

ソフトウェア配布システム RSU の開発

荒川修一 田中功一
三菱電機(株) 情報システム研究所

武田光世 松井陽子
三菱電機(株) 情報システム製作所

近年、急速に発展し、大規模化している分散システム環境において、ソフトウェアのメンテナンス作業は、管理者に大きな負担を与える一因となっている。我々は、それら作業の省力化を図るため RSU の開発を行った。このシステムでは、サーバとして位置付けられた計算機から複数のクライアントに対して、ソフトウェアの転送/インストールを行う。RSU は、エンジニアリングワークステーションおよびパーソナルコンピュータ上に実装され、階層化された大規模なネットワークシステムでの運用を考慮している。

Development of Software Distribution System RSU

Shuichi ARAKAWA, Kouichi TANAKA
Mitsubishi Electric Corp.
Computer & Information Systems Laboratory
5-1-1 Ofuna, Kamakura, Kanagawa 247, Japan

Mitsuyo TAKEDA, Youko MATSUI
Mitsubishi Electric Corp.
Information Systems Engineering Center
325 Kamimachiya, Kamakura, Kanagawa 247, Japan

In recent years, large distributed system environments have rapidly come into wide use. One of the hardest tasks for administrators of such large systems is software maintenance. In order to reduce and simplify the administrators tasks, we have developed "Remote Software Update service(RSU)" which is a software distribution system to transfer and install software from a server to some clients. RSU is implemented on top of engineering workstation and personal computer platforms, and it provides the necessary support to cope with large layered networks.

1 はじめに

近年、急速に発展する分散コンピューティング環境において、ソフトウェアのメンテナンス作業が大幅に増加しており、その作業の省力化が切望されている。

従来、計算機上に置かれたソフトウェアのバージョン・アップ等のメンテナンス作業は、各計算機管理者がフロッピーディスク、DAT、CD-ROM等の媒体により個々に行っていた。しかし、分散環境の発展と共に、それらの作業は計算機管理者にとって大きな負担となっている。そこで、各管理者が行っていたソフトウェアのメンテナンス作業を省力化するための仕組みとしてソフトウェア配信技術が注目されている。

我々は、階層的に接続されたネットワークに柔軟に対応するソフトウェア配布システム、RSU(Remote Software Update service: 以下RSUという)の開発を行ったので報告する。

2 背景

2.1 ソフトウェア配信

分散システムの大規模化に伴う計算機台数の増加により、ソフトウェアのメンテナンスやバージョン管理作業量が増加し、その省力化技術の1つとして、ソフトウェア配信システムが開発された。

ソフトウェア配信とは、ネットワークによって接続された複数の計算機に対して、サーバと呼ばれる計算機からソフトウェアをクライアントに自動的に配布する機構である。ソフトウェアを配布するには、まず関連するファイルをパッケージと呼ぶ単位にまとめ、配布元マシンに格納する。その後、ユーザが指定した時間、配布先マシン名、スケジュール等に従って、配布元マシンがパッケージを配布先のマシンに送信する。送信を受けたマシン(クライアント)では、パッケージを元の形に展開し、インストールを行う(図1)。

ソフトウェア配信システムにより得られる効果には、以下の様なものがある。

- 新規ソフトウェアのインストール作業の省力化
- ソフトウェアのバージョンアップの省力化
- バージョン管理

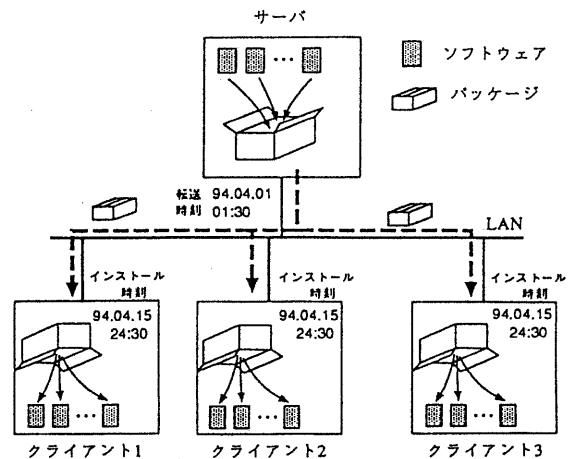


図1: ソフトウェア配信の概要

- サーバによるソフトウェアの一元管理化
- メディア (FD,DAT,MT 等) が不要

現在ソフトウェア配信製品は、ネットワーク管理技術の1つとして、多くの研究開発がなされており、オブジェクト指向技術、高レベルなセキュリティの提供やマルチプラットフォームへの対応などが技術キーワードとなっている。

2.2 ソフトウェア配信の抱える問題点

現在のソフトウェア配信システムには、以下に示すような解決すべきテーマが存在する。

1. サーバの負荷分散
2. ネットワークの負荷を考慮した配信
3. パソコン(PC)の対応
4. 異機種が混在した環境での運用
5. OSを含めた配信
6. DME等の標準化動向への追従

2.3 本開発の目的

我々は、大規模ネットワークでの適用、サーバの負荷分散、複数機種への対応やPCのサポート等に着目し、ソフトウェア配信システムRSUの設計とその実装を行った。

3 RSU の設計と実装

3.1 RSU の設計

ここでは、以下の5つの課題について考察を行い、それらを実現するための検討を行った。以下に、その検討結果について述べる。

(1) サーバの負荷分散

システムの増大に伴い、サーバに掛かる負荷は線形的に高くなってしまい、配信処理速度の低下につながる。従って、極力サーバに負荷をかけないような配信の仕組みが必要となる。

ここでは、サーバからクライアントへは、ソフトウェアではなくスクリプトを転送し、そのスクリプトをクライアント上で起動することにより、ソフトウェアの転送及びインストール処理を行う事とした。また、サーバはスクリプトを起動するための配信スケジューリング、配信状況の管理等を行うのみとすることにより、機能分散を図るようになった。

これにより、従来サーバが自動的に行っていたソフトウェアの転送/インストール処理の起動及びスケジューリングをクライアントが独自に行うこととなり、サーバの負荷を軽減することができる。

(2) 階層構造を持つネットワークへの対応

上述のスクリプトによるクライアントへの配信処理の委譲方法に加え、以下の図2に示したような階層構造のネットワークにも対応可能になるよう処理のリレーが行えるようにした。

図2では、まずサーバからそのクライアントとなる中間サーバに対してスクリプトを転送している。そして、中間サーバ上で転送されたスクリプトを起動することにより、中間サーバ上にソフトウェアを転送し、必要であれば中間サーバ上にインストールを行う。次に、中間サーバからクライアントに対しても先ほどの手順と同様にスクリプトを転送し、それをクライアント上で起動する。

階層構造に対応可能となることによって、(1)で述べたサーバの負荷の軽減にもつながることとなる。

(3) 複数機種への対応

分散システム環境では、一般的に異機種が混在したシステムを構成している場合がほとんどであ

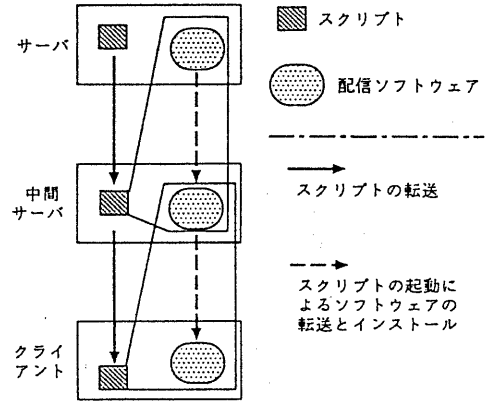


図 2: 階層構造への対応

る。

従来、サーバは予めクライアントの機種タイプを認識しているか、あるいは、サーバがクライアントの機種タイプをその都度判別することによって、複数機種のクライアントに対応していた。

我々は、配信処理のクライアントへの委譲によるサーバの負荷の軽減や階層構造の場合のサーバから直接通信しないクライアントの機種タイプの判別に対応するため、スクリプト内に機種判別を行う処理を組み込むことによって、各クライアントが自主的に判断する仕組みを取り入れた。

また、機種毎に適したソフトウェアの転送及びインストール処理方法を記述したスクリプトを作成することによって、各機種に対応した配信を可能とした(図3)。

(4) PC への対応

現在の分散システム環境では、ネットワーク接続されたPCもシステムの一部として使用される。我々は、オフィスで比較的多く使用されているPCのクライアント化機能の検討を行った。PCのOSとしては現在、利用頻度も高いMS-DOSを考えた。しかし、MS-DOSをOSとしたPCでは、基本的にシングルタスクであるため非同期な処理を行えない。そのため、サーバとのデータのやり取りの方法を別途検討する必要があった。

サーバとPC間の通信方法としては、以下の3通りが考えられる。

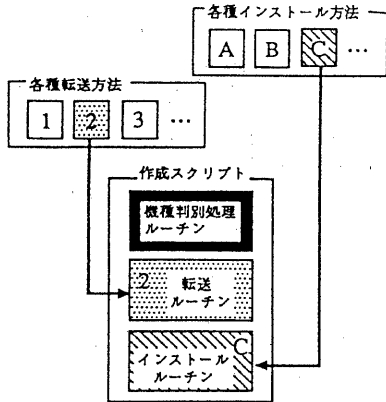


図 3: 複数機種への対応

- a. ソケット通信
- b. ファイルの共有機能
ネットワーク OS や PC-NFS 等の機能により疑似的に配信

上記方式のうち、以下の理由により、本開発におけるサーバ - PC 間の通信方式としてソケット通信による方式を選択した。

- (i) NFS 等のファイルの共有機能を用いた場合、サーバ停止時には PC のアプリケーションが使用不可能となること、カスタマイズ等の処理実行の方式が、別途必要となる。
- (ii) NFS 等のファイルの共有機能では、階層構造ネットワークの配信に対応不可能

更に、前述の図 2 におけるクライアントが、PC などのサーバ及び中間サーバとは異なる OS の機種の場合、サーバからクライアントに送られるスクリプトがサーバ用スクリプトあるいはクライアント用スクリプトだけでは、階層構造の配信が行えない。

そこで、以下の図 4 に示すように、サーバ用のスクリプト中にクライアントの機種に依存したのスクリプトをパッケージ化した別のスクリプトを作成することにより、中間サーバ上では、サーバ用スクリプト部分のみが実行されるようにした。そして、クライアント上では、転送されてきたスクリプト中から機種に依存したスクリプトを取り出

し、それを実行するという仕組みとすることにより、異なる機種が混在したシステム構成にも対応可能とした。

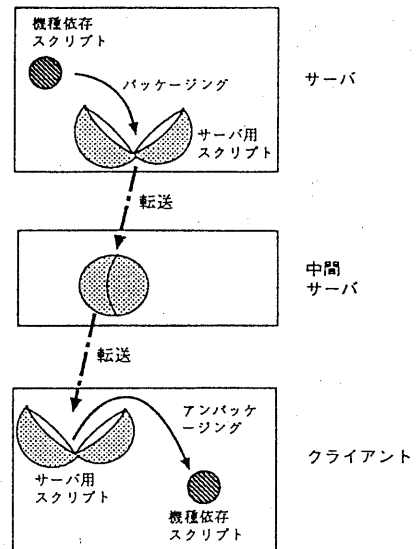


図 4: PC への配信用スクリプトの構造

- (5) ゲートウェイを含んだシステムへの対応
ゲートウェイなどによってサーバとクライアントが直接通信接続出来ない場合がある。

このような場合の対応策として、図 5 のように、ソフトウェアの配信処理を次々にリレー式に渡していくことにより、階層構造の配信を可能にした。そこで、ゲートウェイにそのリレー機能を持たせることにより、サーバからゲートウェイを経由したクライアントへの配信を行う。また、配信結果も配信パスとは逆方向にリレーされてサーバに報告出来るように考えた。

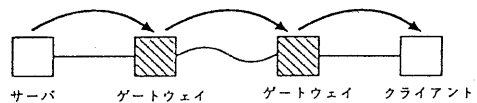


図 5: 配信処理のリレー

(6) 汎用性を考慮した API

本開発では、ソフトウェアの送信および受信の処理は、ハードコーディングせず、サーバからクライアントに対するジョブ(スクリプト)のプーリングを行うことによって実現した。ソフトウェアの転送とインストールの処理は、実際にはクライアント側で起動されるスクリプトに従って実行される。このようなスクリプトによって作業をクライアントに委譲する方式は、スクリプトの内容を自由に変更することによって、ソフトウェア配信だけでなく、他の機能の実現、システムの拡張を容易にする。

3.2 RSU の開発

3.2.1 概要

RSU は、当社製ワークステーション及び PC から構成されるシステムにおいて、ネットワークを介してアプリケーションのソフトウェア及びデータ配布を、サーバ(センタ)主導型で行うプログラムである。

RSU では、LAN 上の任意のクライアント・マシンへのプログラムの転送及びインストールを、サーバ(ホスト)から自動的に行え、サーバにおいて配布物の一括管理が可能である。また、ディスク容量の不足、該当ソフトウェア実行中等の理由により、転送/インストールが行えなかった場合の対処機能(リトライ、管理者への通知)も有している。従って、各クライアントの管理者が個別にソフトウェア管理を意識することなく、サーバからその作業を一元的に行うことにより、今まで必要とされた多量の工数を軽減することが可能である。図 6 に RSU のシステム構成を示す。

3.2.2 機能

以下に、RSU の主な機能を示す。

- 階層構造のネットワークを考慮したサーバ主導型のプログラム配布
- ソフトウェアの配布及びインストール時刻指定
- 失敗時のリトライおよび通知機能
(設定ファイルによりリトライ回数が指定可能)

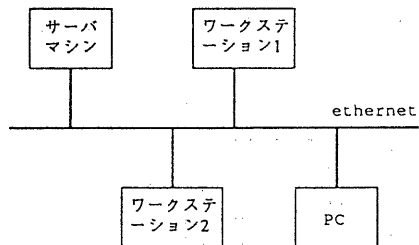


図 6: システム構成

- 各マシン管理者への通知
(配信結果等のメールによる管理者への通知)
- ログ・ファイルへの実行状況の記録
(遠隔サーバのログ・ファイル表示も可能)
- クライアントの配信状況把握
- セキュリティ機能
(ユーザ名やクライアント等の登録リストによる管理)
- PC のサポート
- Motif による GUI(図 7)

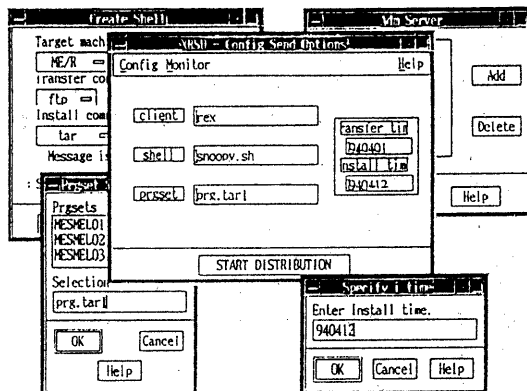


図 7: RSU スケジューラ画面例

3.3 RSU の特徴

RSU には以下のような特徴がある。

- (1) 階層構造における配信が可能

RSUでは、図8で示したように、階層構造におけるトップダウン的配信を可能にしている。

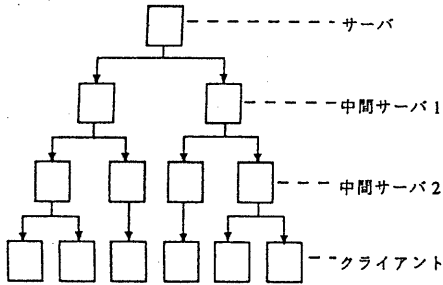


図 8: 階層構造における配信

下位層から上位層に対しては、配信状況の報告が、クライアント→中間サーバ2→中間サーバ1→サーバの順に行われる。

(2) クライアントでのプロセス起動による配信

RSUでは、サーバからクライアントへは、ジョブとシェル・スクリプトのみの転送を行い、実際のソフトウェアの転送及びインストールは、クライアント上でサーバより転送されたシェル・スクリプトを起動することにより実現されている。シェル・スクリプトというUNIXにおける標準的な仕組みを用いたことによって、他機種への展開を容易にし、スクリプトを独自に修正することによる他の機能が実現でき、柔軟性のあるシステム構築を可能とした。

(3) 機種判別

機種判別処理を組み込んだスクリプトをサーバからクライアントに対し転送し、それをクライアントで実行することにより、各クライアントが自主的に機種の判定を行うことが可能である。

(4) PCのサポート

PCのクライアント機能を提供している。サーバ-PC間の通信は、ソケットを実装した。

PCは、シングルタスクであるDOSをOSとしているため、非同期の処理を容易に実装出来ない。従って、本開発におけるPCへの配信処理は、PCのIPL時のみ実行となっている。

(5) 配信処理とソフトウェアの分離

ソフトウェア配信システムの一部には、配信す

るソフトウェアにインストール処理機能を内蔵し、配信されたソフトウェア自身が起動されることにより、インストールが実行される仕様のものがある。

しかしRSUでは、ソフトウェアの転送及びインストール処理と配信ソフトウェアを完全に分離し、転送及びインストール処理は、スクリプトにより実現している。このような方法により、汎用性のあるAPIを実現することが可能である。

(6) サーバ及び中間サーバへの配信

RSUでは、クライアントだけでなく、図8のような階層構造におけるサーバや中間サーバへのソフトウェアのインストールも可能である。

4 おわりに

我々は、ソフトウェア配信に必要とされる機能や大規模分散を考慮したシステムを構築する上で問題を検討し、それらを解決するための提案を行った。また、その提案に従いRSUの実装を行った。以下に、今後の課題を示す。

- ネットワーク管理機能とリンクしたソフトウェア配信システムの開発
- Kerberos 認証機構やACL等のより高いセキュリティ機構
- アプリケーションやデータだけでなく、OSを含めたアップデート機能

5 参考文献

- [1] 瓶家 喜代志, OSF DCE 技術解説, 株式会社ソフトウェア・リサーチ・センター, (1994).
- [2] "System Administration Interface/Software Administration", IEEE Computer Society, POSIX 1003.7.2/D.