B-012

KVM を利用した機密情報の拡散追跡機能におけるファイルアクセス性能の評価
Evaluation of File Access Performance in Function for Tracing Diffusion of Classified Information on KVM

森山 英明† 山内 利宏‡ 佐藤 将也‡ 谷口 秀夫‡
Hideaki Moriyama Toshihiro Yamauchi Masaya Sato Hideo Taniguchi

1. はじめに

計算機内で管理されている機密情報は，外部に漏えいすることで，企業や個人にとって大きな損失となる．この問題に対処するため，これまでに，仮想計算機モニタ(VMM)を利用した機密情報の拡散追跡機能を提案し，実現している．この機能は，機密情報を操作するシステムコールをフックし，VMM 側で解析することで，機密情報の拡散を検知する．一方，システムコールの発行ごとに発生する検知処理のオーバヘッドが課題となる．このため，これまでにオーバヘッド削減による高速化を実現している．

本稿では，機密情報を有するファイルへのアクセスに対して，機密情報の拡散追跡処理による影響とオーバヘッド削減による高速化の効果を評価した結果について述べる．

2. KVM における機密情報の拡散追跡機能

計算機内の機密情報の利用状況を把握するために，仮想計算機モニタ(VMM: Virtual Machine Monitor)を用いた機密情報の拡散追跡機能を提案している．また，KVM (Kernel-based Virtual Machine) 上に機能を実現し，評価結果を報告した[1]．機密情報の拡散追跡機能では，機密情報を有する可能性のあるファイルとプロセス(以降，管理対象ファイルと管理対象プロセス)を拡散情報として記録し，管理する．この機能を VMM 上に実装することにより，オペレーティングシステム(OS)よりも攻撃が困難である VMM で機密情報を管理できる．また，ゲスト OS のソースコードを改変することなく VMM の改変のみで機能を提供できる利点がある．

KVM における機密情報の拡散追跡機能では，仮想計算機上で発行されるシステムコールをフックするために，ハードウェアブレークポイントを用いてシステムコール発行前（SYSCALL 命令）と終了直前（SYSRET 命令）でデバッグ例外を発生させる．これにより，処理をゲスト OS から VMM へ移行させる．VMM 側では，デバッグアドレスレジスタによるデバッグ例外であることを確認し，SYSCALL 命令と SYSRET 命令のどちらの実行前に発生したものか確認する．次に，システムコール番号から，機密情報の拡散に関係するシステムコールであるか否かを判定する．機密情報の拡散に関係するシステムコールの場合は，SYSCALL 命令実行前の拡散追跡処理として，プロセスが発行したシステムコール番号，ページテーブル情報，扱うファイルのファイルディスクリプタの値などを取得する．SYSRET 命令実行前の拡散追跡処理では，システムコール処理の成否，システムコールを発行したプロセスが扱うファイルの情報などを取得する．もし，機密情報の拡散に関係しないシステムコールの場合は，拡散追跡にともなう処理を行わず，システムコール処理を続行する．

機密情報の拡散追跡処理では，仮想計算機上で発行されるすべてのシステムコールに対してフックを行う．このため，機密情報の拡散追跡機能による処理時間がオーバヘッドとなり，オーバヘッドの削減が課題となる．これまでに，オーバヘッドを削減するため，処理の分析，機能の改良，および評価を行っている[2]．以降では，ファイルアクセス処理に着目し，機密情報の拡散追跡機能がファイルアクセス処理に与える影響を評価した結果について述べる．また，機密情報の拡散追跡機能の改善による高速化の効果を明らかにする．

表 1 測定環境

測定用計算機	
CPU	Intel Xeon CPU E5-2609
コア数	8 個
メモリ	64GB
OS	Fedora 18 (Linux Kernel 3.6.10) (64bit)
VMM	KVM-kmod-3.6
仮想計算機	
コア数(vCPU)	1 個
メモリ	1GB
OS	Fedora 18 (Linux Kernel 3.6.10) (64bit)

3. ファイルアクセス性能の評価

3.1 Flexible I/O Tester

ファイルアクセス性能を測定するためのプログラムとして，fio (Flexible I/O Tester) を用いる．fio を利用する際，利用者は，I/O アクセスに関するパラメータとして，ファイルアクセスのパターン，ファイルサイズ，ファイル数，同時に並列実行するジョブ数などを設定する．このパラメータにしたがってファイルを作成して I/O アクセスを行い，ファイルアクセス性能を測定することができる．

3.2 評価方法

本評価の測定環境を表 1 に示す．測定用計算機上に仮想計算機を 1 台用意し，KVM から仮想計算機上の処理を監視し，機密情報の拡散を追跡する．本評価では，以下の 2 つの環境で測定を行い，結果を比較することで，改良による高速化の効果を明らかにする．

(1) 改良前の拡散追跡機能を導入した環境
(2) 改良後の拡散追跡機能を導入した環境

仮想計算機上では，fio を実行し，I/O アクセスの性能を測定した．このとき，以下の 2 つの場合に分けて測定した．

† 有明工業高等専門学校
‡ 岡山大学大学院自然科学研究科

(A) fio を機密情報の拡散に関連するプロセスとして登録する．すなわち，I/O アクセス処理を追跡する．
(B) fio を機密情報の拡散に関連しないプロセスとする．

これにより，追跡処理にかかる処理時間を明らかにする．fio に関するパラメータの設定では，ファイルアクセスのパターンとして，Random Read，Random Write，Sequential Read，Sequential Write の 4 種類を用いる．アクセス対象となるファイルは，4KB のファイルを 1,000 個用意し，ブロックサイズ 4KB でアクセスする．

また，機密情報の拡散追跡の処理時間と比較して，ディスク I/O にかかる処理時間は大きい．したがって，本評価では，tmpfs を用いる．すなわち，fio がアクセス対象とするファイル群を/tmp 以下に置くことで，メモリ上におけるデータのやり取りとし，ディスク I/O による処理時間の影響を小さくする．

3.3 評価結果

fio のアクセスパターンを Random Read としたときの測定結果を図 1 に示す．図 1 において，改良前の拡散追跡機能を導入した環境を高速化前，改良後の環境を高速化後と表している．また，各環境において，機密情報の拡散追跡を行わない場合と行う場合を比較している．図 1 より，拡散追跡を行う場合と行わない場合で，処理時間の差異はほとんど見られない．環境で比較すると，高速化前の環境では約 100ms であるのに対し，高速化後は約 92ms となっており，約 8ms と若干の改善が見られる．図 2 に示す Sequential Read の測定結果でも同様の傾向が見られ，拡散追跡処理を行う場合と行わない場合で比較すると，処理時間の差異はほとんど見られず，高速化前と高速化後の環境を比較すると，高速化前は約 95ms であるのに対し，高速化後は約 87ms となり，約 8ms の改善が見られる．以上の結果より，Read のアクセスパターンの場合は，Random，Sequential にかかわらず，若干の改善が見られる．

次に，Random Write としたときの測定結果を図 3 に示す．図 3 の高速化前の環境において，追跡処理を行わない場合の処理時間は 101ms であるのに対し，追跡処理を行う場合は 926.5ms まで処理時間が増加している．高速化後の環境では，追跡処理を行わない場合の処理時間は 93.5ms，追跡処理を行う場合の処理時間は 104ms となっている．この結果より，追跡処理を行う場合において，高速化前と高速化後では 822.5ms の処理時間の削減ができている．図 4 に示す Sequential Write の測定結果でも同様の傾向が見られ，高速化前の環境において，追跡処理を行わない場合の処理時間は 97ms であるのに対し，追跡処理を行う場合は 938ms まで処理時間が増加している．高速化後の環境では，追跡処理を行わない場合の処理時間は 88ms，追跡処理を行う場合の処理時間は 99ms で，839ms の処理時間の削減ができている．以上の結果より，Write のアクセスパターンの場合は，Random，Sequential にかかわらず，800ms 以上の処理時間の削減が見られ，高速化の効果が高いことがわかった．この理由として，機密情報の拡散追跡機能において，今回の測定では，追跡処理を行う場合，1000 個のファイルを管理対象ファイルとして登録する．高速化では，write システムコールが発行された際の管理ファイル登録処理において，オーバヘッドを削減しているため，高速化の効果が高かったものと思われる．また，Read のアクセスパターンで高い効果が見られなかった理由としては，高速化において，read システムコールが発行された際の 管理プロセス登録処理において，オーバヘッドを削減しているが，今回の測定では，追跡対象となるプロセスは fio のプロセス 1 個であったため，Write 時と比べて効果が低かったものと推察できる．

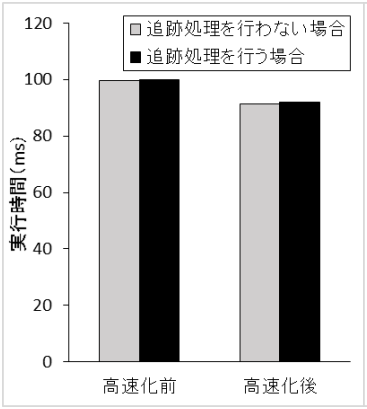

図 1 Random Read の結果

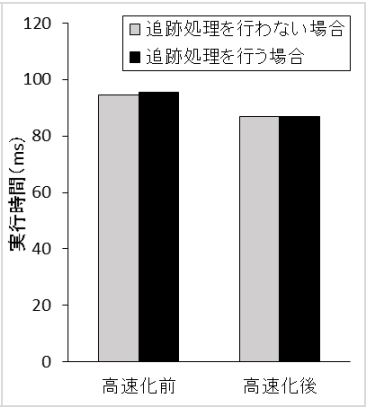

図 2 Sequential Read の結果

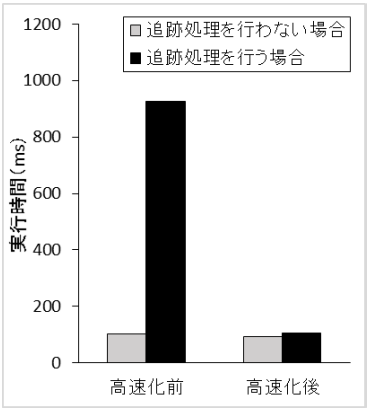

図 3 Random Write の結果

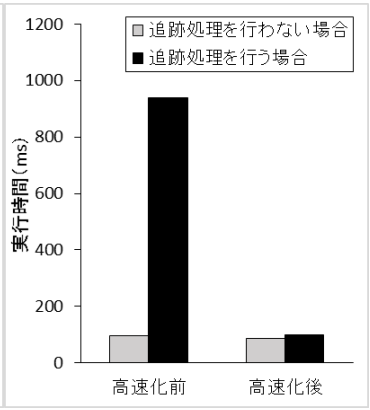

図 4 Sequential Write の結果

4. おわりに

fio によるファイルアクセスを用いた評価より，機密情報の拡散追跡機能がファイルアクセス処理に与える影響を示した．機能の改良による高速化は，管理対象となるファイルの登録数が多い場合，特に効果が大きいことを明らかにした．今後は，ファイルアクセスに関して，より詳細な評価を行う．

謝辞 実験に協力していただいた荒木涼氏（有明工業高等専門学校）に感謝します．本研究の一部は JSPS 科研費 16H02829（基盤研究(B)）の助成を受けたものです．

参考文献

[1] Fujii, S., Sato, M., Yamauchi, T., and Taniguchi, H.: Evaluation and Design of Function for Tracing Diffusion of Classified Information for File Operations with KVM, The Journal of Supercomputing, Vol.72, Issue 5, pp.1841-1861 (2016).
[2] Moriyama, H., Yamauchi, T., Sato, M., and Taniguchi, H.: Performance Improvement and Evaluation of Function for Tracing Diffusion of Classified Information on KVM, 4th International Workshop on Information and Communication Security (WICS 2017), pp.463-468, (11, 2017).