NetApp®

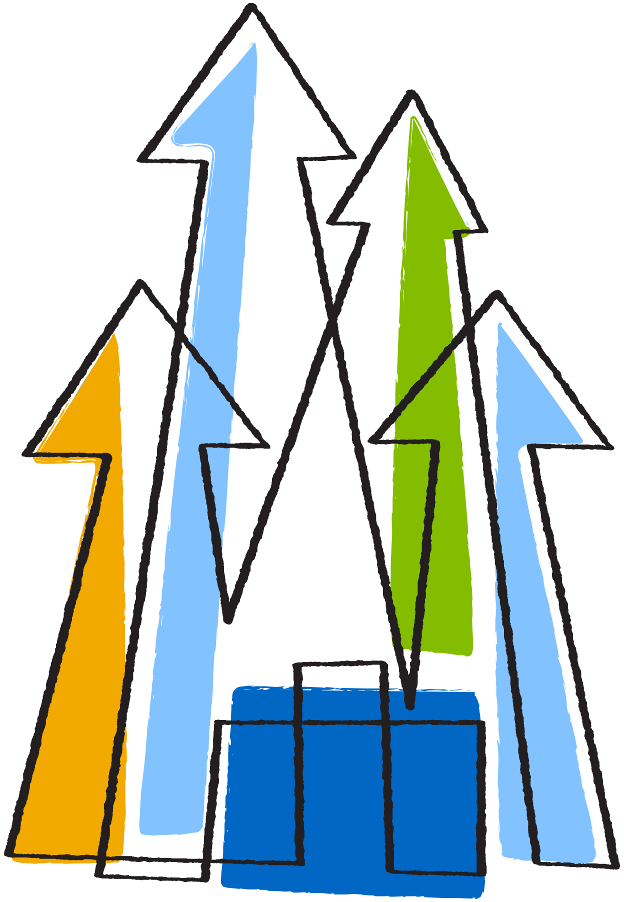
Go further, faster®

Building Large Scale High Performance Backup to Disk Repositories

Peter Buschman

EMEA PS Consultant

NetApp



Agenda

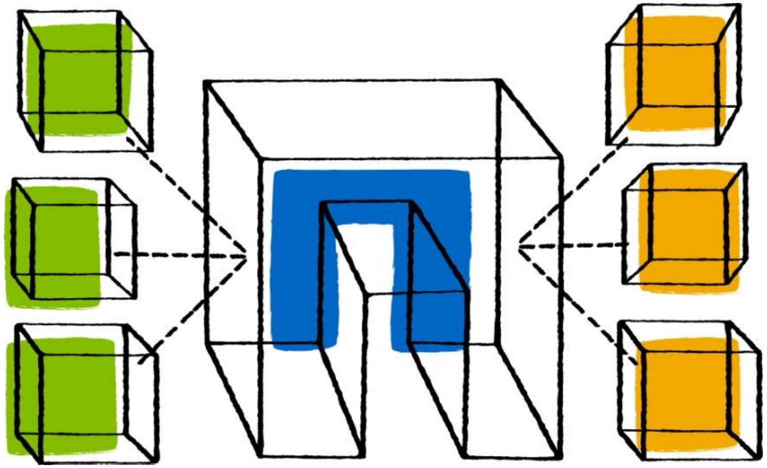
- Introduction
- Data Backup Workload Characteristics
- Designing a Disk Backup Storage System
- Storage Layout Philosophies
- NetApp E-Series Storage Solutions
- Case Study
- Conclusion
- Q&A

Introduction

This presentation focuses on how to design an effective backup-to-disk storage system that sustains performance even under large scale data growth and performance requirements.

Although not specific to any particular backup software, the recommendations made herein are applicable to any Linux/Unix based backup platform that is capable of using a POSIX-compliant filesystem as storage.

Data Backup Workload Characteristics



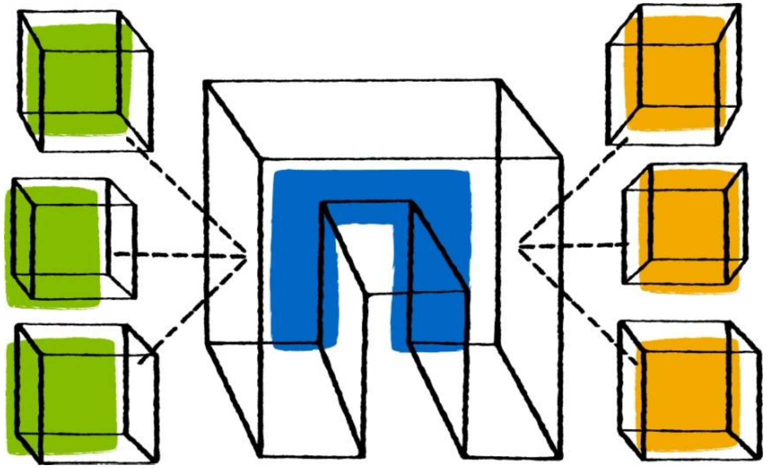
The Backup and Recovery Workload

Backup is different from workloads such as file serving, email, and server virtualization...

- Sequential rather than random
  - Backup is sequential write
  - Restore is sequential read
- High throughput
  - Bulk data movement within a specific time window
  - Sustained throughput vs. burst IO

From a storage perspective, this is not unlike workloads such as video capture or media processing.

Designing a Disk Backup Storage System



Hardware Selection

- Which one of these machines is tuned for capacity? What about performance?

Backup Design Constraints 1/2

Backup architects must operate under constraints many other system designers do not have...

- 1 - Minimize cost of storage (€/TB)
- 2 - Maximize storage utilization (0% wasted space)
- 3 - Achieve performance goals (complete within window)
- 4 - Maximize reliability (little downtime, no data-loss)

Backup Design Constraints 2/2

Meeting these constraints can be difficult in practice...

- The largest drives are often the slowest (7200rpm)
- Filesystem performance drops near 100% full
- Budget limits often mean that the bare minimum of storage is purchased to meet present requirements
- Greater than expected growth can easily overwhelm a minimally sized storage system

Rules for the Modern Backup Architect

Although it might seem like the backup architect has some impossibly conflicting goals to achieve, she is actually in an enviable position. Although the backup workload is intensive and tends to push the limits of storage hardware, it is also extremely predictable and its unique requirements can be accounted for in a sensible and cost-effective storage architecture.

Rule 1 - Do not mix workloads

The backup workload transfers a lot of data sequentially. On a shared storage system, this can reduce performance for other workloads unless done very carefully. Best practice is to dedicate an entire storage system to backup. Performance is guaranteed and backups do not interfere with other applications.

Rule 2 - Do not use NFS/CIFS storage

Storage connections using SAS or FibreChannel are preferable to those using NFS, CIFS, or iSCSI.

- Less protocol overhead
- Higher throughput per-port
- Little tuning required
- NFS / CIFS not designed for large-block IO

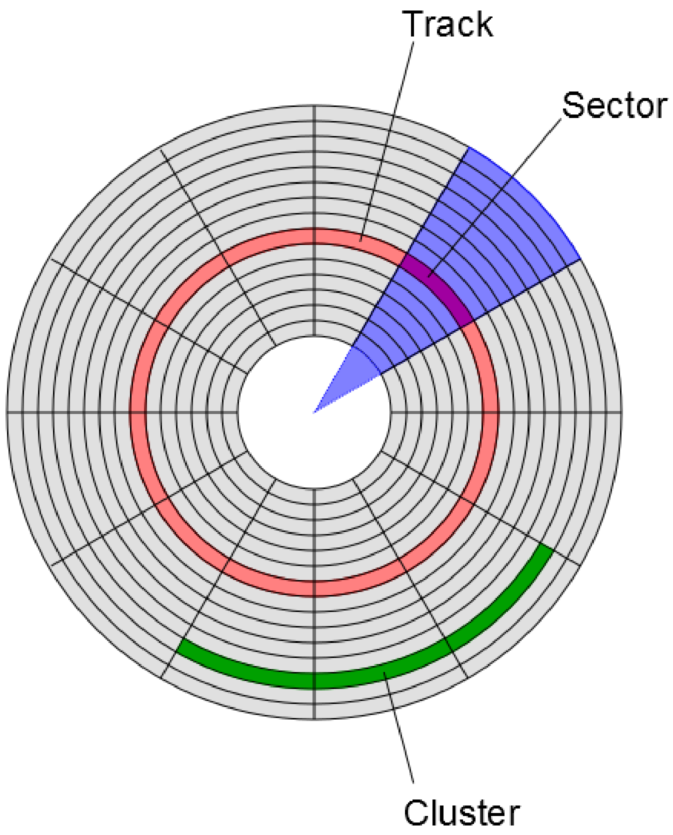
This rule applies to backup storage that is directly attached to the backup server. It can be violated in certain circumstances if the infrastructure available meets certain performance criteria.

Rule 3 - Use external hardware RAID

Modern storage arrays offer a number of advantages over server chassis with lots of internal drive bays.

- The server's CPU does not have to calculate parity
- Easier to cluster for high availability
- Can be upgraded separately from the backup server
- Aggregation of many disk drives together in a single RAID stripe means fewer devices to manage on the server *(also true for internal RAID controllers)* .

Selecting an IO size



- Disk drive IOPs determined by rotational latency and seek-time
- 7200rpm drives can perform fewer IOPs than 10,000rpm or 15,000rpm drives.
- Maximum throughput (MB/s) is best achieved by performing large IOs rather than small ones.
- Optimal IO size depends on what the OS, filesystem, and backup application support.
- For backup, IO sizes from 64k - 1MB are practical.
- Align the IO size to your filesystem's cluster size to prevent fragmentation

Selecting a File System

The next task is to choose a filesystem. Here, it is essential to ensure that the filesystem can also handle IOs of the same size. If it cannot, it may end-up turning the applications's large IOs into many small ones, resulting in sub-optimal performance.

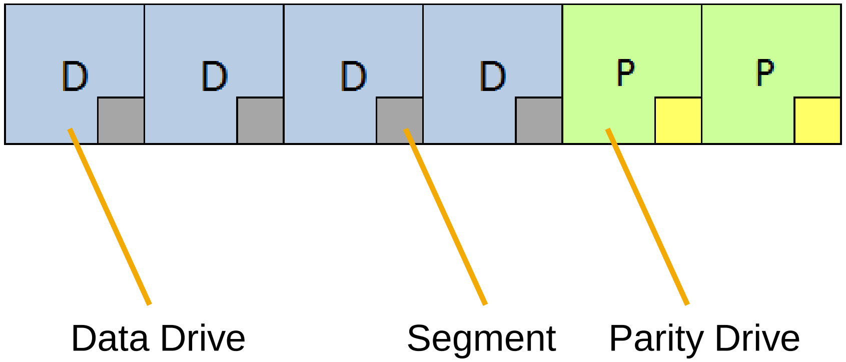
The simplest criteria is to look at the filesystems's supported cluster-sizes. Another is whether the filesystem flushes multiple blocks to disk in a single transaction when under sustained sequential load.

File Systems that Support Large IO Sizes

Filesystem	Max IO or Cluster Size	Notes
ZFS (open-zfs.org)	1MB or 128k	1MB only on Oracle Solaris
Lustre (lustre.org)	1MB	Linux only; Always does 1MB IO but OST filesystem determines allocation size; Requires storage nodes;
EXT4	1MB	Only in Linux kernel 3.2 and higher
XFS	64k	Linux requires pagesize increase
BTRFS	????	Not sure; Anybody know?

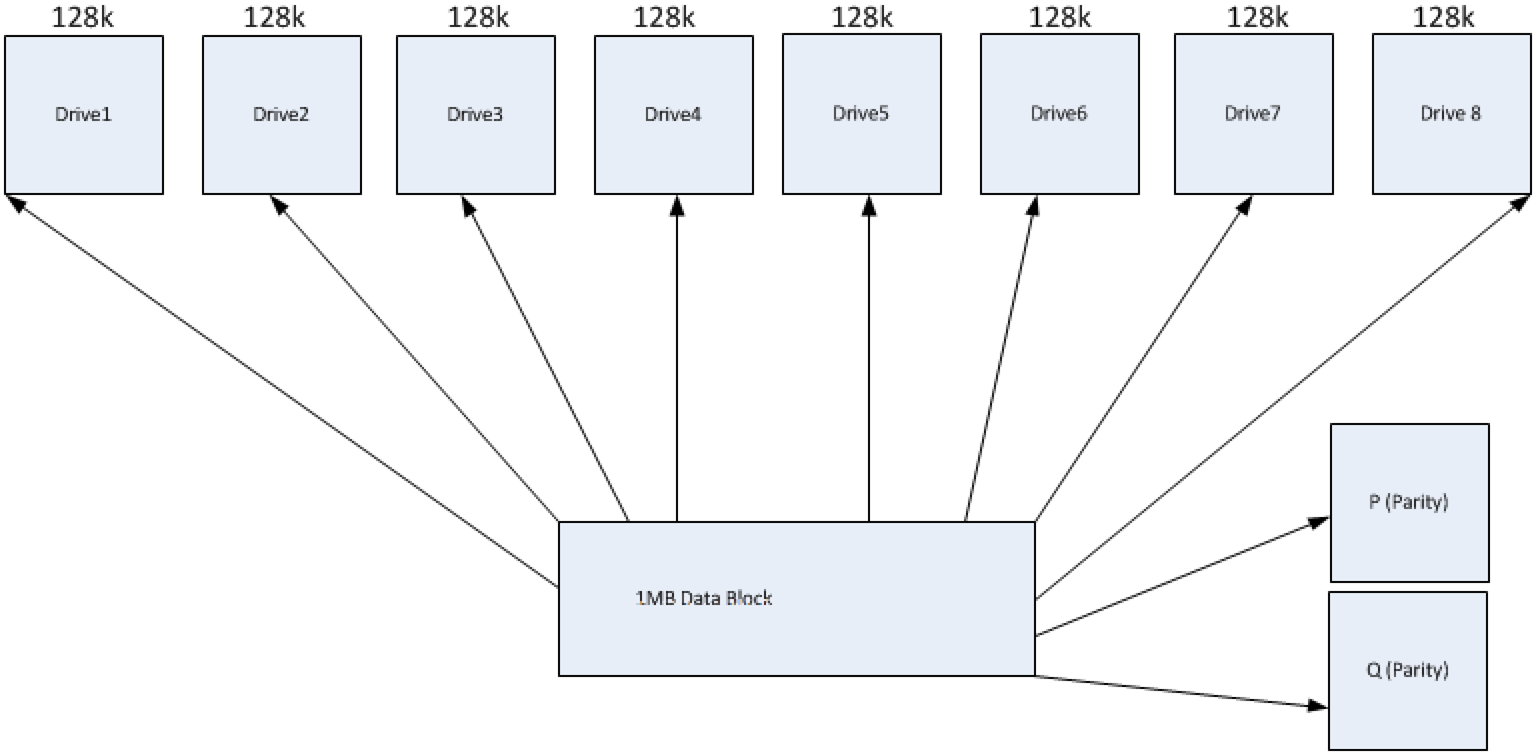
Configuring RAID for Optimal IO

RAID6 Stripe Write



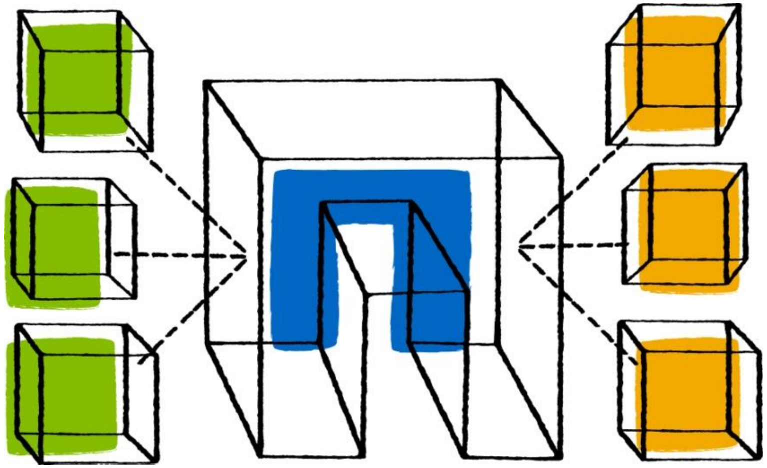
- RAID6 is essential for large repositories
- Parity uses 2 drives on every write
- Remaining drives contain data
- Segment size is the amount of data written to a single drive before proceeding to the next drive in the stripe (8k, 16k, 32k, 64k, 128k, 256k, 512k).
- Optimal IO is when all data drives in a stripe are hit in a single write operation. This is known as a "full stripe write".
- Divide the application IO size by the number of data drives to determine the optimal segment size.
- For common block sizes, aligning the IO size with the segment size to consistently produce full stripe writes is possible when the number of data drives is a power of 2 (2, 4, 8, 16)
- RAID6 8+2 is a best practice middle ground with good capacity and high performance

RAID6 8+2 Full Stripe Writes with 1MB IO



8x128kB=1MB. 1MB is fully striped across all drives in the RAID group. This allows for “well-formed” I/O.

Storage Layout Philosophies



Storage Layout Philosophies

Once the IO size, filesystem, and RAID config have been determined, the final decision remaining is the storage layout. Here, there are 2 possibilities.

- Single Filesystem
- Many Filesystems

Which layout is "better" will be strongly influenced by the choice of operating system, filesystem, and the number of backup servers accessing the storage.

Single Filesystem

Advantages

- Looks really cool
- Most efficient space usage
- Best utilization of disk spindles
- Capacity planning / trending is easy

Disadvantages

- Greatest data loss in event of LUN failure
- Most filesystems lack the ability to relieve a storage LUN from write IO while it is undergoing RAID reconstruct
  - Lustre can do this though
- Filesystem can only be mounted on one backup server at a time
  - Lustre is the exception

Many Filesystems

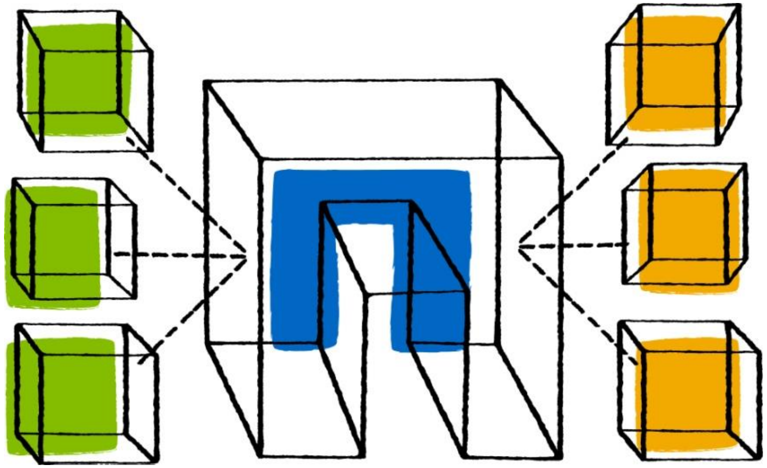
Advantages

- Data loss in case of LUN failure limited to the contents of a single filesystem
- Easier to add new storage / remove old storage without downtime / disruption

Disadvantages

- Space management much more difficult
- Less efficient use of disk spindles
- Each simultaneous filesystem mount may incur overhead
  - ZFS and Luste reserve significant memory buffers for each mountpoint
  - Can be mitigated by unmounting filesystems not in-use

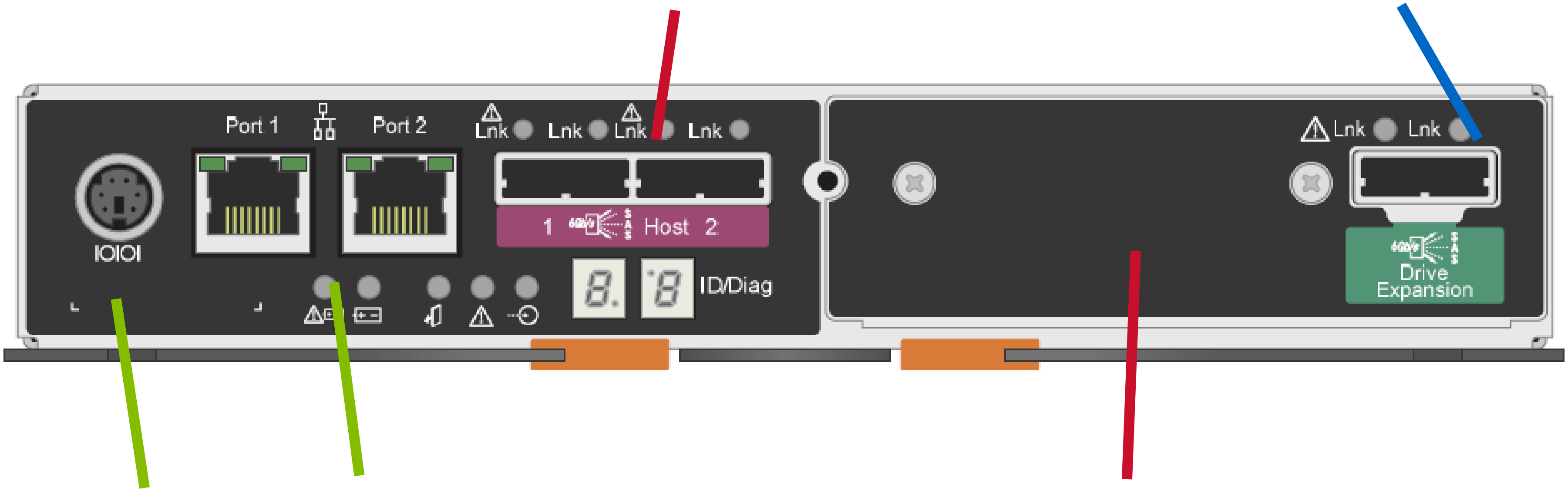
NetApp E-Series Storage Systems



NetApp E-Series Storage Systems

- NetApp Acquisition of Engenio in 2011
- Impressive bandwidth performance powers cost-effective solutions
- Leading density saves data center floor space and lowers operational costs
- Modular flexibility supports configurations optimized for solution requirements
- Bullet-proof reliability and availability designed to ensure continuous high-speed data delivery

E2600 Controller



(2) 6Gb SAS host interface ports

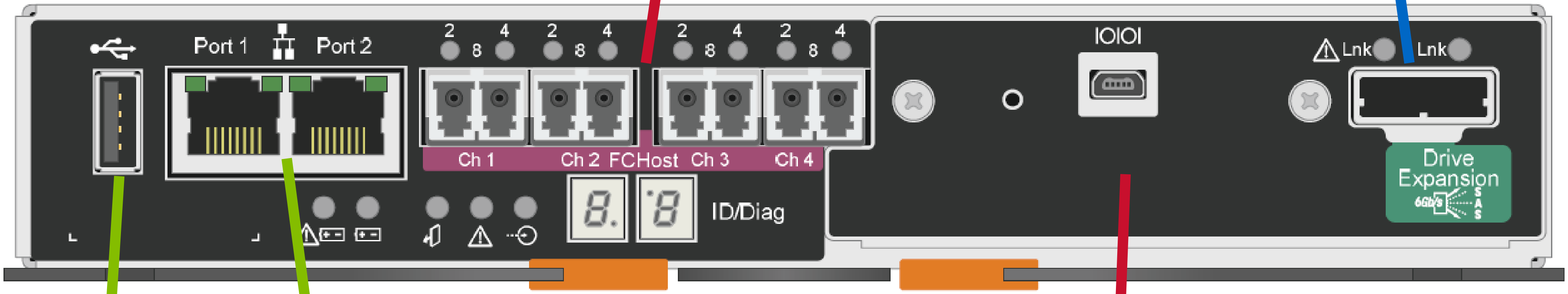
6Gb SAS drive expansion port

RS232

Dual Ethernet connections

Optional host interface card

E5400 Controller



(4) 8Gb FC host interface ports

RS232

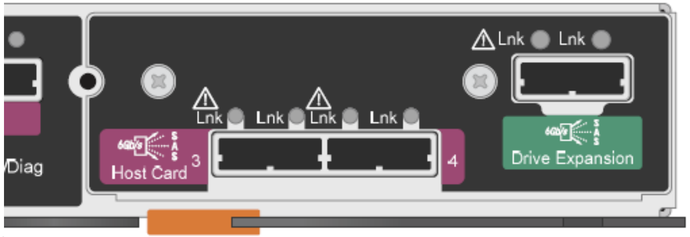
6Gb SAS drive expansion port

USB

Dual Ethernet connections

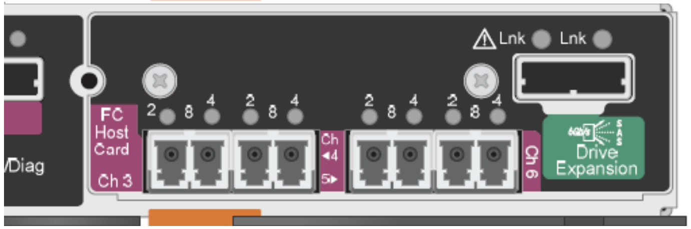
Optional host interface card

E-Series Host Card Options



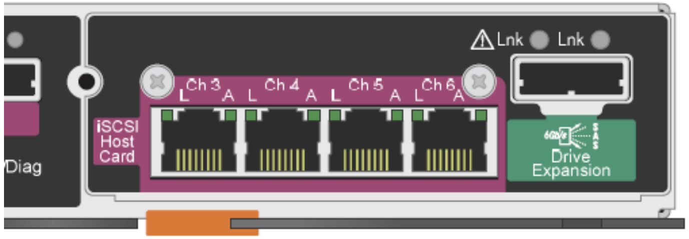
SAS Host card

- (2) 6Gb SAS ports



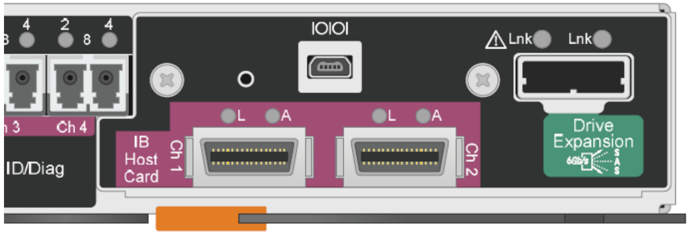
FC Host card

- (4) 8Gb FC ports



iSCSI Host Card

- (4) 1Gb iSCSI ports (shown)
- (2) 10Gb iSCSI ports



InfiniBand Host Card (E5400 only)

- (2) 40Gb IB host interface ports

DE6600 Shelf – Overview

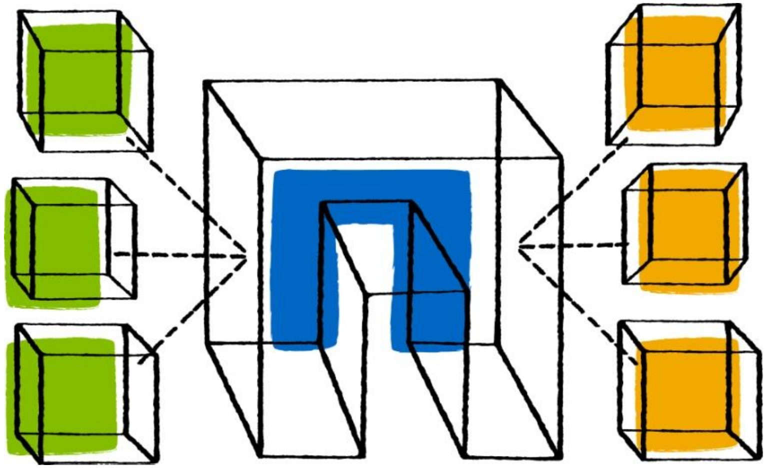
- High-density disk shelf supporting 60 SAS drives
  - 5 horizontal drawers with 12 drives per drawer
  - Just 4U in height and fits standard 19” rack
  - Up to 200 TB usable capacity with 4TB 7.2K SAS drives
- Superior RASUI
  - Drives remain online when drawer is extended for service
  - Individual drawer extension and front access enables safer drive replacement

- “80 plus” high efficiency power supplies
  - Up to 10% reduction in power/cooling
- Support E5400/E2600 controllers or ESMs

DE6600 Disk Shelf

- 4U, 60x 3.5” drive high density SAS enclosure
- 5 - 12 drive horizontal drawers = 60 drives
- Dampeners avoid I/O degradation from acceleration/deceleration of drives
- All drives are active with open drawer
- Designed to fit standard 1000mm cabinets (32” max depth)
- Solution Requires 6, 220V power drops
- Indicators for drive power/activity, fault, and service action allowed per tray

Case Study



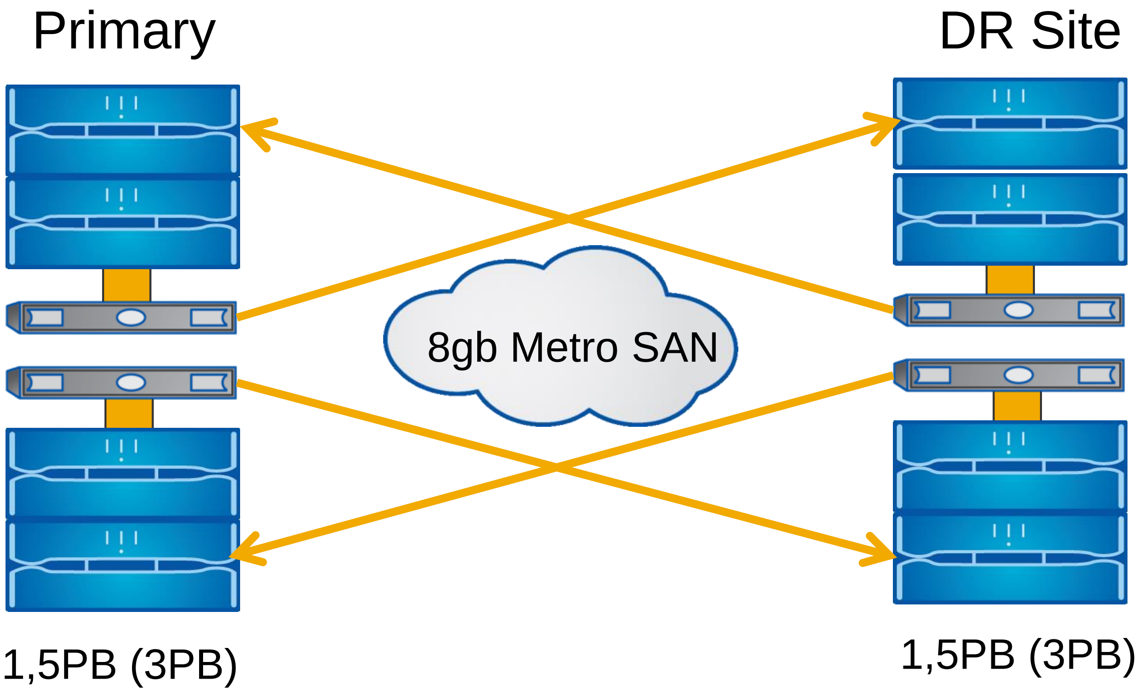
Case Study

Using the principles and hardware described, a disk backup system was implemented during the summer of 2013 in Frankfurt.

With a total of 1200 x 3TB NL-SAS disk drives, a net usable capacity of 3000TB (1TB = 1024GB) was achieved split across a primary and a disaster recovery datacentre. Utilizing ZFS compression, this is further increased to nearly 6PB. The entire solution can fit in just 2 x 42u cabinets.

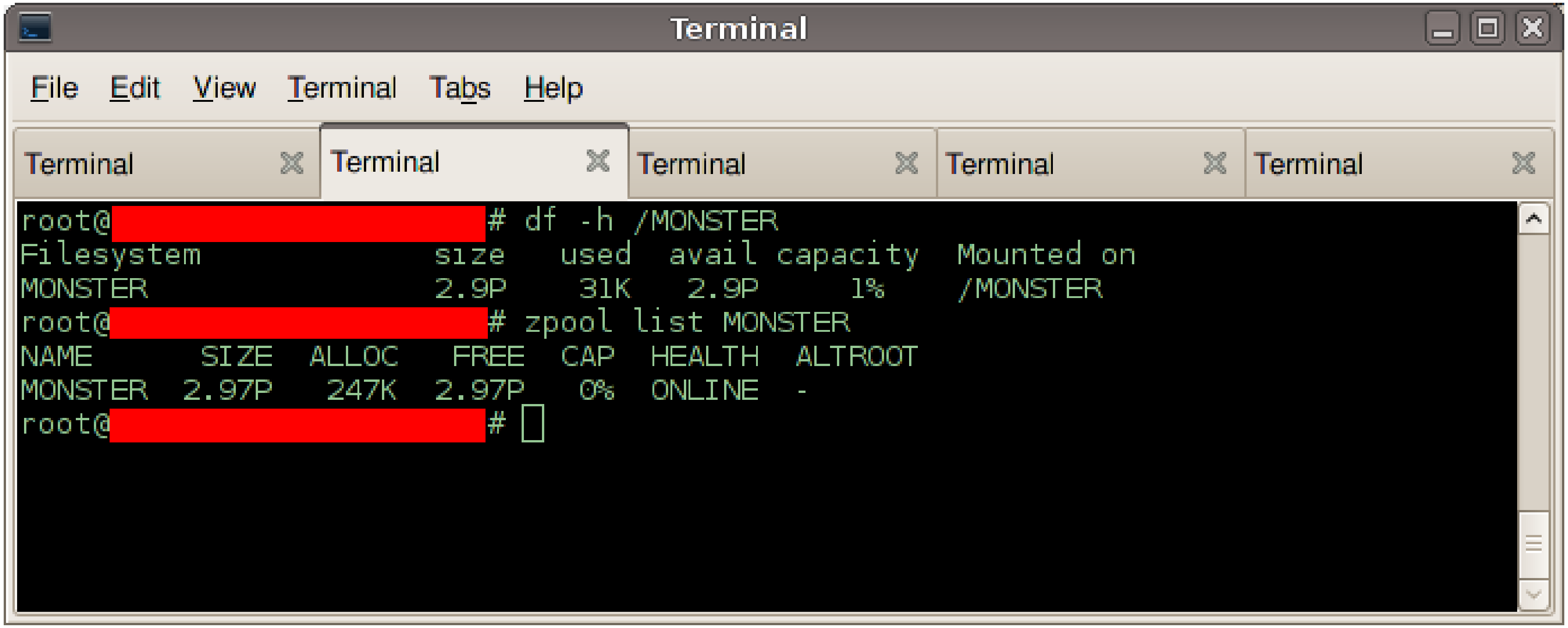
Case Study

6PB Cross Site Backup Repository



- Symantec NetBackup
- 4 x media servers
- ZFS on Solaris with 1MB record-size and compression
- 4 x 75TB storage LUNs with 512k segment size paired into 150TB zpools per-media server
- All backups written to both locations every week
- 5 week retention with 5PB of unexpired backups at any given time (and growing).

3PB ZFS Filesystem



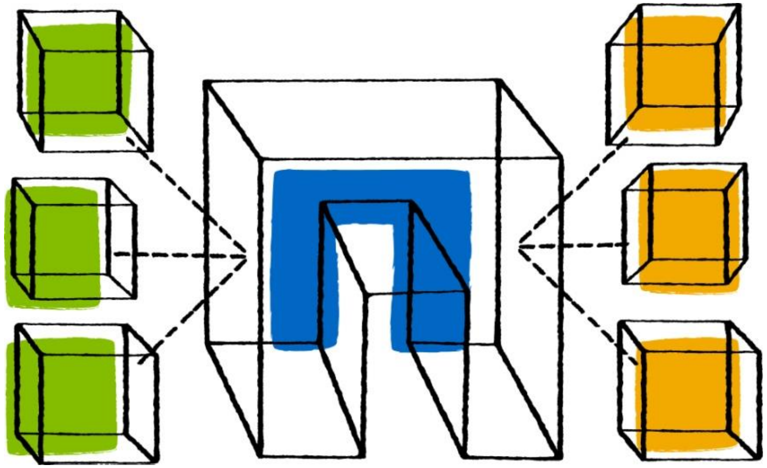
```
root@[redacted]# df -h /MONSTER
Filesystem                  size    used   avail capacity  Mounted on
MONSTER                     2.9P     31K    2.9P      1%    /MONSTER
root@[redacted]# zpool list MONSTER
NAME       SIZE   ALLOC    FREE   CAP  HEALTH  ALTROOT
MONSTER  2.97P    247K  2.97P    0%  ONLINE  -
root@[redacted]#
```

Conclusion

Large scale disk repositories are practical today using readily available and freely available operating systems

- Know your workload
- Optimize the entire IO path from the backup application all the way to the storage
- Architecting both for capacity and performance is possible.
- PetaBytes are the new TeraBytes

Questions & Answers



Thank you

peter.buschman@netapp.com
+49 151 527 555 24