

# ソフトウェア実行環境の特定機能のユーザレベルでの実現と高速化

1180300 大西 史洋 【 分散処理 OS 研究室 】

## 1 はじめに

広域分散システムでは、効率的な運用のために別の計算機へ応用アプリケーション (以下 AP と呼ぶ) の実行環境を移送する技術が重要となっている。実行環境を移送する方法として、実行環境を特定する方法が提案されているが、OS アップデートへの対応が困難となる問題がある [1]。また、移送対象が増加すると、実行環境の特定時間が増加する問題もある。

上記問題を解決するために、本研究では特定機能をユーザレベルで実現する方法を提案する。また、移送対象を特定する追跡機能の高速化を図る。

## 2 提案手法

### 2.1 特定機能のユーザレベルでの実現

提案する特定機能は、監視機能と追跡機能からなる。図 1 は特定機能の処理の流れを表している。

監視機能は、AP が open 関数を実行するとき、システムコールライブラリの open 関数が呼び出されることに着目し、open 関数内部でファイルアクセスを監視する。AP がファイルをオープンするとき、システムコールライブラリの open 関数が呼び出され、open するために必要なデータが渡される。そこで、open 関数内で AP 名とファイル名およびオープンフラグを取得し、名前付きパイプを介して監視追跡プロセスへファイルアクセスのデータを送信しデータを蓄積する。一定数データが蓄積されると、データをログファイルへ書き出す。

追跡機能は、追跡プロセスに対して外部からの追跡開始命令により実行される。このとき、移送対象の AP 名が与えられる。追跡プロセスはログファイルからデータを取得し、取得データより、移送する必要がある AP とファイルを特定する。

### 2.2 追跡機能の高速化

既存手法 [1] では、データの入力数に対して追跡機能の実行時間が大きく増加する問題があるため、追跡機能の改善を行った。

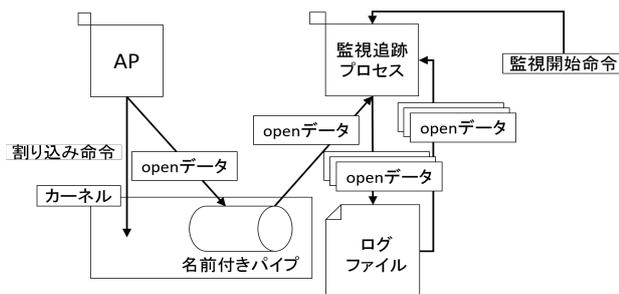


図 1 特定機能の処理の流れ

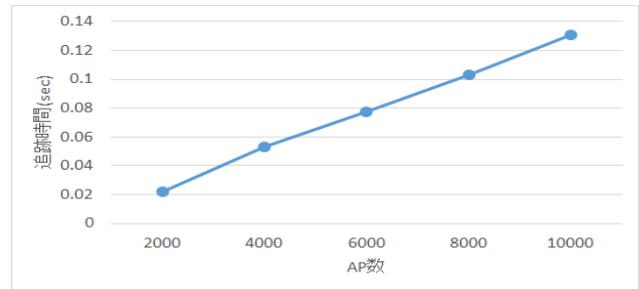


図 2 追跡機能の実行時間

追跡機能の改善点として、ログファイルからデータを取得後、データのソートを実行し、AP がどのファイルを開いているかを高速に追跡できる。追跡が終了した AP のファイルのアクセス履歴を削除することにより、再度追跡対象となることを避け、追跡回数の減少を実現している。

## 3 評価

監視機能を実現していない open 関数と監視を実行している open 関数のオーバーヘッドおよび追跡機能の実行時間を計測した。

監視機能のオーバーヘッドの計測結果は、監視機能を実行していない open 関数の処理時間は 25.6  $\mu$ sec、監視機能を実行している open 関数のオーバーヘッドは 72.4  $\mu$ sec となった。計測結果より、実行しているときのオーバーヘッドは、実行していないときより約 2.8 倍大きいことがわかる。これは、本来の open 処理のほかに、名前付きパイプを開く処理のオーバーヘッドが影響していると考えられる。追跡機能の実行時間の計測結果は図 2 となった。図 2 より、AP が 10000 個のとき、約 0.13 秒となることがわかる。既存手法 [1] では、約 20 秒ほどの実行時間だった。実験データが異なるため、単純に比較はできないが、約 150 倍の速度向上となる。

## 4 おわりに

本研究では、特定機能をユーザレベルで実現する手法を示した。また、特定機能内の追跡機能を改善し、追跡処理の高速化に成功した。

## 参考文献

- [1] 畑 翔太, 谷村 直哉, 横山 和俊, “ファイルの共有関係に着目した移送するプログラムと実行環境の特定方法”, 情報処理学会研究報告, Vol.2015-DPS-164, No.11, pp.1-6, 2015.