ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.7, No.22, August 2011.
E-mail: jslt@cccc.net.cn
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

高档计算机系统中 Cache 性能分析

潘继强

(陕西理工学院 计算机科学与技术系,陕西 汉中 723000)

摘要:现代计算机体系结构中广泛采用 Cache 来缓解处理器运行和存储器访问的速度增长之间的巨大差距,使得 Cache 已经成为影响处理器性能、功耗、价值的重要因素之一。文章根据 Cache 基本工作原理和引入 Cache 后计算机系统的性能分析,介绍了一些改进 Cache 性能的方法。

关键词:高速缓存;命中率;局部性原理;静态随机存储器

中图分类号:TP302 文献标识码:A 文章编号:1009-3044(2011)22-5285-02

随着计算机技术的不断发展和进步,CPU(中央处理器)主频的不断提高对计算机系统性能的提升起到了极大的作用,但是作为一个完整的计算机系统,计算机系统性能的提高并不是仅取决于 CPU 的主频,还与其他因素密切相关,例如,计算机系统结构、数据在各部件间的传送速度、存储部件的存取速度等,其中 CPU 和主存之间的存取速度与计算机系统性能的提高有很大关系。可是主存速度的提高始终跟不上 CPU 的发展,据统计,CPU 的速度平均每年改进 60%,而组成主存的动态 RAM(随机存储器)速度平均每年只改进 7%,结果是 CPU 和主存之间的速度间隙平均每年大增 50%。处理器运行和存储器访问的速度增长之间存在的差距越来越大,这种现象已经成为影响计算机系统性能最主要的瓶颈之一。假设一台计算机的 CPU 工作速度很快,而配备的主存访问速度相对较慢,这样就会造成 CPU 在访存时等待,降低了处理器的工作速度,进而影响计算机的整体性能。

解决 CPU 与主存的速度差距问题在于保持 CPU 的能力,提高主存的速度。使用硬件技术提高存储芯片的存取速度是一个有效的手段,可是在慢速的主存和快速 CPU 之间插入一个容量较小的高速存储器起缓冲作用(即 Cache 技术)也是解决问题的一个行之有效的方法,使得速度和成本之间的矛盾得到较合理的解决。自从 1985 年 Intel80386 问世以来,在后续的微处理器中都采用了 Cache。

1 Cache 的工作原理

在现代计算机系统中,高速缓存 Cache 已经逐渐成为计算机不可缺少的一部分。在计算机系统中设置 Cache 是为了解决高速处理器和低速主存之间速度不匹配的问题,从而可以提高整机的性能。因此要分析 Cache 对计算机性能的影响,必须要了解其工作原理。

Cache 的工作原理是基于程序和数据访问的局部性。任何程序或数据要为 CPU 所使用,必须先放到主存中。CPU 只与主存交换数据,所以主存的速度在很大程度上决定了系统的运行速度。对大量典型程序运行情况的分析结果表明,程序运行期间,在一个较短的时间间隔内,由程序产生的内存访问地址往往集中在主存很小范围的地址空间内。这一点不难理解。指令地址本来就是连续分布的,再加上循环程序段和子程序段要多次重复执行。因此,对这些地址中的内容的访问就自然具有时间上集中分布的倾向。数据分布的这种集中倾向不如指令明显,但对数组的存储和访问以及内存变量的安排都使存储器地址相对集中。这种在单位时间内对局部范围的存储器地址频繁访问,而对此范围以外的地址则访问甚少的现象,就称为程序访问的局部性。

由此可以想到,如果把在一段时间内、一定地址范围内被频繁访问的信息集合成批地从主存中读到一个能高速存取的小容量存储器中存放起来,供程序在这段时间内随时使用,从而减少或不再去访问速度较慢的主存,就可以加快程序的运行速度。这就是 Cache 的设计思想,在主存和 CPU 之间设置一个小容量的高速存储器就是高速缓存 Cache(其结构如图 1 所示),通常由 SRAM(静态随机存储器)构成。它的存取速度比构成主存的动态 RAM 要快的多,故 CPU 在需要访问主存中的程序和数据时,就不必多次直接访问速度较慢的主存,而是从高速缓存中以更快的速度得到必要的程序和数据,从而可以缓解 CPU 和主存速度不匹配的矛盾,进而提高计算机系统的性能。但是 SRAM 的价格比较昂贵,若干设置大容量的高速缓存,就会增加计算机的成本,故 Cache 的容量一般相对主存来说都比较小。

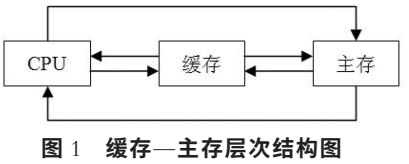

图 1 缓存—主存层次结构图

在 Cache 系统中,将主存和 Cache 都分成若干同样大小的块,每块内又包含若干个字,主存总是以块为单位映射到 Cache 中。根据程序访问的局部性原理,在程序运行期间系统不断地将与当前访问块相关联的后继存储单元块从主存读入到 Cache,其实质就是把主存中的程序和数据分块复制到 Cache 中,然后再和 CPU 进行高速传送。当 CPU 需要访问主存时就会在地址总线上发出访存地址,首先通过主存—Cache 地址变换机构判定访存地址所对应的存储单元块是否已在 Cache 中。如果在 Cache 中(称为 Cache 命中),则经地址变换机构将主存地址变换成 Cache 地址去访问 Cache。如果不在 Cache 中(称为 Cache 不命中),此时就要把要访问的字直接从主存送往 CPU,同时把包括该字的主存块调入 Cache 供 CPU 使用。当然 Cache 的容量是有限的,如果此刻 Cache 处于未被装满的状态,则可将新的主存块直接调入 Cache。倘若 Cache 原来已被装满,即已无法将新的主存块调入 Cache 时,就得采用替换策略,从 Cache 中换出一个旧块,并将新块替换进 Cache。

收稿日期:2011-05-28
作者简介:潘继强(1978-),男,陕西汉中人,讲师,研究方向为计算机基础理论教学。

2 Cache 的性能分析

在计算机系统中设置 Cache 的目的是为了解决 CPU 和主存之间速度不匹配的矛盾,从而提高计算机的性能。尽管引入 Cache 后,使用 SRAM 技术的 Cache 访问速度与 CPU 的速度相当,可以使系统的整体性能得到提高,但由于 SRAM 的制作工艺比较复杂、制作成本比较较高,从计算机系统的性价比方面来考虑,不可能将所有主存都换成 SRAM,也不可能设置大容量的 Cache。例如,80386 的主存最大容量为 4GB,与其配套的 Cache 容量为 16KB 或者 32KB。从 CPU 性能方面考虑,增加 Cache 系统的目的就是使 Cache—主存系统的平均访问时间接近 Cache 访问时间。在 Cache—主存系统中,平均访问时间 t 可定义如下:

$$t=h\cdot t_c+(1-h)\cdot t_m$$

其中 h 表示命中率(数值上等于命中次数比上命中次数与未命中次数之和。当 Cache 不命中时,CPU 就需要访问主存,故未命中次数就等于访问主存的次数),t_c 表示命中 Cache 时的访问时间,t_m 表示未命中 Cache 时访问主存的时间。

假设 CPU 在执行某段程序时,共访问 Cache 命中 2000 次,访问主存 50 次。已知 Cache 的存取周期为 50ns,主存的存取周期为 200ns。则 Cache 的命中率为:

$$h=2000/(2000+50)=0.97$$

于是可计算出 Cache—主存系统的平均访问时间为:

$$t=50ns\times0.97+2000ns\times(1-0.97)=54.5ns$$

通过以上分析可以发现,在使用 Cache 的计算机系统中,Cache—主存系统的平均访问时间非常接近于 Cache 访问的时间,其数值远小于主存的存取周期,这说明增加 Cache 之后,计算机系统的性能有了显著的提高。同时也说明命中率 h 是影响 Cache—主存系统的平均访问时间的关键,因此提高命中率对改善 Cache—主存系统的效率起着关键性的作用。

3 Cache 的性能改进

通过对 Cache 性能分析,可知 Cache 命中率是影响 Cache—主存系统的平均访问时间的关键。Cache 的命中率越高,发生不命中可能性就越小,从 Cache 获取指令和数据的可能性就越大,平均访问时间也就越短,使处理器保持高效率运作,从而可以提高计算机整机性能。通过对 Cache 的基本原理的分析,Cache 不命中的原因有以下几个方面:

1)当程序首次执行时,由于对应的主存块是第一次访问,故该块必然不在 Cache 中,所以必须首先将主存块复制到 Cache 中。而且当 CPU 顺序访问下一个主存块时依然会发生不命中,这个过程一直会持续到 Cache 装满或程序全部调入 Cache 为止。此时发生不命中的次数和分块的大小有直接关系,块长越大,发生不命中的次数自然就越小。

2)若 Cache 的容量太小,受此条件的限制,不可能在执行过程中将所需的指令和数据全部调入 Cache,于是程序执行期间可能会发生频繁的替换。每当发生替换时,CPU 就要访问慢速的主存,而且发生替换的频率越高,越不利于 CPU 性能的发挥。

3)采用替换策略时,由于主存的容量远大于 Cache 的容量,故需替换入的主存块的个数大于 Cache 可容纳的块数,在替换时极有可能会把下次访问的指令或数据替换出去,而且当块长越大,替换时传送的数据量也就越大,造成下次访问的不命中。由于块长的增大,导致缓存中块数的减少,而新装入的块要覆盖旧块,很可能出现少数块刚刚装入就被覆盖,因此命中率反而下降。

影响 Cache 性能的因素比较多,其中 Cache 的容量与块长是影响 Cache 效率的重要因素。目前块长的最优值很难确定,一般每块取 4 至 8 个字或字节较好。而在其他因素方面还有以下方法可以改进 Cache 的性能。

1)增大 Cache 容量,降低不命中率。Cache 的命中率和容量有极大的关系,一般来说,Cache 的存储容量比主存容量小的多,大不能太小,太小会使命中率太低。但是容量也没有必要太大,太大不仅会增加成本,而且当 Cache 容量达到一定值后,命中率随容量的增家将不会有显著的提高,其关系如图 2 所示。因此,Cache 的空间与主存空间在一定范围内应保持适当比例的映射关系,以保证 Cache 有较高的命中率,并且系统成本不过大地增加。一般情况下,可以使 Cache 与主存的空间比为 1:128。

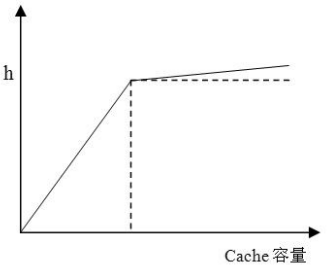

图 2　Cache 命中率与容量的关系图

2)通过 Cache 的结构设计,减少不命中次数。根据主存采用指令、数据分开存储的方案,在设计高速缓存的结构时,则相应的 Cache 采用分立缓存。分立缓存是指指令和数据分别存放在两个缓存中,一个称为指令 Cache,一个称为数据 Cache。有时可以根据 Cache 位置的不同分为片内缓存和片外缓存,片内缓存为第一级,片内缓存为第二级。

3)通过预取技术,提高 Cache 的命中率。根据程序访问的局部性原理,通过缓存控制指令和预取技术,预测将要访问的指令,在当前指令执行过程尚未结束时就提前将下一条准备执行的指令取到 Cache 中,这样 CPU 在取指时只需要访问 Cache 就可以了,从而可以提高 Cache 的命中率,使处理器保持高效率运作。

4)采用适当的映射方式和替换策略来提高命中率。由于 Cache 的容量不可能任意增大,而且当 Cache 容量超过一定值后,命中率随容量的增加将不会明显的提高,所以访存时发生不命中是不可避免的。当访存不命中发生后,可通过采用适当的映射方式和最佳的替换策略,来提高下次访问的命中率,也可改进 Cache 的性能。

在计算机系统中,Cache 技术有效地解决了 CPU 和主存速度不匹配矛盾。目前的微型计算机系统中一般均采用这种方法来提高存储系统的性能,使系统在成本增加不高的情况下,性能有较显著的提升。

参考文献:

[1] 冯博琴.微型计算机硬件技术基础[M].北京:高等教育出版社,2003.
[2] 唐朔飞.计算机组成原理[M].2 版.北京:高等教育出版社,2008.
[3] 白中英.计算机组成原理[M].3 版.北京:科学出版社,2001.
[4] 郑伟明,汤志忠.计算机系统结构[M].2 版.江苏:南京大学出版社,2004.