片上网络延时差异对存储系统公平性的影响及对策

刘 胜 陈书明 尹亚明 陈胜刚 谷会涛 陈小文 王耀华

(国防科学技术大学计算机学院 长沙 410073)

摘 要 研究了在基于片上网络(Network on Chip,NoC)结构的单芯片多处理器(Chip Multiple Processors,CMPs)中,访存请求的 NoC 延时差异对存储系统的公平性带来的影响.针对该问题进行了理论分析、抽象,并构建试验模型,从网络规模、报文比例等 4 个方面对造成访存请求的 NoC 延时差异的原因进行了讨论.最后提出了一种基于片上网络延时的存储器访问调度方法(Scheduling Based on NoC Latency,SBNL),与传统的方法相比,能够将 NoC 延时差异对访存请求公平性的影响降低 20%左右,并带来 15.7%的执行效率提升.

关键词 片上网络;延时差异;存储;公平性;调度

中图法分类号 TP302 **DOI 号**: 10.3724/SP.J.1016.2011.01500

The Effect of NoC Latency Difference on the Fairness of Memory Systems and a Strategy

LIU Sheng CHEN Shu-Ming YIN Ya-Ming CHEN Sheng-Gang GU Hui-Tao
CHEN Xiao-Wen WANG Yao-Hua

(*School of Computer Science, National University of Defense Technology, Changsha* 410073)

Abstract In the CMPs (Chip Multiple Processors) based on NoC (Network on Chip), the NoC latency difference of memory requests would have great effect on the fairness of memory systems. The theoretical and experimental models were constructed and the causations of NoC latency difference were discussed from the scale of network, proportion of pockets and other two aspects. The SBNL (Scheduling Based on NoC Latency) method was proposed, which could reduce the NoC latency difference's side effect on the memory fairness by about 20% and bring 15.7% increment of the execution efficiency, compared with the traditional methods.

Keywords NoC; latency difference; memory; fairness; scheduling

1 引 言

CMPs 是一种公认的能够有效利用单片集成超过 10 亿个晶体管能力的体系结构.目前的 CMPs 已逐渐从多核(multi-core,4～16 cores)向众核(many-core,16～256 cores)发展,如 Intel 公司的 8 核的 Xeon 处理器[1]、STI 联盟的 9 核 Cell[2]、SUN 公司的 8 核 UltraSPARC T2[3]、Tilera 公司 64 核的 TILEPro64 和 100 核的 TILE GM-100[4-5]、Intel 公司的 80 核 Teraflop[6]等.当处理器中核的数目到达一定规模时,传统的共享总线模式在带宽、功耗、延时及全局同步等问题上均遇到了难以逾越的障碍.NoC 技术以成本低廉的点对点分组互连取代传统

收稿日期:2010-06-17;最终修改稿收到日期:2011-04-11.本课题得到国家"八六三"高技术研究发展计划重大项目(2007AA01Z108)、"核高基"重大专项(2009ZX01034-001-001-006)、国家"八六三"高技术研究发展计划(2009AA011704)和国家自然科学基金(61070036)资助.**刘 胜**,男,1984 年生,博士研究生,研究方向为高性能微处理器.E-mail: liusheng83@gmail.com. **陈书明**,男,1961 年生,教授,博士生导师,研究领域为高性能计算机体系结构和 VLSI 设计. **尹亚明**,男,1979 年生,博士研究生,研究方向为高性能微处理器. **陈胜刚**,男,1981 年生,博士研究生,研究方向为高性能微处理器. **谷会涛**,男,1980 年生,博士研究生,研究方向为高性能微处理器. **陈小文**,男,1982 年生,博士研究生,研究方向为高性能微处理器. **王耀华**,男,1985 年生,博士研究生,研究方向为高性能微处理器.

的总线模式,能够较好地解决上述问题[7].

"存储墙"是伴随着现代处理器的出现就一直存在的问题[8].在 CMPs 中,这一问题依然存在甚至更加突出.CMPs 系统中的 PE 一般通过 NoC 进行信息交换并共享存储资源.在所有共享的存储资源中,片外存储资源将对系统的性能产生重要影响[9].由于受到芯片管脚的约束和制造工艺的限制,片外存储器控制器的数目、位置和带宽均为受限的,CMPs 相对于传统的多处理机系统其存储资源更加珍贵,因而需要更加公平合理地在线程之间进行分配.片外存储资源不公平分配会导致以下问题[10-12]:(1)由于某些线程被不公平地赋予较高的优先级,会使另外某些线程的访存请求等待过长,从而影响系统的整体性能;(2)会使系统软件级程序(如操作系统或虚拟机)中的基于线程优先级调度的策略失效;(3)减少了应用程序的性能可预测性.

存储系统的公平性从本质上讲是提高访存请求的服务质量(Quality of Service,QoS)从而提高系统整体性能的问题.随着 NoC 规模的扩大,访存请求的 NoC 延时差异已逐渐成为影响访存公平性的一个重要因素,如在 64 结点的 CMPs 中不同 PE 访存请求的 NoC 延时差异已达 180 个时钟周期(详情见第 4 节),而一般情况下外部存储器的访问延时只有几十到几百拍.已有的存储器调度算法[10-12],只考虑到了不同线程的访存请求在存储器控制器内部的相互影响,在高效利用存储带宽的前提下通过策略保证不同线程的访存公平性,但是忽略了 CMPs 中 NoC 延时差异对访存公平性的影响.本文将对这一因素进行深入分析,并提出解决方案.

本文的主要贡献如下:(1)首次提出 NoC 延时差异将成为影响外部存储器访问公平性的重要因素;(2)对 NoC 延时差异影响访存公平性的现象进行了分析建模并构建了模拟平台,分析了影响外部存储系统公平性的 4 个因素;(3)提出了 SBNL(Scheduling Based on Network Latency)的存储器访问调度方法,该方法与传统的调度方法相比能够有效地减少 NoC 延时差异对访存请求公平性的影响,并带来一定的系统效率提升.

2　相关研究工作

已有很多人员对外部存储系统的公平性及 QoS 进行了研究.关于在存储器控制器(Memory Controller,MC)上采取策略调节不同线程的访存公平性的观点最早出现在 Nesbit 等人[10]的文章,作者基于网络公平队列的概念提出了 NFQ(Network Fair Queuing)调度算法,其 QoS 目标为:一个线程 i 如果被分配了系统存储带宽的 φ_i 部分,那么其运行速度不应比一个相同的线程单独运行在一个频率是原有存储系统频率 φ_i 的存储系统中慢.Mutlu 等人[11-12]认为 NFQ 算法在某些情况下并不能确保公平,并且不同的线程同时运行时,获取存储带宽并不意味着一定能取得相对应的性能.因此提出了一种新的 STFM(Stall-Time Fair Memory)调度算法,该方法的 QoS 定义为:如果 CMPs 中运行的线程具有相同的优先级,那么其存储器相关的减速(在 MC 中由于其它线程的影响)应该相同.已有的存储器调度算法,均是在采用 FR-FCFS(First-Ready First-Come-First-Service)调度方法的基础上增加了 QoS 方面的考虑.这些方法仅考虑到了不同线程的访存请求在存储器控制器内部的相互影响,在高效利用存储带宽的前提下通过策略保证不同线程的访存公平性,但是忽略了 CMPs 中 NoC 延时差异对访存公平性的影响.

将存储器系统的特点和 NoC 控制结合起来研究的工作在最近得到了较多的关注.Dutt 提出了面向存储器的 NoC 设计方法学[13],认为随着芯片集成规模的扩大,片上资源主要被存储资源占据,同时许多存储敏感的应用被映射到芯片上,因而必须在 NoC 设计的早期就考虑存储器相关的事宜(如划分层次和访问结构等).Jang 等人[14]设计了一种考虑 SDRAM 特性的 NoC 路由器,该路由器赋予报文不同的优先级以减少 SDRAM 的 Bank 冲突和数据总线竞争问题,从而提高了 SDRAM 系统的总体性能.Yuan 等人[15]观察到在 GP-GPU(General Purpose-Graphic Processor Unit)体系结构中,每一个核发出的请求本身具有较高的 SDRAM 行访问局部性,但是经过片上网络传输之后行访问局部性遭到了破坏.作者通过在 NoC 设计中实现的"Hold Grant"和"Row-Matching Hold Grant"两种仲裁策略保证了这种行访问局部性,从而采用简单的 FCFS 调度策略就可以得到与 FR-FCFS 相近的效果.

关于 CMPs 片上存储系统特别是片上 Cache 系统的访问公平性问题的研究已经相当广泛.Fedorova 提出了 Cache 公平的线程调度算法[16],可以有效地解决共享 L2 中的线程冲突问题.Chang[17]提出了协同 Cache(Cooperative Cache)机制,通过在私有 L2 基础上引入 Cache 间干净数据的搬

移,可以在不增加存储延时的基础上隔离线程冲突.

此外,Abts 等人[18]提出了在大量的 PE、少量的 MC 的 NoC 中如何放置 MC 的问题,给出了能够使最大通道负载(Maximum Channel Load)最小的 MC 放置方法:菱形放置法.同时在提出了 CDR(Class-based Deterministic Routing)路由算法,能够对处理器-存储器型冲突(Processor-Memory Traffic)进行负载平衡.该文献和本文讨论的出发点不同,但其研究内容却很有参考价值.

据我们所知,目前还没有关于片上网络延时的差异对外部系统的访存公平性的影响这方面的研究成果发表,因而迫切需要对这一问题在理论和试验方面进行分析建模并提出较好的解决方案.

3 片上网络延时差异的现象、原因及参数化建模

在基于 NoC 结构的 CMPs 中,外部存储器的请求起始于最后一级 Cache(Last Level Cache,LLC)的缺失请求.如果是读请求,则在缺失状态处理寄存器组(Miss Status Handling Register,MSHR)记录之后,通过 NoC 传输至目标 MC,在 MC 经过仲裁选择并向 SDRAM 发送访存命令,从 SDRAM 返回的数据再次进入 NoC 中并最终返回给 LLC,同时完成 MSHR 的状态更新.写请求的处理与上述过程类似,只不过前者从 SDRAM 向 LLC 返回的是请求的数据,后者返回的是写确认信息.因而我们能够得到式(1),任何一个访问外部存储器的请求延时可以分为三部分:访存请求发向目标 SDRAM 在 NoC 中的延时 T_{req}、访存请求在目标 SDRAM 中的处理延时 T_{mem} 和从目标 SDRAM 得到的数据或写确认信息返回给请求源结点在 NoC 中的延时 T_{return}.在这里我们忽略了访存请求从网络报文解析为请求信号和返回数打包为网络报文所花费的时间,这样能够简化下述分析过程,并且对问题实质影响不大.

$$T_{total}=T_{req}+T_{mem}+T_{return} \tag{1}$$

式(1)中的 T_{req} 和 T_{return} 均可视为 NoC 中的报文延时,在采用虫孔路由的 NoC 中,报文的无冲突延时包含两部分:报文头传输延时和后续微片(flit)连续传输延时.如式(2)所示[19],Γ_{1p} 是一个报文在 NoC 中传输所花费的时间;$h_{i,j}$ 是结点 i 到结点 j 之间的跳步数;t_c 为无拥塞时一个微片通过一个开关和一条链路所需要的时间;L 为报文的长度;b 为链路带宽.

$$\Gamma_{1p}(N)=h_{i,j}\times t_c+\left\lceil\frac{L}{b}\right\rceil \tag{2}$$

在考虑网络冲突的 NoC 中,引入 t_w 这一参数,t_w 表示存在拥塞时报文头在开关节点处平均等待时间,式(2)将变为式(3):

$$\Gamma_{1p}(N)=h_{i,j}\times(t_c+t_w)+\left\lceil\frac{L}{b}\right\rceil \tag{3}$$

而式(1)中的 T_{mem} 由文献[20]可知,主要由存储器带宽受限导致的停顿 T_{m_bw} 和存储器本身的延时 $T_{m\_latency}$ 两方面的因素构成,如式(4)所示.

$$T_{mem}=T_{m\_bw}+T_{m\_latency} \tag{4}$$

将式(4)和式(3)代入到式(1)中,得到在基于 NoC 结构的 CMPs 中,外部存储器的访问延时,如式(5)所示.

$$T_{total}=2\times h_{i,j}\times(t_c+t_w)+\left\lceil\frac{L_{req}}{b}\right\rceil+T_{m\_bw}+T_{m\_latency}+\left\lceil\frac{L_{return}}{b}\right\rceil \tag{5}$$

在确定的 NoC 参数及报文协议下,式(5)中的参数 t_c、L_{req}、L_{return} 和 b 为固定常数,并且由 SDRAM 的特性可知[21],$T_{m\_latency}$ 的变化范围相对较小(如在采用开页策略的 SDRAM 中行命中与行冲突的延时一般相差十几拍左右).因此不同访存请求的访存延时的差异主要体现在 $h_{i,j}$、t_w 和 T_{m_bw} 这 3 个参数,其中 $h_{i,j}$ 主要由请求结点与 MC 的相对位置、网络规模、MC 在 NoC 中的位置等因数决定,t_w 主要由应用程序的通信、访存特性(如通信流量模型、报文注入率、访存报文所占比重等)决定,T_{m_bw} 主要由存储器带宽、存储器调度策略等决定.显然在基于 NoC 结构的 CMPs 中,外部存储器的访问延时由多种因素共同决定,具有动态特征,因而不能够精确预测.本文第 4 节构建模拟平台,对上述构成 T_{total} 差异的原因进行综合分析.

由式(3)可知,片上网络延时由 $h_{i,j}$ 和 t_w 决定,前者具有确定性,后者具有随机性.我们考虑图 1 所示的情况:在 8×8 的 2 维 mesh 结构下,采用 X-Y 维序路由,MC 放置在芯片的上下两端,结点 a 和结点 b 分别访问同一外存空间(MC1 控制的一段外存空间),由于结点 a 和 MC1 的距离比结点 b 和 MC1 的距离大($h_{i,j\_a}=15$,$h_{i,j\_b}=1$),则结点 a 的访存请求在 NoC 中的延时要远远大于结点 b 的访存请求在 NoC 中的延时.即对应式(1)中 $T_{req\_a}\gg T_{req\_b}$ 且 $T_{return\_a}\gg T_{return\_b}$,现有的存储器调度策略一般采用 FCFS 或修改的 FR-FCFS 调度方法,这些方法保证了不同线程的 T_{m_bw} 相差不大,即 $T_{mem\_a}\approx T_{mem\_b}$.由

式(1)易知 $T_{total\_a} \gg T_{total\_b}$,即不同的结点对同一外存空间的访问延时差异较大(这种差异根据第 4 节的模拟在 256 结点下可达 400 拍以上).同样的道理我们可以推出同一节点对不同外部存储空间的访问延时也具有很大的差异.

由式(5)可以看出在 NoC 规模、MC 数目及位置固定的情况下,不同结点访存的 $h_{i,j}$ 必然存在较大差异,可供我们调节的主要参数是 T_{m_bw},我们可以采用一定的策略使距离 MC 较远的结点在竞争访存通道时具有较高的优先级,即当某个请求的 $h_{i,j}$ 值较大时,使其 T_{m_bw} 值较小,从而避免或减弱 NoC 的延时差异对访存公平性的影响,这也是本文第 5 节提出的 SBNL 调度方法的由来.

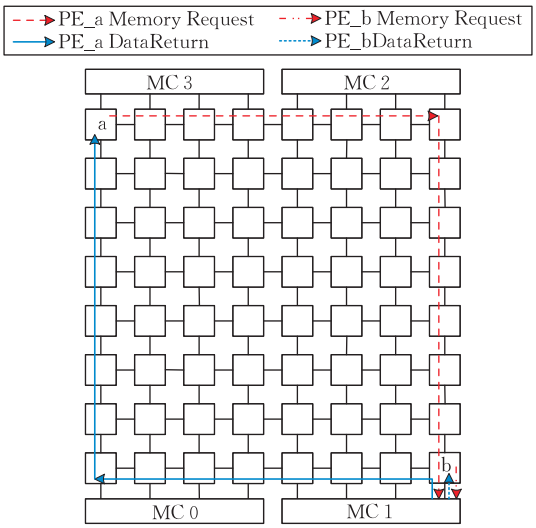


图 1　不同结点对同一外存空间访问的 NoC 差异性示意图

4　片上网络延时差异对存储系统的公平性的影响因素分析

4.1　模拟平台的搭建

本文构建了一个节拍精确的 NoC+MC 模拟平台,并采用合成的流量模型对影响存储系统公平性的各种因素进行分析.该模拟平台主要由一个 NoC 模拟器和多个 SDRAM 的 MC 模型组合而成.能够全面模拟不同 PE 对不同存储空间的访存请求在 NoC 上同时存在,并经过仲裁进而访问 MC,然后返回数据的过程.NoC 的主要参数如表 1 所示.在模拟时我们采用了开环模拟的方法,通过 Bernoulli 过程向每个结点注入报文[19],在模拟器经过预热之后统计每个访存请求花费的节拍数,然后再进一步处理.

表 1　NoC 的主要参数

参数名称	参数值
拓扑结构	2D mesh
PE 的数目	4,16,64 或 256
MC 的数目	1,2,4 或 8
路由延时	1 cycle
路由器之间的线延时	1 cycle
交换策略	虫孔交换
通道缓冲数目	8 flits/channel
虚通道数目	4
路由算法	*X*-*Y* 确定性维序路由
报文大小	普通报文:1～4 flits 访存报文:请求 2 flits 返回 4 flits

NoC+MC 模拟平台中的 MC 的主要参数如表 2 所示,每个 MC 固定连接 4 个 PE,拥有一个存储器通道,控制 8 个 SDRAM Banks 和 256 MB 的空间.SDRAM 的时序参数模型如表 3 所示,该模型从 Micron 公司的 SDRAM 参数手册抽象而来,能够精确地模拟存储器主要的时序约束,如存储器命令的延时、不同存储器命令之间的最小间隔、刷新周期等.

表 2　MC 的主要参数

参数名称	参数值
通道数目	1 channel/MC
Bank 数目	8 banks/channel
Bank 存储空间	32 MB
页策略	开页策略
Buffer 大小	1 KB

表 3　SDRAM 的主要时序参数[21]

参数名称	参数值	参数名称	参数值
t_{RCD}	5 cycle	t_{RC}	23 cycles
t_{CAS}	5 cycle	t_{RRD}	3 cycles
t_{BL}	4 cycle	t_{WR}	4 cycles
t_{RAS}	18 cycles	t_{RFC}	51 cycles
t_{RP}	5 cycles		

4.2　评价函数

对 CMPs 片外存储系统的公平性进行评测并不是一件容易的事情,已有的研究工作从不同的角度定义了存储系统的 QoS 目标[10-11],这些 QoS 目标也可以当成 CMPs 片外存储系统的公平性的评价函数.然而已存在的 QoS 目标都是从系统的角度出发,其本身受许多其它因素的影响(如 Cache 抖动等),并且不利于数学建模.因此本文直接从式(1)定义的 T_{total} 出发,通过对 T_{total} 的不同组成部分进行数学统计,进而对 CMPs 片外存储系统的公平性进行评测.

将一段时间内访存请求的总延时、访存请求在 MC 和存储器中的延时和访存请求在 NoC 中的延时分别定义为集合 A、B 和 C,且假设一共记录了 N

个访存请求，即 $A=\{T_{total\_i}\mid i=1,2,\cdots,N\}$，$B=\{T_{mem\_i}\mid i=1,2,\cdots,N\}$，$C=\{T_{NoC\_i}\mid i=1,2,\cdots,N\}$. 本文采用式(6)～(8) 3 个函数即平均访存延时 $T_{average}$、延时均方差 LSD(Latency Standard Deviation)和 NoC 延时非均匀性比例 UFR_{NoC} 来进行分析. 其中 $T_{average}$ 表示所有访存请求的平均延时；LSD 衡量了集合 A 中的所有访存延时与平均延时的偏离程度，即访存请求的公平性；UFR_{NoC} 表示了 NoC 延时的差异占访存请求非公平性的比例，UFR_{NoC} 的范围在 0 与 1 之间，其值越接近于 1 说明 NoC 延时的差异对外部存储器的公平性影响越严重.

$$T_{average}=\frac{\sum_{i=1}^{N}T_{total\_i}}{N} \tag{6}$$

$$LSD=\sqrt{\frac{\sum_{i=1}^{N}(T_{total\_i}-T_{average})^2}{N}} \tag{7}$$

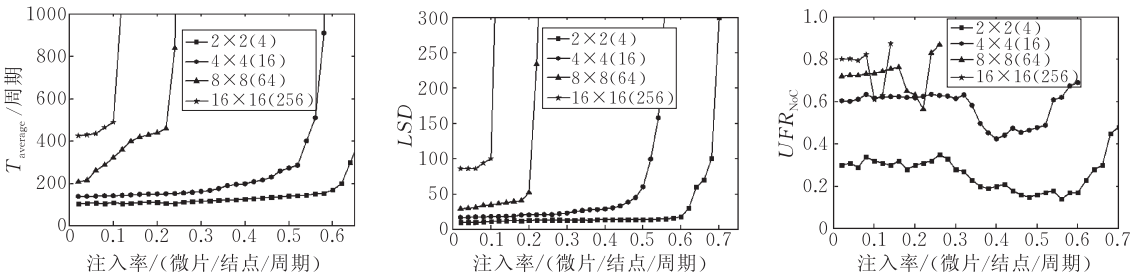
$$UFR_{NoC}=\frac{\text{Standard Deviation}(C)}{\text{Standard Deviation}(C)+\text{Standard Deviation}(B)} \tag{8}$$

4.3 影响访存请求公平性的因素分析

本节分析了网络规模、MC 的位置、报文注入率、报文比例这 4 种因素对外部存储系统公平性的影响. 由于前三者概念明确，这里主要讨论报文比例这一因素的由来. 将应用程序中不同线程之间的片上报文(包片上远程共享 Cache 的访问、Cache 协议的维护等)定义为普通报文，将访问外部存储器的报文(包括外存数据请求和数据返回等)定义为访存报文，普通报文与访存报文的比值称为报文比例.

报文比例的范围，在目前的文献中尚没有明确的结论. 我们可以进行如下定性分析，决定普通报文与访存报文的比例的最重要因素是片上 LLC(假设为二级 Cache，且不考虑片上存储器为便签式存储器的情况)的共享情况. 如果 L2 完全私有，则 NoC 中的报文将全部是访问外部存储器的报文；如果 L2 是分布式共享的，那么普通报文与访存报文的比例取决于远程 L2 和片外存储器的访问比率(和应用程序本身的特征相关)以及所采用的 Cache 一致性方案. 根据文献[17]，在分布式共享 L2 中，远程 L2 和片外存储器的访问比例主要在 1∶1 到 43∶1 范围之间. 因而下文论述中我们分析报文比例从全部是访存报文到 50∶1 这一范围对访存请求公平性的影响，并且以 30∶1 作为一个经验值进行讨论.

我们分别模拟了 2×2、4×4、8×8 和 16×16 的 2D Mesh 结构在不同注入率下，结点直接发送报文的情况(普通报文与访存报文的比例为 30∶1，MC 采用 FCFS 方法调度). 在模拟平台预热(经过 10 万次报文传输)后每个结点向不同的 MC 发送 1 万次报文，记录所有的访存请求的延时并进行处理，得到图 2 所示结果.



(a) $T_{average}$ (b) LSD (c) UFR_{NoC}

图 2 不同网络规模下的访存公平性分析

如图 2(a)所示，在相同的报文注入率下，网络规模越大，每个访存请求的访存延时也越大. 在相同规模的网络中，每个访存请求的延时随着报文注入率的变高而变大. 如图 2(b)所示，在相同的报文注入率下，访存请求的均方差随着网络规模的增加而变大，即访存请求延时随着网络规模的扩大而变得越来越分散. 同时，在注入率比较低时，访存请求的均方差和变化范围都较小，而在中等注入率和高注入率下，访存请求的均方差比较大. 如图 2(c)所示，尽管 NoC 延时差异在访存请求非公平性中的比例存在波动(在中等注入率时存在下降)，但还是占据一定的分量，特别是 64、256 结点时都超过了 50%. 因而能够得出，随着网络规模的扩大，网络负载的增加，NoC 延时差异已逐渐成为影响访存公平性的一个重要因素，必须予以考虑.

在已经商品化的基于 NoC 的众核 CMPs 中[5-6]，MC 均放置在芯片的上下两端(如图 1 所示). 文献[15,18]提出了 MC 的菱形放置方法(如

图 3 所示黑色的 PE 或 router 拥有外部存储器控制器),本文用 NoC+MC 模拟平台对 MC 上下端放置和 MC 菱形放置分别进行模拟(NoC 规模为 8×8,MC 采用 FCFS 调度策略,普通报文与访存报文的比例为 30∶1),得到图 4 所示的结果.

由图 4 可知 MC 菱形放置法比上下端放置法拥有更高的报文吞吐率,这与文献[20]分析结果一致,此外 MC 菱形放置法能够在一定程度上减少访存请求的延时,提高访存请求延时的公平性,因此如果忽略这种放置方法对 NoC 均匀性和可扩展性带来的问题,MC 菱形放置法是一种不错的选择.

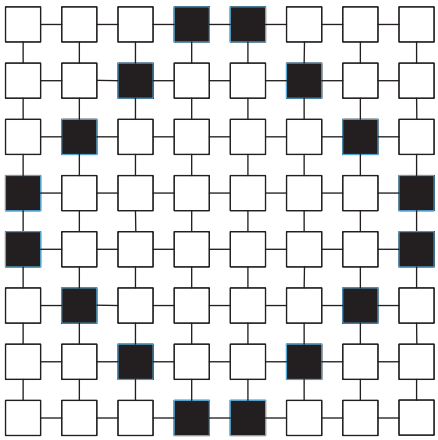
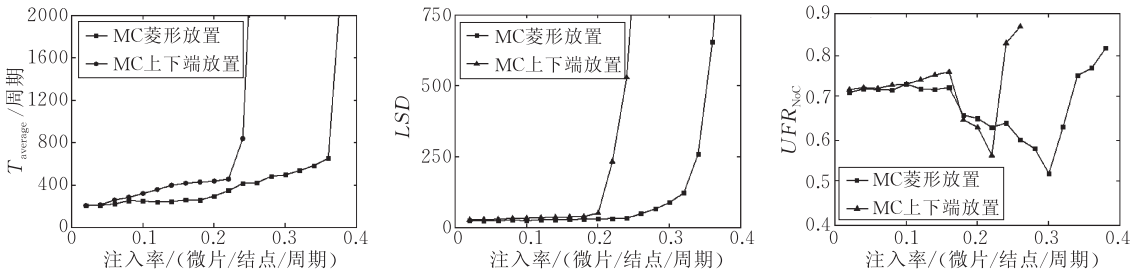


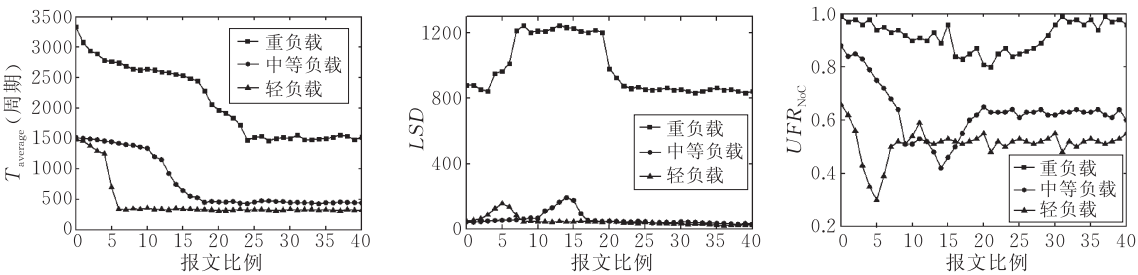
图 3　MC 菱形放置法



(a) $T_{average}$　(b) LSD　(c) UFR_{NoC}

图 4　MC 上下端放置与 MC 菱形放置分析结果对比

报文比例和应用程序相关,是影响访存请求公平性的一个重要因素. 在 64 结点的 NoC 平台下(MC 上下端放置,FCFS 调度),分别在重负载(注入率 0.26)、中等负载(注入率 0.18)、轻负载(注入率 0.06)采用不同的报文比例进行模拟,得出如图 5 所示结果. 可知,报文负载越重,访存请求的 $T_{average}$、LSD 和 UNR_{NoC} 均越大. 在相同的报文负载下,报文比例越小(即访存报文越多),访存请求的平均延时越大,在报文比例大于某一区间之后,访存请求平均延时变小,访存请求的均方差也变小. NoC 延时差异在访存请求非公平性中的比例随着报文比例的变化存在最小值,这个最小值与网络负载相关.



(a) $T_{average}$　(b) LSD　(c) UFR_{NoC}

图 5　不同报文比例下的访存公平性分析

5　SBNL 访存调度方法

通过第 3 节的定性分析和第 4 节的定量模拟,我们发现在 MC 采用上下端放置的芯片布局方法中,NoC 延时差异这一因素对外存访问公平性影响较大,并且随着网络规模、报文注入率的提高而变得日趋严重,因而迫切需要研究人员提出较好的解决方案.

本节我们提出 SBNL 访存调度方法,这种调度方法从不同的线程访问外部存储器的优先级入手,通过在每个 MC 中增加 NoC 延时预估表,赋予距离 MC 较远的请求较高的访存优先级,同时兼顾访存效率,能够对访存公平性和系统性能进行有效的改

善,并且对现有的软硬件结构改动较小.

5.1 主要方法

(1)当访存请求进入 MC 时,由请求源节点坐标查找对应的 NoC 预估延时,作为初始的优先级;当存储器通道处理一个访存请求时,通道中的其它请求的优先级加 t(根据通道中的请求是行命中、行关闭还是行冲突,t 进行相应的变化);

(2)设 P_{max} 为当前通道队列中的请求的最高优先级,将 P_{max} 与参考优先级 P_{ref} 进行比较,若 $P_{max}<P_{ref}$ 则采用 FR-FCFS 调度方法;若 $P_{max}\geq P_{ref}$ 则跳入(3);

(3)具有 P_{max} 优先级的请求优先处理.

SBNL 调度方法是一种充分考虑 NoC 延时差异的存储器调度方法,通过在 MC 的入口处设立 NoC 延时预估表,按照访存请求的源节点在 NoC 延时预估表获取相应的延时预估值.这样在存储器调度时若发现某个线程的请求到达 MC 之前已经花费了过多的时间,则调高该请求的优先级.NoC 延时预估表和 P_{ref} 可由操作系统或程序员根据网络规模、应用程序的特性等提前配置,具有较强的灵活性.

5.2 参数设置依据

在 5.1 节阐述的 SBNL 调度方法中,每个 PE_i 在 MC_j 中的 NoC 预估值由平均网络跳步数、网络负载权重、线程公平因子三方面因素决定,其中平均网络跳步数在一定 NoC 规模和确定性路由策略下是固定的;网络负载权重和应用程序的报文负载相关,操作系统或程序员可以根据不同的应用设置不同的网络负载权重;在之前的论述中我们只考虑应用中不同的线程具有相同的访存优先级的情况,这里增加了线程公平因子这一因素,操作系统或程序员可以通过改变某个 PE 的 NoC 预估值来调整其对应线程的访存优先级.t 的值主要受限于两方面的因素:(1)外部存储器的本身参数.不同的存储模型具有不同的行命中、行关闭、行冲突的访问节拍数(如在我们选取的 DRAM 模型[21]中上述节拍分别为 9、14、18);(2)MC 与外部存储器的频率比值.P_{ref} 的值取决于操作系统或程序员对线程的访存请求延时超出平均访存延时的容忍程度,P_{ref} 越小表示应用程序对外存系统的访问公平性要求越严格,当 P_{ref} 无限大时,SBNL 调度方法就转换成了传统的 FR-FCFS 调度方法.

5.3 实现开销

由上述方法可知,SBNL 方法主要的硬件实现开销包括 NoC 延时预估表、参考优先级 P_{ref} 寄存器、加法器、比较器等.假设网络规模为 $N\times N$,NoC 延时预估表的每一项有 16 bit,MC 为 M 个,每个 MC 中包含一个同时容纳 16 个访存请求的通道,则主要的硬件开销包括 $2M\times N^2$ 个字节的存储空间、$16\times M$ 个 16 位的加法器和 $15\times M$ 个 16 位的比较器,硬件开销不大.

5.4 性能分析

我们将采用以下两种方法分析所提出的 SBNL 调度方法相比于传统的 FR-FCFS 方法在公平性和效率方面的提升.

我们选取 $\Delta LSD\%=\frac{LSD_{\text{FR-FCFS}}-LSD_{\text{SBNL}}}{LSD_{\text{FR-FCFS}}}$ 作为 SBNL 调度方法相比传统的方法在提高系统访问外存公平性的评价参数.分别在 NoC+MC 模拟平台为 2×2、4×4、8×8、16×16 网络规模下,MC 上下端放置、中等报文注入率(在 NoC 为 2×2、4×4、8×8、16×16 规模下,其值分别为采用 0.35、0.25、0.18 和 0.06)和报文比例为 30∶1 的情况下,使每个结点向其它结点均匀发送报文.其中 SBNL 调度方法中的 NoC 延时预估表设置为 $8\times h_{i,j}$,假定 MC 与外部存储器的频率为 4∶1(即根据行命中、行关闭还是行冲突,t 分别设置为 36、56 和 72),假定操作系统或程序员对线程的访存请求延时超出平均访存延时的容忍程度为访存平均延时的 6 倍(则由图 2(a)得:在 2×2、4×4、8×8、16×16 规模下,P_{ref} 的值分别为 720、1000、2100 和 2520).我们统计了不同的网络规模下的 $LSD_{\text{FR-FCFS}}$、LSD_{SBNL} 及 $\Delta LSD\%$ 的值如表 4 所示.由表 4 可知,SBNL 调度方法相比传统的方法,在不同的网络规模下均能显著地降低访问外存请求的延时均方差,提高了访问外存请求的公平性,同时随着网络规模的扩大,SBNL 调度方法对访存公平性的提升效果愈明显.

表 4 不同网络规模下 SBNL 方法和 FR-FCFS 方法的 *LSD* 值对比

网络规模	$LSD_{\text{FR-FCFS}}$	LSD_{SBNL}	$\Delta LSD/\%$
2×2	13.2	12.5	5.0
4×4	25.4	22.0	13.4
8×8	40.9	32.7	20.0
16×16	86.7	65.9	24.0

同时我们给出在网络规模为 8×8 下,不同的报文注入率和报文比例下 $\Delta LSD\%$ 的详细值(SBNL 调度方法的延时预估表值随报文注入率增加而加大,其余参数不变),如表 5 所示.从表 5 中可以得出

如下结论:在网络规模为 8×8 时,SBNL 调度方法均能够有效地降低访存请求的均方差(平均在 20% 左右,最大达到 36.5%),提高系统访存的公平性.随着报文注入率的增加,SBNL 方法对访存公平性的提升愈明显,这是因为报文注入率的增加将会引起更多的网络拥塞,使 NoC 延时的不均匀性更加显著,SBNL 方法能够发挥较大的作用.当报文中包含的访存报文较少时,SBNL 方法对访存公平性的提升较小,这是因为同一时刻 MC 通道中包含的访存请求个数有限,SBNL 方法发挥的余地有限.

表 5　64 结点下 SBNL 与 FR-FCFS 方法的 $\Delta LSD\%$ 值

报文比例	ΔLSD/%			
	注入率=0.26	注入率=0.18	注入率=0.12	注入率=0.06
全部	36.5	29.8	14.1	28.8
5∶1	36.2	28.3	20.3	19.7
10∶1	25.8	22.1	18.6	15.0
15∶1	22.5	23.5	13.4	11.5
20∶1	24.3	20.0	12.7	11.3
25∶1	15.2	14.1	11.9	10.2
30∶1	10.1	8.8	6.5	3.4
35∶1	9.7	8.5	6.5	5.7
40∶1	7.3	6.2	6.3	4.9

此外,为了评估 SBNL 方法对系统全局性能的提升,本文在网络规模为 8×8 的 NoC+MC 平台上进行固定负载情况下线程运行情况的试验.在平台预热之后,为每个 PE 安排了 5000 个访存请求(同一个 PE 的访存请求拥有不同的行局部性,在数据返回之前,每个 PE 允许最多发送 4 个访存请求),在系统运行时,若某个 PE 共接受的访存数据的数目达到 5000,则认为该 PE 的工作完成,不再发送新的请求.在报文比例为 30∶1,中等负载情况下,分别运行 SBNL 调度方法和 FR-FCFS 方法,统计了在不同的时间段(以 500 个时钟周期为采样单位)运行结束的 PE 的数目.

如图 6 所示,采用 SBNL 调度方法时,在系统运行到 17000 节拍时,有 6 个 PE 完成了 5000 个访存请求的既定任务,而在 21500 节拍时,64 个 PE 均完成了既定的访存任务;而采用 FR-FCFS 调度方法,在系统运行到 19500 节拍时,第一个 PE 才完成了既定的访存任务,在 26000 节拍时,全部 PE 才完成了既定的访存任务.可见,相比 FR-FCFS 方法,SBNL 方法能够显著地减少定额工作负载的平均完成时间和最后完成时间(分别为 12.8%和 15.7%).

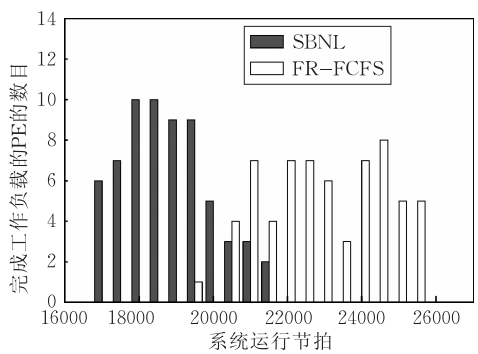


图 6　64 结点下固定工作负载 PE 完成的节拍分布图

5.5　与已有的 NFQ 和 STFM 方法比较

NFQ 方法[10] 和 STFM 方法[11] 均没有考虑 NoC 延时的差异.如 NFQ 方法以线程访存请求到达 MC 的时间作为访存请求虚拟运行时间的开始,这显然是不准确的,在基于 NoC 的系统中,这种方法将会完全忽略访存请求的 NoC 延时;STFM 调度方法中的 T_{share} 直接由 PE 计算并立即传递到 MC 中,这在基于 NoC 的 CMPs 结构中是不可能实现的.本文提出的方法充分考虑 NoC 延时的差异,并且能够与 NFS 和 STFM 方法相结合,因而具有比较明显的优势.

6　总结及进一步的研究

NoC 延时差异逐渐成为影响 CMPs 中不同线程访问外部存储器公平性的一个重要因素,本文对这一问题进行了建模并构建了模拟平台,从 4 个方面分析了对外部存储系统公平性的影响,提出了 SBNL 的存储器访问调度方法.该方法与传统的调度方法相比,能够有效地减少 NoC 延时差异对访存请求公平性的影响.下一步的工作是在系统级模拟平台中寻求 SBNL 方法的优化及改进措施.

参　考　文　献

[1] Rusu S, Tam S et al. A 45nm 8-core enterprise Xeon processor. IEEE Journal of Solid Circuits, 2010, 45(1): 7-14

[2] Hofstee P H. All about the cell processor//Proceedings of the IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips VIII). 2005

[3] Shah M, Barreh J et al. UltraSPARCT2: A highly-threaded, power-efficient, SPARC SoC//Proceedings of the IEEE Asian Solid-State Circuit Conference. Jeju, 2007: 22-25

[4] Agarwal A. On-chip interconnection architecture of the tile processor. IEEE Micro, 2007, 27(5): 15-31

[5] TILERA. Tile-GXTM Processor Family Product Brief. http://www.tilera.com/products/processor.php

[6] An 80-tile 1.28TFLOPS network-on-chip in 65nm CMOS//Proceedings of the IEEE International Solid-State Circuit Conference. San Francisco, 2007: 98-589

[7] Benini L, Micheli G D. Networks on chips: A new SoC paradigm. IEEE Transactions on Computers, 2002, 35(1): 70-78

[8] Wulf W A, McKee S A. Hitting the memory wall: Implications of obvious. Computer Architecture News, 1995, 23(1): 20-24

[9] Burger D, Goodman J R et al. Memory bandwidth limitations of future microprocessors//Proceedings of the International Symposium on Computer Architecture. New York, NY, USA, 1996: 77-78

[10] Nesbit K J, Aggarwal N, Laudon J, Smith J E. Fair queuing memory systems//Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture. Washington, DC, USA, 2006: 208-222

[11] Mutlu O, Moscibroda T. Stall-time fair memory access scheduling for chip multiprocessors//Proceedings of the International Symposium on Micro-Architecture. Washington, DC, USA, 2007: 146-160

[12] Mutlu O, Moscibroda T. Parallelism-aware batch scheduling: Enhancing both performance and fairness of shared dram systems//Proceedings of the International Symposium on Computer Architecture. New York, NY, USA, 2008: 63-74

[13] Dutt N. Memory-aware NoC exploration and design//Proceedings of the Design, Automation& Test in Europe Conference. Munich, Germany, 2008: 1128-1129

[14] Jang W, Pan D Z. An sdram-aware router for networks-on-chip. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2010, 29(10): 1572-1585

[15] Yuan G L, Bakhoda A, Aamodt T M. Complexity effective memory access scheduling for many-core accelerator architectures//Proceedings of the IEEE International Symposium on Microarchitecture. New York, NY, USA, 2009: 34-44

[16] Fedorova A, Seltzer M, Smith M D. Cache-fair thread scheduling for multicore processors. Harvard University: Technical Report TR-17-06, 2006

[17] Chang J, Sohi G S. Cooperative caching for chip multiprocessors//Proceedings of the International Symposium on Computer Architecture. Los Alamitos, CA, USA, 2006: 264-276

[18] Abts D, Natalie D, Jerger E, Kim J. Achieving predictable performance through better memory controller placement in many-core CMPs//Proceedings of the International Symposium on Computer Architecture. Austin, TX, USA, 2009: 451-461

[19] Dally W J, Towles B. Principles and Practices of Interconnection Networks. San Francisco: Elsevier, Inc. 2004

[20] Cuppu V, Jacob B. A performance comparison of contemporary DRAM architectures//Proceedings of the 26th International Symposium on Computer Architecture. Atlanta, Georgia, USA, 1999: 222-233

[21] Micron. 1Gb DDR2 SDRAM Component: MT47H128M8B7-25E, June 2006

LIU Sheng, born in 1984, Ph. D. candidate. His main research interests include microprocessor architecture, memory systems.

CHEN Shu-Ming, born in 1961, professor, Ph. D. supervisor. His main research interests include microprocessor architecture and VLSI design.

YIN Ya-Ming, born in 1979, Ph. D. candidate. His main research interests focus on microprocessor architecture.

CHEN Sheng-Gang, born in 1981, Ph. D.. His main research interests focus on microprocessor architecture.

GU Hui-Tao, born in 1980, Ph. D. candidate. His main research interests focus on microprocessor architecture.

CHEN Xiao-Wen, born in 1982, Ph. D. candidate. His main research interests focus on microprocessor architecture.

WANG Yao-Hua, born in 1985, Ph. D. candidate. His main research interests focus on microprocessor architecture.

Background

It is a very important issue that the limited memory bandwidth should be utilized efficiently and fairly in the homogeneous many-core CMPs. However, the problem of memory unfairness, which would induce unfair utilization of limited memory bandwidth, is manifest in many-core CMPs. From the research of related works, we found there were not so many researchers focused on the off-chip memory fairness problems of many-core CMPs, especially considering the NoC latency difference issue.

This paper constructed theoretical and experimental models to analysis the cautions and effects of NoC latency difference and memory fairness, and proposed the SBNL method to improve the traditional method and could reduce the NoC latency difference's side effect on the memory fairness by about 20% and bring in 15.7% execute efficient increment, compared with the traditional methods.