计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2017.20160142
54(4):775-786, 2017

MPD:结点具有多个并行缓存一致性域的 CC-NUMA 系统

陈继承　赵雅倩　李一韩　王恩东　史宏志　唐士斌
(高效能服务器和存储技术国家重点实验室(浪潮集团有限公司)　北京　100085)
(chenjch@inspur.com)

MPD: A CC-NUMA System with Clump Having Multiple Parallel Cache Coherency Domains

Chen Jicheng, Zhao Yaqian, Li Yihan, Wang Endong, Shi Hongzhi, and Tang Shibin
(*State Key Laboratory of High-End Server & Storage Technology* (*Inspur Group Company Limited*), *Beijing* 100085)

Abstract　Large-scale CC-NUMA system usually employs two-tier architecture to reduce the overhead of cache coherence and enhance the performance of system. In a two-tier system, various processors and a coherence chip are located in an intra-clump cache coherency domain, and various coherence chips are interconnected by a system interconnection network so as to form an inter-clump cache coherency domain. Since every processor occupies at least one processor ID number in the cache coherency domain, and the number of processor ID numbers that can be distinguished by every processor is limited, CC-NUMA system expands the scale only by increasing the number of clumps, not by increasing the scale of clump. This leads to the over-large number of clumps and complicated topology structure in a multi-processor system, thereby increasing the bandwidth and latency of cross-clump memory access. To solve this problem, we propose a new method to construct multi-processor system, called MPD, in which a clump has multiple parallel cache coherency domains. This method solves the problem of limited clump scale brought about by limited number of processor supportable by a processor in a domain. Compared with traditional CC-NUMA system, MPD system not only significantly reduces the system topological complexity, but also effectively improves the system performance. Theoretical analysis and simulation results show: compared with 32-way CC-NUMA system, MPD system constructed by same processors can achieve 75% reduction in the number of nodes, more than 40% savings in consistency directory storage, 27.9% average reduction in access latency and about 14.4% improvement in system performance.

Key words　CC-NUMA (cache coherence non-uniform memory access) system; two-tier architecture; multiple parallel cache coherency domain (MPD); coherence chip (CC); system scalability

摘　要　大规模高速缓存一致性非均匀存储访问(cache coherence non-uniform memory access, CC-NUMA)系统通常采用两级一致性域方法来降低缓存一致性协议维护开销,提升系统性能.两级一致性域系统中,多个处理器互连,形成结点内一致性域;多个结点互连,形成结点间一致性域.然而,受限于处理器直连能力与处理器可识别 ID 数,系统的单结点规模有限,系统规模的扩展不得不依靠增加结点数来实现,使得大规模 CC-NUMA 系统的结点间互连复杂度上升,跨结点访问带宽和延迟急剧增长,影响

收稿日期:2016-03-11;**修回日期**:2016-05-25
基金项目:国家"八六三"高技术研究发展计划基金项目(2013AA011701)
This work was supported by the National High Technology Research and Development Program of China (863 Program) (2013AA011701).
通信作者:赵雅倩(zhaoyaqian@inspur.com)

了系统性能的有效扩展.MPD 系统通过在结点内构建多个并行缓存一致性域,突破了处理器直连能力与可识别 ID 数对单结点规模的限制,能够大幅减少结点数量,并将部分结点间访问转化为结点内访问,实现系统性能的有效扩展.理论分析和实验结果表明:采用同规格处理器的 32 路系统中,结点内 4 个并行缓存一致性域的 MPD 系统可实现结点数目减少 75%、一致性目录存储开销节省 40%以上、平均访问延迟降低约 27.9%、系统整体性能提升约 14.4%.

关键词 CC-NUMA 系统;两级一致性域;并行缓存一致性域;一致性协同芯片;系统可扩展性

中图法分类号 TP303

缓存一致性维护是影响高速缓存一致性非均匀存储访问(cache coherence non-uniform memory access, CC-NUMA)系统性能的关键因素.早期的 CC-NUMA 系统中,处理器数量较少,各处理器间直接互连,系统采用单级一致性域设计即可满足系统性能需求.但随着系统规模的扩展,单级一致性域系统的处理器互连结构越来越复杂,消息全局性传播引发的网络阻塞延迟越来越大,缓存一致性维护开销急剧增长,严重影响系统性能与扩展性.因此,8 路以上的 CC-NUMA 系统通常采用两级一致性域设计抑制缓存一致性维护开销,即:多个处理器互连组成结点后形成结点内一致性域,多个结点互连组成系统后形成结点间一致性域,两级一致性域间的协议转换通过一致性协同芯片(coherence chip, CC)实现[1-3].该方法可将一致性维护操作尽量限制在局部区域以避免一致性消息的全局传播,避免了单级一致性域造成的系统互连结构复杂、跨处理器访问跳步数多、高负载下阻塞延迟急剧增长等难题[4],从而使系统性能得到有效扩展.

受限于处理器的直连能力和处理器可识别的 ID 数,CC-NUMA 系统所能构建的单结点规模有限,系统扩展只能通过增加结点数目来实现.但是,结点数目的增加会导致一致性目录存储开销上升、跨结点访问跳步数和延迟增大、系统规模无法进一步有效扩展.针对上述问题,当前 CC-NUMA 系统采用目录优化[5]、片外缓存扩展[6-8]、缓存数据预取[9-11]、一致性协议优化[12-14]等方法来降低跨结点访问频度,减少一致性开销.

然而,上述优化方法都是通过减少跨结点访问频度间接减少平均访问延迟,对系统拓扑结构没有改变,没有缩短跨结点访问路径和跳步数.针对该问题,本文提出了一种可任意配置结点内处理器规模的 CC-NUMA 系统——多并行缓存一致性域(multiple parallel cache coherency domain, MPD).该系统通过在结点内构建多个并行缓存一致性域来扩大单结点规模,使其不再受限于处理器直连能力和处理器可识别 ID 数,从而减少系统结点数量,简化系统拓扑结构,缩短访问路径和跳步数,直接减少系统平均访问延迟.同时,由于结点内的多个缓存一致性域之间是并行关系,连接至 1 个一致性协同芯片,共同构成结点内一致性域,所以,与传统 CC-NUMA 系统相比,MPD 系统并未增加缓存一致性域的层级.

1 两级一致性域 CC-NUMA 系统

为减少缓存远端访问频率,降低缓存一致性维护开销,CC-NUMA 系统通常将缓存资源划分为 2 个一致性同步域,使用两级缓存一致性协议对各一致性域进行一致性管理.图 1 是一个典型的多结

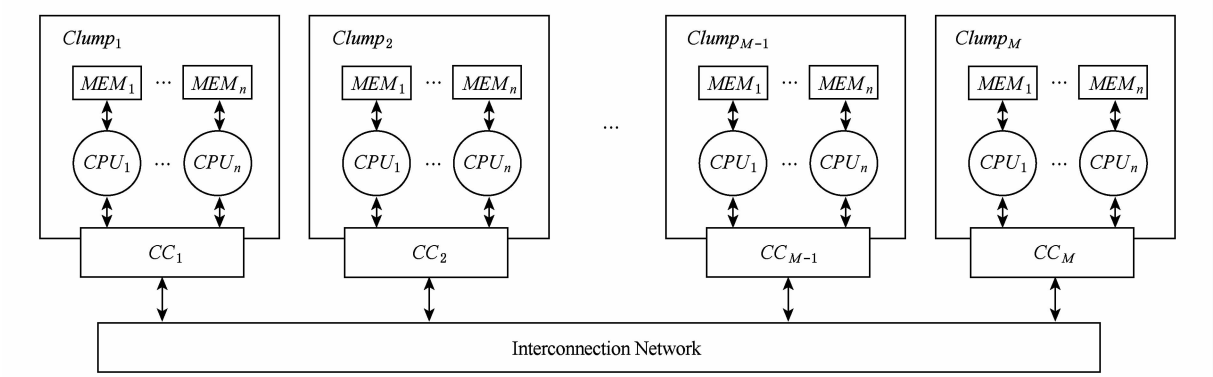


Fig. 1 CC-NUMA system with multiple clumps

图 1 多结点 CC-NUMA 系统

点 CC-NUMA 系统,包含结点内一致性域和结点间一致性域的两级缓存一致性域.

1) 结点内一致性域. n 个处理器与 1 个一致性协同芯片互连,构成结点内一致性域,维护的是本结点内 n 个处理器间的缓存一致性.

2) 结点间一致性域. M 个结点通过一致性协同芯片互连,构成结点间一致性域,维护的是系统内各结点间的缓存一致性.

两级一致性域系统中,一致性协同芯片用于维护两级一致性协议的转换和维护,该芯片通常包含远端代理(remote proxy, RP)和本地代理(local proxy, LP)两个处理单元.

1) RP. 结点一致性协同芯片的远端内存代理,存储远端地址数据在本地结点内处理器的一致性状态,监控本地处理器对远端地址的请求,可与本地处理器及远端结点一致性协同芯片的 LP 单元进行通信.

2) LP. 结点一致性协同芯片的本地内存代理,存储本结点地址的数据在其他远端结点的一致性状态,监控远端结点对本地地址的请求,可与本地根目录及远端结点一致性协同芯片的 RP 单元进行通信.

侦听与目录是对缓存一致性信息的 2 种处理方式[15]. 由于侦听协议可扩展性差,所以,CC-NUMA 系统多采用基于目录的缓存一致性协议. 结点一致性协同芯片的缓存一致性目录记录了各缓存行的一致性状态(State)、共享列表(Share List)和写权限拥有者(Owner),按缓存数据地址的不同,分别由 RP 与 LP 单元维护与更新.

RP 目录存储了远端数据在本结点的一致性状态信息,目录项如图 2 所示:

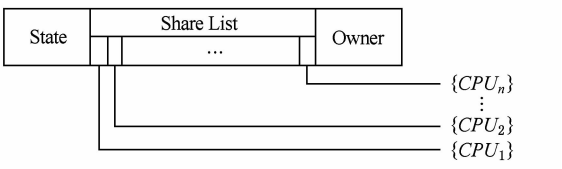


Fig. 2 RP directory entry in CC-NUMA system

图 2 CC-NUMA 系统结点一致性协同芯片 RP 目录项

1) State——状态位,记录远端数据在本结点的一致性状态,其长度与协议有关,如 MI 协议占用 1 b,MESI 协议占用 2 b;

2) Share List——共享列表,记录远端数据在本地结点内处理器的一致性状态,其长度与目录实现技术有关,如全映射目录[3]的共享列表长度为 n;

3) Owner——写权限拥有者,记录远端数据在本结点内处于 M/E 态的处理器 ID,其长度为 n.

LP 目录存储了本地数据在远端结点的一致性状态信息,目录项如图 3 所示:

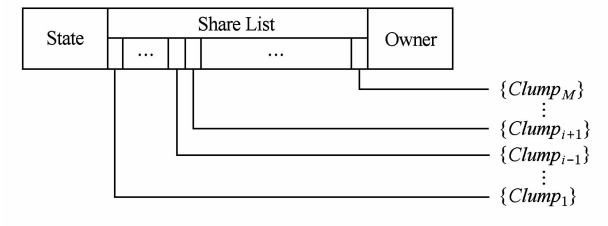


Fig. 3 LP directory entry in CC-NUMA system

图 3 CC-NUMA 系统结点一致性协同芯片 LP 目录项

1) State——状态位,记录本地数据在本结点的一致性状态;

2) Share List——共享列表,记录本地数据在本结点 i 之外的所有远端结点的一致性状态,全映射目录的共享列表长度为 $M-1$;

3) Owner——写权限拥有者,记录本地数据在远端结点处于 M/E 态的远端结点 ID,其长度为 $M-1$.

2 MPD 系统

2.1 MPD 系统结构

MPD 系统是一个结点内包含多个并行缓存一致性域的 CC-NUMA 系统. 如图 4 所示,n 个处理器与结点一致性协同芯片紧耦合互连,构成 1 个缓存一致性域(cache coherence domain, CCD);k 个缓存一致性域并行连接至 1 个结点一致性协同芯片,共同构成结点内一致性域;m 个结点通过结点一致性协同芯片互连,构成结点间一致性域,$m=M/k$. 与图 1 的 CC-NUMA 系统相比,图 4 的 MPD 系统结点数量缩减为原系统的 $1/k$.

MPD 系统中,结点内的多个缓存一致性域并行连接至 1 个一致性协同芯片,并由该协同芯片统一维护其结点内的缓存一致性,也就是说,这些并行的缓存一致性域并不单独维护一致性域. 所以,MPD 系统的一致性域设计与 CC-NUMA 系统相同,都是结点内/结点间两级一致性域架构,使用两级缓存一致性协议,两级一致性域间的协议转换通过一致性协同芯片实现.

1) 结点内一致性域. k 个缓存一致性域与 1 个一致性协同芯片互连,构成结点内一致性域,由本结点的一致性协同芯片统一维护结点内 nk 个处理器间的缓存一致性.

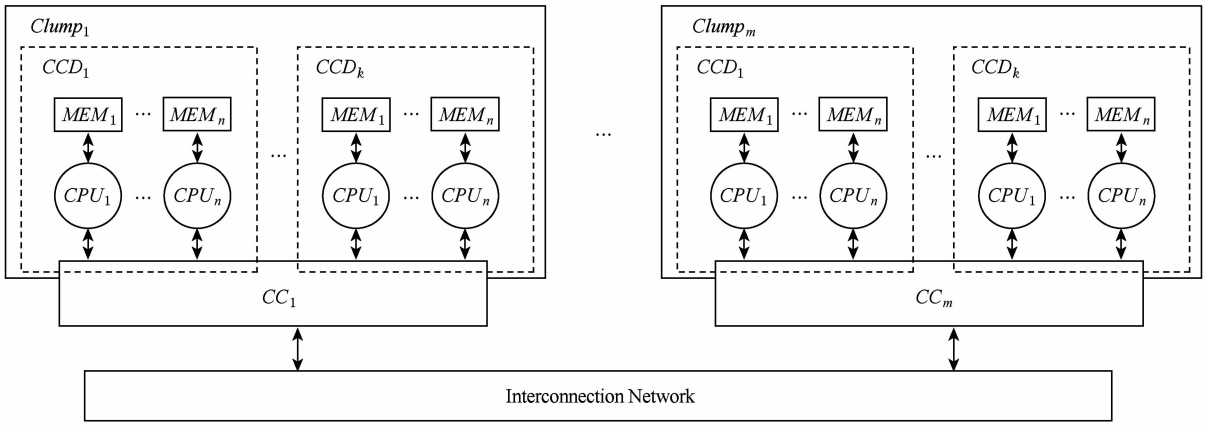


Fig. 4 MPD system with multiple clumps

图 4 多结点 MPD 系统

2) 结点间一致性域. m 个结点通过一致性协同芯片互连,构成结点间一致性域,由各结点的一致性协同芯片共同维护系统各结点间的缓存一致性.

与 CC-NUMA 系统类似,MPD 系统的结点一致性协同芯片也包括远端代理 RP 和本地代理 LP 这 2 个处理单元,但逻辑功能和通信单元略有差异,主要是对 LP 进行了功能扩展,以同时维护结点内所有处理器的缓存一致性.

1) RP. 结点一致性协同芯片的远端内存代理,存储远端地址数据在本地结点内(含本结点内所有缓存一致性域)处理器的一致性状态,监控本结点内所有处理器对远端地址的请求,可与本结点内处理器或远端结点一致性协同芯片的 LP 单元进行通信.

2) LP. 结点一致性协同芯片的本地内存代理,存储结点内各缓存一致性域的本地数据在本结点内其他缓存一致性域各处理器以及系统内所有远端结点的一致性状态,监控各缓存一致性域对结点内其他缓存一致性域的请求以及远端结点对本地地址的请求,可与本结点内处理器、根目录以及远端结点一致性协同芯片的 RP 单元进行通信.

从 CC-NUMA 与 MPD 系统中 RP 与 LP 的逻辑功能对比来看,MPD 系统中的 RP 与 LP 单元需要存储更多处理器的一致性状态信息,所以需要对 CC-NUMA 系统的一致性协同芯片目录的共享列表进行扩展.

RP 目录存储了远端数据在本结点内所有并行缓存一致性域的一致性状态信息,目录项如图 5 所示.

1) State——状态位,记录远端数据在本结点的一致性状态;

2) Share List——共享列表,记录远端数据在本地结点所有缓存一致性域内处理器的一致性状态,全映射目录的共享列表长度为 nk;

3) Owner——写权限拥有者,记录远端数据在本结点内处于 M/E 态的处理器 ID,其长度为 $\mathrm{lb}(nk)$.

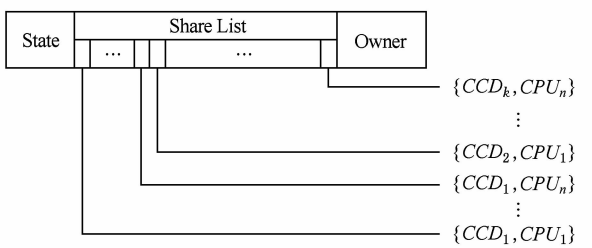


Fig. 5 RP directory entry in MPD system

图 5 MPD 系统 RP 目录项

LP 目录存储了本结点各缓存一致性域的本地数据在结点内其他并行缓存一致性域以及系统远端结点的一致性状态信息,目录项如图 6 所示.

1) State——状态位,记录本地数据在本结点的一致性状态;

2) Share List——共享列表,记录本地数据在远端结点及本地其他缓存一致性域的一致性状态,全映射目录的共享列表长度为 $\left(\frac{M}{k}-1\right)+n(k-1)$;

3) Owner——写权限拥有者,记录本地数据在远端结点处于 M/E 态的远端结点 ID,或缓存一致性域本地数据在结点内其他缓存一致性域处于 M/E 态的处理器 ID,长度为 $\mathrm{lb}\left(\left(\frac{M}{k}-1\right)+n(k-1)\right)$.

2.2 MPD 系统的缓存一致性访问

虽然 MPD 系统与 CC-NUMA 系统的缓存一致性域层次相同,但两者的系统架构不同,因此,2 种系统的缓存一致性交互流程略有不同.

1) 缓存一致性域内访问

MPD 系统与 CC-NUMA 系统的缓存一致性域结构相同,都是由多个处理器与一致性协同芯片互

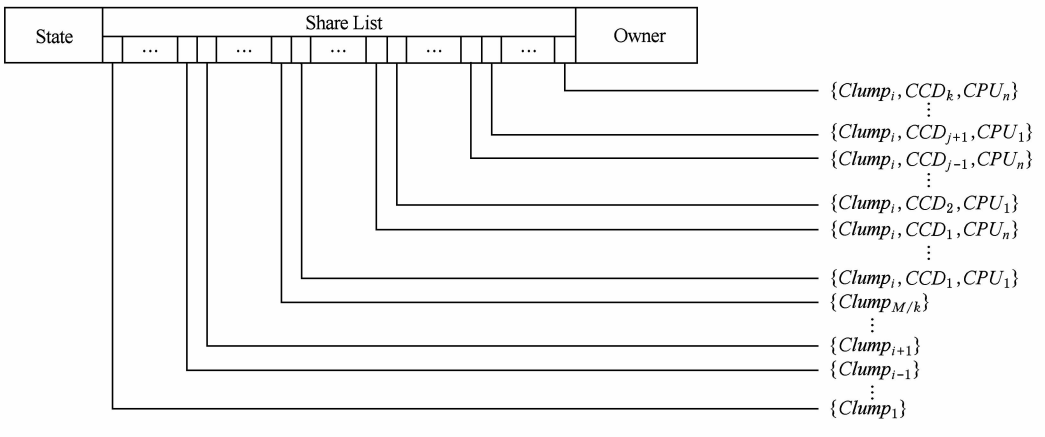


Fig. 6 LP directory entry in MPD system

图 6 MPD 系统 LP 目录项

连而成,因此,两者的缓存一致性域内访问流程相同,均为 CPU 与根目录直接交互,如图 7 所示:

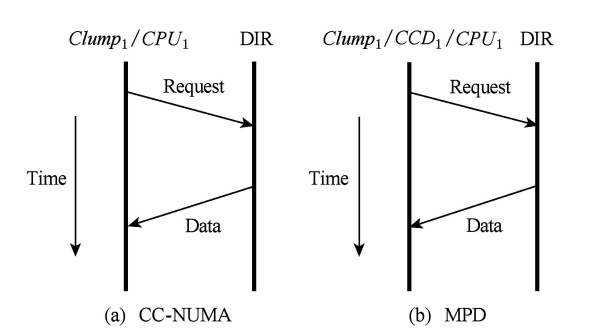


Fig. 7 Access within the same CCD in CC-NUMA or MPD system

图 7 CC-NUMA 系统与 MPD 系统的缓存一致性域内访问

2) 缓存一致性域间访问

CC-NUMA 系统内,结点内一致性域由单个缓存一致性域构成,所以,缓存一致性域间访问即为结点间一致性访问,如图 8 所示:

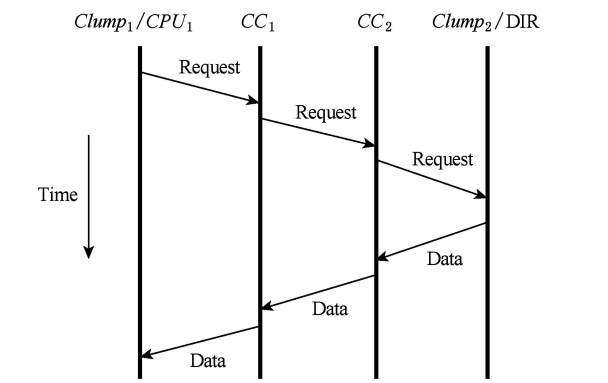


Fig. 8 Access between different CCDs in CC-NUMA system

图 8 CC-NUMA 系统的缓存一致性域间(结点间)访问

MPD 系统内,单个结点一致性域内包含多个缓存一致性域,因此,MPD 系统的缓存一致性域间访问包含了结点内的缓存一致性域间访问和结点间的缓存一致性域间访问 2 种类型,如图 9 所示:

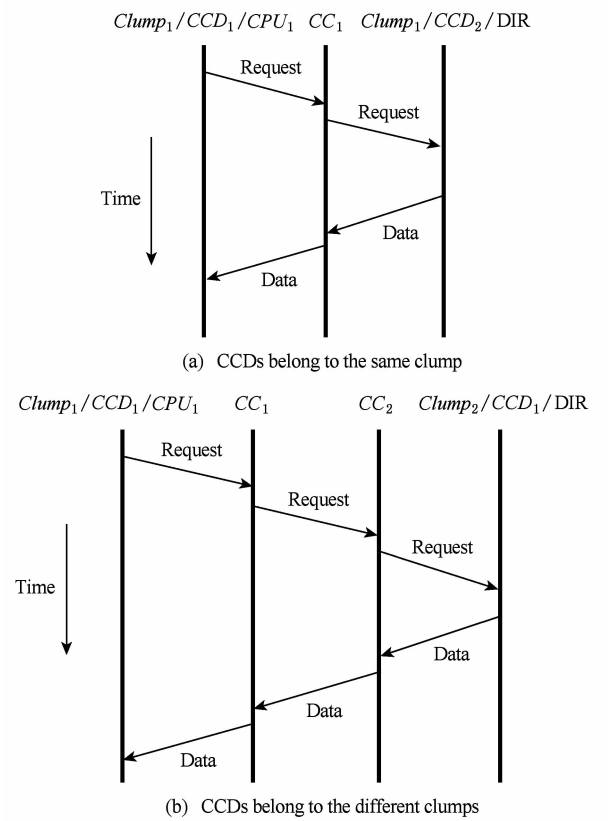


Fig. 9 Access between CCDs in MPD system

图 9 MPD 系统的缓存一致性域间访问

图 9(a)描述了 MPD 系统独有的结点内跨缓存一致性域访问,它将 CC-NUMA 系统中的远端跨结点访问转换为近端结点内跨域访问,仅与本地一致性协同芯片进行交互;图 9(b)描述了与 CC-NUMA

系统相同的结点间的缓存一致性域间访问,需与本地一致性协同芯片及远端一致性协同芯片进行交互.

从 2 种系统的缓存一致性交互流程描述来看,MPD 系统中的缓存一致性访问分为 3 类:

1) 结点内的缓存一致性域内访问. 该类访问对应 CC-NUMA 系统的结点一致性内访问(即缓存一致性域内访问),两者开销相同.

2) 结点内的缓存一致性域间访问. 该类访问对应 CC-NUMA 系统中的结点间访问,前者仅通过本地 CC 即可进行交互,而后者需要通过本地 CC 与远端 CC 才可进行交互,前者开销小于后者.

3) 结点间的缓存一致性域间访问. 该类访问对应 CC-NUMA 系统中的结点间访问,两者开销相同.

因此,与 CC-NUMA 系统相比,MPD 系统并未产生额外的一致性开销,并将部分结点间访问转换为结点内访问,直接缩短了访问路径与跳步数,从而降低系统平均访问延迟,减少一致性维护开销,提升系统性能.

3 MPD 系统性能分析

3.1 结点规模分析

MPD 系统通过在结点内构建多个并行缓存一致性域突破了单结点规模限制,减少了结点数量,从而降低了结点规模和结点间网络复杂度.

假设某系统规模(即处理器总数)为 N,单个缓存一致性域内最多容纳处理器数量为 n,那么,传统 CC-NUMA 系统和 MPD 系统的结点数 Num_C 以及结点间全互连的连接数 T 分别为

$$Num\_C_{\text{CC-NUMA}}=N/n, \tag{1}$$

$$T_{\text{CC-NUMA}}=N(N-n)/2n^2, \tag{2}$$

$$Num\_C_{\text{MPD}}=N/nk, \tag{3}$$

$$T_{\text{MPD}}=N(N-nk)/2(nk)^2, \tag{4}$$

其中,k 为 MPD 系统中单个结点内包含的并行缓存一致性域个数.

由式(1)~(4)可以得出,同等系统规模下,传统 CC-NUMA 系统和 MPD 系统的结点数量的比值 η_{Num_C} 为

$$\eta_{Num\_C}=\frac{Num\_C_{\text{MPD}}}{Num\_C_{\text{CC-NUMA}}}=\frac{N/nk}{N/n}=\frac{1}{k}. \tag{5}$$

结点间全互连的连接数比值 η_T 为

$$\eta_T=\frac{T_{\text{MPD}}}{T_{\text{CC-NUMA}}}=\frac{N-nk}{k^2(N-n)}<\frac{N-n}{k^2(N-n)}=\frac{1}{k^2}. \tag{6}$$

因此,与传统 CC-NUMA 系统相比,MPD 系统的结点规模可缩减至 $1/k$,结点间全互连的连接数可降低到 $1/k^2$ 以下. 例如,当 $N=64$,$n=4$,$k=2$ 时,MPD 系统的结点规模降低 50%,结点间全互连的连接数降低 77%;而当 $N=64$,$n=4$,$k=4$ 时,MPD 系统的结点规模降低 75%,结点间全互连的连接数降低 95%.

3.2 平均访问延迟分析

由于同一结点内不同缓存一致性域间的近端跨域访问延迟远小于不同结点间的远端跨结点访问延迟,所以 MPD 系统的平均访问延迟低于 CC-NUMA 系统,如图 10 所示:

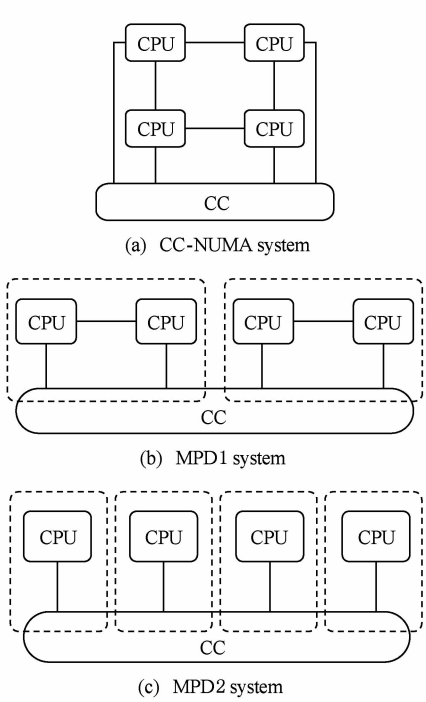


Fig. 10 Clump architecture in different systems built with different CPU

图 10 不同规格处理器搭建的系统单结点架构

假设 MPD 系统中同一缓存一致性域内的平均访问延迟为 l,同一结点内不同缓存一致性域间的平均访问延迟为 $3l$,不同结点间的平均访问延迟为 $7l$,缓存一致性域内的缓存次数占比为 α,近端结点内跨域的访问次数占比为 β,那么,MPD 系统的远端跨结点的访问次数占比为 $1-\alpha-\beta$,而相同规模的 CC-NUMA 系统的远端跨结点的访问次数占比为 $1-\alpha$. 因此,同规模的 MPD 系统与 CC-NUMA 系统的平均访问延迟 L 分别为

$$L_{\text{MPD}}=l\times\alpha+3l\times\beta+7l\times(1-\alpha-\beta)=l\times(7-6\alpha-4\beta), \tag{7}$$

$$L_{\text{CC-NUMA}}=l\times\alpha+7l\times(1-\alpha)=l\times(7-6\alpha). \tag{8}$$

由式(7)(8)可得 2 种系统平均访问延迟的比值为

$$\eta_L=\frac{L_{\text{MPD}}}{L_{\text{CC-NUMA}}}=1-\frac{4\beta}{7-6\alpha}\leqslant 1. \tag{9}$$

假设 $\alpha=10\%$,$\beta=30\%$,则 2 种系统的平均访问延迟之比为 13/16=0.8125,即 MPD 系统的平均访问延迟降低了近 2 成,有效提高了系统性能.

3.3 结点一致性目录存储开销分析

CC-NUMA 系统中,每个缓存一致性域单独维护 1 个结点一致性目录,结点数量较多,系统的结点一致性目录开销较大;而 MPD 系统中,多个并行缓存一致性域共同维护 1 个结点一致性目录,结点数量较少,系统的结点一致性目录开销较小.

以 MESI(modified, exclusive, shared or invalid)协议的全映射目录结构为例,假设 MPD 系统规模为 N,单个缓存一致性域内最多容纳处理器数量为 n,每个结点内共有 k 个缓存一致性域,内存容量为 B,缓存项大小为 b,则该系统内单个结点一致性目录开销为

$$d_{\text{MPD}}=\frac{B}{b}\times\frac{nk}{N}\times d_{\text{MPD}}^{\text{RP}}+\frac{B}{b}\times\frac{N-nk}{N}\times d_{\text{MPD}}^{\text{LP}}=$$
$$\frac{B}{Nb}\times\left(nk+n^2k+\frac{N^2}{nk}+nkN-nN+nk\times\text{lb}(nk)+(N-nk)\text{lb}\left(\frac{N}{nk}-1+n(k-1)\right)\right). \tag{10}$$

该 MPD 系统内所有结点的一致性目录总开销为

$$D_{\text{MPD}}=Num\_C_{\text{MPD}}\times d_{\text{MPD}}=$$
$$\frac{B}{b}\times\left(1+n+\frac{N^2}{n^2k^2}+N-\frac{N}{k}+\text{lb}(nk)+\frac{N-nk}{nk}\times\text{lb}\left(\frac{N}{nk}-1+n(k-1)\right)\right). \tag{11}$$

同等规模的传统 CC-NUMA 系统内单个结点目录开销为

$$d_{\text{CC-NUMA}}=\frac{B}{b}\times\frac{n}{N}\times d_{\text{CC-NUMA}}^{\text{RP}}+\frac{B}{b}\times\frac{N-n}{N}\times d_{\text{CC-NUMA}}^{\text{LP}}=$$
$$\frac{B}{Nb}\times\left(n+n^2+\frac{N^2}{n}+n\times\text{lb}\ n+(N-n)\text{lb}\left(\frac{N}{n}-1\right)\right). \tag{12}$$

该 CC-NUMA 系统内所有结点的一致性目录总开销为

$$D_{\text{CC-NUMA}}=Num\_C_{\text{CC-NUMA}}\times d_{\text{CC-NUMA}}=$$
$$\frac{B}{b}\times\left(1+n+\frac{N^2}{n^2}+\frac{N-n}{n}\times\text{lb}\left(\frac{N}{n}-1\right)\right). \tag{13}$$

由式(11)(13)可得,传统 CC-NUMA 系统与 MPD 系统的结点一致性目录总开销的差值 Δ_D 为

$$\Delta_D=D_{\text{CC-NUMA}}-D_{\text{MPD}}=$$
$$\frac{B}{b}\times\left(\frac{N}{k}\times(k-1)\times\left(\frac{N}{n^2k}+\frac{N}{n^2}-1\right)-\frac{N}{nk}\times\text{lb}\left(\frac{N}{nk}+n(k-1)-1\right)+\frac{N-n}{n}\times\text{lb}\left(\frac{N}{nk}+\frac{N}{nk}(k-1)-1\right)+\text{lb}\left(k+(n-1)(k-1)+\frac{N}{nk}-2\right)-\text{lb}\ k\right).$$

因为在大规模 CC-NUMA 系统中,

$$Num\_N_{\text{CC-NUMA}}=\frac{N}{n}>n,\ Num\_C_{\text{MPD}}=\frac{N}{nk}>\max(n,2),$$
$$Num\_C_{\text{CC-NUMA}}-Num\_C_{\text{MPD}}=\frac{N}{n}-\frac{N}{nk}>1,$$

所以,
$$\Delta_D>\frac{B}{b}\times\left(\frac{N}{k}\times(k-1)\times\frac{N}{n^2k}\right)=\frac{B}{b}\times(k-1)\times Num\_C_{\text{MPD}}^2{}^{①}.$$

因此,对于大规模 CC-NUMA 系统,同等规模的 MPD 系统的结点目录开销更小,而且,内存容量越大,目录开销降低的幅度越大.

以 64 颗处理器组成的 CC-NUMA 系统为例,假设每个缓存一致性域内最多可容纳 2 颗处理器,系统内存容量为 16 GB,单个缓存项大小为 128 KB.那么传统 CC-NUMA 系统共需 32 个结点一致性协同芯片,目录开销为[8×(1027+31×lb 31)]Mb,约为 9.223 Gb;而单结点内含 2 个并行缓存一致性域的 MPD 系统仅需 16 个结点一致性协同芯片,目录开销为[8×(293+15×lb 17)]Mb,约为 2.768 Gb,不到 CC-NUMA 系统目录开销的 1/3;单结点内含 4 个并行缓存一致性域的 MPD 系统仅需 8 个结点一致性协同芯片,目录开销为[8×(118+7×lb 13)]Mb,约为 1.124 Gb,不到 CC-NUMA 系统结点目录开销的 1/8.

3.4 系统构建成本与功耗分析

MPD 系统中,单结点规模不再受处理器直连能力的限制,因此,可以选用直连能力较弱的处理器代替直连能力较强的处理器搭建同等结点规模的系统,以降低系统成本与功耗.不同直连能力的处理器价格和功耗差异较大.支持 8 路直连的处理器有 3 个一致性互连端口,功耗约为 130 W,单价约为

① 该结论仅针对大规模 CC-NUMA 系统,单结点等小规模系统不保证不等式结论成立.

2 500 美元;支持 4 路直连的处理器有 2 个一致性互连端口,功耗为 115 W,单价为 2 000 美元;而支持 2 路直连的处理器仅有 1 个一致性互连端口,功耗为 100 W,单价为 1 500 美元.

假设要搭建 8 结点的 32 路系统,传统 CC-NUMA 系统需要 32 颗具有 8 路直连能力的处理器,结点内结构如图 10(a)所示;而结点内含 2 个并行缓存一致性域的 MPD1 系统则只需要 32 颗具有 4 路直连能力的处理器,结点内结构如图 10(b)所示;结点内含 4 个并行缓存一致性域的 MPD2 系统仅需要 32 颗具有 2 路直连能力的处理器,结点内结构如图 10(c)所示.

假定一致性协同芯片的单价为 1 000 美元,那么,CC-NUMA,MPD1,MPD2 这 3 种 8 结点 32 路系统的硬件成本和处理器功耗如表 1 所示. 相比于传统 CC-NUMA 系统,单结点内含 2 个并行缓存一致性域的 MPD1 系统可降低功耗 11.5%,降低成本 20%;单结点内含 4 个并行缓存一致性域的 MPD2 系统可降低功耗 23.1%,降低成本 40%.

Table 1 Comparison of Power and Cost Among Different Systems

表 1 不同系统功耗与成本对比

Item	CC-NUMA	MPD1	MPD2
Capability of Glueless Connection	8-way	4-way	2-way
Price of CPU/(10^3×USD)	2.5	2	1.5
Power of CPU/W	130	115	100
Number of CCDs in Each Clump	1	2	4
Total Power/kW	4.16	3.68	3.2
Total Cost/(10^3×USD)	81	65	49

4 系统性能模拟

4.1 实验平台

本文选用 gem5[16] 对两级缓存一致性域的 MPD 系统和 CC-NUMA 系统进行模拟,模拟系统中,系统结点无结点缓存功能,对接收的消息均做转发处理. 各级访问延迟比为:缓存域内访问延迟:结点内访问延迟:结点间访问延迟=1:3:7. 实验的主要系统参数如表 2 所示. 实验测试程序选用了 SPLASH2[17] 的 1 个内核程序和 3 个应用程序,以及 PARSEC[18] 中的 3 个应用程序,各配置如表 3 所示.

Table 2 Parameters of System Configuration

表 2 系统参数配置

Item	Value
Number of CPUs	16,32,64
Number of Cores	32,64
Number of CCDs in Each Clump	1,2,4,8
Cache Coherence Protocol	Two-tier MESI
Frequency of CPU/ GHz	1
CPU Model	Timing
Memory Size/GB①	1
Size of Cache Line	64
Set Associativity in L1 Cache	4
Set Associativity in L2 Cache	8
L1 Cache Size/KB	32(I)+64(D)
L2 Cache Size/KB	256
Topology of Network	2D mesh
Ratio of Access Latency	1:3:7

Table 3 Benchmark SPLASH2 and PARSEC

表 3 测试集 SPLASH2 与 PARSEC

Benchmark		Problem Size
SPLASH2	LU	1024×1024 matrix 16×16 blocks
	OCEAN_1 (Non_continuous Partitions)	258×258
	OCEAN_2 (Continuous Partitions)	258×258
	BARNES	16 384 particles, time-step is 0.025
PARSEC	ferret	256 queries, 34 973 images
	bodytrack	4 frames, 4 000 particles
	blackscholes	65 536 options

为验证 MPD 系统对系统性能有效扩展能力的提升,以及低功耗低成本 MPD 系统构建的可行性,本节分别对 16 路、32 路和 64 路的传统 CC-NUMA 系统与不同配置的 MPD 系统进行了对比测试. 测试实验数据显示,不同处理器规模的系统对比结果类似. 因此,为了避免重复,本节仅选用 32 路系统的对比测试数据进行说明. 另外,为便于数据比较的直观性,各组数据均以传统 CC-NUMA 的系统性能为标准进行了归一化处理.

① 本文实验设置较小的内存容量是为了避免部分缓存密集访问,以便更好地模拟大规模系统的多结点间的缓存一致性维护.

4.2 同规格处理器下的系统性能对比

4.2.1 平均访问延迟对比

图 11 显示了 CC-NUMA 系统与各 MPD 系统中不同类型访问的数量占比.图 11 中的横坐标 $m\times k$ 代表了系统的缓存一致性域规模:m 代表结点数,k 代表单结点内缓存一致性域数量.因此,$m\times 1$ 就代表了结点数为 m 的 CC-NUMA 系统.从图 11 中各类访问占比来看,由于 MPD 系统扩大了单结点处理器规模,有效减少了结点数量,部分结点间远端访问转换为结点内跨域近端访问,而且,单结点处理器规模越大,转换为结点内跨域访问的结点间访问量越多.图 11 中结点间访问量减少最多的是 OCEAN_NS.当单结点内有 8 个并行缓存一致性域时,OCEAN_NS 有 58%的结点间访问转换为结点内跨域访问.

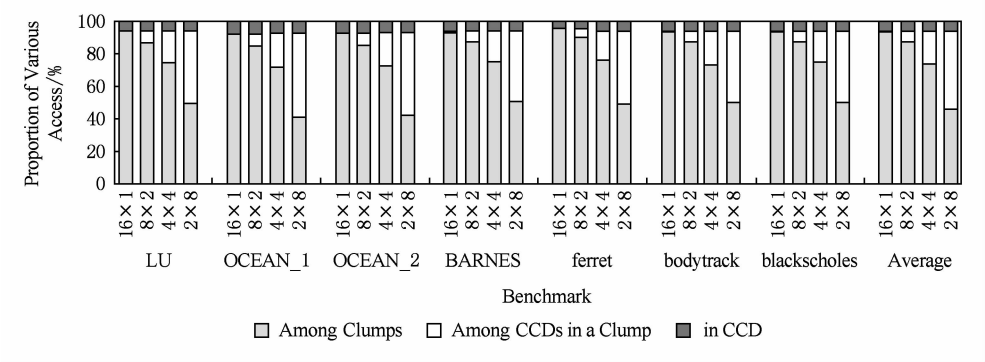


Fig. 11 Comparison of the access ratio among different 32-way systems

图 11 32 路 CC-NUMA 系统与 MPD 系统的各类型访问量对比

因为结点内跨域访问延迟远小于结点间访问延迟,所以结点间访问数量的减少直接降低了系统平均访问延迟.例如,单结点内有 8 个并行缓存一致性域时,OCEAN_NS 的系统平均访存延迟降低 32%.从图 12 的对比结果来看,系统平均访问延迟的变化趋势与结点间访问占比类似,均随结点内缓存一致性域数量 k 的增加而减少.当 $k=2$ 时,与 CC-NUMA 系统相比,MPD 系统的平均访问延迟降低 3.9%;当 $k=4$ 时,与 CC-NUMA 系统相比,MPD 系统的平均访问延迟降低 11.7%;当 $k=8$ 时,与 CC-NUMA 系统相比,MPD 系统的平均访问延迟降低 27.9%.

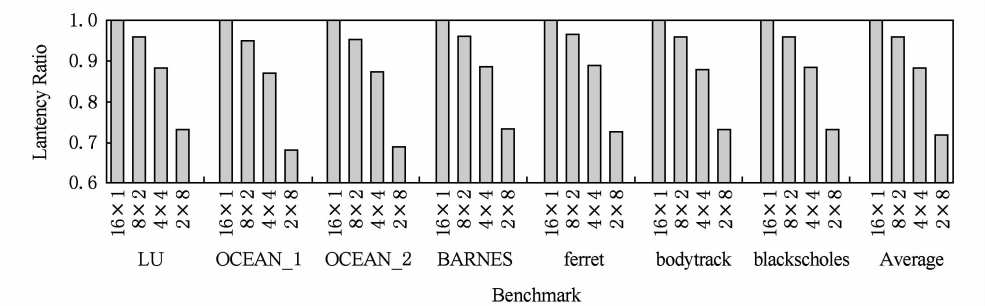


Fig. 12 Comparison of the average access latency among different 32-way systems

图 12 32 路 CC-NUMA 系统与 MPD 系统的平均访问延迟对比

4.2.2 指令平均执行周期对比

相较于系统平均访问延迟,CC-NUMA 系统和 MPD 系统的指令平均执行周期(cycles per instruction,CPI)差异较小,但仍是 MPD 系统性能较优,且系统 CPI 随结点内缓存一致性域数量 k 的增加而减少.这主要是因为,系统平均访问延迟仅与访问类型占比有关,而 CPI 还与程序指令数、指令类型等有关.图 13 中各程序的 CPI 变化幅度不同也是因为这个原因.尽管各测试程序的 CPI 降幅不同,但 MPD 系统整体平均性能仍呈现较明显的线性下降趋势:当 k 取值为 2,4,8 时,MPD 系统比传统 CC-NUMA 系统的 CPI 分别降低 3.6%,7.6%,12.6%,即 MPD 系统性能提升 3.5%,8.2%,14.4%.

4.3 不同规格处理器的系统性能对比

本文 3.4 节所述,MPD 系统中可采用直连能力较弱的处理器代替直连能力较强的处理器搭建同等规模的系统,以降低功耗、节约成本.但是,MPD 系统采用的是单结点内多缓存一致性域架构;原

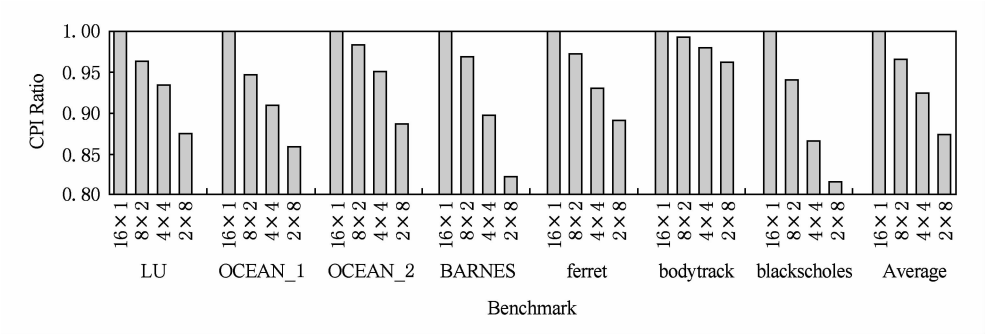


Fig. 13 Comparison of CPI among different 32-way systems

图 13 32 路 CC-NUMA 系统与 MPD 系统的 CPI 对比

CC-NUMA 系统中的部分缓存一致性域内访问将转换为结点内跨一致性域访问,从而导致系统平均访问延迟增加,性能下降.

为测试这种低功耗低成本的 MPD 系统与传统 CC-NUMA 系统的性能差异度,本节对不同规格处理器搭建的 4 结点 16 路系统、8 结点 32 路系统和 16 结点 64 路系统进行了性能测试. 各组测试数据均表明,与传统 CC-NUMA 系统相比,采用较低规格处理器搭建的 MPD 系统性能下降有限. 以图 14 中的 32 路系统为例,当处理器为 4 路直连时,每个结点内有 2 个缓存一致性域,系统 CPI 上升 1.7%;当处理器为 2 路直连时,每个结点内有 4 个缓存一致性域,系统 CPI 上升 2.6%,上升幅度变小.

与表 1 中处理器硬件成本的下降程度相比,这种 MPD 系统的性能下降幅度较小,因此,使用低规格处理器搭建的 MPD 系统具有更高的性价比. 如图 15 所示,与 8 路直连处理器搭建的 CC-NUMA 系统相比,4 路直连处理器搭建的 32 路 MPD 系统性价比提升 1.20 倍,2 路直连处理器搭建的 32 路 MPD 系统性价比提升 1.53 倍.

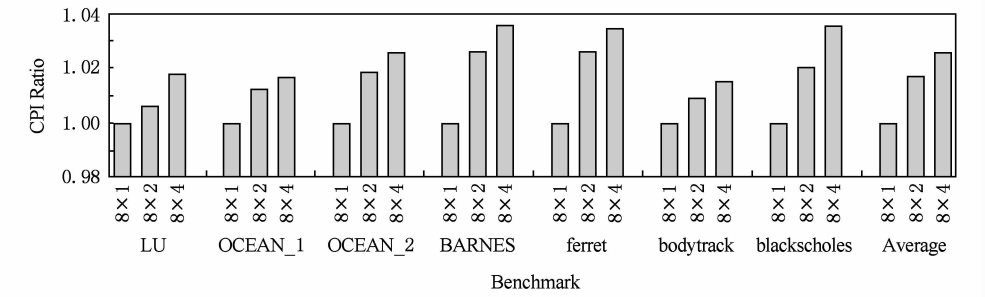


Fig. 14 Comparison of CPI among 8 clumps 32-way systems built by different processors

图 14 不同处理器搭建的 8 结点 32 路系统的 CPI 对比

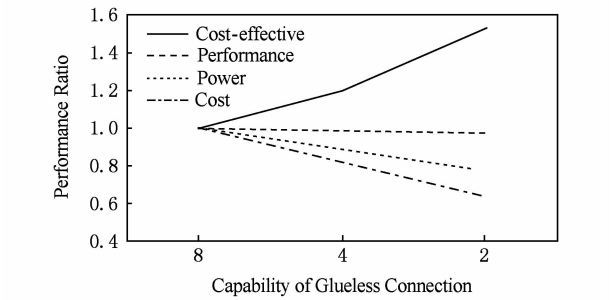


Fig. 15 Cost-effective of different systems

图 15 不同处理器系统的性价比

5 结 论

处理器直连能力和处理器可识别 ID 数的有限性限制了 CC-NUMA 系统的单个结点规模,导致系统规模的扩张只能通过结点数量的增加来实现. 而结点数量的增加致使缓存一致性维护开销增大,结点间互连结构复杂,降低了系统性能的有效扩展. 本文提出的 MPD 系统突破了单结点的处理器规模限制,能够任意设置单结点规模,大幅减少结点数量.

与其他系统性能优化方法相比,这种扩大单结点规模的方法直接缩短了系统平均访问路径和跳步数,从而降低了平均访问延迟,实现了系统性能的有效扩展.另一方面,由于 MPD 系统的单结点规模与处理器直连能力无关,MPD 系统还可用于构建低功耗低成本系统,提升系统性价比.实验数据表明,32 路系统中,采用同规格处理器时,结点内 4 个并行缓存一致性域的 MPD 系统平均访问延迟降低 27.9%,系统性能提升 14.4%;采用不同规格处理器时,与 8 路直连处理器搭建的 CC-NUMA 系统相比,2 路直连处理器搭建的 MPD 系统性价比提升 1.53 倍.

未来的工作中,我们将增加结点缓存功能,进一步优化 MPD 系统的一致性协议实现,提升大规模系统性能.

参 考 文 献

[1] Laudon J, Lenoski D. The SGI Origin: A ccNUMA highly scalable server [J]. ACM SIGARCH Computer Architecture News, 1997, 25(2): 241-251

[2] Lameter C. Numa (non-uniform memory access): An overview [J]. Queue, 2013, 11(7): 40-51

[3] Wang Endong, Hu Leijun, Zhang Dong, et al. Design and Implementation of High-End Fault-Tolerant Computer [M]. Beijing: Tsinghua University Press, 2015 (in Chinese)
(王恩东,胡雷钧,张东,等. 高端容错计算机设计与实现 [M]. 北京:清华大学出版社, 2015)

[4] Wang Endong, Chen Jicheng, Hu Leijun, et al. Design of high-end server based on tightly-coupled single-hop multi-plane architecture [J]. High Technology Letters, 2014, 24 (2): 111-116 (in Chinese)
(王恩东,陈继承,胡雷钧,等. 基于紧耦合单跳步多平面架构的高端服务器设计[J]. 高技术通讯, 2014, 24 (2): 111-116)

[5] Zhang Lunkai, Song Fenglong, Wang Da, et al. Improving the performance of sparse directories [J]. Journal of Computer Research and Development, 2014, 51(9): 1955-1970 (in Chinese)
(张轮凯,宋风龙,王达,等. 提升稀疏目录缓存一致性系统性能的方法[J]. 计算机研究与发展, 2014, 51(9): 1955-1970)

[6] Loh G H, Hill M D. Efficiently enabling conventional block sizes for very large die-stacked DRAM caches [C] //Proc of the 44th Annual IEEE/ACM Int Symp on Microarchitecture. New York: ACM, 2011: 454-464

[7] Sim J, Loh G H, Kim H, et al. A mostly-clean DRAM cache for effective hit speculation and self-balancing dispatch [C] // Proc of the 45th Annual IEEE/ACM Int Symp on Microarchitecture. Piscataway, NJ: IEEE, 2012: 247-257

[8] 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): 3:1-3:13

[9] Vanderwiel S P, Lilja D J. Data prefetch mechanisms [J]. ACM Computing Surveys, 2000, 32(2): 174-199

[10] Somogyi S, Wenisch T F, Ailamaki A, et al. Spatial memory streaming [J]. ACM SIGARCH Computer Architecture News, 2006, 34(2): 252-263

[11] Zhang Jun, Tian Ze, Mei Kuizhi, et al. Node predicting based direct cache coherence protocol for chip multi-processor [J]. Chinese Journal of Computers, 2014, 37(3): 700-720 (in Chinese)
(张骏,田泽,梅魁志,等. 基于节点预测的直接 Cache 一致性协议[J]. 计算机学报, 2014, 37(3): 700-720)

[12] Papamarcos M S, Patel J H. A low-overhead coherence solution for multiprocessors with private cache memories [J]. ACM SIGARCH Computer Architecture News, 1984, 12 (3): 348-354

[13] Culler D E, Singh J P, Gupta A. Parallel Computer Architecture: A Hardware/Software Approach [M]. San Francisco, CA: Morgan Kaufmann, 1999

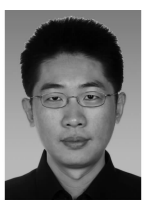
[14] Beers R H, Hum H H J, Goodman J R. Non-speculative distributed conflict resolution for a cache coherency protocol: US, US 6954829B2 [P]. 2005-10-11

[15] Sorin D J, Hill M D, Wood D A. A primer on memory consistency and cache coherence [J]. Synthesis Lectures on Computer Architecture, 2011, 6(3): 1-212

[16] Binkert N, Beckmann B, Black G, et al. The gem5 simulator [J]. ACM SIGARCH Computer Architecture News, 2011, 39(2): 1-7

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

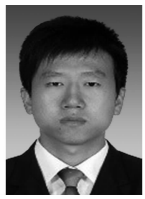
[18] Bienia C, Kumar S, Singh J P, et al. The PARSEC benchmark suite: Characterization and architectural implications [C] //Proc of the 17th Int Conf on Parallel Architectures and Compilation Techniques. New York: ACM, 2008: 72-81



Chen Jicheng, born in 1976. Received his PhD degree from Zhejiang University. Associate professor. Member of CCF. His main research interests include computer architecture, cache coherence and integrated circuit design.

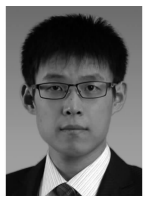
Zhao Yaqian, born in 1981. Received her PhD degree from Beihang University. Her main research interests include computer architecture and machine learning.

Li Yihan, born in 1985. Received his PhD degree from Beihang University. His main research interests include computer architecture and software debugging (liyihan@inspur.com).



Shi Hongzhi, born in 1988. Received his BEn degree from Xidian University. His main research interests include computer architecture and cache coherence (shihzh@inspur.com).

Wang Endong, born in 1966. Received his MSc degree from Tsinghua University. Professor. Fellow member of CCF. His main research interests include computer architecture, hybrid storage system and interconnect protocols for scalable system (wangendong@inspur.com).



Tang Shibin, born in 1986. Received his PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences. His main research interests include computer architecture, cache coherence and on-chip memory system (tangshb@inspur.com).

2017 年《计算机研究与发展》专题(正刊)征文通知

——应用驱动的网络空间安全研究进展

近年来,网络的多维度发展速度令人惊讶,使人兴奋.网络逐渐将虚拟世界与现实世界融合,人类在现实世界的活动越来越多地通过网络空间完成.智能设备、数字货币、物联网、云计算等正在改变人们生活方式.网络空间的安全因而成为大众广泛关注的焦点,没有网络安全的保障,就不会有现代化的生活.

网络空间的信息存储、传输、处理及应用,需要先进的安全技术做保障.区块链技术引起了金融、政府、企业各组织机构的广泛关注和热烈追捧,抗量子密码的标准化正在快速推进,安全的物联网与智能技术相结合将使梦想变为现实.

为推动我国学者在网络空间安全领域的研究,促进信息安全在基础理论、重要应用领域、新兴安全技术的发展,及时报道我国学者的最新研究成果,《计算机研究与发展》将于 2017 年 10 月出版网络与信息安全专题——应用驱动的网络空间安全研究进展,主要聚焦云安全、区块链技术、抗量子密码算法与协议、物联网安全等领域,欢迎相关领域的专家学者和科研人员踊跃投稿.

征文内容

本专辑包括(但不限于)下列主题:

- 密码算法与协议
- 安全大数据技术
- 网络系统安全;
- 网络数据共享的安全与隐私保护
- 安全云计算技术
- 电子货币与区块链技术
- 物联网安全

投稿要求

1) 论文应属于作者的科研成果,数据真实可靠,具有重要的学术价值与推广应用价值,未在国内外公开发行的刊物或会议上发表或宣读过,不存在一稿多投问题.作者在投稿时,需向编辑部提交版权转让协议.
2) 论文应包括题目、作者信息、摘要、关键词、正文和参考文献,论文一律用 word 排版,论文格式体例格式请参考《计算机研究与发展》近期文章.
3) 论文需附通信作者的联系地址、电话或手机及 E-mail 地址.
4) 论文请通过期刊网站(http://crad.ict.ac.cn)进行投稿,并在作者留言中注明"网络与信息安全 2017 专题"(否则按自由来搞处理).

重要日期

征文截止日期: 2017 年 6 月 11 日　　录用通知日期: 2017 年 7 月 20 日
作者修改稿提交日期:2017 年 8 月 6 日　　出版日期:2017 年 10 月

专题特邀编委

徐秋亮　教授　山东大学　xql@sdu.edu.cn
张玉清　教授　中国科学院大学、国家计算机网络入侵防范中心　zhangyq@ucas.ac.cn
董晓蕾　教授　华东师范大学　dongxiaolei@sei.ecnu.edu.cn

联系方式

联 系 人:徐秋亮 xql@sdu.edu.cn
编 辑 部:crad@ict.ac.cn,010-62620696,010-62600350
通信地址:北京 2704 信箱《计算机研究与发展》编辑部
邮政编码:100190