面向非一致 Cache 的智能多跳提升技术

吴俊杰 潘晓辉 杨学军

(国防科学技术大学计算机学院并行与分布处理国家重点实验室 长沙 410073)

摘 要 非一致 Cache 体系结构(Non-Uniform Cache Architecture,NUCA)几乎已经成为未来片上大容量 Cache 的设计趋势.非一致 Cache 中,数据提升技术通过将经常访问的数据放置在距离处理器较近的 Cache bank 中减少处理器对该数据访问的等待时间,对 NUCA 的性能有着重要影响.然而,目前已有的数据提升技术使用固定的提升策略,没有考虑所要提升到目标 bank 的实际状态,容易将目标 bank 中更有用的数据"挤"得远离处理器,从而产生 Cache 污染问题,严重制约了提升技术的性能发挥.针对这一问题,文中提出智能多跳提升技术.智能多跳提升技术能够感知候选目标 bank 的状态,为被提升的数据动态地选择合适的目标 bank,从而提高了提升效率,减少了 Cache 污染.同时,智能多跳提升技术的设计巧妙地利用了处理器访问的反向路径,只是简单地扩充了处理器访问报文的格式,并没有增加对 Cache bank 的额外访问.最后使用全系统模拟器对来自 NAS Parallel Benchmark 和 Livermore Benchmark 的 15 个基准测试程序进行了详细测试,智能多跳提升技术单位提升操作节省的时钟周期数是已有提升技术的 1.50 倍,最多达到 2.61 倍;系统的 IPC 性能平均提高了 6.24%,最高达到 19.03%.

关键词 高速缓存;非一致高速缓存;提升;数据迁移;智能

中图法分类号 TP302 **DOI 号**: 10.3724/SP.J.1016.2009.01887

Smart Multihop Promotion for Non-Uniform Cache Architecture

WU Jun-Jie PAN Xiao-Hui YANG Xue-Jun

(*National Laboratory for Parallel and Distributed Processing*, *School of Computer*, *National University of Defense Technology*, *Changsha* 410073)

Abstract Non-Uniform cache architecture (NUCA) has almost been the trend of the large on-chip cache design. Data promotion technique moves frequently accessed data into cache banks closer to processors for reducing the access latency of NUCA. Hence, data promotion plays a significant role in NUCA optimizing. However, current promotion technique uses fixed promoting policy and does not take the real state of cache banks into account when promoting. Thus, it may push useful data in the promoted cache bank far from the processor, leading to cache pollution problem. This paper proposes smart multihop promotion, which can feel the status of the candidate banks for promotion and select the right target bank dynamically. The smart multihop promotion not only reduces the cache pollution problem, but also improves the performance of data promotion. Meanwhile, the smart multihop promotion makes use of the access path of processor and does not add extra accesses to cache banks. Finally, this paper gives test results to 15 benchmarks from NAS Parallel Benchmark and Livermore Benchmark using a full-system simulator. Compared with the current promotion technique, the evaluation results show that the smart multihop promotion achieves 1.50× improvement in saved cycles per promoting operation on average and 2.61× at best. Meanwhile, the approach improves IPC performance by 6.24% on average over current promotion and by 19.03% at best.

收稿日期:2009-07-15;最终修改稿收到日期:2009-08-26.本课题得到国家自然科学基金(60621003,60873014,60633050)、国家"八六三"高技术研究发展计划项目基金(2007AA01Z102)资助.**吴俊杰**,男,1981 年生,博士研究生,主要研究方向为计算机体系结构和编译技术.E-mail: junjie.ben@gmail.com.**潘晓辉**,女,1977 年生,硕士,助理研究员,主要研究方向为计算机体系结构、虚拟现实.**杨学军**,男,1963 年生,教授,博士生导师,主要研究领域为并行体系结构、并行操作系统和并行编译.

Keywords Cache; NUCA; promotion; data migration; smart

1 引 言

处理器和存储器之间的速度差异一直是计算机系统的性能瓶颈.通过捕获程序中的局部性,Cache技术能够过滤掉很多处理器对主存的直接访问,从而有效缓解存储墙问题.随着集成电路工艺的不断发展,单个芯片上能够集成的晶体管数目已经越来越多,在片内集成更大容量的Cache逐渐成为可能.随着Cache容量的不断增大,线延迟逐渐展现为制约Cache设计的主要问题.为了获得更高的集成度,芯片上全局导线越来越细,线间距也越来越窄,这些因素都导致线延迟的增加.如果采用传统的Cache设计方法,整个Cache的访问时间不得不迁就距离处理器最远的Cache bank的访问时间,将极大地增加Cache的访问延迟.为此,研究人员提出一种非一致Cache体系结构(Non-Uniform Cache Architecture,NUCA)[1],又称为分布Cache.NUCA允许Cache访问拥有不同的访问时间,当处理器访问较近的Cache bank时,等待时间较短;而当处理器访问较远的bank时,访问延迟就较长.NUCA几乎成为未来大容量Cache设计的趋势,学术界和产业界都非常关注.

NUCA上物理分布着很多Cache bank,这些bank通过互连网络进行连接,不同bank根据它们相距处理器的距离拥有不同的访问时间,因此,NUCA研究的一个关键问题就是如何使得处理器经常访问的数据尽可能被放置在距离处理器较近的bank中,从而缩短处理器访问这个数据的等待时间.在NUCA的研究中,动态NUCA允许数据根据访问的需要在bank间移动.动态NUCA里,一个Cache组(set)中不同路(way)的Cache块往往分布于多个bank.当处理器命中某个bank中的数据时,这个数据就被向距离处理器更近的bank移动一步或多步,从而在处理器再次访问该数据时缩短访问时间.这种移动称为数据提升(data promotion)或者Cache块迁移(block migration)[1-2].由于提升操作能够决定数据距离处理器的远近,因此,提升技术对NUCA的访问性能有着非常重要的影响.然而,目前已有的数据提升技术并不考虑目标bank的实际状态,很有可能将目标bank中更有用的数据挤出距离处理器较近的位置,产生Cache污染问题.特别地,往往距离处理器越近的数据也最有可能经常被访问,所以Kim等就曾发现,当数据提升的跳步数增加时,NUCA的访问性能并没有多少改进[1].可见,Cache污染问题严重制约了提升操作的性能发挥.

为了减少NUCA中数据提升操作的盲目性,本文提出一种智能多跳提升技术.智能多跳提升技术能够根据候选bank的状态动态地为提升操作选择合适的目标bank,减少Cache污染.本文的主要贡献包括:

(1)提出了面向非一致Cache的智能多跳提升技术.据我们所知,这项工作第一次提出和研究了感知目标bank状态的智能提升技术的概念和设计方法.

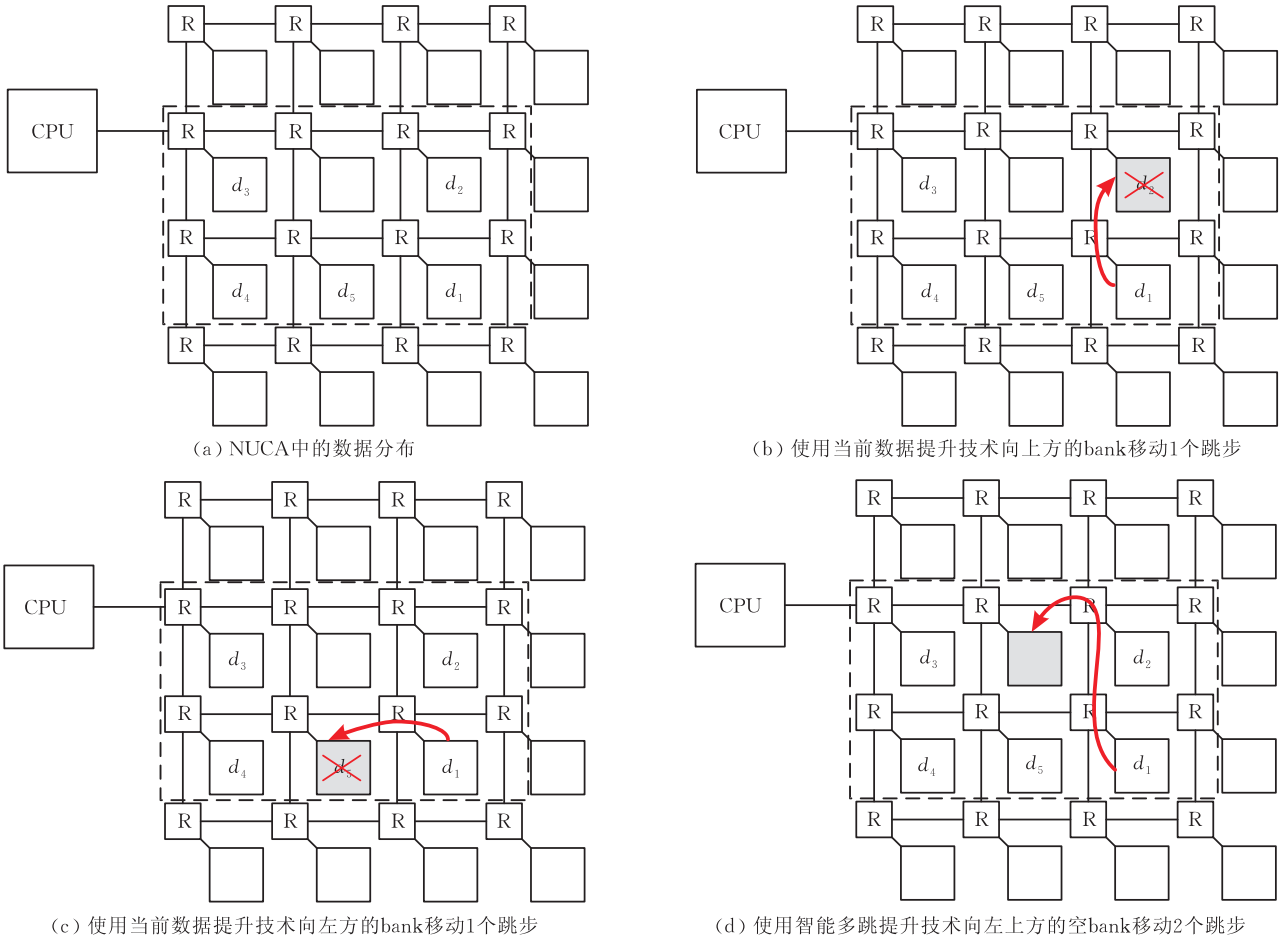
(2)使用来自NAS Parallel Benchmark[3]和Livermore Benchmark[4]的15个基准测试程序对智能多跳提升技术进行了详细评测,实验结果表明,智能多跳提升技术能够有效避免目前现有提升技术的盲目性,减小处理器访问延迟.在测试的8MB容量、16路组相联的L2 NUCA Cache中,相比目前已有的提升技术,智能多跳提升技术平均提高系统IPC性能6.24%,最高达到19.03%.

本文第2节介绍智能多跳提升技术的基本思想;第3节介绍智能多跳提升技术的具体设计;第4节和第5节对本文提出的方法进行了详细实验评测;第6节介绍本文的相关工作;最后,我们在第7节总结全文,展望未来工作.

2 基本思想

2.1 问 题

我们通过图1中的例子说明目前已有提升技术的问题.图1(a)是进行数据提升操作前的Cache视图,Cache共分为16个bank,组织成二维mesh结构,通过互连网络连接.图中用字母"R"标示的是路由器,每个路由器最多有5个端口,其中1个连接Cache bank,剩下的连接其他路由器.简单起见,我们只考虑虚线框中的bank,即提升操作只与这些bank有关.Cache bank中的文字代表在bank中存放的数据,这里,假设数据d_1、d_2、d_3、d_4、d_5属于同一个Cache组中的不同Cache line.数据d_1是要被提升的数据.



(a) NUCA中的数据分布　(b) 使用当前数据提升技术向上方的bank移动1个跳步

(c) 使用当前数据提升技术向左方的bank移动1个跳步　(d) 使用智能多跳提升技术向左上方的空bank移动2个跳步

图 1　当前数据提升技术和智能多跳提升技术

目前已有的数据提升策略由 NUCA 设计时确定的.假设这里采用 1 步(1 跳)的提升策略,那么,对于图 1(a)中的数据分布,对数据 d_1 的提升如图 1(b)或图 1(c)所示.由于在图 1(b)和图 1(c)中演示的目标 bank 中存放有数据 d_2 和 d_5,数据 d_1 的提升将使得数据 d_2 或 d_5 被替换出原有的 Cache bank.如果替换出的 d_2 或 d_5 是处理器经常访问的数据,d_1 的提升就会增加处理器下一次访问对 d_2 或 d_5 的等待时间,产生 Cache 污染问题.

2.2　解决方案

通过分析我们发现,提升操作造成 Cache 污染问题的原因大部分时候在于没有考虑目标 bank 当前存放数据的状态.比如图 1(a)中的情况,一方面,距离数据 d_1 一跳的 bank 分别包含数据 d_2 和 d_5;另一方面,距离 d_1 两跳的 bank(d_1 左上方的 bank)并不包含有效数据.因此,如果像图 1(d)那样将 d_1 提升两跳,移动至这个空的 bank 中,就可以避免 Cache 污染问题的发生.

从这个例子可以看出,没有考虑目标 bank 中的数据状态,盲目地使用设计时确定的提升策略是产生 Cache 污染问题,影响提升操作性能的关键.为此,我们提出智能多跳提升技术,智能多跳提升技术能够感知目标 bank 的数据状态,根据当前 Cache 中的数据分布智能地为提升操作选择目标 bank,从而有效避免和减少 Cache 污染.其中,"智能"体现在能够为提升操作找到合适的目标 bank,而"多跳"代表提升操作能够根据候选目标 bank 的情况选择跳步数,但并不是固定地选择某个跳步值.实际提升过程中,智能多跳提升技术可能一次将数据移动到距离处理器最近的 Cache bank 中,也可能只将数据向处理器方向移动一步.

本文的研究使用 Kim 等提出的 D-NUCA 结构[1].约定一个 Cache 组在每个 bank 中包含一路,使用增量的查找方式.当提升的目标 bank 包含有用数据时,将目标 bank 中的数据与源 bank 中的被提升数据进行交换.

3　智能多跳提升技术

从第 2 节可以看出,智能多跳提升技术选择

bank 的依据是候选 bank 当前包含数据的状态. 因此,智能多跳提升技术的关键是对 bank 状态的收集,如果对 bank 状态的收集过程需要额外的查找时间,无疑会对提升操作的性能产生重要影响,也会由于在互连网络中增加了过多的路由报文影响处理器对 NUCA 的正常访问.

幸运的是,我们发现提升操作的候选目标 bank 集合刚好是处理器访问已经遍历的 bank 集合,也就是说,目标 bank 在处理器访问时遍历的路径上. 如图 2 所示,当处理器发出访问数据 d_1 的请求时,NUCA 为这个请求生成访问报文,沿图 2 中的虚线遍历 bank,最终在虚线路径箭头所指的 bank 中命中数据 d_1. 随后启动数据提升过程,对数据 d_1 进行提升. 处理器访问报文按照距离处理器由近及远的顺序遍历了 NUCA 中的 bank,而包含数据 d_1 的 bank 刚好是这条路径的终点,也就是这条路径中距离处理器最远的 bank. 提升操作将数据向处理器方向移动,移动的目标 bank 刚好在处理器的访问路径上. 因此,智能多跳提升技术只需在处理器沿途寻找数据 d_1 时记录下访问过的 bank 状态,命中 d_1 后根据 bank 的状态完成目标选择即可.

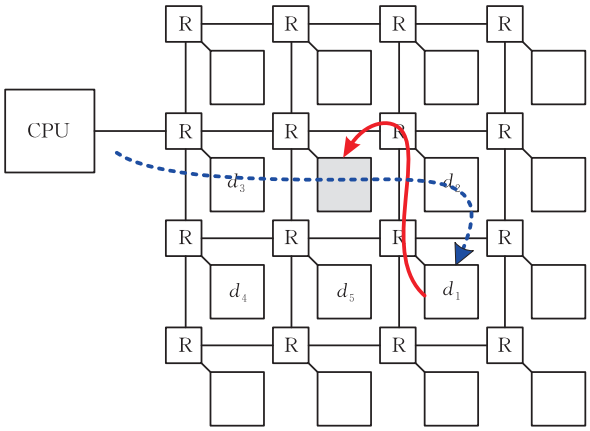


图 2 提升操作的目标 bank 在处理器访问的路径上

为了记录沿途 bank 的状态,我们扩展源处理器访问报文,在其中加入 bank 状态位向量,如图 3 所示. Bank 状态位向量共包含 $2n$ 位,其中,n 是 NUCA 所包含 bank 的数目. 每个 bank 在状态位向量中占有两位,分别是这个 bank 中对应 Cache 块的有效标志位(V)和修改标志位(D). V 为 1 代表这个 Cache 块包含有效数据,否则表示这个块是空的;在 bank 中包含有效数据的情况下,D 为 1 表示这个 Cache 块包含的是被处理器写过的脏数据(dirty data),D 为 0 表示块中是干净数据(clean data). 路由器负责在访问 Cache bank 后根据其中 Cache 块的状态修改 bank 状态位向量.

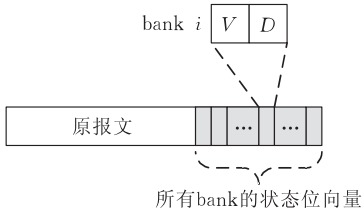


图 3 扩充的处理器访问报文

如图 4 所示,处理器访问报文在 NUCA 中遍历前,bank 状态位向量被置为全 1. 每当访问报文经过一个 Cache bank,根据这个 bank 中对应的 Cache 块状态修改访问报文中相应的位. 图 4 中,访问报文经过 bank B_{10} 时,由于其中包含干净数据 d_3,bank 状态位向量中 B_{10} 对应的位被修改为 10(有效干净数据);B_{11} 中的 Cache 块没有包含有效数据,因此,访问报文经过后,状态位向量改为 00;而 B_{12} 中包含的是脏数据 d_2,访问报文经过它时,状态位向量维持初始值 11.

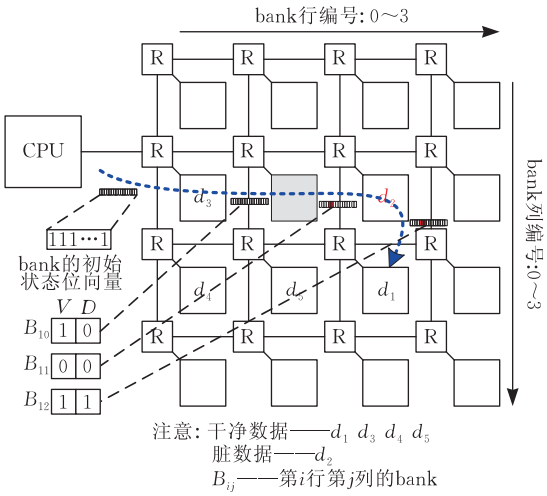


图 4 bank 状态位向量在 NUCA 中的传播

在处理器访问报文到达包含所请求数据的 Cache bank 时,已经遍历了提升操作的所有候选目标 bank,并已经将这些 bank 中对应 Cache 块的状态记录在 bank 状态位向量中. 提升操作开始前,智能多跳提升技术根据状态位向量为被提升的数据选择目标 bank,选择算法如图 5 所示. 最优先选择的是不包含有效数据的空 bank,因为将数据移动至这些 bank 中一定不会引起其它数据被挤得远离处理器. 如果在候选 bank 中有多个空 bank,则选择距离处理器最近的,因为这样能够最大程度地减少处理器下一次访问被提升数据时的等待时间. 如果处理器访问报文遍历的路径上没有空 bank,则优先选择包含干净数据的bank作为提升目标,因为NUCA

```
Algorithm FindTargetBank
input: BSBV, srcBID;
output: desBID;
//BSBV 是 bank 状态位向量
//srcBID 是源 bank
//desBID 是目标 bank

closestEmptyBank=-1;
farthestCleanBank=-1
//BID 为 i 的 bank 比 BID 为 i-1 的 bank 距离 CPU 远
for BID from bank 0 to bank srcBID-1
do
  if (BSBV[BID].V==0) then
  //编号为 BID 的 bank 是空的
    closestEmptyBank=BID;
    break;
  else if (BSBV[BID].D==0) then
  //编号为 BID 的 bank 包含干净数据
    farthestCleanBank=BID;
done

if (closestEmptyBank!=-1) then
  desBID=closestEmptyBank;
else if (farthestCleanBank!=-1) then
  desBID=farthestCleanBank;
else if (srcBID!=0)
  desBID=srcBID-1;
else
  return -1;

return desBID;
```

图 5　目标 bank 的查找算法

一般都是二级或三级 Cache,L2 Cache 中的脏块都是 L1 中脏块被替换时写回产生的,这意味着 L2 中的脏块相比干净块在 L1 中有副本的概率更小,处理器则更有可能会直接访问这些脏块(干净块会以更大的概率在 L1 Cache 中被命中),因此,我们优先选择包含干净数据的 bank 作为提升目标,这样可以将脏块尽可能留在距离处理器较近的 bank 中.如果有多个包含干净数据的候选目标 bank,我们选择距离处理器最远的,因为距离处理器越近的数据往往越有用(被重用的概率越大).

4　实验方法

为了评估智能多跳提升技术,我们修改了全系统模拟器 Simics 的 g-Cache 模块[5],使其支持 NUCA 结构和智能多跳提升技术.实验的详细配置如表 1 所示,8MB、16 路组相联的 L2 Cache 被组织成 NUCA 结构.我们使用 CACTI 6.0[6]得到 NUCA 的具体参数,见表 2,包含 16 个 Cache bank.测试的基准 Cache 为文献[1]中的 D-NUCA 结构.

表 1　实验配置

处理器	L1 指令 Cache	L1 数据 Cache	L2 (NUCA) Cache	操作系统
UltraSPARC-III, in-order	32KB,64B 块大小,4 路组相联	32KB,64B 块大小,4 路组相联	8MB,64B 块大小,16 路组相联	Solaris 10

表 2　由 CACTI 6.0 得到的 NUCA 参数

Bank 分布	水平线延迟/cycles	垂直线延迟/cycles	路由器延迟/cycles	Bank 访问延迟/cycles
4 行×4 列	2	2	3	6

我们使用的基准测试程序主要来自NAS Parallel Benchmark(NPB)[3]和 Livermore Benchmark[4],其中,mm、jac、lap 和 Livermore 的 7 个程序只包含算法核心,而 8 个 NPB 基准测试程序相对复杂,特别是 bt、lu 和 sp 更接近实际应用.表 3 中列出了程序的基本信息.所有程序都使用 GCC 4.2 编译器编译,并使用-O3 优化开关.根据 Kim 等的研究,跳步数对目前已有提升技术的性能影响并不大[1],因此我们使用跳步数为 1 的提升技术作为测试的基准配置.

表 3　测试程序的描述

测试程序	描述	指令数/M	L2 访问数/M	(提升操作数/L2 访问数)/%
bt	Block tridiagonal solver	698.7	58.9	18.63
cg	Conjugate gradient	490.8	15.2	35.84
ep	Embarrassingly parallel	2786.9	21.1	40.00
ft	3-D FFT PDE	821.9	39.4	47.96
is	Integer sort	20.5	1.3	11.48
lu	LU solver	330.3	12.0	31.09
mg	Multigrid	48.1	2.0	29.48
sp	Pentadiagonal solver	361.4	49.6	11.15
mm	Matrix multiplication	30.50	0.44	17.80
jac	Jacobi iteration algorithm	32.79	1.71	11.39
lap	Laplace transformation	53.15	3.79	12.76
iccg	Incomplete Cholesky Conjugate Gradient	24.70	0.71	21.57
es	Equation of state fragment	47.80	0.74	21.19
dp	Difference predictors	48.78	8.86	36.89
lre	General linear recurrence equations	44.62	0.62	25.53

5 实验结果与分析

图 6 列出了在智能多跳提升技术下系统 IPC 性能相比已有提升技术的改进.从图中可以看出,测试的程序在智能多跳提升技术的帮助下都得到了不同程度的性能提升,平均 IPC 增长率达到 6.24%,测试程序 dp 获得的性能提升最多,达到 19.03%.其中,7 个较简单的核心算法测试程序的 IPC 增长率平均为 4.67%,而 NPB 基准测试程序的增长率达到 7.61%. 这是因为在简单的算法核心程序中,数据访问往往较为规整,使得 Cache 中的数据分布也比较规整,而实际应用中,程序的复杂性使得 Cache 中的数据分布也更随机,而智能多跳提升技术能够感知 Cache bank 的动态信息,恰好能够适应这种实际应用中的复杂随机分布,因此,相比传统提升技术,更具性能优势.在所有的测试程序中,jac 和 lap 的性能提升最少,因为在基准测试中,这两个测试程序的提升操作在 L2 Cache 访问的比值非常小(见表 3),也就是说,提升操作对程序总的执行时间的贡献很小.

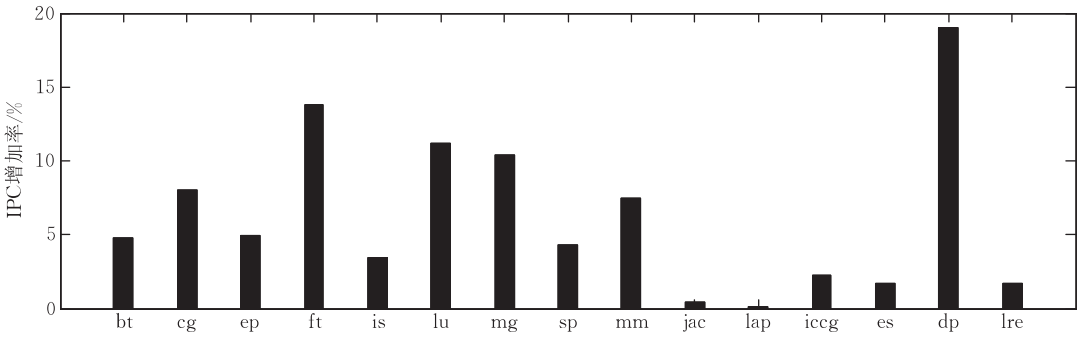


图 6 相比当前数据提升技术,智能多跳提升技术带来的系统 IPC 增加率

为了进一步分析智能多跳提升技术对目标 bank 的选择情况,我们统计了每个程序执行时的平均跳步数,如图 7 所示.从图中可以看出,不同程序的平均跳步数不同,比如,测试程序 cg 的平均跳步数最小,只有 1.04,而 iccg 的平均跳步数最大,达到 4.03.这充分说明了不同测试程序的运行状态不同,要根据访问当时 Cache bank 的动态状态决定提升操作的目标 bank 和跳步数,说明智能多跳提升技术十分必要.

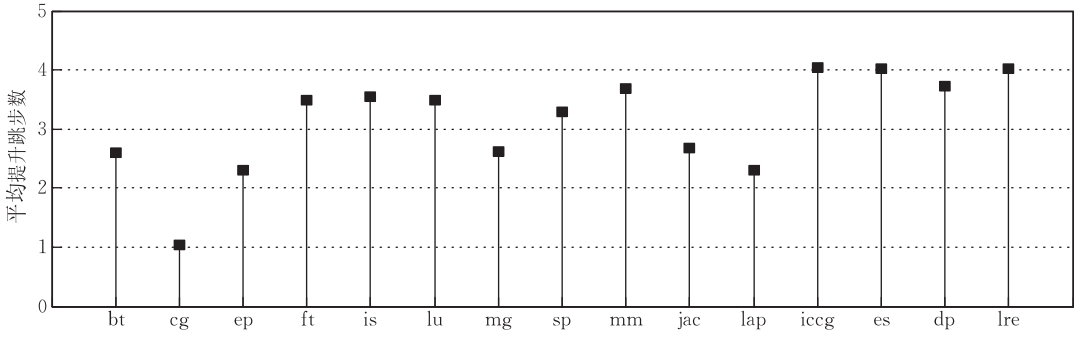


图 7 智能多跳提升技术下测试程序的平均提升跳步数

智能多跳提升技术除了能够智能选择跳步数,减少 Cache 污染外,还能够有效减少提升操作的数目,因为,智能多跳提升技术能够最快速地为被提升的数据找到最合适的目标 bank.如果距离处理器最近的 Cache bank 中没有包含有效数据,智能多跳提升技术就能够一步将被提升的数据移动到这个 bank 中,与已有的提升技术相比,减少了提升的次数,提高了提升效率.图 8 中列出了智能多跳提升技术对提升操作数目的减少率,大多数程序(除 ep 外)的提升操作数目都被不同程度地减少了,平均减少率达到 21.05%.

为了进一步考察智能多跳提升技术提升效率的变化,我们考察了单位提升操作对减少执行时间的贡献 SCPP(Saved Cycles Per Promotion).

$$SCPP=(Cycles_{\text{with\_promotion}}-Cycles_{\text{without\_promotion}})/Num_{\text{promotion}}.$$

图 9 中列出了已有的普通提升技术和智能多跳提升技术的 SCPP,它们已经依据普通提升技术的

SCPP 进行了正规化.在所有程序中,智能多跳提升技术都显著提高了 SCPP 值,在智能多跳提升技术中,单位提升操作节省的时钟周期是现有提升技术的 1.50 倍,最多达到 2.61 倍.可见,智能多跳提升技术提高了提升操作的效率,同时,这也会降低提升操作对互连网络带宽的占用,降低 NUCA 的动态功耗.

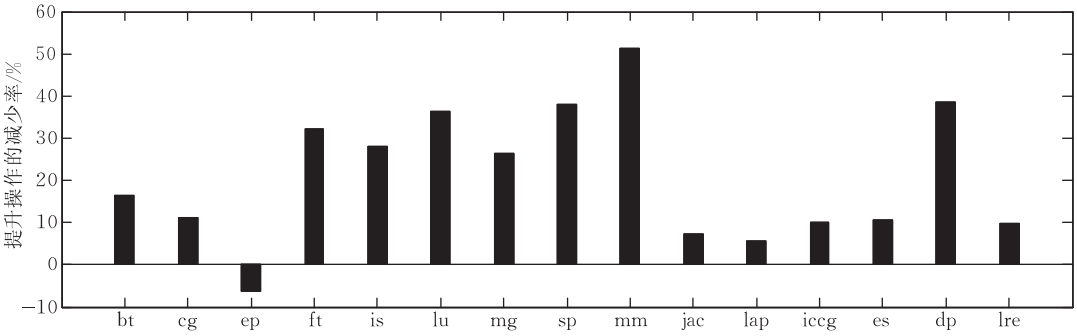


图 8　相比当前数据提升技术,智能多跳提升技术带来的提升操作的减少率

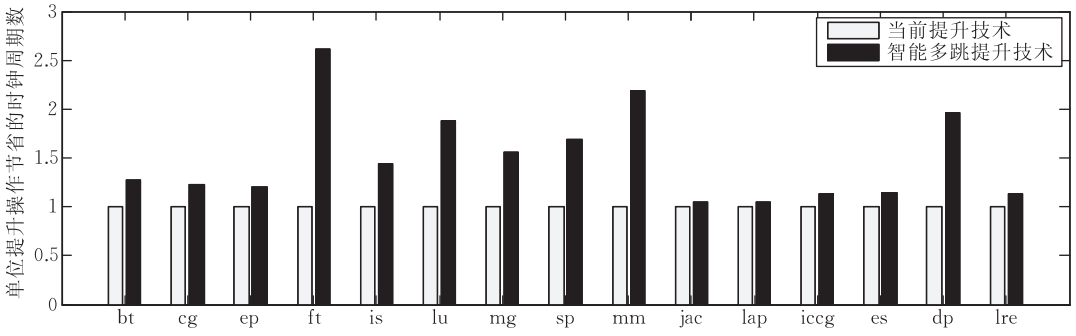


图 9　单位提升操作节省的时钟周期数(已根据当前提升技术进行归一化)

6　相关工作

面向非一致 Cache,本文提出了智能多跳提升技术.这一节,我们分别从 NUCA、NUMA 和互连网络 3 个方面比较与本文最相关的工作.

6.1　非一致 Cache

2002 年,Kim 等第一次提出非一致 Cache 体系结构(NUCA)[1].随后,各种 NUCA 的研究越来越多,NUCA 成为学术界和产业界关于大容量 Cache 设计的关注焦点[7-13].在 Kim 的研究中,一种动态 NUCA(D-NUCA)允许数据在不同的 bank 中移动,Kim 等提出了数据提升技术,也就是我们在实验中比较的基准提升技术,这种数据提升技术对数据的提升跳步数由 NUCA 设计时决定,在运行时不会动态改变,不可避免容易引起 Cache 污染,所以在 Kim 等的实验中,提升跳步数的增大并没有对系统 IPC 性能产生什么影响[1].Chishti 等曾提出距离相联 NUCA 结构——NuRAPID,NuRAPID 通过使用指针将 Tag 和 Data 解耦,一个 Cache 组中的数据并不一定总分布在多个距离组(Cache bank)中,根据需要,可能同一个组中的所有数据都被分布在一个 bank 中,因此,NuRAPID 的数据提升更加灵活,不依赖于 Cache 组的限制[7].然而,NuRAPID 中的提升操作也没有考虑目标 bank 中的数据状态,也不能避免发生数据污染.另外,需要指出的是,虽然本文的实验是以 D-NUCA 结构对智能多跳提升技术进行测试的,我们的方法本身并不依赖 NUCA 结构,也可以用于 NuRAPID 等其它不同的 NUCA 结构中.

6.2　非一致存储访问结构

非一致存储访问(Non-Uniform Memory Access,NUMA)是一种多处理机体系结构,又称为分布共享存储体系结构(Distributed Shared-Memory,DSM)[14].NUMA 结构中,每个处理器物理上拥有自己的本地存储器,而这些存储器在逻辑上是被所有处理器全局共享的.为了减小远程访问延迟对处理器性能的影响,如何优化数据在这些存储器上的分布是 NUMA 结构数据分布研究的关键技术之一[15-20],而提升技术则是 NUCA 结构上的数据分布技术.由于 NUCA 是一种 Cache 结构,在很多方面与 NUMA 也有着本质不同.在 NUMA 上的数据分布优化技术大都属于软件技术,而在 NUCA 上的提升技术是硬件技术,所以,提升技术的性能和实现开

销对其能否进入实际应用更加重要.而我们提出的智能多跳提升技术能够感知候选目标 bank 中的数据状态,极大地提高了提升操作的效率.同时,由于巧妙地利用了处理器访问的反向路径,智能多跳提升技术没有增加对 Cache bank 的额外访问.

6.3 互连网络

在我们提出的智能多跳提升技术中,扩展了目前已有的互连网络报文,并为路由器增加了选择目标 bank 的功能.然而,在我们实验的 16 路组相联 NUCA 结构中,增加的报文长度只有 32 位,与此同时,由于提高了提升操作的效率,提升操作却平均减少了 21.05%,提升操作报文也相应地减少了.对路由器功能增加的开销也远可忽略不计.可见,智能多跳提升技术对互连网络的修改远比其它片上网络技术来得少[21-23].

7 结束语

本文详细研究了目前已有 NUCA 结构中提升技术容易产生的 Cache 污染问题,并提出智能多跳提升技术.智能多跳提升技术能够感知提升的候选目标 bank 中的数据状态,智能地为被提升数据选择合适的目标 bank,从而有效避免 Cache 污染,提高提升技术的性能.通过巧妙地利用了处理器访问的反向路径,智能多跳提升技术没有增加对 Cache bank 的额外访问.本文详细介绍了智能多跳提升技术的设计,并使用全系统模拟器对 15 个来自 NAS Parallel Benchmark 和 Livermore Benchmark 的基准测试程序进行了详细评测.与目前已有的提升技术相比,智能多跳提升技术极大地提高了提升操作的效率,单位提升操作节省的时钟周期是现有提升技术的 1.50 倍,最多达到 2.61 倍;系统的 IPC 性能平均提高 6.24%,最高达到 19.03%.

参 考 文 献

[1] Kim C, Burger D, Keckler S W. An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches//Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X). New York, 2002: 211-222

[2] Beckmann B M, Wood D A. Managing wire delay in large chip-multiprocessor caches//Proceedings of the 37th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 37). Washington, DC, 2004: 319-330

[3] Bailey D H B, Barszcz E, Barton J T, Browning D S, Carter R L, Dagum D, Fatoohi R A, Frederickson P O, Lasinski T A, Schreiber R S, Simon H D, Venkatakrishnan V, Weeratunga S K. The nas parallel benchmarks. The International Journal of Supercomputer Applications, 1991, 5(3): 63-73

[4] McMahon F H. Livermore fortran kernels: A computer test of numerical performance range. Lawrence Livermore National Laboratory, Livermore, CA, USA: Technical Report UCRL-53745, 1986

[5] Magnusson P S, Christensson M, Eskilson J, Forsgren D, Hållberg G, Högberg J, Larsson F, Moestedt A, Werner B. Simics: A full system simulation platform. Computer, 2002, 35(2): 50-58

[6] Muralimanohar N, Balasubramonian R, Jouppi N. Optimizing nuca organizations and wiring alternatives for large caches with cacti 6.0//Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'07). Washington, DC, 2007: 3-14

[7] Chishti Z, Powell M D, Vijaykumar T N. Distance associativity for high-performance energy-efficient non-uniform cache architectures//Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 36). Washington, DC, 2003: 55-66

[8] Chishti Z, Powell M D, Vijaykumar T N. Optimizing replication, communication, and capacity allocation in cmps//Proceedings of the 32nd Annual International Symposium on Computer Architecture (ISCA'05). Washington, DC, 2005: 357-368

[9] Merino J, Puente V, Prieto P, Ángel Gregorio J. SP-NUCA: A cost effective dynamic non-uniform cache architecture. ACM SIGARCH Computer Architecture News, 2008, 36(2): 64-71

[10] Kandemir M, Li F, Irwin M J, Son S W. A novel migration-based nuca design for chip multiprocessors//Proceedings of the 2008 ACM/IEEE Conference on Supercomputing (SC'08). Piscataway, NJ, 2008: 1-12

[11] Li F, Kandemir M, Irwin M J. Implementation and evaluation of a migration-based nuca design for chip multiprocessors//Proceedings of the 2008 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'08). New York, 2008: 449-450

[12] Muralimanohar N, Balasubramonian R. Interconnect design considerations for large nuca caches//Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA'07). New York, 2007: 369-380

[13] Huh J, Kim C, Shafi H, Zhang L, Burger D, Keckler S W. A nuca substrate for flexible cmp cache sharing//Proceedings of the 19th Annual International Conference on Supercomputing (ICS'05). New York, 2005: 31-40

[14] Hennessy J L, Patterson D A. Computer Architecture: A Quantitative Approach. 4th Edition. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007

[15] LaRowe R P Jr, Holliday M A, Ellis C S. An analysis of dynamic page placement on a numa multiprocessor//Proceed-

ings of the 1992 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'92/PERFORMANCE'92). New York, 1992: 23-34

[16] Wilson K M, Aglietti B B. Dynamic page placement to improve locality in CC-NUMA multiprocessors for TPC-C//Supercomputing'01: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM). New York, 2001: 33-33

[17] Nikolopoulos D S, Artiaga E, Ayguadé E, Labarta J. Scaling non-regular shared-memory codes by reusing custom loop schedules. Scientific Programming, 2003, 11(2): 143-158

[18] Nordén M, Löf H, Rantakokko J, Holmgren S. Dynamic data migration for structured amr solvers. International Journal of Parallel Programming, 2007, 35(5): 477-491

[19] Löf H, Holmgren S. Affinity-on-next-touch: Increasing the performance of an industrial pde solver on a CC-NUMA system//Proceedings of the 19th Annual International Conference on Supercomputing (ICS'05). New York, 2005: 387-392

[20] Li Z. Array privatization for parallel execution of loops//Proceedings of the 6th International Conference on Supercomputing (ICS'92). New York, 1992: 313-322

[21] Eisley N, Peh L S, Shang L. In-network cache coherence//Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 39). Washington, DC, 2006: 321-332

[22] Park D, Das R, Nicopoulos C, Kim J, Vijaykrishnan N, Iyer R, Das C R. Design of a dynamic priority-based fast path architecture for on-chip interconnects//Proceedings of the 15th Annual IEEE Symposium on High-Performance Interconnects (HOTI'07). Washington, DC, 2007: 15-20

[23] Aggarwal N, Ranganathan P, Jouppi N P, Smith J E. Configurable isolation: Building high availability systems with commodity multi-core processors//Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA'07). New York, 2007: 470-481

WU Jun-Jie, born in 1981, Ph. D. candidate. His current research interests focus on computer architecture and compilation technology.

PAN Xiao-Hui, born in 1977, M. S., assistant researcher. Her current research interests focus on computer architecture and virtual reality.

YANG Xue-Jun, born in 1963, Ph. D., professor, Ph. D. supervisor. His main research interests include parallel computer architecture, parallel operating system and parallel compilation.

Background

The speed gap between processors and memories has always been the bottleneck of computer performance. Cache memory is an effective technique for alleviating the memory wall problem. Thus, cache studies always attract much attention of industry and academe. Along with the development of integrated circuits, cache capacity has become larger and larger leading that wire delay problem limits the traditional cache design method. In 2002, Non-Uniform Cache Architecture (NUCA) is proposed for large cache designs. When a processor accesses a NUCA, the access time may be one of many different ones. The closer to the processor the bank containing the data is, the faster the access operation is. The key problem concerned by this paper is how to move the frequently accessed data into cache banks closer to processors for reducing the access latency of NUCA. The data migration operation is called promotion. Current promotion technique uses fixed promoting policy and does not take the real state of cache banks into account. Hence, it may push useful data in the target cache bank far from the processor, which is so-called cache pollution problem. This paper proposes smart multihop promotion, which can feel the status of the candidate banks for promotion and select the right target bank dynamically. The approach proposed in this paper not only reduces the cache pollution problem, but also improves the performance of promotion. Meanwhile, the smart multihop promotion makes use of the access path of processor and does not add extra accesses to cache banks.

The project, this paper belongs to, is about cache architecture researches for improving the cache efficiency. This project mainly focuses on the memory wall problem. In this area, the authors have proposed Data-Object Oriented Cache (DOOC) technique, which is published in many papers and gets a national patent of China. This paper belongs to the research of highly efficient data layout technique for NUCA, which is an important part of the project.