POHMELFS usage information.

Mount options.
All but index, number of crypto threads and maximum IO size can changed via remount.

idx=%u
 Each mountpoint is associated with a special index via this option.
 Administrator can add or remove servers from the given index, so all mounts,
 which were attached to it, are updated.
 Default it is 0.

trans_scan_timeout=%u
 This timeout, expressed in milliseconds, specifies time to scan transaction
 trees looking for stale requests, which have to be resent, or if number of
 retries exceed specified limit, dropped with error.
 Default is 5 seconds.

drop_scan_timeout=%u
 Internal timeout, expressed in milliseconds, which specifies how frequently
 inodes marked to be dropped are freed. It also specifies how frequently
 the system checks that servers have to be added or removed from current working set.
 Default is 1 second.

wait_on_page_timeout=%u
 Number of milliseconds to wait for reply from remote server for data reading command.
 If this timeout is exceeded, reading returns an error.
 Default is 5 seconds.

trans_retries=%u
 This is the number of times that a transaction will be resent to a server that did
 not answer for the last @trans_scan_timeout milliseconds.
 When the number of resends exceeds this limit, the transaction is completed with error.
 Default is 5 resends.

crypto_thread_num=%u
 Number of crypto processing threads. Threads are used both for RX and TX traffic.
 Default is 2, or no threads if crypto operations are not supported.

trans_max_pages=%u
 Maximum number of pages in a single transaction. This parameter also controls
 the number of pages, allocated for crypto processing (each crypto thread has
 pool of pages, the number of which is equal to 'trans_max_pages'.
 Default is 100 pages.

crypto_fail_unsupported
 If specified, mount will fail if the server does not support requested crypto operations.
 By default mount will disable non-matching crypto operations.

mcache_timeout=%u

 Maximum number of milliseconds to wait for the mcache objects to be processed.
 Mcache includes locks (given lock should be granted by server), attributes (they should be
 fully received in the given timeframe).
 Default is 5 seconds.

Usage examples.

Add server server1.net:1025 into the working set with index $idx
with appropriate hash algorithm and key file and cipher algorithm, mode and key file:

```
$cfg A add -a server1.net -p 1025 -i $idx -K $hash_key -k $cipher_key
```

Mount filesystem with given index $idx to /mnt mountpoint.
Client will connect to all servers specified in the working set via previous command:

```
mount -t pohmel -o idx=$idx q /mnt
```

Change permissions to read-only (-I 1 option, '-I 2' - write-only, 3 - rw):

```
$cfg A modify -a server1.net -p 1025 -i $idx -I 1
```

Change IO priority to 123 (node with the highest priority gets read requests).

```
$cfg A modify -a server1.net -p 1025 -i $idx -P 123
```

One can check currect status of all connections in the mountstats file:

```
# cat /proc/$PID/mountstats
...
device none mounted on /mnt with fstype pohmel
idx addr(:port) socket_type protocol active priority permissions
0 server1.net:1026 1 6 1 250 1
0 server2.net:1025 1 6 1 123 3
```

Server installation.

Creating a server, which listens at port 1025 and 0.0.0.0 address.
Working root directory (note, that server chroots there, so you have to have appropriate permissions)
is set to /mnt, server will negotiate hash/cipher with client, in case client requested it, there
are appropriate key files.
Number of working threads is set to 10.

```
# ./fserver -a 0.0.0.0 -p 1025 -r /mnt -w 10 -K hash_key -k cipher_key

 -A 6                   - listen on ipv6 address. Default: Disabled.
 -r root                - path to root directory. Default: /tmp.
 -a addr                - listen address. Default: 0.0.0.0.
 -p port                - listen port. Default: 1025.
 -w workers             - number of workers per connected client. Default: 1.
 -K file                - hash key size. Default: none.
 -k file                - cipher key size. Default: none.
 -h                     - this help.
```

Number of worker threads specifies how many workers will be created for each client.
Bulk single-client transafers usually are better handled with smaller number

info.txt

(like 1-3).