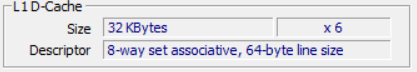
**고급시스템프로그래밍 9주차 실시간수업 과제**

|  |
| --- |
| 제출일: 2021.05.06  이름: 김현용 |

1. 첨부의 9\_1에 대하여 다음을 수행하시오.
2. 각 반복의 배열 별 cache miss ratio의 합을 계산해 보시오. (시스템의 L1 데이터 캐쉬 사이즈를 고려하시오) (3점)

A1-1)

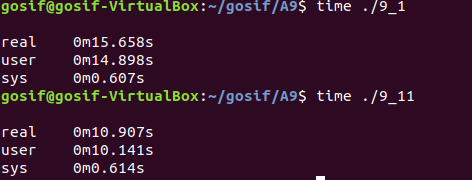
L1 D-Cache Size = 32Kbytes 이고, 그러면 Block Size = 32k/8/64 = 64byte 이다. 이 때의 Misses per inner loop iteration을 구하기 위해 먼저 그림을 그리면 다음과 같다.

R C A B

따라서, Misses per inner loop iteration은 R = 8/64 = 0.125, C = 1, A = 0.125, B = 1이다. 그러므로 0.125+1+0.125+1 = 2.25이다.

1. 반복문 교환(loop interchange)를 사용하여 프로그램의 spatial locality를 조정하여 성능 개선을 확인하시오. (time으로 시간 측정) (2점)

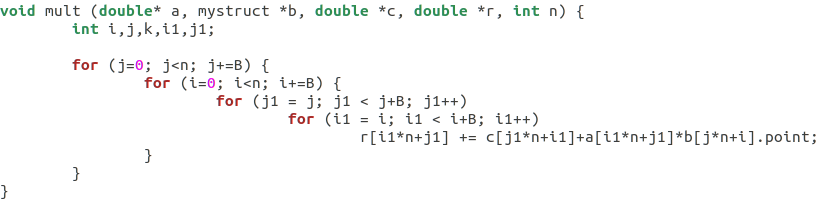
A1-2) ij loop을 ji loop으로 변경해본 뒤 실행 시간의 차이를 비교해 본 결과 다음과 같았다.

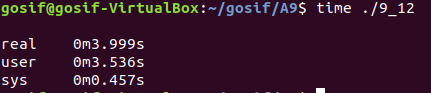


1)번에서 설명한 대로 가로 세로 가로 세로를 ji loop으로 변경해 세로 가로 세로 가로로 변하게 되었고 B는 Struct인데, 1개의 double형과 24개의 char을 가지므로 1 Struct당 총 8+24 = 32Byte 인 것이다. 따라서 B에서 spatial locality가 다른 것에 비해 확연하게 높아졌기 때문에 시간이 줄어들었다.

1. 1번의 주어진 프로그램에 대하여 다음을 수행하시오.
2. 1번의 주어진 프로그램에 대하여 Blocking(Tiling) 기법을 사용하여 Temporal Locality를 향상시켜 프로그램의 성능을 향상시키시오. (전달되는 인자들 간의 메모리 겹침으로 인한 aliasing문제는 발생하지 않는다고 가정한다.) 최적의 Tile 크기는 어느정도가 적당한가?(3점)

A2-1)   
B = 8





Tile의 크기는 8x8 으로 결정했다. 왜냐하면, 현재 cache block size가 64byte이므로 8개의 double이 들어간다. 그러므로, 8x8로 하면 다음 연산에서 이전에 사용했던 a,b,c 등이 cache에서 안 밀려나고 다시 사용가능해져서 Temporal Locality가 향상되기 때문이다.

1. 최적화한 프로그램의 각 반복의 배열 별 cache miss ratio의 합은 어떻게 계산되는가?(2점)

A2-2)

특수한 경우인 구조체를 제외한 나머지의 경우는 8x8 블록당 최초 읽어들일 때 8번 miss만 발생하고 총 블록 개수는 MxM 이므로 8xM/8xM/8번 Miss 발생한다. 그런데 총 read하는 횟수는 MxM 이므로 cache miss ratio는 1/8 = 0.125이다.  
구조체의 경우는 64byte에 32byte짜리 2개 들어가기 때문에 2번당 1번 miss가 발생하고 8x8 블록당 2x8 = 16번 발생한다. 총 블록 개수는 MxM 이므로 16xM/8xM/8번 Miss 발생하고 총 read 하는 횟수는 MxM 이므로 cache miss ratio는 1/4 = 0.25이다. 따라서 총 cache miss ratio은 0.125+0.125+0.125+0.25 = 0.625이다.