

OS HW2 REPORT

E94091071 資訊 113 黃琳

- The method of dispatching works to the worker threads
 - element dispatch

將 result matrix 中的 element 由上到下且由左到右輪流分配給各個 thread 計算，利用 k 表示其為第幾個 thread，將 k 除以 result matrix column 數(即 matrix2 column 數)，商為須負責 element 的 row，餘數為 column，再將 k 加 thread 的數量，在 k 超過 result matrix size 前重複動作。

```
/*matrix multiplication (element dispatch)*/
void *multi(void* num){
    FILE *proc_info;
    int thread_i=(*(int*)num);          /*the number of thread (not thread id)*/

    for(int k=thread_i;k<matrixres_size;k+=thread_num){
        int Q = k/col2;
        int R = k%col2;
        long long unsigned total=0;
        for(int w=0;w<col1;w++){
            total+=matrix1[Q][w]*matrix2[w][R];
        }
        pthread_mutex_lock(&mutex);
        matrix_res[Q][R]=total;
        pthread_mutex_unlock(&mutex);
    }
}
```

計算該 element 的值

假設 matrix1: 3x2

matrix2: 2x4

⇒ result: 3x4

有 5 個 thread: 0, 1, 2, 3, 4

result	0	1	2	3
0	0	1	2	3
1	4	0	1	2
2	3	4	0	1

thread 1 負責:

$$k=1 \Rightarrow Q=1/4=0$$

$$R=1\%4=1$$

$$R[0][1] = M_1[0][0] * M_2[0][1] + M_1[0][1] * M_2[1][1]$$

k+=5

$$\Rightarrow k=6 \Rightarrow Q=6/4=1$$

$$R=6\%4=2$$

$$R[1][2] = M_1[1][0] * M_2[0][2] + M_1[1][1] * M_2[1][2]$$

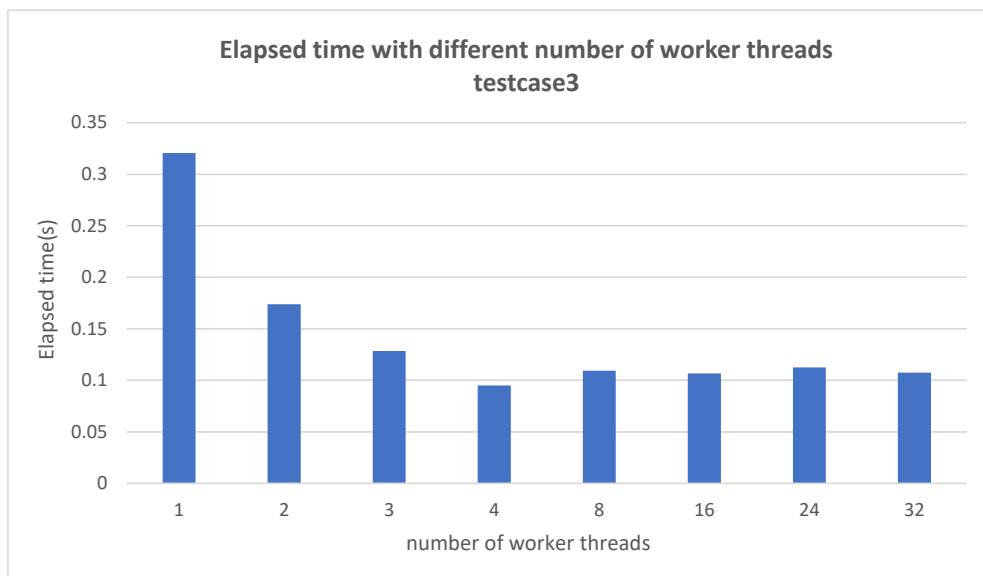
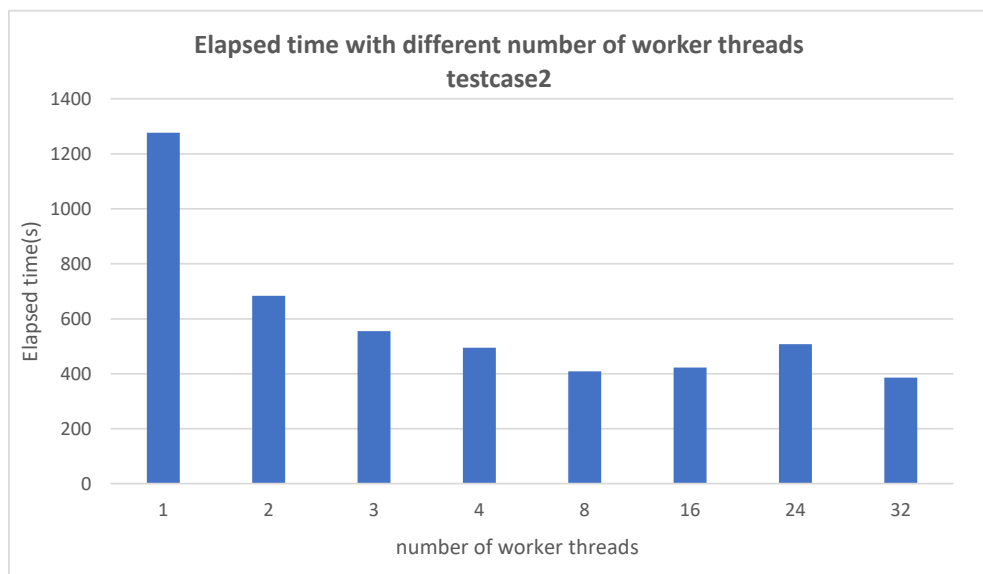
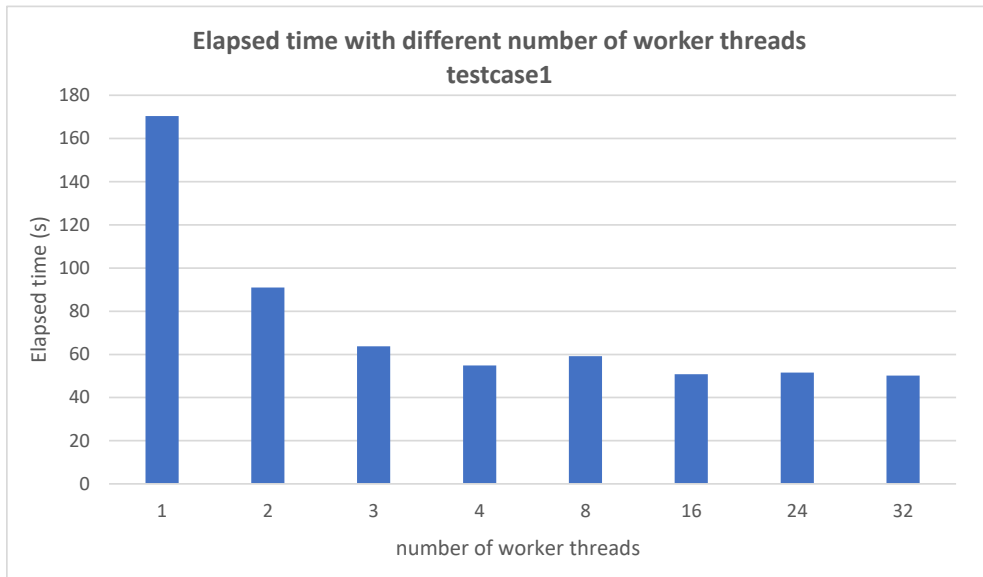
k+=5

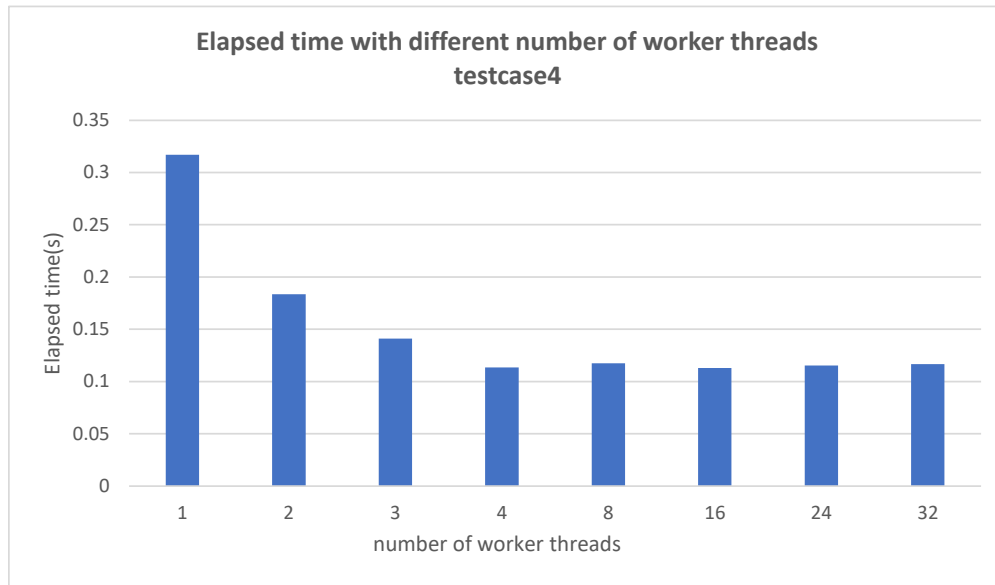
$$\Rightarrow k=11 \Rightarrow Q=11/4=2$$

$$R=11\%4=3$$

$$R[2][3] = M_1[2][0] * M_2[0][3] + M_1[2][1] * M_2[1][3]$$

- **Charts of elapsed time with different number of worker threads**





- **Summarize**

When the number of threads is less than the number of cores, it speeds up with more worker threads, because of parallelization.

However, when the number of threads is greater than the number of cores, the elapsed time doesn't change much with adding more worker threads, because the speedup is limited by the serial portion.

