Before We Start ...

Isn't One Type of Car Enough?

- Functionality
- Efficiency
- Performance
- Emotions

Reliability?

- Functionality
- Efficiency
- Performance
- Emotions

Isn't One Type of filesystem Enough?

- Functionality
- Efficiency
- Performance
- Reliability
- Charged with Emotions

How To Choose a Filesystem

Matthias G. Eckermann
Senior Product Manager
mge@suse.com

Jeff Mahoney
Kernel Developer
Team Lead: File Systems and Storage
jeffm@suse.com

Agenda

- Local Linux Filesystems
- Tuning on the filesystem layer
- Tuning on the block layer
- A few more words on btrfs
  - Copy on Write – and what to do with it
  - Conversion to btrfs

Major Linux (local) Filesystems

Feature	ext 2/3	reiserfs	xfs	ext4	btrfs
Data/Metadata Journaling	•/•	○/•	○/•	•/•	CoW
Journal internal/external	•/•	•/•	•/•	•/•	CoW
Offline extend/shrink	•/•	•/•	○/○	•/•	•/•
Online extend/shrink	•/○	•/○	•/○	•/○	•/•
Inode-Allocation-Map	table	u.B*-tree	B+-tree	table	B-tree
Sparse Files	•	•	•	•	•
Tail Packing	○	•	○	○	•
Defrag	○	○	•	•	•
ExtAttr / ACLs	•/•	•/•	•/•	•/•	•/•
Quotas	•	•	•	•	Subvol.
max. Filesystemsize	16 TiB	16 TiB	8 EiB	1 EiB	16 EiB
max. Filesize	2 TiB	1 EiB	8 EiB	1 EiB	16 EiB

Example – Filesizes

Dataset #1
Documents & Mails

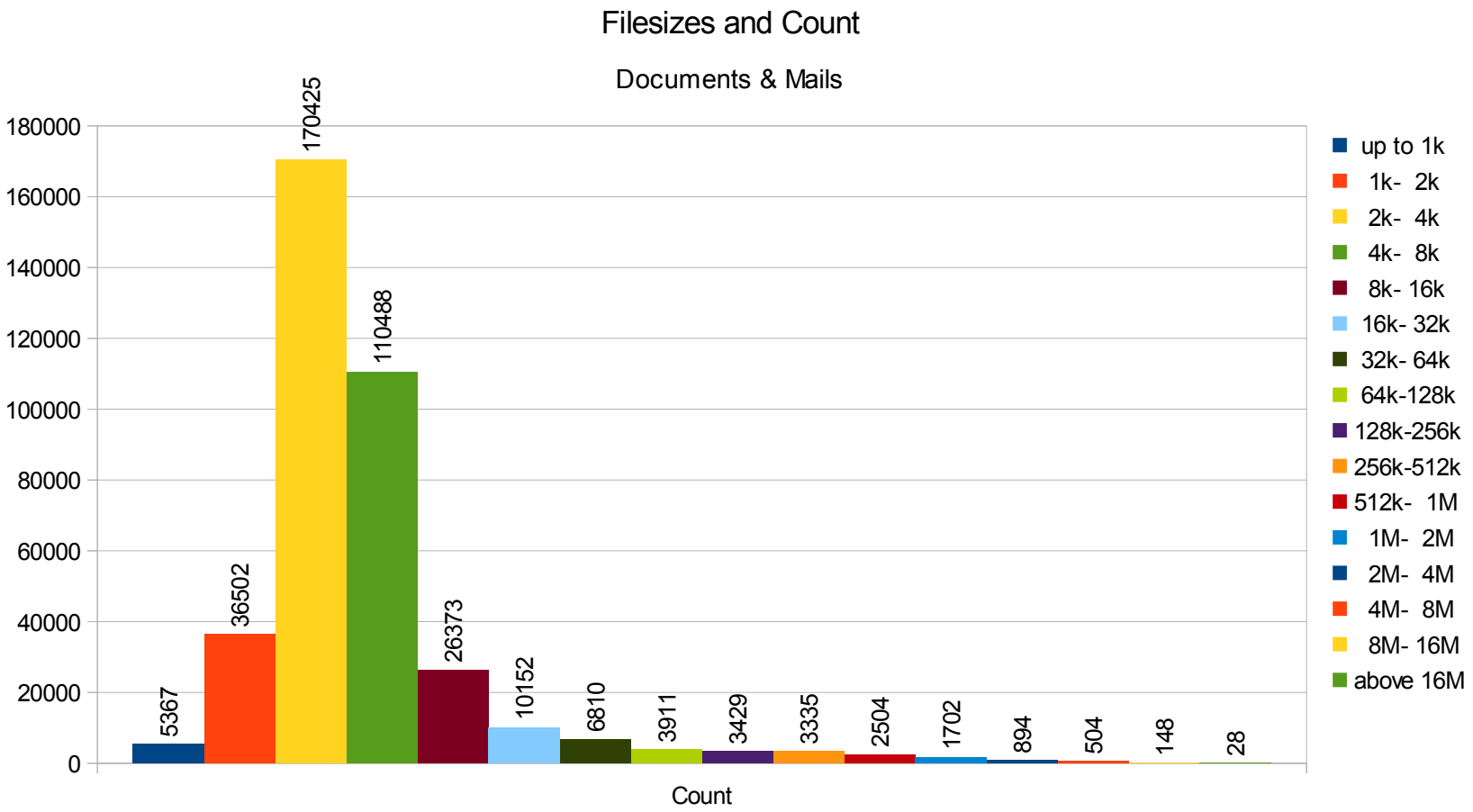
Filesystem Space Efficiency

Dataset#1 – Test parameters

- Partition size: 24626 MiB, SSD
- Data type: E-Mails and Documents

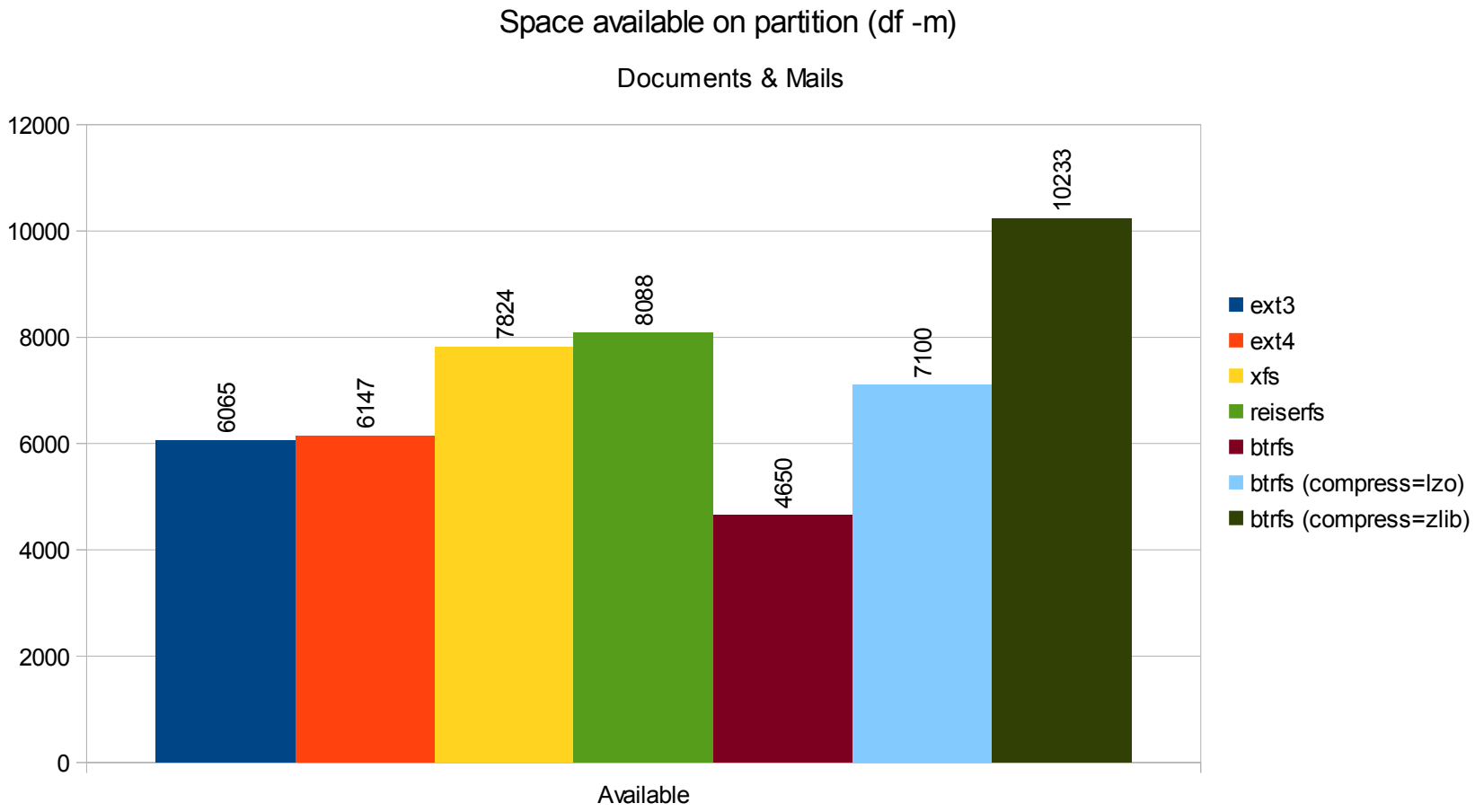
Filesystem Space Efficiency

Dataset#1 – Data typology



Filesystem Space Efficiency

Dataset#1 – Result



Dataset #2
ISOs, RPMs, various

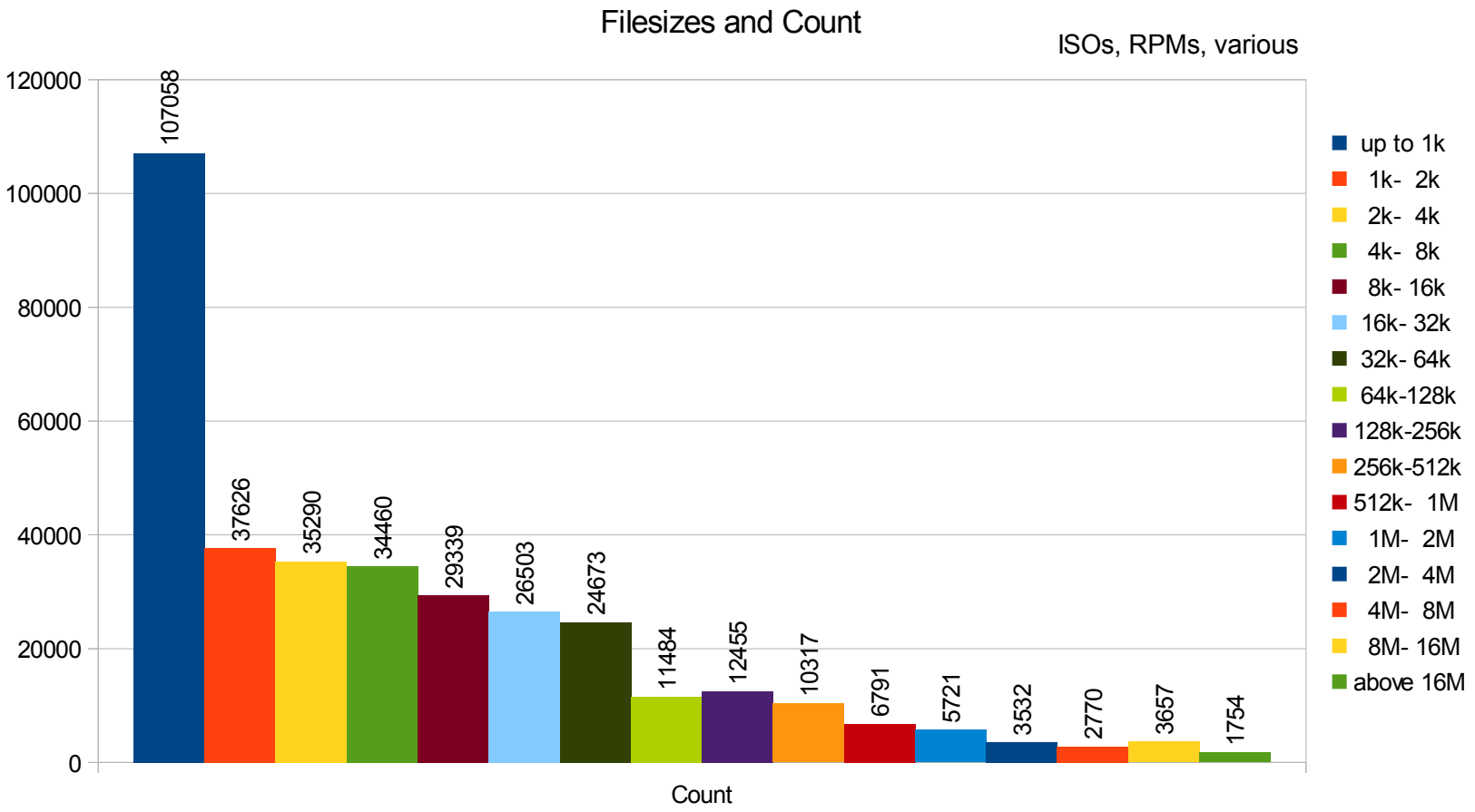
Filesystem Space Efficiency

Dataset#2 – Test parameters

- Partition size: 303884 MiB, HDD
- Data types
  - ISOs (~100 GiB)
  - RPMs(~50 GiB)
  - VMs (~25 GiB)
  - Pictures and Sounds (~20 GiB)
  - Third party software including TeXLive (~16 GiB)
  - /usr/src (including Kernel sources) (~16 GiB)
- Tested on xfs and btrfs only

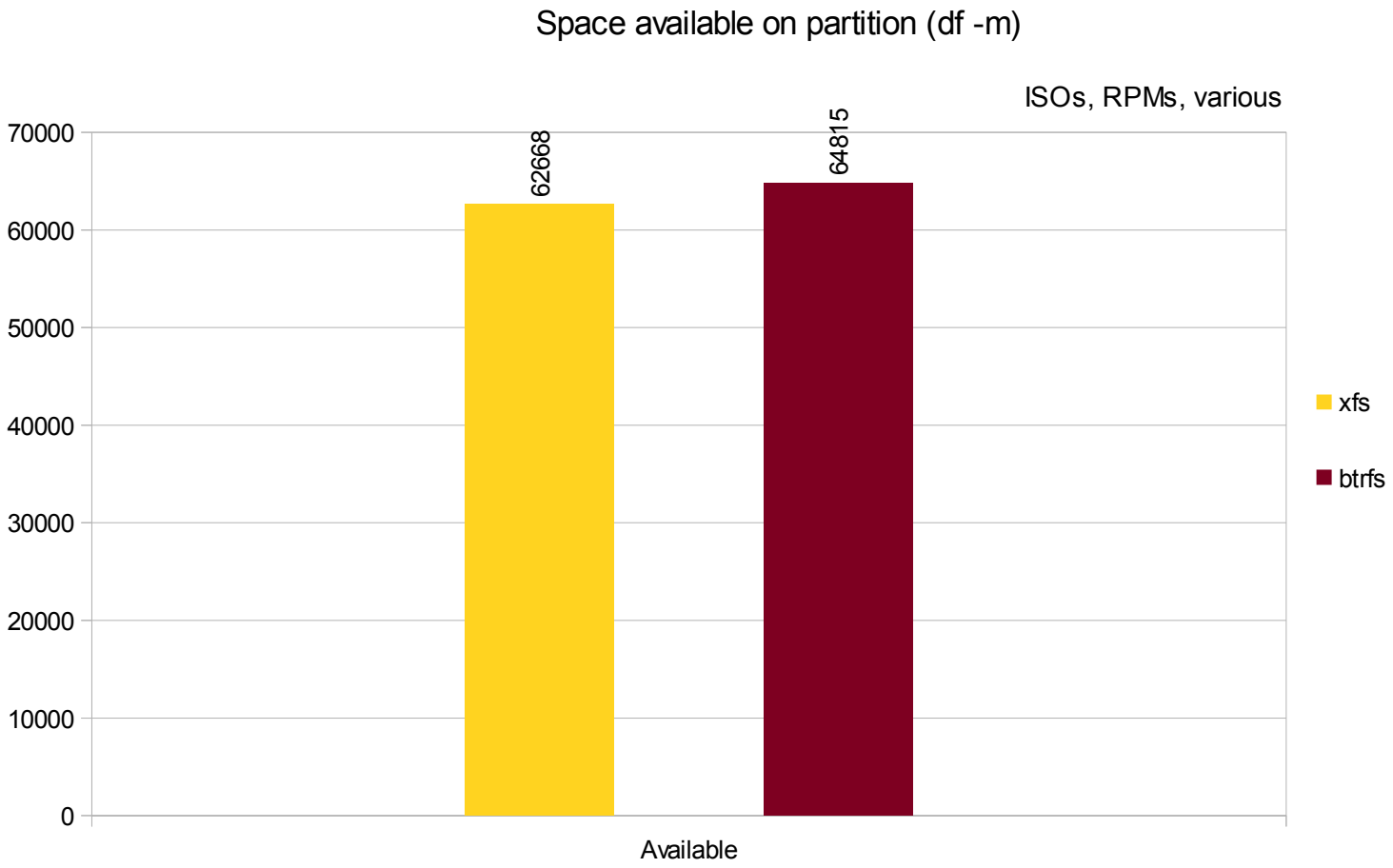
Filesystem Space Efficiency

Dataset#2 – Data typology



Filesystem Space Efficiency

Dataset#2 – Result



Result?

Linux (Local) Filesystems

Major Linux (local) Filesystems

Feature	ext 2/3	reiserfs	xfs	ext4	btrfs
Data/Metadata Journaling	•/•	○/•	○/•	•/•	CoW
Journal internal/external	•/•	•/•	•/•	•/•	CoW
Offline extend/shrink	•/•	•/•	○/○	•/•	•/•
Online extend/shrink	•/○	•/○	•/○	•/○	•/•
Inode-Allocation-Map	table	u.B*-tree	B+-tree	table	B-tree
Sparse Files	•	•	•	•	•
Tail Packing	○	•	○	○	•
Defrag	○	○	•	•	•
ExtAttr / ACLs	•/•	•/•	•/•	•/•	•/•
Quotas	•	•	•	•	Subvol.
max. Filesystemsize	16 TiB	16 TiB	8 EiB	1 EiB	16 EiB
max. Filesize	2 TiB	1 EiB	8 EiB	1 EiB	16 EiB

Picking a File System

Pick the right file system for the task

- Indexed metadata
- File sizes
- Number of files
- Workloads (database, mail server,...)
- AccessPaths
- Dump/Restore

File Systems: ReiserFS

Applications that use many small files

- Mail servers
- NFS servers
- Database servers

or other applications that use synchronous I/O

Mature file system in maintenance mode.

File Systems: Ext-Family

ext3 is default file system in SUSE® Linux Enterprise 11

ext4 is default in other distributions

Primary disadvantage

- Limitations in scaling by size
- Recommended sizes:
  - ext3 < 01 TiB
  - ext4 < 16 TiB

File Systems: XFS

Best suited for

- Medium (>100GiB) to very large file systems (> 100 TiB)
- Large files/many files
- Streaming multimedia (low latencies)

Special features and capabilities

- dump/restore
- online filesystem-check
- online-defragmentation

File Systems: XFS – Advantages

Maturity

- comes from IRIX
- ported to Linux > 10 years ago

Track record for

- Performance
- Scalability
- Stability

Active Development community

- Checksums
- Self-identifying metadata

Filesystems: btrfs

Features

- Copy on Write
- Extents
- Snapshots

Concepts

- B-Tree
- Subvolume
- Metadata
- Raw data

Filesystems: btrfs – Copy on Write

Disadvantages	Advantages
Performance impact on *specific* workloads, such as storing VMs	Efficient Storage Deduplication Snapshots Integrity beyond Journalling

Filesystems: btrfs – Maturity

Mature / Supported	Not (yet) mature
Copy on Write	Inode Cache
Snapshots	Auto Defrag
Subvolumes	RAID
Metadata Integrity	Compression
Data Integrity	Send / Receive
Online metadata scrubbing	Hot add / remove
Manual Defragmentation	Seeding devices
Manual Deduplication	Multiple Devices
Quota Groups	“Big” Metadata

Cluster File System: OCFS2

OCFS2 (Oracle Cluster File System)

- Shared access by multiple nodes
  - Ensures data integrity in case of a node-failure
  - Scale-out for data access
- Generic use
  - POSIX-compliant
  - Cluster-aware POSIX locking
- Higher throughput
  - Parallel I/O
- Disaster Tolerance
  - Integration with data replication for dual node

Tuning On The filesystem Layer

Dedicated Logging Devices

ReiserFS

```
mkreiserfs -j /dev/xxx -s 8192 /dev/xxy
reiserfstune –journal-new-device /dev/xxx -s 8192
```

Ext3/4

```
mke2fs -O journal_dev /dev/xxx
mke2fs -j -J device=/dev/xxx,size=8192 /dev/xxy
tune2fs -J device=/dev/xxx,size=8192 /dev/xxy
```

XFS

```
mkfs.xfs -l logdev=/dev/xxx,size=10000b /dev/xxy
```

File System Tuning

Split file systems based on data access patterns

- Keep commit heavy data away from data that does not have to be synchronous
- Keep streaming writes and reads on different spindles than random I/O

Consider disabling atime updates on files and directories

```
# mount -o noatime,nodiratime
```

Tuning On The Block Layer

I/O Scheduler

Flexible, pluggable I/O scheduler

Selectable via boot parameter elevator=<X>

- noop
- deadline
- as (default in mainline kernels)
- cfq (default in SUSE® Linux Enterprise)

I/O Scheduler per device

- Check

  /sys/block/*DEV*/queue/iosched

- Set

  echo SCHEDNAME > /sys/block/*DEV*/queue/scheduler

I/O Scheduler: Noop

No reordering, just merging

Best for storage with extensive caching and scheduling of its own, such as:

MultiPathing

Virtual Machines

on SSDs

Activated by boot parameter elevator=noop

I/O Scheduler: Deadline

Per-request service deadline

- Caps maximum latency per request
- Maintains good disk throughput

Best for disk-intensive database applications

Activated by boot parameter elevator=deadline

I/O Scheduler: CFQ

Complete Fair Queuing

Treat all competing processes equally by keeping a unique request queue for each and giving equal bandwidth to each queue

- Good compromise between throughput and latency
- Minimal worst case latency on all reads and writes

Suitable for a wide variety of applications

Default in SUSE® Linux Enterprise

Activated by boot parameter elevator=cfq

Block Layer Tuning

Spreading the load across controllers

- Per-target locking for SCSI
- Software RAID bandwidth

Battery backed caching

Blocker Layer Tunables

Block read ahead buffer

/sys/block/<sdX/hdX>/queue/read_ahead_kb

Default is 128. Increase to 512 for fast storage (SCSI disks or RAID)

May speed up streaming reads a lot

Number of requests

/sys/block/<sdX/hdX>/queue/nr_requests

Default is 128. Increase to 256 with CFQ scheduler for fast storage

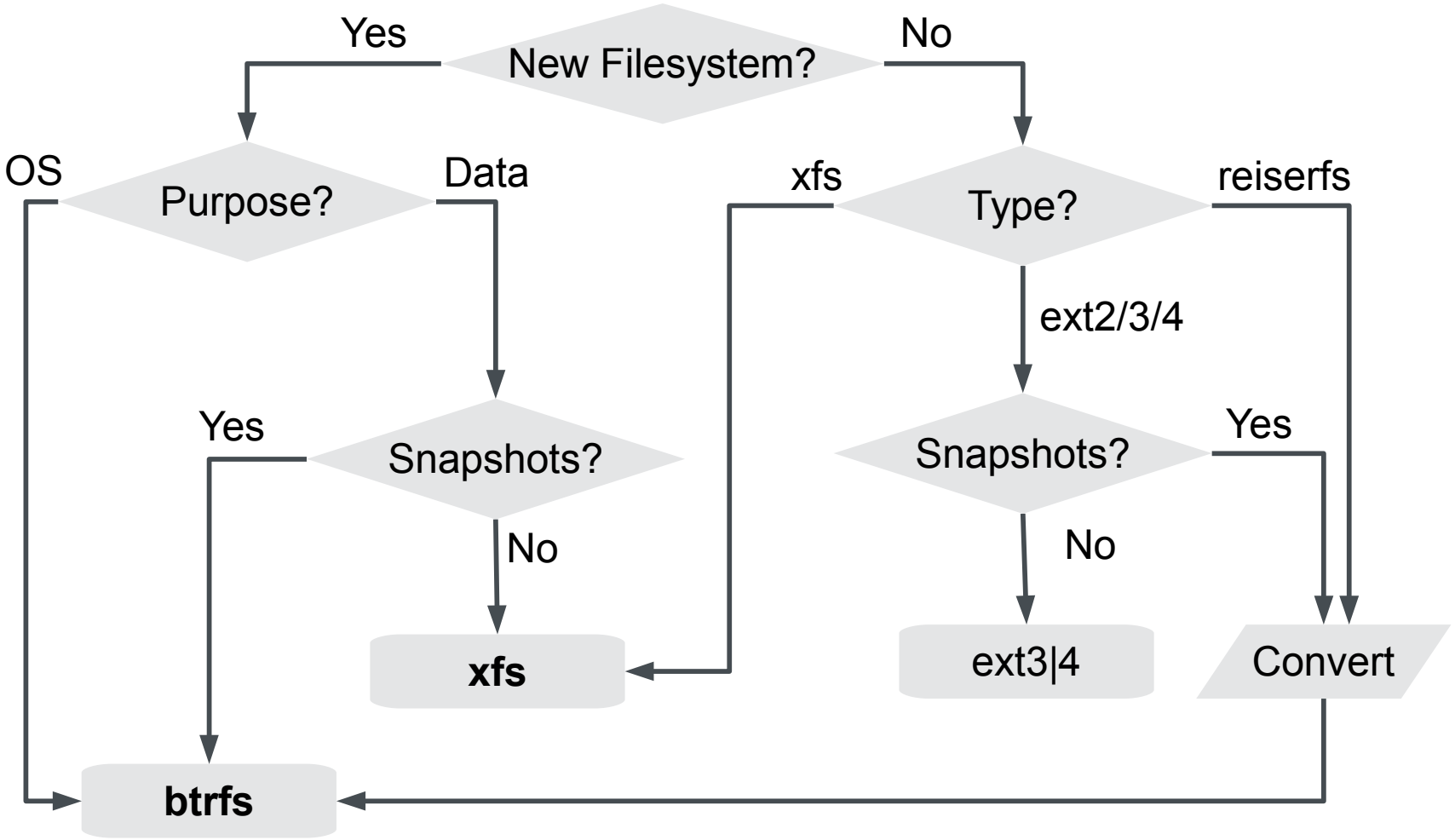
Increases throughput at minor latency expense

Linux (Local) Filesystems

Major Linux (local) Filesystems

Feature	ext 2/3	reiserfs	xfs	ext4	btrfs
Data/Metadata Journaling	•/•	•/•	○/•	•/•	CoW
Journal internal/external	•/•	•/•	•/•	•/•	CoW
Offline extend/shrink	•/•	•/•	○/○	•/•	•/•
Online extend/shrink	•/○	•/○	•/○	•/○	•/•
Inode-Allocation-Map	table	u.B*-tree	B+-tree	table	B-tree
Sparse Files	•	•	•	•	•
Tail Packing	○	•	○	○	•
Defrag	○	○	•	•	•
ExtAttr / ACLs	•/•	•/•	•/•	•/•	•/•
Quotas	•	•	•	•	Subvol.
max. Filesystemsize	16 TiB	16 TiB	8 EiB	1 EiB	16 EiB
max. Filesize	2 TiB	1 EiB	8 EiB	1 EiB	16 EiB

Filesystem Recommendations



Copy On Write – And What To Do With It

Using “snapper” To Manage Operating System Activities

What Is Snapper?

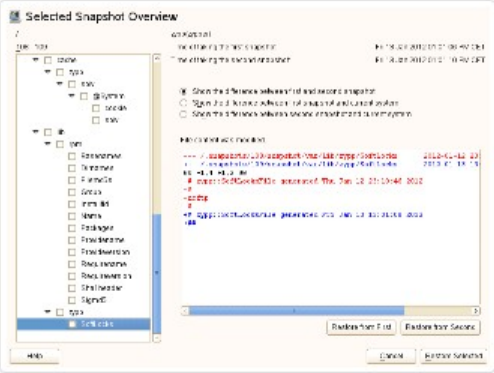
- Tool to manage btrfs snapshots
- Functions:
  - create, modify, delete
  - status (=compare), diff
  - undochange
  - cleanup
- Integration with
  - Package management stack (e.g. yum, zypper)
  - Systems management stack (SUSE: YaST)
- DBUS service

Snapper Home Tutorial Documentation FAQ

Travel back in time and compare...

The ultimate snapshot tool for Linux

Download »



Watch it in action

Greg Kroah-Hartman and Matthias Eckermann play sysadmins and screw a web server configuration.

Arvin Schnell (lead developer) at FOSDEM 2012

Contribute

Snapper is opensource. Port it to your distribution or integrate it with an application.

Fork us on github »

Tweet

Thanks to Snapper, you can mess up system configuration changes or package installations or updates without having to restore from an old backup and risking to lose some files. Just revert to the snapshot before your problematic change and you're fine. *Linux User & Developer Magazine*

http://www.snapper.io/

Snapshotting On The Desktop

/home/$USER

Using Snapper For User Data

Requirements

- /home/$USER is a btrfs subvolume
- “snapper” with DBUS interface
- Snapper configuration per user

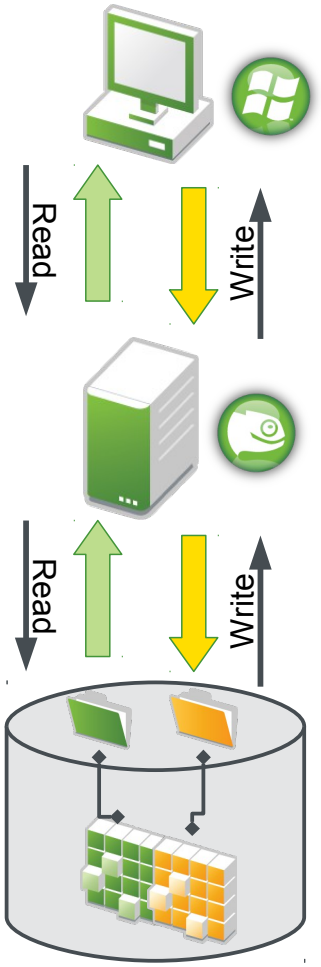
Additional options

- Automated snapshotting on login/logout
  - Requires pam-snapper
- Automated snapshotting on Suspend

Server Side Snapshots

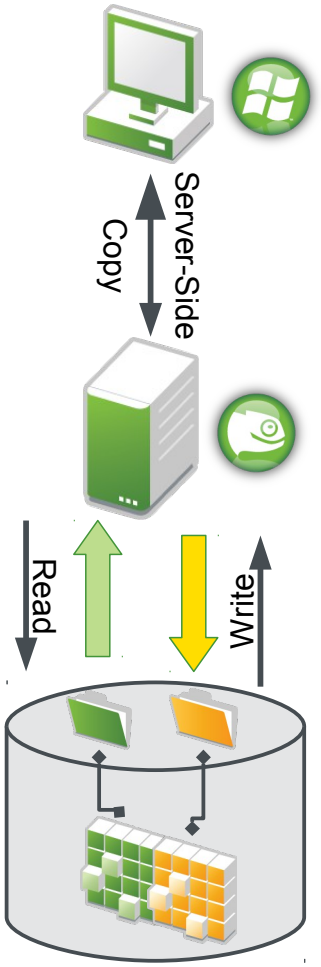
Btrfs as a Samba Backend Server Side Copy

Traditional File Copy



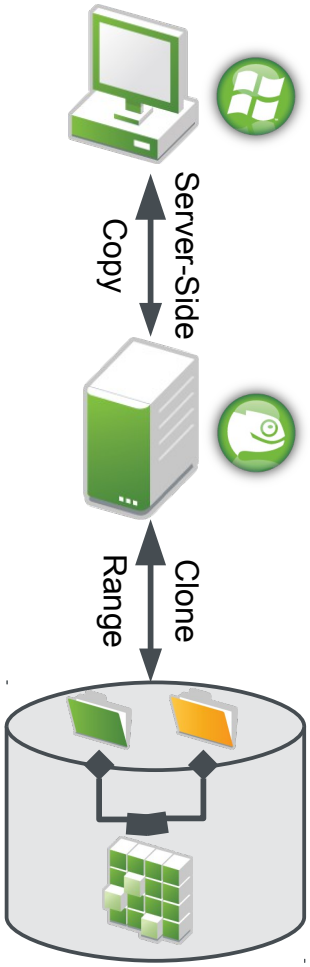
- File data takes disk and network round-trips
- Duplicate data stored on disk

Server-Side Copy



- Network round-trip avoided
- Server copies file data locally
- Duplicate data stored on disk

Btrfs Enhanced Server-Side Copy



- Data avoids network **and** disk round-trips
- No duplication of file data

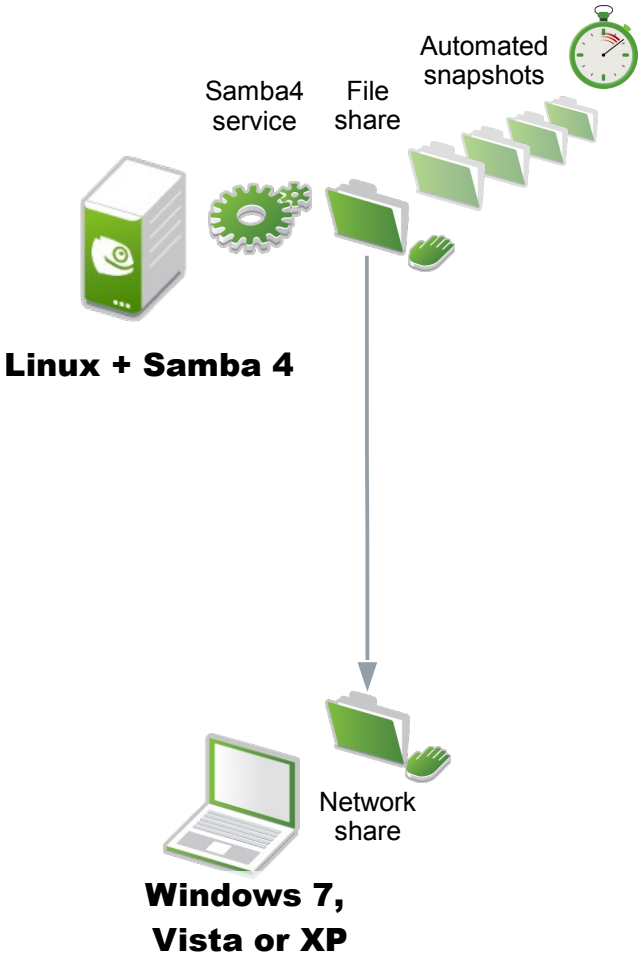
Server Side Snapshots

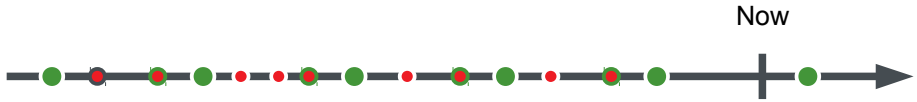
Btrfs as a Samba Backend "Recovery Point"

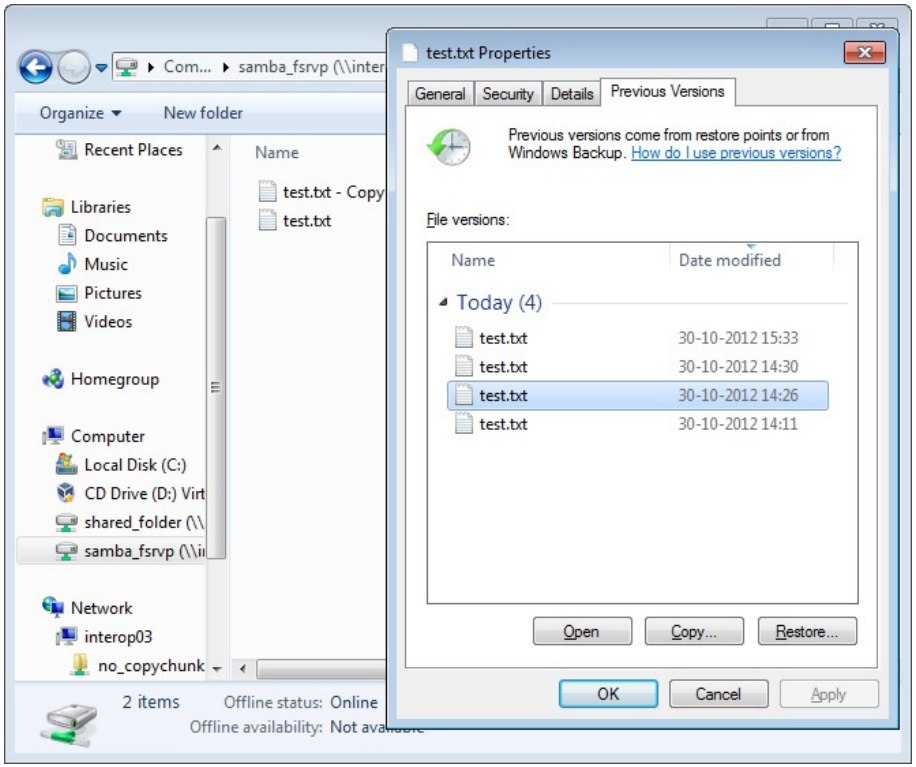
Samba4 and btrfs, snapper

Prototype Samba Implementation of “Recovery Point”

- Automatic snapshots by Snapper
- Previous versions of “test.txt” in Explorer
- File “test.txt” is changed
- File “test.txt” is created







Other Features – Future

Conversion to btrfs

- btrfs-convert
- offline in-place migration from
  - ext2/3/4

    and
  - reiserfs
- Keeps metadata of the old filesystem for a roll-back

demonstration: convert reiserfs to btrfs

Continuously Running Systems

Snapshot / Rollback for full system – Based on

- btrfs
- Snapper
- Bootloader integration
  - Booting directly from a btrfs snapshot
  - Jump back to a former status of the OS, including kernel / initrd

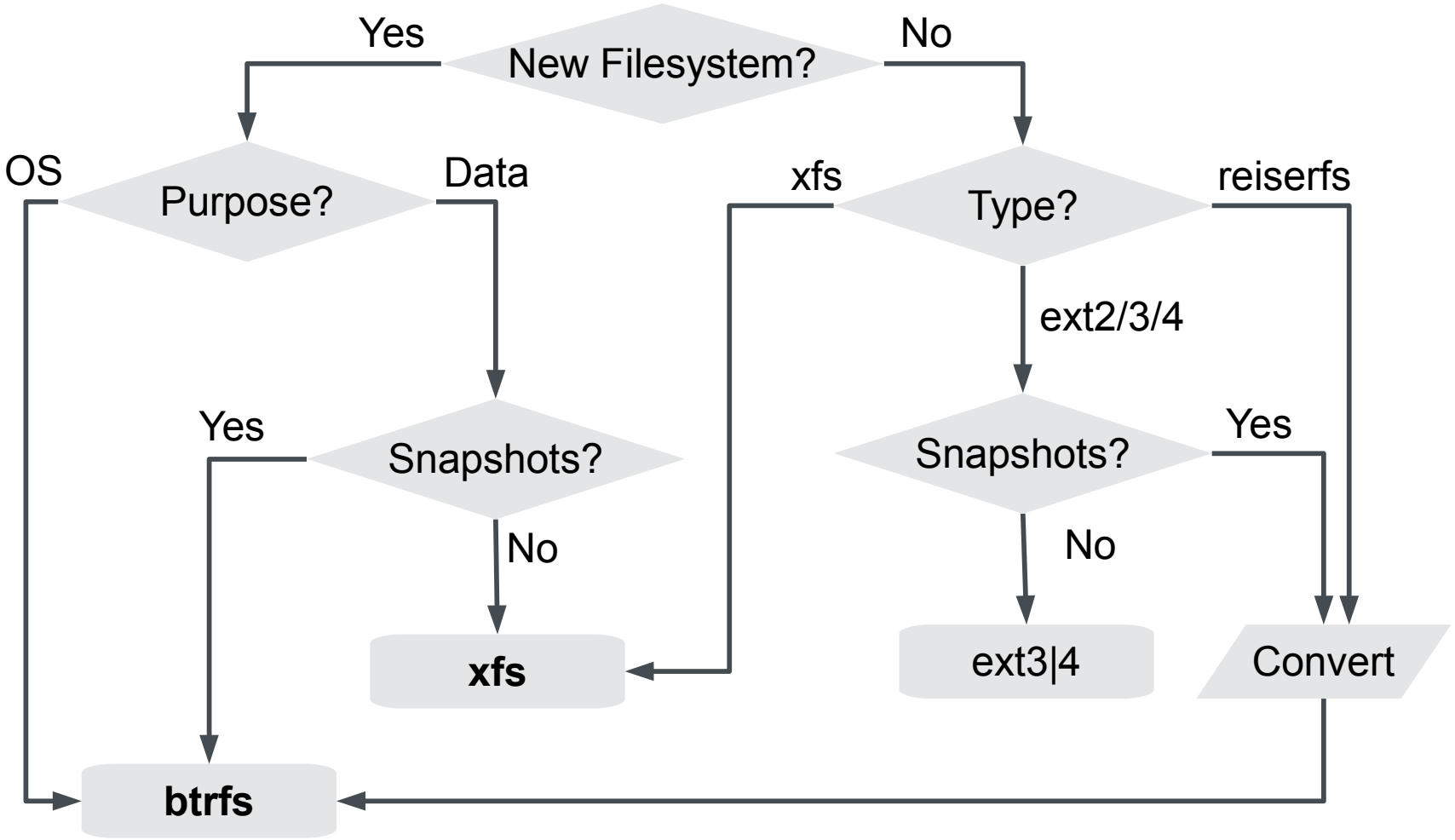
Btrfs – Planned Features

- Data de-duplication:
  - De-duplication during writes
  - Manual De-duplication
- Tiered storage
  - e.g.: combine SSD and HDD

Upstream since 2013-09-12

Summary

Filesystem Recommendations



Go ahead, try xfs, btrfs,
and ocfs2 today!
Start Tuning!
Your questions!?

Thank you.

Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org