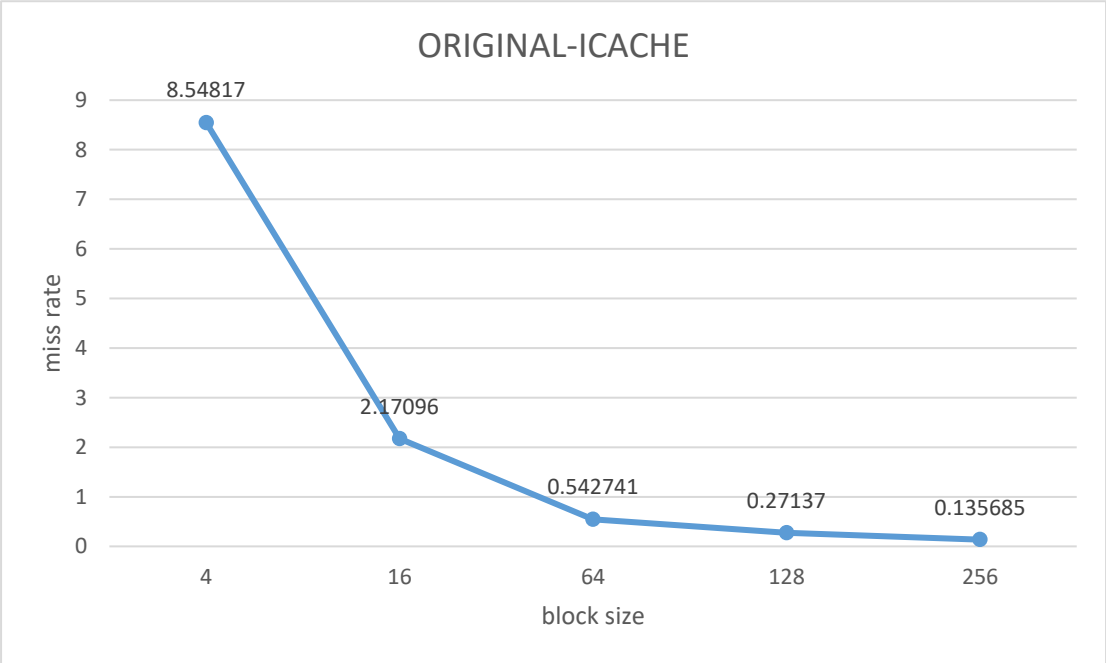
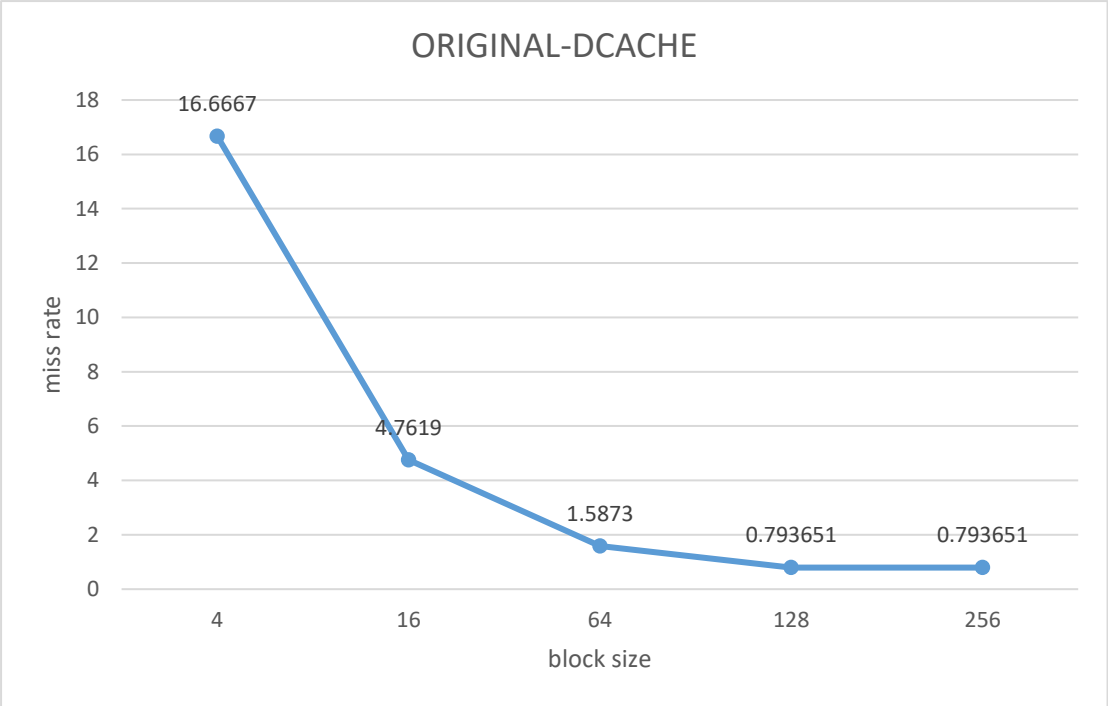
Report

我認為用 PDF 上的圖無法在此做到，因為 PDF 上的圖的 cache size 所能容納的資料量都相對較讀取的 ICACHE 和 DECACHE 的資料數大很多，所以 4K、16K、64K 和 256K 在此所達成的效用是差不多的(得到的 miss rate 在同樣 block size 下都一樣)，因為讀取的資料數不夠多到讓他們顯示出差異，以下是按 PDF 上的數值取得的 ORIGINAL 圖，因產出的數據在 4K、16K、64K 和 256K 都相同，所以只顯示出 1 條折線。




```
ICACHE:
  4* K    4: missrate: 8.54817
  4* K   16: missrate: 2.17096
  4* K   64: missrate: 0.542741
  4* K  128: missrate: 0.27137
  4* K  256: missrate: 0.135685
 16* K    4: missrate: 8.54817
 16* K   16: missrate: 2.17096
 16* K   64: missrate: 0.542741
 16* K  128: missrate: 0.27137
 16* K  256: missrate: 0.135685
 64* K    4: missrate: 8.54817
 64* K   16: missrate: 2.17096
 64* K   64: missrate: 0.542741
 64* K  128: missrate: 0.27137
 64* K  256: missrate: 0.135685
128* K    4: missrate: 8.54817
128* K   16: missrate: 2.17096
128* K   64: missrate: 0.542741
128* K  128: missrate: 0.27137
128* K  256: missrate: 0.135685
256* K    4: missrate: 8.54817
256* K   16: missrate: 2.17096
256* K   64: missrate: 0.542741
256* K  128: missrate: 0.27137
256* K  256: missrate: 0.135685
```



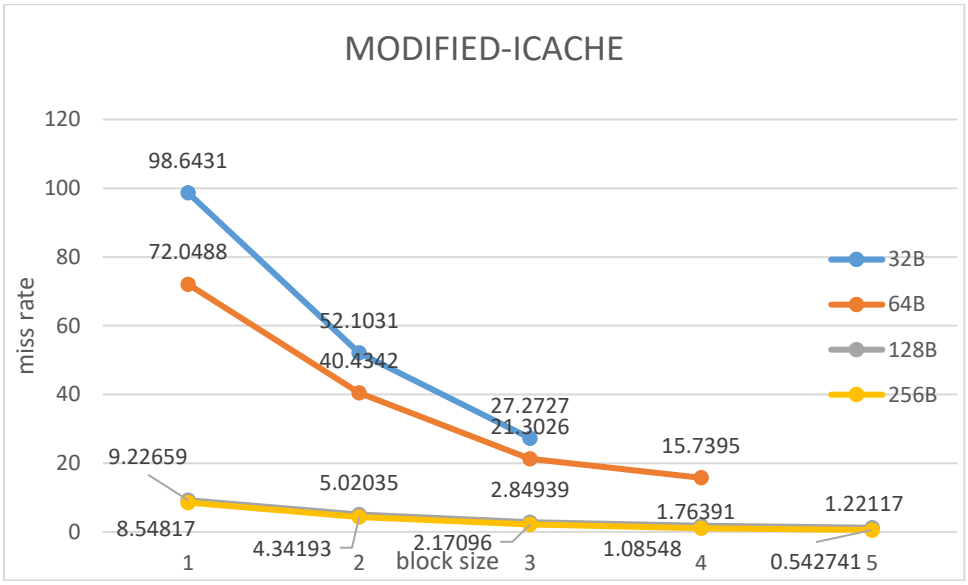

```
DCACHE:
  4* K    4: missrate: 16.6667
  4* K   16: missrate: 4.7619
  4* K   64: missrate: 1.5873
  4* K  128: missrate: 0.793651
  4* K  256: missrate: 0.793651
 16* K    4: missrate: 16.6667
 16* K   16: missrate: 4.7619
 16* K   64: missrate: 1.5873
 16* K  128: missrate: 0.793651
 16* K  256: missrate: 0.793651
 64* K    4: missrate: 16.6667
 64* K   16: missrate: 4.7619
 64* K   64: missrate: 1.5873
 64* K  128: missrate: 0.793651
 64* K  256: missrate: 0.793651
128* K    4: missrate: 16.6667
128* K   16: missrate: 4.7619
128* K   64: missrate: 1.5873
128* K  128: missrate: 0.793651
128* K  256: missrate: 0.793651
256* K    4: missrate: 16.6667
256* K   16: missrate: 4.7619
256* K   64: missrate: 1.5873
256* K  128: missrate: 0.793651
256* K  256: missrate: 0.793651
```

接著我發現 log2 原本的版本有時候會出現計算結果錯誤的情況，我猜可能是小數點和精確度造成的，所以我改用 shift right 的次數來計算 log2 的數值。

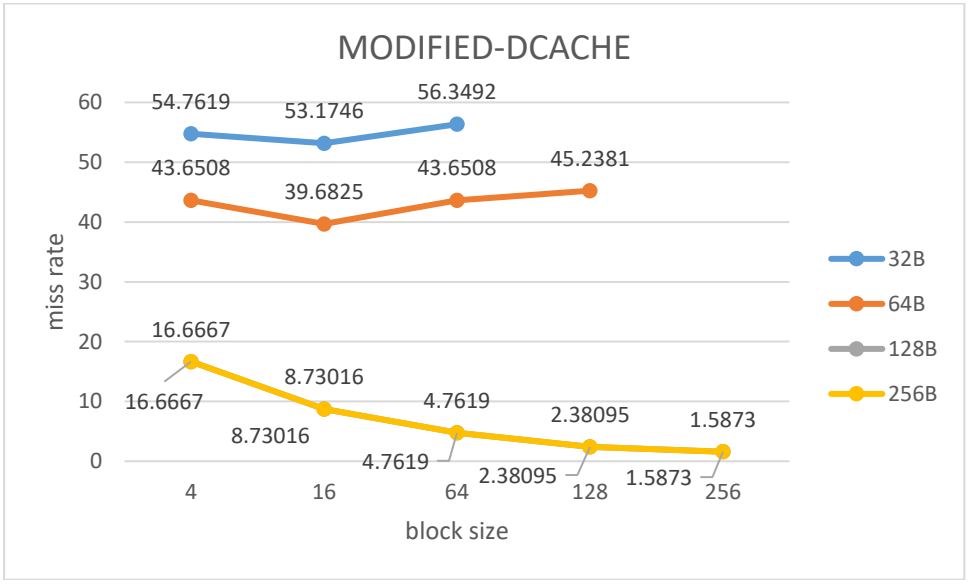
在 pdf 的圖中，block size 會影響 miss rate 的高低，越大的 block size 會因為 spatial locality 的關係降低 miss rate，但因 cache size 有限，越大的 block 也意味著越少的 block 數，造成更多 competition，使 miss rate 上升，所以 miss rate 要視情況而定。

在我修改的部分，因為我認為是 cache size 太大而欲讀取的資料量較少造成 original 那樣的狀況發生，所以我把 cache size 縮小，在 DECACHE 的部分成功找出如 PDF 的圖中那樣的趨勢，但 ICACHE 的部分我認為不可能做到，因為 DECACHE 的資料相較於 ICACHE 的資料較為複雜沒規律，所以較少的 block 對 miss rate 所造成的危害較大，所以在 DECACHE 可以做出兩邊高中間低的圖，而 ICACHE 則只因讀取的 txt 檔資料較有規律，所以較少的 block 影響 miss rate 較少，能做出左邊高右邊低的圖。

以下是修改後的結果:



(128B 和 256B 因數值較接近有稍微重疊)



(128B 和 256B 因數值相同所以重疊)

```
ICACHE:                          DCACHE:
 32       4: missrate: 98.6431   32       4: missrate: 54.7619
 32       8: missrate: 52.1031   32       8: missrate: 53.1746
 32      16: missrate: 27.2727   32      16: missrate: 56.3492
==========                       ==========
 64       4: missrate: 72.0488   64       4: missrate: 43.6508
 64       8: missrate: 40.4342   64       8: missrate: 39.6825
 64      16: missrate: 21.3026   64      16: missrate: 43.6508
 64      32: missrate: 15.7395   64      32: missrate: 45.2381
==========                       ==========
128       4: missrate: 9.22659  128       4: missrate: 16.6667
128       8: missrate: 5.02035  128       8: missrate: 8.73016
128      16: missrate: 2.84939  128      16: missrate: 4.7619
128      32: missrate: 1.76391  128      32: missrate: 2.38095
128      64: missrate: 1.22117  128      64: missrate: 1.5873
==========                       ==========
256       4: missrate: 8.54817  256       4: missrate: 16.6667
256       8: missrate: 4.34193  256       8: missrate: 8.73016
256      16: missrate: 2.17096  256      16: missrate: 4.7619
256      32: missrate: 1.08548  256      32: missrate: 2.38095
256      64: missrate: 0.54274  256      64: missrate: 1.5873
```