Technical University of Cluj-Napoca

Homework 3

Simulation Application

Queue based systems for determining and minimizing clients' waiting time.

Sucitu Stefania

30421

April 2016

1. Abstract

The aim of this project is to develop a standalone application to simulate queue based systems, analyze the waiting time and process time of tasks and find a way to minimize the previously mentioned parameters in order to increase the performance efficiency of the real life situation. In order to simplify the understanding process of what the application in discussion should do, we are going to imagine the clients of a supermarket and how they are going to pay for the chosen products by picking the most convenient cash register and how much they are going to wait until they can leave the supermarket.

2. Introduction

Queues are commonly seen both in real world and in the models. The main objective of a queue is to provide a place for a "client" to wait before receiving a "service", meaning paying for the chosen products. The management of queue based systems is interested in minimizing the time amount its "clients" are waiting in queues. One way to minimize the waiting time is to add more servers, i.e. more queues in the system (each queue is considered as having an associated processor) but this approach increases the costs of the supplier. When a new server is added the waiting clients will be evenly distributed to all current available queues. The application should simulate a series of clients arriving for service, entering queues, waiting, being served and finally leaving the queue. It tracks the time the clients spend waiting in queues and outputs the average waiting time. To calculate waiting time we need to know the arrival time, finish time and service time. The arrival time and the service time depend on the individual clients – when they show up and how much service they need. The finish time depends on the number of queues, the number of other clients in the queue and their service needs.

The application presented in what follows may be useful for a simulating and following up close the performance and flow of events of a real life supermarket. It can be of great use for store managers as they can consult who events flow and if the performance time is the desired one in their business. This application is mainly based on thread processing and queue organization and representation of data.

3. Aim of Application

Besides the imaginary aim of the simulation process presented, the real aim of the application is to display thread processing and the way tasks are efficiently distributed to different servers. The user is prompted with the possibility of picking the simulation time, by choosing the number of tasks to be performed and also the process time limits for each single one of the tasks. By pressing the start button the simulation should start. The user is then prompted with five windows, each representing a single, independent server, which is going to receive certain tasks, which will be displayed in the window. The main feature offered by the program is the easy to observe, task handling capacities of the servers. All the functionalities of the application are open for observation in the graphical user interface provided by the program.

4. Problem Analysis

4. 1 Modelling

The first step into solving the problem we are presented with is to analyze and decide what the main aim of the application is, what are the principal aspects that should be taken into account and what the qualities to be emphasized are. All of these decisions must be taken such that a good, structured and well developed program is created. The packages and class design must be considered of great interest as one good initial design could save a lot of time in the later steps of the development process.

Going further, a good choice of classes must be made. Taking into account that each class may have one main functionality which is advised to be a single, simple and approachable one, we must create such classes. Keeping this in mind the following classes are created. The Task class represents the basic entity the project is working with. The object this class is representing has two attributes: arrival time and process time which characterize one object of type task. Therefore, a constructor for the object is included and also getter and setter methods for the previously mentioned attributes are provided.

The next class created is the main engine of our application, the core of the program. The class server provides the desired functionalities of the application. A blocking queue of tasks is created and attributes for it are also created: waiting time for the task, total process time and number of tasks performed by the server. The constructor is added by default for initializing the fields the class has and also getter methods are provided. As this class is simulating a real life progressive mechanism it implements the Runnable class, therefore a method run is implemented, it running the blocking queue of tasks processes.

The Scheduler class takes care of the actions of the servers. It manages the way the tasks are dispatched on servers depending on the time limit for a waiting time and the number of servers. In order to have an efficient behavior, certain constraints are provided. A new task that just arrived will be assigned to the server with the minimum waiting time and if the already opened servers are not available a new one will be opened and start performing the task right away.

Up until now the expected behavior of this application has been presented. The focus was on how the tasks are created, how the servers should approach them and how they are dispatched and managed by a scheduler. In what follows a more testing inclined approach will be presented. A simulator class is created, in which the behavior of the previous developed classes is open for observing and checking if everything is working properly so far. This class as well implements the Runnable interface, its run method specifying the creation of servers and tasks and creating the relationship between the two of them as well as displaying the result.

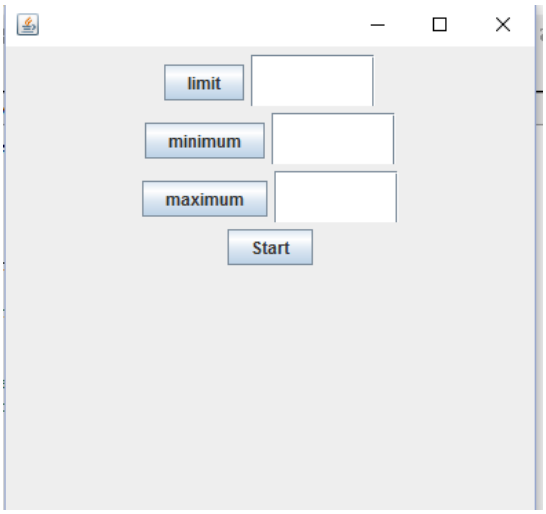
The Class Simulator Frame deals with everything connected to the graphical user interface, as a consequence it is the most complex class of the entire project. Divided into two different functional areas, the user is presented with the task of having to input the parameters defining the simulation and starting by hand the simulation. Once all these operations are performed, the control switches to the other side, which will display using scroll panes the evolution of the

servers. The incoming tasks will be added, displayed on the screen while waiting to be processed and leaving the queue once they receive the service.

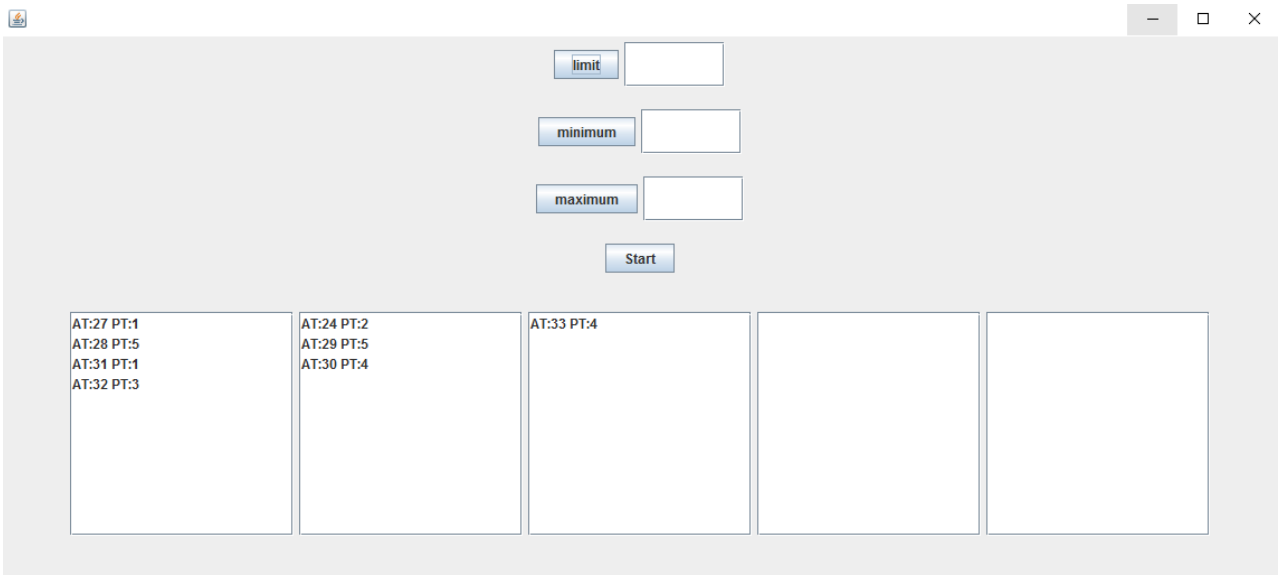
Keeping in mind the fact that the focus on what a class is aimed to do and only perform that specific action is of great importance, the program in discussion is made up of multiple classes, each serving a different but specific purpose.

4.2 User Interface

As mentioned above, the user interface is one of the most important and main features of this application. The user is presented the following window in which he has to input the required parameters and then start the application.



Once the start button is pressed the execution of the simulation starts right away. The serves are displayed as well as the incoming tasks.



6. Implementing and testing

The most used technique of testing was the insertion of the *System.out.println*(…) method in different stages of the development process. Using this method came in handy when the result of a certain operation was needed to be tested for proper working .The main advantage of this method is that it offers quick feedback and it may help locate the unusual behavior source very fast. On the other hand, the fast performance comes with a major disadvantage, its size and many lines of code if it is needed repeatedly and also the impracticality aspect in some cases.

Trial and error was the next step to obtain a perfectly working operation. After developing all the required method, any programmer would think that it should work. Unfortunately, this in not always the case. For the first test of proper performing, some input data will be entered, and a proper result should be obtained, of course considering the entered data is of proper type. If the obtained result does not matches the expected one, the first step is to check if the algorithm is correct and make the needed changes and alterations if necessary, which most of the time are.

A good technique and approach is to verify the correctness of the small modules included into the large project. If the correctness and proper performance of the basic parts is ensured, the errors in the final step of the testing are less likely to appear.

If by any chance, the method implemented does not give the desired results, even after many and may modifications, then a logical rethinking must be performed. Taking the problem step by step may give the result through a new approach, or it may help you figure out what is not correct in your first approach. This method may be very effective but it will surely cost more time. Despite that, in some cases, figuring out a new approach takes less time that figuring out the mistake in a poorly implemented and rapidly, in the urge of the moment developed solution.

7. Results

The most important objectives this project aimed to satisfy have been accomplished. A user friendly interface has been delivered, with its functionalities implemented. All the methods could be further developed and there is also space for improvements in the future. The application offers the basic operations on thread processing and queue based systems which are fully functional and good to go, ensuring the overall success of the project.

8. Conclusion

The project helped gaining more notions of knowledge in the vast and complex domain of thread processing. This project also help me get a better understanding of how the initially simple looking but in fact complex operations on threads work. It was also a good exercise of design, the way each functionality should be put in the right place, so it could offer the best results. One of the main aspects which I understood from developing this application is the fact that a good

initial design and organized initial approach helps a lot when working with the object oriented programming entities. If the base is done correctly and efficiently it saves a lot of time when developing the solution and also excludes some of rethinking processes usually mandatory. Another aspect to be mentioned is that if the initial design of the classes is done properly, the needed calls for the methods and objects can be done in far less words, which is helpful in large, complex projects.