M-014

異なる種類のログの統合的保管に関する提案
A Proposal for Unified Storing of Different Logs

友野 敬大† 上原 稔‡ 島田 裕次‡
AKIHIRO TOMONO MINORU UEHARA YUJI SHIMADA

1. はじめに

近年，米国企業に留まらず，日本企業でも不祥事が多発し，内部統制の必要性が急速に高まっている．ログは，不正アクセスの発見や事後対応を行う際に重要な意味合いをもつとともに，さらに企業改善につながる情報を多分に含んでいると考えられる．

ログを用いて内部統制を実現するためには，より長期的な保存と多くの種類のログが必要となる．我々は，VLSD（Virtual Large Scale Disk）を用いて，低コストで半永久的なログの保存を可能にするシステムを開発した．このシステムでは，生ログに着目し，その保存および管理を行う．しかし，アプリケーションによって異なる形式のログが書きだされる可能性が高いため，横断的な情報検索をするのは容易でない．また，データベースに格納する場合，固有のカラムを持つログが多くなればなるほど，そのカラムを持たないログは NULL 値を持つので，効率的であるとは言えない．また，組織において，長期的な保存をする際に，システムの変更やアプリケーションのアップデートなど，ログのスキーマが変わる可能性を考慮しなければならない．

そこで，本研究では，複数種類あるログを YAML（YAML Ain't Markup Language）によって統一することで，スキーマの変更にオンデマンドに対応できるログを提案する．また，YAML 形式に変換したログをさらに統合化し，1 つのファイルにまとめることで，複数種類のログを横断的に検索可能なログ形式を提案する．

2. 関連研究

2.1 VLSD

VLSDとは，JavaによるソフトウェアRAIDとNBDの実装を含む大規模ストレージ構築のためのツールキットである[1] [2]．VLSDは100% pure Javaであり，Javaが動作するプラットフォームの上ならVLSDも動作する．NBDを用いることで，ファイルシステムに依存しないディスクレベル分散ストレージの実現が可能となる．これにより，高価なストレージの必要性がなくなる．

暗号化や書き込み禁止ディスクを実現するクラスを持ち，これらを組み合わせることで，不正アクセスなどのセキュリティの問題を解決する[3]．

2.2 ログ管理システム

内部統制を実現するのにログ管理は重要であることは，前述のとおりであるが，容量の問題やコストの問題は無視できない．それらの問題を解決するために，我々は VLSD を用いて，半永久的にログを保存可能なシステムの開発した[4]．このシステムは，OS にあらかじめ備わっている機能を用いて実装される．排出されるログを上書きするのではなく，ストレージに保存し続ける．本大学では，常時稼働しているサーバが 25 台あるので，それらから遊休資源を収集し，ストレージを構築する．

2.3 ログ収集エージェント

我々が開発したログ収集エージェント（以降，エージェントとする）は自動的にログを収集するので，ユーザに余計な負担をかけずにログを収集する[5]．エージェントは，業務に関係のないアプリケーションの使用や不審な操作もログとして書き出す．アプリケーションがアクティブになったときに，そのプロセス ID やプロセス名などの情報を取得する．そして，他のウィンドウハンドルがアクティブになったときにログを書き出す．

3. 取得するログとその構造化

関連研究で示したログ収集エージェントを，複数のログが取得できるように改良した．エージェントが取得できるログは，アプリケーションログ，プロセスログ，ファイル操作ログの 3 種類である．

この 3 種類のログのフォーマットを YAML によって構造化し，1 つのログとしてまとめる方法を提案する．

複数の種類のログをデータベースに格納する場合，その種類の数だけテーブルを用意するのが一般的である．1 つのスキーマに複数種類のログを格納すると，あるカラムに対して Null 値が格納される可能性が高く，効率的ではないというのがその理由である．しかし，テーブルを複数用意すると，日時に対して複数のログを横断的に調べる場合に手間がかかる．そこで，本研究では，テキストベースのログを構造化して管理することで，横断的な検索が可能なログを作る．

構造化は YAML によって実現する．YAML は，同様の構造化言語として用いられる XML よりも次の点で優位性があることが知られている．

- ユーザ（人間）にとって，わかりやすく読みやすい
- インデントによるネスト表現で，記述が最小限
- 配列とハッシュとスカラーだけで表現する簡易性

YAML は文章のような非定型的なデータを扱うのは得意でない．アドレス帳や設定ファイルのように定型化されたデータを表現するのに適している．

また，YAML 形式に変換されたものを，さらに CSV 形式に変換する．この CSV 形式のログは一般的なものとは異なり，複数種類のログが 1 つのファイルにまとめられたものである．それぞれのログのスキーマの定義は，*def* で始まるレコードにより定義されている．*def* レコードには，タイムスタンプとログの種類，スキーマの定義が記述されている．これにより，定義が更新されても，新しいタイムスタンプの *def* レコードを参照することで，ログのフォーマットを最新に保つことができる．

† 東洋大学大学院工学研究科
Graduate School of Engineering, Toyo Univ.
‡ 東洋大学総合情報学部
Department of Information Sciences and Arts, Toyo Univ.

4. 考察

ログを用いて内部統制を実現するためには，より長期的な保存と多くの種類のログが必要となる．長期間の保存を実現できれば，より長いスパンでの情報をログから抽出することができるので，内部統制を実現するうえでは有効であると考えられる．しかし，ログの永久保存は費用対効果を考えると，非効率的なものになる．不正アクセス禁止法および電子計算機使用詐欺罪や電磁的記録不正作出および供用罪などのコンピュータ犯罪に関する刑法の時効は，およそ 3~5 年とされている．そのため，裁判証拠としてログを用いる場合，その期間分は保証されるのが望ましい．

データベースにおいて，長期的なログの保存を考えたとき，そのスキーマが変更される可能性を考慮しなければならない．カラムが追加された場合，データベースを更新するため，それまで保存してきたログに対しても処理が必要になる．加えて，ログを書き出すアプリケーションのプログラムの変更も強要される．これに対して，テキストベースである YAML 形式では，オンデマンドにそのスキーマを変更することができる．

あるタイミングでログのフォーマットが変更（カラムの増加，あるいは減少）されたとしても，共通のカラムを持つ場合には，これをキーとしてログのレコードを探索することができる．フォーマットが変更された場合を仮定した図を図 1 に示す．

```
"app","2009/07/29 21:06:19","<2744>","","Yasu","VIOLIN01","Java - vis
"app","2009/07/29 21:06:24","<2744>","","Yasu","VIOLIN01","Java - vis
"app","2010/02/28","9:52:55","2010/02/28","9:54:37","VIOLIN01","Yasu",
"app","2010/02/28","9:54:37","2010/02/28","9:54:43","VIOLIN01","Yasu",
```

図 1　フォーマットの違うログ

実線以降のログには，アプリケーションの内部名およびオリジナルファイル名などのカラムが追加されている．このとき，ユーザ名やホスト名は共通のカラムである．これをキーとして，検索することが可能となる．例えば，図 2 中のユーザ"Yasu"において，実線以前のフォーマットのログも検索できるので，さらに長いスパンでの検索が可能となる．また，ユーザ名やホスト名などは異なる種類のログとも共通のカラムとなりやすい．そのため，これらをキーとすれば，長期間かつ異なる種類のログに対しても，ログの検索が可能になると考えられる．

より長い期間，ログを保存するために，その容量について考える必要がある．アクティブなウィンドウが切り替わるたびに，エージェントは CSV 形式でログを書き出す．このログをそれぞれ，YAML 形式に変換したもの，データベースに格納したものの容量を表 1 に示す．

表 1　それぞれのログの容量（単位：Byte）

	Appli log	Proc log	File log	合計
CSV	69,522	8,541	174,965	253,028
YAML	72,331	21,558	320,926	414,815
MySQL	64,464	7,804	232,164	312,396
SQLite	26,624	10,240	186,368	237,568

データベースは，MySQL5.0 と SQLite3.3.6 を用いる．ここで，MySQL は各テーブルの DATA_LENGTH と INDEX_LENGTH の和，SQLite は各テーブルのファイルのサイズとする．また，単位はすべてバイト（B）である．

もとの CSV に対して，YAML の容量が大きくなるのは，単純に値に対してキーを設けているからである．対して，テキストをバイナリ形式に変換するため，データベースではそれよりも小さくなる．ただし，ファイルログに関しては，ファイルパスを値として保持するため，バイナリ化しても容量は小さくならないと考えられる．容量面を考慮すれば，データベースに格納することが一般的であり，有効である．データベースに格納する場合の利点と欠点について考える．まず利点として，データベースは一般的にインデックスを作成可能なため，検索する際に有用である．一方で，CSV ならびに YAML はテキストベースであるため，シーケンシャルな検索になる．しかし，YAML においてはインデックスを作成することで，検索に対しては特に問題ないとは考えられる．

また，MySQL の現在のバージョンでは，TIME 型としてマイクロ秒を扱うことができない．エージェントが書き出すログは，開始時間と終了時間の差から求めるため，マイクロ秒を扱える必要がある．カラムを分けるなどの方法はあるが，手間がかかるのは事実である．YAML はスカラー値からそのデータ型を判別するため，この点に関して有用である．容量から見ると，YAML はデータベースと比較したとき，不利である．しかし，テーブルの効率性やカラムにおけるデータ型の扱い方などから YAML によって構造化されたログが適していると考えられる．

5. 今後の課題

実際にログを用意して検索を行い，データベースと比較したときのパフォーマンスの定量的な評価をする必要がある．今後，Syslog などのフォーマットを YAML 形式に統一することで，さらに多くの種類のログを横断的に検索することが可能になると考えられる．

6. まとめ

本論文では，複数種類あるログを YAML によって統一することで，スキーマの変更にオンデマンドに対応できるログについて提案した．また，独自のカラムを持つログについても考慮し，データベースとの比較および検討を行った．今後，対応できるフォーマットを増やし，形式を変換する時間やデータベースとの検索時間の比較などの定量的な評価をする必要がある．

参考文献

[1] 上原 稔　“教育環境における仮想大規模ストレージのためのツールキット”，マルチメディア通信と分散処理ワークショップ，pp.205-210, 2006 年 11 月

[2] チャイ エリアント，上原 稔，森 秀樹　”PC 教室のための仮想的大規模ストレージの構築”，マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム論文集，pp.617-622，2007 年 7 月

[3] 上原 稔　“仮想大規模ストレージにおけるセキュリティ”，情報処理学会研究報告書，pp.61-66，2007 年 11 月

[4] Akihiro Tomono, Minoru Uehara, Makoto Murakami, Motoi Yamagiwa: "A Log Management System for Internal Control", In Proc. of 2009 International Conference on Network-Based Information Systems(NBiS2009), pp.432-439, (2009.8.19-21)

[5] Akihiro Tomono, Minoru Uehara, Yuji Shimada: "A Proposal for Method to Manage License based on Log", TeNAS2010, (2010.4.20-23)