



## **CS 426 Project 3 Report**

**Eren Aytüre -**

**21200559 - Section 1**

# 1)Introduction

In this project, sparse matrix - vector multiplication is done according to the given input files. I am going to talk about the parallelisation strategy I made, and the result will be explained.

## 2) Parallelisation strategy

I have used triplet data structure in the form of COO format representation. Then shown in the below figure. I used a 2d array for the result vector due to preventing race condition. However, each thread is parallelised according to the size of the value pointer array. After dividing the work, the result vector takes an index from the row pointer array. Therefore, there could be many same indexes at the same time, It can be retrieved from different threads. To prevent that first, There are # of process, number of resulting vector that each thread works without race condition than these results vectors are reduced in real result vector.

```
result_vector[omp_get_thread_num()][row_ptr[i]-1] +=  
(x[col_ptr[i]-1] * val_ptr[i]);
```

## 3)Results

First I have calculated  $T_p, T_s, S, E$  as

$$T_s = overallTime - f$$

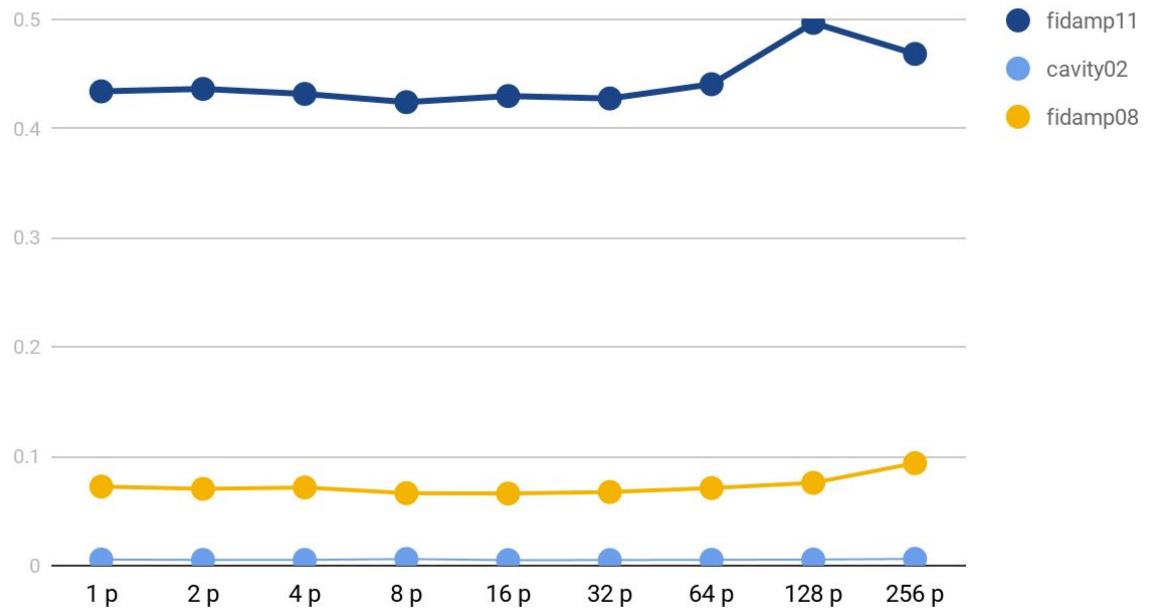
$$T_p = f$$

$$T_p = overallTime - f + \frac{f}{\#p}$$

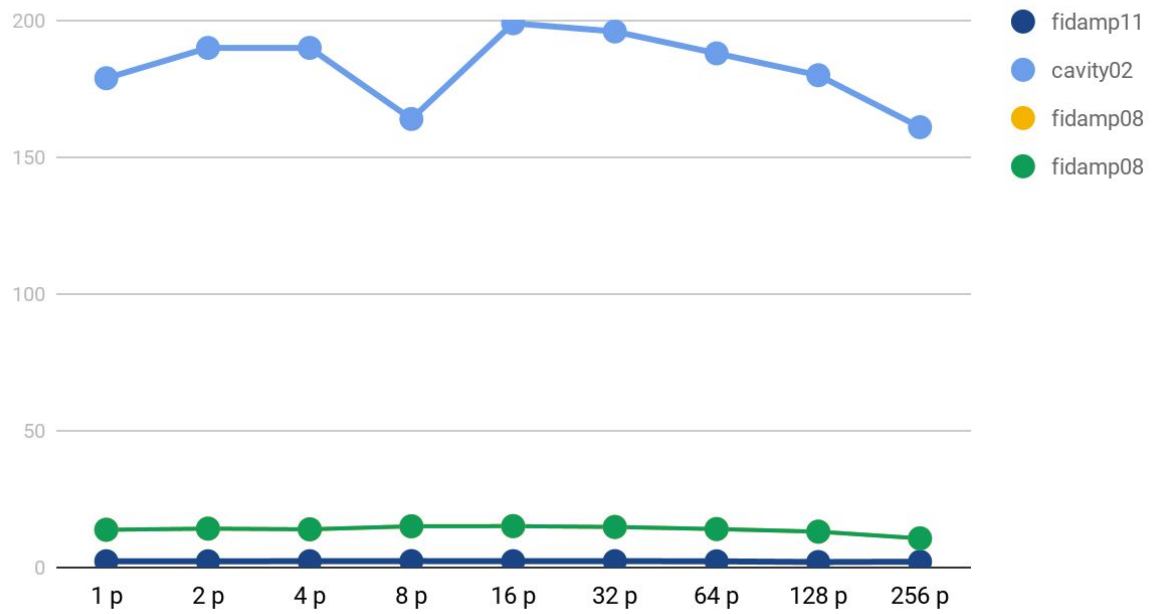
$$S = \frac{1}{T_p}$$

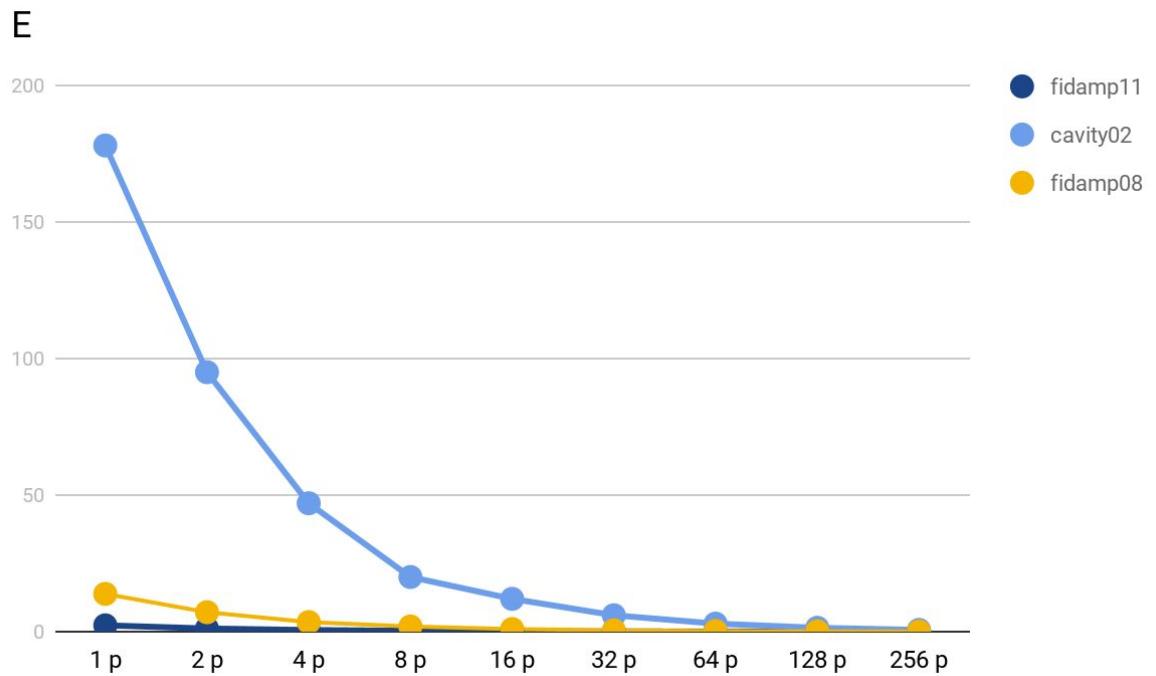
$$E = \frac{S}{\#p}$$

Tp



S





For cavity02 files since size is small, efficiency decreases. That means more processes are appropriate for large size inputs. However, it has the fastest execution times. The others speedup tp and efficiency is proportional to number of processes. The more processes, the more speedup and less Tp time, Since my computer has 8 cores. Best results are received from 8 and 16 processes.