Which filesystem should I use?
LinuxTag 2013

Heinz Mauelshagen
Consulting Development Engineer

TOP

- Major on-disk local Linux filesystems
- Features, pros & cons of each
- Filesystem tools
- Performance/scalability
- Benchmarks
- Conclusions
- Resources & Questions

Local Linux Filesystems

- Major on-disk local filesystems
  - Ext3, Ext4, XFS, BTRFS (all journaled/logged)

- Others are available for special purposes
  - vfat, msdos, udf, cramfs, squashfs, nilfs...
  - network/cluster

Ext3 Filesystem

- Ext3 was the most common file system in Linux (2000)
  - Most distributions historically used it as their default
  - Applications tuned to its specific behaviors (fsync...)
  - Familiar to most system administrators
- Ext3 challenges
  - fsck time can be extremely long for large, populated filesystems
  - Maximum file size of 2TiB, maximum file system size of 16TiB
-> hard scalability limit
  - Maximum 32000/31998 subdirectories
  - Can be significantly slower than other local file systems
    - Direct/indirect block mapping slow
    - Allocation bitmaps throttling free space searches
    - No delayed allocations
  - ...

Ext4 Filesystem

- Ext4 has many compelling new features (2008)
  - Extent based and delayed allocation, preallocation
  - Small files stored more efficiently
  - Higher bandwidth
  - Faster mkfs (-E lazy_itable_init=1) and fsck time (up to 10x over Ext3)
  - (Should be) relatively familiar to experienced ext3 users
  - Ext2 -> Ext3 -> Ext4 in-place migration path
- Ext4 challenges
  - Large device support not polished in its user space tools
  - based on 1980th filesystem design because of Ext2/3 predecessors
  - barely suitable for todays very large file and filesystem sizes
(free space bitmap); optimization being worked on but still bitmap based

XFS Filesystem

- XFS is very robust and scalable (Irix 1994 / Linux 2001/2003)
  - Very good performance for large storage configurations and large servers
  - Many years of use on large (> 16TiB) storage
  - Extent and delayed allocation
  - High bandwidth
- XFS challenges
  - Not as well known by many users and field support people
  - Until recently, had performance issues with meta-data intensive (create/unlink) workloads
  - No in-place migration from Ext*

BTRFS Filesystem

- BTRFS is very scalable and includes enhanced management functionality (2009)
  - the newest local ZFS-type filesystem adding features
    which can't be easily added to others;
    aka Butter/Better/B-tree filesystem
  - Copy on write; nothing will ever be overwritten
  - Has its own internal RAID
  - Snapshot/Clone support
  - Compression support
  - Does full data integrity checks for metadata and user data
    -> proactive error management
  - Can dynamically grow and shrink
  - In-place Ext* conversion (btrfs-convert)
- BTRFS challenges
  - Not as well known by many users and field support people
  - Still no ~~working~~ full-featured fsck
  - Problems with full filesystem (fixed now?)
  - Performance/Reliability constraints -> not (yet) meant for production use!

Filesystem Tools

- e2fsprogs
  - badblocks, debugfs, e2label, resize2fs, tune2fs, ...
- xfsprogs
  - no fsck.xfs but xfs_repair, xfs_admin, xfs_db, xfs_fsr, ...
- btrfs-progs
  - no ~~working~~ fsck.btrfs, btrfs, btfs-image, btrfs-restore, btrfs-zero-log, ...
- They all differ, thus causing administration complexity
  - SSM (System Storage Manager) helping that to a certain degree by providing a unique CLI on them and LVM, MD, ...
- fstrim
  - Used to discard (or trim) blocks the filesystem doesn't use any more
  - Not just on SSDs to help their free space management but also on any thin provisioned storage (HW Array or thin provisioned Lvs)
  - Run regularly as a cron job

Feature Comparison

	Ext3	Ext4	XFS	BTRFS
Online resize	Grow only	Grow only	Grow only	Grow+Shrink
Offline resize	Grow+Shrink	Grow+Shrink	No	No
Online checks	No	No	No	Yes (scrubber)
Snapshots	No	No	No	Yes
Clones	No	No	No	Yes
Internal RAID	No	No	No	Yes
Compression	No	No	No	Yes (zlib/lzo)
Dedupe/Encryption	No	No	No	Not yet
Online Defrag	No	Yes	Yes	Yes
Discard (TRIM)	Yes	Yes	Yes	Yes
FLUSH/FUA(Barrier)	Yes	Yes	Yes	Yes
Metadata CRC	Yes	Yes	Yes	Yes
Data CRC	No	No	No	Yes
Extent allocation	No	Yes	Yes	Yes
Delayed allocation	No	Yes	Yes	Yes
Production-ready	Yes	Yes	Yes	Not yet

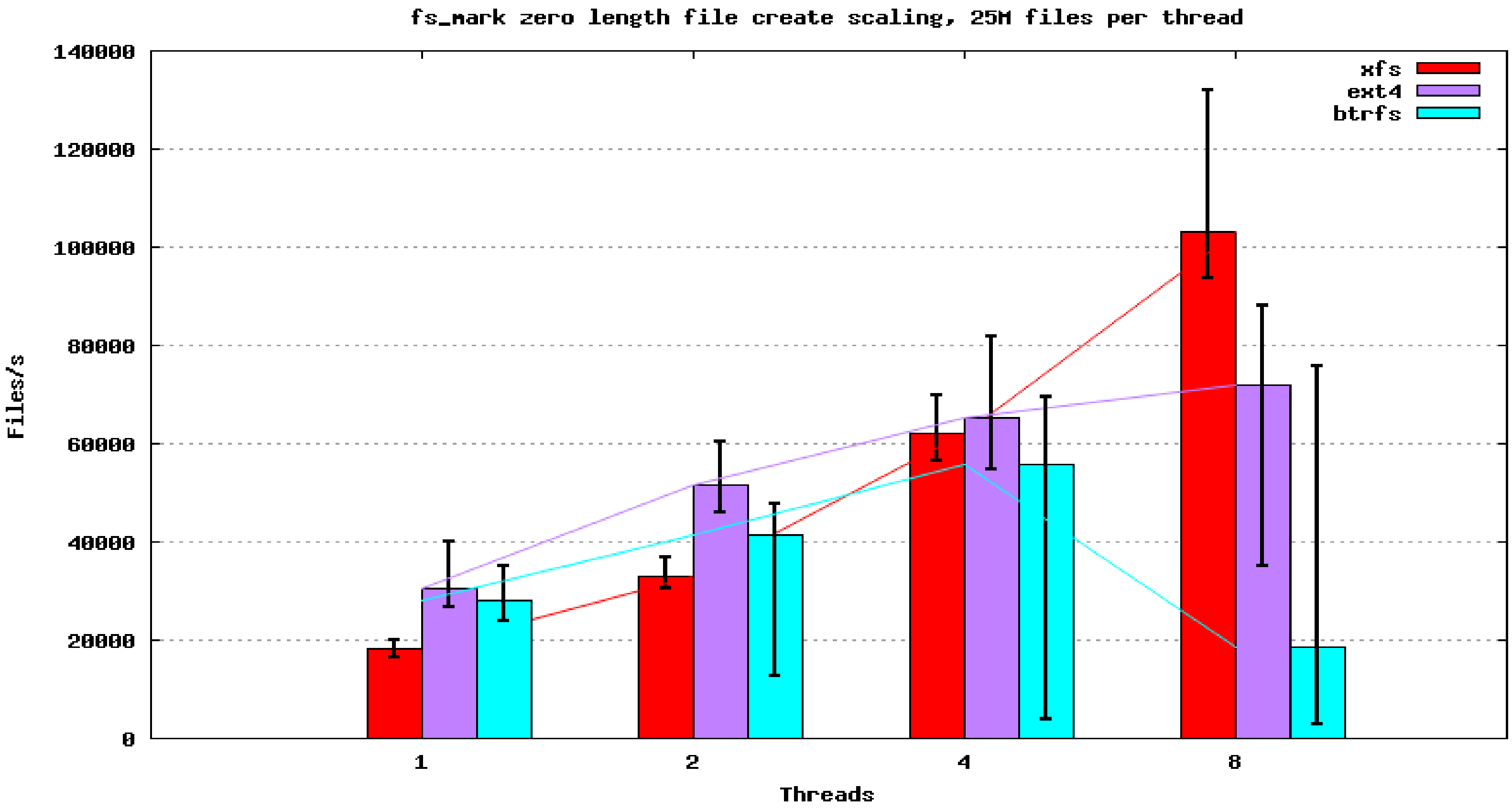
Design Limits

	Max File Size	Max Filesystem Size
Ext3	2 TiB	16 TiB
Ext4	1 EiB	1 EiB (tool limits < !)
XFS	8 EiB	16 EiB
BTRFS	8 EiB	16 EiB

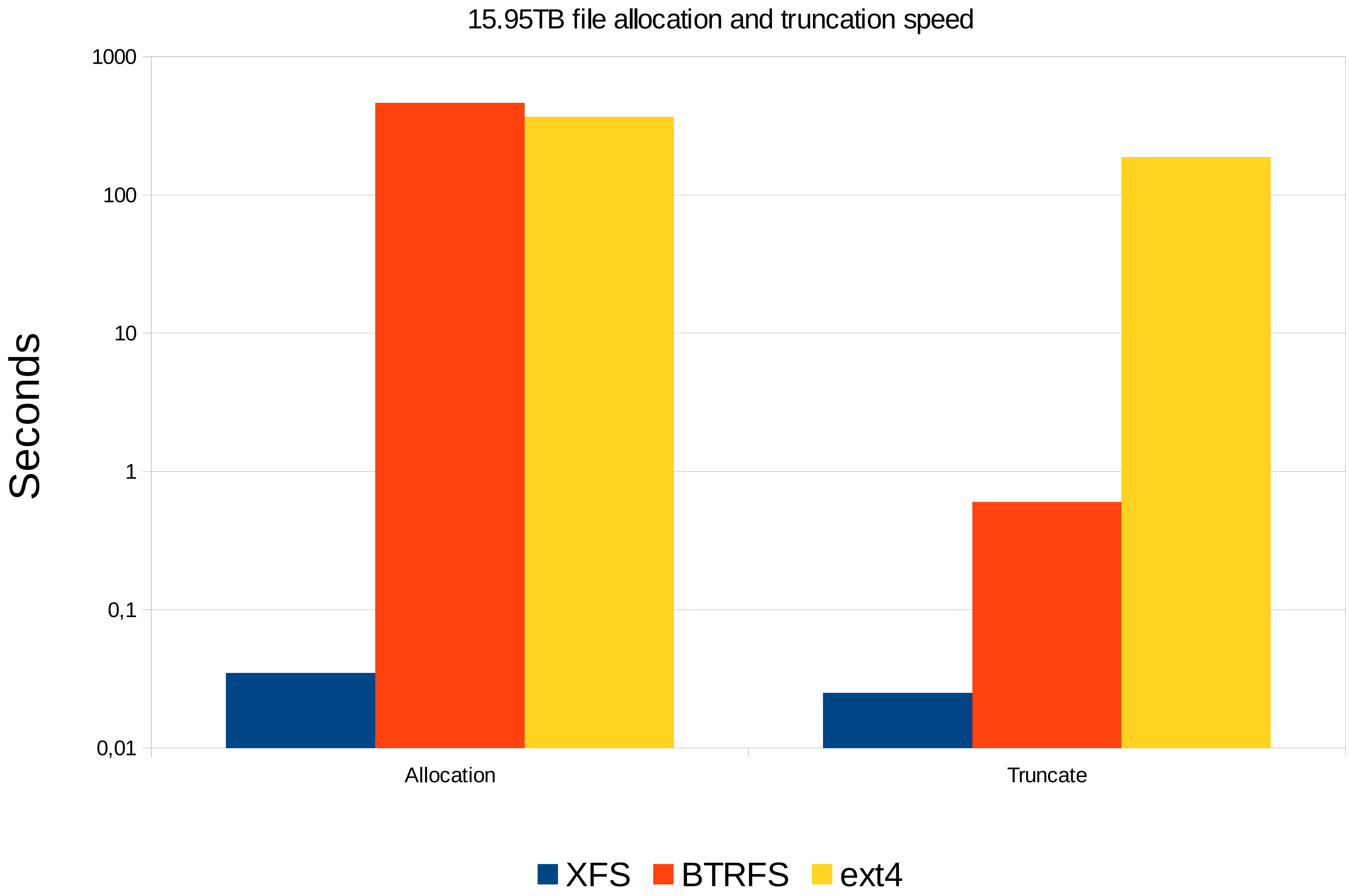
Mind the tested and supported ones!

Benchmarks (or that lies proverb)...

- Dave Chinners LCA talk
  (17TB, 12-disk RAID0; 8P KVM guest, 4G memory)

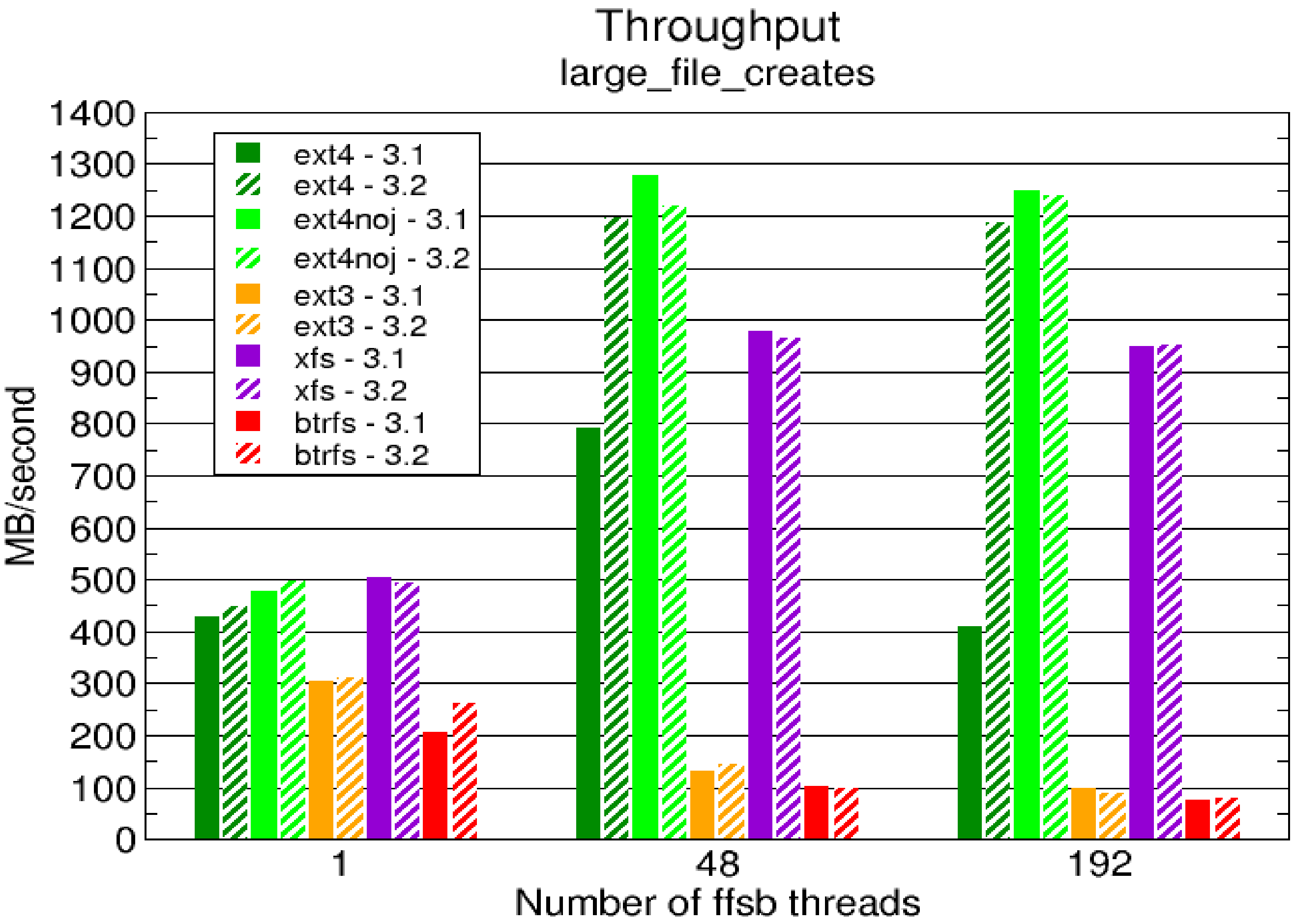


Benchmarks...


15.95TB file allocation and truncation speed
Seconds
1000
100
10
1
0,1
0,01
Allocation
Truncate
XFS
BTRFS
ext4

Benchmarks...

- Eric Whitney's FFSB testing @HP
  (48P, 256G, 7T of SAS disks in RAID0)



Benchmarks...

- enterprisestorageforum.com fsck test

  (md RAID-60 on DDN LUNS; fs_mark population)

FS Size, TiB	Nr of Files (millions)	XFS (seconds)	Ext4 (seconds)
72	105	1629	3193
72	51	534	1811
72	10	161	972
38	105	710	3372
38	51	266	1358
38	10	131	470

...Benchmarks

- mkfs a 128TiB filesystem (sparse LV on one SSD without discard)

Ext3	Ext4	XFS	BTRFS
-EFBIG	3m39s	33.3s	0.04s

Conclusions...

- Ext3 no big data; at least use Ext4
  - File size limit 2 TiB / file system size limit 16 TiB
  - Limited bandwidth due to block allocation
- Ext4 a bit further but still no big data
  - File size limit 16 TiB / file system size limit 1 EiB
  - Tools still limit maximum designed filesystem size
  - More bandwidth than Ext3 because of extent allocation
  - Reasonable performance
- XFS big data and long term field record (20 years)
  - Anything larger than 16 TiB...
- BTRFS big data, many more features but not yet production ready (bug fixes, bug fixes, ...)
  - Test/evaluate for now

- Filesystem tools all individual (almost) without common CLI; SSM (SystemStorageManager) helping here
- Snapshots allowing for OS upgrade rollbacks etc.

...Conclusions

- Challenges for all of these filesystems
  - Ability to scale to real big file and file system sizes
    - Data model (structures)?
    - Algorithms proper?
    - Parallelism on threaded IO
  - Storage integrity
    - Detect errors from disk at runtime with checksums (BTRFS the only for now)?
    - On data? On metadata?
    - Autocorrection on RAID > 1 (BTRFS the only for now)?
  - Consistency and reliability of (new) tools / features

Resources & Questions

- Mailing lists
  - linux-ext4@vger.kernel.org
  - xfs@oss.sgi.com
  - linux-btrfs@vger.kernel.org
- IRC
  - #xfs, #btrfs on irc freenode.net
  - #ext4 on irc.oftc.net
- SSM
  - http://fedoraproject.org/wiki/Features/SystemStorageManager
- Questions?