

利用预测器降低令牌协议的通信量*

付桂涛, 赵天磊, 黄平, 汤先拓, 邢座程
(国防科技大学 计算机学院, 湖南 长沙 410073)

摘要:令牌协议采用广播方式发送一致性消息,导致网络通信量大,限制了协议的扩展能力。提出预测的方法有效地消除令牌协议中的无用消息,从而降低了通信量。根据应用程序的读写失效比例提出3种预测策略。实验结果显示,采用512项的Owner预测器平均降低3.8%的连接通信量,Sharer预测器和Hybrid预测器分别平均降低了11%和7%的连接通信量。预测方法可以有效地降低令牌协议的通信量,提升令牌协议的扩展能力。

关键词:令牌协议;一致性协议;预测器;通信量;广播

中图分类号:TP302.7 **文献标志码:**A **文章编号:**1001-2486(2012)06-0014-06

Reducing the traffic of token protocol with predictors

FU Guitao, ZHAO Tianlei, HUANG Ping, TANG Xiantuo, XING Zuocheng

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

Abstract: Coherence messages in token protocol are always broadcast, which creates heavy network traffic and limits its scalability. An efficient predictor technique was used to eliminate the useless message in token protocol, which employs a predictor to predict the data's sharers and only sends coherence messages to the sharers to avoid broadcasting. Three predictor policies depending on the write miss and read miss ratio of application were proposed. The results show that: the owner predictor with 512 entries can reduce interconnect traffic by 3.8% on average, the sharer predictor and the hybrid reduce interconnect traffic by the average of 11% and 7% respectively. The method proposed efficiently reduces the traffic and improves the scalability of token protocol.

Key words: token protocol; coherence message; predictor; traffic; broadcasting

半导体工艺的进步,使得一颗芯片上集成多个处理器核成为现实,这种片上多核处理器(Chip Multi-Processor, CMP)具有资源利用率高、功耗低等优点,受到了学术界和工业界的热捧^[1-4]。随着多核处理器规模的增大,需要片上存储系统为其提供更多的数据。片上多核处理器通常采用共享Cache的片上存储系统^[1-3]。共享Cache可以平衡Cache需求不均衡的负载,具有空间利用率高等优点。多个处理器核共享存储空间,需要考虑一致性问题。在多核处理器中,Cache一致性协议用来维护Cache中数据的一致性和完整性。

监听协议^[5]和目录协议^[6-7]是常用的两种Cache一致性协议。监听协议依赖于有序的互联网络,比如总线、树形网络。监听协议相对简单,一致性消息采用广播方式发送至目的地,传递延迟较低,适用于规模较小的多核处理器,比如IBM的power6^[2]和power7^[4]两款处理器采用基于广

播的监听协议。

目录协议通过查询目录结构获取数据块的状态信息,目录协议的一致性消息发送至特定的目的地,避免广播消息,具有较好的扩展性。然而,目录协议的失效延迟较大,且存放Cache line状态信息也需要一定的存储开销。

令牌协议^[8-9]是一种在无序网络上实现的广播协议。令牌协议将目录协议中通过复杂的消息组合和顺序要求处理的各种请求,改为利用令牌的传递和收集的方式实现,提高了冲突处理时的性能,是实现Cache一致性的新思路。

令牌协议包括正确性基础和性能协议两部分,通过令牌计数规则避免了对有序互联网络的需求,采用广播方式发送一致性消息,Cache失效延迟小;性能协议实现消息的传递方式。研究发现,基于广播的协议中,存在着大量无用的消息。无用消息不仅浪费网络带宽,还增加失效延迟,影

* 收稿日期:2012-07-11

基金项目:国家自然科学基金资助项目(61170083, 60873016, 61103011);教育部博士点基金资助项目(20114307110001, 20094307120007);国家“核高基”重大专项资助项目(2011ZX01028-001-001, 2009ZX01028-002-002)

作者简介:付桂涛(1980—),男,河南濮阳人,博士生,E-mail:fgt nudt@nudt.edu.cn;

邢座程(通信作者),男,研究员,博士,博士生导师,E-mail:zcxing@nudt.edu.cn

响系统性能。

目前有多种减少广播协议的无用消息的方法。文献[10]提出基于SMP的过滤器,它把过滤器分为包含和非包含两类。包含过滤器跟踪Cache数据用于区分数据是否存在于Cache;非包含过滤器记录了Cache最近被替换的数据块。非包含的过滤器命中的数据块不会存在于Cache,可以过滤对该数据的访问,减少不必要的访问量。文献[11]介绍了一种降低协议通信量的RegionScout方法。RegionScout的项记录了主存的一个连续区域,区域粒度取决于存储开销。项的内容用于判断本地发送的请求在远程节点是否失效。如果失效就不需要发送请求,减少了互连和带宽开销。INCF^[12]提出一种网络一致性过滤器,该过滤器记录非共享的数据块,一致性消息通过互连网络传输,消息到达本地路由器时,如果过滤器发现该数据是非共享数据,路由器不再向下一节点传递该消息,从而降低了网络通信量。INCF方法必须修改路由器的结构,而且该方法不能采用自适应路由算法。VTC^[13]提出虚拟树一致性,在无互连网络上实现一种虚拟有序的链路,一致性消息通过虚拟有序链路传递,通过跟踪粗粒度区域的共享信息,VTC采用组播方式向共享节点发送一致性消息,避免了广播消息。Wang^[14]提出SRC方法降低令牌协议通信量,SRC是类Cache的结构,记录了最近访问的共享数据块的共享者,避免了广播消息。Martin^[16]提出了一种目的地预测方法,通过预测最小的目的地减少广播协议中的通信量。Martin提出了3种预测策略:Owner, Broadcast-If-Shared和Group。Owner策略预测数据的Owner,降低了成对共享的通信量;Broadcast-If-Shared策略用2位计数器记录数据块是否共享,若数据共享需要广播消息,否则从主存获取数据;Group策略用 N 个2位状态计数器记录数据是否共享,计数器大于1时处理器有共享数据,消息发送至有共享数据的节点。

在令牌协议中,读权限至少需要一个令牌,写操作需要所有的令牌。所以发生读失效时请求者发送一致性请求至数据的Owner获取至少一个令牌,而发生写失效时请求者需要发送请求至数据所有的共享节点获取令牌。本文根据程序的读写失效比例提出3种不同的预测策略:Owner策略用于读失效较多的应用;Sharer策略用于写失效较多的应用;Hybrid适用于读写均衡的应用。

1 令牌协议介绍

令牌协议的令牌计数规则保证一致性的正确

性,任何时候都不会违反一致性规则:同时只能有一个写操作或多个读操作,这就保证了协议的正确性;性能协议是引导令牌和数据块在系统中有效移动的方法,性能协议决定了令牌协议的性能优劣。

系统初始化时,系统为每个数据块分配 T 个令牌,其中一个为Owner令牌, T 不小于处理器核的数目。令牌存放于Cache、主存或一致性消息内。令牌和数据在部件之间传递,令牌不能丢失和损坏,任何时候数据的令牌总数量保持不变。通过特定的计数规则可保证访问数据的行为是一致的:读数据至少需要一个令牌;写数据则需要 T 个令牌。同一时刻只有一个写操作或者多个读操作,二者不能同时出现,保证了协议的正确性。MOESI协议的状态可以映射到令牌协议中,通过令牌的数目、是否有Owner令牌以及包含的数据是脏还是干净等信息可以判断Cache块的状态,具体的映射关系如表1所示。

表1 MOESI状态与令牌数目的映射关系
Tab.1 Mapping of MOESI states to token number

状态	令牌数目
L/NP	0
S	1 to $T-1$, but not the owner
O	1 to $T-1$, include the owner token
E	T , dirty bit inactive
M	T , dirty bit active

令牌协议为了处理饿死问题,采用了持续请求的饿死避免机制,当普通请求在一定时间内不能得到数据和令牌时,认为该请求为饥饿状态,请求者发送持续请求,持续请求是一种独占式的优先请求,其他处理器核收到持续请求,释放令牌和数据给持续请求的发送者,从而避免请求者饿死。

2 程序共享度分析

当Cache发生失效,一致性协议需要发送请求从其他Cache或主存获取数据。假定CMP有 N 个核。

在令牌协议中,发生读失效的处理器需要从数据的Owner节点获取数据和令牌,协议中只有到达Owner节点的消息才是有效的,其他消息是无用的。写失效时请求者需作废数据的所有共享副本,只有发送至数据共享者的消息才是有效的,其他的消息是无用的。每次读失效广播 N 个消息,只有1个有效,其余 $N-1$ 个无效;写失效的无用消息量与程序的共享度有关,如果共享度高,数据的共享副本数量多,无用消息量小;否则存在

大量无用消息。

我们测试了 SPLASH-2^[16] 程序的共享度,分析发现 SPLASH-2 程序的写失效产生约 76% 的无用消息。

图 1 为 SPLASH-2 程序的共享度分布情况。在图中,程序的共享度多数为 1,约为 58%,其中 water 和 water-n 共享度为 1 的比例高达 95%,无用消息较多。柱形图的上半部为写失效,下半部为读失效,图中 fft 的读失效比例为 66%,ratrace 的写失效比例为 77.5%。

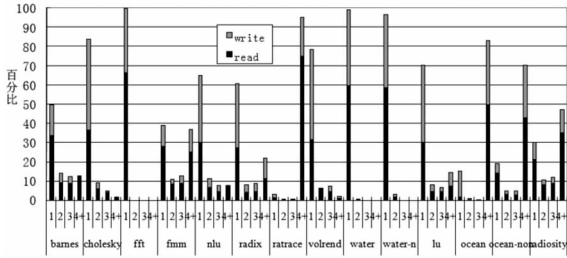


图 1 SPLASH-2 程序共享度分布图

Fig. 1 Sharing degree of the SPLASH-2 applications

3 预测器的实现方法

3.1 预测策略

令牌协议中存在大量的无用消息,无用消息的数量和程序的读写失效比例有关。我们根据程序中读写失效比例提出了三种预测策略: Owner 策略, Sharer 策略和 Hybrid 策略,通过预测数据的 Owner 和 Sharer 降低读/写失效引起的无用消息。三种预测策略的结构和预测操作如表 2 所示。

令牌协议中,某个处理器核的本地 Cache 有数据的 Owner 令牌和有效数据,该处理器的 Cache 为数据的 Owner, Owner 可以为处理器核的私有 Cache、共享 Cache 或主存。在 CMP 中,处理器核一般有私有的 L1 Cache,当 Owner 为 L1 Cache 时,可以用处理器核 ID 表示 Owner。本文只考虑数据的 Owner 为私有 L1 Cache 的情况,通过预测数据的 Owner,避免读失效引起的广播请求在 L1 Cache 之间传递。当数据的 Owner 为共享 L2 Cache 或主存时,请求者直接发送请求至共享 Cache 或主存。Owner 策略记录数据的 Owner,通过设置一个有效位表示 Owner 的存放位置,有效位置 1 表明数据的 Owner 为 L1 Cache,有效位为 0 表明 Owner 是主存或 L2 Cache。读失效时请求消息只发送至数据的 Owner,避免了请求消息的广播。Owner 策略适合读失效较多、Owner 较为稳定的应用。

处理器发生写失效时,令牌协议中发生写失

效时需要广播写请求至所有的处理器核,作废数据共享副本,只有数据的共享者 Sharer 响应请求消息。Sharer 策略用一个位向量记录数据的共享者,每位对应一个处理器核,当该处理器核的 L1 Cache 有数据的副本和令牌时,该处理器核为数据的共享者 Sharer,对应的位向量位置为 1。写失效时查询位向量可以获取数据的共享者列表,发送写请求消息至共享者。Sharer 策略适合共享度较小、写失效比例较大的应用。

Hybrid 策略结合 Owner 和 Sharer 两种策略,记录数据的 Owner 和共享者 Sharer。写失效时请求者发送消息至数据的共享者,读失效时请求者发送至数据的 Owner,避免了读/写失效引起的无用消息,从而降低令牌协议的通信量。该策略适合读写时效较为均衡的应用。

3.2 协议工作流程

当处理器核发生 L1 Cache 失效时,在采用了预测器的令牌协议中,处理器发送请求消息之前通过查找预测器获取数据的共享者信息,查找预测器与访问 L1 Cache 可以并行操作。处理器根据失效类型将读失效请求发送至数据的 Owner,写失效请求发送至数据的共享者 Sharer,最终得到需要的令牌和数据。

当预测器中没有记录数据的共享信息时,请求者在等待两倍平均失效延迟之后仍得不到令牌和数据,需要重发请求消息。为了避免再次失败,重发请求采用广播方式。请求者重发失效请求后等待两倍的平均失效延迟,仍未得到数据和令牌,系统判定该失效请求为饥饿状态,为避免饿死,请求者借助于独占式持续请求,最终保证失效得到足够的令牌和数据。

3.3 预测器的更新

预测信息的准确度影响协议的性能。预测准确,请求者可以快速地获取数据和令牌;若预测失败或预测信息不准确,需要重发请求,由于采用广播方式重发请求,预测失败导致通信量增加。为了保证预测信息的准确性,预测器的内容需要更新。

根据请求消息的来源更新预测器是一种简单的方法。处理器核 B 接收到请求者 A 发来的请求消息,若为写请求消息,请求者 A 得到令牌和数据后修改该数据,将成为数据的 Owner,这时系统只有 A 一个共享者,B 的预测器需要更新内容,若预测器没有记录该数据,需要分配空间记录该数据的共享信息。如果 A 发送的是读请求,A 将

成为数据的 Sharer, B 的预测器也需要更新。

根据接收到的响应消息也可以更新预测器。响应消息分为 Ack 消息和数据消息两类。Ack 消息包含令牌,数据消息包含数据和令牌,数据消息又分为共享类消息和 Owner 类消息,处理器核接收到共享类消息,拥有了至少一个令牌和有效数据,成为数据的共享者 Sharer;处理器核接收到 Owner 类消息,拥有了 Owner 令牌和有效数据,成为数据

的 Owner;接收到 Ack 消息只是增加令牌数目。处理器核接收到数据消息可以更新预测器内容,如果是共享类消息,消息发送者为数据的 Owner;如果是 Owner 类消息,消息发送者释放了数据 Owner 权限,本地 L1 Cache 成为数据的 Owner。令牌协议中发送者发送 Ack 消息响应写请求,意味着 Ack 消息发送者释放了令牌,不再是数据的共享者,需要从 Sharer 列表中删除发送者。

表2 预测器策略

Tab. 2 Predictor policies

策略	Owner	Sharer	Hybrid
结构	Owner ID,有效位	Sharer 位向量	Owner ID,有效位,Sharer 位向量
项大小	$1 + \log_2 N + \text{tag}$	$N + \text{tag}$	$N + 1 + \log_2 N + \text{tag}$
预测操作	请求至 Owner ID	发请求至共享者	读操作发请求至 Owner;写操作发请求至共享者
读写	如果是写请求,把请求者	把请求者加入	写请求,更新 Owner,清空 Sharer,只有请求者一个 Sharer;读请求,把请求者加入 Sharer
更新操作	请求	Sharer	
数据	Owner 型数据,更改	发送者没有保留令	Owner 型数据,更改 Owner,如果发送者没有保留
响应	Owner	牌,从 Sharer 删除	令牌,从 Sharer 删除

当数据从 L1 Cache 被替换,本地的预测器内容不需要更新,其他的预测器需要更新,本文通过一种称为 Hint 的消息完成预测器的更新。如果 Owner 数据被替换写回,数据 Owner 变为 L2 Cache 或主存,本地处理器核发送 Hint 消息更新其他预测器,如果预测器保存数据的预测信息,Owner 的有效位置为 0,表示数据 Owner 为 L2 Cache 或主存。如果共享数据被写回替换,发送 Hint 消息至其他处理器核,从预测器的共享者列表中删除该共享者。Hint 消息带来一定的通信开销,但并不影响协议的性能,因为 Hint 消息发送不在失效请求的关键路径上,丢失 Hint 消息不会破坏协议的正确性。

预测器受芯片面积、访问延迟等因素约束容量不能太大。如果太大,预测器记录的信息有限,可能出现预测失效或信息不准确。我们采用 LRU 替换算法更新预测器的内容。

3.4 预测器开销

从表 2 的预测器结构得知,Owner 预测器记录数据的 Owner ID。假定 CMP 中有 N 个核,保存 Owner ID 需 $\log_2 N$ 位,另外,Owner 预测器用一个有效位区分数据是否存在于 L1 Cache。Owner 预测器的项需要 $\log_2 N + 1 + \text{tag}$ 位。Sharer 预测器保存数据的共享者列表,用 N 位的位向量记录共享者,每位对应一个处理器核 ID,预测器每个项需要 $N + \text{tag}$ 位。Hybrid 预测器保存数据 Owner 和共享者,预测器每项的开销为 $\log_2 N + N + 1 + \text{tag}$ 位。

假定 16 核的 CMP 系统,每个核私有 L1 Cache 大小为 64KB,地址为 64 位,Owner 策略的预测项需要 69 位,约 9Bytes。512 项的 Owner 预测器需要 4.5KB,约为 L1 Cache 的 7%。Sharer 策略的预测项需要 78 位,约为 10 Bytes,512 项的 Sharer 预测器需要 5KB,约为 L1 Cache 的 7.8%。Hybrid 预测器开销较前两种情况稍大。

4 实验结果与分析

4.1 实验平台

本文采用威斯康星大学开发的多核模拟器 GEMS^[17] 模拟平台,基准系统采用 16 个 Sparc V9 组成的瓦片式 CMP。每个瓦片包含处理器核,分离式 L1 cache, L2 cache bank 和预测器。CMP 的参数按照文献[9]设置,CMP 采用 2D-mesh 互联网络和 X-Y 路由算法。预测器、Cache 和主存通过互连网络连接。

本文使用 SPLASH-2^[16] 测试集作为测试程序。SPLASH-2 是并行系统的测试程序,主要用于并行化的测试。为避免系统冷启动的影响,需要对测试程序预热操作。模拟前先把测试程序运行一遍,保存 checkpoint 记录 Cache 的状态,然后开始模拟程序,进入程序的并行阶段后统计相关数据。

4.2 结果分析

4.2.1 通信量分析

衡量协议的通信量有两种指标:一种侧重于

传递消息的带宽需求,测试每次失效传递请求消息需要的字节,称为连接通信量,字节越多表示通信量越大;另一种侧重于存储控制器工作的频率,测试每次失效需要处理的消息数量,称为端点通信量,消息数量越大通信量愈大。连接通信量与互连拓扑结构、路由算法以及消息数量有关,端点通信量与一致性消息的发送方式有关。

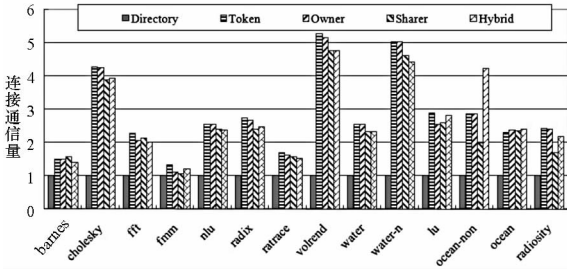


图 2 连接通信量对比

Fig. 2 Interconnect traffic comparison normalized to Directory

目录协议的消息发送有特定的目的地,消息量最小。本文以目录协议的通信量作为比较的基准。采用预测方法可以降低令牌协议的连接通信量。Owner 策略可以减少读失效引起的广播,程序中读失效较多的程序减少的连接通信量越大,如图 2 所示为连接通信量对比图,程序 fmm 的读失效比例为 71%,采用 Owner 策略降低了 17% 的连接通信量。然而失效比例不是影响连接通信量的唯一因素,比如程序 barnes 的读失效比例为 64%,仅降低了 1.2% 的通信量。这是因为程序中存在大量的数据写回消息,Owner 策略不能避免此类消息量。Sharer 策略可以避免写失效引起的广播,适用于共享度小、写失效比较高的程序,比如 ocean-non 和 radiosity 采用 Sharer 策略分别降低了 30% 和 29% 的连接通信量。Hybird 策略对大多数程序都有效,平均降低 7% 的连接通信量。

图 3 为端点通信量对比图。柱形图的上半部是数据类消息,下半部是控制类消息。令牌协议的控制类消息是目录协议的 2.4 倍。采用 Owner 策略的令牌协议,平均减少了 29% 的控制类消息,数据量消息降低了 26%。采用 Sharer 策略令牌协议平均降低 28% 的控制类消息,30% 的数据类消息。

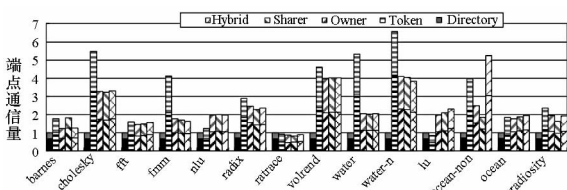


图 3 端点通信量对比

Fig. 3 Endpoint traffic comparison normalized to Directory

4.2.2 预测器容量对通信量的影响

预测器容量是预测器设计的关键问题。预测器容量太大将占用大量芯片面积,查找延迟加长。如果容量太小,预测器记录的信息有限,预测失效率增加。为了研究预测器容量对令牌协议通信量、失效延迟的影响,针对不同容量的预测器进行了对比实验,为了节约篇幅,这里只介绍 Hybrid 预测器容量的分析。

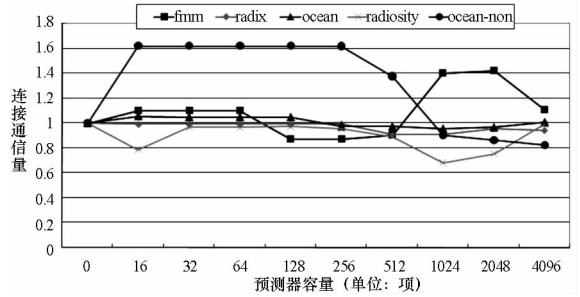


图 4 Hybrid 预测器容量与连接通信量关系图

Fig. 4 Normalized interconnect traffic for Hybrid predictor with increasing entry counts

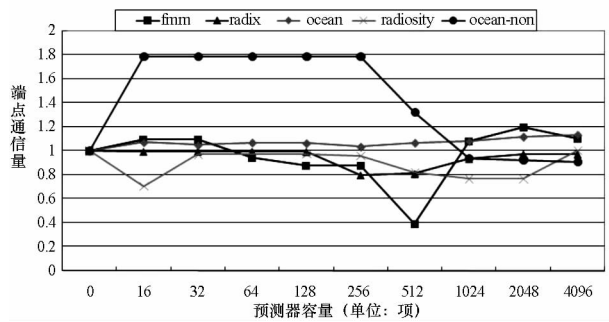


图 5 Hybrid 预测器容量与端点通信量关系图

Fig. 5 Normalized endpoint traffic for Hybrid predictor with increasing entry counts

图 4 和图 5 分别是采用 Hybrid 策略的令牌协议的互连通信量和端点通信量,图中容量为 0 表示未采用预测器的令牌协议。预测器容量增加,预测失效率减少,协议的通信量降低,如程序 fmm 的通信量随预测器容量的增大其通信量变小,当容量为 512 时互连通信量降低 22%,端点通信量降低了 60%。预测器容量增加导致预测器访问延迟变大,程序输入集较小,预测器中存放的共享信息不再准确,导致协议广播重发消息,增加协议通信量。如 fmm 程序在预测器容量为 2048 项时,互连通信量增加了 42%。

4.2.3 带宽受限条件下扩展性分析

带宽受限条件下,令牌协议由于带宽资源竞争,消息需要多次重发,甚至采用持续请求获取令牌和数据,这些重发消息和持续请求增加了令牌协议的消息量。图 6 和图 7 为 flit 等于 1bit 的带

宽条件下协议的互连通信量和端点通信量,令牌协议的互连通信量为目录协议的 4.5 倍。Owner 预测器降低了 23% 的互连通信量和 22% 的端点通信量,Sharer 预测器降低了 30% 的互连通信量和 28% 的端点通信量。与 4.2.1 节相比,预测器方法在带宽受限条件下取得的效果更好,因此本文提出的方法可以提升令牌协议的扩展性。

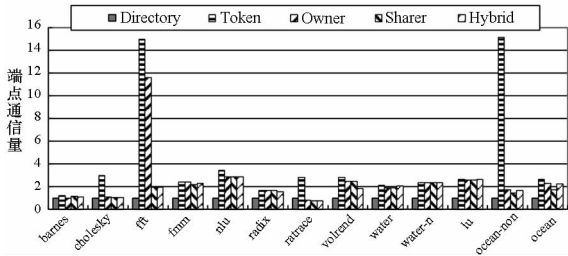


图 6 带宽受限条件下协议的连接通信量比较
Fig. 6 Interconnect traffic comparison normalized to Directory with limited bandwidth

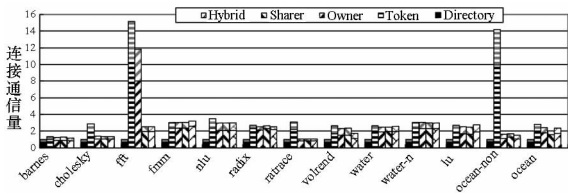


图 7 宽受限条件下协议的端点通信量比较
Fig. 7 Endpoint traffic comparison normalized to Directory with limited bandwidth

5 结束语

令牌协议是一种新的灵活性强的一致性协议。它解耦了性能策略和正确性基础,可以灵活地适应于多种不同的系统。令牌协议采用广播方式发送请求消息,通信量大,限制了令牌协议的扩展能力。本文通过预测数据的共享信息,使用预测器记录数据的共享者信息,减少了广播请求,降低了通信量。实验结果显示:采用预测方法可以显著地降低令牌协议的通信量。

参考文献 (References)

[1] Barroso L, Gharachorloo K, McNamara R, et al. Piranha: A scalable architecture based on single-chip multiprocessing [C]//Proceedings of International Symposium on Computer Architecture, 2000; 282 - 293.
[2] Le Q, Starke J, Fields S, et al. IBM POWER6 micro-architecture[J]. IBM Journal of Research and Development, 2007, 51(6): 639 - 662.

[3] Kongetira P, Aingaran K, Olukotun K. Niagara: A 32-way multithreaded SPARC processor [J]. IEEE Micro, 2005, 25 (2): 21 - 29.
[4] Kalla R, Warnock J, Cargnoni R, et al. POWER7: a highly parallel, scalable multi-core high end server processor [J]. IEEE Journal of Solid-State Circuits, 2011, 25 (1): 145 - 161.
[5] Goodman J R. Using cache memory to reduce processor memory traffic [C]//Proceedings of International Symposium on Computer Architecture, 1983: 176 - 283.
[6] Agarwal A, Simoni R, Hennessy J. An evaluation of directory schemes for cache coherence [C]//Proceedings of International Symposium on Computer Architecture, 1988; 353 - 362.
[7] Tang C K. Cache design in the tightly coupled multiprocessor system [C]//Proceedings of AFIPS Conference Proceedings of National Computer Conference, 1976: 749 - 753.
[8] Martin M, Hill M, Wood D, et al. Token coherence: Decoupling performance and correctness [C]//Proceedings of International Symposium on Computer Architecture, 2003: 182 - 193.
[9] Martin M. Token coherence [D]. University of Wisconsin-Madison, 2003.
[10] Moshovos A, Memik G, Falsafi B, et al. Jetty: filtering snoops for reduced energy consumption in SMP servers [C]// Proceedings of International Symposium on High-Performance Computer Architecture, 2001: 85 - 96.
[11] Moshovos A. RegionScout: Exploiting coarse grain sharing in snoop-based coherence [C]//Proceedings of International Symposium on Computer Architecture, 2005; 234 - 245.
[12] Niket A, Li-Shiuan P, Niraj J. In-network coherence filtering: snoop coherence without broadcasts [C]// Proceedings of IEEE/ACM International Symposium on Micro-architecture, 2009; 232 - 243.
[13] Wang H X, Wang D S, Li P, et al. Reducing network traffic of token protocol using sharing relation cache [J]. Tsinghua Science and Technology, 2007, 12(6): 691 - 699.
[14] Natalie D, Peh L S, Lipasti H. Virtual tree coherence: Leveraging regions and in-network multicast trees for scalable cache coherence [C]//Proceedings of International Symposium on Micro-architecture, 2008; 35 - 46.
[15] Martin M, et al. Using destination-set prediction to improve the latency/bandwidth trade-off in shared-memory multiprocessors [C]//Proceedings of International Symposium on Computer Architecture, 2003; 206 - 217.
[16] Woo S, Ohara M, Torrie E, et al. The SPLASH-2 programs: Characterization and methodological considerations [C] // Proceedings of International Symposium on Computer Architecture, 1995; 24 - 36.
[17] Martin M, Sorin D, Bechmann B, et al. Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset [J]. ACM SIGARCH Computer Architecture News, 2005, 33 (4): 92 - 99.