RED HAT SUMMIT

Linux File Systems
Enabling Cutting Edge Features in RHEL 6 & 7

Ric Wheeler, Senior Manager

Steve Dickson, Consulting Engineer

File and Storage Team
Red Hat, Inc.
June 12, 2013

redhat.

Red Hat Enterprise Linux 6

RHEL6 File and Storage Themes

- LVM Support for Scalable Snapshots and Thin Provisioned Storage
- Expanded options for file systems
- General performance enhancements
- Industry leading support for new pNFS protocol

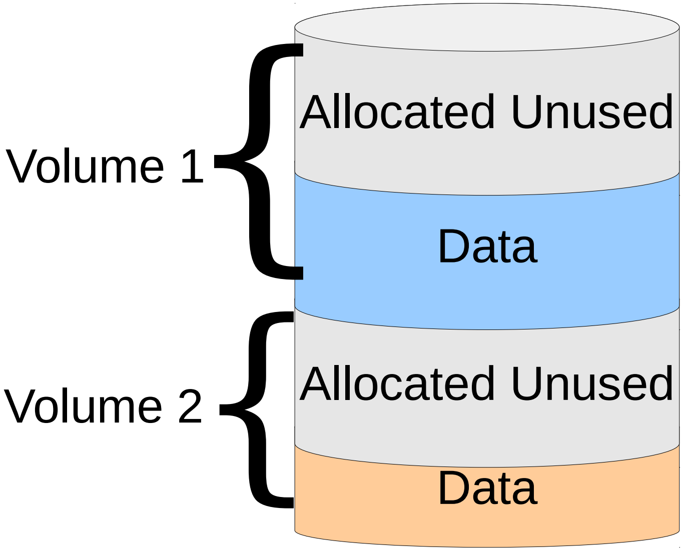
RHEL6 LVM Redesign

- Shared exception store
  - Eliminates the need to have a dedicated partition for each file system that uses snapshots
  - Mechanism is much more scalable
- LVM thin provisioned target
  - dm-thinp is part of the device mapper stack
  - Implements a high end, enterprise storage array feature
  - Makes re-sizing a file system almost obsolete!

RHEL Storage Provisioning
Improved Resource Utilization & Ease of Use

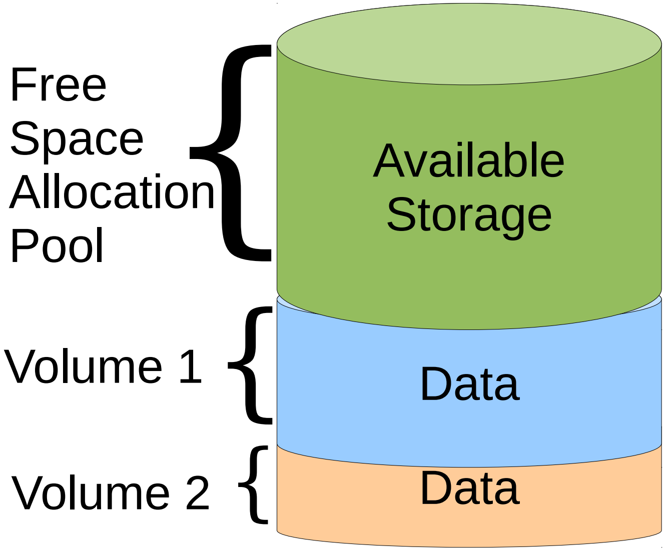
Avoids wasting space
Less administration
* Lower costs *

Traditional Provisioning



Inefficient resource utilization
Manually manage unused storage

RHEL Thin Provisioning

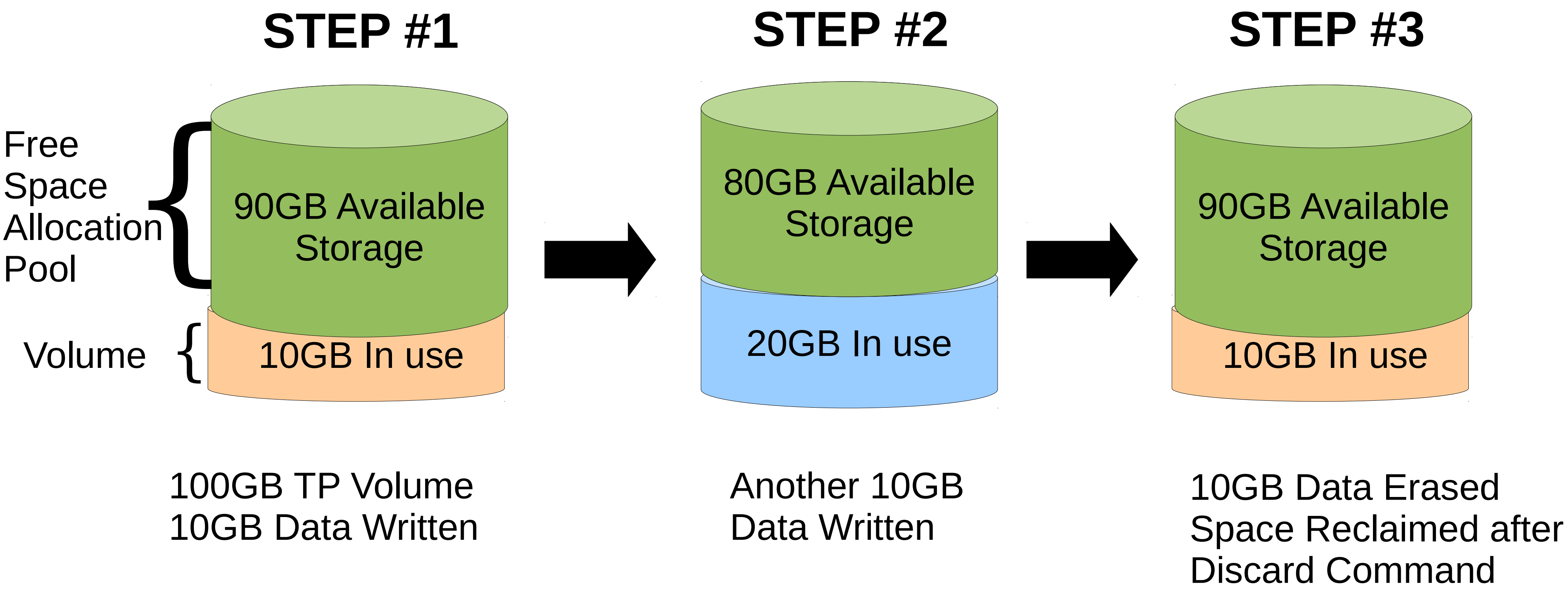


Efficient resource utilization
Automatically managed

Thin Provisioned Targets Leverage Discard

- Red Hat Enterprise Linux 6 introduced support to notify storage devices when a blocks becomes free in a file system
  - Useful for wear leveling in SSD's or space reclamation in thin provisioned storage
- Discard maps to the appropriate low level command
  - TRIM for SSD devices
  - UNMAP or WRITE_SAME for SCSI devices
  - dm-thinp handles the discard directly

RHEL Storage Provisioning
Improved Resource Utilization



STEP #1

Free Space Allocation Pool

90GB Available Storage

Volume

10GB In use

100GB TP Volume
10GB Data Written

STEP #2

80GB Available Storage

20GB In use

Another 10GB
Data Written

STEP #3

90GB Available Storage

10GB In use

10GB Data Erased
Space Reclaimed after
Discard Command

Space Reclamation with Thin Provisioned Volumes

Even More LVM Features

- Red Hat Enterprise Linux LVM commands can control software RAID (MD) devices
  - 6.4 added LVM Support for RAID10
- Introduction of a new user space daemon
  - Daemon stores the device information after a scan in its caches
  - Major performance gains when systems have a large number of disks

Expanding Choices

- Early in RHEL5 there are limited choices in the file system space
  - Ext3 was the only local file system
  - GFS1 was your clustered file system
- RHEL5 updates brought in support for
  - Ext4 added to the core Red Hat Linux platform
  - Scalable File System (XFS) as a layered product
  - GFS2 as the next generation of clustered file system
  - Support for user space file systems via FUSE support

RHEL6 File System Highlights

- Scalable File System (XFS) enhancements support the most challenging workloads
  - Performance enhancements for meta-data workloads
  - Selected as the base file system for Red Hat Storage
- GFS2 reached new levels of performance
  - Base file system for clustered Samba servers
- XFS performance enhancements for synchronous workloads
- Support for Parallel NFS file layout client

RHEL6.4 File System Updates

- Ext4 enhanced for virtual guest storage in RHEL6.4
  - “hole punch” deallocates data in the middle of files
- Refresh of the btrfs file system technology preview to 3.5 upstream version
- Scalable File System (aka XFS) is a layered product for the largest and most demanding workloads
  - Series of performance enhancements learned courtesy of Red Hat Storage and partners
  - Refresh of key updates from the upstream Linux kernel

How to Choose a Local File System?

- The best way is to test each file system with your specific workload on a well tuned system
  - Make sure to use the RHEL tuned profile for your storage
- The default file system will just work for most applications and servers
  - Many applications are not file or storage constrained
  - If you are not waiting on the file system, changing it probably will not make your application faster!

SAS on Standalone Systems

Picking a RHEL File System

xfs **most** recommended

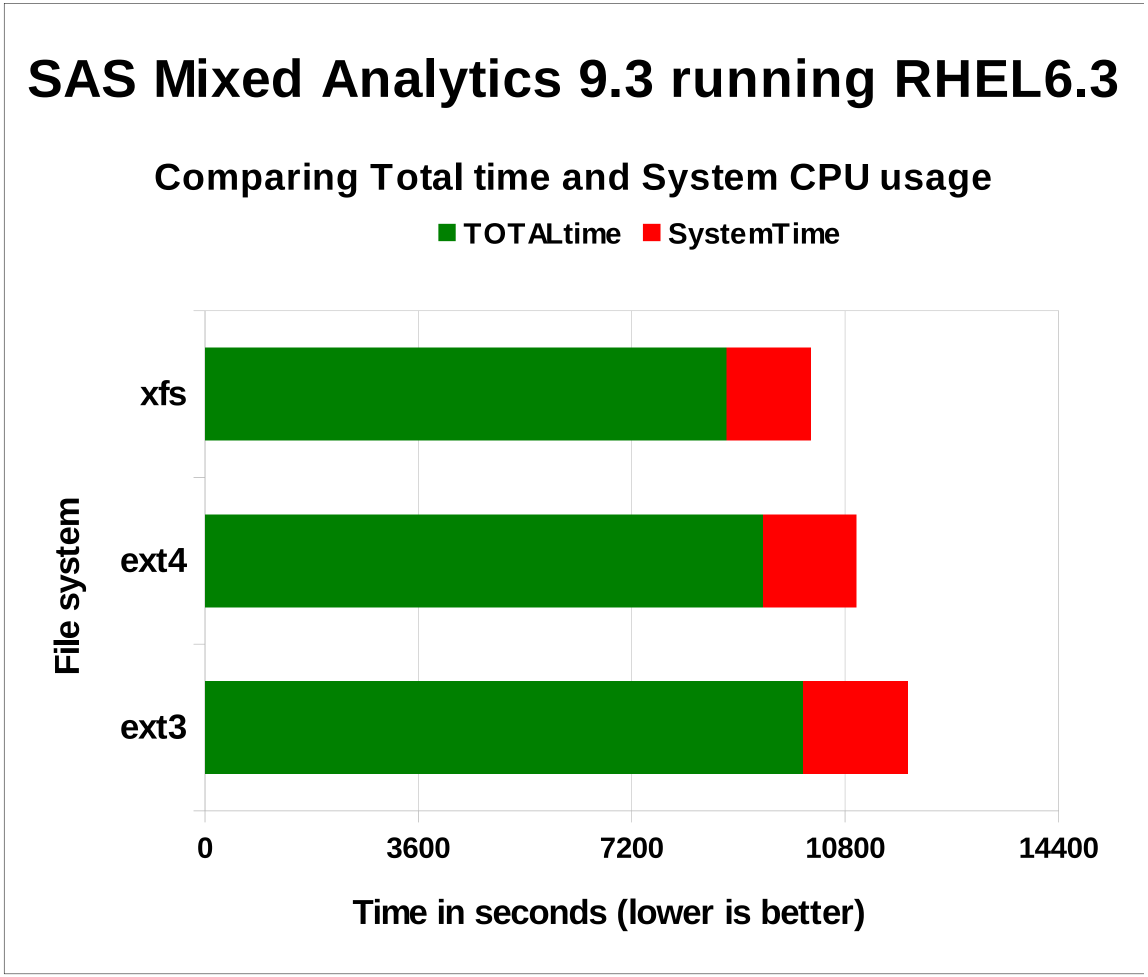
- Max file system size 100TB
- Max file size 100TB
- Best performing

ext4 recommended

- Max file system size 16TB
- Max file size 16TB

ext3 not recommended

- Max file system size 16TB
- Max file size 2TB



Red Hat Enterprise Linux 6
GFS2 New Features

- Performance improvements
  - Sorted ordered write list for improved log flush speed & block reservation in the allocator for better on-disk layout with complex workloads (6.4+)
  - Orlov allocator & better scalability with very large number of cached inodes (6.5+)
- Faster glock dump for debugging (6.4+)
- Support for 4k sector sized devices with TRIM (6.5+)

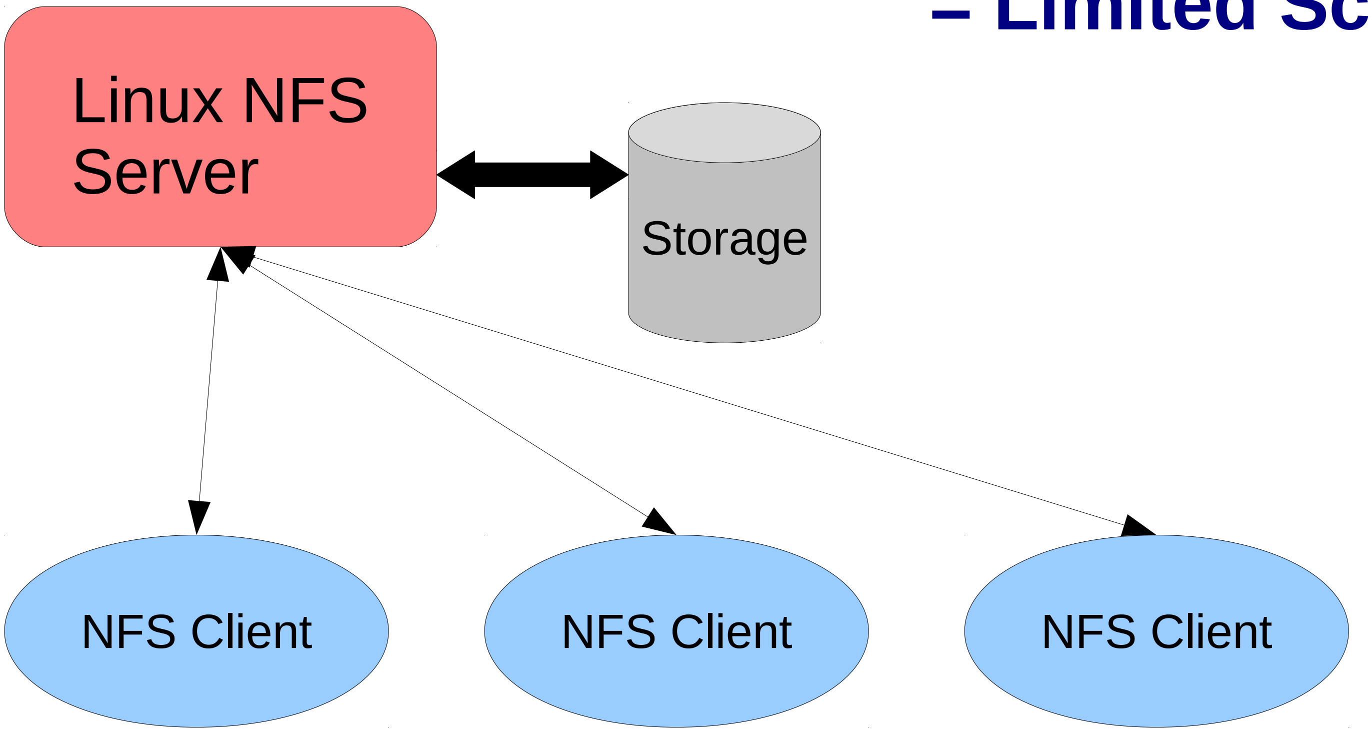
Performance Enhancement for User Space File Systems

- User space file systems are increasingly popular
  - Cloud file systems and “scale out” file systems like HDFS or gluster
- FUSE is a common kernel mechanism for this class of file system
- Work to enhance the performance of FUSE includes
  - Support for FUSE readdirplus()
  - Support for scatter-gather IO
  - Reduces trips from user space to the kernel

RHEL6 NFS Updates

Traditional NFS

One Server for Multiple Clients
= Limited Scalability

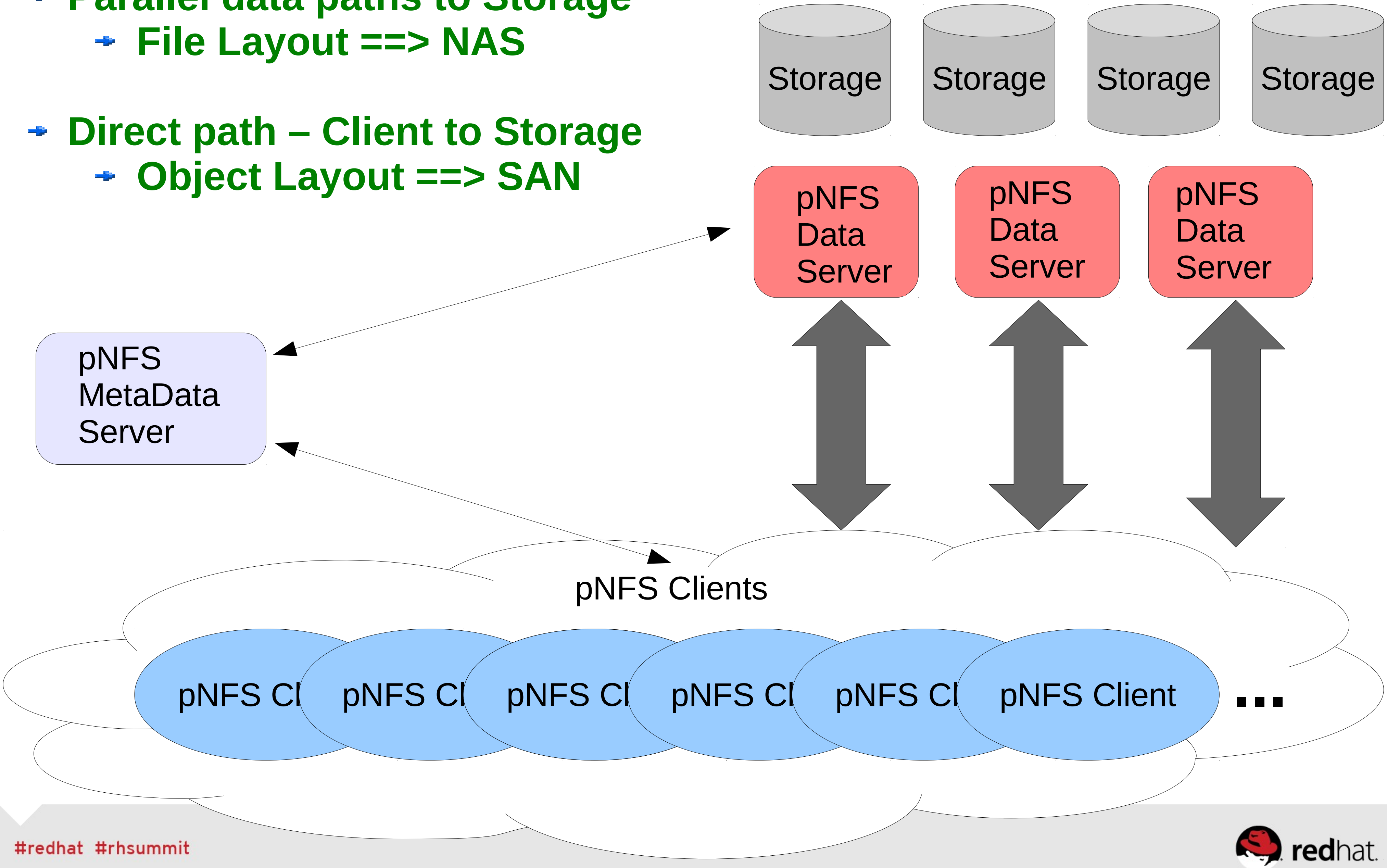


Parallel NFS (pNFS)

- Architecture
  - Metadata Server (MDS) – Handles all non-Data Traffic
  - Data Server (DS) – Direct I/O access to clients
  - Shared Storage Between Servers
- Layout Define server Architecture
  - File Layout (NAS Env) - Netapp
  - Block Layout (SAN Env) - EMC
  - Object Layout (High Perf Env) Pananas & Tonian

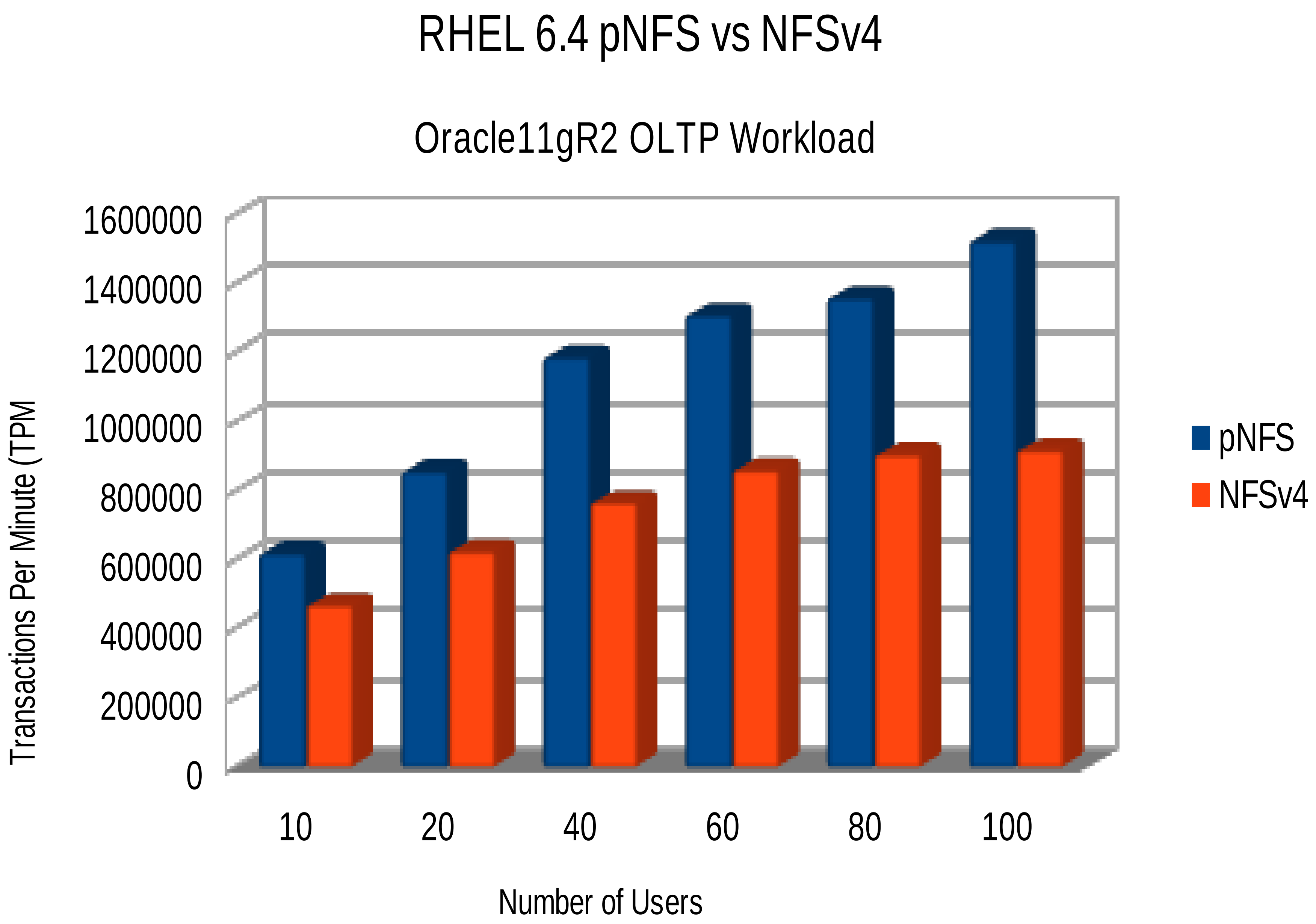
Parallel NFS = Scalability

- Parallel data paths to Storage
  - File Layout ==> NAS
- Direct path – Client to Storage
  - Object Layout ==> SAN

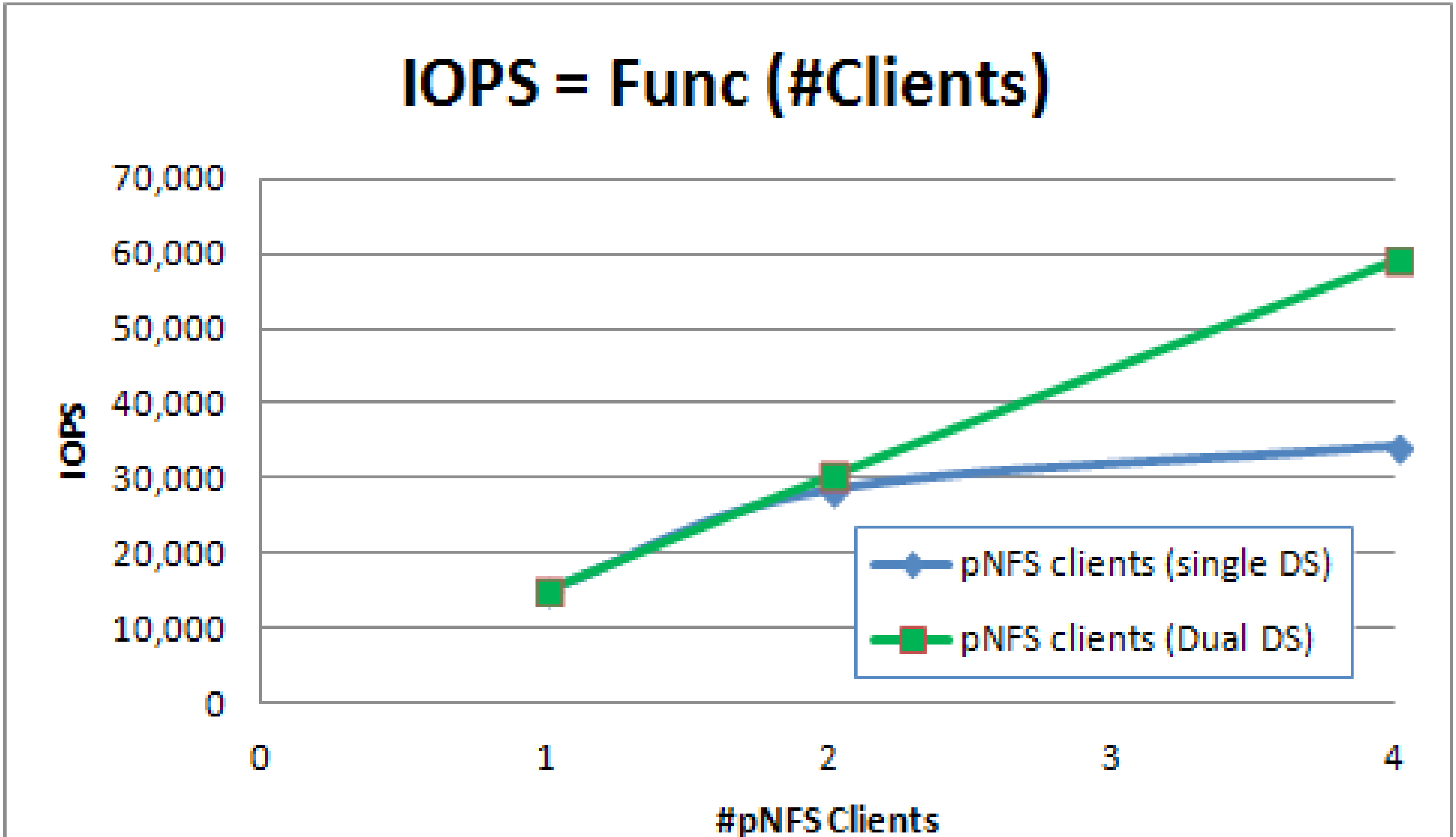


Parallel NFS (pNFS) - RHEL 6.4

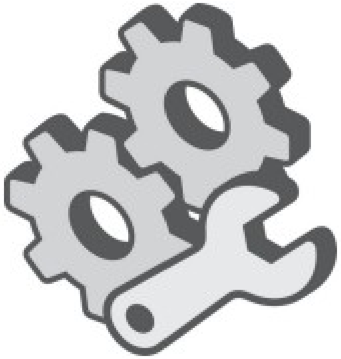
- First to market with Client support (file layout)
  - Thank you very much Upstream and Partners!!!
- Enabling pNFS:
  - **mount -o v4.1 server:/export /mnt/export**
- RHEL-Next
  - Block and Object layout support


RHEL 6.4 pNFS vs NFSv4
Oracle11gR2 OLTP Workload
Transactions Per Minute (TPM
1600000
1400000
1200000
1000000
800000
600000
400000
200000
0
10
20
40
60
80
100
Number of Users
pNFS
NFSv4

Parallel NFS = High Performance and Scalability



Source: Tonian Systems



Red Hat Enterprise Linux 7

RHEL7 Areas of Focus

- Enhanced performance
  - Focus on very high performance, low latency devices
- Support for new types of hardware
  - Working with our storage partners to enable their latest devices
- Support for higher capacities across the range of file and storage options
- Ease of use and management

RHEL7 Storage Updates

Storage Devices are too Fast for the Kernel!

- We are too slow for modern SSD devices
  - The Linux kernel did pretty well with just S-ATA SSD's
  - PCI-e SSD cards can sustain 500,000 or more IOPs and our stack is the bottleneck in performance
- A new generation of *persistent memory* is coming that will be
  - Roughly the same capacity, cost and performance as DRAM
  - The IO stack needs to go to millions of IOPs
  - http://lwn.net/Articles/547903

Dueling Block Layer Caching Schemes

- With all classes of SSD's, the cost makes it difficult to have a purely SSD system at large capacity
  - Obvious extension is to have a block layer cache
- Two major upstream choices:
  - Device mapper dm-cache target will be in RHEL7
  - BCACHE queued up for 3.10 kernel, might make RHEL7
- Performance testing underway
  - BCACHE is finer grained cache
  - Dm-cache has a pluggable policy (similar to dm MPIO)
- https://lwn.net/Articles/548348

Thinly Provisioned Storage & Alerts

- Thinly provisioned storage lies to users
  - Similar to DRAM versus virtual address space
  - Sys admin can give all users a virtual TB and only back it up with 100GB of real storage for each user
- Supported in arrays & by device mapper dm-thinp
- Trouble comes when physical storage nears its limit
  - Watermarks are set to trigger an alert
  - Debate is over where & how to log that
  - How much is done in kernel versus user space?
- User space policy agent was slightly more popular
- http://lwn.net/Articles/548295

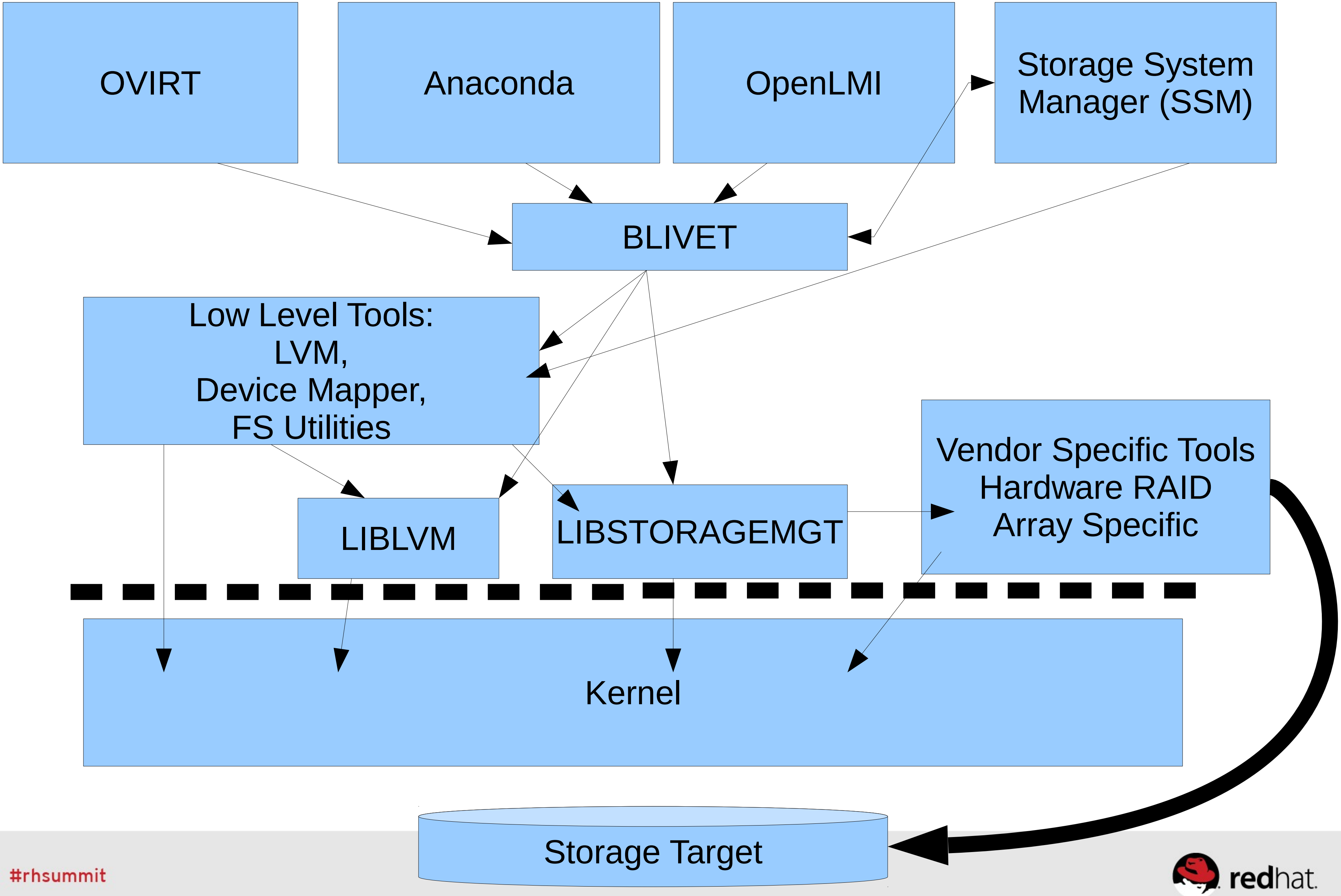
Continued LVM Enhancements for Software RAID

- LVM will support more software RAID features
  - Scrubbing proactively detects latent disk errors in software RAID devices
  - Reshape moves a device from one RAID level to another
  - Support for write-mostly, write-behind, resync throttling all help performance
- BTRFS will provide very basic software RAID capabilities
  - Still very new code

Storage Management APIs

- Unification of the code that does storage management
  - Low level libraries with C & Python bindings
  - libstoragemgt manages SAN and NAS
  - liblvm is the API equivalent of LVM user commands
  - API's in the works include HBA, SCSI, iSCSI, FcoE, multipath
- Storage system manager provides an easy to use command line interface
- Blivet is a new high level storage and file system library that will be used by anaconda and openlmi

Future Red Hat Stack Overview


OVIRT
Anaconda
OpenLMI
Storage System Manager (SSM)
BLIVET
Low Level Tools: LVM, Device Mapper, FS Utilities
Vendor Specific Tools Hardware RAID Array Specific
LIBLVM
LIBSTORAGEMGT
Kernel
Storage Target

RHEL7 Local File System Updates

RHEL7 Will Bring in More Choices

- RHEL 7 is looking to support ext4, XFS and btrfs
  - All can be used for boot, system & data partitions
  - Btrfs going through intense testing and qualification
- Ext2/Ext3 will be fully supported
  - Use the ext4 driver which is mostly invisible to users
- Full support for all pNFS client layout types
  - Add in support for vendors NAS boxes which support the pNFS object and block layouts

RHEL7 Default File System

- In RHEL7, Red Hat is looking to make XFS the new default
  - XFS will be the default for boot, root and user data partitions on all supported architectures
- Red Hat is working with partners and customers during this selection process to test and validate XFS
  - Final decision will be made pending successful testing
- Evaluating maximum file system sizes for RHEL7

XFS Strengths

- XFS is the reigning champion of larger servers and high end storage devices
  - Tends to extract the most from the hardware
  - Well tuned to multi-socket and multi-core servers
- XFS has a proven track record with the largest systems
  - Carefully selected RHEL partners and customers run XFS up to 300TB in RHEL6
- Popular base for enterprise NAS servers
  - Including Red Hat Storage

EXT4 Strengths

- Ext4 is very well know to system administrators and users
  - Default file system in RHEL6
  - Closely related to ext3 our RHEL5 default
- Can outperform XFS in some specific workloads
  - Single threaded, single disk workload with synchronous updates
- Avoid ext4 for larger storage
- Base file system for Android and Google File System

BTRFS – The New File System Choice

- Integrates many block layer functions into the file system layer
  - Logical volume management functions like snapshots
  - Can do several versions of RAID
- Designed around ease of use and has sophisticated metadata
  - Back references help map IO errors to file system objects
- Great choice for systems with lots of independent disks and no hardware RAID

RHEL7 NFS Updates

RHEL7 NFS Server Updates

- Red Hat Enterprise Linux 7.0 completes the server side support for NFS 4.1
  - Support for only-once semantics
  - Callbacks use port 2049
- No server side support for parallel NFS ... yet!

Parallel NFS Updates

- Parallel NFS has three layout types
  - Block layouts allow direct client access to SAN data
  - Object layouts for direct access to the object backend
  - File layout
- RHEL7.0 will add support for block and object layout types
  - Will provide support for all enterprise pNFS servers!

Support for SELinux over NFS

- Labeled NFS enable fine grained SELinux contexts
    - Part of the NFS4.2 specification
- Use cases include
    - Secure virtual machines stored on NFS server
    - Restricted home directory access

Red Hat Enterprise Linux 7.9
GFS2 New Features

- All of the previously mentioned RHEL6 features
- Streamlined journaling code
    - Less memory overhead
    - Less prone to pauses during low memory conditions
- New cluster stack interface (no gfs_controld)
- Performance co-pilot (PCP) support for glock statistics
- Faster fsck
- RAID stripe aware mkfs

Pulling it all Together....

- Ease of Use
- Tuning & automation of Local FS to LVM new features
    - Thin provisioned storage
    - Upgrade rollback
    - Scalable snapshots
- Major focus on stability testing of btrfs
    - Looking to see what use cases it fits best
- Harden XFS metadata
    - Detect errors to confidently support 500TB single FS

Upstream Projects

Performance Work: SBC-4 Commands

- SBC-4 is a reduced & optimized SCSI disk command set proposed in T10
  - Current command set is too large and supports too many odd devices (tapes, USB, etc)
  - SBC-4 will support just disks and be optimized for low latency
  - Probably needs a new SCSI disk drive
- New atomic write command from T10
  - All of the change or nothing happens
  - Supported by some new devices like FusionIO already
- http://lwn.net/Articles/548116/

Copy Offload System Calls

- Upstream kernel community has debated “copy offload” for several years
  - Popular use case is VM guest image copy
- Proposal is to have one new system call
  - int copy_range(int fd_in, loff_t, upos_in, int fd_out, loff_t upos_out, int count)
  - Offload copy to SCSI devices, NFS or copy enabled file systems (like reflink in OCFS2 or btrfs)
- Patches for copy_range() posted by Zach Brown
  - https://lkml.org/lkml/2013/5/14/622
- http://lwn.net/Articles/548347

IO Hints

- Storage device manufacturers want help from applications and the kernel
  - Tag data with hints about streaming vs random, boot versus run time, critical data
  - T10 standards body proposed SCSI versions which was voted down
- Suggestion raised to allow hints to be passed down via struct bio from file system to block layer
  - Support for expanding fadvise() hints for applications
  - No consensus on what hints to issue from the file or storage stack internally
- http://lwn.net/Articles/548296/

Improving Error Return Codes?

- The interface from the IO subsystem up to the file system is pretty basic
  - Low level device errors almost always propagate as EIO
  - Causes file system to go offline or read only
  - Makes it hard to do intelligent error handling at FS level
- Suggestion was to re-use select POSIX error codes to differentiate from temporary to permanent errors
  - File system might retry on temporary errors
  - Will know to give up immediately on others
  - Challenge is that IO layer itself cannot always tell!
- http://lwn.net/Articles/548353

Learning More

Learn more about File Systems & Storage

- Attend related Summit Sessions
  - Linux File Systems: Enabling Cutting-edge Features in Red Hat Enterprise Linux 6 & 7 (Wed 4:50)
  - Kernel Storage & File System Demo Pod (Wed 5:30)
  - Evolving & Improving RHEL NFS (Thurs 2:30)
  - Parallel NFS: Storage Leaders & NFS Architects Panel (Thurs 3:40)
- Engage the community
  - http://lwn.net
  - Mailing lists: linux-ext4, linux-btrfs, linux-nfs, xfs@oss.sgi.com