**CO\_Lab4\_Report**

0710028 陳敬諺

1. Result of cache simulator

Basic problem:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ICACHE | 16 | 32 | 64 | 128 | 256 |
| 4K | 2.17096% | 1.08548% | 0.542741% | 0.27137% | 0.135685% |
| 16K | 2.17096% | 1.08548% | 0.542741% | 0.27137% | 0.135685% |
| 64K | 2.17096% | 1.08548% | 0.542741% | 0.27137% | 0.135685% |
| 256K | 2.17096% | 1.08548% | 0.542741% | 0.27137% | 0.135685% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DCACHE | 16 | 32 | 64 | 128 | 256 |
| 4K | 5.55556% | 3.1746% | 1.5873% | 0.793651% | 0.793651% |
| 16K | 5.55556% | 3.1746% | 1.5873% | 0.793651% | 0.793651% |
| 64K | 5.55556% | 3.1746% | 1.5873% | 0.793651% | 0.793651% |
| 256K | 5.55556% | 3.1746% | 1.5873% | 0.793651% | 0.793651% |

Advanced problem:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| LU | 1 | 2 | 4 | 8 | 16 | 32 | 64 |
| 4K | 5.472020% | 3.410320% | 2.294220% | 1.472640% | 0.976593% | 0.558053% | 0.325531% |
| 16K | 3.162300% | 2.371730% | 1.937680% | 1.705160% | 1.178110% | 0.775074% | 0.496047% |
| 64K | 2.340720% | 2.294220% | 2.278720% | 2.232210% | 1.891180% | 1.643160% | 1.085100% |
| 256K | 2.278720% | 2.278720% | 2.278720% | 2.278720% | 2.154700% | 2.030690% | 1.658660% |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| RADIX | 1 | 2 | 4 | 8 | 16 | 32 | 64 |
| 4K | 19.2569000% | 4.2338700% | 1.1520700% | 0.3960250% | 0.1598500% | 0.0835253% | 0.0489631% |
| 16K | 7.3559900% | 0.9792630% | 0.6322000% | 0.3672240% | 0.2260940% | 0.1310480% | 0.0734447% |
| 64K | 1.0109400% | 0.7517280% | 0.6509220% | 0.5457950% | 0.4205070% | 0.2851380% | 0.1598500% |
| 256K | 0.7517280% | 0.7517280% | 0.7502880% | 0.7488480% | 0.6278800% | 0.5198730% | 0.3528230% |

1. Code and Plot Explanation

Basic problem:

從圖表會發現blocksize增加，miss rate會下降，cachesize增加則不影響miss rate，這是因為blocksize會影響存取的資料範圍，所以blocksize越大，範圍越廣，命中的機會就越高，miss rate則下降。而因為資料區間的關係，所以改變cachesize不影響命中的機率

Advancement Problem:

因為要實現n-ways associate，所以令line/n-ways為block數，cache的架構內是存放tag的vector、use\_t是做為表示該index存放幾筆tag，換言之就是vector的size

，而在vector中，最後(最新)被使用的tag會被移動到或存放在vector的頭，反覆運作後vector尾端就即是最久未使用的tag，若要存入新的tag，直接對該index的cache執行pop\_back()指令。而計算miss rate的方法和basic雷同，只要在該index cache中找尋不到對應的tag就是miss。

1. Conclusion

其實這次lab的功能實現上，概念我覺得不難理解，但在programming上還是花了不少時間，主要是C++有段時間沒用，生疏了不少，甚至大概有超過三個小時以上的時間，我單純是因為忘記了vector資料的存入模式，而在原地打轉，該找點時間做些複習了。