

ファイルの共有関係に着目したアプリケーション実行環境の移送方法

畑 翔太¹ 高田 僚介¹ 谷村 直哉¹ 横山 和俊¹

概要：広域分散システム上でアプリケーションとその実行環境を移送する代表的な手法として仮想マシンを用いる方法がある。しかし、この方法ではディスクイメージをそのまま移送する必要があり、移送時間が増大するといった問題がある。本稿ではアプリケーションが使用している資源を特定することで、そのアプリケーションと資源のみを移送する方法について提案を行う。提案手法は、アプリケーションが利用する資源が他のアプリケーションからも利用されている可能性があることを考慮し、資源の共有関係を追跡し、一緒に移送しなければならない資源を特定する特徴を有している。

1. はじめに

近年、広域分散システムの技術的發展により、クラウドコンピューティングが広く普及している。これらのシステム形態ではアプリケーション（APと呼ぶ）の実行する計算機数を負荷に応じて動的に増減させることや、APを実行する計算機を変更し再配置することが頻繁に行われる。このため、広域分散システム環境ではAPとその実行環境を移送する技術が重要となっている。広域分散システムでAPとその実行環境を移送する代表的な手法として、仮想マシンを用いる手法がある。しかし、移送処理を行うのに多大な時間が必要となるのが問題である。

本研究では、APが必要最小限の実行環境のみを移送する技術の確立を行う。具体的には、APが発行するシステムコールを監視することで利用する資源を把握し、その資源のみを移送する手法を提案する。また、APが利用する資源は、他のAPからも利用されている可能性がある。このため、資源の共有関係に着目し、一緒に移送しなければならない資源を追跡する手法を提案する。

2. 移送モデル

2.1 課題

APが利用する資源は、(1)ファイル、(2)プロセス間通信、(3)ネットワーク通信の3つに分類できる。通常、APは単独で処理を実行するのではなく、他のAPと協調しながら処理を実行する。つまり、他のAPとファイルを共有したり、通信を行いながら処理を実行する。

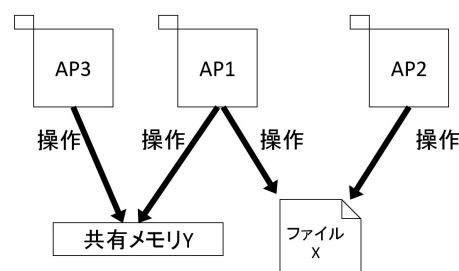


図 1 AP の実行例

図 1 は、3つのAPが資源を共有し実行している例である。AP1とAP2は、共にファイルXを利用している。AP1とファイルXを他の計算機に移送する場合、ファイルXを移送すると移送元の計算機でAP2の実行ができなくなるため、AP2も移送する必要がある。また、AP1とAP3は、共有メモリYを用いたプロセス間通信を行っている。この場合も、AP1のみを移送することはできず、AP3も一緒に移送する必要がある。この例で示したように、あるAPを移送する時、そのAPが利用している資源だけでなく、資源を共有している他のAPとその実行環境を移送する必要がある。つまり、APが利用する資源とその共有関係を追跡し、移送する資源を決定する必要がある。本稿では、利用する資源をファイルに限定した場合について、APが利用している資源とそれらの共有関係の追跡方法を提案する。

2.2 ファイルアクセスの共有関係に基づく移送モデル

ファイルの共有関係に着目した移送モデルについて説明を行う。APのファイル操作は、読み込みのみ（ReadOnlyと略す）と読み書き（ReadWriteと略す）の二種類に分類す

¹ 高知工科大学
Kochi University of Technology.

ることができる。

この操作種別により他の AP を移送するかどうかを判断する。AP1 と AP2 がファイル X を共有し AP1 を移送するとき、2 つの AP がファイル X に対して行う操作の組み合わせにより、AP2 を移送するか否かの判断は以下の 2 つのケースに分類することができる。

(ケース 1) 両者が ReadOnly の場合、AP1 と AP2 は、双方ともファイル X を介して情報を授受する可能性が無い。よって、AP1 を独立して移送することができる。しかし、ファイル X を移送した場合、移送元で AP2 が実行できなくなる。このため、ファイル X をコピーして移送元に残す必要がある。

(ケース 2) 両者のどちらかが ReadWrite の場合、どちらかの AP がファイル X に書き込み、もう片方の AP が読み込み実行する可能性がある。よって、AP1 と AP2 は依存関係にあるので、AP2 を移送対象にする。

3. 実現方式

提案手法を実現するためには、AP が操作するファイルとその共有関係を把握する機能と、共有関係を追跡し、移送対象を決定する機能が必要である。

3.1 ファイルの操作と共有関係の把握

AP がファイル进行操作する際は、操作するファイルを指定し open システムコールを発行する。ここでは、その open システムコールを監視し、AP が利用しているファイルとその共有関係を把握する。UNIX 系 OS の open システムコールの仕様を以下に示す。

```
int open(const char *pathname, int flags);
```

操作するファイルは、open システムコールの引数である pathname により特定できる。ファイルの共有関係は、flags を監視することで実現する。flags は、O_RDONLY, O_WRONLY, O_RDWR の 3 種類の値を指定することができる。ここで、O_RDONLY が指定された場合は、ReadOnly と分類する。一方、O_WRONLY・O_RDWR が指定された場合は、書き込みがあるため、ReadWrite として分類する。

3.2 移送対象ファイルの追跡アルゴリズム

2. で示した共有関係の追跡をするアルゴリズムを示す。この追跡アルゴリズムは、利用者が移送したい AP (起点 AP と呼ぶ) を指定することで開始される。

(ステップ 1) 探索対象 AP が open しているファイル全てについて以下の操作を行う。最初は起点 AP から開始する。

- (1) 探索対象 AP が ReadOnly で操作している場合
 - (a) 探索対象 AP のみが該当ファイル进行操作している場合、探索対象 AP と該当ファイルを移送する。

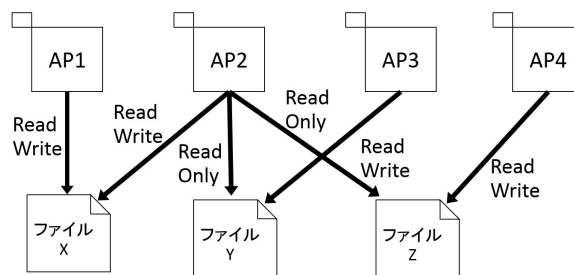


図 2 資源追跡の例

- (b) 他のすべての AP が ReadOnly で操作をしている場合、探索対象 AP と該当ファイルを移送し、該当ファイルのコピーを残す。
 - (c) 他の AP が 1 つでも ReadWrite で操作をしている場合、探索対象 AP、その他全ての AP、及び該当ファイルを移送対象とする。
- (2) 探索対象 AP が ReadWrite で操作している場合
- (a) 探索対象 AP のみが該当ファイル进行操作している場合、探索対象 AP と該当ファイルを移送する。
 - (b) その他の AP が一つでも操作している場合、探索対象 AP、その他全ての AP、及び該当ファイルを移送対象とする
- (ステップ 2) 新たに移送対象となった AP についてステップ 1 のアルゴリズムを適用し実行する。

図 2 に、共有関係を追跡する例を示す。AP1 と AP2 は、ファイル X を ReadWrite で操作している。また、AP2 と AP3 はファイル Y を ReadOnly で操作し、AP2 と AP4 は、ファイル Z を ReadWrite で操作する。アルゴリズムの動作は以下の通りである。

- (i) 起点 AP である AP1 は AP2 とファイル X を ReadWrite で共有している。よって、(ステップ 1) の (2)(b) の処理が適用され、AP2 は移送対象となる。
- (ii) AP2 が移送対象となったため、AP2 に対して (ステップ 1) が適用される。AP2 はファイル Y と Z を操作している。
- (iii) ファイル Y は AP2 と AP3 に ReadOnly で共有されている。よって、(1)(b) の処理が適用され、ファイル Y は移送すると共にコピーして残す。
- (iv) ファイル Z は AP2 と AP4 に ReadWrite で共有されている。よって、(2)(b) の処理が適用され、AP4 は移送対象となる。

4. おわりに

本稿では、ファイル操作を行う open システムコールを監視することで、AP が利用するファイルとその共有関係を把握する方法を示した。また、共有関係を用いることで、移送するファイルとその実行環境の追跡を行う方法を示した。今後は提案手法を実装し、評価を実施する。