Amdahl 定律表明：优化受限于原计算机计算时间中可升级部分所占的比例。

加速比= 整个任务采用该升级时的性能/整个任务未采用该升级时的性能。

产生背景：

内存的带宽高于延迟和程序的空间局域性。运行时的带宽为：409.6GB/s;而DRAM 得带宽仅为25GB/s(他的6%)。

基本公式：

存储器平均访问时间 = 命中时间L1 + 缺失率L1 \* 缺失代价L1

缺失代价L1 = 命中时间L2 + 缺失率L2 \* 缺失代价L2

如此迭代下去，就是多级缓存。

缺失分三种：

强制缺失，容量缺失，冲突缺失。

基本优化方法

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 技术 | 命中时间 | 缺失代价 | 缺失率 | 硬件复杂度 | 备注 |
| 较大的块大小 | // | - | + | 0 | 128字节，64个字节 |
| 较大的缓存大小 | - | // | + | 1 | 应用广泛 |
| 较高的相联度 | -  (可以写的位置多，这就存在寻址上的压力) | // | + | 1 | 应用广泛 |
| 多级缓存 | // | + | // | 2 | 昂贵的硬件 |
| 读取操作优先级高于写入操作 | // | + | // | 1 | 应用广泛 |
| 避免在缓存索引期间进行地址转换 | + | // | // | 1 | 应用广泛 |

// 表示没有影响

* 表示有损害

+ 表示有帮助。

高级方法

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 方法 | 命中时间 | 带宽 | 缺失代价 | 缺失率 | 功耗 | 硬件成本  复杂度 | 注释 |
| 小而简单的缓存 | + | // | // | - | + | 0 | 很普通  应用广泛 |
| 路预测缓存 | + | // | // | // | +  省电 | 1 | 在P4中使用 |
| 流水化缓存访问 | - | + | // | // | // | 1 | 应用广泛 |
| 非阻塞缓存 | // | + | + | // | // | 3 | 应用广泛 |
| 分组缓存 | // | + | // | // | + | 1 | I7和A8使用 |
| 关键字优先和提前重启动 | // | // | +  (缺失的到来后，马上启动处理器，在大块中优势明显) | // | // | 2 | 应用广泛 |
| 合并写缓冲区 | // | // | +  (针对写降低缺失代价) | // | // | 1 | 和直方法一起广泛应用 |
| 以编译器技术减少缓存缺失 | // | // | // | +  (程序员的工作) | // | 0 | 编译器的一个挑战 |
| 指令和数据的硬件预取 | // | // | +  (放到下一级的缓存中降低缺失代价) | + | - | 2(指令)  3(数据) | 大多有预取指令，高端是自动预取 |
| 编译器控制的预取 | // | // | +  (放到下一级的缓存中降低缺失代价) | + | // | 3 | 也可以认为是上面技术的软件实现  需要非阻塞缓存，存在指令开销 |

内存分类：

SRAM: 主要用于缓存，成本高，速度快。

DRAM：主要用于内存，成本适中，速度适中。优化方法：增加定时（列，行行的访问），增加时钟，增加带宽，双倍数据率。

GDRAM：主要用于显存，带宽要求高于DRAM，这种内存局域性较低，没有缓存。

EPRROM：主要用于U盘，成本低，速度低。

错误纠正方法：奇偶校验，ECC校验（和纠正）。