Blocking Ads at *Scale*

Efficient Infrastructure at **Cliqz** and **Ghostery**

Rémi Berson

September 27th, 2019

Cliqz, Ghostery

Summit 2018

New adblocker architecture:

- *fast* decision time (~*0.01 ms* per request)
- *low memory* usage (5.6 MB to 7.7 MB for **170k** filters)
- *compact* memory representation (typed arrays)
- fast serialization and *deserialization* (0.1-*20 ms*)
    - can be stored in IndexedDB for later faster initialization!

And pure JavaScript implementation[1]!

[1] performance mostly comes from efficient data-structures and algorithms

Where Do We Go Now?

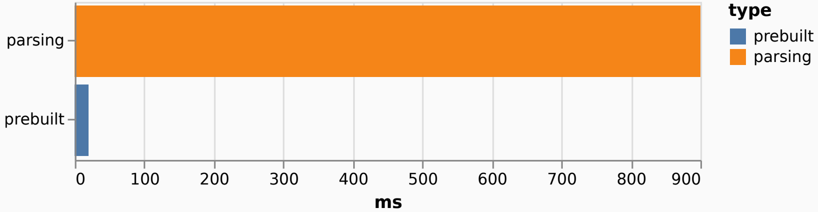
Operating adblocker is really fast, what about *initialization*?

Naive approach:

0. [server] serve lists of filters from CDN (e.g. *EasyList*)
1. [client] *download* assets (network cost)
2. [client] *parse* strings (CPU cost)
3. [client] *initialize* in-memory structures (memory intensive)

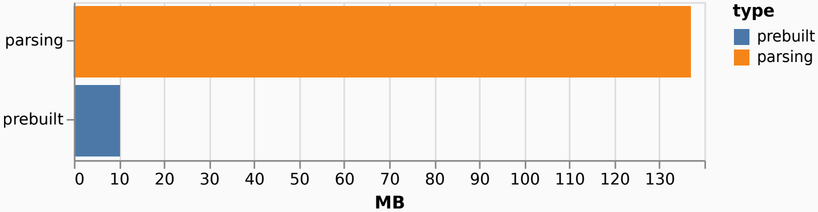
Initialization: Speed

Initialization is CPU intensive.



Initialization: Memory

Initialization is memory intensive.



Engine binary blob of 4.8 MB.

Server Side Building

What if server builds and distributes serialized adblocker?

1. [server] build once and serialize
2. [server] serve via CDN (binary blob)
3. [client] do not pay the CPU and memory cost!

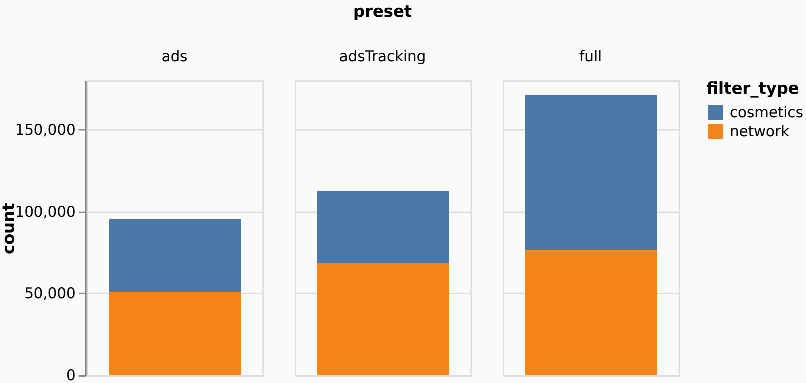
Initial start can benefit from *embedded engine*.

Presets

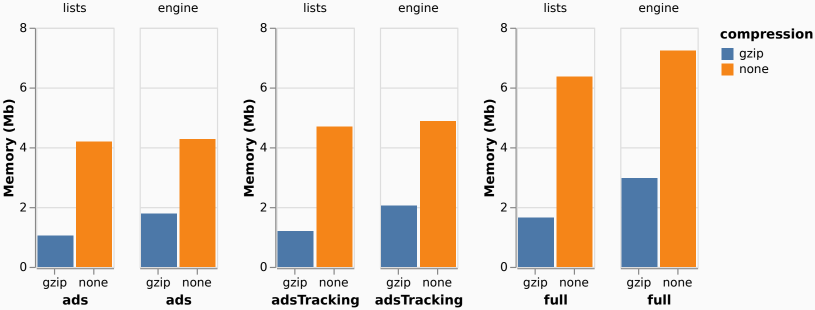
Presets for blocking:

- ads only
- ads + trackers
- ads + trackers + annoyances

Presets: Number of Filters



Presets: Size



- Small network cost for big speed-up memory and savings
- Engine compresses less but there is room for improvement

Updates?

- How do clients **update** (added and removed filters)?
- How to **customize** with extra lists (e.g. regional)?

Adblocker can be updated with added filters!

Better Updates: Diffs

- Still wasteful to download redundant data
- What about downloading what changed instead?
- From `checksum1 -> checksum2`

```
cdn.cliqz.com/lists/{name}/{checksum1}/diffs/{checksum2}
```

Adblocker can be updated with *added* and removed filters!

Diffs: Sizes

- More than *200* days since builder in prod'
- EasyList is most active (7953 added, 8466 removed)
- From 1 to 1126 lines changed in single day
- **Max 1-2%** of total number of filters!

Diffs: EasyList

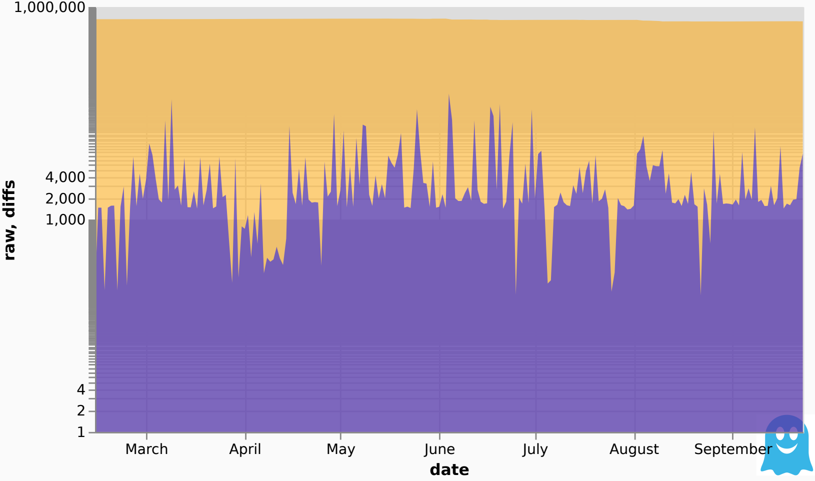
For EasyList:

- Naive: $\sim$244 MB per year, per user.
- Diffs: $\sim$**1.9** MB per year, per user (*x128 less!*)

Especially beneficial for mobiles.

What Gives?

Diffs are **100-1000x smaller** than full list!



Cache Control

- `Cache-Control` HTTP headers (`max-age`, `s-maxage`)
- Most resources are immutable (`immutable` directive)
- Only `index` can change (`max-age=3600`, `s-maxage=86400`)

`https:`*//cdn.cliqz.com/lists/{name}/{checksum}/list.txt*

In a Nutshell

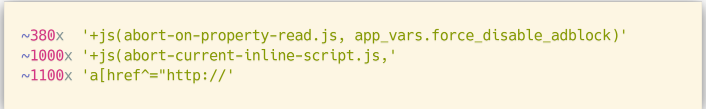
We serve three kinds of resources through the CDN:

1. raw lists (daily updates = ~16 files)
2. diffs (up to 7 days = ~112 files)
3. serialized engines (all *versions* of the library x *presets*)
4. small *index* with links to resources

```
16 x lists + 112 x diffs + 96 x engines = 224 files
```

Going Further

Raw strings take **50%** of total size + are highly redundant.



```
~380x  '+js(abort-on-property-read.js, app_vars.force_disable_adblock)'
~1000x '+js(abort-current-inline-script.js,'
~1100x 'a[href^="http://'
```

- compress specific cases? (e.g.: trie, group similar)
- compress all strings somehow?

We need the ability to get *individual filters* lazily.

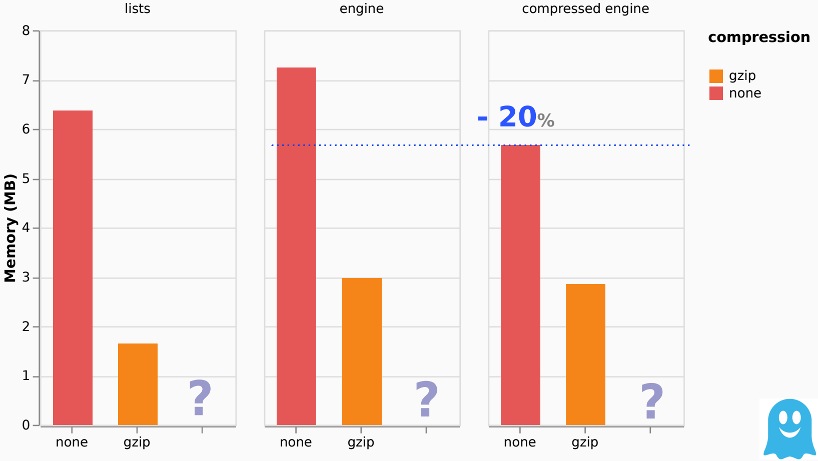
On The Fly Compression

“small strings compression”

- variant of *smaz* (i.e. “tsmaz”)
- transparent at the *DataView* layer
- compression ratio of ~**50%** on strings
- global memory reduction of **20-25%**
- no measurable impact on matching speed!

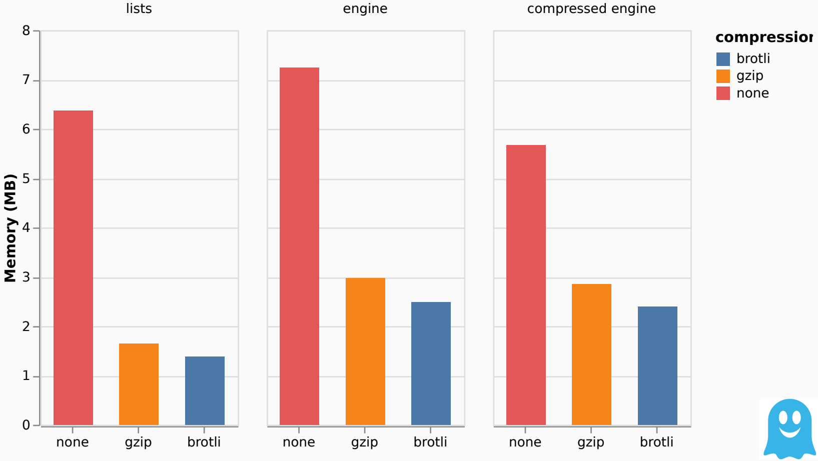
On The Fly Compression: Savings

String compression brings extra **20%** *memory usage reduction*.



Better Assets Compression

Brotli reduces size by **13-18%** compared to *gzip*.



Benefits: clients

- *less* network cost (blob downloaded once then diffs)
- *less* CPU (no parsing, only updates)
- *less* memory used
- *faster* initialization!

Benefits: server

- *equivalent* total data volume
- will be less once brotli + strings compression are enabled
- expected **20%** extra size reduction (CDN)

Cost: more complex system to maintain.

What's Next?

- **Cliqz** is hiring!
- Ping me: remi@cliqz.com
- Twitter: **@Pythux**

— github.com/**cliqz-oss**/**adblocker**

- npm @cliqz/adblocker-webextension
- npm @cliqz/adblocker-electron
- npm @cliqz/adblocker-puppeteer

Questions?

- **Cliqz** is hiring!
- Ping me: remi@cliqz.com
- Twitter: **@Pythux**

— github.com/**cliqz-oss**/**adblocker**

- npm @cliqz/adblocker-webextension
- npm @cliqz/adblocker-electron
- npm @cliqz/adblocker-puppeteer