

Foto: Roselito Fávero da Silva



Utilização de Rsync com Windows e Linux para cópias automáticas de segurança

Roselito Fávero da Silva¹
Robson Rodrigues Santiago²

Introdução

O Rsync, um utilitário de código livre e disponível sob a licença *GNU – General Public License (GNU GPL)* – para rápida cópia de arquivos, é bastante versátil, pois pode copiar localmente, de uma máquina/para outra máquina sobre qualquer *shell* remoto, ou de/para um *daemon* remoto. Também oferece um grande número de opções que controlam cada aspecto de seu comportamento, e permite especificações muito flexíveis do conjunto de arquivos a ser copiado. Além disso, é famoso por seu algoritmo delta-transfer, que reduz a quantidade de dados enviados pela rede, enviando somente as diferenças entre os arquivos de origem e os arquivos existentes no destino, diminuindo, com isso, o tráfego de dados locais. O Rsync é amplamente usado para cópias de segurança, para espelhamento e como um comando de cópia aperfeiçoado para uso no dia-a-dia (TRIDGELL e MACKERRAS, 2009).

Os procedimentos utilizados para efetuar cópias de segurança de máquinas que operam com sistemas Windows e para enviar as cópias para servidores Linux passaram por profundas modificações na Embrapa Pecuária Sudeste.

O uso do utilitário Rsync tornou possível a automatização total da configuração e do agendamento das cópias de segurança, além de permitir o aumento da confiabilidade nos arquivos armazenados nos repositórios de dados.

Para promover essas soluções, diversas tecnologias disponíveis sob licença GNU GPL foram utilizadas, dentre elas, o Samba, o Plone, o Python e o *shell-script*, integrados por meio de um sistema disponibilizado via *web*, em rede local.

O objetivo deste trabalho é mostrar como o Rsync foi utilizado para a criação de cópias de segurança na Embrapa Pecuária Sudeste.

¹ Matemático, Assistente da Embrapa Pecuária Sudeste, São Carlos, SP. <roselito@cppse.embrapa.br>

² Analista de Sistemas da Embrapa Pecuária Sudeste, São Carlos, SP. <robson@cppse.embrapa.br>

Configuração do Rsync como servidor em máquinas Windows

Não há uma versão de Rsync para ambientes Windows, mas é possível executar o Rsync sob Windows por meio da instalação da biblioteca Cigwin.

O Cygwin é um conjunto de ferramentas eficientes, usadas para ajudar os desenvolvedores no processo de migração de aplicativos do UNIX/ Linux para a plataforma Microsoft Windows. O Cygwin fornece o padrão de código aberto do compilador gcc Red Hat GNU gcc, e o depurador gdb no Windows. Além disso, fornece um padrão UNIX/Linux no ambiente de desenvolvimento e APIs do Windows, incluindo os ambientes de comando. A biblioteca *Cygwin.dll*, incluída com o Cygwin, fornece o subconjunto SVR4 UNIX, BSD e POSIX APIs para permitir rápida portabilidade dos aplicativos UNIX/Linux para a plataforma Windows. O Cygwin pode ser utilizado em Windows 2000, 2003 Server, XP, Vista, 2008 Server, e versões superiores (RED HAT, 2009).

Apesar de todo o potencial das ferramentas do Cygwin, e da possibilidade de se ter um terminal Linux executando dentro do Windows, foi suficiente utilizar apenas a biblioteca *cygwin1.dll* e o aplicativo *cygrunsrv.exe* nas máquinas Windows, o que garante a execução do Rsync e torna o processo transparente para o usuário.

Para configurar o Rsync como servidor, deve ser criada uma pasta *C:\Windows\rsyncd*, na qual serão colocados os seguintes arquivos:

- *rsync.exe*
- *cygrunsrv.exe*
- *cygwin1.dll*
- *rsyncd.conf*
- *rsyncd.secrets*

Os três primeiros podem ser obtidos pelo endereço: <http://www.itfix.no/i2/node/10650>. O arquivo *rsyncd.conf* deve conter a configuração do módulo que será acessado por um cliente Rsync, ou seja, a máquina onde ficarão armazenadas as cópias de segurança. Recomenda-se deixar o acesso ao disco inteiro da máquina Windows liberado no módulo, escolhendo-se as pastas que serão copiadas no cliente, como será visto mais adiante. A seguir, um exemplo de configuração do módulo no arquivo *rsyncd.conf*:

```
# final do arquivo rsyncd.conf
[cDrive]
  path = /
  comment = Entire Drive
  auth users = backupmaster
  secrets file = c:/windows/rsyncd/rsyncd.secrets
  hosts allow = xxx.xxx.xxx.xxx
  read only = true
  list = false
```

É importante notar que o usuário permitido para acessar o módulo não precisa estar cadastrado no servidor, nem no cliente. Esse usuário existe apenas nos arquivos de configuração e de senhas do Rsync. Também é importante verificar as cláusulas *read only* e *list* – que garantem maior segurança para o processo – e o IP da máquina que poderá acessar o conteúdo do módulo.

O arquivo *rsyncd.secrets* contém somente o usuário que pode acessar o conteúdo do módulo e sua respectiva senha, conforme o exemplo:

```
# arquivo rsyncd.secrets
backupmaster:podeacessar
```

As linhas de comando a seguir mostram como o aplicativo *cygrunsrv.exe* executa e inicia o *daemon* do Rsync em uma máquina Windows:

```
c:\windows\rsyncd\cygrunsrv.exe
-I rsyncd
-e CYGWIN=nontsec
-p c:\windows\rsyncd\rsync.exe
-a "--config=c:/windows/rsyncd/rsyncd.conf
--daemon --no-detach"
```

```
c:\windows\rsyncd\cygrunsrv.exe--start rsyncd
```

Essas linhas podem ser colocadas em um arquivo de execução em lote, por exemplo: *rsync.bat*. Em seguida, o registro do Windows pode ser configurado para executar o arquivo *rsync.bat* automaticamente, quando o sistema iniciar. Para facilitar a configuração, pode ser criado um arquivo *rsync.reg* com o conteúdo a seguir:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\windows\Currentversion\Run]
"Rsync"="\"c:\\windows\\rsyncd\\rsyncd.bat\""
```

Para diferentes versões do Windows, alguns ajustes devem ser feitos. No caso do Windows 98, o conteúdo do arquivo de registro deve ser semelhante ao detalhado a seguir:

```
REGEDIT4
[HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows\Currentversion\RunServices]
"rsync"="\windows\rsyncd\rsync98.bat"
```

O arquivo de registro deve ser executado pelo menos uma vez, após a configuração dos demais arquivos. Dessa maneira, o serviço *rsync* será executado quando a máquina for iniciada.

Máquinas Windows com *firewall* devem ter a porta 837 e os arquivos executáveis *c:\windows\rsyncd\cygrunsrv.exe* e *c:\windows\rsyncd\rsync.exe* liberados. Esse procedimento encerra a configuração do servidor.

Uso do Rsync nos repositórios de dados

Nos repositórios de dados da Embrapa Pecuária Sudeste, o ambiente é baseado na distribuição Debian do Linux. Nesse ambiente, o uso do Rsync se resume em linhas de comando organizadas em *shell-scripts*, que acessarão os servidores Windows para efetuar a cópia dos arquivos, como mostra a listagem a seguir:

```
# arquivo copia_servidor1
rsync -gopavc
    --password-file=/etc/p
    backupmaster@servidor1::cDrive/pasta
    /backup_rsync/servidor1 >> servidor1.txt
```

Na linha acima, os valores "usuário", "cDrive" e "pasta" são definidos na configuração da máquina Windows cujo *hostname* na rede é "servidor1". Esses itens indicam, respectivamente, o usuário que pode acessar o servidor Rsync, o módulo que poderá ser acessado e a pasta localizada no caminho definido na configuração do módulo. A pasta *backup_rsync/servidor1* deve ser criada antes que a cópia seja efetuada.

O arquivo *servidor1.txt* criado ao término da execução do comando Rsync é posteriormente enviado ao endereço eletrônico do responsável pela máquina Windows.

Para cada pasta que o usuário queira a respectiva cópia de segurança, deve ser criada uma linha no *shell-script*. Por exemplo:

```
# arquivo copia_servidor1
rsync -gopavc
    --password-file=/etc/p
    backupmaster@servidor1::cDrive/pasta1
    /backup_rsync/servidor1 >> servidor1.txt
```

```
rsync -gopavc
    --password-file=/etc/p
    backupmaster@servidor1::cDrive/pasta2
    /backup_rsync/servidor1 >> servidor1.txt
```

O arquivo */etc/p* contém a mesma senha de acesso que foi definida no servidor Windows, colocada em uma linha única. No caso de várias máquinas Windows atuando como servidoras do Rsync, é aconselhável que haja penas um usuário e uma senha, para facilitar a administração. A seguir, o conteúdo do arquivo */etc/p*:

```
podeacessar
```

Em ambientes Linux, esse arquivo precisa ter permissão 600; caso contrário, a máquina-cliente não conseguirá o acesso ao servidor (TRIDGELL e MACKERRAS, 2009).

Depois de montado o *shell-script*, ele pode ser colocado no *crontab* da máquina, com escolha dos dias, das horas e dos minutos nos quais as cópias serão realizadas. Para automatizar o processo e diminuir a manutenção do administrador, uma linha ao final do arquivo pode mandar uma mensagem ao responsável pela máquina Windows:

```
# arquivo copia_servidor1
rsync -gopavc
    --password-file=/etc/p
    backupmaster@servidor1::cDrive/pasta
    /backup_rsync/servidor1 >> servidor1.txt
mailx -s "Relatorio de copia" < servidor1.txt
responsavell@dominio.xxx.br
```

Todo o resultado do comando *rsync -gopavc* será enviado ao responsável por meio de mensagem eletrônica gerada automaticamente, para que ele analise e peça a inclusão de alguma pasta em sua lista de cópias de segurança.

As cópias de segurança feitas com as opções *gopavc* são incrementais e não removem arquivos. Ou seja, arquivos novos ou modificados serão sobrepostos, mas, se o usuário apagar um arquivo da máquina Windows, ele continuará no servidor até que seja removido manualmente pelo administrador.

Interface gráfica para manutenção dos scripts

A Embrapa Pecuária Sudeste utiliza o gerenciador de conteúdos Plone em sua *intranet*, para a hospedagem e o desenvolvimento de páginas.

Para automatizar o processo de criação de *scripts* nos repositórios por meio de formulários que podem ser acessados via *web*, é preciso ter acesso aos comandos do sistema Linux em cada repositório.

Segundo McKay (2004), um método externo é um módulo Python escrito no sistema de arquivos e então acessado no Plone. Como foi escrito no sistema de arquivos, o módulo não executa em modo restrito do Python, e, portanto, enquadra-se aos padrões de segurança do Plone. Isso significa que podemos escrever um *script* que executa qualquer atividade que se deseje e então chamá-lo de uma página no Plone.

Como mostra Brockmann et al. (2002), um método externo deve ser colocado na pasta **Extensions** da instalação Zope do servidor *web*. A seguir, mostra-se um exemplo de módulo externo escrito em Python e que possibilita a gravação de arquivos em sistema de arquivos montado com Samba em servidor *web*:

```
def deletasistemabk(repositorio,maquina,local):
    saida = os.popen('rm /sistemabk/'+repositorio+'bk/'+local+'copia_'+maquina)
    saida = os.popen('rm /sistemabk/'+repositorio+'cr/'+maquina)
    return saida

def gravasistemabk(texto,repositorio,maquina,hora,local):
    saidabk = os.popen('mkdir -p /sistemabk/'+repositorio+'bk/'+local+maquina)
    saidabk = os.popen('touch /sistemabk/'+local+repositorio+'bk/'+local+maquina)
    saidabk = os.popen('touch /sistemabk/'+local+repositorio+'bk/'+local+'copia_'+maquina)
    arquivo = open('/sistemabk/'+repositorio+'bk/'+local+'copia_'+maquina,"w")
    arquivo.write(texto)
    arquivo.close()
    geracron = os.popen('echo "0 '+hora+' * * 1,3,5 root '+local+'copia_'+maquina+' # Copia do '+maquina+' " > /sistemabk/'+repositorio+'cr/'+maquina)
    return geracron
```

Esse módulo cria a estrutura necessária para o armazenamento dos dados, cria o *shell-script* que conterà os comandos *rsync* para efetuar as cópias, modifica a tabela *cron* da máquina cliente e modifica a data da pasta */var/spool/cron/crontabs* para que o *cron* seja reiniciado automaticamente.

Os parâmetros são enviados por um formulário construído com Plone.

Conclusões

O Rsync mostrou-se uma ferramenta adequada para efetuar cópias de segurança de maneira automatizada, sem prejudicar o tráfego da rede local, e mantendo a integridade dos arquivos, já que, em vez de criar uma cópia em forma de pacote, preserva a estrutura de pastas da máquina original. Além disso, efetua a cópia incrementalmente, sem excluir arquivos que deixam de existir na máquina original. Outra vantagem é que não há custos agregados, uma vez que as licenças das ferramentas utilizadas são disponíveis gratuitamente na internet (*software livre*).

Referências

BROCKMANN, M.; KIRCHNER, K.; PRATT, M.; LÜHNSDORF, S. **Zope: kit de construção de aplicativos de Web**. Rio de Janeiro: Alta Books, 2002. p. 520.

MCKAY, A. **The definitive guide to Plone**. New York: Apress, 2004. 156 p.

RED HAT. **Cygwin**. 2009. Disponível em: <<http://www.redhat.com/services/custom/cygwin/>>. Acesso em: 23 setembro 2009.

TRIDGELL, A.; MACKERRAS, P. **Rsync manual pages**. 2009. Disponível em: <<http://www.samba.org/ftp/rsync/rsync.html>>. Acesso em: 23 setembro 2009.

Comunicado Técnico, 89

Exemplares desta edição podem ser adquiridos na:
Embrapa Pecuária Sudeste
Endereço: Rod. Washington Luiz, km 234
Fone: (16) 3411-5600
Fax: (16) 3361-5754
E-mail: sac@cppse.embrapa.br

1ª edição on line: (2009)

Ministério da
Agricultura, Pecuária
e Abastecimento



Comitê de publicações

Presidente: Ana Rita de Araujo Nogueira.
Secretário-Executivo: Simone Cristina Méo Niciura.
Membros: Maria Cristina Campanelli Brito,
Milena Ambrosio Telles, Sônia Borges Alencar.

Expediente

Revisão de texto: Milena Ambrosio Telles.
Editoração eletrônica: Maria Cristina Campanelli Brito