共有メモリを用いたptraceの高速化

山 本 航 洋[†1]　大 山 恵 弘[†1]

1. 背　　景

Linux kernel にはプロセスのトレースを行うためのシステムコールとして ptrace が提供されている。これを用いてシステムコールトレーサーやデバッガ、サンドボックス、ファイルシステムなどが実装されている。ptrace の機能としてトレースされているプロセス (アプリケーション) のメモリの読み書きが行える。

ptrace はシステムコールなので実行するときにユーザーモードとカーネルモードで遷移が行われる。そのためトレースを行うプロセス (トレーサー) が ptrace を呼び出すたびにコンテキストスイッチが発生する。また ptrace によるアプリケーションのメモリへの読み書きは 1 ワードづつしか行えない。つまりトレーサーがワードサイズより大きいデータをアプリケーションのメモリに対して読み書きする際、多量のコンテキストスイッチが発生し、その部分が大きなオーバーヘッドとなりうる可能性がある。

2. 目　　的

本研究では ptrace よりも高速にメモリの読み書きの行えるライブラリの実装を目的とする。具体的にはトレーサーとアプリケーションの間に共有メモリを作り、それを用いて一度にデータをやり取りさせるライブラリを実装する。

このライブラリによって ptrace のみでデータの読み書きを行うよりもコンテキストスイッチの回数を抑えることが可能となりオーバーヘッドが減少すると考えられる。ユーザーレベルのライブラリとして実装するため、管理者でなくとも扱える。

3. 設 計 方 針

まずトレーサーは mmap でファイルのマッピングを行う。その後アプリケーションに対してコード注入を行い同じファイルを mmap でマッピングさせ、トレーサーとアプリケーションの間に共有メモリを作る。メモリの読み書きは以下の関数を呼び出して行う。

- int fast_peekdata(pid_t pid, void *addr, void *data, int len)

この関数はプロセス ID:pid のプロセスのメモリアドレス data から長さ len を addr に格納する関数である。動作としてはコード注入によってアプリケーションに指定アドレスのデータを共有メモリにコピーさせる。その後トレーサーは共有メモリに移されたデータを格納アドレスにコピーする (図 1)。

- int fast_pokedata(pid_t pid, void *addr, void *data, int len)

この関数は data にある長さ len のデータをプロセス ID:pid のプロセスのメモリアドレス addr に書き込む関数である。動作としては書き込むデータを共有メモリにコピーする。その後アプリケーションにコード注入で共有メモリの中身を指定アドレスにコピーさせる命令を実行させる (図 2)。

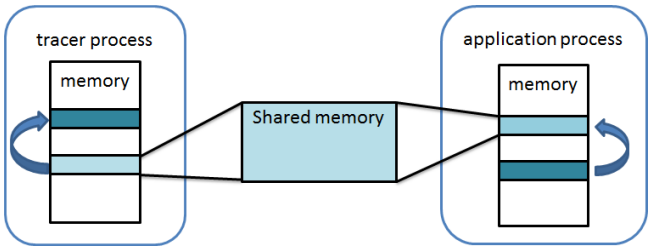

図 1　ライブラリ概要図

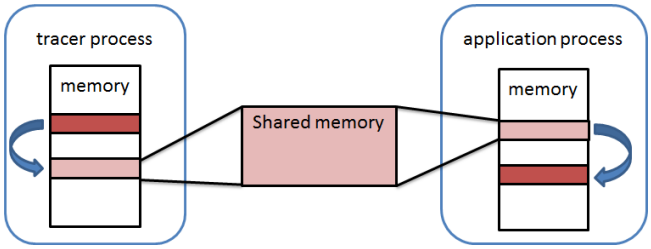

図 2　ライブラリ概要図

†1 電気通信大学 電気通信学部 情報工学科

4. 関 連 研 究

ptrace によるメモリの読み書きの改善に関する研究としてファイルシステムフレームワーク Goanna[1)] がある。この研究では ptrace によるメモリの読み書きを/proc/[pid]/mem を読み書きすることで改善を行っている。

5. 現状と今後の予定

現状は、トーレス中のプロセスに対してコード注入を行い mmap によるファイルのマッピングの実装を行った。またトレーサーと正しく共有メモリが張られているか実験も行い、動作の確認を行った。

今後は、ライブラリの実装を行いオーバーヘッドの測定を行う予定である。またライブラリを用いずに ptrace のみを用いた場合とのオーバーヘッドの比較を行う予定である。さらにライブラリの ptrace 互換 API の実装も行いたいと考えている。

参 考 文 献

1) Richard P. Spillane, Charles P. Wright, Gopalan Sivathanu, Erez Zadok ”Rapid File System Development Using ptrace”, ExpCS ’07 Proceedings of the 2007 workshop on Experimental computer science Article No. 22