

Upstart

... das andere `/sbin/init`

der Status Quo

- (Linux) System wird nach SysV oder BSD gestartet
- /sbin/init übernimmt die Kontrolle mit PID 1
- Runlevel entscheidet über die laufenden Prozesse
- Startscripte werden sequenziell abgearbeitet

flashback

Runlevel:

0 = halt

S = startup

1 = single user mode

2 = multiuser w/o
network

3 = multiuser + network

4 = reserved

5 = X11

6 = reboot

- sysV startet von S in default RL

- bsd geht Runlevel durch bis default RL

Upstart und Ubuntu

Klassischer Ansatz "Kernel kennt die Hardware zur Bootzeit" entspricht nicht mehr dem realen Umfeld

- Hotplugging von
 - Massenspeichern
 - Netzwerkadaptoren
- Dateisysteme ebendort
- Services müssen nicht laufen, wenn HW nicht verfügbar (CUPS)

was Upstart anders macht

- kein Runlevel-Konzept
- Steuerung durch Events
- Scripts ("Jobs") können parallel starten
- Upstart gibt Auskunft über den Status der "Jobs" und behält sie im Auge

Jobs

in `/etc/event.d/` werden Jobs definiert

Jobs können:

- Daemons
 - starten (optional: pre-start scripts)
 - stoppen (optional: post-stop scripts)
 - automatisch neustarten (respawn)
- Events aussenden
- beliebige Aktionen ausführen

Job Definition

Stanzas

- description
- author
- start on
- stop on
- respawn
- emits
- daemon
- instance
- limit
- nice
- normal
- exec
- script
- pre start
- post start
- pre stop
- post stop
- console
- env
- umask
- chroot
- chdir

Job Definition

Jobs müssen `script` oder `exec` Stanza enthalten

Dateien dürfen nicht ausführbar sein

Kommentare beginnen mit `#`

Whitespace wird ignoriert, ausser in `'''` oder `''`

```
# hello world
```

```
description "simple job"  
author "Goesta Smekal"
```

```
start on hello
```

```
console output
```

```
script  
    echo "Hello world!"  
end Script
```

start stop respawn

Kontrolliert, wann ein Job gestartet oder gestoppt wird, oder ob er automatisch wieder starten soll.

`start` und `stop` verlangen einen Event als Parameter

`respawn` kann mit einem `limit` und `timeout` versehen werden, wie oft der Neustart versucht werden soll

daemon instance

- `daemon` bewirkt, dass der Job in den Hintergrund forkt
- `instance` Jobs sind immer in waiting und erzeugen weitere Instanzen von sich selbst

exec script

Was der Job tun soll bestimmen `exec`, bzw `script`
`exec` kann genau ein Kommando mit Parametern ausführen

`script` umschließt eine Reihe von Befehlen, die wie von `/bin/sh` interpretiert werden

```
script
  your shellscript here
end script
```

pre/post start/stop

Werden verwendet um den Start vorzubereiten, bzw nach dem Stop aufzuräumen

```
pre-start script
    mkdir /var/run/mydaemon
end pre-start script
post-stop script
    rm -rf /var/run/mydaemon
end post-stop script
```

nice kill normal

- `nice` bestimmt den Nicelevel
- `kill` setzt die Wartezeit zwischen TERM und KILL Signal
- `normal` listet alle exitcodes, die normalen Abschluss bedeuten

console env

`console` bestimmt was mit input und output geschehen soll:

- `logged`: stdout geht an den Logger
- `output`: stdout geht an aktuellen STDOUT
- `owner`: Prozess wird Eigner der Konsole, kann auch Keyboardinput empfangen
- `none`: `/dev/null`

`env` setzt Umgebungsvariablen

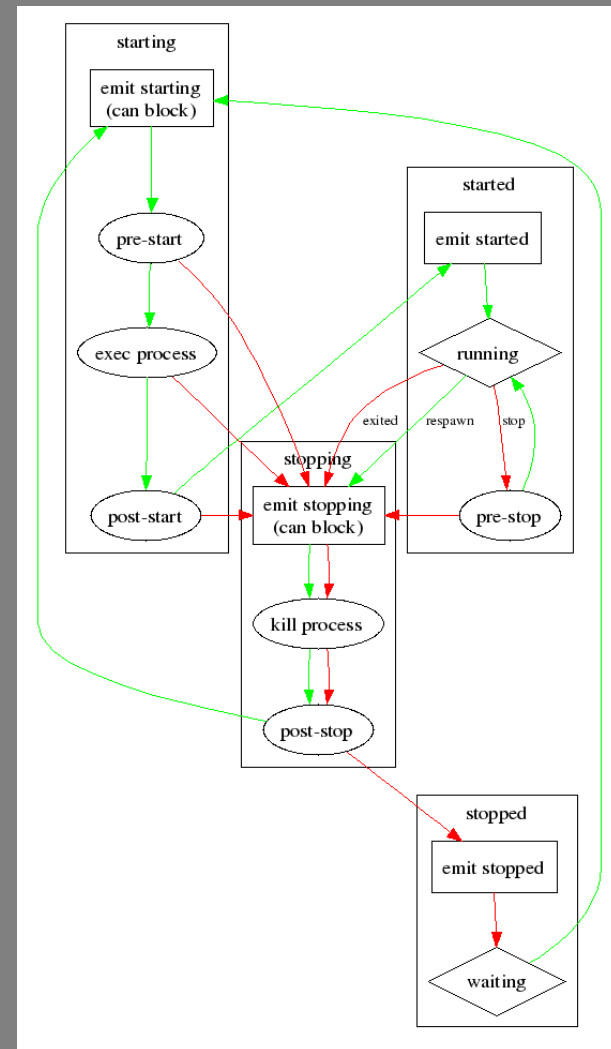
Job Events

Jeder Job sendet automatisch Events aus:

- starting
- started
- stopping
- stopped

andere Jobs können sich darauf beziehen:

start on started
network



Commandline Tools

- Jobcontrol

- start
- stop
- status

- Events

- initctl emit
- initctl events

- Status

Stand der Dinge

Derzeit gibt es upstart bei Ubuntu (ab Gutsy Gibbon) und Debian (experimental)

Beide Distributionen nutzen aber die Stärken von Upstart nicht, sondern haben sysv-compatibility

Howtos für Gentoo, Fedora, Suse und Frugalware

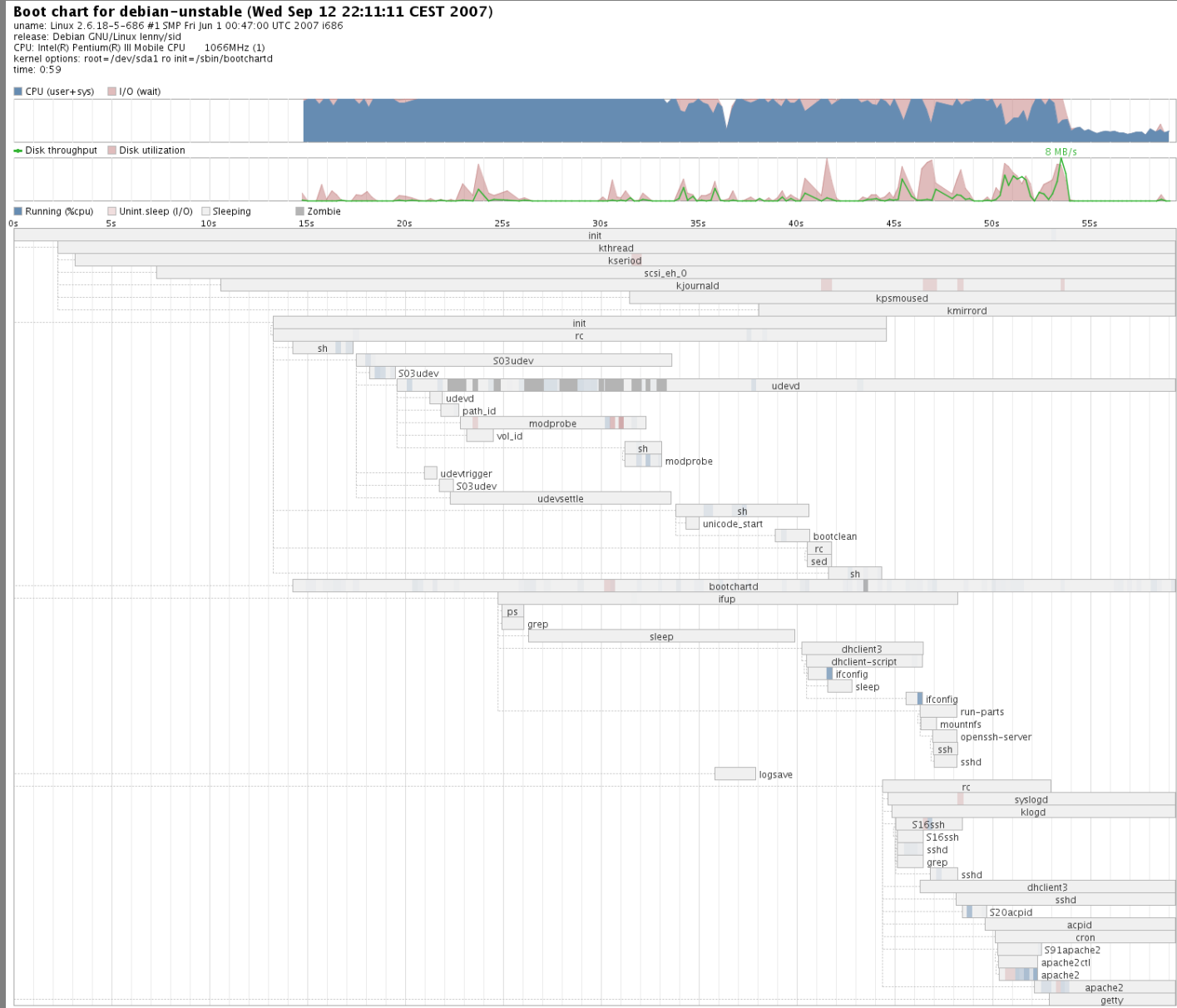
Vergleich

`bootchart` kann Systemstart verfolgen und graphisch auswerten

Selbst bei SysV compatibility ergibt sich ein kleiner Vorteil in der Geschwindigkeit

bootchart sysV

t=59sec

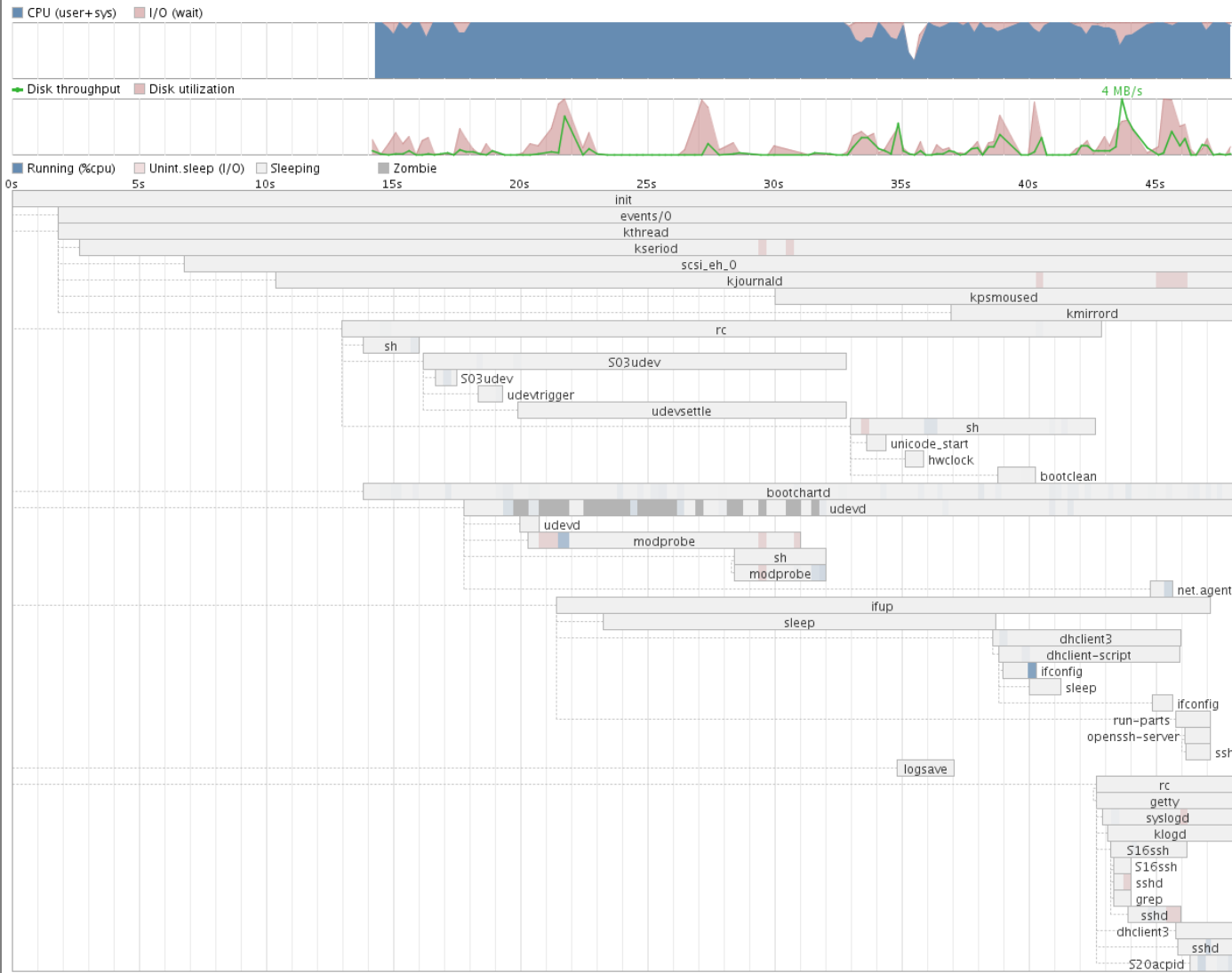


bootchart upstart

t=48sec

Boot chart for debian-unstable (Wed Sep 12 22:31:02 CEST 2007)

uname: Linux 2.6.18-5-686 #1 SMP Fri Jun 1 00:47:00 UTC 2007 i686
release: Debian GNU/Linux lenny/sid
CPU: Intel(R) Pentium(R) III Mobile CPU 1066MHz (1)
kernel options: root=/dev/sda1 ro init=/sbin/bootchartd
time: 0:48



geplante Features

- zeitgesteuerte Events (löst cron ab)
- Hardwarebasierte Events (Erweiterung/Kompliment zu hotplug)
- Events bei Datei- Verzeichnisänderung
- User Services (unprivilegierter Start)

initng

- Fokus liegt offenbar auf Beschleunigung durch asynchronen Start
- <http://www.initng.org>
- gestartet Mai 2006
- aktuell: v0.6.10.2 (11. Nov. 2007)
- läuft angeblich auf allen Mainstream Distros (.deb auf Website)
- 238 bekennende User, 5454 Hosts

launchd runit

launchd

- Apple Plattform
- vereinigt
 - init
 - cron
 - inetd
 - ...

runit

- kommt von Solaris
- Fokus liegt auf Servicemanagement

Quellen

- <http://upstart.ubuntu.com>
- `man init(8)`
- <http://www.linuxfromscratch.org>
- <http://tldp.org/HOWTO/From-PowerUp-To-Bash-Prompt-HOWTO.html>
- <http://www.linux.com/articles/57213>