

Btrfs

Current Status and Future Prospects

Oct 13 2014
Satoru Takeuchi
Linux Development Div.
Fujitsu LTD.

Agenda

- Background
- Core Features
- Developments statistics
- Future Prospects

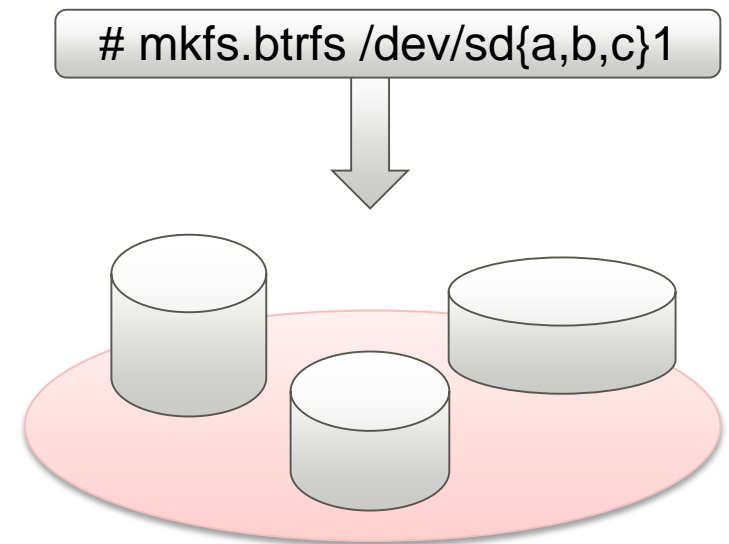
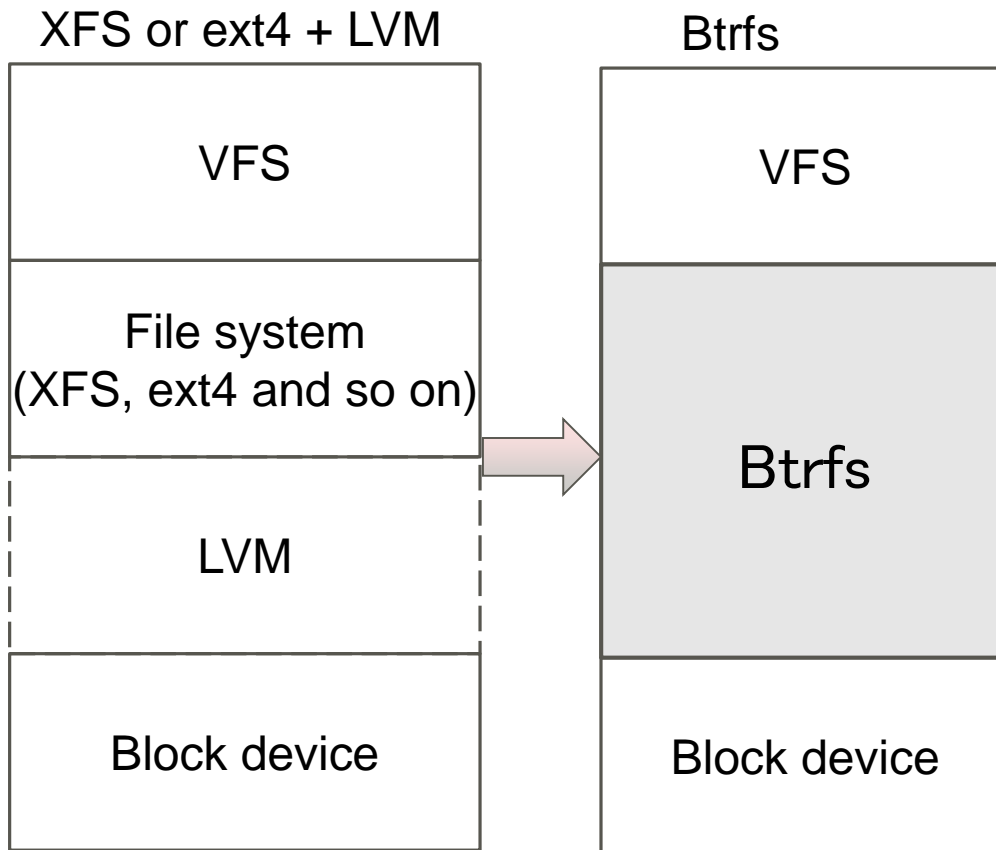
- **Background**
- Core Features
- Developments statistics
- Future Prospects

- Fujitsu has developed Btrfs for Mission Critical (MC) systems since 2010
- Requirements of MC systems
 - High robustness
 - Don't crash: data duplication
 - Error detection: checksum
 - Repair, recovery: snapshot, backup/restore, repairing tools
 - High availability: Should work 365days/24h
 - Limited maintenance time: enlarge storage size and backup online
- Btrfs is designed for such the requirements

- Background
- **Core Features**
- Developments statistics
- Future Prospects

- Multi-volumes
- Copy-on-Write Style Update
- Data/Metadata Checksum
- Subvolume
- Snapshot
- Transparent Compression

- Btrfs file system can consists of multiple volumes
 - Low layered and low overhead than LVM
 - Many features: RAID, online {add/remove/replace} devices

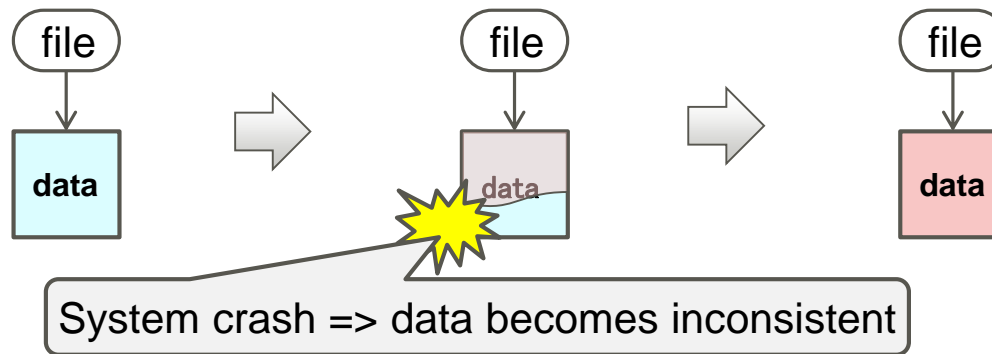


Copy-on-Write(CoW) style update

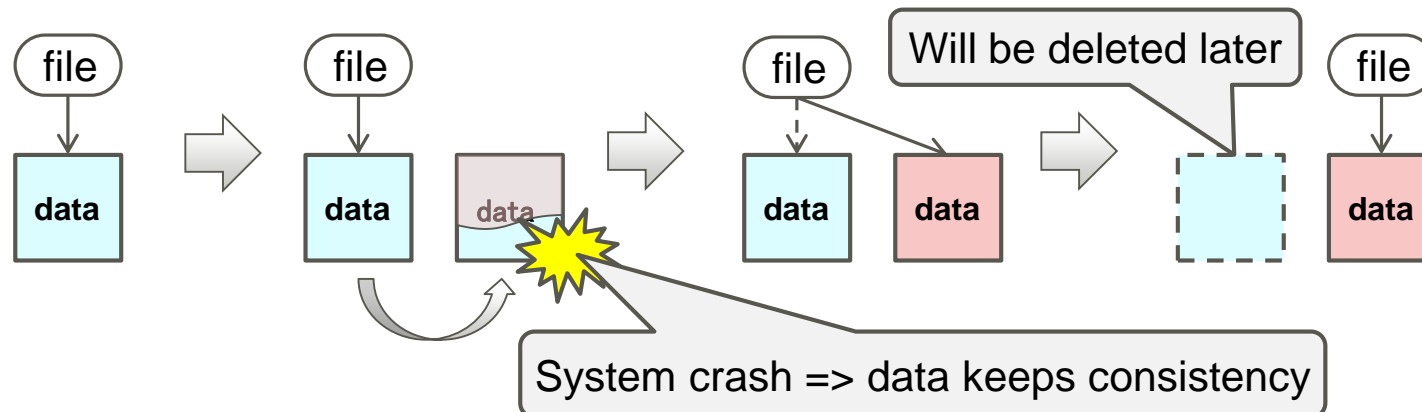
■ Btrfs uses CoW style data/metadata update

- Safer than overwrite style update by design

■ Overwrite style: Update the data in place



■ CoW style: Copy, update, and replace pointer



■ 1,000 surprising power failure test

- Linux File System Analysis for IVI system, Mitsuharu Ito, Fujitsu

http://events.linuxfoundation.jp/sites/events/files/slides/linux_file_system_analysis_for_IVI_systems.pdf

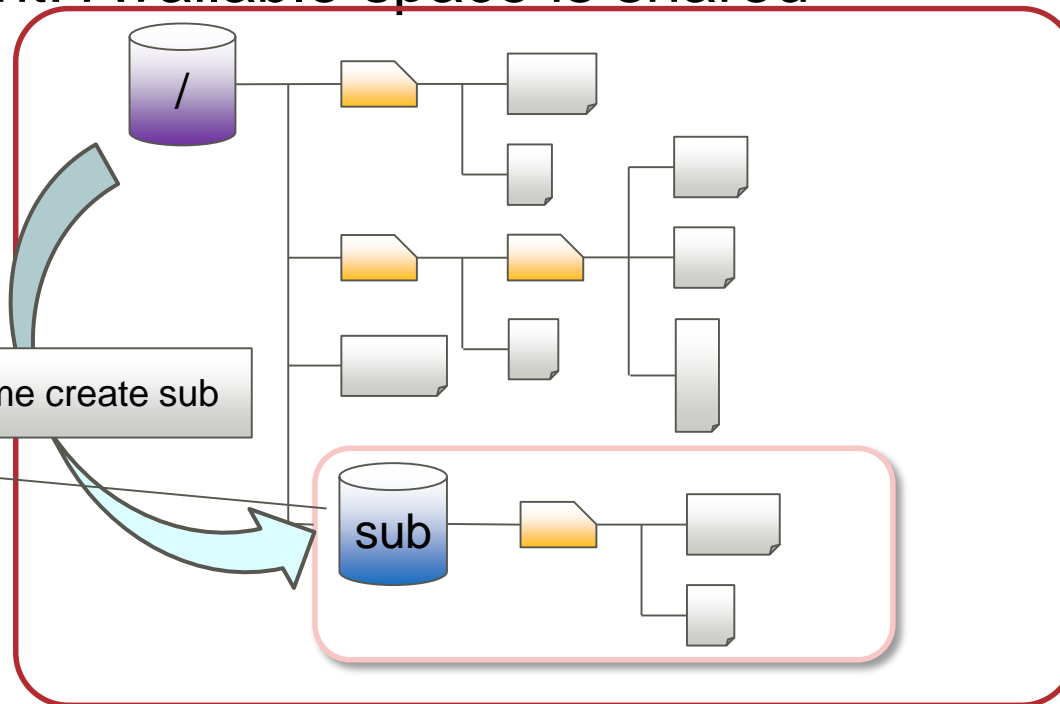
■ Result

- Ext4: Metadata was corrupted
- Btrfs: Worked fine without any problem

■ In my internal similar testing, XFS corrupted too.

- Btrfs has checksum for each data/metadata extent to detect and repair the broken data
- When Btrfs reads a broken extent, it detects checksum inconsistency
 - With mirroring: RAID1/RAID10
 - Read a correct copy
 - Repair a broken extent with a correct copy
 - Without mirroring
 - Dispose a broken extent and return EIO
- With “btrfs scrub”, Btrfs traverses all extents and fix incorrect ones
 - Online background job

- A subvolume is a file system inside file system
 - Can be treated as a file system root
 - Mountable: most mount options are shared
 - Own inode namespace and quota limit
 - Efficient: Available space is shared



Snapshot

■ Copy of a subvolume

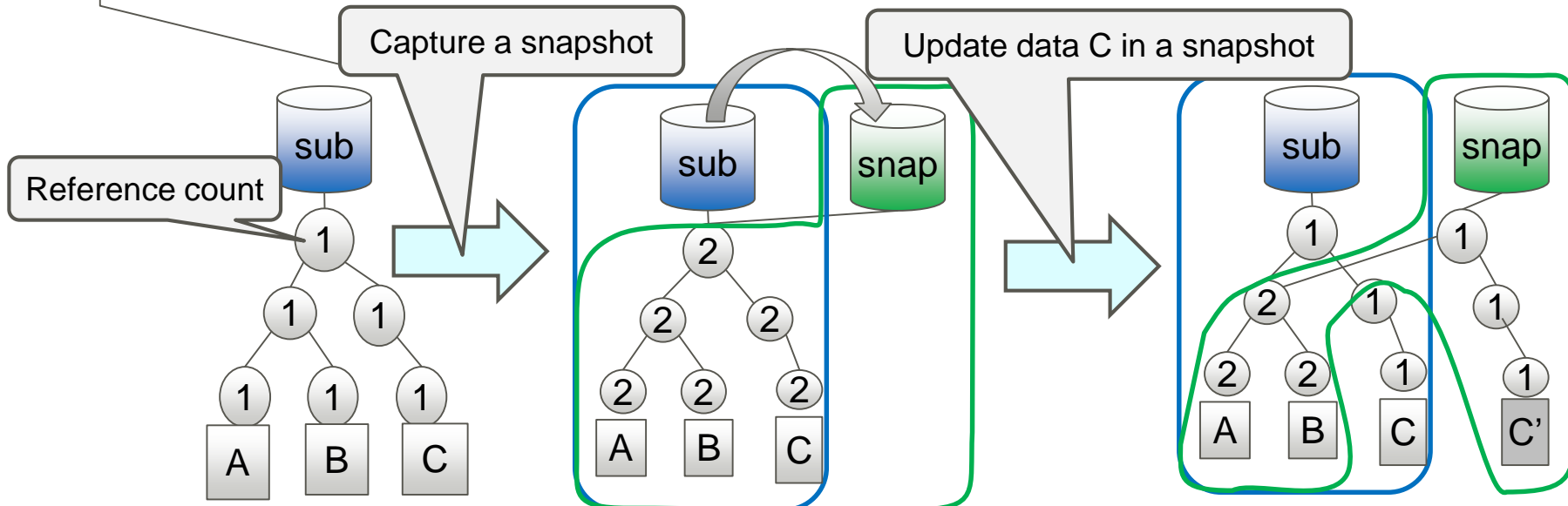
■ Far faster than LVM

- Not a full copy, but only update metadata in CoW style

■ Readonly snapshot: with `-r` option

■ Incremental snapshot: snapshot of snapshot

```
# btrfs subvolume snapshot [-r] ./sub ./snap
```



1. Copy the following data to a volume

- Consists of 100 directories and 100 files for each directory
 - File size: 1MB

2. Capture a snapshot of the volume

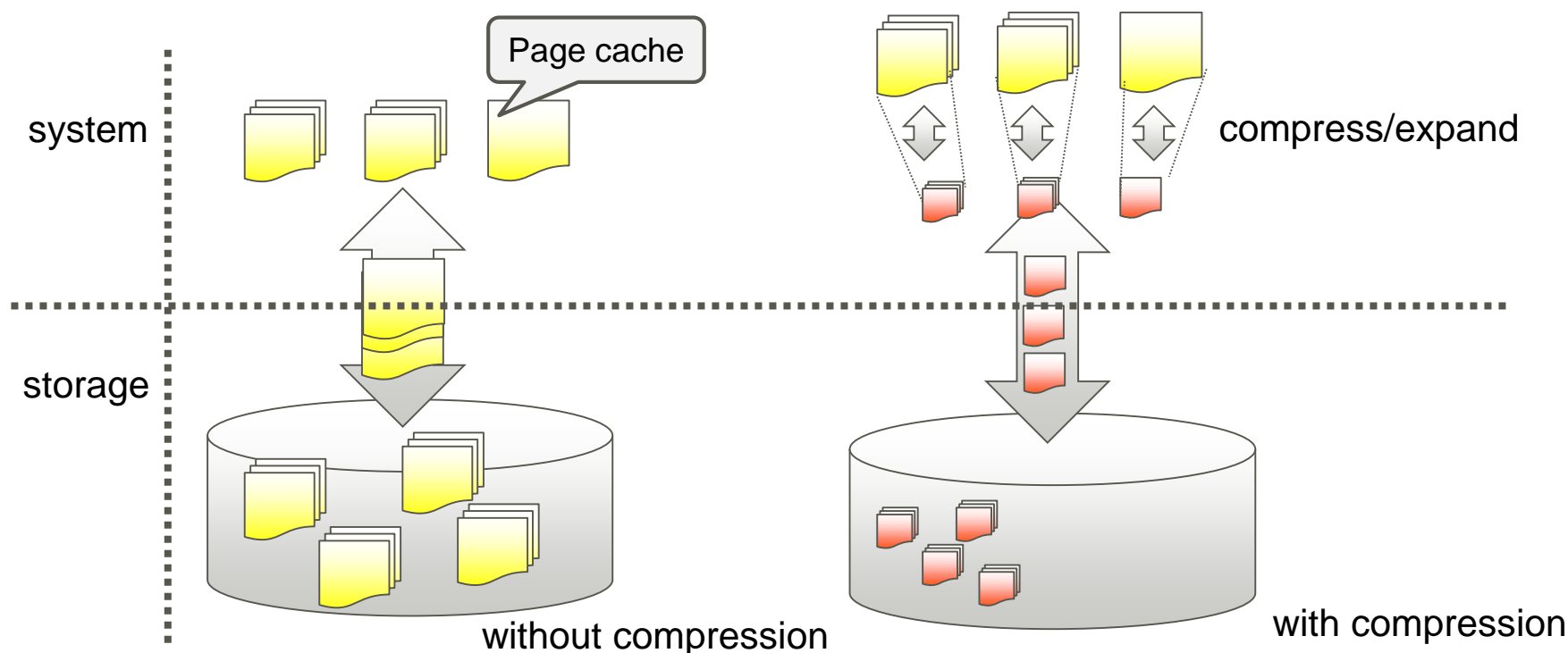
Hardware Environment	Software Environment
<ul style="list-style-type: none">• PRIMERGY RX300 S6<ul style="list-style-type: none">• CPU: Intel Xeon X5690 3.47GHz x12 core• Memory: 16GiB• Storages: 100GB HDD x 2	<ul style="list-style-type: none">• Red Hat Enterprise Linux 7.0• File systems<ul style="list-style-type: none">• Btrfs<ul style="list-style-type: none">• Data/metadata: RAID1• Other options: default• XFS: default options• Volume manager for XFS<ul style="list-style-type: none">• dm-thinp: chunksize is 256KiB• LVM: RAID1

- Copy: Btrfs > LVM >>> dm-thinp
- Snapshot: Btrfs > dm-thinp >>> LVM

Volume type	Copy	Snapshot	
		Without page cache	With page cache
Btrfs	106s	0.126s	11.7s
XFS on dm-thinp	209s	0.260s	15.5s
XFS on LVM	133s	1.03s	45.2s

Transparent compression

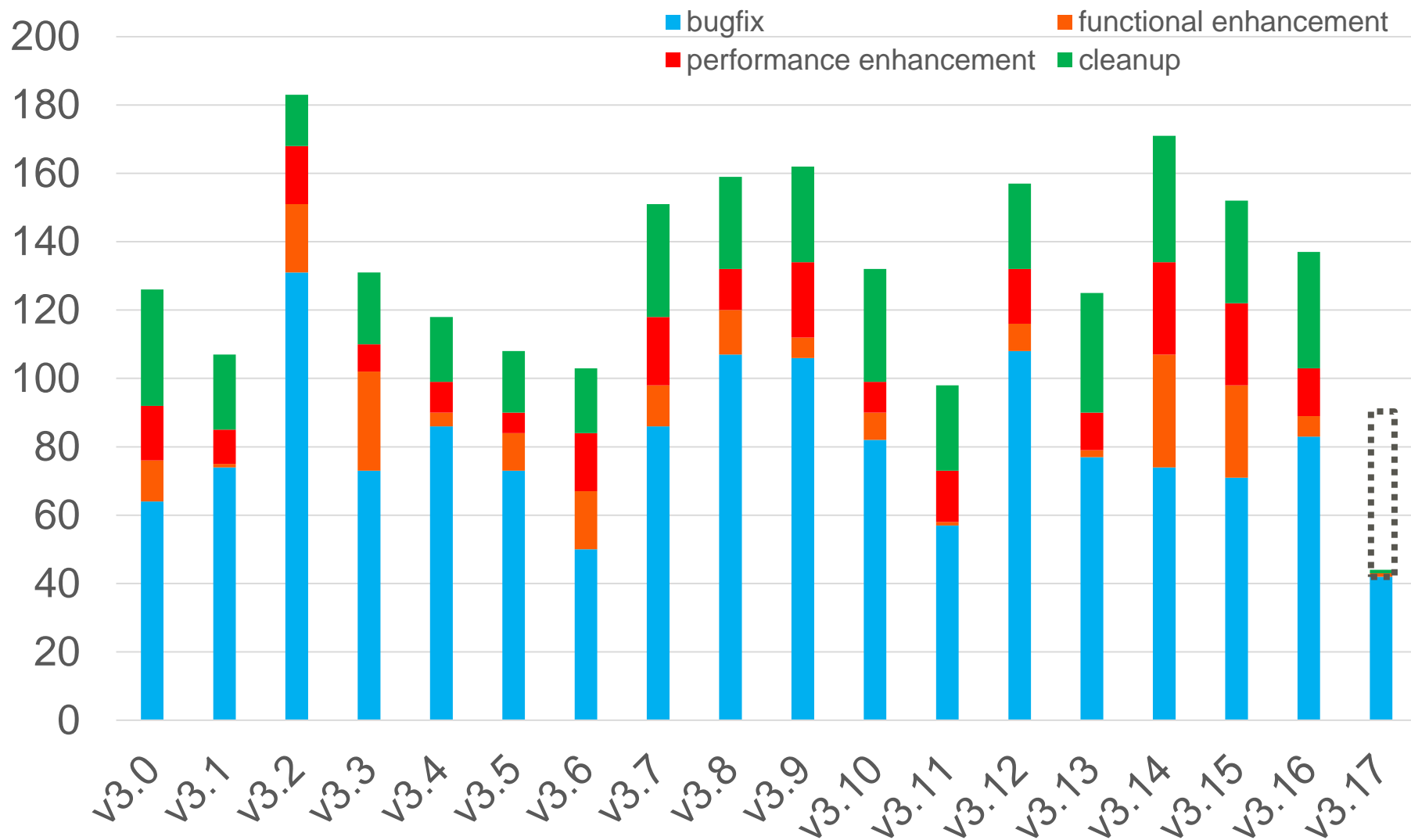
- Automatically compress/expand file data on I/O
 - Low space consumption and high I/O performance
 - Need some extra CPU time
 - Usage: `mount -o compress={lzo,zlib} <device> <mnt point>`
 - Can also be enabled/disabled for each file



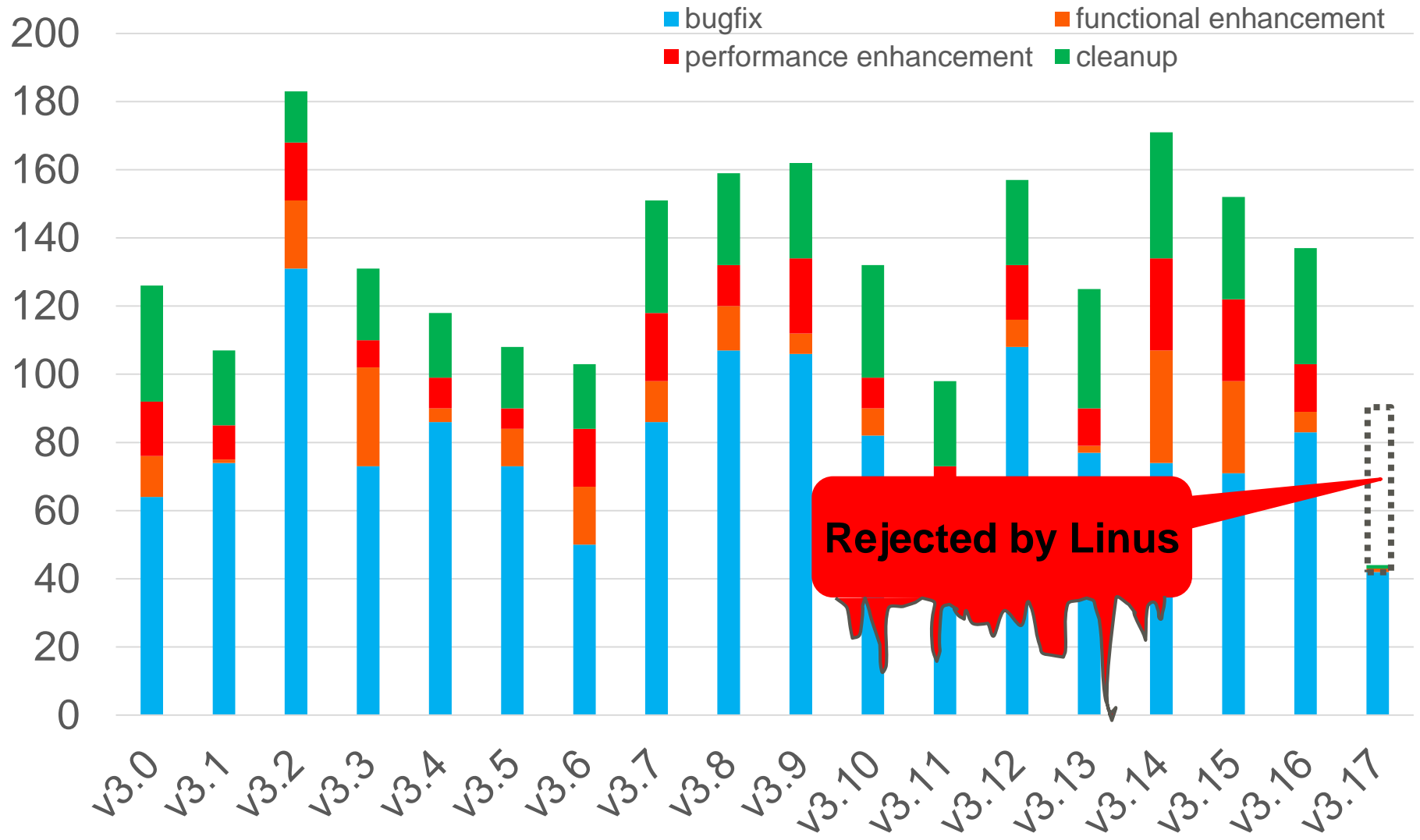
- Background
- Core Features
- **Developments statistics**
- Future Prospects

- Patch statistics
- Performance
- Summary

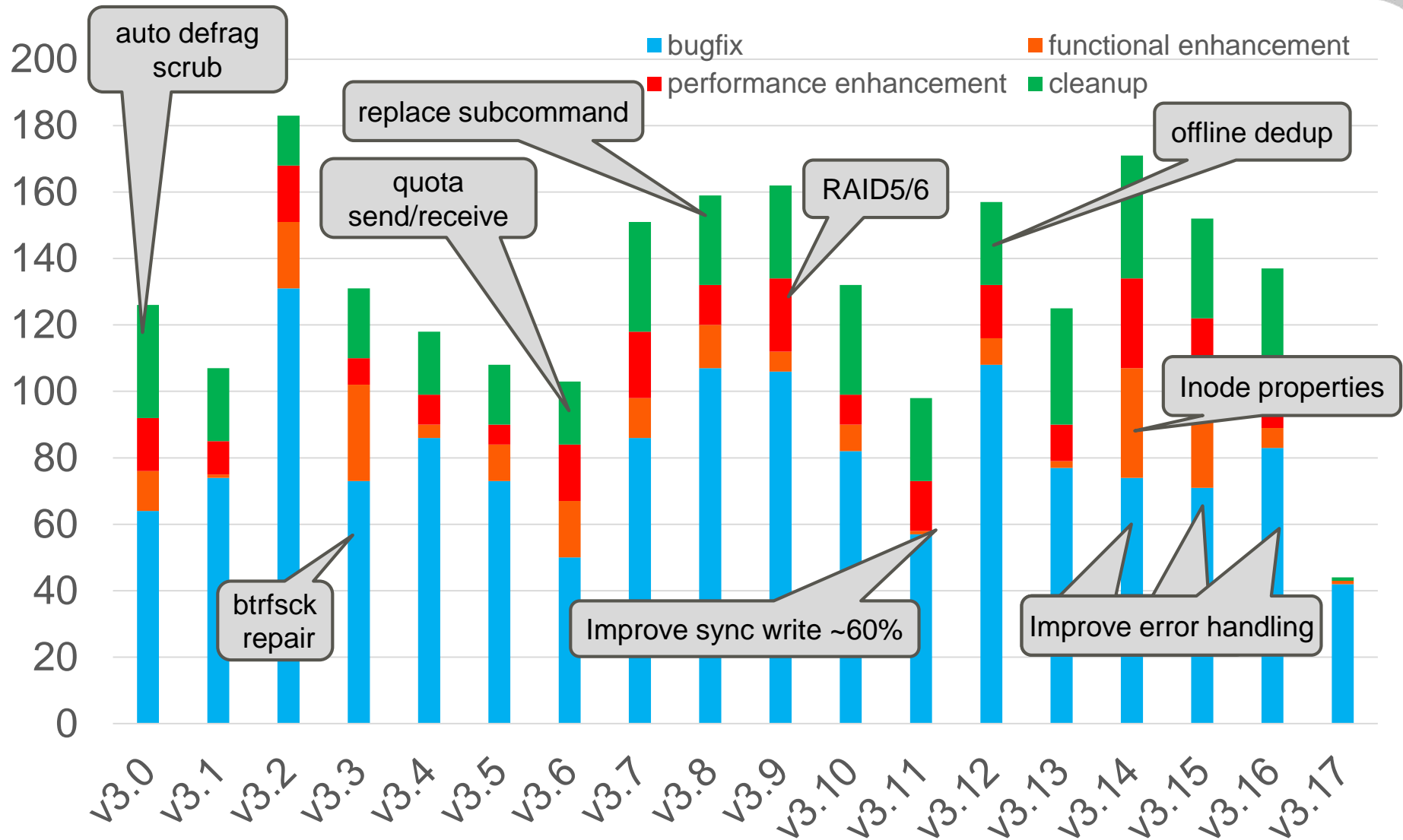
Patch Statistics



Patch Statistics: Tips of v3.17

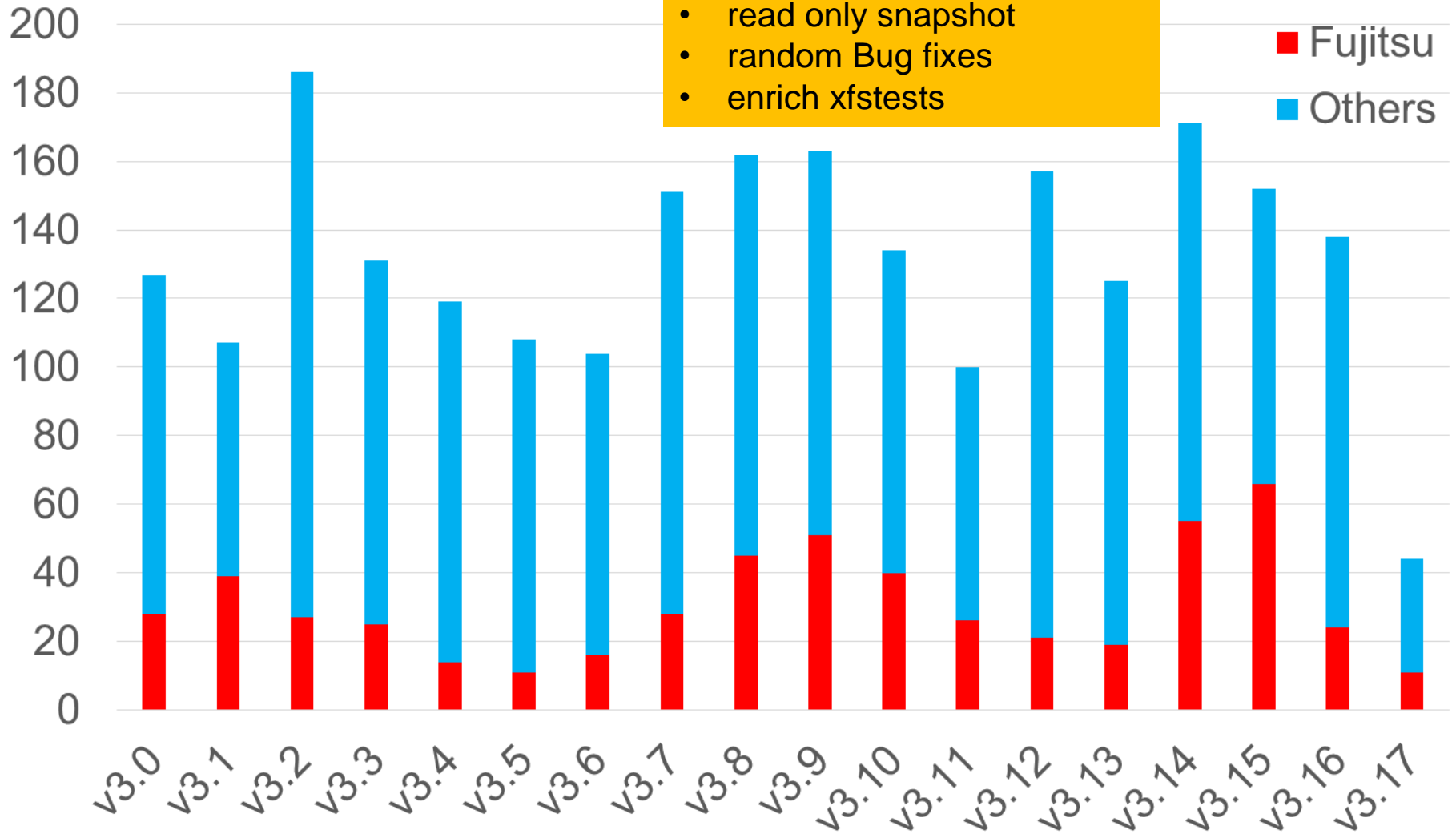


Patch statistics: Main changes

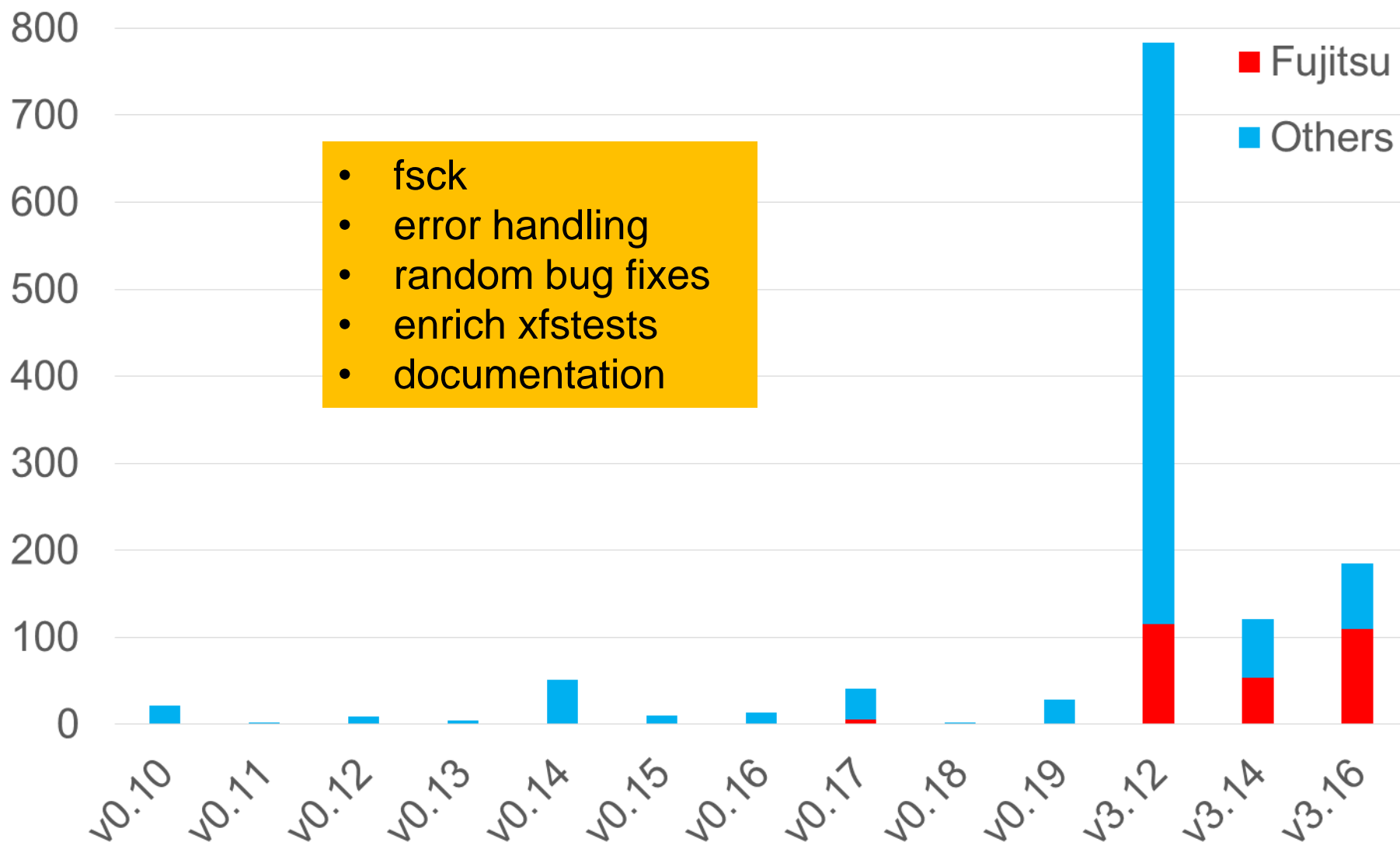


Fujitsu's contribution

- btrfsck, error handling
- fast {random/async} write
- LZO compression
- read only snapshot
- random Bug fixes
- enrich xfstests

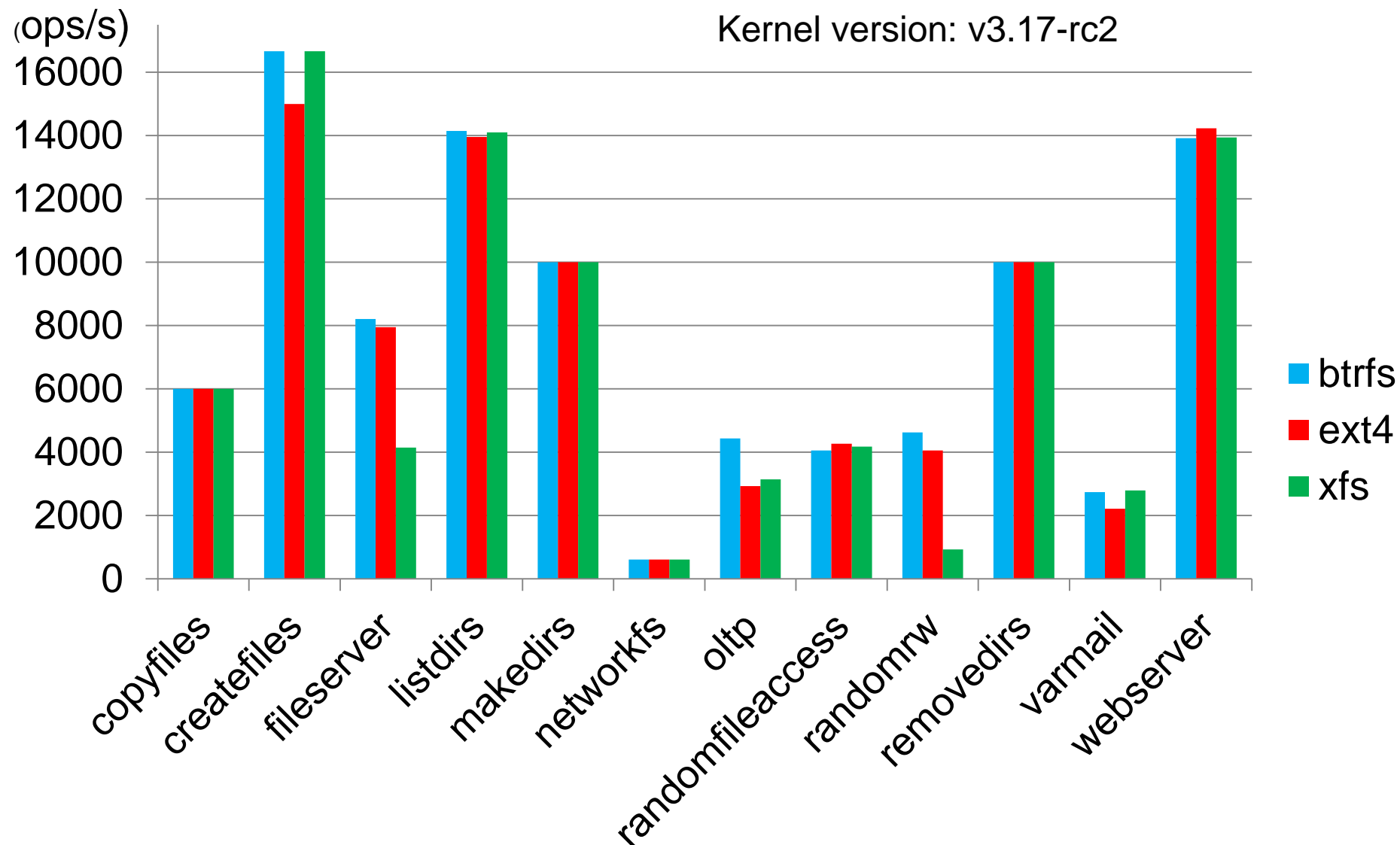


Fujitsu's contribution: btrfs-progs

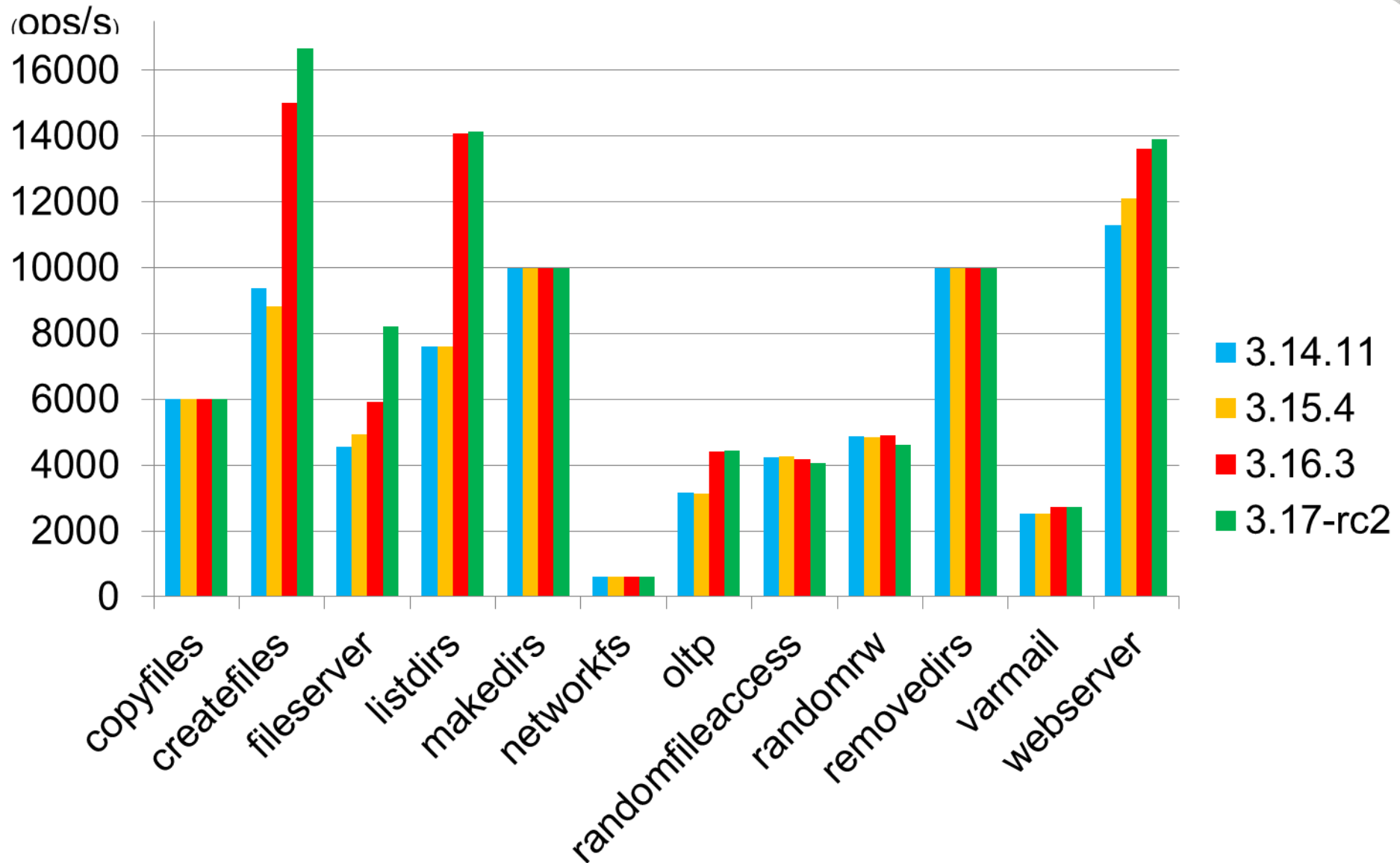


Hardware Environment	Software Environment
<ul style="list-style-type: none">• PRIMERGY TX300 S6<ul style="list-style-type: none">• CPU: Xeon x5670 x 2<ul style="list-style-type: none">• 12 core• HT is disabled• Memory: 4GB• HDD: 300GB x 1<ul style="list-style-type: none">• MegaRAID SAS, HITACHI HUS156030VLS600	<ul style="list-style-type: none">• Benchmark software: filebench• Kernel: 3.14.2, 3.15.3, 3.16.0, and 3.17-rc2<ul style="list-style-type: none">• I/O scheduler: deadline• File systems: Btrfs(single volume), XFS, and ext4• default mkfs options and mount options

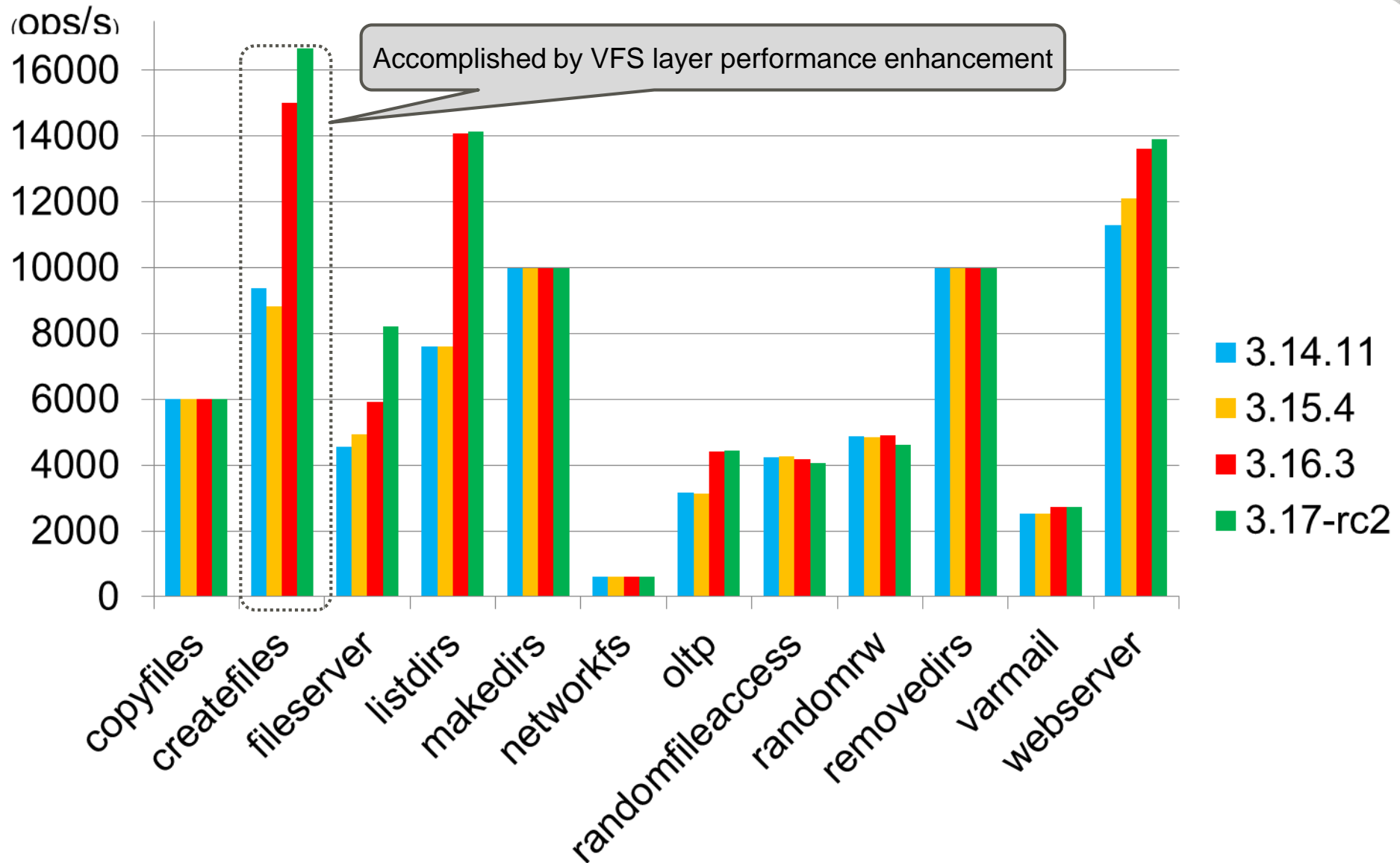
The result: Compare with other file systems



The result: Compare with old Btrfses



VFS has also improved performance



■ Ready to use without RAID5/6

- Performance: OK

- Stability: OK

- # of new features has decreased
- Test coverage has increased

- Features: almost OK

- RAID5/6: Lack of scrub and replace subcommands

■ RAID1 and RAID10 are the best choice

- Especially safe and stable

- Background
- Core Features
- Developments statistics
- **Future Prospects**

■ RAID 5/6 enhancement

■ Add scrub and replace subcommands

- We're testing patches now and will post it to linux-btrfs ML soon

■ Add five tests for these features to xfstests

■ Further enhancement of robustness and performance

■ Repairing tools and so on

■ Education and documents for this purpose

■ Operation know-how

- Btrfs operations are different from other file systems
 - e.g. Btrfsの基礎 part1 機能編(It's in Japanese. Now translating to English...)

http://www.slideshare.net/fj_staoru_takeuchi/btrfs-part1

■ File system structure

■ Code logic

Future Prospects: Btrfs users are increasing

- OpenSuSE13.2: Will be used as its default
- Ubuntu: Support
- RHEL7: Available as tech-preview
- Will be used for In Vehicle Infotainment(IVI) system
 - Linux File System Analysis for IVI system, Mitsuharu Ito, Fujitsu
http://events.linuxfoundation.jp/sites/events/files/slides/linux_file_system_analysis_for_IVI_systems.pdf

- **Please try Btrfs**

- **It's ready to use**

- **RAID1/10 are the best choice**

- **RAID5/6 need some more work**

- **Recommend the newest stable kernel**

- Linux File System Analysis for IVI system, Mitsuharu Ito, Fujitsu

http://events.linuxfoundation.jp/sites/events/files/slides/linux_file_system_analysis_for_IVI_systems.pdf

- Btrfsの基礎 part1 機能編

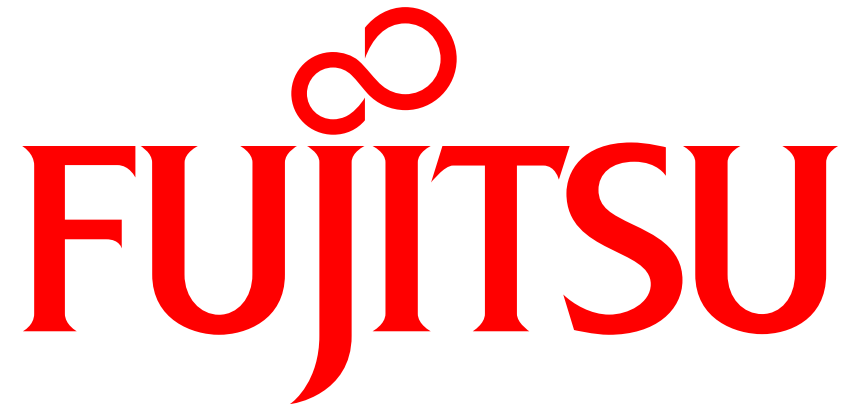
http://www.slideshare.net/fj_staoru_takeuchi/btrfs-part1

- Linux-btrfs ML

linux-btrfs@vger.kernel.org

- Btrfs wiki

https://btrfs.wiki.kernel.org/index.php/Main_Page



shaping tomorrow with you