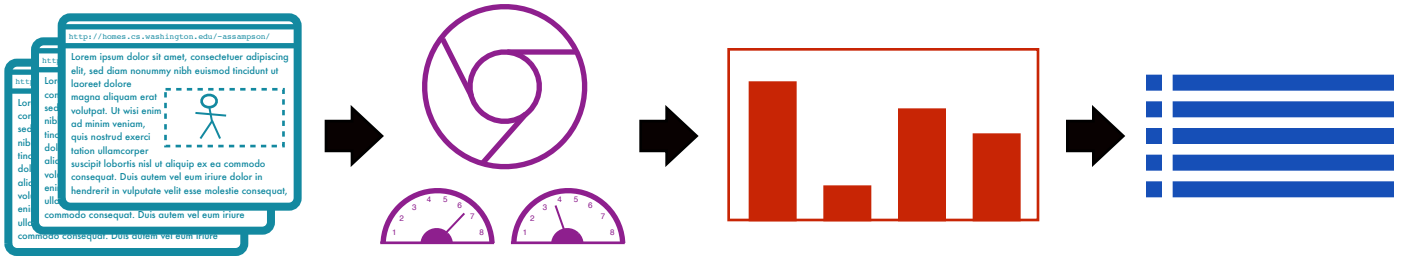
Automatic Discovery of Performance and Energy Pitfalls in HTML and CSS

WebChar answers the questions: *What makes some Web pages slower than others? Why do some sites seem to guzzle battery life while others sip it?*



Web Pages

We start with a set of snapshots of a large set of popular Web pages. WebChar analyzes Web content in the aggregate to look for common trends in real-world use.

WebChar Analysis

WebChar observes browser behavior to measure the time and energy taken to render each page. It also extracts content features.

Performance and Energy Model

Using a machine learning technique, WebChar correlates page features with performance and energy. The model infers browser behavior based on a page's features.

Hypotheses

WebChar mines the model to produce a set of verifiable hypotheses about the content factors that lead to poor performance and energy efficiency.

We used an **HTTP record-and-replay system** to capture popular pages for analysis. Then, we instrumented each page to measure its page load time in an unmodified browser. We measured browser behavior on two resource-constrained systems: **an Atom netbook and an Android smartphone**. For the smartphone, we replaced the battery with an external power supply that measured voltage and current during page load.

WebChar learns a function that relates a vector of feature values (e.g., the frequency of `<IMG>` tags) to the resulting render time and energy consumption. We mine the model to find features that tend to correlate with bad browser behavior. These are WebChar's **hypotheses**.

Browser developers can use hypotheses to identify optimization targets. **Content developers** use them to write faster, more energy-efficient HTML and CSS.

Laying out **tables** can be expensive on the netbook. Unless tabular data must be displayed, content developers should avoid placing content into tables.

Background fills cost significant energy on Android. Even when pages do not load slowly, they can spend a large amount of energy just drawing backgrounds.

On both platforms, **"floating" layout** is expensive. Since modern Web page layouts often use floats for structure, browser implementors should optimize for this common pattern.

The CSS **opacity controls** carry a significant performance impact, especially on mobile. Mobile browser developers should investigate using hardware-accelerated compositing.

sampa QUALCOMM

Adrian Sampson
Călin Caşcaval
Luis Ceze
Pablo Montesinos
Dario Suarez Gracia
IISWC 2012

Download the WebChar source code:
http://sampa.cs.washington.edu/sampa/WebChar