DDSS: Dynamic Dedicated Servers Scheduling for Multi Priority Level Classes in Cloud Computing

Husnu Saner Narman

Md. Shohrab Hossain

Mohammed Atiquzzaman

School of Computer Science

University of Oklahoma, USA.

atiq@ou.edu

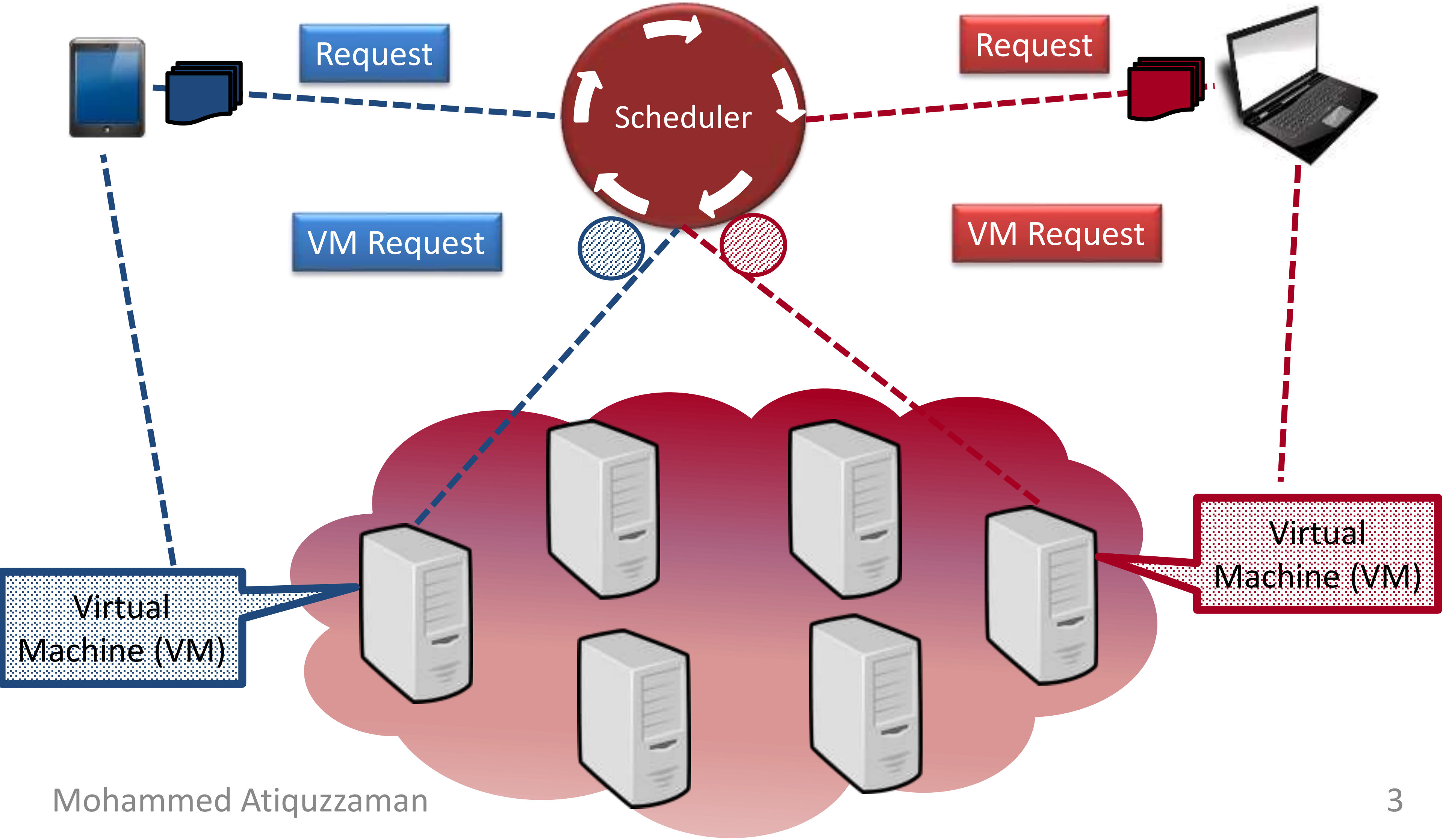
www.cs.ou.edu/~atiq

June 2014

Presentation Outlines

- Cloud Computing
- Dedicated Servers Scheduling (DSS)
- Proposed Dynamic Dedicated Scheduling (DDSS)
- Analytical Models
- Results
- Conclusion

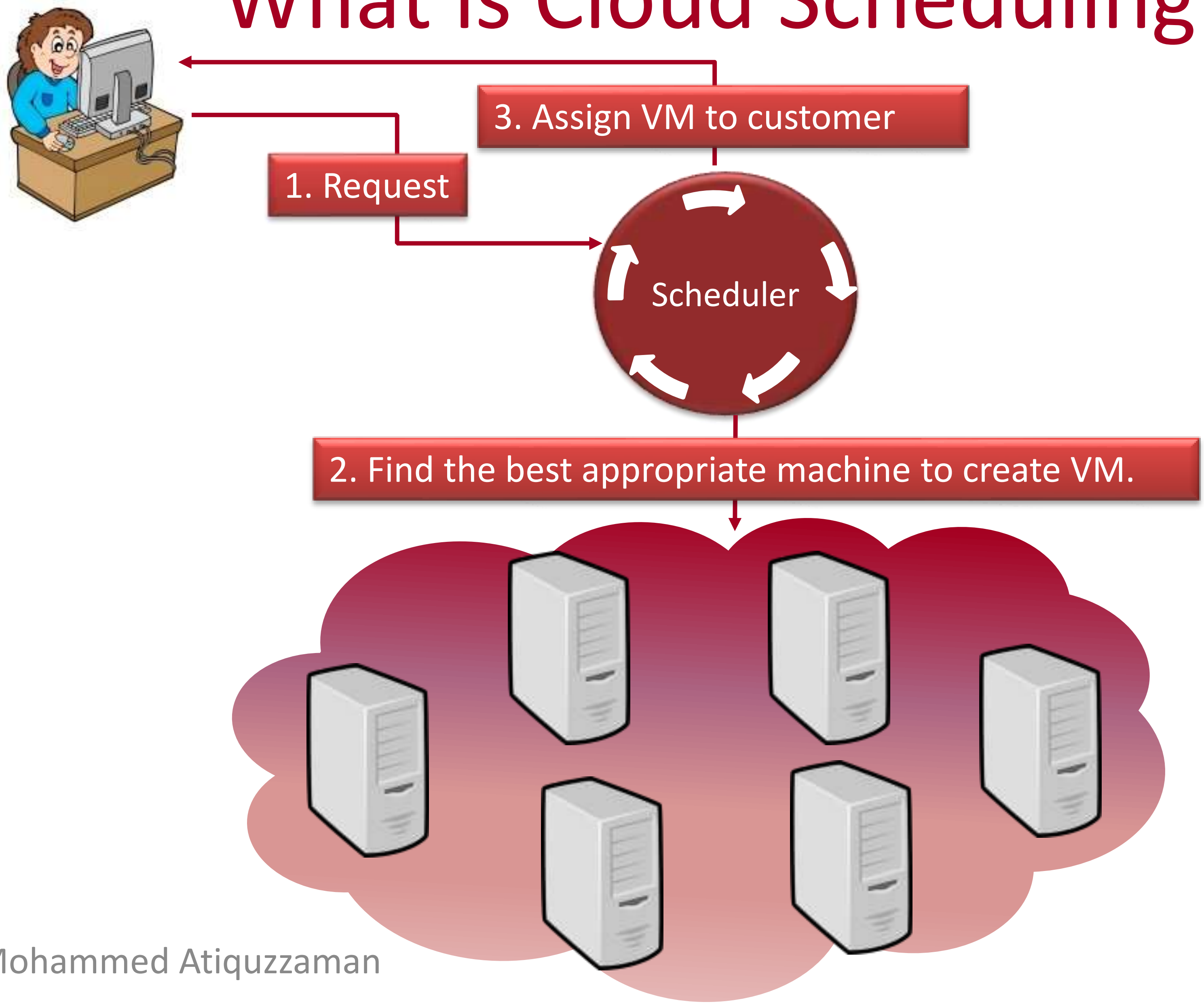
What is Cloud Computing



Why Cloud Computing

- Simplicity
  - No need to set up software/hardware
- Flexibility
  - Easily extending memory/CPU capacity
- Maintenance
  - IT services
- Time and energy
  - No consume time or extra effort to have desired environment
- Pay as you go
  - Not pay for unused hardware or software

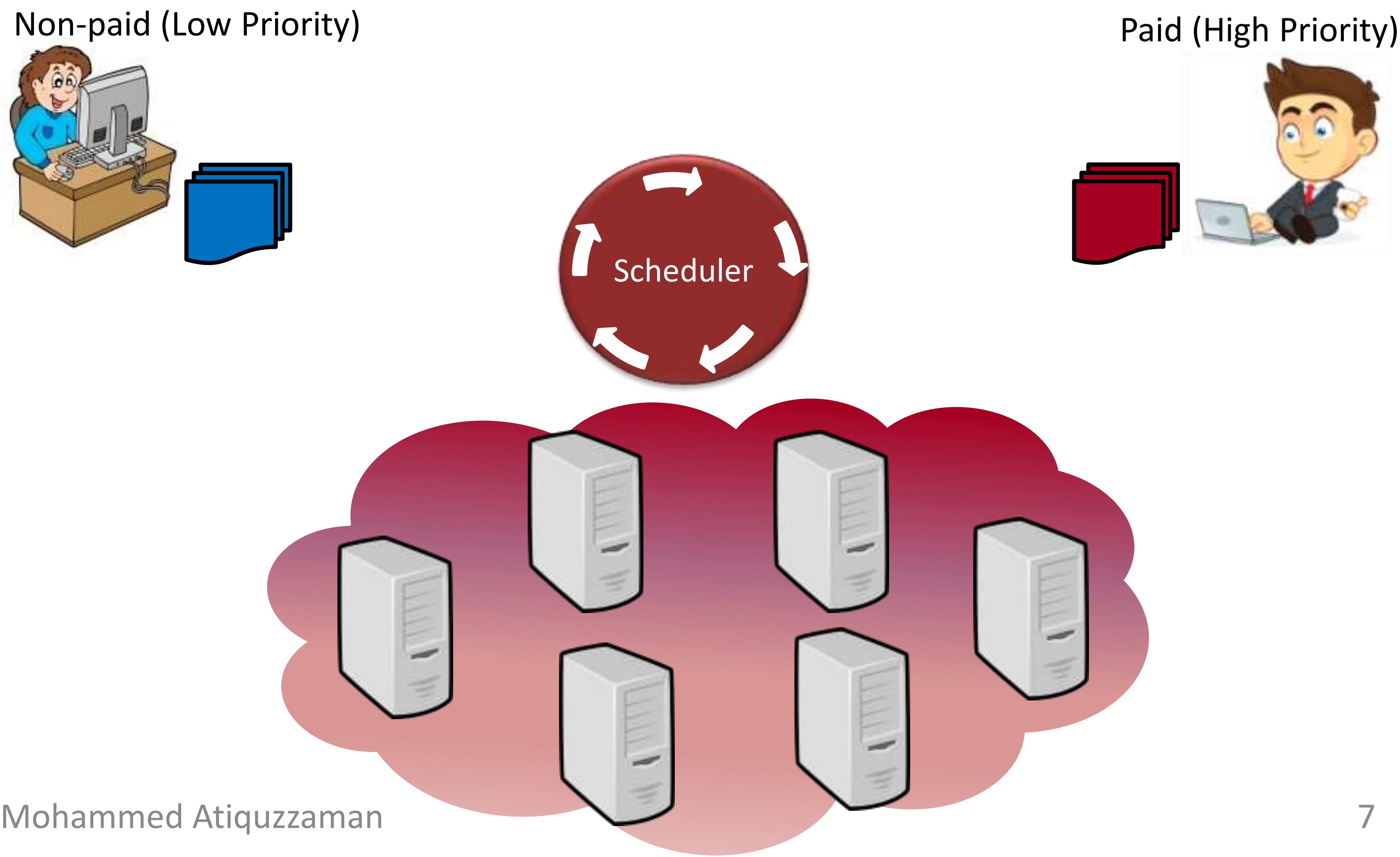
What is Cloud Scheduling



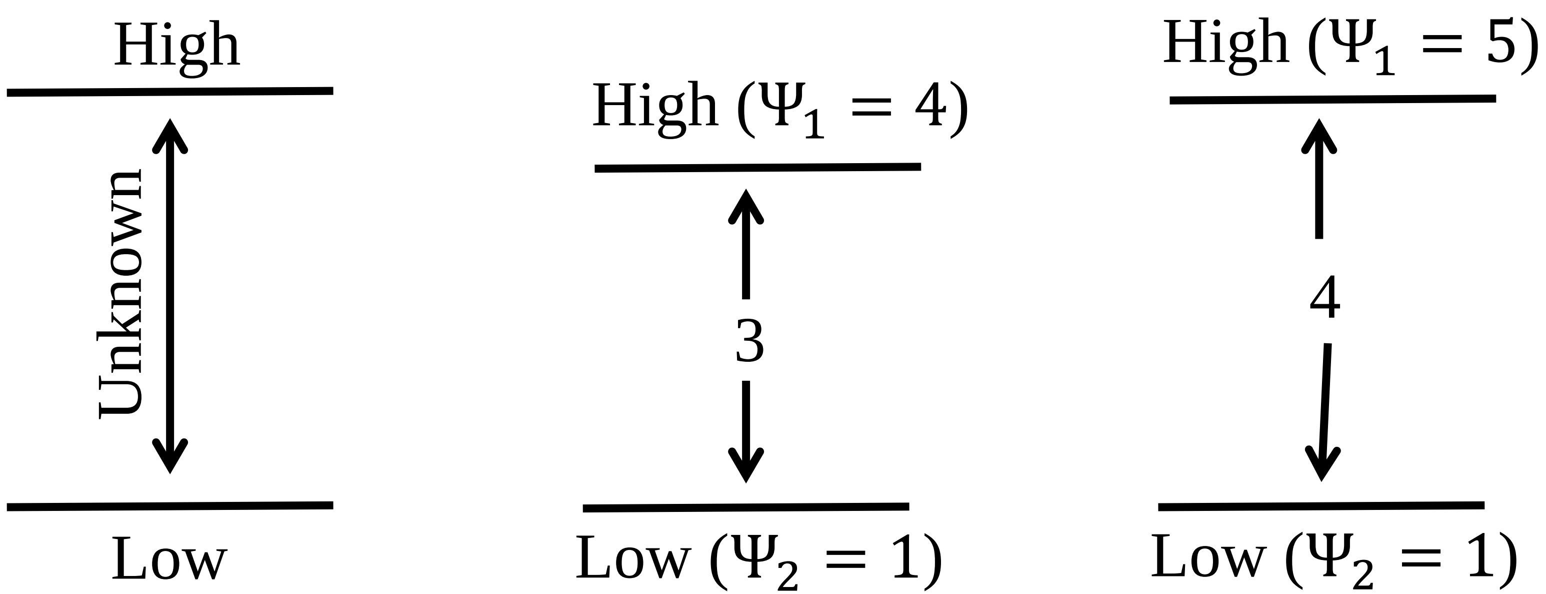
Customer Type

- Different customers classes?
  - Paid and non-paid
- Customer requirements
  - Desired Platform based on Service Level Agreement
- How to satisfy different customer classes?
  - Reserve servers for each customer types
    - Dedicated Servers Scheduling
  - Priority
    - High or Low

Customer Priority



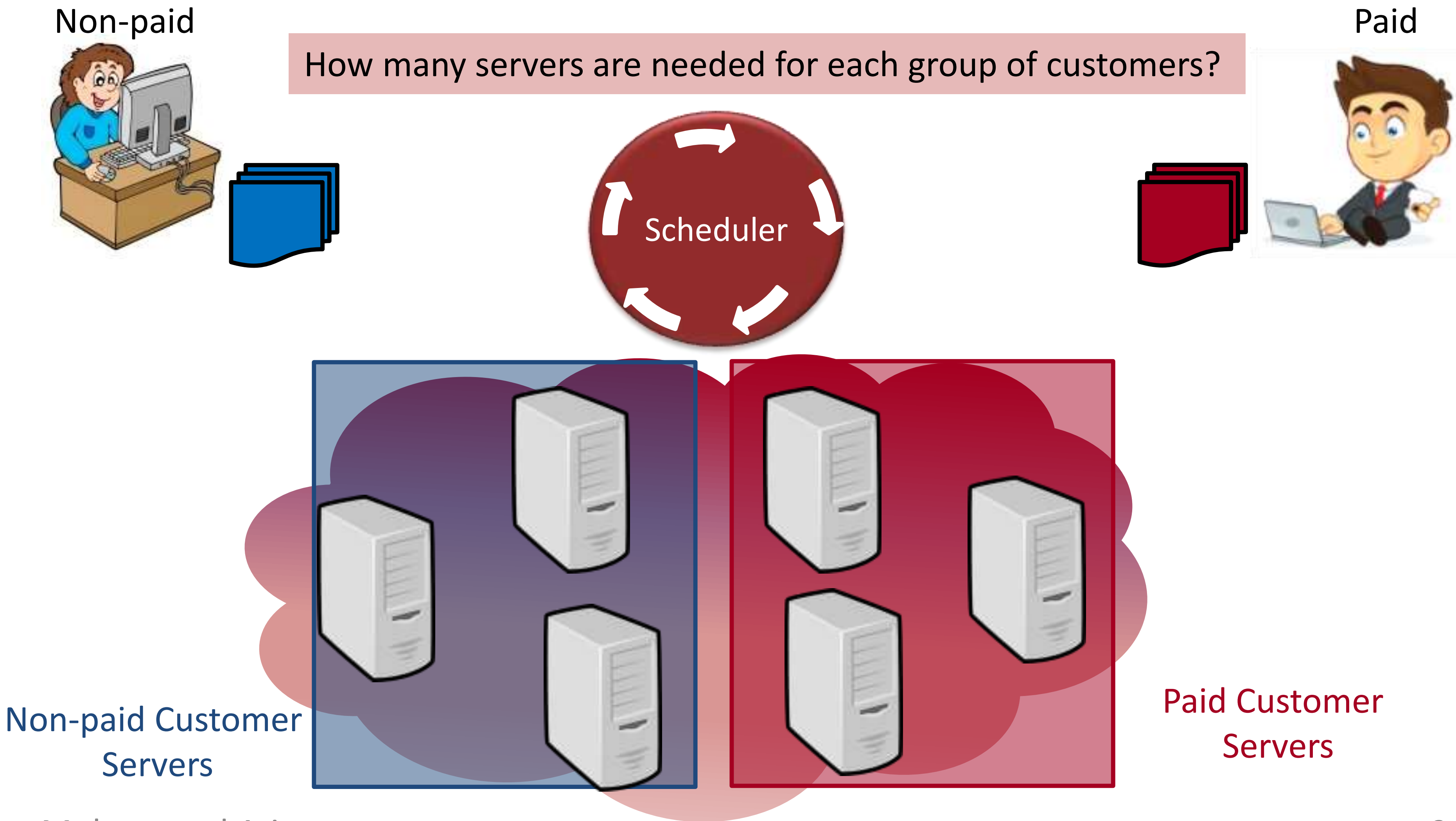
Priority Level



Without priority level in queuing theory

With priority level in cloud computing

Reserved Servers

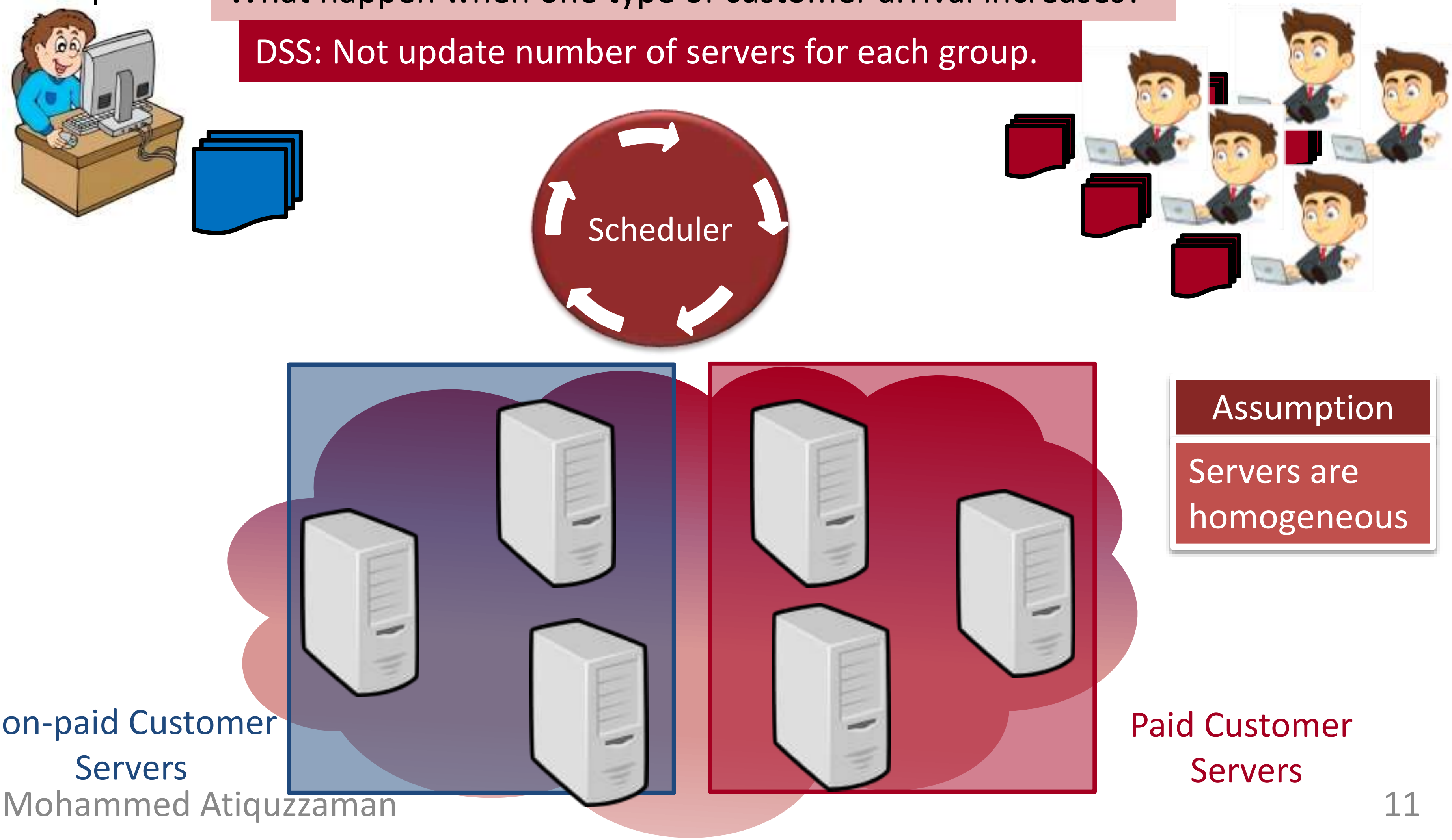


Dedicated Server Scheduling (DSS)

Dedicated Servers Scheduling

What happen when one type of customer arrival increases?

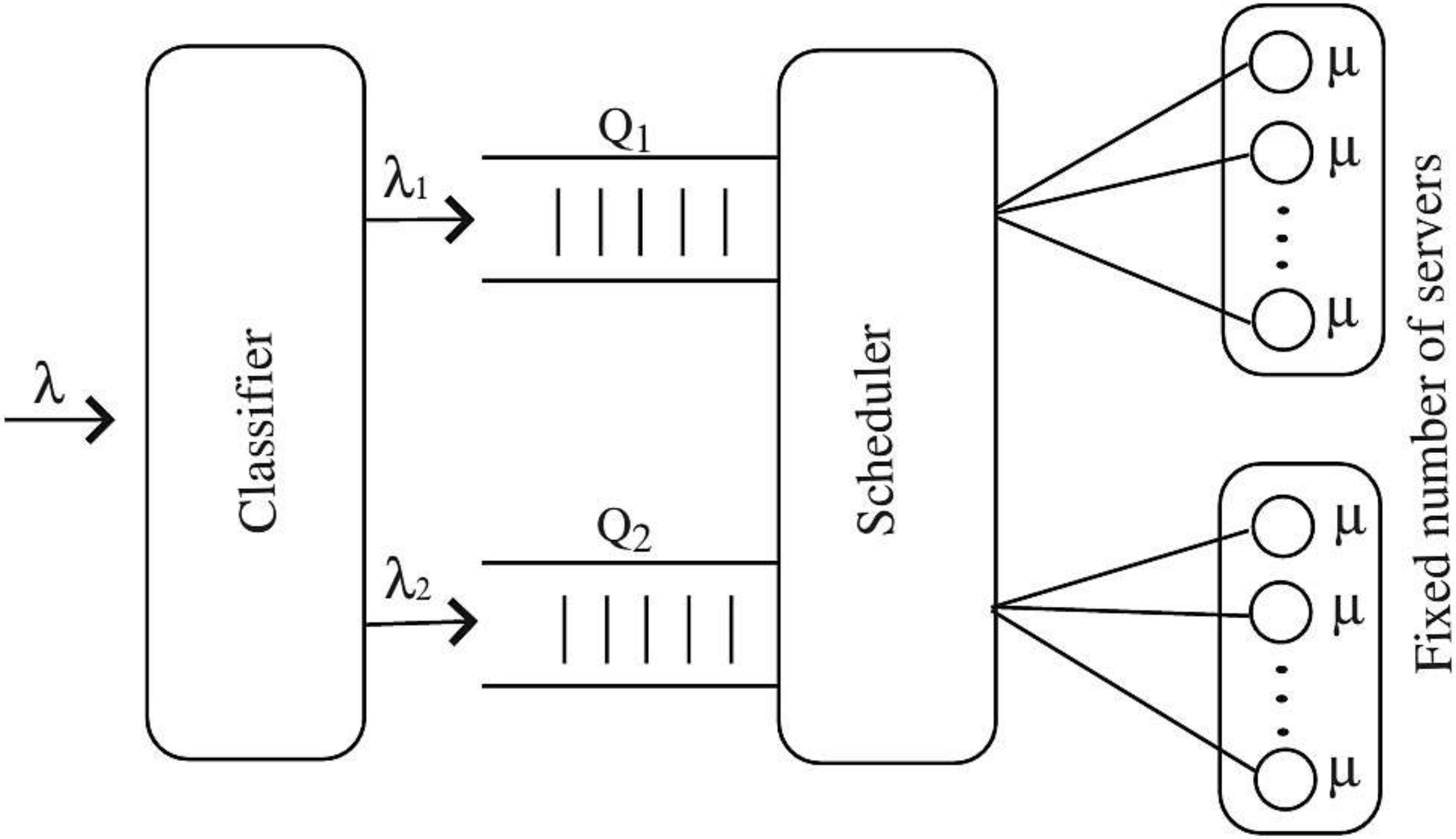
DSS: Not update number of servers for each group.



Assumption

Servers are homogeneous

Dedicated Servers Scheduling



Problems with DSS

- Not dynamically update number of servers for each group
  - If arrival rate changes
  - If priority level changes

Dynamic Dedicated Server Scheduling (DDSS)

Objective

- Improve performance of cloud systems
  - Allowing servers to be dynamically allocated to customer classes based on:
    - Priority level.
    - Arrival rate.

Contribution

- Propose Dynamic Dedicated Servers Scheduling
- Develop Analytical Model to evaluate performance
  - Average occupancy,
  - Drop rate
  - Average delay
  - Throughput
- Comparing performance of
  - Dynamic Dedicated Servers Scheduling
  - Dedicated Servers Scheduling

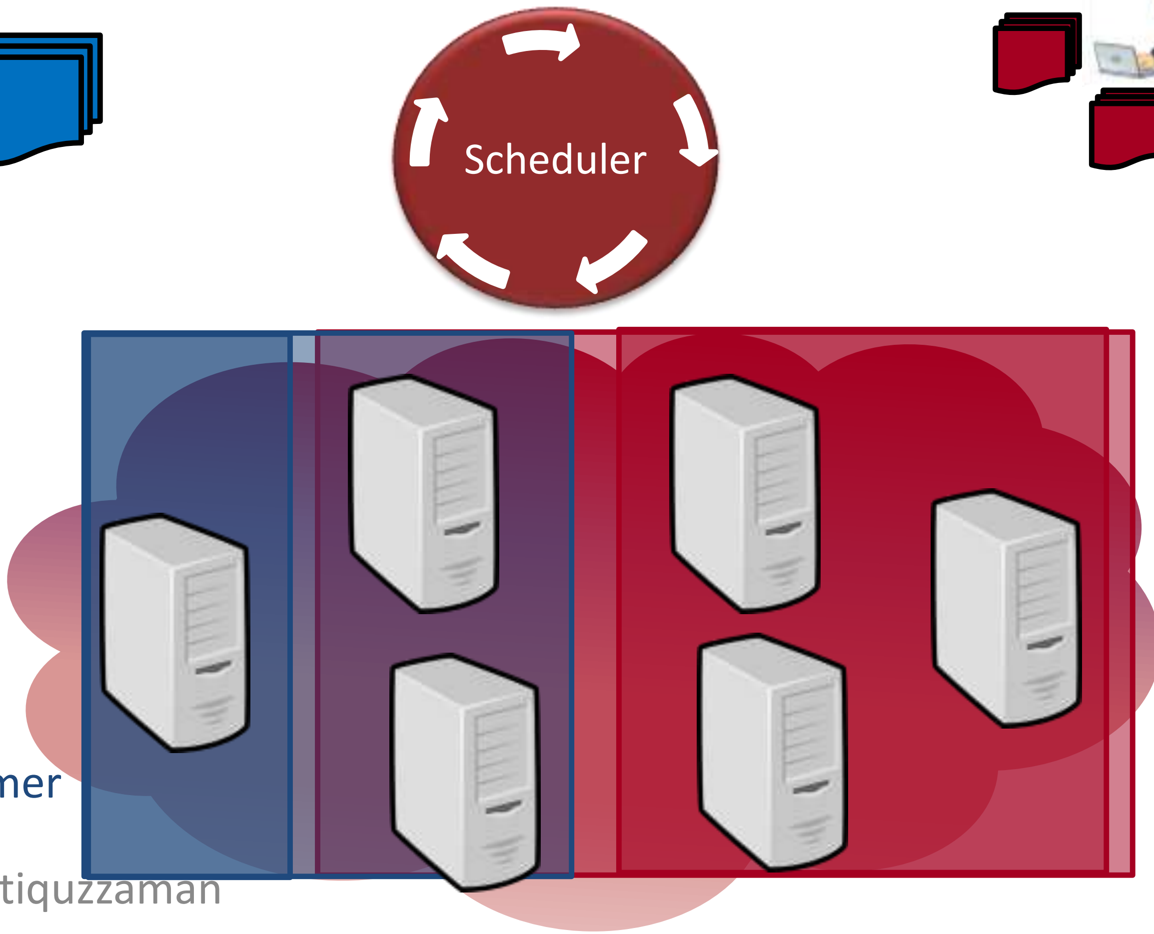
Dynamic Dedicated Servers Scheduling

Non-paid

What happen when one type of customer arrival increases?

DDSS: Updating number of servers for each group.

Paid



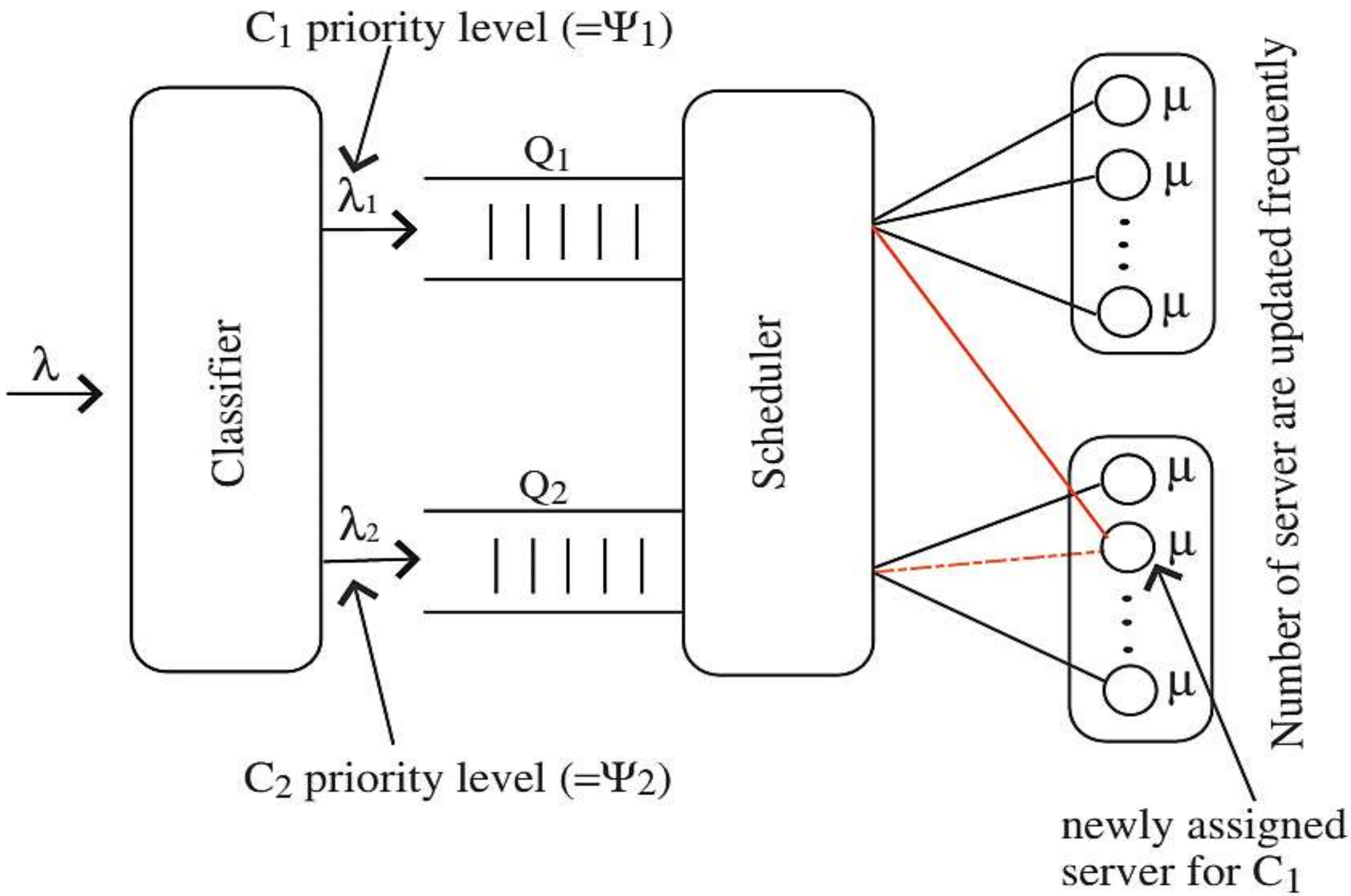
Assumption

Servers are homogeneous

Non-paid Customer Servers

Paid Costumer Servers

Dynamic Dedicated Servers Scheduling



Dynamic Approach

l: Total number of servers

Ψ_1: Priority level of C_1 customers

λ_1: Arrival rate of C_1 customers

m: Number of servers assigned for C_1 customers

$$m = \left\lfloor \frac{l\Psi_1\lambda_1}{\Psi_1\lambda_1 + \Psi_2\lambda_2} \right\rfloor$$

λ_2: Arrival rate of C_2 customers

$$k = l - m$$

Ψ_2: Priority level of C_2 customers

k: Number of servers assigned for C_2 customers

$$m_1 = \left\lfloor \frac{l\Psi_1\lambda_1}{\Psi_1\lambda_1 + \Psi_2\lambda_2 + \ldots + \Psi_r\lambda_r} \right\rfloor$$

This formula can be used for r different number customer types.

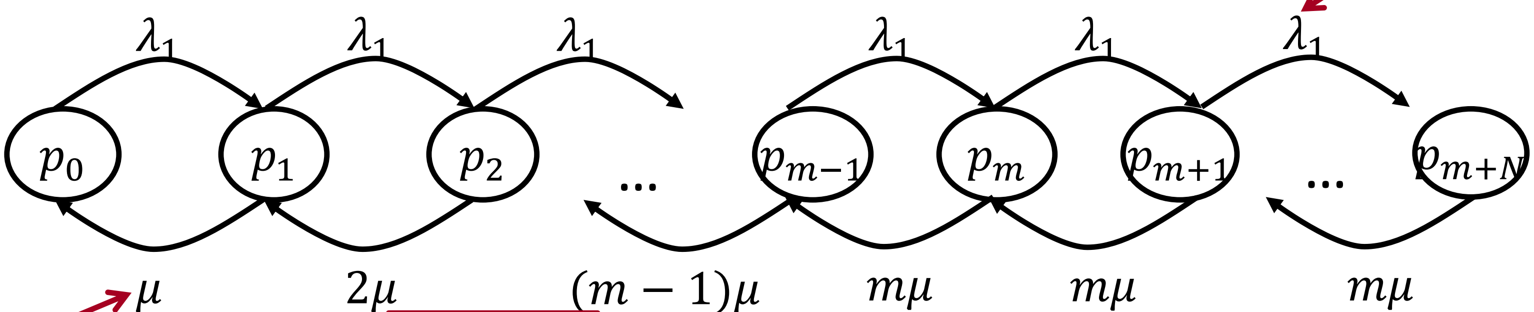
Modeling Assumptions

- System is under heavy traffic flows.
- Arrivals follow Poisson distribution, and service times for customers are exponentially distributed.
- Type of queue discipline used in the analysis is FIFO.
- Service rate of all servers are equal.

Analytical Model

- Only C_1 customers performance metric developed.
- Markov Chain Model :

λ_1: Arrival rate of C_1 customers



μ: Service rate of C_1 customers

$$\rho = \frac{\lambda_1}{\mu}$$

m: number of servers for C_1 customers

$$p_i = \begin{cases} p_0 \frac{\rho^i}{i!} & , 1 \leq i \leq m \\ p_0 \frac{m^m}{m!} \rho_2^i & , m < i \leq m + N \end{cases}$$

p_i: Probability of i C_1 customer in the system

N: Queue size

$$\rho_2 = \frac{\lambda_1}{m\mu}$$

Analytic Model (Contd.)

- Drop Probability : $D = p_0 \frac{m^m}{m!} \rho_2^{m+N}$
- Throughput: $\gamma = \lambda_1(1 - D)$
- Occupancy: $n = \begin{cases} p_0 \rho_2 \frac{m^m}{m!} \left(\frac{1-(N+1)\rho_2^N + N\rho_2^{N+1}}{(1-\rho_2)^2} \right) & \rho_2 \neq 1 \\ p_0 \frac{m^m}{m!} \left(\frac{N(N+1)}{2} \right) & \rho_2 = 1 \end{cases}$
- Delay: $\delta = \frac{n}{\gamma}$

Drop probability

Rate of dropped customers from the systems buffer.

Throughput

Number of customers served in the systems.

in the systems buffer.

Delay

Average waiting time of a customer in the systems buffer.

Results

- We have used discrete event simulation to implement by following M/M/N/N and proposed scheduling.
- Each queue holds 30 customers.
- We ran simulation with 20000 customers for each arrival rate.

Traffic Arrival Rates

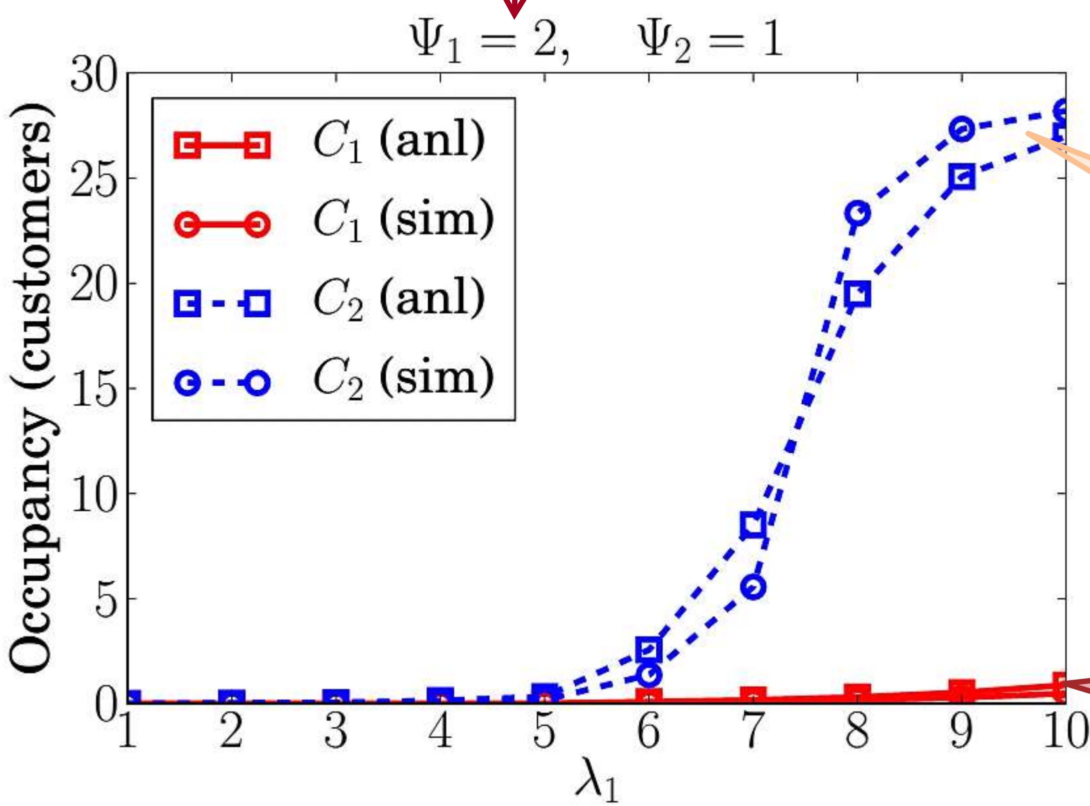
- Simulations were with increased arrival rates of all types of customers to observe the impact of heavy traffic on the system.

- Customer arrival rates at different trials:

$$\lambda_1 = \{1, 2, 3, 4, 5,6, 7,8,9,10\},$$
$$\lambda_2 = \{1,2,3,4,5,12,14,16,18,20\}$$
$$\Psi_1 = \{1.5, 2, 5\}, \; \Psi_2 = \{1\}$$
$$\mu = 5, \quad l = 6$$

Validation of Analytic Formulas: Occupancy

Ψ_1 - Priority level of C_1 customers
Ψ_2 - Priority level of C_2 customers



Occupancy
Number of customers in the systems buffer.

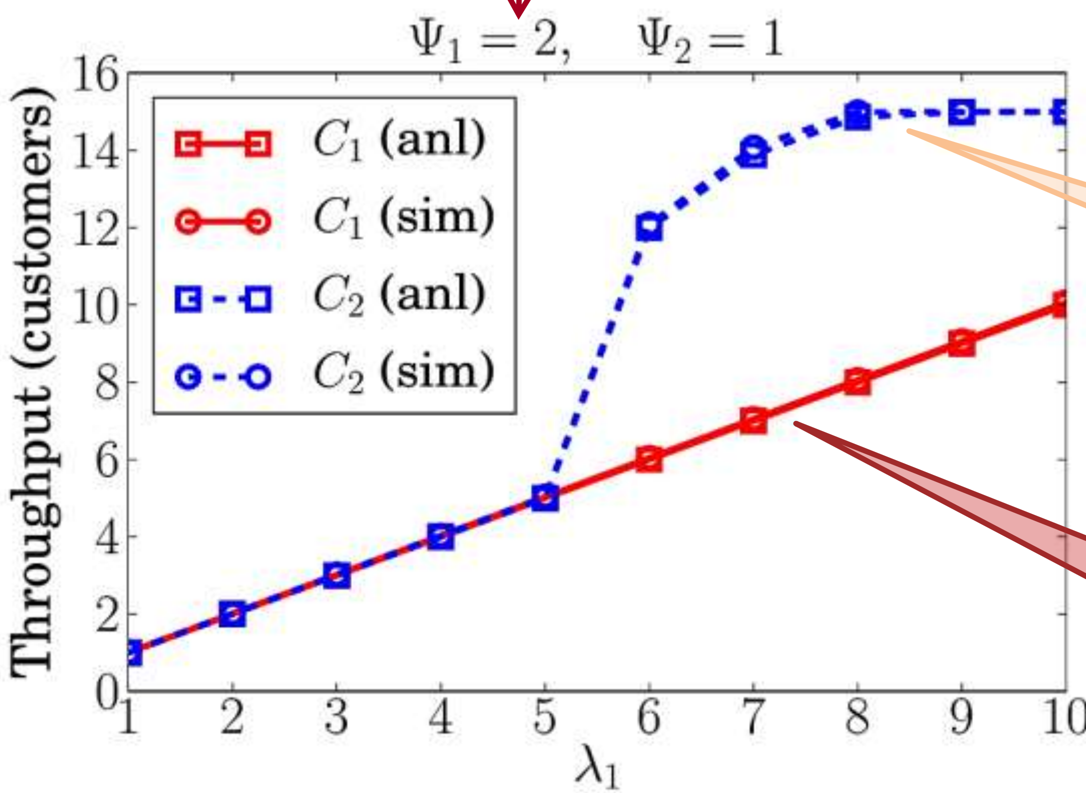
Occupancy of C_2 for analytical and simulation matches.

Occupancy of C_1 for analytical and simulation closely matches.

Occupancy model matches with simulation.

Validation of Analytic Formulas: Throughput

Ψ_1 - Priority level of C_1 customers
Ψ_2 - Priority level of C_2 customers



Throughput
Number of customers are served in the systems.

Throughput of C_2 for analytical and simulation closely matches.

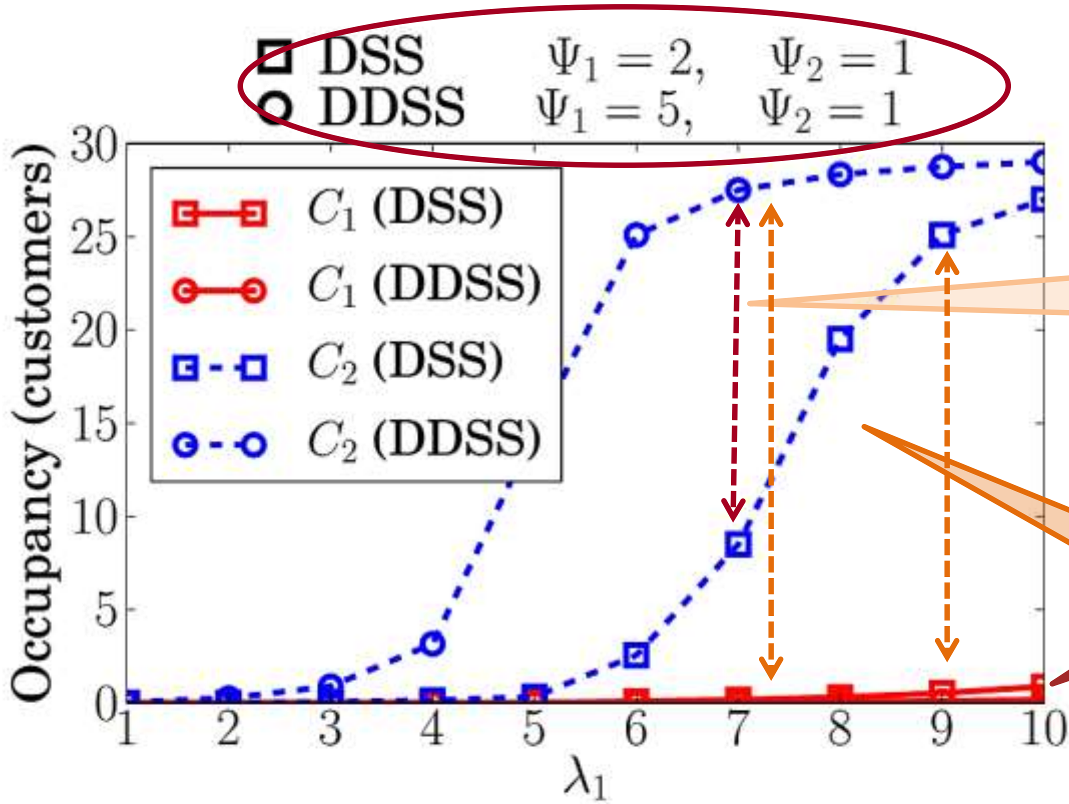
Throughput of C_1 for analytical and simulation closely matches.

Throughput model matches with simulation.

DDSS can arrange dynamically based on priority and arrival rate.

DDSS vs DSS

Assumption: DSS can arrange dynamically based on arrival rate.



Objective

We would like to see effects of priority level $\Psi_1 = 5$ on occupancy.

Occupancy of C_2 for DDSS is higher than occupancy of C_2 for DSS.

Occupancy of C_1 for DDSS and DSS are same.

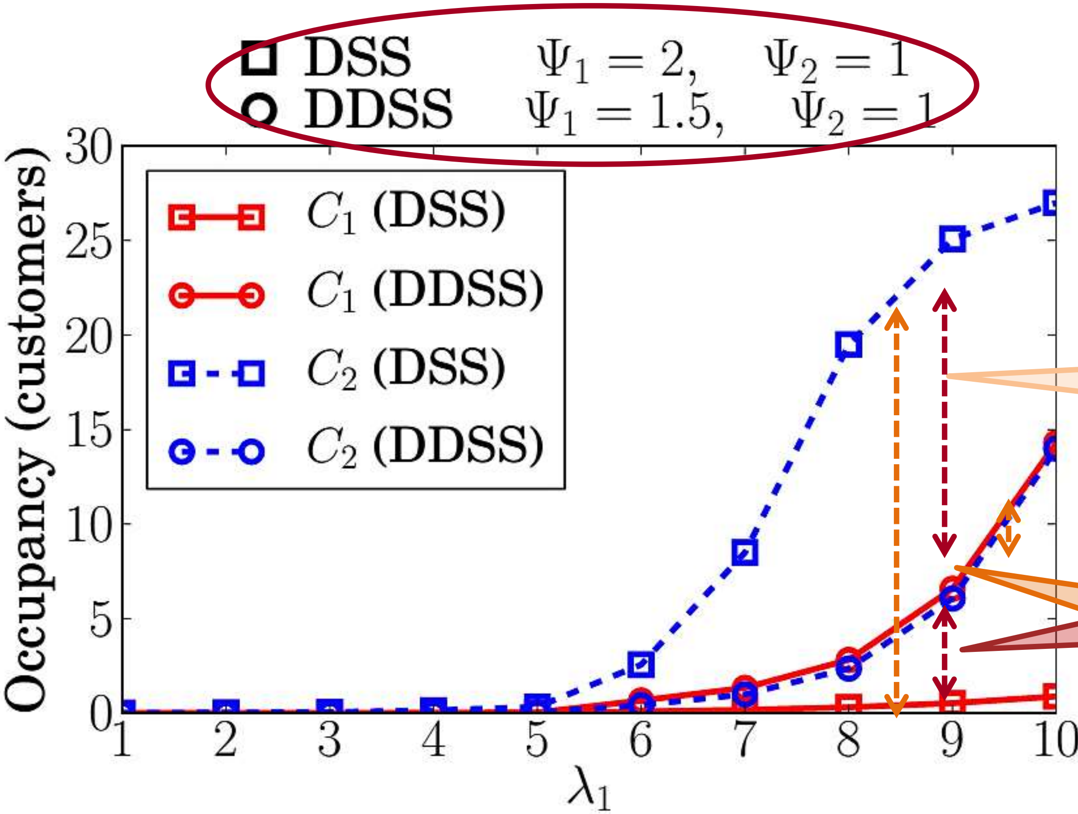
The gap between C_1 and C_2 for DDSS is higher than the gap between C_1 and C_2 for DSS.

DSS shows better occupancy than DDSS for these priority levels.

DDSS can arrange dynamically based on priority and arrival rate.

DDSS vs DSS

Assumption: DSS can arrange dynamically based on arrival rate.



Objective

We would like to see effects of priority level $\Psi_1 = 1.5$ on occupancy.

Occupancy of C_2 for DDSS is lower than occupancy of C_2 for DSS.

Occupancy of C_1 for DDSS is higher than occupancy of C_1 for DSS.

The gap between C_1 and C_2 for DDSS is lower than the gap between C_1 and C_2 for DSS.

DDSS shows better occupancy than DSS for these priority levels.

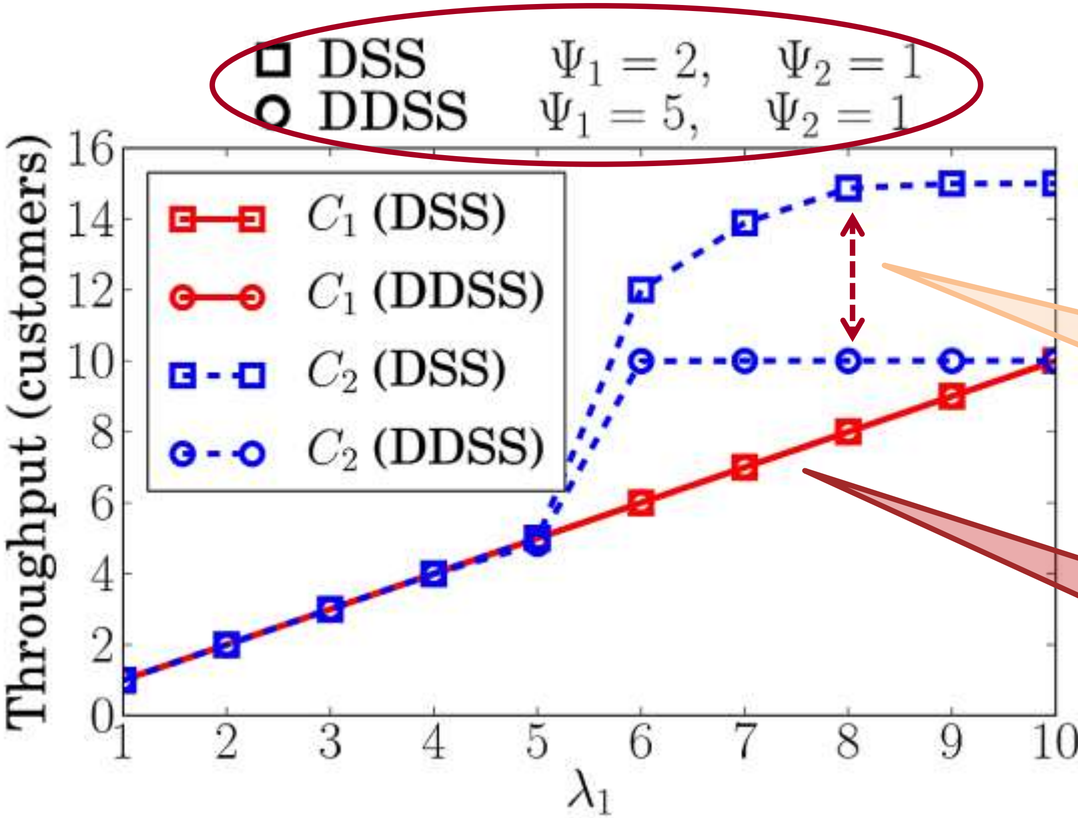
DDSS vs DSS

DDSS can arrange dynamically based on priority and arrival rate.

Assumption: DSS can arrange dynamically based on arrival rate.

Objective

We would like to see effects of priority level, $\Psi_1 = 5$ on throughput.



Throughput of C_2 for DDSS is lower than throughput of C_2 for DSS.

Throughput of C_1 for DDSS and DSS are same.

DSS shows better throughput than DDSS for these priority levels.

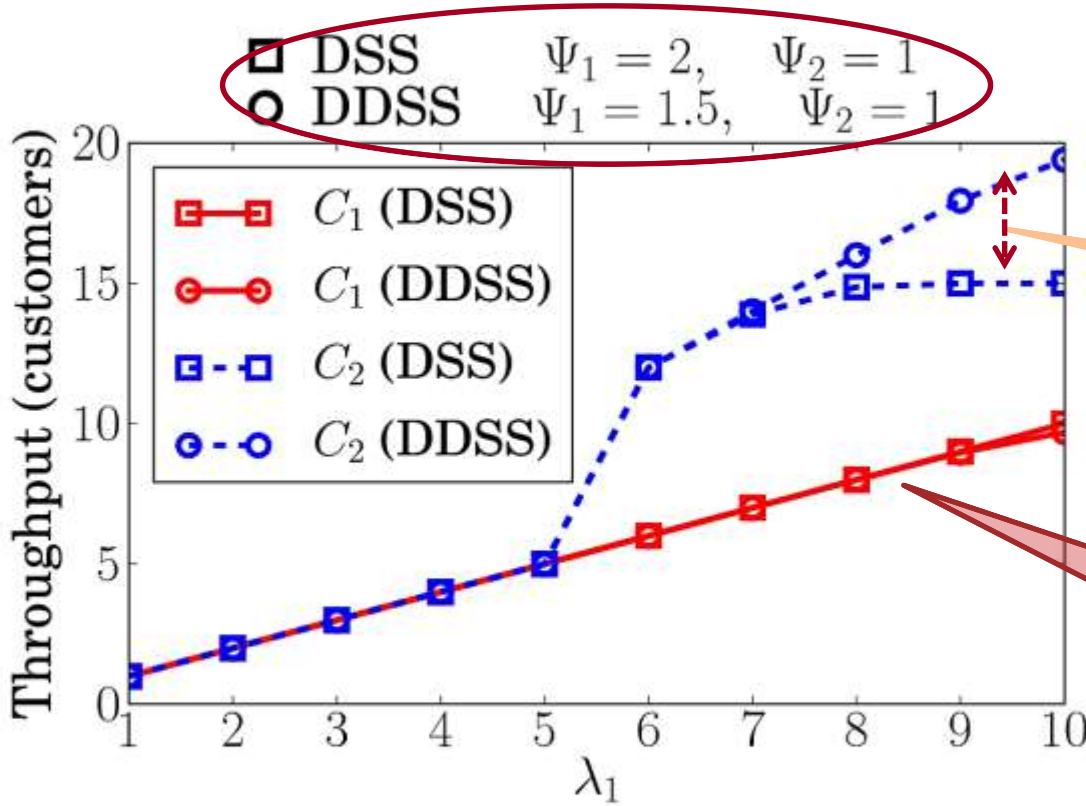
DDSS vs DSS

DDSS can arrange dynamically based on priority and arrival rate.

Assumption: DSS can arrange dynamically based on arrival rate.

Objective

We would like to see effects of priority level $\Psi_1 = 1.5$ on throughput.



Throughput of C_2 for DDSS is higher than throughput of C_2 for DSS.

Throughput of C_1 for DDSS and DSS are same.

DDSS shows better throughput than DSS for these priority levels.

Summary of Results

- The class priority levels do not affect the performance of DSS and DDSS architectures under low traffic.
- Under heavy traffic, the class priority levels have significantly effects on performances of DDSS architecture.
- The system can become more efficient based on priority levels in DDSS.
- DDSS shows better performance than DSS although assuming DSS can dynamically update servers.

Conclusion

- We have proposed a novel scheduling algorithm for cloud computing considering priority and arrival rate.
- Performance metrics of the proposed cloud computing system are presented through different cases.
- DDSS and DSS are compared under different priority levels.
- Proposed scheduling algorithm can help Cloud Computing Platforms have higher throughput and be more balanced.

Thank You

http://cs.ou.edu/~atiq

atiq@ou.edu