PSU CS 461/561
Open Source Software Development
Summer 2018

Author : Yokesh Thirumoorthi
Project : Rusher
License: MIT
Contact: yokesh@pdx.edu; yokeshthirumoorthi@gmail.com

Description

Full stack web solution

- Data Persistence
  - SQLite
- Websocket Server
  - Rust
- Presentation layer
  - React
    - Collaborative Text Editing
    - Chat

Benchmarked

- ~270,000 simultaneous open connections

Who and How?

Who is this software meant for?

- Anyone who wants to build their own realtime applications with websockets.
  - Startups

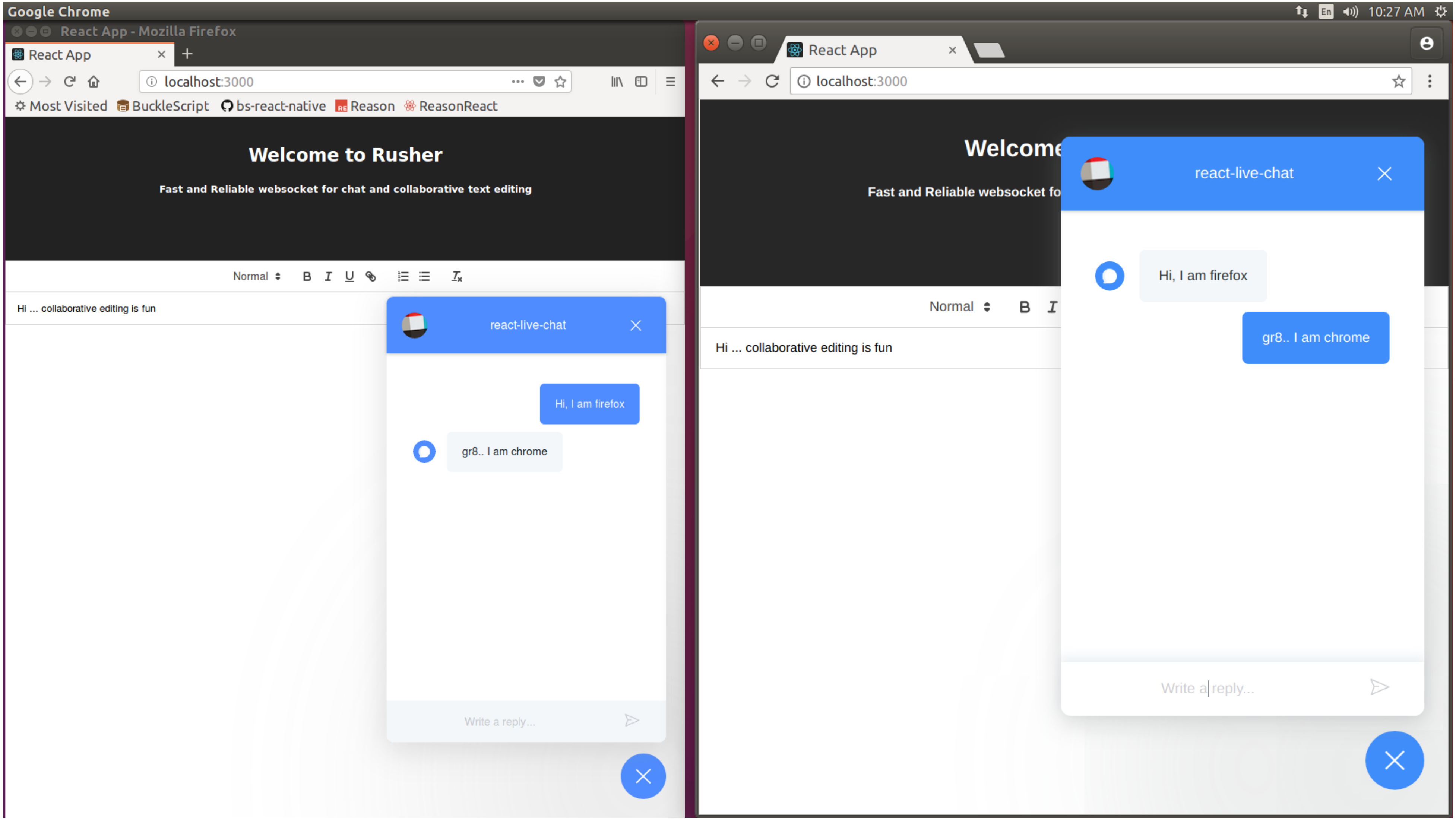
    * Code is not production ready yet

Who and How?

How does it meet their needs?

- Low cost
  - Easy Maintenance and Good Reliability with Rust
  - Performance with in memory DB
- Quick start
  - Example – Shows how to glue your needs.
  - Try your alternatives

Web Screens



Welcome to Rusher

Fast and Reliable websocket for chat and collaborative text editing

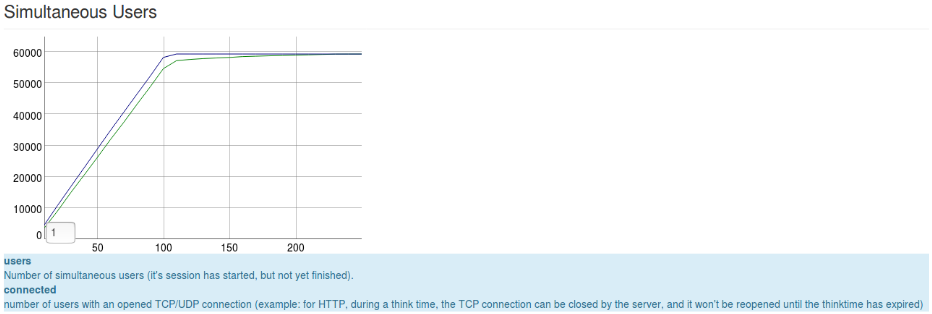
Hi ... collaborative editing is fun

react-live-chat

Hi, I am firefox

gr8.. I am chrome

Benchmarking Stages



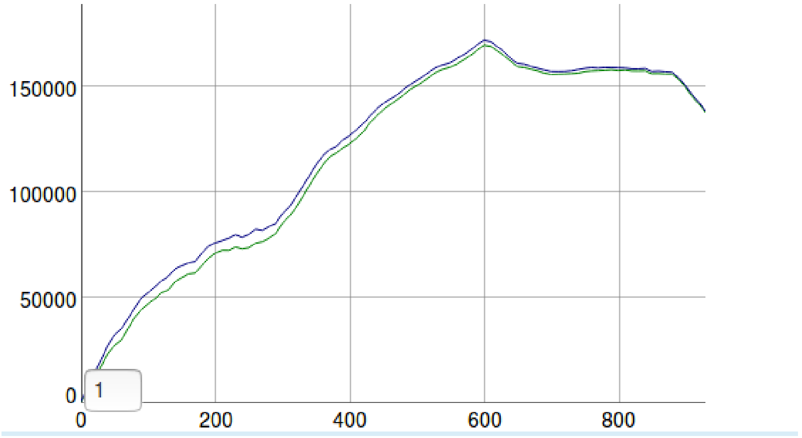
Simultaneous Users

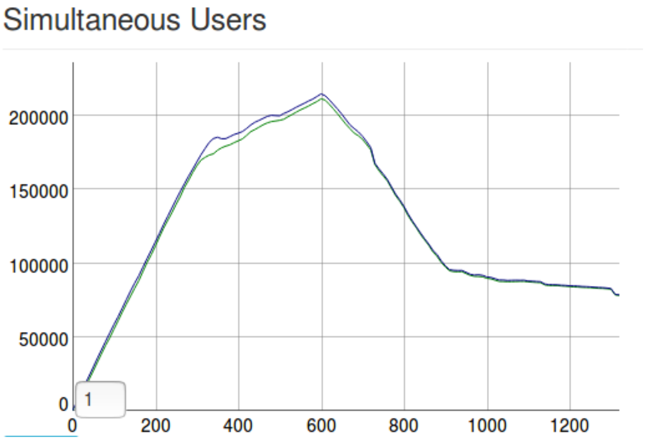
users

Number of simultaneous users (it's session has started, but not yet finished).

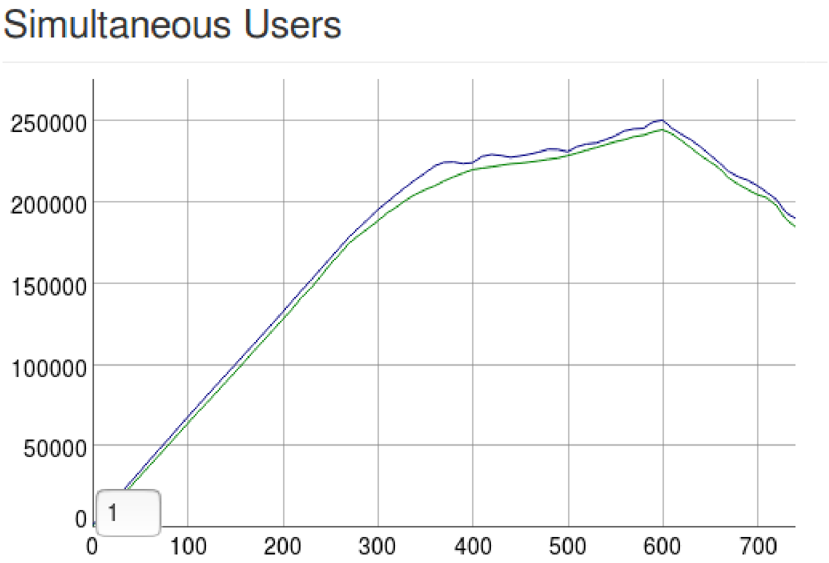
connected

number of users with an opened TCP/UDP connection (example: for HTTP, during a think time, the TCP connection can be closed by the server, and it won't be reopened until the thinktime has expired)



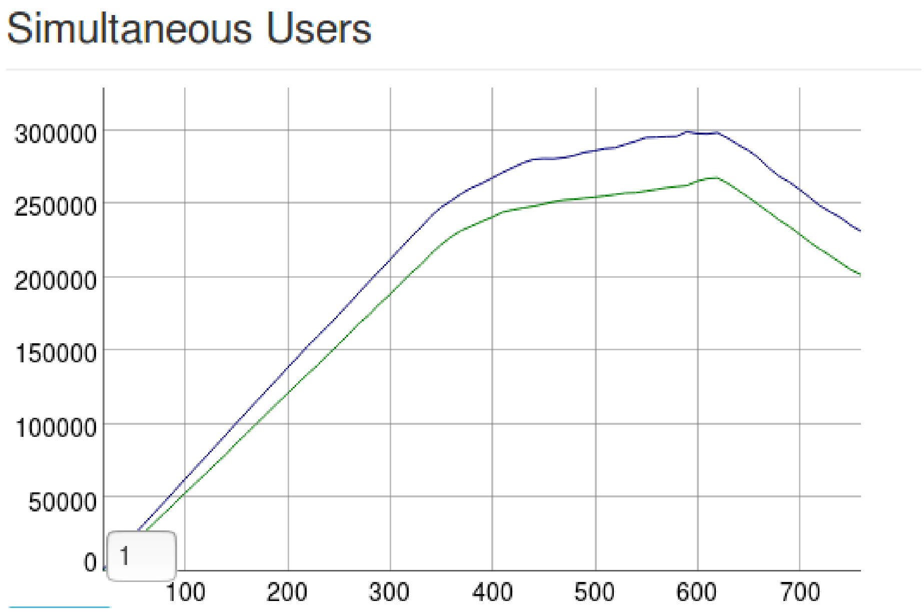


Simultaneous Users



Simultaneous Users

Benchmarking Result ~270,000



- Client 4 machines
  - 3 boxes – 6 cpu;16 GB
  - 1 box – 4 cpu;8 GB
- Server
  - 1 box - 4 cpu;8 GB

Other options vs This Project

- Commercial:
  - Firebase, Pusher.
- Non-Commercial:
  - Any opensource Websocket.
  - Successfully benchmarked for high number of connections ?
    - Erlang or Haskell

Other options vs This Project

This project provides a combination of all these.

- Data persistence with websocket connections.
- Some nice apis.
- Strong benchmarking records.

Motivations

- Rust programming
- My own startup experiences
  - Firebase and Cost
  - Javascript

Success

- Rust with actix, serde and diesel
- Collaborative editor
- More controlled environment
- Tsung based testing environment

Failures

- Not 2 million yet
- Clusters with Google cloud platform

Lessons

- Power of composing softwares
  - By composing nice and small open source projects, we could create really powerful software solutions.

- Benchmarking is hard

Future

- Improve Implementation Part
  - Security around this websocket server
  - Other DBs, such as Redis, Mongo, SQL etc to improve the data archiving and retrieve choices
  - An example with react native
  - More apis
    - Activity feeds within friend networks
    - Realtime bets and
    - Realtime trading and more

Future

- Improve Benchmarking Part
  - For bigger numbers
  - With different types of Dbs
  - Provide data visualization by comparing results

Thank you

- Summary
- Code @ https://github.com/Yokeshthirumoorthi/rusher