—論文—

高岡短期大学紀要　第9巻　平成9年3月
Bull.Takaoka National College, Vol.9, March 1997

分散ネットワークにおけるサーバの集中管理法
—WWWホストへの実装—

近藤　潔・米川　覚

（平成8年11月5日受理）

要　旨

インターネットに代表される分散ネットワークシステムの普及と拡大にともない，ネットワーク上のサーバ数が増加した結果システム運営の負荷が増大している。これを解決するためにWWWサーバ上からネットワーク上の複数のサーバをリモートで管理するためのツールを開発し，それを用いて高岡短期大学のネットワークにサーバ管理システムを構築した。このツールはコマンド実行を要求するマネジャーとリモートマシンに常駐してコマンド実行を行なうエージェントで構成される。マシンによる相違を吸収する機能とコマンドを同期・非同期で実行する機能を持つとともに，GNUのシステム管理ツールであるGNU cfengineをリモートで実行することにより複雑な処理を可能とした。また，WWWサーバ上でのシステム管理はシステム関連情報を公開するという側面を持っており，システム運営の工数削減とユーザサービスの向上も期待できる。

キーワード

システム管理，サーバ管理，集中管理，SA Manager，RPC，非同期実行，GNU cfengine，WWW

1　はじめに

学内ネットワークシステムの運営の効率化はシステム部門を持たない本学にとって重要な課題である。先に筆者らはユーザのセルフサービスを可能とする利用環境の整備と処理の自動化によるシステム運営の効率化について報告した[1]。その後システムの拡大とともにサーバ数が増え，異なるOSをもつ異なるベンダーのマシンを対象にマシンごとに異なる設定ファイル等を維持管理する必要性が高まり，システム管理の業務はますます複雑になっている。

本学のシステム運営は主として数人の情報処理を専門とする教官が協力してグループで行なっており，運営そのものの効率化以外にもグループ内での意思の疎通や管理方法の共通化といった課題がある。これらの課題を解決するために以下の方針で開発を進めてきた。

1．Web上でのサーバの集中管理
- 効率化
  サーバの増加に備え，1つのWWWサーバからリモートでシステム管理を行なう仕組みを開発する。
- 共通化
  Web上で共通のツールを使用する。

2．システム情報の公開
- グループ内での情報共有

システム運営するグループ内での共通認識を育てる。

- ユーザ向けの情報公開
  一般ユーザ向けに管理情報の一部を公開し，セルフサービスを促進する。

集中管理のためのツールとして大規模なものも発表されているが[2]，対象が本学と異なり大規模ネットワークであり，設定が複雑なので，柔軟性と維持管理の容易さに重点を置いたツールを開発することにした。

2 Web上でのサーバの集中管理

2.1 集中管理の概要

本学のサーバは1台のWindows NTマシンを除いてすべてUNIXマシンなので，集中管理の対象はUNIXマシンに限定する。また，システム管理の内容を，比較的単純なシステム関連のコマンドを実行してサーバの状態を監視するような場合と種々のシステム設定についての複雑な処理を行なう場合に分け，後者については新たにコーディングする部分を少なくするために既存のツールをできるだけ利用する方針のもとに開発を行なった。

2.1.1 リモートコマンドの実行

UNIX上ではすべての操作はコマンドで実行できる。そこで，原理的には1つのマシンからリモートのマシンに対してコマンドを実行できればシステムの集中管理が可能となる。このような仕組みを実現するものとして，すでにrsh, rcpといったいわゆるrコマンドが用意されているが，システムの集中管理で用いるには以下のような問題点がある：

- セキュリティ関連の設定
  rコマンドの実行には，.rhostsやhosts.equivのようなファイルの設定が必要で，設定の手間がかかる上にこれらの設定ファイルが汎用的に用いられるためにセキュリティの問題が生ずる。
- マシンによるコマンドの相違
  OSや機種が異なると，一般にコマンドのパラメータ等に微妙な相違がでてくる。コマンドを透過的に扱うためにはこれらの相違を吸収する仕組みが必要である。

そこで，rコマンドの設定に依存せず，しかもマシンによる相違を意識せずにコマンド実行を可能にするシステム管理ツールの“SA Manager”（2.2節で詳述）を開発した。

2.1.2 システム設定関連の処理

システムのファイル，ディレクトリ，およびネットワークの設定等に関するチェックや処理の作業は複雑で，システム管理の大きな部分を占める。そこで，文献3）のGNU cfengineを利用することで開発及び保守の効率化を図った。

2.2 システム管理ツール：SA Manager

SA Manager (System Administration Manager)は図1に示すように，コマンドのリクエストを発行するマネジャーとリモートマシンでコマンドを実行するエージェントで構成される。ここで用いる「エージェント」という語は文献4）で用いられているような「行為を行なう個人」という意味ではなく，マネジャーに管理されるシステム内の管理オブジェクト[5]という意味で用いる。また，マネジャーはcfengineの設定ファイルを集中的に

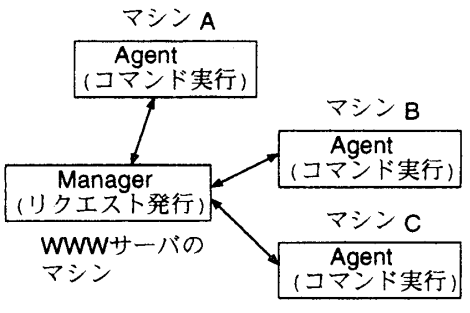

図1：SA Manager

管理して変更が生じた場合にはいつでも各エージェントに送付できるように，エージェントに対するファイル転送の機能を持っている。

2.2.1 コマンド実行の仕組み

リモートでのコマンドの実行はRPC (Remote Procedure Call)を用いて実装した[6]。

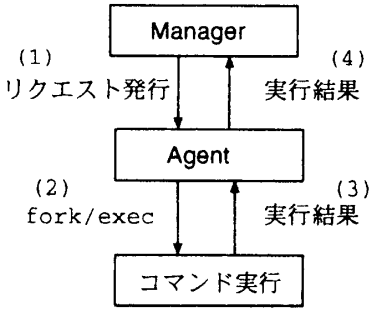

図2：コマンドの同期実行

コマンドの同期実行

マネジャーがエージェントにコマンド実行を要求する際にTCPプロトコルを用いるとエージェントが結果を返してくるまで待ち続けることになる。この同期実行の場合には一度に1つのエージェントしか対象にできないが，返してくる結果の量に対する制限はない。従って，実行時間の短い処理や大量の結果が出力される処理に適している。図2に処理の流れを示す。

以下に処理の内容を示す：

1. マネジャーから1つのリモートのエージェントにコマンド実行のリクエストを発行する。
2. エージェントは子プロセスをforkし，受けとったコマンドをexecして実行する。結果の受渡しはpipeを使用する。
3. 子プロセスは結果をpipeに出力して終了する。
4. エージェントは実行結果をマネジャーに返す。

なお，同期実行のマネジャー機能は“scsh”コマンドが行ない，scshは以下のように実行する。

% scsh　リモートホスト名　コマンド名

また，同期実行ではscshによるコマンドの実行の他にファイル転送の機能がある。これは“transf”コマンドを使い，以下のように実行する。

% transf リモートホスト名　ファイル名
〈リモートホストのパス名〉

(注)〈 〉内は省略可。省略時にはファイルはエージェントのカレントディレクトリにコピーされる。

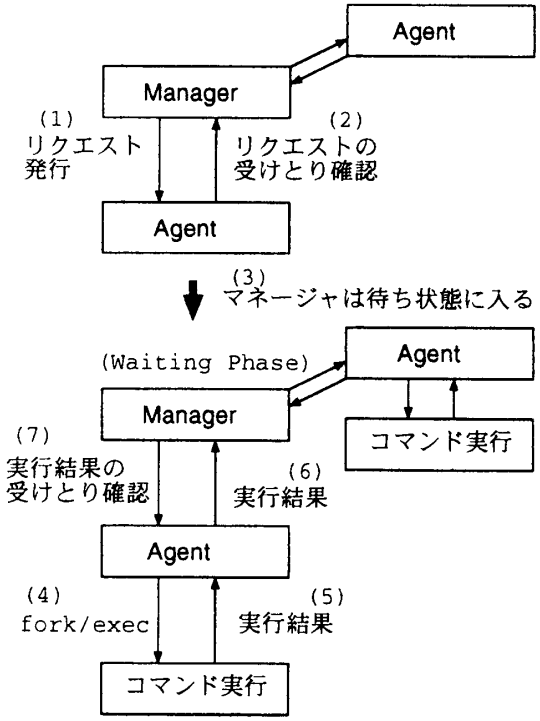

図3：コマンドの非同期実行

コマンドの非同期実行

コマンドの実行時間が長い場合には，ホストごとに順番に同期実行していてはホスト数とともに実行終了までの時間が増大してしまい，能率が悪い。そこで，UDPのプロトコルを用いてまずコマンド実行のリクエストを各ホストに発行し，コマンドの実行が完了したものから順に結果を受けとる非同期実行の機

能を実装した。処理の流れを図3に示す。

この手法はfollow-up RPC(FRPC)と呼ばれ，クライアント（マネジャー）側で新たなプロセスを生成せずに複数のサーバ（エージェント）に並列的に処理を実行させるところに特徴がある[6]。但し，UDPプロトコルの条件から，実行結果のサイズは8Kバイト以内に制限される。

以下，図3にしたがい処理の内容を示す：

1. マネジャーから複数のエージェントに対してコマンド実行のリクエストを発行する。
2. リクエストの受け取りを確認する。処理の単純化のため，リクエストを受けとらなかったエージェントは以後無視する。
3. マネジャーはエージェントの実行結果を受けとるための待ち状態に入る。待ちの間，エージェントからの応答をselect関数でチェックする。その際，タイムアウトを0に設定してpollingモードで行なう。
4. エージェントは子プロセスをforkし，受けとったコマンドをexecして実行する。結果の受渡しはpipeを使用する。
5. 子プロセスは結果をpipeに出力して終了する。
6. エージェントは実行結果をマネジャーに返す。
7. マネジャーは受けとり確認を返す。全てのエージェントからの実行結果を受けとってから処理を終了する。

なお，非同期実行のマネジャー機能は“ascsh”コマンドが行ない，ascshは以下のように実行する。

% ascsh コマンド名 ホスト名-1 ホスト名-2 ホスト名-3....

2.2.2 マシンによる相違の吸収

各マシンに固有の情報をSA Managerの設定ファイルに記述しておき，各マシンにおいてエージェントが起動時に読み込むようにした。マネジャーは全てのエージェントに対するただ1つの設定ファイルを集中管理し，変更が生じた場合は“transf”コマンドで全てのエージェントにファイル転送して更新する。

別名コマンドの設定

OSや機種が異なると，コマンドの引数やコマンドが存在するディレクトリが異なってくる。パス変数はエージェント起動時の環境を受け継ぐが，正確を期するために絶対パス指定を用いる場合もあり，絶対パスは一般にOSにより異なる。例えば，psコマンドはBSD系とSYSV系でパラメータが異なり，また，SYSV系のOSでBSD互換コマンドを使うときには，/usr/ucb/psのように絶対ディレクトリ指定が必要となる場合がある。このような環境ごとに異なる設定を以下のように行なう：

```
Cmd PS = ps -aux
kokufu, katakago:
Cmd PS = /usr/ucb/ps -aux
futagami:
Cmd PS = ps -ef
```

この例では，別名コマンドPSのデフォールトとしてBSD系コマンド形式 (ps -aux) を設定し，SYSV系のマシンであるkokufuとfutagamiではBSD互換コマンドのpsを使用し，別のSYSV系マシンであるfutagamiではSYSV系のpsコマンドを使うためにデフォールトと異なる引数を用いてPSを定義している。このように設定しておくと，システムに透過的に1つのコマンドで複数のリモートマシンのプロセスを表示することができる：

% ascsh PS kokufu katakago futagami fire

この例では，fireはBSDマシンなので，デフォールトの“ps -aux”が適用される。

ルート権限で実行禁止のコマンド

システム管理のために，エージェントにル

ート権限でコマンドを実行させたい場合がある。そのような場合のためにマネジャーがルート権限でリクエストを発行したときにはエージェントはルート権限でコマンドを実行するように設定している。しかし，rm等のコマンドを誤ってルート権限で実行するとシステムを破壊する等の危険性があるので，ルート権限での実行を禁止するコマンドを設定ファイルに書けるようにした。これは以下のように記述する：

```
CmdList rm rmdir cp mv chown
katakago:
CmdList rm rmdir
```

この例は，デフォールトでは，rm，rmdir，cp，mv，chownのルート権限での実行を禁止するが，katakagoだけはrm，rmdirのみの実行を禁止するという意味である。

その他の設定

現在，エージェントのカレントディレクトリを設定できる。今後，エージェントのユーザIDの設定やマネジャーによるコマンド実行のタイムアウト時間の設定等を追加する予定である。以下にカレントディレクトリの設定例を示す：

```
SAHome /home/saman
kokufu:
SAHome /var/www/saman
```

2.3 GNU cfengineの利用

文献3）にある通りcfengineは言語ベースのシステム管理ツールで，システム維持管理業務の自動実行と，ネットワークで結ばれたホストの設定を1つの集中ファイルで管理できることが特徴である。このツールは発表から1年程しか経過しておらず，まだ実績はあまりないがGNUのパッケージとして登録されておりFree Software Foundationが今後の維持管理の母体となることから安心して使用できる。

機能的には特にファイル関連の設定と維持管理に優れており，また適用する対象のシステム側に特別な設定変更などを必要としないので利用が容易である。機能の詳細については省略するが，本論文のシステムに組み込む際に変更・追加を行なった部分を中心に述べる。

2.3.1 RPCベースでの実行

Cfengineは図4に示すように，1台のホストで集中的に管理している設定ファイル（cfengine.conf）を各ホストがリモートマウントして共通に用いることを想定している。また，各ホストでのcfenigineの実行方法は基本的に自由であるが，cron等によりホストごとにスケジューリングして行なうか，または各ホストで管理者がコマンドにより実行することを想定している。

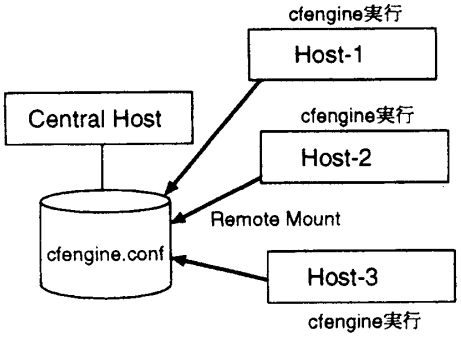

図4：cfengineの実行方法(1)

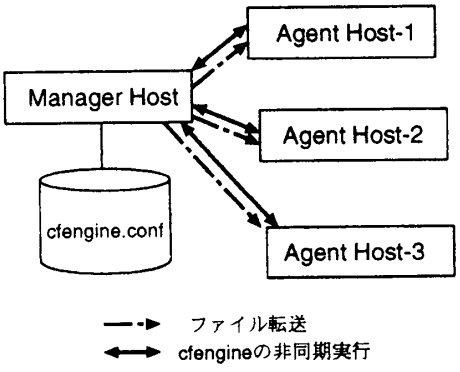

図5：cfengineの実行方法(2)

本学のネットワークは小規模ではあるが4つのサブネットに分かれており，特別に大容量のサーバもないので，全てのホストでcfengine.confをマウントして利用すると設定や維持管理が繁雑となる。また，サーバの集中管理を目的としているので1つのホストからリモートでcfengineを起動したい。そこで図5に示すようにcfengine.confファイルはSA Managerのtransfコマンドにより1つのホストから各ホストにファイル転送し，cfengineの起動はascshによりリモートで実行する方法を採用した。ここで，cfengineの実行には時間がかかる場合が多いので非同期実行のascshを用いる必要がある。

2.3.2 サポートされていないOSへの移植

本学にあるUNIXサーバのOSの種類を表1に示す。cfengineがサポートがされていないEWS-UX/V (Rel-4.2MP) とBSD/OSについてプログラムを移植した。移植の際に，configure.in，cf.defs.h，cfengine.c，classes.c，ifconf.cのファイルを変更する必要があるが，あらかじめ新しいOSが増えることを想定して作られているために比較的容易であった。

表1：OSの種類

OS名称	cfengine サポート	備　考
EWS-UX/V	×	EWS4800
Solaris2.5	○	SunOS 5.5
ESD/OS 2.1	×	PC/UNIX
FreeBSD 2.1R	○	PC/UNIX

2.4 システム情報ボード

SA Managerの機能はコマンドベースで実装してあるが，グループでシステム運営するためには使いやすいユーザインタフェースが重要である。そこで，3節で述べるシステム情報公開用のユーザインタフェースと一緒にWWWサーバ上に図6に示す「システム情報ボード」を設け，SA Managerの機能はその中の「システム管理者用ボード」から実行することにした。

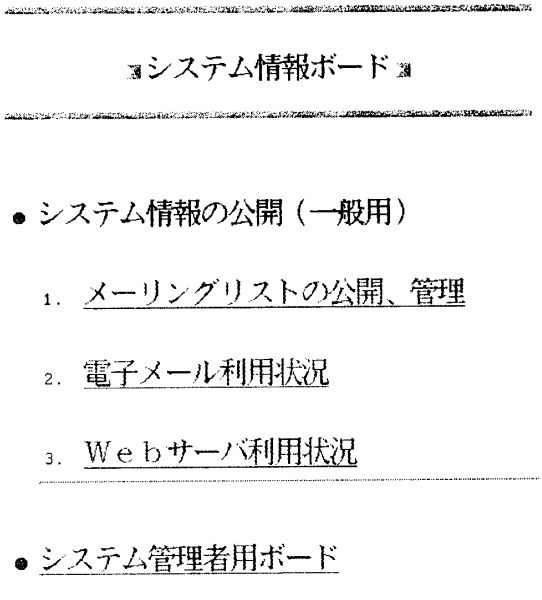

図6：システム情報ボード

2.5 セキュリティの確保

SA Managerはリモートホストに対して任意のコマンドを実行できるのでセキュリティの確保が必要である。

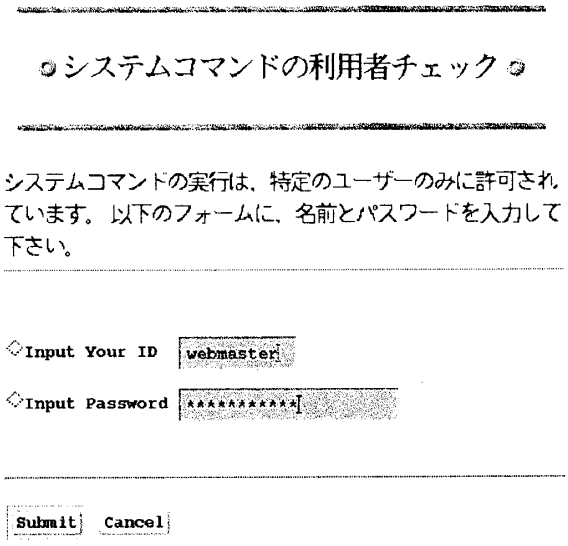

図7：ユーザ認証

2.5.1 ユーザの認証

システム管理者用ボードへのアクセスをシステム運営グループのメンバーに限定するため，パスワードによるユーザの認証を行なう。これに用いるパスワードはログイン用のパスワードと共通にすることも可能であるが，パスワードが破られた場合にシステム全体のセキュリティに波及するので独自のパスワードを用いることにした。

パスワードの登録にはNCSA httpd[7]の配布ファイルに含まれているhtpasswdと呼ぶプログラムを用いる。これは，ログイン時の認証と同様に暗号化規格(DES)[8]に基づいた手法を採用したcrypt関数を用いている。

図6でユーザが「システム管理者用ボード」を選択すると，図7の画面が現れ，ここで登録名とパスワードを入力することでユーザ認証を行なう。

データの入力にはHTMLのフォーム機能[9]を用い，認証用のCGIプログラム[10]はPerl[11]で作成した。PerlにはCの関数と同様の機能を持つcrypt関数があるのでこれを用いた。

2.5.2 ルート権限での実行

2.2.2節で述べたように，マネジャーがルートの実行権限を持っている場合にはエージェントはルート権限でコマンドを実行する。これはcfengineを使ってファイルの操作を行なう場合に必要となるが，それ以外の場合にはほとんど必要としない。WWWサーバをルートの実行権限で作動させればSA Managerもルートの実行権限で実行することになるが，汎用的に使用されるWWWサーバ自体をルートの実行権限で作動させることはセキュリティの観点から望ましくない。

そこで，以下のようにルートにSUIDしたascshのコマンド(sascsh)を作成し，WWWサーバのグループである“nobody”に属するユーザとルートのみが実行できるように設定してセキュリティに配慮した。

-rwsr-x--- 1 root nobody ... sascsh

但し，この設定ではcgiプログラムからは自由に実行できてしまうことになるのでWWWサーバ上での自作のcgiプログラムの実行を一般ユーザに開放すると問題が多い。このようなセキュリティの問題については今後も検討が必要である。

2.6 コマンド実行のユーザインタフェース

コマンド実行のユーザインタフェースは同期実行型のscsh及びtransfを使う場合と，非同期実行型のascshを使う場合に分けて作成してある。

2.6.1 同期実行のユーザインタフェース

図8に同期実行のユーザインタフェースを示す。transfによるファイル転送は汎用的に使う必要がないので，当面はcfengineの統一的な設定ファイルであるcfengine.confを各エージェントに転送する場合に固定した。実行結果の例を図9に示す。

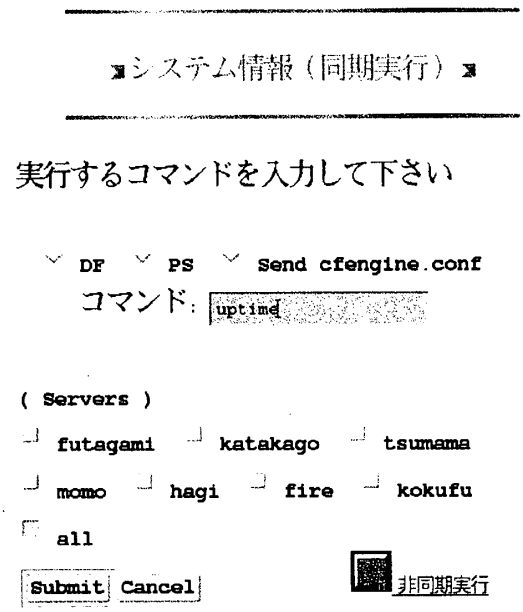

図8：コマンドの同期実行

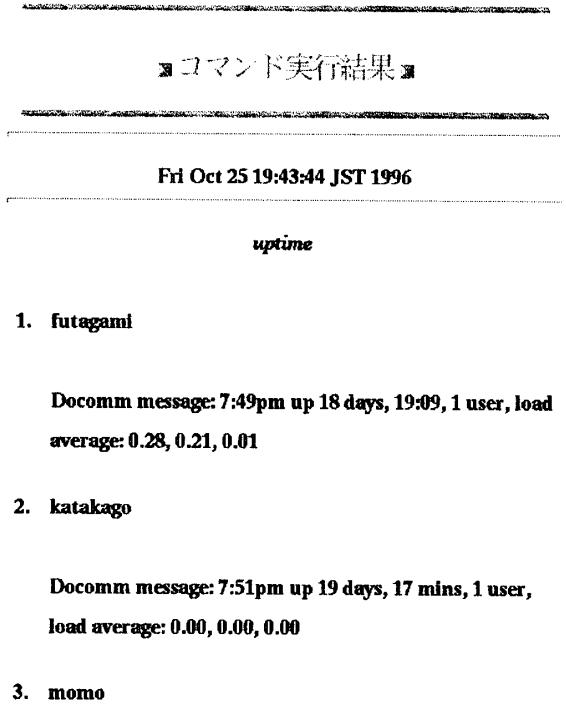

図9：同期実行の結果(例)

2.6.2 非同期実行のユーザインタフェース

図10に非同期実行のユーザインタフェースを示す。cfengine以外のコマンドはWWWサ

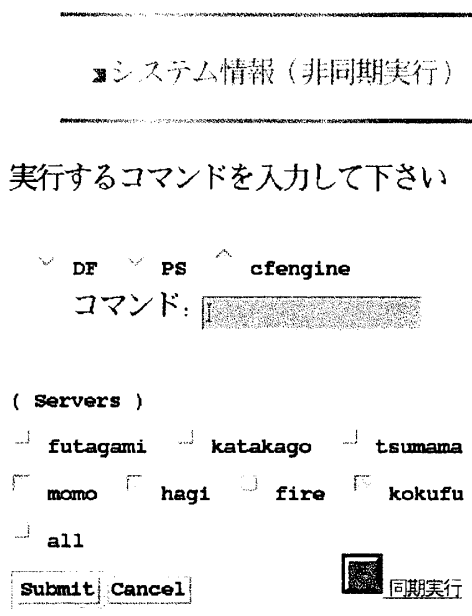

図10：コマンドの同期実行

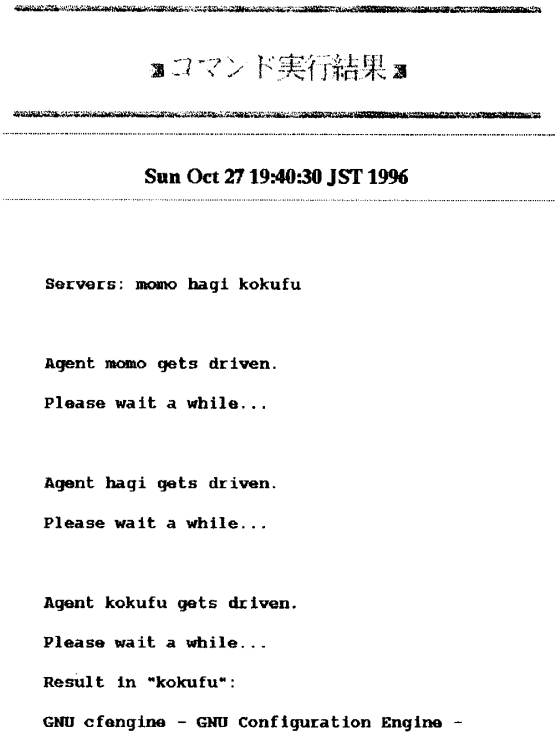

図11：非同期実行の結果(例)

一バと同じnobodyのユーザ権限で実行し，ラジオボタンの選択画面から選んだcfengineのみをルート権限で実行するようにしてセキュリティに配慮している。なお，cfengineをルート権限で実行しない場合はコマンド入力の欄を使用することで可能である。実行結果の例を図11に示す。

3 システム情報の公開

3.1 システム情報公開の目的

文献1）ではシステム管理の効率化とユーザの使い勝手の向上を目指してユーザ情報を公開し，一般ユーザが検索できるようにしたことを述べた。このシステムは自分でアカウントの情報を検索することで，個人のアカウントについてのユーザからの問い合わせをなくすことが最初の狙いであった。実際に稼働してみるとクラスの名簿作成やメーリングリストの作成等にも活用されている。

このように，システム側が管理すべきと考えられている情報の中には，一般ユーザに積

極的に公開することでユーザサービスを充実させるだけでなく，システム管理自体の工数削減につながるものがある。管理情報を公開することでユーザからの問い合わせが少なくなり結果的にシステム管理の効率が向上する効果が期待できる。また，本学では専任のシステム管理者を置かずにグループで運営を行なっていることから，システム運営のグループ内で情報を共有する効果も大きい。

3.2 公開情報の内容

公開情報は図6のシステム情報ボードから見ることができる。以下に個々の内容を紹介する。

3.2.1 メーリングリストの内容公開と自主管理

メーリングリストによる任意グループ内のコミュニケーションは今後ますます重要になると考え，メーリングリストの内容をWWW上で公開するとともにその場でメーリングリストの管理者（作成者）による変更ができるようにした。この結果メーリングリストの自主管理が容易になり，問い合わせやサポートの依頼が減るものと期待できる。

メーリングリストの表示

1. 95all
2. 96fujitasemi
3. 96kondo-semi
4. 96sato-semi
5. active-mac
6. active
7. c-club-2
8. c-club-a

図12：メーリングテスト一覧

96kondo-semiのパスワードチェック

オーナー名: kondo
パスワード: *******

Submit Cancel

図13：オーナー認証

図12にメーリングリスト一覧，図13にメーリングリストのメンバー変更のためのオーナー認証の画面を示す。オーナー認証のパスワードはセキュリティの問題が小さいため暗号化を用いず，メーリングリスト作成時に入力したパスワードとここで入力した文字をそのまま比較している。

3.2.2 電子メールの利用集計

電子メールの利用状況はLANの利用状況のバロメータとしてシステム管理者のみならず全体の関心が高い。そこで，sendmailのログを解析して電子メールの利用集計をWWW上で公開した。集計は日別，月別に行ない，表及びグラフにまとめた。図14に月別の集計グラフの例を示す（図中で，例えば学内—学外

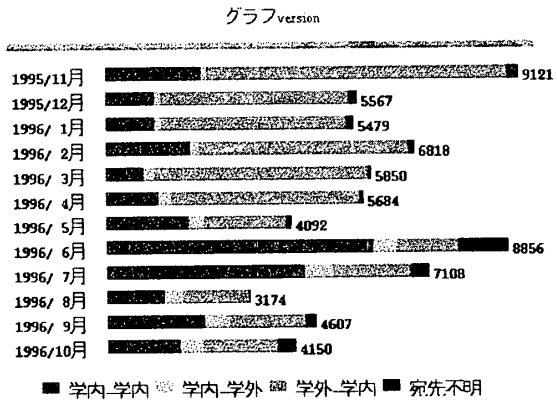

図14：月別メール利用状況図

は学内から学外へ向けて発信されたメールの意味)。なお，グラフの作成にはJava[12]を用いた。

3.2.3 WWWの利用集計

WWWサーバのアクセスログを解析し，WWWの利用集計を行なっている。集計項目は以下の通り：

1. 月別・日別・時間別アクセス件数
2. ドメイン別アクセス件数
3. コンテンツ別アクセス件数

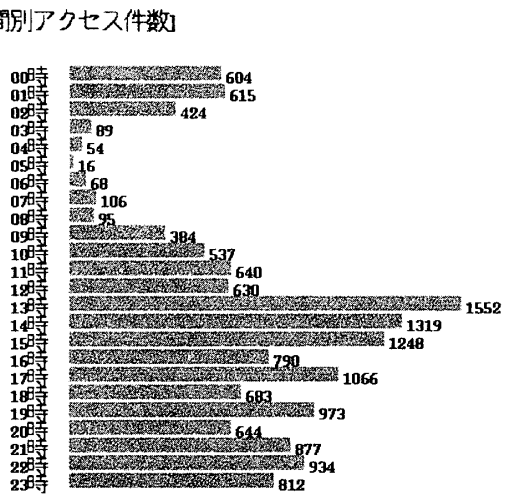

図15：時間別アクセス件数の例

図15に時間別のアクセス件数の例を示す。

WWWサーバのアクセス集計はシステム管理者のみならず一般ユーザにとっても関心度が高い。例えば，いつ頃が空いているか，自分のホームページにどこからどの程度のアクセスがあるか等を知ることができる。さらにプロキシサーバのログ解析を行なって全体像を把握することも可能である。

4 結　言

今後の増大するネットワークに備えてシステム運営を効率化することは重要な課題である。インターネット技術の進歩とともに異なるOS，異なる機種のマシン同士が接続し，同じユーザインタフェースで操作できるシステムを構築するための環境が整ってきた。このような技術を基礎に置くことで，多数のサーバを集中的に管理するために必ずしも大規模なプログラム開発は必要でなくなってきた。一度開発したシステムはそれ自体の維持管理が必要となる。システム管理を効率化するためのシステムの維持管理に多くの労力を割くことは避けねばならない。

システムの維持管理の観点から，本論文のシステムの新規開発部分は最小限に抑えた。コンパイラ系言語のCで開発したのはSA Managerのコマンド実行部分のみで，複雑な機能を必要とするシステムの設定関連の処理には既存のツールであるcfengineを利用できるようにした。ユーザインタフェース部分はWWWの機能であるフォームとCGIを用い，CGIのプログラミングには開発効率の良いPerlを用いた。

まだ，稼働して間もないのでほとんど効果は報告できないが，ascshによる非同期の実行は，対象とするサーバ数の増大とともに並列処理の効果が出てscshの数倍の性能がでることを確認している。

今後の課題を以下に示す：

1. 非同期実行機能の充実
   非同期実行ではマネジャーとエージェントの間に同時に1対多の関係が生まれ，しかも処理が終了するまでの待ち時間に制限がない。HTMLとCGIを使ったプログラミングだけではこのような状況に十分対応できるインタフェースを作るのが難しく，今後の検討が必要である。
2. コマンド実行の制限
   SA Managerのコマンド実行に対する制限としてパスワード認証を採用したが，WWWサーバにログインできるユーザであればパスワード認証を経ないで直接コマンドを実行できるという問題

があり，今後の検討が必要である。

最後に，研究の機会を与えて下さった高岡短期大学の宮本匡章学長及び数々の貴重なアドバイスを与えていただいた高岡短期大学産業情報学科の佐藤孝紀教授に厚く謝意を表する。

引用文献

1）近藤，藤田，米川：学内LANにおけるシステム運営の効率化とユーザサービス向上の試み，高岡短期大学紀要第7巻，pp.29-38，1996年3月
2）Magnus Harlander：Central System Administration in a Heterogeneous Unix Environment：GeNUAdmin, Eighth System Administration Conference(LISA'94)，pp.1-8，1994
3）Mark Burgess：A Site Configuration Engine, Computing Systems (The Journal of the USENIX Association)，Vol.8，No.3，pp.309-337，1995(http://www.iu.hioslo.no/mark/cfengine.html)
4）所　真理雄：マルチエージェントシステム研究の目指すもの，コンピュータソフトウェア，Vol12，No.1，pp.78-84，1995
5）(社)電子情報通信学会：情報ネットワークハンドブック，オーム社
6）John Bloomer：Power Programming with RPC, O'Reilly & Associates, Inc
7）C. Liu et al.：Managing Internet Information Services, O'Reilly & Associates, Inc(http://hoohoo.ncsa.uiuc.edu/)
8）S. Garfinkel他「UNIXセキュリティ」アスキー出版局
9）C. Musciano et al.：HTML The Definitive Guide, O'Reilly & Associates, Inc
10）S. Gundavaram：CGI Programming on the World Wide Web, O'Reilly & Associates, Inc
11）L. Wall他：Perlプログラミング，ソフトバンク
12）(http://www.sun.co.jp/java.jp/docs/)

Central Administration of Servers in Distributed Network Syestms

—Implementing a WWW Host—

Kiyoshi KONDO, Satoru YONEKAWA

(Received November 5, 1996)

ABSTRACT

As distributed network systems spread and grow, the number of server computers grows, increasing the amount of system management work. The authors have developed a tool which manages remote servers from a central WWW host to control many servers efficiently., The server management system of Takaoka National University uses the tool; It consists of a manager and agents. The manager sends requests for command execution in either synchronous or asynchronous mode to agents which reside in remote hosts. The alias function enables the manager to use common commands transparently among different systems. GNU cfengine can be driven by the manager to execute complex configuration in remote hosts. As a side effect of using a WWW server, it is easy to open the results of sytem management to users. This will improve the quality of user service as well as reduce inquiries from users by giving more information to them.

KEY WORDS

System Administration, Server Management, Central Administration, SA Manager, RPC, Asynchronous Execution, GNU cfengine, WWW