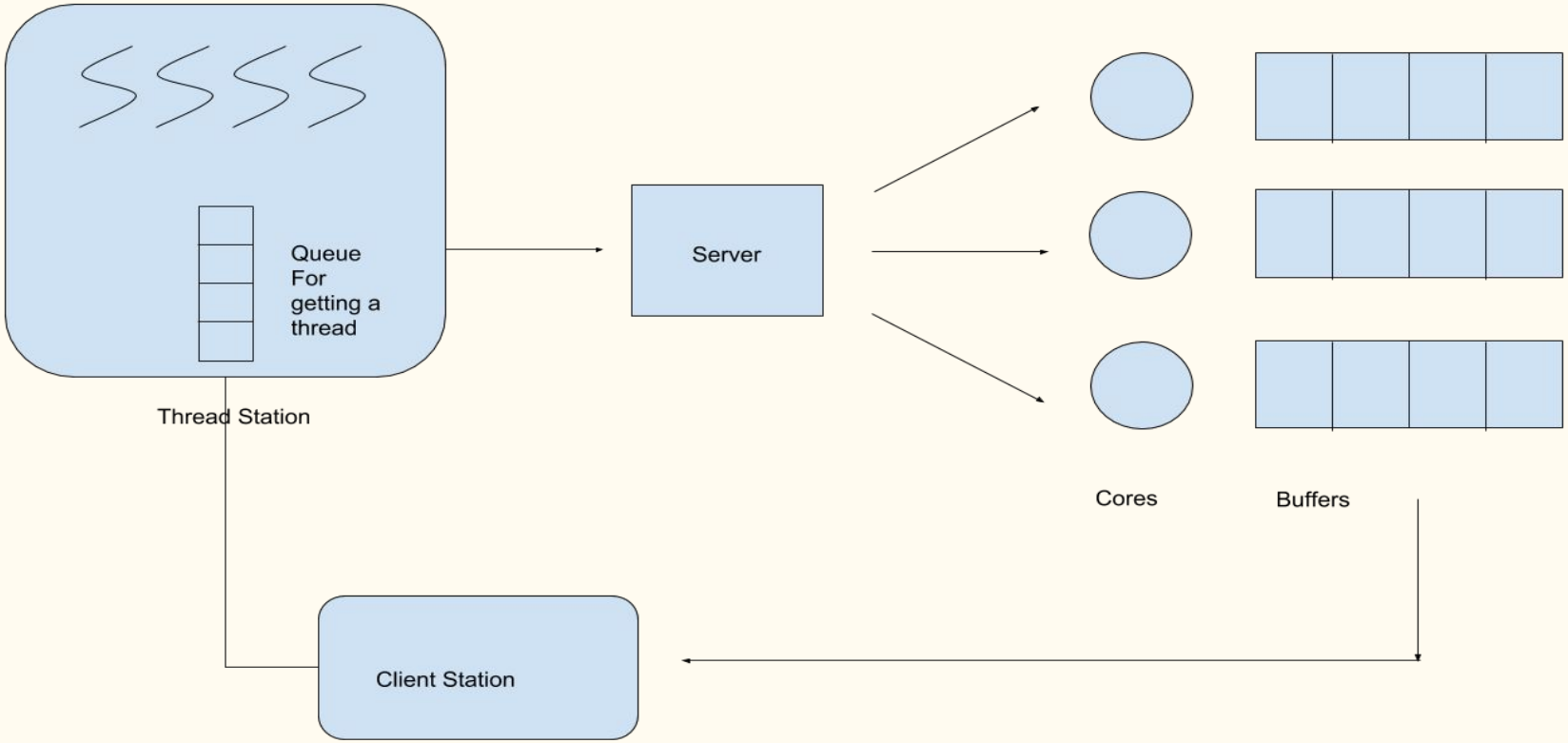
CS 681

Discrete Event Simulation

—

Mayank Singhal (160050039)
Sunchu Rohit (160050097)

Flow Chart (Modelling of The System)



Design of Classes

Server

- number_of_Cores
- list_of_cores
- get_Core()

Core

- Service_time_distribution
- Core_buffer
- Policy
- is_idle()
- Departure()
- Timeout()

Core_buffer

- Capacity
- List_of_threads
- get_next_job()

User

- Number_of_users
- Think_time_distribution
- create_Request()

Request

- Id
- Timeout
- Time_required
- Timestamp
- Time_spent_on_cpu

Thread

- Request
- Is_running
- thread_id

Design of Classes (Contd)

Event

- Event_type
- Event_start_time

Event_type

- Create_request
- switch_context
- Get_next_job
- Departure
- Timeout
- Drop_Request

Thread_List

- number_of_threads
- List_of_threads
- is_thread_availabable_for_a_request()
- get_thread_to_run_on_cpu()

Thread_queue

- Queue_length
- List_of_requests
- add_to_queue()
- remove_from_queue()
- Drop_request()

Program Logic

We intend to write the program using events. We create a event list (which is a priority queue) and pick most imminent event from the event_list and process them.

The events are of 6 types:

- Create_request - It is handled by calling Users.create_Request()
- switch_context - It adds a get_next_job event with context switch overheads to event list to pick next job.
- Get_next_job - This event calls Buffer.get_next_job()
- Departure - This calls Core.Departure()
- Timeout - It calls Core.Timeout()
- Drop_Request - It calls Thread_queue.Drop_request()