

片上多核处理器共享末级缓存动静结合地址映射机制

曹 非¹ 刘志勇²

(西北工业大学计算机学院 西安 710072)¹ (中国科学院计算技术研究所前瞻研究中心 北京 100190)²

摘 要 片上多核处理器(CMP)通常采用私有或者共享的末级高速缓存(cache)结构,而共享末级 cache 一般使用静态地址映射机制。该机制将各处理器临时私有访问的数据映射于分布在其他处理器的末级 cache 中,使得各处理器对临时私有数据的访问延时增加。针对该问题,提出了一种动静结合的共享末级 cache 地址映射方法。该方法可将原来静态映射于其他处理器末级 cache 中的临时私有数据动态映射于访问者处理器的本地末级 cache 中,减少了大量静态映射所造成的长延时非本地末级 cache 访问,从而有效降低了整个共享末级 cache 的访问延时,在提高性能的同时降低了功耗和带宽使用。实验结果表明,动静结合的地址映射方式应用于采用环连接互连结构和侦听顺序环协议的 CMP 结构时,可获得的平均性能提升为 9%,最大性能提升为 38%。

关键词 片上多核处理器,共享末级高速缓存,地址映射机制,环,侦听顺序环协议

中图分类号 TP302 **文献标识码** J

Combined Method of Dynamic and Static Address Mapping for Shared Last Level Cache of CMP

CAO Fei¹ LIU Zhi-yong²

(School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China)¹

(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)²

Abstract The shared last level cache(LLC) of CMP often uses a static address mapping method. This method may map some processor's temporary private data to other processor's last level cache. The processor needs longer access latency on these data than on the data mapped to local. This paper proposed a combined method of dynamic and static address mapping. The method can map most temporary data to their accessing processor's cache, so that these data's access latency can be reduced to local LLC access latency and the power and bandwidth of interconnection wasted for these data are saved. The experiment results show that the combined method of static and dynamic address mapping used in a CMP with a ring interconnection and SOR cache coherence protocol can obtain average performance increase of 9%, and the maximum is 38%.

Keywords CMP, Shared last level cache, Address mapping method, Ring, SOR cache coherence protocol

1 引言

片上多核处理器(CMP)中的末级缓存(cache)可以是私有或者共享的。私有的优点是各处理器仅需要访问位于本地的末级 cache,不需要访问位于其他处理器的末级 cache,末级 cache 访问延时短;其缺点是当某处理器的访问数据集大于其本地私有末级 cache 容量时,无法利用其他处理器的末级 cache 空余空间来存放部分数据,造成本地末级 cache 频繁失效,需要经常访问片外内存,从而增加访存延时。而共享末级 cache 的优点是可以充分利用片内的 cache 容量,尽量减少到片外访问内存的操作,从而降低访存延时;其缺点是共享的末级 cache 通常在物理上分布于各处理器,当采用静态地址映射方式时,处理器要访问的数据有可能映射于位于其他处理器的末级 cache 中,从而造成长于本地末级 cache 的访问延时。理想的末级 cache 能同时达到私有末级 cache 的 cache

访问延时短和共享末级 cache 的片外访存次数少的优点。

为达到或接近上述的理想设计目标,有 3 种思路。

1)在共享末级 cache 结构的基础上进行改进。基于该设计思路,文献[1]提出了基于共享二级 cache(L2)结构的被替换块复制(VR)和被替换块移动(VM)两种结构。通过使用本地 L2 区域来保存从一级 cache(L1)中替换出来的 cache 块,使未来对该块的访问在本地 L2 命中,将访问位于其他处理器的 L2 的延时降低为访问本地 L2 的延时。

2)在私有末级 cache 结构的基础上进行改进。基于该思路,文献[2]提出各私有 L2 间进行协作的方法以减少片外失效次数。所谓协作,是指当某个处理器的私有 L2 空间不足以放下全部的访问数据时,可以将部分数据放入相邻处理器有空闲空间的 L2 中,该处理器访问其私有 L2 失效时可以通过 cache 间的数据传递来访问放在其他处理器 L2 中的数据,以减少片外失效次数。

到稿日期:2011-11-08 返修日期:2012-02-17 本文受国家自然科学基金项目(60736012,60773223,61003037,61173047),国家“863”基金项目(2009AA01Z110)资助。

曹 非(1978-),男,博士后,主要研究方向为片上多核处理器设计,E-mail:caofei@nwpu.edu.cn;刘志勇(1946-),男,博士,研究员,博士生导师,主要研究方向为高性能算法与体系结构、并行处理。

3)使用混合结构。基于该思路,文献[3]提出一种每个核都有自己的私有 tag 阵列,所有核共享数据阵列的混合结构,实现了副本控制、当地通信和容量窃取 3 种机制,以达到同时减少 cache 访问延时和片外访存次数的目的。

上述 3 方面的工作都针对所有类型的数据进行优化,没有区分和利用不同的数据共享模式,均在不同程度上存在 cache 一致性协议和硬件结构复杂化、系统实现和验证困难等问题。

本文基于使用逻辑上共享、物理上分布于各处理器的末级 cache 的 CMP 结构,针对由于静态地址映射方式造成的各处理器临时私有访问的数据存放于分布在其他处理器的末级 cache 中,使得各处理器对临时私有数据的访问延时增加的问题,提出了一种动静结合的共享末级 cache 地址映射方法。该方法根据处理器所访问的数据是否还有其他处理器访问来动态检测数据是否临时私有,然后使用动态地址映射的方法,将各处理器的临时私有数据尽量映射于本地末级 cache 中。本文提出的方法可以减少大量静态映射所造成的长延时非本地末级 cache 访问,从而有效降低整个共享末级 cache 的访问延时,在提高性能的同时降低功耗和带宽使用。我们将该方法应用于使用侦听顺序环协议的环连接 CMP^[4],使用 Godson-Ring CMP 模拟器^[5]对动静结合和静态地址映射方式进行对比实验评估。实验结果表明,动静结合地址映射方式可获得的平均性能提升为 9%,最大性能提升为 38%;带宽降低百分比最高值达到了请求信息槽的 73.3%和数据信息槽的 44.8%;请求信息槽和数据信息槽传递次数降低百分比最高值为 84%和 66%。

2 动静结合的地址映射方法

采用共享末级 cache 结构的 CMP 通常使用静态地址映射方式。本节首先介绍该静态映射的共享末级 cache 结构及其不足之处,然后提出动静结合的地址映射方法并对其进行改进。

2.1 采用静态地址映射方式的共享末级 cache 结构

该结构中末级 cache 在逻辑上由 CMP 中的所有处理器共享,而物理上分布于各处理器。静态映射是指,由数据地址中固定的若干位(如高位交叉或低位交叉等)来决定数据存放的末级物理 cache 和该 cache 分布于的处理器,该 cache 和处理器分别称为数据的宿主(home)和宿主处理器。

位于末级 cache 中的数据的数据的宿主处理器访问该数据时仅需要访问本地末级 cache 即可,其访问延时仅为末级 cache 的访问延时。而非宿主处理器访问数据时需要经过互连网络来访问远程的 home,其访问延时包括访问请求通过互连网络到达 home 的延时、home 的访问延时以及从 home 得到数据后通过互连网络将数据传回请求者的延时。这两种数据访问的过程可以用图 1 来描述,图中 P0 到 Pn 表示 n 个处理器,而 LLC0 到 LLCn 表示分布于这 n 个处理器的末级 cache,interconnect 表示互连网络。

当数据在一段时间内仅由某个处理器单独多次访问时,称该数据为临时私有数据。如果该处理器是宿主处理器,则只需要访问本地末级 cache,否则则需要通过互连网络来访问远程的 home。在静态地址映射方式下,临时私有数据的访问者处理器有很大的可能不是该数据的宿主处理器,这样大量的临时私有数据访问需要通过互连网络来访问远程末级

cache。如果能将这些远程访问变成本地访问,则将大大降低处理器对临时私有数据的访问延时,提高处理器的性能,同时可以减少网络的信息流量,降低信息传递引起的功耗和带宽使用。为了达到该目的,需要有能将临时私有数据映射于位于访问者处理器末级 cache 的更加灵活的共享末级 cache 数据地址映射机制,因此我们提出了下述的动静结合地址映射方法。

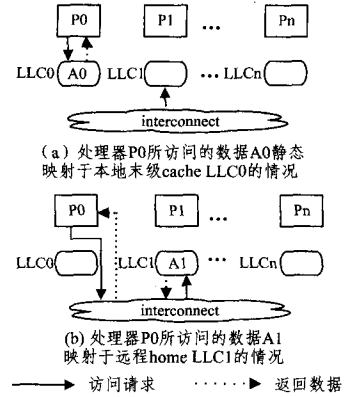


图 1 静态映射方式下宿主处理器和非宿主处理器访问数据的过程对比

2.2 动静结合地址映射方法

该方法以静态映射方式的共享末级 cache 结构为基础。针对临时私有数据的访问者处理器不是该数据的宿主处理器的情况,将这些数据动态映射于其临时私有访问者处理器的末级 cache 中,以减少临时私有数据的远程访问,从而改进处理器的性能、功耗和带宽等特性。

图 2 描述了该动静结合的映射方法,图中 P0 到 Pn 表示 n 个处理器,而 LLC0 到 LLCn 表示分布于这 n 个处理器的末级 cache,interconnect 表示互连网络。图 2(a)表示处理器 P0 第一次访问数据 A1 时将其放入由静态映射决定的 home(P1 的末级 cache)中。之后由 home 记录对 A1 的访问历史,以判断 A1 是由多个处理器共享访问还是仅由一个处理器单独访问。图 2(b)表示如果访问历史表明仅有处理器 P0 单独访问过 A1,则将该数据从 home 转移到处理器 P0 的末级 cache 中,并建立起该数据 A1 的地址到使用者处理器 P0 的动态映射关系。当 P0 再次访问此数据时,就根据动态映射直接到本地的末级 cache(LLC0)中取数据,不需要再到静态映射决定的 home(P1 的 LLC1)去取数,从而降低临时私有数据的访问延时。而当有多于一个处理器访问已经建立起动态映射关系的临时私有数据时,需要将该数据送回到静态映射决定的 home,并取消该动态映射关系。

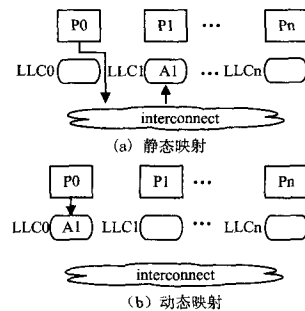


图 2 动静结合的映射方法

该方法的实现需要解决 4 方面的问题。(一)如何判断数据是否由某处理器临时私有;(二)如何表示动态映射关系;(三)如何区分共享数据和临时私有数据;(四)临时私有数据变回共享数据的检测方法。问题一的解决需要记录数据的访问历史,根据历史判断数据是否只有一个处理器访问过,从而预判该数据是否为临时私有。问题二需要有能记录较大的临时私有数据地址的集合,以便进行地址的加入和删除,从而实现动态映射关系的建立和取消的结构;并且其硬件实现应该结构简单、面积小,可以快速确定映射关系,以减少动态映射过程所增加的 cache 访问延时。问题三是由于用于静态地址映射的地址段不包含在末级 cache 的 tag 中,造成静态映射于不同处理器的末级 cache 的数据(数据的地址除了用于静态映射的地址段外相同)如果动态映射于同一个处理器的末级 cache 时,在访问 tag 时无法区分,就可能引起错误的访问。问题四是判断为临时私有的数据,随后可能由其他处理器访问从而变回共享的。该判断是一种根据数据访问历史的预测,可能不准确。随后可以根据是否有不同的处理器访问被判断为临时私有的数据,来决定是否需要将该数据变回共享,并解除该数据的动态映射关系,同时将数据放回静态映射决定的 home。这样做可以避免该数据再占用其动态映射到的末级 cache 的空间,使得各末级 cache 中的共享数据均匀放置。

针对上述问题,我们的解决方法如下。数据是否由某处理器临时私有的判断方法为:数据由内存取回时先按照静态映射关系放在 home 中,在 tag 中增加记录访问该数据处理器号的域,如果多次访问该数据的都是同一个处理器,则可以认为该数据为该处理器临时私有。

区分共享数据和临时私有数据的方法为:增加 cache 的 tag 段,用于区分是静态映射的共享数据还是动态映射的临时私有数据。所增加 tag 段即为用于静态映射的地址段。当增加的 tag 段静态映射决定的 home 处理器号和本地处理器号不同时为临时私有数据,相同时为共享数据。

临时私有数据变回到共享数据的检测方法为:如果使用侦听的 cache 一致性协议,则当非 home 侦听到其他处理器对临时私有数据的请求时,即可检测到临时私有数据到共享数据的转变。如果使用目录 cache 一致性协议,则需要由 home 来跟踪临时私有数据的所有者处理器,以便决定是否还有其他处理器访问了临时私有数据,引起临时私有到共享的转变,并通知临时私有数据的所有者将临时私有数据送回 home,变成共享数据。

由于布龙过滤器(bloom filter)^[6,7]具有和上述问题二的要求相吻合的特点,因此动态映射关系使用布龙过滤器来表示。该过滤器代表的是集合的超集,即集合的元素都在过滤器表示的超集中,并且过滤器所表示的超集还包括集合以外的元素。如果过滤器判断某个地址不在集合的超集中,则该地址就一定不属于此集合;而如果判断某地址在集合的超集中,则仅仅表示该地址有较大的可能性在此集合中。每个处理器都使用一个布龙过滤器来记录动态映射于本处理器的临时私有数据地址集合,过滤器可以方便地进行新的临时私有数据地址的加入和转变为共享数据的旧临时私有数据地址的删除。当过滤器判断某地址可能属于该集合时,需要访问本地 cache;而如果判断为不属于该集合,则需要到 home 或者其他处理器寻找该数据。

3 动静结合的地址映射方法在环连接 CMP 中的实现

我们提出了一种使用侦听顺序环协议的环连接 CMP 结构^[4]。本节首先描述该结构与协议,然后介绍上述动静结合的地址映射方法在该结构中的实现以及对侦听顺序环协议的优化,包括了对环连接 CMP 结构的 L2 地址映射方式、cache 结构和侦听顺序环协议等方面的改变。

3.1 环连接 CMP 结构和侦听顺序环协议

八处理器环连接 CMP 结构如图 3^[4]所示。图中 P1 到 P8 表示 8 个处理器,cache 分为两级,分别是一级 cache(L1)和二级 cache(L2)。L1 分成指令 cache(L1I)和数据 cache(L1D),对于所有者处理器私有。L2 物理上分布于各个处理器,逻辑上由所有处理器共享,其数据地址采用低位交叉的静态映射方式。单向分槽环位于 L1 和 L2 之间,连接所有处理器核。

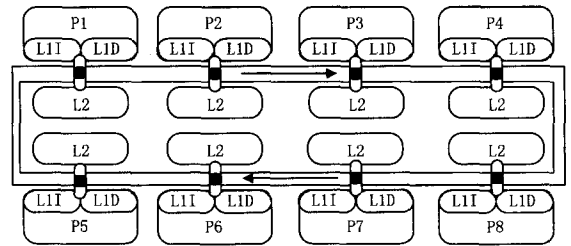


图 3 八处理器环连接 CMP 结构图

文献[4]中的侦听顺序环协议是一种侦听协议,协议利用了环的顺序特性有效避免了基于环的侦听协议^[8]所使用的重发应答操作和许多不必要的侦听操作,从而改进了系统的性能、功耗和带宽等特性。该协议还使用了灵活的侦听策略来避免可能出现的死锁和活锁情况。实验表明,该协议是一种硬件实现简单、非常适合 CMP 系统的高效协议。

3.2 L2 地址映射方式

在原 CMP 采用的 L2 低位交叉静态地址映射方式的基础上,增加动态映射的方式,每个处理器在访问或侦听 L2 时都需要经过动态映射和静态映射两种方式来决定是否访问或侦听本地 L2。用于动态映射的布龙过滤器结构如图 4 所示,图中 addr 表示数据的地址,被分成多个位段,用于索引不同的计数器序列。每个位段的每个值对应于一个计数器。当某个地址加入布龙过滤器所表示的地址集合时,该地址的每个位段所对应的计数器加 1,而当某地址从该集合中删除时,该地址的每个位段所对应的计数器减 1。判断某个地址是否属于该过滤器所代表的集合时,需要看该地址所有位段的对应计数器是否都大于 0,如果是,则该地址可能在此集合中,如果不是,则该地址一定不在该集合中。这里之所以说可能在集合中,是由于布龙过滤器表示的是一个集合的超集,会出现错误判断某地址属于集合的情况。

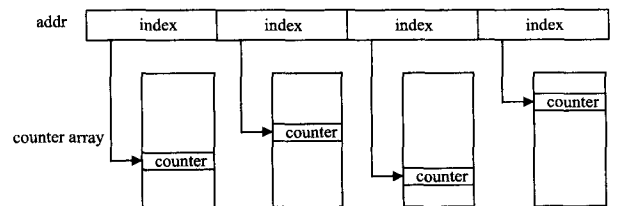


图 4 布龙过滤器的结构

在分布于 CMP 各处理器的 L2 中均加入一个这样的布龙过滤器,用于表示动态映射于各处理器的临时私有数据地址的集合。当来自本地处理器的 L1 失效请求访问 L2 或者来自其他处理器的失效请求侦听 L2 时,可以利用布龙过滤器快速决定数据不在本地 L2 中,以避免不必要的 L2 访问或者侦听操作,及由此产生的功耗。当过滤器判断数据可能在本地时,需要访问或侦听本地 L2,如果是来自本地的 L1 失效请求引起的访问命中且本地 L2 非 home,则直接返回数据,不需要再发请求到环上,减少了本地 L1 失效请求的访问延时;如果是来自其他处理器的失效请求引起的侦听命中,且本地 L2 非 home,则需要将临时私有数据到共享数据的转变,从本地的布龙过滤器中删除该地址,并且将数据传回 home。

Home 处理器访问本地 L2,或者其他处理器的失效请求引起侦听 home 时,如果不命中,在仅使用静态映射机制的情况下,就可以确定请求的数据不在 L2 中,同时立刻向内存发出失效请求。而如果使用了动静结合的映射机制,还无法判断数据是否不在 L2 中,则是因为数据有可能动态映射于某个处理器的 L2 中,此时需要将该失效请求发到环上,让所有其他处理器进行侦听。如果侦听结果表示该数据也不存在于其他所有处理器的 cache 中,则将结果带回给 home,由 home 向内存发出失效请求。可见使用动静结合映射相对于仅使用静态映射增加了 L2 失效时的访存延时,所增加延时为失效请求在环上传递一圈进行侦听的延时。为了尽量避免上述延时,可以再给每个处理器加入一个上述的布龙过滤器,用于记录以本处理器 L2 为 home 且动态映射于其他处理器 L2 的临时私有数据地址集合。当访问或者侦听 home 失效时,如果失效地址不在过滤器所代表的集合中,则可以立刻向内存发出失效请求,不需要再侦听其他处理器,从而避免了侦听延时。但由于布龙过滤器是集合的超集,还是会有少部分共享数据被错误判断为临时私有数据,从而引起侦听延时。

3.3 cache 结构

L2 的 tag 结构变化如图 5 所示,其中地址格式表示了静态映射 L2 地址的组成部分,而动静结合映射 L2 的 tag 由静态映射 L2 的 tag 和 home 选择两部分组成。这里的 home 选择部分就是用于静态映射决定 home 处理器号的地址位。

地址格式:

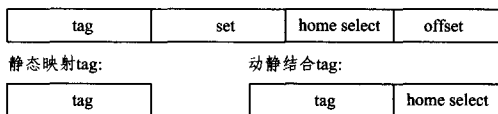


图 5 L2 的 tag 变化

L2 的 tag 阵列中加入表示数据访问历史和表示同一处理器访问该数据次数的域。历史域用于记录访问该数据的处理器号,当有新的处理器访问该数据时,将该处理器号和历史记录的处理号进行比较,如果不相同,则对访问次数域清“0”,并将新的处理器号记入历史域,如果相同,则对访问次数域加“1”。当访问次数域达到了预先设定的次数阈值时,认为该数据为请求者处理器临时私有,将数据传给请求者处理器,并在请求者处理器建立起动态映射关系。

建立了动态映射关系的临时私有数据从 L2 中替换出去时需要将该地址从布龙过滤器中删除,取消该动态映射关系。

3.4 侦听顺序环协议

在原协议的基础上增加两种数据消息类型:临时私有和

共享。临时私有类型数据消息在 home 侦听并判断请求者所请求数据为请求者临时私有产生,产生的同时 home 中的数据置为无效。该类型数据消息用于通知接收该数据的请求者处理器,将该数据地址加入布龙过滤器,建立起动态映射关系,同时将数据替换入该处理器的 L2 中。共享类型数据消息在建立了动态映射关系的临时私有数据侦听到其他处理器对该数据的失效请求时产生,用于通知 home 处理器在接收到该类型数据消息时将该数据替换入 home。共享类型的数据消息产生的同时,将该临时私有数据状态置为无效,并且将其地址从布龙过滤器中删除,取消动态映射关系。

增加 L1 数据块状态位,用于表示数据是否临时私有。当侦听 L1 时,如果数据为可读可写态(M),则需要根据该状态位来判断数据是否临时私有,如果临时私有,则产生共享类型的数据消息,并将该数据块状态置为无效。

4 实验评估

4.1 评估方法

本文使用 Godson-Ring CMP 模拟器^[5]对动静结合和静态地址映射方式进行对比评估。该模拟器以 Sim-Godson 模拟器^[9]作为处理器核建模的基础,龙芯 2 号处理器^[10]作为处理器核的硬件原型,使用事件驱动等模拟方法详细模拟了环形互连结构,其是对环连接 CMP 硬件结构和缓存一致性协议的硬件行为进行周期精确刻画的用户态模拟平台。

表 1 和表 2 列出了模拟器中处理器核和系统的参数。测试程序使用 SPLASH-2 测试程序集^[11]的一个子集。表 3 列出了它们的问题规模。

表 1 处理器核参数

参数	值与描述
流水线	4 发射,乱序
分支预测器	Gshare,4096-项 PHT
ROQ 大小	64 项
MSHR 大小	8 项
LSQ 大小	32 项
功能单元	2 个定点运算单元,1 个访存部件 2 个浮点运算单元

表 2 系统参数

参数	值与描述
处理器个数	8
L1I cache	64kB,4-路,2-周期访问延时
L1D cache	64kB,4-路,2-周期访问延时
L2 cache	8MB,4-路,4-周期访问延时
互连网络	单向分槽环
线延时	2 个周期
DRAM	100 个周期

表 3 测试程序

LU	512 * 512 matrix, 16 * 16 blocks
Ocean	130 * 130 array, 1e-7 error tolerance
Radix	256k keys, radix 1024
FFT	64k points
FMM	16k particles
Water-spatial	512 molecules

下面对使用 L2 地址静态映射和动静结合映射方式的 CMP 系统性能、带宽和功耗特性分别进行评估。图中都以 org 代表使用静态地址映射方式 L2 的 CMP,以 L2_D 代表使用动静结合地址映射方式 L2 的 CMP。

4.2 性能评估

图 6 是两种地址映射方式在运行不同的测试程序时的归一化运行时间。所有的运行时间都归一化为使用静态地址映射方式 CMP 的运行时间。

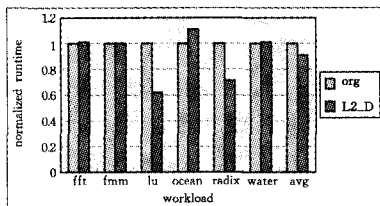


图 6 运行时间

从图中可以看出,动静结合映射方式相对于静态映射方式,在运行测试程序 fft, fmm 以及 water 时运行时间基本相同;运行 lu 和 radix 时均有较大的性能提升,分别为 38% 和 29%;而在运行 ocean 时有 11% 的性能下降。平均性能提升为 9%,最大性能提升为 38%。由于动静结合地址映射方式是针对临时私有数据访问延时的优化,因此测试程序中有越多的临时私有数据访问就会有越好的性能表现,比如 lu 和 radix。而如果程序中的临时私有数据访问较少,则该方法的效果就不明显,还可能对临时私有数据的错误判断,引起额外的数据传递和 L2 访问延时,使性能反而下降,比如 ocean。

4.3 带宽评估

图 7 和图 8 统计了请求信息和数据信息的平均带宽。纵轴表示环上平均每周期所使用的信息槽的数量;横轴表示不同的测试程序和它们的平均值。

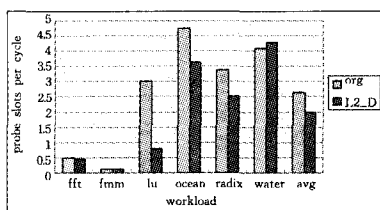


图 7 请求信息槽带宽

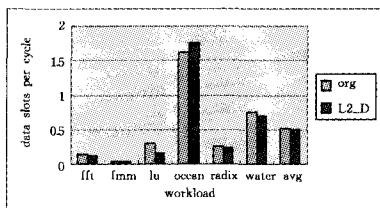


图 8 数据信息槽带宽

从图 7 中可以看出,动静结合方式仅在运行程序 water 时使用了更多的请求信息槽带宽,而在运行其他测试程序时总是使用更少的请求信息槽带宽。动静结合和静态映射方式运行所有测试程序的平均带宽分别为 1.98 和 2.64。动静结合方式所节约带宽的最大和平均值分别为 2.21 和 0.66,带宽节约百分比最大和平均值分别为 73.3% 和 25%。动静结合地址映射方式相对于静态地址映射方式对请求信息槽带宽的降低主要是由于处理器在访问动态映射到本地 L2 的临时私有数据时在本地命中,不会再在环上产生失效请求。

从图 8 可以看出,动静结合方式仅在运行程序 ocean 时使用了更多的数据信息槽带宽,而在运行其他测试程序时总是使用更少的数据信息槽带宽。动静结合和静态映射方式运行所有测试程序的平均带宽分别为 0.5 和 0.52。动静结合方式所节约带宽的最大和平均值分别为 0.13 和 0.02,带宽节约百分比最大和平均值分别为 44.8% 和 3.8%。动静结合地址映射方式相对于静态地址映射方式对数据信息槽带宽使用的降低,主要是由于处理器在访问动态映射到本地 L2 的临时私有数据时在本地命中,不需要再到远程的 home 处理器的 L2 去取数据,不会引起数据信息槽的传递。而运行 ocean 时带宽的增加是由于对临时私有数据的误预测,导致不必要的的数据转移,从而增加了数据信息槽的带宽。这种误预测可以通过调整预测时使用的同一处理器对同一数据块的访问次数阈值来减少,不过该阈值的增加也会引起延迟预测和漏预测,可能会影响性能的提高。

4.4 功耗评估

图 9 和图 10 显示了动静结合方式相对于静态映射方式所减少请求信息槽和数据信息槽传递次数的百分比。

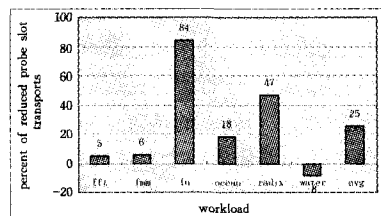


图 9 请求信息槽传递的减少

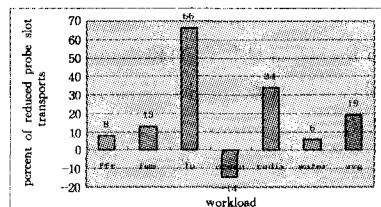


图 10 数据信息槽传递的减少

从图中可以看出,动静结合方式可以减少大量信息槽的传递,所减少请求信息槽最大和平均值分别为 84% 和 25%,而所减少数据信息槽最大和平均值分别为 66% 和 19%。图中信息槽传递减少最多的程序 lu 和 radix 也是性能改进最明显的两个程序。信息槽的传递操作会产生大量功耗,传递次数的减少就意味着功耗的降低。侦听操作会产生大量功耗,而请求信息槽传递的减少也意味着侦听操作的减少,也就意味着功耗的降低。

5 相关工作

文献[1]提出了基于共享 2 级 cache 结构的被替换块复制(VR)和被替换块移动(VM)两种结构,所谓被替换块是指从 L1 中替换出来的 cache 块。VR 结构的想法是使用本地 L2 来保存从本地 L1 中替换出来的 cache 块,该块是已经存在于 home 的 cache 块的复本,使未来对该块的访问在本地 L2 命中,将访问延时从访问外地 L2 降低到访问本地 L2。该方法的不足之处在于在 L2 中引入了同一数据的多个副本,

会消耗部分 L2 容量。VM 结构在 VR 结构的基础上给每个 L2 增加一个 VM tag 阵列,大小和相联度等同于原来 L2 的 tag 阵列,用于存储 home 的 L2 中在其他处理器的 L1 或 L2 中有共享块的 cache 块的 tag,而对应的数据块可以释放作他用,这样就避免了 home 中不必要的数据副本,相对 VR 增大了 cache 容量。这两种方法和本文的相同之处在于利用本地的共享末级 cache 来缓存部分 home 不在本地的数据,从而减少对远程 home 的访问,降低末级 cache 的访问延时。不同之处在于文献[1]的方法应用于目录协议,针对从 L1 中替换出来的数据进行缓存,VM 结构虽然减少了 VR 结构存在的多个数据副本对 L2 容量的消耗,但硬件开销较大,且每次都需要先访问本地 L2,如果不命中才访问 home 的 L2,造成了本地 L2 的无效访问,引起了访问 home 延时和本地 L2 访问功耗的增加。本文的方法可用于侦听或者目录协议,主要针对临时私有数据进行动态映射,不会引起共享末级 cache 容量的减少,并且利用布龙过滤器(bloom filter)^[6,7]避免了大部分对位于本地的共享末级 cache 的无效访问。

文献[2]针对私有 2 级 cache 的 CMP 结构,提出各私有 2 级 cache 间进行协作的方法,以减少片外失效次数。文献[3]提出一种每个核都有自己的私有 tag 阵列,所有核共享数据阵列的混合式 CMP 结构,以达到同时减少 cache 访问延时和片外访存次数的目的。这两种方法都针对所有类型的数据进行优化,没有区分和利用不同的数据共享模式,存在 cache 一致性协议和硬件结构复杂化、系统实现和验证困难等问题。我们的方法针对临时私有数据和共享末级 cache 的 CMP 结构,对协议的改动少,硬件简单,系统实现和验证相对容易。

为了消除目录协议中每次数据失效请求的间接目录访问带来的延时,文献[12]提出了利用上下文信息的 cache 一致协议,这里的上下文是指程序运行时所表现出的数据共享模式。文中提出了检测生产者消费者数据共享模式的方法,若检测到该模式,则将目录转移到生产者,消除了目录访问带来的间接延时,进而使用推测更新使得消费者的读请求变成本地失效。文献[12]和本文的相同之处在于,将部分对于远程 home 的失效访问变成本地失效,降低失效延时;不同之处在于,文献[12]中方法应用于目录协议,针对生产者消费者数据共享模式,而本文的方法可适用于侦听或者目录协议,主要针对临时私有数据访问延时的优化。

文献[6]首先提出了布龙过滤器,它可以通过一个长度有限的位向量签名(signature)有效表达一组数值的集合,支持数值插入该集合和判断某数值是否属于该集合两种操作。其不足之处在于,没有从集合中删除数值元素的操作,随着集合中加入的数值不断增多,过滤器受到污染,判断精度会不断降低。文献[7]提出了计数式布龙过滤器,它将原来的位向量变成了一系列的计数器组,通过计数器的加 1 减 1 实现元素的加入和删除,避免了原来过滤器无法删除元素造成的判断精度降低的问题。本文使用计数式布龙过滤器来记录各处理器访问的临时私有数据集合,建立起临时私有数据到分布于各处理器的共享末级 cache 的动态映射关系,利用布龙过滤器硬件实现简单、速度快的特点有效降低了临时私有数据的末

级 cache 访问延时,提高了性能。

文献[13]针对使用嵌入式单向环连接的多 CMP 系统,利用和本文类似的计数式布龙过滤器来过滤对每个 CMP 的侦听请求,以达到降低侦听延时和减少侦听操作、降低功耗的目的。文献[14]针对 SMP 利用和本文类似的计数式布龙过滤器来过滤不需要进行侦听的地址,以减少 L2 的侦听操作,降低功耗。文献[15]利用了一个简单的计数式布龙过滤器来记录本地缓存的地址区域集合,用于侦听时通知请求者本地是否有所请求数据地址所在的区域,以便请求者建立起一个仅本地访问的数据区域表,从而在下次访问该地址区域时不再广播失效请求,直接到本地访问,以减少失效请求,从而减少了侦听操作和功耗。本文利用计数式布龙过滤器来记录各处理器独占访问的临时私有数据,建立起临时私有数据到位于所有者处理器末级 cache 的动态映射关系,降低了各处理器对临时私有数据的末级 cache 访问延时,并减少了对临时私有数据失效请求的广播和临时私有数据从远程 home 到使用者处理器的移动,在提高性能的同时降低了功耗。

结束语 本文针对共享末级 cache 的 CMP 结构存在的静态地址映射将部分各处理器单独访问的临时私有数据映射到分布于远程处理器的末级 cache,导致临时私有数据访问的延时和功耗增加的问题,提出了动静结合的末级 cache 地址映射方式。利用 home 判断出临时私有数据,使用计数式布龙过滤器来记录临时私有数据,建立起动态映射关系,并将临时私有数据从远程末级 cache 转移到分布于使用者处理器本地的末级 cache 中。各处理器可以利用该动态映射关系直接在本地末级 cache 中访问临时私有数据,以降低临时私有数据的访问延时和功耗。将该方法应用于使用侦听顺序环协议的环连接 CMP 中,使用 Godson-Ring CMP 模拟器对静态和动静结合两种 L2 地址映射方式进行了对比评估。结果表明,动静结合相对于静态地址映射方式,平均性能提升为 9%,最大性能提升为 38%;请求信息槽带宽节约的百分比最大和平均值分别为 73.3%和 25%,数据信息槽带宽节约百分比最大和平均值分别为 44.8%和 3.8%;所减少请求信息槽传递次数百分比最大和平均值分别为 84%和 25%,所减少数据信息槽传递次数百分比最大和平均值分别为 66%和 19%。由此可见,动静结合地址映射方式可以有效地减少临时私有数据的远程访问,提高处理器性能,降低功耗和带宽使用。

参 考 文 献

- [1] Zhang M. Latency Reduction Techniques in Chip Multiprocessor Cache Systems [D]. Massachusetts Institute of Technology, 2006
- [2] Jichuan C, Gurindar S S. Cooperative Caching for Chip Multiprocessors[C]//Proceedings of the 33rd annual international symposium on Computer Architecture. IEEE Computer Society, 2006
- [3] Zeshan C, Michael D P, Vijaykumar T N. Optimizing Replication, Communication, and Capacity Allocation in CMPs[C]//Proceedings of the 32nd annual international symposium on Computer Architecture. IEEE Computer Society, 2005

- [4] Cao Fei, Liu Zhi-yong. Snooping and Ordering Ring-An Efficient Cache Coherence Protocol for Ring Connected CMP[C]// Proceedings of the Fifteenth International Conference on Parallel and Distributed Systems (ICPADS'09), 2009
- [5] 曹非. 环连接片上多核处理器缓存一致性协议研究[D]. 北京: 中国科学院计算技术研究所, 2010
- [6] Bloom B. Space/Time Tradeoffs in Hash Coding with Allowable Errors[M]. Communications of the ACM, 1970
- [7] Fan P C L, Almeida J, Broder A Z. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol[J]. IEEE/ACM Transactions on Networking (TON), 2000, 8: 281-293
- [8] Barroso L, Dubois M. Cache Coherence on a Slotted Ring[C]// Proceedings of the International Conference on Parallel Processing. 1991: 230-237
- [9] 张福新, 章隆兵, 胡伟武. 基于 SimpleScalar 的龙芯 CPU 模拟器 Sim-Godson[J]. 计算机学报, 2007, 30: 68-73
- [10] Hu F Z W, Li Z. Microarchitecture of the Godson-2 Processor [J]. Journal of Computer Science and Technology, 2005, 20: 243-249
- [11] Cameron W S, Moriyoshi O, Evan T, et al. The SPLASH-2 programs; characterization and methodological considerations[C]// Proceedings of the 22nd annual international symposium on Computer architecture S. Margherita Ligure, Italy, ACM, 1995
- [12] Liqun C. Context-aware coherence protocols for future processors[D]. University of Utah, 2007
- [13] Karin S, Xiaowei S, Josep T. Flexible Snooping; Adaptive Forwarding and Filtering of Snoops in Embedded-Ring Multiprocessors[C]// Proceedings of the 33rd annual international symposium on Computer Architecture. IEEE Computer Society, 2006
- [14] Andreas M, Gokhan M, Alok C, et al. JETTY: Filtering Snoops for Reduced Energy Consumption in SMP Servers[C]// Proceedings of the 7th International Symposium on High-Performance Computer Architecture. IEEE Computer Society, 2001
- [15] Andreas M. RegionScout: Exploiting Coarse Grain Sharing in Snoop-Based Coherence[C]// Proceedings of the 32nd annual international symposium on Computer Architecture. IEEE Computer Society, 2005

(上接第 283 页)

5.2 识别结果

为了比较算法的有效性,采用 CASIA-3DFACE 人脸库进行了实验。取库中的 30 幅中性表情的图像进行训练,取 40 幅表情人脸图像进行识别(30 幅有原型向导,10 幅没有原型向导)。使用文中算法表情弱化后识别得到结果:28 人在数据库中、11 人找不到、1 人识别错误。直接识别得到结果:23 人在数据库中、16 人未找到、1 人识别错误。通过分析识别结果发现,表情弱化后的识别中 2 人未能够准确匹配,原因是表情人脸同时带有幅度较大的偏转。表情弱化对表情变化大的情况效果更明显。

5.3 算法比较

用位平面法、特征块法、小波变换法等表情弱化算法与本文中算法比较。在 orl 人脸库上进行了实验,取库中的 100 幅中性表情的图像进行训练,然后取 50 幅表情人脸图像进行识别,在相同的条件下得到表 1 所列的识别数据。

表 1 不同算法的识别数据

识别项目 \ 算法	位平面	特征块法	小波变换	本文算法
正确识别数量	37	40	41	45
错误识别数量	4	3	2	1
未识别数量	9	7	7	4
识别准确率	0.74	0.80	0.82	0.90
平均耗时(s)	1.05	1.50	1.22	1.30

从以上的数据可以看出,位平面法、特征块法、小波变换法与本文算法比较,它们使用结构信息进行识别,虽然能够减弱表情对人脸识别的影响,但未明显提升识别准确性;使用基于向导的表情区域变换,时间无太大变化,识别准确率有明显提高,说明该算法是一种较好的算法。

结束语 首先介绍了常见的表情弱化的方法。在此基础上,提出了基于向导的局部表情变换方法,实现了局部表情弱

化处理,并在重建时添加了刚性区域约束条件,保证了类内相似度和类间的差异;将使用基于向导的表情区域变换方法与 PCA 算法进行比较,在不影响识别速度的同时,其识别准确率达到 90% 以上,说明该算法是一种较好的算法。但表情弱化时未考虑姿态等复杂人脸情况;另外,通过确定个体表情变化的特点来提高识别率需要进一步研究。

参考文献

- [1] Chua C S. 3D Human Face Recognition Using Point Signature [C]// Int'l Conf. on Automatic Face and Gesture Recogniton. 2000: 254-259
- [2] Chang K, Bowyer K, Flynn P. Adaptive rigid multi-region selection for handling expression variation in 3D face recognition[C]// IEEE Workshop on Face Recogniton Grand Challenge Experiments. 2005: 11-14
- [3] Jahanbin S, Choi H, Liu Y, et al. Three dimensional face recognition using ISO-geodesic and ISO-depth curves[C]// Proceedings of the 2nd IEEE International Conference on Biometrics; theory, Applications and Systems. Piscataway; IEEE Press, 2008: 1-6
- [4] Xu D, Li H. Geometric moment invariants[J]. Pattern Recognition, 2008, 41(1): 240-249
- [5] 胡平, 曹伟国, 李涛涛. 曲面上一种等距不变量的构造[J]. 系统科学与数学, 2009, 29(9): 1178-1188
- [6] 胡平, 曹伟国, 李华. 一类等距不变量及其在三维表情人脸识别中的应用[J]. 计算机辅助设计与图形学学报, 2010, 22(12): 2089-2094
- [7] 周竞文, 程志全, 金士尧. 基于 Dijkstra 距离剪枝的测地线求解算法[J]. 系统仿真学报, 2009, 21(1): 141-144
- [8] 王跃明. 表情不变的三维人脸识别研究[D]. 杭州: 浙江大学, 2007