

GUUG Frühjahrsfachgespräch 2019

SmartOS: OpenSolaris in den Wolken

Volker A. Brandt

Brandt & Brandt Computer GmbH

<http://www.bb-c.de/>

vab@bb-c.de

Karlsruher Institut für Technologie, 11.04.2019

Übersicht

- 1 Vorwort
- 2 Illumos
- 3 Joyent
- 4 SmartOS - Was ist das?
- 5 SmartOS - Installation
- 6 SmartOS - File-Server
- 7 SmartOS - iSCSI-Server
- 8 Ausblick
- 9 Links
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS

1 Vorwort

2 Illumos

3 Joyent

4 SmartOS - Was ist das?

5 SmartOS - Installation

6 SmartOS - File-Server

7 SmartOS - iSCSI-Server

8 Ausblick

9 Links

10 Backup 1:
RAM-Disk für
das
Root-Filesystem
größer machen

11 Backup 2:
Virtualisierung
mit SmartOS

Vorwort

Dieser Vortrag ist

- subjektiv...
- eine Momentaufnahme...
- unvollständig...
- eine Diskussionsanregung...
- eine Anknüpfung an die Bestandsaufnahmen vom FFG 2013, 2016, 2017...

...und er hätte besser „SmartOS angekettet im Bergwerk“ heißen sollen!

- A set of navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other slide controls.

Start von Illumos



- Idee: der letzte freie Source-Stand von ON “reicht fast” für ein komplettes OS
- man muß sich von Oracle abnabeln
- Gründer: Garrett D’Amore, ex-Sun und damals bei Nexenta
- Ankündigung am 3. August 2010

(ON = “Operating System and Network”, das Basis-Bündel der OpenSolaris-Source)

Bedeutung von Illumos

- keine Distribution, sondern Nachfolger und Ersatz von ON
- zunächst vom Ansatz her plattform-agnostisch (x86/x64 und SPARC werden gleich gut unterstützt)
- Diskussionen über die weitere SPARC-Unterstützung, aber inzwischen ist SPARC wieder gesetzt (“keeps us honest”)
- immer wieder Pläne für Ports nach ARM und Power, bisher ist nichts draus geworden

Illumos-Verfügbarkeit

- Der Illumos-Sourcetree wird per `git` zur Verfügung gestellt
- Github:
`https://github.com/illumos/illumos-gate`
- Der Inhalt von `gate` wird immer wieder diskutiert
- Es gibt auch noch `userland`, ist aber nicht so relevant...
- Der “Peer Review”-Prozeß wird gelebt

(Mercurial benutzt niemand mehr)

Illumos-Verfügbarkeit

Illumos bauen:

- der Bauprozess ist inzwischen nicht mehr so komplex; einzelne Distributionen haben das komplette Tooling zum Illumos-Bauen mit dabei (am besten ist OmniOS)
- der Compiler aus dem nicht (mehr) frei verfügbaren Solaris Studio verliert an Bedeutung
- `gcc` ist der Standard, LLVM/Clang kommt (erst als “shadow compiler”)
- man kann auch mit aktuelleren `gcc`-Versionen bauen
- `Smatch` (“The Source Matcher”) als `lint`-Ersatz

- 1 Vorwort
- 2 Illumos
- 3 Joyent**
- 4 SmartOS - Was ist das?
- 5 SmartOS - Installation
- 6 SmartOS - File-Server
- 7 SmartOS - iSCSI-Server
- 8 Ausblick
- 9 Links
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS

Joyent und SmartOS



- Joyent war schon Vorzeigekunde von Sun
- Cloud-Anbieter mit virtuellen Kunden-Servern
- außer Nexenta der einzige kommerzielle Player mit Gewicht (gehört jetzt zu Samsung)
- viele gute Ex-Sun-Engineers arbeiten dort
- Paketverwaltung über `pkgsrc` (bekannt aus NetBSD)
- strikt auf Hosting- und Cloud-Ansprüche getrimmt
- Support per Mailing-List und IRC (`#smartos`)

Entwicklung von SmartOS

- häufige Releases, schnelle Reaktion auf Probleme (CERT-Alerts)
- aktiv im Illumos-Review-Prozeß
- diverse ZFS-Weiterentwicklungen
- Virtualisierung: native Zones, KVM, LX brand, bhyve
- PR (Auftritt auf Konferenzen, Giveaways, ...)
- Hardcore-Enterprise-Cloud-Einsatz einer Illumos-basierten Umgebung

- 1 Vorwort
- 2 Illumos
- 3 Joyent
- 4 SmartOS - Was ist das?**
- 5 SmartOS - Installation
- 6 SmartOS - File-Server
- 7 SmartOS - iSCSI-Server
- 8 Ausblick
- 9 Links
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS

Was ist SmartOS?



- OpenSolaris-basierende Server-Betriebssystemdistribution
- Illumos-Distribution
- Teil von Triton, einer “RZ-Komplettlösung”
- frei herunterzuladen, frei zu benutzen, auch kommerziell
- Source gehostet auf Github (zumindest teilweise)
- verwendet Package Source (NetBSD)
- optimiert für Hosting und als VM-Träger
- nur für **x86/x64**
- Boot-Medium ist **read-only** (USB-Stick, CD-ROM, PXE)

Wozu ist es gut?

- kleine, mittlere, große gehostete Umgebungen
- Virtualisierungs-Server für alles und jedes
- Basis für Joyent Triton (“managed datacenter”)

Wozu taugt es nicht?

- Desktop
- Plattform für fertige (gekaufte) Software
- persistentes Arbeiten in der Global Zone ;-)

Verfügbarkeit

- Joyent baut alle zwei bis drei Wochen neue Boot-Images
- einfach USB-Stick aktualisieren und neu booten
- Netzwerk-Boot geht natürlich auch
- kostenpflichtiger Support über Triton/Hosting-Angebote
- sehr fleißiges Upstreamen von Änderungen

Was bekommt man?

```
-r--r--r-- 1 32530 Mar 14 05:39 changelog.txt
-r--r--r-- 1 424 Mar 14 05:39 md5sums.txt
-r--r--r-- 1 158827651 Mar 14 05:19 platform-20190314T022406Z.tgz
-r--r--r-- 1 17 Mar 14 05:39 SINGLE_USER_ROOT_PASSWORD.txt
-r--r--r-- 1 158494288 Mar 14 05:39 smartos-20190314T022406Z-USB.img.bz2
-r--r--r-- 1 311343104 Mar 14 04:32 smartos-20190314T022406Z.iso
-r--r--r-- 1 158330313 Mar 14 05:39 smartos-20190314T022406Z.vmwarevm.tar.bz2
```

```
# tar tzf platform-20190314T022406Z.tgz
platform-20190314T022406Z/
platform-20190314T022406Z/root.password
platform-20190314T022406Z/i86pc/
platform-20190314T022406Z/i86pc/amd64/
platform-20190314T022406Z/i86pc/amd64/boot_archive.manifest
platform-20190314T022406Z/i86pc/amd64/boot_archive
platform-20190314T022406Z/i86pc/amd64/boot_archive.hash
platform-20190314T022406Z/i86pc/amd64/boot_archive.gitstatus
platform-20190314T022406Z/i86pc/kernel/
platform-20190314T022406Z/i86pc/kernel/amd64/
platform-20190314T022406Z/i86pc/kernel/amd64/unix
```

- 1 Vorwort
- 2 Illumos
- 3 Joyent
- 4 SmartOS - Was ist das?
- 5 **SmartOS - Installation**
- 6 SmartOS - File-Server
- 7 SmartOS - iSCSI-Server
- 8 Ausblick
- 9 Links
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS

First Boot: Serielle Console mit dem USB-Stick

```
grub> variable os_console ttyb
```

Selected OS console device is 'ttyb'..

To change OS console device, enter command-line modee

and use 'variable os_console <dev>', then Esc to return..

Valid <dev> values are: ttya, ttyb, ttyc, ttzd, vga

(und dann noch das übliche (... -B console=ttyb, ...).

First Boot: Start

SmartOS Setup

Joyent

<https://wiki.smartos.org/install>

You must answer the following questions to configure your SmartOS node.
You will have a chance to review and correct your answers, as well as a
chance to edit the final configuration, before it is applied.

At the prompts, if you type ^C you will be placed into a shell. When you
exit the shell the configuration process will resume from where it was
interrupted.

Press [enter] to continue

First Boot: Network 1

SmartOS Setup

Networking

<https://wiki.smartos.org/install>

To set up networking you must first configure a network tag. A network tag refers to a physical NIC or an aggregation. Virtual machines will be created on top of a network tag. Setup will first create a network tag and configure a NIC so that you can access the SmartOS global zone. After setup has been completed, you will have the option of creating additional network tags and configuring additional NICs for accessing the global zone through the `nictagadm(1M)` command.

Press [enter] to continue

First Boot: Network 2

SmartOS Setup
Networking - Admin

<https://wiki.smartos.org/install>

The admin network is the primary network in SmartOS. It is the default network that is created. The configured NIC will be used to access the global zone. If you wish to use a VLAN on this network, you must configure VLAN ACCESS mode for this network.

Number	Link	MAC Address	State	Network
1	bge0	94:18:82:37:1a:dc	up	-
2	bge1	94:18:82:37:1a:dd	down	-

Enter the number of the NIC for the 'admin' interface: 1
(admin) IP address (or dhcp): 192.168.222.98
(admin) netmask [255.255.255.0]:

First Boot: Gateway und DNS

SmartOS Setup

Networking - Continued

<https://wiki.smartos.org/install>

The default gateway will determine which router will be used to connect the global zone to other networks. This will almost certainly be the router connected to your 'admin' network. Use 'none' if you have no gateway.

Enter the default gateway IP [none]: 192.168.222.254

The DNS servers set here will be used to provide name resolution abilities to the SmartOS global zone itself. These DNS servers are independent of anything you use to create virtual machines through vmadm(1M).

Enter the Primary DNS server IP [8.8.8.8]: 192.168.222.254

Checking connectivity...OK

Enter the Secondary DNS server IP [8.8.4.4]: 217.0.43.33

Checking connectivity...OK

Default DNS search domain: bb-c.de

By default the headnode acts as an NTP server for the admin network. You can set the headnode to be an NTP client to synchronize to another NTP server.

Enter an NTP server IP address or hostname [0.smartos.pool.ntp.org]: 192.168.222.254

Checking connectivity...OK

First Boot: Fallstricke – Netzwerk

Worauf sollte man achten:

- es gibt keine deutsche Tastatur! Mimimi!
- man **muß** zwei DNS-Server angeben
- nur eine “DNS search domain” angeben (es geht mit "double quotes", aber lieber später konfigurieren)
- bei Problemen mit dem Netzwerk: Die Abhängigkeiten des SMF-Dienstes `network/physical` sind anders; beispielsweise muß `smartdc/config` separat ge-“clear“-t werden

First Boot: zpool

SmartOS Setup

Storage

<https://wiki.smartos.org/install>

SmartOS will automatically determine what we think is the best zpool layout from your current disks. You may use this suggestion, change to another built in storage profile, or simply create your own zpool.

vdevs: c2t0d0 total capacity: 2794.52 GB

This is the 'default' storage configuration. To use it, type 'yes'.

To see a different configuration, type: 'raidz2', 'mirror', or 'default'.

To specify a manual configuration, type: 'manual'.

Selected zpool layout [yes]:

First Boot: Fallstricke – Storage

- Man kann den Wert für `ashift` des Pools `zones` hier nicht ändern
- ...im Zweifel 'default' wählen und später nachbessern (die ZFS-Kommandos sind ja alle da)

First Boot: Password und Hostname

```
SmartOS Setup
System Configuration                                https://wiki.smartos.org/install
```

```
Setup will now go through and prompt for final pieces of account configuration.
This includes setting the root password for the global zone and optionally
setting a hostname.
```

```
Enter root password:
```

```
Confirm password:
```

```
Enter system hostname [press enter for none]: basalt
```

Wir erinnern uns: Vorsicht mit Umlauten und Sonderzeichen!

First Boot: Zusammenfassung

SmartOS Setup

Verify Configuration

<https://wiki.smartos.org/install>

Please verify your SmartOS Configuration. After this point the system will set up and all data on the disks will be erased.

Net	MAC	IP addr.	Netmask
Admin 94:18:82:37:1a:dc		192.168.222.98	255.255.255.0

DNS Servers: (192.168.222.254, 217.0.43.33), Search Domain: bb-c.de

Hostname: basalt

NTP server: 192.168.222.254

Is this correct, proceed with installation? [y]:

Creating pool zones...

done

Making dump zvol...

dump on /dev/zvol/dsk/zones/dump size

done

Initializing config dataset for zones...

Creating config dataset...

done

Creating global cores dataset...

done

Creating opt dataset...

done

Initializing var dataset...

done

Creating swap zvol...

done

System setup has completed.

Press enter to reboot.

rebooting...

Nach dem Reboot...

```
# zpool list
NAME      SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP   HEALTH  ALTROOT
zones  10.9T   5.94T   4.97T        -         -       0%   54%  1.00x  ONLINE   -

# zfs list
NAME                                     USED   AVAIL   REFER  MOUNTPOINT
zones                                     5.97T   4.59T    804K    /zones
zones/config                             172K   4.59T    172K    legacy
zones/opt                               2.15G   4.59T   2.15G    legacy
zones/dump                              1.00G   4.59T   1.00G    -
zones/swap                              16.5G   4.61T   31.7M    -
zones/usbkey                             132K   4.59T    132K    legacy
zones/var                               132M   4.59T    132M    legacy
...

# mount
/ on /devices/ramdisk:a read/write/setuid/devices/intr/largefiles/xattr/...
/var on zones/var read/write/setuid/devices/nonbmand/exec/xattr/noatime/...
/etc/zones on zones/config read/write/setuid/devices/nonbmand/exec/xattr/...
/opt on zones/opt read/write/setuid/devices/nonbmand/exec/xattr/noatime/...
/usbkey on zones/usbkey read/write/setuid/devices/nonbmand/exec/xattr/...
/etc/ssh on /usbkey/ssh read/write/setuid/devices/dev=1690025 on Sun Apr ...
/usr on /devices/pseudo/lofi@1:disk read only/setuid/devices/intr/...
...

# lofiadm
Block Device      File      Options
/dev/lofi/1       /usr.lgz  Compressed(gzip)
```

Konfiguration

```
# cat /usbkey/config
#
# This file was auto-generated and must be source-able by bash.
#

admin_nic=94:18:82:37:1a:dc
admin_ip=192.168.222.98
admin_netmask=255.255.255.0
admin_network=
admin_gateway=192.168.222.98

headnode_default_gateway=192.168.222.254

dns_resolvers=192.168.222.254,217.0.43.33
dns_domain=bb-c.de

ntp_hosts=192.168.222.254
compute_node_ntp_hosts=192.168.222.98

hostname=basalt
```

Die Herausforderung: Das OS ist auf einem USB-Stick!

- das Betriebssystem ist im Wesentlichen read-only
- Das root-Filesystem ist eine RAM-Disk, ziemlich klein und hat wenig freien Platz
- `/usr` ist ein Loopback und read-only
- **nach einem Reboot ist alles weg, was man nicht selber persistiert hat**

SmartOS hat eine Art Nachkonfiguration für Dateien

```
# ls -l /usbkey/config.inc
total 9
-r--r--r-- 1 root      root          615 Jan 27 16:12 authorized_keys

# cat /usbkey/config
...
hostname=basalt

root_authorized_keys_file=authorized_keys
default_keymap=germany
```

Das reicht aber nicht!

Die Lösung: Persistenz in /opt/custom

SmartOS ist eigentlich so entworfen, daß in der globalen Zone nichts Persistentes verbleiben soll. Wie erstelle ich also Konfigurationsdateien in `/etc`, starte SMF-Dienste, mache `crontab`-Einträge usw. usw. usw.? Der Trick:

- SMF-Manifeste in `/opt/custom/smf` werden beim Booten eingelesen
- man baut damit eine Art Init-Dienst für arme, der alles Gewünschte nachstartet
- ich kann alles machen, was ich in einem Shell-Script ausdrücken kann

- 1 Vorwort
- 2 Illumos
- 3 Joyent
- 4 SmartOS - Was ist das?
- 5 SmartOS - Installation
- 6 SmartOS - File-Server**
- 7 SmartOS - iSCSI-Server
- 8 Ausblick
- 9 Links
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS

Fileserver: Was soll er denn können?

- NFS v4
- SMB (Samba, kein Kernel-CIFS)
- `rsync`
- ...

Fileserver: Die Hardware



Fileserver: Specs

- HP ProLiant MicroServer Gen8
- Intel(R) Celeron(R) CPU G1610T @ 2.30GHz
- 16 GB DDR3-RAM
- 4x HGST HDN726060ALE614 6TB SATA
- 2x Broadcom NetXtreme BCM5720

Dieser Microserver hat eine serielle Console!

Fileserver: Was brauchen wir?

- Netzwerk-Konfiguration, falls mehr als ein Interface vorhanden
- Samba-Software und Konfiguration
- `rsync`-Konfiguration
- Persistenz in der globalen Zone

(Die NFS-Konfiguration merkt sich ZFS freundlicherweise direkt im `zpool` und aktiviert sie beim Boot wieder.)

Fileserver: pkgsrc

Die benötigte Software (Samba, `rsync` und so weiter) liefert `pkgsrc`. Dafür stellt Joyent ein Bootstrap-Archiv bereit, das einfach heruntergeladen und ausgepackt werden kann:

```
# ARCHIVE=bootstrap-2019Q1-x86_64.tar.gz
# cd /var/tmp
# wget --no-check-certificate https://pkgsrc.joyent.com/packages/SmartOS/bootstrap/$ARCHIVE
# cd /
# tar xvfzfp /var/tmp/$ARCHIVE
```

Danach wird die Software aus dem `pkgsrc`-Repository heruntergeladen. Details: Siehe `pkg_add(1)`, `pkgin(1)`

SMF-Persistenz-Dienst: Manifest

```
# cat /opt/custom/smf/local-config.xml
...
<service_bundle type='manifest' name='local-config'>
  <service type='service' name='local/config'
    version='1.0.0'>
    <create_default_instance enabled='true' />
    <single_instance/>
  ...
  <exec_method type='method' name='start'
    timeout_seconds='300' exec='/opt/custom/local/config-service %m' />

  <exec_method type='method' name='refresh'
    timeout_seconds='300' exec='/opt/custom/local/config-service %m' />

  <exec_method type='method' name='restart'
    timeout_seconds='300' exec='/opt/custom/local/config-service %m' />

  <exec_method type='method' name='stop'
    timeout_seconds='1' exec=':true'>
  </exec_method>
  ...
</service_bundle>
```

SMF-Manifeste in anderen Verzeichnissen können mit symbolischen Links aktiviert werden.

SMF-Persistenz-Dienst: Start/Stop-Scripts

```
# cd /opt/custom/local ; ls -gol S* K*  
-r-xr-xr-x  1    1127 Feb 12  2018 K20_rsyncd  
-r-xr-xr-x  1     210 Feb 12  2018 S00_bge1  
-r-xr-xr-x  1     344 Jan 27  2018 S01_add_users  
-r-xr-xr-x  1     250 Jan 27  2018 S02_root_profile  
-r-xr-xr-x  1     400 Feb 12  2018 S03_smf_services  
-r-xr-xr-x  1    1263 Feb 17  2018 S20_rsyncd  
-r-xr-xr-x  1     534 Feb 11  2018 S30_samba
```

SMF-Persistenz-Dienst: Netzwerk

```
# cat /opt/custom1/local/S00_bge1
...
IF=bge1
AD=192.168.222.96
NM=24

[ "x$1" != "xstart" ] && exit 0

ipadm create-if $IF
ipadm create-addr -T static -a $AD/$NM $IF/v4
exit 0
```

SMF-Persistenz-Dienst: root home

```
[ "x$1" != "xstart" ] && exit 0

cd /root
echo "export TZ=MET" >> .profile
tar xjf /opt/custom/local/root_profile.tar.bz2
chown -R root:root .
exit 0
```

SMF-Persistenz-Dienst: Samba

```
...
NMBD="svc:/pkgsrc/samba:nmbd"
SMBD="svc:/pkgsrc/samba:smbd"

start_smf()
{
    # Make smbd shut up:
    touch /etc/printcap
    chmod 0644 /etc/printcap

    $SVCADM enable $NMBD $SMBD
}

stop_smf()
{
    $SVCADM disable $NMBD $SMBD
}
...
```

Samba in `pkgsrc` ist so gebaut, daß die Konfiguration in `/opt/local/etc/samba` ist – und das ist ja schon persistent.

Fileserver: Ergebnis

- Fileserver mit „ausreichender“ Performance
- die benötigten Protokolle (NFS, rsync, SMB auch für TimeMachine) sind da
- viel Platz auf kleinstem Raum
- alle Disk-Slots für „User-Storage“ ausgenutzt
- Root-FS ist read-only, also kann sich nichts dort einnisten
- alle „bekannten“ Solaris- und ZFS-Tools funktionieren
- leicht erweiterbar (z.B. Webserver)
- ...und Virtualisierung als Bonus!

- 1 Vorwort
- 2 Illumos
- 3 Joyent
- 4 SmartOS - Was ist das?
- 5 SmartOS - Installation
- 6 SmartOS - File-Server
- 7 SmartOS - iSCSI-Server**
- 8 Ausblick
- 9 Links
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS

iSCSI-Server: Was soll er denn können?

- nur iSCSI, sonst nix
- alle iSCSI-Target-Features sollen nutzbar sein
- möglichst kleiner Memory-Footprint

iSCSI-Server: Specs

- HP ProLiant MicroServer N40L @ 1.5GHz
- AMD Turion(tm) II Neo N40L Dual-Core Processor
- 8 GB DDR3-RAM
- 2x Hitachi HDS723030ALA640 SATA
- 1x Broadcom NetXtreme BCM5723 Gigabit Ethernet
- 4x Intel PRO/1000 Gigabit Ethernet

Dieser Microserver hat leider keine serielle Console.

iSCSI-Server: Was brauchen wir?

- Netzwerk-Konfiguration für die iSCSI-Interfaces
- iSCSI-Target-Konfiguration
- Persistenz in der globalen Zone

Die iSCSI-Software ist bereits auf dem USB-Stick. Ansonsten brauchen wir nur noch unseren „Init-Service für arme“. Weil alles so klein wie möglich sein soll, wird auch (fast) nichts aus `pkgsrc` verwendet.

iSCSI: ZFS-Volumes

```
# zfs create -o mountpoint=/iscsi zones/iscsi
# foreach f(000 001 002 003 004 005 006 007 008 009)
foreach? zfs create zones/iscsi/vol$f
foreach? zfs create -V 10G zones/iscsi/vol$f/data
foreach? end
```

```
# zfs list -rtall zones/iscsi
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
zones/iscsi	103G	2.51T	27K	/iscsi
zones/iscsi/vol000	10.3G	2.51T	24K	/iscsi/vol000
zones/iscsi/vol000/data	10.3G	2.52T	12K	-
zones/iscsi/vol001	10.3G	2.51T	24K	/iscsi/vol001
zones/iscsi/vol001/data	10.3G	2.52T	12K	-
...				

iSCSI: iSCSI-LUNs

```
# foreach f ( 00 01 02 03 04 05 06 07 08 09 )
foreach? stmfadm create-lu /dev/zvol/rdisk/zones/iscsi/vol_${f}/data
foreach? end
Logical unit created: 600144F0E798A61100005CAA2BE60001
Logical unit created: 600144F0E798A61100005CAA2BE60002
Logical unit created: 600144F0E798A61100005CAA2BE60003
...

# stmfadm list-lu
LU Name: 600144F0E798A61100005CAA2BE60001
LU Name: 600144F0E798A61100005CAA2BE60002
LU Name: 600144F0E798A61100005CAA2BE60003
...
```

iSCSI: iSCSI-Views

```
# stmfadm add-view 600144F0E798A61100005CAA2BE60001
# stmfadm add-view 600144F0E798A61100005CAA2BE60002
# stmfadm add-view 600144F0E798A61100005CAA2BE60003
...

# stmfadm list-view -l 600144F0E798A61100005CAA2BE80007
View Entry: 0
  Host group   : All
  Target group : All
  LUN          : 6
```

iSCSI: iSCSI-Target

```
# svcadm enable -r svc:/network/iscsi/target:default

# itadm create-target
Target iqn.2010-08.org.illumos:02:a5520d79-6b70-609e-cl68-fd5d2d31c369 successfully created

# itadm list-target -v
TARGET NAME                                     STATE      SESSIONS
iqn.2010-08.org.illumos:02:a5520d79-6b70-609e-cl68-fd5d2d31c369 online      0
    alias:                                     -
    auth:                                     none (defaults)
    targetchapuser:                           -
    targetchapsecret:                         unset
    tpg-tags:                                 default

# itadm list-defaults
iSCSI Target Default Properties:

alias:          <none>
auth:           <none>
radiusserver:   <none>
radiussecret:   unset
isns:           disabled
isnsserver:     <none>
```

iSCSI: Target Portal Group

```
# ipadm set-prop -p hostmodel=strong ipv4
# ipadm set-prop -p hostmodel=strong ipv6

# ipadm create-if e1000g0
# ipadm create-if e1000g1

# ipadm create-addr -T static -a 192.168.20.1/24 e1000g0/v4
# ipadm create-addr -T static -a 192.168.21.1/24 e1000g1/v4

# itadm create-tpg GROUP1 192.168.20.1 192.168.21.1
# itadm list-tpg -v
TARGET PORTAL GROUP          PORTAL COUNT
GROUP1                        2
  portals:    192.168.21.1:3260,192.168.20.1:3260

# itadm modify-target -t GROUP1 iqn.2010-08.org.illumos:02:a5520d79-6b70-609e-c168-fd5d2d...
Target iqn.2010-08.org.illumos:02:a5520d79-6b70-609e-c168-fd5d2d31c369 successfully modified
```

iSCSI: iSCSI-Initiator

```
# uname -a
SunOS t5120-2-cd 5.11 11.3 sun4v sparc SUNW,SPARC-Enterprise-T5120

# svcs initiator
STATE      STIME      FMRI
online      20:19:49   svc:/network/iscsi/initiator:default

# ipadm set-prop -p hostmodel=strong ipv4
# ipadm set-prop -p hostmodel=strong ipv6

# ipadm create-ip net2
# ipadm create-ip net3

# ipadm create-addr -T static -a 192.168.20.2/24 net2/v4
# ipadm create-addr -T static -a 192.168.21.2/24 net3/v4

# iscsiadm add discovery-address 192.168.20.1
# iscsiadm add discovery-address 192.168.21.1

# iscsiadm modify discovery --sendtargets enable
```

iSCSI: Messages auf dem iSCSI-Initiator

```
Apr  7 20:36:47 t5120-2-cd scsi: ssd0 at scsi_vhci0: unit-address g600144f0e798a61100005...
Apr  7 20:36:47 t5120-2-cd genunix: ssd0 is /scsi_vhci/ssd@g600144f0e798a61100005caa2be60001
Apr  7 20:36:47 t5120-2-cd genunix: /scsi_vhci/ssd@g600144f0e798a61100005caa2be60001
(ssd0) online
Apr  7 20:36:47 t5120-2-cd genunix: /scsi_vhci/ssd@g600144f0e798a61100005caa2be60001 (ssd0)
multipath status: degraded: path 1 iscsi0/ssd@0000iqn.2010-08.org.illumos:02:a5520d79-...

Apr  7 20:36:47 t5120-2-cd scsi: ssd1 at scsi_vhci0: unit-address g600144f0e798a61100005...
Apr  7 20:36:47 t5120-2-cd genunix: ssd1 is /scsi_vhci/ssd@g600144f0e798a61100005caa2be60002
Apr  7 20:36:47 t5120-2-cd genunix: /scsi_vhci/ssd@g600144f0e798a61100005caa2be60002
(ssd1) online
Apr  7 20:36:47 t5120-2-cd genunix: /scsi_vhci/ssd@g600144f0e798a61100005caa2be60002 (ssd1)
multipath status: degraded: path 2 iscsi0/ssd@0000iqn.2010-08.org.illumos:02:a5520d79-...
...
```

iSCSI: iSCSI-Target-Connections

```
# iscsiadm list target -v
Target: iqn.2010-08.org.illumos:02:a5520d79-6b70-609e-c168-fd5d2d31c369
  Alias: -
  TPGT: 2
  ISID: 4000002a0000
  Connections: 1
    CID: 0
      IP address (Local): 192.168.20.2:48623
      IP address (Peer): 192.168.20.1:3260
      Transport Type: socket
      Discovery Method: SendTargets
      Login Parameters (Negotiated):
        Data Sequence In Order: yes
        Data PDU In Order: yes
        Default Time To Retain: 20
        Default Time To Wait: 2
        Error Recovery Level: 0
        First Burst Length: 65536
        Immediate Data: yes
        Initial Ready To Transfer (R2T): yes
        Max Burst Length: 262144
        Max Outstanding R2T: 1
        Max Receive Data Segment Length: 32768
        Max Connections: 32
        Header Digest: NONE
        Data Digest: NONE
```

iSCSI: iSCSI-Initiator-Node

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:002128b0e5ee.5b977d03
Initiator node alias: t5120-2-cd
    Login Parameters (Default/Configured):
        Header Digest: NONE/-
        Data Digest: NONE/-
        Max Connections: 65535/-
    Authentication Type: NONE
    RADIUS Server: NONE
    RADIUS Access: disabled
    Tunable Parameters (Default/Configured):
        Session Login Response Time: 60/-
        Maximum Connection Retry Time: 180/-
        Login Retry Time Interval: 60/-
    Configured Sessions: 1
```

iSCSI: iSCSI-Initiator über mehr als ein Interface

```
# iscsiadm modify initiator-node -c 192.168.20.2,192.168.21.2

Apr  7 20:48:44 t5120-2-cd genunix: /scsi_vhci/ssd@g600144f0e798a61100005caa2be60001 (ssd0)
multipath status: optimal: path 11 iscsi0/ssd@0001iqn.2010-08.org.illumos:02:a5520d79-6b70-
609e-c168-fd5d2d31c3690002,0 is online: Load balancing: round-robin

Apr  7 20:48:44 t5120-2-cd genunix: /scsi_vhci/ssd@g600144f0e798a61100005caa2be60002 (ssd1)
multipath status: optimal: path 12 iscsi0/ssd@0001iqn.2010-08.org.illumos:02:a5520d79-6b70-
609e-c168-fd5d2d31c3690002,1 is online: Load balancing: round-robin

# mpathadm list lu
    /dev/rdisk/c0t600144F0E798A61100005CAA2BE60001d0s2
        Total Path Count: 2
        Operational Path Count: 2
    /dev/rdisk/c0t600144F0E798A61100005CAA2BE60002d0s2
        Total Path Count: 2
        Operational Path Count: 2
...

```

iSCSI: LU auf dem Initiator

```
# mpathadm show lu /dev/rdisk/c0t600144F0E798A61100005CAA2BE9000Ad0
Logical Unit: /dev/rdisk/c0t600144F0E798A61100005CAA2BE9000Ad0s2
  mpath-support: libmpscsi_vhci.so
  Vendor: SUN
  Product: COMSTAR
  Revision: 1.0
  Name Type: unknown type
  Name: 600144f0e798a61100005caa2be9000a
  Asymmetric: yes
  Current Load Balance: round-robin
  Logical Unit Group ID: NA
  Auto Failback: on
  Auto Probing: NA

Paths:

  Initiator Port Name: iqn.1986-03.com.sun:01:002128b0e5ee.5caa3f3a,400...
  Target Port Name: 4000002a0000,iqn.2010-08.org.illumos:02:a5520d79-6b...
  Override Path: NA
  Path State: OK
  Disabled: no

  Initiator Port Name: iqn.1986-03.com.sun:01:002128b0e5ee.5caa3f3a,400...
  Target Port Name: 4000002a0001,iqn.2010-08.org.illumos:02:a5520d79-6b...
  Override Path: NA
  Path State: OK
  Disabled: no
```

...

iSCSI: Volume-Status auf dem iSCSI-Target

```
# stmfadm list-lu -v 600144F0E798A61100005CAA2BE60001
LU Name: 600144F0E798A61100005CAA2BE60001
  Operational Status: Online
  Provider Name      : sbd
  Alias              : /dev/zvol/rdsk/zones/iscsi/vol_00/data
  View Entry Count   : 1
  Data File           : /dev/zvol/rdsk/zones/iscsi/vol_00/data
  Meta File           : not set
  Size                : 10737418240
  Block Size          : 512
  Management URL      : not set
  Vendor ID           : SUN
  Product ID          : COMSTAR
  Serial Num          : not set
  Write Protect        : Disabled
  Writeback Cache      : Enabled
  Access State         : Active
```

```
# itadm list-target -v
```

TARGET NAME	STATE	SESSIONS
iqn.2010-08.org.illumos:02:a5520d79-6b70-609e-c168-fd5d2d31c369	online	2
alias:	-	
auth:	none (defaults)	
targetchapuser:	-	
targetchapsecret:	unset	
tpg-tags:	GROUP1 = 2	

iSCSI: Verwenden der Volumes

```
# format
Searching for disks...done

c0t600144F0E798A61100005CAA2BE9000Ad0: configured with capacity of 9.94GB
c0t600144F0E798A61100005CAA2BE60001d0: configured with capacity of 9.94GB
...

AVAILABLE DISK SELECTIONS:
  0. c0t600144F0E798A61100005CAA2BE9000Ad0 <SUN-COMSTAR-1.0 cyl 323 alt 2 hd 254 ...
    /scsi_vhci/ssd@g600144f0e798a61100005caa2be9000a
  1. c0t600144F0E798A61100005CAA2BE60001d0 <SUN-COMSTAR-1.0 cyl 323 alt 2 hd 254 ...
    /scsi_vhci/ssd@g600144f0e798a61100005caa2be60001
...

# zpool create TESTPOOL mirror c0t600144F0E798A61100005CAA2BE9000Ad0 c0t600144F0E798A611 ...

# zpool status -v TESTPOOL
...

```

NAME	STATE	READ	WRITE	CKSUM
TESTPOOL	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
c0t600144F0E798A61100005CAA2BE9000Ad0	ONLINE	0	0	0
c0t600144F0E798A61100005CAA2BE60001d0	ONLINE	0	0	0

```
...
```

iSCSI-Server: Was passiert denn beim Reboot?

Tja, dann sind all diese schönen Sachen wieder weg!

Naja, die ZFS-Volumes sind noch da. Aber alles andere müssen wir neu machen. Und zwar mit denselben GUIDs/WWNs, demselben Target-Namen undsoweiter! Sonst kotzt der Initiator...

Lösung: Wir müssen uns alle diese Daten merken und brauchen dann ein Persistenz-Script, das diese Daten auswertet und alles neu anlegt.

SMF-Persistenz-Dienst: Start/Stop-Scripts

```
# cd /opt/custom/local ; ls -gol S* K*  
-r-xr-xr-x  1      250 Jan 27  2018 S02_root_profile  
-r-xr-xr-x  1      503 Apr  7 18:28 S03_smf_services  
-r-xr-xr-x  1      626 Apr  7 22:57 S10_start_iscsi_target
```

SMF-Persistenz-Dienst: iSCSI-Target

```
BASE=/iscsi
PATTERN=vol

guid=""
# view=""

for vol in $BASE/$PATTERN* ; do
    . $vol/stmf
    stmfadm create-lu -p guid=$guid /dev/zvol/rdisk/zones$vol/data
    stmfadm add-view $guid
done

[ ! -f $BASE/interfaces ] && exit 0

. $BASE/interfaces

[ ! -f $BASE/groups ] && exit 0

while read group ips ; do
    itadm create-tpg $group $ips
done < $BASE/groups

[ ! -f $BASE/targets ] && exit 0

while read group target ; do
    itadm create-target -n $target -t $group
done < $BASE/targets
```

Konfigurationsdateien: Interfaces und Volumes

```
# cat /iscsi/interfaces
...

ipadm create-if e1000g0
ipadm create-if e1000g1

ipadm create-addr -T static -a 192.168.20.1/24 e1000g0/v4
ipadm create-addr -T static -a 192.168.21.1/24 e1000g1/v4

svcadm enable -r svc:/network/iscsi/target:default

# cat /iscsi/targets
GROUP1 iqn.2010-08.org.illumos:02:a5520d79-6b70-609e-c168-fd5d2d31c369

# cat /iscsi/groups
GROUP1 192.168.20.1 192.168.21.1

# cat /iscsi/vol000/stmf
guid=600144F0E798A61100005CAA2BE60001

# cat /iscsi/vol001/stmf
guid=600144F0E798A61100005CAA2BE60002
...
```

iSCSI-Server: Tuning

- die Performance skaliert mit dem System (für echtes I/O ist so ein Microserver zu klein)
- Jumbo Frames einschalten
- Nagle's Algorithm (TCP_NODELAY) abschalten
- Beim Pool-Anlegen auf `ashift` achten
- Blockgröße mit Blick auf die Anwendung wählen (8k ist auch ein guter Wert, weil das in ein Jumbo Frame hineinpaßt)

iSCSI-Server: Ergebnis

- iSCSI-Server mit „ausreichender“ Performance für Test/Lab/...
- „echtes“ iSCSI dank COMSTAR
- viele iSCSI-Volumes auf kleinstem Raum
- alle Disk-Slots für Storage ausgenutzt
- Root-FS ist read-only, also kann sich nichts dort einnisten
- alle „bekannten“ Solaris- und ZFS-Tools funktionieren

- 1 Vorwort
- 2 Illumos
- 3 Joyent
- 4 SmartOS - Was ist das?
- 5 SmartOS - Installation
- 6 SmartOS - File-Server
- 7 SmartOS - iSCSI-Server
- 8 Ausblick**
- 9 Links
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS

Entwicklungen in Illumos

- Übernahme von Commits aus dem ZFS-on-Linux-Repository
- Integration von SMB2
- neue Treiber (hauptsächlich HBAs und NICs)
- Umstieg auf aktuelle `gcc`-Versionen zum Bauen
- Smatch als `lint`-Ersatz

Entwicklungen bei SmartOS

- bessere Unterstützung von AMD-Chips (Epyc, Ryzen)
- bhyve wird die präferierte Virtualisierungstechnik
- Umstellung auf BSD-Loader statt grub

- 1 Vorwort
- 2 Illumos
- 3 Joyent
- 4 SmartOS - Was ist das?
- 5 SmartOS - Installation
- 6 SmartOS - File-Server
- 7 SmartOS - iSCSI-Server
- 8 Ausblick
- 9 Links**
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS

Links

<https://wiki.illumos.org/>

<https://www.joyent.com/>

<https://www.smartos.org/>

<https://smartos.org/blog/>

<https://wiki.smartos.org/display/DOC/Home>

https://us-east.manta.joyent.com/Joyent_Dev/public/SmartOS/latest.html

<https://pkgsrc-eu-ams.joyent.com/images/>

<http://www.pkgsrc.org/>

<https://pkgsrc.joyent.com/>

<https://pkgsrc.joyent.com/install-on-illumos/>

Das war's! Fragen? Anmerkungen?



Bild: BrokenSphere/Wikimedia Commons

Volker A. Brandt, vab@bb-c.de

- 1 Vorwort
- 2 Illumos
- 3 Joyent
- 4 SmartOS - Was ist das?
- 5 SmartOS - Installation
- 6 SmartOS - File-Server
- 7 SmartOS - iSCSI-Server
- 8 Ausblick
- 9 Links
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS

USB-Stick mounten und Boot-Archive extrahieren

```
# mount -F pcfs /dev/dsk/clt0d0p1 /mnt
# cp /mnt/platform/i86pc/amd64/boot_archive /opt/custom/local
# lofiadm -a /opt/custom/local/boot_archive
# mkdir /tmp/boot_archive
# mount -F ufs /dev/lofi/2 /tmp/boot_archive
```

Jetzt ist das Boot-Archive unter `/tmp/boot_archive` gemountet und kann bearbeitet werden.

```
# umount /tmp/boot_archive
# lofiadm -d /dev/lofi/2
```

... und wieder auf den Stick zurückkopieren!

RAM-Disk größer machen

Die Datei `boot_archive` ist ein UFS-Dateisystem und enthält die RAM-Disk. Vergrößern geht mit dem Shell-Script `growfs`, das in zahlreichen Illumos-Distributionen enthalten ist:

```
# dd if=/dev/zero bs=1024 count=16384 >> /var/tmp/boot_archive
16384+0 records in
16384+0 records out
16777216 bytes transferred in 0.201458 secs (83278939 bytes/sec)

# lofiadm -a /var/tmp/boot_archive
/dev/lofi/2

# /tmp/growfs /dev/rlofi/2
/dev/rlofi/2: 576600 sectors in 961 cylinders of 1 tracks, 600 sectors
281.5MB in 61 cyl groups (16 c/g, 4.69MB/g, 384 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 9632, 19232, 28832, 38432, 48032, 57632, 67232, 76832, 86432,
489632, 499232, 508832, 518432, 528032, 537632, 547232, 556832, 566432,
576032

# lofiadm -d /var/tmp/boot_archive
```

... und auch wieder auf den Stick zurückkopieren!

- 1 Vorwort
- 2 Illumos
- 3 Joyent
- 4 SmartOS - Was ist das?
- 5 SmartOS - Installation
- 6 SmartOS - File-Server
- 7 SmartOS - iSCSI-Server
- 8 Ausblick
- 9 Links
- 10 Backup 1: RAM-Disk für das Root-Filesystem größer machen
- 11 Backup 2: Virtualisierung mit SmartOS**

SmartOS als kompromißloser Hypervisor

- die Umgebung dient nur einem Zweck: Hosting von Gästen
- läuft von einem USB-Stick; alle Platten im System sind für Gäste nutzbar
- KVM + bhyve: alle Gäste funktionieren “automatisch”
- Zonen: effiziente Container-Virtualisierung für “native” SmartOS-Anwendungen (meist Infrastruktur)
- LX: Container auf “bare Metal”, schneller/höher/weiter als auf Linux-gehosteten Umgebungen
- darauf basieren die kommerziellen Angebote von Joyent, aber die Basistechnologie ist frei verwendbar
- viele vorgefertigte Gast-Images

Was braucht man?

- Server mit einer einigermaßen “modernen” Intel-CPU (wegen VT-x)
- genügend RAM für die Gäste
- genügend Plattenplatz für die Gäste
- Internet-Verbindung, falls man fertige Gast-Images verwenden will
- sonst nichts!

SmartOS Gast-Images

```
# imgadm sources
https://images.joyent.com
```

```
# imgadm avail | grep base-64
```

c02a2044-c1bd-11e4-bd8c-dfc1db8b0182	base-64-lts	14.4.0	smartos	zone-dataset	2015-03-03
24648664-e50c-11e4-be23-0349d0a5f3cf	base-64-lts	14.4.1	smartos	zone-dataset	2015-04-17
4166f6d6-ea5f-11e4-addd-8351b159d9b6	base-64	15.1.0	smartos	zone-dataset	2015-04-24
[...]					

```
# imgadm avail | grep centos-7
```

5e164fac-286d-11e4-9cf7-b3f73eefcd01	centos-7	20140820	linux	zvol	2014-08-20
[...]					
d1c80032-83d2-11e5-b89f-c317cd0ed1fd	centos-7	20151105	linux	lx-dataset	2015-11-05
aae64e42-c88d-11e5-a49d-87f422b1820b	centos-7	20160201	linux	lx-dataset	2016-02-01

```
# imgadm list
```

UUID	NAME	VERSION	OS	TYPE	PUB
3b811030-4554-11e5-a198-0733b8a63f6c	freebsd-10	20150818	bsd	zvol	2015-08-18
f58ce4f2-beb9-11e5-bb02-e30246d71d58	base-multiarch-lts	15.4.0	smartos	zone-dataset	2016-01-15
aae64e42-c88d-11e5-a49d-87f422b1820b	centos-7	20160201	linux	lx-dataset	2016-02-01

SmartOS: Gast-Konfiguration

```
{
  "brand": "lx",
  "kernel_version": "3.10.0",
  "zfs_io_priority": 30,
  "quota": 20,
  "nowait": true,
  "image_uuid": "aae64e42-c88d-11e5-a49d-87f422b1820b",
  "max_physical_memory": 512,
  "alias": "cent-01",
  "nics": [
    {
      "nic_tag": "igb1",
      "ip": "192.168.222.157",
      "netmask": "255.255.255.0",
      "gateway": "191.168.222.254",
      "primary": 1
    }
  ]
}

{
  "brand": "joyent",
  "zfs_io_priority": 30,
  "quota": 20,
  "nowait": true,
  "image_uuid": "f58ce4f2-beb9-11e5-bb02-e30246d71d58",
  "max_physical_memory": 512,
  [...]
```

Gäste == Zonen

```
# vmadm list
UUID                                TYPE  RAM   STATE  ALIAS
0ce59ad5-5820-e8e8-a779-c901d9fd82df KVM   256   running freebsd0
55d49c2b-2573-cdc3-ceed-b46c4b4a376b LX    512   running cent-01
8c99e36c-09f5-c6c4-857c-f33085a0e615 OS    512   running szone0
3549f837-1a28-67af-d5bf-d8d1bc67d35e KVM   4096  stopped -
```

```
# zoneadm list -vci
ID  NAME          STATUS  PATH                                BRAND  IP
0  global        running /                                     liveimg shared
5  55d49c...    running /zones/55d49c...                  lx     excl
4  8c99e3...    running /zones/8c99e3...                  joyent excl
7  0ce59a...    installed /zones/3549f8...                  kvm    excl
```

```
# zfs list -tall
NAME                                USED  AVAIL  REFER  MOUNTPOINT
zones                              105G   793G   547K   /zones
zones/0ce59a...                   46.0K  20.0G  46.0K   /zones/0ce59a...
zones/0ce59a...df-disk0           10G    803G   1.60G   -
zones/3549f8...                   27.3K  10.0G  27.3K   /zones/3549f8...
```

Noch ein Gäste-Beispiel

```
# vmadm list
UUID                                TYPE  RAM    STATE    ALIAS
728b420c-7d26-c2d6-8110-b9b3d443b1a1  BHYV  1024   stopped  kayak
07b701b5-5a11-c42e-9214-886cd13143ba  LX    6144   stopped  diamond
39cdb0e9-c97d-62d4-e6ab-a9da758927e7  LX    6144   stopped  onyx
472d8bc9-149e-4c7c-c420-c68c200c7cbf  OS    6144   stopped  blue
862c1939-6714-6bda-fd02-d279d293efe9  LX    6144   stopped  sapphire
f880fb39-6a6e-666d-869b-e9316e2807c1  KVM   6144   stopped  jade

# cat /etc/zones/728b420c-7d26-c2d6-8110-b9b3d443b1a1.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
    DO NOT EDIT THIS FILE.  Use zonecfg(1M) instead.
-->
<!DOCTYPE zone PUBLIC "-//Sun Microsystems Inc//DTD Zones//EN" "file:///usr/share/lib/...
<zone name="728b420c-7d26-c2d6-8110-b9b3d443b1a1" zonepath="/zones/728b..."
    autoboot="false" brand="bhyve" ip-type="exclusive"
    limitpriv="default,-file_link_any,-net_access,-proc_fork,-proc_info,...
...
    <attr name="com1" type="string" value="/dev/zconsole"/>
    <attr name="com2" type="string" value="socket,/tmp/vm.ttyb"/>
...

```

Branded-Linux-Prozesse sind Solaris-Prozesse!

```
# ps -z 55d49c2b-2573-cdcb-ceed-b46c4b4a376b
  PID TTY          TIME CMD
 12769 ?              0:00 sleep
 10630 ?              0:00 cgrpmgr
 10588 ?              0:01 systemd
...
```

```
# pargs 12769
12769:  sleep 1h
argv[0]: sleep
argv[1]: 1h
```

```
# pcred 12769
12769:  e/r/suid=0  e/r/sgid=0
```

```
# pldd 12769
12769:  sleep 1h
/usr/lib/amd64/lx_brand.so.1
/usr/lib/amd64/libmapmalloc.so.1
/lib/amd64/libc.so.1
/lib/amd64/librpcsvc.so.1
/lib/amd64/libnsl.so.1
/usr/lib/brand/lx/amd64/lx_vdso.so.1
/zones/55d49c2b-2573-cdcb-ceed-b46c4b4a376b/root/usr/lib64/libc-2.17.so
/zones/55d49c2b-2573-cdcb-ceed-b46c4b4a376b/root/usr/lib64/ld-2.17.so
```

Die Solaris-Tools funktionieren!

```
# ps -z 55d49c2b-2573-cdcdb-ceed-b46c4b4a376b | grep systemd
10588 ?                0:01 systemd
10728 ?                0:00 systemd-

# pfiles 10588
10588:  systemd
  Current rlimit: 65536 file descriptors
    0: S_IFCHR mode:0620 dev:554,3 ino:2870873893 uid:0 gid:5 rdev:92,1
      O_WRONLY|O_NOCTTY|O_LARGEFILE FD_CLOEXEC
      /zones/55d49c2b-2573-cdcdb-ceed-b46c4b4a376b/root/dev/console
      offset:0
  ...
    2: S_IFCHR mode:0666 dev:553,0 ino:23068676 uid:0 gid:3 rdev:44,0
      O_RDWR FD_CLOEXEC
      /devices/pseudo/poll@0:poll
      offset:0
    3: S_IFCHR mode:0000 dev:554,3 ino:65081 uid:0 gid:0 rdev:293,1
      O_RDONLY|O_NONBLOCK|O_LARGEFILE FD_CLOEXEC
      /zones/55d49c2b-2573-cdcdb-ceed-b46c4b4a376b/root/dev/signalfd
      offset:2432
    4: S_IFDIR mode:0755 dev:0,0 ino:16777217 uid:0 gid:0 rdev:0,0
      0x200001|O_NONBLOCK|O_NOCTTY|O_LARGEFILE FD_CLOEXEC
      /zones/55d49c2b-2573-cdcdb-ceed-b46c4b4a376b/root/sys/fs/cgroup/systemd
    5: S_IFCHR mode:0000 dev:554,3 ino:62489 uid:0 gid:0 rdev:292,1
      O_RDWR|O_NONBLOCK|O_LARGEFILE FD_CLOEXEC
  ...
```