doi:10.3772/j.issn.1002-0470.2018.02.003

片上网络与系统域网络的协同设计探索①

刘小丽②*** 邬志轩*** 曹 政* 孙凝晖*

(*中国科学院计算技术研究所 北京 100190)

(**中国科学院大学计算机与控制工程学院 北京 100190)

摘 要 为进一步提升网络性能,在处理器内集成网络设备逐渐成为趋势,但随着片上处理器数目以及高维度系统域网络对片间互连端口数目需求的不断增加,传统的集成集中式 Router 的结构(如 IBM Blue Gene/Q)将面临可扩展性、片上流量均衡及片间接入公平性等问题。基于此,提出了片上网络和片间系统域网络协同设计方法,以分布式的虚拟 Router 架构替代传统集中式的 Router 架构,一方面通过将片间网络接口分布在片上网络,为处理器核提供均衡、公平的网络服务,另一方面省去了集中式 Router 中的交叉开关,进而消除了片间交换的扩展性限制,降低了 CPU 集成片间网络设备的成本。从协同路由、协同网络接口布局和协同参数设置三个方面对该协同设计方法进行了系统性的探索,并通过对 TMesh 网络架构进行实例研究,给出了虚拟 Router 架构的性能和可行性分析。

关键词 高性能计算(HPC),片上网络,系统域互连网络,网络死锁,路由器

0 引 言

随着半导体工艺的进步,单芯片晶体管数目不断增加,芯片内可集成的处理器核数越来越多,多核/众核处理器成为当前处理器的主流。从平衡设计角度出发,处理器核数的增加大大提高了处理器对网络性能的需求。目前,将网络路由器集成在处理器内是提高网络 I/O 性能的主要方法之一,它省去了桥片的转发开销,同时可以利用片上缓存和片上网络实现网络数据的高速传输。IBM Blue Gene/Q[1]、K computer[2,3]等已经使用集成网络设备的处理器构建了高性能计算机,Intel 的 Xeon Phi 也已集成 Omnipath 网络接口控制器 HFI[4],成为这一架构的重要推动者。本文提出了片上网络和片间系统域网络协同设计方法。

1 问题阐述

当前处理器设计仅侧重片内,即片上网络仅关注片内处理器核间通信,忽略了处理器核的片间通信需求。当前典型的集成网络设备的处理器结构如图 1(a)所示,它将一个集成的网络路由器(Router)挂载于片上网络,Router 仅通过一个片上网络接口与片上网络进行数据交换,若干个片间网络接口控制器用于跟其他处理器间的通信,所有端口之间通过交叉开关实现数据交换,IBM Blue Gene/Q、K

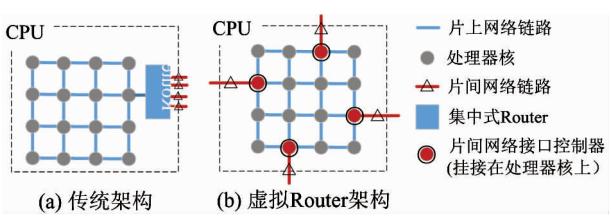


图 1 两种片上集成片间网络设备的架构

① 国家重点研发计划(2016YFB0200204,2016YFB0200205),国家自然科学基金(61572464,61331008)和国家重点实验室开放课题基金(2016GZKF0JT006)资助项目。

② 女,1986 年生,博士生;研究方向:计算机系统结构,高性能互连网络,I/O 虚拟化等;联系人,E-mail: liuxiaoli@ncic.ac.cn

(收稿日期:2017-10-09)

Computer 等都采用了类似的结构，但是该结构存在三个问题：

（1）扩展性问题：整个 CPU 的网络接入带宽受限于唯一的片上网络接口带宽，单端口带宽的提升能力约束了整个 CPU 的互连能力。

（2）网络拥塞问题：众多处理器核向单个片上网络接口的多对一通信，导致出口拥塞，在非丢包的片上网络中，该拥塞会进一步向内传导，严重干扰片上通信性能。

（3）不公平问题：不同处理器核到达片上网络接口的跳步数存在差异，这种差异将导致处理器核的通信性能差异，一般来说，距离片上网络接口越近，处理器核获得的延迟更低、带宽更高。

为了获得更优的性能和更低的实现成本，本文提出一种片上与片间网络协同的设计方案，以分布式的虚拟 Router 替代之前的集中式 Router，如图 1(b)所示，多个网络接口控制器被均匀部署在片上网络之上，它们之间的数据交换通过片上网络完成，与片上网络一起构成虚拟 Router。这样集中式 Router 结构中的单一片上网络接口被多个片间网络接口控制器(NIC)替代，使得流量分布更加均匀，同时交叉开关功能被可扩展的片上网络替代，处理器获得了更宽的片间网络接口带宽，Router 从一种集中式的 Scale up 架构，演化为一种分布式的 Scale out 架构。Shaw 等提出的 Anton 2[5] 系统使用了类似的架构，但是他们仅介绍了一种具体实现，没有对该结构对片上和片外网络性能产生的影响开展研究。

本文对虚拟 Router 架构、片上片间网络协同设计进行系统性的探索，涵盖了协同路由、协同接口布局和协同参数设置三个方面，并对 TMesh 网络架构进行了实例分析。TMesh 网络架构如图2所示，系统域网络为 3D Torus，每一个网络节点是多核/众核处理器计算节点，其片上网络采用 2D Mesh 网络。TMesh 的提出基于如下考虑：Mesh 和 Torus 分别是当前众核处理器、TOP10 超级计算机[6] 的主流拓扑之一；相比树形拓扑，Torus 类网络存在最复杂的死锁和多路径问题，有利于对虚拟 Router 架构进行更充分深入的研究。

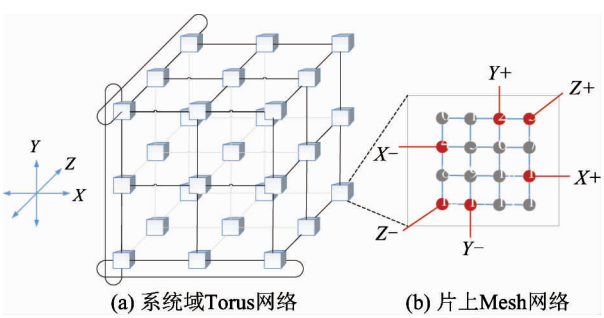


图 2　TMesh 网络架构

2　方法的关键点

协同路由死锁避免、协同网络接口布局及协同网络参数设置是本文方法的关键。

2.1　协同路由死锁避免

网络路由中一个数据流的包总在请求其他数据流的包占用的资源，而自己又占用着其他数据流的包所请求的资源，这种循环依赖导致数据包永远被阻塞的现象称为死锁。直接网络中常用的环绕网 Torus 拓扑存在三种死锁风险：(1)维序(dimension order)死锁：由于消息在不同维度转向形成环而造成死锁，如图 3(a)所示，这种死锁也存在于网状(Mesh)网络中；(2)环绕网死锁：由于物理上存在环路(wrap-around connection)，因此即使没有转向，在同一个维度传输也会形成环而造成死锁，如图 3(b)所示；(3)请求/应答死锁：请求通路和其响应通路间会形成环而造成死锁。

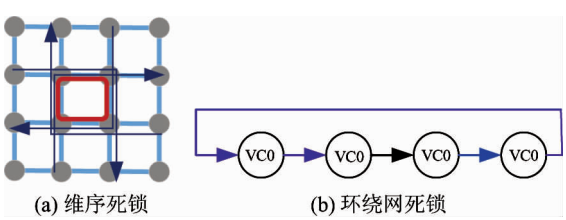


图 3　路由死锁

针对维序死锁，可以使用维序路由，即规定包路由时所使用的维度的顺序，打破不同维度间的循环依赖，从而避免死锁。如图 4(a)所示，假定 Mesh 内的路由总是先沿 1 所示的方向路由，然后再沿 2 所示的方向路由，根据 Dally 的死锁避免理论，此时形成了一个资源偏序[7]，从而死锁得以避免。针对环绕网死锁，通过增加一条虚通道，即在每一个维度内使用两条虚通道(VC0，VC1)，并设置一个虚通道变

更线(Dateline)[8],如图4(b)所示,在包穿越变更线之前使用低编号虚通道(VC0),在包穿过变更线之后则使用高编号虚通道(VC1),形成这样的一个虚通道依赖偏序(VC0→VC1),可打破由环路引入的同一维度内部的虚通道的循环依赖;同时,不同维度间使用维序路由,并在包从一个维度转向另一个维度时,将其使用的虚通道编号重置为低编号虚通道VC0,这样使用两条虚通道就可以避免环绕网中的死锁。

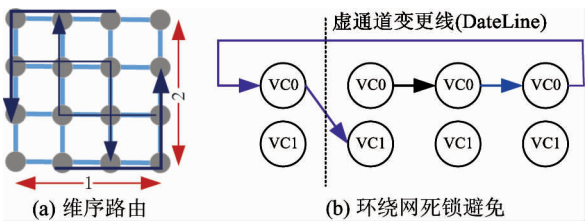


图4　传统死锁避免方法

然而,在片上片间网络协同场景下,网络层次的增加使得网络路由死锁问题变得更加复杂。以TMesh(Mesh片上网络+Torus片间系统域网络)为例,如图5所示,假定Torus和Mesh上的维序路由都使用*Y*-*X*顺序,Torus上使用2条虚通道避免死锁,虚通道切换规则使用日期变更线策略,且日期变更线位于每一维的回环通路上(如图5 Dateline_*X*和Dateline_*Y*所示)。图中0号虚通道(VC0)使用黑虚连线标出,1号虚通道(VC1)使用黑实连线标出。假定有一个包P0从S_0处理器核出发,路径记为P0_VC*i*(*i*表示使用的虚通道号);另有其他包P1和P2,同理,路径分别记为P1_VC*i*和P2_VC*i*。如图所示,P0首先经过*Y*维的时间变更线Dateline_*Y*沿路径P0_VC1到达CPU_n,然后由于要切换到*X*维,因此P0需要切换虚通道(如图5中维度切换(VC1→VC0)所示)并沿路径P0_VC0路由。同时,P1沿着P1_VC0路由;而P2由于穿越了时间变更线Dateline_*X*,首先沿着P2_VC0路由然后又切换到P2_VC1路由,并与P0交汇在CPU_n。可以发现,由于在CPU_n内引入了*X*维度内VC1到VC0的依赖关系(图示"维度切换(VC1→VC0)"),打破了原来只有VC0→VC1的虚通道依赖偏序,形成了一个维度内VC0→VC1→VC0循环依赖,进而形成死锁。这就使得传统使用维序路由及多个虚通道利用日期变更线策略解决死锁的方法失效,因此需要对协同路由死锁避免方法进行重新的考虑和设计,实现片上片间网络死锁隔离。

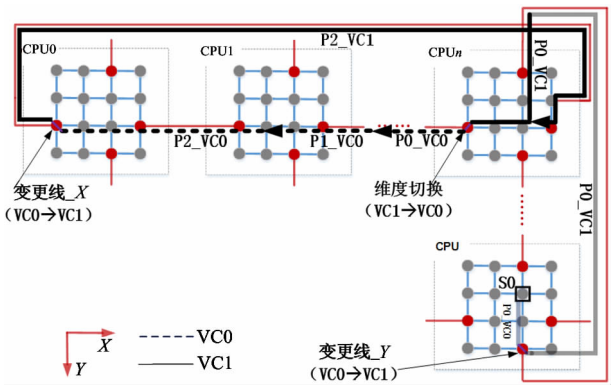


图5　协同网络死锁示意图

2.2　协同网络接口布局

利用片上网络的节点作为片上流量接入系统域网络的网络接口即为全局流量的转发点,网络接口的布局将直接影响全系统网络的性能。如图6(a)所示的网络接口布局,网络接口均匀分布在片上网络的边缘,这种策略可以更好地实现片上流量接入系统域网络时的负载均衡,但是在转发全局流量时,会增加全局流量的转发跳数,增加延时。如图6(b)所示的网络接口布局方案中,网络接口节点集中分布于片上网络的中心位置,这种策略可以降低全局流量转发的跳步数,减少片上流量接入系统域网络的出口距离,但是容易出现片上网络拥塞热点。

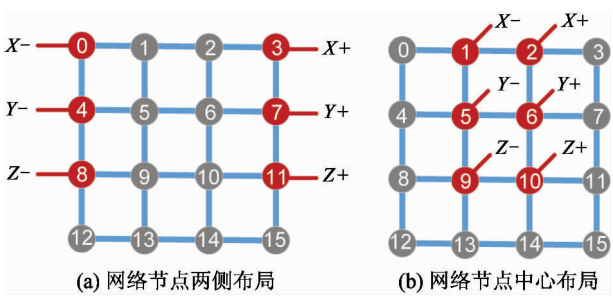


图6　不同网络接口节点布局示意图

因此在设计选择桥接点布局时,应结合布局策略对片上片间系统域网络性能的影响进行综合考虑。网络接口布局的影响可以归纳总结为以下几个方面:

(1) 对片上流量负载均衡的影响。如果需要网络接口的布局对片上流量具有比较好的疏导作用,则需要尽量将网络接口节点均匀分布在片上网络中。这样,片上流量在向外流出时可以充分利用片上网络内的链路,以避免产生拥塞热点,实现负载均衡的效果。

(2) 对片间系统域流量网络延迟的影响。网络延迟与路由跳数有很大关系,为了使路由时延迟尽量低,应当使得片间系统域数据流路由时,在片上网络内部经过的跳数尽量少。这就要求网络接口节点之间的距离尽可能地近,而这与第一个考虑因素是矛盾的,因此需要在这两个因素之间寻找一个均衡点。

(3) 对片上流量注入公平性的影响。网络接口节点的布局应当尽量减少与流量注入点的距离,以方便片上流量尽快从出口离开片上网络,减少对片上网络资源的占用。这就要求网络接口节点不应当集中在片上网络的某一边缘区域,因为这样会使所有路由出去的包在片上网络内部走过较长的路径。

2.3 协同网络参数设置

片上网络与系统域网络协同设计中,利用片上网络进行全局流量的桥接转发,在路由时片内的大量流量和片间的全局流量势必会在一定程度相互影响,影响程度直接决定这种协同设计架构的实用性。如果片内流量对全局通信影响较大,则说明片上网络拓扑的通信能力不足以支撑起进行额外全局流量路由的设计,此时需要对片上网络的通信能力进行加强,或者换用其他性能更好的片上网络拓扑结构。因此在协同设计时,需要考虑片内网络和系统域网络参数设置。

3 案例分析:TMesh 协同网络架构

本节以 TMesh 结构为示例,对第 2 节提出的关键问题进行量化分析,通过相关的评测对片上和系统域网络协同设计的优势进行分析和论证。

3.1 网络体系结构

3.1.1 TMesh 网络拓扑

TMesh 协同网络拓扑如图 2 所示,其网络拓扑有两个层次,片间系统域网络为 3D Torus,其三个维度分别表示为 X、Y、Z,每维长度记为 d_x、d_y、d_z。每个 Torus CPU 节点的内部是一个 2D Mesh 的片上网络,其两个维度分别表示为 M、N,每维的长度记为 d_m、d_n。整个 TMesh 网络拓扑中,共有 $d_x \times d_y \times d_z \times (d_m \times d_n - 2 \times d_{\text{Torus}})$ 个处理器核,其中 $(d_m \times d_n) > |2 \times d_{\text{Torus}}|$。

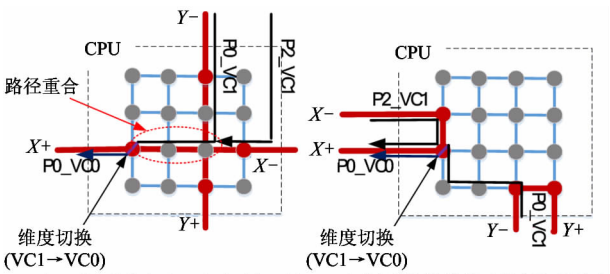
每个 CPU 节点实现为一个虚拟 Router 架构,如图 2(b)所示,在片上 2D Mesh 网络中选取 6 个节点不连接 CPU 核心,实现接入系统域网络的网络接口,每个网络接口表示系统域 3D Torus 网络中一个维度的一个方向。为了便于描述,2D Mesh 上的节点按照从左至右、从上到下的顺序编号,选取的实现网络接口的节点按照其连接到 Torus 的方向进行分组,连接到每个维度递减方向的节点归入 LN 组 $\{X-, Y-, Z-\}$,连接到每个维度递增方向的节点归入 RN 组 $\{X+, Y+, Z+\}$,每个组内的节点按照 $\{X, Y, Z\}$ 的顺序排列。如图 6(a)示意的网络接口节点的选取,可以记为 LN{0,4,8} + RN{3,7,11}。

3.1.2 死锁避免路由算法

片上网络与系统域网络的协同设计中,由于网络层次的增加,传统使用维序路由及多个虚通道利用日期变更线策略解决死锁的方法失效。从第 2.1 节所述的分析可以看出,传统死锁避免路由算法失效的关键在于系统域网络的不同维度在片上网络内存在交叉,如图 7(a)所示 Y 维和 X 维垂直交叉,Y 维向 X 维转换的路径与 X 维内路径有一段重合,进而在 CPU_n 内引入了同一维度内 VC1 到 VC0 的依赖关系,形成了系统域网络同一个维度内 VC0→VC1→VC0 循环依赖。因此,只需要保证在系统域网络上每一维度内的路由路径和不同维度之间转向的路由路径,在片上网络内没有重合,即可不借助额外的虚通道,保证路由算法无死锁。

在本案例 TMesh 中,系统域网络 Torus 和片上网络 Mesh 上都使用维序路由,通过对网络接口节点的布局做出适当限制,即可保证不同维度之间路由路径的链路无关性。如图 7(b)所示,将 $X+$ 和 $X-$ 网络接口节点沿同一 Y 维度坐标放置,$Y+$ 和 $Y-$ 网络接口节点沿同一 X 维度坐标放置,且保证

$X+$ 和 $X-$ 的 Y 维度坐标不同于 $Y+$ 和 $Y-$,$Y+$ 和 $Y-$ 的 X 维度坐标不同于 $X+$ 和 $X-$,即可使得路由算法无死锁。可以发现,转弯路径不与沿同一维度路由的路径重合。由于虚通道切换仅发生在包从桥接节点离开 Mesh 的时刻,因此可以认为这些转弯路径属于 Torus 的 Y 维度,从而这种桥接节点的布局使得 TMesh 拓扑对于维序路由呈现了与标准 Torus 相同的性质,这也是时间变更线策略在这种布局下可以生效的原因。



(a) 不同系统域维度交叉存在路径重合 (b) 不同系统域维度不存在路径重合

图 7 TMesh 网络死锁避免示意图

一般地,对于任意 TMesh 拓扑结构,设 Torus 有 n 维,第 i 维记为 E_i;Mesh 有 m 维,第 i 维记为 D_i,每一维长度为 k_i,$i \in \{0, 1, 2, \cdots, m-1\}$。不失一般性,假设 Torus 上的路由维序为 $E_0 \rightarrow E_1 \rightarrow \cdots \rightarrow E_{n-1}$,Mesh 上的路由顺序为 $D_0 \rightarrow D_1 \rightarrow \cdots \rightarrow D_{m-1}$。记节点 A 在 Mesh 内的 D_i 维度的坐标为 $MD_i(A)$,Torus 的第 i 维的桥接节点为 B_i,其中 $B_{li} \in B_i$ 且 $B_{ri} \in B_i$。要使使用 2 条虚通道和日期变更线切换策略的维序路由在 TMesh 上无死锁,需要满足以下条件约束:

$$\begin{cases} m \geqslant n \\ k_i \geqslant 3 \\ MD_i(Blp) = MD_i(Brp) = MD_i(Bp) = a \\ MD_j(Blq) = MD_j(Brq) = MD_j(Bq) = b \\ \qquad\qquad 其中\ i > j\ \&\ p < q \\ 若\ MD_i(Bp) = a\ \&\ j \neq i\ \&\ p \neq q \\ \qquad\qquad 则\ MD_j(Bq) \neq a \end{cases}$$

根据上述约束条件,在 Torus 的每一维度内的正、负方向桥接点 B_{lp} 和 B_{rp},沿 Mesh 同一维度 D_i 放置;不同桥接点放置的维度不同(D_i 和 D_j,i 不等于 j)。不同桥接点 B_p 和 B_q 它们之间的转向路径,在 Mesh 内都使用不同链路,从而在使用标准 Torus 的虚通道切换策略时,虚通道依赖图中没有环,因此路由算法无死锁。

3.1.3 性能评测与结果分析

仿真环境配置:本文所有的评测都是基于大规模并行网络模拟器 HiNetSim[9] 开展的,该模拟器可以配置层次化网络,并进行包级别细粒度的网络模拟。本文配置 TMesh 网络中系统域互连网络采用 6×6×6 3D-Torus 网络,片上采用 4×4 Mesh 网络,共包含 216 个处理器,3456 个处理器核心,并参考 IBM Blue Gene/Q 芯片的参数[10,11],设置片间单向链路带宽为 2GB/s,片上网络单向链路带宽25.6GB/s,片上网络与 Router 接口带宽76.8GB/s(单向),Router 的片间聚合双向带宽为24GB/s(3D Torus,所以是 2GB/s×6×2),片上网络单跳转发延迟为集中式 Router 转发延迟的 50%。可以看到,片上网络与 Router 的接口带宽远高于 Router 的交换容量,因此本文限定 16 个处理器核产生的聚合片间注入流量不超过 4GB/s。

本节的评测用于验证无死锁路由算法的正确性,并通过与标准 Torus 网络的对比,初步分析 TMesh 网络的性能。其中标准 Torus 网络采用如图 1(a)所示的传统 CPU 架构。

仿真结果分析:图 8 为均匀随机流量下,使用 LN{0,2,4} + RN{1,3,5} 网络接口节点布局时的片间网络吞吐率-延迟曲线,带黑色圆点曲线为标准 Torus 的吞吐率。在注入率较低的情况下,由于 TMesh 增加了传输过程中的片内跳步数(但在源和目的处理器内的跳步数低于集中式Router架构),因

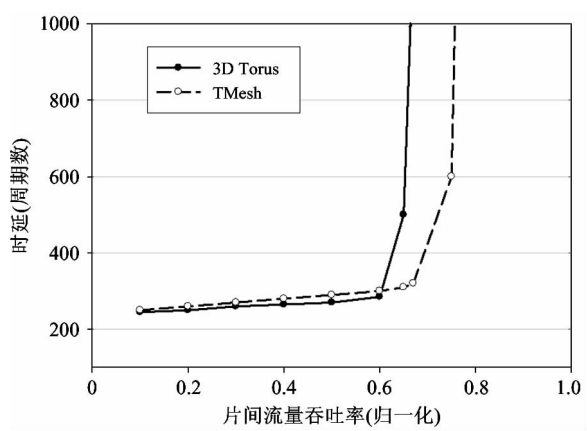


图 8 TMesh 与标准 Torus 的片间网络性能对比

此延迟略高，但随着片间网络流量不断增加，分散的布局使得 TMesh 可以利用片上多路径，降低了流量在片内队头阻塞和碰撞的概率，均衡疏导流量的优势得以发挥，片间通信最大吞吐率相比标准 Torus 提升了 13.4%。

3.2　虚拟 Router 架构

针对第 2.2 节所述网络接口布局问题，本研究从片上网络流量拥塞、全局流量转发性能、以及片上网络流量注入公平性三个方面，提出 6 种不同的虚拟 Router 架构，其核心在于网络接口的布局策略：

（1）中置并列式布局。属于同一个 Torus 维度的两个网络接口节点两两邻接，水平放置，不同 Torus 维度的两个网络接口节点从上到下依次排开，且所有桥接节点位于 Mesh 的中心位置，如图 9 所示。这种策略主要为了降低 Torus 路由跳步数，同时减少流量与出口之间的距离。属于这种策略的方案有 LN{1,5,9} + RN{2,6,10}（如图 9 所示）和 LN{5,1,9} + RN{6,2,10}。另外，这种策略的变种是将同一个 Torus 维度的两个节点垂直放置，不同维度的桥接节点水平排开。属于这种变种的方案有 LN{5,4,6} + RN{9,8,10} 和 LN{4,5,6} + RN{8,9,10}。

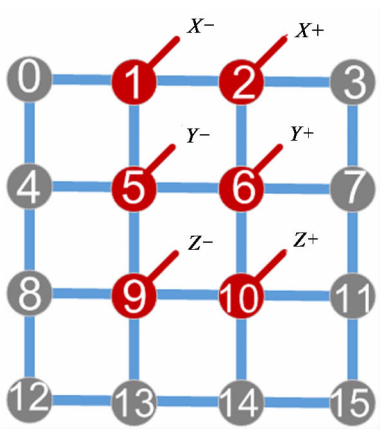


图 9　中置并列式布局

（2）集中式布局。所有网络接口节点按照 Mesh 节点编号顺序依次排开，且集中于 Mesh 上的一个角落，如图 10 所示。这种策略的目的在于减少 Torus 路由的跳步数，同时尽量减轻全局流量对于片内通信的干扰。属于这种策略的方案有 LN{0,2,4} + RN{1,3,5}（如图 10 所示），LN{10,12,14} + RN{11,13,15}，LN{4,6,8} + RN{5,7,9} 等。

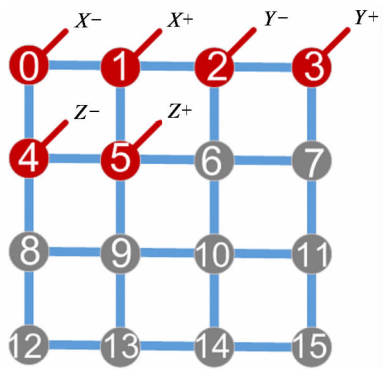


图 10　集中式布局

（3）边缘两列式布局。LN 节点与 RN 节点分布于 Mesh 的两个对边，且沿同一个维度依次排开，如图 11 所示。这种策略的目的在于尽量将网络出口分散开，并追求对 Mesh 内信道的负载均衡。属于这种策略的方案有 LN{0,4,8} + RN{3,7,11}（如图 11 所示）、LN{4,0,8} + RN{7,3,11} 等。

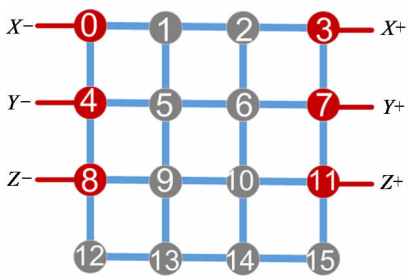


图 11　边缘两列式布局

（4）垂直式布局。属于 Torus 的 X 维的两个网络接口节点垂直放置，Y 和 Z 维度的两个网络接口节点与 X 维节点相邻且平行放置，如图 12 所示。这种方案在缩短流量与网络出口距离的同时，将

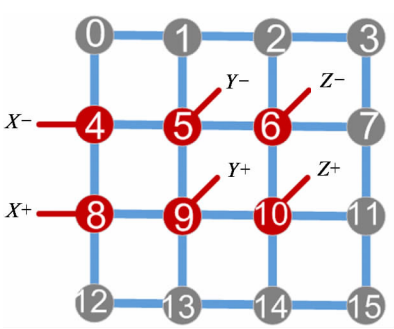


图 12　垂直式布局

Torus 路由的跳步数降低到最低。属于这种策略的方案有 LN{4,9,5} + RN{8,10,6}(如图 12 所示)和 LN{4,5,9} + RN{8,6,10}。

(5) 交叉平行式布局。所有 6 个网络接口节点放置于 Mesh 网络的中央位置,*X* 维度的两个网络接口节点单独位于一行,*Y* 和 *Z* 维度的 4 个网络接口节点位于同一行,属于同一 Torus 维度的网络接口节点两两相邻,且每个维度中有一个节点其 Mesh 维度中的一维与 *X* 维度的节点相同,如图 13 所示。这种策略在减少流量注入点与网络出口距离的同时,考虑 Torus 上从 *X* 维度转向 *Y* 和 *Z* 维度的流量与沿 *Y* 和 *Z* 维度路由的流量相互影响,利用布局将其所使用的链路分离开来。属于这种策略的方案有 LN{9,4,6} + RN{10,5,7}(如图 13 所示)和 LN{9,6,4} + RN{10,7,5}。

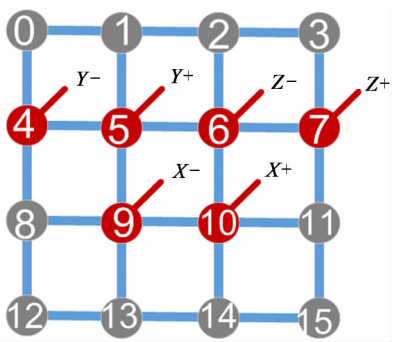


图 13　交叉平行式布局

(6) 翼展式布局。将 *X* 维度的两个网络接口节点放置于 Mesh 中一个维度的两端,然后在同一维度以及相邻维度的同一位置放置 *Y* 和 *Z* 维度的网络接口节点,如图 14 所示。这种策略同样从减少流量源与出口距离的角度出发,且针对 Torus 上的转向流量特别进行优化,使得从 *X* 转到 *Y* 和从 *Y* 转向到 *Z* 的流量经过跳数尽量少,从 *X* 转到 *Z* 的流量使用负载较低的 Mesh 链路。属于这种策略的方案有 LN{4,5,6} + RN{7,9,10}(如图 14 所示),LN{4,6,5} + RN{7,10,9},LN{4,5,9} + RN{7,6,10}和 LN{4,9,5} + RN{7,10,6}等。

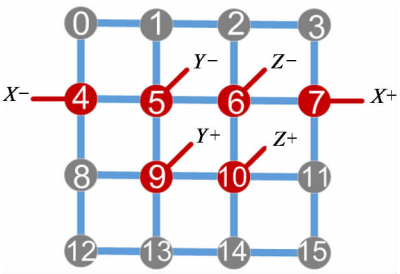


图 14　翼展式布局

图 15 是 7 种不同的布局策略对网络性能的影响。可以发现,实验结果印证了上面的结论。边缘两列式布局方案 LN{8,4,0} + RN{11,7,3}将网络接口节点分布于 Mesh 的两侧,数据包在 Torus 上路由时将在 Mesh 内部经过相当长的路径,因此其路由延迟是所有方案里最高的。同时,也需要注意到,由于这种方案充分利用了 Mesh 内部的链路,负载均衡效果比较好,因此其最大吞吐率也是最高的,Anton2[5] 采用了这种布局方案。

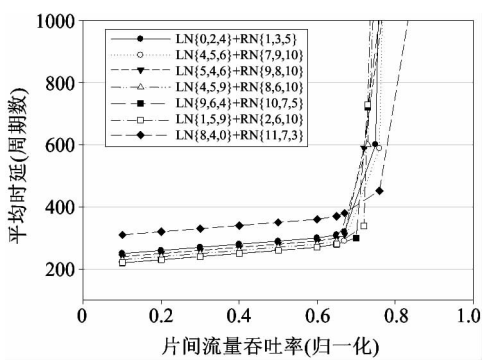


图 15　不同网络接口节点布局下的片间网络吞吐率

表 1 是各布局方案的饱和点延迟(如图 15 所示,饱和点约在吞吐率 60% ~70% 的位置)、最大吞吐率和 Mesh 内平均跳步数。为了探究网络接口节点布局对延迟的影响,表中最后一行给出了按照维序路由的 Mesh 内平均跳数,此跳数即所有可能的路由路径的平均跳数。例如,从 *X* − 节点出发的所有可能的路径,其终点是 *X* +、*Y* −、*Y* +、*Z* −、*Z* + 节点,即所有其他网络接口节点;从 *Y* − 节点出发的所有可能的路径,其终点则是 *Y* +、*Z* −、*Z* +,而从 *Y* − 到 *X* − 节点的路径则是非法的,因为维序路由决定了不可能有从 *Y* 维度转向到 *X* 维度的包,因而在计

算平均跳数时不会被统计。

表 1 各桥接节点布局方案性能

LN	8,4,0	0,2,4	4,5,6	9,6,4	4,5,9	5,4,6	1,5,9
延迟周期数	383.1	341.3	333.9	321	321.2	318.8	315.5
最大吞吐率(%)	84.0	75.9	77.2	78.0	79.6	80.2	74.0
Mesh 内平均跳步数	2.89	1.78	1.78	1.67	1.56	1.56	1.56

从表 1 中可以看出,在典型负载下,维序路由平均跳数与网络延迟成负相关关系,平均跳数越小,网络的延迟越低。同时,通过对网络接口节点进行巧妙摆放,可以更加均衡地利用 Mesh 内的链路,从而提升最高吞吐率。例如布局 LN{1,5,9} + RN{2,6,10}和 LN{5,4,6} + RN{9,8,10},其维序路由平均跳数均为1.56,其最高吞吐率却有 6% 的差别。这说明,信道的负载均衡对最高吞吐率有较高的影响。另外,某些情况下,信道的负载均衡程度甚至会超过维序路由平均跳数,成为影响网络最高吞吐率的第一因素。例如 LN{8,4,0} + RN{11,7,3}布局的维序路由平均跳数高达2.89,最高吞吐率约为 84%,而布局 LN{1,5,9} + RN{2,6,10}的维序路由平均跳数为 1.56,其最高吞吐率反而仅有 74%。因此,对网络接口节点布局时,需要尽量均衡网络接口节点之间的链路负载,以达到更高的吞吐率。

3.3 协同网络参数设置

TMesh 使用片上网络进行片间流量的传输,因此片上和片间转发能力协同是系统性能的关键。真实场景中,片上网络链路带宽通常是片间网络链路带宽的数倍至数十倍(例如 Blue Gene/Q 中是 12.8 倍,Tofu2 中是 4 倍),因此本节沿用第 3.1.3 节中的参数设置,仅改变片上网络链路带宽为 8GB/s、12GB/s、16GB/s、25.6GB/s 进行测试,设置片内流量为 0,即所有处理器核发出的流量都是片间流量,仍然限定单处理器注入的聚合带宽峰值为 4GB/s(单核 2Gb/s)。如图 16 所示,在片上网络链路带宽为 8GB/s 时,片间网络的性能就已经与片上网络链路带宽为 25.6GB/s 时相差无几。因此,虚拟 Router 架构不会对当前片上网络带宽提出更高的需求。

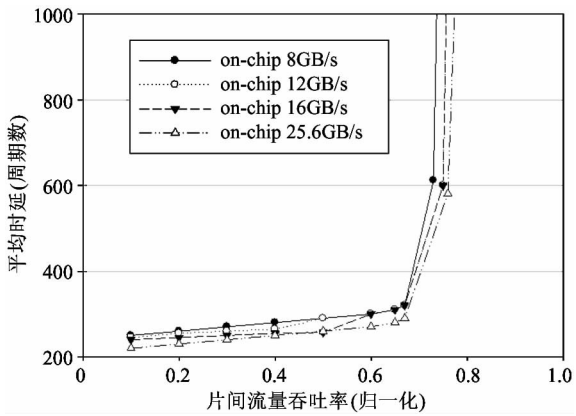


图 16 片上网络带宽对片间流量的影响

进一步地,本文限定片上网络链路带宽为8GB/s,改变片内流量(背景流量),探究片内流量对系统域通信性能的影响。如图 17 所示,在片内流量占据处理器核峰值流量 30% 和 50% 时,片上网络对片间流量仍然是轻载状态,系统域片间网络的通信性能几乎不受影响,仅达到 80% 时,片内流量才导致延迟上的些许差异。因此,片上Mesh网络具有足够的

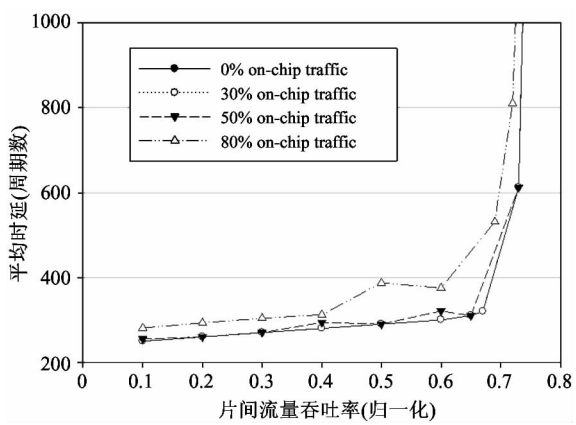


图 17 片内流量对片间网络的影响

交换能力,在进行片上流量转发的同时,支撑系统域全局流量的桥接转发。

4 讨 论

为了满足日益增加的网络规模需求,路由器端口/交换机端口正向高阶网络的方向发展。但如何在处理器内集成高阶路由器就成为一个严峻的问题,而本文的方法将高阶路由器的交换功能与片上网络耦合在一起,避免了专门为高阶路由器实现交换资源的开销。但是协同网络设计面临更加复杂的网络设计选择,本文仅对协同网络的无死锁算法、协同接口布局和协同调参开展了初步的探索,未来将针对更多流量模式进行测试评估,通过对更多具体案例的分析,理清三个关键问题之间的复杂关系,抽象泛化出协同网络的性能模型。

此外,本文探讨的问题也可以归纳为"隐性层次网络"问题。本文探讨的直接网络中,片上网络相当于为系统域网络增加了一个隐性层次网络,使得网络架构的设计变得越来越复杂。然而对于间接网络也存在类似的问题,传统使用单级交叉开关(Crossbar)方式构建方法,受限于芯片资源及后端设计布线等挑战,难以实现高阶的交换芯片。因此目前采用的手段是使用多个小端口的交换芯片通过网络互连构建高阶的交换机,如构成 Cray BlackWidow 系统的 YARC[12] 交换结构。因此不论是直接网络还是间接网络,目前都有网络拓扑的隐性层次增加的问题,这使得热点、死锁、负载均衡等一系列网络设计问题更加复杂。本文提出的片上网络和系统域互连网络协同设计思路同样适应于间接网络的设计。

5 结 论

本文针对直接网络互连维度及其互连的节点片上网络的发展趋势和挑战,提出了片上网络和片间系统域网络协同设计思路,以分布式的虚拟 Router 架构替代传统集中式的 Router 架构,并对协同设计方法的关键问题进行了系统性的探索。通过实例化设计一种 TMesh 网络架构,量化分析并评测验证了本文提出的网络协同设计方法不仅能为片上网络互连的众多核心提供可扩展的、均衡及公平的网络服务,有效提升片间网络性能,还能降低 CPU 集成网络部件的硬件开销。当越来越多的密集计算核心或者交换功能单元、片上网络互连及节点间系统域互连网络接口集成到单个芯片时,本文提出的片上网络和系统域互连网络协同设计方法具有非常重要的可行性和实践指导意义。

参考文献

[1] Chen D, Eisley N A, Heidelberger P, et al. The IBM Blue Gene/Q interconnection fabric[J]. *IEEE Micro*, 2012, 32(1): 32-43

[2] Ajima Y, Inoue T, Hiramoto S, et al. The Tofu Interconnect[J]. *IEEE Micro*, 2012, 32(1): 21-31

[3] Shimizu T. 2014, Fujitsu's new supercomputer delivering the next step in Exascale capability[EB/OL]. https://www.fujitsu.com/global/Images/fujitsu-new-supercomputer-delivering-the-next-step-in-exascale-capability.pdf: Fujitsu, 2014

[4] Sodani A. 2015, Intel® Xeon Phi™ processor "Knights Landing" architectural overview[EB/OL]. https://www.nersc.gov/assets/Uploads/KNL-ISC-2015-Workshop-Keynote.pdf: Nersc, 2015

[5] Towles B, Grossman J P, Greskamp B, et al. Unifying on-chip and inter-node switching within the anton 2 network[C]. In: Proceedings of the 41st International Symposium on Computer Architecture (ISCA'14), Minneapolis, USA, 2014. 1-12

[6] Top500. org. 2017, TOP500 list[EB/OL]. http://top500.org/lists/2017/06: Top500.org, 2017

[7] Dally W J. Virtual-channel flow control[J]. *IEEE Transactions on Parallel and Distributed Systems*, 1992, 3(2): 194-205

[8] Duato J, Yalamanchili S, Ni L. Interconnection Networks: An Engineering Approach[M]. San Francisco: Morgan Kaufmann Publishers, 2003

[9] Fan Z G, Cao Z, Su Y, et al. HiNetSim: a parallel simulator for large-scale hierarchical direct networks[C]. In: Proceedings of the 11th IFIP International Conference on Network and Parallel Computing, Yilan, China,

2014. 120-131

[10] Haring R A, Ohmacht M, Fox T W, et al. The IBM blue gene/Q compute chip[J]. *IEEE Micro*, 2012, 32(1): 48-60

[11] Ohmacht M, Blumrich M, Chiu G L T, et al. Design of the IBM blue gene/Q compute chip[J]. *IBM Journal of Research and Development*, 2013, 57(1): 1-13

[12] Scott S, Abts D, Kim J, et al. The blackwidow high-radix clos network[C]. In: Proceedings of the 33rd Annual International Symposium on Computer Architecture, Boston, USA, 2006. 16-28

Co-design of on-chip networks and system interconnection networks

Liu Xiaoli[*,**], Xun Zhixuan[*,**], Cao Zheng[*], Sun Ninghui[*]

([*] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**] Institute of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100190)

Abstract

It is pointed that to further improve network performance, integrating network devices into processors becomes an important trend, but with the increase of the on-chip processor number and the number of interchip interconnected ports required by high dimensional system interconnection networks, the traditional processor architecture, IBM Blue Gene/Q, for example, will face the problems of scalability, on-chip traffic balancing and intership access fairness, etc. In consideration of this, a method for co-design of on-chip networks and system interconnection networks is proposed, which on one hand, provides balanced and fair network service to processing cores by distributing network interfaces into the on-chip network, and on the other hand, solves the scalability issue and reduces processor implementation cost by avoiding using crossbar in the centralized router. This proposed cooperative design method is systematically investigated from three aspects of collaborative routing, collaborative deployment of network interfaces, and collaborative parameter setting. Also, a case study on the network architecture of TMesh is conducted, and the performance and feasibility of the virtual router architecture are analyzed.

Key words: high performance computing (HPC), on-chip network, system interconnection network, deadlock, router