

Solution to the teaser:

The application uses AJAX and a single thread for the mock server. The minimum time for ajax to service multiple instantaneous requests is 4ms, so making call every 100 microseconds is out of the question.

Though the application I have developed could service requests upto 4ms, it would not display/service requests every 100 microseconds. So here is my solution to the problem:

1. Use multithreading on the server to make DB entries every 100 microseconds.(This is because even Thread.sleep can take time till 1 millisecond). So having multiple threads (~10) that sleep every 1 ms could do the job.
2. In global.js (front-end), make an ajax call, say, every 400 milliseconds.
3. In app.js (nodeJS) , write a query that takes only the first updated row. Calculate the sum of all the bytes and packets (like a buffer) until (currentTime - startTime > 390) (10ms buffer time to prevent overrides) . Send data as response to the AJAX call.

Other enhancements:

Use WebSocket instead of AJAX to lower latency