Tahoe-LAVS, storage distribuito fatto in casa

Fabrixxm

Linux Day 2026



#me

- 20+ anni di FOSS e di GL-Como
- spippolatore seriale
- utente linux (Debian e Arch, btw)



GL-Como - Gruppo Linux Como

- si forma nel 2002
- diventa associazione nel 2008
- nel 2018 torna ad essere un gruppo informale



GL-Como

- sito/blog - https://gl-como.it
- social federato - https://social.gl-como.it/~gl-como
- raccolta link - https://linkopedia.gl-como.it/
- server chat XMPP - prosody



Tahoe-LAFS

The Least-Authority File Store

Tahoe-LAFS è un sistema di storage decentralizzato libero ed opensource. Distribuisce i tuoi dati su più server. Anche se alcuni dei server dovessero guastarsi o venire compromessi, l'intero archivio di file continuerbbe a funzionare correttamente, preservando privacy e sicurezza.

- https://tahoe-lafs.readthedocs.io



Tahoe-LAFS

- Creato nel 2006 da “All My Data”, una società di servizi di backup online
- Nel 2009 “All My Data” chiude, Tahoe-LAFS diventa opensource sotto licenza GPL
- Ultima release: v1.20.0 2024-12-18



Tahoe-LAFS

- Una “griglia” di “nodi”
  - “storage”
    - ha uno o più dischi su cui vengono salvati i dati
  - “client”
    - si collega alla griglia senza avere uno storage suo
    - si occupa di salvare, recuperare e verificare i dati dell’utente
  - “introducer”
    - permetti ai “client” e gli “storage” di trovarsi a vicenda



Tahoe-LAFS - Caricamento di un file

- il file viene criptato
  - con AES
  - la chiave è ricavata dal contenuto stesso più un “segreto” legato al client (“*convergence secret*”)



Tahoe-LAFS - Caricamento di un file

- viene spezzato attraverso un algoritmo “erasure-code”
  - in `N` pezzi (*“shares”*)
  - in modo che sia recuperabile con solo `K` (con `K < N`)
    - es, 10 pezzi, ne bastano 3 qualunque per ricreare il file
  - i pezzi vengono poi distribuiti su `H` nodi “storage” diversi

https://github.com/tahoe-lafs/zfec



Tahoe-LAFS - Caricamento di un file

Dalla chiave vengono derivate le *"capability"*.

Una *capability* è una stringa di bit che identificano univocamente un file.

Conoscere questo identificativo è necessario e sufficiente per ottenere accesso ai dati che identifica.



Tahoe-LAFS - Capabilities

In Tahoe un file puo' essere caricato come *immutabile* o *modificabile*

Ogni file *immutabile* ottiene due *capabilities*:

- verify cap : per verificare l'integrità dei dati

  `URI:CHK-Verifier:wzlpt64jaxomjv2vqwqtwj6t6e:tqlhtq46vnbkbsryx7hy4xb5bkjgadv5fi2rian5ueqpddm2juya:3:10:990`

- read-only cap: consente l'accesso in sola lettura ai dati

  `URI:CHK:ijd6kv7z46v3mptvm2nlimei4e:tqlhtq46vnbkbsryx7hy4xb5bkjgadv5fi2rian5ueqpddm2juya:3:10:990`

Un file *modificabile* ottiene anche una *capability* "read-write"

`URI:SSK:ooblc5bwdrojxc3gx7nnq6issu:o4qjslh34lzm7wwu2xxm56tspuebukc7p2j2wbxqgjnpednsr67q`



Tahoe-LAFS - Sharing

Condividere una *capability* equivale condividere i permessi relativi a quel file.

File writecap	`URI:SSK:bepahximuzdghofk5jezj3xrvm:fswtjrrbul3kxs57lfwsqghjgy22tkphgyxmlqsgunpydzm2fc2q`
File readcap	`URI:SSK-RO:drqhykehxs7qcazfadobxxsxvq:fswtjrrbul3kxs57lfwsqghjgy22tkphgyxmlqsgunpydzm2fc2q`
File verifycap	`URI:SSK-Verifier:vyn65w4t7v4dnqkwddjtlppcx4:fswtjrrbul3kxs57lfwsqghjgy22tkphgyxmlqsgunpydzm2fc2q`



Thaoe-LAFS - File system

Un oggetto JSON che descrive il contenuto della directory con nome, metadati e *capabilities* del file in questione

Tahoe-LAFS Directory SI=mwpp6

Type	Filename	Size	Times
FILE	file.txt	13	lcr: 2026-07-24 18:38:31 lmo: 2026-07-24 18:50:32
FILE	firma.txt	990	lcr: 2026-07-22 17:19:38 lmo: 2026-07-24 18:56:53
DIR-RO	rowiki	-	lcr: 2026-07-22 20:36:17 lmo: 2026-07-22 20:36:17
FILE	test.bas	218	lcr: 2025-11-14 11:56:10 lmo: 2025-11-14 11:56:10
DIR	wiki	-	lcr: 2026-06-05 09:17:41 lmo: 2026-06-05 09:17:41



Thaoe-LAFS - File system


```
[
 "dirnode",
 {
  "mutable": true,
  "children": {
        ...
  },
  "verify_uri": "URI:DIR2-Verifier:mwp...t7u:clx...y4a",
  "ro_uri": "URI:DIR2-RO:2x5...rs4:clx...y4a",
  "rw_uri": "URI:DIR2:xng...ziu:clx...y4a",
 }
]
```




Thaoe-LAFS - File system

Un file immutabile


```
"firma.txt": [
 "filenode",
 {
  "mutable": false,
  "format": "CHK",
  "size": 990,
  "metadata": {
   "ctime": 1784729776, "mtime": 1760368937, "st_mode": 33188,"st_ino": 18877788,"st_dev": 2052,"st_uid": 1000, "st_gid": 1000,
   "tahoe": {
    "linkcrtime": 1784733578.0753376,
    "linkmotime": 1784912213.5291348
   }
  },
  "verify_uri": "URI:CHK-Verifier:wzl...t6e:tql...uya:3:10:990",
  "ro_uri": "URI:CHK:ijd...i4e:tql...uya:3:10:990"
 }
],
```




Thaoe-LAFS - File system

Una sottocartella modificabile


```
"wiki": [
 "dirnode",
 {
  "mutable": true,
  "metadata": {
   "tahoe": {
    "linkcrtime": 1780643861.2542055,
    "linkmotime": 1780643861.2542055
   }
  },
  "verify_uri": "URI:DIR2-Verifier:wn4...wom:ixq...cga",
  "ro_uri": "URI:DIR2-RO:ept...7my:ixq...cga",
  "rw_uri": "URI:DIR2:pc5...lou:ixq...cga"
 }
]
```




Thaoe-LAFS - File system

La stessa sottocartella di prima, ma immutabile


```
"rowiki": [
 "dirnode",
 {
  "mutable": true,
  "metadata": {
   "tahoe": {
    "linkcrtime": 1784745377.5771527,
    "linkmotime": 1784745377.5771527
   }
  },
  "verify_uri": "URI:DIR2-Verifier:wn4...wom:ixq...cga",
  "ro_uri": "URI:DIR2-RO:ept...7my:ixq...cga"
 }
],
```




Tahoe-LAFS - User Interface

CLI

```
$ tahoe
Usage: tahoe [global-options] <command> [command-options]
Commands:
    backup                  Make target dir look like local dir.
    cp                      Copy one or more files or directories.
    get                     Retrieve a file from the grid.
    ln                      Make an additional link to an existing file
                            or directory.
    ls                      List a directory.
    mkdir                   Create a new directory.
    put                     Upload a file into the grid.
    unlink                  Unlink a file or directory on the grid.
    webopen                 Open a web browser to a grid file or
                            directory.
```



Tahoe-LAFS - User Interface

```
$ tahoe put example.txt
200 OK
URI:CHK:6tp...syu:yy6...e5q:3:10:2595

$ tahoe ln URI:CHK:6tp...syu:yy6...e5q:3:10:2595 \
    URI:DIR2:xng...ziu:clx...y4a/example.txt

$ tahoe ls URI:DIR2:xng...ziu:clx...y4a
example.txt
file.txt
firma.txt
rowiki
test.bas
wiki

$ tahoe get URI:DIR2:xng...ziu:clx...y4a/example.txt
[...]
```



Tahoe-LAFS - Aliases

```
$ tahoe add-alias home: URI:DIR2:xng...ziu:clx...y4a

$ tahoe put example.txt home:example.txt
200 OK
URI:CHK:6tp...syu:yy6...e5q:3:10:2595

$ thaoe ls home:
example.txt
file.txt
firma.txt
rowiki
test.bas
wiki

$ tahoe ls home:wiki/
logo-glc.png
wiki.html
```



Tahoe-LAFS - Backup

```
$ tahoe backup --help
Usage: tahoe [global-options] backup [options] FROM ALIAS:TO

[...]

Add a versioned backup of the local FROM directory to a timestamped
subdirectory of the TO/Archives directory on the grid,
sharing as many files and directories as possible with earlier backups.
Create TO/Latest as a reference to the latest backup.
```



Bakup

- viene creata `Archives/YYYY-MM-DD_hh:mm:ss`
- al suo interno viene ricreata la struttura file locale
- i file vengono caricati come immutabili (deduplicati)
- viene creata la cartella `Latest` che fa riferimento allo stesso oggetto directory dell'ultimo `Archives/YYYY-MM-DD` (hardlink)



Backup

Sulla VPS

```
prosody@vps $ tahoe backup --exclude="*upload*" \
                /var/lib/prosody/ tahoe:prosody
```



Backup

In locale

```
$ tahoe ls -l glbackup:prosody
drwx - Jul 19 02:00 Archives
dr-x - Jul 19 02:02   Latest

$ tahoe ls -l glbackup:prosody/Archives
dr-x - Jul 19 02:02 2026-07-19_00:02:57Z
dr-x - Jul 26 02:00 2026-07-26_00:00:38Z
[...]
dr-x - Oct 11 02:00 2026-10-11_00:00:41Z
dr-x - Oct 18 02:02 2026-10-18_00:02:13Z

$ tahoe ls -l glbackup:prosody/Archives/2026-10-18_00:02:13Z
dr-x - Jul 18 17:05                        .tahoe
dr-x - Oct 17 17:32 conference%2egl%2dcomo%2eit
dr-x - Oct 17 17:32               gl%2dcomo%2eit
dr-x - Oct 17 17:32                    localhost
dr-x - Oct 15 16:50     pubsub%2egl%2dcomo%2eit

$ tahoe ls -l glbackup:prosody/Latest
dr-x - Jul 18 17:05                        .tahoe
dr-x - Oct 17 17:32 conference%2egl%2dcomo%2eit
dr-x - Oct 17 17:32               gl%2dcomo%2eit
dr-x - Oct 17 17:32                    localhost
dr-x - Oct 15 16:50     pubsub%2egl%2dcomo%2eit
```



Backup

```
$ tahoe ls -l --uri glbackup:prosody/Archives
dr-x - Oct 11 02:00 2026-10-11_00:00:41Z URI:DIR2-CHK:nrbyzydtqybnrbg73yvv6spz7y:xxeej23wfqwcr7yfyrvp3t6mf7rsrqmk7e3shxsdq6ad7khmuvza:3
dr-x - Oct 18 02:02 2026-10-18_00:02:13Z URI:DIR2-CHK:yzawl5xxjtep5tkzpzc6qscznm:37vheru246lm7vkqqouq7acucnznp27d4hjru5ce37qfavnuqwla:3
[...]

$ tahoe ls -l --uri glbackup:prosody/
drwx - Jul 19 02:00 Archives URI:DIR2:eynu74jdkrcpzvb3wk4a24vr7e:unjbvpexvkhua42iwbcfz3ceqxxvw5hxeolcrpokjdkds4xuo5oq
dr-x - Jul 19 02:02   Latest URI:DIR2-CHK:yzawl5xxjtep5tkzpzc6qscznm:37vheru246lm7vkqqouq7acucnznp27d4hjru5ce37qfavnuqwla:3:10:1394
```



La Griglia del GL-Como

- 4 storage servers
- 2 introducers
- nodo ‘client’ che gira sulla VPS
  - total (`N`): 10
  - needed (`K`): 3
  - happy (`H`): 3

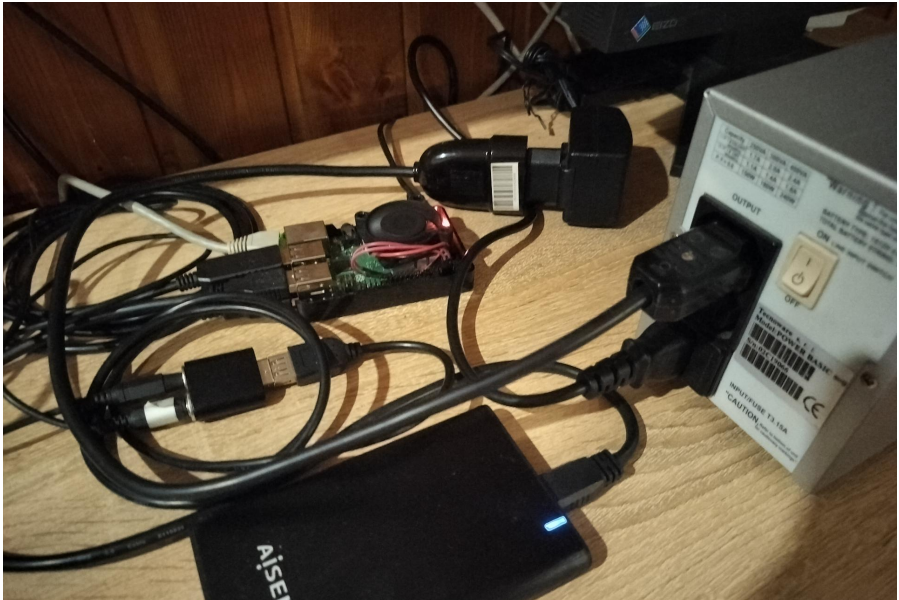


La Griglia del GL-Como

- Raspberry Pi 3 B+, 1GB RAM , 1TB disco USB dedicato, Raspbian
- SoC A20, 1 GB di RAM, 1TB disco USB, Debian
- VPS con 2cpu, 2Gb di RAM, 600Gb di disco, Debian
- Raspberry Pi 4 B, 8Gb di RAM, ~3T a disposizione su Raid 10 software (brr), Debian



La Griglia del GL-Como



Il Backup

Sulla VPS

- Nodo client configurato per collegarsi alla griglia
  - in esecuzione come servizio
  - interfaccia http in ascolto su localhost
- CLI 'nudo' configurato per l'utente che ha accesso ai file di cui fare backup
  - non è un nodo, ma si collega via api http al nodo locale
- Cronjob che lancia `tahoe backup ...` per conto dell'utente



Risultati

In ‘test’ da luglio 2026

Un backup alla settimana dei file riguardanti prosody (~72Mb)

Spazio su disco della cartella dello storage Tahoe: 2,5G (altri file sono caricati, tra cui una immagine squashfs della vecchia VPS da ~1G)

Abbiamo avuto (e avremo) nodi down, nessun problema a interagire con i file.



Da vedere

Aggiungere altri dati ai backup tendendo d'occhio lo spazio occupato negli storage.

Cosa succede se si rimuove una cartella di backup vecchia?

Dopo quanto si libera lo spazio? Si libera lo spazio?



Lease

Ogni share ha una o più "lease", che sono gestiti dai client che vogliono che il file/directory sia conservato. Lo storage accetta ogni share per un periodo predeterminato di tempo, e gli è permesso cancellarlo se tutti i lease scadono.



Lease

Gli storage sono configurati con la scadenza a 31 giorni

Uno script sul server aggiorna i *lease* di tutti gli *share* che puo' raggiungere partendo da un alias.

Scollegando un oggetto dall'albero, in teoria il *lease* non viene aggiornato, e in teoria il nodo dovrebbe cancellare i dati alla scadenza del *lease*.



Cose che non vanno benissimo

- Il progetto è ‘rallentato’
  - ultima release: v1.20.0 2024-12-18
  - poca attività sul tracker e su github
  - ultimo messaggio in ‘tahoe-dev’ il 2026-01-14
    https://lists.tahoe-lafs.org/pipermail/tahoe-dev/2026-January/010135.html
- L’installazione diventa difficile
  - in Debian stabile il pacchetto non dichiara le dipendenze, vanno installate a mano
    - il pacchetto è corretto su Sid
  - pypi?
- Python+Twisted = mesasggi di errore poco user-friendly
- Rimasugli della transizione python2 -> python3
- Manca un comando per il restore del backup



Altre funzionalità

- Interfaccia web / api REST
  - possibilità di creare client alternativi
- Autoconfigurazione via Magic Wormhole
- Accesso via SFTP
- Tahoe-LAVS su Tor o I2P (leento, ma anonimo)



Linki

https://tahoe-lafs.org/

https://github.com/tahoe-lafs/

