Enterprise Filesystems

Eric Sandeen
Principal Software Engineer,
Red Hat

Feb 21, 2013

What We'll Cover

- Local “Enterprise-ready” Linux filesystems
  - Ext3
  - Ext4
  - XFS
  - BTRFS
- Use cases, features, pros & cons of each
- Recent & future work
  - Features
  - Scalability
- Benchmarks

Local Filesystems in RHEL6

- We ship what customers need and can rely on
- We ship what we test and support
  - Major on-disk local filesystems
    - Ext3, Ext4, XFS, BTRFS*
  - Others are available for special purposes
    - fat, vfat, msdos, udf, cramfs, squashfs...
  - We'll cover the “big four” today

The Ext3 filesystem

- Ext3 ~~is~~ was the most common file system in Linux
  - Most distributions historically used it as their default
  - Applications tuned to its specific behaviors (fsync...)
  - Familiar to most system administrators
- Ext3 challenges
  - File system repair (fsck) time can be extremely long
  - Limited scalability - maximum file system size of 16TB
  - Can be significantly slower than other local file systems
  - direct/indirect, bitmaps, no delalloc ...

The Ext4 filesystem

- Ext4 has many compelling new features
  - Extent based allocation
  - Faster fsck time (up to 10x over ext3)
  - Delayed allocation, preallocation
  - Higher bandwidth
  - Should be relatively familiar for existing ext3 users
- Ext4 challenges
  - Large device support not polished in its user space tools
  - Limits supported maximum file system size to 16TB*
  - Has different behavior over system failure

The XFS filesystem

- XFS is very robust and scalable
  - Very good performance for large storage configurations and large servers
  - Many years of use on large (> 16TB) storage
  - Red Hat tests & supports up to 100TB
- XFS challenges
  - Not as well known by many customers and field support people
  - Until recently, had performance issues with meta-data intensive (create/unlink) workloads

The BTRFS filesystem

- BTRFS is the newest local file system – copy on write
- Shipped in RHEL6 as a tech preview item
- Has its own internal RAID and snapshot support
- Does full data integrity checks for metadata and user data
- Compression support
- Can dynamically grow and shrink
- Developers very interested in feedback and testing
- Not (yet) meant for production use!

The BTRFS filesystem

- But …
  - Still no ~~working~~ full-featured fsck
  - ENOSPC took a while (fixed now!) (?)
  - Encryption yet to come
  - COW can fragment oft-written files
  - Perf issues still being worked out
  - Just missed “default” status for Fedora ~~17~~ 18

Generic but interesting features

- Ext4, XFS, btrfs all have:
  - Delayed allocation
  - Per-file space preallocation
  - Hole punch ~~(not on btrfs yet)~~
  - Trim / discard
  - Barrier (now flush/FUA) support
  - Defragmentation

How to Choose? Use Cases

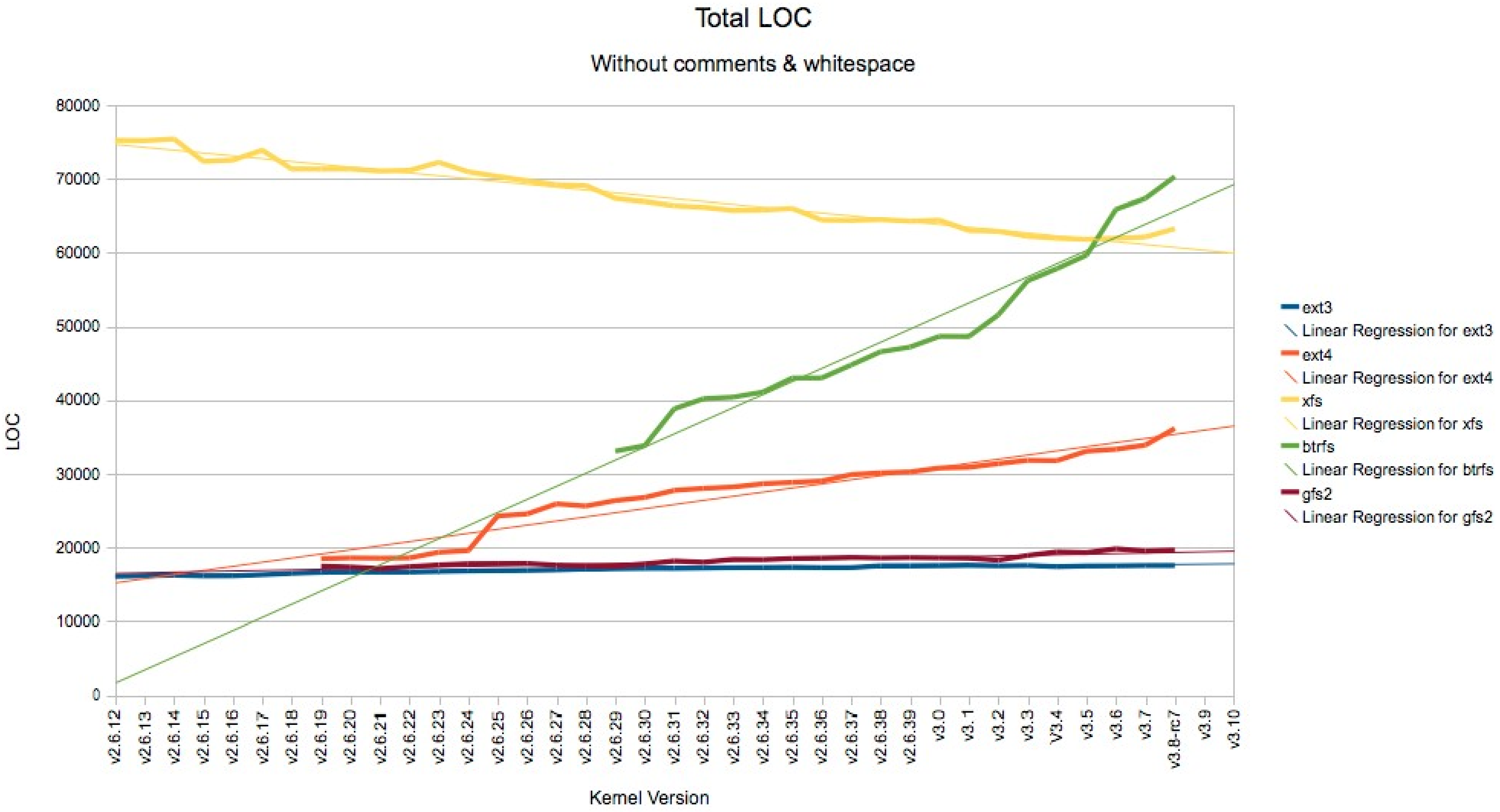
- Ext4
  - Fine for general use, familiar
  - Reasonable performance, scalability somewhat limited
- XFS
  - “If you have large or lots, use XFS” - Valerie Aurora
  - Anything over 16T
  - Don't fear the metadata!
- BTRFS
  - For now, testing
  - Nice admin features – OS upgrade rollback, etc

Active Maintenance and Development

- Since kernel v2.6.18 (~RHEL5):
  - Ext3 : 568 commits, ~123 authors
  - Ext4 : 1803 commits, ~218 authors
  - XFS : 2577 commits, ~152 authors
  - Btrfs : 2956 commits, ~162 authors
- Most are eavily weighted towards some authors, i.e.:

```
# git log –no-merges v2.6.18.. fs/ext4 fs/jbd2 | grep ^Author | awk -F "<" '{print
$1}' | sort | uniq -c | sort -n | tail -n 8
     50 Author: Christoph Hellwig
     55 Author: Dmitry Monakhov
     63 Author: Yongqiang Yang
     67 Author: Tao Ma
     90 Author: Jan Kara
    111 Author: Eric Sandeen
    155 Author: Aneesh Kumar K.V
    423 Author: Theodore Ts'o
```

Active Maintenance and Development



Where to now?

- All of these filesystems face challenges in the future
  - Ability to scale in sheer filesystem size
    - Containers mostly in shape
    - But structures & algorithms …?
  - Integrity with large storage
    - Detect errors from disk at runtime with checksums
    - On data? On metadata?
    - Then what?
  - More Features!

Ext3 Scaling & Features

- None.

Ext4 Scaling & Features

- Bigalloc (since kernel v3.2)
  - Workaround for bitmap scalability issues
  - Allocates *multiples* of 4k blocks at a time
  - Not true large filesystem blocks, but close?
- Inline Data – (since kernel v3.8)
  - Store data inline in (larger) inodes
  - Mitigate bigalloc waste?
- Metadata Checksumming – (since kernel v3.5)

XFS Scaling & Features

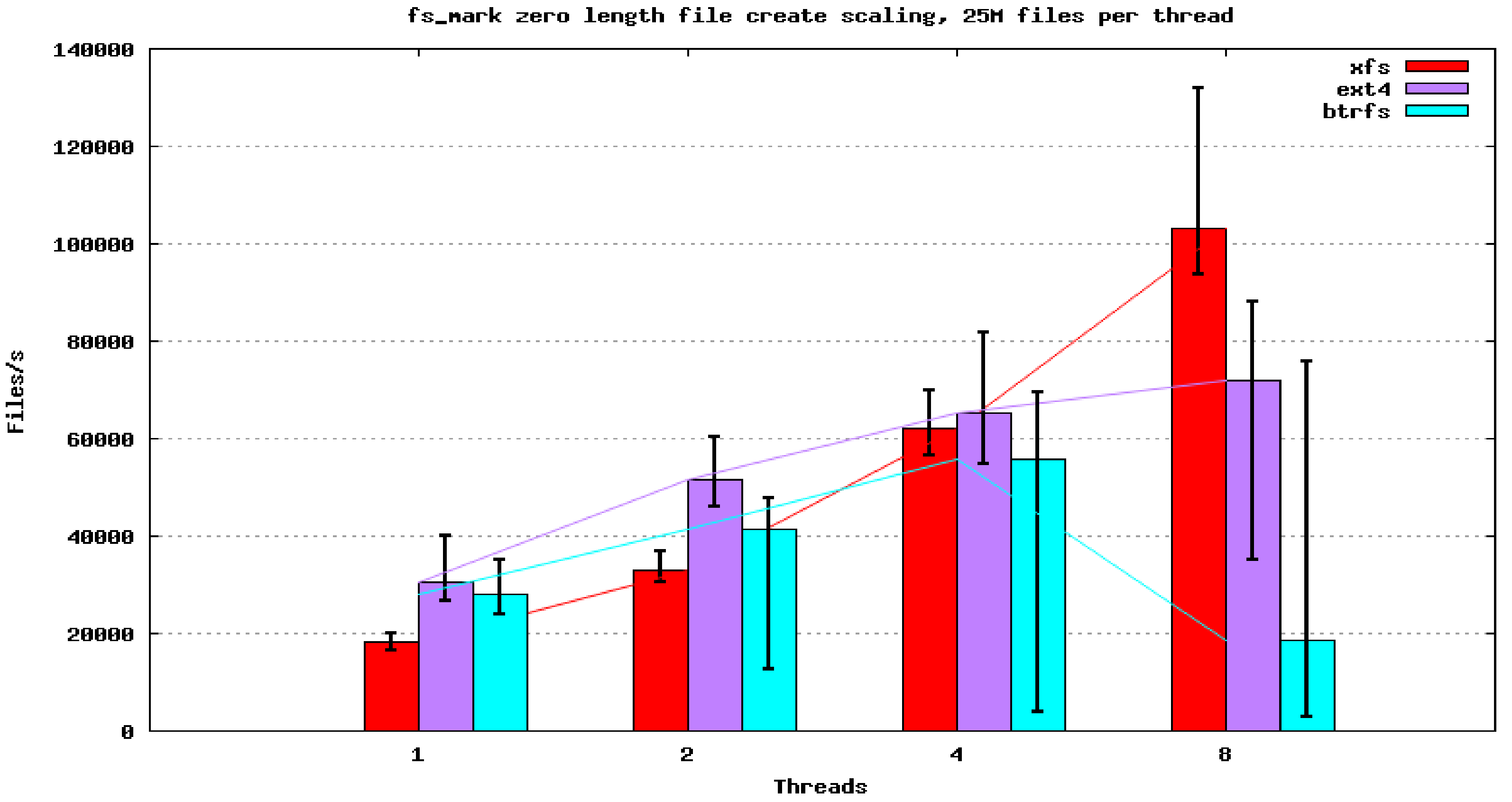
- “Delayed logging” is done
  - dramatically improved metadata performance
  - default since v2.6.39
  - Last™ big performance issue
- Integrity work is next
  - CRCs on all metadata and log
  - FS UUID to detect misdirected writes
  - Transaction rollback in the face of errors
  - Background scrub

BTRFS Scaling & Features

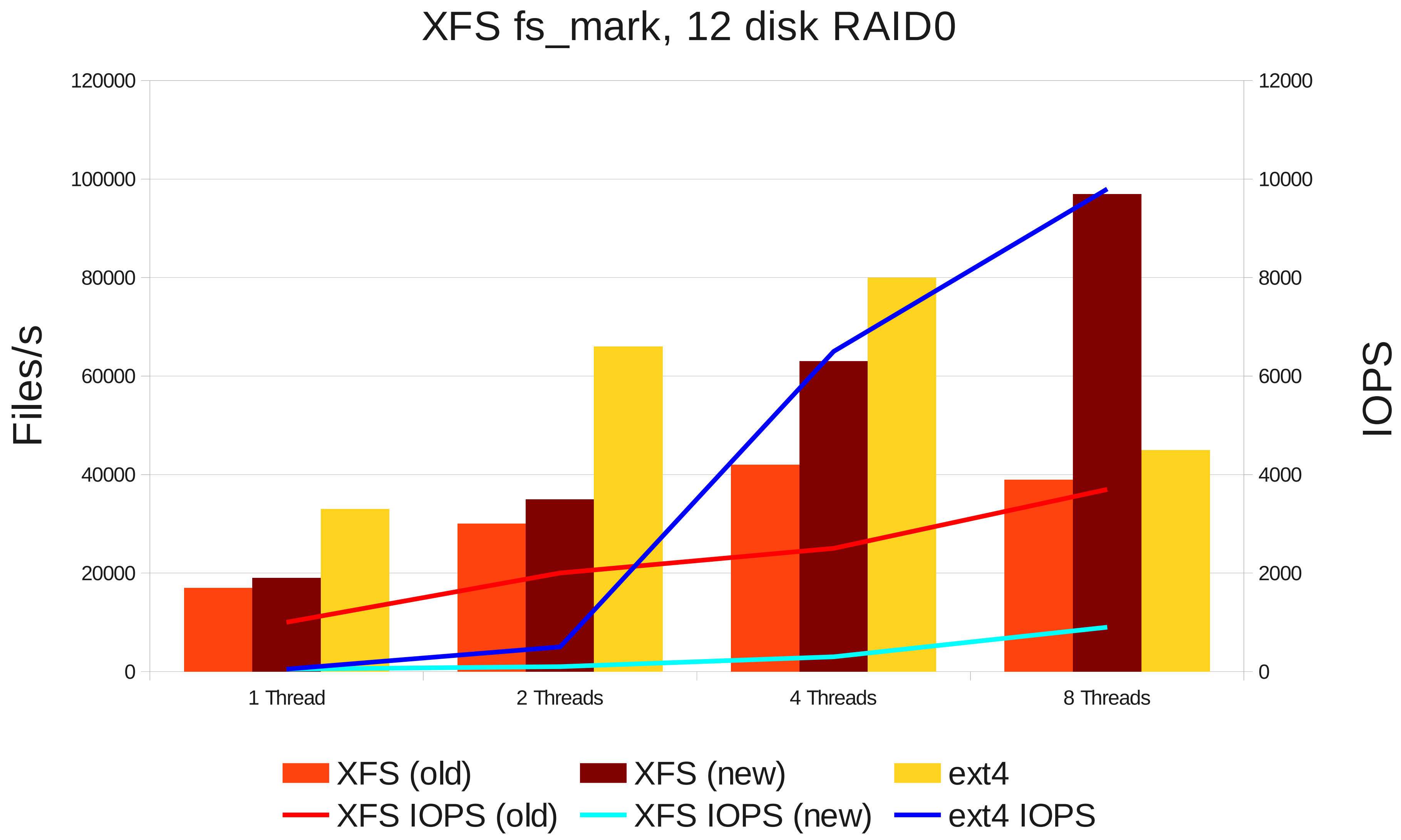
- Scaling work here and there
- Mostly still fleshing out features
  - Checksumming was done early
  - fsck.btrfs will be ready tomorrow™
  - RAID 5/6 (just released)
  - Quotas
  - Dedupe
  - Encryption
- Bugfixing, bugfixing, bugfixing

Lies, Damned Lies, and Benchmarks

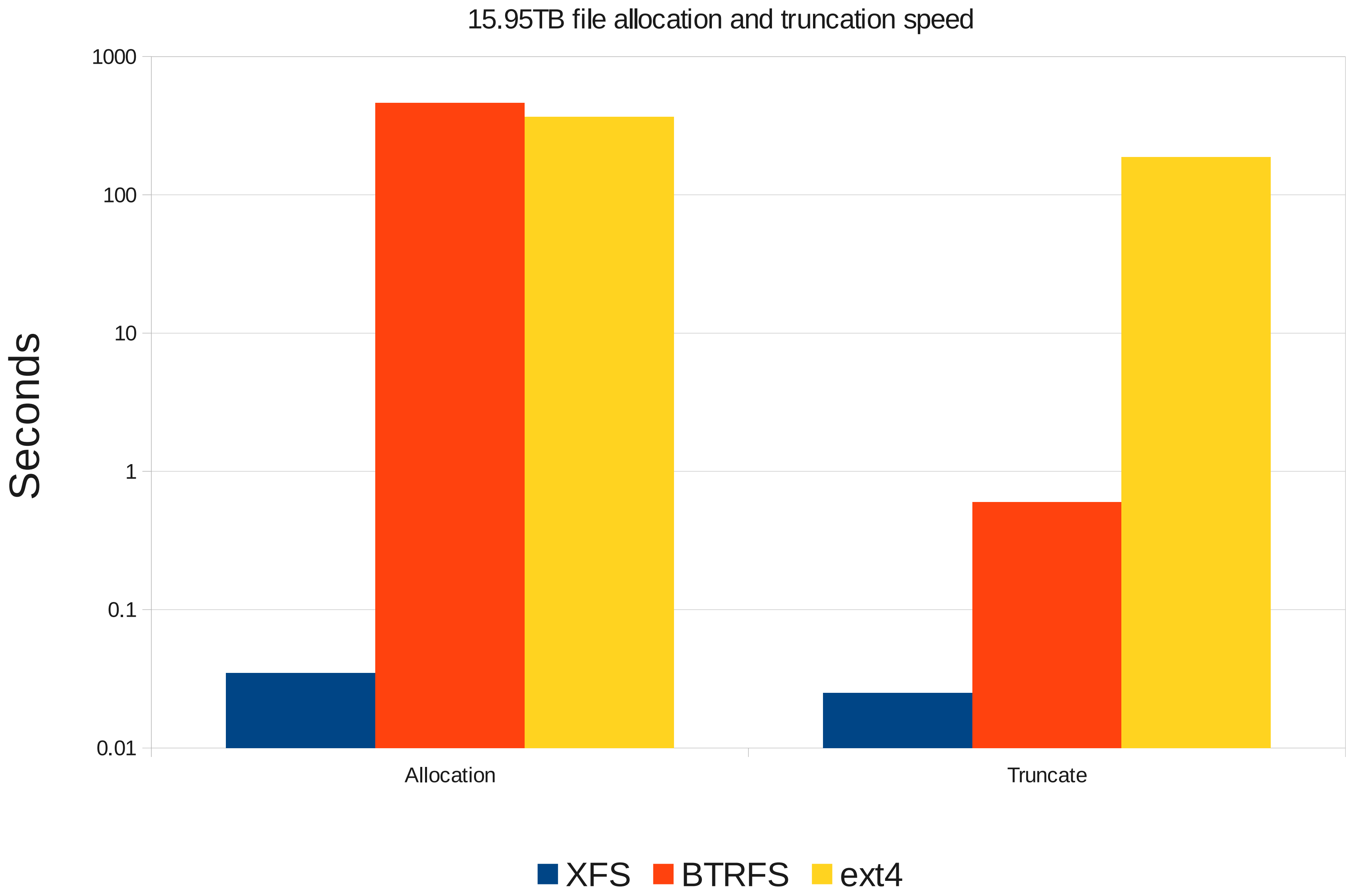
- Dave Chinner's LCA Talk
  - 17TB, 12-disk RAID0; 8P KVM guest, 4G memory



Lies, Damned Lies, and Benchmarks

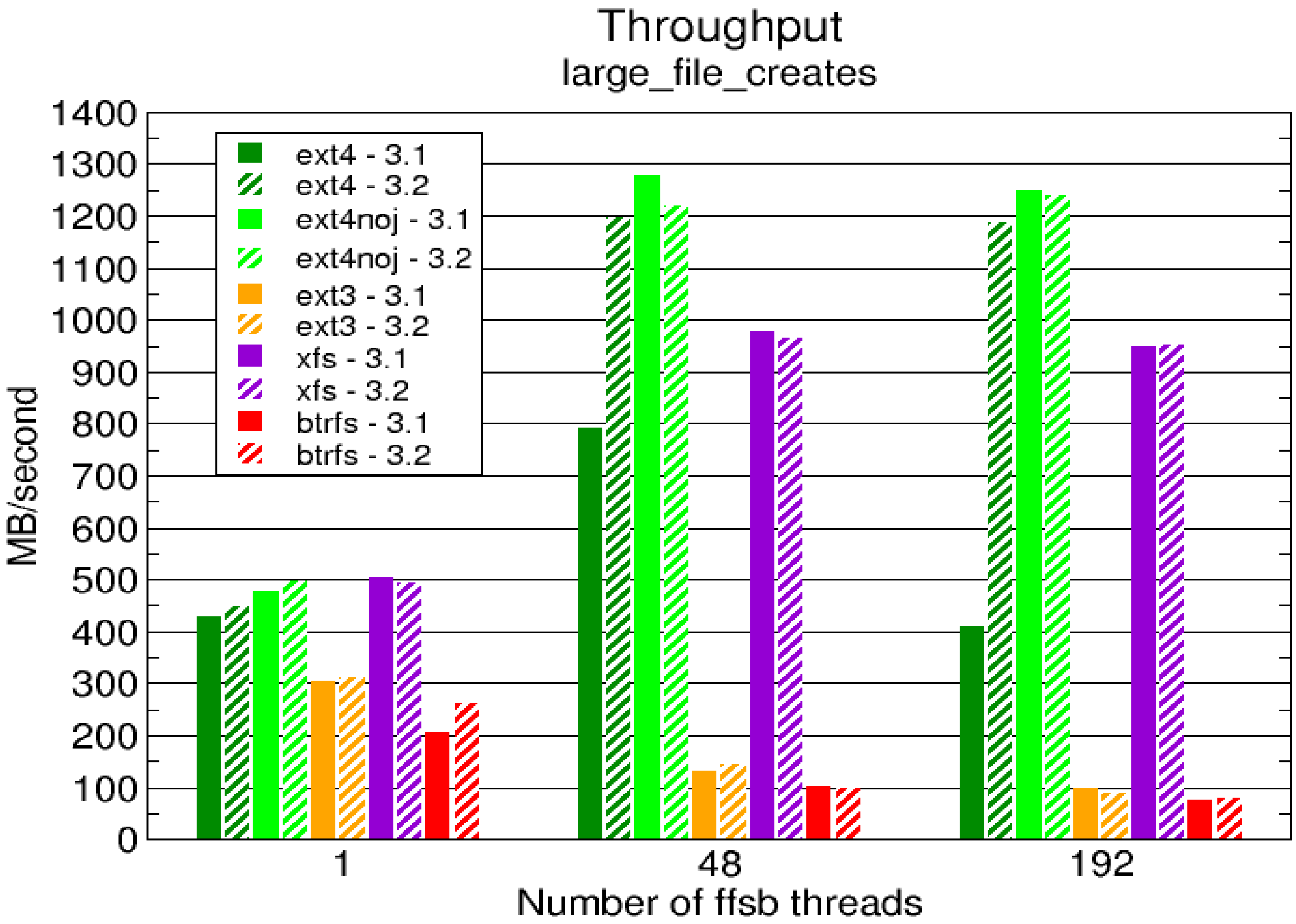


Lies, Damned Lies, and Benchmarks

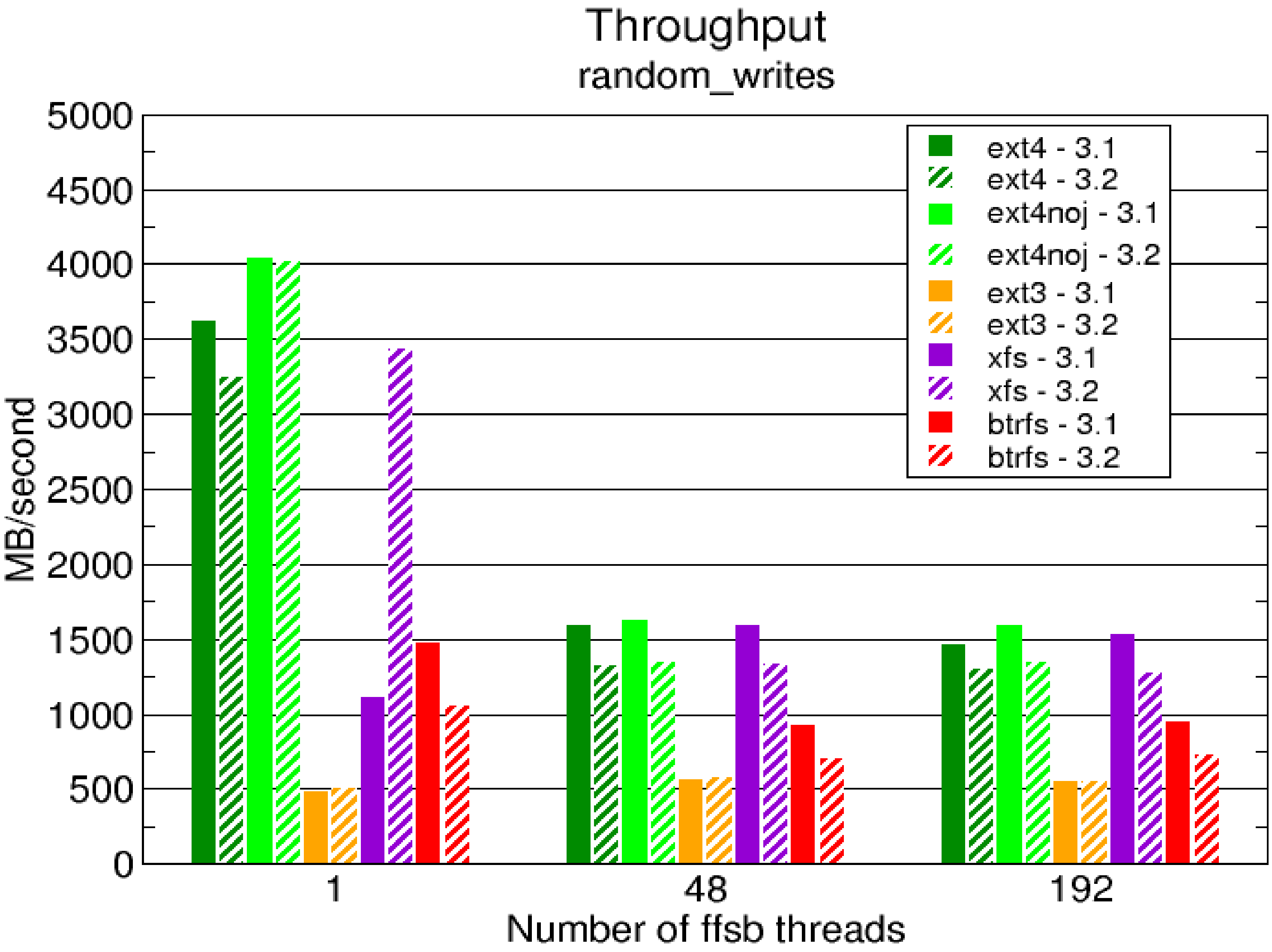


Lies, Damned Lies, and Benchmarks

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



Lies, Damned Lies, and Benchmarks



Lies, Damned Lies, and Benchmarks

- enterpriseestorageforum.com fsck test
- md RAID-60 on DDN LUNS; fs_mark population

FS Size, TB	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

RHEL6 File System Updates

- RHEL 6.2
  - Clustered Samba on GFS2 brings high performance
  - Parallel NFS (pNFS) client supports (tech preview)
  - XFS performance gain for meta-data intensive workloads
- RHEL 6.3
  - GFS2 enhanced performance
  - O_DIRECT support for FUSE file systems
- RHEL 6.4
  - Full support for pNFS client file layout
  - Rebase of btrfs to 3.5 upstream kernel code
  - Hole punch support for ext4
  - Backport of key FUSE patches (scatter-gather IO, readdirplus)

RHEL7 Will Bring in More Choices & Change

- RHEL 7 plans to support ext4, XFS and btrfs (boot, system & data partitions)
- Ext2/Ext3 will be fully supported & use the ext4 driver
  - Should be invisible to the user
  - Reduces code maintenance
- Storage system manager provides a unified easy to use CLI for all supported file systems
  - – FS creation, adding disks to an FS, etc
  - – http://sourceforge.net/p/storagemanager/home

File System Scalability

- Maximum file system size needs to keep up with the ever expanding capacity of storage
- RHEL5 and RHEL6 broke the 16TB limit
  - GFS2 and XFS both raised the limit to 100TB
- RHEL7 limits jump again
  - GFS2 goal of 250TB
  - XFS goal of 500TB
  - Btrfs and ext4 will both exceed 16TB
- Our limits are tested limits, not theoretical ones!

RHEL7 Ongoing Work

- Ease of Use
- Major focus on stability testing of btrfs
    - Looking to see what use cases it fits best (desktop? Local disks without hardware RAID?)
- Harden XFS metadata
    - Detect errors so we can have confidence in 500TB single FS
- Tuning & automation of Local FS to LVM new features
    - Thin provisioned storage
    - Upgrade rollback
    - Scalable snapshots

Resources & Questions

- Mailing lists
  - xfs@oss.sgi.com
  - linux-ext4@vger.kernel.org
  - linux-btrfs@vger.kernel.org
- IRC
  - #xfs on irc freenode.net
  - #ext4, #btrfs on irc.oftc.net
- Questions?