1ZA-9

CPU 拡張命令の対応の有無による CPU アーキテクチャの推測

桐生　直輝†　　後藤　浩行‡　　齋藤　孝道†
明治大学†　　明治大学大学院‡

1. はじめに

個人の趣味嗜好によってユーザ側に表示する内容を変更する web サービスが増えてきている．趣味嗜好を判断する材料として，個人の web 閲覧履歴やオンラインショップでの商品購入履歴を利用する方法が存在する．この方法は，ユーザとユーザの行動を関連付けるために，ユーザを特定する必要がある．

ユーザを特定する手法として，ユーザのマシン上に保存された cookie などに格納された識別子を用いる手法が存在するが，近年のブラウザには cookie を削除，またはブロックする機構が存在する．

その一方で，cookie などのストレージを用いることなく，ユーザのマシンの特徴点を用いることでユーザを特定する手法が，Peter Eckersley[1]によって，提案されている．

本論文では，新たな特徴点としてユーザのマシンの CPU アーキテクチャを，拡張命令の有無によって推測する手法を提案し，実装を示す．

2. web 行動追跡

2.1 概要

web 行動追跡とは，web サービス提供者が，ユーザの過去のページ閲覧履歴や商品購入履歴などの情報を得るための行為である．

本論文では，識別子を用いる手法を確定的トラッキング，ユーザの特徴点を用いてユーザを推定する手法を推定的トラッキングと呼ぶ．

2.2 確定的トラッキング

閲覧したページに広告画像などが埋め込まれていた際，その広告画像を提供しているサーバとの間で cookie が発行される．その cookie に格納された識別子とユーザが閲覧したページの URL を紐付けることで，広告サーバはユーザの web 閲覧履歴を取得することが可能となる．

ユーザ毎に一意な識別子を割り当てるため，確実にユーザを特定できるが，cookie などを削除されると追跡できなくなる．

2.3 推定的トラッキング

この手法では，HTTP ヘッダやブラウザの plugin，利用可能なフォント，画面解像度などのユーザの特徴点をサーバに送信させ，それらの特徴点情報を基にユーザを推定する．閲覧したページの URL とユーザのマシン情報を紐付けることで行動追跡を行うことが可能となる．

これらの情報は JavaScript や Flash から取得することが可能である．推定的トラッキングは，HTTP cookie などを用いないため防ぐことは難しいが，特徴が類似しているユーザが存在する場合は一意に識別することができない．

3. SSE 拡張命令

3.1 概要

SSE 拡張命令は x86 アーキテクチャの CPU に実装された拡張命令群である．SSE 拡張命令を用いる際は，CPU に用意された SSE 専用の 128 ビットのレジスタのデータに対して転送や演算などの命令を実行する．

本論文の提案手法では，単精度浮動小数点数を扱う SSE と，倍精度浮動小数点数を扱う SSE2 を用いる．

3.2 SSE

SSE は，単精度浮動小数点数演算のための拡張命令である．単精度浮動小数点数とは，32 ビットの浮動小数点数であり，符号部 1 ビット，指数部 8 ビット，仮数部 23 ビットで構成される．SSE 専用レジスタに単精度浮動小数点数を 4 つ格納し，1 命令で同時に処理することが可能である．

An Estimation Method of CPU Architecture by Existence of Support of CPU Extended Instructions.
† Naoki Kiryu
‡Hiroyuki Goto,
†Takamichi Saito
Meiji University(†), Graduate school of Meiji University(‡)

3.3 SSE2

SSE2 は，倍精度浮動小数点数演算のための拡張命令である．倍精度浮動小数点数とは 64 ビットの浮動小数点数であり，符号部 1 ビット，指数部 11 ビット，仮数部 52 ビットで構成される．SSE 専用レジスタに倍精度浮動小数点数を 2 つ格納し，1 命令で同時に処理することが可能である．

3.4 浮動小数点数における演算誤差

浮動小数点数を用いて表現できない値が存在する．例えば，10 進数の 0.1 を 2 進数の固定小数点数として表現すると，0.0001100110011001100…と循環小数になるため，単精度浮動小数点数や倍精度浮動小数点数のようなビット数が有限である形式ではこれを表現することは不可能である．

浮動小数点数は表現不可能な値を取ろうとした際，表現可能な値へ変換されるため，丸め誤差が生じる．

変換の方法は，桁溢れになったビットの最上位のビットが，1 の場合切り上げ，0 の場合切り捨てとなる．ただし，溢れた桁が 1 桁のみだった場合は，変換後の値が偶数になるように，つまり仮数部の最下位ビットが 0 になるように変換される．10 進数の 0.1 の例を用いると，単精度浮動小数点数では，00111101110011001100110011001101 となり，倍精度浮動小数点数では，0000011110111001100110011001…1010 となる．

4. 提案手法

4.1 概要

浮動小数点数には表現不可能な値が存在し，そのような値をとろうとした際，表現可能な値へ変換されるため，誤差が生じる．

単精度浮動小数点数と倍精度浮動小数点数では精度に差があり，SSE2 への対応の有無によって，同じ計算でも結果が異なる場合がある．

SSE2 への対応の有無によって結果に差が生じる JavaScript のコード(後述)を実行させることで，ユーザの CPU アーキテクチャを推測する材料にすることが可能であるとした．

実際，JavaScript のプログラムが実行される際に生成された機械語が，SSE2 の命令を用いているかどうかを，JavaScript エンジンである google v8 で機械語を逆アセンブルすることによって確認した．

4.2 動作

以下にブラウザで実行させる JavaScript コードを示す．

このプログラムでは，0 ラジアンから 10 万ラジアンまでの sin，cos の値を繰り返し計算する．1 ラジアンを意味する変数 rad は，180 を Math.PI で割ることによって算出する．Math.PI は無理数のため，単精度浮動小数点数や倍精度浮動小数点数で正確な値を取ることができず，SSE2 への対応の有無によって計算結果に差が生じる．

SSE2 は Pentium4 以降で実装されているため，これによってユーザの CPU が Pentium4 以降の物かどうかを判別することが可能となる．

```
var a = new Array();
var b = new Array();
var rad = 180/Math.PI;
var sum = 0;
for(var i=0;i<100000; i++){
  a[i]= Math.sin(rad*i);
  b[i] = Math.PI(rad*i);
  sum += a[i]*b[i];
}
```

図 1．JavaScript コード

動作環境

-CPU: Intel Core i7-3770K
-CentOS6.3 32bit
-apache 2.2.15
-browser: google chrome 22.0.1229.94

拡張命令への対応の有無は仮想マシンの設定ファイルから CPUID を変更することで実現した．

SSE2 対応	0.12732268535134178
SSE2 非対応	0.1273226857058138

図 2．計算結果の比較

5. まとめ

ユーザの特徴点を用いる行動追跡手法における，新たな特徴点として，CPU アーキテクチャを推測する方法を提案し，実装した．

演算結果の差によって，CPU の SSE 拡張命令への対応の有無を判断することが可能となった．

6. 参考文献

[1]Peter Eckersley, 2010,
How Unique Is Your Web Browser?
https://panopticlick.eff.org/browser-uniqueness.pdf
[2]Intel 64 and IA-32 Architecture Software Developer's Manual
http://download.intel.com/design/processor/manuals/253665.pdf