第五章 存储系统

第六讲 减少Cache不命中开销

谢长生

武汉光电国家研究中心

华中科技大学计算机科学与技术学院

5.6.1 采用两级Cache

1. 应把Cache做得更快？还是更大？

答案：二者兼顾，再增加一级Cache

- 第一级Cache(L1)小而快
- 第二级Cache(L2)容量大

2. 性能分析

平均访存时间 ＝ 命中时间$_{L1}$＋不命中率$_{L1}$×不命中开销$_{L1}$

不命中开销$_{L1}$ ＝ 命中时间$_{L2}$＋不命中率$_{L2}$×不命中开销$_{L2}$

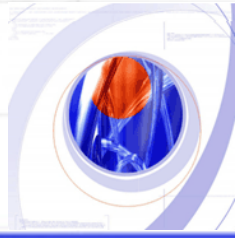
平均访存时间 ＝ 命中时间$_{L1}$＋不命中率$_{L1}$×

（命中时间$_{L2}$＋不命中率$_{L2}$×不命中开销$_{L2}$）

3. 局部不命中率与全局不命中率

- 局部不命中率＝该级Cache的不命中次数/到达该级Cache的访问次数
- 全局不命中率＝该级Cache的不命中次数/CPU发出的访存的总次数
- 全局不命中率$_{L2}$＝不命中率$_{L1}$×不命中率$_{L2}$

评价第二级Cache时，应使用全局不命中率这个指标。它指出了在CPU发出的访存中，究竟有多大比例是穿过各级Cache，最终到达存储器的。



5.6 减少Cache不命中开销

1. 对于第二级Cache，我们有以下结论：
   - 在第二级Cache比第一级 Cache大得多的情况下，两级Cache的全局不命中率和容量与第二级Cache相同的单级Cache的不命中率非常接近。
2. 第二级Cache不会影响CPU的时钟频率，因此其设计有更大的考虑空间。
   - 两个问题需要权衡：
     - 它能否降低CPI中的平均访存时间部分？
     - 它的成本是多少？

3. 第二级Cache的参数

- 容量

  第二级Cache的容量一般比第一级的大许多。

  大容量意味着第二级Cache可能实际上没有容量不命中，只剩下一些强制性不命中和冲突不命中。

- 相联度

  第二级Cache可采用较高的相联度或伪相联方法。

- 块大小
  - 第二级Cache可采用较大的块

    如 64、128、256字节
  - 为减少平均访存时间，可以让容量较小的第一级Cache采用较小的块，而让容量较大的第二级Cache采用较大的块。
  - 多级包容性

    需要考虑的另一个问题：

    第一级Cache中的数据是否总是同时存在于第二级Cache中。

5.6.2 让读不命中优先于写

1. Cache中的写缓冲器导致对存储器访问的复杂化

在读不命中时，所读单元的最新值有可能还在写缓冲器中，尚未写入主存。

2. 解决问题的方法(读不命中的处理)

- **推迟对读不命中的处理，直到写缓冲器清空。**

（缺点：读不命中的开销增加）

- **检查写缓冲器中的内容(让读不命中优先于写)**

3. 在写回法Cache中，也可采用写缓冲器。

5.6.3 写缓冲合并

1. 提高写缓冲器的效率
2. 写直达Cache

依靠写缓冲来减少对下一级存储器写操作的时间。

- 如果写缓冲器为空，就把数据和相应地址写入该缓冲器。

  从CPU的角度来看，该写操作就算是完成了。

  - 如果写缓冲器中已经有了待写入的数据，就要把这次的写入地址与写缓冲器中已有的所有地址进行比较，看是否有匹配的项。如果有地址匹配而对应的位置又空闲，就把这次要写入的数据与该项合并。这就叫写缓冲合并。
  - 如果写缓冲器满且又没有能进行写合并的项，就必须等待。

5.6 减少Cache不命中开销

提高了写缓冲器的空间利用率，而且还能减少因写缓冲器满而要进行的等待时间。

写地址	V		V		V		V	
100	1	Mem[100]	0		0		0	
108	1	Mem[108]	0		0		0	
116	1	Mem[116]	0		0		0	
124	1	Mem[124]	0		0		0	

（a）不采用写合并

写地址	V		V		V		V	
100	1	Mem[100]	1	Mem[108]	1	Mem[116]	1	Mem[124]
	0		0		0		0	
	0		0		0		0	
	0		0		0		0	

（b）采用了写合并

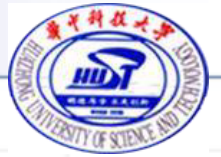
5.6.4 请求字处理技术

1. 请求字

从下一级存储器调入Cache的块中，只有一个字是立即需要的。这个字称为请求字。

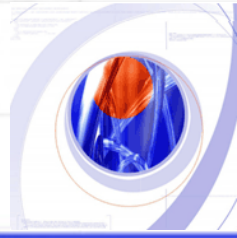
2. 应尽早把请求字发送给CPU

- 尽早重启动：调块时，从块的起始位置开始读起。一旦请求字到达，就立即发送给CPU，让CPU继续执行。
- 请求字优先：调块时，从请求字所在的位置读起。这样，第一个读出的字便是请求字。将之立即发送给CPU。



3. 这种技术在以下情况下效果不大：

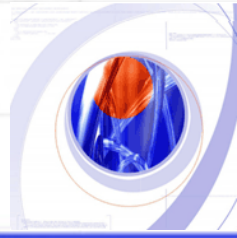
- Cache块较小
- 下一条指令正好访问同一Cache块的另一部分



5.6.5 非阻塞Cache技术

1. 非阻塞Cache：Cache不命中时仍允许CPU进行其它的命中访问。即允许“不命中下命中”。
2. 进一步提高性能：
   - “多重不命中下命中”
   - “不命中下不命中”

   （存储器必须能够处理多个不命中）
3. 可以同时处理的不命中次数越多，所能带来的性能上的提高就越大。



谢谢大家！