PL HW4
Matrix Multiplication

第18組
張哲 孫上智 郭彥松 許雁婷

使用語言

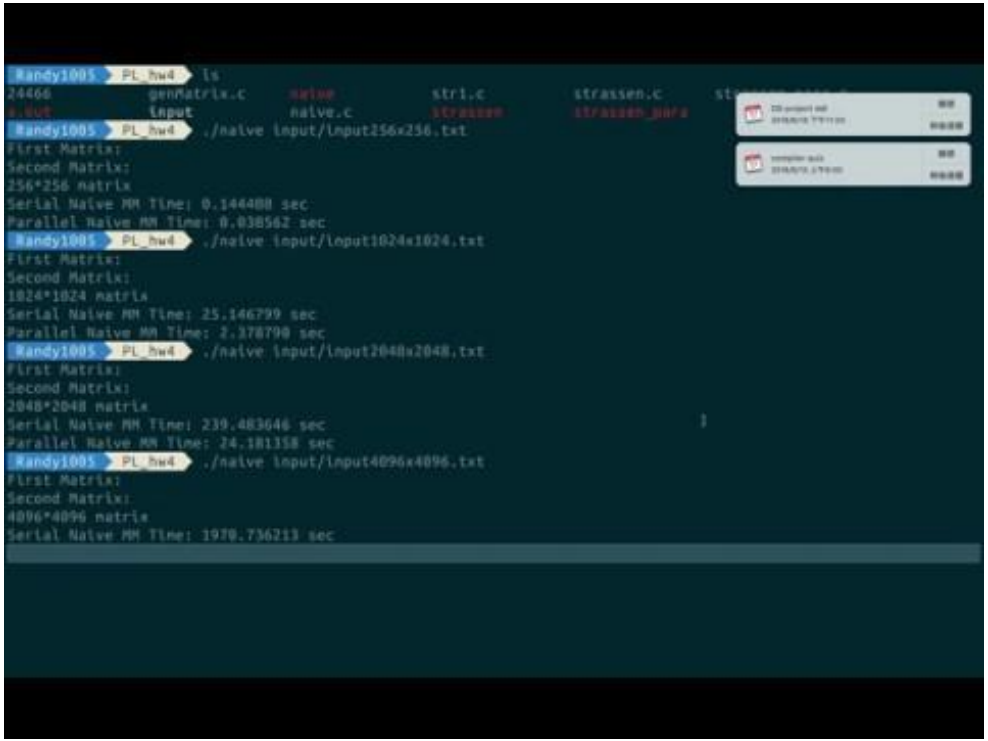
- C
- OpenMP

傳統矩陣乘法 (Serial + Parallel)

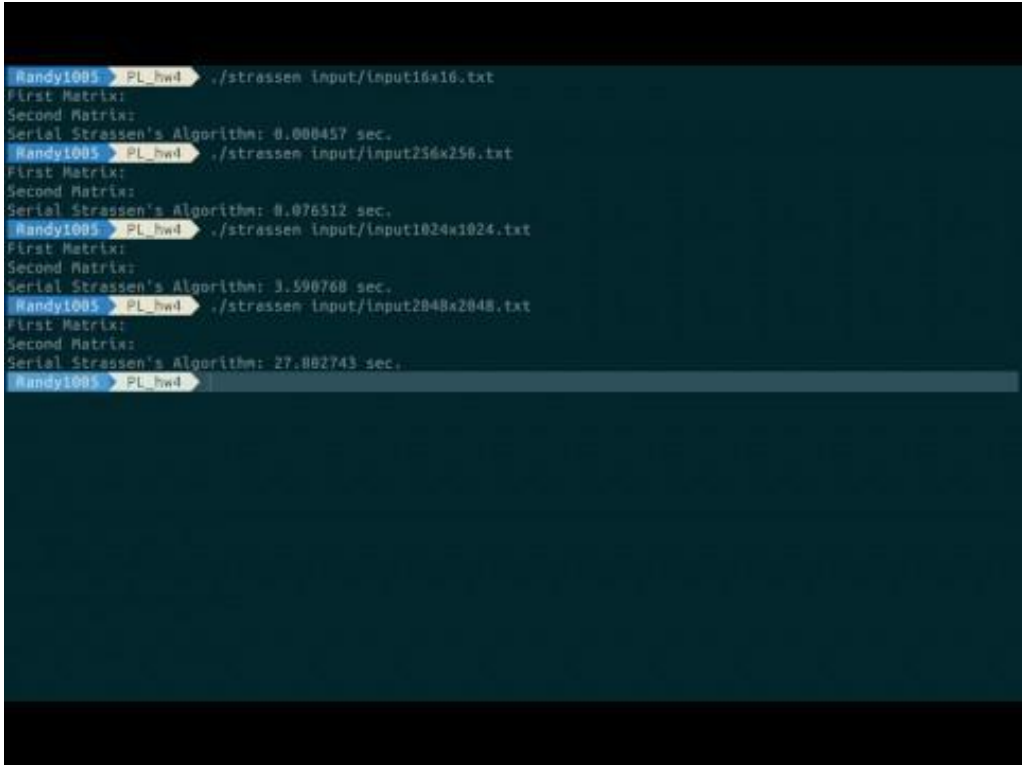
平行for迴圈
openmp for
#pragma omp parallel for

```
#pragma omp parallel for
for (int i=0; i < row; i++){
    for (int j=0; j < col; j++){
        for(int k=0; k < tmp; k++){
            ansMatrix[i][j] += fMatrix[i][k]*sMatrix[k][j];
        }
    }
}
```

傳統矩陣乘法 (Serial + Parallel)

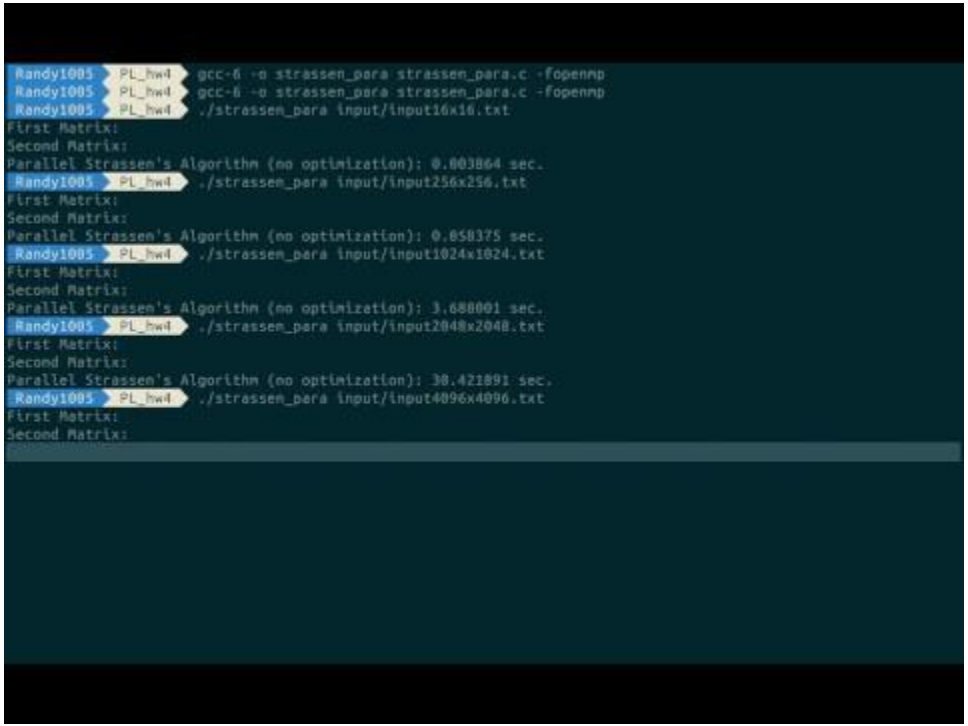


Strassen’s ALGO (Serial)

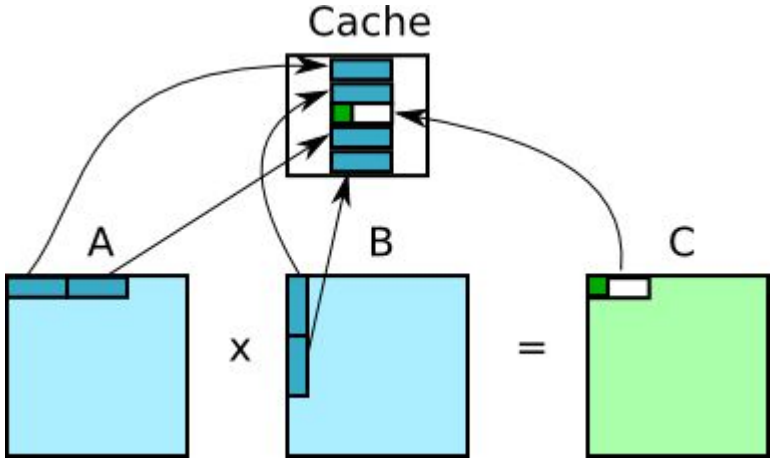


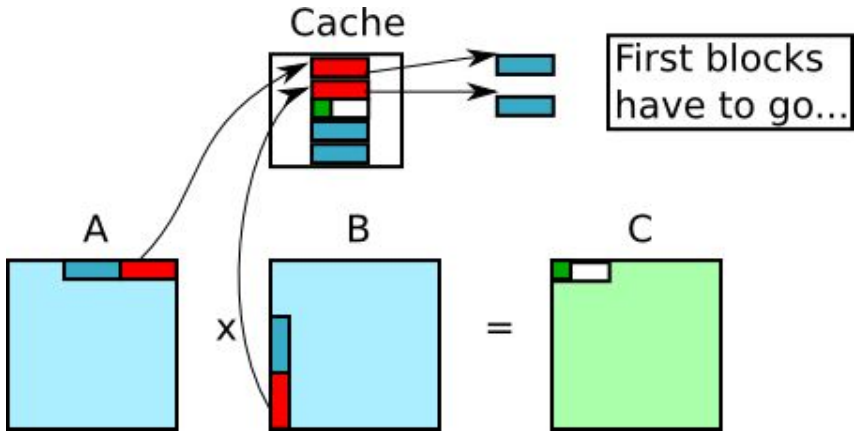
Strassen’s ALGO (Parallel/no optimization)

平行 baseline矩陣乘法的 for迴圈
openmp for
#pragma omp parallel for

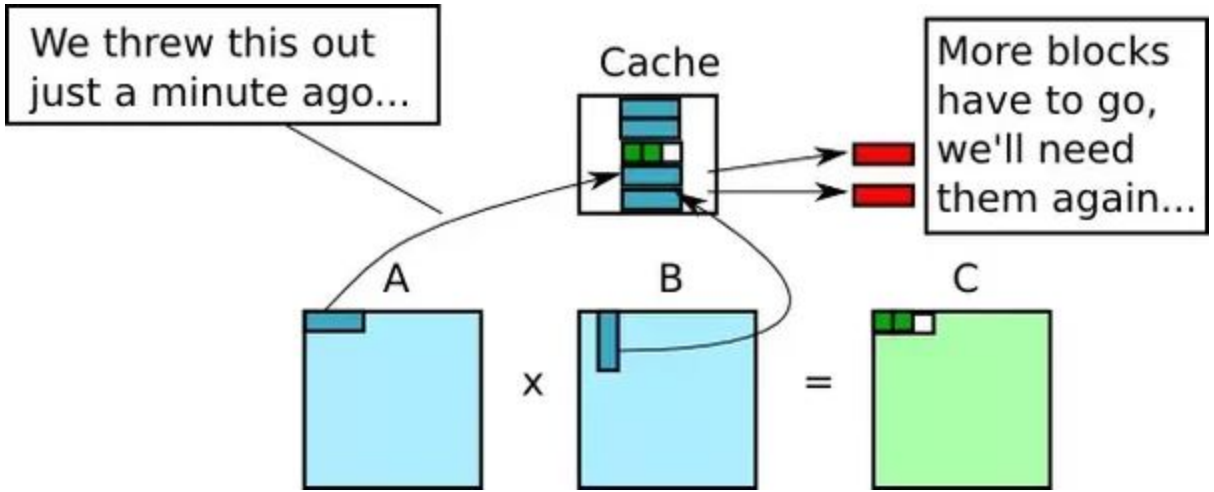


其他加速方法：cache locality





其他加速方法




```
for (int i=0; i < row; i++){
    for (int j=0; j < col; j++){
        for(int k=0; k < tmp; k++){
            ansMatrix[i][j] += fMatrix[i][k]*sMatrix[k][j];
        }
    }
}
```

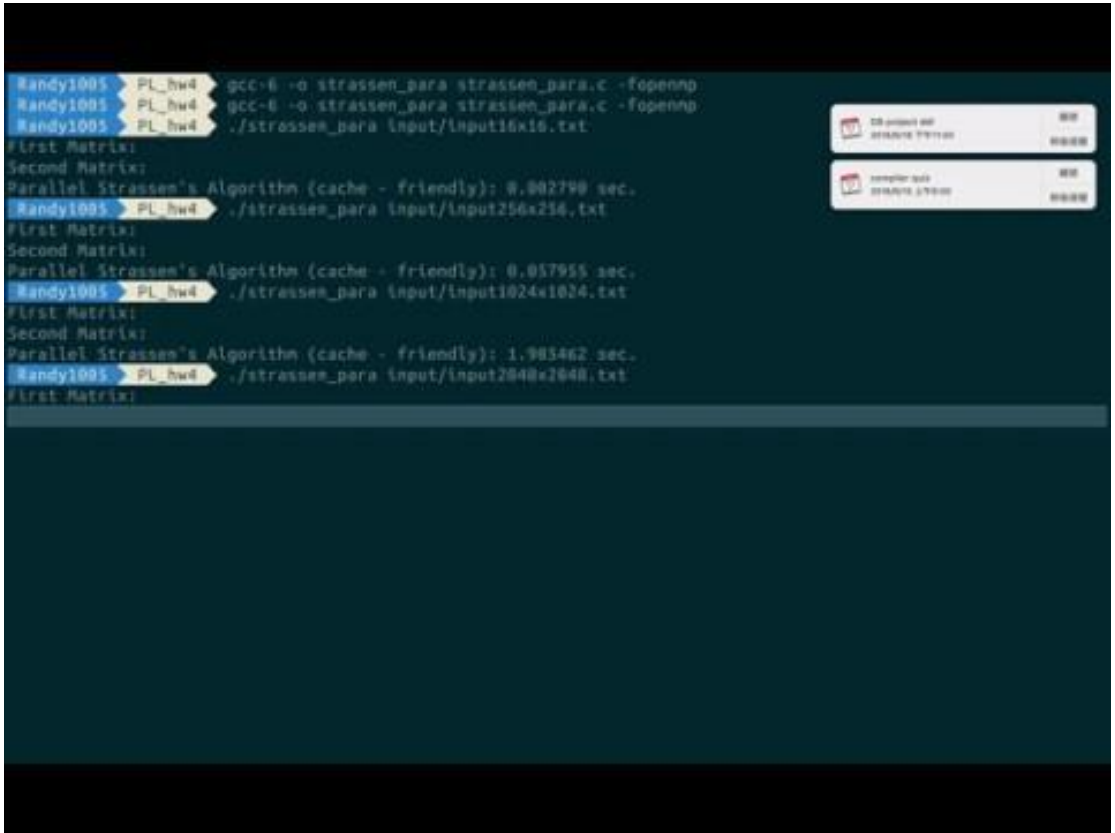
```
for (int i=0; i < row; i++){
    for (int k=0; k < col; k++){
        for(int j=0; j < tmp; j++){
            ansMatrix[i][j] += fMatrix[i][k]*sMatrix[k][j];
        }
    }
}
```

Strassen’s ALGO (Parallel/optimization)

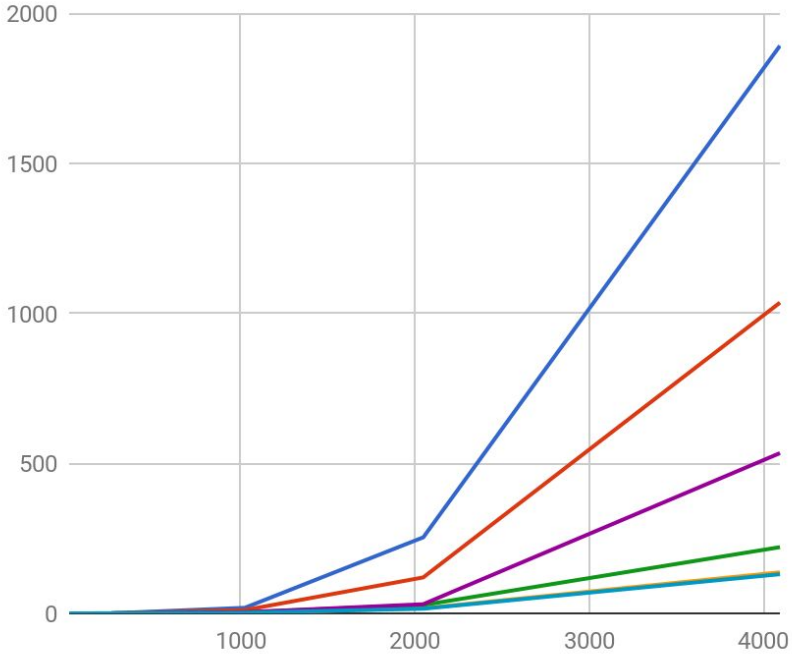
- cache locality + 只做2層recursion 就換baseline的乘法

```
if(size <= (row/4)){
    mat_mul(size, fMat, sMat, ansMat);
}
```

Strassen’s ALGO (Parallel/optimization)



比較



size(n)	Naive	P Naive	Strassen	P Strassen	Optimize
16	0.000076	0.000598	0.000388	0.003864	0.000598
256	0.132198	0.042979	0.076512	0.058375	0.042979
1024	18.512002	2.005506	3.590768	3.688001	2.005506
2048	253.222766	16.767355	27.802743	30.4211891	16.767355
4096	1891.756694	136.73551	220.610812	534.010275	136.73551

分工

張哲:strassen

孫上智: strassen加速

郭彥松:傳統

許雁婷:傳統加速