



rear for corporate use

**Ralf Dannert**  
Systems Engineer  
SUSE Linux GmbH

23.Sept 2014



# Agenda

- Overview Disaster Recovery
- Relax-and-Recover (rear)
- rear in SLE12(btrfs subvolumes)
- Demo
- rear-SUSE



# Disaster Recovery Strategies

- Disk Imaging
- Copy files, store disk layout and bootinfo
  - consistency problems with btrfs
- Put it somewhere else..
- Bootable media
- Recover later

# Disaster Recovery – How it works

- Store the disk layout
  - Partitions(RAID), filesystems, labels
  - Bootloader
- Store files(tar, backup, backup software)
- Create bootable rescue media with system config
- Do it online
- 100% compatible with original system(driver, firmware,..)

# Rescue media

- Use what's there
- Mini rescue linux from running system
  - Guarantees compatibility between original and backup
- kernel, ramdisk(base for DR system)

# Actual Recovery

- Boot system from rescue media
- Restore disk layout
  - Partitions, filesystems, labels, mountpoints
- Restore backup data
- Restore bootloader
- Reboot

Relax and Recover - rear



# Relax and Recover - rear

- Only bash scripts
- Uses binaries of host
- GPL, mailinglists, [github.com/rear](https://github.com/rear)
- <http://relax-and-recover.org/>
- Main tasks:
  - recreate system as it was before
  - Use rear as Addon to provide Disaster Capabilities for existing backup software
- Linux philosophy



# Features 1/3

- Integrates into your enterprise backup software
  - ReaR uses backup software for data storage and retrieval
  - Modular design supports all backup software vendors
- Supported backup and output formats

NAME	TYPE	Description
NETFS	BACKUP	Copy files to NFS, CIFS share or local file system
TAPE	BACKUP	Copy files to tape(s)
CDROM	BACKUP	Copy files to CD/DVD
NSR	BACKUP	Use EMC <sup>2</sup> Legato Networker ( <b>SUSE Consulting Project</b> )
TSM	BACKUP	Use Tivoli Storage Manager
DP	BACKUP	Use HP Data Protector
BACULA	BACKUP	Use opensource Bacula
BAREOS	BACKUP	Use Backup Archive REcovery Open Sourced
ISO	OUTPUT	Write result to an ISO9660 image
CDROM	OUTPUT	Write result to a CD/DVD
OBDR	OUTPUT	Create an OBDR Tape
PXE	OUTPUT	Create a PXE bootable files on TFTP server
USB	OUTPUT	Create a bootable USB device

# Features 2/3

- Support every hardware
  - Rescue media is generated from production system
- High security disaster recovery
  - No private data on rescue system (backup SW dependent)
  - Secure drop-off storage of rescue media
- Disaster Recovery as an Infrastructure Service
  - All systems are covered by default
  - Tiny data storage requirements (ISO image ~ 30-100MB / per system)
- Restore on newer hardware possible
  - With enterprise brand hardware (if original OS supports it)
- Physical to virtual machine migration (p2v) possible

# Features 3/3

- rear can be done online!
- One-Button-Solution
  - No know-how required for disaster recovery
  - “Bare Metal” restore to last backup
- Supports all Linux features
  - SW/HW RAID, LVM, weird partitioning ...
- Supports HA cluster
  - DRBD detection
  - exclude option for cluster controlled shared storage



# Relax and Recover - rear

- highly modular disaster recovery framework
- rear is part of the SLE High Availability Extension
  - rear is only part of the x86 and x86\_64 architectures
  - support level is inherited by underlying SLES
  - rear was added with SLE HA 11 SP1
  - SUSE 3<sup>rd</sup> level support & consulting services available
- rear versions in SLES:
  - rear version 1.16 in SLES12
  - rear version 1.10 in SLES11 → version 1.15 in SLES11 SP3
- Environment: SLES, SLE-HA, SAP HANA, Oracle RAC

# rear 1.16 Command Line Interface

```
rear -v help
Usage: rear [-dDsSvV] [-c DIR ] [-r KERNEL] COMMAND [-- ARGS...]
```

Relax-and-Recover comes with ABSOLUTELY NO WARRANTY; for details see the GNU General Public License at: <http://www.gnu.org/licenses/gpl.html>

## Available options:

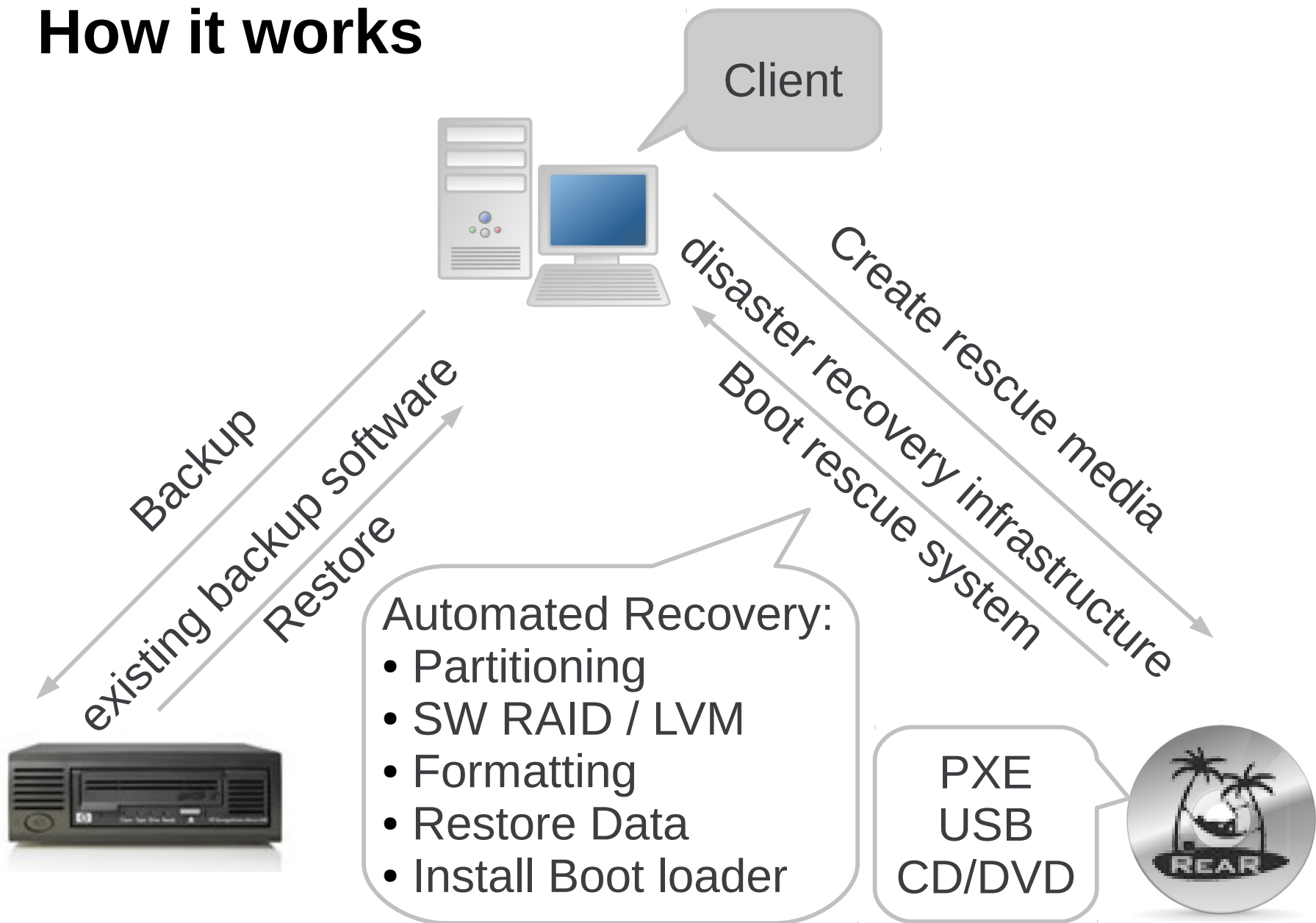
```
-c DIR          alternative config directory; instead of /etc/rear
-d             debug mode; log debug messages
-D            debugscript mode; log every function call
-r KERNEL      kernel version to use; current: '3.12.25-2-default'
-s           simulation mode; show what scripts rear would include
-S          step-by-step mode; acknowledge each script individually
-v         verbose mode; show more output
-V         version information
```

## List of commands:

```
checklayout    check if the disk layout has changed
dump           dump configuration and system information
format         format and label media for use with rear
mkbackup       create rescue media and backup system
mkbackuponly   backup system without creating rescue media
mkrescue       create rescue media only
savelayouth    save the disk layout of the system
shell          start a bash within rear; development tool
udev           udev handler; triggered by udev rule
validate       submit validation information
```



# How it works



# Files and Directories

FS layout: ReaR tries to be as much LSB compliant as possible. Therefore rear will be installed into the usual locations:

- /etc/rear/ Configuration
- /usr/sbin/rear Main program
- /usr/share/rear/ Internal scripts
- /var/lib/rear/ Recovery and disk and file system layout information
- /var/log/rear/ Log files of ReaR is kept here
- /tmp/rear.\$\$/ Build area

# Configuration Examples

- Configuration is normally done in `/etc/rear/local.conf`
- See `/usr/share/rear/conf/default.conf` for all possible parameters and variables to use
- See `/usr/share/rear/doc/configuration-examples.txt` and <http://sourceforge.net/projects/rear/files/documentation/> for more examples
- The simplest configuration is

```
~# cat /etc/rear/local.conf

# default backup and output targets
BACKUP=NETFS
OUTPUT=ISO
```



# The OUTPUT Variable

There are 2 very important variables to define, OUTPUT and BACKUP. The OUTPUT variable defines where the rescue image should be send to. The BACKUP setting defines our backup/restore strategy.

- OUTPUT=ISO : creates a bootable ISO9660 image
- OUTPUT=PXE : creates on a remote PXE/NFS server the required files
- OUTPUT=TSM : use IBM Tivoli Storage Manager programs
- OUTPUT=DP : use HP DataProtector programs
- OUTPUT=NBU : use Symantec NetBackup programs
- OUTPUT=BACULA: use Bacula programs
- OUTPUT=OBDR: create a bootable OBDR tape including the backup archive
- OUTPUT=USB : create a bootable USB disk (with syslinux)

# Using the NETFS Method

To backup to a local disk, NFS, CIFS or USB disk and using an ISO image as boot medium:

```
OUTPUT=ISO
```

```
BACKUP=NETFS
```

NETFS\_URL setting depends on the type of disk

- local disk: `NETFS_URL=file:///directory/path/`
- NFS disk: `NETFS_URL=nfs://nfs-server-name/directory/path`
- CIFS disk: `NETFS_URL=cifs://cifs-server/directory/path`
- USB disk: `NETFS_URL=usb:///dev/sdb1`

Together with `OUTPUT=USB` everything is on an USB device.

```
NETFS_KEEP_OLD_BACKUP_COPY=y
```

If you want to keep the previous backup archive

# Example: Bootable USB Drive

Initial USB Drive format (done once):

```
~# rear format /dev/sdX # It will be labeled REAR-000
```

## Configuration

```
~# cat /etc/local/rear.conf
```

```
BACKUP=NETFS
```

```
OUTPUT=USB
```

```
OUTPUT_URL="usb:///dev/disk/by-label/REAR-000"
```

Create bootable USB drive with the backup

```
~# rear mkbackup
```

# Example: NFS Share

```
# Create ReaR rescue media as ISO image
OUTPUT=ISO
# optionally define backup software, e.g. TSM, NBU, DP, BACULA
BACKUP=NETFS
NETFS_URL=nfs://e6500os123/srv/backup
NETFS_KEEP_OLD_BACKUP_COPY=Y

# the following is required on older VMware VMs
# MODULES_LOAD=( vmxnet )
# line below was automatically added by 21_include_dhclient.sh
DHCLIENT_BIN=dhpcd
# line below was automatically added by 21_include_dhclient.sh
DHCLIENT6_BIN=
# Enable the use of cfg2html (set to non-empty to disable). cfg2html must be
installed independently from ReaR.
SKIP_CFG2HTML=N
# Warn about NETFS not being a professional backup solution
NETFS_SKIP_WARNING=Y
# additional tools
PROGS=( "${PROGS[@]}" lsfd parted fdisk cfdisk )
```

```
e6500os123:/srv/backup/sles11sp3 # ll
total 1632460
-rw-r--r-- 1 nobody nogroup      202 Oct 31 14:27 README
-rw-r--r-- 1 nobody nogroup       290 Oct 31 14:27 VERSION
-rw-r--r-- 1 nobody nogroup 15340384 Oct 31 14:45 backup.log
-rw-r--r-- 1 nobody nogroup 1589051033 Oct 31 14:45 backup.tar.gz
-rw-r--r-- 1 nobody nogroup   65562624 Oct 31 14:27 rear-sles11sp3.iso
-rw-r--r-- 1 nobody nogroup    22474 Oct 31 14:27 rear.log
e6500os123:/srv/backup/sles11sp3 # █
```



# Exkurs: btrfs in SLES12

# Filesystems: btrfs – Features

## Mature / Supported

Copy on Write

Snapshots

Subvolumes

Metadata Integrity

Data Integrity

Online metadata scrubbing

Manual Defragmentation

Manual Deduplication

Quota Groups

## Not (yet) mature

Inode Cache

Auto Defrag

RAID

Compression

Send / Receive

Hot add / remove

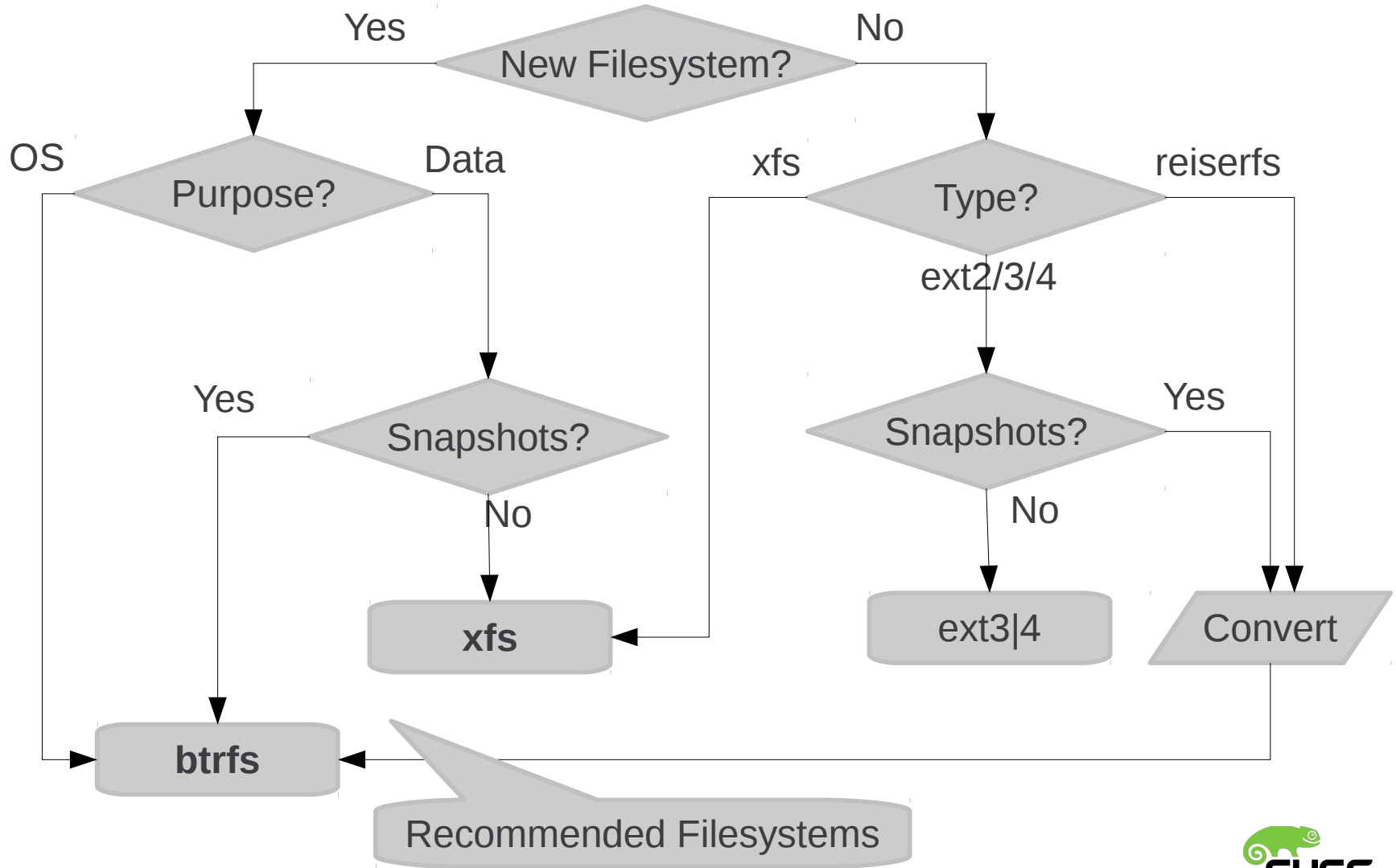
Seeding devices

Multiple Devices

“Big” Metadata



# Filesystem recommendations



# btrfs

Copy on Write on a

- full subvolume tree

Trees for

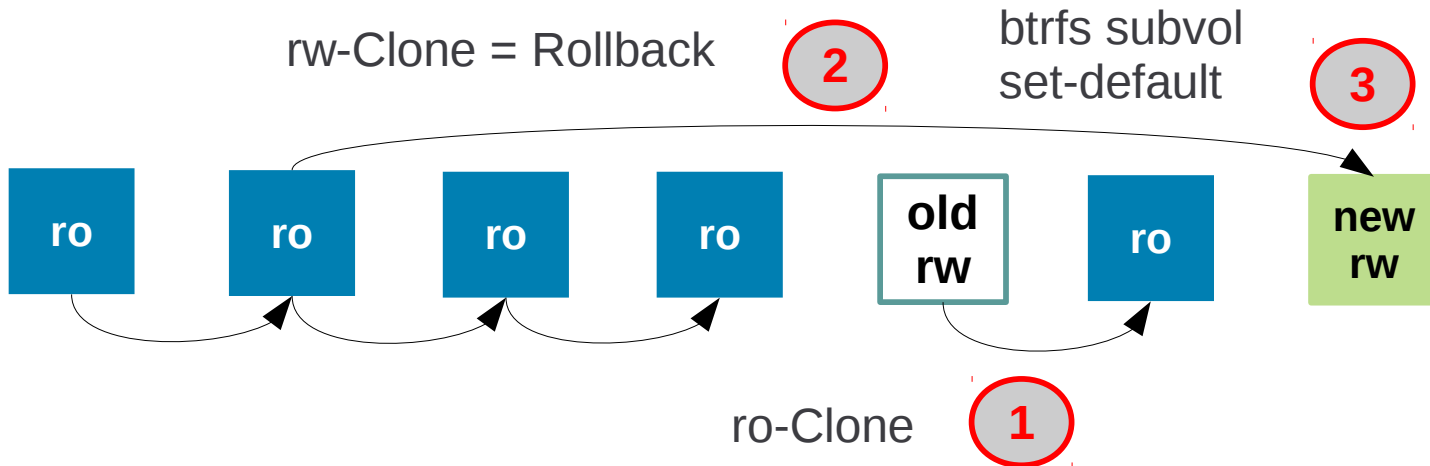
- Data
- Metadata

Details

- Every snapshot is again a subvolume of its own
- Snapshots (as subvolumes) can be mounted and accessed as every other subvolume
- Snapshots can be created read-only
- Expected ENOSPC behaviour



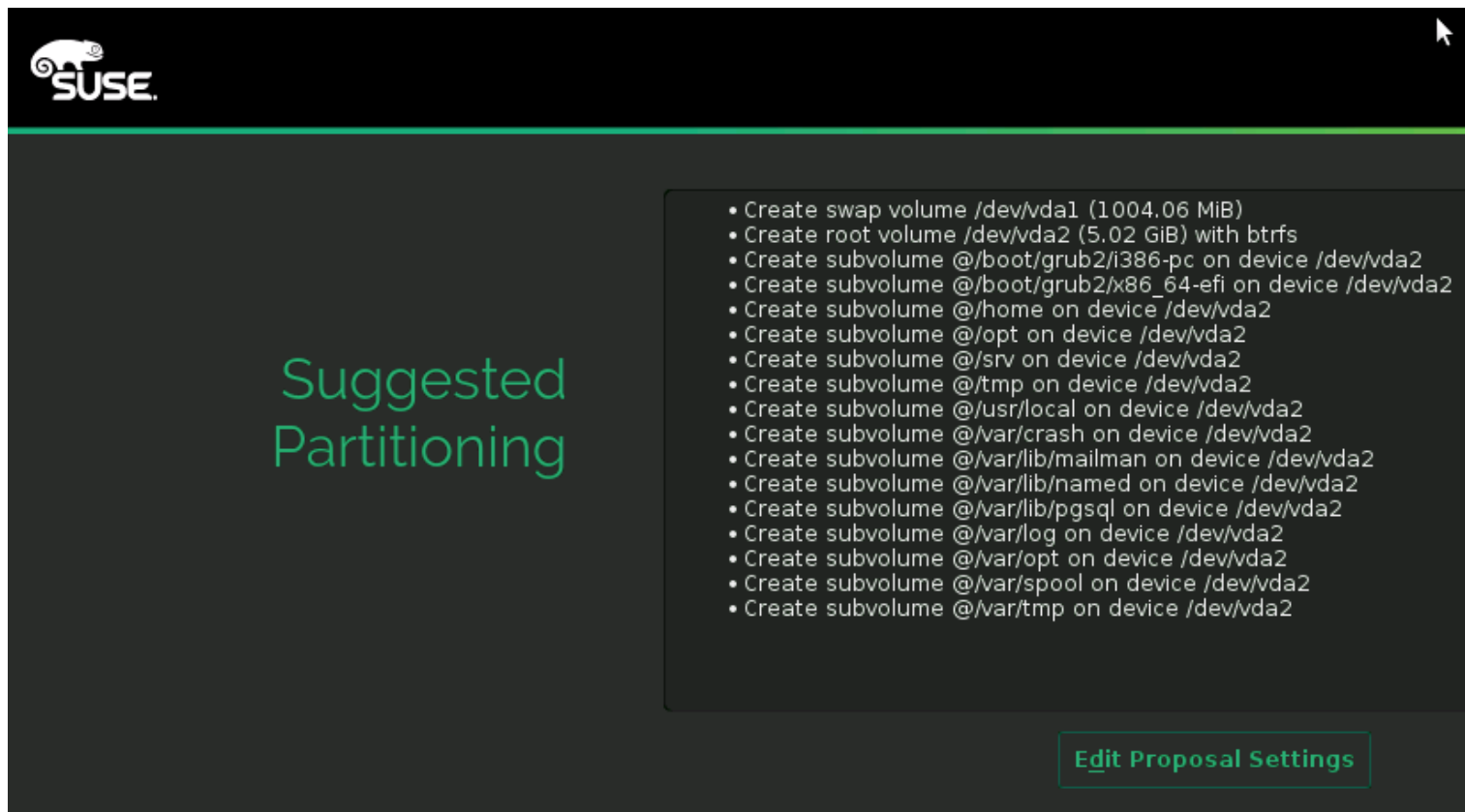
# User view on Snapshot History



# Btrfs subvolumes

- a btrfs subvolume has its hierarchy and relations between other subvolumes(unlike LVM)
- Access
  - from parent subvolume - just like a directory
  - Or as separate mounted filesystem(subvolid mount option)
- default subvolume as its initially top-level subvolume, whose subvolume id is 5(FS\_TREE)
- Due to the capabilities of COW, modifications inside a snapshot will only show in a snapshot but not in its source subvolume
- A btrfs snapshot is much like a subvolume, but shares its data(and metadata) with other subvolume/snapshot
- snapshot can not be made from normal directories

# Recommendation subvolume layout



The screenshot shows the SUSE partitioning tool interface. On the left, the text "Suggested Partitioning" is displayed in green. On the right, a list of subvolumes is shown, each with its path and device. At the bottom right, there is a button labeled "Edit Proposal Settings".

- Create swap volume `/dev/vda1` (1004.06 MiB)
- Create root volume `/dev/vda2` (5.02 GiB) with btrfs
- Create subvolume `@/boot/grub2/i386-pc` on device `/dev/vda2`
- Create subvolume `@/boot/grub2/x86_64-efi` on device `/dev/vda2`
- Create subvolume `@/home` on device `/dev/vda2`
- Create subvolume `@/opt` on device `/dev/vda2`
- Create subvolume `@/srv` on device `/dev/vda2`
- Create subvolume `@/tmp` on device `/dev/vda2`
- Create subvolume `@/usr/local` on device `/dev/vda2`
- Create subvolume `@/var/crash` on device `/dev/vda2`
- Create subvolume `@/var/lib/mailman` on device `/dev/vda2`
- Create subvolume `@/var/lib/named` on device `/dev/vda2`
- Create subvolume `@/var/lib/pgsql` on device `/dev/vda2`
- Create subvolume `@/var/log` on device `/dev/vda2`
- Create subvolume `@/var/opt` on device `/dev/vda2`
- Create subvolume `@/var/spool` on device `/dev/vda2`
- Create subvolume `@/var/tmp` on device `/dev/vda2`

[Edit Proposal Settings](#)

- mountpoint no longer corresponds to one whole filesystem
- "vda2 on /opt type btrfs" is by default only a btrfs subvolume

# SUSE specific adaptations for SLE12

```
[rdannert@sled11sp1 home:jsmeix]$ osc co rear116
A   rear116
A   rear116/SLE12-btrfs-example.conf
A   rear116/adaptions_for_btrfs_for_SLE12.diff
A   rear116/avoid_unary_operator_expected_error.diff
A   rear116/rear-1.16.tar.gz
A   rear116/rear-release-notes.txt-1.16
A   rear116/rear116.changes
A   rear116/rear116.spec
A   rear116/traditional_style_for_bash_3.diff
At revision 2.
```

# specfile: disable version upgrades

## SLE12 specific

```
Name:          rear116
Version:       1.16
Release:       0
# Automatic version upgrades are not possible in practice.
# The user must explicitly specify his intended version.
# When users have a working disaster recovery procedure, they should not upgrade
# (see "Version upgrades" at http://en.opensuse.org/SDB:Disaster\_Recovery).
# Users who already use it and like to upgrade, must re-validate that their
# particular disaster recovery procedure still works.
# For one product (e.g. SLE11 or SLE12) we provide several versions in parallel
# so that users where version N does not support their particular needs
# can upgrade to version M but on the other hand users who have a working
# disaster recovery procedure with version N do not need to upgrade.
# Therefore the package name contains the version and all packages conflict with each other
# to avoid that an installed version gets accidentally replaced with another version:
Provides:      rear = %{version}
Conflicts:     rear < %{version}
Conflicts:     rear > %{version}
Summary:       Relax-and-Recover (abbreviated rear) is a Linux Disaster Recovery framework
License:       GPL-2.0+
Group:         Productivity/Archiving/Backup
Url:           http://relax-and-recover.org/
# as GitHub stopped with download section we need to go back to Sourceforge for downloads
Source0:       http://sourceforge.net/projects/rear/files/rear/%{version}/rear-%{version}.tar.gz
# Source1 rear-release-notes.txt-1.16 was created by calling
# w3m -dump -cols 78 http://relax-and-recover.org/documentation/release-notes-1-16 >rear-release-notes.txt-1.16
# because rear-1.16.tar.gz does not contain an up to date doc/rear-release-notes.txt file:
Source1:       rear-release-notes.txt-1.16
# Source2 SLE12-btrfs-example.conf is a working example for SLE12 with default btrfs subvolume
# that gets installed as /usr/share/rear/conf/SLE12-btrfs-example.conf
```



# hidden '/@' btrfs subvolume

## SLE12 specific

- SLE12 includes **'/@' btrfs subvolume** that is made the default btrfs subvolume
- /sbin/btrfs subvolume create '/tmp/libstorage-9vKYd4/tmp-mp-loBzwl/@'
  - "btrfs subvolume list /"
- hides the '/@' in its output so that one must use
  - "btrfs subvolume list -a /"
  - "btrfs subvolume get-default /"
- to find out this hidden special '/@' btrfs subvolume
  
- The default btrfs subvolume gets mounted when no subvolume is specified for the mount command
  
- This means the /etc/fstab entry
  - "UUID=7ab751c6-b075-4066-9403-c225fdda91d6 / btrfs defaults 0 0"
- belongs to the default btrfs subvolume '/@' that is mounted at the mount point '/'

Demo

# Demo

1. Set up NFS share (on auxiliary machine)

```
# cat /etc/exports
```

```
/nfs/space/rear *(crossmnt,rw,no_root_squash,sync,no_subtree_check)
```

```
# rcnfsserver reload
```

2. Set up rear and make backup

```
# zypper in rear116*.rpm
```

```
# vi /etc/rear/local.conf
```

```
# rear mkrescue
```

```
# rear mkbackuponly
```

```
# halt/poweroff
```

3. Restore system

```
boot: rear
```

```
login: root
```

```
# rear recover
```

```
# ls /mnt/local/
```

```
# reboot
```

} rear mkbackup

```
Disaster recovery console
Recover and restore rear-test
Other actions
Boot First Local disk (hd0)
Boot Second Local disk (hd1)
Boot Next device
Hardware Detection Tool
ReBoot system
Power off system
Press [Tab] to edit, [F2] for help, [F1] for version info
```

```
Welcome to SUSE Linux Enterprise Server 11 SP2 (x86_64) - Kernel 3.0.13-0.27-default (tty1).
rear-test login: root
Welcome to Relax and Recover. Run "rear recover" to restore your system !
RESCUE rear-test:~ # rear recover_
```





# /etc/rear/local.conf

## example setup for SLE12

- SLE12-btrfs-example.conf from OBS package **home:jsmeix rear116**
- SLE12 with default btrfs subvolumes.
- /home/\* in backup or separate partition on xfs?
- Recovery of btrfs snapshot subvolumes **not** possible
- support database article at [http://en.opensuse.org/SDB:Disaster\\_Recovery](http://en.opensuse.org/SDB:Disaster_Recovery)
  
- OUTPUT=ISO
- BACKUP=NETFS
- BACKUP\_OPTIONS="nfsvers=3,nolock"
- BACKUP\_URL=nfs://<nfsserver>/directory
- BACKUP\_PROG\_INCLUDE=( '/home/\*' '/var/spool/\*' '/var/opt/\*' '/var/log/\*' '/var/lib/pgsql/\*' '/var/lib/mailman/\*' '/var/lib/named/\*' '/usr/local/\*' '/srv/\*' '/boot/grub2/x86\_64-efi/\*' '/opt/\*' '/boot/grub2/i386-pc/\*' )
- EXCLUDE\_RECREATE=( "\${EXCLUDE\_RECREATE[@]}" "fs:/home" "fs:/.snapshots" "fs:/var/tmp" "fs:/var/spool" "fs:/var/opt" "fs:/var/log" "fs:/var/lib/pgsql" "fs:/var/lib/mailman" "fs:/var/lib/named" "fs:/usr/local" "fs:/tmp" "fs:/srv" "fs:/var/crash" "fs:/boot/grub2/x86\_64-efi" "fs:/opt" "fs:/boot/grub2/i386-pc" )

rear-SUSE

# Disaster Recovery with AutoYaST together with a ReaR NETFS backup

- Workflow:
  - Script **RecoveryImage** creates a bootable ISO image to recover this particular system
- RecoveryImage
  - Run 'rear mkbackuponly' on NFS server
  - Run AutoYaST clone\_system.ycp → autoinst.xml
  - Make a bootable system recovery ISO image based on install medium
  - 'chroot script' is added to autoinst.xml to restore backup

# RecoveryImage Workflow

- Recovery medium(from ISO) runs AutoYaST to recreate basic system
  - Partitioning, mountpoints
- AutoYaST runs 'chroot script' to fill in backup data
  - recreated system mountpoints still below /mnt
  - Install bootloader after restore inside chroot
- Boot first time
- Autoyast system configuration(network)
- Switch to final runlevel

# rear-SUSE:

## Restrictions And Shortcomings

- Only supported with SLES11 SP3
- rear BACKUP\_URI of the form 'nfs://host/path/file.tar.gz'
- All parameters required
  - RecoveryImage [ -d BASE\_URI
  - -l { log-to-base-dir | LOG\_DIR }
  - -b { make-rear-backup | use-existing-rear-backup | BACKUP\_URI }
  - -a { clone-system | AUTOINST\_FILE | use-autoinst-from-base-dir }
  - -m { autodetect-dvd | MEDIUM\_URI | use-existing-medium-ISO
  - | use-existing-ISO-files }
  - -i { install-RPMs | skip-RPM-install | no-RPM-payload }
  - -r { restore-all | restore-exclude-default | RESTORE\_EXCLUDE }
  - -c { configure-all | CONFIGURE\_EXCLUDE | skip-second-stage } ]

# SUSE Consulting and rear

- SUSE Consulting had partner contracts with two of the core developers, so we have easily access to rear premium consulting services.
- SUSE Consulting had done successfully customer projects enhancing rear with missing features like
  - Adding Legato Networker to rear
  - Adding support for VLAN tagging
  - Proof of Concepts
- SUSE Consulting tries to add the enhancements upstream (git and SLE HAE), so every customer can benefit from it and get a supported solution.

# rear Keywords

- Enterprise Solution
  - Built for enterprise customers – fully automated
- Open Source (GPL)
  - Source code included (100% bash)
- Modular Design
  - Easily extendible with new functions
- Scalability
  - One solution for all Linux systems, unlimited scale-out
- Usability
  - Documentation, community & commercial support



# More Info Resources

## URLs

<http://www.suse.com/products/server/>

[http://www.suse.com/releasenotes/x86\\_64/SUSE-SLES/11-SP3/](http://www.suse.com/releasenotes/x86_64/SUSE-SLES/11-SP3/)

<http://www.suse.com/products/highavailability>

[http://www.suse.com/releasenotes/x86\\_64/SLE-HA/11-SP3/](http://www.suse.com/releasenotes/x86_64/SLE-HA/11-SP3/)

[http://www.suse.com/documentation/sle\\_ha/](http://www.suse.com/documentation/sle_ha/)

[https://www.suse.com/documentation/sle\\_ha/singlehtml/book\\_sleha/book\\_sleha.html#cha.ha.rear](https://www.suse.com/documentation/sle_ha/singlehtml/book_sleha/book_sleha.html#cha.ha.rear)

[www.suse.com/products/highavailability/geo-clustering/](http://www.suse.com/products/highavailability/geo-clustering/)

[www.suse.com/documentation/sles11/](http://www.suse.com/documentation/sles11/)

<http://relax-and-recover.org/>

## FATE entries for ReaR

- |        |  |
|--------|--|
| 316651 | ReaR to support BTRFS snapshots                              |
| 316508 | ReaR update to latest stable upstream                        |
| 315057 | Relax-and-Recover (formerly ReaR): Support for EMC NetWorker |
| 314698 | ReaR to support BTRFS subvolumes                             |
| 317032 | Relax-and-Recover (formerly ReaR) - Add VLAN tagging support |





# Appendix

# Snapshot/Rollback – Overview

## Past & Present

- “snapper undochange”
- Selective Rollback for
  - Package updates
  - Administrative changes
- No rollback of
  - Kernel / initrd
  - Bootloader
  - System data, e.g. /var/log

High Demand

## Present & Future

- “snapper rollback”
- Full Rollback for
  - Package updates
  - Administrative changes
  - **Kernel / initrd (initramfs)**
- No rollback of
  - Bootloader
  - Customer data: “/home”, if on own partition (default)
  - System data, e.g. /var/log

# Snapshotting “/” – Challenges

- Kernel and initrd / initramfs = “/boot”
  - Grub2 booting from a snapshot = subvolume
  - Mark snapshots with /boot relevance as such
- System integrity and Compliance
  - Don't allow to roll back certain log-files etc.
  - Solution: subvolumes instead of directories for
    - /tmp
    - /opt
    - /srv
    - /var/spool
    - /var/log
    - /var/run
    - /var/tmp
    - ...

# Btrfs: Root filesystem

- According to <https://btrfs.wiki.kernel.org/index.php/UseCases>
- How do I mount the real root of the filesystem once I've made another subvolume the default?
- `#mount -o subvolid=0 <filesystem> <mount-point>`
- With kernel 3.2 and newer you can specify `subvol=/some/PATH` for the subvolume to mount
- `#mount -o subvol=/path/to/subvol /dev/sdx /mnt`
- The PATH is always relative to the toplevel subvolume, ie. independent of currently set default subvolume.

# Btrfs subvolume mounting

```
- # root@sle12 # ls /
- # bin boot dev etc home lib lib64 media mnt opt proc root run sbin selinux .snapshots
  srv sys tmp usr var
- # root@sle12 # mkdir /tmp/btrfs-filesystem-toplevel
- # root@sle12 # mount -t btrfs -o subvolid=0 /dev/sda2 /tmp/btrfs-filesystem-toplevel
- # root@sle12 # ls /tmp/btrfs-filesystem-toplevel
- # @
- # root@sle12 # ls /tmp/btrfs-filesystem-toplevel/@
- # bin boot dev etc home lib lib64 media mnt opt proc root run sbin selinux .snapshots
  srv sys tmp usr var
- # root@sle12 # ls /home
- # johannes
- # root@sle12 # ls /tmp/btrfs-filesystem-toplevel/home
- # ls: cannot access /tmp/btrfs-filesystem-toplevel/home: No such file or directory
- # root@sle12 # ls /tmp/btrfs-filesystem-toplevel/@/home
- # johannes
```



**Corporate Headquarters**  
Maxfeldstrasse 5  
90409 Nuremberg  
Germany

+49 911 740 53 0 (Worldwide)  
[www.suse.com](http://www.suse.com)

Join us on:  
[www.opensuse.org](http://www.opensuse.org)

## **Unpublished Work of SUSE. All Rights Reserved.**

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

## **General Disclaimer**

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

