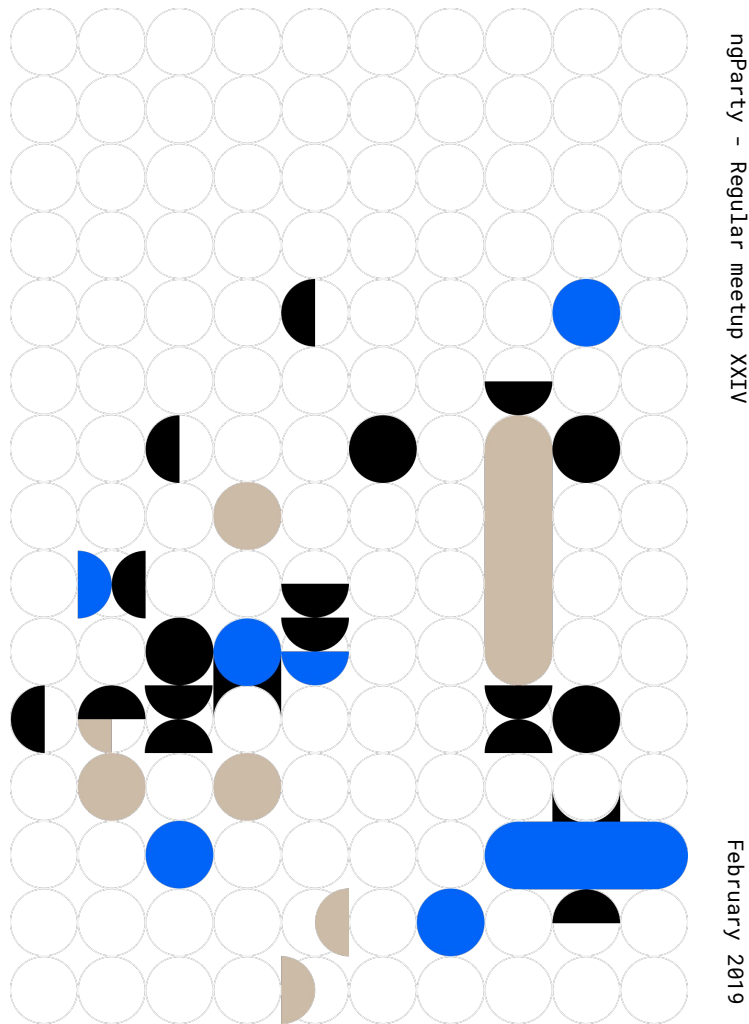


CEAi

React app #webperf tuning

Pavel Gavlík
Product Engineer

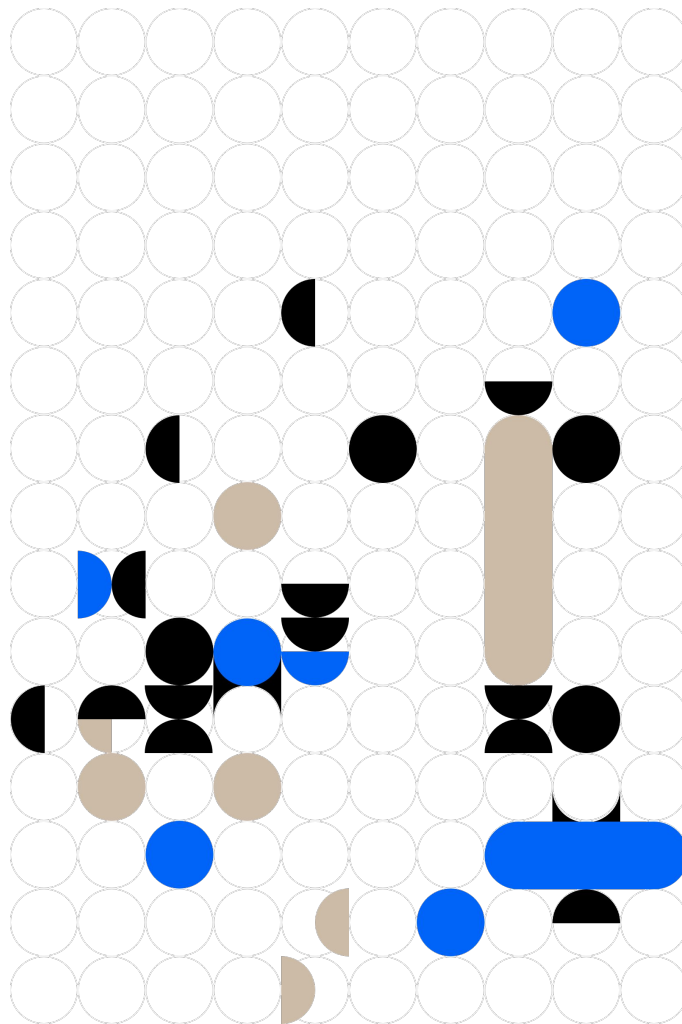


CEAi



React app #webperf tuning

Pavel Gavlík
Product Engineer



Why optimize web performance?

It affects both business metrics & UX

- **How the BBC builds websites that scale**

The BBC found they lost an additional 10% of users for every additional second their site took to load.

- **Page weight matters**

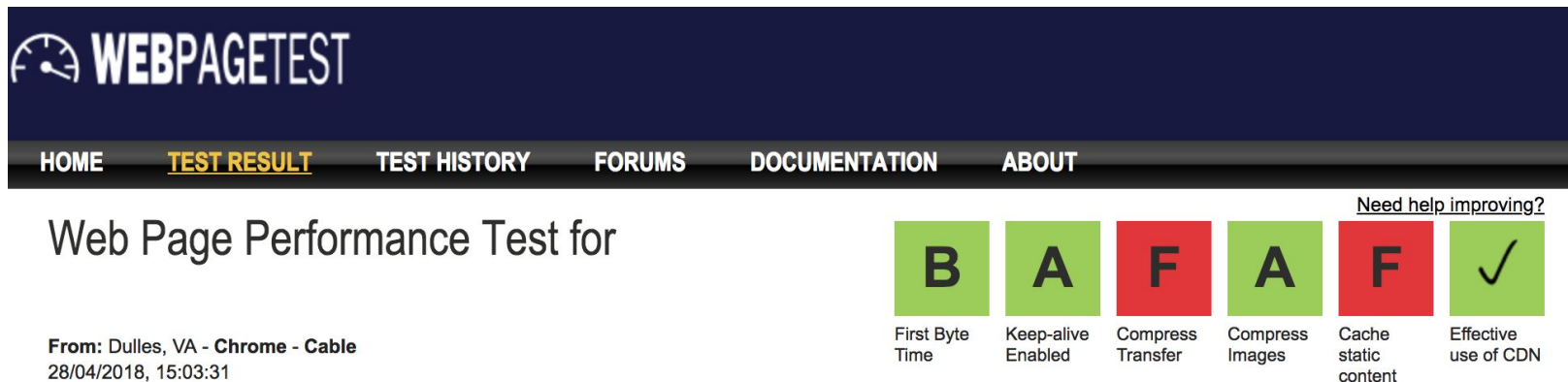
After optimization Youtube became usable for people from Southeast Asia, South America, Africa, and even remote regions of Siberia

- **Merlon Intelligence story**

Global expansion wouldn't be possible without reducing page size

Start With the Obvious

Enable Gzip compression



Start With the Obvious

Enable Gzip compression - Results

- **4X Smaller Static Assets**
- **Up to 3X Faster Page Load**

CEAi

Demo

Source-map-explorer

Load Only What is Necessary

Code splitting and async loading (-0.82M)

3.3M main.195b8edd.js



2.5M main.080ecae7.js

9.2K 4.db886c53.chunk.js

11K 5.a767b89a.chunk.js

3.0K 6.316244b4.chunk.js

2.0K 7.3ab990b1.chunk.js

12K 8.6bc666cf.chunk.js

251K 9.dfb2c210.chunk.js

...

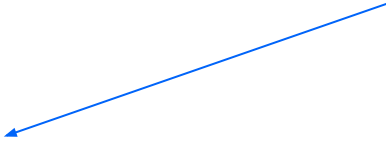
Problem of One Huge File

graphql-document-collector

- [graphql-document-collector](#) has a straightforward CLI interface to precompile all of your GraphQL documents in one definition file.

Problem of One Huge File

graphql-document-collector

- [graphql-document-collector](#) has a straightforward CLI interface to precompile all of your GraphQL documents in **one** definition file.
 - **documents.json • 643.07 KB**
(makes it hard to use code splitting)
- 

Problem of One Huge File

graphql-document-collector → gql imports (-0.61M)



```
-import { withGql } from 'graphql/queries';  
+import { graphql } from 'react-apollo';  
+import FiltersData from './FiltersData.graphql';  
  
...  
  
-const FiltersWithData = compose(withGql('FiltersData'))(Filters);  
+const FiltersWithData = compose(graphql(FiltersData))(Filters);
```

Clean Your Code Regularly

Remove unused packages (-0.32M)

- **react-datepicker** (122K including moment dependency)
- **ion-range slider** (41K + 87K jquery dependency)
- **common** (61K including ramda)

Use Build System, But Check Its Output

Optimized imports (-0.12M)



```
-import R from 'ramda';  
+import { lensPath, set } from 'ramda';  
  
-import omit from 'lodash/omit';  
+import { omit } from 'ramda';  
  
-import { has } from 'lodash';  
+import has from 'lodash/has';
```

Other Work

- **Optimize & inline SVG logo**
 - <https://github.com/jakearchibald/svgomg>
- **Preloading fonts**
 - not very noticeable impact
- **Warm up GraphQL client cache**
 - prefetching of query in critical path

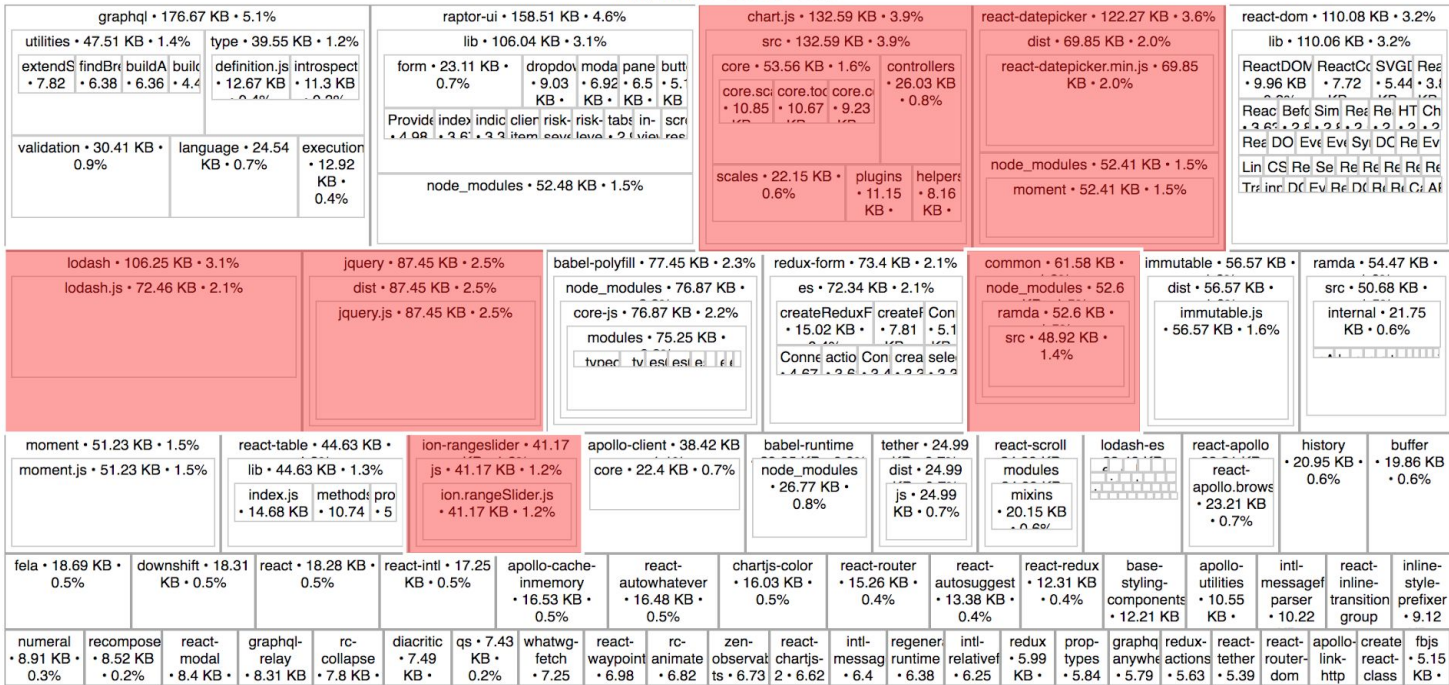
CEAi

Demo

Source-map-explorer
(after webperf optimizations)

.. • 2.3 MB • 67.0%

node_modules • 2.03 MB • 59.2%



static • 269.14 KB • 7.8%

js • 269.14 KB • 7.8%

graphql • 643.53 KB • 18.7%
documents.json • 643.07 KB • 18.7%

packages • 395.69 KB • 11.5%

NMS • 160.78 KB • 4.7%

pages • 118.47 KB • 3.4%

components • 33.23 KB • 1.0%

he

Desktop • 137.14 KB • 4.0%

pages • 134.08 KB • 3.9%

ListManagement • 42.49

pages • 39.64 KB • 1.2%

Acknowledged • 23.01 KB • 0.7%

Triage • 16.61 KB • 0.5%

Terms • 10.7 KB • 0.3%

components • 47.25 KB • 1.4%

form • 14.65 KB • 0.4%

Dropd • 5.98 KB • 0.2%

data.js • 12.0 KB • 0.4%

helpers • 3 KB • 0.3%

reduc • 10.0 KB • 0.3%

Final Results

3.44M → 1.58M

2x Smaller Main Bundle

4x Smaller Using Gzip

Final Results

3.44M → 1.58M

2x Smaller Main Bundle

4x Smaller Using Gzip

Final Results

3.44M → 1.58M

2x Smaller Main Bundle

4x Smaller Using Gzip

What Could Work in the Future?

- [Bundlesize](#) to keep bundle size within budget
- HTTP/2 / Brotli compression
- React → Preact
- Remove graphql package (Is it necessary for apollo-client?)

Next Steps

Learn more from experts:

[Ilya Grigorik](#), [Tim Kadlec](#), [#webperf](#)

Don't guess, measure!

Pavel Gavlík
Product Engineer

