

Warum SUSE?

J. Engelhardt

SourceTalkTage 28.–29. August 2012

1 Einführung

Im Privatbereich wählt man sich seine Distribution oft nach eigenen Vorlieben. Beginnt man erst neu und hat eine solche Präferenz noch nicht getroffen, so wird oft das eigene Umfeld als Referenz genommen. Insbesondere durch den steigenden Bekanntheitsgrad von “Linux” sind damit installierte Systeme immer häufiger anzutreffen (passive Werbung durch Präsenz).

Andere Möglichkeit ist die aktive Werbung mittels Kommunikation — Zeitschriften, Foren/Chats mit anderen Nutzern, die jeweils ihre Präferenz kundtun. Wünschenswert wäre im besten Fall, sich mehrere Alternativen anzuschauen und durch Ausprobieren zu einer eigenständigen kritischen Betrachtung kommen.

2 Produktiv-/Businesseseinsatz

Während Sie im Privatbereich noch Ihre eigene Distribution nach Gutdünken nutzen können, liegen im Produktiveinsatz die Vözüge anders. Insbesondere dann wenn Support gewünscht ist, fallen viele der heutzutage unzähligen Distributionsableger aus dem Bild.

Support wird in vielfältiger Weise angeboten, darunter durch den Hersteller einer Enterprise-Distribution selber, wie auch durch Drittanbieter (IT-Consultants). [Anm.: CentOS — das von sich selbst behauptet enterprisig zu sein — bietet keinen eigenen professionellen (bezahlten) Support, sodass man exklusiv auf Drittanbieter angewiesen ist.]

Anwendungsspezifische Software, also solche, die dem Nutzerziel zuarbeiten soll und die durch das eigentliche Betriebssystem-Offering nicht abgedeckt ist, birgt zuweilen Einschränkungen hinsichtlich der Distributionswahl. Für die Hersteller solcher Software stellt sich die Frage, welche Distributionen man denn unterstützen möchte, da damit auch Arbeit verbunden ist.

Gemeint ist damit nicht nur Paketierung — das ist ein relativ kleiner Teil — sondern die Anpassung der eigenen Software an distributionsspezifische Eigenheiten (sofern zutreffend). Ein Office-Paket braucht sich dabei nicht so sehr um so etwas zu kümmern, jedoch haben die Entwickler von systemnahen Konfigurationsprogrammen, wie Webmin, Plesk, cPanel, usw., wiederkehrenden Aufgaben.

Bsp. für Diskrepanzen: `/etc/sysconfig/networkscripts/ifcfg-*` (RedHat), `/etc/sysconfig/network/ifcfg-*` (SUSE), `/etc/network/interfaces` (Debian).

Da erscheint es logisch, dass sich einige Softwarehersteller auf wenige Distributionen beschränken, die zusammen einen möglichst großen Marktanteil abdecken.

Die Hauptakteure: RedHat und SUSE, in dieser Reihenfolge. Ich stelle dies mal als allgemeingültige Meinung dahin, da Zahlen zu aktiven Installationen schwer beizukommen sind.

Fest steht jedoch: Z. B. betreibt ein Drittel der Top 25 bzw. die Hälfte der Top500-Supercomputer SUSE Linux Enterprise. Unter diesen ist auch Europas momentan stärkster, der SuperMUC. Nun sollte Marktanteil nicht ausschlaggebend sein für die Wahl der Distribution, sondern eher der Nutzen aus dieser. Das sollte man auch viel öfter auf dem Desktop berücksichtigen, finde ich.

3 Vorteile im Serverbetrieb

Features die mindestens mir wichtig erscheinen.

3.1 Xen

Der Haushalt des öffentlichen Dienstes kann manchmal beklemmend klamm sein, meistens ist er es auch chronisch. Das Mathematische Institut bspw. hat einige ältere, nichtsdestotrotz weiterhin gut funktionierende SUN X4100-Server Baujahr ca. 2005 mit AMD64-Opteron-Prozessoren, die noch keine hardwareunterstützte Virtualisierung (SVM/VT-*) besitzen, was die Nutzung von KVM ausschließt. Mit Xen in Paravirtualisierung lässt sich der Einsatz von Virtualisierungslösungen wie VirtualBox oder VMware, die ggf. privilegierte Operationen interpretieren (müssen), effizient umgehen.

SUSE bietet Xen für SLES und OpenSUSE an.

RedHat entfernte Xen aus seinem Supportportfolio mit RHEL6.

Natürlich könnte man sich auch immer selber einen Kernel mit Xen zusammenbauen, jedoch werden Eigenlösungen selten von Supportverträgen abgedeckt, und sind auch sonst zeitaufwendiger als mancher Administrator gerne hätte.

3.2 Moderne Paketverwaltung mit zypper

zypper ist ein auf RPM aufbauendes Frontend zur Paketinstallation, ähnlich wie es apt-get ein Frontend für dpkg ist. Zur Auflösung von Abhängigkeiten verwendet zypper eine sonst bisher wenig genutzte Methode, nämlich mittels eines SAT-Solver. Andere Frontends wie apt-get, smart, yum, etc. verwenden heuristische Suchen, oft eine einfache rekursive Suche, die nicht immer das beste Ergebnis liefert, und Alternativen nicht durchleuchtet oder anbietet.

Als Beispiel dazu der Versuch, den systemwichtige Paket "postfix" zu entfernen:

```
# yum remove postfix
--> Running transaction check
---> Package postfix.x86_64 2:2.6.6-2.2.el6_1 will be erased
--> Running transaction check
---> Package cronie.x86_64 0:1.4.4-7.el6 will be erased
--> Running transaction check
---> Package cronie-anacron.x86_64 0:1.4.4-7.el6 will be erased
--> Running transaction check
---> Package crontabs.noarch 0:1.10-33.el6 will be erased
```

```
# zypper rm postfix
The following NEW package is going to be installed:
  exim
The following package is going to be REMOVED:
  postfix
1 new package to install, 1 to remove.
```

Bei yum-Systemen fehlt Ihnen nun die crontab-Funktionalität, bei zypp-Systemen wird versucht, einen alternativen Mail-Daemon anzubieten, ohne Pakete zu entfernen. Gibt es nur schlechtbewertete Lösungen, so ist manuelle Intervention gefragt, und die könnte so aussehen:

```
# zypper in libjpeg-normal
Problem: gimp-accelerated requires libjpeg-turbo,
but this requirement cannot be provided
Solution 1: Following actions will be done:
  deinstallation of gimp-accelerated
Solution 2: do not install libjpeg-normal
Solution 3: break gimp-accelerated by ignoring some of its dependencies
Choose from above solutions by number or cancel [1/2/3/c] (c):
```

Mittels zypper bzw. dessen bibliothekifizierte Variante "libzypp" werden auch Distributionsupgrades angeboten und hauptsächlich durchgeführt, ähnlich wie mit apt-get dist-upgrade unter Debian.

RHEL hingegen unterstützt offiziell kein dist-upgrade. Das Mantra hier lautet Neuinstallation. Dennoch eine Art dist-up mit yum durchzuführen stößt auf vielseitige Arbeit.¹

3.3 btrfs

btrfs gilt als Next-Generation-Dateisystem [zum Zeitpunkt dieses Vortrags]. Der Funktionsumfang schneidet sich mit ZFS, jedoch ist/wird die Installationsmasse ("installation base") um vieles größer. Es unterstützt u. a. Storagepools mit verschiedenen RAID-Levels, Subvolumes, Snapshots, Prüfsummen, Kompression.

SUSE nimmt an der Entwicklung von btrfs teil, und portiert insbesondere Fixes zurück in die Enterprise-Kernel, und ist somit zugleich einer der wenigen, die btrfs bereits für den Produktiveinsatz anbieten statt nur als "Technology Preview". (Oracles Unbreakable Linux ist die andere mir bekannte Distribution, die es voll unterstützt. RHEL6 hat btrfs nur als Preview.)

Wer noch nicht so experimentierfreudig ist, der nimmt XFS, das sich als sehr gut skalierendes Dateisystem für parallele Zugriffe präsentiert, obgleich es ein "klassisches" Dateisystem darstellt, das vollständig auf darunterliegende Schichten (d. h. RAID, LVM, etc.) für seine Konsistenz vertraut.

XFS tauchte erstmalig in RHEL6 (2012) als supported auf²; bei SUSE war es seit SLES 8 (2006) dabei³.

¹<http://www.it-hure.de/2011/10/update-rhel5-to-rhel6/>

²https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/6.0_Release_Notes/filesystems.html

³http://www.novell.com/connectionmagazine/2010/12/two_paths_to_server_performance_two.html

Normalnutzer bekommen bei der Installation oft ext4 als Default angesetzt, welches bei Parallelität schnell einbricht⁴.

3.4 Backups mit btrfs

Für Backups wird btrfs bereits durch das Mathematische Institut eingesetzt, da die copy-on-write Snapshot-Funktionalität dies platzsparend und auf einfache Weise erlaubt.

Wo früher Software von Drittanbietern oder eigene Skripte mit find(1)/tar(1)-Logik zunächst alle Dateien mit Datum neuer als der letzte Backupzeitpunkt ausfindig machen musste und diese dann in ein Archiv schrieb — und das meist auch nicht atomar tun konnte — so genügt es heute, von einem Subvolume einen Schnappschuss zu erzeugen.

Neue Datenblöcke werden an noch freie Stellen geschrieben, ohne alte zu verwerfen oder unnötig zu duplizieren. Alle solche Subvolumes sind für Nutzer wie ein normaler Dateibaum navigierbar. Die Nachteile von klassisch-inkrementellen Backups, bei denen Dateien ggf. über mehrere Archive verstreut sind, und klassisch-vollen Backups, die viel Platz benötigen, entfällt.

Im Betrieb sieht das folgendermaßen aus:

```
# ls -l
total 8
dr-xr-xr-x  1 root root 428 Aug 16 13:54 .
drwxr-xr-x 25 root root 4096 Jul  9 15:51 ..
drwxr-xr-x  1 root root 504 Jul 11 11:24 HEAD
drwxr-xr-x  1 root root 504 Jul 11 11:24 snap-2012-08-15
drwxr-xr-x  1 root root 504 Jul 11 11:24 snap-2012-08-16
drwxr-xr-x  1 root root 504 Jul 11 11:24 snap-2012-08-17
drwxr-xr-x  1 root root 504 Jul 11 11:24 snap-2012-08-18
[...]
# btrfs sub list /top.srv/
ID 256 top level 5 path HEAD
ID 1154 top level 5 path snap-2012-08-15
ID 1155 top level 5 path snap-2012-08-16
ID 1184 top level 5 path snap-2012-08-17
ID 1193 top level 5 path snap-2012-08-18

# df -BG .
Filesystem 1G-blocks  Used  Available  Use%  Mounted on
/dev/sdf          5590G 3470G      1925G   65%  /top.srv
```

⁴<http://www.youtube.com/watch?v=FegjLbCnoBw> @ 20:50

```

# btrfs fi df .
Data, RAID10: total=2.10TiB, used=1.68TiB
System, RAID10: total=128.00MiB, used=260.00KiB
System: total=4.00MiB, used=0.00
Metadata, RAID10: total=112.00GiB, used=14.42GiB

# btrfs fi show
Label: 'srv' uuid: 78951b79-300a-4cea-b8ee-f70f580d58b9
Total devices 4 FS bytes used 1.69TiB
devid 1 size 1.36TiB used 1.11TiB path /dev/sdc
devid 3 size 1.36TiB used 1.11TiB path /dev/sde
devid 4 size 1.36TiB used 1.11TiB path /dev/sdf
devid 2 size 1.36TiB used 1.11TiB path /dev/sdd

```

Da Belegung bei COW-Daten Ansichtssache ist, zeigen verschiedene Tools unterschiedliche Werte an, die jedoch alle im Prinzip konsistent sind. `'btrfs fi show'` und `'df'` zeigen Rohkapazität und -nutzung an, werden also von Optionen wie RAID1/10/DUP, Kompression, beeinflusst. `'btrfs fi df'` hingegen zeigt Nutzdatenmasse, zumindest eine Art davon. Es hängt alles davon ab, wie man geCOWtes zählt. Entsprechend ist auch die Arithmetik bei Ausgabe von `'du'` gebogen: Sowohl eine Datei A als auch eine Datei B belegen — einzeln gesehen — x Bytes, zusammen aber zwischen $1 \cdot x$ und $2 \cdot x$.

Einen Nachteil in dieser Backup-Nutzung des btrfs-Dateisystems gibt es leider: die Aneinanderreihung von Snapshots mittels dem täglich verwendeten Befehl `'btrfs sub snap HEAD $(date +%F)'` führt dazu, dass das Verzeichniseinträge “HEAD”-genannte Subvolume — mit fortschreitender Zeit — Referenzen querbeet über das Storage Device enthält, also praktisch fragmentiert. Dies hat bereits zu messbaren Einbußen beim Lese-Schreib-Zyklus geführt. Insbesondere das Update (`'rsync --inplace'`) eines VM-Imagefiles (4 GB und mehr) dümpelte mit weniger als 10 MByte/s dahin.

4 Mitwirkungsmöglichkeiten

- Distribution ausprobieren und bei Fehlern rückmelden (Bugzilla)
- eigenständige Behebung von Fehlern, Paketierung von Software (Build Service)
- Artwork, Übersetzung
- “politische Ebene”, Boosters, Board

5 Build Service

Die “Open Build Service”-Software ermöglicht es, selbst Pakete für nicht nur SUSE-Distributionen zu erstellen. OBS kann bereits als etwas gehobenes Buildsystem angesehen werden, da es selbstständig notwendige Neukompilierungen von Reverse-Abhängigkeiten anstößt (also bspw. gimp nach libjpeg), und dabei die Last auch auf einen Worker-Cluster verteilen kann.

Man kann sich selbst eine sog. Instanz des Services installieren, oder ein öffentliches Angebot wie das von `build.opensuse.org` nutzen. Öffentliche haben u. U. Beschränkungen hinsichtlich

proprietärer Software. Auch Software, deren Quellcode nicht publik werden soll, sollte man auch besser in seiner eigenen Instanz behalten. Eine eigene Instanz lässt trotzdem den lesenden Zugriff auf öffentliche Instanzen zu, sodass bereits existierende Pakete, insbesondere wie Compiler, direkt ohne zusätzlichen Importschritt genutzt werden können.

Zu den OpenSUSE Build Service Repositories:

Debian und deren User mochten ja immer gerne mit Paketvielfalt werben (Ubuntu auch), aber tatsächlich hat OpenSUSE (mit Einbeziehung der Build Service-Repos) die beiden schon längst überholt. Mit einer Wachstumsrate von ca. 30 Pakete/Tag stehen momentan über 53000 BRPMS allein für OpenSUSE 12.1⁵ zur Verfügung. (Debian/Ubuntu: 35000–37000)

Trotz diesen Wachstums ist openSUSE weiterhin eine zusammenhaltende Community; signifikante Abspaltungen gibt es nicht. (Vgl.: Debian–Ubuntu–Mint)

6 Sonstiges

SUSEs Zusammenarbeit mit Herstellern:

Blast from the past: Als einer der ersten bot SUSE Linux die Installation von CD. Da der ATAPI-Standard noch nicht etabliert war, galt es, Kernaltreiber für die Laufwerke in Kollaboration mit den Hardwareherstellern erstellt zu bekommen, um die Installation auf diese Weise flächendeckend zu ermöglichen. (3.5-Zoll-Disketten mit Möglichkeit zur Installation über andere Medien wurden noch mindestens bis 2000 geliefert.)

⁵Ohne Projekte aus dem “home:”-Namespace, und auch nur erfolgreich produzierte BRPMS.

Warum SUSE?

Jan Engelhardt <jan@inai.de>

2012-Aug-29

- 1 Einleitung
- 2 Serverbetrieb
- 3 Mitwirken
- 4 Build Service

Distributionswahl persönlich

Wahl der eigenen Linuxdistro geschieht durch

- Beeinflussung durch Umfeld
 - vorinstallierte Systeme
 - persönliche Präferenzen von anderen Nutzern
- eigenständige kritische Betrachtung und Durchprobieren anderer Wahlmöglichkeiten

im Produktiveinsatz

- Wunsch nach Support schränkt Distro-Wahl ein
- Wunsch nach spez. Software schränkt Distro-Wahl ggf. ein
- Anpassungen an Distribution mit Aufwand für SW-Hersteller verbunden

im Produktiveinsatz

- Wunsch nach Support schränkt Distro-Wahl ein
- Wunsch nach spez. Software schränkt Distro-Wahl ggf. ein
- Anpassungen an Distribution mit Aufwand für SW-Hersteller verbunden

Example

Diversität

- `/etc/sysconfig/networkscripts/ifcfg-*`
- `/etc/sysconfig/network/ifcfg-*`
- `/etc/network/interfaces`

- Folge: SW-Hersteller beschränken sich auf Marktführer

Xen

- KVM erfordert Hardwareunterstützung (fehlt bisweilen)
- homogene Linuxumgebung ermöglicht Paravirtualisierung
- eigenes Kernelcompilat oft nicht tragbar

zypper

Moderne Paketverwaltung mit zypper

- zypper (als Frontend zu libzypp) als Frontend zu rpm
- SAT-Solver: schnell, korrekt, allwissend

Beispiel: “postfix” entfernen

Rekursive Auflösung von Abhängigkeiten

```
# yum remove postfix
-> Running transaction check
--> Package postfix.x86_64 2:2.6.6-2.2.el6_1 will be erased
-> Running transaction check
--> Package crontabs.x86_64 0:1.4.4-7.el6 will be erased
-> Running transaction check
--> Package crontabs.noarch 0:1.10-33.el6 will be erased
```

- cron würde nicht mehr funktionieren

Beispiel: “postfix” entfernen

Featureverlust vermeiden

```
# zypper rm postfix
```

```
The following NEW package is going to be installed:
```

```
exim
```

```
The following package is going to be REMOVED:
```

```
postfix
```

```
1 new package to install, 1 to remove.
```

- Angebot einer alternativen Software
(cron will auch hier einen smtp_daemon)

Beispiel: conflicting package

Installation eines konflikt erzeugendes Paketes

```
# zypper in -- libjpeg-normal -libjpeg-turbo
Problem: gimp-accelerated requires libjpeg-turbo, but this
requirement cannot be provided
  Solution 1: Following actions will be done:
    deinstallation of gimp-accelerated
  Solution 2: do not install libjpeg-normal
  Solution 3: break gimp-accelerated by ignoring some of
its dependencies
Choose from above solutions by number or cancel [1/2/3/c]
(c):
```


Filesysteme

- btrfs als modernes FS mit integriertem RAID auf Objektebene, Subvolumes, Snapshots, Checksums, Kompression
- XFS als skalierendes FS
- ext4: wechselndes Verhalten
- David Chinner über XFS Scalability (u.a. im Vergleich mit ext4 und btrfs) – <http://www.youtube.com/watch?v=FegjLbCnoBw>

Backups mit btrfs

- früher: inkrementell, Timestamps vergleichen, tar-Archiv o.ä.
- btrfs: sofortiger atomarer Snapshot
- btrfs: voll navigierbar, schnell durchsuchbar

btrfs subvols

Ansicht

```
# ls -l
drwxr-xr-x 1 root root 504 Jul 11 HEAD
drwxr-xr-x 1 root root 504 Jul 11 snap-2012-08-15
drwxr-xr-x 1 root root 504 Jul 11 snap-2012-08-16
drwxr-xr-x 1 root root 504 Jul 11 snap-2012-08-17
```

btrfsprogs

```
# btrfs sub list /top.srv/
ID 256 top level 5 path HEAD
ID 1154 top level 5 path snap-2012-08-15
ID 1155 top level 5 path snap-2012-08-16
ID 1184 top level 5 path snap-2012-08-17
```

btrfs disk stats

```
df
```

```
# df -BG
Filesystem 1G-blocks  Used  Available  Use%  Mounted on
/dev/sdf          5590G 3470G      1920G    65%  /top.srv
```

```
btrfsprogs
```

```
# btrfs fi df .
Data, RAID10: total=2.10TiB, used=1.68TiB
System, RAID10: total=128.00MiB, used=260.00KiB
System: total=4.00MiB, used=0.00
Metadata, RAID10: total=112.00GiB, used=14.42GiB
```

- Zählweise manchmal gewöhnungsbedürftig

btrfs info

btrfsprogs

```
# btrfs fi show
```

```
Label: "srv" uuid: 78951b79-300a-4cea-b8ee-f70f580d58b9
Total devices 4 FS bytes used 1.69TiB
devid 1 size 1.36TiB used 1.11TiB path /dev/sdc
devid 3 size 1.36TiB used 1.11TiB path /dev/sde
devid 4 size 1.36TiB used 1.11TiB path /dev/sdf
devid 2 size 1.36TiB used 1.11TiB path /dev/sdd
```

- du(1) über Objekte mit geteilten Extents problematisch
- Fragmentierung

Mitwirkungsmöglichkeiten

- Distribution ausprobieren und bei Fehlern rückmelden (Bugzilla)
- eigenständige Behebung von Fehlern, Paketierung von Software (Build Service)
- Artwork, Übersetzung
- “politische Ebene”

Open Build Service

- wenn rpmbuild -bb/dpkg -b nicht genug ist
- OBS hilft bei der Einrichtung des Buildroots mit allen notwendigen Abhängigkeiten

Open Build Service

- wenn rpmbuild -bb/dpkg -b nicht genug ist
- OBS hilft bei der Einrichtung des Buildroots mit allen notwendigen Abhängigkeiten
- automatische Neukompilierung (bspw. gimp nach libjpeg)
- Verteilung der Buildjobs auf einen Workercluster

Open Build Service

- wenn `rpmbuild -bb/dpkg -b` nicht genug ist
- OBS hilft bei der Einrichtung des Buildroots mit allen notwendigen Abhängigkeiten
- automatische Neukompilierung (bspw. `gimp` nach `libjpeg`)
- Verteilung der Buildjobs auf einen Workercluster
- Nutzung von öffentlichen Instanzen möglich – `build.opensuse.org`
- proprietäre Pakete in einer privaten Instanz bauen

Open Build Service

- wenn `rpmbuild -bb/dpkg -b` nicht genug ist
- OBS hilft bei der Einrichtung des Buildroots mit allen notwendigen Abhängigkeiten
- automatische Neukompilierung (bspw. `gimp` nach `libjpeg`)
- Verteilung der Buildjobs auf einen Workercluster
- Nutzung von öffentlichen Instanzen möglich – `build.opensuse.org`
- proprietäre Pakete in einer privaten Instanz bauen
- größtes Repository