

ブラウザ経由による内部ストレージ種別の推定手法の提案と実装

種岡 優幸† 石川 貴之‡ 細井 理央‡ 高橋 和司‡ 安田 昂樹‡
 田邊 一寿† 細谷 竜平† 齋藤 孝道†
 明治大学† 明治大学大学院‡

1. はじめに

HTTP ヘッダや JavaScript から採取可能な情報を組み合わせて端末を識別する Browser Fingerprinting という手法がある。海外の Web 広告事業者の一部はこの手法を用いて端末を識別している。ユーザーエージェント文字列のような端末の識別に利用される情報のいくつかは時間経過に伴い変化することが先行研究[1]により示された。ゆえに、端末の長期的な識別を行うためには、時間経過による値の変化が生じにくいハードウェアに依存する情報を得ることは必要である。先行研究[2][3]では、様々なハードウェア情報の取得手法を提案している。本論文では、HTML5 の Web Storage API[4]を利用し、ハードウェアに依存する新たな情報として端末内のストレージ種別が HDD か SSD かを推定する手法を提案する。

2. Web Storage API

Web Storage API はキーと値のペアをブラウザに保存する仕組みである。Web Storage には sessionStorage と localStorage の 2 種類があるが、本論文では、localStorage を利用する。データの保存は setItem(), データの読み出しはキーを引数とした getItem()を用いる。

3. 提案手法

3.1. 読み込み時間による推定手法

一般に、ストレージへのアクセス速度は HDD よりも SSD の方が速い。そこで、localStorage での読み込み時間も同様に SSD の方が速いと推測し、読み込み時間の差を利用してストレージの識別を行う。次に示す手順によりブラウザ経由での内部ストレージ種別の推定を行う。

(A)データの保存

setItem()によりブラウザに対し、文字'a'を 1024×1024 個つなげた 1MB のデータを保存

A Proposal and Implementation of Method to Estimate Internal Storage Type via Browser

†Masayuki TANEOKA ‡Takayuki ISHIKAWA ‡Rio HOSOI

‡Kazushi TAKAHASHI ‡Koki YASUDA

†Kazuhiisa TANABE †Ryohei HOSOYA †Takamichi SAITO

†Meiji University

‡Graduate School of Meiji University

する。

(B)読み込み時間の計測

getItem()により保存した 1MB のデータを読み出す。この際、getItem()の前後で High Resolution Time API を用いて現在時刻を取得し、その差を読み込み時間として計測する(図 1 参照)。

```
var start = performance.now();
localStorage.getItem(...);
var diff = performance.now() - start;
```

図 1 読み込み時間の計測

(C)計測時間の平均化

計測は (B)の計測を 50 回行い、その平均値を推定に用いる読み込み時間とする。

(D)閾値の設定

ストレージへのアクセス速度はブラウザ経由であっても HDD と SSD で差が生じると考えられる。そこで、読み込み時間についての閾値を設定することでストレージ種別を推定する。閾値はアクセスした端末の内部ストレージ種別を正しく推定できる数が最大となるように定める。

3.2. 端末の処理性能を考慮した推定手法

Web Storage での読み込み時間がストレージ性能以外に端末の処理性能に依存する可能性について検討する。Octane 2.0[5]と JetStream 1.1[6]の 2 種類の JavaScript ベンチマークソフトを利用することで端末の処理性能を得る。ベンチマークソフトの結果を利用することで計測した時間から端末の処理性能に依存せず、ストレージ性能の差のみに依存する結果を得ることを狙う。式(1)から算出した値に対して、3.1 節の (D)と同様に閾値を設定する。

$$\text{比率} = \frac{\text{読み込み時間の逆数}}{\text{ベンチマークソフトの各計測項目の値}} \quad (1)$$

4. 評価

4.1. 概要

実験に利用するブラウザとして Chrome と

Firefox を選択した。まず、3 節における提案手法の(A), (B), (C)を行う調査用 Web ページを作成した。読み込み時間は同一端末内の Chrome, Firefox からこの調査用 Web ページへそれぞれアクセスすることで計測した。

全体のサンプル数のうち、内部ストレージ種別を正しく推定した割合を精度として、評価の指標に用いる。

実装される JavaScript エンジンの違いからブラウザごとに処理性能に差異が生じ、読み込み時間にも差が生じた。よって、Chrome と Firefox に対して同一の閾値を設定することが困難である。そこで、Chrome と Firefox に分けて分析を行った。

4.2.読み込み時間による推定手法

Chrome における端末ごとの読み込み時間の関係を図 2 に示す。横軸はアクセスした端末、縦軸は読み込み時間である。端末は読み込み時間の昇順で並べ、黒色が SSD、白色が HDD の端末を示す。また、図中の横の点線は閾値であり、読み込み時間がこの閾値以下の端末を SSD、閾値以上の端末を HDD と推定している。

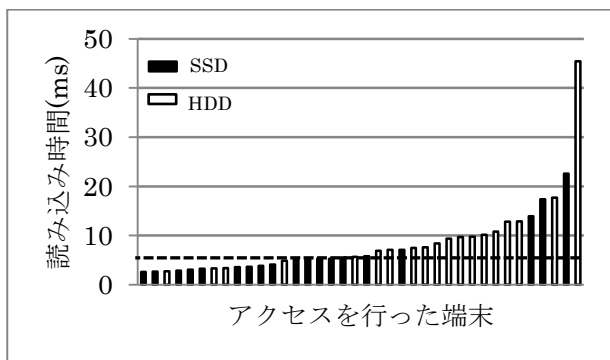


図 2 Chrome における端末ごとの読み込み時間

Chrome のサンプル数 38 のうち内部ストレージ種別を正しく推定できた数は 29 であり、精度は 76.3%であった。誤判定の内訳として HDD を SSD と判定した数が 4, SSD を HDD と判定した数は 5 であった。

Firefox からアクセスした際の端末ごとの読み込み時間の関係を図 3 に示す。Firefox のサンプル数 34 のうち内部ストレージ種別を正しく推定できた数は 27 であり、精度は 79.4%であった。誤判定の内訳として HDD を SSD と判定した数が 2, SSD を HDD と判定した数は 5 であった。

Chrome と Firefox を比較すると、読み込み時間の値自体は大きく異なるが、推定精度はどちらも 80%に近い値となった。

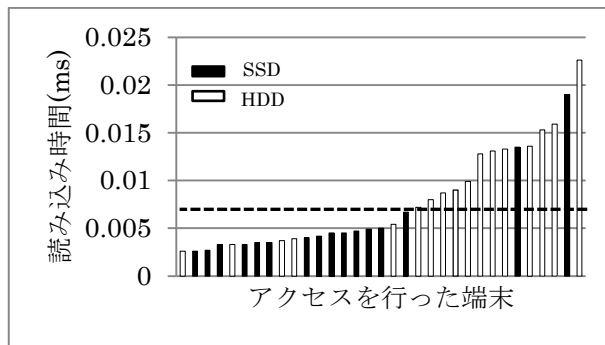


図 3 Firefox における端末ごとの読み込み時間

4.3.端末の処理性能を考慮した推定手法

読み込み時間に対して、3.2 節で示した式(1)の比率による判定をしたところ、4.2 節で示した精度からの向上は確認できなかった。

本節の実験結果を以下の表 2 に示す。

表 2 実験結果

	サンプル数	正判定	HDD 誤判定	SSD 誤判定	精度
Chrome	38	29	4	5	76.3%
Chrome (処理性能を考慮)	38	29	6	3	76.3%
Firefox	33	27	5	2	79.4%
Firefox (処理性能を考慮)	33	27	1	6	79.4%

5.まとめ

本論文では Web Storage API を用いることでブラウザ経路による内部ストレージ種別を推定する手法を提案し調査した。提案手法でストレージを正しく推定できた割合は Chrome で 76.3%, Firefox で 79.4%であった。

6.参考文献

- [1] 磯侑斗, 桐生直輝, 塚本耕司, 高須航, 山田智隆, 武居直樹, 齋藤孝道, “Web Browser Fingerprint を採取する Web サイトの構築と採集データの分析”, コンピュータセキュリティシンポジウム 2014, 2014
- [2] 桐生直輝, 磯侑斗, 金子洋平, 齋藤孝道, “Web Workers を用いた演算処理性能の差による CPU コア数の推定”, コンピュータセキュリティシンポジウム 2014, 2014
- [3] 安田昂樹, 高須航, 山田智隆, 武居直樹, 西倉裕太, 石川貴之, 細井理央, 高橋和司, 齋藤孝道, “Web Browser Fingerprinting 技術を用いた CPU 拡張機能の推定法の提案と実装”, SCIS2016 暗号と情報セキュリティシンポジウム, 2016
- [4] <https://html.spec.whatwg.org/multipage/webstorage.html>
- [5] <https://chromium.github.io/octane/>
- [6] <http://browserbench.org/JetStream/>