SUMMIT

JBoss WORLD

PRESENTED BY RED HAT

LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

Picking the Right File & Storage System for your Application

Ric Wheeler
Architect & Manager, Red Hat
June 23, 2010

Overview

- Introduction
- Local File Systems
- Networked File Systems
- Shared Disk File Systems
- Storage Overview
- New RHEL6 FS Features
- Performance Results
- Futures

File System Types

- Local file systems
  - ext3, ext4, xfs and btrfs
- Network file systems
  - NFS and CIFS
- Shared disk file systems
  - GFS2
- Cloud file systems

Which File System is Best?

- It always depends on your specific application and circumstances
  - Budget?
  - Performance requirements?
  - Capacity needs?
  - Availability?
  - Robustness in the face of power outages & crashes?
  - IO Workload generated by your application?
- Different answers for every combination of answers!

Data Integrity over System Crash

- Systems can fail for multiple reasons
  - Power outage, hardware fault, software failure
- Modern file systems use a journal mechanism to maintain consistent state
  - Similar to a database transaction
  - Correctness tied to order that data makes it to safe storage
- “barrier” support manages volatile storage device write cache

Alignment on Storage

- Most storage has a preferred IO size and alignment
  - Simple disks have a 512 byte IO size and alignment need
  - New drives move to 4096 byte IO and alignment
- Historic default to sector 63
  - Does not work for some storage at all
  - Can be a big performance hit for some sophisticated storage devices

Discard Support

- File systems now issue “discard” hints to block layer
  - Informs storage of unused ranges of blocks
  - Allows storage to keep an accurate picture of what is utilized
- SSD devices see this as a “TRIM” command
  - Used for wear leveling, pre-erase, etc
- SCSI devices see this as an UNMAP command
  - Used for thinly provisioned LUNs

Overview

- Introduction
- Local File Systems
- Networked File Systems
- Shared Disk File Systems
- Storage Overview
- New RHEL6 FS Features
- Performance Results
- Futures

EXT3 Pros & Cons

- **ext3 is the most common file system in Linux**
  - Most distributions have used it as their default
  - Applications tuned to its specific behaviors
  - 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

EXT4 Pros & Cons

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

XFS Pros and Cons

- 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
  - Performance issues with meta-data intensive (small file creation) workloads

BTRFS

- Btrfs is the newest local file system
  - Has its own internal RAID and snapshot support
  - Does full data integrity checks for metadata and user data
  - Can dynamically grow and shrink
- Supported in RHEL6 as a tech preview item
  - Developers very interested in feedback and testing
  - Not meant for production use!

RHEL5 Local File Systems

- ext3 is our default file system for RHEL5
  - ext4 is supported as a tech preview in (5.4)
- xfs offered as a layered product (5.5+)

RHEL6 Local FS Summary

- FS write barrier enabled for ext3, ext4, gfs2 and xfs
- FS tools warn about unaligned partitions
  - parted/anaconda responsible for alignment
- Size Limitations
  - XFS for any single node & GFS2 for clusters up to 100TB
  - Ext3 & ext4 supported < 16TB

Overview

- Introduction
- Local File Systems
- Networked File Systems
- Shared Disk File Systems
- Storage Overview
- New RHEL6 FS Features
- Performance Results
- Futures

NFS Overview

- Supported by a huge range of hardware
  - NFS servers range from consumer devices up to high end NAS arrays
  - Performance varies with network & hardware
  - Scales up to very large file systems
- Popular uses
  - Users' home directories
  - Read-mostly workloads in scale out configurations of dozens of nodes
- See Steve Dickson's talk on NFS for details

NFS Limitations

- Traditional NFS servers can be a bottleneck
  - Parallel NFS (pNFS) is a new standard that allows direct client to data connections
  - Object, block and file versions
- Does not provide SMP-like coherency for clients
  - Client A needs to wait to see data written by client B
  - Similar issue with newly created files in a directory
  - NFS V4.0 delegations improve this situation

CIFS and Samba

- Samba is a server that speaks Microsoft SMB protocols
  - Allows RHEL to provide networked storage for windows guests
- CIFS is the client side file system that provides access to SMB servers
  - Allows RHEL clients of windows or Samba servers
- See Jeff Layton's CIFS or Simo Sorce's Samba talk for details

Overview

- Introduction
- Local File Systems
- Networked File Systems
- Shared Disk File Systems
- Storage Overview
- New RHEL6 FS Features
- Performance Results
- Futures

Shared Disk File Systems

- Design goal is to provide tight coherence and high availability
  - Avoids most of the issues and lags seen with NFS clients and servers
  - Achieves this by aggressive use of distributed locks
  - Requires shared storage
- Shared disk file systems pay for this tighter coherency
  - Tend be slower than a dedicated local file system
  - Complex to set up and maintain
  - Application tuning needed to avoid lock thrashing

Choosing Between NFS & GFS2?

- GFS2 is a layered product aimed at deployments that need high availability
  - Supported on clusters from 2-16 nodes
  - GFS1 support is dropped in RHEL6
  - Maximum FS size is 100TB
  - Users are encouraged to review configuration with Red Hat
- NFS deployments are much easier to set up and configure

Overview

- Introduction
- Local File Systems
- Networked File Systems
- Shared Disk File Systems
- Storage Overview
- New RHEL6 FS Features
- Performance Results
- Futures

Storage Systems Overview

- Different types of storage have wildly varying performance characteristics
  - Random write?
  - Random read?
  - Streaming read?
  - Streaming write?
- File systems historically have been tuned to run best on traditional, single rotating disk drives
- See Tom Coughlan's talk on storage for details

Traditional Spinning Disk

- Spinning platters store data
  - Modern drives have a large, volatile write cache (16+ MB)
  - Streaming read/write performance of a single S-ATA drive can sustain roughly 100MB/sec
  - Seek latency bounds random IO to the order of 50-100 random IO's/sec
- This is the classic platform that operating systems & applications are designed for
- Write barrier support needed on these devices

External Disk Arrays

- External disk arrays can be extremely sophisticated
  - Large non-volatile cache used to store data
  - IO from a host normally lands in this cache without hitting spinning media
- Performance changes
  - Streaming reads and writes are vastly improved
  - Random writes and reads are fast when they hit cache
  - Random reads can be very slow when they miss cache
- No need for write barrier support on these devices

SSD Devices

- S-ATA interface SSD's
  - Streaming reads & writes are reasonable
  - Random writes normally slow
  - Random reads great!
- PCI-e interface SSD's enhance performance across the board
- Both types of devices tend to use internal DRAM as a buffer
  - Some might need write barrier support

Overview

- Introduction
- Local File Systems
- Networked File Systems
- Shared Disk File Systems
- Storage Overview
- New RHEL6 FS Features
- Performance Results
- Futures

RHEL6 Support for Alignment

- New standards allow storage to inform OS of preferred alignment and IO sizes
  - Few storage devices currently export the information
- Partitions must be aligned using the new alignment variables
  - fdisk, parted, etc snap to proper alignment
  - FS tools warn of misaligned partitions
- Red Hat engineering is actively working with partners to verify and enhance this for our customers

RHEL6 Support for Discard

- File system level feature that informs storage of regions no longer in active use
  - SSD devices see this as a TRIM command and use it to do wear leveling, etc
  - Arrays see this as a SCSI UNMAP command and can enhance thin lun support
- Discard support is off by default
  - Some devices handle TRIM poorly
  - Might have performance impact
  - Test carefully and consult with your storage provider!

RHEL6 NFS Features

- NFS version 4 is the default
  - Per client configuration file can override version 4
  - Negotiates downwards to V3, V2, etc
- Support for industry standard encryption types
- IPV6 Support added for NFS and CIFS
  - NFS clients fully supported in 6.0
  - NFS server support for IPV6 aimed at 6.1

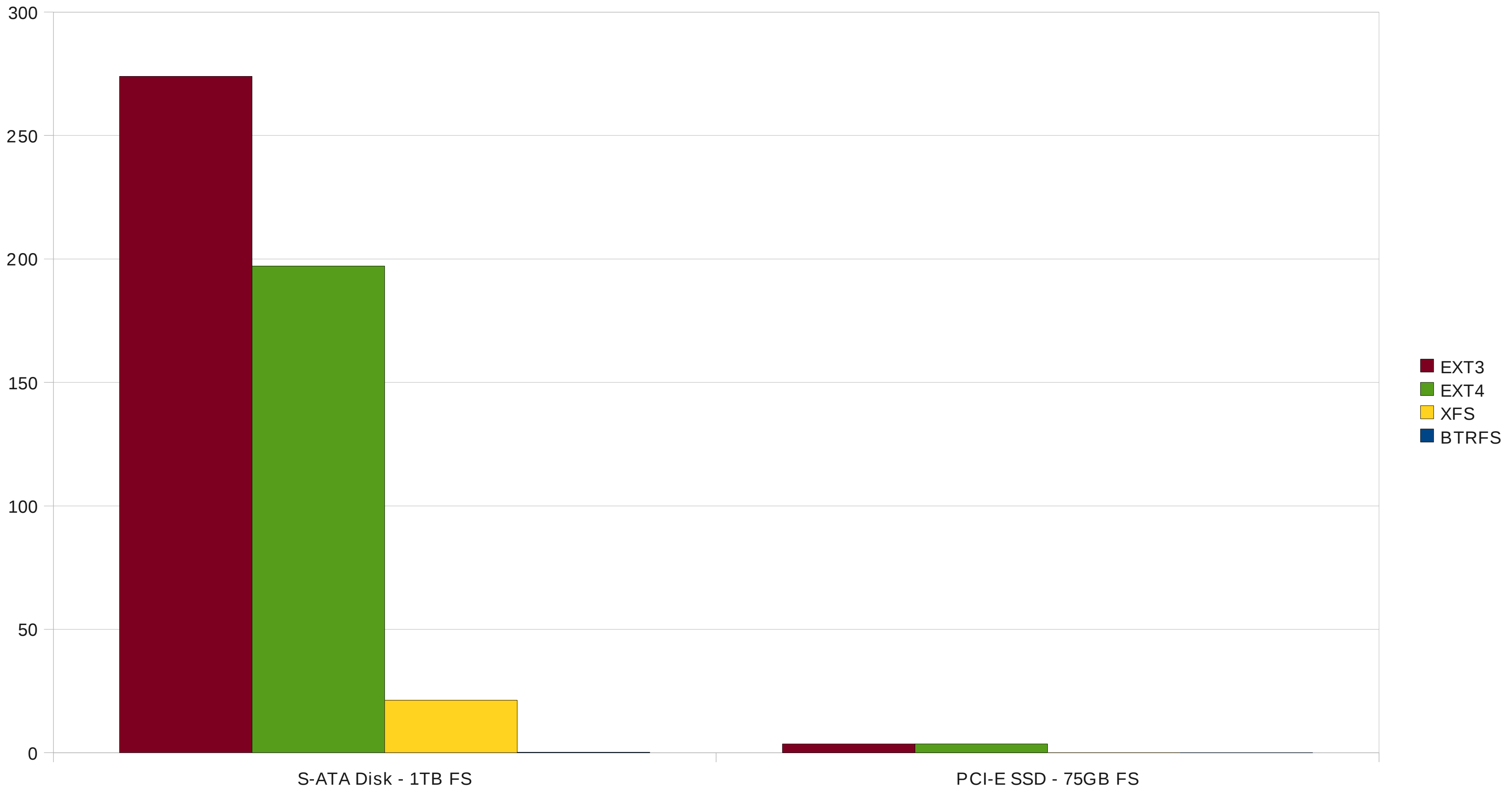
Overview

- Introduction
- Local File Systems
- Networked File Systems
- Shared Disk File Systems
- Storage Overview
- New RHEL6 FS Features
- Performance Results
- Futures

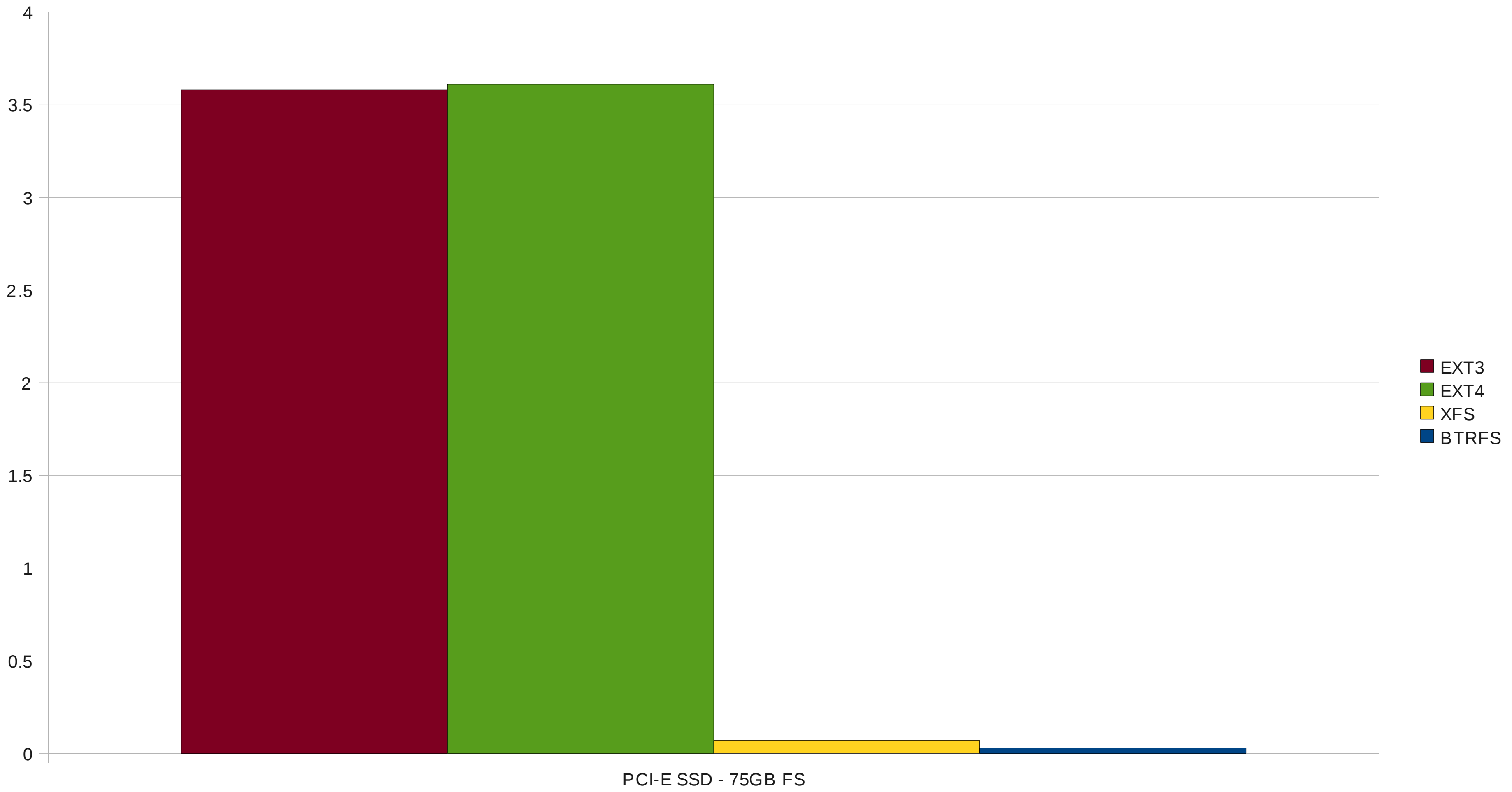
Performance & Measurement

- Workload, storage device and server type all have a huge impact
  - Always measure your actual application on your real system if possible!
  - Same test run on different storage can give opposite results
- Various file systems have special tuning that can help
- See talks by our performance team – Rao & Wagner and Shakshober & Woodman

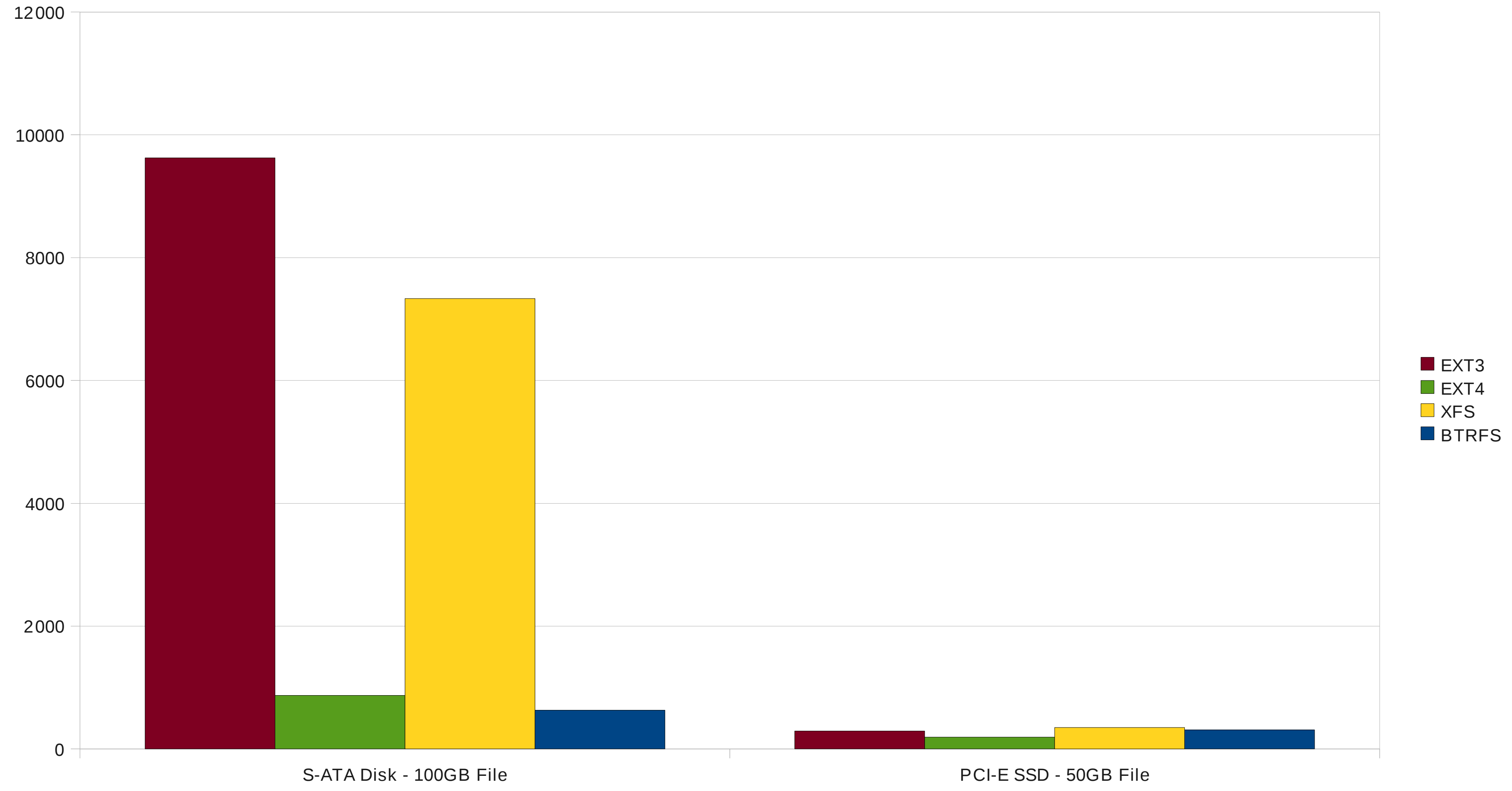
Making a File System – Elapsed Time (sec) Smaller is Better



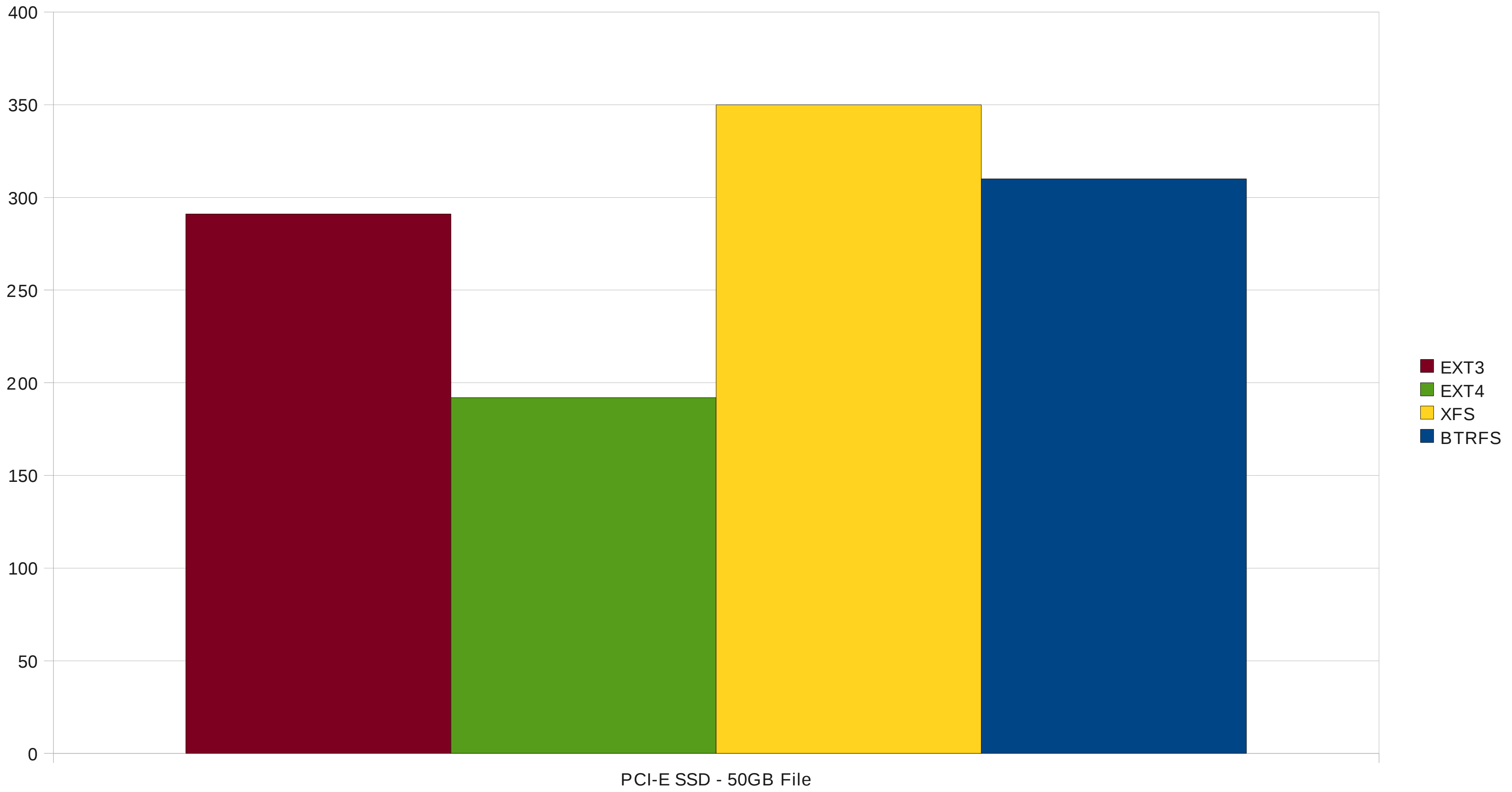
Making a File System – Elapsed Time (sec) Smaller is Better (Zooming in on SSD)



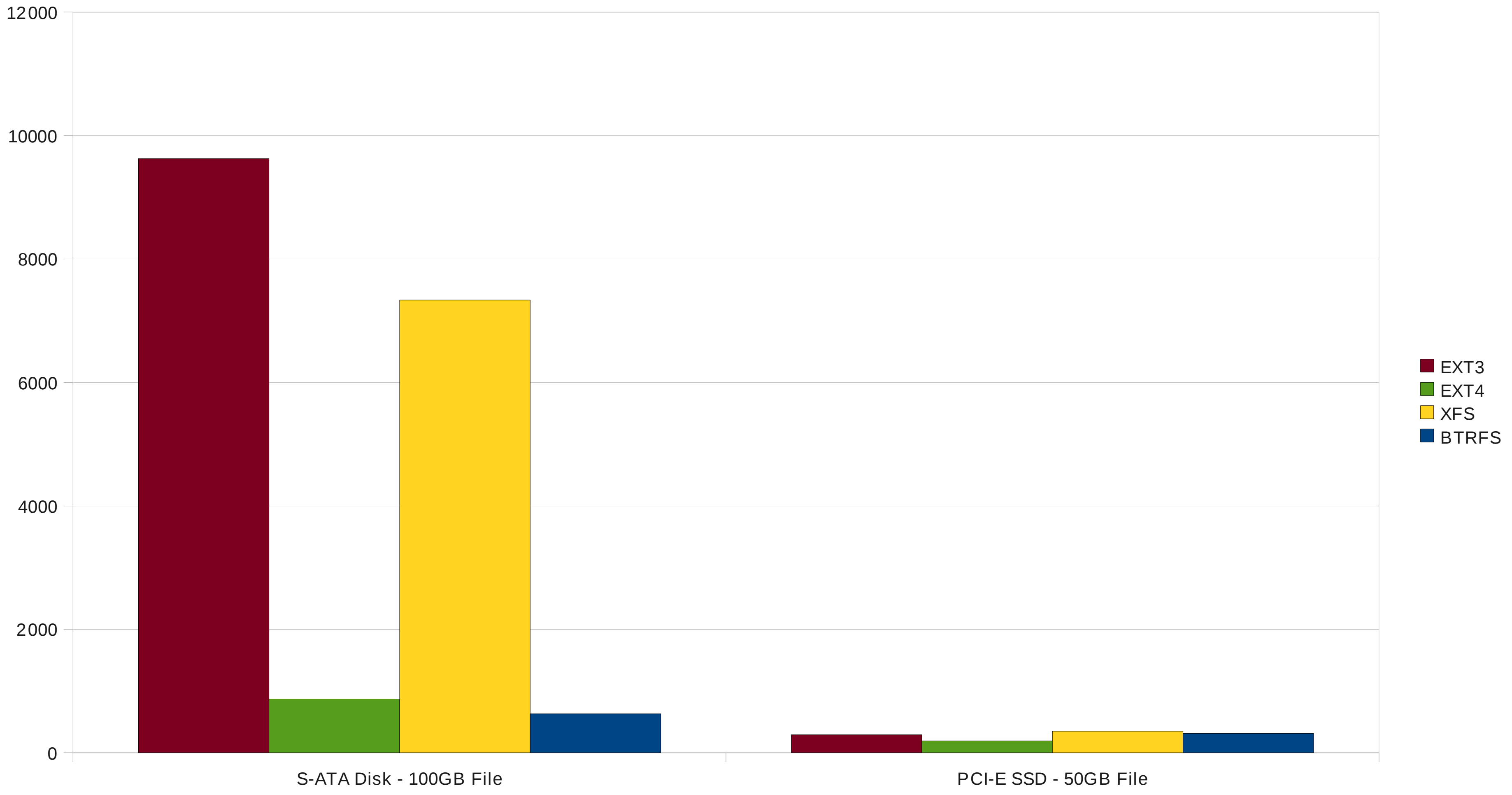
Creating Lots of Small Files – Elapsed Time (sec) Smaller is Better



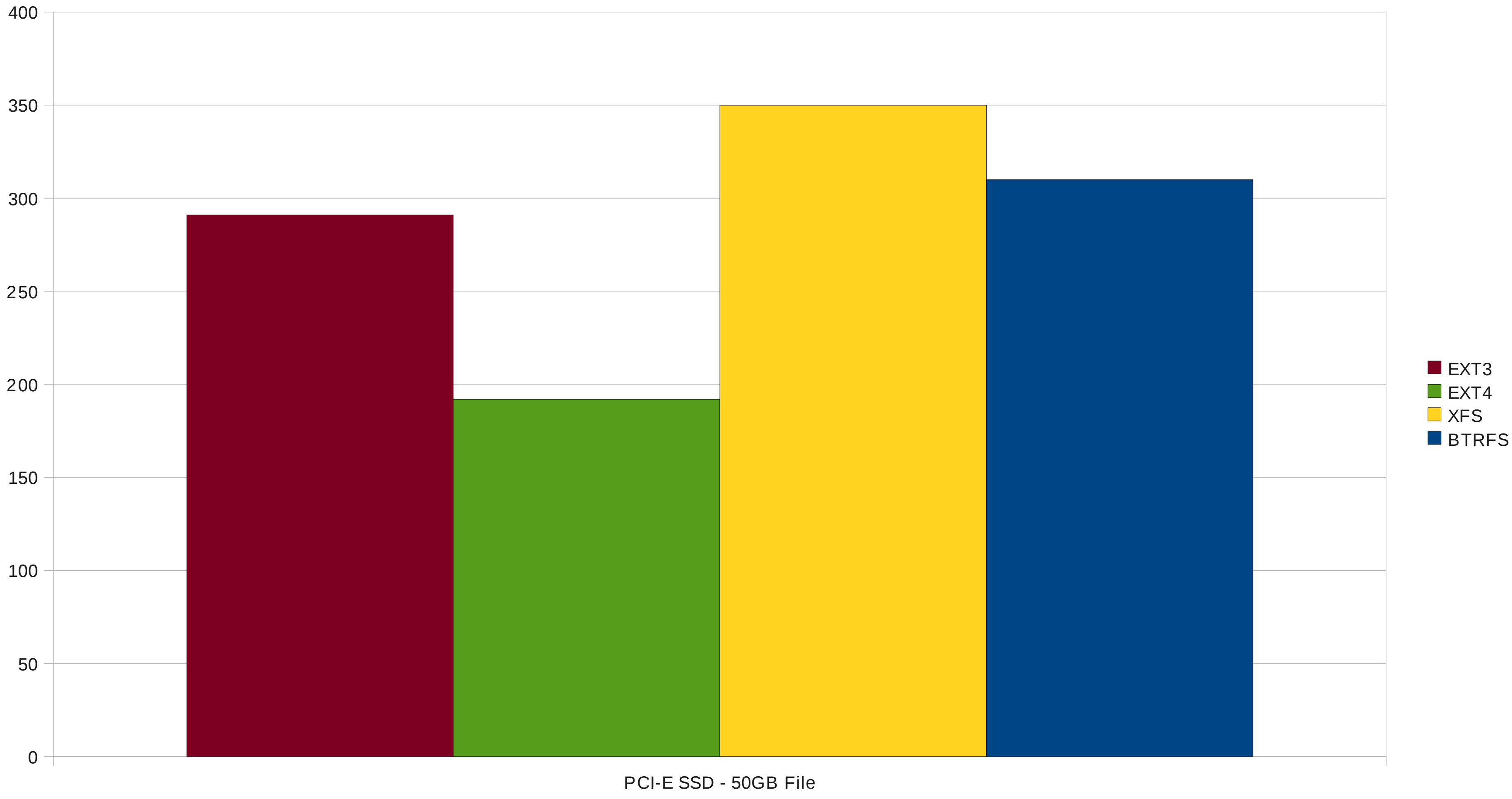
Creating Lots of Small Files – Elapsed Time (sec) Smaller is Better (Zooming in on SSD)



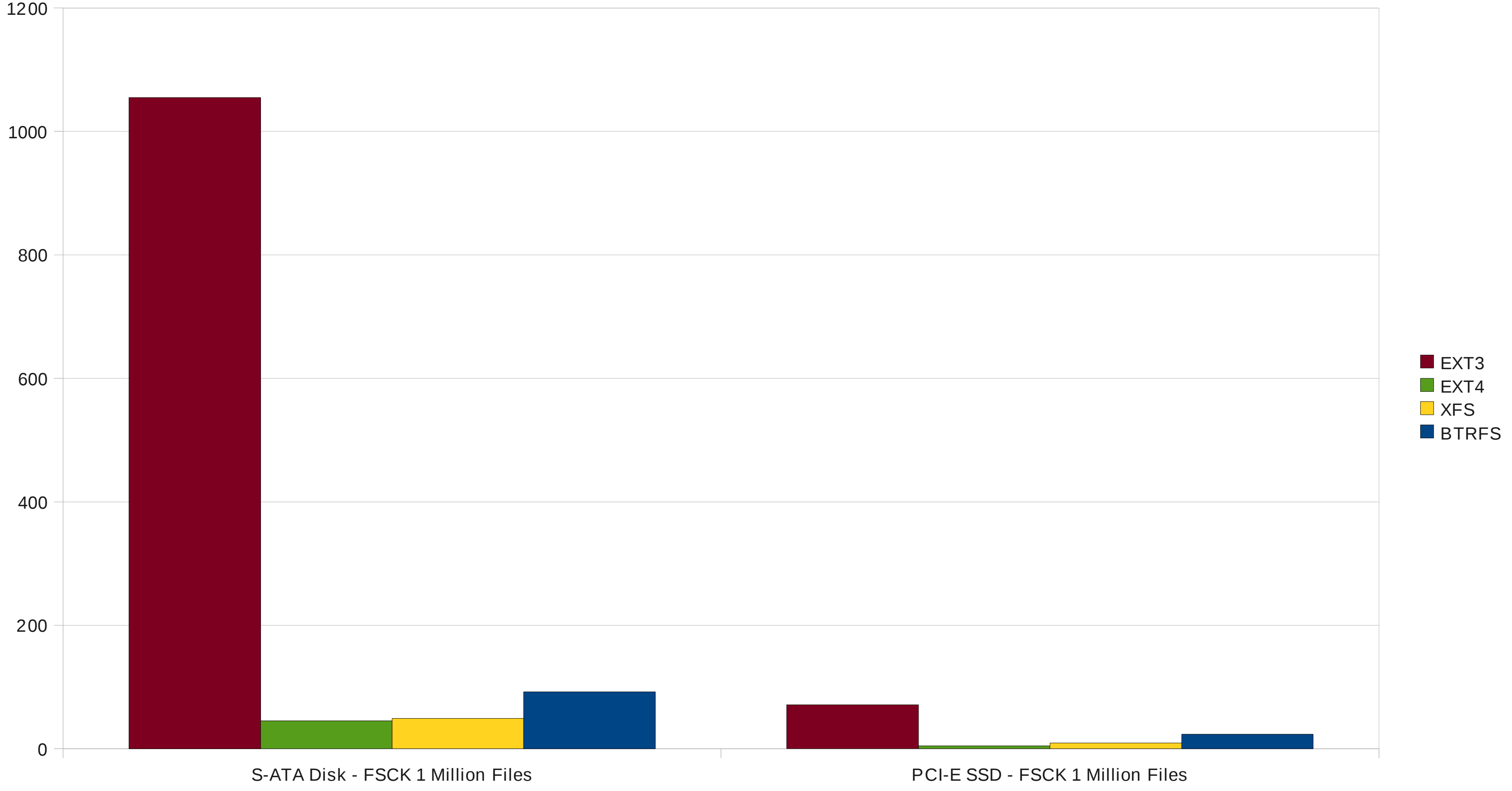
Creating Lots of Small Files – Elapsed Time (sec)
Smaller is Better



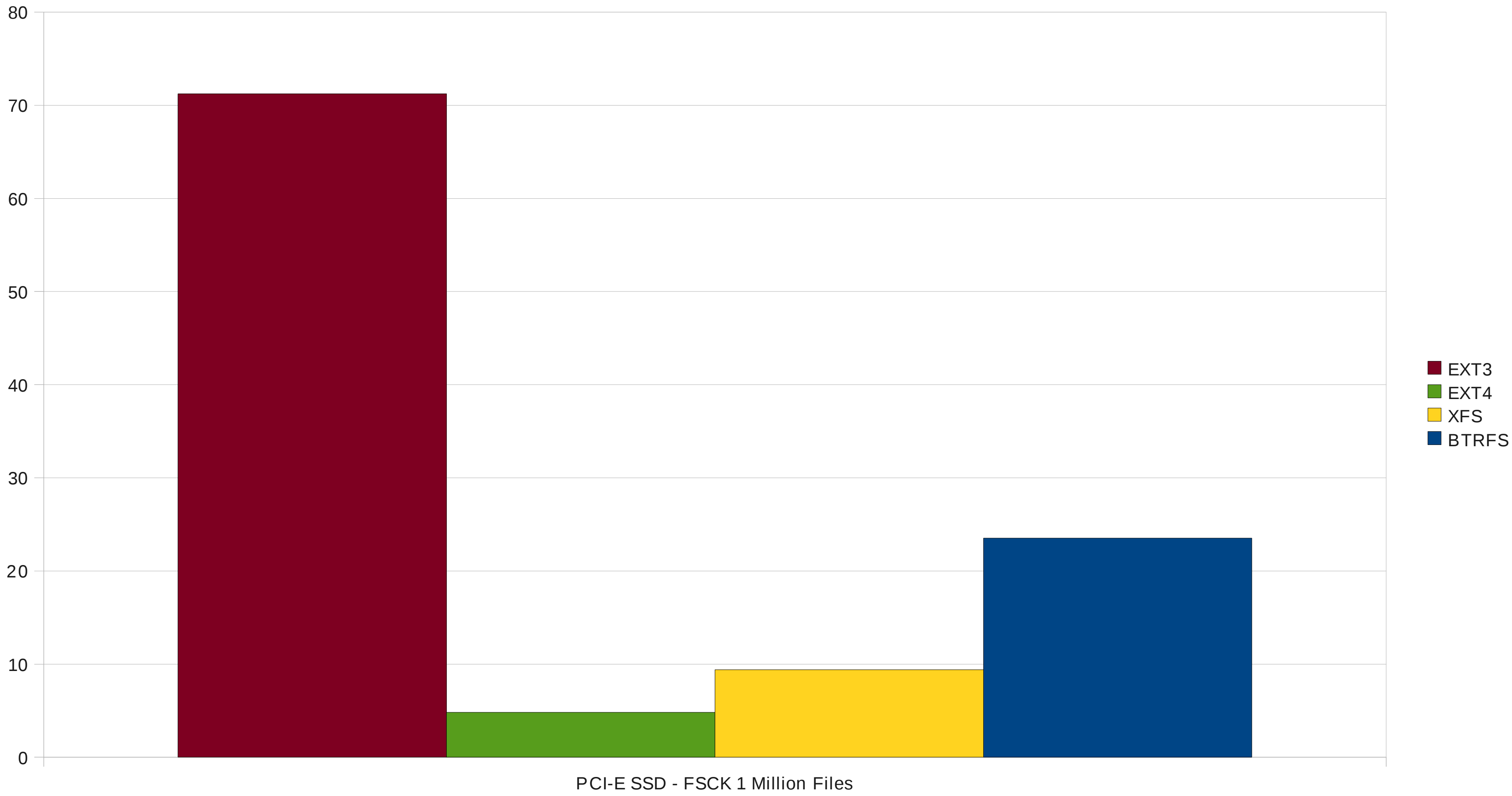
Creating Lots of Small Files – Elapsed Time (sec) Smaller is Better (Zooming in on SSD)



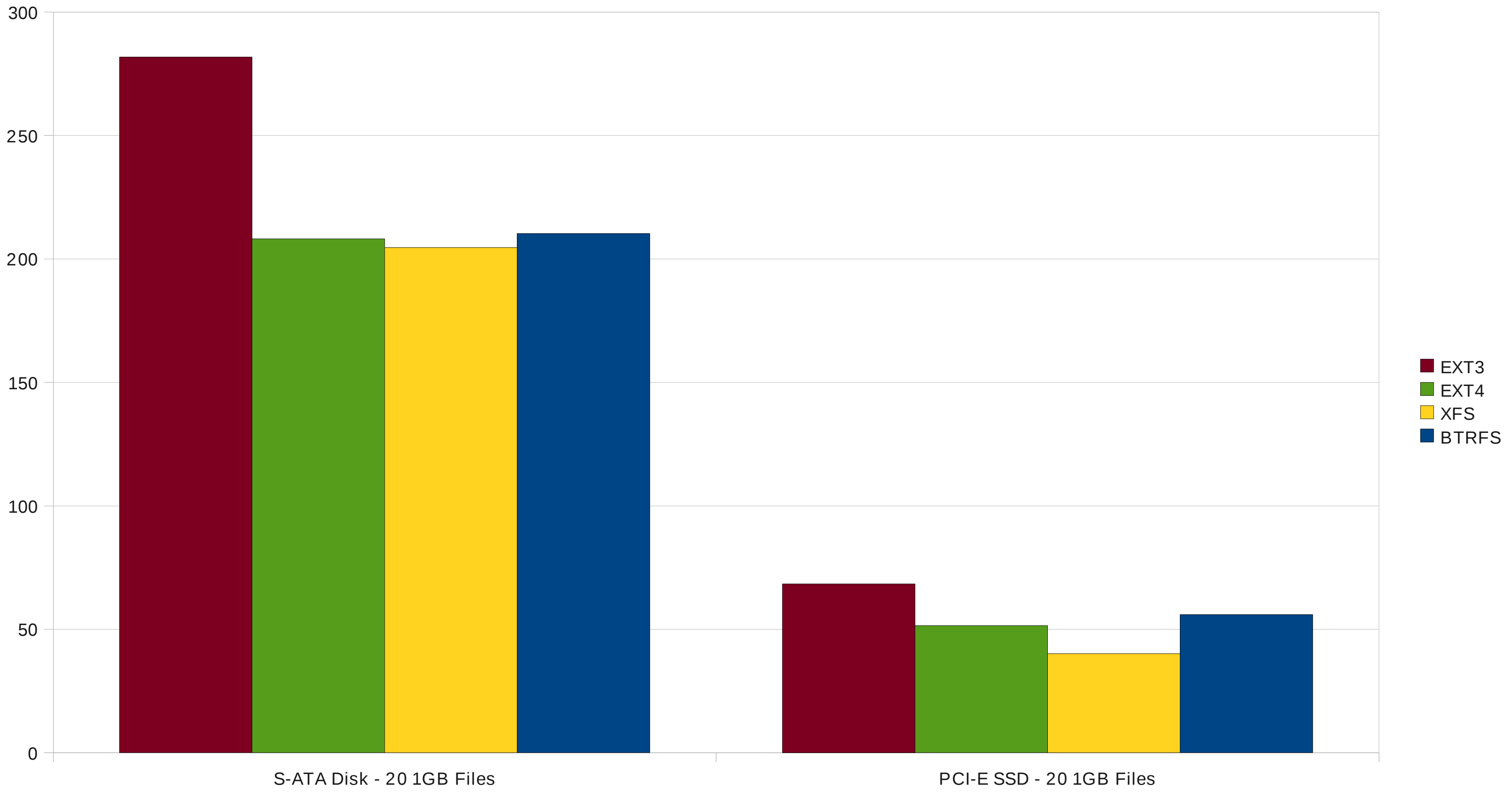
File System Repair – Elapsed Time (secs) Smaller is Better



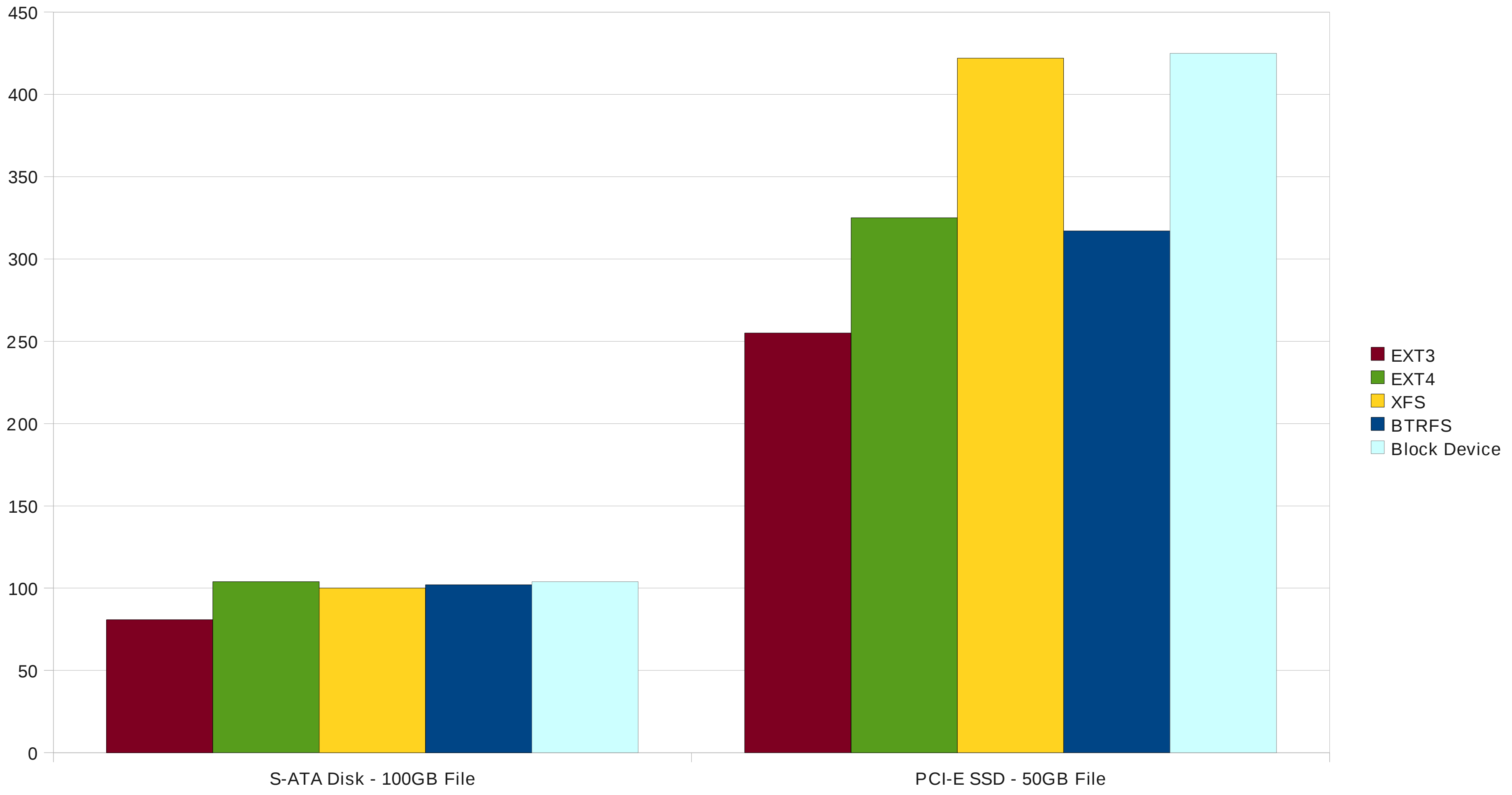
File System Repair – Elapsed Time (secs) Smaller is Better (Zooming in on SSD)



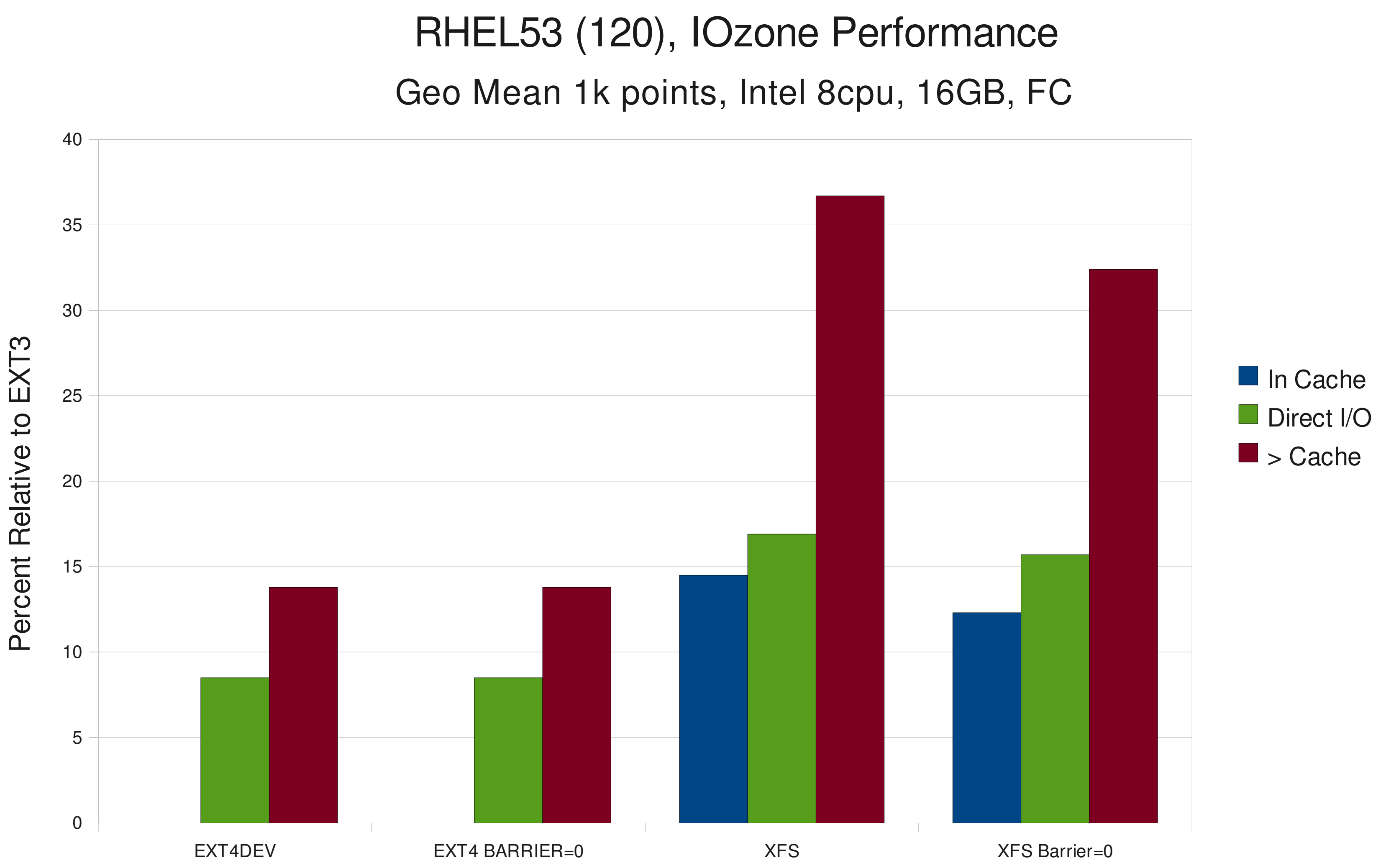
Writing a Few Medium Files – Elapsed Time (secs)
Smaller is Better



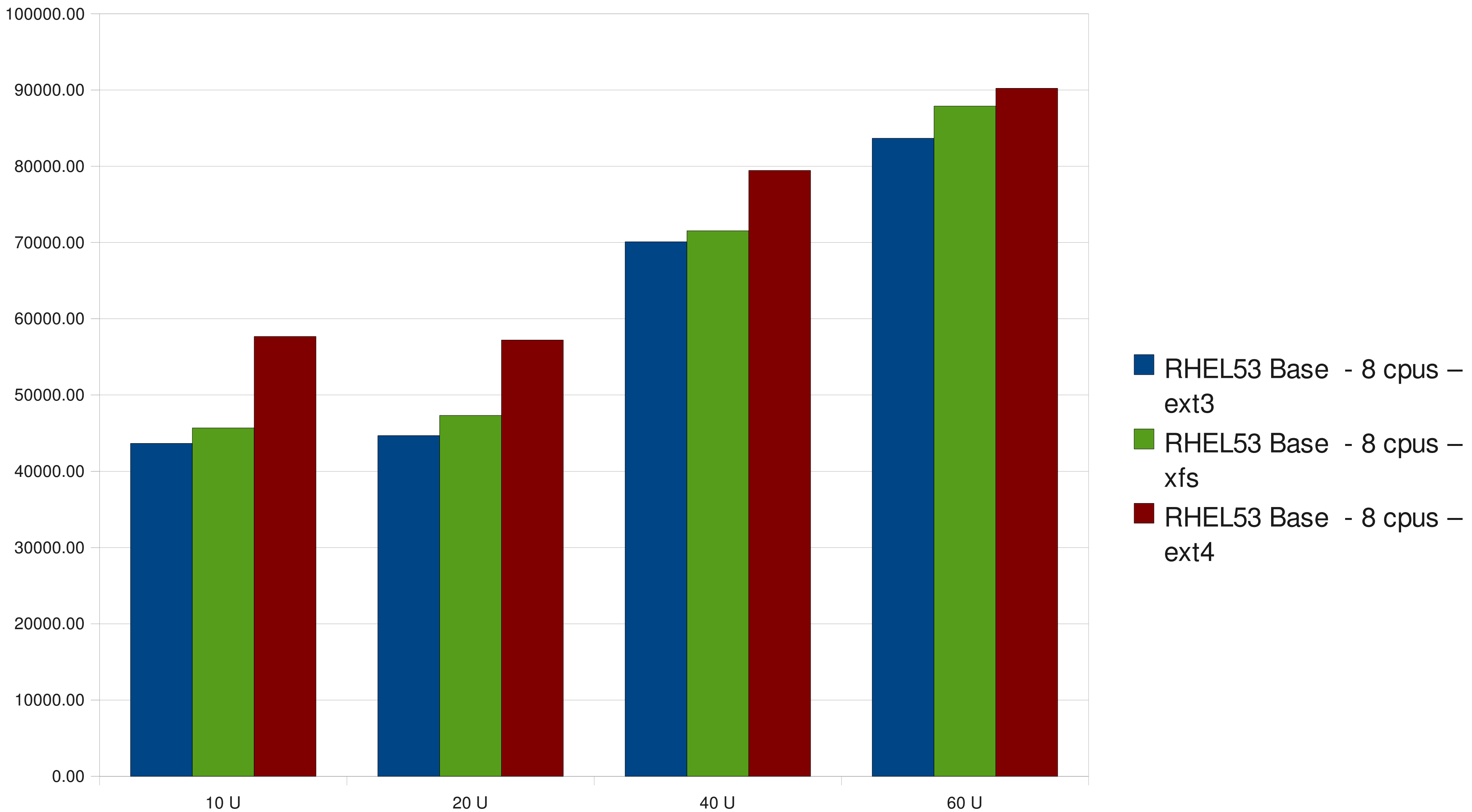
Writing 1 Really Big File – MB/sec
Bigger is Better



RHEL5.3 IOzone EXT3, EXT4, XFS eval
Bigger is Better



RHEL5 Oracle 10.2 Performance Filesystems
Intel 8-cpu, 16GB, 2 FC MPIO, AIO/DIO
Bigger is Better



Performance Summary

- Always measure performance of your application on your real system!
  - No single file system out performs every other one
- Expensive storage can hide performance issues
- Retest when moving to a new OS or application version
- Faster is not always better
  - Trade offs include reduced data integrity
  - Less features like extended attributes, system security

Overview

- Introduction
- Local File Systems
- Networked File Systems
- Shared Disk File Systems
- Storage Overview
- New RHEL6 FS Features
- Performance Results
- Futures

Upcoming Local File System Features

- Union mounts
  - Allow a read-write overlay on top of a read-only base file system
  - Useful for virt guests storage, thin clients, etc
- Continuing to help lead btrfs development towards an enterprise ready state
- Support for ext4 on larger storage
- Enhanced XFS performance for meta-data intensive workloads

Upcoming NFS Features

- PNFS support
  - pNFS and more 4.1 features aimed at a minor 6.x release
  - No commercial arrays support pNFS yet
  - Ongoing work on open source (GFS2, object, etc) pNFS servers
- Working with standards body to add support for passing extended attributes over NFS
  - Goal is to enable SELinux over NFS

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#summitjbw

READ THE BLOG

http://summitblog.redhat.com/