计算机体系结构 课程报告

15307110273-张星宇

本次课程作业中，主要完成了对阅读材料《计算机体系结构：量化研究方法（第五版)》中存储器结构层次相关内容的阅读与研究。主要涉及内容为材料附录 B 及第二章的内容。

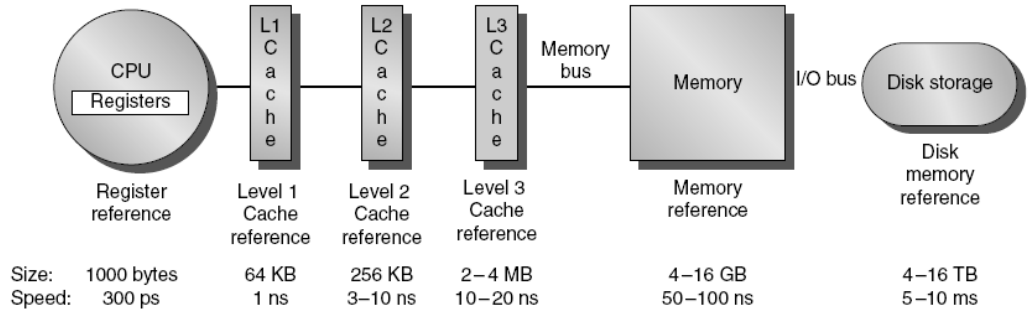
在存储系统的相关研究中，计算机系统结构设计的关键问题之一就是，如何以合理的价格，设计容量和速度都满足计算机系统要求的存储器系统？从这个问题我们可以知道，人们对上述三个指标的主要要求为 :容量大、速度快和价格低。其中，这里的价格指的是每位价格。然而，存储器的固有特性为：速度越快，每位价格就越高；容量大，价格低；而相应地，容量越大，速度就越慢。

针对上述问题，往往使用的解决方法是：采用多种存储器技术，构成多级存储层次结构。实现这种方法的依据为指令和数据访问的局部性原理，即对于绝大多数程序来说，程序所访问的指令和数据在地址上不是均匀分布的，而是相对簇聚的。这里的簇聚包含两个方面的含义：

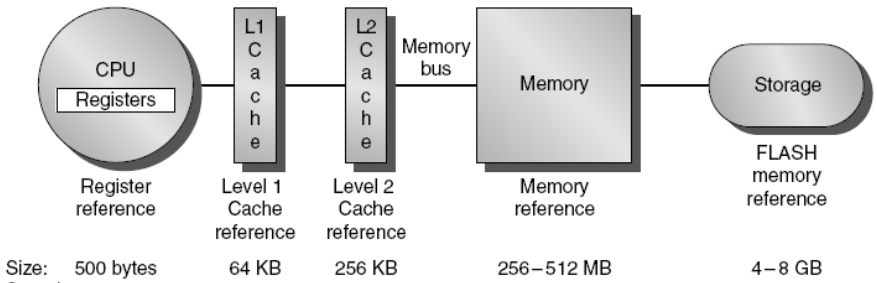
时间局部性：程序马上将要用到的信息很可能就是现在正在使用的信息。

空间局部性 :程序马上将要用到的信息很可能与现在正在使用的信息在存储空间上是相邻的。

结合这两种局部性，我们可以获得以 “Cache－主存” 和 “主存－辅存” 层次为常见的两种层次结构的存储系统，其中 “Cache－主存” 层次用来弥补主存速度的不足；而 “主存－辅存” 层次用来弥补主存容量的不足。材料中给出



(a) Memory hierarchy for server



(b) Memory hierarchy for a personal mobile device

了如下图所示的存储系统示例：

在设计存储器时，在存储层次方面主要需要解决四个问题 :其一，映像规则，即当把一个块调入高一层(靠近 CPU)存储器时，可以放在哪些位置上的问题；其二，查找方法问题，即当所要访问的块在高一层存储器中时，如何找到该块的问题；其三，替换算法啊的选择问题，即当发生不命中时，应替换哪一块？最后，写策略的选择问题，即当进行写访问时，应进行哪些操作的问题。阅读材料中分别对该四个问题的处理方法和研究成果进行了详细介绍。

在映像规则方面，阅读材料给出了 cache 的直接映像、全相联映像和组相联映像三种映像规则。这一块内容和计算机系统基础课程中的基础知识基本重叠，这里不在报告中详细介绍阅读材料中的内容。对于查找方法的介绍也较为简单，现在用到的主要是查找目录表的方法，并存在并行查找和顺序查找两种查找方式。

替换算法所要解决的问题主要为：当新调入一块，而 Cache 又已被占满时，该替换哪一块的问题。直接映像 Cache 中的替换很简单，因为只有一个块，别无选择；而在组相联和全相联 Cache 中，则有多个块供选择，则需要使用到替换算法。目前主要的替换算法有三种：随机法、先进先出法 FIFO 和最近最少使用法 LRU；其中 LRU 和随机法分别因其不命中率低和实现简单而被广泛采用。模拟数据表明，对于容量很大的 Cache，LRU 和随机法的命中率差别不大。

写策略是区分不同 Cache 设计方案的一个重要标志。两种写策略是写直达法：执行“写”操作时，不仅写入 Cache，而且也写入下一级存储器；写回法：执行“写”操作时，只写入 Cache。仅当 Cache 中相应的块被替换时，才写回主存。(设置“修改位”来对其是否已经被修改进行标识)。比较两种写策略，可知写回法的优点是速度快，所使用的存储器带宽较低；而写直达法的优点为易于实现，数据一致性好。

基于上述基本知识分析我们可知，对存储系统结构进行研究的最终目的就是要获得效率更高、性价比更高的存储系统，那么，我们可以获得哪些改进方式，来对 cache 进行优化呢？由以下公式我们可以知道：

平均访存时间 = 命中时间 + 缺失率×缺失代价

针对现有的 cache，我们可以从三个方面改进 Cache 的性能：降低缺失率、减少缺失代价和减少 Cache 命中时间。

基于上述分析，阅读材料中给出了六种 cache 的基本优化方法，具体介绍如下所示：

方法 1：增加 Cache 块大小以降低缺失率

结合实验数据，对该方法进行分析：对于给定的 Cache 容量，当块大小增加时，缺失率开始是下降，后来反而上升了。分析原因，一方面是它减少了强制性缺失；另一方面，由于增加块大小会减少 Cache 中块的数目，所以有可能会增加冲突缺失。 此外，Cache 容量越大，使缺失率达到最低的块大小就越大；而增加块大小会增加缺失代价，故该方法在实现的过程中硬件复杂度较小，而在不同方面的评估中有利有弊。

方法 2：增加 Cache 的容量以降低缺失率

这种方法的主要思想是增加 Cache 的容量以减少容量缺失。而它也存在缺点：增加成本和功耗、可能增加命中时间。从实践的角度来看，这种方法在片外 Cache 中用得比较多。

方法 3：提高相联度以降低缺失率

实验表明，采用相联度超过 8 的方法实际意义不大，故在实践中往往选择相联度在 8 以内的 cache。同时，我们还获得了 2:1 Cache 经验规则：容量为 N 的直接映象 Cache 与容量为 N/2 的两路组相联 Cache 具有大体相同的缺失率。然而，该方法也存在一定的问题，即提高相联度会增加命中时间。

方法 4：采用多级 Cache 以降低缺失代价

把 Cache 做得更快还是更大的问题中，阅读材料中给出的答案是：二者兼顾，故采用再增加一级 Cache 的方法。在这种两级 cache 的问题中，第一级 Cache(L1)小而快，而第二级 Cache(L2)容量大。这样的方法硬件难度较大，但也大大降低了缺失代价，目前使用得较为广泛。在设计的过程中，还需要考虑另一个多级包容性的问题：第一级 Cache 中的数据是否总是同时存在于第 二级 Cache 中。同时，对块大小的经验为：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。

方法 5：使读取缺失的优先级高于写入缺失，以降低缺失代价

Cache 中的写缓冲器导致对存储器访问的复杂化: 写缓冲器进行的写入操作是滞后进行的，所以该缓冲器也被称为后行写数缓冲器。解决问题的方法(读失效的处理)为：推迟对读失效的处理，或检查写缓冲器中的内容。在写回法 Cache 中，如采用写缓冲器，也可以采用类似方法（检查写缓冲器中的内容）。

方法 6：避免在索引 cache 期间进行地址转换，以缩短命中时间

虚拟 Cache 指的是访问 Cache 的索引以及 Cache 中的标识都是虚拟地址。然而，在实际设计中，我们并非都采用虚拟 Cache，其重要原因有两个：其一，要对内容提供保护；其二：每次切换进程时，虚拟地址会指向不同的物理地址，需要对 Cache 进行刷新。对上述为题的解决方法为，在 Cache 地址标识中增加 PID 字段(进程标识符)。实验表明，该方法在失效率方面有较好的表现。

本次课程作业，对阅读材料《计算机体系结构：量化研究方法（第五版）》中存储器结构层次相关内容进行了学习。后续学习中还将不断查阅存储器结构层次相关资料，不断深化自己对该方面学习的理解，从而提高自己对计算机体系结构的理解。