Postfixのログ解析

SendmailやPostfixはメールの送受信にキューを使用しており、メールのログもキューにメールを追加して配送完了や破棄までイベントごとのログが出力される。そのためメールが無事に配送されたかどうかは、キューIDを頼りに解析していかないとわからないことになる。

mail.log

```
Jun 26 04:19:51 raspberrypi postfix/smtpd[8834]: connect from
localhost[::1]
Jun 26 04:19:51 raspberrypi postfix/smtpd[8834]: 1CD3A5E5F7:
client=localhost[::1]
Jun 26 04:19:51 raspberrypi postfix/cleanup[8837]: 1CD3A5E5F7: message-
id=<xSQ83dtbxSuOCbmh3AFIodVVyVoBKrGcDg2IodzOTqI@ultraviolet.zapto.org>
Jun 26 04:19:51 raspberrypi postfix/smtpd[8834]: disconnect from
localhost[::1] ehlo=1 mail=1 rcpt=1 data=1 quit=1 commands=5
Jun 26 04:19:51 raspberrypi postfix/qmgr[1017]: 1CD3A5E5F7:
from=<wordpress@ultraviolet.zapto.org>, size=2978, nrcpt=1 (queue
active)
Jun 26 04:19:53 raspberrypi postfix/smtp[8838]: 1CD3A5E5F7:
to=<kijima.minoru@gmail.com>,
relay=smtp.gmail.com[2404:6800:4008:c02::6c]:587, delay=2.6,
delays=0.12/0.17/1.1/1.2, dsn=2.0.0, status=sent (250 2.0.0 OK
1687720793 r2-20020a62e402000000b00672401787c6sm1251779pfh.109 - gsmtp)
Jun 26 04:19:53 raspberrypi postfix/qmgr[1017]: 1CD3A5E5F7: removed
```

そこでPostfixのメールログを**from**,**to**といった情報や処理結果(**status**)を1行に集約することにする。

日時、ホスト名、キューIDの分離

まずは日時とホスト名、キューIDの取得とPostfixログを分離する処理を書いてみる。 難しいことはなくスペースをセパレーターとしたカラムに分離するだけである。キューIDより後ろはスペース区切りの分離ではむしろ扱いにくくなるため分離個数を設定しておく。
メールの情報はカンマ(,)をセパレーターとして分離する。

postfix-log-analyzer.pl

```
#!/usr/bin/perl
# postfix log analyzer

%month_num = (
    'JAN', 1,
    'FEB', 2,
    'MAR', 3,
    'APL', 4,
    'MAY', 5,
```

```
    'JUN', 6,
    'JUL', 7,
    'AUG', 8,
    'SEP', 9,
    'OCT', 10,
    'NOV', 11,
    'DEC', 12 );

while(<>) {
    chop;
    s/\r$//;
    $syslog = $_;
    @col = split(/\s+/, $syslog, 7);
    unless ($col[4] =~ /^postfix/) {next}
    $mon = $month_num{uc($col[0])};
    ($day, $time, $host, $qid) = @col[1,2,3,5];
    $qid =~ s/:$//;
    printf("%02d/%02d %s %s %s\n", $mon, $day, $time, $host, $qid);
    @stack = split(/\s*,\s*/, @col[6]);
    while (@stack) {
        $word = shift(@stack);
        printf(": %s\n", $word);
    }
}
```

処理結果

このように分離できればメール情報の解析は簡単になる。取得対象のワードが出現したら、以降の部分を変数に格納すればよい。

```
06/26 04:19:51 raspberrypi connect
: from localhost[::1]
06/26 04:19:51 raspberrypi 1CD3A5E5F7
: client=localhost[::1]
06/26 04:19:51 raspberrypi 1CD3A5E5F7
: message-
id=<xSQ83dtbxSuOCbmh3AFIodVVyVoBKrGcDg2IodzOTqI@ultraviolet.zapto.org>
06/26 04:19:51 raspberrypi disconnect
: from localhost[::1] ehlo=1 mail=1 rcpt=1 data=1 quit=1 commands=5
06/26 04:19:51 raspberrypi 1CD3A5E5F7
: from=<wordpress@ultraviolet.zapto.org>
: size=2978
: nrcpt=1 (queue active)
06/26 04:19:53 raspberrypi 1CD3A5E5F7
: to=<kijima.minoru@gmail.com>
: relay=smtp.gmail.com[2404:6800:4008:c02::6c]:587
: delay=2.6
```

```
: delays=0.12/0.17/1.1/1.2
: dsn=2.0.0
: status=sent (250 2.0.0 OK  1687720793
r2-20020a62e402000000b00672401787c6sm1251779pfh.109 - gsmtp)
06/26 04:19:53 raspberrypi 1CD3A5E5F7
: removed
```

取得対象のワード

下記のワードを変数に格納しておき、配送や破棄の処理が確定した時点でログを出力する。

- client=
- message-id=
- from=
- size=
- to=
- relay=
- status=

解析スクリプト

変数への格納とログ出力、それからログ出力が済んだ変数を削除を追加するとこうなる。
オリジナルのsendmail対応版を20年近く前に書いたのだが処理はほぼ同じである。

postfix-log-analyzer.pl

```
#!/usr/bin/perl
# postfix log analyzer

%month_num = (
    'JAN', 1,
    'FEB', 2,
    'MAR', 3,
    'APL', 4,
    'MAY', 5,
    'JUN', 6,
    'JUL', 7,
    'AUG', 8,
    'SEP', 9,
    'OCT', 10,
    'NOV', 11,
    'DEC', 12 );

while(<>) {
    chop;
    s/\r$//;
    $syslog = $_;
    @col = split(/\s+/, $syslog, 7);
    unless ($col[4] =~ /^postfix/) {next}
```

```
    $mon = $month_num{uc($col[0])};
    ($day, $time, $host, $qid) = @col[1,2,3,5];
    $qid =~ s/:$//;
    @stack = split(/\s*,\s*/, @col[6]);
    while (@stack) {
        $word = shift(@stack);
        if ($word =~ s/^client=//) {
            $log_client{$qid} = $word;
        } elsif ($word =~ s/^message-id=//) {
            $log_messageId{$qid} = $word;
        } elsif ($word =~ s/^from=//) {
            $log_from{$qid} = $word;
        } elsif ($word =~ s/^size=//) {
            $log_size{$qid} = $word;
        } elsif ($word =~ s/^to=//) {
            $log_to{$qid} = $word;
        } elsif ($word =~ s/^relay=//) {
            $log_relay{$qid} = $word;
        } elsif ($word =~ s/^status=//) {
            $log_status{$qid} = $word;
        } elsif ($word =~ /^removed$/) {
            printf("%02d/%02d %s %s %s client=%s, from=%s, to=%s,
size=%s, message-id=%s, relay=%s, status=%s\n", $mon, $day, $time,
$host, $qid, $log_client{$qid}, $log_from{$qid}, $log_to{$qid},
$log_size{$qid}, $log_messageId{$qid}, $log_relay{$qid},
$log_status{$qid});
            delete $log_client{$qid};
            delete $log_from{$qid};
            delete $log_to{$qid};
            delete $log_size{$qid};
            delete $log_messageId{$qid};
            delete $log_relay{$qid};
            delete $log_status{$qid};
        }
    }
}
```

ログ解析結果

ログの解析結果**はこうなる**。

```
06/26 04:19:53 raspberrypi 1CD3A5E5F7 client=localhost[::1],
from=<wordpress@ultraviolet.zapto.org>, to=<kijima.minoru@gmail.com>,
size=2978, message-
id=<xSQ83dtbxSuOCbmh3AFIodVVyVoBKrGcDg2IodzOTqI@ultraviolet.zapto.org>,
relay=smtp.gmail.com[2404:6800:4008:c02::6c]:587, status=sent (250 2.0.0 OK
1687720793 r2-20020a62e402000000b00672401787c6sm1251779pfh.109 - gsmtp)
```

```
06/28 00:01:02 raspberrypi 48E515E675 client=,
from=<pi@mail.kijima.mydns.jp>, to=<pi@mail.kijima.mydns.jp>, size=891,
message-id=<20230627150102.48E515E675@mail.kijima.mydns.jp>, relay=local,
status=sent (delivered to mailbox)
06/29 20:28:46 raspberrypi ED3765E688 client=localhost[::1],
from=<wordpress@ultraviolet.zapto.org>, to=<kijima.minoru@gmail.com>,
size=25390, message-
id=<zxlWHZovSm73XRYuZ4DPhkIDdkoTIydCmvdlhZ9U8@ultraviolet.zapto.org>,
relay=smtp.gmail.com[64.233.188.108]:587, status=sent (250 2.0.0 OK
1688038126 u12-20020a170902e5cc00b001aae625e422sm7349239plf.37 - gsmtp)
06/29 20:29:04 raspberrypi E4A4D5E688 client=localhost[::1],
from=<wordpress@ultraviolet.zapto.org>, to=<kijima.minoru@gmail.com>,
size=1153, message-
id=<RHdqGoxIeA4gsLFDBZ8txaKrYj3Df8qbceOHZL6Yq4@ultraviolet.zapto.org>,
relay=smtp.gmail.com[2404:6800:4008:c19::6d]:587, status=sent (250 2.0.0 OK
1688038144 oj3-20020a17090b4d8300b0024e4f169931sm10906240pjb.2 - gsmtp)
06/29 20:48:18 raspberrypi F38C95E688 client=localhost[::1],
from=<wordpress@ultraviolet.zapto.org>, to=<kijima.minoru@gmail.com>,
size=1037, message-
id=<v1xInvbMwMx9Esb4D2mh0g5dFuup4AjS6PgsV19j0@ultraviolet.zapto.org>,
relay=smtp.gmail.com[173.194.174.109]:587, status=sent (250 2.0.0 OK
1688039298 d14-20020a170902cece00b001b864add154sm262846plg.154 - gsmtp)
06/30 17:19:57 raspberrypi 66CA45E612 client=localhost[::1],
from=<wordpress@ultraviolet.zapto.org>, to=<kijima.minoru@gmail.com>,
size=1039, message-
id=<nGwbPgeH3Hm5kvL79Y1Zijjbrew0F1HaKsnZMg3dqT4@ultraviolet.zapto.org>,
relay=smtp.gmail.com[142.251.170.109]:587, status=sent (250 2.0.0 OK
1688113197 r2-20020a62e402000000b00672401787c6sm8267636pfh.109 - gsmtp)
06/30 17:32:19 raspberrypi 20D705E612 client=localhost[::1],
from=<wordpress@ultraviolet.zapto.org>, to=<kijima.minoru@gmail.com>,
size=2977, message-
id=<lS8rWnRbSC5rn6Qyq931IWtx8RlIwMNS46Pao14fsA@ultraviolet.zapto.org>,
relay=smtp.gmail.com[142.251.8.109]:587, status=sent (250 2.0.0 OK
1688113938 14-20020a630b0e000000b0054fb537ca5dsm9810814pgl.92 - gsmtp)
```

似たような解析事例もあったのだが、わざわざリングバッファを構成していて情報が欠けることもあるらしい。そんな面倒なことをしなくともキュ□IDが存在する間だけ変数が存在すればよいので、意外とメモリ消費は増えないものである。とある万以上のユーザーが使うメールログ解析でも十分使えたので大丈夫であろう。