(/)

OSYNC

Soumis par -badmin- le ven 28/04/2017 - 09:54

Version Stable: 1.2
Version dev: 1.2.1-dev
Code Source: https://github.com/deajan/osync (https://github.com/deajan/osync)
Documentation HTML:
osync_v1.2.html (http://www.netpower.fr/sites/default/files/soft/html-doc/osync_v1.2.html)
Documentation PDF: osync_v1.2.pdf (http://www.netpower.fr/sites/default/files/soft/pdf-doc/osync_v1.2.pdf)

SYNC

A two way filesync script with fault tolerance, resuming, deletion backup and conflict backups running on linux and virtually any system supporting bash. File synchronization is bidirectional, based on rsync, and can be run manually, by cron, or triggered whenever a file changes on master.

ABOUT

Osync provides the following capabilities

- Fault tolerance with resume scenarios
- Email alerts
- Logging facility
- Soft deletition and multiple backups handling
- Before / after command execution
- Time control
- Directory monitoring
- Running on schedule or as daemon
- Batch runner for multiple sync tasks with rerun option for failed sync tasks

Osync uses a master / slave sync schema. It can sync local to local or local to remote directories. By definition, master replica is always a local directory on the system osync runs on. Osync uses pidlocks to prevent multiple concurrent sync processes on/to the same master / slave replica. Be sure a sync process is finished before launching next one, or use osync-batch. You may launch concurrent sync processes on the same system but only for different master replicas.

Currently, it has been tested on CentOS 5.x, 6.x, 7.x, Debian 6.0.7, Linux Mint 14, 15 and 16, Ubuntu 12.04 and 12.10, FreeBSD 8.3 and 10.1. Some users report osync to work on MacOS X too. Microsoft Windows is supported via MSYS environment.

INSTALLATION

Osync has been designed to not delete any data, but rather make backups of conflictual files or soft deletes. Nevertheless, you should always have a neat backup of your data before trying a new sync tool.

You can download the latest stable release of Osync at www.netpower.fr/osync (http://www.netpower.fr/osync) or https://github.com/deajan/osync/archive/stable.tar.gz (https://github.com/deajan/osync/archive/stable.tar.gz)

You may also get the last development version at https://github.com/deajan/osync (https://github.com/deajan/osync) with the following command

```
$ git clone -b "stable" https://github.com/deajan/osync

$ bash install.sh
```

Osync will install itself to /usr/local/bin and an example configuration file will be installed to /etc/osync

Osync needs to run with bash shell. Using any other shell will most probably result in errors. If bash is not your default shell, you may invoke it using

```
$ bash osync.sh [options]
```

On *BSD, be sure to have bash installed. On MSYS, On top of your basic install, you need msys-rsync and msys-coreutils-ext packages.

USAGE

Osync can work with in three flavors: Quick sync mode, configuration file mode, and daemon mode. While quick sync mode is convenient to do fast syncs between some directories, a configuration file gives much more functionnality. Please use double quotes as path delimiters. Do not use escaped characters in path names.

QUICKSYNC EXAMPLE

```
# osync.sh --initiator="/path/to/dir1" --target="/path/to/remote dir2"

# osync.sh --initiator="/path/to/another dir" --target="ssh://user@host.com:22//path/to/dir2"
--rsakey=/home/user/.ssh/id_rsa_private_key_example.com
```

RUNNING OSYNC WITH A CONFIGURATION FILE

You'll have to customize the sync.conf file according to your needs. If you intend to sync a remote directory, osync will need a pair of private / public RSA keys to perform remote SSH connections. Also, running sync as superuser requires to configure /etc/sudoers file. Please read the documentation about remote sync setups. Once you've customized a sync.conf file, you may run osync with the following test run:

```
# osync.sh /path/to/your.conf --dry
```

If everything went well, you may run the actual configuration with one of the following:

```
# osync.sh /path/to/your.conf

# osync.sh /path/to/your.conf --verbose

# osync.sh /path/to/your.conf --no-maxtime
```

Verbose option will display which files and attrs are actually synchronized and which files are to be soft deleted / are in conflict. You may mix "--silent" and "--verbose" parameters to output verbose input only in the log files. No-Maxtime option will disable execution time checks, which is usefull for big initial sync tasks that might take long time. Next runs should then only propagate changes and take much less time.

Once you're confident about your fist runs, you may add osync as cron task like the following in /etc/crontab which would run osync every 30 minutes:

```
*/30 * * * * root /usr/local/bin/osync.sh /etc/osync/my_sync.conf --silent
```

BATCH MODE

You may want to sequentially run multiple sync sets between the same servers. In that case, osync-batch.sh is a nice tool that will run every osync conf file, and, if a task fails, run it again if there's still some time left. The following example will run all .conf files found in /etc/osync, and retry 3 times every configuration that fails, if the whole sequential run took less than 2 hours.

```
# osync-batch.sh --path=/etc/osync --max-retries=3 --max-exec-time=7200
```

Having multiple conf files can then be run in a single cron command like

```
00 00 * * * root /usr/local/bin/osync-batch.sh --path=/etc/osync --silent
```

DAEMON MODE

Additionnaly, you may run osync in monitor mode, which means it will perform a sync upon file operations on master replica. This can be a drawback on functionnality versus scheduled mode because this mode only launches a sync task if there are file modifications on the master replica, without being able to monitor the slave replica. Slave replica changes are only synced when master replica changes occur, or when a given amount of time (default 600 seconds) passed without any changes on master replica. File monitor mode can also be launched as a daemon with an init script. Please read the documentation for more info. Note that monitoring changes requires inotifywait command (inotify-tools package for most Linux distributions). BSD, MacOS X and Windows are not yet supported for this operation mode, unless you find a inotify-tools package on these OSes.

```
# osync.sh /etc/osync/my_sync.conf --on-changes
```

Osync file monitor mode may be run as system service with the osync-srv init script. Any configuration file found in /etc/osync will then create a osync daemon instance. You may run the install.sh script which should work in most cases or copy the files by hand (osync.sh to /usr/bin/local, osync-srv to /etc/init.d, sync.conf to /etc/osync).

```
$ service osync-srv start
```

```
$ chkconfig osync-srv on
```

TROUBLESHOOTING

You may find osync's logs in /var/log/osync-*.log (or current directory if /var/log is not writable). Additionnaly, you can use the --verbose flag see to what actions are going on.