基于局部故障块三维 mesh/torus 网的容错路由

向 东[1)] 陈 爱[2)] 孙家广[1)]

[1)](清华大学软件学院 北京 100084)

[2)](清华大学微电子学研究所 北京 100084)

摘 要 当系统包含很少的故障点时，mesh/torus 网整个系统就有可能是不可靠的.该文采用扩展的局部可靠性信息来指导三维 mesh/torus 网的容错路由.扩展的局部可靠性信息在每个平面内部对无故障节点分类,所以系统中的故障块也是在不同的平面上构成的,而不是基于整个系统.很多基于整个系统不可靠的节点在二维的平面中都会变成可靠的节点.不管是在可靠的系统内,甚或不可靠的系统内,扩展的局部可靠性信息都能有效地指导容错路由.不同于以往的方法,作者的方法不会将任何无故障节点设置为无效节点.所有的故障块都是在平面内构成的,而不是基于整个系统;在一个平面内,任何包含在故障块里的无故障节点仍然可作为出发点或者目标点,这样将大大提高系统的计算能力和性能.模拟结果表明该文方法大大优于已有的方法.

关键词 容错路由;扩展的局部可靠性信息;可靠节点;不可靠系统;三维 mesh/torus 网

中图法分类号 TP302

Fault-Tolerant Routing in 3D Meshes/Tori Based on Locally Formed Fault Blocks

XIANG Dong[1)] CHEN Ai[2)] SUN Jia-Guang[1)]

[1)](*School of Software*, *Tsinghua University*, *Beijing* 100084)

[2)](*Institute of Microelectronics*, *Tsinghua University*, *Beijing* 100084)

Abstract A 3D mesh/torus network may be unsafe even if it contains only a few number of faulty nodes. A new scheme to form fault blocks planarly is proposed to direct fault-tolerant routing in a 3D mesh/torus network. Many unsafe nodes in the whole system become locally safe now. Any fault-free nodes inside a planarly formed fault block can still be a source or a destination. This scheme can greatly improve performance and computational power of the system. Extensive simulation results show that the proposed method outperforms the pipelined-circuit-switching method and two representative methods using wormhole routing and globally formed fault blocks.

Keywords fault-tolerant routing; extended local safety information; safe node; unsafe system; 3D mesh/torus

1 引 言

在近 10 年的实验与商用多计算机系统中,torus 网与 mesh 网得到了广泛的应用.这类多计算机系统有 J-machine, T3D, T3E, Touchstone 及 m-machine.上述多计算机系统大都采用三维的 mesh 或 torus 网.多计算机系统的性能很大程度上决定于点对点的通信代价.最常见的通信方式就是把消息从出发点送到目标点,该方式称为路由.当网络中

收稿日期:2003-04-04;修改稿收到日期:2004-02-23.本课题得到教育部 985 基础研究计划资助.**向 东**,男,1966 年生,博士,副教授,研究方向包括数字系统的设计和测试(可测性设计、可测性分析和 BIST)以及容错计算、分布式计算和计算机网络.E-mail: dxiang@tsinghua.edu.cn.**陈 爱**,男,1980 年生,硕士研究生,研究方向包括容错计算和分布式计算.**孙家广**,男,1946 年生,教授,博士生导师,中国工程院院士,主要研究方向包括计算机图形学、计算机辅助设计和计算机辅助管理.

的节点增加时,节点和连线产生故障的可能性也大大增加.在一个存在故障节点系统中的通信称为容错通信.

文献[1～12]广泛研究了多计算机网络中的容错通信.Gaughan 和 Yalamanchili[4] 提出了一种称为 PCS(Pipelined Circuit Switching)的策略,该策略是虫孔交换技术的一种改进策略.在传送消息之前,PCS 需要先设置一条路径.当系统中包含大量的故障点时,预先设定路径能保证消息成功地传送.Chien 和 Kim[3] 提出了基于平面的路由算法,该算法虽然限制了路由的自由度,但只需 3 条虚拟通道就能避免 n 维 mesh 或 torus 网的死锁.对该算法进行适当的改进就能有效地处理 n 维网络的容错路由.Boppana 和 Chalasani[1] 在 e-cube 路由算法和故障块模型的基础上,提出了一种 mesh 网络的容错路由算法.这种方法利用局部故障信息指导消息传输.使用故障环和故障链使消息能够沿着故障区域周围传输.在任意维 mesh 网中实现完全适应性算法最多只需 4 条虚拟通道[1].文献[1～3]均采用了虫孔交换技术.文献[5,12]提出基于虫孔交换技术的容错路由算法,上述方法可容忍凸型的故障块,并需将部分无故障点标志为不可用节点.

在利用虫孔交换技术的网络中,当系统中只有一个消息或者负载很低时消息在出发点和目标点的延时主要决定于启动时间(start-up time).当系统的通信量很高时,最短路径的选择是非常重要的.最短路径路由具有下述吸引人的特点:首先,基于最短路径的路由不会浪费任何系统资源,因为所有消息不会向远离目标点的方向移动,所以网络的通信能力可得到充分利用;其次,最短路径路由允许消息传送被限制在一个子网络里.与非最短路径路由相比,可提高系统的吞吐量.前面提到的绝大部分基于局部故障信息的通信方法只能利用每个节点的邻节点的可靠信息进行路由,所以虽然在很多情况下最短路径存在,却不能保证能够沿着最短路径传送消息.

矩形故障块模型非常简单却能提供标准化的路由.但是系统含有很少的故障,如果故障块是在整个系统内形成[1～3,7～9],大量的甚至所有无故障节点被标记为不可用.本文的方法在不同的平面内独立构造故障块,因此每个无故障节点保存了包含它的 3 个平面的可靠信息.扩展的局部可靠性(Extended Local Safety)信息将用于指导三维 mesh/torus 网上的容错路由.包含在每个平面构成的故障块里的所有无故障节点也能够当作出发点或者目标点,所以没有一个无故障节点是不可用的,这将大大提高系统的计算能力.当系统不可靠时,采用局部故障信息[1～3,5,6,12]效果没有 PCS 好.同样,当系统包含很大的故障块时,如果故障块之外的消息不能进入故障块将限制路由算法的适应性.我们提出的方法在平面内构造故障块,不同于基于平面的适应性路由算法[3]和扩展的可靠性参数算法[8],任何时候消息传送都不被限制在一个平面内,这样能提高路由算法的适应性和性能.我们的算法需要处理的信息量是以前算法[1～3,7～9]的两倍左右,但大大提高系统的计算能力和性能,并在很大程度上提高了找到最短路径路由的可能性.所增加的时间开销比因此而节省的消息传送时间要短得多,代价是可以接受的.

任何容错路由算法只要利用局部或全局的故障信息都存在及时更新的问题.本文算法也是这样,但并不需要每时每刻地更新,只要定期更新即可.

2 预备知识

一个 $k\times k\times k$ 的三维 mesh 网包含 k^3 个节点,其中每条边包含 k 个节点.在 $k\times k\times k$ 的三维网络中,如果两个标记为 (a_3,a_2,a_1) 和 (b_3,b_2,b_1) 的节点其标记只有在一个维度 i 是不相等的,且 $a_i=(b_i+1)\bmod k$,则这两个节点是相邻的;而如果在 mesh 网中,则要求 $|a_i-b_i|=1$ 时,这两个节点才相邻($i\in\{1,2,3\}$).

假定二维 mesh 网的故障点的形状是矩形.二维 mesh 网中,当故障点集 F 具有一个以上的矩形,并且每个矩形具有以下性质:①直角边上不包含任何故障点;②矩形内部包含 F 的所有故障点;③矩形内部不包含任何不属于 F 的节点,称 F 为一个故障块.在以往的方法[1～3,7,9]中,三维 mesh 网的故障块模型如下定义:一个节点 $x=(x_3,x_2,x_1)$ 被称为节点 $y=(y_3,y_2,y_1)$ 的基节点(base-node),当且仅当对于所有的 $i\in\{1,2,3\}$,$x_i\leq y_i$;如果 x 是 y 的基节点,则 y 称为 x 的顶节点(apex-node);以基节点 x 和顶节点 y 构成的一个块 B_{xy} 包含 $N_{xy}=\{(z_3,z_2,z_1)\mid x_i\leq z_i\leq y_i,1\leq i\leq 3\}$ 的所有节点.

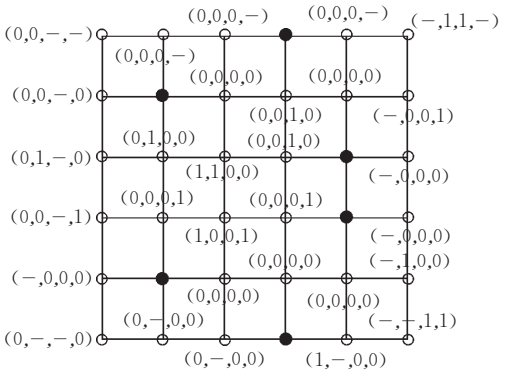
块 B_{xy} 也包含 N_{xy} 中任意两个节点的连线.如果一个节点 $z=(z_3,z_2,z_1)$ 对于某个 $i\in\{1,2,3\}$,有 $z_i=x_i$ 或者 $x_i=y_i$ 则称 z 是块 B_{xy} 的边界节点.当且仅当 v 和 v' 都是边界节点时,连线 (v,v') 称为边界连线.B_{xy} 的内部包含除 B_{xy} 的边界以外的所有节点和连线.

定义 1.　在一个子 mesh 网中,故障点集 F 称为故障块 $T(Q)$,当且仅当存在节点 x 和节点 $y(x,y\in T(Q))$,并且具有如下性质:(1)B_{xy} 的边界连线和节点都没有故障;(2) B_{xy} 的内部包含且只包含 F 的所有故障;(3) 二维子 mesh 网的故障形成如前定义的故障块.

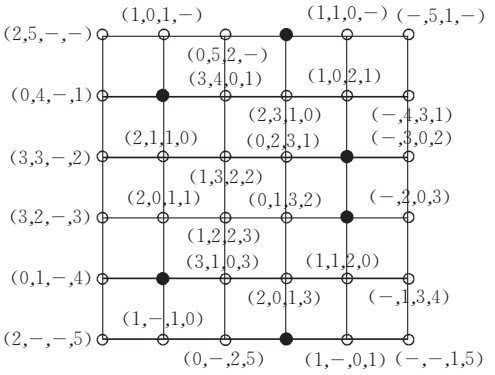
三维 mesh 网上一个节点的扩展的可靠参数[8]用 6 元组$(v_1',v_2',\cdots,v_6')$来表示,$v_i'(1\leq i\leq 6)$代表该节点在方向 i 上到最近的故障块的距离.二维 mesh 网上的局部可靠性[10]可扩展到三维 mesh 网上.

定义 2.　三维 mesh 网上节点 v 的局部可靠性由一个 6 元组$(v_1,v_2,\cdots,v_6)$确定,$v_1,v_2,\cdots,v_6$代表分别沿不同方向从 v 点出发最长的可行路径长度(不进入故障块内部).

扩展的可靠参数[8]与局部可靠性[10]之间最明显的差别是:扩展的可靠参数指导消息传送尽量远离故障块,但是局部可靠性指导消息可靠地沿最短的可行路径路由.图 1(a)和(b)标示了边长为 6×6 mesh 网中所有无故障节点扩展的可靠参数与局部可靠性,其中"−"代表此方向不用考虑.4 元组(a,b,c,d)分别为东、西、南、北方的可靠参数.任意分布的故障模型可将一些无故障点变成不可靠点而形成故障块.



(a) 扩展的可靠性参数



(a) 局部可靠性

图 1　扩展的可靠性参数与局部可靠性

3　矩形故障块的性质和不可靠三维 mesh 网的特征

定义 3.　mesh 网中的节点分为故障节点、不可靠节点和可用节点.如果一个无故障节点在至少两个不同维度上与故障节点或者不可靠节点相连,称为不可靠节点;否则称为可用节点.如果一个系统中的所有无故障节点都是不可靠的,此系统称为不可靠的.

上面的定义是一个递归定义.显然所有的不可靠节点和故障节点形成一系列故障块.应该指出,我们说故障块断开网络指的是网络被故障块断开,但不意味着网络真的被故障断开了.我们有如下引理.

引理 1.　一个可靠的 mesh 网具有如下性质:(1)如果系统没有被任何故障块断开,则所有可用节点是相连的;(2)如果系统被故障块断开,则每个连通片内的所有可用节点是相连的.

较少的故障点就可使得一高维的 mesh 网不可靠.不可靠的三维 mesh 网或者三维 mesh 中的一个故障块的内部存在下述有趣的性质.

引理 2.　如果 $k\times k\times k$ 的三维 mesh 网中的一个故障块为一个三维子 mesh 网,此故障块的每个二维子 mesh 网都至少包含一个故障节点.

定理 1.　如果一个三维 mesh 网是不可靠的,该网每个平面都至少包含一个故障节点.

证明.　如果一个平面不包含任何故障节点.此平面内的任意一个无故障节点 v 最多只包括沿同一维度的两个邻接点是不可靠的或者是故障节点.因此,根据定义 3,节点 v 在整个三维立方中不可能是不可靠的.所以整个系统将不是不可靠的,这与前提冲突.　　证毕.

引理 2 的证明类似.考虑图 2 所示的 5×5×5 mesh 网,每个平面至少有一个故障节点.虽然网络仅含 7 个故障节点,该网整个系统是不可靠的.根据定义 3 我们有如下定理.

定理 2.　一个边长为 k 的三维不可靠 mesh 网至少包含$[3(k-1)/2]+1$ 个故障节点.如果 $k-1$ 是偶数,任意一对故障节点不能相连;如果 $k-1$ 是奇数,最多有一对故障节点能够相连.

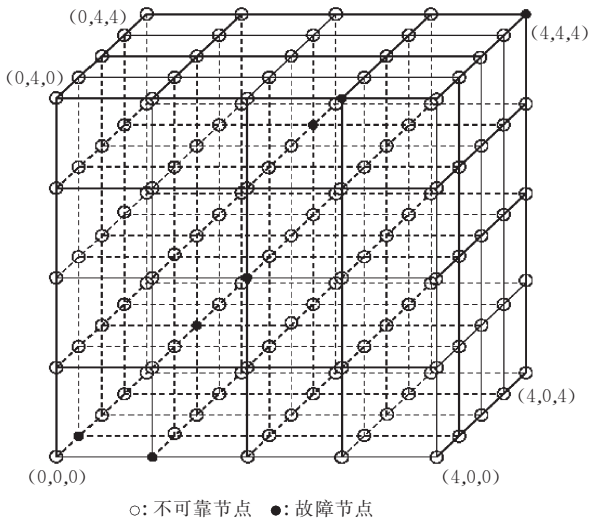


图 2 一个不可靠的 5-ary 三维 mesh 网络

4 扩展的局部可靠性

局部可靠性信息已有效地应用于超立方多计算机的容错通信[9,11]. 当一个三维立方系统不可靠时,许多子 mesh 网内部可靠的消息路由仍然是可行的. 让我们再次考虑图 2 所示 5×5×5 mesh 网. 除了(0,0,0)和(1,0,1)在平面(*,0,*)中,其它所有无故障节点在每个平面中都是局部可用的. 实际上,消息能在任意一对无故障节点之间可靠地传递. 我们将考虑每个无故障节点在包含它的每个平面中的局部可靠性,此时不考虑平面之外故障的影响. 以上的策略使许多全局不可靠的节点在特定平面中成为可用的.

定义 4. 一个容错三维 mesh 网中的无故障节点在特定的子 mesh 网中进行分类:如果一个无故障节点在一个子 mesh 网中至少在两个不同维度与故障节点或者不可靠节点相连,则称此节点为这个子 mesh 网中的局部不可靠节点;否则称为局部可用节点.

引理 3. 如果节点 v 在子 mesh 网 SM_1 中是局部不可靠的,则它在包含 SM_1 的子 mesh 网 SM_2 中也是局部不可靠的;如果一个节点在 SM_2 中是局部不可靠的,并不意味着它在 SM_1 中是局部不可靠的.

证明. 假设节点 v 在 n_1 维的子 mesh 网 SM_1 中是局部不可靠的,在 SM_1 中 v 有 $k(k\leq 2n_1)$ 个邻接点. 在这 k 个邻接点中至少有 2 个故障节点或者局部不可靠节点. 在包含 SM_1 的 $n_2(n_1\leq n_2)$ 维子 mesh 网 SM_2 中,节点 v 具有 $k'-k$ 个额外的邻接点(在 SM_2 中 v 有 k' 个邻接点,$k'\leq 2n_2$). 所以,在 SM_2 中 v 不可能成为局部可用的. 证毕.

引理 4. 如果节点 v 在子 mesh 网 SM_2 中是局部可用的并且 SM_2 包含 SM_1,则它在子 mesh 网 SM_1 中也是局部可用的.

证明. 在 n_2 维的子 mesh 网 SM_2 中 v 有 $k'(k'\leq 2n_2)$ 个邻接点,因为 v 在 SM_2 中是局部可用的,根据定义 4 这些邻接点中至多有一个故障节点或者局部不可靠节点. 在 $n_1(n_1\leq n_2)$ 维的子 mesh 网 SM_1 中 v 有 $k(k\leq 2n_1)$ 个邻接点,这 k 个邻接点包括在前面的 k' 个邻接点之内. 显然,在 SM_1 中的 k 个邻接点最多有一个故障节点或者局部不可靠节点. 所以,在 SM_1 中 v 仍旧是局部可用的. 证毕.

定理 3. E_1 为子 mesh 网 SM 的所有局部可用节点集合,E_2 为整个 mesh 网中包含在 SM 中的可用节点集合,则 E_2 必定是 E_1 的子集.

根据引理 3 和引理 4,定理 3 显然成立. 考虑图 2 所示的三维 mesh 网,在整个系统中所有无故障节点都是不可靠的. 但是除了(0,0,0)和(1,0,1),其它所有节点在包含它们的所有平面中都是局部可用的(可靠的). 节点(0,0,0)和(1,0,1)在平面(*,0,*)中是局部不可靠的,因为在这个子 mesh 网中它们有两个故障邻接点.

定义 5. 三维 mesh 网中的一个无故障节点 v 的扩展的局部可靠性用 6 个数$(v_1,v_2,\cdots,v_6)$确定,$v_i(1\leq i\leq 6)$的值通过公式 $v_i=\min_j\{v_{ij}\}$ 得到. v_{ij} 表示 v 在沿 i 方向和 j 方向形成的平面中,沿 i 方向的局部可靠性.

一个无故障节点在不同的子网络里可能为不同类型的节点. 如图 2 所示,节点(0,0,0)在平面(0,*,*)和(*,*,0)中是局部可用的,但在平面(*,0,*)中是局部不可靠的. 显然与以前的方法[1~3,8~10]相比,扩展的局部可靠性能够大大提高 mesh 网络的计算能力,因为以前的方法会使大量的无故障节点成为不可用节点. 与局部可靠性[10]相比,本文提出的方法能够为容错路由提供更多的信息,因为许多基于局部可靠性而不可靠的节点在平面中成为局部可用的. 一个无故障节点扩展的局部可靠性信息可以通过下面的简单程序获得:

extended-local-safety ()

对于系统中的每个无故障节点 v,同步进行如下操作:

local-class(v);

同步操作结束.

重复上述处理,直到获得稳定的状态.

local-class (i)

1. 节点 i 获得它在包含它的不同平面中的状态;

2. 如果节点 i 在一个平面中沿不同的维度包含至少两个故障邻接点或者不可靠邻接点,则把 i 在这个平面中的状态设置为不可靠的.

在三维 mesh 网中,一个无故障节点扩展的局部可靠性可以用 (E,S,F,W,N,B) 表示,E,S,F,W,N,B 分别代表该节点沿东、南、前、西、北、后 6 个方向的扩展的局部可靠性值. 如图 2 所示,节点(0,1,0)的扩展的局部可靠性为(1,2,—,—,2,4),节点(3,2,2)为(1,2,2,3,2,2) . 我们考虑无故障节点在每个平面中的可靠性,并有以下引理.

引理 5.　在一个三维 mesh 网中,$(V_1,V_2,\cdots,V_6)$,$(V_1',V_2',\cdots,V_6')$和$(V_1'',V_2'',\cdots,V_6'')$代表节点 v 的扩展的局部可靠性、局部可靠性和扩展的可靠性参数,则有 $V_i\geq V_i'\geq V_i''(i\in\{1,2,\cdots,6\})$.

证明.　扩展的可靠性参数[7,8]指导容错路由远离故障块,使之不会达到故障块的边界;局部可靠性信息则允许到达故障块的边界,其目标是沿着最短路径可靠地传送消息,所以总是有 $V_i'\geq V_i''(i=1,2,\cdots,6)$. 然而基于局部可靠性信息的可用路径永远不会进入故障块内部;扩展的局部可靠性考虑每个平面内部的可靠性,许多在整个系统中是不可靠的节点在平面中成为局部可靠节点,基于扩展的局部可靠性的可用通信路径可以进入整个系统的故障块内部,则 $V_i\geq V_i'(i=1,2,\cdots,6)$.　　证毕.

定义 6.　一个节点 s 相对于结点 d 是扩展的且局部可靠的,如果 $V_{i1}\geq d_{i1},V_{i2}\geq d_{i2},\cdots,V_{il}\geq d_{il}$. 其中 s 和 d 在方向 $i_1,i_2,\cdots,i_l$ 上是不同的,$V_{i1},V_{i2},\cdots,V_{il}$ 是节点 s 沿方向 $i_1,i_2,\cdots,i_l$ 的扩展的局部可靠性的值,$d_{i1},d_{i2},\cdots,d_{il}$ 是 s 和 d 沿这些方向的彼此间的距离.

5　基于扩展的局部可靠性信息的容错路由

5.1　容错路由算法

先定义下述启发式函数来指导路由的路径选择

$$h_i=\begin{cases}0, & d_i\geq L_i(s,d)\\ d_i-L_i(s,d), & \text{其它}\end{cases}\qquad(1)$$

$L_i(s,d)$表示在维度 i 上,s 和 d 之间的距离;d_i表示 s 在维度 i 上的扩展的局部可靠性信息,h_i表示 s 相对于 d 在维度 i 上的启发值. 在传递消息时,尽量沿启发值最小的维度上传递消息. 因为如果在某个维度上的启发值越大,表明沿此维度传递消息越不受阻碍. 为了尽可能找到最短路径,应该先沿启发值小的维度上传递消息. 我们的模拟试验证明这个策略大大提高了找到最短路径路由的可能性.

我们将介绍一种新的路由算法,利用启发式函数来引导从出发点到目标点最短路径的选择. 在传递消息之前,此算法需要预先设置路径. 公式(1)所定义的启发值将用来指导消息的传递. 每个消息在特定的虚拟子网(virtual subnetwork)中传送. 我们假设出发点并不知道目标点扩展的局部可靠性信息. 先把一个信号从出发点向目标点传送,如果从出发点到目标点的最短路径已经被找到,目标点将把一个信号返回到出发点,然后消息从出发点沿着设定的路径通信. 否则,根据公式(1)把一个信号沿启发值最小的方向从目标点向出发点传送. 如果已找到最短路径,则沿此最短路径传送消息. 否则,根据目标点扩展的局部可靠性信息和当前点的位置引导消息传送路径的选择. 在设置路径和消息的传送过程中,出发点或者目标点相对于当前点的启发值是不断变化的. 路由算法的描述如下:

route-message-in-3D-mesh ()

1. 在特定的虚拟子网中,保持出发点 s 的扩展的局部可靠性信息的同时建立一条从出发点 s 到目标点 d 的路由路径.

2. 如果已经建立了 s 和 d 之间可用的最短路径或者 s 和 d 只在一个维度上不同,则沿着建立的路径把一个信号从 d 送到 s,然后把消息沿建立的路径从 s 传送到 d;否则转步 3.

3. 在对应的虚拟子网中,尝试建立从 d 到 s 可用的最短路径,执行 *select-path*().

4. 如果可用的最短路由路径已经建立,把消息在已设定的虚拟子网中沿建立的路径从 s 传送到 d,否则执行 *send-message*().

select-path ()

1. 根据当前节点 v 的位置和出发点的扩展的局部可靠性信息及公式(1)计算出不同维度的启发式函数值. 如果维度 t 具有最小的启发值 h_t 并且 v 沿此维度最短路径上的邻接点在某个平面中是局部可用的,则设置路径通过此邻接点,否则转步 2.

2. 如果上述 v 的邻节点在所有平面中都是局部不可用的,则在剩下的维度中选择启发值最小的方向的局部可用邻接点设置路径.

3. 继续上面的步骤直到当前点和出发点 s 只相差一个维度,如果在这个节点与 s 之间找到了可用路径,则可用的路由路径已经建立,否则转步 4.

4. 把信号沿 s 具有扩展的局部可靠性最大值的维度上从当前点向其邻接点推进,并且尽可能沿最短路径推进. 继续上述步骤直到到达出发点 s.

send-message ()

1. 根据当前节点 v 的位置和目标点 d 扩展的局部可靠性信息计算出不同维度的启发值. 如果维度 t 具有最小的启发值 h_t 并且 v 沿此维度最短路径上的邻节点在某个平面中是局部可用的,则把消息传送到此邻接点,否则转步 2.

2. 如果上述 v 的邻节点在所有平面中都是局部不可用的,则在剩下的维度中选择启发值最小方向的局部可用邻接点传送消息.

3. 继续上面的步骤直到消息所在的当前点和目标点 d 只相差一个维度.

4. 如果当前点和目标点之间存在可用的最短路径,则沿此路径把消息传送到目标点;否则把消息推进到一个无故障的邻接点直到当前点和目标点在其它维度上最短路径上的邻接点是无故障的.

5. 继续上述步骤直到消息到达目标点.

5.2 采用虚拟子网划分来避免死锁

一个物理网络可划分为几个虚拟子网,每种消息在唯一的虚拟子网中路由. 如果在虚拟子网内部或者几个虚拟子网之间不存在回路,将能够避免死锁.

一个三维 mesh 网可以划分为 8 个不同的虚拟子网: $x+y+z+$, $x+y+z-$, $x+y-z+$, $x+y-z-$, $x-y+z+$, $x-y+z-$, $x-y-z+$, $x-y-z-$. 这 8 个虚拟子网可以合并成 4 个不同的虚拟子网: $x+y+z*(c_1+,c_1+,c_1)$, $x-y*z+(c_2-,c_2,c_2+)$, $x-y*z-(c_1-,c_3,c_2-)$, $x+y-z*(c_2+,c_1-,c_3)$. 括号内的标记表示每个虚拟子网所分配的虚拟通道. 所有消息根据出发点和目标点的位置进行分类. 例如,一个从节点(0,1,0)传送到节点(3,0,3)的消息被划分为第四类消息. 两条额外的虚拟通道 c_4, c_5 被用来引导消息沿着非最短路径传播. 显然,在任何虚拟子网内部和虚拟子网之间不存在回路依赖关系. 以上策略为路由算法提供了更强的适应性.

6 模拟结果

基于扩展的局部可靠性信息(els)的容错路由算法我们实现了一个新的模拟器. 我们还实现了基于平面的适应性路由算法(wh1)[3]、Boppana 和 Chalasani 提出的基于虫孔交换技术的算法(wh2)[1]和基于 PCS 算法(pcs)[4]的模拟器以便与本文的算法进行比较. 模拟结果显示 Boppana 和 Chalasani 提出的 wh2 算法[1]效果比基于平面的适应性路由算法[4]要略好,这是因为后者限制了路由的自由度,而且对于每条物理通道后者只用了 3 条虚拟通道而前者用了 4 条虚拟通道. 然而,当系统中的故障节点数增加后,上述两方法的效果都迅速下降. 所有的模拟结果都是考虑三种不同故障分布模式的平均值. 消息的长度和每个节点的缓存数目分别设定为 16 个 flit 和 90 个 flit(通常一个 flit 为 4 个 64 位字,或 32 字节). 两个重要的指标分别为延时(latency,传送一个消息的周期数)和吞吐率(throughput,flit/node/cycle)用于评价系统的性能.

图 3 提供了当 8×8×8 的 mesh 网中包含 20 个故障点时,在不同的负载下,4 种方法的效果比较. 图 4 提供了当 8×8×8 的 mesh 网中负载为 0.10 时,在不同的故障节点数下,4 种方法的效果. 图 5 供了当 16×16×16 的 mesh 网中包含 50 个故障点时,在不同的负载下,4 种方法的效果比较. 图 6 提供了当 16×16×16 的 mesh 网中负载为 0.05 时,在不同的故障节点数下,4 种方法的效果比较.

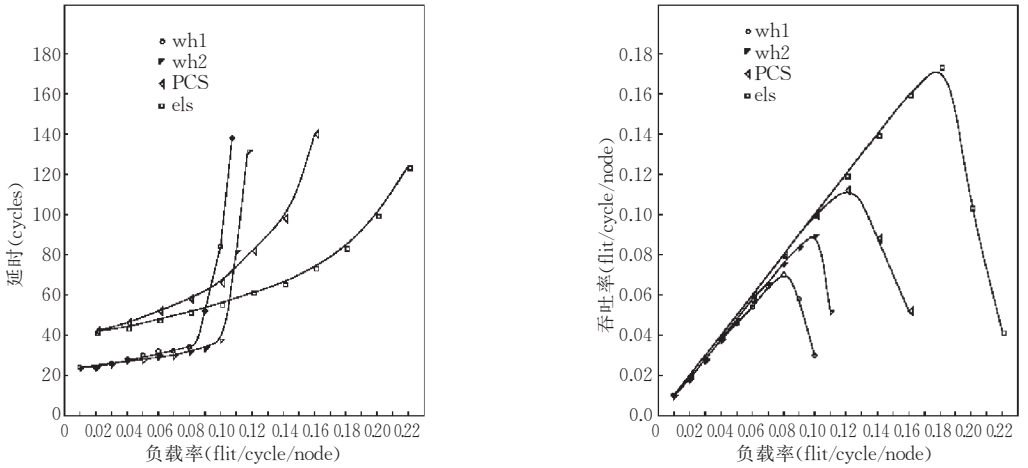


图 3 8×8×8 mesh 网含固定故障的性能评价

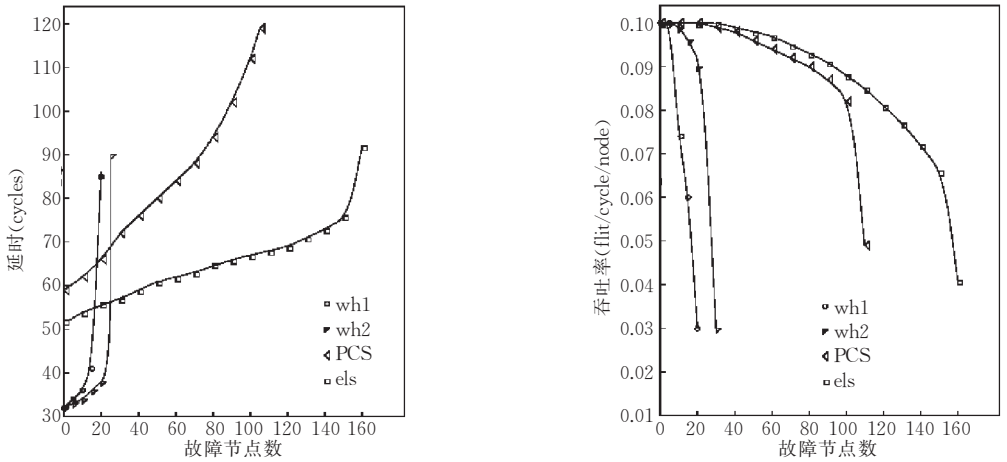


图 4　8×8×8 mesh 网在负载固定时的性能评价

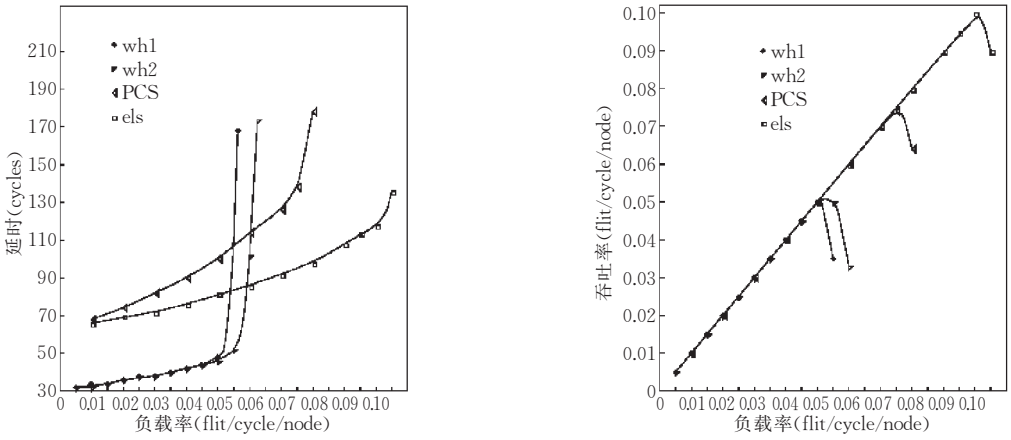


图 5　16×16×16 mesh 网含固定故障点的性能评价

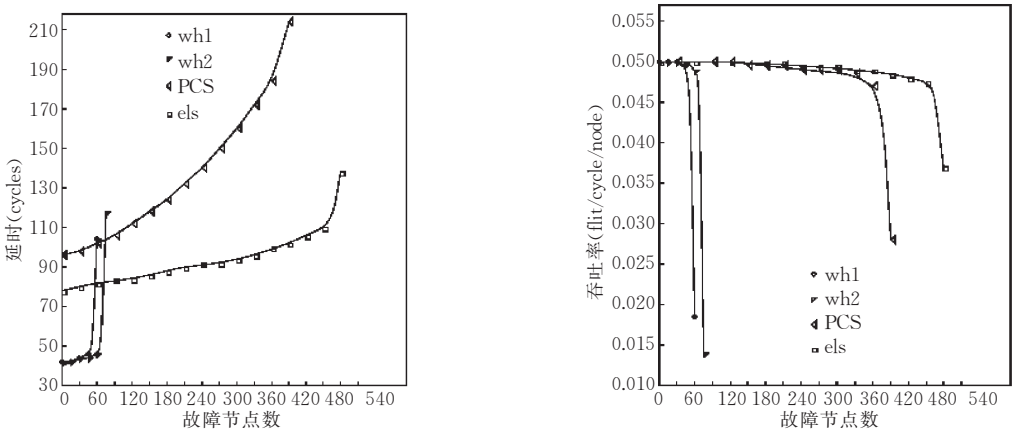


图 6　16×16×16 mesh 网在负载固定时的性能评价

7　总　结

本文提出了一种称为扩展的局部可靠性来指导三维 mesh 网中的容错路由.该方法基于三维 mesh 网中每个平面构造故障块,很多在整个系统中是不可靠的节点在平面内成为可用的.本文方法允许故障块中的无故障点作为出发点和目标点传送消息,因而不需要像文献[1～3,5～8,12]那样需将网络中任意无故障点标志为不可用接点,可大大提高系统的计算能力和系统性能.模拟结果表明本文方法优于已有的方法.

参 考 文 献

1 Boppana R. V., Chalasani S.. Fault-tolerant wormhole routing algorithms for mesh networks. IEEE Transactions on Computers, 1995,44(7):848～864

2 Boura Y. M., Das C. R.. Fault-tolerant routing in mesh networks. In:Proceedings of International Conference on Parallel Processing, 1995, I106-I109

3 Chien A. A., Kim J. H.. Planar adaptive routing: Low-cost adaptive networks for multiprocessors. Journal of ACM, 1995, 42(1):91～123

4 Gaughan P. T., Yalamanchili B. V., Dao S., Schimmel D. E.. Distributed, deadlock-free routing in faulty, pipelined, direct interconnection networks. IEEE Transactions on Computers, 1996, 45(6):651～665

5 Park S., Youn J. H., Bose B.. Fault-tolerant wormhole routing algorithms in meshes in presence of concave faults. In: Proceedings of IEEE International Parallel and Distributed Processing Symposium, 2000, 633～638

6 Su C. C., Shin K. G.. Adaptive fault-tolerant deadlock-free routing in meshes and hypercubes. IEEE Transactions on Computers, 1996, 45(6): 666～683

7 Wu J.. Fault-tolerant adaptive and minimal routing in mesh-connected multicomputers using extended safety levels. In:Proceedings of IEEE International Conference on Distributed Computing Systems, 1998, 428～435

8 Wu J.. A fault-tolerant adaptive and minimal routing approach in 3D meshes. In: Proceedings of the 7th IEEE International Conference Parallel and Distributed Systems, 2000, 256～263

9 Xiang D.. Fault-tolerant routing in faulty hypercube multicomputers based on local safety information. IEEE Transactions on Parallel and Distributed Systems, 2001, 12(9):942～951

10 Xiang D., Chen A.. Fault-tolerant routing in 2D tori or meshes using limited global safety information. In:Proceedings of the 31th IEEE International Conference on Parallel Processing, 2002,231～238

11 Xiang D., Chen A.. Reliable broadcasting in wormhole-routed hypercube-connected networks using local safety information. IEEE Transactions on Reliability, 2003, 52(2):245～256

12 Zhou J., Lau F. C. M.. Adaptive fault-tolerant wormhole routing in 2D meshes. In: Proceedings of IEEE International Symposium on Parallel and Distributed Processing, 2001, 56～61

XIANG Dong, born in 1966, received the B. S. in 1987 and the M. S. in 1990 in Computer Science from Chongqing University. He received the Ph. D. in 1993 in Computer Engineering from the Institute of Computing Technology, the Chinese Academy of Sciences, Beijing. He visited Concordia University, Montreal, Canada as a post-doctor from 1994 to 1995, and the University of Illinois, Urbana Champaign from 1995 to 1996. He visited Nara Institute of Science and Technology, Japan as a JSPS fellow from Apr. to Sept., 2003. He is now with the School of Software at Tsinghua University as an Associate Professor. His research interests include design and test of digital systems (design for testability, testability analysis, and BIST) and fault-tolerant computing, distributed computing, and computer networking.

CHEN Ai, born in 1980, received the B. S. in 2001 in Electronic Engineering from Tsinghua University. He is working toward the M. S. degree at the Institute of Microelectronics, Tsinghua University. His research interests include fault-tolerant computing and distributed computing.

SUN Jia-Guang, born in 1946 and graduated from the Department of Automation of Tsinghua University in 1970. He is a professor and doctoral candidate supervisor in the Department of Computer Science and Technology, the dean of the School of Software at Tsinghua University. He is a member of the Chinese Academy of Engineering. His research interests include computer graphics, computer-aided design, and computer aided management.

Background

This work is supported in part by the 985 Fundamental research grant of the Education Ministry "Fault-Tolerant Communication for Multicomputer Systems". The work in this paper is fault-tolerant routing for mesh-connected networks. It is an important part of the project. Authors have done a couple of work on fault-tolerant high-performance communication algorithms for multicomputer networks.