

RapidIO 网络中一种基于最小隔离块的流量驱动机制

殷从月, 张兴明, 魏 帅, 汪 欣

(国家数字交换系统工程技术研究中心, 郑州 450002)

摘 要: 针对 RapidIO 路由网络枚举问题, 提出一种基于最小隔离块的流量驱动机制。在基本 RapidIO 路由拓扑结构上给出最小隔离块的概念, 对路由网络进行隔离块划分, 并以隔离块为基础, 深度优先遍历整个 RapidIO 路由网络。同时引入流量驱动机制, 通过流量驱动能耗算法和流量路径分配算法对隔离前后的能耗值、功耗值以及 RapidIO 交换机的平均工作时间进行比较, 从而验证算法的可靠性。实验结果表明, 该机制可以减少 RapidIO 交换机的枚举数量, 避免重复枚举, 而且随着路由结构层数的拓展, 隔离效果越明显。

关键词: RapidIO 路由网络; 最小隔离块; 深度优先遍历; 流量驱动; 重复枚举; 能耗; 功耗

开放科学(资源服务)标识码(OSID):



微信扫码
听作者独家介绍本文

中文引用格式: 殷从月, 张兴明, 魏 帅, 等. RapidIO 网络中一种基于最小隔离块的流量驱动机制[J]. 计算机工程, 2018, 44(9): 118-123, 135.

英文引用格式: YIN Congyue, ZHANG Xingming, WEI Shuai, et al. A flow driven mechanism based on minimum isolation block in RapidIO network[J]. Computer Engineering, 2018, 44(9): 118-123, 135.

A Flow Driven Mechanism Based on Minimum Isolation Block in RapidIO Network

YIN Congyue, ZHANG Xingming, WEI Shuai, WANG Xin

(National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450002, China)

[Abstract] Aiming at the enumeration problem of RapidIO routing network, a flow driven mechanism based on minimum isolation block is proposed. The concept of minimum separation block is given, and the isolation block is divided into the routing network. Then a depth first traversal search is carried out for the entire RapidIO routing network. And the flow driven mechanism is introduced to compare the energy consumption value, power consumption value and the average operation time of the RapidIO switches before and after isolation, to verify the reliability of the algorithm through the flow driven energy algorithm and flow path allocation algorithm. Experimental results show that the enumeration number of RapidIO switches can be reduced and repeated enumeration can be avoided. With the expansion of the routing structure layer, the isolation effect is more obvious.

[Key words] RapidIO routing network; minimum isolation block; depth-first traversal; flow driving; repeated enumeration; energy consumption; power consumption

DOI: 10.19678/j.issn.1000-3428.0049197

0 概述

近年来, 随着互联网、微处理器和存储器的高速发展, RapidIO 作为一项系统内部的互连技术, 在嵌入式设备市场中的需求越来越大^[1]。RapidIO 行业协会具有开放性, 一直以来, 许多类型的设备供应商都对嵌入式市场有相应的支持, 在这种环境下, 此类开放性就会显得尤为适合^[2]。

RapidIO 总线互连技术是一种以包交换为基础的

系统级内部互连协议, 它采用的是高速串行总线技术, 主要包括面向相关串行控制平面应用、串行背板和数字信号处理(Digital Signal Processing, DSP)的串行接口, 以及面向系统级互连和高性能微处理器的并行接口^[3]。相比于高速串行计算机扩展总线标准(Peripheral Component Interconnect express, PCIe)和万兆以太网, RapidIO 更加具备低时延、高吞吐量、高带宽、高性能等特点, 能够支持板之间以及芯片之间的灵活通信, 其性能可以达到 60 Gb/s 的水平^[4]。

基金项目: 国家自然科学基金(61572520, 61521003); 国家科技重大专项(2016ZX01012101)。

作者简介: 殷从月(1994—), 女, 硕士研究生, 主研方向为 RapidIO 路由网络; 张兴明, 教授; 魏 帅, 讲师; 汪 欣, 工程师。

收稿日期: 2017-11-06 **修回日期:** 2018-01-03 **E-mail:** 503637088@qq.com

随着对 RapidIO 研究的深入,研究者由对 RapidIO 协议本身的探讨逐渐转向更加复杂的 RapidIO 路由网络系统,而枚举作为该系统特有的一种搜索方式,是研究的一个热点。文献[5]列举了 RapidIO 系统网络的枚举过程,文献[2]也给出了相应的系统枚举启动实例,两者对以 RapidIO 为基础,由 1 个交换机、2 个系统主机、1 个高速 I/O 卡和 1 个 DSP 阵列组合而成的框架系统运行机制进行逐步阐述,但都没有进一步叙述如何对 RapidIO 路由组合网络进行优化。文献[6]提出了一种 RapidIO 网络递归枚举的组网方法,它采用的是以深度优先为基础的路径加权方式,若只是单一采用深度优先的枚举算法,将无法获得最优路径。该方法对于解决该类问题起到了一定的作用,但是对于更加复杂化的环形拓扑结构、星型拓扑结构、星型与环形相互叠加的拓扑结构以及其他类型的拓扑结构未必适用,实际枚举过程会显得很繁琐,甚至会出现重复枚举的情况。

为了更好地解决 RapidIO 路由网络枚举的实现问题,需要对 RapidIO 路由组合结构进行一定的等效优化,对复杂的路由网络进行简化。文献[7]描述了在配电网拓扑结构中,将部分对故障后果性质相同的设备进行合并后,以合并后的设备为单位,进行深度优先搜索枚举分析,这样能够大大降低枚举对

象的数量,又可以避免重复枚举同一类型的设备。

本文对文献[8]中 RapidIO 胖树结构的路由网络进行横向和纵向的拓展,并采取相应的整合优化措施,将性质和结构相同的路由网络组合成一个模块,称每个模块为最小隔离块,以此减少枚举数量,同时避免重复枚举。为更好地证明上述思想对 RapidIO 路由网络确实能起到优化作用,引入流量驱动机制,根据流量计算出相应的能耗值,通过路由网络对前后能耗值的降低来证明上述思想的可靠性,将该算法称为流量驱动能量感知路径算法^[9]。

1 流量驱动机制描述与定义

定义 1 基于最小隔离块的描述,为了更好地定义最小隔离块,以图 1 为例说明确定最小隔离块的方法。图 1 所示为一个 RapidIO 路由拓扑结构,其中包括 1 个 RapidIO 主机和 3 个 RapidIO 设备, RapidIO 交换机组成一个 $2n \times 2n$ 的阵列。以与 RapidIO 主机相连的虚线框为例,由于 RapidIO 交换机 1、2、3、4 的结构和性质相同,可以将这 4 个交换机进行组合,构成一个模块,并称该模块为最小隔离块。同理,对其余的 RapidIO 交换机进行类似的隔离块划分,最终可以得到如图 2 所示的 RapidIO 路由拓扑简化结构。

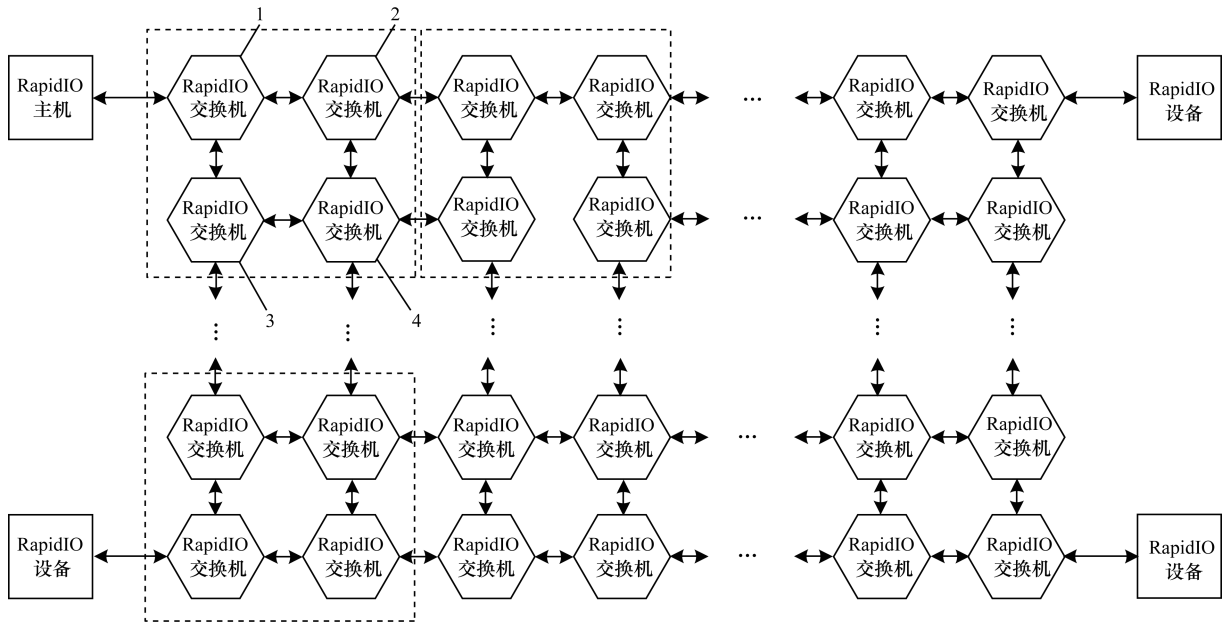


图 1 $2n \times 2n$ 交换机阵列的 RapidIO 路由拓扑结构

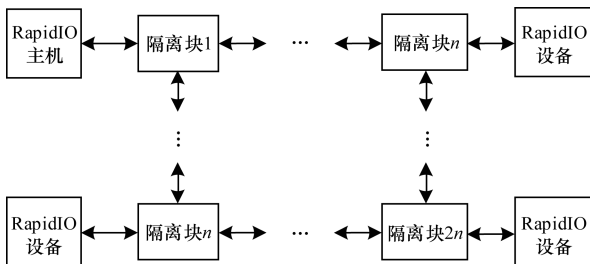


图 2 $2n \times 2n$ 交换机阵列的 RapidIO 路由拓扑简化结构

定义 2 树形拓扑结构中每个 RapidIO 交换机的位置设定为坐标 (i, j) , 其中 $i, j \in \{0, 1, \dots\}$, i 按照从下至上的顺序表示交换机的号, j 按照从左至右的顺序表示交换机的号。当 $i > 0$ 时, j 在区间 $[0, \frac{k^2}{4} - 1]$ 的范围以内; 当 $i = 0$ 时, j 在区间 $[0, \frac{k^2}{4} - 1]$ 的范围以内, 其中 k 代表交换机的端口数量^[10]。

定义 3 本文以单个 RapidIO 交换机的功耗模型为基础,建立整个 RapidIO 树形网络结构功耗的模型,如式(1)所示。其中, $t_{i,j}$ 表示交换机 (i,j) 的工作时间, $t_{i,j,k}$ 表示交换机 (i,j) 第 k 个端口的工作时间。式(1)表明,在建立单个交换机功耗模型时需要考虑线卡数量、线卡类型、底板类型以及交换机网络接口的功耗^[11]。式(2)表示 RapidIO 路由网络能耗的构成,其中 $P_{i,j}^{\text{fix}}$ 表示交换机 (i,j) 的固定功耗, $P_{i,j,k}^{\text{port}}$ 表示交换机 (i,j) 第 k 个端口的功耗。

$$\begin{aligned} \text{Power}_{\text{switch}} = & \text{Power}_{\text{chassis}} + \text{num}_{\text{linecards}} \times \text{Power}_{\text{linecards}} + \\ & \sum_{i=0}^{\text{configs}} \text{numports}_{\text{configs}_i} \times \text{Power}_{\text{configs}_i} \times \\ & \text{utilizationFactor} \end{aligned} \quad (1)$$

$$\begin{aligned} E = & \sum_{0 \leq i \leq 1} \sum_{0 \leq j \leq \frac{k^2}{2} - 1} (P_{i,j}^{\text{fix}} \cdot t_{i,j} + \sum_{k \in K} P_{i,j,k}^{\text{port}} \cdot t_{i,j,k}) + \\ & \sum_{i=2} \sum_{0 \leq j \leq \frac{k^2}{4} - 1} (P_{i,j}^{\text{fix}} \cdot t_{i,j} + \sum_{k \in K} P_{i,j,k}^{\text{port}} \cdot t_{i,j,k}) \end{aligned} \quad (2)$$

定义 4 RapidIO 主机传输速率与交换机端口转发速率相同,记为 r 。假设在 RapidIO 网络拓扑结构中存在两股流量,分别称作为 f_1 和 f_2 ,其中, f_1 的大小记为 m_1 , f_2 的大小记为 m_2 ,并且 $m_1 > m_2$ ^[12]。交换机端口的工作时间为 $t_{i,j} = t_{i,j,k} = \frac{m_1 + m_2}{r}$ 。

定义 5 路由网络能耗模型的主要目的是为了式(3),而该模型的关键是如何计算出端口 $t_{i,j,k}$ 的工作时间以及交换机 $t_{i,j}$ 的工作时间。

$$E = E_{\min} \quad (3)$$

假定路由网络中流量的数目为 ρ ,并且 ρ 的区间范围为 $[0, +\infty]$ 。将 $r_{i,j,k}$ 定义成交换机 (i,j) 第 k 个端口的传输速率, $f(x)$ 是流量 x 的大小, $y_{i,j,k}$ 是交换机 (i,j) 第 k 个端口的工作时间, $S_{i,j,k}(x)$ 是状态标志。即如果 $S_{i,j,k}(x) = 1$,则代表流量 x 经过交换机 (i,j) ;如果 $S_{i,j,k}(x) = 0$,则代表流量 x 未经过交换机 (i,j) 。式(4)给出了上述变量的关系。

$$y_{i,j,k} = \frac{\sum_{x=0}^{\rho} [S_{i,j,k}(x) \cdot f(x)]}{r_{i,j,k}} \quad (4)$$

当计算工作时间 $t_{i,j,k}$ 时,必须清楚路由网络中的传输要以流量为基础,两者不可分割,如式(5)所示。此外,交换机 (i,j) 中第 k 个端口的工作时间并不是整个流量流经端口的数量与端口转发速率的比值,而是与流量流经的所有端口相关。因此,要先分别计算每个流量流经交换机端口的工作时间,然后从中选出最大值作为流量 x 交换机端口的工作时间,如式(6)所示,记为 $t_{i,j,k}^x$ 。接着,比较所有流量交换机端口的工作时间 $t_{i,j,k}^x$,从中选出最大值作为交换机 (i,j) 第 k 个端口的工作时间,如式(7)所示,记为 $t_{i,j,k}$ 。最后,可以通过比较交换机每个端口的工作时

间以及选出最大值 $t_{i,j}$ 来获得交换机 (i,j) 的工作时间,如式(8)所示。式(9)给出了交换机的相关路径信息。

$$\sum_{0 \leq i \leq 2k \in K} S_{i,j,k}(x) \leq 2, x \in \rho \quad (5)$$

$$t_{i,j,k}^x = \max \{y_{i,j,k}/i, j, k \in \text{Path}, x \in \rho\} \quad (6)$$

$$t_{i,j,k} = \max \{t_{i,j,k}^x, x \in \rho\} \quad (7)$$

$$t_{i,j} = \max \{t_{i,j,k}, k \in K\} \quad (8)$$

$$\text{Path} = \left\{ \frac{(i,j,k,x)}{S_{i,j,k}(x)} = 1, x \in \rho \right\} \quad (9)$$

在路由网络能耗模型中,上述公式均可作为约束条件。

2 核心算法设计

2.1 深度优先遍历算法

根据 RapidIO 协议规范^[5],在 RapidIO 路由网络拓扑结构中,一般采用深度优先遍历的枚举算法,因为广度优先遍历算法空间复杂度大,在规模较大的 RapidIO 网络中容易找不到最优路径。具体算法流程如下^[13]:

算法 1 深度优先遍历算法

步骤 1 将与主机 Host 所连的交换机 Switch M 作为出发点。

步骤 2 判断与该交换机 M 端口相连的是交换机 Switch N 还是终端 EndPoint。

步骤 3 如果交换机 N 未被访问过,则将 N 标记为已被访问,并从 N 点继续进行深度优先搜索。

步骤 4 如果交换机 N 已被访问过,则只与交换机 N 建立连接关系。

步骤 5 若相连的是终端 EndPoint,则将该终端与交换机 M 建立一定的关系。

步骤 6 如果任何一个交换机 Switch 周围相邻的交换机结点均已被访问过,则从最近一个被访问的节点开始按逆序依次退回,直至初始顶点。

步骤 7 当所有 Switch 顶点均已被访问过,或者从任何已被访问过的 Switch 顶点出发,再也无法到达未被访问过的 Switch 顶点,则终止该算法。

算法 1 为 RapidIO 路由网络中较为普遍的一种可以遍历各个交换机 Switch 的方式,其算法时间复杂度为 $O(n)$,其中, n 代表 RapidIO 路由网络中各个路径所经过的交换机 Switch 结点的平均数量。经过算法 1,可以得到 RapidIO 路由网络中所有遍历的路径,继而能够从中筛选出符合约束要求的某些路径。

2.2 路由拓扑网络最小隔离块的生成

对文献[8]中的 RapidIO $K=4$ 的胖树结构进行纵向拓展。一般地, $k=0$ 的胖树结构从上而下由核心层、汇聚层以及边缘层组成,此处核心层的部分不

作改变,对汇聚层和边缘层的部分纵向拓展一层,得到如图 3 所示的 RapidIO 胖树拓扑结构拓展图。为了减少不必要的能耗,图中有 16 个与链路相关的 RapidIO 交换机是开放的,其余交换机均关闭。结合定义 4, f_1 链路是从主机 h_0 到 h_4 , f_2 链路是从主机 h_2 到 h_6 。以 f_1 链路为例,一般情况下,若要从主机 h_0 到 h_4 的路径访问,需要按照上文中所述的深度优先遍历算法来完成各个 RapidIO 交换机节点之间的

相互搜索和枚举。但是根据定义 1,可以将 RapidIO 交换机 (0,0)、(0,1)、(1,0) 和 (1,1) 组成一个最小隔离块,将交换机 (2,0)、(2,1)、(3,0) 和 (3,1) 组成第 2 个最小隔离块,其余的 RapidIO 交换机同理。接着在局部上,分别在各个最小隔离块内进行深度优先遍历;在整体上,又以各个隔离块为单位,在各隔离块之间进行深度优先遍历,大大减少了 RapidIO 交换机的枚举数量,又可以避免重复枚举。

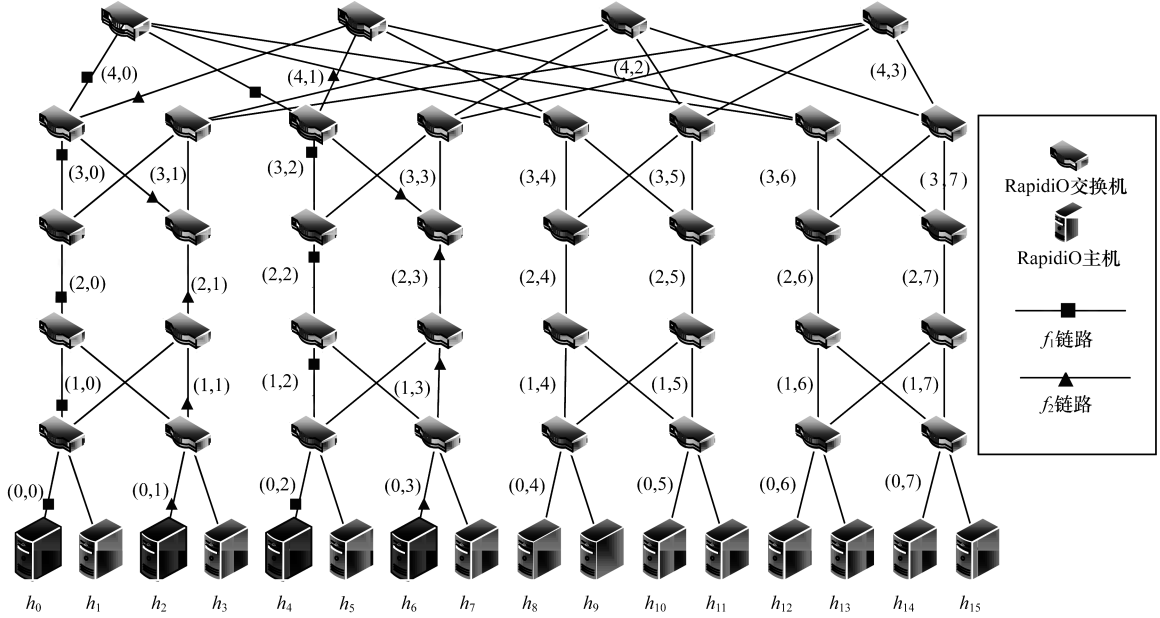


图 3 RapidIO 胖树拓扑结构拓展

2.3 流量驱动能耗算法

设置该算法的目的是通过计算能耗值,评估系统的性能改善程度。根据定义 3 和定义 4,式(2)表示了整个 RapidIO 路由网络结构能耗的构成,由此可以得到图 3 中 RapidIO 胖树拓扑结构拓展后的整个能耗值。对最小隔离块使用前后路由网络的能耗值进行对比,如式(10)和式(11)所示, E_1 代表隔离块使用前的能耗值, E_2 代表隔离块使用后的能耗值。

$$E_1 = P_{i,j}^{\text{fix}} \times \frac{9m_1 + 8m_2}{r} + 26 \times P_{i,j,k}^{\text{port}} \times \frac{m_1 + m_2}{r} \quad (10)$$

$$E_2 = P_{i,j}^{\text{fix}} \times \frac{9m_1 + 8m_2}{r} + 22 \times P_{i,j,k}^{\text{port}} \times \frac{m_1 + m_2}{r} \quad (11)$$

在计算 E_2 时,可以将隔离块与隔离块之间衔接的端口合并成一个端口,抽象成类似图 2 所示的理想化数学模型,从而表示由于最小隔离块的使用,在流量驱动局部范围内已经减少了交换机的枚举数量,降低了整体能耗值。整个流量驱动能耗算法如下^[14]。

算法 2 流量驱动能耗算法

输入 FT, TR // RapidIO 路由网络结构和流量矩阵

输出 PATH, E, P, TIME_AVE // 每个流量分配的路径, RapidIO 路由网络的总能耗, RapidIO 路由网络的总功

耗, RapidIO 交换机的平均工作时间

1. Topology() // 胖树网络拓展结构初始化
2. Traffic() // 流量矩阵初始化
3. while (TR) // 给流量矩阵中未分配的流量分配路径
4. {
5. for $i \in \text{TR}$
6. FindPath(FT, src, des, P) // 以有限制性的流量驱动
7. E += Energy_Cost() // 计算整个能耗值
8. }
9. end while

算法 2 提供了一种路由网络在流量驱动模式下的能耗值计算方法,但在该算法中涉及到流量路径的分配问题,下文介绍的流量路径分配算法能够解决这个问题。

2.4 流量路径分配算法

在流量路径分配问题中,会标记分配的流量以及通过每个端口的流量数。当给新的流量分配路径时,该算法会根据各个端口的状态来选择路径。如果流经每个端口的流量数小于该端口流量的最大数量,路径将会被分配给该流量,具体算法过程如下。

算法 3 流量路径分配算法

1. FindPath(FT, src, des, P)
2. {

```

3. switch_port_flag[ ][ ] // 标记流经每个端口的流量数
4. traffic_flag[ ][ ] // 标记分配路径
5. flow_num // 流经一个端口的最大流量数
6. if src/POD == des/POD // 源和目的在相同的 POD 中
7. for the port the flow passes && switch_port_flag[ ][ ] <
flow_num
8. write the subscript information of the port to P
9. else // 源和目的在不同的 POD 中
10. for the port the flow passes && switch_port_flag[ ][ ] <
flow_num
11. write the subscript information of the port to P
12. traffic_flag[src][des] = 1 // 标记分配路径
13. switch_port_flag[ ][ ] + = 1 // 更新 RapidIO 交换机
// 和端口的分配信息
14. }

```

算法 3 可以使流量路径得到合理的分配,并能即时更新 RapidIO 交换机和端口的分配信息,做到不重复分配路径。

3 实验设置与结果分析

3.1 实验设置

实验在 VxWorks 嵌入式操作系统中完成仿真,该操作系统是由美国 WindRiver 公司设计开发的,具有用户开发环境友好和持续发展能力良好等特点。系统中使用的是 wind 内核,此内核具有较高的稳定性和可移植性,能够用于高度裁剪,性能优良,形式精简。仿真实验参照以下实际内容进行: RapidIO 嵌入式互连系统以 MPC8548 处理器作为端点处的主机设备, MPC8548 是由 Freescale 公司生产的处理器,适用于高速低时延的环境,处理能力比较强大^[15];此外,交换器件将采用 Tsi578 芯片,该芯片由 IDT 公司所生产,是第三代 Tundra 串行 RapidIO 交换机,所支持的聚合带宽高达 80 GB,操作便捷,使用模式灵活,易于修改^[16];迄今为止, RapidIO 协议已经更新至 4.0 的版本,但在实际操作环境中仍然使用 2.0 版本的 RapidIO 协议。

实验依据图 3 进行,在该路由拓扑结构中包括 36 个 RapidIO 交换机和 16 个 RapidIO 主机,比较最小隔离块使用前后整个路由网络的能耗值、交换机平均工作时间以及整个路由网络的功耗值。设置 RapidIO 默认接口速率为 2.5 GB/s,流量大小满足均值为 64 MB 的指数分布,采用的 RapidIO 交换机的固定功耗为 100 W,交换机端口频率可以配置为 1.25 GB、2.5 GB 以及 3.125 GB^[17],端口设置为 1 × 模式。

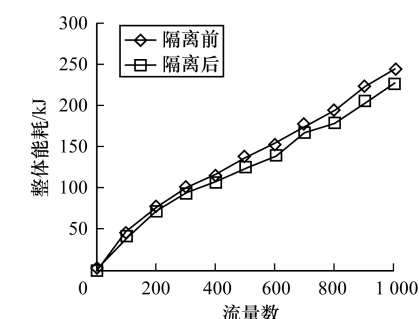
3.2 实验结果分析

为更好地验证最小隔离块对 RapidIO 路由网络的优化作用,从 RapidIO 路由网络意识流和 RapidIO

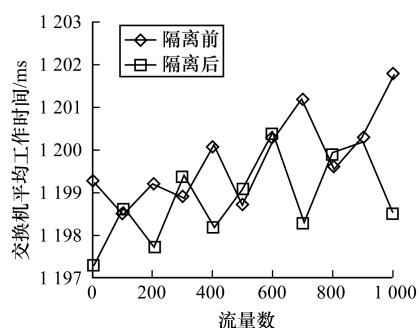
主机意识流两方面入手,与文献[8]方法比较 RapidIO 路由网络整体能耗、功耗以及交换机平均工作时间的大小。隔离前为文献[8]的实验结果,隔离后为本文的实验结果。所谓路由网络意识流,即单个流量的最大传输速率受到网络可用带宽的限制;而所谓主机意识流,即单个流量的最大传输速率受到主机转发速率的限制^[8]。

3.2.1 RapidIO 路由网络意识流

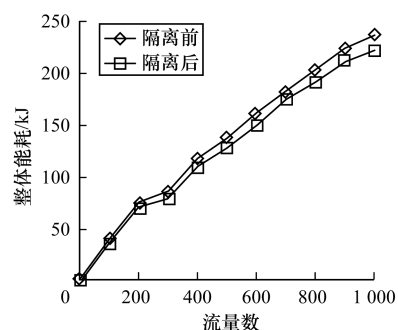
RapidIO 主机转发速率和交换机传输速率均设置为 2.5 Gb/s,实验结果如图 4 所示。实验表明,在使用隔离块之后, RapidIO 路由网络意识流整体能耗平均减少约 9.7%,功耗平均减少约 8.6%,而 RapidIO 交换机的平均工作时间在隔离前后均值相差不多,但也有所减少。



(a) 路由网络意识流整体能耗对比



(b) 路由网络意识流交换机平均工作时间对比



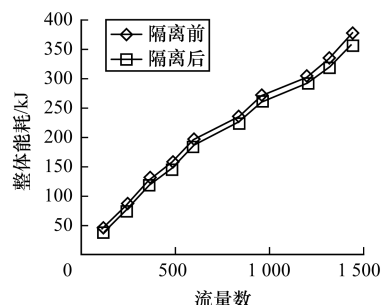
(c) 路由网络意识流整体功耗对比

图 4 RapidIO 路由网络意识流各项指标隔离前后对比

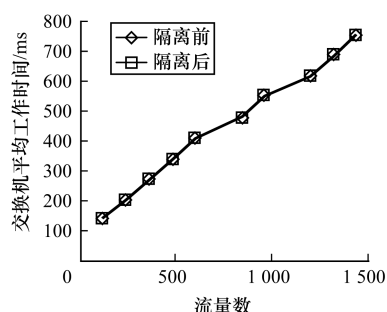
3.2.2 RapidIO 主机意识流

RapidIO 主机转发速率设置为 2.5 Gb/s,交换机传输速率设置为 3.125 Gb/s,实验结果如图 5 所示。

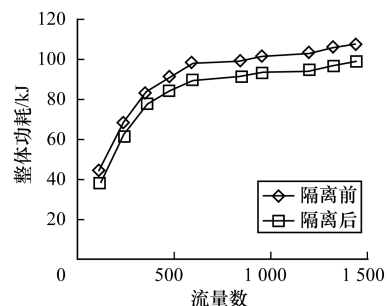
实验结果表明,在使用隔离块之后,RapidIO 主机意识流整体能耗平均减少约 7.9%,功耗平均减少约 12.3%,而 RapidIO 交换机的平均工作时间在隔离前后均值近乎相同,略微有所减少。



(a)主机意识流整体能耗对比



(b)主机意识流交换机平均工作时间对比



(c)主机意识流整体功耗对比

图5 RapidIO 主机意识流各项指标隔离前后对比

通过上述实验的对比结果可以明显看出,隔离块的使用对 RapidIO 路由网络性能起到了很好的改善作用。通过隔离块,可以减少 RapidIO 交换机的枚举数量,避免重复枚举。本文只是对常规($k=4$)的 RapidIO 胖树结构纵向拓展了一层,随着拓展层数的增多,上述对比图中的能耗值、功耗值和交换机平均工作时间会下降得更多,性能改善的效果更明显。

4 结束语

本文针对 RapidIO 路由网络枚举问题,提出了最小隔离块的概念,并以此为基础,引入流量驱动机制,对 RapidIO 路由网络简化前后的性能进行评估。对 RapidIO 路由网络进行深度优先遍历时,利用最

小隔离块的思想,最终实现 RapidIO 交换机枚举数量的减少以及避免重复枚举。然而,本文给出的数学理想化模型以隔离块为基础,同时用流量对该模型进行性能评估,在计算过程中可能会因为隔离块衔接处端口的合并而导致流量拥塞。虽然该问题几乎不会对评判、结果造成影响,但可以将减少流量拥塞作为今后工作的研究方向。

参考文献

- [1] FULLER S. RapidIO: The embedded system interconnect[M]. New York, USA: John Wiley & Sons, 2005.
- [2] 章 乐,李雅静,倪 明,等.一种基于 RapidIO 接口的嵌入式系统[J]. 计算机工程,2008,34(s1):1-3.
- [3] SIEGLE F, HABINC S, MARINIS K. SpaceFibre and serial RapidIO network layers (GRSPFI/GRSRIO): SpaceFibre, short paper[C]//Proceedings of SpaceWire Conference. Washington D. C., USA: IEEE Press, 2016: 1-4.
- [4] 林 玲,蒋 俊,倪 明,等. RapidIO 在多处理器系统互连中的应用[J]. 计算机工程,2006,32(4):244-246.
- [5] RapidIO interconnect specification annex1: software/system bring up specification, rev. 4. 0 [EB/OL]. [2017-11-06]. <http://www.rapidio.org>.
- [6] 钟海林,李国超,高伟林.一种 RapidIO 网络递归枚举方法:中国,103746910[P]. 2014-04-23.
- [7] 徐珍霞,周江昕.复杂配电网可靠性评估的改进故障遍历法[J]. 电网技术,2005,29(14):64-67.
- [8] SHU Lin, HAO Jie, SONG Yafang, et al. Optimal many-to-many personalized concurrent communication in RapidIO-based fat-trees[C]//Proceedings of International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. Washington D. C., USA: IEEE Press, 2016:343-350.
- [9] DUAN Pengfei, WANG Kun, YU Xiaoshan, et al. Flow driven energy-aware routing algorithm in data center network [C]//Proceedings of the 17th International Conference on Parallel and Distributed Computing, Applications and Technologies. Washington D. C., USA: IEEE Press, 2016:280-285.
- [10] KARANJOT S G, AMIYA N. Per-packet based energy aware segment routing approach for data center networks with SDN [C]//Proceedings of the 24th International Conference on Telecommunications. Washington D. C., USA: IEEE Press, 2017:1-6.
- [11] GUO Zehua, HUI Shufeng, XU Yang, et al. Dynamic flow scheduling for power-efficient data center networks[C]//Proceedings of International Symposium on Quality of Service. Washington D. C., USA: IEEE Press, 2016: 11200-11214.
- [12] SHANG Yunfei, LI Dan, XU Mingwei. Greening data center networks with flow preemption and energy-aware routing [C]//Proceedings of the 19th IEEE Workshop on Local and Metropolitan Area Networks. Washington D. C., USA: IEEE Press, 2013:1-6.

(下转第 135 页)

5 结束语

本文利用代理重签名、多分支路径树和随机掩码等技术,提出了一种支持用户撤销和数据动态更新操作的云端数据完整性验证方案,不仅在验证过程中能确保数据块的隐私性,而且还能有效抵抗CSP的伪造攻击、重放攻击和删除攻击。但该方案的存在不可伪造性依赖于理想的随机预言机,因此,下一步将设计标准模型下安全的云端数据完整性验证方案。

参考文献

- [1] SOOKHAK M, GANI A, KHAN M K, et al. Dynamic remote data auditing for securing big data storage in cloud computing [J]. Information Sciences, 2017, 380: 101-116.
- [2] KUMARI M P, PAUL R K. A study for authentication and integrity of data files in cloud computing [J]. International Journal of Science and Research, 2017, 2(9): 1-8.
- [3] ATENIESE G, BURNS R, CURTMOLA R. Provable data possession at untrusted stores[C]//Proceedings of the 14th ACM Conference on Computer and Communications Security. New York, USA: ACM Press, 2007: 598-609.
- [4] ERWAY C, KUPCCU A, PAPAMATHOU C, et al. Dynamic provable data possession[C]//Proceedings of the 16th ACM Conference on Computer and Communications Security. New York, USA: ACM Press, 2009: 213-222.
- [5] WANG Qian, WANG Cong, LI Jin, et al. Enabling public verifiability and data dynamics for storage security in cloud computing [C]//Proceedings of European Symposium on Research in Computer Security. Berlin, Germany: Springer, 2009: 355-370.
- [6] 李勇, 姚戈, 雷丽楠, 等. 基于多分支路径树的云存储数据完整性验证机制[J]. 清华大学学报(自然科学版), 2016, 56(5): 504-510.
- [7] WANG Cong, WANG Qian, REN Kui, et al. Privacy-preserving public auditing for data storage security in cloud computing [C]//Proceedings of the 29th Conference on Information Communications. Berlin, Germany: Springer, 2010: 525-533.
- [8] WANG Cong, CHOW S S M, WANG Qian, et al. Privacy-preserving public auditing for secure cloud storage [J]. IEEE Transactions on Computers, 2013, 62(2): 362-375.
- [9] WANG Cong, WANG Qian, REN Kui, et al. Toward secure and dependable storage services in cloud computing [J]. IEEE Transactions on Services Computing, 2013, 5(2): 220-232.
- [10] ZHU Yan, AHN G J, HU Hongxin, et al. Dynamic audit services for outsourced storages in clouds [J]. IEEE Transactions on Services Computing, 2013, 6(2): 227-238.
- [11] WANG Boyang, LI Baochun, LI Hui. Knox: privacy-preserving auditing for shared data with large groups in the cloud [C]//Proceedings of the 10th International Conference on Applied Cryptography and Network Security. Berlin, Germany: Springer, 2012: 507-525.
- [12] WANG Boyang, LI Baochun, LI Hui. Panda: public auditing for shared data with efficient user revocation in the cloud [J]. IEEE Transactions on Services Computing, 2015, 8(1): 92-106.
- [13] YU Y, NI J, AU M H, et al. Comments on a public auditing mechanism for shared cloud data service [J]. IEEE Transactions on Services Computing, 2015, 8(6): 998-999.
- [14] 张新鹏, 许春香, 张新颜, 等. 基于代理重签名的支持用户可撤销的云存储数据公共审计方案[J]. 计算机应用, 2016, 36(7): 1816-1821.
- [15] ATENIESE G, HOHENBERGER S. Proxy re-signatures: new definitions, algorithms, and applications [C]//Proceedings of the 12th ACM Conference on Computer and Communications Security. New York, USA: ACM Press, 2005: 310-319.
- [16] 冼友伦, 纪小明, 王显跃. 基于TS1578的RapidIO互连技术[J]. 电子与封装, 2014, 14(11): 26-30.
- [17] KLILOU A, BELKOUCH S, ELMAIZI A, et al. Case studies of data traffic management on a high-performance computing system based on multi-DSPs and serial RapidIO interconnect [C]//Proceedings of International Conference on Information Technology for Organizations Development. Washington D. C., USA: IEEE Press, 2016: 1-6.

编辑 顾逸斐

编辑 顾逸斐

(上接第123页)

- [13] 石炜, 邓伟. 基于RapidIO高速信号处理系统的网络枚举技术[J]. 电子设计工程, 2013, 21(4): 36-39.
- [14] DUBEY A, DUBE J. Energy aware routing technique for improving network: an analysis and performance [C]//Proceedings of International Conference on ICT in Business Industry and Government. Washington D. C., USA: IEEE Press, 2016: 1-6.
- [15] 黄振中, 柴小丽, 黎想, 等. 基于Vxworks的PCI-RapidIO桥驱动设计[J]. 计算机工程, 2010, 36(4): 239-240.