Project tracks

Ivano Malavolta

VRIJE UNIVERSITEIT AMSTERDAM

LOOKING FURTHER

WARMLY Suggested reading

Home | Blog | Docs | Tools | JS/Wasm features | Research

The cost of JavaScript in 2019

Published 25 June 2019 · Tagged with internals parsing

Note: If you prefer watching a presentation over reading articles, then enjoy the video below! If not, skip the video and read on.

"The cost of JavaScript" as presented by Addy Osmani at #PerfMatters Conference 2019.

One large change to the cost of JavaScript over the last few years has been an improvement in how fast browsers can parse and compile script. **In 2019, the dominant costs of processing scripts are now download and CPU execution time.**

https://v8.dev/blog/cost-of-javascript-2019

Other interesting presentations: https://www.youtube.com/user/estellevw/videos

Tranco list

https://tranco-list.eu

Technical priority tasks (for all)

1. **Check if you are able to apply the treatments to your subjects**
   > Are you able to programmatically modify the index.html of a web app?
   > Are you able to run the ML algorithms in a larger pipeline?
2. **Check if you can apply the treatments to your co-factors**
   > Are you able to throttle the network?
   > Are you able to execute the subjects in both Chrome and Firefox?
3. **Check if you can execute the experiment**
   > Are you able to programmatically switch between the WebGL and WebGPU versions of a web app?
   > Are you able to execute a prefixed usage scenario in your apps?
4. **Check if you can collect all the metrics of your experiment**
   > Are you able to properly collect the page load time of a web page?
   > Do the energy measures you collected make sense?

   Do the math!

VU

Team assignments

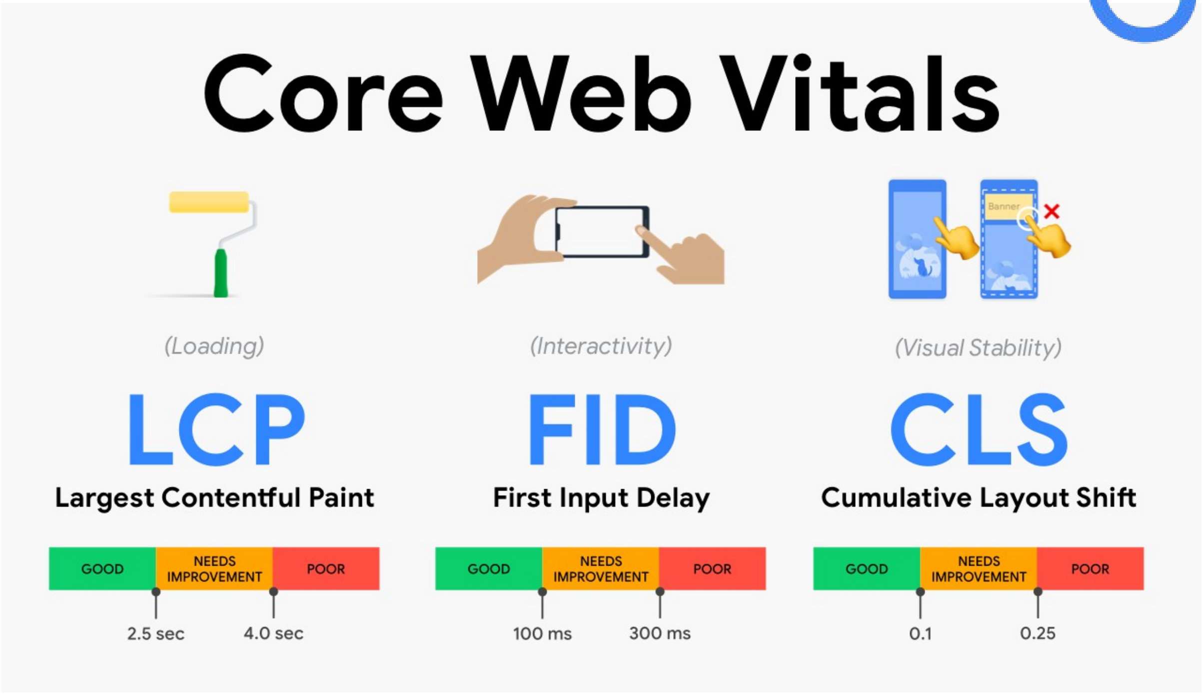
1. Core Web Vitals and energy
2. Android Location APIs
3. Greener ML software - Desktop
4. Greener ML software - GPU
5. ML - anonymized
6. ML - synthesized

0 – Core Web Vitals

Informal RQ

Do Core Web Vitals metrics correlate with energy consumption on mobile web apps?

In collaboration with Google



0 – Core Web Vitals

A previous study have found a negative correlation between Lighthouse performance scores and the energy requirements of mobile websites

However, **little is known about how CWVs correlate with energy consumption**

This study aims to answer the following (informal) research questions:

- Is there a correlation between Core Web Vitals and mobile web app energy consumption?
- If so, what is the strength and direction of this correlation?
- How does each CWV impact energy consumption individually?

0 – Core Web Vitals

- **Factors**:
  - Google Chrome – fixed factor
    - but depending on the complexity of your design, you might add others like Mozilla Firefox
  - device type (low-end VS high-end)
  - Available bandwidth
    - you can use [throttle-proxy]() or the [Charles]() proxy for this
- **Measurement**: study in details and try beforehand the JS library for collecting the metrics ([https://github.com/GoogleChrome/web-vitals]())
  - Collect all metrics (raw values)

```
interface Metric {
  /**
   * The name of the metric (in acronym form).
   */
  name: 'CLS' | 'FCP' | 'FID' | 'INP' | 'LCP' | 'TTFB';

  /**
   * The current value of the metric.
   */
  value: number;

  /**
   * The rating as to whether the metric value is within the "good",
   * "needs improvement", or "poor" thresholds of the metric.
   */
  rating: 'good' | 'needs-improvement' | 'poor';
```

Tranco list

- **Dataset**: randomly sample (~50 websites?)
- **Readings**:
  - Core Web Vitals (CWVs)

1 – Android Location APIs

Informal RQ

How can developers access Android location APIs more efficiently?

In collaboration with

UNIVERSITY OF ALBERTA

- Android guides to read in depth:
  - Build location-aware apps
  - Optimize location for battery
  - Improving urban GPS accuracy for your apps
- You will need to implement a microbench app:
  - Single app with different configurations
  - Each configuration accessing user's location with a different strategy

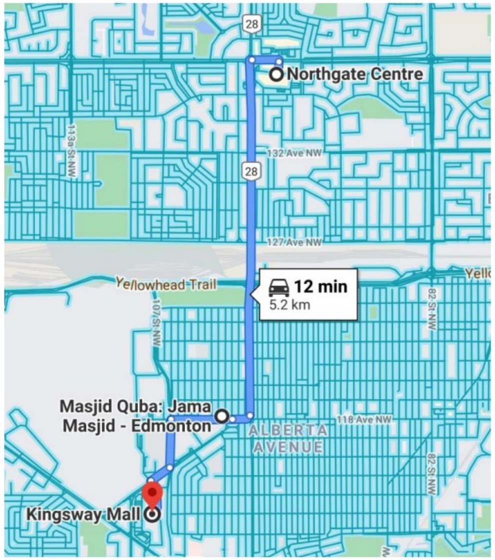


Fig. 3: A path for a user traveling from Northgate Centre to Kingsway Mall.

1 – Android Location APIs

In collaboration with
UNIVERSITY OF ALBERTA

2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)

Energy Efficient Guidelines for iOS Core Location Framework

Abdul Ali Bangash*, Daniil Tiganov†, Karim Ali‡ and Abram Hindle§
Department of Computing Science, University of Alberta
Edmonton, AB, Canada
Email: *bangash@ualberta.ca, †tiganov@ualberta.ca, ‡karim.ali@ualberta.ca, §abram.hindle@ualberta.ca

...e and Evolution (ICSME) | 978-1-6654-2882-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSME52107.2021.00035

***Abstract*—Several types of apps require accessing user location, including map navigation, food ordering, and fitness tracking apps. To access user location, app developers use frameworks that the underlying platform provides to them. For the iOS platform, the Core Location framework enables developers to configure various services to obtain user location information. But how does a particular configuration affect the energy consumption of an app? The available Core Location framework documentation is insufficient to help developers reason about the tradeoff between choosing a particular configuration and energy consumption.**

In this paper, we present a set of guidelines that will help developers make an energy-efficient design choice while configuring the Core Location framework for their app. To achieve that, we have created microbenchmark configurations of the various services that the Core Location framework provides. We have then run several test-scenarios on these configurations to extract their energy profiles. To extract energy-efficient guidelines for developers, we have carefully examined those energy profile results. The guidelines show several configurations that not only reduce energy consumption but also access locations more frequently than other configurations. To evaluate those guidelines, we analyzed three real-world apps and a location service sample app provided by Apple. Our results show that the guidelines help reduce energy: 0.42% for a property search app, 10.59% for a weather app, 26.91% for a location utility app, and 11.37% for Apple's sample app. Additionally, our empirical evaluation shows that choosing an energy-hungry configuration can increase the energy consumption by up to a maximum of 23.97%. Our guidelines are effective on 3 real-world apps, and our methodology may be used to extract energy-efficient guidelines for frameworks other than the Core Location framework.

***Index Terms*—software energy consumption, developers guide, iOS development, smartphone apps**

efficient guidelines for developers. A
to focus on reducing the energy cons
networking, location, and graphics [
ticularly important for location servic
use of location information in variou
turn map navigation, social networki
ordering, carpooling, and vehicle hiri
location information may prevent the
sleep mode, which keeps the location

To access user location, Apple p
the Core Location framework with mu
Standard Location Service, Significant
Location Service, and Regional Mo
service is configurable to access user
frequency and accuracy. This frequen
at the cost of battery life through energ
life and accuracy is an engineering tra
oper needs to make while developing
the current Apple developer's guide
information about such tradeoff [10].
most energy-efficient service but it d
about which services are most energy

*"Always choose the most power-effi
the needs of your app. Disable loca
do not need the location data offer
example, you might disable location
is in the background and would not u*
–AppleInc [11] (Core Location Documentation)

Methodology: replication of a controlled experiment on iOS

Paper: [PDF]

Dataset: https://github.com/themaplelab/iosCoreLocationEnergy

2 – Greener ML software - Desktop

Informal RQ

How to improve ML software from an energy point of view?

https://github.com/IntelPython/scikit-learn_bench

README.md

Machine Learning Benchmarks

Azure Pipelines succeeded

Machine Learning Benchmarks contains implementations of machine learning algorithms across data analytics frameworks. Scikit-learn_bench can be extended to add new frameworks and algorithms. It currently supports the scikit-learn, DAAL4PY, cuML, and XGBoost frameworks for commonly used machine learning algorithms.

Follow us on Medium

We publish blogs on Medium, so follow us to learn tips and tricks for more efficient data analysis. Here are our latest blogs:

- Save Time and Money with Intel Extension for Scikit-learn
- Superior Machine Learning Performance on the Latest Intel Xeon Scalable Processors
- Leverage Intel Optimizations in Scikit-Learn
- Optimizing CatBoost Performance
- Intel Gives Scikit-Learn the Performance Boost Data Scientists Need
- From Hours to Minutes: 600x Faster SVM
- Improve the Performance of XGBoost and LightGBM Inference
- Accelerate Kaggle Challenges Using Intel AI Analytics Toolkit
- Accelerate Your scikit-learn Applications
- Optimizing XGBoost Training Performance
- Accelerate Linear Models for Machine Learning
- Accelerate K-Means Clustering
- Fast Gradient Boosting Tree Inference

2 – Greener ML software - Desktop

https://github.com/IntelPython/scikit-learn_bench

Benchmark supported algorithms

algorithm	benchmark name	sklearn (CPU)	sklearn (GPU)	daal4py	cuml	xgboost
DBSCAN	dbscan	✅	✅	✅	✅	❌
RandomForestClassifier	df_clfs	✅	❌	✅	✅	❌
RandomForestRegressor	df_regr	✅	❌	✅	✅	❌
pairwise_distances	distances	✅	❌	✅	❌	❌
KMeans	kmeans	✅	✅	✅	✅	❌
KNeighborsClassifier	knn_clsf	✅	❌	❌	✅	❌
LinearRegression	linear	✅	✅	✅	✅	❌
LogisticRegression	log_reg	✅	✅	✅	✅	❌
PCA	pca	✅	❌	✅	✅	❌
Ridge	ridge	✅	❌	✅	✅	❌
SVM	svm	✅	❌	✅	✅	❌
train_test_split	train_test_split	✅	❌	❌	✅	❌
GradientBoostingClassifier	gbt	❌	❌	❌	❌	✅
GradientBoostingRegressor	gbt	❌	❌	❌	❌	✅

2 – Greener ML software - Desktop

Main steps before running the experiment:

1) Be able to execute all ML algorithms across all ML frameworks (with timestamps!)
   - Tool: https://github.com/powerapi-ng/pyJoules
2) Define the plan of interventions

   - aka the settings/configurations that you want to compare

   - see here for some examples on scikit-learn (but there are more – get creative!)

NOTE: this experiment can be run on either:

- a **board** (--> IoT flavour to your research) or

- one of our **servers** in the Cloud (--> datacenter flavour to your research), it is up to you to

choose which flavour you want to give to your project

Relevant reading:

- Video: https://www.dropbox.com/s/ak4bl3f9mf0h65q/video1065367371.mp4?dl=0
- Paper: https://stefanos1316.github.io/my_curriculum_vitae/GKSSZ22.pdf

3 – Greener ML software - GPU

Same as the previous track, but the ML algorithms will run on a GPU, instead of a Desktop/laptop

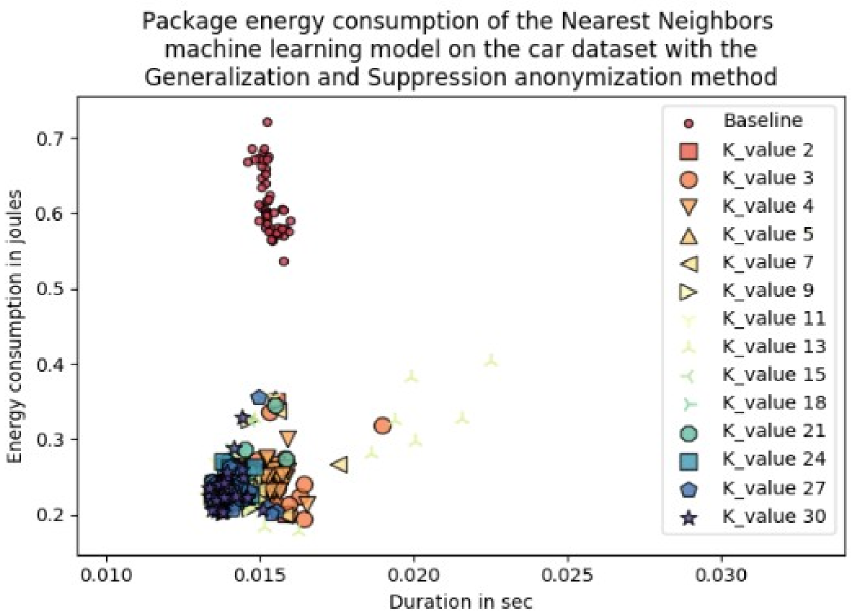
4 - ML training on anonymized datasets

What is the impact of k-anonymizing datasets on the energy efficiency (**and accuracy**) of ML algorithms?

- Use repository with datasets https://archive.ics.uci.edu/ml/datasets.php (beta site https://archive-beta.ics.uci.edu)
- Datasets features:
  - Multivariate
  - Default task – what the dataset was originally used for
    - Classification, Regression
  - Attribute types
    - Categorical, Numerical, Mixed
  - What is the impact of the number of attributes?
    - < 10, 10 to 100, > 100
- Machine learning algorithms:
  - Random forest
  - kNN

In collaboration with
Ana Oprescu, Zoltan Mann

5 - ML training on k-anonymized datasets



https://ieeexplore.ieee.org/abstract/document/9830083

- K-anonymization
  - Microaggregation better energy efficiency
  - Generalisation and suppression better ML accuracy, as long as hierarchy is well constructed

- Technology
  - ARX to create k-anonymised versions of the datasets
  - Code using generalisation and suppression: https://github.com/PepijndeReus/ThesisAI/tree/main/Anonymisation

k-value
3
10
27

5 - ML training on synthesized datasets

What is the impact of data synthesis techniques on the energy efficiency (**and accuracy**) of ML algorithms?

- Use repository with datasets https://archive.ics.uci.edu/ml/datasets.php (beta site https://archive-beta.ics.uci.edu)
- Datasets features:
  - Multivariate
  - Default task – what the dataset was originally used for
    - Classification, Regression
  - Attribute types
    - Categorical, Numerical, Mixed
  - What is the impact of the number of attributes?
    - < 10, 10 to 100, > 100
- Machine learning algorithms:
  - Random forest
  - kNN

5 - ML training on synthesized datasets

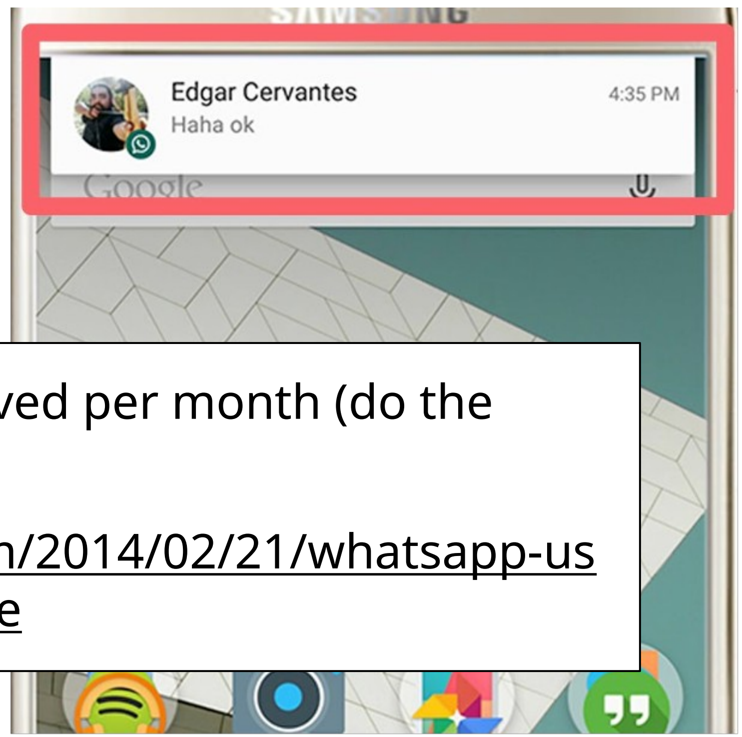
	Avg. time (s)	Avg. CPU (J)	Avg. DRAM (J)	Total Joules used
Preprocessing	0.550 ±0.067	15.090 ±0.654	2.014 ±0.11	17.104
Generation of data	67.478 ±0.287	3584.957 ±24.995	276.832 ±1.066	3861.788
k-nearest neighbours	7.748 ±0.032	274.949 ±2.26	40.734 ±0.144	315.683
Logistic regression	1.282 ±0.105	60.097 ±5.285	6.182 ±0.514	66.279
Neural network	9.767 ±0.076	132.844 ±3.286	23.968 ±0.674	156.812

Energy consumption for ML training on the synthetic Adult data set [1]

- Technology
  - DataSynthesizer to synthesize the datasets
    https://github.com/PepijndeReus/ThesisAI/tree/main/Synthetic_data

6 – Messaging

What is the impact of receiving different types of messages in Instant Messaging apps on the energy consumption of Android devices?



2267 messages received per month (do the math)

https://mashable.com/2014/02/21/whatsapp-user-chart/?europe=true

In collaboration with

UNIVERSITÀ DEGLI STUDI FIRENZE

6 – Messaging

- **Methodology:**
  - A replication of this study, but with the following differences:
    - focus on sending various types of messages (text, audio, video, etc.)
    - Target these apps (source):
      - WhatsApp
      - WeChat
      - Facebook Messenger
      - Telegram
      - Signal
        - this is not used much, but it is open-source → it allows you to "open the hood" (also Telegram)
- **Priority-1 task**: check if you are able to programmatically send messages to all the targeted apps
- Interesting related work: https://eprint.iacr.org/2023/071.pdf

GO!