

Introduzione a Rsync

Trasferimento file ottimizzato

Montebelluna 29 novembre 2007



Licenza d'utilizzo

Copyright © 2007, Manuel Dalla Lana

Questo documento viene rilasciato secondo i termini della licenza Creative Commons
Attribuzione-Non commerciale-Condividi allo stesso modo 3.0 Unported

(<http://creativecommons.org>).

L'utente è libero di:

***riprodurre, distribuire, comunicare al pubblico, esporre in pubblico,
rappresentare, eseguire, recitare e modificare quest'opera***

alle seguenti condizioni:

Attribuzione

Deve riconoscere la paternità dell'opera all'autore originario.

Non commerciale

Non può utilizzare quest'opera per scopi commerciali.

Condividi allo stesso modo Se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con una licenza identica o equivalente a questa.

- Ogni volta che usi o distribuischi quest'opera, devi farlo secondo i termini di questa licenza, che va comunicata con chiarezza.
- In ogni caso, puoi concordare col titolare dei diritti utilizzi di quest'opera non consentiti da questa licenza.
- Questa licenza lascia impregiudicati i diritti morali.

Questo è un riassunto in lingua corrente dei concetti chiave della licenza completa (codice legale), reperibile sul sito Internet

<http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>



Cos'è Rsync?



Rsync (Remote Sync) è uno strumento estremamente versatile e veloce per copiare file in locale e da/verso sistemi remoti.

Rsync è stato creato da Andrew Tridgell (autore anche di Samba) e Paul Mackerras, attualmente mantenuto da Wayne Davison (autore di trn) ed è rilasciato con licenza GNU GPL.



Installazione

Sorgente

```
$ wget http://rsync.samba.org/ftp/rsync/rsync-2.6.9.tar.gz
$ tar zxvf rsync-2.6.9.tar.gz
$ cd rsync-2.6.9/
$ ./configure
$ make
# make install
```

Debian e derivate

```
# apt-get install rsync
```

Red Hat Enterprise Linux (RHEL)

```
# up2date rsync
```

CentOS/Fedora Core Linux

```
# yum install rsync
```



Come funziona?

Rsync sincronizza file e directory da una locazione all'altra minimizzando la quantità di dati da spostare grazie all'algoritmo rsync che utilizza (quando possibile) il delta encoding, spostando quindi solo ciò che differisce tra due versioni di un file.

Inoltre utilizza una sola trasmissione di dati per ogni direzione di comunicazione (upload e download)

Altra caratteristiche utilizzabili sono la compressione dei dati e la ricorsione nelle directory.



L'algoritmo

Ricevente:

- spezza i file in blocchi contigui di dimensione **S**
- per ciascuno dei blocchi calcola 2 checksum (MD4 e uno simile a adler-32)
- spedisce al mittente i checksum

Mittente:

- calcola il checksum per tutti i possibili blocchi di lunghezza **S** del file
- compara i suoi checksum ciclici con quelli inviati dal ricevente, per verificare se ci sono delle corrispondenze
- Se ci sono, viene calcolato e controllato anche il corrispondente hash MD4
- il mittente invia al ricevente solo le parti del file per cui non sono stati trovati blocchi corrispondenti, insieme a istruzioni su come ricomporre i blocchi già presenti e le parti nuove in modo da ottenere una copia esatta del file originale

Risultato: Se le due versioni del file hanno molte sezioni in comune, com'è spesso il caso quando si tratta di versioni diverse dello stesso file, rsync può effettuare la sincronizzazione trasmettendo molti pochi dati rispetto alla dimensione dell'intero file.



Principali opzioni del comando

- `-v` : verbose (`-vv` more verbose)
- `-a` : archive mode; same as `-rlptgoD` (no `-H`, `-A`)
- `-r` : recurse into directories
- `-z` : compress file data
- `--delete` : delete files that don't exist on sender (system)
- `-e "ssh options"` : specify the remote shell to use



Sicurezza prima di tutto!

Utilizzare SEMPRE ssh quando si utilizza rsync attraverso una rete non fidata (LAN, Wifi, Internet,...)

Operazioni semplificate con ssh passwordless

- `local$ ssh-keygen -t dsa`
- `local$ scp ~/.ssh/id_dsa.pub remote`
- `local$ ssh username@remote`
- `remote$ cat ~/id_dsa.pub >> ~/.ssh/authorized_keys`
- `remote$ chmod 644 ~/.ssh/authorized_keys`
- `remote$ exit`
- `local$ ssh username@remote`



Il demone rsyncd

rsync può essere demonizzato, e lavorerà (di default) sulla porta TCP 873

/etc/rsyncd.conf

```
motd file = /etc/rsyncd.motd
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsync.lock
```

```
[simple_path_name]
  path = /rsync_files_here
  use chroot = yes
  comment = My Very Own Rsync Server
  uid = nobody
  gid = nobody
  read only = no
  list = yes
  auth users = username
  secrets file = /etc/rsyncd.secrets
```

/etc/rsyncd.secrets

```
endelwar:miapassword
```



Esempi di utilizzo

```
rsync -t *.c foo:src/
```

```
rsync -avz foo:src/bar /data/tmp
```

```
rsync -avz foo:src/bar/ /data/tmp
```

```
rsync -av host::src /dest
```

```
rsync somehost.mydomain.com::
```

```
rsync host::'modname/dir1/file1  
        modname/dir2/file2' /dest
```

```
rsync -av --rsh=ssh host::module /dest
```

```
rsync -av -e "ssh -l ssh-user" rsync-  
        user@host::module /dest
```

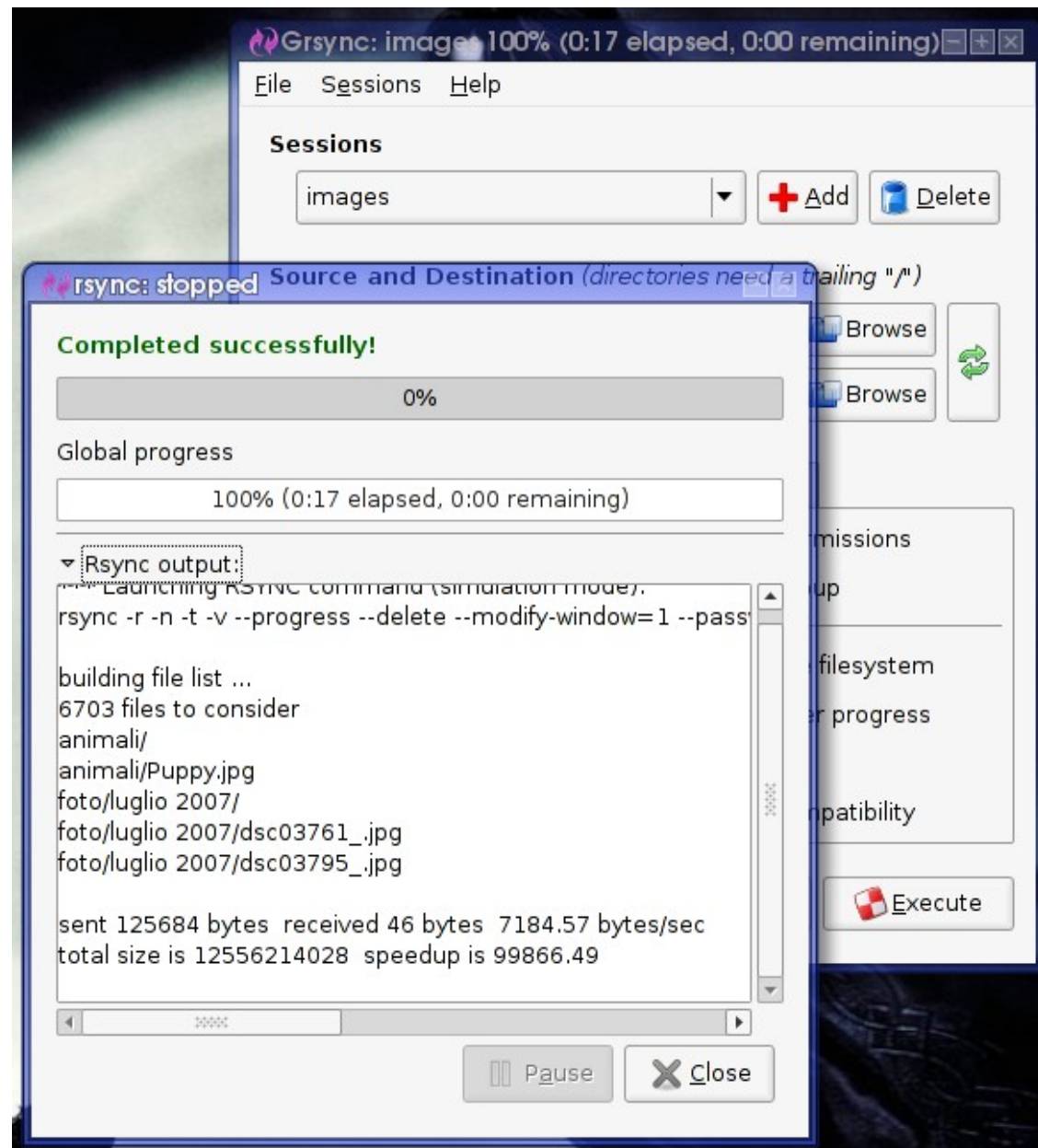


Graphic User Interface/1 - GTK

Grsync

<http://www.opbyte.it/grsync/>

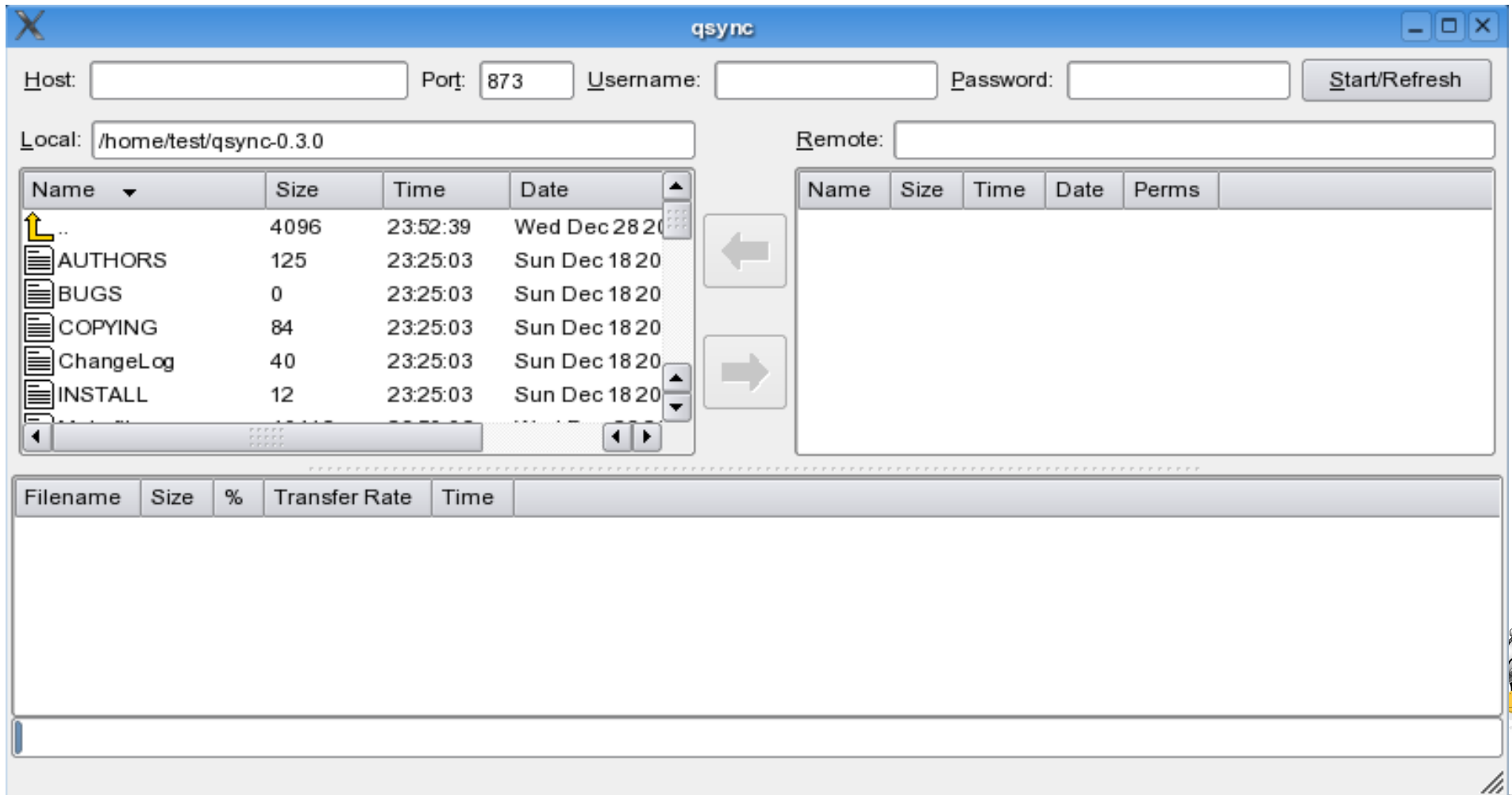
Ultima release 0.6.1 27/11/07



Graphic User Interface/2 - QT

Qsync

<http://transamrit.net/projects/qsync/> Ultima release 0.3 28/12/2005



3.0: aria di novità

A breve sarà rilasciata la versione 3.0 di rsync

- new incremental-recursion algorithm
- Saved memory in the non-incremental-recursion (21-29 bytes per file)
- default --delete algorithm is now --delete-during
- Rsync now allows multiple remote-source args to be specified (`rsync -av host:path/{f1,f2} .`)
- Added the --acls (-A) and --xattrs (-X) option to preserve Access Control Lists and extended attributes
- Added the --iconv option, which allows rsync to convert filenames from one character-set to another during the transfer
- Added the --skip-compress=LIST option to override of the default list of file suffixes that will not be compressed when using --compress
- Protocol 30 now uses MD5 checksums instead of MD4
- Changed the --append option to not checksum the existing data in the destination file, which speeds up file appending.
- Added the --append-verify option, which works like the older --append option (verifying the existing data in the destination file).



Grazie per l'attenzione

Su! Sotto con le domande!

