Schrittmacher 1.8

Users Manual

Purpose

Schrittmacher's purpose is to keep a pair of applications up and running at all times. If one instance is up, the other will be on standby. If the first one fails, the second one will go into "hot" mode and take over the load.

It was developed for the MessageMover SMSC, a software based on UniversalSS7 stack.

Operation

Schrittmacher runs as a daemon on a Unix system. It talks to another Schrittmacher instance on another host by sending each other UDP packets over a common lan. Both instances send each other heartbeats at predefined intervalls where they negotiate who is HOT and who is STANDBY.

The application which wants to be protected makes uses of Schrittmacherclient library. It has to instantiate a schrittmacherClient object and set the resource-id and the schrittmacher UDP port. And start sending status messages at regular intervall. The client library will signal to the schrittmacher daemon over localhost UDP packets its statuts which are

LHOT Local Application is considering itself "Hot"
LSBY Local Application is considering itself "Standby"
LUNK Local Application doesnt know its status (because it just started after a crash for example)
LFAI Local Application is requesting a failover (asking the other side to go Hot and itself going standby)

The Schrittmacher daemon signals back to the application by the use of signals

SIG_USR1 The Application should go to HOT status (if not already)
SIG_USR2 The Application should go to Standby status (if not already)

The communication between the schrittmacher daemons happens over the following messages:

HOTT Local Daemon is considering itself "Hot"
STBY Local Daemon is considering itself "Standby"
UNK Local Daemon doesnt know its status (because it just started after a crash for example)
TREQ Local Daemon tells the remote it wishes to take over
TCNF Daemon confirms the takeover to the remote
TREJ Daemon rejects the takeover
FAIL Daemon tells it is in failed state and implies a takeover confirmation

Command line options

```
--config-file {filename}     read the filename for its configuration
--daemonize                  put the application to background and
return
--make-pidfile               create the pid file for schrittmacher
itself
--pidfile {filename}         specifiy the filename where the pid is
written to
--parachute                  create a parachute process which
watches the schrittmacher daemon
--version                    print the version and exit
```

The config file

Schrittmacher's config file is normally `./schrittmacher.conf` or `/etc/schrittmacher/schrittmacher.conf`

It contains the following sections

```
group = core
local-address   = the local IP we are listening to
remote-address  = the peer's IP address
shared-address  = a virtual IP which is shared between the two
port            = udp port number
http-port       = tcp port number for the admin interface
log-dir         = directory where the log is written
heartbeat       = how often the heartbeat is sent (Default 2.0
seconds)
timeout         = how long until other side is considered dead
(default: 3*heartbeat)

group = resource
name                    = the resource-id
start-action            = script to call on start
stop-action             = script to call on stop
priority                = default priority on this daemon.
pid-file                = pid file name
interface-activate      = activate action script
interface-deactivate    = deactivate action script
startup-delay           = seconds to wait before starting
heartbeat-intervall     = heartbeat intervall in seconds
```

The environment

the scripts called from schrittmacher can use the following environmental variables

```
LOCAL_ADDRESS    the local ip address (from core config)
REMOTE_ADDRESS   the remote ip address (from core config)
SHARED_ADDRESS   the shared ip address (from core config)
RESOURCE_NAME    the resource name (from resource config)
PID_FILE         the pid file (from resource coding)
HEARTBEAT_INTERVAL the heartbeat intervall (from resource config)
ACTION           the action to take:
                 'activate','deactivate','start' or 'stop'


if it goes from Standby to hot
the actions activate & start are called

if it goes from Hot to Standby
the actions stop & deactivate are called
```

```
#
# schrittmacher config example
#
group = core
local-address = 10.22.52.190
remote-address = 10.22.52.191
shared-address = 10.22.52.189
http-port      = 7700
port           = 7700


group = resource
name = smsc-i400
start-action = "kill -10 `cat /var/run/MessageMover/i400.pid`"
stop-action = "kill -12 `cat /var/run/MessageMover/i400.pid`"
interface-activate = ip addr add 10.22.52.189/25 dev eth0 label eth0:9
interface-deactivate = ip addr del 10.22.52.189/25 dev eth0 label eth0:9
```