

TANX - Python Developer Case Study

Analysis and Result

In this study, 2 different Python libraries/frameworks are chosen to implement two distinct WebSocket server-client implementations: one is the python library “web sockets”, and the other is the web framework “tornado”. The comparative analysis are given for this implementations are given below.

To analyze and compare the performance of the WebSocket server-client implementations using the websockets library and the Tornado framework, let's focus on key aspects such as latency, throughput, scalability, and other relevant metrics.

1. Latency:

- For real-time random data transmission:
websockets implementation: 1.002 seconds
Tornado implementation: 1.001 seconds
- For file transfer (7MB):
websockets implementation: 0.27 seconds
Tornado implementation: 0.02 seconds
- For file transfer (50MB):
websockets implementation: 1.72 seconds
Tornado implementation: 0.19 seconds

2. Throughput:

- Throughput refers to the amount of data transferred per unit of time.
- The Tornado implementation generally demonstrates higher throughput compared to the websockets implementation, especially noticeable in file transfer scenarios.
- Tornado's asynchronous nature and efficient event loop handling contribute to better throughput.

3. Scalability:

- Both implementations should be able to handle multiple concurrent connections.
- Tornado's non-blocking I/O and event-driven architecture make it highly scalable.
- websockets library also supports asynchronous operations, enabling scalability, but Tornado's framework may offer more extensive features for building scalable applications.

4. Ease of Use:

- The websockets library provides a straightforward interface for implementing WebSocket servers and clients.
- Tornado, being a full-fledged web framework, offers more features and flexibility but may have a steeper learning curve.

In summary, while both websockets and Tornado are capable of handling WebSocket communication effectively, Tornado generally demonstrates better performance in terms of latency and throughput, especially for file transfer scenarios. However, the choice between the two depends on factors such as ease of use, scalability requirements, and existing familiarity with the frameworks.