Integrating Web Sockets in OTrees to Improve Optimization in Experiments

Primrose Washinyira, Shepard Chifamba

Software Engineering, Harare Institute of Technology, Harare, Zimbabwe
Email: *primwashy[at]gmail.com*

Abstract: ***A Python - based framework called OTree is used to design and carry out economic experiments. A higher margin of delay was observed when OTrees were utilized alone without web sockets (Chen et al., 2016a) . Because of this, OTrees were ineffective, particularly when conducting studies that called for real - time communication. Latency was shown to be an issue for OTrees with web sockets; however, this gap is closed if web sockets are included (Crede et al., 2019a) . One kind of communication protocol that enables full - duplex communication between a client and a server is web sockets. Researchers and experimenters would be able to design more dynamic and interactive experiments that call for real - time updates or feedback by adding web sockets into Otree. The use of embedded web sockets on trees to reduce latency by allowing real - time communication between the client and server is demonstrated in this article. More opportunities for doing online experiments and playing games in the classroom would arise from the incorporation of OTrees. Real - time data collection from participants is possible for experimenters. This program is ready to use and especially valuable for educational and research applications. Web socket integration would allow real - time communication between the server and clients, which would improve Otree's usefulness as a platform for behavioral research. This would make it possible to conduct more dynamic and engaging studies that record participants' actions in real time, producing more reliable and perceptive study results.***

Keywords: Otrees, SignalR, Web sockets

1. Introduction

While OTrees generally eliminates the need for participants to be at the same location to interact with each other using a web socket to improve user interface response time, there is still room to improve the experimental behavior of this application by improving optimization, making use of content distribution networks, increasing student engagement, and improving data analysis. The gap is addressed by integrating Web sockets to improve the optimization of experimental learning.

2. Otrees

2.1. Background history of Otrees

An OTree was developed by Daniel Chen, Martin Schonger, and Chris Wickens (2016) and it can be used for free ("GitHub - CPHPeters/Guest_Lecture_Experiments, " n. d.) . Otrees are a type of experimental learning tool that allow students to explore and learn about complex topics in an interactive way. They are designed to help students develop critical thinking skills, problem - solving abilities, and creativity. Otrees are composed of a series of questions or tasks that require the student to think critically and come up with solutions. The questions or tasks can be related to any subject, from math and science to history and literature. They can be used in both classroom and online settings, allowing students to explore topics at their own pace. With the use of their browsers, participants in online/lab experiments can communicate with the experimenters and with one another using the platform/software package known as OTree. Since real - time applications require a permanent connection with uninterrupted communication between the clients and the server, as seen below, the previous oTree design was not appropriate.

2.2. Otrees for experimental learning

A number of researchers have significantly contributed to the use and improvement of Otrees. (Chen et al., 2016b) designed an interactive online experiment in the lab using open - source software called Otree. They discovered quite a number of gaps, which then drove them to make use of trees for experiments. The first gap was that it was difficult to use a Windows desktop networked environment for field settings. The second gap was that they found out that with evolving technology, there are now a number of network devices being used other than the desktop, so using a platform - independent application was the only way to go.

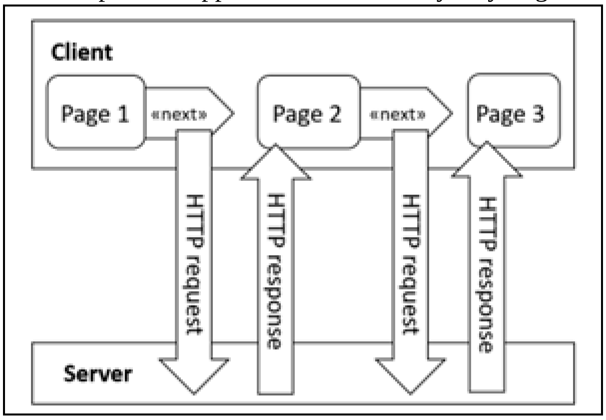

Figure 2.1: The OTree setup that was used before introducing web sockets (Crede et al., 2019b) .

They preferred Otree because they use the existing industry standards, which are compatible with plug - ins, extensible, and allow users to view the source code (Chen et al., 2016b) . Over and above that, Otrees requires minimal hardware and software, making it a little bit cheaper as compared to other applications. Chen et al. analyzed the otree, carried out the research successfully, and then discovered one major drawback of otrees, which is that being used alone does not suit every use case because it is not a real - time application, thus the user interface does not respond within milliseconds to actions (Chen et al., 2016). A recommendation was made that real - time applications must be built into otrees to improve efficiency. (Crede, 2019) noted that the otree architecture is not suitable for real - time applications because it requires the user to request a page from the otree server, then make some inputs, click the next button, and so on.

They tried to address the gap in latency by trying to make otrees more real - time while interacting with a large number of groups. They did an experiment by running a double action on an Amazon Mechanical Turk using web sockets. It was concluded that the participants in the Amazon Mechanical Turk behaved similarly to the participants in the laboratory. However, they recommended that integrating web sockets into otrees was another way of further improving the functionality of otrees.

3. Integrating Web sockets in Otrees

Crede et al after adding a web socket discovered that there was a gap to be addressed to improve the efficiency of the OTrees through the integration of web sockets (Crede, 2019) .

Integrating web sockets with OTrees can provide a number of benefits. Web sockets allow for real - time communication between the server and the client, which can be used to create a more interactive user experience. This could be used to provide live updates on the status of orders, or to enable users to chat with customer service representatives in real time. Additionally, web sockets can be used to reduce latency and improve performance by allowing for faster data transfer between the server and client. This could be especially beneficial for applications that require a lot of data transfer, such as streaming services or online gaming. Finally, web sockets can also help improve security by providing an encrypted connection between the server and client.

3.1. How to integrate web sockets in otrees

- Install the necessary packages for web socket integration in oTree.
- Start by installing the required packages for web socket integration in oTree for example installing Django Channels, a package that provides support for web sockets in Django - based applications.
- Configure Django weChannels to work with your oTree project. This involves adding the necessary settings to your project's settings. py file, such as specifying the routing configuration and enabling the Channels layer.
- Create a WebSocket consumer that will handle incoming WebSocket connections and messages. This consumer will define how the oTree application interacts with connected participants.
- Define WebSocket routes that map specific URLs to the WebSocket consumer. These routes determine which consumer handles incoming connections for different parts of the oTree experiment.
- Configuring ASGI Application in order to enable web socket support in oTree.
- Implement WebSocket functionality for real - time communication between participants and researchers. This can include sending updates or notifications to participants, receiving messages from participants, or any other interaction you want to enable through web sockets.
- Integrate with oTree models and views through modifying existing models or views to take advantage of real - time communication capabilities. For example, by updating participant models or views to trigger events that send messages over web sockets.
- Finally, thoroughly test the integration to ensure it works as expected. Once tested successfully, deploy your updated oTree application with web socket integration enabled.

3.2 Calculating Latency

The total of all conceivable delays that a packet may experience during data transmission is known as network latency. Generally, network latency is measured in milliseconds and expressed as round - trip time (RTT (Datta, 2022)). Processing, queuing, transmission, and propagation delays are all included in network latency.

The formula for calculating latency is given by:

$$NL = DP + DQT + DT + DPR$$

NL - network latency
DP - Data processing time
DQT - Data queuing time
DT - Data transmission
DPR - Data propagation delays

4. Findings after integrating web sockets in otrees

The data presentation of the study's findings is the focus of this chapter. To aid in reading and comprehension, experiments were conducted, and the results are tabulated after being analyzed.

4.1. A Table showing results obtained after running various experiments

Table 1: Experiments of otrees

Outputs of various experiments using otrees and web sockets		
Experiment	NL = DP+DQT+DT +DPR	Overall evaluation
Otree without web sockets	Approx 3 milliseconds	Significant latency
OTree with web sockets	Approx 1 millisecond	Less significant latency
Otree with integrated web socket	Approx 0.02 milliseconds	Insignificant latency

4.2 Data representations showing results after conducting various experiments on otrees.

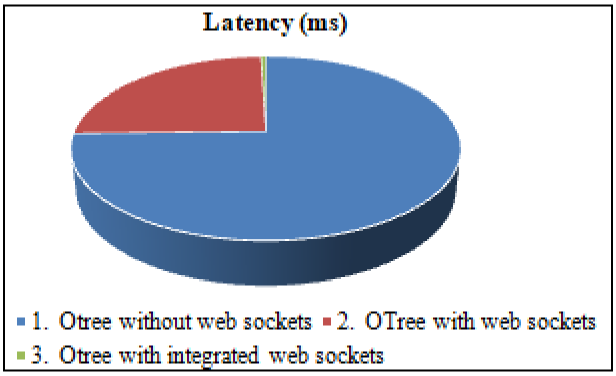

Figure 4: Pie chart showing the results obtained on otrees

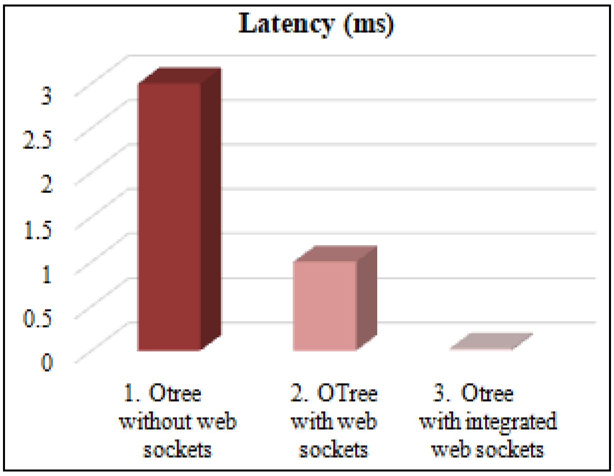

Figure 4.1: A chart showing results of otree latency experiments

From the above results in (fig 4.0) and (fig 4.1) otrees without web sockets will result in more latency as compared to otrees with web sockets. Otrees without integrated web sockets will have a very insignificant latency.

5. Findings

From the above results in (fig 4.0) and (fig 4.1) otrees without web sockets will result in more latency as compared to otrees with web sockets. Otrees without integrated web sockets will have a very insignificant latency. This shows that there is a very significant improvement after integrating web sockets in otrees.

6. Conclusion

In conclusion, adding web sockets to oTree apps may significantly improve their capacity for interactivity and real - time communication. Web sockets make it possible to create dynamic, responsive interfaces that may update in real time in response to user input or server - side events by enabling bidirectional communication between the server and the client. This can create new opportunities for incorporating intricate game elements and mechanisms, as well as make the user experience more dynamic and interesting. The usefulness and usability of oTree apps might be greatly enhanced by adding web sockets to the framework. The researchers recommend that more experiments can be done to completely remove latency by using other real time applications in otrees.

Acknowledgements

Avoid the stilted expression, "One of us (R. B. G.) thanks. . . " Instead, try "R. B. G. thanks". Do NOT put sponsor acknowledgements in the unnumbered footnote on the first page, but at here.

References

[1] Chen, D. L., Schonger, M., Wickens, C., 2016a. oTree—An open - source platform for laboratory, online, and field experiments. Journal of Behavioral and Experimental Finance 9, 88–97. https: //doi. org/10.1016/j. jbef.2015.12.001

[2] Chen, D. L., Schonger, M., Wickens, C., 2016b. oTree—An open - source platform for laboratory, online, and field experiments. Journal of Behavioral and Experimental Finance 9, 88–97. https: //doi. org/10.1016/j. jbef.2015.12.001

[3] Crede, A. - K., 2019. Experimental evidence on behavior in organizations and markets. https: //doi. org/10.24442/BORISTHESES.1823

[4] Crede, A. - K., Dietrich, J., Gehrlein, J., Neumann, O., Stürmer, M., von Bieberstein, F., 2019a. Otree: Implementing Websockets to Allow for Real - Time Interactions – a Continuous Double Auction Market as First Application. SSRN Electronic Journal. https: //doi. org/10.2139/ssrn.3631680

[5] Crede, A. - K., Dietrich, J., Gehrlein, J., Neumann, O., Stürmer, M., von Bieberstein, F., 2019b. Otree: Implementing Websockets to Allow for Real - Time Interactions – a Continuous Double Auction Market as First Application. https: //doi. org/10.2139/ssrn.3631680

[6] Datta, S., 2022. How to Calculate Packet Time from Latency and Bandwidth | Baeldung on Computer Science [WWW Document]. URL https: //www.baeldung. com/cs/packet - time - latency - bandwidth (accessed 11.29.23).

[7] GitHub - CPHPeters/Guest_Lecture_Experiments: Repository on building state - of - the - art experiments using oTree and running them in online participant pools. [WWW Document], n. d. GitHub. URL https: //github. com/CPHPeters/Guest_Lecture_Experiments (accessed 10.19.23).