ネットワークのトラフィックのトラブル発見及び警告通知

03H007 井上 忠昭

目次

1 はじめに 2

2 システム 3

2.1 システムの概要 3

2.2 説明 4

3 作成したシステム 5

4 実行結果 6

5 考察・まとめ 8

付録 A 付録 1 9

A.1 抽出プログラム 9

A.2 時間変換プログラム 9

A.3 メールプログラム 9

A.4 曜日判定プログラム 10

1　はじめに

近年、ネットワーク網が広がり、ネットワーク通信を気軽に行えるようになっている。そして、学校や会社などの多くのユーザーが利用するネットワークシステムでは、規模によって要求されるシステムの質に違いはあるが、24 時間 365 日ノンストップで稼働することが理想として求めらている。

しかし、絶対に止まらないシステム作成することは不可能である。だが、ノンストップに近い状態で稼働させる事が管理者の仕事である。ノンストップに近い状態で稼働させるには、管理者がシステムを監視し、トラブルが発生した時には、問題の解決しなければならない。特にシステムが大きいほど、監視する対象も多くなり、管理者の仕事も多くなる。また、理想を言うならば、ユーザーにトラブル発生を悟られる前に、問題解決をすることが管理者には求められる。

そこで、システム管理を手助けしてくれるのがシステム監視ツールである。システム監視ツールも色々なものがあり、例を上げると、BigBrother や MRTG や NetSaint などがある。

本研究室では MRTG を利用している。MRTG は SNMP を使っており、SNMP でルータやコンピュータなど、通信機器をネットワーク経由で監視し、出た結果を InPut/OutPut パケットでグラフ化でき、Web データとして作成できる。そして、本研究室では誰がいつ、研究室内の PC を利用したかを Web ページで存在確率として公表している。

しかし、MRTG でグラフ化できるだけでは、管理者がグラフを見続けていないといけない。また、トラブルが発生したときに、グラフを見続けていないとグラフの異常な変化に気がつかない。それでは管理者の負担が大きいので、グラフを見続けていなくても、トラブルの発見ができないかと考えた。

そこで、グラフの異常な変化が起きたとき、警告をメールで送信することで、グラフを見なくても発見でき、グラフを監視することなく、管理者が異変に気がつき、管理者の負担を減らせると考えた。

2　システム

2.1　システムの概要

普段の研究室におけるルータを流れるトラフィックのデータの統計をとり、その結果を基本として、ルータのトラフィックデータを比較する。そのなかで、比較の差が大きすぎたり、急激なトラフィックの増大などは障害が発生する前兆の可能性があるので、警告を表示するシステムを作成しようと考えた。もちろん、平日と土曜、日曜、祝日によって利用率が違うので平日と休みの日のそれぞれの統計をとる。祝日で日が決まってる日は日にちで設定し、成人の日など、曜日で決まってる祝日 (ハッピーマンデー)[*1]は、○ 月第○ 月曜日で設定する。夏休みや、法改正で祝日が変わった時、プログラムがたいへんなので、変更が簡単なテーブルを利用する。テーブルはテキストファイルとし、プログラムで読み込みます。以下は簡単な概要図 (図 1) を貼付する。

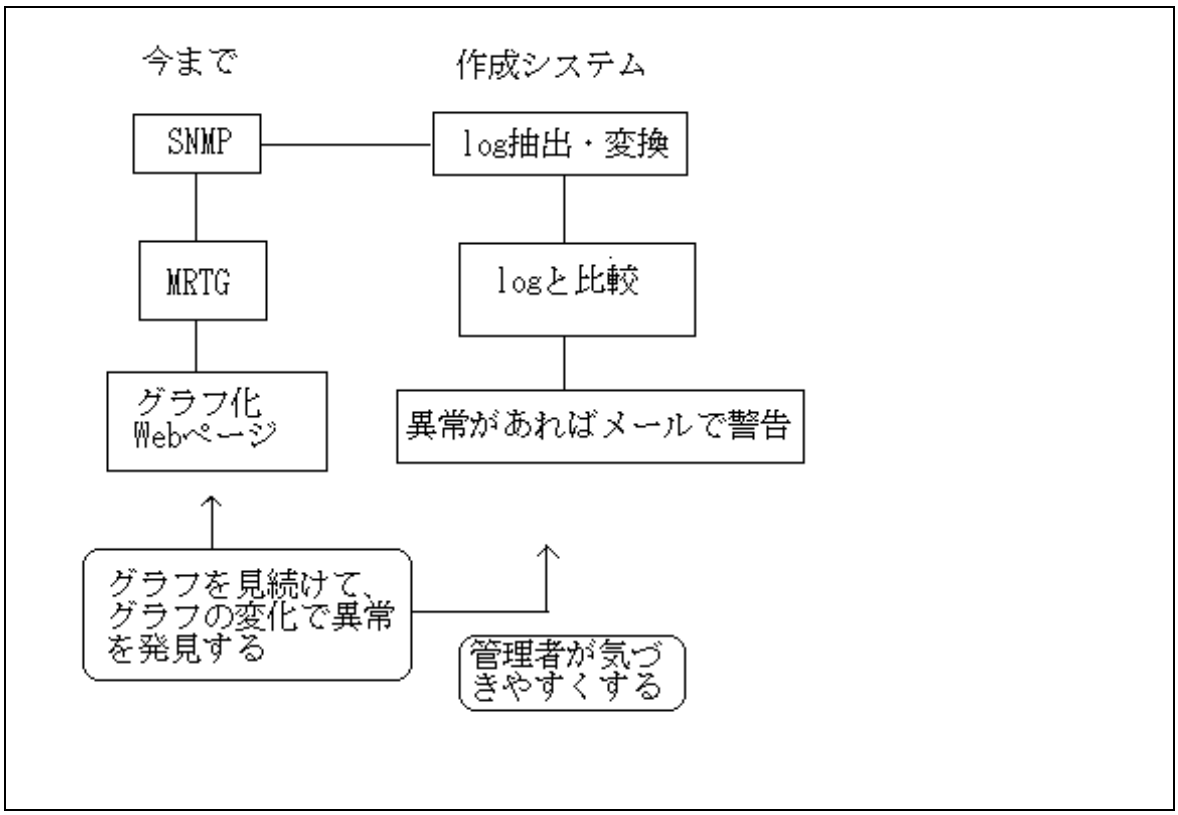

図 1　概要図

[*1] ハッピーマンデー制度とは、国民の祝日の一部を従来の日から月曜日に移動させる制度を指す。

2.2　説明

1. プログラム
   → 本研究のプログラムは Perl で作成する。
   なぜ Perl 使用してるかというと、MRTG が Perl でかかれているため、Perl で作成するのがよいと考えたからである。
2. MRTG
   → SNMP でネットワークを監視した結果をもとに、ネットワーク機器の利用状況を、InPut/OutPut パケットでグラフ化ができ、グラフィックイメージを含む HTML ページを生成してくれるツールである。
   またマシンに関するものから関係ない室温などまで、グラフ化することも可能である。MRTG もグラフ例以下にを貼付する。(図 2)

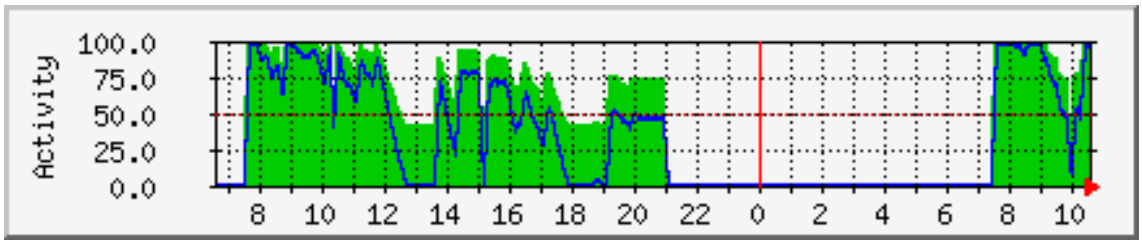

図 2　MRTG のグラフ例

3. SNMP
   → TCP/IP ネットワークにおいて、ルータやコンピュータ、端末など、ネットワークに接続された通信機器をネットワーク経由で監視・制御するためのプロトコルである。
4. corn →指定時間にコマンドやプログラムを実行する機能である。

3　作成したシステム

- MRTG でグラフ化される前のトラフィックデータの検索
- トラフィックデータを抽出する
- 抽出したものを UNIXTime と数値 log にわける
- UNIXtime を JST に変換する
- JST に変換したものと数値 log を表示する
- 変換したデータを基に基データ (平日と土曜と日曜、祝日にわけて) 作成する
- メール送信するプログラムを作成する
- データを基に比較し、比較結果によってはメールをするプログラムを作成する
- 上記のプログラムにテーブルを利用して曜日の設定をする
- 平日と土曜と日曜、祝日で別けて比較するようにする
- ハッピーマンデーと振替休日も祝日に設定する
- 春分・秋分の日の計算を毎年自動でおこない、祝日判定を行なうように設定する
- cron を使い常に監視できるよにした

4　実行結果

なんとか Log ファイルをみつけることができた。SNMP の Log 発見に時間がかかりすぎたために、現状の結果となった。log 抽出プログラム (図 3) と log 変換、プログラムメール送信のプログラム (図 5) も作成できた。

日で決まっている祝日と日曜、土曜、平日でわけて比較し、結果によって違う結果を表記にできるようにできた。また振替休日やハッピーマンデーによる月曜日の休みの設定を行えた。前ページを参照すると 1 ~ 13 まで作成できた。

```
[tadaaki@lindow04 perl]$ cat rwho.yada.log |more
1167097036 99 94
1167097036 99 94 99 94
1167096716 99 98 99 98
1167096600 99 98 99 99
1167096300 99 99 99 99
1167096000 99 99 99 99
1167095700 99 99 99 99
1167095400 99 98 99 99
1167095100 99 98 99 98
1167094800 99 97 99 98
1167094500 99 97 99 97
1167094200 99 97 99 99
1167093900 99 99 99 99
1167093600 99 99 99 99
1167093300 99 99 99 99
1167093000 99 99 99 99
1167092700 99 99 99 99
1167092400 99 99 99 99
1167092100 99 99 99 99
1167091800 99 99 99 99
1167091500 99 99 99 99
1167091200 99 99 99 99
--続ける--
```

図 3　log 抽出結果

```
[tadaaki@lindow01 perl]$ perl tyusyutu.pl |more
 2007/01/16 10:15:16 <2> 99 97    2007/01/16 10:15:16 <2> 99 97  99 97
 2007/01/16 10:05:36 <2> 99 92  99 92
 2007/01/16 10:05:00 <2> 99 92  99 92
 2007/01/16 10:00:00 <2> 98 91  99 92
 2007/01/16 09:55:00 <2> 0 0  0 0
 2007/01/16 09:50:00 <2> 0 0  0 0
 2007/01/16 09:45:00 <2> 0 0  0 0
--続ける--
```

図 4　log 変換結果

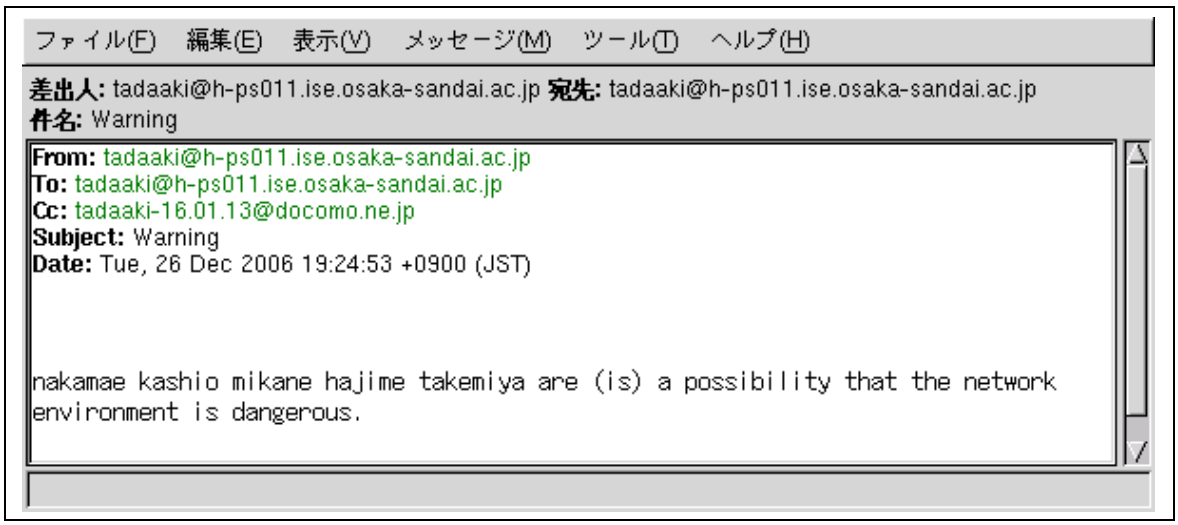
ファイル(F) 編集(E) 表示(V) メッセージ(M) ツール(T) ヘルプ(H)
差出人: tadaaki@h-ps011.ise.osaka-sandai.ac.jp 宛先: tadaaki@h-ps011.ise.osaka-sandai.ac.jp
件名: Warning
From: tadaaki@h-ps011.ise.osaka-sandai.ac.jp
To: tadaaki@h-ps011.ise.osaka-sandai.ac.jp
Cc: tadaaki-16.01.13@docomo.ne.jp
Subject: Warning
Date: Tue, 26 Dec 2006 19:24:53 +0900 (JST)
nakamae kashio mikane hajime takemiya are (is) a possibility that the network environment is dangerous.

図 5　受信メール

5　考察・まとめ

当初、数値データのみで判定し、メール送信をしようと考えていたが、数値データのみだと、トラフィックが 0 の時など、警告メールが大量に送信されてしまう。例として、ログインしていない時や、コマンド入力をしていないときなどが上げられる。なので、他のデータなどを参照し、システムを考える必要がある。

処理に時間がかかっている。そして、管理者がメール受信するまでに時間がかかる。なので、プログラムについて勉強し、現在のプログラムより早く処理できるようにする必要がある。また、負荷も大きいので、もっと軽くする。

謝辞

本研究を進める上で、御指導頂及び御協力頂いた大垣斉講師・藤井 信夫教授、採点をしてくださった松本助教授、ならびに fken.a4w ML メンバーの方々、卒研生、居候の方々に深く感謝いたします。

参考文献

[1] 三省堂辞書 [http://dictionary.sanseido-publ.co.jp/topic/10minnw/011ubiquitous.html]
[2] Linux 管理者への道 [http://www.atmarkit.co.jp/flinux/rensai/root08/root08a.html]
[3] IT 用語辞典, [http://e-words.jp/]
[4] MRTG[http://www.mrtg.jp/doc/]
[5] リソース監視 [http://www.jp-z.jp/linux/mrtg.html]
[6] 監視を自動化する SNMP[http://www.atmarkit.co.jp/fnetwork/rensai/snmp01/01.html]
[7] すぐわかる Perl(ISBN-7741-0817-0 C3055)
[8] はてなダイアリー [http://d.hatena.ne.jp/keyword/]
※ URL はすべて 2006 年 12 月 8 日 (金) 時点のもの

付録 A　付録 1

A.1　抽出プログラム

!/usr/bin/perl
use Net::FTP;
@fken = (tadaaki,nakamae,yada,oguri,ohnishi,kashio,fujikawa,uetsuji,mikane,hajime,takemiya);
my $host =' h-ps011.fken'$;
my $user =' tadaaki'$;
my $pass =' ********'$; ←パスワードは * で表している
while(@fken[i])
my $remote_file =$ "/tmp/warbird.fken/SNMP/rwho.@fken[i].log";
my $local_file =$ "/home/2006/tadaaki/perl/rwho.@fken[i].log";
my $ftp = Net :: FTP- > new$(host) or die "can not connection: @";
$ftp- > login$(user, $pass)ordie$ftp-¿message;
$ftp- > binary$;
$ftp- > get$(remote$_{file}$,local$_{file}$)orwarn$ftp-¿message;
$ftp- > quit$;
$i++$;

A.2　時間変換プログラム

!/usr/bin/perl
@data =();
$file =$ "/home/2006/tadaaki/perl/rwho.tadaaki.log";
open(FH,$file$);
$file =< FH >$;
@data = split(/+/,$file$, 5);
print(" ",@data[1]," ",@data[2]," ",@data[3]," ",@data[4],"");
close(FH);

A.3　メールプログラム

!/usr/bin/perl
@data =();
@fken = (tadaaki,nakamae,yada,oguri,ohnishi,kashio,fujikawa,uetsuji,mikane,hajime,takemiya);
AA:
while(@fken[i])
$file =$ "/home/2006/tadaaki/perl/rwho.@fken[i].log";
open(FH,$file$);
@file = ¡FH¿;
$file2 =$file[2];
@data = split(/+/,$file2$, 5);

if(”@data[1]” ¿= ”40” ”@data[1]” ¡ ”100” ”@data[2]” ¿= ”40” ”@data[2]” ¡ ”100” ”@data[3]” ¿= ”40” ”@data[3]” ¡ ”100” ”@data[4]” ¿= ”40” ”@data[4]” ¡ ”100”)

print” @fken[i]$isNormality$,”;

$i++$;

goto AA;

else

@aa[j] = @$fken$[i];

$j++$;

close(FH);

$i++$;

if(@aa == 0)

exit;

else

$sendmail = '$ /$usr/sbin/sendmail'$;

$from = '$ $tadaaki@h-ps011.ise.osaka-sandai.ac.jp'$;

$to = '$ $tadaaki@h-ps011.ise.osaka-sandai.ac.jp'$;

$cc = '$ $tadaaki-16.01.13@docomo.ne.jp'$;

$subject = '$ $Warning'$;

msg =<< ”$_TEXT$”; メールの本文 (ヒアドキュメントで変数に代入)

@aa are (is) a possibility that the network environment is dangerous. $_TEXT$

open(SDML,”— $sendmail-t-i$”)||$die'sendmailerror'$;

print SDML ”From: $from$”;

print SDML ”To: to”;

print SDML ”Cc: cc”;

print SDML ”Subject: $subject$”;

print SDML ”Content-Transfer-Encoding: 7bit”;

print SDML ”Content-Type: text/plain;”;

print SDML ”msg”;

close(SDML);

exit;

A.4 曜日判定プログラム

!/usr/bin/perl

use POSIX ’strftime’;

$jikan = strftime$”

$jikan2 = strftime$”

$youbi = strftime$”

$manth =$ ”

$day =$ ”

$day2 =$ ”

$year =$ ”

if(($manth = 1)||(manth == 10))

$holiday = '/home/2006/tadaaki/perl/holiday2.txt'$; 第 2 ハッピーマンデー
open (FILE, $holiday$);
while($holiday =< FILE >$)
if($holiday = m/$day/)
$youbi = 0$;
close (FILE)
elsif(($manth == 7$)||(manth == 9))
$holiday = '/home/2006/tadaaki/perl/holiday3.txt'$; 第 3 ハッピーマンデー
open (FILE, $holiday$);
while($holiday =< FILE >$)
if($holiday = m/$day/) $youbi = 0$;
close (FILE)
else
$holiday = '/home/2006/tadaaki/perl/holiday1.txt'$; 普通の祝日
open (FILE, $holiday$);
while($holiday =< FILE >$)
if($holiday = m/$jikan2/) $youbi = 0$;
close (FILE);
$holiday = '/home/2006/tadaaki/perl/holiday4.txt'$; 振替
open (FILE, $holiday$);
while($holiday =< FILE >$)
if($holiday = m/$day2/) $youbi = 0; close(FILE)$;
$holiday = '/home/2006/tadaaki/perl/holiday5.txt'$; 春分・秋分
open (FILE, $holiday$);
while($holiday =< FILE >$)
if($holiday = m/$jikan/) $youbi = 0; close(FILE)$;
if($youbi == 0$)
require'/home/2006/tadaaki/perl/mail3.pl'; 日曜・祝日
elsif($youbi == 6$)
require'/home/2006/tadaaki/perl/mail2.pl'; 土曜
else
require'/home/2006/tadaaki/perl/mail.pl'; 平日