

Marat Yerkebayev

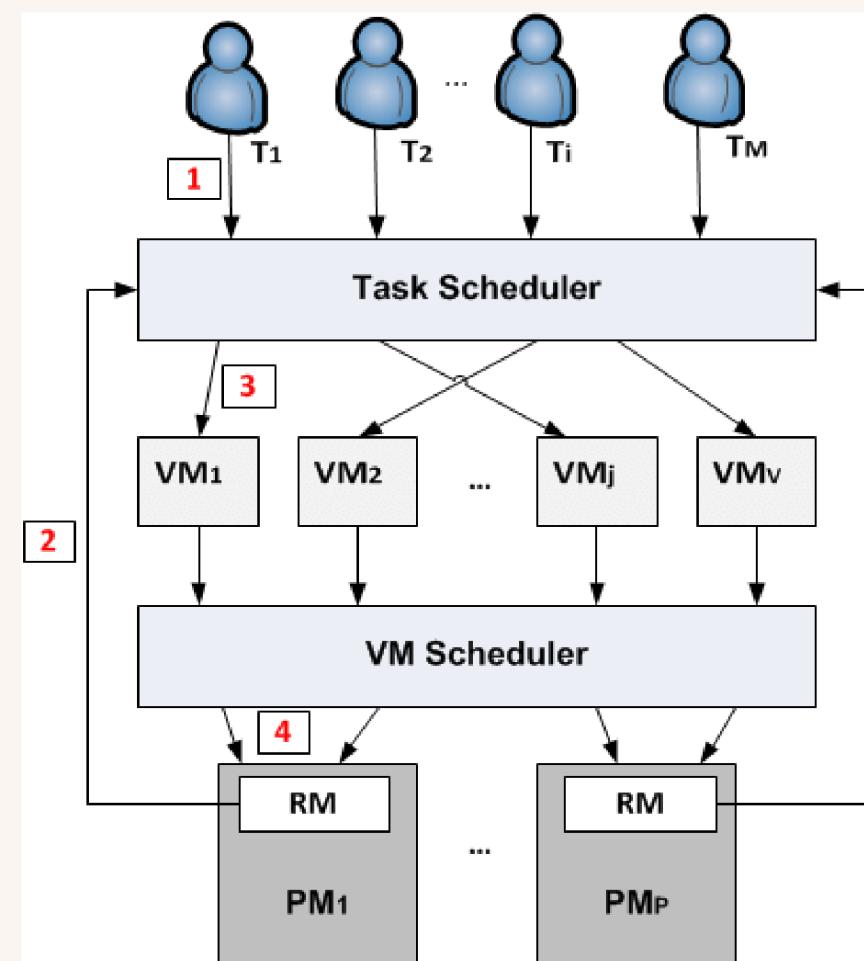
# Flower pollination algorithm

Optimizing Task Scheduling in Cloud  
Computing with Flower Pollination  
Algorithm

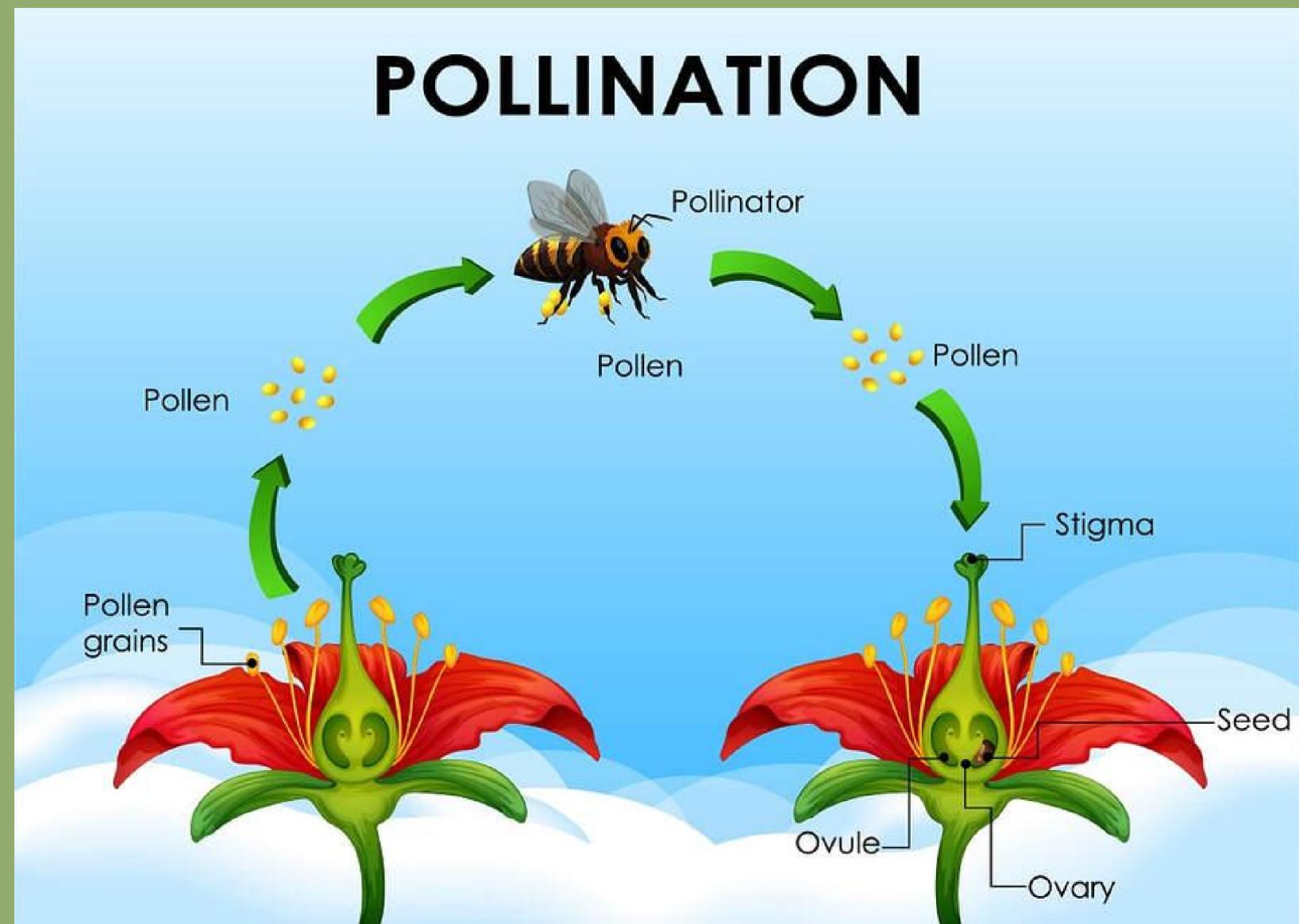


# Introduction

- Task scheduling: Efficient allocation of computing resources to tasks in cloud environments.
- Importance: Efficient task scheduling enhances resource utilization and improves performance in cloud computing environments.



# What is Flower Pollination?



01.

Flower pollination is a crucial biological process essential for the reproduction of flowering plants.

02.

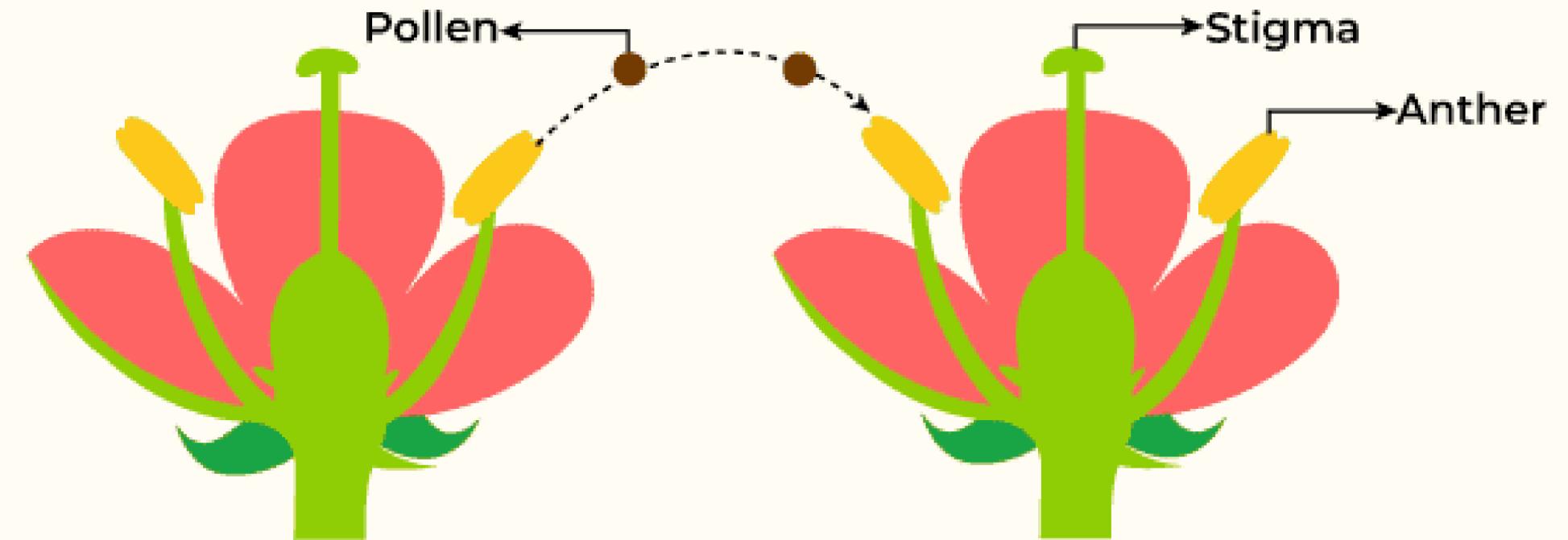
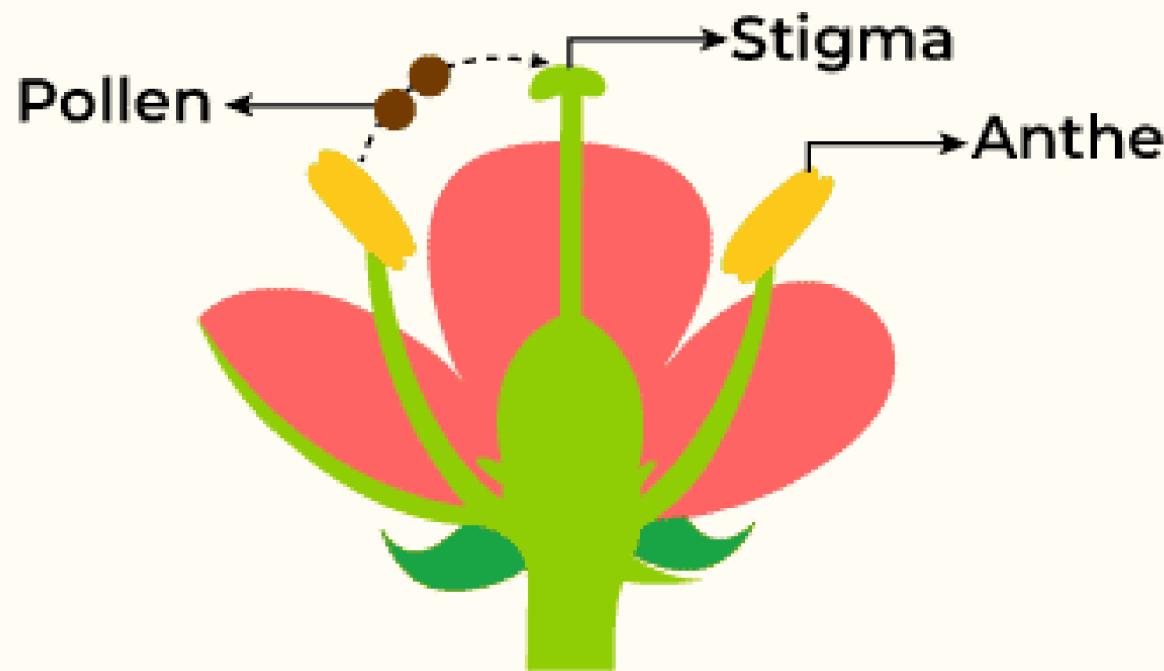
It involves the transfer of pollen grains from the male reproductive organs of a flower to the female reproductive organs of the same or another flower.

03.

This transfer can occur through various agents such as wind, water, animals (like bees, butterflies, birds), or even self-pollination within the same flower.

# Two types of Pollination

## Self-Pollination Vs Cross-Pollination



# How FPA adapted in Task scheduling?

## 1. Problem Representation

**Flower**

is a potential solution  
to the optimization  
problem

**Fitness**

of a flower refers to  
how well the  
corresponding  
solution performs.

For example, we have 10  
cloudlets and 2 Vms and  
one Flower is like

1, 3, 6, 8, 9 -> cloudlets for  
1st Vm  
2, 4, 5, 7, 10 -> cloudlets for  
2nd Vm

# How FPA adapted in Task scheduling?

## 2. Initialization

### Create Population

Generate the initial population of solutions. It randomly assigns cloudlets to Vms. It prepares variables to execute algorithm.

### Number solutions

Number of solutions that we check

### Best Flower

We aim to find best flower inside population and first of all we find it by comparing their fitness



# How FPA adapted in Task scheduling?

## 3. Main Part

1. Iterate  
each flower



2. Random number  
- Math.random()

3. Local  
pollination  
or Global  
pollination



4. Compare  
with BestFlower



vs

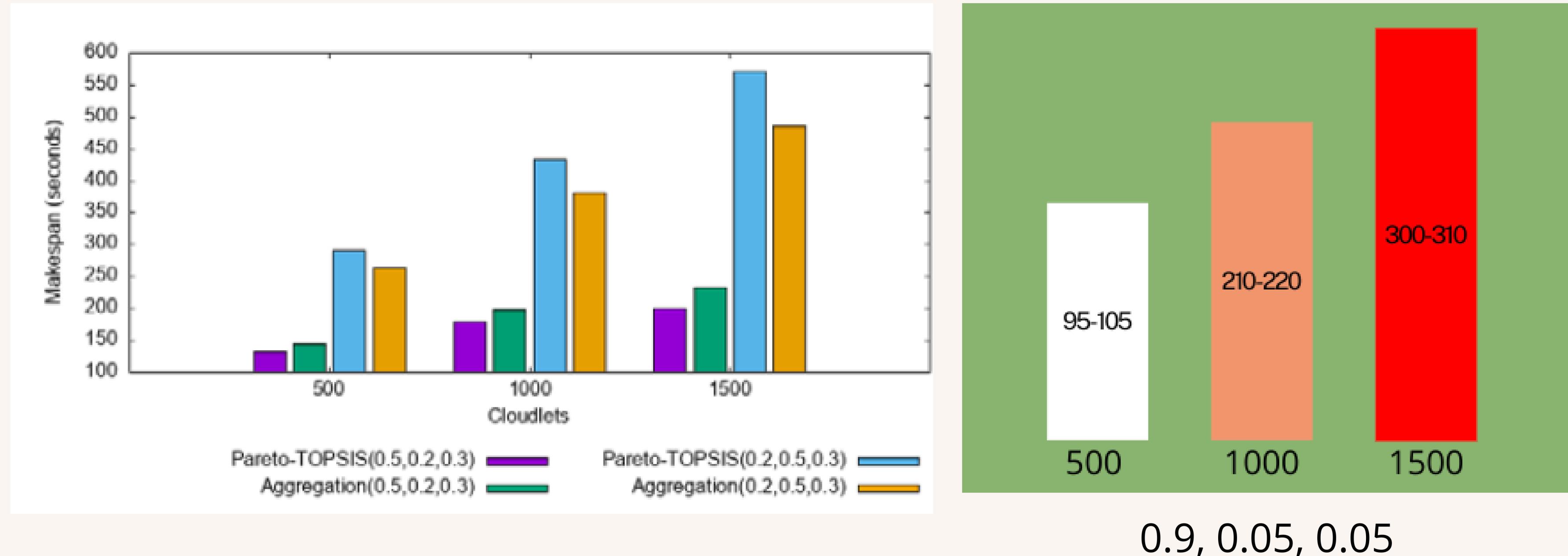


# How FPA adapted in Task scheduling?



4. Return  
Best Flower

# Statistics



Thank  
you very  
much!

