

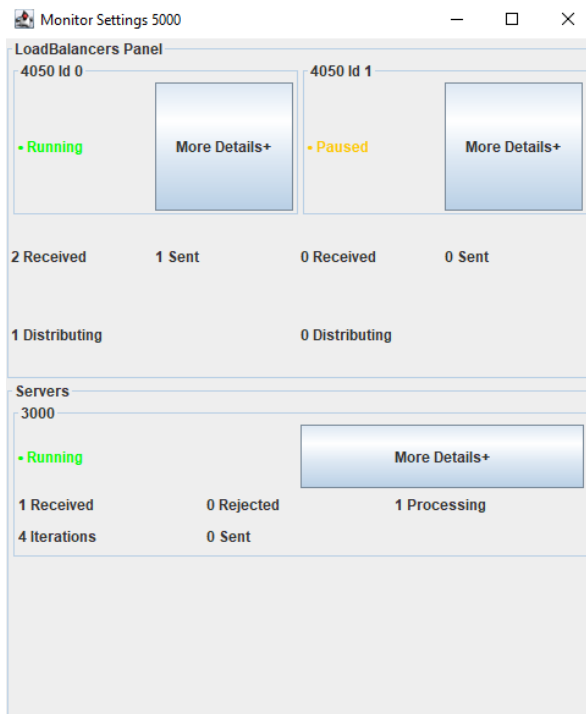
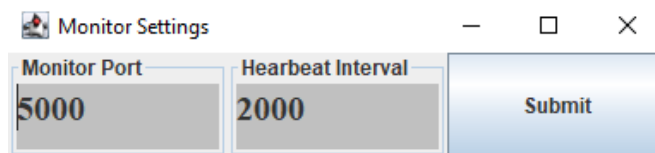
## Report of Assignment 3

# Quality Attributes

Pedro Marques, 92926 Inês Leite, 92928

## Monitor

The Monitor is essential for the program to fully function. If the Monitor is not working, initializing the individual components will not be successful. However, the opposite is not true. Load Balancer, clients and servers can be initialized and terminated without restarting the Monitor.



The first thing the Monitor asks upon initialization is some settings, including the Monitor's port and heartbeat interval. After submitting and initializing 2 Load Balancers and 1 Server the GUI is as displayed on the left.

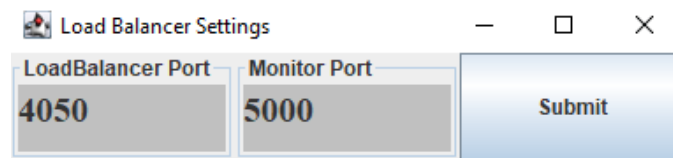
It shows information about each service which is constantly updated.

Clicking on more details for each service will show another frame with all the received and sent requests/replies.

The Monitor constantly sends heartbeat requests after a defined interval and receives updates/ other requests for information from the services. It is also responsible for defining when the second Load Balancer should wake up.

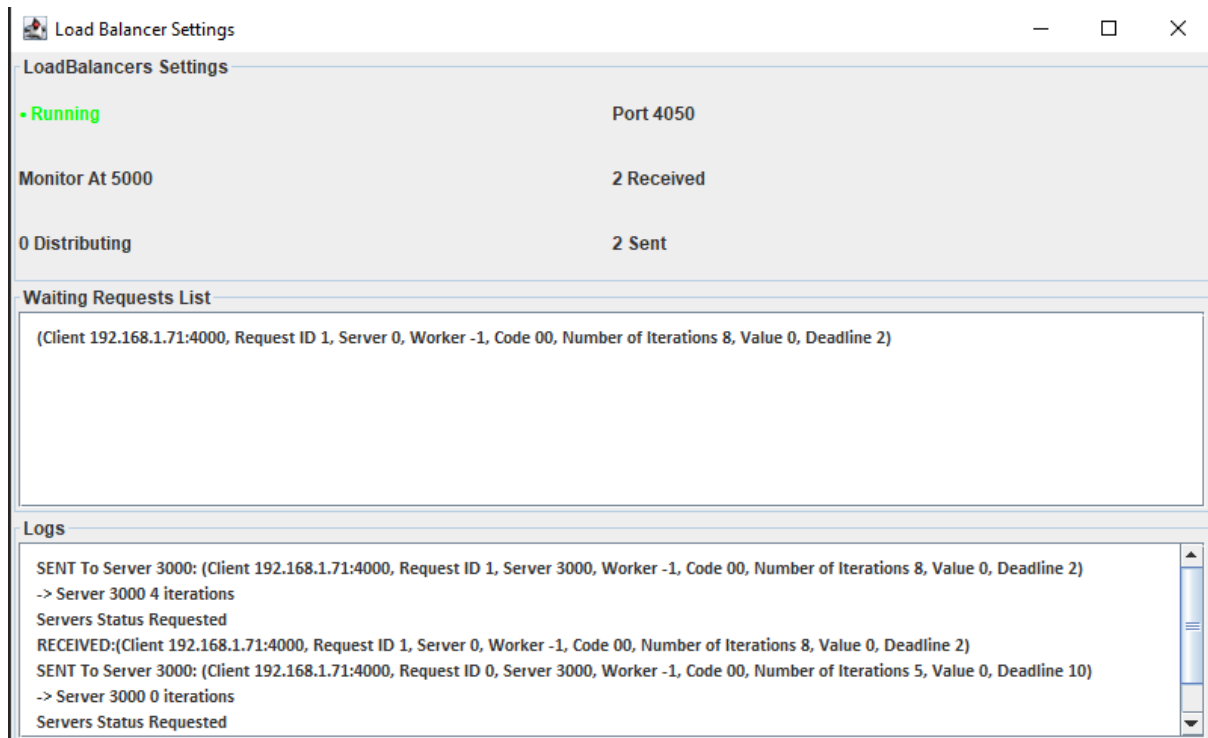
# Load Balancer

The Load Balancer communicates with the Monitor to decide where to send requests. The LB asks the Monitor for information about the server's status and the Monitor responds with the server's current iteration count. The one with the lowest count is chosen to be forwarded the request. The LB does not reject requests, that competence falls to the server.



The Load Balancer starts by asking for the Monitor's port and its own port, where it will listen for the client's requests.

When loading the Load Balancer, it awaits for the Monitor's permission to start its process. As such, if one Load Balancer fails, the second one starts listening on the same port.

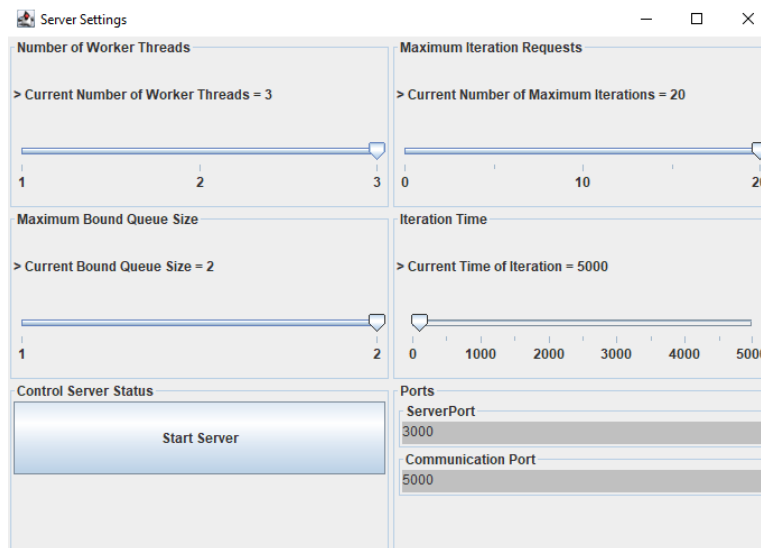


As shown, the LB's interface shows its current status, the list of requests waiting to be distributed according to the Earliest-Deadline priority and the full list of received requests, distributed requests and everytime the LB asks the Monitor for server information.

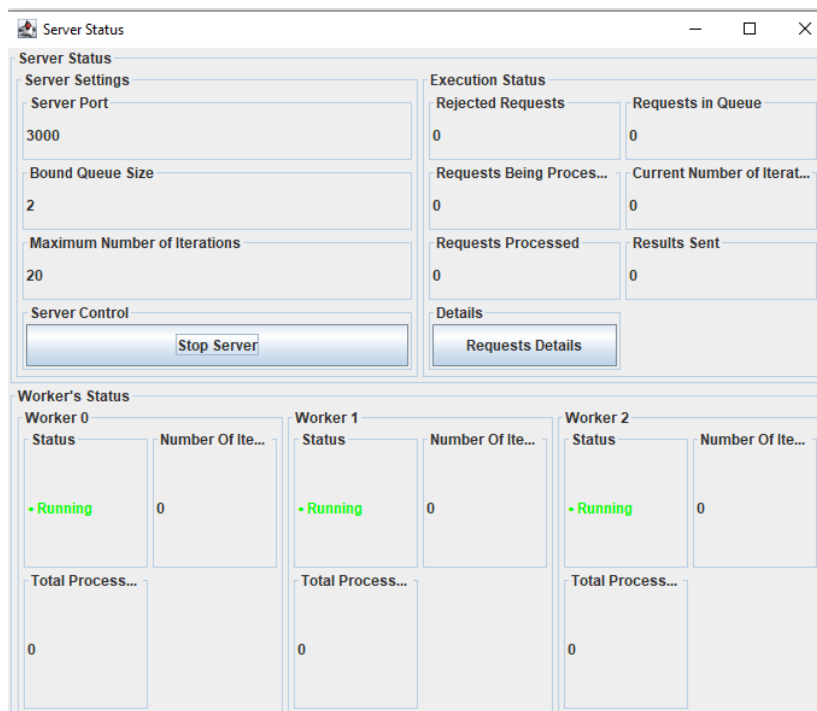
# Server

The server receives requests and adds them to the inbound requests queue if possible. After that it gives the work to different workers which will process it, calculate the results and the server will proceed to send these replies to the client who issued the request.

If there is no space in the server for the request to be processed the request is rejected and sent back to the client with code 03. This happens when the number of iterations (work load) added by this request surpasses the maximum number of iterations allowed or when there is no space in the queue.



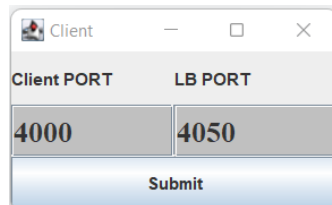
The server's initial GUI asks for a lot of configuration, including its own socket port and the Monitor's communication port.



As shown, the Server provides a lot of information, including the number of iterations left for each worker and the total number of requests processed by them. The “Requests Details” button includes a log of all the received requests and sent replies.

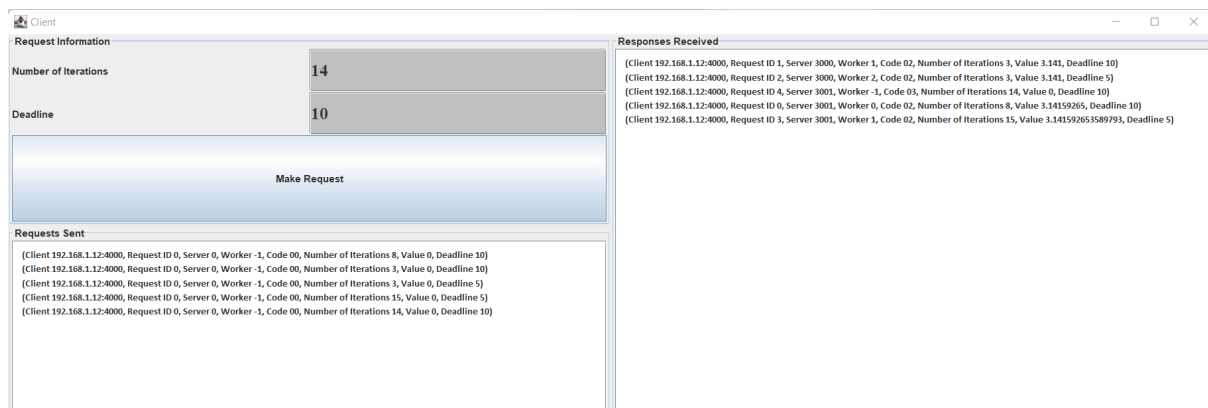
## Client

The Client sends requests to the Load Balancer and receives the responses from the Server. The Client starts by asking for its own port and the LB’s port, to be able to send the requests.



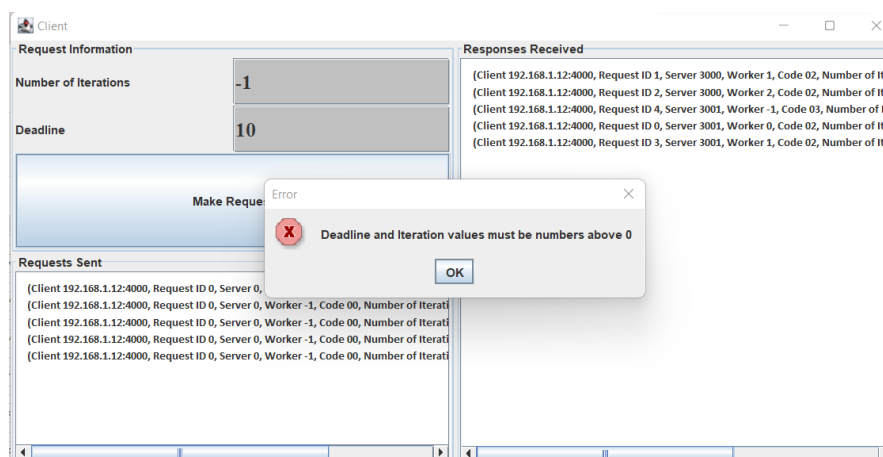
A small window titled "Client" with two input fields: "Client PORT" containing "4000" and "LB PORT" containing "4050". Below these fields is a "Submit" button.

After that, another window appears, with 3 sections: one for sending the requests, being able to input the number of interactions and the deadline, with a button to make the request; another one to see the requests made; and lastly, one to show the responses received to the requests.



The "Client" window is divided into three main sections. The top-left section, "Request Information", has input fields for "Number of Iterations" (value: 14) and "Deadline" (value: 10), with a "Make Request" button below. The bottom-left section, "Requests Sent", displays a list of request logs. The right section, "Responses Received", displays a list of response logs.

There are some checks, to see if the information inputted is valid, for example, no negative numbers, no text, etc.



The "Client" window is shown with an error dialog box overlaid. The "Number of Iterations" field contains "-1" and the "Deadline" field contains "10". The error dialog has a red "X" icon and the text: "Deadline and Iteration values must be numbers above 0". An "OK" button is at the bottom of the dialog.

## What is not implemented or not working correctly

Everything is implemented.

Even though the Load Balancer distributes the requests fairly and, when it dies, the requests left are sent to the next Load balancer, there could be a certain amount of data replication. This occurs when the Load balancer is killed between sending the data to the client and notifying the Monitor that it actually sent the data.

## Contribution

- Inês Leite, 50%
- Pedro Marques, 50%