \tau - 1$ ($\tau \geq 4$) 时成立, 并假设对于任意的 $F \subset V(C_\tau)$ 且 $|F| = \tau + 1$, 若 $C_\tau - F$ 是不连通的, 则可令 $F_i = F \cap V(C_{\tau-1}^i)$, 且 $i \in \{0, 1\}$. 不失一般性, 假设 $|F_0| \leq |F_1|$, 此时分以下 3 种情形来讨论 $|F_0|$ 和 $|F_1|$.

情形 1 $|F_0| = 0$. 此时 $F_1 = F$. 由定义 2, $C_{\tau-1}^1 - F_1$ 中每个顶点与 $C_{\tau-1}^0 - F_0 = C_{\tau-1}^0$ 中顶点相邻. 故 $C_{\tau} - F$ 是连通的, 这与假设矛盾.

情形 2 $|F_0| \geq 2$. 此时 $|F_0| \leq |F_1| \leq \tau - 1$. 由定理 1, $C_{\tau-1}^1 - F_1$ 和 $C_{\tau-1}^0 - F_0$ 都是连通的. 当 $\tau \geq 4$ 时, 存在 $|V(C_{\tau-1}^0 - F_0)| - |F_1| = 2^{\tau-1} - |F| = 2^{\tau-1} - (\tau + 1) > 0$. 那么, $|V(C_{\tau-1}^0 - F_0)| > |F_1|$ 成立, 这意味着 $C_{\tau-1}^0 - F_0$ 中至少存在一个顶点与 $C_{\tau-1}^1 - F_1$ 中顶点相邻. 故 $C_{\tau} - F$ 是连通的, 这与假设矛盾.

情形 3 $|F_0| = 1$ 且 $|F_1| = \tau$. 此时 $|F_0| = 1 \leq \tau$. 根据定理 1, $C_{\tau-1}^0 - F_0$ 是连通的. 假设 $C_{\tau-1}^1 - F_1$ 是连通的. 当 $\tau \geq 4$ 时, $|V(C_{\tau-1}^0 - F_0)| - |F_1| = 2^{\tau-1} - |F| = 2^{\tau-1} - (\tau + 1) > 0$. 那么, 存在 $|V(C_{\tau-1}^0 - F_0)| > |F_1|$, 这意味着 $C_{\tau-1}^0 - F_0$ 中至少存在一个顶点与 $C_{\tau-1}^1 - F_1$ 中顶点相邻, 此时 $C_{\tau} - F$ 是连通的, 这与假设矛盾.

因此, $C_{\tau-1}^1 - F_1$ 是不连通的. 根据归纳假设, 存在一个顶点 $u \in V(C_{\tau-1}^1)$ 满足 $F = \Gamma(C_{\tau-1}^1, u)$ 且 $C_{\tau-1}^1 - F$ 有且仅有两个连通分支, 其中一个连通分支是 $C_{\tau-1}^1[\{u\}]$, 另一个连通分支是 $C_{\tau-1}^1 - (F \cup \{u\})$. 对于 $\tau \geq 4$, 有 $|V(C_{\tau-1}^0 - F_0)| - |F_1 \cup \{u\}| = 2^{\tau-1} - (\tau + 2) > 0$. 这意味着 $C_{\tau-1}^0 - F_0$ 中至少存在一个顶点与 $C_{\tau-1}^1 - (F \cup \{u\})$ 中顶点相邻.

若 u 与 $C_{\tau-1}^0 - F_0$ 中顶点相连, 则 $C_{\tau} - F$ 是连通的, 这与假设矛盾; 若 u 与 F_0 中唯一的顶点相连, 则 $C_{\tau} - F$ 是连通的, 这与假设矛盾. 若 $C_{\tau} - F$ 是不连通的, 则存在一个顶点 $u \in V(C_{\tau})$ 满足 $F = \Gamma(C_{\tau}, u)$ 且 $C_{\tau} - F$ 有且仅有两个连通分支, 其中一个连通分支是 $C_{\tau}[\{u\}]$, 另一个连通分支是 $C_{\tau} - (F \cup \{u\})$. 证毕.

根据定理 1 很容易推导出, 相比顶点个数 2^n , 规模为 n 的点割集合总共有 $\binom{2^n}{n}$ 种情形. 如果是 n 无穷大的, 这意味着 C_n 中顶点的所有邻居都为故障顶点的概率是非常小的.

引理 2 当 $n \geq 3$ 时, 对于任意的顶点集合 $F \subset V(C_n)$ 且 $|F| \leq 2n - 3$. 令 $F_0 = F \cap V(C_{n-1}^0)$ 且 $F_1 = F \cap V(C_{n-1}^1)$, 如果 $C_{n-1}^0 - F_0$ 和 $C_{n-1}^1 - F_1$ 都是连通的. 那么 $C_n - F$ 是连通的.

证 当 $n \geq 3$ 时, 有 $|F_0 \cup F_1| = |F| \leq 2n - 3 < 2^{n-1} = |V(C_{n-1}^0)| = |V(C_{n-1}^1)|$ 和 $|V(C_{n-1}^0) - F_0| > |F_1|$. 因此, $C_{n-1}^0 - F_0$ 中存在顶点与 $C_{n-1}^1 - F_1$ 中顶点相邻, 故 $C_n - F$ 是连通的. 证毕.

由定义 2, 可以推导出以下引理:

引理 3 对于任意的整数 $n \geq 3$, C_n 中任意一个顶点都位于一个顶点个数为 4 的完全图上.

由引理 3 和定义 2, 可以推导出以下引理:

引理 4 当 $n \geq 3$ 时, 对于任意的边 $(u, v) \in E(C_n)$. 如果 $u[n-1:2] = v[n-1:2]$, $|(\Gamma(C_n, u) - \{v\}) \cap (\Gamma(C_n, v) - \{u\})| = 2$; 否则, $|(\Gamma(C_n, u) - \{v\}) \cap (\Gamma(C_n, v) - \{u\})| = 0$.

引理 5 当 $n \geq 3$ 时, 对于任意的顶点集合 $F \subset V(C_n)$ 且 $|F| \leq 2n - 3$. 如果 C_n 中每个顶点都至少有一个无故障邻居, 则 $C_n - F$ 是连通的.

证 该引理的证明仅需验证下述断言成立即可: 对于任意两个不同顶点 $x, y \in V(C_n - F)$, $C_n - F$ 中 x 和 y 之间存在一条无故障路径.

令 $F_i = F \cap V(C_{n-1}^i)$, 且 $i \in \{0, 1\}$. 不失一般性, 假设 $|F_0| \leq |F_1|$, 则 $|F_0| \leq \lfloor \frac{|F|}{2} \rfloor \leq \lfloor \frac{2n-3}{2} \rfloor \leq n-2$. 由定理 1, $C_{n-1}^0 - F_0$ 是连通的. 此时, 可分以下 3 种情形来讨论.

情形 1 $x, y \in V(C_{n-1}^0)$, 由于 $C_{n-1}^0 - F_0$ 是连通的且 $C_{n-1}^0 - F_0 \subset C_n - F$, 则 $C_n - F$ 中 x 和 y 之间存在一条无故障路径.

情形 2 $x \in V(C_{n-1}^0)$ 且 $y \in V(C_{n-1}^1)$, 令 u 表示 y 在 C_{n-1}^0 中的邻居. 考虑到 y 在 $C_n - F$ 中至少有一个无故障邻居, 此时有 1), 2) 两个子情形.

1) $u \notin F_0$. 若 $u = x$, 则 (x, y) 是 $C_n - F$ 中 x 和 y 之间的一条无故障路径; 否则, 由于 $|F_0| \leq n-2$, 根据定理 1, $C_{n-1}^0 - F_0$ 是连通的, 那么 x 和 u 之间在 $C_{n-1}^0 - F_0$ 中存在一条无故障路径 Q , 则 (Q, y) 是 $C_n -$

F 中 x 和 y 之间的一条无故障路径.

2) $u \in F_0$. 由于 y 在 $C_n - F$ 中至少有一个无故障邻居, 那么 $C_{n-1}^1 - F_1$ 中存在顶点 v , 使得 $(y, v) \in E(C_n)$. 令 $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$ 分别表示 y 在 $C_{n-1}^1 - F_1$ 中除了 v 的邻居, 且令 $\beta_1, \beta_2, \dots, \beta_{n-1}$ 分别表示 v 在 $C_{n-1}^1 - F_1$ 中除了 y 的邻居. 此时, 有 ①, ② 两个子情形.

① $u[n-1; 2] = v[n-1; 2]$. 由引理 4, 可得 $|\{\alpha_1, \alpha_2, \dots, \alpha_{n-1}\} \cap \{\beta_1, \beta_2, \dots, \beta_{n-1}\}| = 2$. 不失一般性, 假设 $\{\alpha_1, \alpha_2, \dots, \alpha_{n-1}\} \cap \{\beta_1, \beta_2, \dots, \beta_{n-1}\} = \{\alpha_{n-2}, \alpha_{n-1}\} = \{\beta_{n-2}, \beta_{n-1}\}$. 进一步, 令 $\gamma_1, \gamma_2, \dots, \gamma_{n-1}$ 分别表示 $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$ 在 $C_{n-1}^0 - F_0$ 中的邻居, 且 $\omega, \delta_1, \delta_2, \dots, \delta_{n-3}$ 分别表示 $v, \beta_1, \beta_2, \dots, \beta_{n-3}$ 在 $C_{n-1}^0 - F_0$ 中的邻居. 由定义 2, $\{\gamma_1, \gamma_2, \dots, \gamma_{n-1}\} \cap \{\omega, \delta_1, \delta_2, \dots, \delta_{n-3}\} = \emptyset$. 因此, 存在 $2n-3$ 条不相交路径 $P_1 = (y, \alpha_1, \gamma_1)$, $P_2 = (y, \alpha_2, \gamma_2)$, \dots , $P_{n-1} = (y, \alpha_{n-1}, \gamma_{n-1})$, $P_n = (y, v, \omega)$, $P_{n+1} = (y, v, \beta_1, \gamma_1)$, $P_{n+2} = (y, v, \beta_2, \gamma_2)$, \dots , $P_{2n-3} = (y, v, \beta_{2n-3}, \delta_{2n-3})$.

由于 $|F| - |\{u\}| \leq 2n-4 < 2n-3$. 因此, 至少存在一条无故障路径 P_j 在这 $2n-3$ 条路径 $P_1, P_2, \dots, P_{2n-3}$ 中, 其中 $j \in \{1, 2, \dots, 2n-3\}$. 接下来, 若 $1 \leq j \leq n-1$, 有 $P_j = (y, \alpha_j, \gamma_j)$; 若 $j = n$, 有 $P_j = (y, v, \omega)$; 若 $n+1 \leq j \leq 2n-3$, 有 $P_j = (y, v, \beta_j, \delta_j)$. 因此, 存在一条路径 P 连接 y 和 C_{n-1}^0 中顶点 z .

若 $z = x$, 则 P^{-1} 是 $C_n - F$ 中 x 和 y 之间的一条无故障路径; 否则, 由于 $C_{n-1}^0 - F_0$ 是连通的, x 和 z 之间在 $C_{n-1}^0 - F_0$ 中存在一条无故障路径 Q , 那么 (Q, P^{-1}) 是 $C_n - F$ 中 x 和 y 间的一条无故障路径.

② $u[n-1; 2] \neq v[n-1; 2]$. 由引理 4, 可得 $|\{\alpha_1, \alpha_2, \dots, \alpha_{n-1}\} \cap \{\beta_1, \beta_2, \dots, \beta_{n-1}\}| = 0$. 类似于情形 2.2.1 中讨论, 可构造出 $C_n - F$ 中从 y 到 C_{n-1}^0 的 $2n+1$ 条不相交路径. 由于 $|F| - |\{u\}| \leq 2n-4 < 2n+1$, 因此, 存在一条路径 P 连接 y 和 C_{n-1}^0 中顶点 z . 若 $z \neq x$, 由于 $C_{n-1}^0 - F_0$ 是连通的, x 和 z 之间在 $C_{n-1}^0 - F_0$ 中存在一条无故障路径 Q , 那么 (Q, P^{-1}) 是 $C_n - F$ 中 x 和 y 间的一条无故障路径; 否则, P^{-1} 是 $C_n - F$ 中 x 和 y 之间的一条无故障路径.

情形 3 $x, y \in V(C_{n-1}^1)$, 若 $(x, y) \in E(C_n)$, 则 (x, y) 是 $C_n - F$ 中 x 到 y 之间的一条无故障路径; 否则, 类似于情形 2.2, 可以构造出 x 到 $C_{n-1}^0 - F_0$ 中顶点 u 之间的一条无故障路径 P_0 , 以及 y 到 $C_{n-1}^0 - F_0$ 中顶点 $v \in V(C_{n-1}^0 - F_0)$ 之间的一条无故障路径 P_1 .

若 $V(P_0) \cap V(P_1) = \emptyset$, 令 z 表示 P_0 和 P_1 间的第一个公共点, 路径 $(Path(P_0, x, z), Path(P_1^{-1}, z, y))$ 是 $C_n - F$ 中 x 到 y 之间的一条无故障路径; 否则, 由于 $C_{n-1}^0 - F_0$ 是连通的, $C_{n-1}^0 - F_0$ 中 u 和 v 之间存在一条无故障路径 P_2 . 那么, (P_0, P_2, P_1^{-1}) 是 $C_n - F$ 中 x 和 y 之间的一条无故障路径.

综上所述, 对于任意两个不同顶点 $x, y \in V(C_n - F)$, $C_n - F$ 中 x 和 y 之间存在一条无故障路径, 故 $C_n - F$ 是连通的. 证毕.

定理 2 当 $n \geq 3$ 时, 若 C_n 中每个顶点都至少有一个无故障邻居, 则 $\kappa'(C_n) = 2n - 2$.

证 通过引理 2, $\kappa'(C_n) \geq 2n - 2$. 接下来, 将证明 $\kappa'(C_n) \leq 2n - 2$ 成立.

对于任意的整数 $n \geq 3$, 如果 C_n 中每个顶点都至少有一个无故障邻居, 则存在一条边 $(u, v) \in E(C_n)$, 使 $\Gamma(C_n, \{u, v\}) = 2n - 2$ 且 $C_n - \Gamma(C_n, \{u, v\})$ 有且仅有两个连通分支: 其中一个连通分支为 $C_n[\{u, v\}]$, 另外一个连通分支为 $C_n - N(C_n, \{u, v\})$.

首先证明当 $n = 3$ 时该结论成立. 当 $n = 3$ 时, 选择一条边 $(000, 001)$, 令故障集合为 $\Gamma(C_3, \{000, 001\}) = \{010, 011, 100, 101\}$. 明显的 $|\Gamma(C_3, \{00, 01\})| = 4 = 2 \times 3 - 2$. 容易验证 $C_3 - \Gamma(C_3, \{00, 01\})$ 有且仅有两个连通分支: 其中一个连通分支为 $C_3[\{00, 01\}]$, 另外一个连通分支为 $C_3 - N(C_3, \{00, 01\})$.

接下来, 假设该结论在 $n = \tau - 1 (\tau \geq 4)$ 时成立. 当 $n = \tau$ 时, 通过归纳假设, 存在一条边 $(u, v) \in E(C_{\tau-1}^0)$ 使得 $\Gamma(C_{\tau-1}^0, \{u, v\}) = 2\tau - 4$ 且 $C_{\tau-1}^0 - \Gamma(C_{\tau-1}^0, \{u, v\})$ 有且仅有两个连通分支: 其中一个连通分支为 $C_{\tau-1}^0[\{u, v\}]$, 另外一个连通分支为 $C_{\tau-1}^0 - N(C_{\tau-1}^0, \{u, v\})$. 令 x 和 y 分别表示 u 和 v 不在 $C_{\tau-1}^0$ 中的

邻居. 接下来, 有 $\Gamma(C_\tau, \{u, v\}) = \Gamma(C_{\tau-1}^0, \{u, v\}) \cup \{x, y\}$ 和 $|\Gamma(C_\tau, \{u, v\})| = 2\tau - 2$. 由定理 1, 可以验证 $C_{\tau-1}^1 - \{x, y\}$ 是连通的. 对于任意的整数 $\tau \geq 4$, 有 $|C_{\tau-1}^1 - \{x, y\}| = 2^{\tau-1} - (2\tau - 2) > 1$. 因此, $C_{\tau-1}^1 - \{x, y\}$ 中存在顶点连接 $C_{\tau-1}^0 - N(C_{\tau-1}^0, \{u, v\})$ 中一个顶点. 故 $C_\tau - \Gamma(C_\tau, \{u, v\})$ 有且仅有两个连通分支: 其中一个连通分支为 $C_\tau[\{u, v\}]$, 另外一个连通分支为 $C_\tau - N(C_\tau, \{u, v\})$. 进一步, 令故障集合为 $\Gamma(C_\tau, \{u, v\})$, 容易验证 C_τ 中每个顶点都至少有一个无故障邻居, 通过上述证明可以验证 $\kappa'(C_\tau) \leq 2\tau - 2$.

综上所述, $\kappa'(C_n) = 2n - 2$. 证毕.

3 交错立方体基于限制故障集的单播算法

定理 2 证明了 n 维交错立方体上限制连通度的精确值. 本节提出 n 维交错立方体上基于限制故障顶点集合的单播算法 CFRouting, 如下所示.

Algorithm: CFRouting

Input: An n -dimensional cross-cube, C_n , a faulty node set $F \subset V(C_n)$ with $|F| \leq 2n - 3$, and two nodes $u, v \in V(C_n)$.

Output: A path from u to v in $C_n - F$.

print(CFRouting(C_n, F, u, v));

function CFRouting(C_n, F, u, v)

1. if $(u, v) \in E(C_n)$ then return (u, v) ;
 2. else if $n \leq 3$ or $|F| \geq 2n - 2$ then
 3. return (BFS($C_n - F, u, v$));
 4. else if $F = \emptyset$ then return (CFRouting(C_n, u, v));
 5. end if
 6. Let $F_0 \leftarrow F \cap V(C_{n-1}^0)$, $F_1 \leftarrow F \cap V(C_{n-1}^1)$;
 7. Let $m \leftarrow \min\{|F_0|, |F_1|\}$;
 8. for $i \in \{0, 1\}$ then
 9. Let $C_0 \leftarrow C_{n-1}^i$, $C_1 \leftarrow C_{n-1}^{1-i}$;
 10. if $u, v \in V(C_0)$ and $|F_i| = m$ then
 11. return (CFRouting(C_0, F_i, u, v));
 12. else if $u, v \in V(C_0)$ and $|F_{1-i}| = m$ then
 13. Let $P \leftarrow \text{CMapping}(u, C_0, C_1, F)$;
 14. Let $Q \leftarrow \text{CMapping}(v, C_0, C_1, F)$;
 15. if $V(P) \cap V(Q) = \emptyset$ then
 16. $S \leftarrow \text{CFRouting}(C_1, F, P[-1], Q[-1])$;
 17. return(P, S, Q^{-1});
 18. end if
 19. Find the first common node x from P and Q ;
 20. return(Path(P, u, x), Path(Q^{-1}, x, v));
 21. else if $u \in V(C_0)$ and $v \in V(C_1)$ and $|F_i| = m$
 22. Let $Q \leftarrow \text{CMapping}(v, C_1, C_0, F)$;
 23. if $u \in V(Q)$ then return(Path(Q, u, v));
 24. return(CFRouting($C_0, F, u, Q[-1]$), Q);
 25. else if $u \in V(C_0)$ and $v \in V(C_1)$ and $|F_{1-i}| = m$ then
 26. Let $P \leftarrow \text{CMapping}(u, C_0, C_1, F)$;
 27. if $v \in V(P)$ then return(Path(P, u, v));
 28. return($P, \text{CFRouting}(C_1, F, P[-1], v)$);
 29. end if
 30. end for
- end function
-

Algorithm: CFRouting

function CMapping(u, G_0, G_1, F)

1. Let $\{v\} \leftarrow N(G_1, u)$;
2. if $v \notin F$ then return(u, v);
3. for v in $N(G_0 - F, u)$ then
4. for x in $N(G_1 - F, v)$ then
5. return(u, v, x);
6. end for
7. end for
8. for v in $N(G_0 - F, u)$ then
9. for x in $N(G_0 - F, v)$ then
10. for y in $N(G_1 - F, x)$ then
11. return(u, v, x, y);
12. end for
13. end for
14. end for

end function

function CRouting(C_n, u, v)

1. Let $i \leftarrow n-1, P \leftarrow (u)$;
2. while $i > 1$ then
3. Let x be the last node of P ;
4. if $x[i] \neq v[i]$ then
5. $P \leftarrow (P, x[n-1:i] + v[i:i] + x[i-1:0])$;
6. end if
7. $i \leftarrow i-1$;
8. end while
9. if $u[1:0] = v[1:0]$ then return(P);
10. return(P, v);

end function

接下来, 定理 3 给出算法 CFRouting 的设计思路和执行过程, 并分析其时间复杂度.

定理 3 当 $n \geq 3$ 时, 对于任意的顶点集合 $F \subset V(C_n)$ 使 $|F| \leq 2n-3$. 算法 CFRouting 可以构造出 $C_n - F$ 中任意两个不同顶点间的一条无故障路径, 在最坏情形下时间复杂度为 $O(\lceil \log |F| \rceil n^3)$.

证 当 $n \geq 3$ 时, 对于任意的顶点 u 和 v , 以及集合 $F \subset V(C_n)$ 使 $|F| \leq 2n-3$. 算法 CFRouting 可以构造出 $C_n - F$ 中任意两个不同顶点间的一条无故障路径. 为了便于描述该算法, 引入两个函数, CMapping 和 CRouting. 其中 CMapping 构造从 G_0 中顶点 u 到 $G_1 - F$ 的一条无故障路径; CRouting 构造出 C_n 中 u 到 v 的一条路径.

假设函数 CFRouting 中构造的路径被保存在一个向量中, 其中路径中顶点用一个长度为 n 的二进制串表示. 函数 CFRouting 中第 1 行在静态时间内构造出一条边; 函数 CFRouting 中第 2-3 行, 采用广度优先搜索算法(BFS) 构造一条无故障路径, 其最坏情形下, 时间复杂度为 $O(2^n + n2^n) = O(N \log N)$, 其中 $N = 2^n$; 函数 CFRouting 中第 4 行调用函数 CRouting 构造一条无故障路径, 时间复杂度为 $O(n)$; 函数 CFRouting 中 6-7 行, 花费 $O(n)$ 时间去计算 F_0, F_1 和 m 的值.

用 $U(u, v, n, F)$ 表示构造 $C_n - F$ 中 u 和 v 之间一条无故障路径的时间. 假设 n 是无穷大的, 令 $T(n) = \max\{U(u, v, n, F) \mid u, v \in V(C_n - F), u \neq v\}$. 当 $F = \emptyset$ 时, $T(n) = O(n)$. 接下来, 根据 $n \geq 3$ 和 $|F| \leq 2n-3$ 分以下几种情形:

根据函数 CFRouting 中第 11 行, 有

$$T(n) \leq T(n-1) + O(n) \quad (1)$$

根据函数 CFRouting 中第 13 – 20 行, 有

$$T(n) \leq T(n-1) + O(n^3) \quad (2)$$

根据函数 CFRouting 中第 21 – 24, 25 – 29 行, 有

$$T(n) \leq T(n-1) + O(n^3) \quad (3)$$

因此, 基于公式(1) – (3), 有

$$T(n) \leq \max \left\{ \sum_{i=1}^{\lceil \log |F| \rceil} O((n-i+1)^3) + O(n - \lceil \log |F| \rceil), O(n^3), O(n) \right\} \leq \max \{ O(\lceil \log |F| \rceil n^3), O(n^3), O(n) \} \leq O(\lceil \log |F| \rceil n^3) \quad (4)$$

由公式(4), 在最坏情形下, 算法 CFRouting 的时间复杂度为 $O(\lceil \log |F| \rceil n^3)$, 其中 $|F| \leq 2n - 3$.

证毕.

进一步, 定理 4 分析算法 CFRouting 构造出的路径长度的最大值.

定理 4 当 $n \geq 3$ 时, 给定故障顶点集合 $F \subset V(C_n)$ 且 $|F| \leq 2n - 3$. 算法 CFRouting 在最坏情形下, 构造出的无故障路径的最长路径长度的上界为 $5m + n - 1$, 其中 $m = \lceil \log |F| \rceil$.

证 用 $M(n)$ 表示算法 CFRouting 构造得到的路径的长度. 容易验证, 当 $|F| = 0$ 时, 有 $M(n) \leq n - 1$. 接下来, 分以下几种情形讨论: 根据函数 CFRouting 中第 11 行, 有 $M(n) \leq M(n-1)$; 根据函数 CFRouting 中第 12 – 20 行, 有 $M(n) \leq M(n-1) + 6$; 根据函数 CFRouting 中第 21 – 24, 25 – 29 行, 有 $M(n) \leq M(n-1) + 3$.

因此, 令 $m = \lceil \log |F| \rceil$, 有

$$M(n) \leq \max \left\{ \sum_{i=1}^m 6 + n - m - 1, \sum_{i=1}^m 3 + n - m - 1, n - 1 \right\} \leq \max \{ 6m + n - m - 1, 3m + n - m - 1, n + 1 \} \leq 5m + n - 1 \quad (5)$$

由公式(5), 在最坏情形下, 算法 CFRouting 构造出的无故障路径的最长路径长度的上界为 $5m + n - 1$, 其中 $m = \lceil \log |F| \rceil$. 证毕.

综上所述, 本文提出的时间复杂度为 $O(\lceil \log |F| \rceil n^3)$ 的算法 CFRouting 与时间复杂度为 $O(N \log N)$ 的广度优先搜索算法相比, 在最坏情形下具有明显的优越性.

4 模拟实验及结果分析

本文对算法 CFRouting 用 Python 语言编程实现, 用一台配置为 Intel(R) Core(TM) i5 – 7Y54 CPU 1.61GHz, 8 GB 内存的计算机来评估算法的性能, 并分析实验结果. 实验中, 顶点故障是随机生成的.

实验 1 给定限制故障集合 $F \subset V(C_n)$ 且 $|F| \leq 2n - 3 (2 \leq n \leq 13)$, 此时网络的最大规模为 $2^{13} = 8192$ 个顶点. 将算法 CFRouting 构造出 $C_n - F$ 上一条单播路径花费的 CPU 时间与使用算法 BFS 所得的结果进行比较. 算法 CFRouting 和 BFS 重复运行 100 次的最坏情形如图 2(a) 所示, 平均情形如图 2(b) 所示.

根据实验结果, 随着维度 n 的增加, 构造 $C_n - F$ 上一条单播路径所花费的 CPU 时间逐步增多. 根据实验结果, 当 $n = 13$ 时, 算法 CFRouting 和 BFS 所花费的最坏 CPU 时间分别为 189 ms 和 968 ms, 平均 CPU 时间分别为 163 ms 和 794 ms. 由实验数据可知, 本文算法的执行效率优于广度优先搜索算法.

实验 2 给定限制故障集合 $F \subset V(C_n)$ 且 $2 \leq n \leq 12$, 此时网络的最大规模为 4096 个顶点. 将算法 CFRouting 在 $|F| = 2n - 3$ 所构造的单播路径长度的最大值, 与 $|F| = 0$ 情形下的结果相比较. 算法 CFRouting 重复运行 100 次, 构造出的路径长度的上界情况如图 3(a) 所示, 路径长度的方差如图 3(b)

所示.

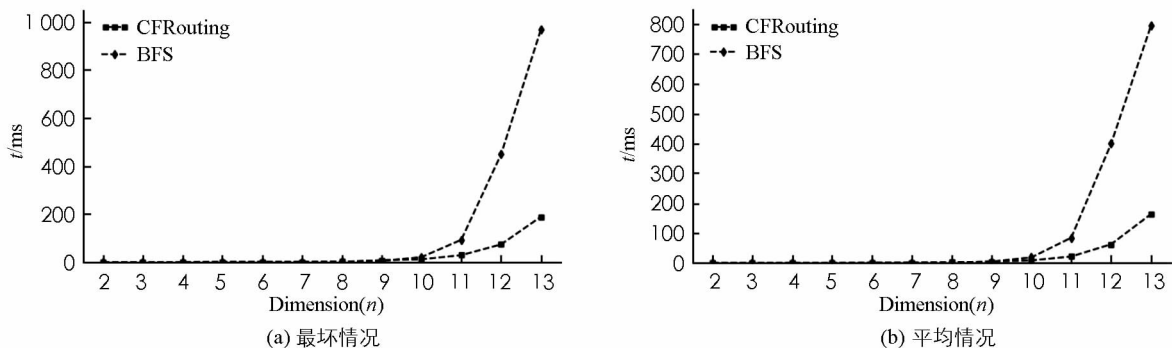


图 2 2 种算法消耗 CPU 时间比较

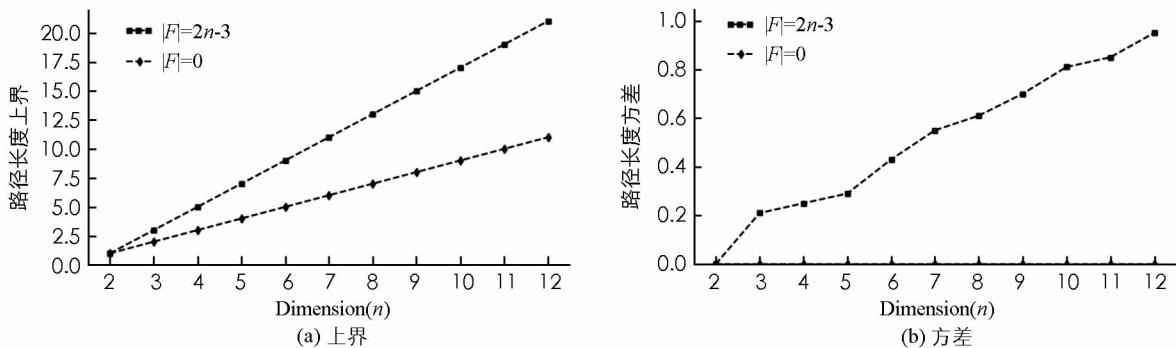


图 3 不同 F 情形下的单播路径长度的上界与方差

根据实验结果,随着维度 n 的增加,算法 CFRouting 构造 $C_n - F$ 上一条单播路径长度的上界逐步增大.根据模拟结果,两种情形下构造的路径长度的上界在 n 较小时差别不大.当 n 较大时,模拟结果可以验证定理 4 结论,两者相差不超过 $5m$,其中 $m = \lceil \log |F| \rceil$.另外,模拟结果可以验证在 $|F| = 2n - 3$ 情形下方差不大.

5 结 论

在一个互连网络中,容错性能的好坏和互连网络的连通度密切相关.随着互连网络规模的扩大,如果用连通度来评估其容错性能则会低估互连网络的容错性能.本文证明了 n 维交错立方体的限制连通度为 $2n - 2$,这一结果几乎是交错立方体上传统连通度的两倍;然后提出了基于该情形下的容错单播算法,并分析了算法的时间复杂度和最长路径长度,进一步利用上述算法进行了仿真.

参考文献:

- [1] 海蓝. 神威·太湖之光继续“称霸”超级计算机排行榜,美国无缘前三 [EB/OL]. (2017-06-19)[2018-03-13]. <http://www.techweb.com.cn/world/2017-06-19/2536748.shtml>.
- [2] HARARY F. Conditional Connectivity [J]. Networks, 1983, 13(3): 347-357.
- [3] ESFAHANIAN A. Generalized Measures of Fault Tolerance with Application to n -Cube Networks [J]. IEEE Transactions on Computers, 1989, 38(11): 1586-1591.
- [4] OH A, CHOI H. Generalized Measures of Fault Tolerance in n -cube Networks [J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 4(6): 702-703.
- [5] LIN R, ZHANG H. The Restricted Edge-Connectivity and Restricted Connectivity of a Ugmented k -ary n -Cubes [J]. International Journal of Computer Mathematics, 2015, 93 (8): 1281-1298.
- [6] WANG X, FAN J, ZHOU J, et al. The Restricted h -Connectivity of the Data Center Network D Cell [J]. Discrete Ap-

plied Mathematics, 2016, 203: 144–157.

- [7] HSIEH S, HUANG H, LEE C. $\{2, 3\}$ -Restricted Connectivity of Locally Twisted Cubes [J]. Theoretical Computer Science, 2016, 615: 78–90.
- [8] LIN L, XU L, ZHOU S, et al. The Extra, Restricted Connectivity and Conditional Diagnosability of Split-Star Networks [J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(2): 533–545.
- [9] HAQ E. Cross-Cube: A New Fault Tolerant Hypercube-based Network [C] // Processing of the 5th International Parallel Processing Symposium, New York: IEEE Computer Society, 1991: 471–474.
- [10] 闫少华, 樊建席. Cross-cube 在 PMC 诊断模型下的诊断性[J]. 计算机工程与应用, 2011, 47(17): 83–86.

A Restricted Fault-Free Unicast Algorithm in Cross-Cubes

WANG Xi^{1,2}, HE Fu-nan¹, ZHANG Shu-kui²

1. Suzhou Institute of Industrial Technology, Suzhou Jiangsu, 215004, China;

2. School of Computer Science and Technology, Soochow University, Suzhou Jiangsu, 215006, China

Abstract: Supercomputers based on parallel systems have always been the hot research topic in industry and academia. The interconnection network, which is the basis of the parallel system, its properties determine the performance of the parallel system directly. The n -dimensional Cross-cube, denoted by C_n , is a variant of the hypercube, which has some features superior to the hypercube, for example: the diameter of the cross-cube is less than that of the hypercube. In this paper, we first introduce the concept of the set of restricted faulty nodes into cross-cubes. We further prove that under the condition that each node of the C_n has at least one fault-free neighbor its restricted connectivity is $2n-2$, which is almost as twice as that of C_n under the condition of arbitrary faulty nodes. Moreover, we give an $O(\lceil \log |F| \rceil n^3)$ fault-free unicast algorithm, and the maximal length of the path constructed by the algorithm is no more than $5m+n-1$, where $m=\lceil \log |F| \rceil$. Finally, we give the simulation result of the expected length of the fault-free path gotten by our algorithm.

Key words: cross-cube; parallel systems; connectivity; restricted connectivity

责任编辑 张 枸