·体系结构与软件技术· 文章编号:1000-3428(2017)07-0038-06 文献标志码:A 中图分类号:TP332

一种应用于多路直连 CMP 的混合一致性协议

王云霏,王 飙,李 媛,孙战先

(上海高性能集成电路设计中心,上海 200120)

摘 要: Cache 一致性协议对系统性能和带宽需求具有重要影响,而当前广泛采用的广播协议带宽需求较高,目录协议访存延迟较大,均不适用于国产服务器 CPU 直连接口带宽较低及延迟较大的应用场景。针对上述问题,基于 Token 广播协议和目录协议,设计混合一致性协议,采用 Simics 结合 GEMS 搭建多路直连片上多核处理器仿真系统,通过运行 SPLASH-2 测试协议的相关性能。实验结果表明,混合协议的系统性能优于目录协议,与 Token 协议相比,混合协议以较小的性能代价,大幅降低片间通信带宽需求,且在带宽资源受限系统中具有更好的系统性能。

关键词: 片间直连;Cache 一致性;Token 协议;目录协议;混合协议

中文引用格式: 王云霏,王 飙,李 媛,等. 一种应用于多路直连 CMP 的混合一致性协议[J]. 计算机工程,2017,43(7):38-43.

英文引用格式: Wang Yunfei,Wang Biao,Li Yuan,et al. A Hybrid Coherence Protocol Applied to Multi-channel Direct Connection CMP[J]. Computer Engineering,2017,43(7):38-43.

A Hybrid Coherence Protocol Applied to Multi-channel Direct Connection CMP

WANG Yunfei,WANG Biao,LI Yuan,SUN Zhanxian

(Shanghai High Performance IC Design Center,Shanghai 200120,China)

【Abstract】 Cache coherence protocol impacts the system performance and the demand of bandwidth. Snoopy protocols and directory protocols are widely used in modern server systems,but the former needs large bandwidth while the latter has long latency,so they are not suitable for domestic server CPU used in the scenario that the bandwidth is relatively small and latency is relatively long. To solve this problem,this paper proposes a hybrid coherence protocol based on Token protocol and directory protocol,and uses Simics and GEMS to construct a system in which multi-channel Chip Multi-processors(CMP) are directly connected. It then runs SPLASH-2 benchmark to test related performance. The experimental results show that the hybrid protocol has better performance than the directory protocol. Compared with the Token protocol,the hybrid protocol significantly reduces the demand of inter-chip bandwidth at relatively little cost of performance,and has better performance in the system that the bandwidth is not sufficient.

【Key words】 direct connection among chips;Cache coherence;Token protocol;directory protocol;hybrid protocol

DOI: 10.3969/j.issn.1000-3428.2017.07.006

0 概述

服务器 CPU 除了需要具有超强的计算能力和极高的事务处理吞吐率,还必须具备与之适应的大主存容量、高访存带宽和高 I/O 带宽。受限于芯片引脚数量和信号传输率,单个芯片所能够连接的主存容量有限,且访存带宽不能完全满足服务器的需求。

将多个服务器 CPU 直连构成基于 CC-NUMA 结构的共享多路服务器节点已经成为国际主流服务器 CPU 的必备特性。采用该架构可以方便地搭建基于分布共享主存结构[1]的多路服务器硬件平台,在此平台基础上易于构建单一映像的操作系统,并为上层应用提供统一、高效的编程接口。片间直连结构下的 Cache 一致性协议及实现技术是实现片间直连的重要基础之一。

目前,基于片间直连的一致性协议已经有了广泛的应用,Intel,AMD,IBM 等主流服务器处理器设计公司都已提出了一些成熟的设计方法。Intel 根据系统规模大小选择使用广播协议或者目录协议[2];AMD

基金项目: "核高基"重大专项(2013ZX0102-8001-001-001)。

作者简介: 王云霏(1992—),男,硕士研究生,主研方向为微处理器体系结构;王 飙、李 媛,高级工程师;孙战先,助理工程师。

收稿日期: 2016-06-23 **修回日期:** 2016-07-27 **E-mail:** flylucas_10@163.com

在 L3 Cache 维护 filter[3-4] 起到了目录结构的作用,减少出片的一致性请求;IBM 综合运用状态设计、区域划分、目录跟踪预测等技术降低广播所需带宽[5]。从发展趋势来看,采用目录跟踪、预测等方式,将目录协议与广播协议进行融合设计,进一步提高运行性能和通信效率,降低通信带宽需求,以支持更大规模的系统扩展是服务器系统现阶段的发展方向。

本文针对 Token 一致性协议和双层目录一致性协议的不同特点进行平衡设计,实现一种应用于多路直连片上多核处理器(Chip Multi-processors, CMP)的混合一致性协议。采用 Simics + GEMS (General Execution-driven Multiprocessor Simulator)模拟器对该混合协议的相关性能进行测试,并与 Token 一致性协议和双层目录协议进行比较和分析。

1　相关工作

目录协议和广播协议在现代服务器处理器直连中得到广泛应用,且表现出不同的性能特点。如图 1 所示[6],在 Cache-to-Cache 的数据响应中,目录协议[7-8]需要查询目录和转发请求,具有更高的访问延时。广播协议需要对所有存储节点发送请求,具有较高的带宽需求,且随着系统规模扩大片间通信流量增长更为迅速。对于给定的有限带宽,广播所有请求所引起的网络拥挤和排队也会增加访存延时。Intel 在其服务器处理器中集成了 2 种协议,以应对不同的应用场景:4 路直连及以下规模采用广播协议;8 路直连及以上规模采用目录协议[9]。与 Intel 的 QPI 直连接口相比,国产服务器直连接口的带宽仍然较小,面临的问题也更为严峻。

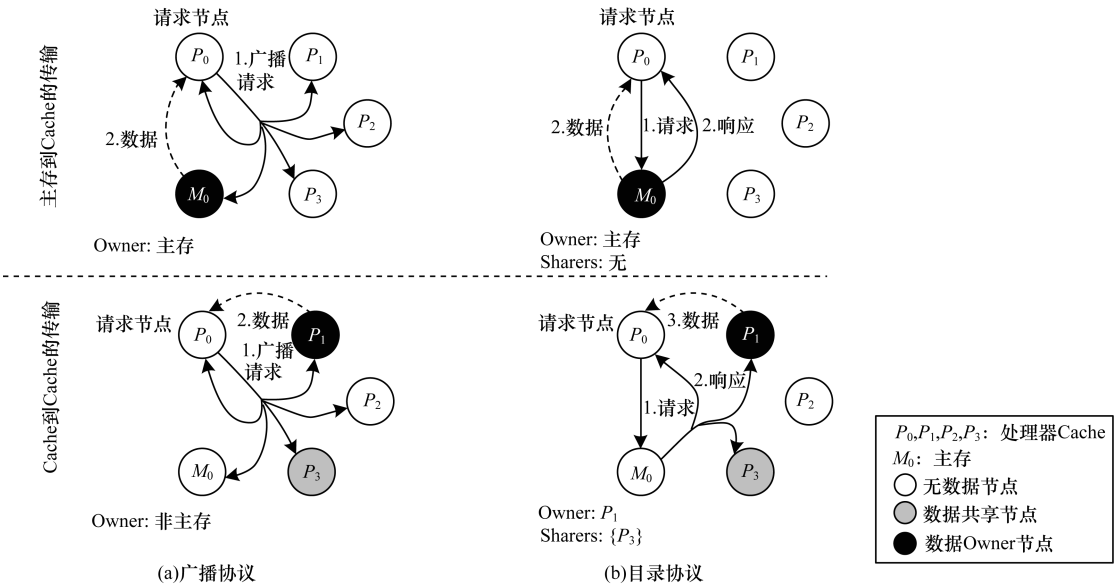


图 1　广播协议、目录协议示意图

Token 协议[10-11]是广播协议的一种,通过维护 Token 数目保证系统正确性,减少响应数目,降低通信开销。分析 Token 协议设计结构及实验数据,如图 2 所示,Token 协议在处理 persistent 请求时,采用完全扁平化方式,即当 L1 Cache 长时间未得到所需的数据及权限时,会广播 persistent 请求。由于该请求的范围包含每一个 L1 Cache、对应的 L2 Cache 和 Memory,因此会产生大量的片间一致性请求,给直连带宽较小的系统带来巨大的带宽压力。

本文通过采用层次化设计方案,在片内维护目录协议,在片间采用 Token 广播协议既能避免目录协议片间 3 跳的转发延时,又能降低完全扁平化 Token 协议的片间带宽需求。

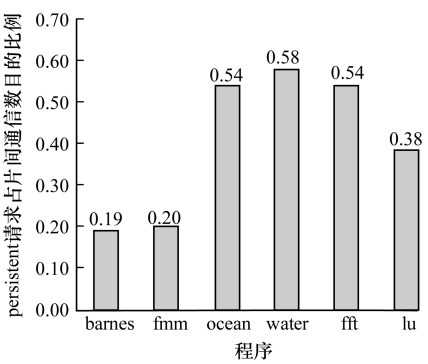


图 2　Token 协议 persistent 请求占片间通信数目的比例

2　混合协议设计

2.1　整体设计方案

在本文讨论的混合一致性协议中,L1 Cache 与

L2 Cache 之间采用目录协议，L2 Cache 与 L2 Cache、主存间采用 Token 协议。L2 Cache 是整个存储系统的核心节点，采用目录结构维护片内数据副本的分布情况，使用 Token 计数确定当前处理器所拥有的读写权限。当 L1 Cache 发生访存不命中时，向地址对应的片内 L2 Cache 发送请求，L2 Cache 根据自身状态和 Token 数目，确定当前处理器是否具有满足 L1 Cache 请求的权限。当具备权限时，根据 L2 Cache 数据状态和所维护的目录，直接完成响应，或是向片内其他 L1 Cache 发送二次请求；当权限不具备时还需向地址对应的其他处理器 L2 Cache 和主存广播 transient 请求。当片间广播的 transient 请求长时间得不到响应时，还需广播 persistent 请求。

考虑到各种请求及响应的复杂性，为避免死锁，混合一致性协议在各存储节点设置了虚通道[12]。虚通道是一种避免网络死锁的常用技术，通过为每个物理通道设置多个缓冲区，将通过链路的数据流到达输出端时被分解到独立的通道缓冲中。若一个虚通道被阻塞，则其他的虚拟通道还能继续通向输出端口。混合协议中虚通道的具体设置见表 1。

表 1 混合一致性协议中的虚通道设置

虚通道	应用	保序	阻塞
0	片内 L1 与 L2 间的请求	否	是
1	L2 与 L2 及主存间的 transient 请求	否	是
2	L2 与 L2 及主存间的 persistent 请求	是	是
3	响应网络	否	否

2.2 persistent 请求及串行化

本文所采用的混合一致性协议在 L2 Cache 与 L2 Cache、主存之间采用 Token 协议，与完全扁平化的 Token 协议相比，可降低 persistent 请求源节点的数目，同时缩小 persistent 请求的广播范围，减少 Token 协议的 persistent 请求消息数目。本文采用分布式仲裁处理 persistent 请求冲突，在 L2 Cache 和主存维护持续请求表(persistent-table)，由 persistent 请求网络输入 persistent 请求，依照事先规定的优先级(L2 Cache ID)响应 persistent 请求。为防止优先级较低的节点出现饥饿，L2 Cache 节点在发送 persistent 请求时，标记当前 persistent-table 中所有 persistent 请求，当该节点再次发送 persistent 请求时，需检查 persistent-table，确认无标记项。对于正在等待 L1 Cache 响应、处于暂态中的 L2 Cache 数据，本文采用阻塞 persistent 请求队列方式，先完成片内处理，再响应 persistent 请求。响应信息具有最高优先级。因此，本文所采用的一致性消息的串行化优先级为：响应信息 > persistent 请求(依据 L2 Cache ID 划分优先级) > transient 请求 > L1 Cache 请求信息。

2.3 Token 维护

由于主存访问延时较大，因此为尽可能避免对主存的访问出现在关键路径上，以提升系统性能，混合协议设计采用以下 2 种策略使数据块和 Token 尽可能存储在 L2 Cache 中：1)对主存的访问会获取主存数据的所有权限；2)当替换 L2 Cache 时，按照文献[13]方式检测最近数据副本请求、响应情况，预测其他数据副本分布情况，根据预测结果，转移数据权限。

2.4 状态机设计

为进一步降低混合一致性协议的片间通信流量，在进行状态机设计时，混合协议保证整个存储系统在任意时刻均有且仅有一个存储节点负责向片外节点提供数据转发[14]，并且提供数据转发的权限随片间请求转移，以均衡各节点的负载。各层 Cache 和主存的基态设计如表 2 ~ 表 4 所示。L2 Cache 的状态“L2_AB”中有 2 位标志，“A”表示处理器所拥有的数据权限，“B”表示 L1 Cache 的副本状态。

表 2 L1 Cache 基态设计

状态	说明
L1_M	L1 数据修改
L1_S	L1 数据共享
L1_I	L1 数据无效

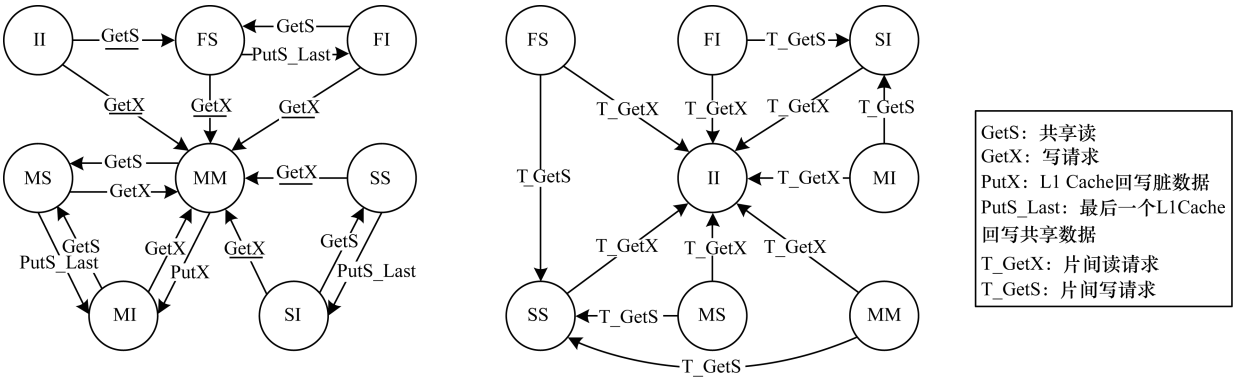
表 3 L2 Cache 基态设计

状态	说明	数据转发
L2_II	L2 数据无效，片内 L1 无数据	否
L2_SI	L2 数据共享，片内 L1 无数据	否
L2_SS	L2 数据共享，片内 L1 拥有共享数据	否
L2_FI	L2 数据共享，片内 L1 无数据	是
L2_FS	L2 数据共享，片内 L1 拥有共享数据	是
L2_MI	L2 数据独占，片内 L1 无数据	是
L2_MM	L2 数据无效，L1 数据独占	是
L2_MS	Chip 数据独占，L2，L1 数据共享	是

表 4 主存基态设计

状态	说明	数据转发
Mem_O	主存数据共享	是
Mem_NO	主存数据无效	否
Mem_L	主存锁定	否

图 3(a)为 L2 Cache 处于各基态，当接收到 L1 Cache 请求且得到响应时，各基态之间状态转换的状态机。带下划线的请求表示当前处理器不具有满足该请求的权限，需发送 transient 请求。图 3(b)为 L2 Cache 处于各基态，当接收到 transient 请求且 L2 Cache 进行响应时，各基态之间状态转换的状态机。



(a)L2 Cache收到L1 Cache处理器请求的情况　　(b)L2 Cache收到其他处理器请求的情况

图 3　L2 Cache 状态转换简图

2.5　协议对比

与其他混合协议相比,文献[6,15]为单核 CPU 互联,L1 Cache 和 L2 Cache 数据关系较简单。本文所设计的混合协议基于片上多核系统,L2 Cache 需根据片内数据副本分布状况进行请求和响应,状态设计更复杂,重点关注片间直连网络。文献[2,6,15]均在 L2 Cache 上维护了 2 种协议,与其相比本文混合协议能够节省一部分硬件开销。

3　实验方法

本文采用 Virtutech 公司的 Simics 全系统模拟器及 Wisconsin 大学的 GEMS 模拟器[16]搭建基本的实验环境。Simics 是一款全系统模拟器,通过配置可以模拟多路多核服务器系统。GEMS 模拟器的 Ruby 部件可对一致性协议的不同存储层次(主存、各 Cache 层次)的状态机和时序进行描述建模。使用 Ruby 替换 Simics 自带的存储层次,即搭建存储层次结构和 Cache 一致性协议可重构的实验平台。

在该实验环境中使用专用描述语言 SLICC(Specific Language for Implementing Cache Coherence)对服务器各存储层次的协议状态机和时序进行建模;SLICC 语法在设计之初就考虑到了与硬件描述语言(Hardware Description Language, HDL)之间的转换,符合SLICC语法的设计通常能够使用硬件描述语言实现。

本文测试程序来自 SPLASH-2(Stanford Parallel Applications for Share Memory)[17]测试程序集。SPLASH-2 包括 12 个典型的并行应用程序,所选取的测试程序及其规模如表 5 所示。

表 5　测试程序集属性

程序	应用	规模
barnes	三维多体问题	65 536
fmm	三维多体问题	65 536
ocean	大规模海洋模拟	258 ×258
water	水模拟	512 ×512
fft	傅里叶变换	65 536
lu	矩阵 LU 分解	512 ×512

本文所采用的 CMP 为带有 3 个直连接口的4 核处理器,各存储层次的主要参数见表 6。

表 6　仿真环境的存储结构及参数

存储结构	容量及结构	访问延时/cycle
L1 ICache/DCache	32 KB,4 路组相连	4
L2 共享 Cache	2 MB,4 个分体,16 路组相连	15
MEM 主存	每个核心处理器配置 2 GB 内存	150

本文在进行 4 路和 8 路直连时采用的网络拓扑结构如图 4 所示。

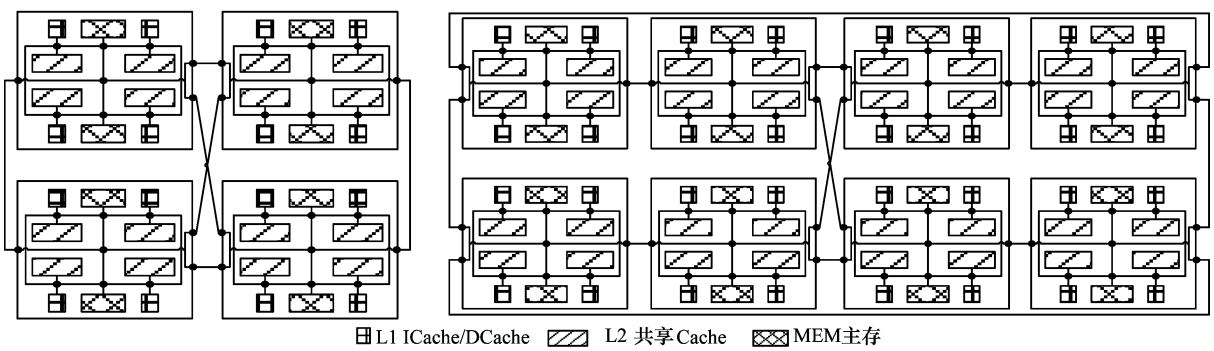


图 4　网络拓扑结构

连接 L1 ICache/DCache, L2 共享 Cache 和 MEM 的片内环形链路的单步延迟为 1 cycle,带宽为 16 Byte/cycle;连接 MEM 和直连接口的片内交叉开关的穿透延迟为 4 cycle,带宽为 16 Byte/cycle;片间直连链路的延迟为 110 cycle,带宽为 2 Byte/cycle(以 16 核 CMP,8 Byte/cycle 为参考,参照文献[6]缩放)。

4 实验结果与分析

4.1 4 路直连系统

图 5 和图 6 分别对比了 4 路 16 核直连系统 Token 协议、双层目录协议和混合协议的运行时间、片间一致性消息流量。实验结果表明,混合协议比目录协议性能平均提高 8.0%;与 Token 协议相比,以 5.7% 的性能为代价,降低了 51.6% 的带宽需求。

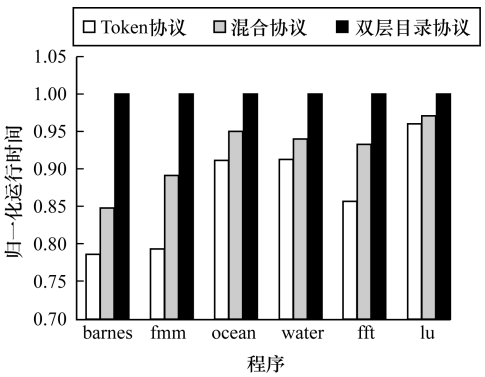


图 5 4 路直连系统中协议运行时间比较

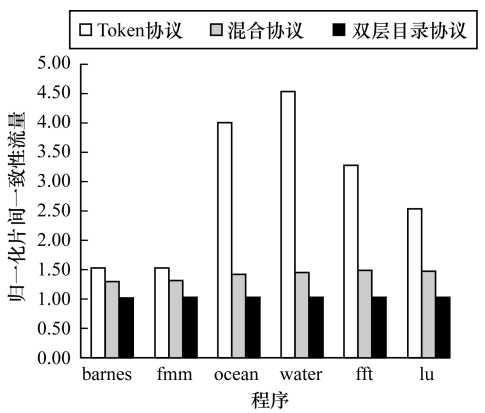


图 6 4 路直连系统中协议片间一致性消息流量比较

4.2 8 路直连系统

图 7 和图 8 分别对比了 8 路 32 核直连系统 Token 协议、双层目录协议和混合协议的运行时间、片间一致性消息流量。与双层目录协议相比,混合协议性能提升 4.2%。与 Token 协议相比,两者性能相当,但混合协议带宽需求降低了 32.0%。

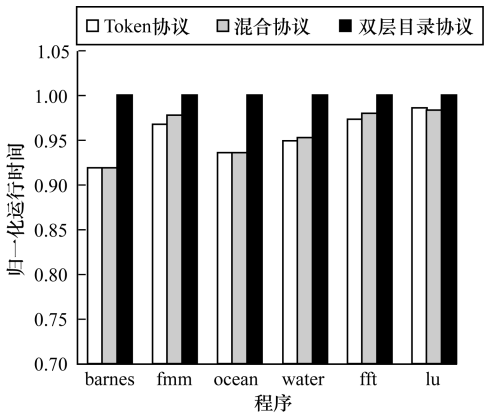


图 7 8 路直连系统中协议运行时间比较

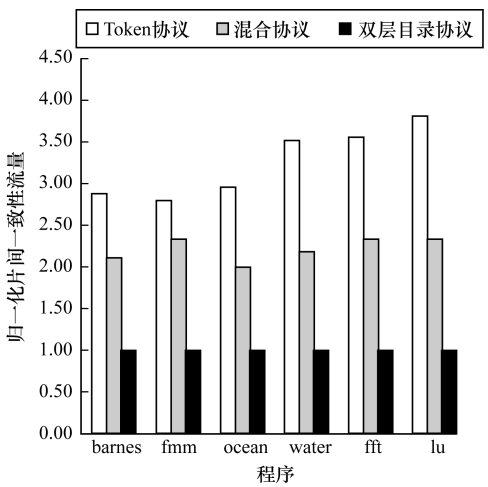


图 8 8 路直连系统中协议片间一致性消息流量比较

4.3 带宽压力测试

在上述实验中,片间网络利用率均小于 5%,体现的是不同协议在片间直连带宽较充裕情况下的性能。根据文献[6],在网络较拥挤时使用广播协议。片间网络利用率会达到 30% 以上。受限于仿真环境和测试程序,本文采用文献[6]的方式仿真 16 核处理器 4 路直连增加片间通信量,同时降低网络带宽,以达到相近的网络利用率。实验中在直连网络带宽为 0.5 Byte/cycle 时,Token 广播协议片间网络利用率达到 30% 左右,以仿真网络拥挤时不同协议的性能。实验结果如图 9 所示,与目录协议和混合协议相比,Token 协议在网络发生拥堵时性能恶化更为迅速,此时混合协议表现出优于 Token 协议和目录协议的性能。

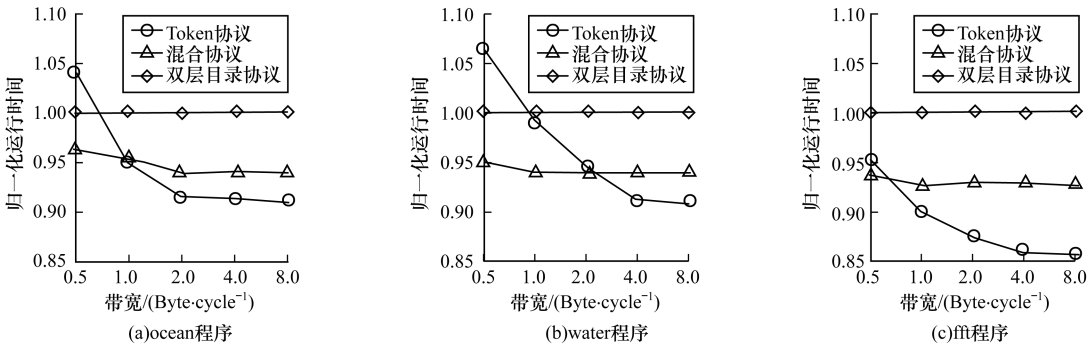


图 9　片间直连带宽对性能的影响

5　结束语

本文针对国产服务器处理器片间直连接口带宽较低、延迟较高的应用现状,权衡性能与带宽需求,设计基于 Token 协议和目录协议的混合一致性协议,讨论了其整体设计方案,确定了其状态机实现和串行化优先级,并通过 SLICC 进行方案实现。该方案在 L1 Cache 与 L2 Cache 及主存间维护了存在且唯一的数据响应节点,进一步降低了片间通信流量。在 L2 Cache 发生替换时,通过预测方式使 Token 和数据转移到其他 L2 Cache,避免对主存的访问出现在关键路径,提升系统性能。通过构建基于 Simics 和 GEMS 的仿真实验平台,对比分析了该混合一致性协议的相关性能,并指出该协议集合了目录协议和 Token 协议的优势,具有较好的应用前景。

参考文献

[1] 胡伟武. 共享存储系统结构[M]. 北京:高等教育出版社,2001.

[2] Molka D, Hackenberg D, Schone R, et al. Cache Coherence Protocol and Memory Performance of the Intel Haswell-EP Architecture[C]//Proceedings of the 44th International Conference on Parallel Processing. Washington D. C., USA:IEEE Press,2015:739-748.

[3] Ahmed A, Conway P, Hughes B, et al. AMD Opteron Shared Memory MP Systems[EB/OL]. [2016-07-14]. http://www. cse. wustl. edu/ ~ roger/569M/28_AMD_Hammer_MP_HC_v8. pdf.

[4] Conway P, Kalyanasundharam N, Donley G, et al. Cache Hierarchy and Memory Subsystem of the AMD Opteron Processor[J]. IEEE Micro, 2010, 30(2):16-29.

[5] Starke W J, Stuecheli J, Daly D M, et al. The Cache and Memory Subsystems of the IBM POWER8 Processor[J]. IBM Journal of Research and Development, 2015, 59(1): 31-33.

[6] Martin M M K, Sorin D J, Hill M D, et al. Bandwidth Adaptive Snooping [C]//Proceedings of the 8th Symposium on High-performance Computer Architecture. Washington D. C., USA:IEEE Press, 2002:251-262.

[7] Acacio M, Gonzaález J, Garciía J, et al. A New Scalable Directory Architecture for Large-scale Multiprocessors[C]// Proceedings of the 7th International Symposium on High-performance Computer Architecture. Washington D. C., USA: IEEE Computer Society, 2001: 97-106.

[8] Acacio M, Gonzaález J, Garciía J, et al. A Two-level Directory Architecture for Highly Scalable CC-NUMA Multiprocessors[J]. IEEE Transactions on Parallel and Distributed Systems, 2005, 16(1):67-79.

[9] Intel. An Introduction to the Intel Quick Path Interconnect [EB/OL]. [2016-03-11]. http://www. intel. com/content/www/us/en/io/quickpath-technology/quick-path-interconnect-introduction-paper. html.

[10] Hill M D, Marty M R. Cache Coherence Techniques for Multicore Processors[D]. Madison, USA: University of Wisconsin at Madison, 2008.

[11] Marty M R, Bingham J D, Hill M D, et al. Improving Multiple-CMP Systems Using Token Coherence[C]// Proceedings of the International Conference on High-performance Computer Architecture. Washington D. C., USA:IEEE Press, 2005:328-339.

[12] 陈国良,吴俊敏,章　锋,等. 并行计算机体系结构[M]. 北京:高等教育出版社,2002.

[13] Martin M M K, Harper P J, Sorin D J, et al. Using Destination-set Prediction to Improve the Latency/Bandwidth Tradeoff in Shared Memory Multiprocessors [C]// Proceedings of the 30th Annual International Symposium on Computer Architecture. Washington D. C., USA:IEEE Press, 2003:206-217.

[14] Hum H H J, Goodman J R. Forward State for Use in Cache Coherency in a Multiprocessor System: US6922756[P]. 2005-07-26.

[15] Raghavan A, Blundell C, Martin M M K. Token Tenure: PATCHing Token Counting Using Directory-based Cache Coherence[C]//Proceedings of the 41st Annual IEEE/ACM International Symposium on Microarchitecture. Washington D. C., USA: IEEE Computer Society, 2008: 47-58.

[16] Martin M M K, Sorin D J, Bechmann B M, et al. Multifacet's General Execution-driven Multiprocessor Simulator (GEMS) Toolset [J]. ACM SIGARCH Computer Architecture News, 33(4), 2005:92-99.

[17] Woo S C, Ohara M, Torrie E, et al. The SPALSH-2 Programs: Characterization and Methodological Considerations[J]. ACM SIGARCH Computer Architecture News, 1995, 23(2):24-36.

编辑　陆燕菲