

کد مربوط به موازی سازی سطری :

```
void add_row_parallel(DataSet dataSet) {
    #pragma omp parallel for num_threads(NUM_THREADS)
    for (int i = 0; i < dataSet.n; i++) {
        for (int j = 0; j < dataSet.m; j++) {
            dataSet.C[i * dataSet.m + j] = dataSet.A[i * dataSet.m + j] +
dataSet.B[i * dataSet.m + j];
        }
    }
}
```

داده های برداشته شده از اجراها و تسريع به دست آمده :

تعداد نخ	1mb	10mb	100mb	1gb	تسريع
8	0.001833	0.005723	0.041776	0.378428	3.48085
4	0.001609	0.006261	0.040674	0.364387	3.61498
2	0.000943	0.006995	0.071831	0.695543	1.89384
1	0.001523	0.013248	0.132549	1.317252	1

کد مربوط به موازی سازی بلاکی :

```
void add_block_parallel(DataSet dataSet) {
    int row_block = (int)ceil(dataSet.n / (double)BLOCK_SIZE);
    int col_block = (int)ceil(dataSet.m / (double)BLOCK_SIZE);
    #pragma omp parallel for num_threads(NUM_THREADS) collapse(2)
    for (int i = 0; i < row_block; i++) {
        for (int j = 0; j < col_block; j++) {
            //printf("block : %d -- %d\n", i, j);
            add_block(dataSet, i, j);
        }
    }
}
```

در این ماتریس به بلاک های ۱۶ در ۱۶ (بسته به BLOCK_SIZE) تقسیم می شود و محاسبه جمع آن به ترد ها واگذار می شود. اطلاعات استخراج شده از اجرا ها :

تعداد نخ	1mb	10mb	100mb	1gb	تسریع
8	0.001200	0.002963	0.030040	0.316247	4.5598
4	0.000937	0.004304	0.046205	0.421973	3.4173
2	0.000724	0.005089	0.053105	0.818489	1.76182
1	0.001084	0.009973	0.098086	1.442031	1

اعداد به دست آمده از اجرای هر الگوریتم به ازای میانگین گیری از ۱۰ بار اجرا به دست آمده اند.