

Mission: Performance Impossible - Real time chat webapp performance

Mohammed Zubair Ahmed

Mattermost

Open source platform for secure collaboration across the entire software development lifecycle for communities and enterprises.

The screenshot displays the Mattermost web interface. On the left is a sidebar with a navigation menu for 'R&D Org' containing sections for 'FAVORITES', 'COMPANY', 'MOBILE', and 'CLOUD'. The main area shows a channel named 'Mobile DevOps' with a message from Amara Nuñez at 10:33 AM. She has shared a PDF file titled 'Mobile User Analytics.pdf' (15KB) and mentioned '@Michael' and '@Susan'. Below this, John Vu at 10:35 AM asks '@Alex' about triggering the release pipeline from GitLab. Alex Rodriguez at 10:37 AM responds that it's possible but complicated. John Vu at 10:40 AM agrees and mentions creating a Jira ticket. A 'Zoom Meeting' card is displayed with the Personal Meeting ID (PMI) 3271823343 and a 'Join Meeting' button. On the right, a 'Thread' view shows a message from Ayanna Moore at 10:34 AM explaining the use of Splunk for monitoring logging points and including a sample dashboard image. The dashboard image shows a network diagram and a table of services.

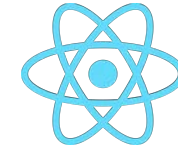
Service	Uptime	Error Rate	95th Percentile	99th Percentile
api-gateway	1.00	0%	200ms	400ms
cache	1.00	0%	10ms	20ms
db	1.00	0%	10ms	20ms
frontend	1.00	0%	10ms	20ms
mail	1.00	0%	10ms	20ms
monitoring	1.00	0%	10ms	20ms
notification	1.00	0%	10ms	20ms
payment	1.00	0%	10ms	20ms
search	1.00	0%	10ms	20ms
storage	1.00	0%	10ms	20ms
webhooks	1.00	0%	10ms	20ms



Architecture of webapp

Core technologies

- React
- Redux with Thunks
- WebSocket API



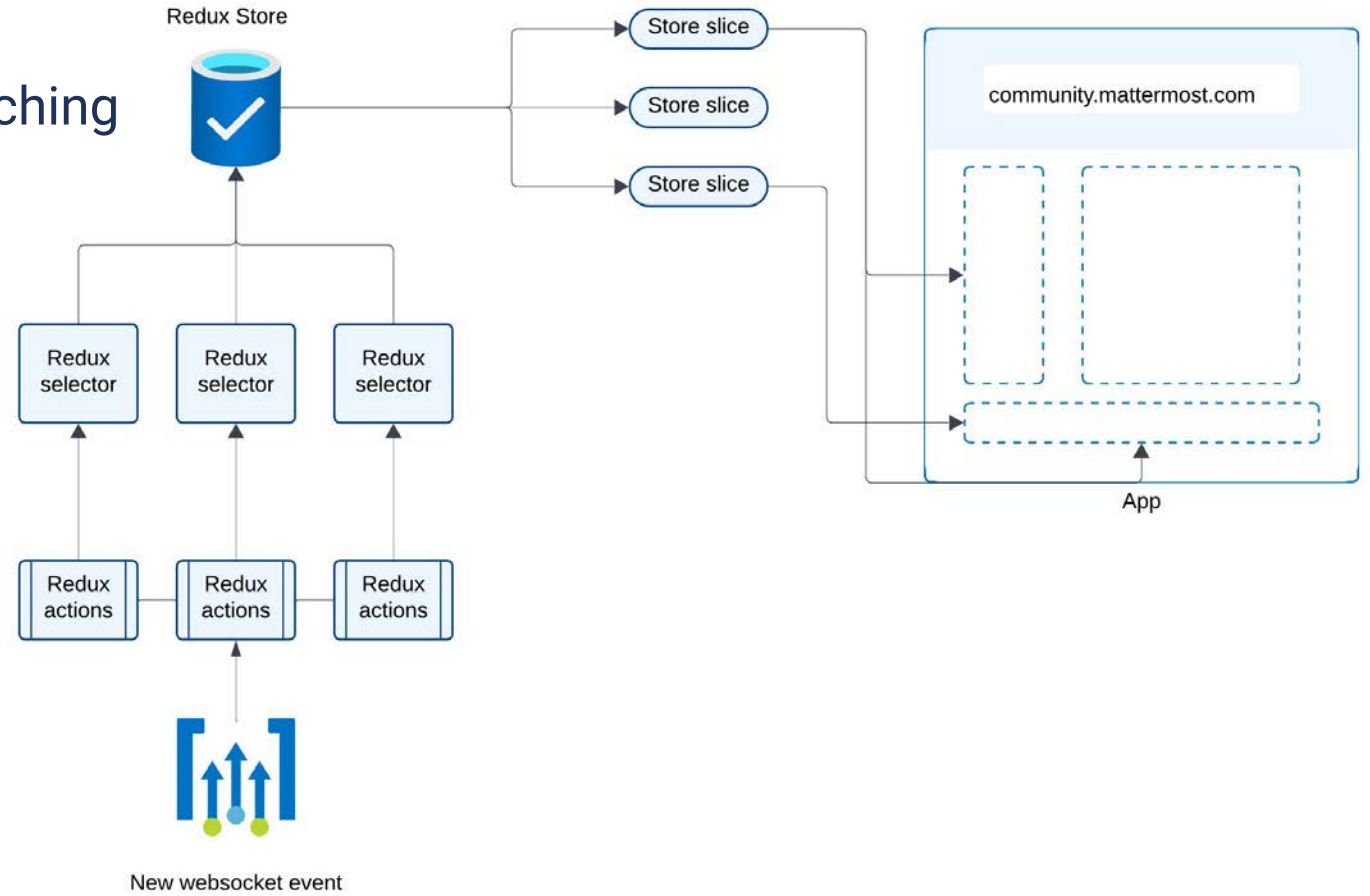
Highlights

- Single page app
- Client driven webapp
- Realtime updates



Challenges with Real time data

- High inflow volume in elevated usage
- Balancing over fetching and under fetching
- UI responsiveness



Optimising performance



[MM-58463] Remove defer for main/entry JS scripts #2727

[MM-57744] Improve status received reducers #26670



[MM-59299] Investigate further breaking down chunks of components/
node_modules for initial load #27845

[MM-57384] Investigate app performance on repeated calls to `users/`
`users/ids` and `users/ids` calls on posted event #26644



[MM-58456] Make react library be loaded at DOMContentLoaded
onload #27192

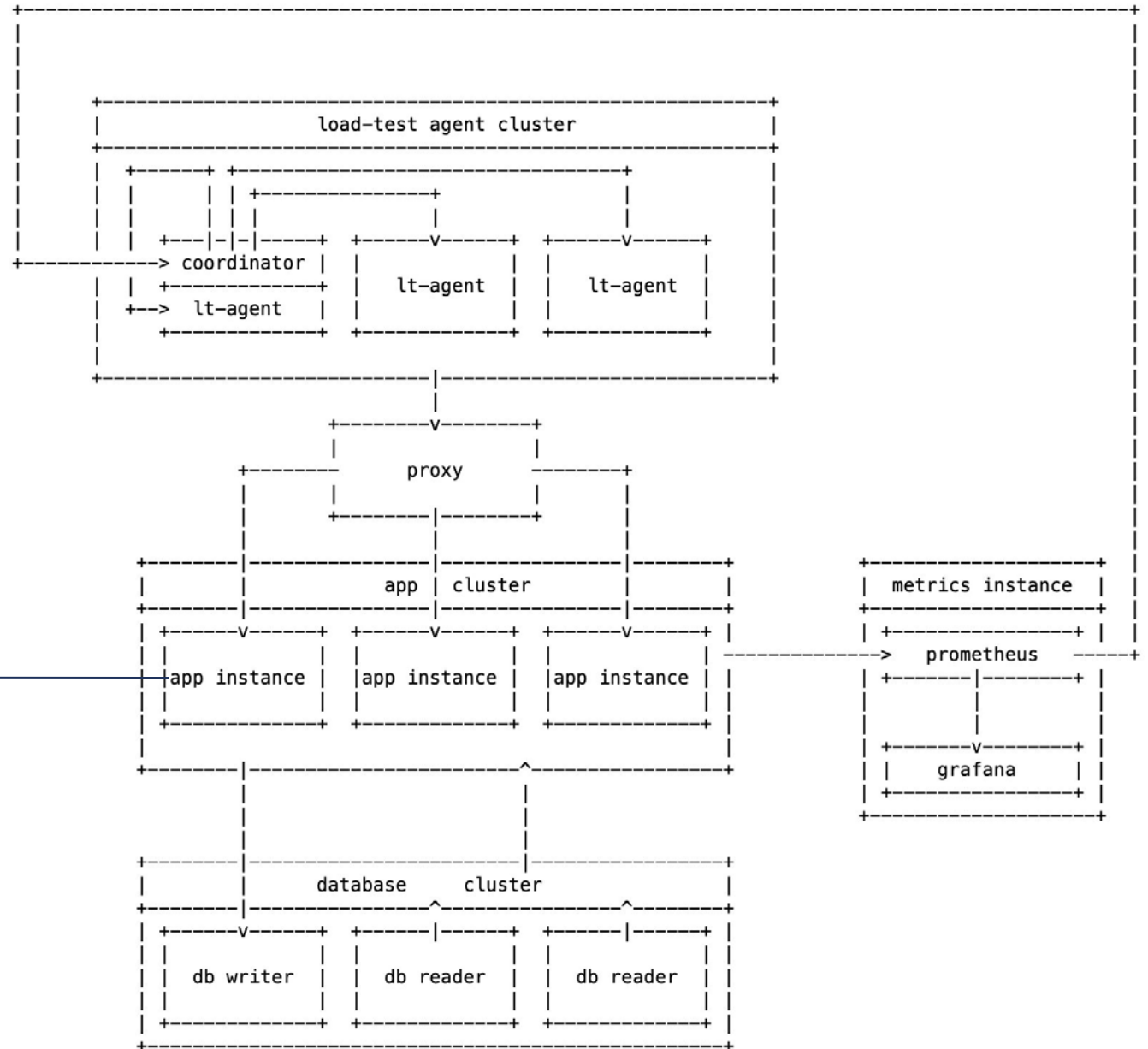
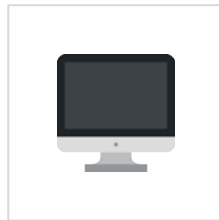


[MM-58445] Try deferring plugins scripts loading #27179



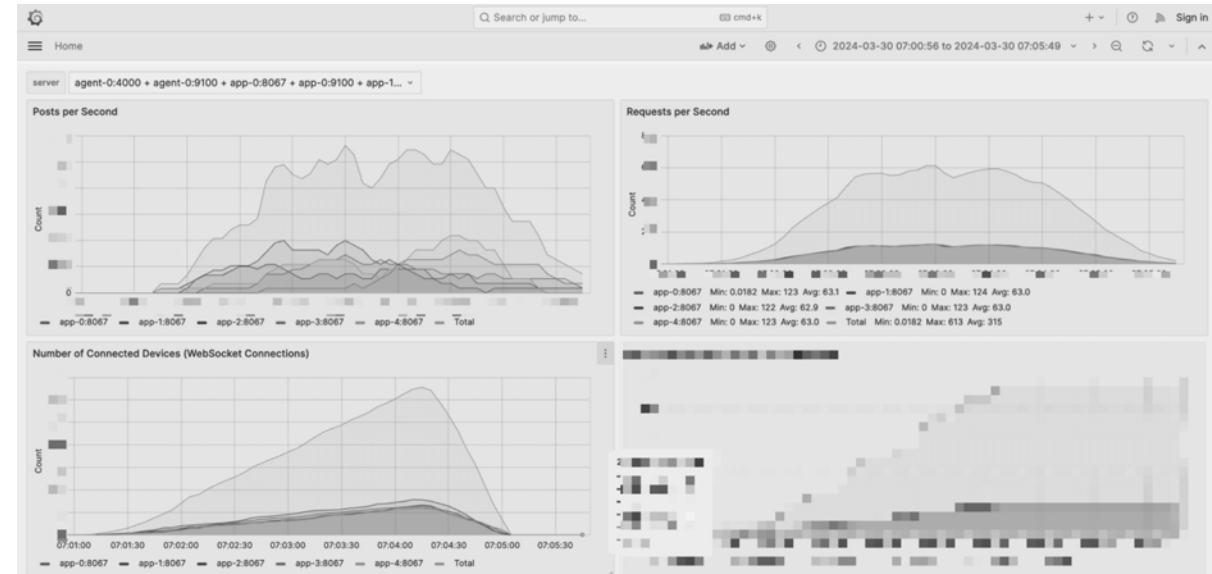
mattermost-load-test-ng

- UserController - in charge of controlling and mimicking user behaviour in loop.
- Coordinator - in charge of managing cluster of load testing agents.



Comparison

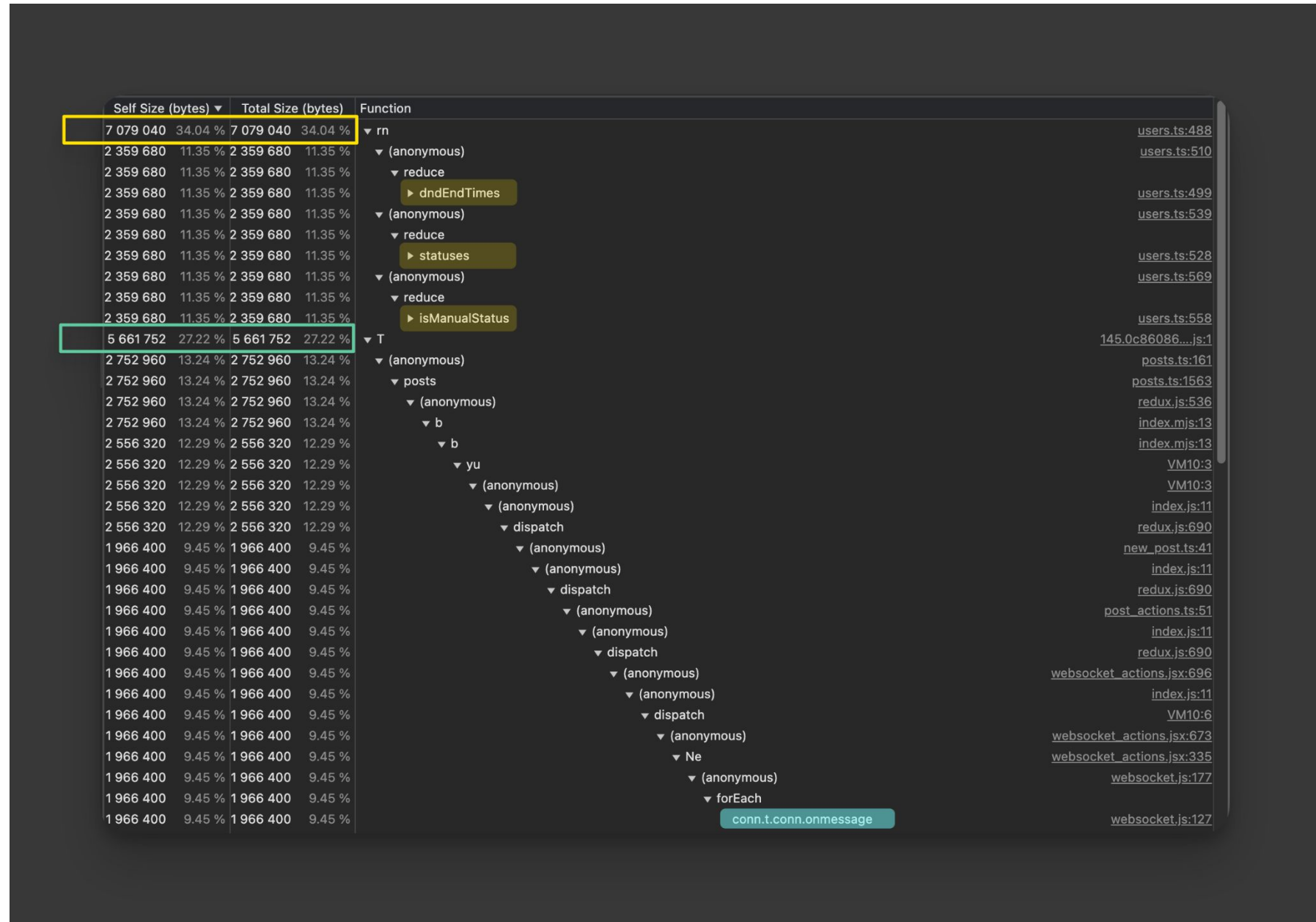
- Run the load test
- Wait for equilibrium set conditions to reach
- Get the app node's URL
- Run client automation
- View metrics in Grafana
- Take other metrics from browser



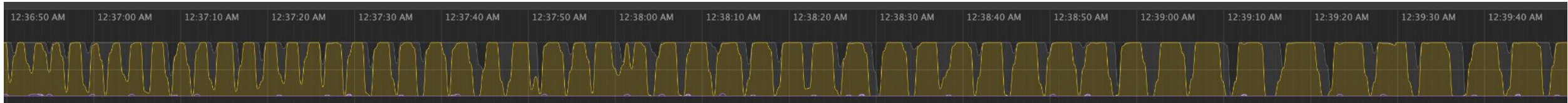
Quantifying performance



Memory profiles

























CPU usage



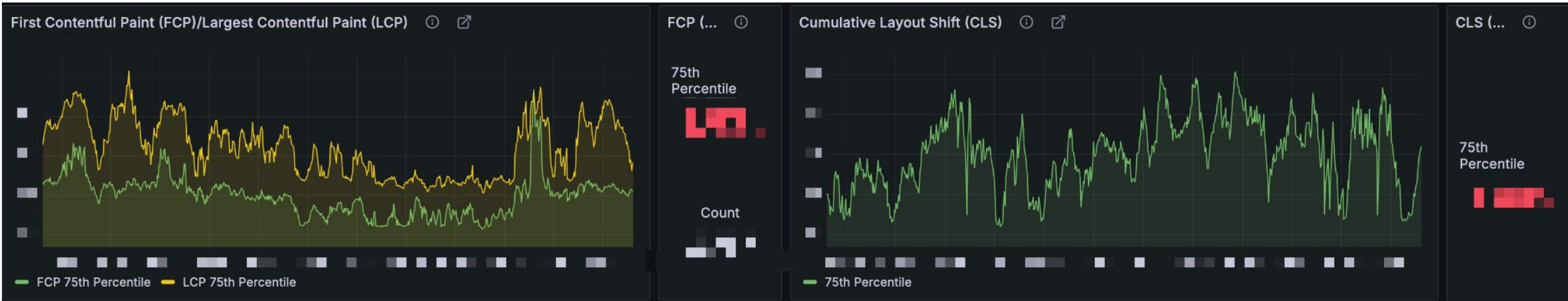
Frame per second

The screenshot displays a Discord server interface. On the left, a sidebar shows system performance metrics: 'Frame Rate' at 48.6 fps, 'GPU raster' at 43.6 fps, and 'GPU memory' at 87.8 MB used (882.5 MB max). Below these are channel and direct message lists. The main content area features a video player showing a diagram with a central play button. The diagram consists of a central square with a play button, surrounded by four boxes: 'Installation (Automotive)' (top-left), 'Installation (Clear Workspaces)' (top-right), 'Cloud Provisioner' (bottom), and 'Installation (Automotive)' (bottom-left). Below the video is a chat log with multiple messages from users like 'testuser-33216', 'testuser-5725', 'testuser-34086', 'testuser-40481', 'testuser-49871', 'testuser-25078', 'testuser-35346', 'testuser-28661', 'testuser-34086', 'testuser-21896', and 'testuser-6345'. The chat messages contain various words and phrases, some of which appear to be generated or repeated. At the bottom, there is a text input field for sending a message to the 'Off-Topic' channel.

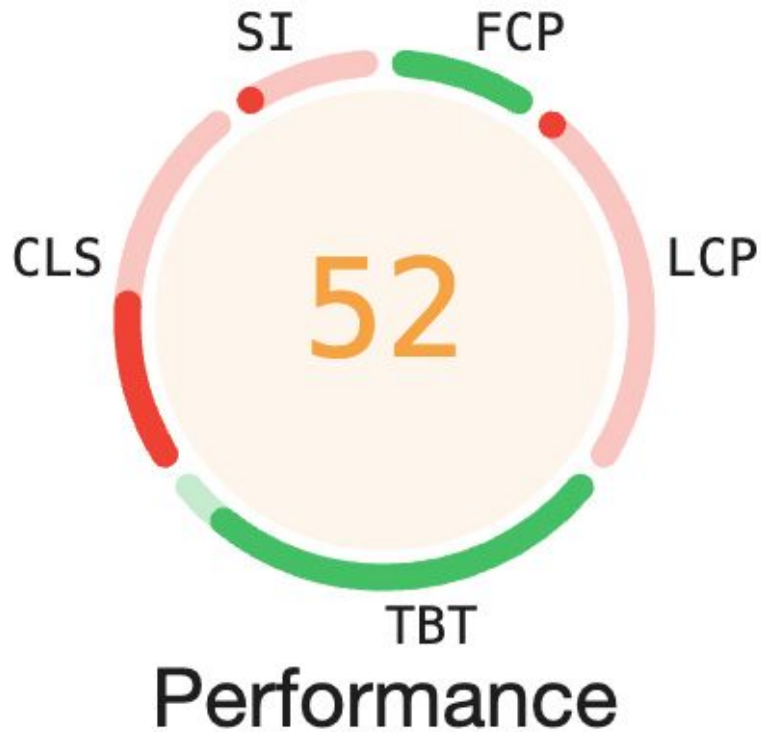
HTTP archives

Time	Response	Req. Size	Resp. Size	Analysis	Total time	Timing
18:21:29.161	200 POST /api/v4/users/status/ids	2557	2067	  	489 ms	 
18:21:29.257	200 POST /api/v4/users/ids	2637	3028	  	429 ms	 
18:21:29.517	200 GET /api/v4/posts/bo8p8kzhmtgq7kddxii87hiwhr/thread	1076	1289	  	289 ms	
18:21:29.517	200 GET /api/v4/posts/tapear8j1irq8fpssszxwak9rh/thread	1076	1333	  	296 ms	
18:21:29.517	200 GET /api/v4/posts/n1n1fdwfaini7rrfa96fdco94o/thread	1076	1217	  	286 ms	

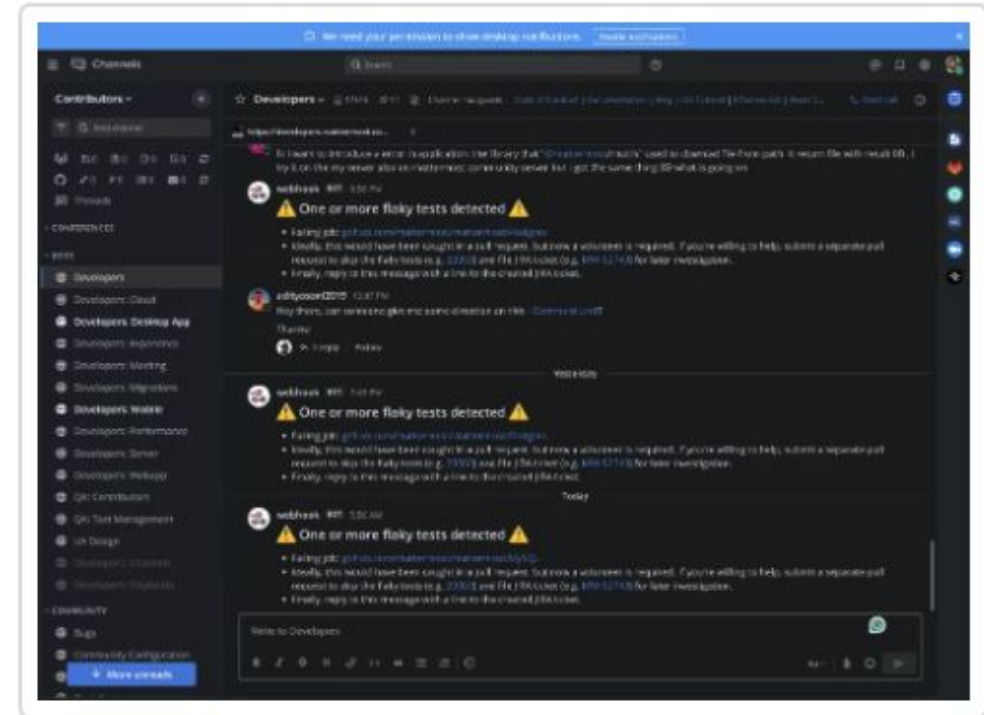
Web vitals



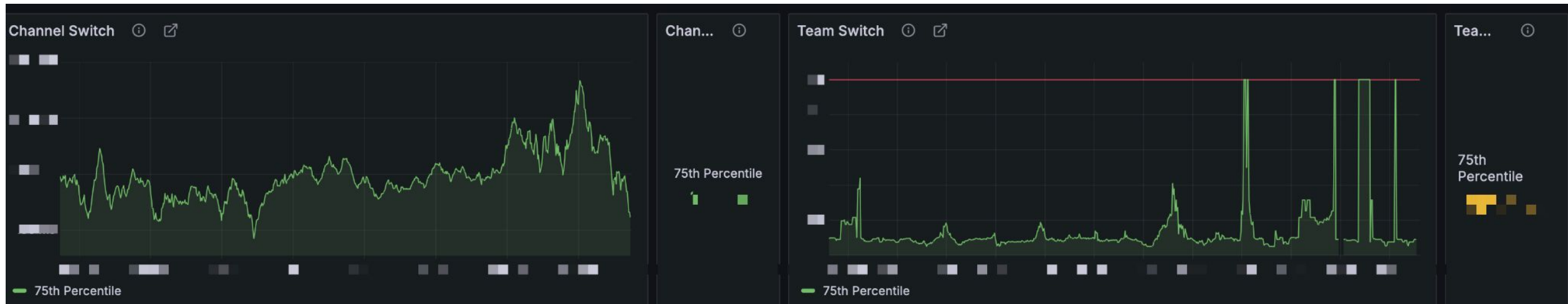
Lighthouse



Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)



Custom app specific metrics



Results from analysis








- <https://github.com/mattermost/mattermost/pull/26670>
- <https://github.com/mattermost/mattermost/pull/26644>
- <https://github.com/mattermost/mattermost/pull/26420>
- <https://github.com/mattermost/mattermost/pull/26662>

... many more checkout github.com/mattermost/mattermost 🙄



Caveats

- Network latency will eat away the hard work. 
- Client side performance measurement has lots of inter dependent factors. 
- Comparison is the key. 
- Manage user expectations regarding performance. 
- Performance improvements are non linear. 



Thank you

بہت شکریہ

شُکراً جَزِيلاً

Connect @ <https://mzubairahmed.com>