

cache命中率专项总结

【2010统考真题】某计算机的主存地址空间大小为256MB,按字节编址。指令Cache和数据Cache分离,均有8个Cache行,每个Cache行大小为64B,数据Cache采用直接映射方式。现有两个功能相同的程序A和B,其伪代码如下所示:

<p>程序A:</p> <pre> int a [256][256]; ... int sum_array1() { int i ,j ,sum=0; for (i=0 ; i<256 ; i++) for(j=0 ; j<256 ; j++) sum += a [i] [j]; return sum; } </pre>	<p>程序B:</p> <pre> int a [256][256]; ... Int sum_array2() { int i ,j ,sum=0 for (j=0; j<256; j++) for(i=0 ; j<256 ; j++) sum += a [i] [j]; return sum; } </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

假定int类型数据用32位补码表示,程序编译时,i、j和sum均分配在寄存器中,数组a按行优先方式存放,其首地址为320(十进制数)。请回答下列问题,要求说明理由或给出计算过程。

- 1) 不考虑用于Cache一致性维护和替换算法的控制位,数据 Cache 的总容量为多少?
- 2) 数组元素a[0][31]和a[1][1]各自所在的主存块对应的Cache行号是多少 (Cache行号从0开始)?
- 3) 程序A和B的数据访问命中率各是多少? 哪个程序的执行时间更短?

思路:

① 明确以下信息:

- 1) cache的映射方式、形状和大小,初始cache是否为空
- 2) 数组的存储方式(行优先还是列优先)、数组的访问方式(行优先还是列优先)、每个数组元素的大小是多大,起始地址在哪里,将数组划分成多个块(组)。
- 3) 确定主存地址的划分方式
- 4) 数据元素的访问方式,是直接读出/写回,例如sum+=a[i][j],还是读出后又写回,例如a[i][j]+=5;

② 画出cache,模拟cache访问的过程,计算命中率时可以不求甚解,模拟出总体规律即可。

步骤:

- ① 数据cache有8个cache行,每个行大小为64B,采用直接映射方式,即每个主存块只能映射到特定的cache块中

Cahce0	Cahce1	Cahce2	Cahce3	Cahce4	Cahce5	Cahce6	Cahce7

主存地址空间大小为256MB，由于按字节编址，因此也就是 $2^{28}B$ ，即28位的主存地址，由于采用直接映射方式，且有8个cache行，因此主存地址划分如下：

Tag (19)	行号 (3位)	块内地址 (6位)
----------	---------	-----------

数组a的起始地址为320， $320 \div 64 = 5 \dots 0$ ，因此cache行号是5，块内地址是0。

由于一个cache行的大小是64B，即可以存放 $64B/4B=16$ 个数据元素。

由于数组连续存储的特性，因此 $a[0][0] \sim a[0][15]$ 处于同一块中，接下来每16个元素处于同一个块中。

$Sum += a[i][j]$ 仅有读出，因此每次访问 $a[i][j]$ 只有1次访存。

② 模拟cache访问的过程。

运行程序A时，按照行优先的方式访问数组a[]。

首先访问 $a[0][0]$ ，未命中，因此将 $a[0][0] \sim a[0][15]$ 所在的块给调入cache5中。

Cahce0	Cahce1	Cahce2	Cahce3	Cahce4	Cahce5	Cahce6	Cahce7
					$a[0][0] \sim a[0][15]$		

显然，接下来访问 $a[0][1] \sim a[0][15]$ 时均命中。

接着访问 $a[0][16]$ 时，未命中，因此将 $a[0][16] \sim a[0][31]$ 所在的块给调入cache6中。

Cahce0	Cahce1	Cahce2	Cahce3	Cahce4	Cahce5	Cahce6	Cahce7
					$a[0][0] \sim a[0][15]$	$a[0][16] \sim a[0][31]$	

由于a[0]总共有256个元素， $256/16=16$ 组，接下来16组每一组都会发送同样的情况，因此命中率为 $15/16$ 。

运行程序B时，按照列优先的方式访问数组a[]。

首先访问 $a[0][0]$ ，未命中，因此将 $a[0][0] \sim a[0][15]$ 所在的块给调入cache5中。

Cahce0	Cahce1	Cahce2	Cahce3	Cahce4	Cahce5	Cahce6	Cahce7
					$a[0][0] \sim a[0][15]$		

接下来访问 $a[1][0]$ ，显然， $a[1][0]$ 为命中，且 $a[1][0]$ 是整个数组的第17组， $17 \% 8 = 1$ ，因此其映射的cache块为cache5，即覆盖掉 $a[0][0] \sim a[0][15]$ ，同理，接下来访问 $a[2][0] \dots$ 的访问过程类似，因此命中率为0。

【2020统考真题】假定主存地址为32位，按字节编址，指令Cache和数据Cache与主存之间均采用8路组相联映射方式，直写(Write Through)写策略和LRU替换算法，主存块大小为64B，数据区容量各为32KB。开始时Cache均为空。请回答下列问题。

1) Cache 每一行中标记(Tag)、 LRU位各占几位？是否有修改位？

2) 有如下C 语言程序段：

```
for (k= 0; k<1024; k++)
    s[k] = 2 * s [k];
```

若数组s及其变量k均为int型，int型数据占4B，变量k分配在寄存器中，数组s在主存中的起始地址为0080 00C0H，则在该程序段执行过程中，访问数组s的数据Cache缺失次数为多少？

3) 若CPU最先开始的访问操作是读取主存单元0001 0003H中的指令，简要说明从Cache中访问该指令的过程，包括Cache缺失处理过程。

步骤：

① 数据cache采用8路组相联映射方式， $32\text{KB}/64\text{B}=512$ ，总共有512个cache行， $512/8=64$ ，总共有64组，因此cache结构如下：

	路0	路1	路2	路3	路4	路5	路6	路7
组0								
组1								
组2								
组3								
组4								

组59								
组60								
组61								
组62								
组63								

由于主存地址为32位，按字节编址，且采用组相联映射方式，因此主存地址划分如下：

Tag (20)	组号 (6位)	块内地址 (6位)
----------	---------	-----------

$s[k] = 2 * s[k]$ 涉及将 $s[k]$ 的内容取出并写回，总共2次访存操作。

数据 $s[]$ 为一维数组，总共有1024个元素，起始地址为0080 00C0H, 0C0h=0000 1100 0000B。

显然，块内地址为0，组号为3。

由于一个块大小为64B， $64B/4B=16$ ，即一个块有16个元素。因此 $s[0] \sim s[15]$ 都处于同一个块中。

$1024/16=64$ ，总共是64组。

由于是一维数组，因此没有行列优先之说。

② 模拟cache访问的过程。

访问 $s[0]$ ，首先是读操作，cache未命中，因此将 $s[0] \sim s[15]$ 读入cache，接着写回命中。

	路0	路1	路2	路3	路4	路5	路6	路7
组0								
组1								
组2								
组3	$s[0] \sim s[15]$							
组4								

组59								
组60								
组61								
组62								
组63								

接着在访问 $s[1] \sim s[15]$ 的过程中，共有30次访存，全部命中。

访问 $s[16]$ ，首先是读操作，cache未命中，因此将 $s[16] \sim s[31]$ 读入cache，接着写回命中。

	路0	路1	路2	路3	路4	路5	路6	路7
组0								
组1								
组2								
组3	$s[0] \sim s[15]$							
组4	$s[16] \sim s[31]$							

接着在访问 $s[17] \sim s[31]$ 的过程中，共有30次访存，全部命中。

接下来的过程类似，可以基本判断，命中率为 $31/32$ ，缺失次数为64次。

全相联映射方式下由于数组元素可以映射到任何一个位置，相对简单，一般常见的考察方式是组相联映射，这里情况更多变一点。

【2023统考真题】已知计算机M的字长为32位，按字节编址，采用请求调页策略的虚拟存储管理方式，虚拟地址为32位，页大小为4KB；数据Cache采用4路组相联映射方式，数据区大小为8KB，主存块大小为32B。现有C语言程序段如下：

```
int a[24][64];
...
for(i=0; i<24; i++)
    for(j=0; j<64; j++) a[i][j]=10;
```

已知二维数组a按行优先存放，在虚拟地址空间中分配的起始地址为0042 2000H，sizeof(int)=4，假定在M上执行上述程序段之前数组a不在主存，且在该程序段执行过程中不会发生页面置换。请回答下列问题：

- 1) 数组a分布在几个页面中？对于数组a的访问，会发生几次缺页异常？页故障地址各是什么？
- 2) 不考虑变量 i 和 j，该程序段的数据访问是否具有时间局部性？为什么？
- 3) 计算机M的虚拟地址（A31~A0）中哪几位用作块内地址？哪几位用作Cache组号？a[1][0]的虚拟地址是多少？其所在主存块对应的Cache组号是多少？
- 4) 数组a占用多少主存块？假设上述程序段执行过程中数组a的访问不会和其他数据发生Cache访问冲突，则数组a的Cache命中率是多少？若将循环中i和j的次序按如下方式调换：

```
for(j=0; j<64; j++)
    for(i=0; i<24; i++) a[i][j]=10;
```

则数组a的Cache命中率又是多少？

步骤：

① 虚拟地址的低位部分（页内地址）和物理地址的低位部分是先相同的。

由于页大小是4KB，按字节编址，因此页内地址为12位。

数据区大小为8KB，主存块大小为32B，因此cache块数为8KB/32B=256块。

由于采用4路组相联，因此256/4=64组，因此cache结构如下：

	路0	路1	路2	路3
组0				
组1				
组2				
组3				
.....				
组60				
组61				
组62				
组63				

主存地址划分如下：

Tag	组号 (6位)	块内地址 (5位)
-----	---------	-----------

由于数组a的起始地址为0042 2000H, 000H=0000 0000 0000B, 因此a[0][0]所在块会映射到组0中。32B/4B=8, 因此a[0][0]~a[0][7]一组。

数组a[24][64]共有24行, 每行有64个元素, 因此一行有8组, 共有24*8组。

a[i][j]=10只涉及到一次写操作, 即1次访存。

② 模拟cache访问的过程。

行优先方式下访问过程如下:

	路0	路1	路2	路3
组0	a[0][0]~a[0][7]			
组1	a[0][8]~a[0][15]			
组2				
组3				
组4				
.....				
组59				
组60				
组61				
组62				
组63				

最开始访问a[0][0], 映射到组0的任意一路中, 不妨假设是路0中, 此时未命中, 因此将a[0][0]~a[0][7]读入到cache中。随后访问a[0][1]~a[0][7]命中。

接着访问a[0][8], 过程类似。后续访问类似, 因此命中率为7/8。

列优先方式下访问过程如下:

	路0	路1	路2	路3
组0	a[0][0]~a[0][7]			
组1				
组2				
组3				
组4				
.....				
组59				
组60				
组61				
组62				
组63				

最开始访问 $a[0][0]$ ，映射到组0的任意一路中，不妨假设是路0中，此时未命中，因此将 $a[0][0] \sim a[0][7]$ 读入到cache中。

接着访问 $a[1][0]$ ，显然，未命中， $a[1][0]$ 为第9组，对应组号为组8，访问过程类似。

	路0	路1	路2	路3
组0	$a[0][0] \sim a[0][7]$			
组1				
组2				
组3				
组4				
.....				
	路0	路1	路2	路3
组8	$a[1][0] \sim a[1][7]$			
组9				
组10				
组11				
组12				

同理，接下来访问 $a[2][0], a[3][0], \dots$ ，其映射组号为间隔为8的等差数列。

$64 \text{组} / 8 = 8$ ，显然 $a[0][0]$ 、 $a[1][0]$ 、 $a[2][0], \dots, a[7][0]$ 将占满第一路。

	路0	路1	路2	路3
组0	$a[0][0] \sim a[0][7]$			
组1				
组2				
组3				
组4				
.....				
	路0	路1	路2	路3
组8	$a[1][0] \sim a[1][7]$			
.....				
	路0	路1	路2	路3
组16	$a[2][0] \sim a[2][7]$			
.....				
	路0	路1	路2	路3
组56	$a[7][0] \sim a[7][7]$			

接下来访问 $a[8][0]$ 时， $a[8][0]$ 前有64组，因此其组号又从0开始了。

	路0	路1	路2	路3
组0	$a[0][0] \sim a[0][7]$	$a[8][0] \sim a[8][7]$		
组1				
组2				
组3				
.....				
	路0	路1	路2	路3
组8	$a[1][0] \sim a[1][7]$	$a[9][0] \sim a[9][7]$		

	路0	路1	路2	路3
组16	$a[2][0] \sim a[2][7]$	$a[10][0] \sim a[10][7]$		
.....				
组56	$a[7][0] \sim a[7][7]$	$a[15][0] \sim a[15][7]$		

$a[8][0]$ 、 $a[9][0]$... $a[15][0]$ 访问过程类似。

$a[16][0]$ 、 $a[17][0]$... $a[23][0]$ 访问过程类似。

组0	$a[0][0] \sim a[0][7]$	$a[8][0] \sim a[0][7]$	$a[16][0] \sim a[16][7]$	
.....				
组8	$a[1][0] \sim a[1][7]$	$a[9][0] \sim a[9][7]$	$a[17][0] \sim a[17][7]$	
.....				
组16	$a[2][0] \sim a[2][7]$	$a[10][0] \sim a[10][7]$	$a[18][0] \sim a[18][7]$	
.....				
组56	$a[7][0] \sim a[7][7]$	$a[15][0] \sim a[15][7]$	$a[23][0] \sim a[23][7]$	

分析整个访问流程：

列优先访问时

$a[0][0]$	$a[0][1]$	$\sim a[0][7]$
$a[0][0]$	$a[1][0]$	$\sim a[1][7]$
\vdots	\vdots	
$a[23][0]$	$a[23][1]$	$\sim a[23][7]$
$a[0][8]$	$a[0][9]$	$\sim a[0][15]$
\vdots	\vdots	
$a[23][8]$	$a[23][9]$	$\sim a[23][15]$

不难发现， $a[][0]$ 列的元素均未命中，随后的 $a[][1] \sim a[][7]$ 均命中，同理，接下来访问其他列的过程类似，因此命中率为7/8。

【2014统考真题】某程序中有循环代码段P: “for(int i=0; i<N; i++)sum+=A[i];”。假设编译时变量sum和i分别分配在寄存器R1和R2中。常量N在寄存器R6中，数组A的首地址在寄存器R3中。程序段P的起始地址为0804 8100H，对应的汇编代码和机器代码如下表所示。

编 号	地 址	机器代码	汇编代码	注 释
1	08048100H	00022080H	loop: sll R4, R2, 2	(R2) << 2 → R4
2	08048104H	00083020H	add R4, R4, R3	(R4) + (R3) → R4
3	08048108H	8C850000H	load R5, 0(R4)	((R4) + 0) → R5
4	0804810CH	00250820H	add R1, R1, R5	(R1) + (R5) → R1
5	08048110H	20420001H	add R2, R2, 1	(R2) + 1 → R2
6	08048114H	1446FFFAH	bne R2, R6, loop	if(R2) ≠ (R6) goto loop

【2014统考真题】假设对于上题中的计算机M和程序P的机器代码，M采用页式虚拟存储管理；P开始执行时，(R1)=(R2)=0，(R6)=1000，其机器代码已调入主存但不在Cache中；数组A未调入主存，且所有数组元素在同一页，并存储在磁盘的同一个扇区。请回答下列问题并说明理由。

- 1) P执行结束时，R2的内容是多少？
- 2) M的指令Cache和数据Cache分离。若指令Cache共有16行，Cache和主存交换的块大小为32B，则其数据区的容量是多少？若仅考虑程序段P的执行，则指令Cache的命中率为多少？
- 3) P在执行过程中，哪条指令的执行可能发生溢出异常？哪条指令的执行可能产生缺页异常对于数组A的访问，需要读磁盘和TLB至少各多少次？

注意：这里问的是指令cache的命中率。

分析：循环代码段的整个长度是24B，而一个cache块大小是32B，因此，程序段p执行过程中，除了第一条指令未命中，后续指令的执行均命中。