Task scheduling algorithm for cloud computing environment.

## Abstract

Cloud computing is one of the popular models that gives us flexibility and is on-demand. Scheduling in the cloud is responsible for selection of best suitable resources for task execution, by taking some static and dynamic parameters and restrictions of tasks into consideration. The user perspective of efficient scheduling may be based on parameters like task completion time or task execution cost etc. Task scheduling problem in cloud computing environment is NP-hard problem, which is difficult to obtain exact optimal solution and is suitable for using intelligent optimization algorithms to approximate the optimal solution. This project implements and compares different task scheduling algorithms that addresses these major challenges of task scheduling in the cloud. The selection of resources for execution of tasks is based on the greedy approach.

# ACKNOWLEDGEMENT

We wish to express our deep gratitude to our mentor Asst. Prof. Avita Katal, for all advice, encouragement, and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank our cluster head, Prof. (Dr.) Neelu J. Ahuja, for her great support in doing our project at SoCS.

We are also grateful to Dr. Ravi S. Iyer Dean, SoCS, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all faculties from UPES for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

**Introduction**

In this era of rapidly increasing technology, every system needs to be fast and highly efficient. Moreover, so as to achieve high throughput and to make everything cost effective it is very important to find the best task scheduling algorithms. Task scheduling is the main problem in cloud computing which reduces the system performance. To improve system performance, there is a need for an efficient task-scheduling algorithm. Existing task-scheduling algorithms focus on task-resource requirements, CPU memory, execution time and execution cost.The process of deciding which task will utilize the CPU time is called task scheduling.The task scheduling is basically static or dynamic. The scheduling of the task may be on the basis of their priorities. Task scheduling is an essential part of a Multiprogramming operating system. Such operating systems allow more than one process to be loaded into the executable memory at a time and the loaded process shares the CPU using time multiplexing. Multi­resource scheduling is for minimum task completion time, in order to have high resource utilization and minimum queuing delay.

Nowadays, we do have many task scheduling algorithms for our local systems but as everything is shifting to the cloud environment rapidly, so as to keep the pace we need to find better algorithms for the cloud systems. Some of the most efficient algorithms that could be used are MIN-MIN, MAX-MIN , RASA , Enhanced MAX-MIN and Multilevel Queue Scheduling. Proper comparison will be made in the project so as to find the best algorithm for different tasks.

## Literature Review

In paper [1],of “Optimization of tasks in cloud computing based on MAX-MIN, MIN-MIN and priority” it is found that considering the growing use of cloud computing and the need for optimal use of resources in the cloud, and attention to users that pay for services they use based on their pay-as-you-go basis. There should be a quicker way for users to decrease the user's waiting time and task’s waiting time. [2] Task scheduling is the main problem in cloud computing which reduces the system performance. To improve system performance, there is a need for an efficient task-scheduling algorithm. Existing task-scheduling algorithms focus on task-resource requirements, CPU memory, execution time and execution cost. However, these do not consider network bandwidth.

Task scheduling not only improves execution time, cost, response time, flow time, throughput, and average resource utilization, but also improvement is required in some areas like makespan, Time/space complexity & execution cost. They observed that Optimization based task scheduling may be further considered for achieving more effective task scheduling. In this paper of “Review paper on various scheduling techniques in cloud environments. They take existing scheduling algorithms and they are compared by using different parameters as well as tools. Mostly they all are worked on to minimize the execution time, faster response time and maximum utilization of resources. They found that Existing scheduling algorithms do not consider load balancing, availability and reliability. Therefore, there is a need to implement such scheduling algorithms that can improve the reliability, availability and load balancing in a cloud computing environment. In the future, algorithms based on migration of tasks from one machine to another can also be introduced.

## Objectives

* To implement task scheduling algorithms for cloud environments.
* To understand Data Structures and Algorithms.
* To manage cloud computing performance and Quality of Service.
* To schedule tasks on a real-time system. Achieving a high system throughput. Improving load balance

## Methodology

The project flow (Fig 1) starts from taking the input from the data set file. Then the data set is allocated to different resources in the methods. Each method performs a different algorithm. The result for each type of algorithm is compared and then the result will be represented graphically.

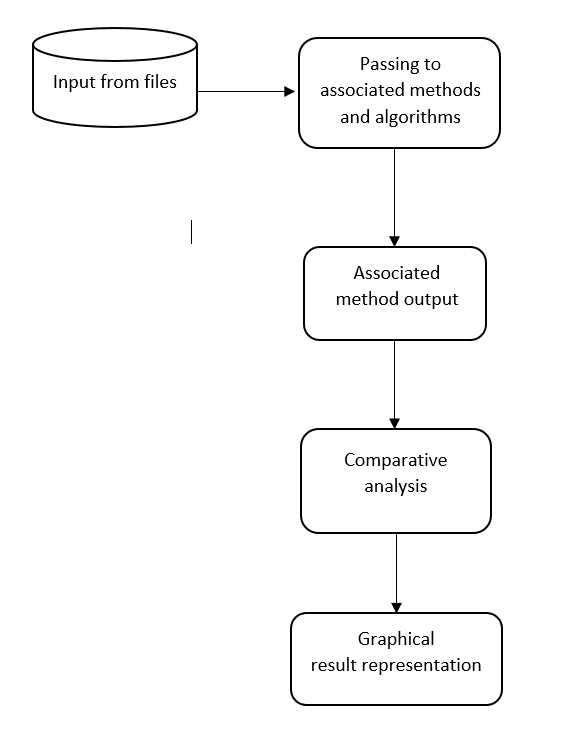


Fig 1: Overall Flow of the Project

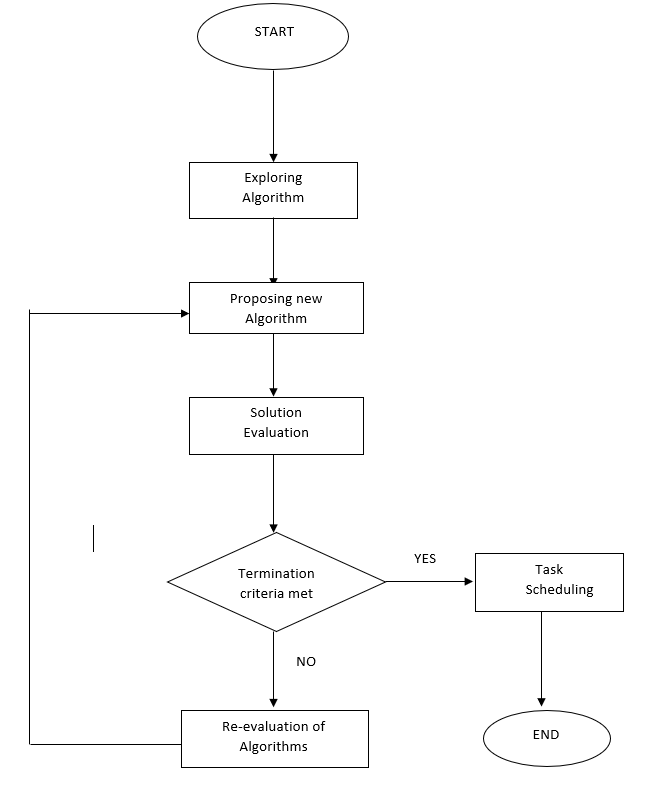
****

Fig 2: Detailed overall flow of the project

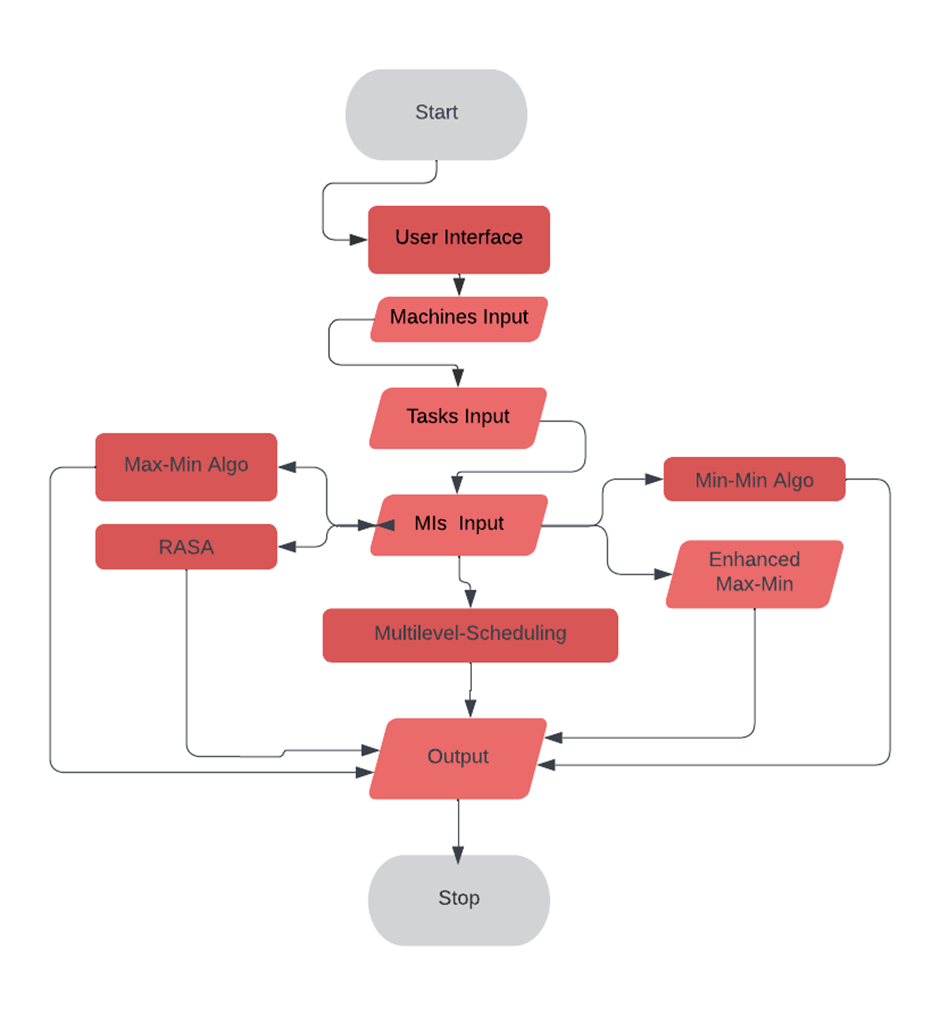


Fig 3. Process flo

#### Algorithms

MIN-MIN- The Min-Min algorithm first finds the minimum execution time of all tasks. Then it chooses the task with the least execution time among all the tasks. The algorithm proceeds by assigning the task to the resource that produces the minimum completion time. The same procedure is repeated by Min-Min until all tasks are scheduled.

MAX-MIN- The Max-Min algorithm first finds the minimum execution time of all tasks. Then it chooses the task with the maximum execution time among all the tasks. The same procedure is repeated by Max-Min until all tasks are scheduled.

RASA - A new algorithm named RASA follows the Max-min and Min-min strategy alternatively in order to assign tasks over available resources. As it adapts the advantage of both the existing algorithm, it supports synchronized execution of large and small tasks which avoids delay in execution of larger tasks.

ENHANCED MAX-MIN - The Max Min algorithm schedules the task as per the overall maximum execution time over the resource that provides overall minimum completion time. This algorithm supports load balancing of available resources and allows concurrent execution of the submitted tasks. Total makespan is calculated over larger task execution.

MULTILEVEL QUEUE SCHEDULING - This algorithm introduced a modified Max-min algorithm [8]. This algorithm selects the task with the average or nearest to average execution time instead of selecting the largest task for execution. This selected task is then scheduled over the resource with minimum completion time. This algorithm reduces the overall makespan and also balances load across resources.

In a cloud environment, job scheduling is done by adapting the existing scheduling algorithms. These algorithms when scheduled tasks follow the same principal in each situation sometimes leading to increase in the makespan of the processes. Our proposed algorithm adapts the advantages of various existing task scheduling algorithms and applies one of them on the basis of the situation and calculations performed. Since high throughput and load balancing is equally important, the distribution of the tasks over available resources is done in order to achieve lesser makespan.

#### Data Structures:

Queue Data structure

Priority Queue Data Structure

Stack Data Structure

Hash Map.

Array List

Array

**Area Of Application:**

The target beneficiaries of the proposed methodology are all the organizations which are process dependent and need to complete their task of any sort , as well as cloud users and all the people who are using cloud systems. Even the bigger companies like aws, azure and banking sector, all IT industry and many more people or organizations who are totally system dependent and working on cloud environments are the beneficiary target because using better task scheduling algorithms helps in a greater aspect.

## 7. Implementation and Results

The MIPS (Million of Instructions per second) of the System and the MIs (Million of Instructions) of the Tasks are taken from the user.

The user chooses from the different “Task Scheduling Algorithms” (MIN-MIN , MAX-MIN RASA , Enhanced MAX-MIN) .

**MAIN CLASS :**

**Output**

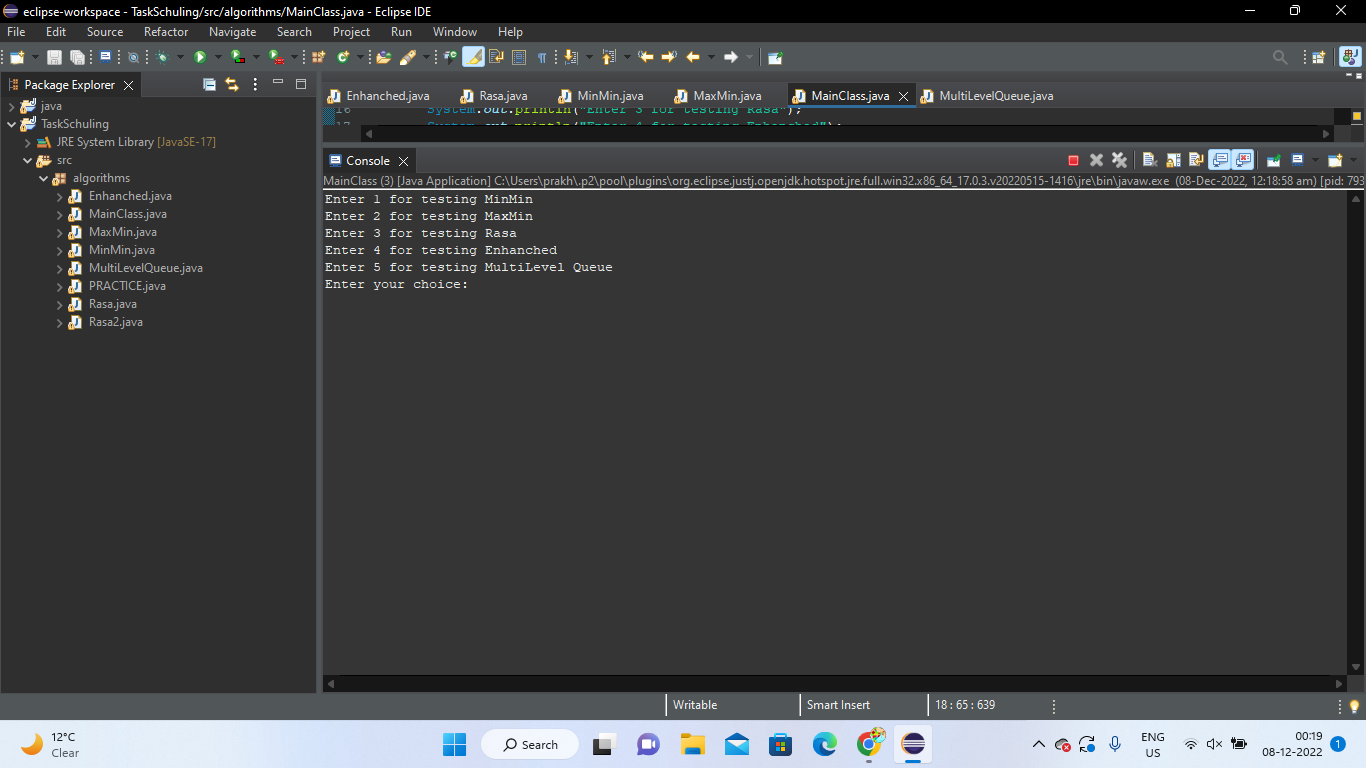


Fig 4 : Output for main class

**MIN- MIN:**

**OUTPUT:**

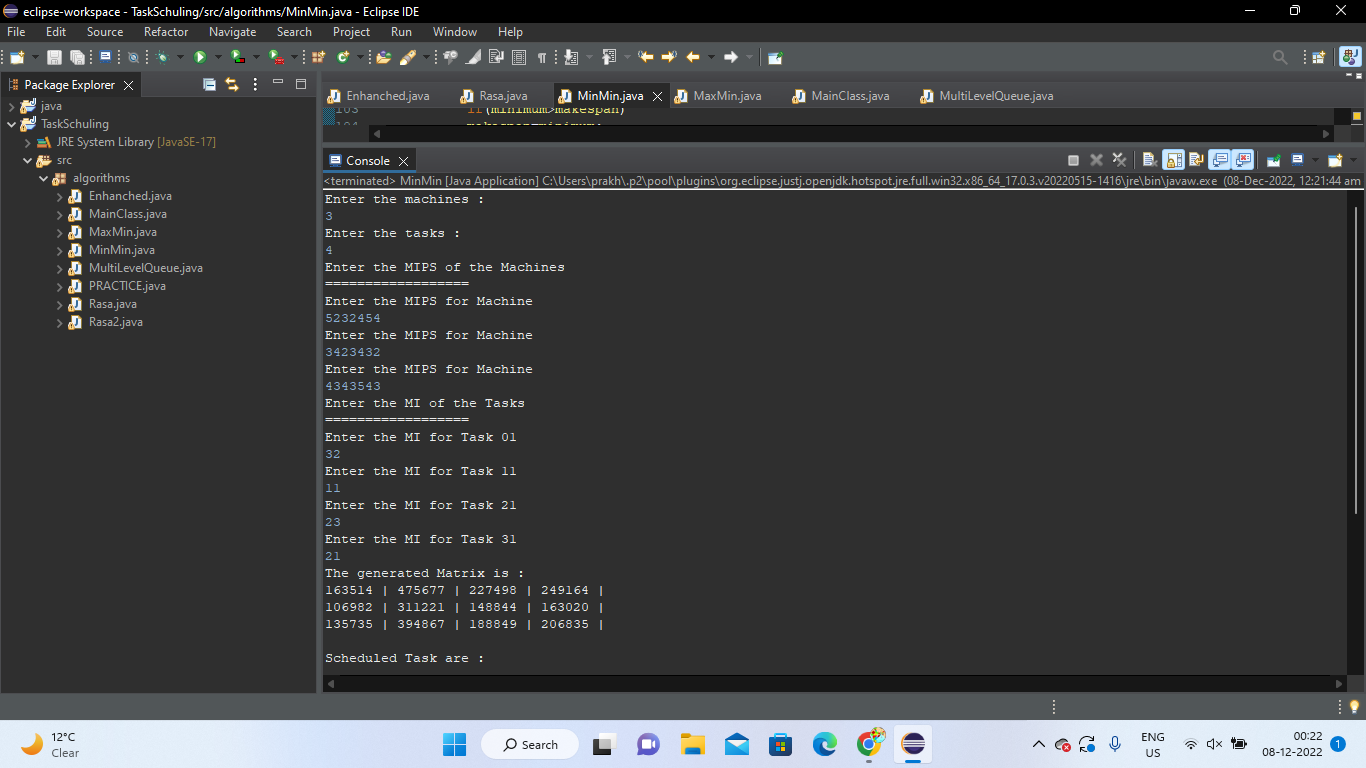


Fig 5 : Output for MIN-MIN

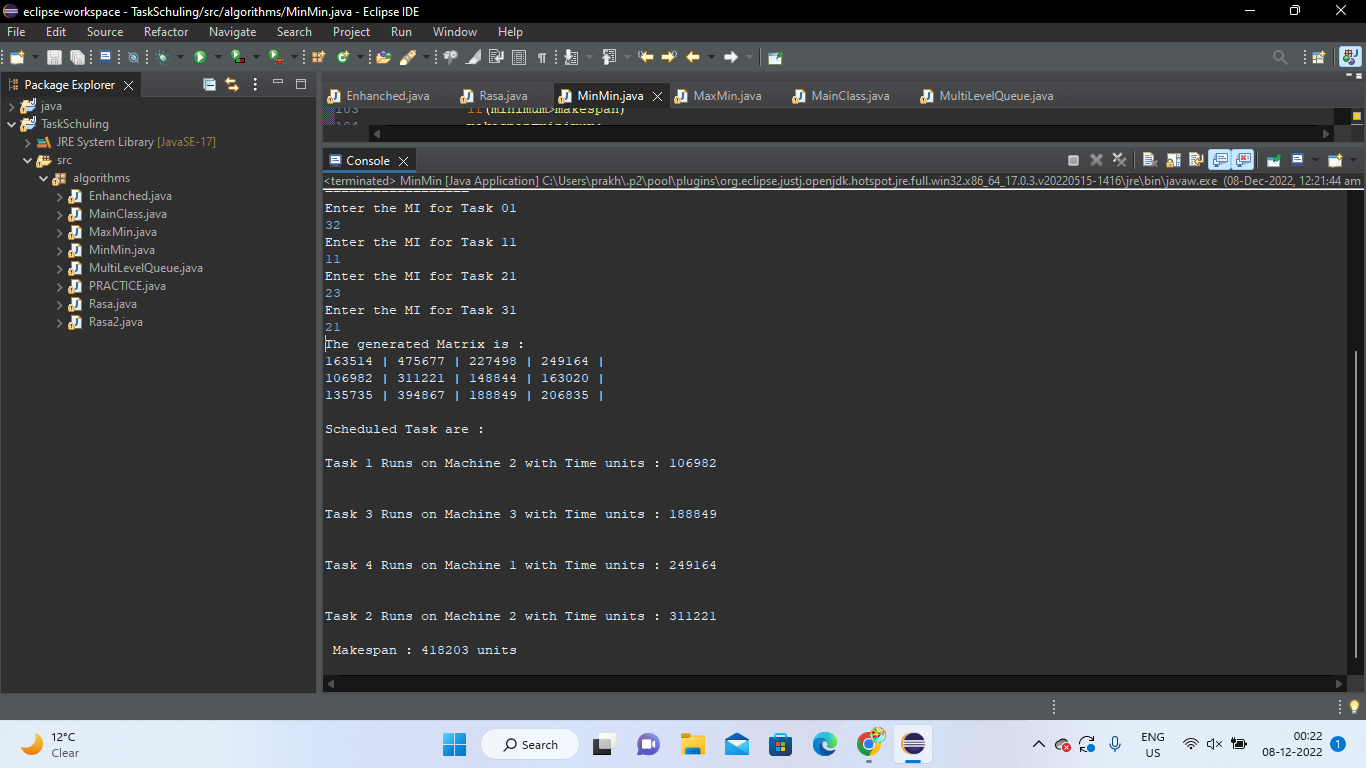


Fig 6 : Output for MIN-MIN

**MAX-MIN:**

**Output:**

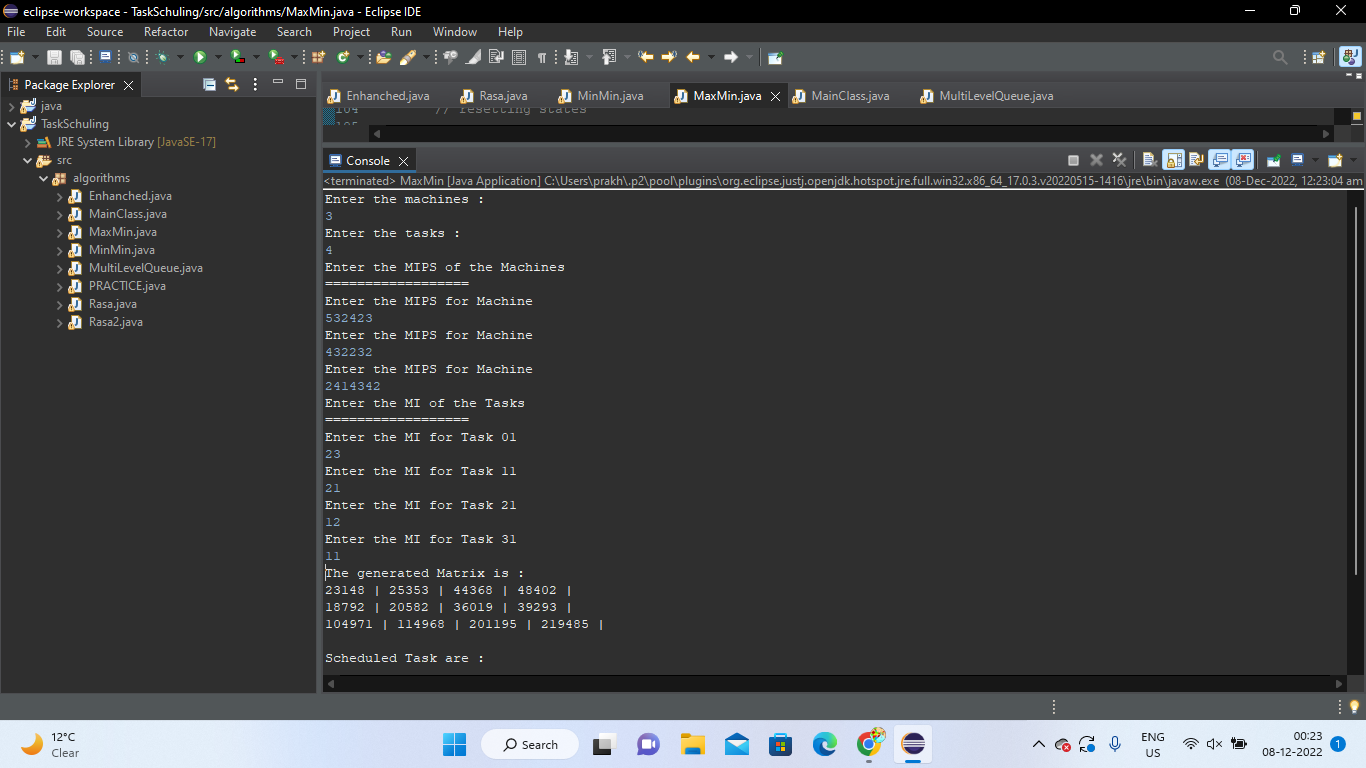


Fig 7: Output for MAX-MIN

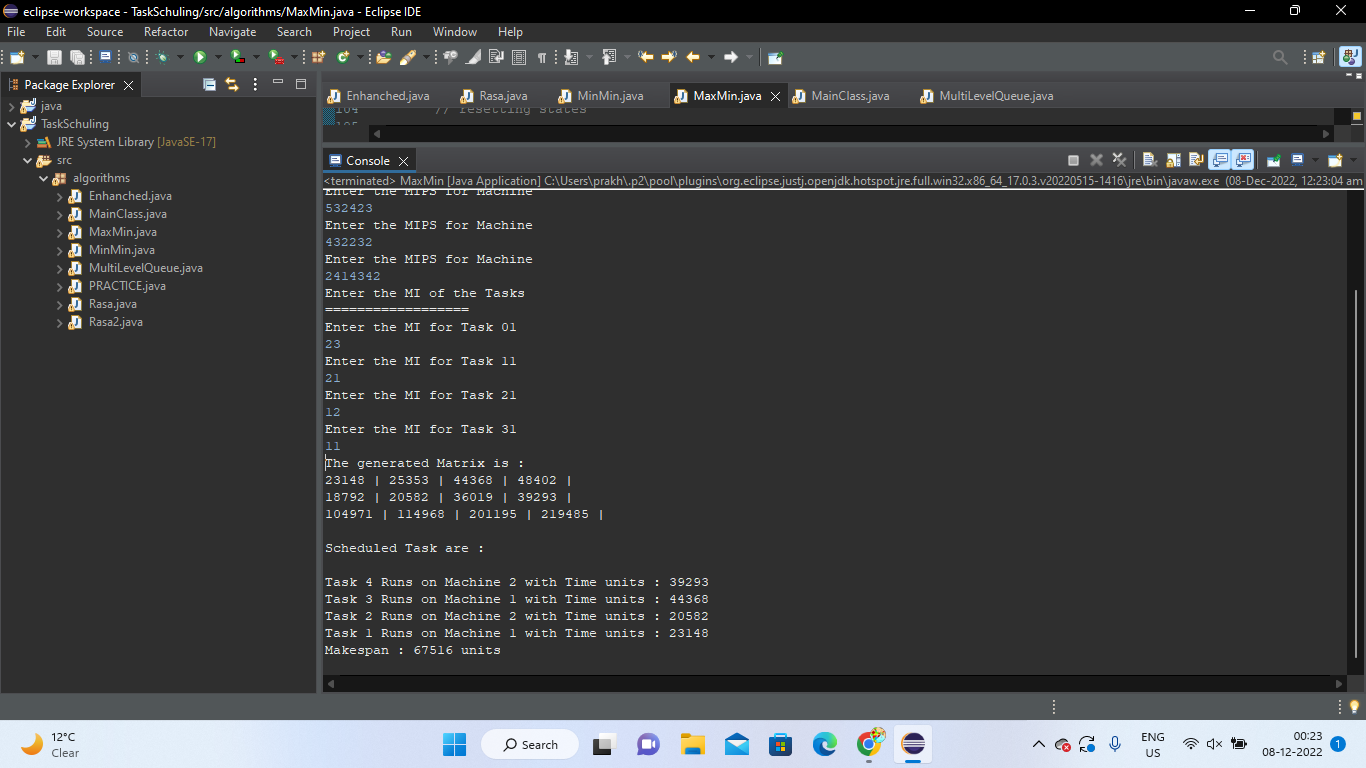


Fig 8 : Output for MAX-MIN

**RASA:**

**Output:**

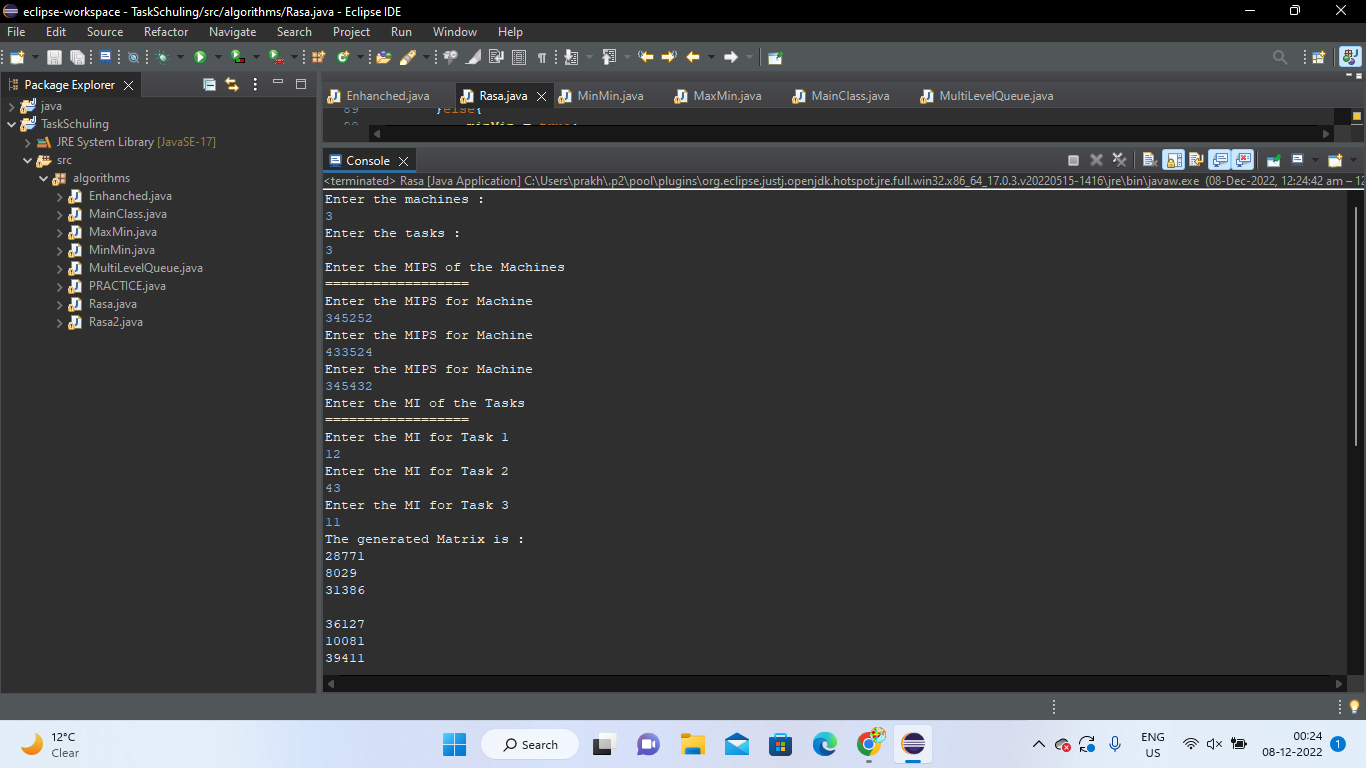


Fig 9 : Output for RASA

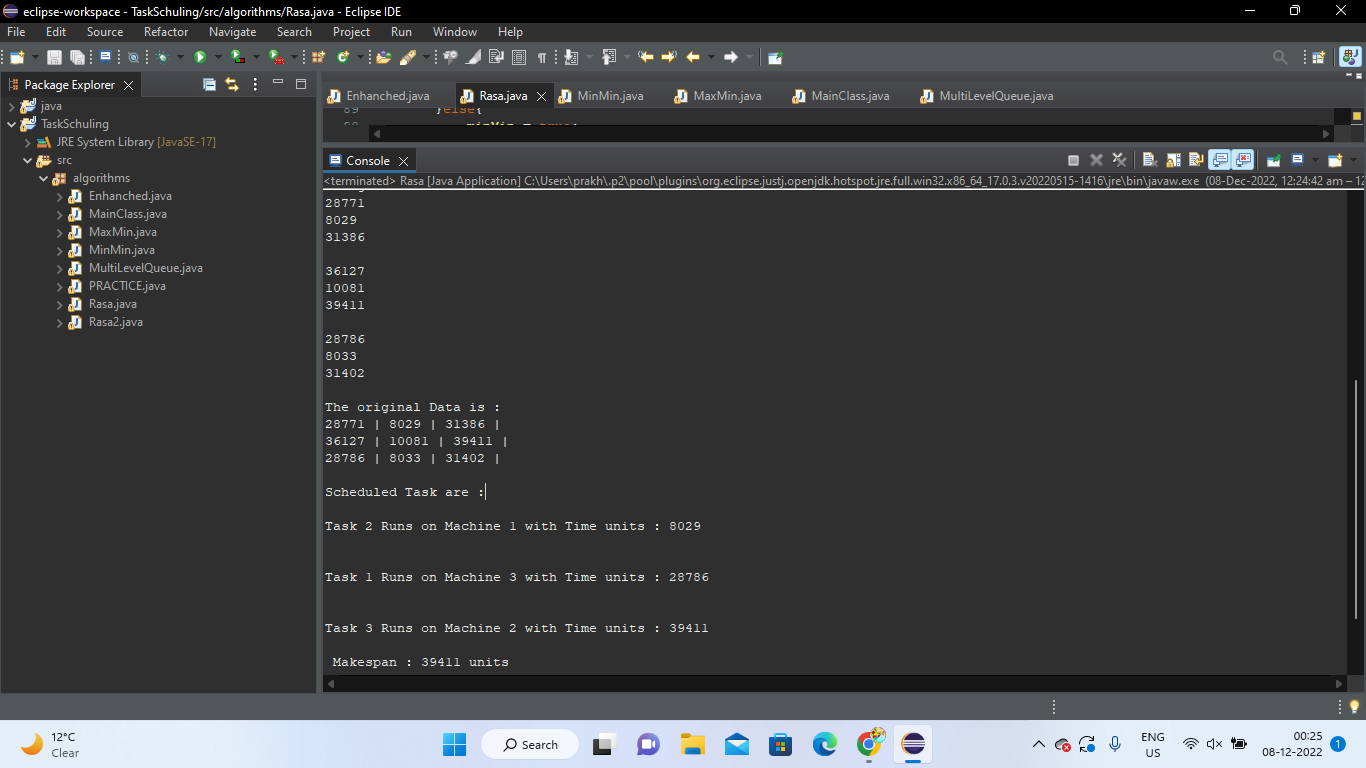


Fig 10 : Output for RASA

**ENHANCED:**

**Output:**

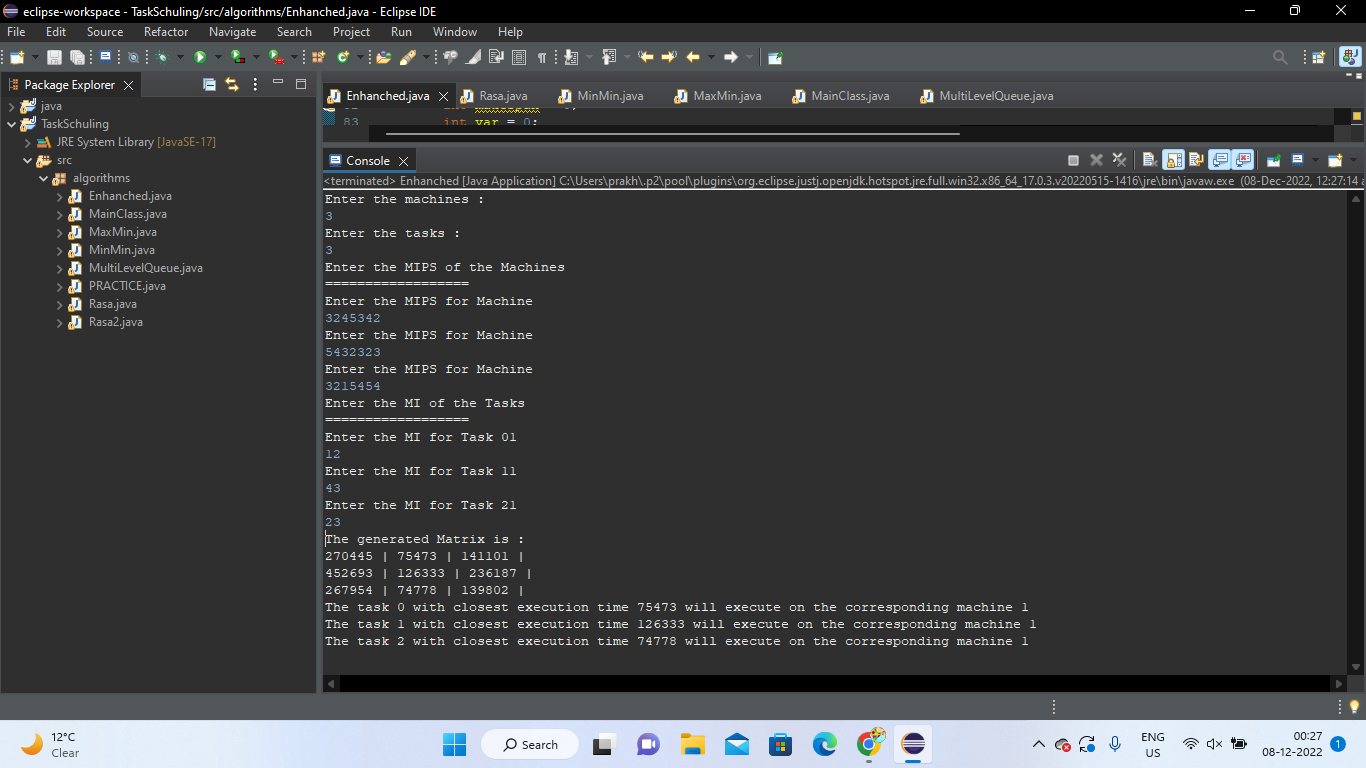


Fig 11 : Output for ENHANCED

**MULTI-LEVEL QUEUE:**

**Output:**

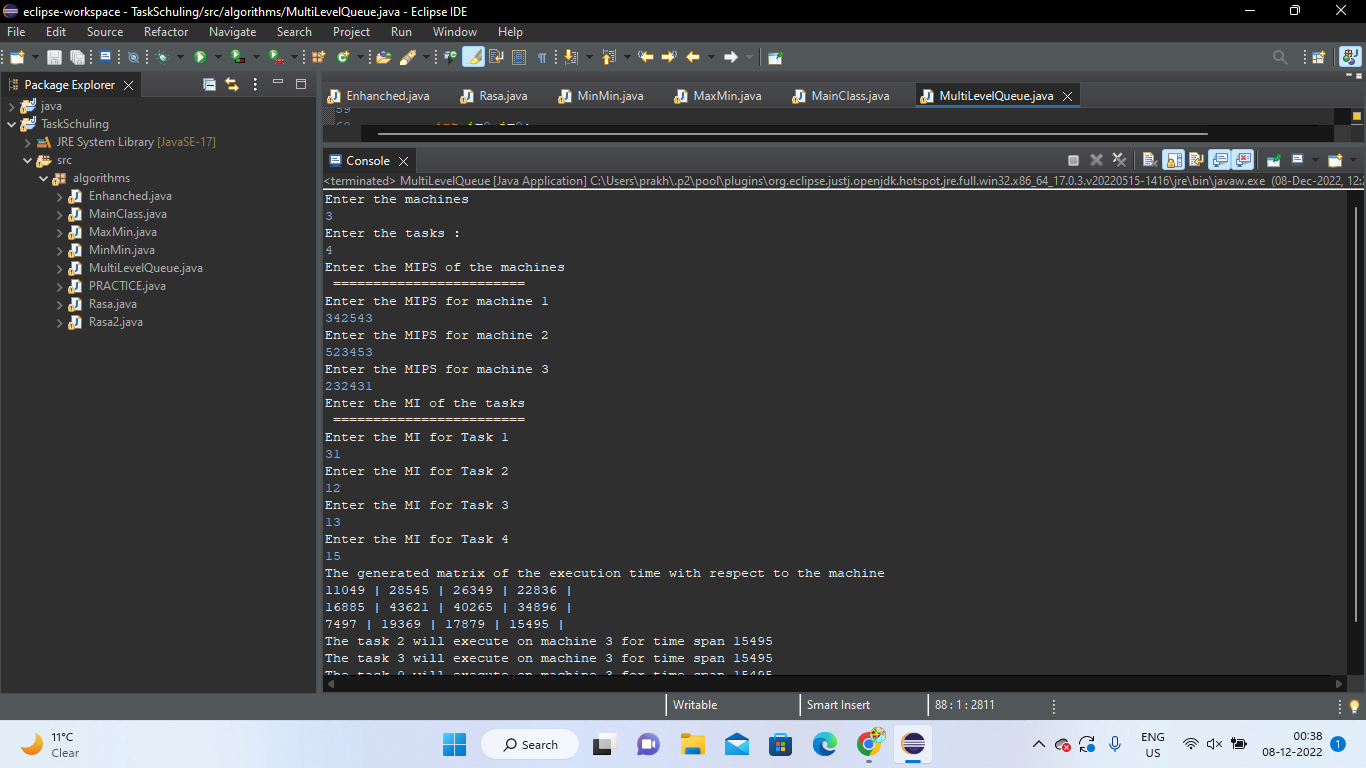


Fig 12 : Output for MULTI-LEVEL QUEUE

## 8. SWOT Analysis

Strength: The utmost strength of our project is that it helps to reduce the waiting time for the tasks, helps in achieving high system throughput. Helps in load balancing , better utilization of the CPU , memory , resources and many such system resources. In the future, we can develop a new algorithm that minimizes the execution cost and execution time and give better results than the proposed algorithm. Changes can be made in some parameters of the proposed algorithm.

Weakness: The weakness of our project is that the level of saturation is quite close, so it is very difficult to optimize the scheduling at the higher end. The interdependency of the tasks aren't into consideration.

Opportunities: It can have a wide range of valid possibilities that this project can help to reduce the waiting time in achieving high system throughput. This helps to give better results with respect to different task scheduling algorithms to minimize the cost and time of execution while meeting the deadline and budget constraints.

## Threats: An improper machine that isn’t compatible for high end tasks can harm our project as when we are trying to execute our task. The project focuses on the independent task, so it won’t be considering the sequence and work as an independent task.