4J-07

複数のクロック信号源における相対的なクロック特性の高精度な観測

高井 淳光 † 千川 尚人 † 下馬場 朋禄 ‡ 伊藤 智義 ‡
国立高等専門学校機構 小山高専 † 千葉大学 ‡

1 はじめに

IoT 機器を用いたサービスでは, 利用する機器が公の場に設置されている可能性がある. そのため悪意のある人物による機器のすり替え, 不正利用のリスクが高まるため, これに対応する技術が求められている. 我々はこの課題に対して, 機器が持つ半導体チップが生成するクロック特性を基準装置の特性 (Clock FingerPrints CFP) と比較することで識別する手法を提案している. しかし, この手法では基準装置に高精度な時刻情報の提供能力が必須となることが課題である. そこで, ローカル環境下で機器識別に利用できる特徴量を抽出する手法と機器識別手法を提案する. 本稿では提案する特徴量抽出法と識別手法を示し, その有効性について論じる.

2 既存手法

既存研究 [1] では, コンピュータに搭載されたクロックカウントによって計算されるローカル時刻と基準時刻のずれ (時刻ドリフト) に固有の差が現れることが明らかにしており, 時刻ドリフトの大きさとマイクロプロセッサのコア温度の関係は線形的に近似できることが確認されている. この研究おける識別手法は計測した時刻ドリフトとコア温度の関係に対して回帰直線を求め, その傾きと切片を特徴量として用いて機器の識別を行う. このとき, 基準時刻を提供する時刻基準機には GNSS(Global Navigation Satellite System / 全球測位衛星システム) モジュールを取り付け, 高精度な時刻供給能力を持たせている.

3 提案手法

3.1 提案手法における CFP

Linux カーネルでは, 時刻管理に利用する変数である jiffies や xtime の更新のために Timer Tick と呼ばれるタイマ割り込みが発生, 処理される. 省電力化を目的として更新の頻度を下げる Tickless 機能が存在するが, これを無効化したカーネルでは予め決めた周波数で定期的に割り込みが起こる. この割り込みは, Linux カーネルにおける Clockevents と呼ばれる タイマデバイスを抽象化した機能により引き起こされる. また, 割り込み機能を持たず, 高精度な時刻を提供するための機能として, Clocksource と呼ばれる機能も存在する.

既存研究 [1] によれば各タイマは固有の特性を持つため, 割り込み周波数を f とすれば, Clockevents デバイスごとに $1/f$ 秒のカウントにかかる絶対的な時間は異なる. 加えて, Clocksource デバイスごとにカウントする速度も異なる. したがって, Clockevents デバイスが 1/f 秒のカウントする間の Clocksource デバイスのカウンタの増加量も機器ごとに異なった値が得られる. 本稿ではこの値を CFP 特徴量とする.

3.2 CFP の抽出方法

CFP を抽出するため, Timer Tick の処理を行う tick_periodic 関数に, 現在の TSC (Time Stamp Counter) のカウント値を予め確保したメモリに記録する処理を加える. これにより得られるデータを RawCFP と呼ぶ. RawCFP をユーザプログラムに渡すために, Linux におけるスペシャルファイルを生成するカーネルモジュールを作成する. ユーザプログラムはこのスペシャルファイルから RawCFP を定期的に読み取り, 保存する. 機器 A における p 個の RawCFP は $CFP_A = (c_{A_1}, ..., c_{A_s})$ と表す.

3.3 個体特徴量の算出

抽出した RawCFP から, 割り込み周期当たりの TSC の増加量（CFP 特徴量）を算出する. この増加量を tsc_sum と呼ぶ. 機器 A における p 個の RawCFP を元に計算できる tsc_sum 集合 $tsum_A$ を次のように表す.

$$tsum_A = (c_{A_2} - c_{A_1}, c_{A_3} - c_{A_2}, ...c_{A_s} - c_{A_{s-1}}) \quad (1)$$

m 個の候補機器群を $\boldsymbol{C} = (C_1, C_2, ...C_m)$ とする. 予め, 候補機器群 $\boldsymbol{C}$ に属する機器ごとに, p 個の RawCFP を 2 回計測し, それぞれの tsc_sum 集合を算出する. 候補機器 C_j の 1 つ目の tsc_sum 集合は $tsum_{Cj_1}$ と表し, 2 つ目は $tsum_{Cj_2}$ と表す. これらの tsc_sum 集合をそれぞれ次式のように $\boldsymbol{TG1}$ と $\boldsymbol{TG2}$ にグループ分けする.

$$\boldsymbol{TG1} = (tsum_{C1_1}, tsum_{C2_1}, ..., tsum_{Cm_1}) \quad (2)$$
$$\boldsymbol{TG2} = (tsum_{C1_2}, tsum_{C2_2}, ..., tsum_{Cm_2}) \quad (3)$$

C_j における個体特徴量は, $\boldsymbol{TG2}$ に属する $tsum_{Cj_2}$ と $\boldsymbol{TG1}$ に属する全ての tsc_sum 集合との類似度を要素に持つベクトルとする. この類似度を計算するため, 次式で定義される 2 つの確率密度分布同士の距離を計算できる L^2 距離を導入する. [2]

$$L^2(p, p') = \int (p(x) - p'(x))^2 dx \quad (4)$$

ここで, α から β までの範囲で区間の幅が γ の度数分布を tsc_sum 集合を基に作成し, これを離散的な確率密度分布として捉えて L^2 距離を計算することで, 2 つの tsc_sum 集合の類似度を計算することが出来る. この類似度を $S(\alpha, \beta, \gamma, tsum_A, tsum_B)$ と表す.

また, 区間の幅を c から d ずつ k 回広げていくことで, 得られる類似度を増やすことができ, 個体特徴量を k 次元ベクトルとして得ることが出来る. これを A の B に対する個体特徴量ベクトルと呼び, このベクトルの要素は次のように計算出来る.

$$f_{k_{AB_i}} = S(a, b, c + id, tsum_A, tsum_B) \quad (i = 0, \cdots, k-1) \quad (5)$$

High-Precision Observation Method of Relative Clock Drift in Multiple Clock Signal Sources
†Akihiro TAKAI, †Naoto HOSHIKAWA,
‡Tomoyoshi SHIMOBABA, and ‡Tomoyoshi ITO
†National Institute of Technology, Oyama College
‡Chiba University

次に機器 C_j の $\boldsymbol{TG1}$ 全体に対する機器の個体特徴量を表すため, $\boldsymbol{TG2}$ に属する C_j の tsc_sum 集合である $tsum_{Cj_2}$ と, $\boldsymbol{TG1}$ に属する全ての tsc_sum 集合とそれぞれの間における個体特徴量ベクトルを計算し, これらのベクトルを行ベクトルとする行列を計算する. これを個体特徴量行列 $\boldsymbol{FM}$ と呼ぶ. $tsum_{Cj_2}$ の $\boldsymbol{TG1}$ に対する個体特徴量行列は次のように表すことができる. このとき, a は $\boldsymbol{TG1}$ に属する tsc_sum 集合と $tsum_{Cj_2}$ の中における tsc_sum の最小の値であり, b は最大の値とする.

$$\boldsymbol{FM_j} = \begin{bmatrix} \boldsymbol{f_{k_{C_jC_1}}} \\ \boldsymbol{f_{k_{C_jC_2}}} \\ \vdots \\ \boldsymbol{f_{k_{C_jC_m}}} \end{bmatrix} \tag{6}$$

3.4 機器識別手法

3.4.1 判定係数の算出

ある機器 A, B から得られた 2 つの個体特徴量行列 $\boldsymbol{FM_A}, \boldsymbol{FM_B}$ の近さを示すため, それぞれの行ベクトルの残差平方和 RSS を計算する. 同じ機器の個体特徴量行列同士の残差平方和は小さくなる. この値を判定係数と呼び, 機器の識別に利用する. $\boldsymbol{FM_A}, \boldsymbol{FM_B}$ の判定係数 h_{AB} は次式のように表せる.

$$h_{AB} = \sum_{i=1}^{m} RSS(\boldsymbol{FM_{A_i}}, \boldsymbol{FM_{B_i}}) \tag{7}$$

3.4.2 識別処理

識別対象機器を機器 X としたとき, 機器 X の $\boldsymbol{TG1}$ に対する個体特徴量行列が $\boldsymbol{FM_X}$ ならば, 全ての候補機器の $\boldsymbol{TG1}$ に対する個体特徴量行列と $\boldsymbol{FM_X}$ の判定係数を算出し, その値が最も小さかった候補機器を識別対象機器 X と識別する.

3.5 提案手法の評価実験

候補機器には Intel Core i5 8250U 1.6GHz, RAM 8GB を搭載するパーソナルコンピュータ (以下 PC) 3 台を用いる. これらの PC には全て, Ubuntu 18.04 をインストールし, RawCFP を抽出するための機能を追加した Linux カーネル v4.19.75 に置き換えている. これら 3 台の PC を候補機器 C_1, C_2, C_3 とし, それぞれ識別対象機器 X に設定して判定係数を計算する. この判定係数を元に識別を行う. 類似度を計算する時の定数は $c = 100, d = 100, k = 100$ とする.

4 実験結果

4.1 算出した特徴量行列

X 軸を度数分布の区間, Y 軸を L^2 距離として, 機器 C_1 を機器 X としたときの $\boldsymbol{FM_{C_1}}, \boldsymbol{FM_{C_2}}, \boldsymbol{FM_{C_3}}, \boldsymbol{FM_X}$ の各行ベクトルをプロットすると, 図 1 のような散布図を得ることができる.

4.2 識別結果

機器 C_1, C_2, C_3 の個体特徴量行列と識別対象機器 X を C_1, C_2, C_3 とした時の個体特徴量行列の判定係数を表 1 に示す. 機器 X が C_1 の場合 h_{XC1} が, C_2 の場合 h_{XC2} が, C_3 の場合 h_{XC3} が最小になっている. したがって, それぞれの機器を識別出来ていると言える.

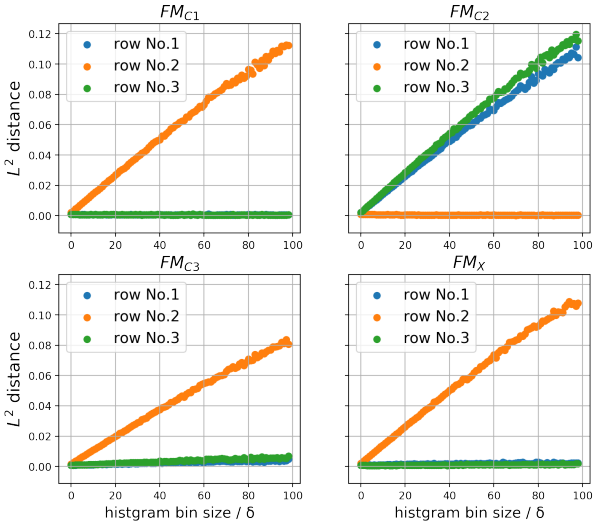

図 1 $\boldsymbol{FM_{C1,C2,C3}}, \boldsymbol{FM_X}$ の各行ベクトル

表 1 各機器における判定係数

識別対象機器 X	h_{XC_1}	h_{XC_2}	h_{XC_3}
C_1	0.0006623	1.3450701	0.0284545
C_2	1.3009506	0.0051635	1.0189870
C_3	0.0335815	1.1118373	0.0003080

5 考察

提案手法における個体特徴量行列は, 候補機器群の数が増えるとサイズが大きくなり, 判定係数を計算する式 (7) の項が増える. 一般的に, 候補機器の数が増えると, 似た特徴を持つ機器が現れる可能性が高くなり, 識別が難しくなる. しかし, この手法では候補機器が増えることで, より多面的に一次特徴量を捉えることができるため, このような状況にも対応できると考えられる.

6 結論・今後の予定

今回提案した手法では, CFP データ抽出が数秒で完了し, PC 3 台の識別に成功した. したがって, 本手法は識別速度において既存手法に対し優位性がある. しかし, 実験に用いた機器が少なく, 候補機器が大きくなった場合の信頼性を示せていない. 今後はこれらの課題に対して検討を進めていく.

謝辞

本研究は矢崎財団（Yazaki Memorial Foundation for Science and Technology）の支援を受けた.

参考文献

[1] 並木涼 , 干川尚人 , 下馬場朋禄 , 伊藤智義. “ ディジタル機器におけるシステム時刻のずれと環境温度の変動との相関性 ”. 電子情報通信学会第 18 回ネットワークソフトウェア研究会 , January 2019.

[2] 杉山将：確率分布間の距離推定:機械学習分野における最新動向, 日本応用数理学会論文誌,Vol.23, No.3 (2013) pp.439-452.