研究与开发

支持低成本快速局部重构的快速Benes网络

秦梦远[1,2]，刘宏伟[1]，郝沁汾[1]
（1. 中国科学院计算技术研究所，北京 100095；
2. 中国科学院大学，北京 101408）

摘　要：为了解决互连规模大于100时快速可重构光互连网络单次局部重构代价过高的问题，提出了快速Benes网络与配套局部重构算法，利用预留空置链路减少局部重构对已有链路的影响，在互连规模超过100时性能优异。在处理单一节点的路由变更时，快速Benes网络仅影响平均2~4个接入节点对应的既有通信链路，略差于Crossbar网络，而远好于Benes网络（一次平均影响0.71N个接入节点，N为互连规模），降低重构代价达98%。基于该算法的现场可编程门阵列（field-programmable gate array，FPGA）硬件加速器，局部路由求解速度为79 ns/次，与Crossbar网络相近，比Benes网络快2个数量级。

关键词：快速Benes网络；局部重构；节点同步成本
中图分类号：TN401；TP302.2
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2026003

Fast Benes network that supports fast low-cost partial route reconfiguration

Qin Mengyuan[1,2], Liu Hongwei[1], Hao Qinfen[1]
1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100095, China
2. University of Chinese Academy of Sciences, Beijing 101408, China

Abstract: To address the problem of excessively high costs for single partial reconfiguration in a fast reconfigurable optical interconnection network when the interconnection scale is greater than 100, a fast Benes network and a supporting partial reconfiguration algorithm were proposed. By utilizing reserved idle links, the impact of partial reconfiguration on existing links was reduced, and it exhibited excellent performance when the interconnection scale exceeded 100. When dealing with the routing change of a single node, the existing communication links corresponding were affected to an average of 2~4 access nodes by the fast Benes network. It was only slightly inferior to the Crossbar network, and was much better than the Benes network. The reconfiguration cost was significantly reduced by up to 98%. Based on the FPGA hardware accelerator of this algorithm, the partial routing solution speed was 79 nanosec-

收稿日期：2025−03−17；**修回日期：**2025−07−10
通信作者：郝沁汾，haoqinfen@ict.ac.cn
基金项目：国家重点研发计划项目（No.2022YFB4401501）；江苏省重点研发计划项目（No.BE2023006-4）
Foundation Items: The National Key Research and Development Program of China (No.2022YFB4401501), The Key Research and Development Program of Jiangsu Province (No.BE2023006-4)

onds per time, which was similar to that of the Crossbar network and two orders of magnitude faster than the Benes network.

Key words: fast Benes network, partial reconfiguration, node synchronization cost

0　引言

目前，数据中心带宽和互连规模需求暴增，可重构光互连网络对比传统电互连网络具有显著的互连规模、带宽与功耗优势，并且在重载应用场景下通信时延性能远好于电互连网络。可重构光互连网络的本质为可动态配置的点对点网络，其交换功能的实现依赖于对网络内链路的重构操作，越高的可用重构频率意味着网络越能及时响应新的路由请求，链路切换耗时越短，交换性能越好。近年来，基于2×2快速光开关单元构建的开关网络将开关状态切换时延降至纳秒级[1-2]，理论重构频率相比以往数据中心光互连网络[3-4]提升了6个数量级，解决了文献[5]指出的光互连网络性能问题，使之具有承载高性能动态交换需求的潜力。

但使用快速光开关单元构建互连规模$N>100$的开关阵列时，传统Crossbar结构将不再可用：其需要N^2个开关单元构建，导致构建成本高昂，且链路平均穿过N个开关单元，插入损耗过高影响数据通信。因此基于快速光开关单元的阵列通常采用Benes[6]这一紧凑的网络结构[7-9]，将开关单元数降低至$N\times(2\,\mathrm{lb}N-1)/2$，使链路穿过开关单元数均匀且为$2\,\mathrm{lb}\,N-1$，同时保有可重排无阻塞特性。256×256交换规模的Benes网络拥有与64×64 Crossbar网络近似的开关单元数，以及与16×16 Crossbar网络近似的低插入损耗，有效降低了百节点互连规模下的构建成本与插入损耗。因其构造紧凑的特性，Benes网络也被用于片上众核网络构建[10]或领域专用处理器构建[11]。

但Benes结构自身链路耦合度非常高，网络内常见的单一链路的路由变更很难不影响其余既有链路。尽管相关研究[12-17]给出了任意非冲突路由请求下与之匹配的Benes网络全局重构求解算法，但这些算法仅关注如何更好地进行并行求解[13-14]，或更快地进行硬件加速求解[15-17]，而不讨论对既有链路的保持，也不统计具体哪些既有链路的内部路由在当次重构后发生了变化。针对这些算法的链路保持测试发现，在仅变更两路链路，交换它们的路由终点并保持其余链路的路由不变时，这些算法平均破坏$0.71N$条既有链路，且被破坏既有链路中位数高达$0.92N$，说明绝大多数既有链路均在当次重构过程中发生了内部路由改变。在应用这些算法的互连系统中[18]，由于无法确认仍能保持的链路，只能假定全部链路均会被破坏，每次重构前需要排空整个网络内的数据流量，网络实际吞吐量大幅降低，单次路由重构的总开销高昂。文献[18]通过TopoCoin方法合并尽量多的局部路由请求为一次全局路由重构以降低全局重构频率，但仅在网络内全部节点的通信请求均可预测时，能够通过预先计算掩盖毫秒级时延，获得性能提升。其同样指出，若通信请求难以预测，则TopoCoin方法效果甚微。

不难发现，Benes网络实际吞吐量不高的原因在于缺乏高效的局部重构手段，当其面临常见的单节点路由变更请求时，总是退化为代价高昂的全局重构模式，被迫触发全局流量排空。若在使用Benes网络时，不再接入全部端口，而是留空一部分端口构建空置链路，则有望通过破坏空置链路实现低成本的局部重构，使受影响的既有链路数大幅降低，对应重构前仅需排空这些链路的流量即可，进而大幅降低重构代价。与Benes网络构型相似的扩张型Benes网络，通过提供一

倍的空置链路实现更高的信噪比[19-20]，相应的路由求解算法[21-23]研究也较为成熟，但与现有Benes网络全局重构算法一样，缺乏对既有数据链路保持能力的相关研究，不能有效降低排空流量链路数。

本文针对电路交换网络大规模互连与低重构代价难以兼得的问题，提出快速Benes网络，并基于扩张型Benes网络预留空置链路的手段降低单节点局部路由重构造成的链路影响。同时提出对应的局部重构求解算法，优化方向不同于扩张型Benes网络的降低器件串扰，而是降低局部重构破坏的链路数与缩短大规模互连下局部路由的求解耗时。本文主要贡献如下。

（1）利用预留的空置链路代替既有数据链路在局部重构时被破坏，从而大幅减少单节点路由变更所破坏的既有链路数至平均2~4，使单次局部重构所需排空流量的链路数减少约98%。

（2）提出核心网络路由重整方法，使局部重构的节点同步成本整体可控且可掩盖，并提升完全不破坏额外既有链路的局部重构占比，使其高达72.4%。

（3）实现了百节点互连，对其进行单次局部重构求解的耗时也低至62 ns，接近Crossbar网络的水平，避免了局部重构求解导致的性能瓶颈。

1 局部重构求解原理与快速Benes网络

1.1 局部重构底层逻辑

既有全局重构方法[13-17]的求解结果仅给出一组开关状态可行解，但单一全局路由请求对应的开关状态解并不唯一。传统路由求解算法按Benes网络的构建特性，从外向里逐层求解，每一层将链路均匀分配给2个子网进行路由。交换这些链路分配的子网，仍能得到相同的外部路由结果，但开关状态将完全不同，各自对应一个可行解。8×8 Benes网络外围开关状态的两个可行解如图1所示。图1（a）与图1（b）分别展示了8×8 Benes网络在一次边缘开关状态求解过程中应对同一组路由请求的两个可行解，注意到每个对应开关的状态均相反。除了2×2 Benes网络状态会被输入条件严格限定为唯一解，其余任意规模的Benes网络，其最外层均满足上述性质。因此，以16×16 Benes网络为例，假设每层开关状态依赖关系仅为一个环路，则其共包含1个16×16边缘、2个8×8边缘和4个4×4边缘，对同一输入输出关系存在至少$2\times2^2\times2^4=128$个可行解。

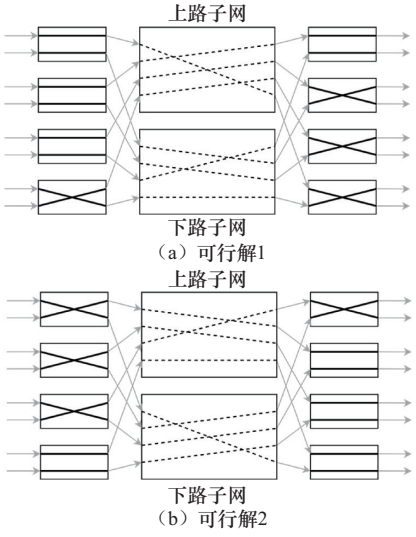


图1 8×8 Benes网络外围开关状态的两个可行解

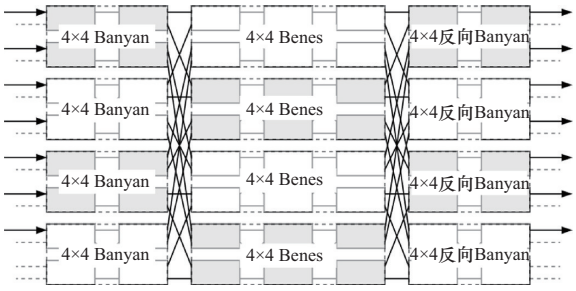
因此，若已知全局路由**R**，给定待求解路由状态**R'**，可从**R'**对应的可行解集合$\boldsymbol{S'}$中选出与当前可行解S最为相似的可行解$S'\in\boldsymbol{S'}$，通过变更不同部分实现局部重构，以减少需要变更的开关数，进而减少被破坏的既有链路数量。但Benes网络链路耦合度极高，若要求仅破坏与之相关的2路链路而不破坏其余的，则必然只能变更1个开关的状态，此时单节点路由变更的求解成功率会非常低：取$N=128$，则当前可能发生的单节点路由变更请求有$128\times127=16\,256$种，但对应规模的Benes网络，其开关单元总数仅为$64\times13=832$个，占比5.2%。若放宽单次局部重

构可变更状态的开关数且仍然希望尽可能不额外破坏既有链路，则可引入部分不承载数据流量的空置链路，在开关状态变更时代替既有链路被破坏，从而进一步降低被破坏的既有链路数，实现低成本局部重构。将M路链路划为加入空置链路后，Benes网络原有全局路由$\mathbf{R}$可分为有效链路的路由$\mathbf{R}_1$与空置链路的路由$\mathbf{R}_2$，$\mathbf{R}=\mathbf{R}_1\cup\mathbf{R}_2$。由于仅关心$\mathbf{R}_1$而不关心$\mathbf{R}_2$，此时全部$\mathbf{R}_1$相同但$\mathbf{R}_2$不同的$\mathbf{R}$均可视为相同，对应可行解数量进一步扩大为原有可行解数量的$M!$倍，使低成本局部重构占比大幅提升，M为空置链路数量。经过实验验证，在$N>100$时取$M=9/16N$可保证求解成功率与阵列规模膨胀率的平衡。

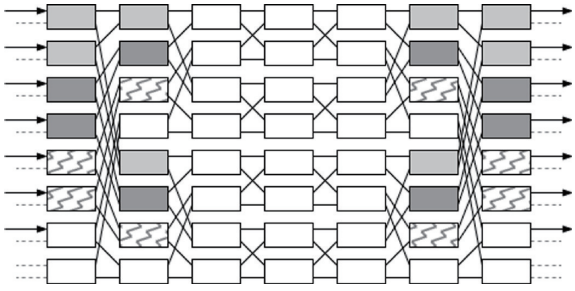
1.2　快速Benes网络与Banyan网络

加入9/16的空置链路并调整内部开关单元相对位置（但不改变开关单元连接方式）后的Benes网络称为快速Benes网络。快速Benes网络构建如图2所示。其可视为由若干Banyan网络[24]与反向Banyan网络以及更小规模的Benes网络以Clos连接方式[25]构成的3层交换网络，第i张Banyan网络的第j个内层输出端口连接第j张核心Benes网络的第i个输入端口，如图2（a）所示。由于开关单元连接方式未改变，快速Benes网络与Benes网络同构。图2（a）所示的16×16快速Benes网络与图2（b）所示的常规Benes网络结构等效，不同颜色的开关单元组成对应的Banyan网络。若Banyan网络的互连规模为N_E，核心Benes网络的互连规模为N_C，则快速Benes网络满足$N=N_\mathrm{E}\times N_\mathrm{C}$，记该快速Benes网络为$N_\mathrm{E}-N_\mathrm{C}$快速Benes网络。

快速Benes网络同样可以与Banyan网络一起，以Clos连接方式构建更大的快速Benes网络。例如，使用32张8×8 Banyan网络、32张反向8×8 Banyan网络与8张8-4快速Benes网络可构建嵌套网络，记为8-8-4快速Benes网络。互连规模N>100时，适宜使用此类5层网络进行建模与求解。



（a）16×16快速Benes网络的4-4视图



（b）与4-4视图同构的16×16 Benes网络

图2　快速Benes网络构建

2　局部重构求解方法

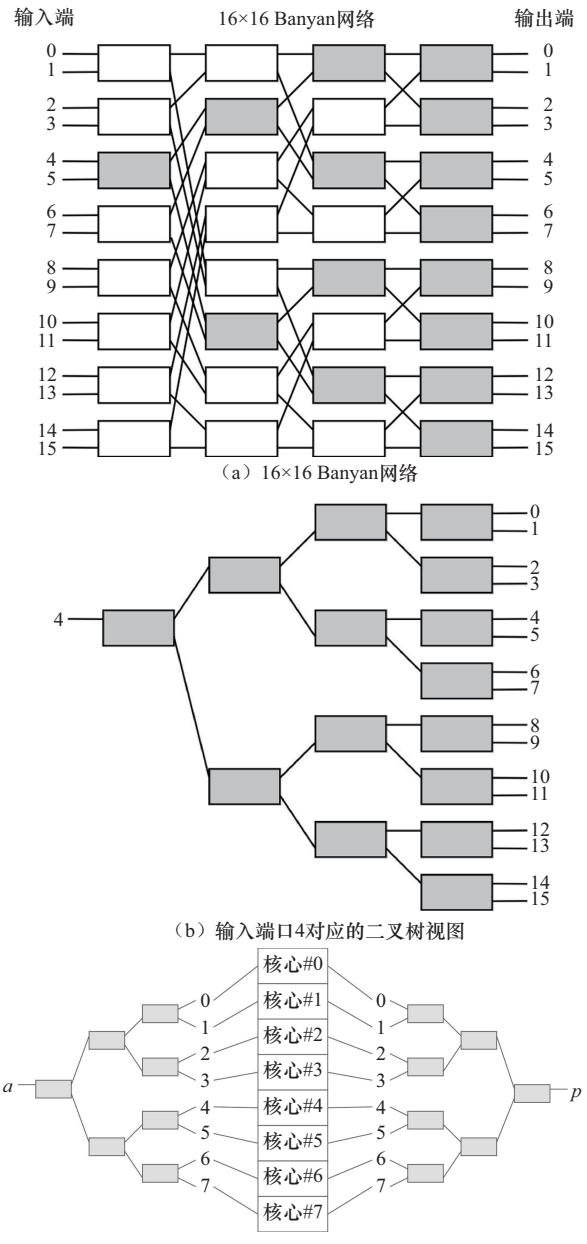
2.1　分级求解模型

由于采用Banyan网络替代2×2开关单元作为边缘网络，快速Benes网络的层数增长较Benes网络更为平缓。在$112\leqslant N\leqslant 448$时，快速Benes网络总能构建出$N_\mathrm{Eo}-N_\mathrm{Ei}-N_\mathrm{C}$三级结构，实现二级求解，其中$N_\mathrm{C}$始终为4，对应4×4 Benes网络，$N_\mathrm{Eo}$与$N_\mathrm{Ei}$则视情形不同取8或16。

$N\times N$ Banyan网络由2张规模减半的$N/2\times N/2$ Banyan网络和$N/2$个2×2开关单元构成，每个开关单元的两个输出端口分别连接其中一张Banyan网络。Banyan网络、二叉树视图与求解视图如图3所示。图3（a）为16×16 Banyan网络，其内部任意输入端口可达的开关单元构成一个二叉树网络，称其为Banyan二叉树视图，如图3（b）所示。对于单节点a的路由终点请求p，分别从输入端与输出端同时进行求解。由快速Benes网

络的连接方式可知，任意外围Banyan网络，其相同序号的输出端口总是连接到相同的核心Benes网络中，故提取a与p对应的Banyan二叉树视图及全部核心网络，构成求解视图如图3（c）所示。求解视图下，记待求解路由为(a,p)，选择合适的核心编号x并获取其对应核心网络的输入输出端口$a'=\lfloor a/N_{\mathrm{E}}\rfloor$与$p'=\lfloor p/N_{\mathrm{E}}\rfloor$。由于核心网络均为无阻塞网络，路由$(a',p')$必然可被成功构造，实现路由$(a,p)\rightarrow(a,x,p)\rightarrow(a,a',p',p)$的细化过程。过程中不难发现，若构建路由$(a,a')$与$(p',p)$时没有额外破坏既有链路，则后续构建路由$(a',p')$过程仅对应核心网络参与重构，最大破坏链路数仅为核心网络的互连规模N_{C}，进而实现单次重构破坏既有链路数量的降低与链路的可预测。在多层快速Benes网络结构中，核心网络仍可视为快速Benes网络并进行$N_{\mathrm{E}}-N_{\mathrm{C}}$拆分，从而进一步缩小局部重构的影响范围并降低重构破坏的既有链路数量。

对Banyan网络的路由求解分为2种类型：简单型（TYPE1）与复杂型（TYPE2），分别对应不破坏任何既有数据链路与仅破坏1条既有数据链路。在空置9/16N条链路时，外层Banyan网络仅用TYPE1与TYPE2求解即可覆盖全部情形，但对于内层Banyan网络，则额外多出一种求解类型，称之为路由重整操作，用于处理TYPE1与TYPE2均无法求解的极少数情形，并提升核心网络中后续的TYPE1与TYPE2路由成功率。当核心网络为4×4 Benes网络时，由于总求解空间仅有百余项，使用预先计算并查表的方式完成求解。根据各级路由求解结果，确定需要改变状态的开关单元、被破坏的既有链路编号，完成求解。具有3级$N_{\mathrm{Eo}}-N_{\mathrm{Ei}}-N_{\mathrm{C}}$结构的快速Benes网络局部重构求解整体流程如图4所示。



（a）16×16 Banyan网络

（b）输入端口4对应的二叉树视图

（c）快速Benes网络的求解视图

图3　Banyan网络、二叉树视图与求解视图

2.2　简单型Banyan路由求解

简单型（TYPE1）路由求解的目标是寻找核心网络x，使得a和p对应的Banyan网络在路由至x时均只破坏空置链路。以16×16 Banyan网络的二叉树视图为例，其内部有且仅有7条数据链路，并有9条空置链路，且每条数据链路均独占一个输入侧开关单元。简单型（TYPE1）局部重构原理如图5所示，为路由请求(a,p)对应的一次

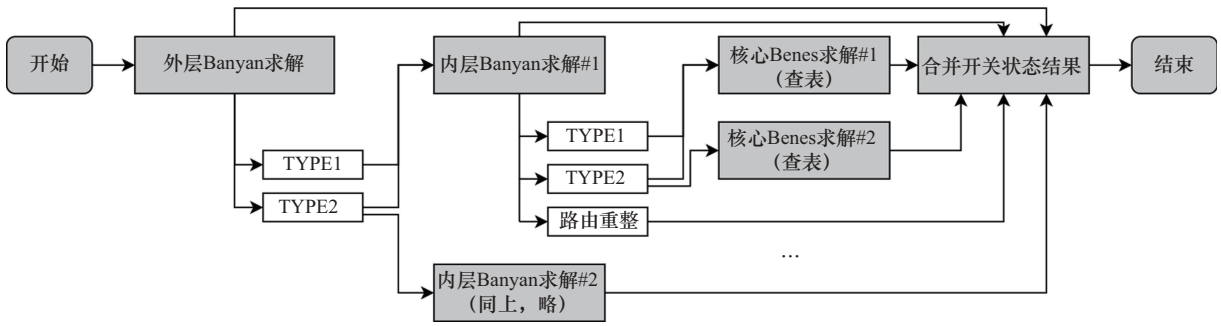


图 4　局部重构求解整体流程

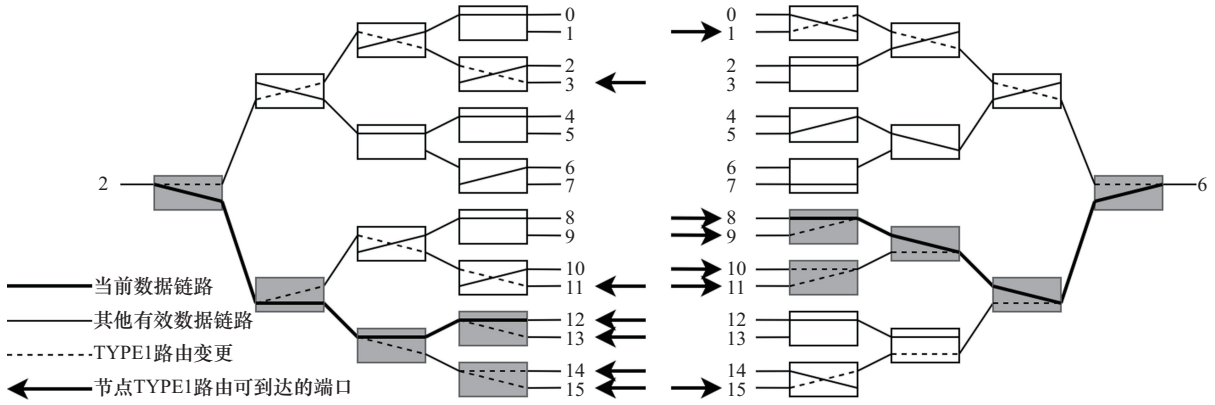


图5　简单型（TYPE1）局部重构原理

完整的TYPE1重构样例，不难发现，通过按需切换图5中左侧5个深色着色开关单元的开关状态，即可让输入链路路由至5条空置链路对应的出口且不破坏既有数据链路，加上链路原本的输出端口，即可让a路由至存在6张核心网络x_L满足端口a路由至x_L同时不破坏既有数据链路。同理，输出端口p对应的右侧反向Banyan网络，也存在6张核心网络x_R满足要求。定义a和p的可达性集合$A^L=\{x_L\}$与$A^R=\{x_R\}$，分别代表其可不破坏既有数据链路能够路由的核心网络编号x的集合，则图3中，$A^L=\{3, 11, 12, 13, 14, 15\}$，$A^R=\{1, 8, 9, 10, 11, 15\}$，令$A^M=A^L\cap A^R$，则$\forall x\in A^M$均可满足TYPE1重构要求，对应图3中的核心网络编号11和15。实际操作时，会选择内部有效链路数较少的x作为路由目标，以平衡核心网络的负载并提高后续局部重构成功率。

2.3　复杂型Banyan路由求解

在进行TYPE1求解时，并非任意的输入端口a与输出端口p都能同时实现路由至x且不破坏其他数据链路，具体表现为可达性集合$A^M=\varnothing$。TYPE1重构成功率取决于选取的边缘Banyan网络规模，边缘网络为8×8 Banyan时约为98.8%；边缘网络为16×16 Banyan时则降至95.5%。对于$A^M=\varnothing$情形，可通过破坏并重建1条既有数据链路(c,r)实现目标链路(a,x,p)的构建，称该类局部重构为复杂型局部重构求解（TYPE2求解），简记为TYPE2重构。

复杂型（TYPE2）求解的目标是在TYPE1失败时，破坏并重建1条既有数据链路(c,r)以实现目标链路(a,x,p)的构建。复杂型（TYPE2）局部重构原理如图6所示，如果选择改变开关单元1的状态将破坏既有有效链路#1，但可令输入端口a路由至TYPE1重构模式下无法到达的核心网络

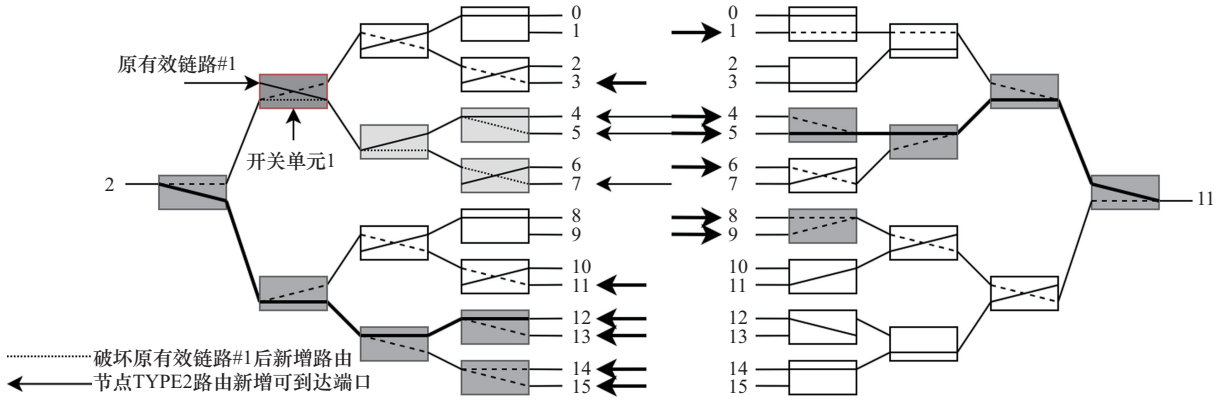


图6　复杂型（TYPE2）局部重构原理

{4, 5, 7}；改变其余开关节点的状态，破坏其他既有有效链路则可实现额外的可达路由终点。在TYPE1重构无法找到满足要求的$x \in A^M$时，通过此法可额外构造出满足要求的x，即图6中的{4, 5}，完成局部重构。简记被破坏的既有路由为(c,y,r)，则接下来需要在满足路由(a,x,p)的基础上完成路由(c,y',r)，$y' \neq y$。

以输入侧Banyan网络的TYPE2重构为例，在破坏既有链路(c,y,r)后，切换到c对应的Banyan二叉树视图，并改变链路(c,y,r)与原链路(a,q)交汇点（即图6中的开关单元1）的开关状态，执行对c的TYPE1求解得到A^{L2}。切换到输出端口r对应的Banyan二叉树视图，执行TYPE1求解得到A^{R2}。注意到亦可利用输出端口p进行TYPE2重构求解，故求解前开关状态的变更亦可发生在r的Banyan二叉树视图中，但不会同时发生在两者中。令$A^{M2}=A^{L2} \cap A^{R2}$，从中选取$\forall y' \in A^{M2}$且$y' \neq y$，即可完成路由$(c,y,r) \to (c,y',r)$。

TYPE2重构会破坏1条既有链路(c,r)，故额外引入的c与r对应另一张核心网络y'。TYPE2重构后续求解视图如图7所示，c对应输入端口编号#0，r对应输出端口编号#15，y'则由原先的4号核心网络变成12号或13号核心网络。对应核心网络y的输入输出端口c'与r'转化同理。TYPE2重构比TYPE1重构额外破坏1条既有链路，但在外围局部路由求解时可覆盖TYPE1重构未覆盖的剩余局部重构请求。

2.4　内层Banyan网络路由重整

对于内层Banyan网络，TYPE1与TYPE2求解方法依然有效。唯一不同的是，由于外层Banyan网络内每次成功的TYPE1和TYPE2重构均利用了空置链路，进入内层Banyan网络的路由请

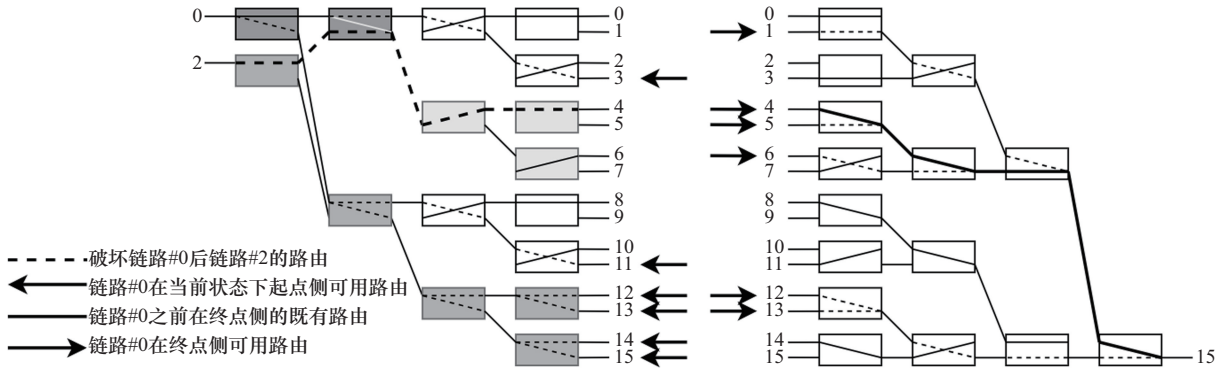


图7　TYPE2重构后续求解视图

求，其输入输出端口必然是之前空置的端口，导致可用空置链路占比降低。核心网络局部重构成功率下降原因如图 8 所示，为外围网络选择 11 号链路/子网进行路由后，在 11 号子网内 TYPE1 重构仅能到达 12 号与 15 号子网，可用 TYPE1 路由终点数少于外围网络。如此，导致 TYPE1 局部重构成功率下降，且 TYPE2 局部重构存在约 0.2% 的失败率。

本文提出一种全局路由重整算法用以解决内层 Banyan 网络局部重构失败问题。该算法基于 Benes 网络路由求解算法改进得到，可将内层 Banyan 网络所在的快速 Benes 网络进行全局路由重构，并对其路由进行压缩，将尽可能多的链路通过其内部的一个 Benes 子网进行路由，从而空出另一个子网用于局部重构。经过路由重整后，空出的子网既扩充了可达性集合 A 的大小，也增加了两侧 Banyan 网络 TYPE1 重构交集 A^M 的大小，能够在短期内大幅提高后续 TYPE1 重构的成功率至 100%。

算法基于文献[17]，按递归方式从外向里逐层求解，共分为 3 步：染色、边缘开关状态求解、确定子网路由请求。染色阶段，和文献[17]描述的方法一样，利用约束条件递推全部输入端口的染色状态。不同于常规 Benes 求解，在本文求解上下文条件中，并不关心当前标记为空置的输入端口具体连接到哪个输出端口，仅需要保证其连接到的输出端口也标记为空置即可。同时，“将尽可能多的链路通过一张 Benes 子网构建”这一需求成为新的约束条件，使得染色过程与文献[15]不同。具体如算法 1 I_COLORING 所示。

算法 1　I_COLORING

输入　有效链路的全量映射关系 $\mathbf{M}=\left\{(a,p)\middle|\begin{matrix}a\text{为输入端口编号}\\p\text{为输出端口编号}\end{matrix}\right\}$

输出　输入端口染色状态 $C^i=\left\{C_n^i\middle|C_n^i\in\{\text{"}U\text{"},\text{"}L\text{"}\}\right\}$

let $\mathbf{M}^p(i)=\begin{cases}q,\ i\text{对应有效链路}\left|(i,q)\in\mathbf{M}\right.\\-1,\ i\text{对应空置链路或}i=-1\end{cases}$

let $\mathbf{M}^n(i)=\begin{cases}q,\ i\text{对应有效链路}\left|(q,i)\in\mathbf{M}\right.\\-1,\ i\text{对应空置链路或}i=-1\end{cases}$

let $\mathrm{NB}(i)=M^n(\mathrm{Mutex}(i))$//注：Mutex 引用文献[15]内的 Mutex 算法

let $c\leftarrow 0$

let $s\in\{\text{"}U\text{"},\text{"}L\text{"},\varnothing\}\leftarrow\varnothing$

for i in 0 to $N-1$ do:

　if $s\neq\text{"}U\text{"}$ do:

　　$C_c^i\leftarrow\text{"}U\text{"}$, $s\leftarrow\text{"}U\text{"}$

　else do:

　　$C_c^i\leftarrow\text{"}U\text{"}$, $s\leftarrow\text{"}U\text{"}$

　end if

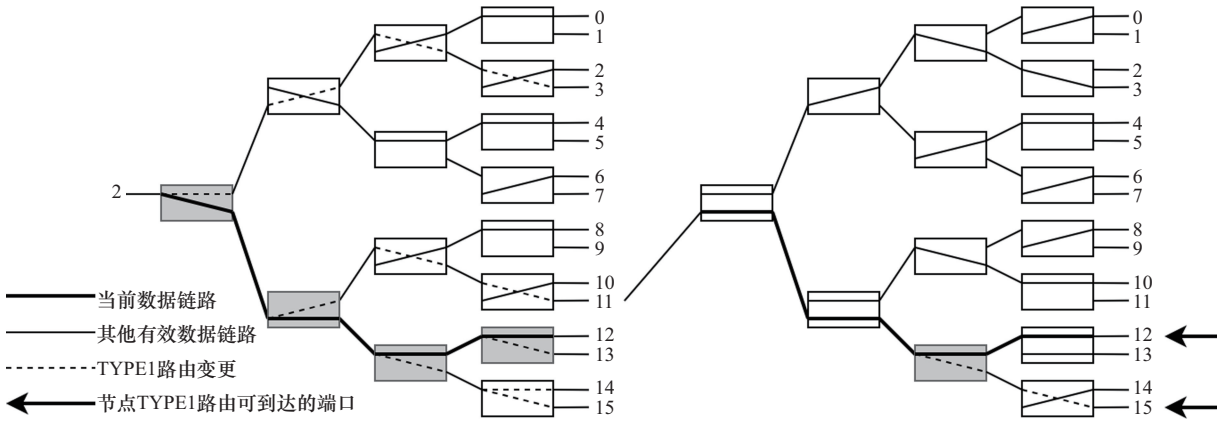


图 8　核心网络局部重构成功率下降原因

$c \leftarrow \mathrm{Mutex}(c)$ if c 是奇数 else $\mathrm{NB}(c)$

if $c=-1$ do:

$c \leftarrow$ 任意t满足C_t^i尚未被赋值，$s \leftarrow \varnothing$

end if

end for

注意到**M**仅包含有效链路的映射关系而不包含空置链路的，因此原文献[15]中对输出端口的染色状态求解亦需要替换，如算法2 O_COLORING所示。

算法2 O_COLORING

输入 有效链路的全量映射关系 $\mathbf{M}=\left\{(a,p)\middle|\begin{matrix}a\text{为输入端口编号}\\p\text{为输出端口编号}\end{matrix}\right\}$

输入端口染色状态 $C^i=\left\{C_n^i|C_n^i\in\{"U","L"\}\right\}$

输出 输出端口染色状态 $C^o=\left\{C_n^o|C_n^o\in\{"U","L"\}\right\}$

for (a,p) in **M** do:

$C_p^o \leftarrow C_a^i$

end for

for i in 0 to $N-1$ do:

if $C_{i\times 2}^o=\varnothing \wedge C_{i\times 2+1}^o=\varnothing$ do:

$C_{i\times 2}^o \leftarrow$ "U"，$C_{i\times 2+1}^o \leftarrow$ "L"

else if $C_{i\times 2}^o=\varnothing$ do:

$$C_{i\times 2+1}^o \leftarrow \begin{cases}"L",\ C_{i\times 2}^o="U"\\"U",\ C_{i\times 2}^o="L"\end{cases}$$

else if $C_{i\times 2+1}^o=\varnothing$ do:

$$C_{i\times 2}^o \leftarrow \begin{cases}"L",\ C_{i\times 2+1}^o="U"\\"U",\ C_{i\times 2+1}^o="L"\end{cases}$$

end if

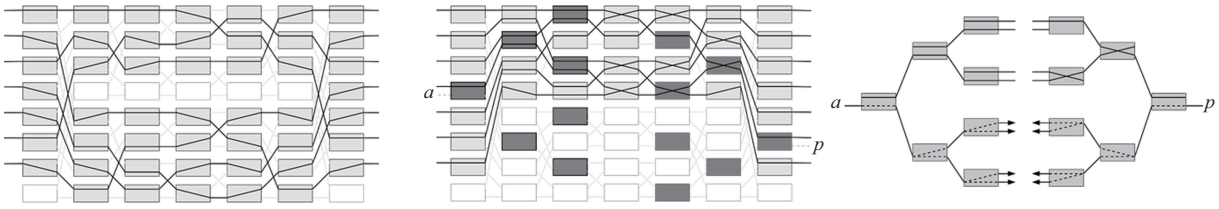
end for

路由重整原理如图9所示。利用算法1 I_COLORING与算法2 O_COLORING分别替换文献[17]中对输入端口和输出端口的染色算法，即可实现链路不满配时Benes网络的全局路由求解，并保证绝大多数链路仅经过其中一个子网进行路由，从而空出另一个，如图9（a）与图9（b）所示。经路由重整后的边缘Banyan网络，其支持TYPE1路由的终点数将达到最高，且求解视图下路由终点的重合度也极高。图9（c）为图9（b）中被深色着色的开关为输入端口a与输出端口p对应的求解视图，注意到其输入端口和输出端口均对应空置链路，但依然能够实现经过下方4张核心网络x执行TYPE1重构，可用核心网络占比高达50%。路由重整算法成功解决了核心网络局部路由难度较高的问题，并且一次路由重整可确保核心网络在较长时间内均能完成极低节点同步成本的局部重构。

2.5 网络的容错与故障恢复

快速Benes网络支持一定程度的简单容错。在进行TYPE1或TYPE2路由求解时，若已知其对应的Banyan网络内存在故障开关单元，则在可达性集合求解过程中，将经过该故障开关的对应路由设置为失效即可。若以此法设置路由，则对应的可达性集合中，不会包含经过故障开关的路由，即实现了路由求解绕过故障开关单元。

快速Benes网络的容错能力是有限的。每存在一个完全故障（两路链路均无法工作）的开关单元，都会导致边缘Banyan网络求解时，端口可达的核心网络数量减少，从而降低TYPE1与



（a）路由重整前　（b）路由重整后　（c）路由重整后的求解视图

图9 路由重整原理

TYPE2 求解的成功率。当触发核心网络路由重整时，对应核心网络的故障开关单元也会使路由重整算法失效，即算法不能保证全部接入的有效链路能够同时实现路由请求而不阻塞。

2.6　求解方法复杂度分析

Banyan 网络的 TYPE1 与 TYPE2 求解均基于 Banyan 构建规律实现，其并行求解时间复杂度为 $O(M)$，其中 M 为 Banyan 网络内单一信号经过的开关数量。具体而言，对于 TYPE1 求解，对每个可达终点的判断需要访问相应 Banyan 网络每层交换层级内的一个开关单元，且只读访问没有数据依赖，故并行时间复杂度为 $O(M)$；注意到因全部判断逻辑不涉及数据依赖，故可完全并行处理，并行时间复杂度保持不变仍为 $O(M)$。对于 TYPE2 求解，尽管新增破坏既有链路的操作会引入其余链路的判断，其时间复杂度也为 $O(M)$，但同样由于判断逻辑不涉及数据依赖，并行处理后仅增加时间复杂度为 $O(1)$的比较操作；对于 TYPE2 求解其后续对被破坏的既有链路重新路由，其并行求解复杂度同样为 $O(M)$，因此快速 Benes 网络的每级并行求解复杂度为 $O(M)$。内层路由重整的并行时间复杂度与 Benes 网络一致，为 $O(N\log N)$，并行求解复杂度参考文献[17]，为 $O(N)$，N 为该层快速 Benes 网络互连规模。由于没有数据依赖，在同一级的 Banyan 与反向 Banyan 网络可并行 TYPE1 求解。同样，同一级 Banyan 网络内 TYPE1 求解、TYPE2 求解与路由重整也可并行执行，最坏时间复杂度对应路由重整的时间复杂度 $O(N)$，平均时间复杂度则取决于 TYPE1、TYPE2 与路由重整的触发占比。

由于 $M=\mathrm{lb}(N_\mathrm{B})$，$N_\mathrm{B}$ 为 Banyan 网络的互连规模，在仅有 TYPE1 与 TYPE2 求解时，快速 Benes 网络的并行求解复杂度为 $O(\log N)$，N 为快速 Benes 网络的互连规模。在路由重整介入时，求解复杂度会上升为 $O(N_\mathrm{C}\log N_\mathrm{C})$，其中 N_C 为对应路由重整的快速 Benes 网络互连规模。

空间复杂度方面，由于求解采用并行处理，TYPE1 并行求解队列数每次为 2^M，M 为 Banyan 网络内单一信号经过的开关数量；TYPE2 并行求解则进一步对每个输入信号建立 2^M 条并行求解队列，总求解队列数为 $2^M\times2^M=2^{2M}$。因此，TYPE1 求解空间复杂度为 $O(2^M)$，TYPE2 求解空间复杂度为 $O(2^{2M})$。路由重整的空间复杂度则可参考传统 Benes 网络求解算法，为 $O(N)$，N 为对应 Benes 网络互连规模；若将传统 Benes 网络每层求解过程展开，则空间复杂度上升至 $O(N\log N)$。

3　硬件求解加速器实现

3.1　整体求解架构与功能单元实现

快速 Benes 网络硬件求解加速器采用嵌套架构分层求解。每层求解器仅求解该层对应的 Banyan/反向 Banyan 网络局部重构，确定路由的核心网络后，将后续求解工作交给下一层求解器。$112\leqslant N\leqslant448$ 时，求解器共分 3 层，快速 Benes 硬件求解加速器结构如图 10 所示。每层由一个可同时求解 TYPE1 与 TYPE2 的 Banyan 求解单元、一个链路复用 TYPE1 求解单元和与之镜像的 2 个反向 Banyan 求解单元构成，同时设有状态存储单元用于存储本层全部 Banyan/反向 Banyan 网络的开关状态，并按需选择其中一个 Banyan 和反向 Banyan 载入对应求解单元进行求解。链路复用 TYPE1 求解单元用于处理 TYPE2 路由时额外被破坏既有链路的重新路由。由于出现 TYPE2 路由时需要处理 2 条链路的路由，对应 2 个不同的子网，需要准备 2 个下一层求解单元。由第 2.1 节和第 2.4 节可知，外层 Banyan 网络求解可被 TYPE1 与 TYPE2 求解完全覆盖，但内层 Banyan 网络求解将存在 TYPE1 与 TYPE2 均覆盖不到的求解缺口，故增设路由重整单元用于处理这部分求解情形的核心网络路由重整需求。在求解器之外设有相应规模的快速 Benes 仿真网络，用于同

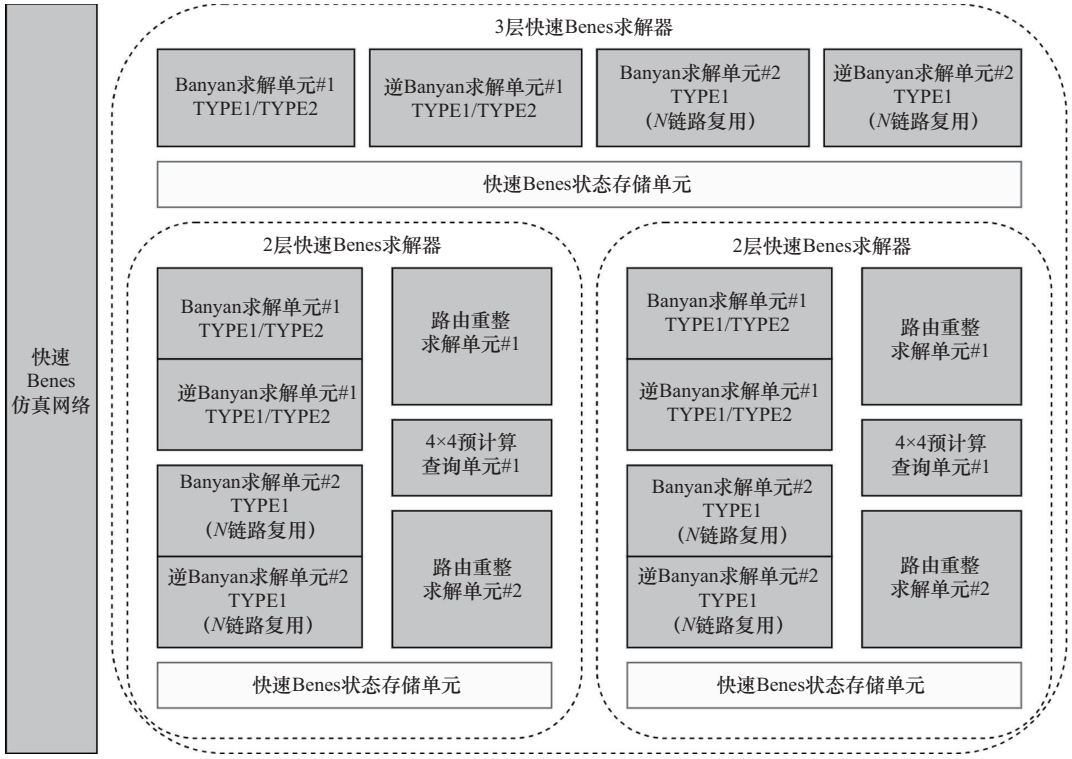
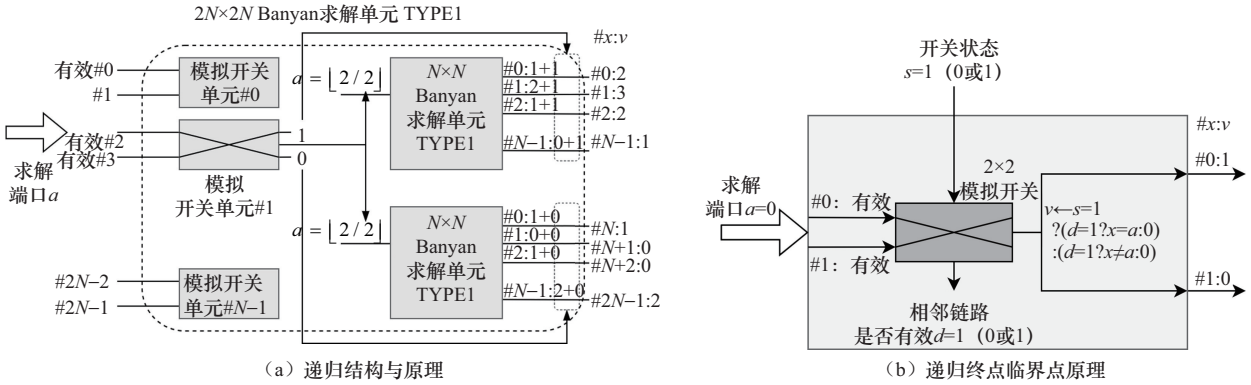


图10　快速Benes硬件求解加速器结构

步当前待求解网络的开关与路由状态。

TYPE1求解采用资源复用的方式，仅使用1个$N_{Eo}\times N_{Eo}$ Banyan求解单元、1个反向$N_{Eo}\times N_{Eo}$ Banyan求解单元和1组核心快速Benes求解器，在求解时即时传入对应Banyan网络的基本信息进行求解，以大幅降低硬件面积开销。使用递归方式构建Banyan求解单元，给定当前求解单元所需求解的链路编号，定位所在的开关单元，判断该单元内另一条数据链路是否有效，并结合两个次级Banyan求解单元的求解结果，得出本求解单元在给定求解端口a时，其路由至全部输出端口须破坏的既有链路数集合$\{(\#x{:}v)\}$。TYPE1 Banyan求解单元原理如图11所示。在图11（a）中，次级Banyan求解单元的求解端口$a'=\lfloor a/2\rfloor$。从求解端口a路由至上路Banyan网络需要破坏既有链路#3，故代价为1，而端口a原本就路由至下路Banyan网络，因此对应代价为0。将之叠加到对应次级Banyan求解单元的求解结果，即可得

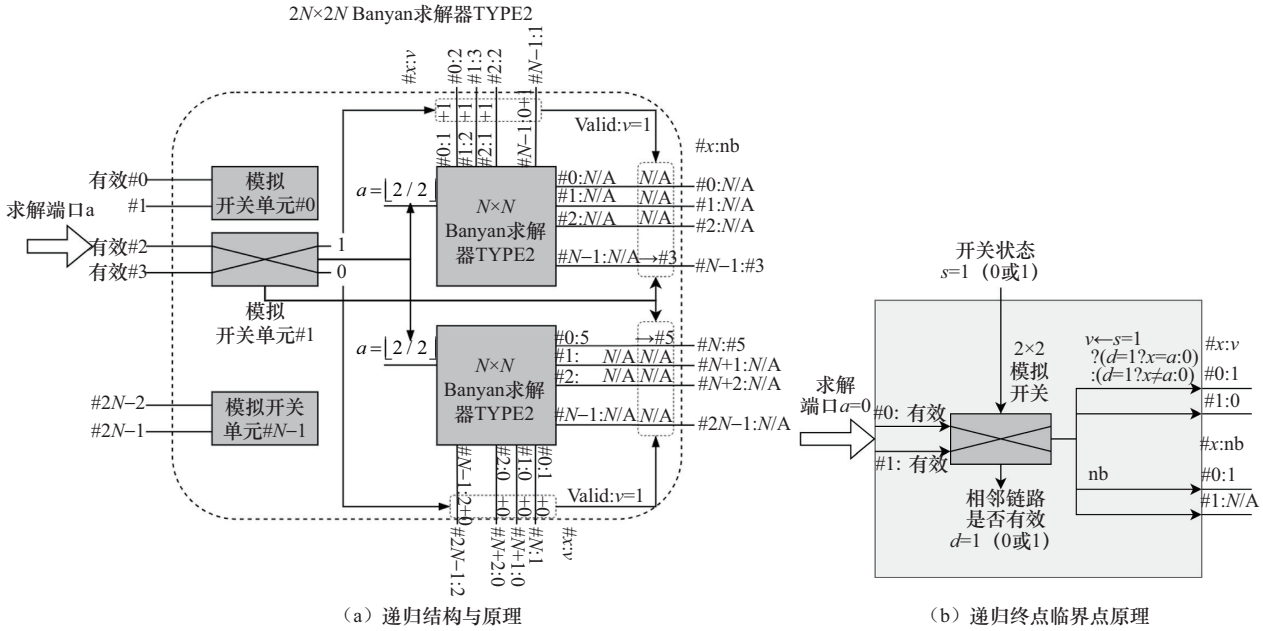


（a）递归结构与原理　　（b）递归终点临界点原理

图11　TYPE1 Banyan求解单元原理

到本级 TYPE1 求解的初步结果$\{\#x{:}v\}$，其中代价为 0 的输出端口即可选 TYPE1 重构的路由目标，即$\forall(\#x{:}v)$所有满足$v=0$的x都可以作为TYPE1路由目标，均可实现TYPE1局部重构。TYPE1 Banyan求解单元的递归终点临界点原理如图11（b）所示，通过对仅有的几种情形进行预先枚举完成求解。Banyan求解单元可在单周期内并行求解给定输入端口a时全部$\{(\#x{:}v)\}$的集合。完整的内层TYPE1求解耗时2周期，外层TYPE1求解则多1周期，为3周期，用于满足时序约束。

TYPE2 求解复用 TYPE1 求解的基础设施，并额外增设 1 个$N_{\mathrm{Eo}}\times N_{\mathrm{Eo}}$ Banyan求解单元和反向$N_{\mathrm{Eo}}\times N_{\mathrm{Eo}}$ Banyan求解单元用于处理被破坏的无关链路的重新路由$(c,y,r)\to(c,y',r)$，能够实现8条或16条重新路由的并行求解。TYPE2 Banyan求解单元原理如图12所示。图12（a）为TYPE2 Banyan求解单元的构造与工作原理，相比于TYPE1求解，TYPE2 Banyan求解单元需要额外求解在代价v恰好为1时，路由至目标输出端口会破坏的链路编号（即图12中nb值）。被破坏的链路必然在某一开关单元内与当前求解端口a对应的链路交汇，因此分为两种情况：（1）在本级Banyan网络的最左侧开关单元内，本级路由代价为1且后续路由代价为0，对应图12（a）中的$\#(N-1){:}\,\#3$；（2）在次级Banyan网络内，本级路由代价为0且后续路由代价为1，对应图12（b）中的$\#N{:}\,\#5$。对于前者，若后续路由至次级Banyan网络的代价恰好为1，且在对应次级Banyan网络内路由代价为0，则nb=$\#b$，$\#b$为与a相邻的链路号，对应图12(a)中的$\#(N-1){:}\,\#3$。若在次级Banyan网络内路由代价恰好为1，且路由至对应次级Banyan网络的代价为0，则nb=$\#b'$，$\#b'$为次级Banyan网络求解得到的nb值，对应图12（a）中的$\#N{:}\,\#5$。在定位可用路由终点与对应nb值（图12（a）中$\#(N-1){:}\,\#3$与$\#N{:}\,\#5$）后，需要查找映射关系表补全路由(nb,r)，并需要通过额外的TYPE1求解单元进行$(\mathrm{nb},y,r)\to(\mathrm{nb},y',r)$重新路由，对应增设的求解单元。TYPE2 Banyan求解单元的递归终点临界点原理如图12（b）所示，同样通过对仅有的几种情形进行预先枚举完成求解。TYPE2与TYPE1同步进行求解，求解耗时6周期，多出的周期用



（a）递归结构与原理　　（b）递归终点临界点原理

图12　TYPE2 Banyan求解器原理

于处理额外的$(\mathrm{nb},y,r)\rightarrow(\mathrm{nb},y',r)$重新路由。

当前开关阵列的路由映射关系表可通过对当前快速Benes网络的行为模拟快速得出。在加速器内维护一张与待求解快速Benes网络开关状态完全同步的仿真网络，令仿真输入信号为对应的端口编号，即可通过在输出端口侧读取信号内容的方式获得输入–输出映射关系表，输入–输出映射关系求解原理如图13所示。

路由重整求解单元基于文献[17]中的流水线求解加速器实现，并以算法1 I_COLORING和算法2 O_COLORING代替其原有算法。为了配合其余工作模块，将文献[17]中的每四级流水线合并为一级，故在求解64×64核心网络时，消耗20 min周期即可完成求解。求解32×32核心网络则只需10 min周期。

求解器工作时，对于外层Banyan网络总是同时进行TYPE1与TYPE2求解，对于内层Banyan网络则同时进行TYPE1、TYPE2求解与路由重整，以确保即便出现最坏情况也不损失求解时间。

3.2 求解器的FPGA验证

使用Xilinx Virtex 7 690T FPGA进行百节点互连规模下硬件求解加速器硬件验证。整张快速Benes网络使用三级$N_{\mathrm{Eo}}-N_{\mathrm{Ei}}-N_{\mathrm{C}}$结构，其中256×256快速Benes网络采用8–8–4结构，512×512快速Benes网络采用16–8–4结构，1 024×1 024快速Benes网络采用16–16–4结构。分别对相应规模的硬件求解加速器进行静态分析，FPGA资源消耗、求解时钟频率与求解耗时见表1，其中查找表（lookup table，LUT）用于实现组合逻辑，触发器（flip-flop，FF）用于实现时序逻辑，LUT和FF为现场可编程门阵列（field-programmable gate array，FPGA）的主要硬件资源。求解器硬件资源消耗量总体和互连规模N呈线性关系，求解耗时略有增加但远低于互连规模增加的速度。

表1 FPGA资源消耗、求解时钟频率与求解耗时

Benes互连规模	256×256	512×512	1 024×1 024
有效链路数	112	224	448
LUT	29 514	82 240	228 843
FF	12 431	28 612	63 823
频率/MHz	150	118	75
单次局部重构求解平均耗时/ns	61.67	78.39	123.33

设计性能评估实验以验证求解器的实际性能。在最理想状态下，求解器应对任意到来的局部路由变更请求均有较好支持，故设计在求解器

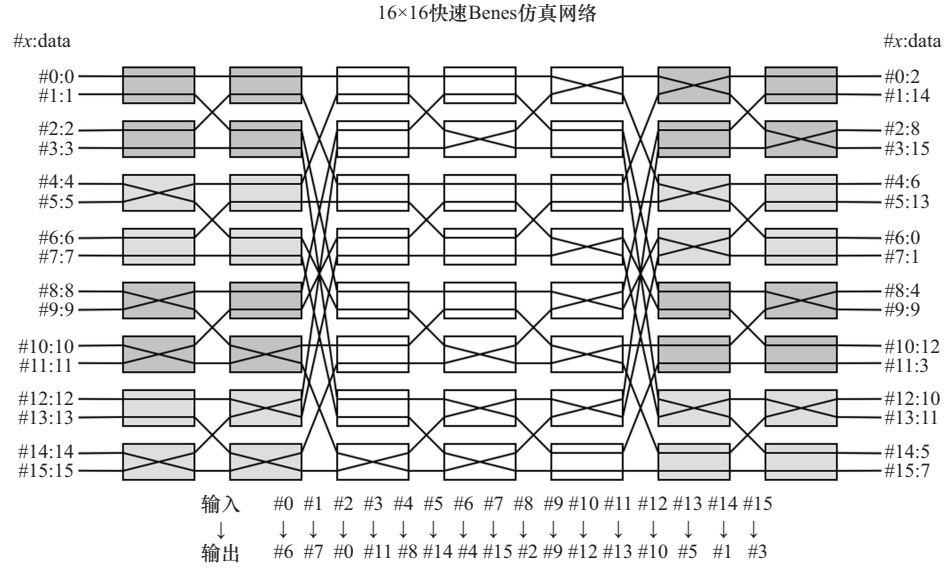


图13 输入–输出映射关系求解原理

对应当前网络状态下随机选择变更 1 个节点的路由终点作为当前局部路由请求，使用求解器进行局部重构求解并应用于当前网络（执行一次网络重构操作）。使用支持 448 条有效链路的求解器，重复上述实验百万次并统计每次重构的具体信息，得到各类求解类型所需时钟周期数与占比统计见表 2。TYPE2 求解后内层需要并行求解两个子网，故求解所需周期数由最为耗时的一路内层求解决定。外层和内层网络均适用 TYPE1 求解时，总时钟周期数最短为 8，其中外层 TYPE1 求解消耗 3 周期，内层消耗 2 周期，核心查表消耗 1 周期，汇总消耗 2 周期。每触发一层的 TYPE2 求解，则外层需要消耗 7 周期，内层消耗 6 周期；触发内层路由重整则消耗 10 周期，但内层路由重整也会同时完成核心 4×4 Benes 网络的路由求解，故省下 1 周期，实际消耗 9 周期。若外层触发 TYPE2 路由，意味着将存在 2 路并行的内层求解，此时内层求解耗时将取决于这 2 路求解中更耗时的那一路，故分类中将其分为：2 路均为 TYPE1、存在至少 1 路 TYPE2 但无路由重整、存在至少 1 路路由重整。求解所需时钟周期数平均值为 9.28，且超过 95% 的路由请求对应的求解时钟周期数≤12。

表 2　各类求解类型所需时钟周期数与占比统计

外层 Banyan 求解类型	内层 Banyan 求解类型	求解所需时钟周期数	占总求解请求比例
TYPE1	TYPE1	8	72.4%
	TYPE2	12	17.8%
	路由重整	17	0.6%
TYPE2	均为 TYPE1	12	5.7%
	存在 TYPE2，无路由重整	16	3.4%
	存在路由重整	21	0.1%
备注	—	平均值 9.28	合计 100%

实验也统计了每次局部重构操作造成的既有链路破坏数与占比。有效链路数为 448 时，单次局部重构额外破坏链路数累积频率如图 14 所示。其中，单次局部重构额外破坏有效链路的平均值为 1.59，完全不额外破坏有效链路的局部重构占比高达 61.2%，有 96.7% 的局部重构操作额外破坏链路数<8，说明快速 Benes 网络在绝大多数情况下，单一接入节点路由变更引发的局部重构平均只需要不到 4 个接入节点的流量排空操作。

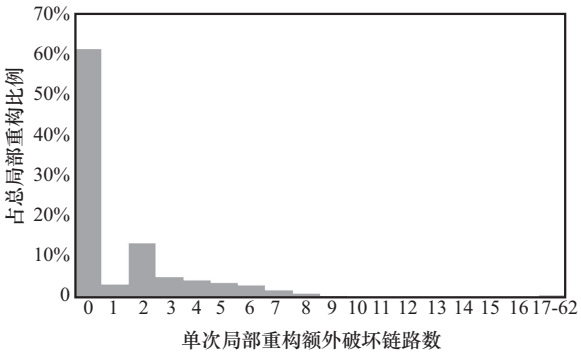


图 14　单次局部重构额外破坏链路数累积频率

4　快速 Benes 网络评估与应用探索

4.1　求解器性能评估

由于尚无 Benes 网络局部重构相关算法与加速器研究成果，选择当前最佳的 Benes 网络硬件求解加速器作为对比对象，并加入典型的严格无阻塞网络 Crossbar 作为性能对比的上限。选择进行比较的互连规模为 8×8 至 448×448，没有选择更大的互连规模，原因在于：互连规模超过 16×16 时，基于半导体光开关的 Crossbar 阵列就已经难以物理实现；以往 Benes 网络缺乏低成本局部重构方法的缺点也导致研究者鲜少关注超过 100×100 互连规模的 Benes 阵列制造；基于当前工艺水平和光模块插损容忍程度的反推，未来数年内可实现 448×448 互连规模的快速 Benes 开关阵列，但进一步扩大互连规模则充满困难。当互连规模大于 448×448 时，求解器的设计压力也会大幅上升：快速 Benes 网络将以四级网络构建，求解更加复杂，求解器规模也会空前庞大，导致求解性能快速下降。

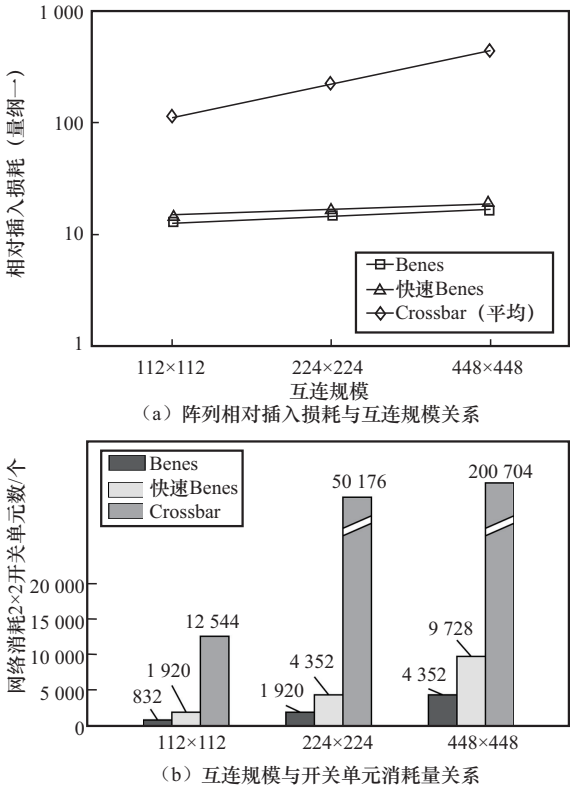
本文实现的硬件加速器与现有Benes网络硬件加速器和Crossbar硬件加速器的路由求解耗时对比见表3，其中Crossbar硬件加速器为自行设计的FPGA加速器。在互连规模为64×64以下时，本文参与对比的硬件加速器对应N_E-N_C二级快速Benes网络而非三级。可以发现，Crossbar加速器的求解效率几乎不随互连规模的提升而下降，极为理想，但受限于阵列自身的物理限制而极难实现$N>30$的互连规模；Benes加速器的求解耗时则随互连规模N增长而线性增长；文献[15]提出的Benes加速器实现了更优的对数增长。然而上述3篇研究成果均未探讨在大于128×128互连规模下的求解效率，但可预见的是，在互连规模超过百节点时，上述研究成果将面临巨大的求解困难。当$112\leqslant N\leqslant 448$时，本文实现的快速Benes硬件加速器仅需进行3层求解：外层Banyan、内层Banyan和核心Benes网络，因此求解所需时钟周期数不变，仅时钟频率因求解规模扩大而有所降低，性能优势尤为突出，局部路由求解速度具有较高的实际意义，可有效提升网络的灵活性。

表3　路由求解耗时对比

互连规模	本文	Crossbar	文献[13]	文献[14]	文献[15]
8×8	—	2~3	100	96	12
16×16	—	2~3	150	256	26
32×32/28×28	20	—	200	960	72
64×64/56×56	24	—	250	3 072	—
128×128/112×112	62	—	360	—	—
224×224	78	—	—	—	—
448×448	123	—	—	—	—

4.2　快速Benes网络成本评估

网络的插入损耗与开关阵列完成交换所需的开关单元级数线性相关，而构建成本则与构建开关阵列所需的2×2开关单元数近似线性相关。不难发现，快速Benes网络由于空置了9/16N路链路，其最大互连规模为额定互连规模的2倍有余，而在最大互连规模下，其又能与Benes网络相互转化，故额定互连规模为N的快速Benes网络，其插入损耗与构建成本均和$M\times M$ Benes网络相似，其中$\mathrm{lb}(2N)+1=\mathrm{lb}M$。互连规模为$N$时，快速Benes网络需经过$2\mathrm{lb}N+1$个开关单元完成路由，仅比Benes网络多经过2个开关，远远好于Crossbar网络的平均N个/最大（$2N-1$）个。快速Benes网络成本对比如图15所示。当使用相同规格的2×2开关单元构建阵列时，不同交换网络的相对插入损耗对比如图15（a）所示。可以发现，快速Benes网络的相对插入损耗仅略高于相同互连规模的Benes网络，而优于Crossbar网络超过一个数量级。



（a）阵列相对插入损耗与互连规模关系

（b）互连规模与开关单元消耗量关系

图15　快速Benes网络成本对比

互连规模与开关单元消耗量关系如图15（b）所示。不难发现相同互连规模下，快速Benes网络的开关单元需求量仅约为Benes网络的2.5倍，

比Crossbar网络少一个数量级。更低的开关单元需求量既能直接降低网络的一次性部署成本，亦能在相同工艺水平下提升总体良率，进一步降低生产成本。

4.3　快速Benes网络性能评估

快速Benes网络为电路交换网络，其网络性能评价指标主要有两点。

（1）并发交换能力：网络内支持起终点不冲突链路同时工作的最大值。由于快速Benes网络具有可重排无阻塞特性，其并发交换能力在任意时刻均等于其额定互连规模，为最大值。

（2）信号占空比：网络存在重构操作，重构期间无法支持数据传输；同时为避免重构期间数据丢失或损坏，重构前需要排空对应链路的流量，故其对应链路在该重构前一段时间和重构期间处于空闲状态。称空闲状态时间占总通信时间的比例为占空比，由于网络总能以最大带宽传输数据，链路占空比越低，则网络吞吐量越高，性能也越好。

快速Benes网络为电路交换网络，故提出占空比概念用于衡量其性能与吞吐量指标，其定义为：链路空闲状态时间占总通信时间的比例。占空比与网络内数据流量类型、网络实际重构频率、网络重构的副作用均高度相关。忽略计算耗时，则：

$$D_C = \frac{T_W}{T_I + T_W} \tag{1}$$

其中，D_C为占空比，T_W为传输中断等待耗时，T_I为数据传输理论耗时。

当计算任务相同时，数据传输理论耗时T_I也相同，此时可用传输中断等待耗时T_W间接比较D_C大小。显然，T_W越小，则D_C越小。T_W可量化为$n\times(T_R+T_T)$，其中n为单次重构受影响的无关链路数，T_R为开关元件重构耗时，单一链路T_T为流量排空耗时。使用相同的光发射器与接收器接入光开关阵列时，T_T相同；使用相同的开关单元构建开关阵列时，T_R相同。因此，降低D_C需要更小的n值。不同网络单节点局部重构破坏的链路数分布如图16所示。不难发现，Crossbar网络拥有最低n值为2，快速Benes网络在112×112、224×224、448×448互连规模下的平均n值分别为2.63、3.31、3.59，离群值最高也仅为21、32、62，且占比低于1%。相比之下，Benes网络的n值则分布极广，从最低的2到最高与互连规模持平，且50%以上的n值大于$0.5N$，平均值也高达$0.71N$。

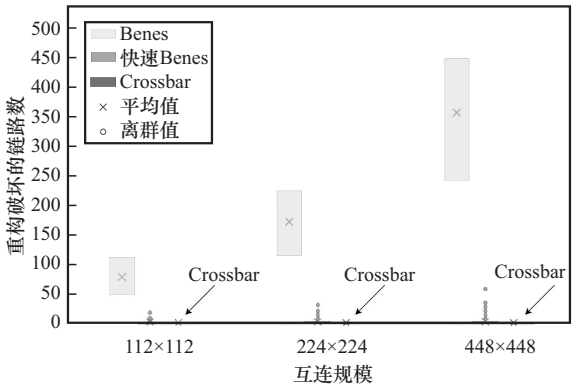


图16　不同网络单节点局部重构破坏的链路数分布

在互连规模为224×224、单一接入终端速率为400 Gbit/s、单一接入终端平均每发送10 000 Byte会变动一次通信目标时，T_I=200 ns；按点对点光路平均10 m长度计算，需要约50 ns完成信号传播，即$T_T\approx$50 ns；使用2×2快速光开关单元构建阵列，则T_R=3 ns。此时网络单一链路受重构影响次数平均为n，单链路占空比=$n\times(T_T+T_R)/(T_I+n\times(T_T+T_R))$。不同交换网络在特定工况下占空比与吞吐量对比见表4。

快速Benes网络在上述工况下，单链路吞吐量仍能达到Crossbar网络的81.5%，而传统Benes

表4　不同交换网络在特定工况下占空比与吞吐量对比

网络构型	占空比	单链路吞吐量/(Gbit·s^{-1})	单链路吞吐量（归一化）
Crossbar	0.346	261.6	1
快速Benes	0.467	213.2	0.815
Benes	0.977	9.2	0.035

网络仅有Crossbar网络的3.5%，快速Benes网络相比传统Benes网络提升了约23倍的吞吐量。注意到传统Benes网络单一链路受重构影响次数随互连规模N的扩大而增加，故互连规模越大，快速Benes网络对比传统Benes网络的吞吐量优势也越大。

4.4 快速Benes网络应用探索

Benes网络依靠其可重排无阻塞特性和较低的构建成本，特别适宜作为数据中心网络中负责高吞吐量交换的补充网络。但Benes网络高昂的重构代价导致其在实际使用场景中倾向于合并局部重构请求、尽可能减少重构频率或在运行时不重构，使其在路由目标频繁变化的通信场景下时延表现糟糕。快速Benes网络继承了Benes网络的可重排无阻塞特性，同时保证了局部重构的低成本，故其支持频繁的局部重构操作而不会大幅损失吞吐量。因此，快速Benes网络可用于以下通信场景。

（1）云服务商内部网络：云计算服务提供商自身拥有超大规模的计算集群，但对外租售算力资源的粒度较细，因此其内部流量特征会随着单用户租赁规模、用户数量以及用户具体用途变化而差异极大。使用可重构光互连网络可以根据用户具体需求将网络动态重构为最优拓扑构型，但以往的工程实践重构粒度很粗，往往倾向于在用户启动虚拟机前就将网络拓扑构建完毕，并不会基于用户需求动态变更网络拓扑。使用快速Benes网络，既可利用其低成本局部重构特性实现应用级的动态网络适配，又能确保尽可能不影响其他用户的通信负载，并且能实现百节点以上的互连规模，为算力资源的动态调配提供非常大的浮动空间。

（2）高性能计算集群：无论是传统的一致性内存访问（uniform memory access，UMA）模型还是非一致性内存访问（non-uniform memory access，NUMA）模型的计算节点，受物理特性与综合成本限制，其内存容量不能无限扩大，故超算集群往往使用高性能互连网络将数以万计的计算节点互连以实现超大内存容量，并配合如Map-Reduce等计算模型实现并行计算。传统互连网络由于是静态网络，在对不同目标通信时通常会触发多次数据转发，占用数据链路导致通信效率下降，已有相当多的研究针对此类场景优化通信模式以降低效率折损。但对于可动态重构拓扑的网络则不然，网络总是能在存在高带宽通信需求时动态重构出对应的点对点物理链路以避免数据转发。在此类应用场景中使用快速Benes网络，则可快速构建应用所需的点对点物理链路，减少数据转发频率，在保证单一交换装置构建的扁平化互连网络有足够大互连规模的同时，网络能快速响应接入节点的路由请求并避免影响其他无关节点的通信过程，确保计算任务对带宽和时延的双重高要求。

5 结束语

本文针对百节点以上互连规模可重构光互连网络难以兼顾局部重构成本、链路损耗、路由求解速度的问题，提出可重排无阻塞的快速Benes网络和与之对应的单接入节点快速局部重构算法。快速Benes网络以交换网络开关级数比Benes网络增加2、构建所需开关总数变为后者2倍多为代价，实现了稳定的低成本局部重构，单一接入节点路由变更额外影响链路数从平均0.71N条/最高N条降低至平均1.59条/最高N/8条，N为互连规模。算法求解局部重构也会给出具体被破坏的既有链路和对应接入节点，将需要同步并进行数据流量排空的节点数从N降低至平均不到4，使其与Crossbar等严格无阻塞网络相仿，避免了使用全局重构前大规模流量排空导致的网络吞吐量大幅降低。

针对百节点互连规模下快速Benes网络内核心网络局部重构求解难度较高、成功率较低的

问题，本文提出了路由重整算法，可大幅提高两次路由重整之间的局部路由变更成功率，且平均每 143 次局部重构才会发生一次路由重整，实际操作时可使用定期路由重整规避其相对较高的流量排空开销。本文设计并实现了与算法对应的局部重构求解硬件加速器，112×112 至 448×448 互连规模下求解消耗时钟周期数保持不变为 8~21 周期，且单节点局部路由变更平均求解周期数仅为 9.28，可实现 224×224 互连规模下 79 ns/次的求解频率，远好于以往百节点规模单次全局重构耗时的接近微秒级，为大规模互连网络的高性能路由提供基础。

快速 Benes 网络充分结合了 Benes 网络开关单元消耗量小与 Crossbar 网络局部重构求解快、重构开销小的特点，并有效弥补了二者的不足。使用高性能光开关单元配合快速 Benes 网络结构构建的电路交换网络，在实现单一接入节点路由变更快速求解的同时，可获得与 Crossbar 网络近似的极低重构开销与重载场景下极高的实际吞吐量，使灵活的路由调度策略得以实施，展现了快速 Benes 网络在未来光通信领域的潜力。

参考文献：

[1] Parsonson C W F, Shabka Z, Chlupka W K, et al. Optimal control of SOAs with artificial intelligence for sub-nanosecond optical switching[J]. Journal of Lightwave Technology, 2020, 38(20): 5563-5573.

[2] Maniotis P, Dupuis N, Schares L, et al. Intra-node high-performance computing network architecture with nanosecond-scale photonic switches[J]. Journal of Optical Communications and Networking, 2020, 12(12): 367.

[3] Farrington N, Porter G, Radhakrishnan S, et al. Helios: a hybrid electrical/optical switch architecture for modular data centers[C]// Proceedings of the ACM SIGCOMM 2010 Conference. New York: ACM Press, 2010: 339-350.

[4] Wang G H, Andersen D G, Kaminsky M, et al. C-Through: part-time optics in data centers[C]//Proceedings of the ACM SIGCOMM 2010 Conference. New York: ACM Press, 2010: 327-338.

[5] Bazzaz H H, Tewari M, Wang G H, et al. Switching the optical divide: fundamental challenges for hybrid electrical/optical datacenter networks[C]//Proceedings of the 2nd ACM Symposium on Cloud Computing. New York: ACM Press, 2011: 1-8.

[6] Beneš V E. Permutation groups, complexes, and rearrangeable connecting networks[J]. Bell System Technical Journal, 1964, 43(4): 1619-1640.

[7] Zhao S Y, Lu L J, Zhou L J, et al. 16×16 silicon Mach–Zehnder interferometer switch actuated with waveguide microheaters[J]. Photonics Research, 2016, 4(5): 202.

[8] Lu L J, Zhao S Y, Zhou L J, et al. 16×16 non-blocking silicon optical switch based on electro-optic Mach-Zehnder interferometers[J]. Optics Express, 2016, 24(9): 9295-9307.

[9] Qiao L, Tang W J, Chu T. 32×32 silicon electro-optic switch with built-in monitors and balanced-status units[J]. Scientific Reports, 2017(7): 42306.

[10] Hou W G, Guo P X, Guo L, et al. O-star: an optical switching architecture featuring mode and wavelength-division multiplexing for on-chip many-core systems[J]. Journal of Lightwave Technology, 2022, 40(1): 24-36.

[11] Wang Y, Qin Y B, Deng D Z, et al. A 28 nm 27.5 TOPS/W approximate-computing-based transformer processor with asymptotic sparsity speculating and out-of-order computing[C]// Proceedings of the 2022 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE Press, 2022: 1-3.

[12] Waksman A. A permutation network[J]. Journal of the ACM, 1968, 15(1): 159-163.

[13] Nassimi, Sahni. Parallel algorithms to set up the Benes permutation network[J]. IEEE Transactions on Computers, 1982, C-31(2): 148-154.

[14] Lee K Y. A new benes network control algorithm[J]. IEEE Transactions on Computers, 1987, C-36(6): 768-772.

[15] Koloko L, Matsumoto T, Obara H. Design and implementation of fast and hardware-efficient parallel processing elements to set full and partial permutations in Beneš networks[J]. The Journal of Engineering, 2021(6): 312-320.

[16] Nikolaidis D, Groumas P, Kouloumentas C, et al. Novel Benes network routing algorithm and hardware implementation[J]. Technologies, 2022, 10(1): 16.

[17] 秦梦远, 刘宏伟, 郝沁汾 . 高性能 Benes 网络路由求解算法及硬件加速器[J]. 计算机工程与应用, 2025, 61 (14): 163-175.
Qin M Y, Liu H W, Hao Q F. High-performance Benes network routing algorithm and hardware accelerator[J]. Computer Engineering and Applications, 2025, 61 (14): 163-175.

[18] Wang C, Yoshikane N, Elson D, et al. Modoru: Clos nanosecond

optical switching for distributed deep training[J]. Journal of Optical Communications and Networking, 2024, 16(1): A40-A52.

[19] Padmanabhan K, Netravali A. Dilated networks for photonic switching[J]. IEEE Transactions on Communications, 1987, 35(12):1357-1365.

[20] Kabacinski W. Modified dilated Benes networks for photonic switching[J]. IEEE Transactions on Communications, 1999, 47(8): 1253-1259.

[21] Lu E Y, Zheng S Q. Fast reconfiguration algorithms for time, space, and wavelength dilated optical Benes networks[J]. International Journal of Parallel, Emergent and Distributed Systems, 2007, 22(1): 39-58.

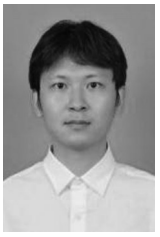
[22] 张金花, 武保剑, 邱昆. 扩张型Benes光交换集成芯片路由算法[J]. 光通信技术, 2019, 43(2): 1-8.
Zhang J H, Wu B J, Qiu K. Routing algorithm of dilated Benes optical switching integrated chip[J]. Optical Communication Technology, 2019, 43(2): 1-8.

[23] 张金花, 武保剑, 邱昆. 扩张型Benes光交换芯片未满配置情形下的约束链路路由算法[J]. 激光与光电子学进展, 2019, 56(21): 211301.
Zhang J H, Wu B J, Qiu K. Constrained link routing algorithm for dilated Benes optical switching chips under non-full configuration[J]. Laser & Optoelectronics Progress, 2019, 56(21): 211301.

[24] Goke L R, Lipovski G J. Banyan networks for partitioning multiprocessor systems[C]//Proceedings of the 1st Annual Symposium on Computer Architecture - ISCA '73. New York: ACM Press, 1973: 21-28.

[25] Clos C. A study of non-blocking switching networks[J]. Bell System Technical Journal, 1953, 32(2): 406-424.

[作者简介]



秦梦远（1994– ），男，中国科学院计算技术研究所博士生，主要研究方向为新型计算机系统结构、Benes网络、可重构光互连系统等。

刘宏伟（1984– ），男，博士，中国科学院计算技术研究所高级工程师，主要研究方向为高性能处理器设计、异构计算、硬件安全与加密芯片、软件定义芯片等。

郝沁汾（1969– ），男，中国科学院计算技术研究所正高级工程师、博士生导师，主要研究方向为高性能计算机、高端SMP服务器、CPU等。