1) mmvec

USER CPU TIME: 151.207017

USER SYSTEM CPU TIME: 0.312006

MAXIMUM RESIDENT SET SIZE: 1050408

2) mmvec

USER CPU TIME: 148.595638

USER SYSTEM CPU TIME: 0.271992

MAXIMUM RESIDENT SET SIZE: 1050536

3) mmvec

USER CPU TIME: 148.534394

USER SYSTEM CPU TIME: 0.343996

MAXIMUM RESIDENT SET SIZE: 1050492.

1. Mmnovec

USER CPU TIME: 348.251613

USER SYSTEM CPU TIME: 0.283999

MAXIMUM RESIDENT SET SIZE: 1050544

1. Mmnovec

USER CPU TIME: 346.941634

USER SYSTEM CPU TIME: 0.372001

MAXIMUM RESIDENT SET SIZE: 1050584

1. Mmnovec

USER CPU TIME: 344.531697

USER SYSTEM CPU TIME: 0.343995

MAXIMUM RESIDENT SET SIZE: 1050496

Table for three runs using n=8192

|  |  |  |  |
| --- | --- | --- | --- |
| mmvec | 1 | 2 | 3 |
| User Time | 151.207017 | 148.595638 | 148.534394 |
| System Time | 0.312006 | 0.271992 | 0.343996 |
| Sum |  |  |  |
| Speed Up |  |  |  |
| Maximum Resident Size |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| mmnovec | 1 | 2 | 3 |
| User Time | 348.251613 | 346.941634 | 344.531697 |
| System Time | 0.283999 | .372001 | .343995 |
| Sum |  |  |  |
| Speed Up |  |  |  |
| Maximum Resident Size |  |  |  |

Implementation:

My first attempt was multiplying the matrix by row and column for second matrix, but the only aligned matrix was the first one, so this took a long time because data was not aligned for both matrix and it deprived the compiler to maximize the matrix multiplication. However, to have a better result I transpose the second matrix to align the data and this improved the performance and achieved better vectorization. Another implementation to have a better performance was to set the alignment to 32 bits because that is the size of a float, so I wanted to aligned the data base on the float size.