

CONFIGURAÇÕES DE SERVIDORES DE CORREIO ELECTRÓNICO

Pedro Vale Pinheiro¹, Mário Bernardes²
Centro de Informática da Universidade de Coimbra
3000 COIMBRA
vapi@student.dei.uc.pt, mjb@ci.uc.pt

Sumário

Este documento é o resultado de um estudo de configurações de servidores de correio electrónico com vista a maximizar a escalabilidade e disponibilidade do serviço, com base em plataforma Unix utilizando a distribuição de *sendmail* sem qualquer desenvolvimento adicional ou produtos proprietários.

Introdução

A importância do serviço de correio electrónico na Internet atingiu um nível tal que se tornou numa preocupação permanente para a grande maioria dos administradores de sistemas envolvidos na gestão deste serviço. Paralelamente ao crescimento exponencial do número de utilizadores, revelaram-se requisitos – disponibilidade, escalabilidade – para os quais a plataforma tradicional de suporte ao serviço de correio electrónico Unix/*sendmail* não foi concebida. A esta necessidade respondeu o mercado com diversos produtos proprietários que muitas vezes não estão ao alcance de instituições do *campus* de uma universidade, de um hospital ou de outras com características idênticas. Neste âmbito, propusémo-nos analisar configurações de servidores de correio electrónico baseadas exclusivamente em plataforma Unix, utilizando o *sendmail*[1][2] como *mail transfer agent* (MTA) e com recurso a ficheiros de configuração simples, sem utilizar qualquer tipo de programação adicional.

Assim, começaremos por uma configuração convencional baseada num único servidor sendo apresentados, de seguida, dois cenários evolutivos onde procuramos identificar, em cada um, as facilidades e limitações em termos de disponibilidade e escalabilidade. Os protótipos foram construídos com base no *Sendmail* 8.9.0, tendo os ficheiros de configuração – *sendmail.cf* – sido gerados a partir de ficheiros de macros *m4*[3]. O domínio que compreende o serviço de correio é designado por *inst.uc.pt*. Os programas clientes do serviço de correio (MUA, *mail user agent*) são externos aos servidores e a sua parametrização deverá permanecer inalterada apesar das alterações na configuração da plataforma do servidor. Os clientes do serviço de correio deverão apenas conhecer dois parâmetros:

```
POP Account: user@pop.inst.uc.pt
SMTP: smtp.inst.uc.pt
```

1º CASO

Num cenário convencional, o serviço de correio é suportado por um único servidor, designado na **Figura 1**

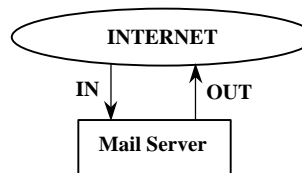


Figura 1

por *Mail Server*, que recebe as mensagens do exterior, envia as mensagens, aloja as caixas de correio dos utilizadores e ainda o serviço de POP[4].

Para suportar a configuração de clientes proposta, este sistema poderá ser registado no servidor de DNS da instituição

```
mail-server IN A 192.168.0.1
pop IN CNAME mail-server.inst.uc.pt.
smtp IN CNAME mail-server.inst.uc.pt.
```

O *mail-server* tem que aceitar as mensagens para o domínio *inst.uc.pt*, pelo que se cria o ficheiro */etc/sendmail.cw* contendo a linha

```
inst.uc.pt
```

Pretende-se proteger o servidor contra SPAM[5] e garantir que as mensagens que deixam o servidor tenham a linha “From:” do cabeçalho com *inst.uc.pt*. Para tal foi construído um ficheiro *.mc* com macros *m4* do tipo:

```
FEATURE(masquerade_envelope)
FEATURE(masquerade_entire_domain)
FEATURE(access_db)

MASQUERADE_AS(inst.uc.pt)
MASQUERADE_DOMAIN_FILE('/etc/mail/md')
```

A macro *masquerade_envelope* vai mascarar todo o cabeçalho de uma mensagem. Sem esta opção só seria mascarado a linha “From:”. A opção *masquerade_entire_domain* vai mascarar como *inst.uc.pt* todas as máquinas que usam o servidor para envio de correio e que pertençam aos domínios incluídos no ficheiro */etc/mail/md*. Este ficheiro terá que conter, pelo menos, a linha

¹ Estudante do Departamento de Eng. Informática da U. Coimbra.

² Assessor Principal do Centro de Informática da U. Coimbra

```
inst.uc.pt
```

O `FEATURE(access_db)` serve para implementar as medidas *anti-spam*[6]. No ficheiro `/etc/mail/access.txt` deverão estar os endereços das máquinas que poderão fazer relay através da máquina `mail-server.inst.uc.pt`. Por exemplo,

```
192.168.0 RELAY3
```

Neste caso todas as máquinas da rede 192.168.0, poderão utilizar o servidor como *relay*.

Para evitar a perda de mensagens por falha do servidor `mail-server.inst.uc.pt`, são adicionados os seguintes registos MX ao ficheiro de configuração de DNS do domínio

```
IN MX 10 mail-server.inst.uc.pt.  
IN MX 20 backup.inst.uc.pt.
```

Assim, enquanto o servidor não estiver disponível, as mensagens são encaminhadas para `backup.inst.uc.pt`, o que é garantido atribuindo o peso 20 ao *backup*. Se neste servidor o `sendmail` for executado com

```
sendmail -bd -qlh
```

ao fim de cada hora é tentado o envio das mensagens que permanecem na fila. Deste modo, quando o `mail-server` estiver de novo acessível, irá receber as mensagens armazenadas no servidor `backup.inst.uc.pt`, pois possui registos MX de menor peso.

Esta configuração apenas suporta escalabilidade vertical, isto é, o desempenho é melhorado por adição de componentes da própria máquina – processador, memória, discos.

2º CASO

Esta configuração difere da anterior, na medida em que o envio de mensagens é garantido não pelo servidor de correio, mas por um conjunto de uma ou mais máquinas (`smtput1`, `smtput2`, ...) designado na **Figura 2** por *Mail Relay*. O `mail-server` continua a receber as mensagens do exterior, a armazenar as caixas de correio dos utilizadores e com o servidor de POP. O *Mail Relay* é ainda utilizado para receber as mensagens em caso de falha do servidor de correio, comportando-se como o servidor `backup.inst.uc.pt` do caso anterior. Esta configuração é bastante escalável relativamente ao envio de mensagens, podendo utilizar-se uma distribuição de carga (de envio) por *round-robin*[7] entre os diversos `smtput`. Estas funcionalidades são garantidas com a seguinte configuração de DNS:

```
IN MX 10 mail-server.inst.uc.pt.  
IN MX 20 smtput1.inst.uc.pt.  
IN MX 20 smtput2.inst.uc.pt.  
...  
  
smtp IN CNAME smtput1.inst.uc.pt.  
      IN CNAME smtput2.inst.uc.pt.  
      ...
```

³ Entre o endereço IP e a palavra RELAY, tem que existir pelo menos um tabulador.

Em caso de falha de um dos `smtput`, há apenas um atraso no envio de algumas mensagens. Ao utilizar-se *round-robin*, é atribuído um mesmo nome (neste caso `smtput.inst.uc.pt`) a um conjunto de máquinas (`smtput1.inst.uc.pt`, `smtput2.inst.uc.pt`); o nome `smtput.inst.uc.pt` é resolvido pelo DNS de forma circular, isto é, quando cliente pergunta o endereço de `smtput.inst.uc.pt`, o DNS responde da primeira vez com `smtput1`, da segunda com `smtput2`, etc. Se falhar o `smtput1`, o DNS, no primeiro pedido, resolve o nome `smtput.inst.uc.pt` com uma máquina não disponível. No entanto, num segundo pedido, o resultado já é `smtput2`. Deste modo, as mensagens dos MUA poderão permanecer mais tempo nas respectivas filas em função da resposta do DNS.

Se fosse admissível que os utilizadores utilizassem MUAs no próprio servidor de correio `mail-server.inst.uc.pt`, este deveria ser configurado por forma a que todas as mensagens fossem enviadas para o

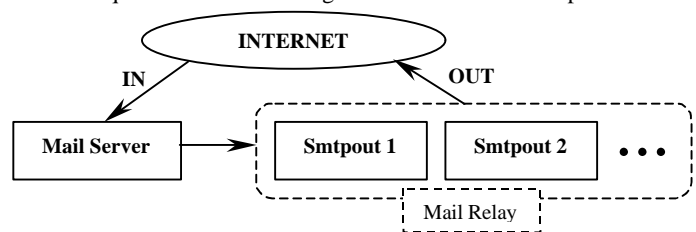


Figura 2

`smtput.inst.uc.pt`. Para isso, bastaria incluir no ficheiro `.mc`:

```
define(`SMART_HOST', `smtput.inst.uc.pt.')
```

A restante configuração do ficheiro `.mc` é idêntica à apresentada no 1º caso.

3º CASO

Esta última configuração utiliza os conceitos anteriores adaptados a uma instituição que mantém um *mail-relay* agora constituído por um conjunto de máquinas que não só enviam, mas também recebem todo o correio dessa instituição. Este *mail-relay* é configurado de modo a suportar diversos servidores de correio de grupos de trabalho (faculdades, departamentos, etc.). Isto é, aceita mensagens dirigidas para `user@depart.uc.pt` e reenvia estas mensagens para o `mail-server` departamental que armazena as caixas de correio dos utilizadores de `depart.uc.pt` e suporta o serviço de POP.

Este *mail-relay* oferece escalabilidade horizontal no envio e na recepção do correio, permite a implementação centralizada de medidas comuns *anti-spam* e a simplificação da configuração dos servidores departamentais. A disponibilidade do serviço de recepção também é melhorada, na perspectiva de “mensagens dirigidas para a instituição”.

Notar que neste cenário não é feito qualquer *masquerade* das linhas de cabeçalho das mensagens. De facto o *mail-relay* só pode ser configurado de modo a mascarar apenas um domínio e, neste caso, pretende-se que o *mail-relay* não só receba, mas também envie mensagens com origem em domínios virtuais. O domínio de origem

das mensagens deve então ser correctamente atribuído na parametrização dos MUAs. No servidor de DNS de cada departamento, indicam-se os servidores que recebem correio dirigido a esse departamento:

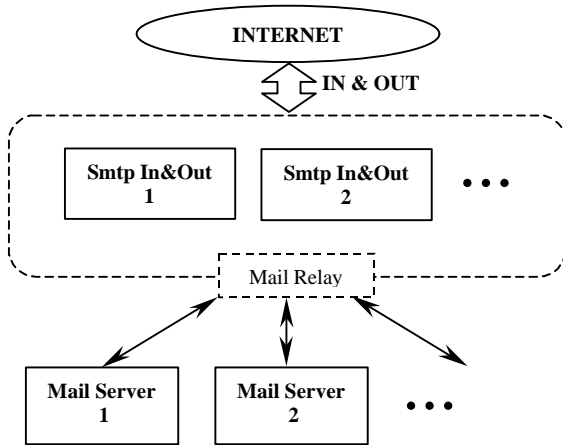


Figura 3

```
IN MX 10 mrelay1.inst.uc.pt.
IN MX 10 mrelay2.inst.uc.pt.
IN MX 10 mrelay3.inst.uc.pt.
```

Notar que não é necessário introduzir o MX secundário porque há servidores suficientes para que, caso um deles deixe de funcionar, o outro o substitua. Nos testes efectuados com o *sendmail* verificou-se que, caso seja tentado o envio para um *mail-relay* que não esteja a responder, este MTA vai tentar automaticamente o próximo MX que se encontre no *round-robin*. Por forma a garantir a configuração inicial estipulada para os MUA externos ao servidor, é necessário definir o endereço *smtp.inst.uc.pt* no DNS de *inst.uc.pt*. Esse endereço aponta agora para o *mail-relay* da seguinte forma,

```
smtp IN CNAME mrelay1.inst.uc.pt.
      IN CNAME mrelay2.inst.uc.pt.
      IN CNAME mrelay3.inst.uc.pt.
```

Passemos a analisar a configuração dos *mail-servers* e *mail-relay*.

Os *mail-servers* departamentais, devem conhecer os *smtpout* como os sistemas capazes de enviar as suas mensagens dirigidas para o exterior. Para tal é necessário definir um *SMART_HOST*. O ficheiro *.mc* será então do tipo:

```
Define(`SMART_HOST`, `smtp.inst.uc.pt`)
FEATURE(masquerade_envelope)
FEATURE(access_db)

MASQUERADE_AS(depart.uc.pt)
```

A macro *MASQUERADE_AS* mascara todo o correio enviado a partir da máquina. Como o servidor departamental não permite relay, o ficheiro */etc/mail/access.txt* fica vazio.

O ficheiro *sendmail.cw* de cada *mail-server* departamental deve conter o nome do domínio pelo qual o servidor aceita correio.

Todos os *mail-relay* serão configurados da mesma forma. Os ficheiros */etc/sendmail.cw* deverão conter a lista dos domínios pelos quais os *mail-servers* aceitarão mensagens.

É no *mail-relay* que serão implementadas as medidas anti-spam. Estas configurações são idênticas às explicadas anteriormente (*access_db*).

A correspondência entre as mensagens recebidas pelo *mail-relay* e o *mail-server* de destino é definida incluindo no ficheiro *.mc* a linha

```
FEATURE(virtusertable,hash -o /etc/mail/virtusertable)
```

No ficheiro */etc/mail/virtusertable.txt* deverão ser colocadas linhas do tipo

```
@depart.uc.pt %!@mail-server1.inst.uc.pt
```

Esta funcionalidade permite encaminhar as mensagens dirigidas ao utilizador *exemplo@depart.uc.pt* para *exemplo@mail-server1.inst.uc.pt*.

Os utilizadores terão que incluir na configuração dos seus MUA o seu endereço de origem correcto. Por exemplo,

Return Address: *exemplo@depart.uc.pt*

O endereço do servidor de POP deve ser alterado para

POP Account: *exemplo@pop.depart.uc.pt*

O **SMTP** permanece inalterado.

Conclusão

Apresentadas as três configurações de servidores de correio electrónico podemos avançar, como primeira conclusão, a limitação da disponibilidade do serviço em caso de falha do sistema que armazena as caixas de correio dos utilizadores. Em tal situação, em qualquer das configurações, as mensagens são armazenadas num sistema de recurso, ao qual os utilizadores não têm acesso, sendo posteriormente enviadas para o servidor logo que este esteja de novo disponível. Este problema só pode ser ultrapassado com *hardware* de alta disponibilidade – sub-sistemas de discos RAID (*Redundant Array of Inexpensive Disks*) e *clustering* de servidores. Uma segunda limitação resulta do elevado número de caixas de correio que poderão ter que ser armazenadas num mesmo directório do servidor, tendo como consequência uma quebra de desempenho ao nível do sistema de ficheiros.

Contudo, podemos constatar que é possível e de simples implementação uma significativa escalabilidade horizontal distribuindo o envio de mensagens por vários servidores. Quando aplicável, a configuração descrita no 3º. caso pode representar uma melhoria significativa em termos de escalabilidade e disponibilidade. Permite ainda suportar um serviço de correio centralizado, onde são mantidos domínios virtuais que podem, contudo, ser transferidos para domínios reais em função das opções das instituições utilizadoras do serviço.

Embora possa parecer trivial, considerámos relevante, atendendo à sua relativa complexidade, incluir os ficheiros de configuração desenvolvidos, que reunimos no anexo apresentado no fim desta comunicação.

As configurações apresentadas são certamente insuficientes para o nível de disponibilidade de serviços de um ISP, mas poderão constituir um compromisso interessante em termos de custos, disponibilidade e

escalabilidade para muitos ambientes de instituições de dimensões e recursos diversos que pretendam administrar o seu próprio serviço de correio electrónico.

Referências:

- [1] Sendmail Consortium, <http://www.sendmail.org/>
- [2] Bryan Costales, *SENDMAIL*, O'Reilly & Associates, Inc.

- [3] Eric Allman, Sendmail Consortium, <http://www.sendmail.org/m4/readme.html>
- [4] Post Office Protocol, RFC 1957, 1734
- [5] Fight Spam on the Internet, <http://spam.abuse.net/>
- [6] Anti-Spam Configuration Control, Sendmail Cons., <http://www.sendmail.org/m4/anti-spam.html>
- [7] Paul Vixie, DNS, BIND and load balancing, <http://www.isc.org/bind-lb.html>

ANEXOS

1º Caso

Configuração do DNS

```
;/etc/namedb/inst.db para o dominio inst.uc.pt
; -----
;
inst.uc.pt.      IN      SOA      ns.uc.pt. postmaster.ns.uc.pt. (
                98071302 ; Serial
                28800  ; Refresh - 8 hours
                7200   ; Retry   - 2 horas
                604800 ; Expire  - 7 dias
                86400  ) ; Minimum - 24 horas
                IN      NS      ns.uc.pt.
                IN      NS      ns.secundario.pt.
; Mail Exchangers
;
                IN      MX      10 mail-server.inst.uc.pt.
                IN      MX      20 backup.inst.uc.pt.

; Atenção que estes exemplos te^m IP's invalidos
mail-server     IN      A      192.168.0.1
                IN      HINFO   "Maquina" "S.O."
                IN      TXT     "Local"
;
pop             IN      CNAME   mail-server.inst.uc.pt.
smtp           IN      CNAME   mail-server.inst.uc.pt.
;
backup         IN      A      192.168.0.2
                IN      HINFO   "Maquina" "S.O."
                IN      TXT     "Local"
```

Ficheiro *.mc*

```
divert(0)
VERSIONID(`@(#)inst.uc.pt      8.9 (Berkeley) 5/19/98')
OSTYPE(mklinux)

FEATURE(nouucp)
FEATURE(use_cw_file)
FEATURE(masquerade_envelope)
FEATURE(masquerade_entire_domain)
FEATURE(access_db)

MASQUERADE_AS(inst.uc.pt)
MASQUERADE_DOMAIN_FILE(`/etc/mail/md')

MAILER(local)
MAILER(smtp)
```

2º Caso

Configuração do DNS

```
;/etc/namedb/inst.db para o dominio inst.uc.pt
; -----
;
inst.uc.pt. IN      SOA      ns.uc.pt. postmaster.ns.uc.pt. (
                98071302 ; Serial
                28800  ; Refresh - 8 hours
                7200   ; Retry   - 2 horas
                604800 ; Expire  - 7 dias
                86400  ) ; Minimum - 24 horas
                IN      NS      ns.uc.pt.
                IN      NS      ns.secundario.pt.
; Mail Exchangers
;
                IN      MX      10 mail-server.inst.uc.pt.
                IN      MX      20 smtpout1.inst.uc.pt.
                IN      MX      20 smtpout2.inst.uc.pt.
                IN      MX      20 smtpout3.inst.uc.pt.

; Atenção que estes exemplos te^m IP's invalidos
; mail-server
mail-server     IN      A      192.168.0.1
                IN      HINFO   "Maquina" "S.O."
```

MAILER(local)
MAILER(smtp)
