小規模ネットワーク環境におけるサーバのマシン間移転機構の設計と実装

中村禎宏　安田絹子　多田好克
電気通信大学大学院情報システム学研究科

1. はじめに

ネットワーク上には Web、Mail、DNS などの様々なサービスが存在し、それらはネットワーク機能を提供する上で必要不可欠なものである。そのためネットワーク管理者はサーバが停止してしまう事を考慮し、ミラーリングなどによる多重化で対応している。しかし、小規模の環境ではそのようなコストをかけられない場合が多い。

本研究では稼動中のサーバに対してサーバプログラムが利用する外部コマンドやライブラリのディレクトリの配置情報などを取得し、その情報を基に他の計算機上で臨時的なサーバの代行を実現する。また、その際、情報取得の自動化により手間の削減を行う。

以下、この論文では、2章でシステムの概要を説明する。次に、移転すべき情報に関する考察を3章で、本システムで採用した移転情報取得法とその他の実装を4章で、それぞれ議論する。また、5章で本システムの評価と今後の課題を述べる。

2. システム概要

本システムでは専用計算機などを用いず、手間のかからないサービスの代行を実現する。そのため、サーバを稼動させるための各種情報をサーバ計算機から取得し、その情報を基に他の計算機上へサーバを移転させ、新たにサーバ環境を構築するという処理を行う。

各種情報には、サーバプログラム本体のみではなく、サーバが稼動するために必要となるファイル群のディレクトリ内配置情報やサーバが利用する外部コマンド、ライブラリ関数なども含まれる。通常、これらの情報の抽出には多くの手間を要し、それらの情報の必要と不必要の切り分けを明確に行わなければ、データ量は増加し、移転にかかる時間の増加につながってしまう。そのため、本システムでは各種情報の切り分けと取得の自動化を実現している。

3. 移転すべき情報

ネットワーク機能には様々なものがあるが、ここでは、Web、Mail、DNS を対象として移転のために取得しなければならない情報を考える。

現在、サーバプログラムはパッケージ化して配布されているものが多く、これらのサーバパッケージは、サーバプログラム本体、設定ファイル、ライブラリ、マニュアルなどを含む。また、これら以外にサーバプログラムが使用するOSの標準コマンドやサーバが動作する過程で生成、配置されたログファイル等のファイルも動作に必要となる。

システム設計にあたり、実際にいくつかのサーバパッケージについて調査を行った。サーバパッケージのディレクトリの配置は製作者によって異なっており、同一サーバプログラムだとしてもパッケージの種類によってディレクトリの配置は異なっている。

4. システムの設計、実装

サーバ代行までの処理の流れは以下のようになる(図1)。

(1) サーバを代行先で稼動させるために必要な各種情報の取得
(2) ファイル群と情報のアーカイブ、代行先への転送
(3) サーバの代行環境の構築

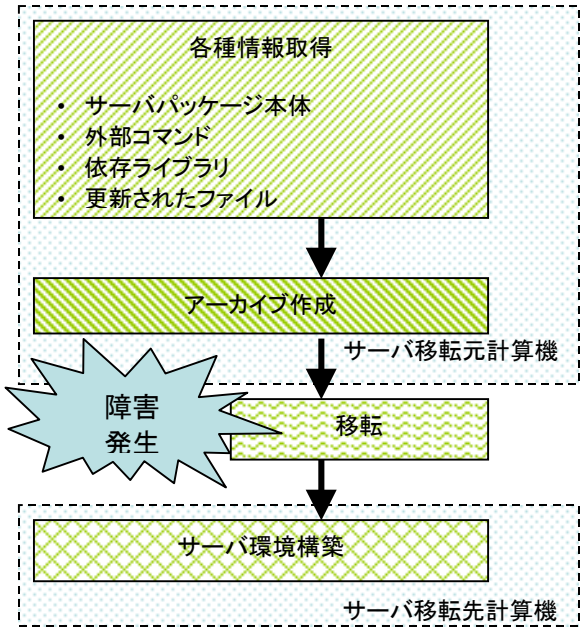

図1. サーバ移転の流れ

設計方針として、1)複数のサーバを一台の計算機で代行できるように転送するデータ量の削減を行う、2)複数のサーバが一台の計算機環境で影響を与えあわないようにサーバ環境を隔離する、3)複数の OS で対応できるように特定の OS に大きく依存した設計は行わないことを目指した。

実装においては、サーバ用計算機に多く使われていることから Linux 上で実装を行った。また、当面は移転元と移転先で同一の OS が動作していることを仮定する。

4．1 ファイル群のディレクトリ情報の取得

4．1．1 サーバパッケージ本体

パッケージの移転にはその対象となるパッケージの情報を取得する必要がある。多くの OS で使われているパッケージ管理ソフトを用いて、こうしたパッケージ情報を取得する。実装したシステムでは Linux で採用されているRPM（RedHat Package Manager）のパッケージを使用した。しかし、RPMによるパッケージ内のファイル、ディレクトリ配置のリストは当然ながらインストール後にユーザ自ら生成したファイルなどを考慮していない。そのため、それらのファイルについては特別な対応方法はなく別途リストに追加を行っている。

4．1．2 外部コマンド

パッケージ管理ソフトはプログラムの起動、終了をシェルスクリプトで行っている。それらのシェルスクリプト内では、サーバ起動に関する初期化処理を行っており、その過程に touch、echo など様々なコマンドが使用される。サーバ代行環境を構築するには、こうした必要なコマンド情報も取得しなければならない。

スクリプト内で使用されるコマンド情報を抽出するため、プロセストレースツールを使用している。これにより、稼動しているプロセスに対して終了するまでにプロセス内で呼び出されるシステムコール情報を抽出し、その引数情報から必要な外部ファイルを自動的に抽出するようにした。

4．1．3 依存ライブラリ

プログラムはディスク上に実行ファイルとして保存されている。プログラム中の関数の多くは、どのプログラマにも利用可能なサービスルーチンのオブジェクトコードを含む共有ライブラリを使用している。そのため、サーバプログラムを動作させるには対象となるサーバパッケージ内の実行可能ファイルが依存している全共有ライブラリの情報を抽出する必要がある。一般的な OS には、こうした共有のライブラリ情報を調べるコマンドが付属しているので、本研究ではこれらのコマンドを活用して必要な情報を得ることにした。具体的には、Linux 標準のlddコマンドを使用している。

4．1．4 ファイルの追加、更新

サーバの中には Web サーバのようにファイル更新が頻繁に行われるものがある。本システムではファイル群のディレクトリ情報を取得した後、その情報を基にアーカイブを生成して代行先の計算機へ転送する。この処理の後のファイル更新については定期的にサーバのファイル群に対して更新情報を調べ、更新されたファイルをアーカイブに追加する。

4．2 サービス代行

サーバプログラムを稼動させるために必要となる全ファイル群の情報を基にサーバの計算機間移転が行われる。サーバ復旧後、移転先の環境を削除した際に、サーバの残骸を残さないために、運用時は chroot で隔離して稼動する。こうすることで、代行先の計算機環境に影響を与えることなく一台の計算機で複数のサービスを代行させることができる。また、移転先と移転元の環境において同じライブラリ、ファイルがある場合、移転先に予め保持してあるものを使用することで、転送データ量の削減を図った。

計算機間でサーバを移転する際に IP アドレスの変更が必要となる。計算機が停止する場合に対しては IPalias を用いることで解決することができるが、移転元の計算機が稼動したままサーバプロセスのみが停止した場合、移転先の計算機にサービスを代行させる前に何らかの処理が必要となる。

5． 評価と今後の課題

本システムはシェルスクリプトを使用して実装を行った。代行までの処理時間の結果は以下のようになった(表 1)。

表1、代行までの処理時間

	DNS	Web	Mail
転送データ量(MB)	15.64	16.20	11.38
取得+アーカイブ(s)	5.5+5.3	15.1+5.9	3.4+4.2
環境構築(s)	4.62	6.63	2.97

表1では各サーバのアーカイブを行ったデータ量、各情報取得時間とアーカイブ時間、移転先での環境構築時間を示している。Web サーバの情報取得時間が他のサーバに比べて多くかかっているが、これは全ファイル数と実行可能ファイル数が他に比べて多いためである。また、情報取得とアーカイブは 1 度しか発生しないため、移転時の負担とはならない。

本システムでは Linux 上で実装を行ったが Linux 独自の機能を使用していないため、他の OS 上でも一部の修正のみで実現可能である。また、サーバダウンの自動検出、IP アドレスに関する問題は未解決である。

6． まとめ

本研究ではサーバパッケージの計算機間の移転および代行を行うためのシステムを設計し、Linux 上で実装を行った。いくつかの未解決問題も残っているが、同じOSが稼動していれば、特別に代行専用の計算機を確保することなく、ほぼ自動的なサーバの移転、代行を実現することが出来た。また、評価によって低オーバヘッドであることを示すことが出来た。

参考文献

[1] Daniel P.Bovet Macro Cesati , “詳細 Linux カーネル”, オライリー・ジャパン, 2002