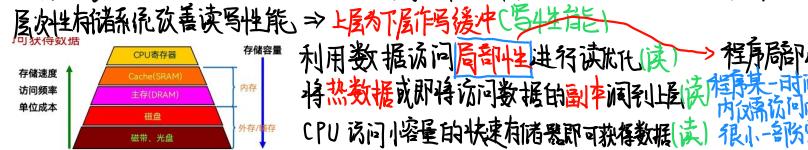


cache: 现代处理器标配, 由SRAM构成, 部分计算机配置有L1, L2, L3多级缓存



层次性有利于改善读写性能 \Rightarrow 上层为下层作缓存 (写回能)

利用数据访问局部性进行读优化 (读) \Rightarrow 程序局部性

将热数据或即将访问数据的副本调到上层 (时间间隔)

提高命中率, 提高访问效率, 减少访存冲突

命中率 hit: CPU访问数据在cache中

miss: CPU访问数据不在cache中

写写入: 同时写至cache \Rightarrow 多线程数据无数据风险, 写速度慢

写回: 只写cache \Rightarrow 有脏数据, 有丢失数据风险, 实现写速度快

块块块: cache与主存块级映射实现预读

行/槽 line/slot: 行选, 行地址, 数据块索引

有效位, 查找标记, 标记位, 置换标志, 数据快副本

命中率 hit rate: 主存中 cache命中比例, 缺失率 miss rate: 1-命中率

命中时间 hit time: 数据查找时间, cache访问时间, 总访问时间

缺失损失 miss penalty: 主存访问cache, 数据块传输到处理器的时间, 远大于命中时间, 所以一些相对较小的访问可能慢

cache关键技术

1. 数据查找 \Rightarrow 主存地址对应 cache地址

* 主存块地址对应 cache块地址

软件方法: 如何用数据结构, 如何快速查找

硬件方法: 如何硬件存储, 如何快速定位数据块

② 相联存储器: 按内容进行访问的存储器(CAM) \Rightarrow 提高命中率

用关键字检索存储器内部关键字, 对包含关键字的有效单元进行读写,

以内容作为地址访问的存储器称为相联存储器

特点:

按内容进行访问(key, value)

以关键字作全局并发比较 \Rightarrow 速度快, 硬件成本高, 有较多比较器

常用于存放查找表 \Rightarrow CAM在cache中有有效表, 页表

容量 = 寻找表容量 = 表项数 \times 表项大小 = 表项数 \times (valid, key, value)

CPU Cache基本组织开环

由较快的SRAM构成;

Cache与主存分为固定大小的数据块, 以块为单位交换数据;

相联存储器存放找表 \Rightarrow 容量 = Cache块数 \times 表项大小;

表项离: 有效位, 调入Cache的主存块地址, Cache块地址

表项一: 有效位, 主存块地址, cache line中数据, 用途及其它标记

CPU给出块地址与查找表中单元相同且有效位表示命中

块块块地址

主存 11位块地址 + 4位块内偏移

cache 8位块地址 + 4位块内偏移

总容量 20×2^8 块

Cache读过程及读操作

CPU给出主存地址 \Rightarrow 分解为块地址 + 块内地址

块块块地址

块块块地址