

《 计算机体系结构 》

实验报告

|  |  |
| --- | --- |
| 班 级： | **计211** |
| 学 号： | **21013017** |
| 姓 名： | **蒋文强** |
| 指导教师： | **梁建宁** |

信息科学与工程学院

2023年 11 月

**实验名称**Cache性能分析**实验地点** **实验日期** 11.22

1. **实验目的**

1、加深对Cache基本概念、基本组织结构以及工作原理的理解。

2、掌握Cache容量、相关度、块大小对Cache性能的影响

3、掌握降低Cache不命中率的各种方法以及它们对提高Cache性能的好处

4、理解LRU与随机法的基本思想以及对Cache性能的影响。

1. **实验设备**

实验平台采用Cache模拟器，使用个人电脑。

1. **实验原理**

**1、Cache容量对不命中率的影响**

（1）启动MyCache

（2）单击“复位”按钮，把各参数设置为默认值。

（3）选择地址流文件all.din。方法：选择“访问地址”→“地址流文件”选项，然后单击“浏览”按钮，从本模拟器所在的文件夹下的“地址流”文件夹中选取。

（4）选择不同的Cache容量，分别执行模拟器（单击“执行到底”），在表1.1中记录各种情况下的不命中率。

表1.1 不同容量下Cache的不命中率

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 容量KB | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
| 不命中率 | 9.87% | 7.19% | 4.48% | 2.56% | 1.42% | 0.89% | 0.60% | 0.49% |

（5）以容量为横坐标，画出不命中率随Cache容量变化的曲线。

1. 根据该模拟结果，你能得出什么结论？

增加缓存容量对缓存不命中率的影响呈现递减的趋势。一开始，增加容量会显著减少不命中率，但随着容量的增加，降低不命中率的效果逐渐减弱，最终趋于平稳。

一个典型的性能递减（diminishing returns）现象，即在一定条件下，进一步投入资源不再产生相同比例的增益。在缓存设计和优化中，需要权衡缓存容量、成本和性能之间的关系，以找到适当的平衡点。

**2、相联度对不命中率的影响**

（1）单击“复位“按钮，把各参数设置为默认值。

（2）选择地址流文件eg.din。

（3）选择不同的Cache相联度，分别执行模拟器，在下表记录各种情况的不命中率。

表1.2 当Cache容量为64KB时，不同相联度下Cache的不命中率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 相联度 | 1 | 2 | 4 | 8 | 16 | 32 |
| 不命中率 | 0.89% | 0.53% | 0.47% | 0.45% | 0.44% | 0.44% |

（4）把Cache容量改为256KB，重复（3）中工作，并填表1.3

表1.3 当Cache容量为256KB时，不同相联度下Cache的不命中率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 相联度 | 1 | 2 | 4 | 8 | 16 | 32 |
| 不命中率 | 0.49% | 0.38% | 0.36% | 0.36% | 0.35% | 0.35% |

（5）以容量为横坐标，画出在容量为64KB和256KB1情况下，不命中率随相联度的曲线。

（6）根据该模拟结果，你能得出什么结论？

根据实验结果以及曲线图可知，在Cache容量一定的情况下，提高相关联度可以降低Cache不命中率，但是相关联度提高到一定程度Cache不命中率几乎不再改变。

**3、Cache块大小对不命中率的影响**

（1）单击复位按钮，把参数设置为默认值。

（2）选择地址流文件all.din。

（3）选择不同的Cache块大小，不同的Cache容量，分别执行模拟器，记录各种情况下的不命中率。

表1.4 各种情况下Cache的不命中率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 块大小（B） | Cache容量（KB） | | | | | |
| 2 | 8 | 32 | 64 | 128 | 512 |
| 16 | 12.02% | 5.79% | 1.86% | 1.26% | 0.95% | 0.71% |
| 32 | 9.87% | 4.48% | 1.42% | 0.89% | 0.60% | 0.42% |
| 64 | 9.36% | 4.03% | 1.20% | 0.71% | 0.43% | 0.27% |
| 128 | 10.49% | 4.60% | 1.08% | 0.64% | 0.35% | 0.20% |
| 256 | 13.45% | 5.35% | 1.19% | 0.67% | 0.34% | 0.16% |

1. 分析Cache块大小对不命中率的影响。

在实验中观察到的现象可以通过空间局部性和冲突不命中的权衡来解释。具体来说，当增大缓存块大小时，会产生双重效应，导致缓存不命中率的变化。

增加空间局部性： 增大缓存块大小有助于提高空间局部性，因为更多的数据被加载到缓存中，利用了数据在空间上的临近性。这减少了强制性不命中，即因为缓存块太小而导致的必须从主存中加载数据的情况。因此，在块大小较小时，这个因素的影响通常会导致不命中率下降。

增加冲突不命中： 然而，随着缓存块大小的增加，每个缓存块可以容纳的数据减少，这可能增加了冲突不命中的可能性。冲突不命中是由于多个内存块映射到相同的缓存组，导致竞争访问的情况。因此，当块大小增加到一定程度时，由于冲突增加，不命中率可能会再次上升。

综合来看，缓存块大小的选择需要在空间局部性和冲突不命中之间找到平衡。过小的块大小可能导致强制性不命中过多，而过大的块大小可能增加了冲突不命中。因此，在进行缓存优化时，需要考虑应用程序的访存模式和缓存架构的特点，以确定最佳的缓存块大小。这种权衡是计算机体系结构和性能优化中的重要考虑因素。

**4、替换算法对Cache不命中率的影响**

（1）单击复位按钮，把参数设置为默认值。

（2）选择地址流文件all.din。

（3）对于不同的替换算法、Cache容量和相联度，分别执行模拟器，记录各种情况下的的不命中率。

表1.5 LRU和随机替换算法的不命中率比较

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cache  容量 | 相联度 | | | | | |
| 2路 | | 4路 | | 8路 | |
| LRU | 随机算法 | LRU | 随机算法 | LRU | 随机算法 |
| 16KB | 1.71% | 2.22% | 1.33% | 2.17% | 1.21% | 3.08% |
| 64KB | 0.53% | 0.68% | 0.47% | 0.70% | 0.45% | 0.83% |
| 256KB | 0.38% | 0.41% | 0.36% | 0.39% | 0.36% | 0.36% |
| 1MB | 0.35% | 0.35% | 0.35% | 0.35% | 0.35% | 0.35% |

（4）分析不同替换算法对Cache不命中率的影响。

随着缓存容量的增加，不同替换策略之间的性能差异可能会减小。在足够大的缓存容量下，缓存的命中率主要受到容量本身的限制，而不再受到替换策略的显著影响。此时，LRU和随机替换策略可能表现相对接近，因为大部分数据都可以被缓存，不论采用何种替换策略，都有足够的空间存储经常访问的数据。

1. **实验中出现的问题和解决方法**

问题：一开始选择地址流文件eg.din，Cache的不命中率变化不明显

解决办法：选择地址流文件all.din,重新记录数据

1. **实验总结**

替换算法的影响： 观察到了LRU和随机替换算法在不同条件下的表现差异。强调了替换算法对缓存性能的重要性。理解不同替换算法的优缺点，以及在不同工作负载下它们的表现如何变化，对于设计和优化缓存系统至关重要。

块大小的影响： 实验结果显示，缓存块大小的选择对缓存性能有显著影响。体现了对空间局部性和冲突不命中之间的平衡的需求。了解何时增加块大小会带来好处，何时可能导致性能下降，对于缓存优化至关重要。

相联度和Cache容量： 观察到了相联度和缓存容量对不命中率的影响。强调了在实际设计中需要平衡相联度和容量的问题。更高的相联度和更大的缓存容量可能会提高性能，但需要考虑成本和复杂性。

全面性的数据集： 通过使用更全面的地址流文件，认识到不同的访问模式对缓存性能的影响。这强调了在实验和性能评估中选择代表性和全面性数据集的重要性。

性能递减： 可能注意到在某些情况下，增加某些参数并不总是导致性能的线性提升。体现了性能递减的概念，即在某一点之后，进一步的增加可能会带来较小的收益。