

## 面向网络延迟均衡性的片上网络交叉开关分配策略\*

王子聪, 陈小文, 郭 阳

(国防科技大学 计算机学院, 湖南 长沙 410073)

**摘要:**针对路由器中的交叉开关分配,提出面向网络延迟均衡性的分配策略,旨在通过更加公平的交叉开关分配策略提升网络延迟均衡性,从而进一步提升系统性能。通过在全系统模拟器上运行 SPEC CPU2006 实验表明,与传统的分离式分配策略和最新的 TS - Router 分配策略相比,延迟标准差分别平均降低了 13.8% 和 3.9%,而最大延迟分别平均降低了 45.6% 和 15.1%。在系统性能方面,相比于 TS - Router,所提分配策略在 IPC 上平均提升了 0.8%。在分离式分配器的基础上完成了硬件实现,并给出了速度、面积和功耗方面的开销评估。

**关键词:**片上网络;交叉开关分配;均衡性

**中图分类号:**TP302 **文献标志码:**A **文章编号:**1001-2486(2018)06-082-07

## Network latency equalization-oriented switch allocation strategy for networks-on-chip

WANG Zicong, CHEN Xiaowen, GUO Yang

(College of Computer, National University of Defense Technology, Changsha 410073, China)

**Abstract:** According to the problem of switch allocation, a network latency equalization-oriented switch allocation strategy was proposed to improve system performance by designing a more fairness switch allocation strategy and improving the degree of network latency equalization. The evaluation of the design with SPEC CPU2006 benchmarks was performed in a full-system simulator. Compared with the canonical separable switch allocator and the TS - Router (recently proposed switch allocator), the experiments show that the approach decreases LSD (latency standard deviation) by 13.8% and 3.9% respectively, as well as ML (maximum latency) by 45.6% and 15.1% respectively. The approach improves system throughput by 0.8% over that of TS - Router. Finally, the design is implemented based on the separable switch allocator, and it can perform the evaluation in speed, area and power.

**Key words:** networks-on-chip; switch allocation; equalization

随着片上多核处理器(Chip MultiProcessor, CMP)集成的处理核数越来越多,支持核间通信的片上互联系统在规模和复杂度上也在持续增长,而片上网络(Networks-on-Chip, NoC)由于具备优良的可扩展性逐渐成为现代 CMP 的主流互连方式。

然而, NoC 规模的不断增长会导致网络中各个节点之间的通信距离和延迟的差异性逐渐变大,距离相近的节点之间相比于距离较远的节点之间在通信上更具优势。随着 NoC 网络直径的不断增长,网络中的最大通信距离和最小通信距离之间的差异持续扩大,造成不同网络报文之间的延迟差异逐渐增大。将网络报文延迟的差异性随着网络规模扩大而不断增长的现象称之为网络延迟不均衡问题。

由于 Cache 容量的需求不断扩大,现代 CMP 通常会基于 NoC 采用非一致 Cache 访问(Non-Uniform Cache Access, NUCA)体系结构来组织最后一级缓存(Last Level Cache, LLC)。在基于 NoC 的 NUCA 结构中, LLC 通常在物理上分布于各个节点,每个节点的 Cache 存储体(Bank)在逻辑上构成一个统一的共享 Cache。在这样的 NUCA 结构中,当处理核发出 Cache 访问请求时,其访问时间由请求处理核所在节点与访问数据所在的 Bank 所在节点的距离有关。当距离较近时,访问时间较小;当访问距离较远的 Bank 时,访问时间较大。然而,随着网络规模的扩大和节点个数的增多, Cache 访问延迟逐渐由网络延迟主导,从而使得网络延迟不均衡问题传导至 Cache 访问

\* 收稿日期:2017-08-06

基金项目:国家自然科学基金资助项目(61502508, 61572025);湖南省自然科学基金资助项目(2015JJ3017)

作者简介:王子聪(1989—),男,河南开封人,博士研究生, E-mail:wangzicong@nudt.edu.cn;

陈小文(通信作者),男,助理研究员,博士, E-mail:xwchen@nudt.edu.cn

延迟上,导致不同的 Cache 访问请求延迟差异增大,造成了 Cache 访问延迟的不均衡。这种情况会引起部分 Cache 访问请求的延迟非常大,从而阻塞发出 Cache 访问请求的处理核的执行进程,成为系统瓶颈并严重影响系统整体性能。

为了提升系统的网络延迟均衡性,希望通过优化路由器设计,使得网络报文之间的延迟差异尽可能地缩小。在路由器结构中,交叉开关位于报文数据通路中的核心<sup>[1]</sup>,通过交叉开关报文被分配并传输至相应的输出端口<sup>[2]</sup>。因此,交叉开关分配器是调度报文并决定报文延迟的重要模块。本文旨在设计一种面向网络延迟均衡性的交叉开关分配策略(network latency Equalization-oriented Switch Allocation, ESA),并以此缩小不同网络报文之间的延迟差异,从而改善网络报文的延迟均衡性,提升系统性能。

## 1 相关工作

本文从网络延迟均衡性问题出发,致力于通过优化片上网络路由器的交叉开关分配策略缓解均衡性问题对于系统性能的影响。因此,本文将从片上结构优化技术和片上网络交叉开关分配策略两个方面论述与本文相关的工作。

### 1.1 片上结构优化技术

在分布式存储并行系统中,远程数据访问模式对于程序的运行效率具有较大影响。Hsieh 等<sup>[13]</sup>在数据迁移(data migration)、线程迁移(thread migration)等远程数据访问模式之外提出了计算迁移(computation migration)机制,通过迁移部分线程及其相关状态至请求数据所在处理器节点,减少了迁移时的开销。Michaud 等<sup>[4]</sup>发现当多核系统中运行单个串行程序时,只有程序所在核的 Cache 得到利用,造成其他核上的 Cache 空间资源的浪费。为了提升 Cache 空间资源利用效率,文献<sup>[4]</sup>提出了 affinity 程序迁移算法,用于支持程序的运行时动态迁移,从而能够利用总的片上 Cache 资源。Chakraborty 等<sup>[5]</sup>观察到在并行程序中不同线程之间存在大量的计算代码重叠,而这些冗余的计算代码却需要共同竞争私有资源(Cache 空间、分支预测器等),导致私有资源的利用率低。因此,Chakraborty 等提出了计算传播(Computation Spreading, CSP)模型,该模型可以将线程中的计算代码分成不同的计算片段(computation fragment),然后将来自不同线程的相似计算片段分配到同一个核上,而将同一个线程中的不相似计算片段分配到不同核上,每个核

可以通过保存计算状态动态地分时执行计算片段,从而提升了系统资源利用率。

### 1.2 片上网络交叉开关分配策略

NoC 的网络性能主要取决于路由器的性能,因此许多工作都是围绕设计高效的路由器结构展开<sup>[6-7]</sup>的。在路由器设计中,交叉开关分配器的设计很大程度上决定了路由器的性能,因而交叉开关分配器的设计占据重要地位。在之前的研究工作中,人们针对交叉开关分配器在结构以及算法上都进行了大量的研究。分离式分配策略(Separable Switch Allocation, SSA)是众多的分配器的结构基础,即许多分配策略是基于分离式策略设计的。传统的分离式分配器通常在每个输入端口和输出端口分别设置一个基于轮循(round-robin)的仲裁器,通过两轮仲裁完成一个周期的分配决策<sup>[2]</sup>。一种对于传统的分离式分配器的改进设计是 iSLIP 分配器<sup>[8]</sup>。在 iSLIP 分配器中,只有那些最终匹配成功的通路上的仲裁器才会更新仲裁器的优先级,而对于其他的仲裁器则保持优先级不变。SSA 分配器和 iSLIP 分配器虽然在硬件实现上较为高效,但却牺牲了匹配效率,同时也并未关注到网络延迟的均衡性问题。文献<sup>[9]</sup>提出了一种基于时序的分配策略 TS-Router。在 TS-Router 中,某一周期内的分配决策不仅取决于当前的报文请求,还会通过前瞻的方式预测下一周期的报文请求,从而在更宽的时序范围内实现更高的匹配效率。然而,这种分配策略却忽视了公平性问题,在一定程度上加剧了“饿死”问题,恶化了网络延迟的均衡性。以上所提到的这些分配策略通常是面向匹配效率而设计的,因此,本文将称之为面向匹配效率的交叉开关分配策略。

综上所述,考虑到网络延迟均衡性对于系统性能的重要影响,在路由器结构尤其是交叉开关分配器的设计上亟须一种面向网络延迟均衡性的分配策略,从而能够通过改善网络延迟均衡性来提升系统性能。

## 2 面向网络延迟均衡性的交叉开关分配器设计

### 2.1 分配策略设计

ESA 主要基于 SSA 而设计。分配的过程包括两组仲裁器,一组在输入端口,另一组在输出端口。本文以一个包含 5 个端口和 4 个 VC 的路由器结构为例介绍 ESA 分配策略的设计。图 1 所示的是 ESA 分配策略示意图。将每个输

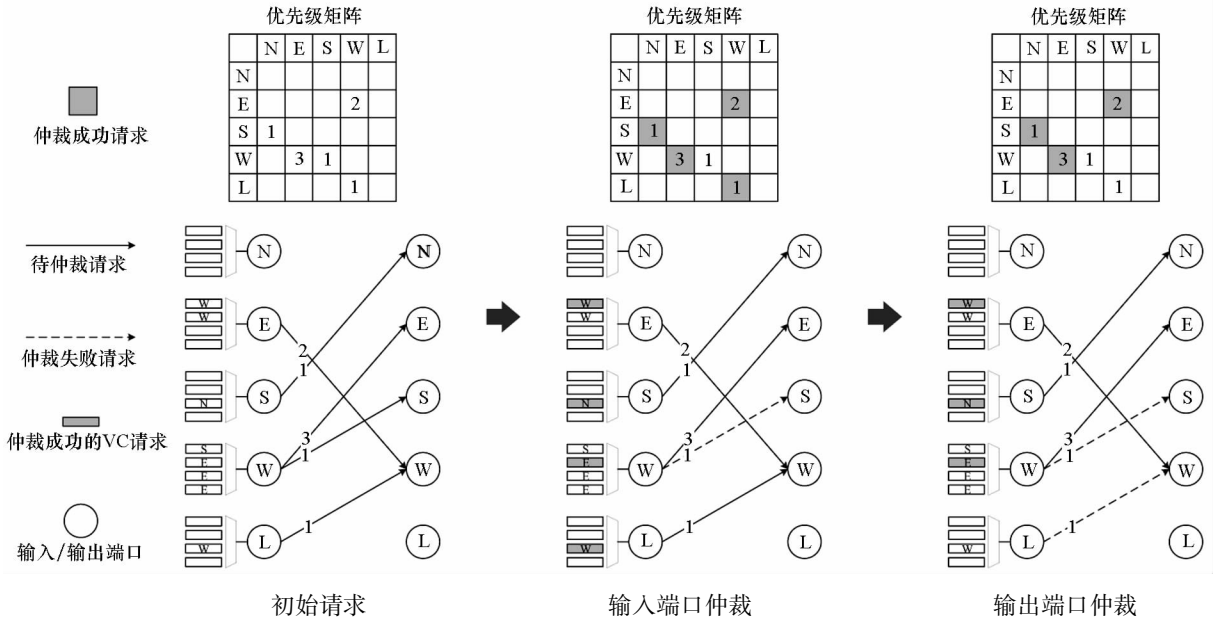


图 1 ESA 分配策略示意图

Fig. 1 Procedure of ESA strategy

入端口到各个输出端口的请求 VC 数量定义为该输入 - 输出端口的公平因子。公平因子越大,表明该输入 - 输出请求上的请求报文越多,从而越有可能造成该条通路上的拥塞并引发较高的网络报文延迟。因此,从缓解拥塞、减少大网络延迟报文数量和增强网络延迟均衡性的角度应当优先响应具有较高的公平因子的输入 - 输出请求。由于每个输入端口到输出端口都会有一个相应的公平因子,因此这些公平因子集合起来就构成了输入端口到输出端口的优先级矩阵。后续的仲裁过程主要依据该优先级矩阵对报文进行仲裁。

仲裁过程主要分为两个阶段:输入端口仲裁和输出端口仲裁。在图 1 所示的初始请求中,注意在输入端口 W 中有 3 个 VC 请求了输出端口 E,有 1 个 VC 请求了输出端口 S;在输入端口 E 和 L 中分别各有 2 个和 1 个 VC 同时请求了输出端口 W。在第一阶段的输入端口仲裁中,需要完成两部分的仲裁工作:一是对来自同一个输入端口的请求中,仲裁出一个输出端口请求方向;二是在请求该输出端口的多个(单个)VC 请求中,仲裁出一个获胜的 VC 请求。在图 1 的示例中,对于输入端口 W,由于输入端口 W 对输出端口 E 的公平因子大于对输出端口 S 的公平因子,因此请求输出端口 E 的 3 个 VC 请求仲裁成功。同时,由于交叉开关中一个通路在单个周期只能通过一个报文,因此需要在 3 个 VC 请求中再仲裁出一个获胜的 VC,对于 VC 请求之间的仲裁,本文采用的是基于轮循的仲裁策略,因此 3 个 VC 请求

中仲裁成功的是 VC1。

在输出端口仲裁过程中,需要在每个输出端口中对输入端口请求进行仲裁。由于同时请求输出端口 W 的输入端口 E 和 L 的公平因子分别为 2 和 1,因此输入端口 E 的请求仲裁成功,从而完成整个交叉开关分配过程。

图 2 所示的是 SSA 分配策略的结构图。假设路由器有  $V$  个 VC,  $P$  个输入/输出端口。在 SSA 分配器中,输入端口  $i$  的各个 VC 的请求信号组表示为  $r_i^v[\cdot]$ ,该信号组经过一个  $V:1$  的轮循仲裁器选出一个获胜的 VC 请求信号,并用  $g_i^v[\cdot]$  表示输入端口  $i$  的经过仲裁后的 VC 请求信号。 $g_i^v[\cdot]$  经过译码器转换为获胜 VC 的输出端口请求信号,并传向输出端口进行第二轮仲裁。在每个输出端口  $j$  会收到来自各个输入端口的请求信号,  $P$  个请求信号经过  $P:1$  的轮循仲裁器选出一个获胜的输入端口请求信号,并作为输出端口  $j$  的输入端口仲裁结果,在图中用  $g_j^p[\cdot]$  表示。 $g_j^p[\cdot]$

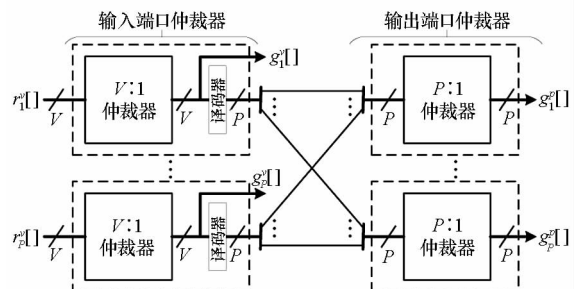


图 2 SSA 分配器结构图

Fig. 2 Structure of SSA

和  $g_j^p[\ ]$  分别指明了输入端口  $i$  的获胜 VC 和输出端口  $j$  的获胜输入端口,从而共同决定了最终的仲裁结果。

为了能够支持 ESA 分配策略,需要在传统的 SSA 分配器结构的基础上进行修改设计。图 3 所示的是 ESA 分配器结构图。与 SSA 分配器不同的是,每个输入端口增加了一个额外的  $P:1$  仲裁器。同时,在输入端口的仲裁流程方面,输入端口  $i$  的请求信号  $r_i^v[\ ]$  会首先经过译码器转换为输出端口请求信号,并通过一个大小为  $P:1$  的基于公平因子的仲裁器选出公平因子最大(即 VC 请求数量最多)的输出端口请求,而  $V:1$  的轮循仲裁器的作用是选择出输入端口  $i$  的获胜 VC,并传向输出端口进行第二轮仲裁。在第二轮仲裁中,每个输出端口中大小为  $P:1$  的基于公平因子的仲裁器会仲裁出具有最大公平因子的输入端口请求,从而保证每个输出端口产生一个获胜的输入端口请求。

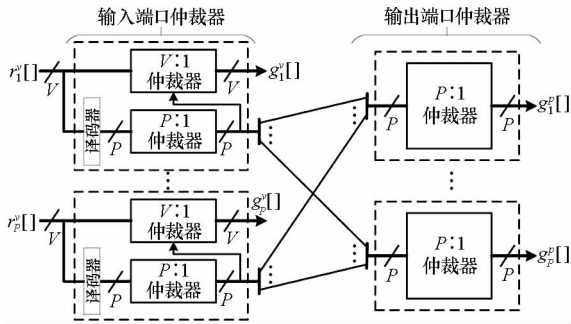


图 3 ESA 分配器结构图  
Fig.3 Structure of ESA

### 2.2 基于公平因子的仲裁器设计

从上述分配过程可以看出,ESA 分配器中包含三次仲裁,其中有两次基于公平因子的仲裁过程。因此,需要设计相应的仲裁器来支持在输入和输出端口中的两次基于公平因子的仲裁。图 4 所示的是基于公平因子的仲裁器结构图。该仲裁器在轮循仲裁器的基础上加入了公平因子比较逻辑,包括公平因子计算模块以及比较器树模块。每个 VC 请求信号首先会通过公平因子计算模块得到该输入-输出的公平因子,这些公平因子输入到比较器树模块,从而选出那些拥有最大公平因子的请求信号。由于可能存在多个请求拥有相同的最大公平因子,因此需要在这些请求信号中通过轮循仲裁器再次选出一个获胜的请求信号,这样就保证了每个端口产生一个获胜的请求信号。另外,通常情况下请求某个输出端口的 VC 数量(即公平因子)会远小于每个端口的最大 VC

请求数,因此为了降低硬件开销,我们可以将公平因子的大小约束在  $W$  位( $0 < W < \lceil \log_2 V \rceil$ ),当某个请求的公平因子超过  $W$  位所能表示的最大数值时,用  $W$  位能够表示的上限值代替,这样在损失一定的精度的情况下大大降低硬件开销以及关键路径延迟。

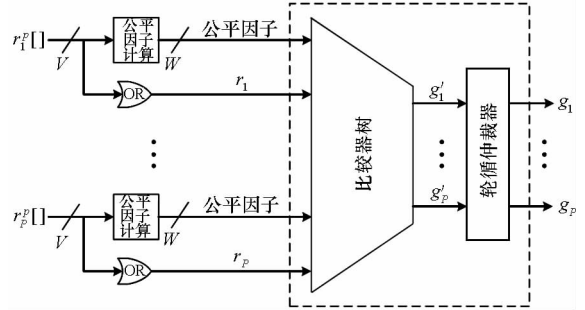


图 4 基于公平因子的仲裁器结构图  
Fig.4 Structure of fairness factor arbiter

### 2.3 防饿死机制

当某些输出端口请求持续保持较大的公平因子时,可能会导致该输出端口请求长时间获胜,从而发生其他输出端口请求饿死的情况,使得那些公平因子较小的输出端口成为新的拥塞和大延迟报文的产生源。因此,在基于公平因子的仲裁器的基础上,将停顿周期引入到仲裁过程中,在每个公平因子计算模块中添加一个停顿周期寄存器(Stall Cycle Register, SCR),该寄存器主要用于记录每个输出端口请求的停顿周期,并在计算公平因子时将该寄存器中的值计算在内。需要注意的是,SCR 只有满足如下两个条件才会在某个周期内自增:一是 SCR 所记录的输出端口在该周期内的确有请求(即请求的 VC 数不为 0);二是输出端口请求在输入端口和输出端口的仲裁过程中仲裁失败(即未在分配过程中取得开关通路)。另外,当输出端口请求仲裁成功后,SCR 会复位为 0。

根据上述讨论,假设第  $N$  个时钟周期,在输入端口  $i$  对输出端口  $j$  的公平因子计算模块中,输入端口  $i$  请求输出端口  $j$  的 VC 数为  $n_{ij}^{(N)}$ ,SCR 值为  $v_{ij}^{(N)}$ ,公平因子为  $f_{ij}^{(N)}$ 。在第  $N$  个时钟周期输入端口  $i$  对输出端口  $j$  的公平因子为:

$$f_{ij}^{(N)} = n_{ij}^{(N)} + v_{ij}^{(N)} \quad (1)$$

因此,在第  $N+1$  个时钟周期 SCR 的更新规则为:

$$v_{ij}^{(N+1)} = \begin{cases} 0 & \text{仲裁成功} \\ v_{ij}^{(N)} & n_{ij}^{(N)} = 0 \\ v_{ij}^{(N)} + 1 & \text{仲裁失败, } n_{ij}^{(N)} \neq 0 \end{cases} \quad (2)$$

### 3 实验评估

#### 3.1 实验配置

实验评估主要包括性能模拟和硬件开销评估两个方面。在性能模拟实验中,本文采用基于 Garnet2.0<sup>[10]</sup>的全系统体系结构模拟器 GEM5<sup>[11]</sup>开展了相关实验,将 ESA、SSA 和 TS - Router 三种交叉开关分配策略进行了比较。实验配置参数如表 1 所示。性能模拟包括两部分内容。首先,将 GEM5 模拟器切换至网络模拟模式,即网络中的每个节点的处理核只作为网络流量的注入点,而不模拟运行实际的 CPU 指令,各个节点按照指定的合成流量模型向网络注入报文。其次,将 GEM5 模拟器切换至全系统模式,在每个处理核上运行真实的 CPU 指令。另外,硬件开销评估基于开源的片上网络路由器 RTL Router<sup>[12]</sup>,在该路由器结构的基础上,本文进行了修改并利用 Synopsys 的 Design Compiler 工具进行了综合。

表 1 实验配置参数

Tab.1 Configuration of performance simulation

配置参数	参数值
处理核数	64
指令集	X86
Cache 一致性协议	MESI
网络拓扑结构	二维 8 × 8 网格网络
路由算法	XY 维序路由策略
切片大小	128 bit
Cache 块大小	128 B
一级指令/数据 Cache 容量	16 KB
二级 Cache Bank 容量	128 KB

#### 3.2 合成流量模拟实验

合成流量模拟实验是在 GEM5 的网络模拟模式下进行的。本文选择了最为常用的 Uniform-random 合成流量,并选取了两个指标来考察网络延迟均衡性能表现,即延迟标准差 (Latency Standard Deviation, LSD) 和最大延迟 (Maximum Latency, ML)。其中,LSD 反映了报文延迟的均衡程度,LSD 越小表明报文延迟之间的差异性越小,且均衡性越好。ML 反映了所有报文延迟中的最大值,ML 越小表明报文延迟对系统产生的瓶颈作用越小。从图 5 和图 6 中可以看出,相比于 SSA 和 TS - Router,ESA 在各个注入率下都有较低的 LSD 和 ML,表明了 ESA 能够有效提升网络报文延迟的均衡性,并降低了网络报文延迟对

于系统性能的瓶颈作用。

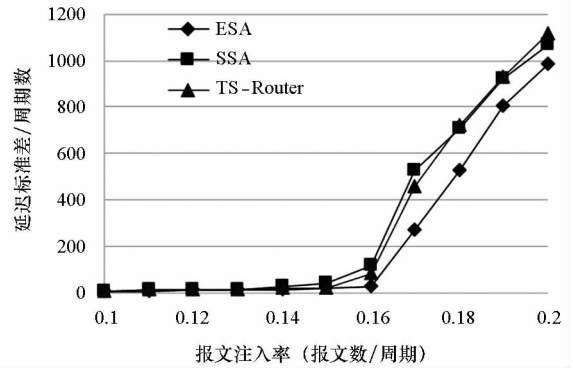


图 5 LSD 随注入率变化图

Fig.5 LSD comparison with different injection rate

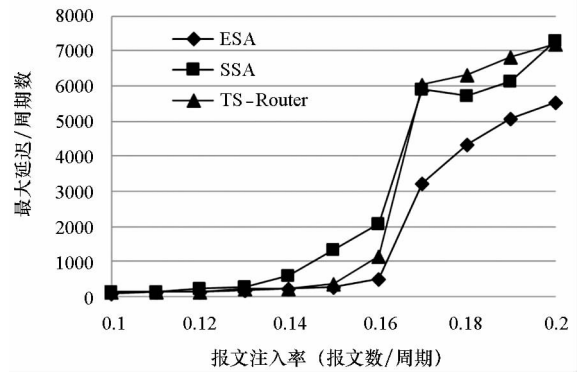


图 6 ML 随注入率变化图

Fig.6 ML comparison with different injection rate

为了进一步验证 ESA 的延迟均衡效果,在 Uniform-random 流量模型的饱和注入率下(即 0.16)收集了各个分配周期策略下 10 000 个周期内的网络报文延迟数据。图 7 显示的是网络报文延迟的累积分布函数图。由图 7 中各个分配策略下的报文比例可知,对于相同的网络报文延迟界限,ESA 相比于 SSA 和 TS - Router 小于该界限的网络报文更少,从而表明 ESA 具有更少的大延迟。另外,图 8 显示的是网络报文延迟的概率密

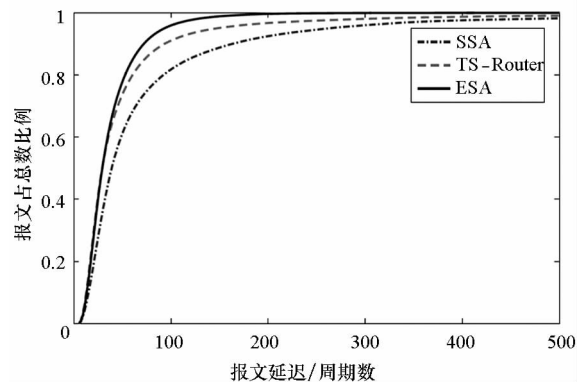


图 7 网络延迟累积分布函数图

Fig.7 Cumulative distribution function of the network latencies

度分布函数图。从图 8 中可以观察到,相比于 SSA 和 TS - Router, ESA 的报文延迟分布更加集中,促使部分高延迟报文分布转变为低延迟报文。因此,ESA 能够使得报文延迟分布更加集中,进而实现均衡网络报文延迟的目标。

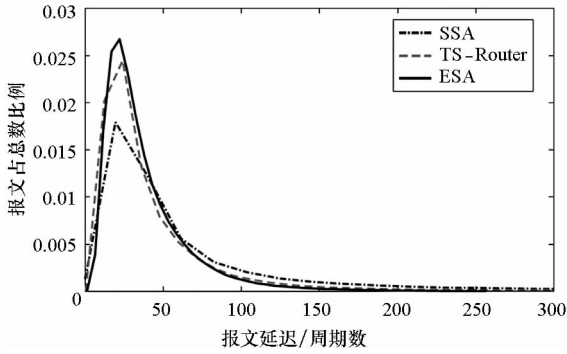


图 8 网络延迟概率密度分布图

Fig. 8 Probability density function of the network latencies

### 3.3 全系统模拟实验

从 SPEC CPU2006 基准测试程序集中选取了 11 个典型的应用程序进行全系统模拟实验。针对每一个应用程序,本文在每个处理核上运行一个该程序的实例,即一次模拟同时运行 64 个程序实例。收集了程序运行过程中的网络报文延迟,并分别统计了网络报文延迟的 LSD、ML,如图 9 和图 10 所示。从图中可以看出,ESA 在 LSD 和 ML 上都有显著改进,相比于 SSA 和 TS - Router, ESA 在 LSD 上分别平均降低了 13.8% 和 3.9%,在 ML 上分别平均降低了 45.6% 和 15.1%。其中,LSD 的降低表明网络报文延迟之间的差异性缩小,而 ML 的降低则表明制约系统性能的瓶颈延迟变小,进而能够促进系统性能得到提升。图 11 中显示的是 11 个应用程序的模拟实验中 ESA 相比于 TS - Router 的归一化单位周期执行指令数 (Instruction Per Cycle, IPC)。从图 11 中可以看出,ESA 在系统性能上得到了一定提升,在 11 个应用程序的模拟实验中,IPC 相比于 TS -

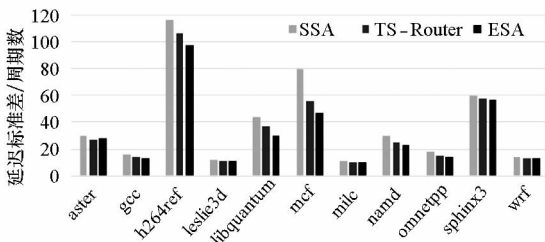


图 9 全系统模拟实验下 LSD 比较图

Fig. 9 LSD comparison under full system experiments

Router 在总体上平均提升了 0.8%。

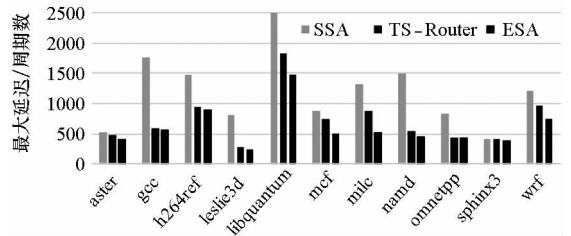


图 10 全系统模拟实验下 ML 比较图

Fig. 10 ML comparison under full system experiments

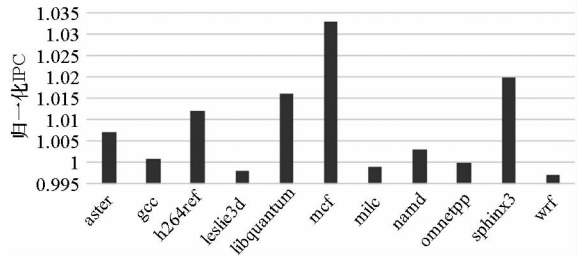


图 11 全系统模拟实验下归一化 IPC 统计图

Fig. 11 Normalized IPC under full system experiments

### 3.4 硬件开销评估

基于图 3 所示的 ESA 分配器结构以及图 4 所示的基于公平因子的仲裁器结构图,给出了如图 12 所示的 ESA 硬件实现结构图。基于开源的片上网络路由器 RTL Router 实现了 ESA 的硬件结构,并用 Synopsys 的 Design Compiler 工具在 TSMC 的 65 nm 工艺库下 (1.0 V, 25 °C) 进行了综合。实验结果表明,由于 ESA 在 SSA 的基础上用基于公平因子的仲裁器替代了部分轮循仲裁器,因此延长了关键路径,使得最高频率从 SSA 的 500 MHz 下降到了 333 MHz。尽管频率有一定程度的降低,但可以一定的额外硬件开销来流水化 ESA,从而保证运行频率。另外,ESA 相比于 SSA

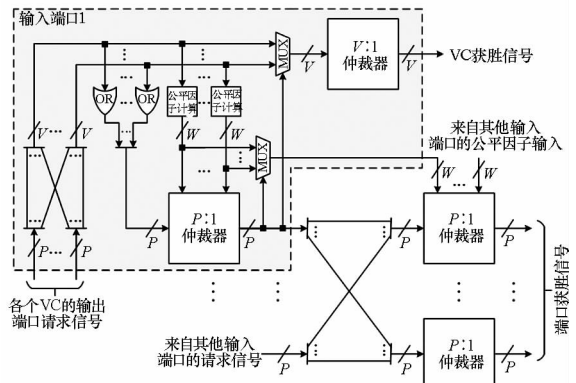


图 12 ESA 分配器硬件实现结构图

Fig. 12 Implementation structure of ESA

大约需要额外的  $47 \mu\text{m} \times 47 \mu\text{m}$  面积开销。同时在功耗方面, ESA 相比于 SSA 增加了约 0.291 mW 的额外功耗。相比于 TS - Router 的面积和功耗的额外开销 ( $61 \mu\text{m} \times 61 \mu\text{m}$ , 0.325 mW), ESA 通过采用面向均衡性的设计目标实现了更低的硬件开销。

## 4 结论

随着芯片网络规模的不断扩大,网络延迟不均衡的现象愈加突出,从而导致更多的大延迟网络报文,严重制约系统性能。本文在路由器的交叉开关分配设计领域提出了一种新的策略,该策略能够有效缓解网络延迟不均衡问题,与传统的分离式分配策略和最新的 TS - Router 分配策略相比,在网络性能和系统性能上都得到了一定的提升。在今后的工作中,仍将持续关注网络延迟不均衡现象,同时从均衡网络报文路径长度方面解决该问题,进一步挖掘由此带来的性能提升。

## 参考文献 (References)

- [1] Jerger N D E, Peh L S. On-chip networks[M]. Synthesis Lectures on Computer Architecture, 2009.
- [2] Dally W, Towles B. Principles and practices of interconnection networks [M]. USA: Morgan Kaufmann Publishers, 2003.
- [3] Hsieh W C, Wang P, Weihl W E. Computation migration: enhancing locality for distributed-memory parallel systems[C]// Proceedings of the fourth ACM SIGPLAN Symposium on PRINCIPLES and Practice of Parallel Programming, 1993: 239 - 248.
- [4] Michaud P. Exploiting the cache capacity of a single-chip multi-core processor with execution migration [C]// Proceedings of 10th International Symposium on High Performance Computer Architecture, 2004.
- [5] Chakraborty K, Wells P M, Sohi G S. Computation spreading: employing hardware migration to specialize CMP cores on-the-fly [J]. ACM SIGOPS Operating Systems Review, 2006, 40(5): 283 - 292.
- [6] Mullins R, West A, Moore S. Low-latency virtual-channel routers for on-chip networks[J]. ACM SIGARCH Computer Architecture News, 2004, 32(2): 188.
- [7] Park D, Das R, Nicopoulos C, et al. Design of a dynamic priority-based fast path architecture for on-chip interconnects[C]// Proceedings of 15th Annual IEEE Symposium on High-Performance Interconnects, 2007: 15 - 20.
- [8] Mckeown N. The iSLIP scheduling algorithm for input-queued switches[J]. IEEE/ACM Transactions on Networking, 2002, 7(2): 188 - 201.
- [9] Chang Y Y, Huang S C, Poremba M, et al. TS-Router: on maximizing the quality-of-allocation in the on-chip network[C]// Proceedings of 19th International Symposium on High Performance Computer Architecture, 2013: 390 - 399.
- [10] Agarwal N, Krishna T, Peh L S, et al. GARNET: a detailed on-chip network model inside a full-system simulator [C]// Proceedings of International Symposium on Performance Analysis of Systems and Software, 2009: 33 - 42.
- [11] Binkert N, Beckmann B, Black G, et al. The gem5 simulator[J]. ACM SIGARCH Computer Architecture News, 2011, 39(2): 1 - 7.
- [12] Becker D U. Efficient microarchitecture for network-on-chip routers[Z]. Stanford Digital Repository, 2012.