Systemd

François Ducrot
Département de Mathématiques, Université d'Angers

10 octobre 2012

Contenu

1. Introduction
2. Différentes stratégies de démarrage
3. Architecture de Systemd
4. Contrôle des processus
5. Systemctl
6. Conclusion

Pourquoi mon intérêt pour systemd ?

On m'appris, quand j'étais petit, qu'un serveur applicatif se doit de fonctionner en mode non graphique.
Dans le bon vieux temps de init system V, je savais faire :

/etc/inittab

```
id :3 :initdefault :
```

Pourquoi mon intérêt pour systemd ?

On m'appris, quand j'étais petit, qu'un serveur applicatif se doit de fonctionner en mode non graphique.
Depuis le passage à Fedora 15, puis Fedora 17 :

/etc/inittab

```
# inittab is no longer used when using systemd.
#
# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
```

Depuis ce temps, par flemme de notre part, nos serveurs sont en mode graphique.
Il est temps de réagir et de comprendre de nouveau comment ça marche.

Plus sérieusement

Comprendre le système de démarrage d'une machine, et en particulier le processus de PID 1, qui est responsable du lancement de tous les autres processus.
Différentes alternatives

- Init system V
- Upstart : Ubuntu (vers. $\geq$ 6.10), Fedora (9 $\leq$ vers. $\leq$ 14), RedHat 6
- Systemd : Fedora (vers. $\geq$ 15), RedHat 7 (?), openSuse (vers. $\geq$ 12.1), Mandriva, ...
- Mais aussi : Launchd (Mac OS X), Solaris SMF

Les buts de Systemd

- Accélérer le démarrage du système (intérêt ? ?)
- Éliminer le recours à des scripts shell (robustesse et rapidité)
- Diminuer le nombre de processus lancés au démarrage
- Rationaliser la gestion des services
- Uniformiser la gestion des services (architecture commune à toutes les distributions)
- Inclure de nouvelles fonctionnalités dans le système d'init
- Tout régenter (un point de vue souvent assez impérialiste !)
- Linux Only !

Mes sources

- Les nombreuses pages de manuel de `systemd` et de ses utilitaires et fichiers de configuration.
- Le blog de l'auteur de systemd Lennart Poettering : `http://0pointer.de/blog`
- Le blog du projet Fedora consacré à systemd : `http://fedoraproject.org/wiki/Systemd`
- L'autopsie d'une Fedora 17

Init System V

- Un niveau d'éxécution (runlevel) défini par `/etc/inittab`
- Des scripts dans `/etc/init.d` lançant les processus
- Des répertoires `/etc/rcx.d`, pour `x` = 0,1,...,6 contenant des liens symboliques vers les scripts de `/etc/init.d`

Init System V

Un comportement séquentiel, géré par les noms des liens dans `/etc/rcx.d` :

```
# ls -l /etc/rc3.d/
total 292
K02avahi-dnsconfd -> ../init.d/avahi-dnsconfd
K02dhcdbd -> ../init.d/dhcdbd
K02NetworkManager -> ../init.d/NetworkManager
K02NetworkManagerDispatcher -> ../init.d/NetworkManagerDispatcher
K02oddjobd -> ../init.d/oddjobd
K05conman -> ../init.d/conman
K05innd -> ../init.d/innd
K99readahead_later -> ../init.d/readahead_later
S00microcode_ctl -> ../init.d/microcode_ctl
S04readahead_early -> ../init.d/readahead_early
S95atd -> ../init.d/atd
S97yum-updatesd -> ../init.d/yum-updatesd
S98avahi-daemon -> ../init.d/avahi-daemon
S98haldaemon -> ../init.d/haldaemon
S99firstboot -> ../init.d/firstboot
S99local -> ../rc.local
S99smartd -> ../init.d/smartd
```

Init System V

Les scripts gèrent le démarrage, l'arrêt, le redémarrage, les verrous

```
#!/bin/sh
#
### BEGIN INIT INFO
# Provides : nbd-server
# Default-Start :
# Default-Stop : 0 1 6
# Should-Start :
# Required-Start : network
# Required-Stop :
# Short-Description : Start and stop the NBD server
# Description : Network Block Device
### END INIT INFO
#
# The fields below are left around for legacy tools (will remove later).
#
# chkconfig : - 98 35
# description : Network Block Device
# processname : nbd-server
# pidfile : /var/run/nbd.pid
. /etc/init.d/functions

RETVAL=0

prog=nbd-server
nbd=/usr/local/bin/nbd-server
NBDARGS=
lockfile=/var/lock/subsys/nbd
pidfile=/var/run/nbd.pid
```

Init System V

Les scripts gèrent le démarrage, l'arrêt, le redémarrage, les verrous

```
 start() {
    [ -x $nbd ] || return 5

    pidofproc $prog >/dev/null 2>&1
    RETVAL=$?
    [ $RETVAL -eq 0 ] && return $RETVAL
    echo -n "Starting nbd-server :"
    daemon $nbd $NBDARGS  2>/dev/null
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && touch $lockfile
    return $RETVAL
}
stop() {
    pidofproc $prog >/dev/null 2>&1
    if [ $? -ne 0 ]; then
        RETVAL=7
        return $RETVAL
    fi
    echo -n "Shutting down nbd-server :"
    killproc $prog
    RETVAL=$?
    [ $RETVAL = 0 ] && success || failure
    echo
    [ $RETVAL = 0 ] && rm -f $lockfile
    return $RETVAL
}
```

Init System V

Les scripts gèrent le démarrage, l'arrêt, le redémarrage, les verrous

```
 case "$1" in
    start)
        start
        RETVAL=$?
       ;;
    stop)
        stop
        RETVAL=$?
       ;;
    restart)
        stop; start
        RETVAL=$?
       ;;
    status)
        status $nbd
        RETVAL=$?
       ;;
    *)
        echo "Usage : $0 start|stop|restart|status"
        RETVAL=3
       ;;
esac

exit $RETVAL
```

Upstart

- Des Jobs décrits par des fichiers `*.conf`, dans `\etc\init`.

```
networking - configure virtual network devices
#
# This task causes virtual network devices that do not have an associated
# kernel object to be started on boot.

description     "configure virtual network devices"

emits static-network-up
emits net-device-up

start on (local-filesystems
          and (stopped udevtrigger or container))

task

pre-start exec mkdir -p /run/network

exec ifup -a
```

Upstart

- Des Jobs décrits par des fichiers `*.conf`, dans `\etc\init`.
- Pilotage par événements : Un job émet des événements

```
networking - configure virtual network devices
#
# This task causes virtual network devices that do not have an associated
# kernel object to be started on boot.

description     "configure virtual network devices"

emits static-network-up
emits net-device-up

start on (local-filesystems
          and (stopped udevtrigger or container))

task

pre-start exec mkdir -p /run/network

exec ifup -a
```

Upstart

- Des Jobs décrits par des fichiers `*.conf`, dans `\etc\init`.
- Pilotage par événements : Un job émet des événements et démarre lors de l'émission d'autre événements

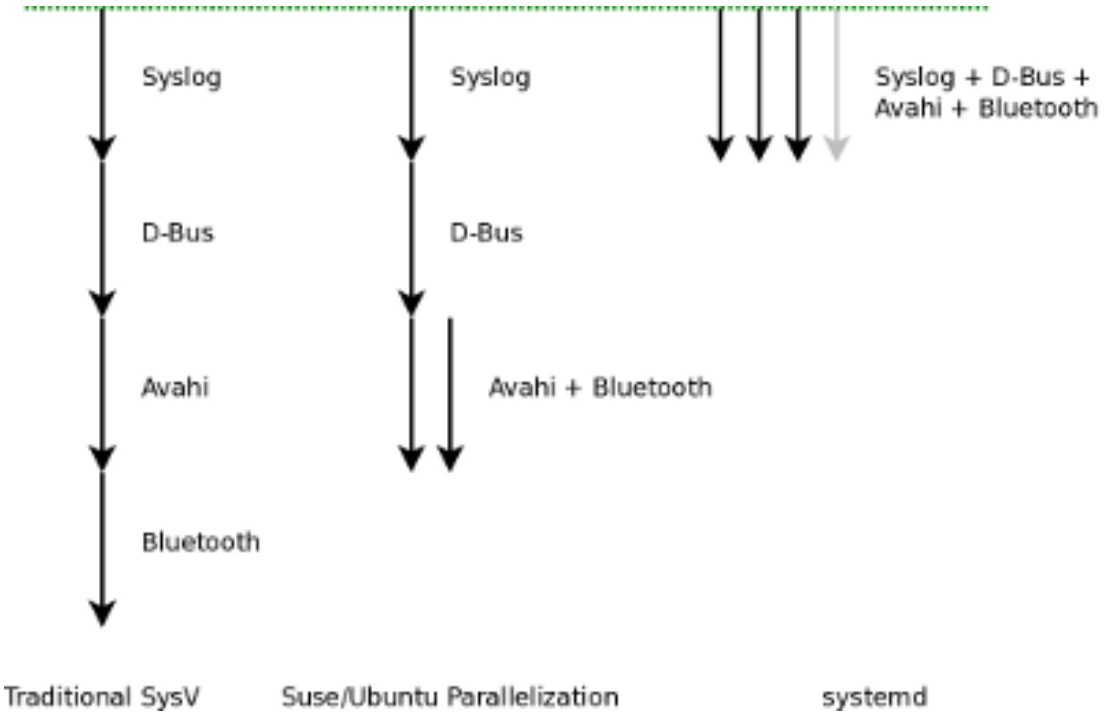
```
networking - configure virtual network devices
#
# This task causes virtual network devices that do not have an associated
# kernel object to be started on boot.

description     "configure virtual network devices"

emits static-network-up
emits net-device-up

start on (local-filesystems
          and (stopped udevtrigger or container))

task

pre-start exec mkdir -p /run/network

exec ifup -a
```

Systemd

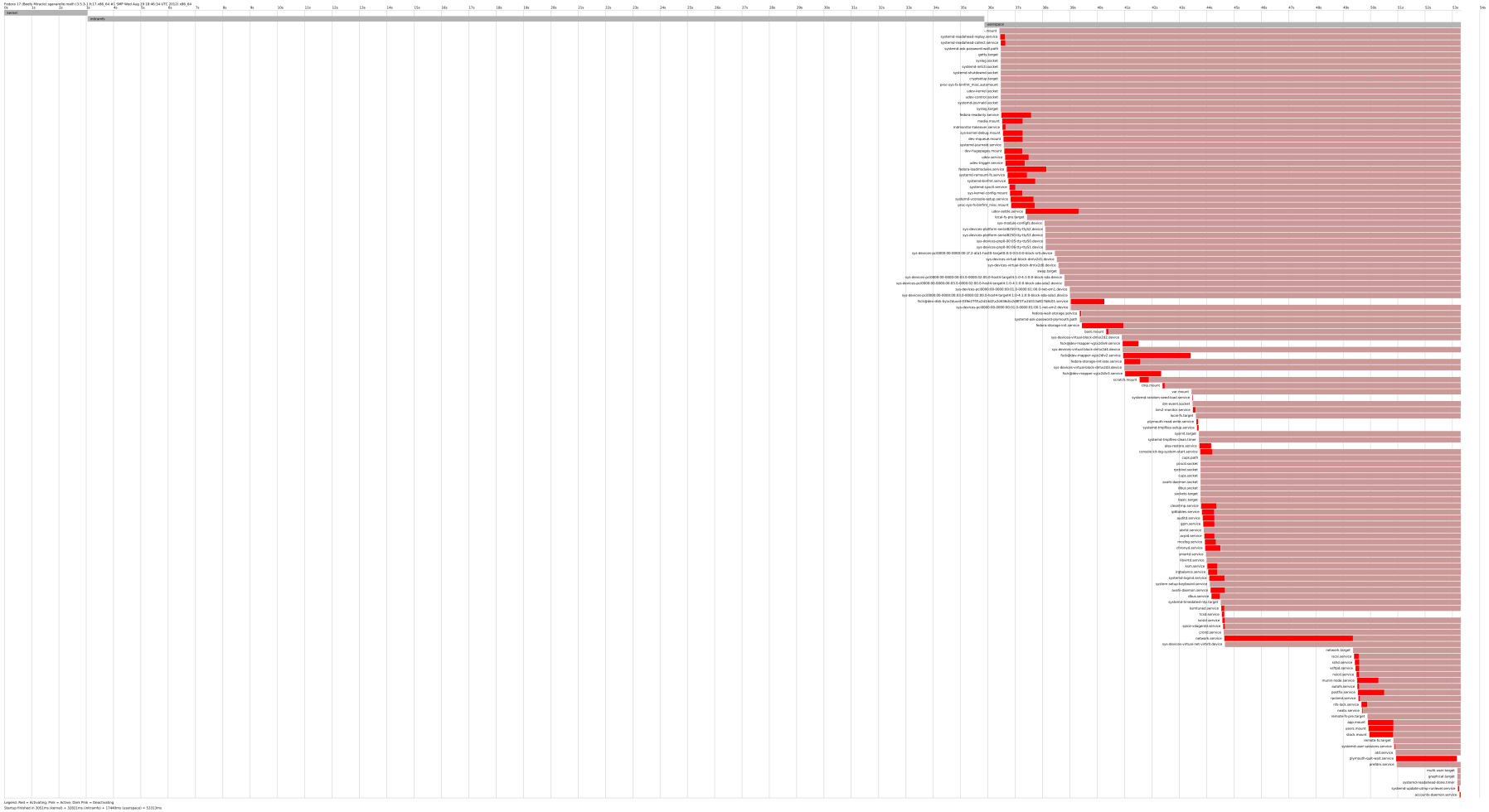
- Parallélisation des processus par ouverture de sockets : systemd peut lancer en parallèle deux processus, même s'ils sont interdépendants.
- Pour chaque processus à lancer, systemd ouvre une socket unix.
- Les processus dépendant d'un processus donné vont se connecter à la socket correspondante, alors que le service réel n'est pas encore forcément actif.
- Lorsque le processus sera actif, systemd le connectera à sa socket et les clients pourront être servis.

Parallelisation des processus : exemple publicitaire

Exemple extrait du blog de L. Poettering :
Les services Avahi et Bluetooth nécessitent tous deux le fonctionnement de D-Bus, qui, à son tour, réclame Syslog.

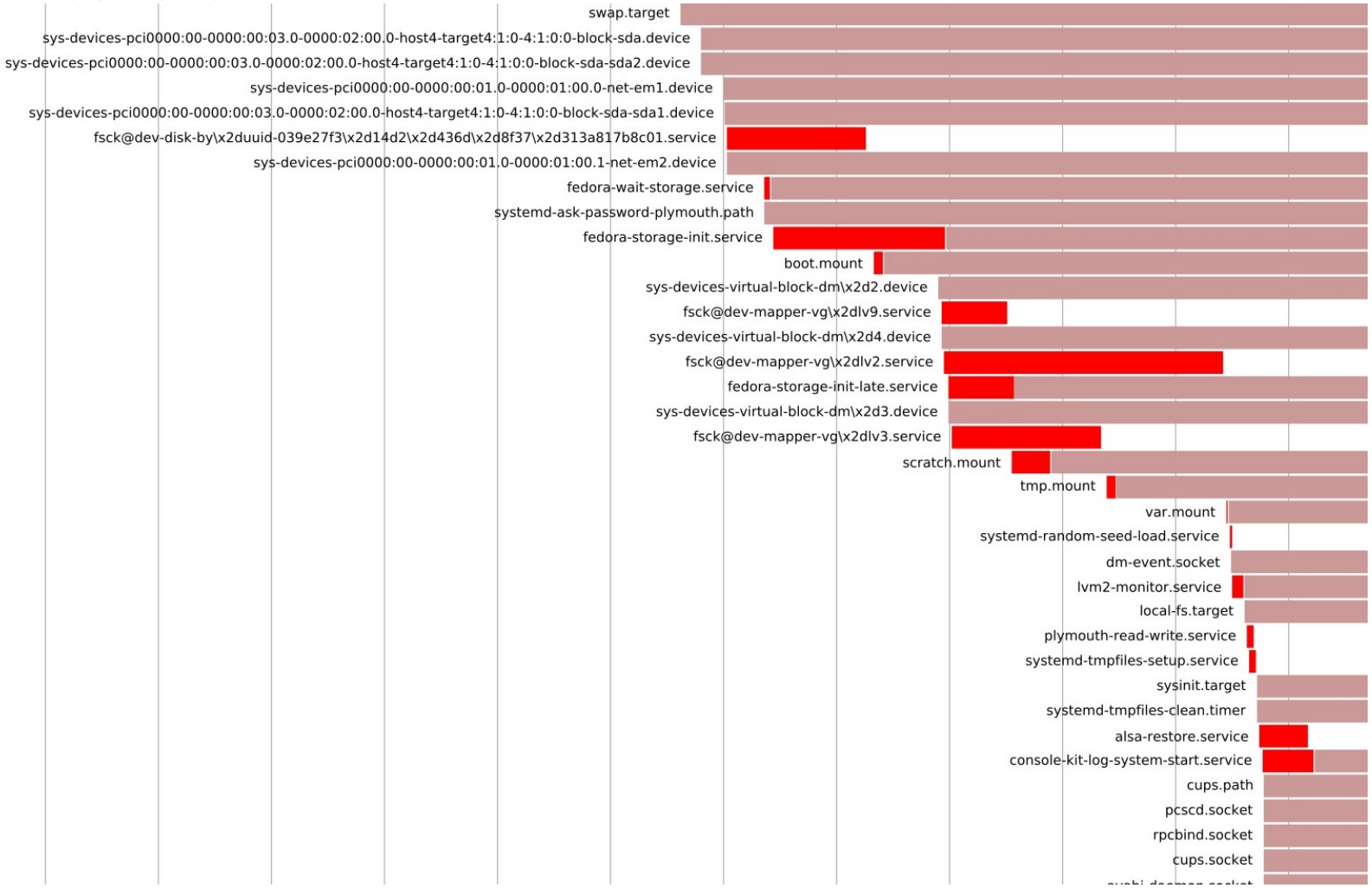


Parallelisation : carte du boot d'une fedora 17



Parallelisation : carte du boot d'une fedora 17

en zoomant :



Les composants de systemd

- Le moteur : `systemd`
- La tour de contrôle : `systemctl`
- Différents outils de monitoring : `systemd-analyze`, `systemd-cgls`...
- Un répertoire `/usr/lib/systemd`, contenant des fichiers de configuration. Chaque fichier décrit une unité et a un nom de la forme `sshd.service`, `graphical.target`, `cups.socket`, etc, suivant son type.
- Un répertoire `/etc/systemd`, contenant les configurations modifiées par l'administrateur. Les fichiers de ce répertoire sont prééminents sur ceux de `/usr/lib/systemd`.

Différents types d'unités

- `service` : lancement et administration de processus analogues à ceux de system V.
- `socket` : lancement d'une socket de type unix ou réseau, associée à un service.
- `target` : groupe fonctionnel de services et sockets. Généralise la notion de niveau d'éxécution.
- `mount` : gestion des montages, créés à la volée après lecture de `/etc/fstab`.
- `automount` : auto-montage par autofs.
- mais aussi : `path`, `timer`, `snapshot`, `swap`.

La syntaxe des fichiers service

- Syntaxe déclarative inspirée de celle des fichiers `.desktop`
- Différents groupes fonctionnels : `[Unit]`, `[Service]`, `[Install]`

/usr/lib/systemd/system/sshd.service

```
[Unit]
Description=OpenSSH server daemon
After=syslog.target network.target auditd.service

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStartPre=/usr/sbin/sshd-keygen
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target
```

La syntaxe des fichiers service

[Unit] contient la description du service, et la gestion des interactions avec les autres services

/usr/lib/systemd/system/sshd.service

```
[Unit]
Description=OpenSSH server daemon
After=syslog.target network.target auditd.service

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStartPre=/usr/sbin/sshd-keygen
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target
```

La syntaxe des fichiers service

[Service] décrit les commandes lancées par le service, et éventuellement celles pour le relancer, le tuer...

/usr/lib/systemd/system/sshd.service

```
[Unit]
Description=OpenSSH server daemon
After=syslog.target network.target auditd.service

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStartPre=/usr/sbin/sshd-keygen
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target
```

La syntaxe des fichiers service

`[Install]` décrit comment il sera installé par une commande `systemctl enable sshd.service`

/usr/lib/systemd/system/sshd.service

```
[Unit]
Description=OpenSSH server daemon
After=syslog.target network.target auditd.service

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStartPre=/usr/sbin/sshd-keygen
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target
```

Un fichier .socket

Ce fichier vient en complément du fichier `cups.service`.
Il lance une socket unix, une socket TCP et une socket UDP, écoutant toutes les deux sur le port 631.
S'il est installé, il sera activé par la cible socket

/usr/lib/systemd/system/cups.socket

```
[Unit]
Description=CUPS Printing Service Sockets

[Socket]
ListenStream=/var/run/cups/cups.socket
ListenStream=631
ListenDatagram=0.0.0.0 :631
BindIPv6Only=ipv6-only

[Install]
WantedBy=sockets.target
```

Un fichier .socket

Ce fichier vient en complément du fichier `cups.service`.
Il lance une socket unix, une socket TCP et une socket UDP, écoutant toutes les deux sur le port 631.
S'il est installé, il sera activé par la cible socket

/usr/lib/systemd/system/cups.socket

```
[Unit]
Description=CUPS Printing Service Sockets

[Socket]
ListenStream=/var/run/cups/cups.socket
ListenStream=631
ListenDatagram=0.0.0.0 :631
BindIPv6Only=ipv6-only

[Install]
WantedBy=sockets.target
```

Un fichier .socket

Ce fichier vient en complément du fichier `cups.service`.
Il lance une socket unix, une socket TCP et une socket UDP, écoutant toutes les deux sur le port 631.
S'il est installé, il sera activé par la cible socket

/usr/lib/systemd/system/cups.socket

```
[Unit]
Description=CUPS Printing Service Sockets

[Socket]
ListenStream=/var/run/cups/cups.socket
ListenStream=631
ListenDatagram=0.0.0.0 :631
BindIPv6Only=ipv6-only

[Install]
WantedBy=sockets.target
```

Un fichier .socket

Ce fichier vient en complément du fichier `cups.service`.
Il lance une socket unix, une socket TCP et une socket UDP, écoutant toutes les deux sur le port 631.
S'il est installé, il sera activé par la cible socket

/usr/lib/systemd/system/cups.socket

```
[Unit]
Description=CUPS Printing Service Sockets

[Socket]
ListenStream=/var/run/cups/cups.socket
ListenStream=631
ListenDatagram=0.0.0.0 :631
BindIPv6Only=ipv6-only

[Install]
WantedBy=sockets.target
```

Les cibles

Une cible (target) est un groupe fonctionnel d'unités.
La cible graphical lance une interface graphique (runlevel 5).

/usr/lib/systemd/system/graphical.target

```
[Unit]
Description=Graphical Interface
Documentation=man :systemd.special(7)
Requires=multi-user.target
After=multi-user.target
Conflicts=rescue.target
AllowIsolate=yes

[Install]
Alias=default.target
```

Les cibles

Une cible (target) est un groupe fonctionnel d'unités.
La cible graphical lance une interface graphique (runlevel 5).

/usr/lib/systemd/system/graphical.target

```
[Unit]
Description=Graphical Interface
Documentation=man :systemd.special(7)
Requires=multi-user.target
After=multi-user.target
Conflicts=rescue.target
AllowIsolate=yes

[Install]
Alias=default.target
```

`graphical.target` requiert la cible `multi-user.target` et démarrera après que `multi-user` ait achevé son démarrage.

Les cibles

Une cible (target) est un groupe fonctionnel d'unités.
La cible graphical lance une interface graphique (runlevel 5).

/usr/lib/systemd/system/graphical.target

```
[Unit]
Description=Graphical Interface
Documentation=man :systemd.special(7)
Requires=multi-user.target
After=multi-user.target
Conflicts=rescue.target
AllowIsolate=yes

[Install]
Alias=default.target
```

`graphical.target` ne peut pas être activée en même temps que la cible `rescue.target`.

Les cibles

Une cible (target) est un groupe fonctionnel d'unités.
La cible graphical lance une interface graphique (runlevel 5).

/usr/lib/systemd/system/graphical.target

```
[Unit]
Description=Graphical Interface
Documentation=man :systemd.special(7)
Requires=multi-user.target
After=multi-user.target
Conflicts=rescue.target
AllowIsolate=yes

[Install]
Alias=default.target
```

La commande `initctl isolate graphical.target` passera la machine en mode graphique (le `init 5` du bon vieux temps)

Les cibles

Une cible (target) est un groupe fonctionnel d'unités.
La cible graphical lance une interface graphique (runlevel 5).

/usr/lib/systemd/system/graphical.target

```
[Unit]
Description=Graphical Interface
Documentation=man :systemd.special(7)
Requires=multi-user.target
After=multi-user.target
Conflicts=rescue.target
AllowIsolate=yes

[Install]
Alias=default.target
```

La commande `initctl enable graphical.target` effectuera un lien symbolique vers `/etc/systemd/default.target`.

Que lance la cible /etc/systemd/default.target ?

- C'est un lien vers /usr/lib/systemd/system/graphical.target
- La cible graphical est associée à deux répertoires

/usr/lib/systemd/system/graphical.target.wants

```
display-manager.service -> /usr/lib/systemd/system/display-manager.service
```

/usr/lib/systemd/system/graphical.target.wants

```
system-setup-keyboard.service -> /usr/lib/systemd/system/system-setup-keyboard.service
```

- Elle requiert la cible multi-user qui a elle même ses répertoires .wants
- multi-user requiert à son tour la cible basic.target, qui à son tour ...

Que lance la cible /etc/systemd/default.target ?

systemctl show -p Wants multi-user.target

```
Wants=systemd-readahead-replay.service systemd-readahead-collect.service
systemd-update-utmp-runlevel.service system-setup-keyboard.service
prefdm.service spice-vdagentd.service jexec.service
```

systemctl show -p Wants multi-user.target

```
Wants=systemd-update-utmp-runlevel.service gpm.service sshd.service vsftpd.service
nslcd.service cups.path munin-node.service abrtd.service acpid.service
mcelog.service  autofs.service postfix.service mdmonitor.service ntpd.service
abrt-vmcore.service crond.service auditd.service smartd.service abrt-oops.service
ksmtuned.service libvirtd.service ksm.service rsyslog.service rpcbind.service
arp-ethers.service avahi-daemon.service irqbalance.service nfs-lock.service
atd.service remote-fs.target systemd-logind.service plymouth-quit-wait.service
systemd-user-sessions.service getty.target systemd-ask-password-wall.path
dbus.service plymouth-quit.service cleantmp.service tcsd.service
neatx.service network.service iscsi.service iscsid.service
```

etc...

Quelques cibles importantes

Pour plus de détails : `man systemd.special`

- `basic.target`
- `local-fs.target` `remote-fs.target`
- `socket.target`
- `network.target`
- `runlevel0.target` : lien vers `halt.target`
- `runlevel1.target` : lien vers `rescue.target`
- `runlevel3.target` : lien vers `multi-user.target`
- `runlevel5.target` : lien vers `graphical.target`
- `halt.target`
- `ctrl-alt-del.target` : lien vers `reboot.target`

Ecrire un service simple

Je veux mettre en place un démon `nbd`, en compilant le binaire.

- Je crée un fichier

/etc/systemd/system/nbd-server.service

```
[Unit]
Description=Serveur NBD
After=network.target

[Service]
ExecStart=/usr/local/bin/nbd-server

[Install]
WantedBy=multi-user.target
```

- Je fais relire la configuration de systemd et j'active le démon

```
# systemctl daemon-reload
# systemctl enable nbd-server.service
ln -s '/etc/systemd/system/nbd-server.service'
'/etc/systemd/system/multi-user.target.wants/nbd-server.service'
```

- Je démarre éventuellement le démon tout de suite

```
# systemctl start nbd-server.service
```

Et pour un service moins simple ?

Le passage de scripts system V à un système de services natifs pour systemd est manifestement un travail de grande ampleur. Si on regarde la liste des fonctionnalités de différentes versions de Fedora :

- `fedoraproject.org/wiki/Releases/16/FeatureList`

30%	SysV to Systemd	Porting from sysVinit init scripts to systemd unit file.	2011-05-18

- `fedoraproject.org/wiki/Releases/17/FeatureList`

40%	SysV to Systemd	Porting from sysVinit init scripts to systemd unit files.	2011-11-04

- `fedoraproject.org/wiki/Releases/18/FeatureList`

60%	SysV to Systemd	Porting from sysVinit init scripts to systemd unit files.	2012-06-05

Autopsie d'un boot - 1

```
Welcome to Fedora 17 (Beefy Miracle)

         Starting Replay Read-Ahead Data...
         Starting Collect Read-Ahead Data...
[OK] Reached target Login Prompts.
[OK] Listening on Syslog Socket.
[OK] Listening on /dev/initctl Compatibility Named Pipe.
[OK] Listening on Delayed Shutdown Socket.
[OK] Reached target Encrypted Volumes.
[OK] Set up automount Arbitrary Executable File Formats File System Automount Point.
         Expecting device dev-mapper-vg/x2dlv1.device...
[OK] Listening on udev Kernel Socket.
[OK] Listening on udev Control Socket.
         Expecting device dev-disk-by/x2duuid-039e27f3/x2d14d2.device...
         Expecting device dev-mapper-vg/x2dlv9.device...
[OK] Listening on Journal Socket.
[OK] Reached target Syslog.
         Starting Configure read-only root support...
         Mounting Media Directory...
         Starting Software RAID Monitor Takeover...
         Mounting Debug File System...
         Mounting POSIX Message Queue File System...
         Starting Journal Service...
[OK] Started Journal Service.
         Mounting Huge Pages File System...
         Starting udev Kernel Device Manager...
         Starting udev Coldplug all Devices...
[OK] Started Collect Read-Ahead Data.
[OK] Started Replay Read-Ahead Data.
```

Autopsie d'un boot - 2

```
[OK] Started Software RAID Monitor Takeover.
         Starting Load legacy module configuration...
         Starting Remount Root and Kernel File Systems...
         Starting Set Up Additional Binary Formats...
         Starting Apply Kernel Variables...
         Mounting Configuration File System...
         Starting Setup Virtual Console...
         Mounting Arbitrary Executable File Formats File System...
[OK] Started Apply Kernel Variables.
[OK] Mounted Media Directory.
[OK] Mounted Debug File System.
[OK] Mounted POSIX Message Queue File System.
[OK] Mounted Huge Pages File System.
[OK] Mounted Configuration File System.
[OK] Started udev Coldplug all Devices.
         Starting udev Wait for Complete Device Initialization...
[OK] Started Remount Root and Kernel File Systems.
[OK] Reached target Local File Systems (Pre).
[OK] Started udev Kernel Device Manager.
[OK] Started Configure read-only root support.
[OK] Started Setup Virtual Console.
[OK] Mounted Arbitrary Executable File Formats File System.
[OK] Started Set Up Additional Binary Formats.
[OK] Started Load legacy module configuration.
[OK] Found device /dev/mapper/vg-lv1.
         Activating swap /dev/mapper/vg-lv1...
[OK] Activated swap /dev/mapper/vg-lv1.
[OK] Reached target Swap.
```

Autopsie d'un boot - 3

```
[OK] Found device VIRTUAL_DISK.
         Starting File System Check on /dev/disk/by-uuid/039e27f3-14d2-436d-8f37-313a817b8c01...
[OK] Started udev Wait for Complete Device Initialization.
         Starting Wait for storage scan...
[OK] Started Wait for storage scan.
         Starting Initialize storage subsystems (RAID, LVM, etc.)...
systemd-fsck[667] : /dev/sda1 : recovering journal
systemd-fsck[667] : /dev/sda1 : clean, 352/51200 files, 96069/204800 blocks
[OK] Started File System Check on /dev/disk/by-uuid/039e27f3-14d2-436d-8f37-313a817b8c01.
         Mounting /boot...
[OK] Mounted /boot.
[OK] Found device /dev/mapper/vg-lv9.
         Starting File System Check on /dev/mapper/vg-lv9...
[OK] Started Initialize storage subsystems (RAID, LVM, etc.).
         Starting Initialize storage subsystems (RAID, LVM, etc.)...
systemd-fsck[713] : /dev/mapper/vg-lv9 : recovering journal
systemd-fsck[713] : /dev/mapper/vg-lv9 : clean, 11/1602496 files, 144597/6400000 blocks
[OK] Started File System Check on /dev/mapper/vg-lv9.
         Mounting /scratch...
[OK] Mounted /scratch.
         Starting Load Random Seed...
[OK] Listening on Device-mapper event daemon FIFOs.
         Starting Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling...
[OK] Started Load Random Seed.
[OK] Started Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling.
[OK] Reached target Local File Systems.
         Starting Tell Plymouth To Write Out Runtime Data...
         Starting Recreate Volatile Files and Directories...
[OK] Started Tell Plymouth To Write Out Runtime Data.
[OK] Reached target System Initialization.
```

Autopsie d'un boot - 4

```
[OK] Listening on PC/SC Smart Card Daemon Activation Socket.
[OK] Listening on RPCbind Server Activation Socket.
[OK] Listening on CUPS Printing Service Sockets.
[OK] Listening on Avahi mDNS/DNS-SD Stack Activation Socket.
[OK] Listening on D-Bus System Message Bus Socket.
[OK] Reached target Sockets.
[OK] Reached target Basic System.
         Starting SYSV : cleantmp : clean /tmp and /var/tmp...
         Starting IPv4 firewall with iptables...
         Starting IPv6 firewall with ip6tables...
         Starting Console Mouse manager...
         Starting NTP client/server...
         Starting Kernel Samepage Merging...
         Starting Login Service...
         Starting Avahi mDNS/DNS-SD Stack...
         Starting System Logging Service...
[OK] Started System Logging Service.
         Starting D-Bus System Message Bus...
[OK] Started Restore Sound Card State.
[OK] Started Console System Startup Logging.
[OK] Started IPv4 firewall with iptables.
[OK] Started IPv6 firewall with ip6tables.
[OK] Started Console Mouse manager.
[OK] Started ACPI Event Daemon.
[OK] Started Machine Check Exception Logging Daemon.
[OK] Started SYSV : cleantmp : clean /tmp and /var/tmp.
[OK] Started Kernel Samepage Merging.
[OK] Started D-Bus System Message Bus.
[OK] Started NTP client/server.
[OK] Reached target Network Time Protocol.
```

Autopsie d'un boot - 5

```
         Starting OpenSSH server daemon...
         Starting Vsftpd ftp daemon...
         Starting Naming services LDAP client daemon....
         Starting Munin Node Server....
         Starting Automounts filesystems on demand...
         Starting Postfix Mail Transport Agent...
         Starting RPC bind service...
[OK] Started OpenSSH server daemon.
[OK] Started Vsftpd ftp daemon.
[OK] Started Naming services LDAP client daemon..
[OK] Started Automounts filesystems on demand.
[OK] Started RPC bind service.
         Starting NFS file locking service....
         Starting SYSV : Creates /tmp/.X11-unix/ if required...
[OK] Started SYSV : Creates /tmp/.X11-unix/ if required.
[OK] Started NFS file locking service..
[OK] Reached target Remote File Systems (Pre).
         Mounting /users...
[OK] Started Munin Node Server..
[OK] Started Postfix Mail Transport Agent.
[OK] Mounted /users.
[OK] Reached target Remote File Systems.
         Starting Permit User Sessions...
[OK] Started Permit User Sessions.
         Starting Job spooling tools...
[OK] Started Job spooling tools.
         Starting Wait for Plymouth Boot Screen to Quit...
         Starting Display Manager...
[OK] Started Display Manager.
```

Utilisation des cgroups

- Un système d'init qui se respecte doit pouvoir tuer les processus qu'il a lancés.
- Mais par exemple, un cgi lancé par apache peut faire un double fork (lancer une nouvelle instance de lui même et se tuer). Dans ce cas, tuer apache ne permet pas de tuer cette instance du cgi.
- Le noyau linux fournit le mécanisme des control groups (`cgroups`), que systemd utilise à plein.
- Les cgroups sont stockés comme un système de fichiers usuel.

/sys/fs/cgroup/systemd

```
# ls /sys/fs/cgroup/systemd/system/
accounts-daemon.service    dbus.service          prefdm.service
avahi-daemon.service       fsck@.service         rsyslog.servicegpm.service
cgroup.clone_children      libvirtd.service      systemd-journald.service
cgroup.event_control       mcelog.service        systemd-logind.service
cgroup.procs               munin-node.service    system-setup-keyboard.service
chronyd.service            notify_on_release     tasks
```

Utilisation des cgroups

Affichons ici un certain nombre de processus, avec leurs cgroup

- à l'aide de la commande ps

ps axwf -eo pid,cgroup,args

```
  PID CGROUP                        COMMAND
    2 -                             [kthreadd]
    3 -                              _ [ksoftirqd/0]
 8202 -                              _ [kworker/0 :1]
    1 name=systemd :/system          /usr/lib/systemd/systemd
  297 cpuacct,cpu :/system/udev.se /usr/lib/udev/udevd
  458 cpuacct,cpu :/system/udev.se  _ /usr/lib/udev/udevd
  630 cpuacct,cpu :/system/rsyslog /sbin/rsyslogd -n -c 5
 6031 cpuacct,cpu :/system/prefdm.       _ /usr/bin/Xorg :0 -br -verbose
-auth /var/run/gdm/auth-for-gdm-8ve6Rc/database -seat seat0 -nolisten tcp
 6093 name=systemd :/user/fd/14          _ gdm-session-worker [pam/gdm-password]
 6115 name=systemd :/user/fd/14              _ gnome-session
 6302 name=systemd :/user/fd/14                  _ /usr/libexec/gnome-settings-daemon
 6334 name=systemd :/user/fd/14                  _ /usr/bin/gnome-shell
  776 cpuacct,cpu :/system/sshd.se /usr/sbin/sshd -D
  902 name=systemd :/system/rtkit- /usr/libexec/rtkit-daemon
```

Utilisation des cgroups

Affichons ici un certain nombre de processus, avec leurs cgroup.

- à l'aide de l'outil `systemd-cgls` de systemd

systemd-cgls

```
├ user
│ └ fd
│   └ 14
│     ├ 6093 gdm-session-worker [pam/gdm-password]
│     ├ 6111 /usr/bin/gnome-keyring-daemon --daemonize --login
│     ├ 6115 gnome-session
│     ├ 6302 /usr/libexec/gnome-settings-daemon
└ system
  ├ 1 /usr/lib/systemd/systemd
  ├ rtkit-daemon.service
  │ └ 902 /usr/libexec/rtkit-daemon
  ├ prefdm.service
  │ ├  645 /usr/sbin/gdm-binary -nodaemon
  │ ├ 6031 /usr/bin/Xorg :0 -br -verbose -auth /var/run/gdm/auth-for-gdm-8v...
  │ └ 6093 gdm-session-worker [pam/gdm-password]
  ├ rsyslog.service
  │ └ 630 /sbin/rsyslogd -n -c 5
  └ udev.service
    ├  297 /usr/lib/udev/udevd
    ├  458 /usr/lib/udev/udevd
```

Arrêt d'un service

- La commande

```
systemctl stop httpd.service
```

arrête le serveur apache et lui fait confiance pour se terminer proprement.

- La commande

```
systemctl kill -s SIGKILL httpd.service
```

envoie le signal `SIGKILL` au serveur apache, ainsi qu'à tous les processus qui sont dans son cgroup.

Lancer un service dans un chroot

Systemd gère nativement les chroot :

/usr/bin/systemd/system/foobar.service

```
[Unit]
Description=A chroot()ed Service

[Service]
RootDirectory=/srv/chroot/foobar
ExecStartPre=/usr/local/bin/setup-foobar-chroot.sh
ExecStart=/usr/bin/foobard
RootDirectoryStartOnly=yes
```

- L'éxécutable est lancé dans le chroot `/srv/chroot/foobar`, et le chemin `/usr/bin/foobard` est relatif à ce répertoire.
- Le script `setup-foobar-chroot.sh` sert à configurer le chroot (montage de `/proc`,...) avant de lancer `foobard`.

Utiliser les espaces de nommage

Systemd peut utiliser la fonctionnalité d'isolation par espaces de nommage du noyau linux pour restreindre les accès d'un service, sans faire un chroot :

/usr/bin/systemd/system/foobar.service

```
[Unit]
Description=Un service a domaine restreint

[Service]
ExecStart=/usr/bin/foobard
InaccessibleDirectories=/home
ReadOnlyDirectories=/etc
```

Utiliser les espaces de nommage

Les espaces de nommage peuvent être utilisés pour mettre en place des containers, suivant le principe de LXC, mais en plus élémentaire, et en plus limité.
Voici deux exemples extraits de la page de manuel de `systemd-nspawn` :

lancer une fedora 16 dans une fedora 17

```
# yum --releasever=16 --nogpgcheck --installroot ~/fedora-tree/
    install yum passwd vim-minimal rootfiles systemd
# systemd-nspawn -D ~/fedora-tree /usr/lib/systemd/systemd
```

lancer une debian dans une fedora 17

```
# debootstrap --arch=amd64 unstable ~/debian-tree/
# systemd-nspawn -D ~/debian-tree/
```

Système de Logs

- Systemd a son propre système de logs, lancé par le service `systemd-journald.service`. Il peut fonctionner indépendemment ou non de `syslog`.
- Dans le fichier de configuration `/etc/systemd/systemd-journald.conf`, on peut ainsi renseigner la ligne

```
ForwardToSyslog=yes
```

- On peut consulter les logs grâce à `journalctl` :

```
# journalctl _SYSTEMD_UNIT=sshd.service
Logs begin at Fri, 14 Sep 2012 11 :05 :09 +0200, end at Mon, 24 Sep 2012 09 :00 :01 +0200.
Sep 14 11 :05 :21 sganarelle.math sshd[1335] : Server listening on 0.0.0.0 port 22.
Sep 14 11 :05 :21 sganarelle.math sshd[1335] : Server listening on : : port 22.
```

```
# journalctl /usr/sbin/vsftpd
Logs begin at Fri, 14 Sep 2012 11 :05 :09 +0200, end at Mon, 24 Sep 2012 09 :00 :01 +0200.
```

L'outil de contrôle Systemctl

- Démarrer/arréter un service

```
# systemctl start/stop sshd.service
```

- Activer/désactiver un service

```
# systemctl enable sshd.service
ln -s '/usr/lib/systemd/system/sshd.service' '/etc/systemd/system/multi-user.target.wants/sshd.service'
# systemctl disable sshd.service
rm '/etc/systemd/system/multi-user.target.wants/sshd.service'
```

- Masquer un service : de cette façon, le service ne sera jamais lancé par systemd, même en suivant des dépendances.

```
# systemctl mask sshd.service
ln -s '/dev/null' '/etc/systemd/system/sshd.service'
```

L'outil de contrôle Systemctl

- Lister les services

```
# systemctl --type=service --all
UNIT                        LOAD   ACTIVE   SUB     JOB DESCRIPTION
abrt-oops.service           error  inactive dead        abrt-oops.service
accounts-daemon.service     loaded active   running     Accounts Service
alsa-restore.service        loaded inactive dead        Restore Sound Card State
bluetooth.service           loaded failed   failed      Bluetooth Manager
```

- Examiner un service

```
# systemctl status sshd.service
sshd.service - OpenSSH server daemon
   Loaded : loaded (/usr/lib/systemd/system/sshd.service ; enabled)
   Active : active (running) since Fri, 14 Sep 2012 11 :05 :21 +0200 ; 3 weeks and 3 days ago
   Main PID : 1335 (sshd)
   CGroup : name=systemd :/system/sshd.service
            | 1335 /usr/sbin/sshd -D
```

- Changer de un niveau d'activité

```
# systemctl isolate rescue.target
```

Pour conclure

Il aurait suffi de lire jusqu'au bout dès le début !

/etc/inittab

```
# inittab is no longer used when using systemd.
#
# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# Ctrl-Alt-Delete is handled by /etc/systemd/system/ctrl-alt-del.target
#
# systemd uses 'targets' instead of runlevels. By default, there are two main targets :
#
# multi-user.target : analogous to runlevel 3
# graphical.target : analogous to runlevel 5
#
# To set a default target, run :
#
# ln -s /lib/systemd/system/<target name>.target /etc/systemd/system/default.target
#
```

Mais alors, je n'aurais pas fait d'exposé !