類似度を用いたファイル追跡に関する一手法の提案

高田慎也† 藤木直人† 松村隆宏† 中原慎一† 元田敏浩†

†日本電信電話株式会社 セキュアプラットフォーム研究所
180-8585 東京都武蔵野市緑町 3-9-11 NTT 武蔵野研究開発センタ
{takada.shinya,fujiki.naoto,matsumura.takahiro,nakahara.shinichi,
motoda.toshihiro}@lab.ntt.co.jp

あらまし クラウドに情報を預託する際の不安を払拭するため，我々はファイルI/Oを監視し情報流通を可視化するトレーサビリティ基盤の開発に取り組んできた．ファイルI/O監視により情報を漏れなく把握できる一方，ファイルアクセス時の一時ファイル利用などアプリケーションに依存したファイル操作の推定や，動作環境によるI/O動作の違いの補正が必要など，トレース精度に限界があり，定常的な維持管理を必要とした．本稿では，ファイルトレースにデジタルフォレンジック分野で研究されているファイル類似度を併用する事でアプリケーション依存性を低減し，更にはクリップボードやネットワーク経由の情報トレースにまで応用可能な手法を提案する．

A Method for File Tracing using Similarity

Shinya Takada† Naoto Fujiki† Takahiro Matsumura† Shinichi Nakahara† Toshihiro Motoda†

†NTT Secure Platform Laboratories
3-9-11, Midori-cho, Musashino-shi, Tokyo 180-8585 JAPAN
{takada.shinya,fujiki.naoto,matsumura.takahiro,nakahara.shinichi,
motoda.toshihiro}@lab.ntt.co.jp

Abstract An information tracing system "TRX" was developed to visualize the flow of information on the Cloud. It monitors the entire file I/O of the Cloud, and extracts its essential operational information from enormous raw file I/O data. Such filtering algorithm needs periodical maintenance according to every version of applications and OS. This paper propose a method for information tracing which uses similarity of the information with file I/O, which reduces the cost of maintenance, and also applicable for tracing the information on the clipboard and/or network.

1 はじめに

クラウドサービスはその経済性や機動性から利用拡大が進む一方で，重要な情報資産を他社が運営する環境に委ねる事への不安感を持たれる場合があり，普及の阻害要因になる可能性がある．これは，クラウド事業者の運用の失敗により情報が消失してしまう懸念や，外部に流出するなど意図しない形で流通してしまうリスクに対する懸念である．オンプレミスの場合，例えば自社の社屋内の物理サーバに閉域網経由でアクセスするなど物理的に防御していたケースでありインフラの運用を外部委託するクラウド固有のリスクと言える．

この不安感払拭のための1つのアプローチとして我々はこれまでに，トレーサビリティ基盤

TRX（以下”TRX”）を開発してきた．TRX ではクラウド上のデスクトップサーバにエージェントを組み込んで，アプリケーションのファイルアクセスを監視しユーザ操作に基づく情報の削除・移動・更新等を履歴として保存し，ユーザ本人や管理者からの求めに応じて「ファイルトレース」として可視化することでクラウド利用での不安感払拭に貢献してきた．[1][2][3][4]

2　ファイルトレースについて

TRX でのファイルトレースは，ユーザ操作の可視化の観点からファイルに対する「新規作成」「参照」「更新」「複製」「変名」「削除」等の基本操作を単位として履歴化を行なっている．

これら履歴生成に必要な情報収集のため，エージェントによるファイル I/O の監視機能を用いている．例えば Microsoft Windows の場合には Installable File System(IFS)としてファイル I/O 監視機能が提供されており，ファイルシステム経由の全アクセスを把握する事が可能となるため，ウィルス対策ソフトのリアルタイム監視等でも用いられている．しかし漏れなくファイルアクセス状況が把握出来る一方，ユーザ操作が細かく分解された個々の I/O として記録される場合や，ユーザ操作とは直接関連しない一時ファイル作成・削除や OS のファイル操作などユーザ操作に比べ大量の詳細情報も混在してしまう．

TRX では必要情報抽出のため，ファイルパス名および Read/Write/Delete 等のファイルに対するオペレーションとファイル内容の代表値としてのハッシュ値，そして最前面アプリケーションのウィンドウタイトル文字列を組み合わせて利用している．例えばユーザがファイル C:¥abc を D:¥abc に移動した場合，実際には C:¥abc の読込み，D:¥abc の生成・書込み，C:¥abc の削除といった一連のファイル I/O が記録されるが，ハッシュ値が同一であればドライブをまたがったファイルの移動であり，それを解釈して「C:¥abc から D:¥abc への移動」という1つの操作履歴として残す．

また，文書作成や表計算，プレゼンテーションなど，ビジネスで多く利用されるいわゆる「オフィスアプリケーション」では，単一のユーザ操作が多数のファイル操作に分解されており，ウィンドウタイトルに表示されるファイル名文字列も参考にしてユーザが意図して行った操作に近い履歴に変換して記録している．

2.1　ファイルトレースの課題

(1)逆変換の完全性向上のコスト

ファイル I/O 監視機能は全ファイル I/O を監視可能な反面，本来のユーザ操作を単位とする履歴への逆変換には困難を伴う．

図 2-1 は PC 環境での情報変換モデルである．形態の変化とメディアが組み合わされた形だが，メディアをファイルに限定しても様々な形態変化がある．一般にアプリケーションの内部動作仕様は非公開（ブラックボックス）であり，ユーザがアプリケーションを操作するとアプリケーションがファイル I/O を発生させるが操作と 1:1 対応するとは限らない．また，アプリケーションのバージョン・OS・ファイル形式等他の条件によっても対応関係は変化し得る．

TRX では外部観察によって，ユーザ操作からファイル I/O への対応関係を求め，プログラムロジックおよび変換ルールにより，ユーザ操作

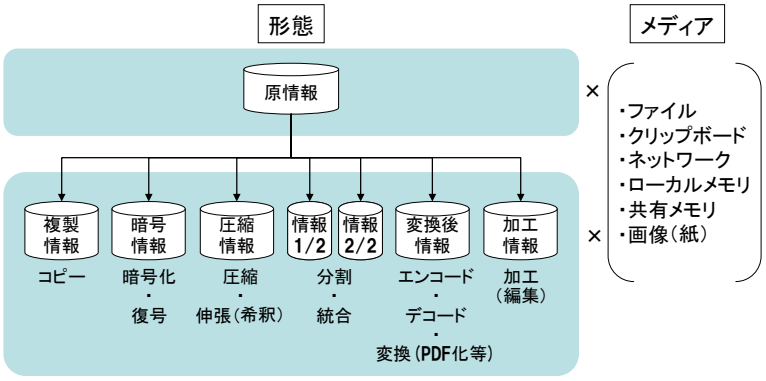

図2-1 PC環境での情報変換モデル

への逆変換を行い履歴として記録している.

しかし, ブラックボックスであるアプリケーション仕様を外部観察から完全に特定する事は困難であり, 推定エラーを完全に無くすことは原理的に不可能である.

また, アプリケーションや OS のバージョンアップ, あるいは推定エラーの申告を受けてプログラムロジックや変換規則の修正を実施し続け, 更にそれらの更新版を利用者に配布・インストールする必要があり, 維持管理のために定常的なコストを必要とする.

(2)非ファイル媒体のトレース

アプリケーションが入出力する情報はファイルシステム経由のみではない. 図2-1に示す通り情報は各種メディアに変換され得る.

クリップボードは編集手段として頻繁に利用されるがファイル I/O は発生しない.

また, 電子メールやファイル転送等ネットワーク通信としての情報送受信はファイル I/O 監視では検出できない. 更に, ファイル I/O とは異なり通信プロトコル上で分割・変換されたデータ本体が送受信されている.

2.2 課題解決に向けて

課題(1)の解決には, 逆変換ロジックのメンテナンス省力化が必要である. ファイル内容を見て入出力ファイル間の関連を推定出来ればアプリケーション動作の違いを自動吸収する事が期待される. 例えば, アプリケーションによって新規生成されたファイルでもそれ以前に同じプロセスが読み込んだファイルとの類似性が高ければ「編集」, 低ければ「新規作成」と判断出来る事が期待される. 更に編集時の書込みが新規生成からテンポラリファイルへの書込み＋複製＋削除へと内部仕様が変化 してもファイルの中身に沿った判定であれば複数動作をまとめて「編集」と判断出来る事が期待される.

課題(2)の解決には, クリップボードやネットワークパケット監視など, 新たな監視を行う必要があるがそれに伴う新たな課題も存在する. クリップボードでは, コピーされたデータの中身や, コピーしたアプリケーションの特定は可能だが, どのアプリケーションに情報が渡ったかを検知出来ない.

ネットワークパケット監視では, キャプチャしたデータを識別するため通信プロトコルの解釈とそれに基づくデータ分離が必要であるが, 全てのプロトコルへの対応にはファイル I/O からユーザ操作への逆変換以上の困難を伴う. TCP/IP 等の低レイヤプロトコルの解釈は行いつつ, ペイロードに流れるデータそのものを比較する事で厳密なプロトコル分析を代替できる可能性がある.

以上のように, デジタルデータの内容に踏み込んで, どのデータがどのメディアを通じてやり取りされたのかを「類似度」を利用して推定する事で前述の課題解決が期待される.

3 類似度について

デジタルデータの比較手法は多くの分野で研究が行われている[5]-[11]. 本稿では, 2種類のデジタルデータがどの程度似ているかを示す一次元の評価値を類似度と定義する.

表4-1に主な類似度計算方式を挙げた. デジタルフォレンジック分野では完全一致を判定するハッシュ値, デジタルデータを複数区間に分割し部分一致も判定可能としたファジーハッシュ, エントロピーや情報量あるいは重み付けエントロピー等が使われている. ファジーハッシュが部分一致まで判定しているのに対し, エントロピーに基づく方式はファイル全体の平均情報量に基づく低精度の比較であるものの, アルゴリズムが簡単で計算量が少ない利点がある.

自然言語分野では単語出現率, N 文字の並びの出現傾向を比較するN-gramが知られる.

音声認識や遺伝子探索の分野では柔軟な一致のため探索パターンを伸縮させマッチングを行う DP マッチングが用いられる. ただ DP マッチングは計算量が大きいため遺伝子探索分野では, これを高速化した FASTA や BLAST 等の類縁度近似手法が用いられている.

その他の手法としては，2種類のデジタルデータを合わせたデータの圧縮率から類似度を計算する NCD[6]や画像に特化した色相ヒストグラムを用いた方式[11]等が提案されている.

4　類似度適用ファイルトレース

4.1　新手法について

新手法ではファイル I/O に加えクリップボードやネットワークパケットの監視を基本とし，その補助として類似度を用いる.

すなわち，ファイル I/O やネットワークパケット監視においては，個々のプロセスをノードとし当該プロセスが入出力したファイルやネットワークパケットが OS の提供する機能を用いて特定される. ただしクリップボードに関してはクリップボードへの書出しのみが特定される.

次の段階として類似度を用いた枝刈りを行う. すなわち，あるプロセスの入力と出力について類似度を計算し，一定以上の類似度がある入出力はその間で情報が入力から出力に向かって流通したとみなす. これにより，複雑なファイル操作が行われ多数の入出力がある場合でも，ユーザが意図した操作としての情報の流通や加工が行われた枝を機械的に特定する事が出来る.

また，ネットワーク経由の情報流通でプロトコルの解釈によるデータ分離が困難だったとしても，デジタルデータとしてペイロード部分と他のデジタルデータの類似度から通信経由のデータ送受信の判定を行う事が出来る.

4.2　新手法に適用する類似度の要件

新手法では，類似度を2段階に分けて計算する. すなわち，(1)デジタルデータから256バイト程度の大きさの「特徴量」への変換と，(2)2つの特徴量から「類似度」への変換である. (1)はファイル入出力等が発生するたびにリアルタイムに計算・記録する必要があるが，(2)は可視化のタイミングまでに計算すれば良い.

そのため，新手法に適用する類似度は，(A)適用可能なデジタルデータの種類を問わないことが必要であるのみならず，(B)一旦「特徴量」に変換する事が可能でなければならない上に，特徴量自体は履歴として記録するため(C)特徴量は出来るだけコンパクトに表現出来ることが望ましい. また，特徴量への変換はリアルタイム性が必要となるため，(D)なるべく計算量や必要記憶容量が少ないことが望ましい. 更に，(E)特徴量からの類似度計算も計算量や必要記憶容量が少ないことが望ましい.

4.3　類似度の適用性に関する考察

新手法に適用する類似度としてこれまでに提案されている既存方式の適用性を評価する. 表4-1 は新手法への適用性の観点(A)～(E)で既存方式を定性的に比較した結果である.

先ず，NCD や DP マッチング，類縁度近似は特徴量化が困難な上，計算量も比較的多いため方式として除外する. また，色相ヒストグラムは画像にしか適用出来ないためこれも除外する. 通常のハッシュは前述の通り類似度として使えないため除外する.

ファジーハッシュは ssdeep[7]や基になった

表4-1　主な類似度計算方式とその特徴

方式	適用領域	特徴量				類似度		
		特徴量化	計算量	記憶容量	サイズ	計算量	記憶容量	精度
ハッシュ	一般ファイル	◎	△	○	○	◎	◎	×※
ファジーハッシュ	一般ファイル	○	△	◎	△	△	○	○
N-gram	テキスト	△	◎	△	△	△	△	○
	一般ファイル	△	◎	×	△	△	×	○
エントロピー	一般ファイル	◎	○	○	◎	◎	◎	△
情報量/重み付けエントロピー	一般ファイル	◎	○	○	◎	◎	◎	△+
NCD (CompLearn)	一般ファイル	×	−	−	−	×	△	△
DPマッチング	音声、遺伝子	×	−	−	−	△	△	◎
類縁度近似(FASTA, BLAST)	遺伝子	×	−	−	−	○	△	◎
画像検索(色相ヒストグラム)	画像	○	△	○	△	○	○	△

※ハッシュは完全一致か否かの判定のみ可能で類似判定には不適の意味

spamsum[8]に代表される通り，デジタルデータの複数分割を行い各区分のハッシュ値の集合を 64 文字以下の特徴量として保持する．ハッシュとは言えセキュリティ分野で用いられる SHA2 に代表される数百ビット程度のハッシュ値を出力するハッシュ関数ではなく，計算量の少ない FNV ハッシュを 6 ビットの出力で用いている．そのため各区分について 1/64 程度の確率でハッシュ衝突が発生するため，それに基づく類似度も一定の誤差を含む事を考慮する必要がある．

ファジーハッシュでは，各区分の分割点が一致しなければ比較が出来ないため，デジタルデータの文脈に沿った分割を行なっている．これには入力データの N バイト分によってのみ値が定まる特殊なハッシュ関数（Rolling hash と呼ばれる）を用いて，その出力値の整数 M による剰余が M-1 に一致した点を分割点にする手法を用いる．それでも M の値が一致しなければ比較出来ないため，1 つの特徴量の中に M の値，M による分割ハッシュ 2M による分割ハッシュの両方を保持する工夫を行なっている．

芹田らは M として 2 のべき乗 2^t を選択し，入力データが最初に分割される t の値 t_max から 1 ずつ係数を減らして入れ子で分割数を増やした複数階層のハッシュ値を保持する事で編集によって長さが大きく変化した部分データの比較を可能とした方式を提案している[5]．これは例えばクリップボードに切り出した部分データや本体ファイルの一部を抽出したデータを検出するための類似度として望ましい性質である．ただし，ファジーハッシュは分割点の最適化のために計算量がやや多い傾向がある．

N-gram はテキストの比較では Tri-gram 等が用いられているが，比較対象となるデジタルデータから Tri-gram を生成し直接比較を行う処理が基本となっており，特徴量化は考慮されていない．一般ファイルに単純に Tri-gram を適用した場合には $256^3 \fallingdotseq 1678$ 万通りのパターン×出現頻度を数百バイトの特徴量に変換する必要があり，圧縮の工夫が必要である．

エントロピーは N-gram の特殊形と言える．すなわち Uni-gram の出現頻度の偏りを1つの数値として特徴量化したものとみなせる．出現頻度の計数はカウンタのみで済むため計算量は小さいが，特徴量化の際に対数計算が必要となるため対象となるデジタルデータが小さい場合には相対的にオーバーヘッドが大きくなる．また，デジタルデータ全体として1つの数値を特徴量とする事から，識別精度には限界があるほか，部分データの検出も難しい．

情報量や重み付けエントロピーはエントロピーにファイル長やファイル長の対数値を掛け合わせた数値で，特に後者は数値の桁数増大が抑えられる事から Weighted Entropy としてデジタルフォレンジックの分野で国際特許として出願され[9]用いられている．また，通常のエントロピーはデジタルデータを構成する各 1 バイトの出現頻度のみを用いているためその順序性は考慮されないが，前後のデータの差分のエントロピーや一次マルコフ過程のエントロピーを用いる Weighted Entropy も提案されており，計算量や必要とする記憶容量は増加するものの一般的なエントロピーに比べて識別精度が若干高まる．

また，情報量に関しては可逆圧縮において情報量が不変であるという原理から，図 2-1 のモデルにおける圧縮の前後における類似度として高い値が出力される事が期待される．

4.4 既存類似度の評価

上記考察から，エントロピー系およびファジーハッシュ系の類似度の適用性が高いと見込まれるため具体的なデータを用いて評価する．

エントロピー系として，単純なエントロピー×ファイル長による情報量（図中 ”Entropy (InfoSize)”と表記），および一次マルコフ過程のエントロピー値×ファイル長による情報量（図中”Entropy (M1 InfoSize)”と表記)を特徴量として用いた．なお，類似度 $L = 100 * (1 - |E_0-E_1|/(E_0+E_1))$として計算した．ただし E_0,E_1 は比較対象の 2 つの特徴量(=情報量)である．

ファジーハッシュ系は，Spamsum[8]および

芹田らの方式[5][10](図中「Hitachi」と表記)を選択した. いずれも類似度は同一レベルの区分ハッシュの集合を文字列とみなした場合の編集距離から計算した.

評価として, 類似度の識別能力を評価するため類似ファイルおよび異なるファイル同士で類似度を計算し, それらが数値的に識別可能であるか否かを確認する. この観点で以下のファイル及びファイルセットを用いた:

(a)16 スライド構成の Power Point ファイル

(b)上記(a)を 1 スライド毎に分割したファイル

(c)上記(a)と, その1スライドを少しずつ修正した 4 種類のファイルで合計 5 種類のファイル

(d)異なる作成者による 16 個の Power Point ファイルから1スライドずつ抽出したファイル.

先ず, ファイルセット(b)のスライド間の類似度をヒストグラム化したものを図 4-1 に示す. 各スライドの記述内容は異なるため, 類似度が低い事が期待されたがエントロピー系・ファジーハッシュ系とも多くが 69～99 の高い類似度を示した.

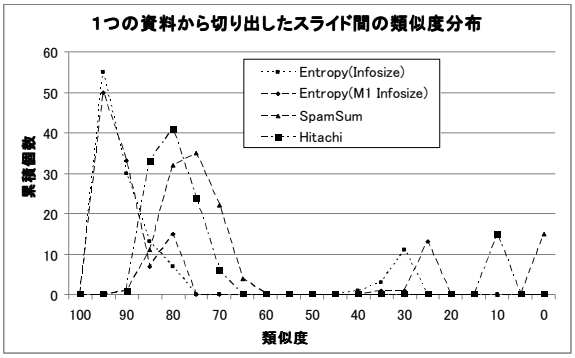

図 4-1 切り出したスライド間の類似度分布

計算に用いたファイルの一部を図 4-2 に示す. 大半のスライドは上中段の(b)-3・(b)-4 と同様に高類似度を示し, 下段の(b)-2 スライドのみ他の全てのスライドとの類似度が低い. 原因は, (b)-2 スライドのみファイルサイズが 1,914[KB] と大きく, 背景として貼り付けられた世界地図のビットマップが大半を占めており他スライドとの一致部分が少なかったためと推測される. 逆に (b)-3・(b)-4 の場合はテキストや簡単な基本図形のみが存在するシンプルなスライドだが, タイトルの背景にビットマップが使われており, 文字列データよりも大きなサイズを持っている事が高類似度の原因と推定される.

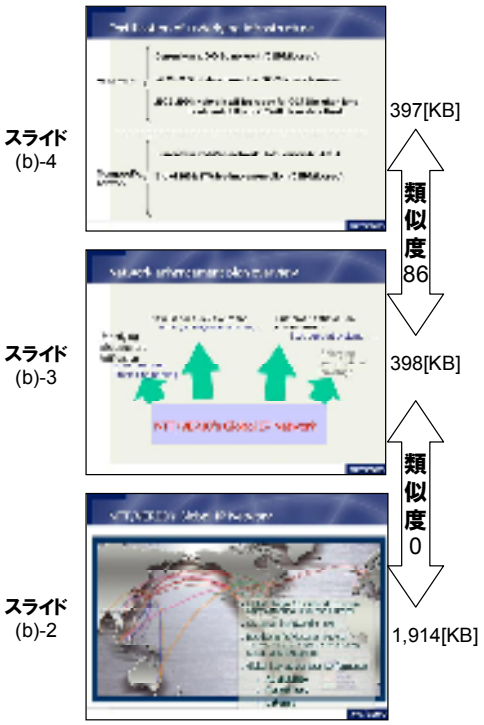

図 4-2 評価用スライドファイル

Power Point ファイルの比較では内包するサイズの大きなビットマップの影響を非常に受けやすいと言える.

次に母体ファイルからの部分切り出しの検出を確認するため, ファイル(a)とファイルセット(b)との類似度の分布を図 4-3 に示す.

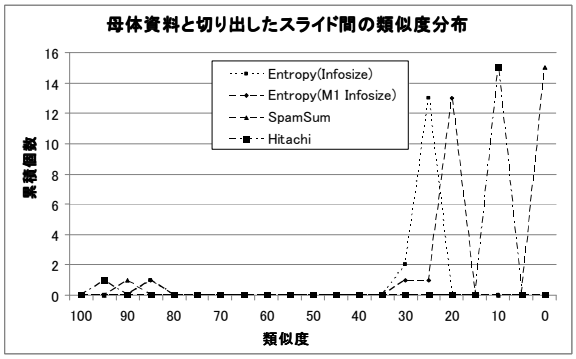

図 4-3 一部切り出しに関する類似度分布

この場合, 4 方式とも図 4-2 の(b)-2 スライドと母体ファイルの類似度が 89～97 と高く他は低い結果になった. これも, 内包する巨大なビットマップに影響された結果であると言えるが, そ

の他のスライドについても一定程度の類似度が計算されている点に注目する必要がある.

特にSpamsumで類似度0となっている部分について芹田らの方式(Hitachi)では11～12と一定の類似性があると判定されており, 部分抽出の検出能力が高い特徴[5]が確認された.

今度は, スライドの編集に対する類似度の変化を確認するためファイルセット(c)について, 母体ファイルとその編集ファイル4種類について相互の類似度を計算した結果を図 4-4 に示す.

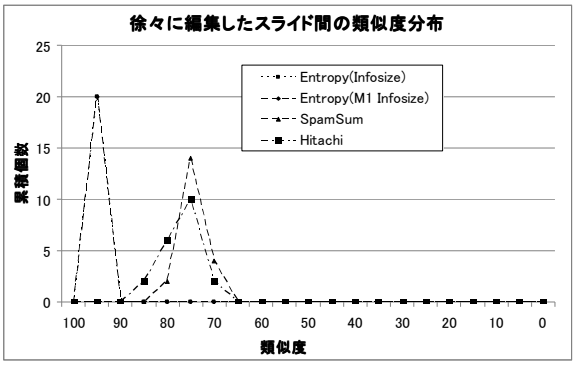

図 4-4 スライド編集に伴う類似度分布

16スライドのうちサイズ的に大きくない1スライドの文字の修正や図形の削除等の操作であり, 人間の感覚的にはほぼ同一とみなせるレベルである.

エントロピー系はほぼ 99 と完全一致に近い値であるが, ファジーハッシュ系は74～84の値となり感覚的な差分より類似度が低く評価されている. これはアプリケーションのファイルフォーマットの特性として部分的な編集でもファイル内の複数箇所に影響を与えているものと推測される. そのため, 編集量が多い場合に類似度が大きく低下する事も考えられる.

これまでは 1 つの母体ファイルから切り出した個々のファイルの類似度で評価したが, ファイルセット(d)の全く無関係な複数ファイルの類似度を比較した結果が図 4-5 である.

ファジーハッシュ系の類似度がいずれも低く, エントロピー系類似度が比較的高めの値となり両者で大きな違いが出た.

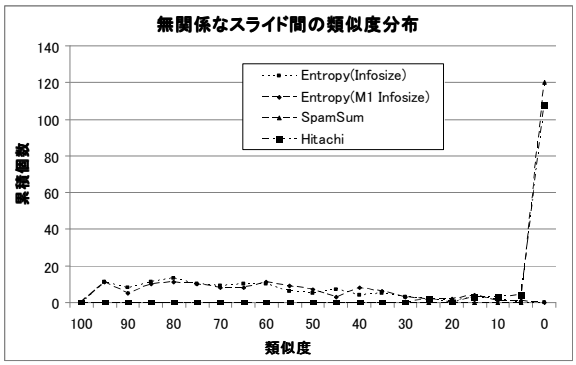

図 4-5 無関係なファイル間の類似度分布

詳細に確認すると, Spamsum では全て 0 で, 芹田らの方式(Hitachi)では殆ど 0 で一部 2～26(平均 2)の類似度となった. 比較的ファイルサイズが小さいファイル同士が高い値である事からファイルフォーマット上存在する共通要素に反応したのではないかと推測される.

エントロピー系類似度は, 9～99 の類似度となり単純エントロピーによる情報量で平均 67, 一次マルコフ過程エントロピーによる情報量では平均 65 となった. これはファイルフォーマットとして含まれる情報の分布が比較的似通っているため, 内容に依らず類似と判定してしまう事に依るものと推測される.

最後に, 圧縮における類似度を評価するためファイルセット(d)とそれを WinZip の「Maximum (portable)」で圧縮したファイルの類似度の分布を図 4-6 に示す.

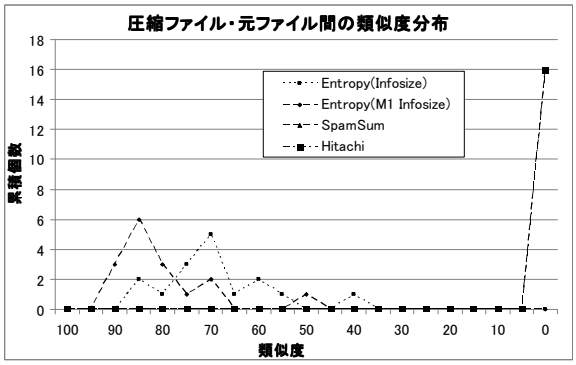

図 4-6 WinZip 圧縮における類似度分布

ファジーハッシュではほぼ全て 0 となったが, エントロピー系では 43～93 となった. 単純エントロピーによる情報量で平均 71, 一次マルコフ

過程エントロピーによる情報量で平均 82 となった. エントロピー系では高い類似度を期待していたがそれよりもかなり低くなってしまったのは WinZip の圧縮アルゴリズムがエントロピー計算よりも正確に情報量を見極めて圧縮しているためであり, より優秀な圧縮アルゴリズムの場合に更に類似度が低くなると予想される.

5 まとめ

ファイル I/O の監視に基づくデジタルデータの追跡である「ファイルトレース」について, その課題解決のために類似度を併用する新手法を提案した. 更に新手法に必要な類似度について, エントロピー系およびファジーハッシュ系アルゴリズムを精度の観点で評価した. 通常編集ではファジーハッシュ系, 圧縮においてはエントロピー系の適正が高い傾向が見られたが, ファジーハッシュは計算量が大きいため実用化には精度と計算量のバランスの観点で改善を図った方式が必要である. 今後は提案手法に適用するために特徴量計算の際の計算量と必要記憶容量が少なく, かつ精度の高い類似度計算手法の研究を進める.

適切な類似度計算方式が伴えば, ユーザ操作への逆変換アルゴリズムのうち操作対象ファイルの推定やファイル修正の有無の判定, 複数ファイルを編集した場合のコンテンツの複製等の対応関係の把握を自動化する事が可能となるため維持管理コストを低減させる事が出来るのみならず, クリップボード経由の情報流通やネットワーク経由の情報流通の可視化も可能になる事が期待される.

最後に, 本研究を進めるにあたって日頃より数々のコメントを頂いている永吉剛主幹研究員をはじめとする所員の皆様に感謝いたします.

参考文献

[1] S.Nakahara, H.Ishimoto, “A Study on the requirements of accountable cloud services and log management,” APSITT, 2010

[2] 中原ほか,“クラウドトレーサビリティ(CBoC TRX)”,NTT 技術ジャーナル 2011.10, p.31-35

[3] 張 一凡ほか,”MapReduce を用いたログ間の依存関係ツリーの抽出アルゴリズムの提案,” 情処学会全国大会論文集 74th-1, pp.277-278

[4] 中原ほか, “クラウドサービスの説明能力とトレーサビリティ技術,” 信学技報 112(22):2012.5.10・11, pp.81-85

[5] 芹田ほか, “ファイル伸縮に耐性のある類似ハッシュ算出方式の考察,” IEICE Technical Report ISEC2010-54 LOIS2010-33 pp.31-36

[6] Paul Vitanyi,渡辺(訳), “圧縮度にもとづいた汎用な類似度測定法”, 数理科学 No.521,Nov 2006, pp.1-8

[7] Jesse Kornblum,”Identifying almost identical files using context triggered piecewise hashing,” Digital investigation 3S(2006) pp.91-97.

[8] Tridgell Andrew. Spamsum README: http://samba.org/ftp/unpacked/junkcode/spamsum/README; 2002.

[9] GUIDANCE SOFTWARE , INC., “System and method for entropy-based near-match analysis.” 国際特許 WO2010/107659 A1

[10] 藤井ほか, “デジタルシーケンス特徴量算出方法及びデジタルシーケンス特徴量算出装置”, 日本国公開特許広報 特開 2012-18549

[11] Swain, M. J. and Ballard, D. H.: “Color Indexing,” International Journal of Computer Vision, 7(1), pp.11-32, 1991