Trends in the Enterprise Storage Market

Tom Coughlan
Sr. Engineering Manager, Red Hat
Feburary, 2012

Hot Topics

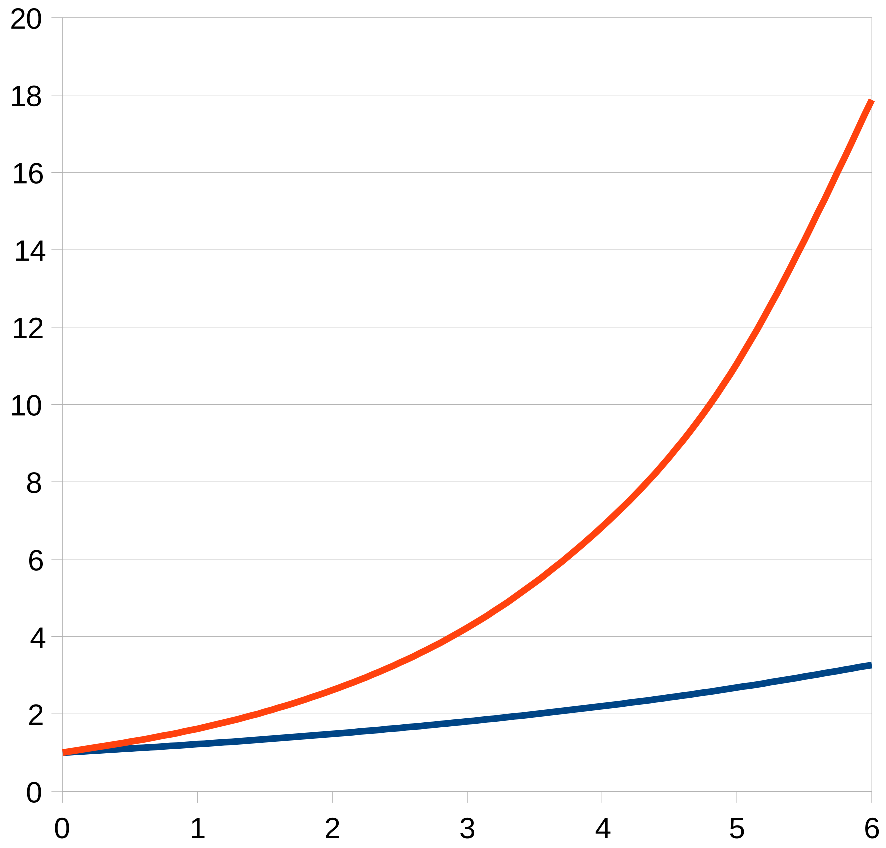
- Big Data,
- Unstructured Data,
- Scale-Out vs. Scale-Up,
- Virtualization,
- pNFS
- Solid State Storage...

What is the future of SAN, NAS, DAS?

What role will Linux play in the new environment?

The amount of data is exploding

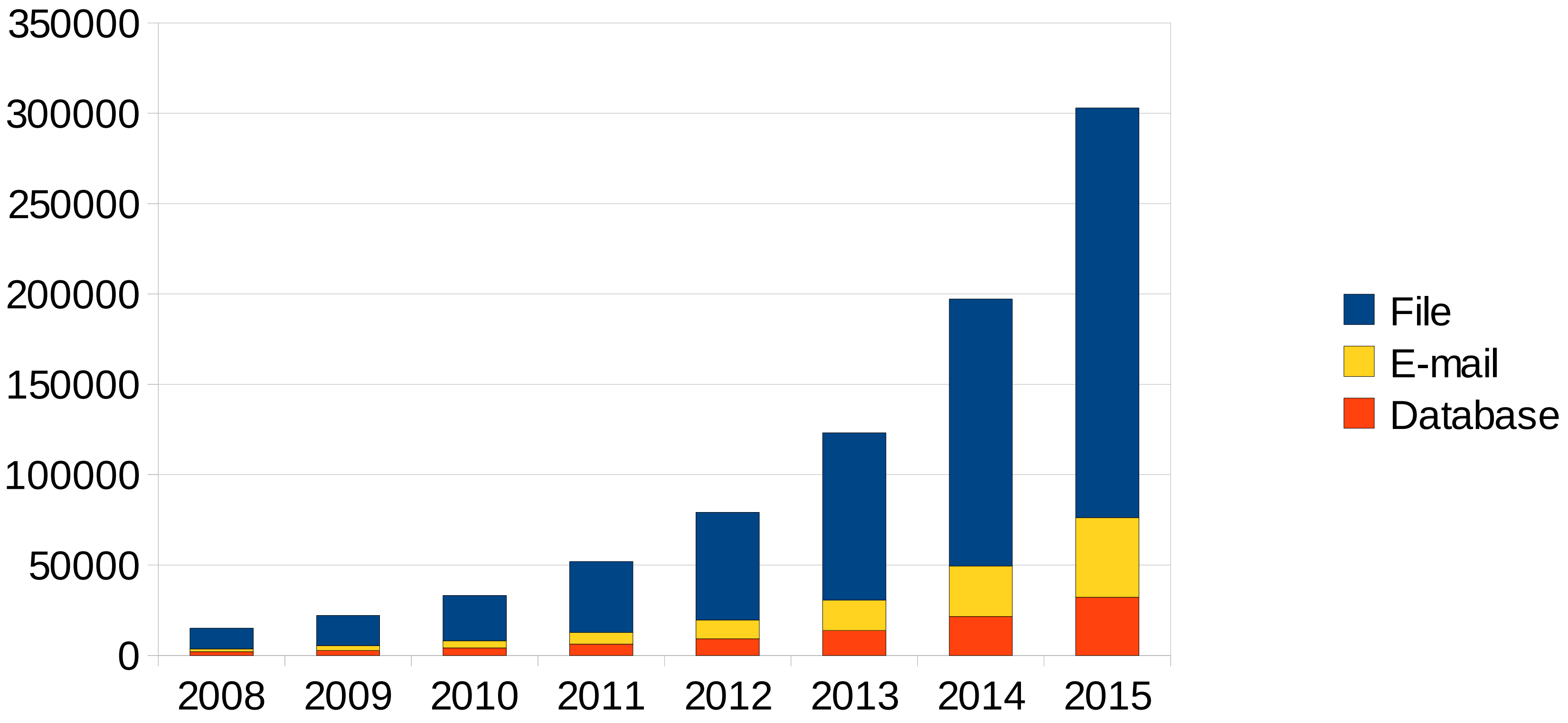
- IBM estimates:
  - Every day, 2.5 exabytes of data are created.
  - 90% of the data in the world today was created within the past two years.
- IDC projections:
  - transactional data will grow at a 21.8% CAGR
  - unstructured data will grow at a 61.7% CAGR



http://www-01.ibm.com/software/data/bigdata/
http://searchstorage.techtarget.com/magazineContent/Object-storage-gains-steam-as-unstructured-data-grows

Much of this data is unstructured

- Total Archived Capacity, by Content Type, Worldwide, 2008-2015 (Petabytes) (ESG)



h20195.www2.hp.com/v2/GetPDF.aspx/4AA0-4382ENW.pdf

Big Data Analytics

- Healthcare, government, entertainment, social networking, oil and gas, retail...
- Business intelligence, strategy, product support, product panning, development, just-in-time capacity planning...
- Requires:
  - High volume (so cost control is critical)
  - Low latency (streaming)
  - Data integration: text, audio, video..., from retail, medical, sensor, seismic, climate, satellite, (even databases)...

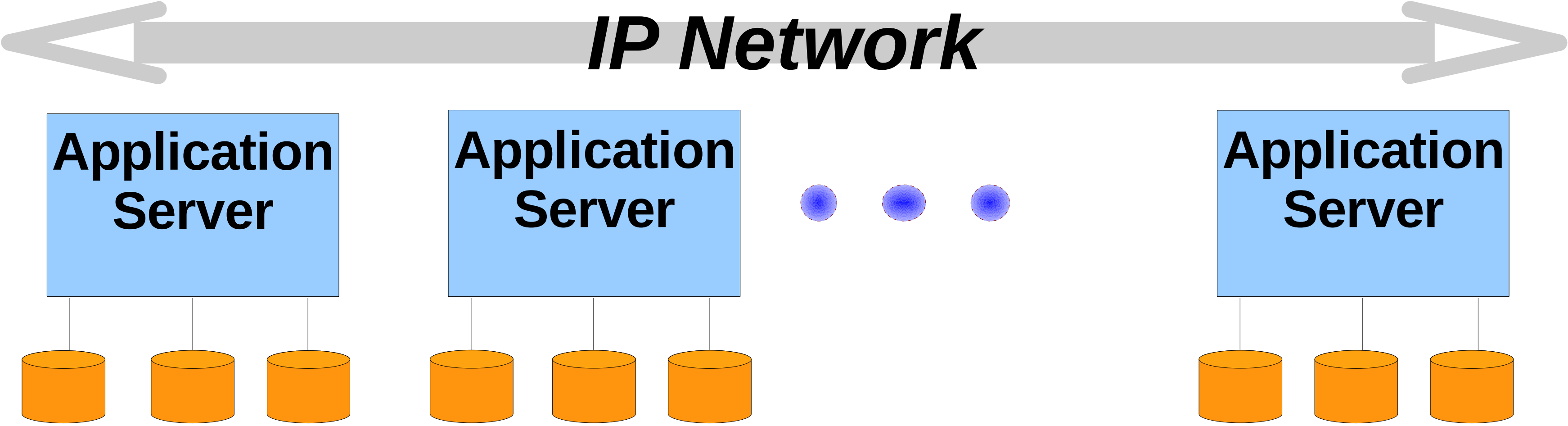
How to deal with this?

- So far, big data adherents have

1) Preferred to avoid shared storage, to minimize latency, and cost.

2) When more capacity is required, **scale-out** (add nodes)

   - Keep the storage close to the processor
   - Add processing power and storage together
   - Use commodity parts
   - File replication for data persistence, as needed



Scale-out, shared nothing

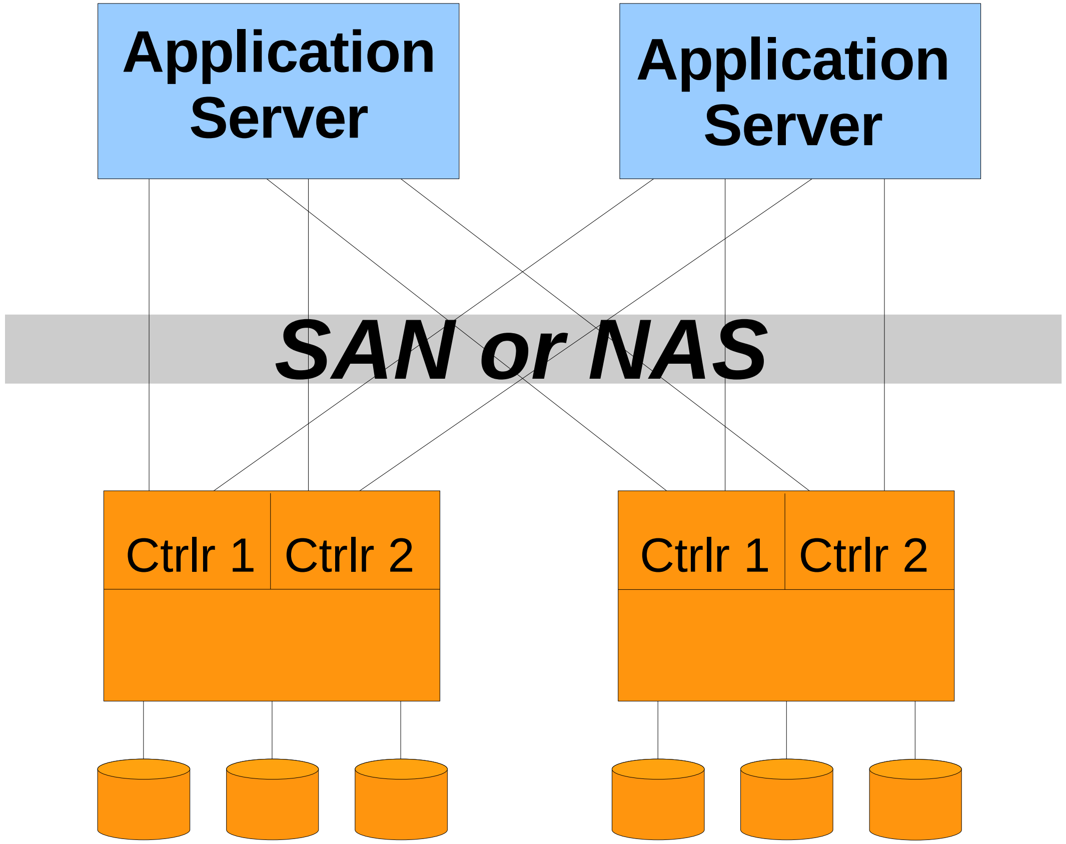
- Advantages:
  - Performs well for highly distributable problems
  - Inexpensive commodity hardware
  - Takes advantage of high performance local storage (PCIe flash)

Scale-out, shared nothing

- Disadvantages:
  - Latency increases for queries that span nodes, and for replication.
  - Specialized functions previously done in the storage controller must be implemented in the o.s.:
    - distributed fs, global namespace, data replication, backup, encryption, snapshot, thin provisioning, remote replication, deduplication, compression, proactive error detection, ease of management, ...

Contrast with the more traditional approach - shared storage, scale-up

- Smaller number of nodes, more tightly-coupled, shared resources, specialized storage servers.
- When more capacity is required, **scale-up** existing nodes.



Scale up – add horsepower to existing nodes

- Advantages:
  - Some applications (single-threaded with large data sets) can not be easily partitioned.
  - Centralized data protection, management, backup...
- Disadvantages:
  - Scaling limits... eventually you hit a wall
    - ...and, if you do add another server+storage cluster, the lack of a global namespace can make it difficult to manage/load-balance the environment
  - Proprietary, vendor lock-in
  - Generally more expensive

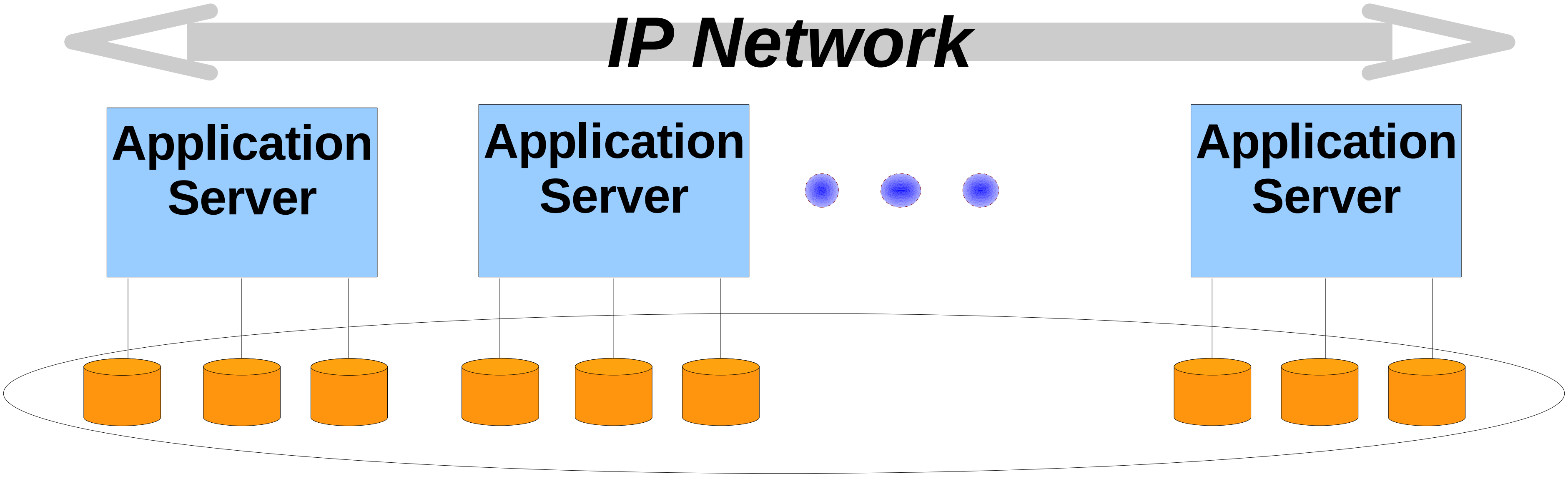
Shared storage

- Advantages:
  - Data is available to multiple machines
    - Server virtualization provides load balancing
  - Centralized data protection, management, backup...
- Disadvantages:
  - Access coordination can impact performance
  - Can be more expensive than DAS

As big data moves to the enterprise

- Take advantage of the scale-out approach
  - Control cost
- but keep shared storage
  - Virtualization
  - Ease-of-management, data protection, specialized functions.

Scale-out, with shared storage

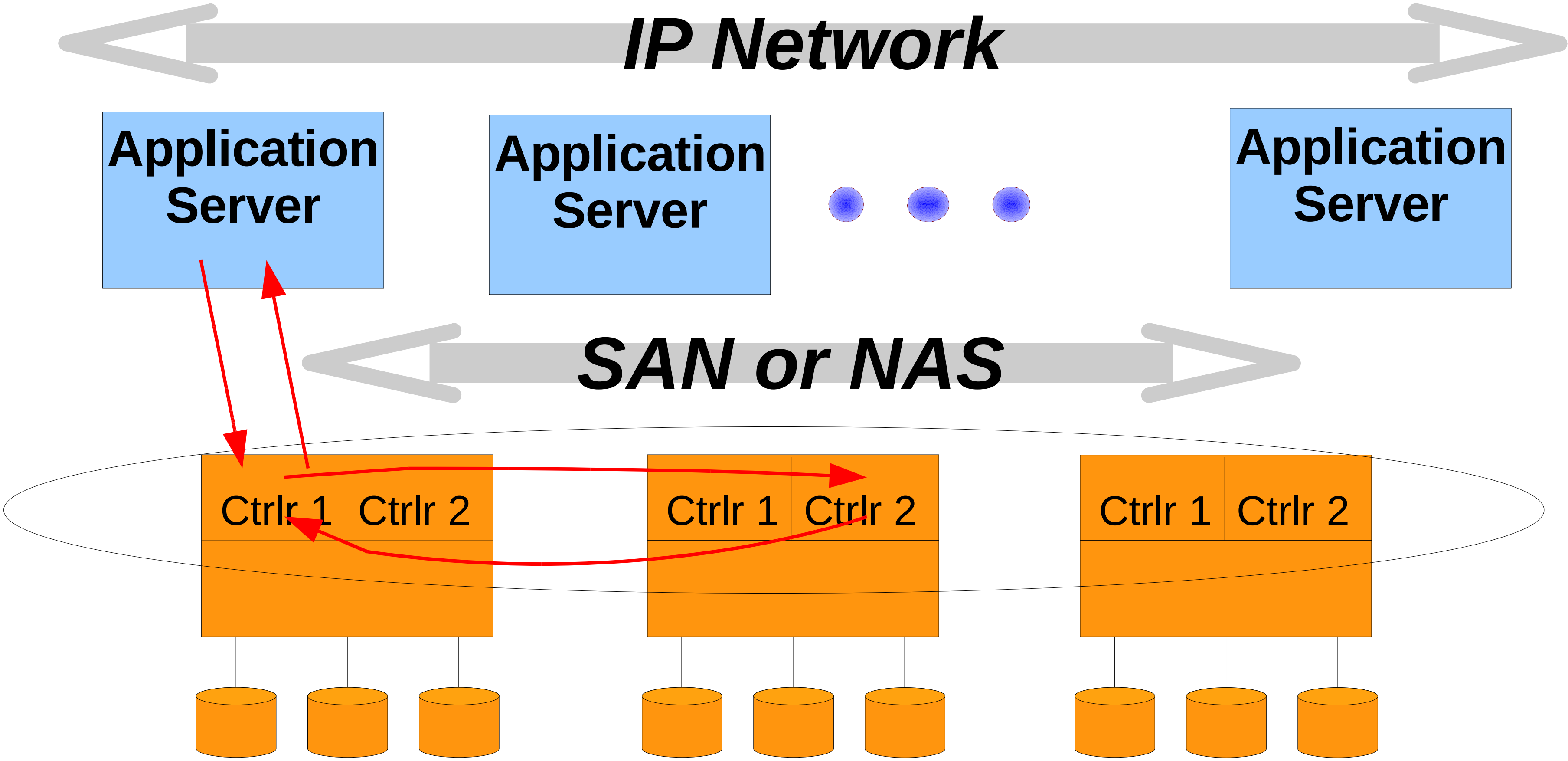


NAS/SAN Shared Storage

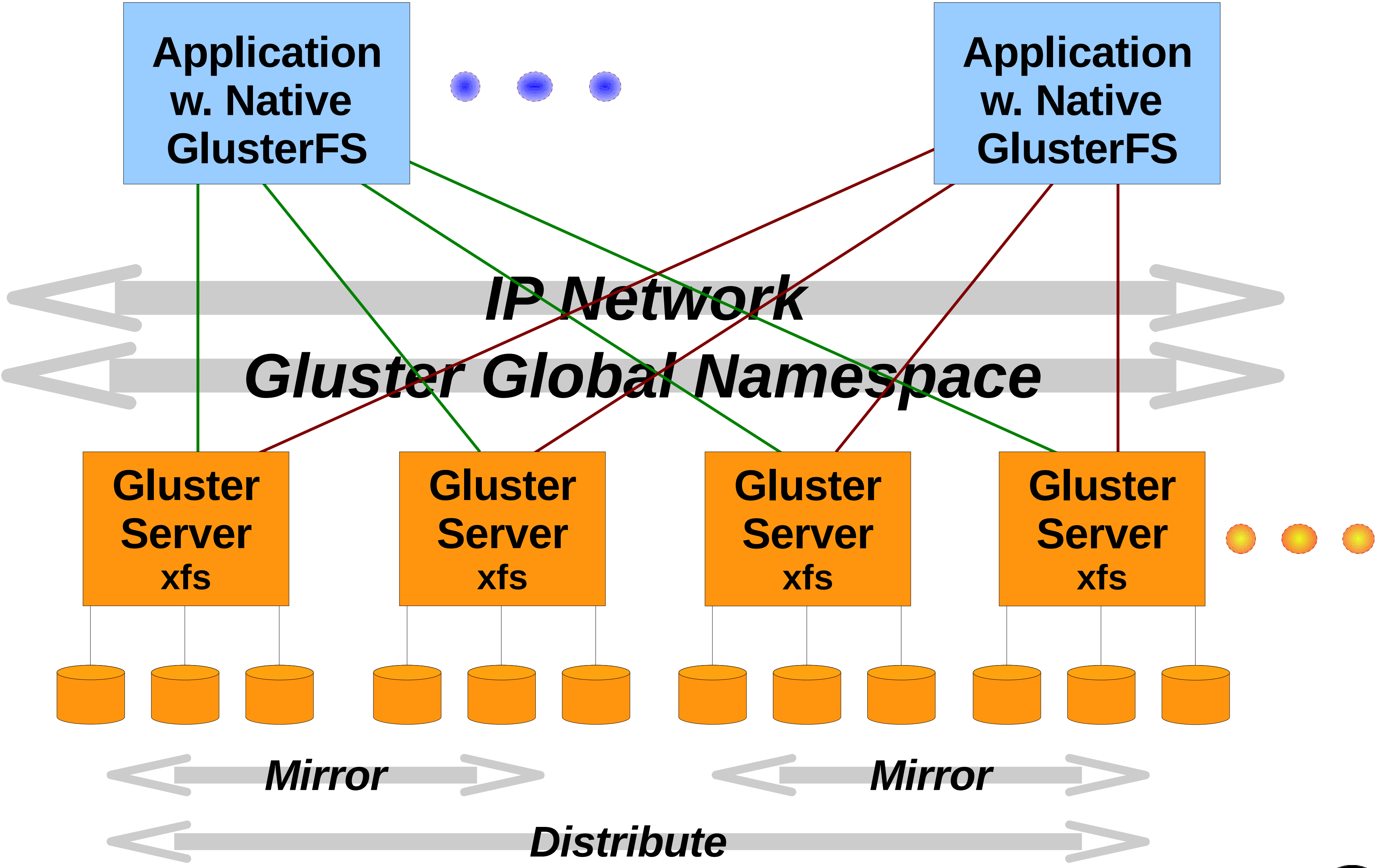
- Scale-out NAS
- pNFS
- iSCSI and FCoE

Scale-out NAS (the hardware approach)

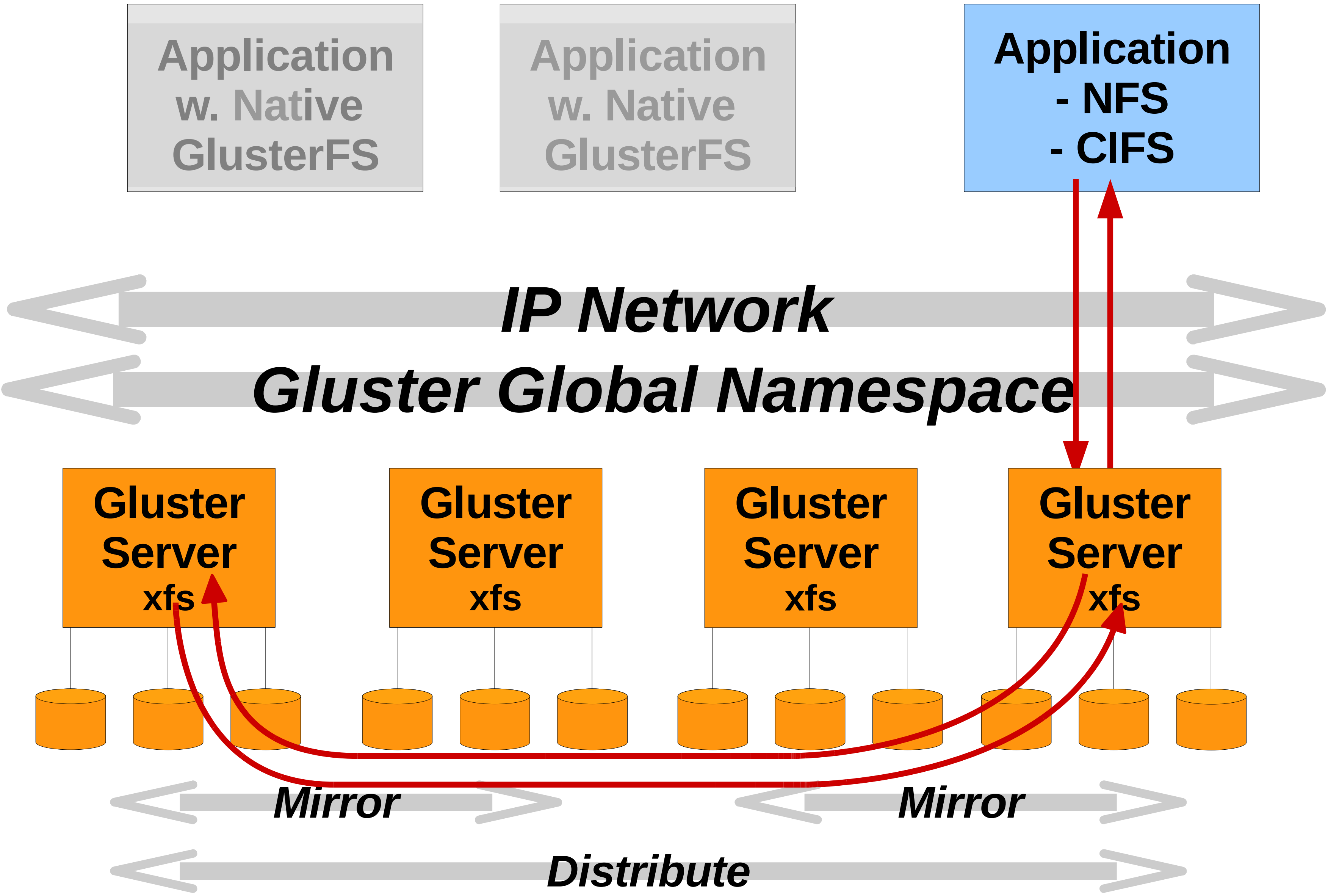
- Hardware vendors solve the storage controller bottleneck by “clustering” the controllers together.
    - The group appears as one to the o.s.



Scale-out NAS (the software approach)
- Gluster Distributed Filesystem

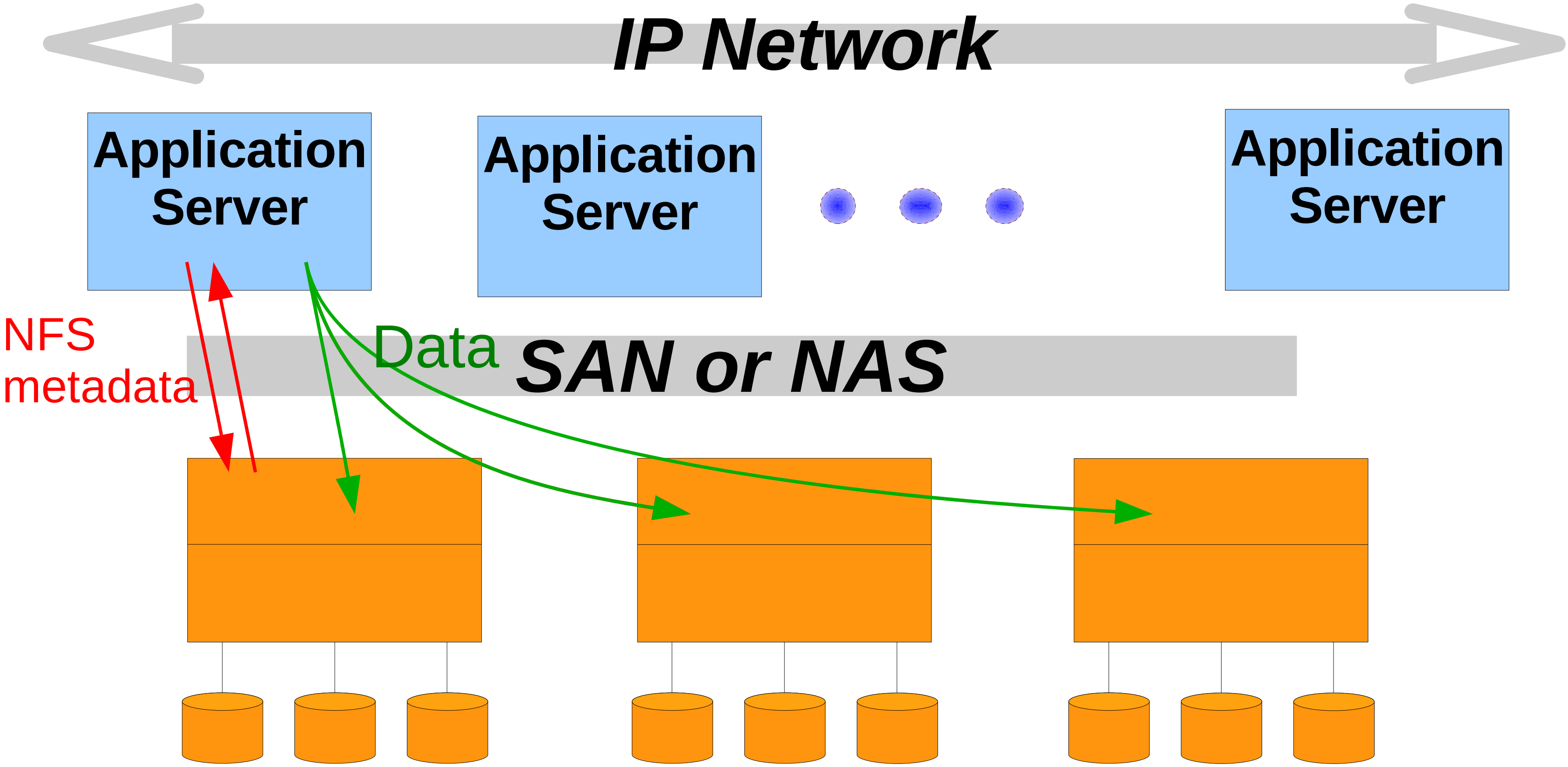


Gluster with Non-native Clients



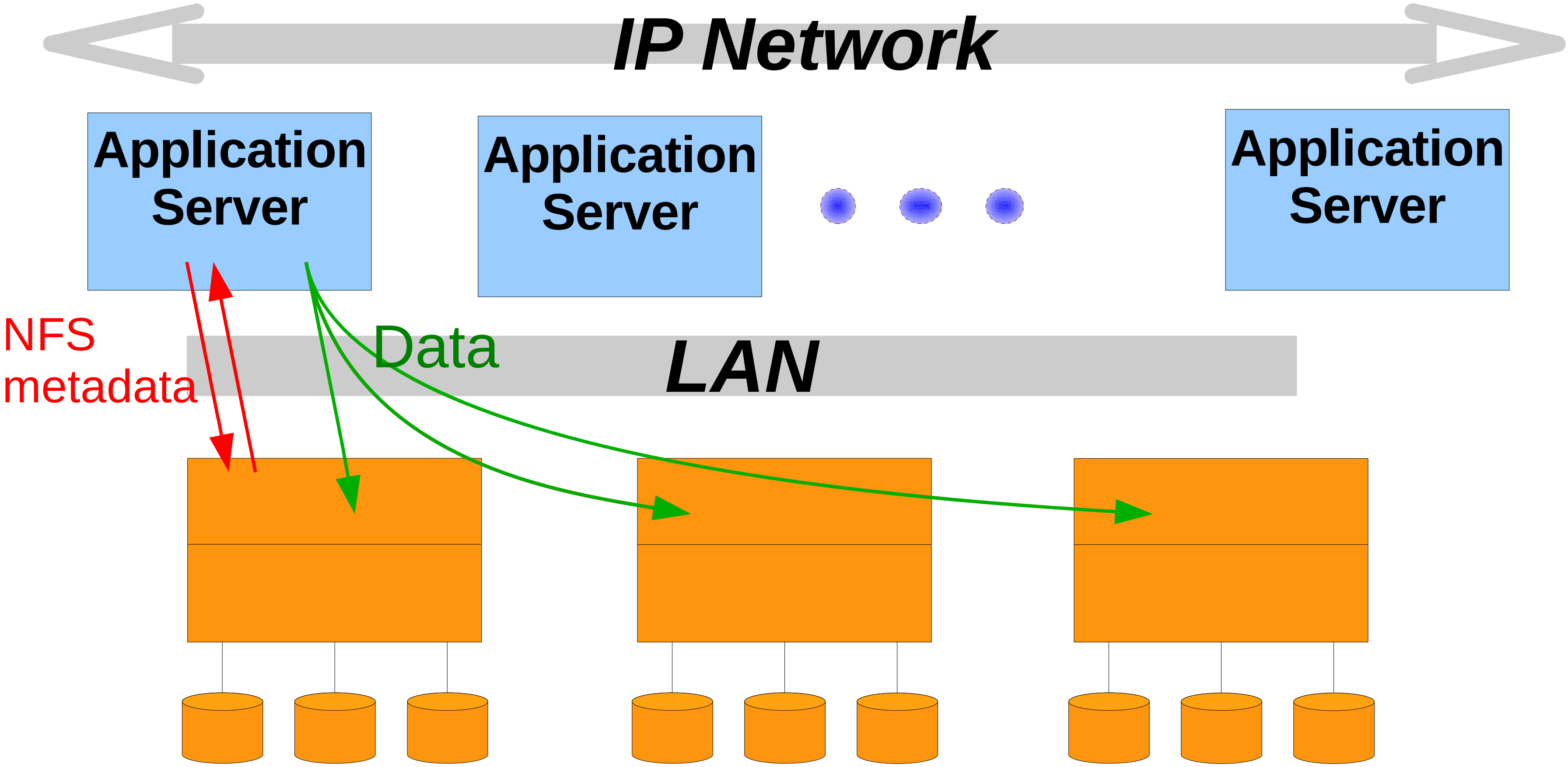
pNFS

- Access an NFS Metadata Server, then R/W the storage directly
  - Data may be File, or Object, or Block Based



iSCSI and FCoE

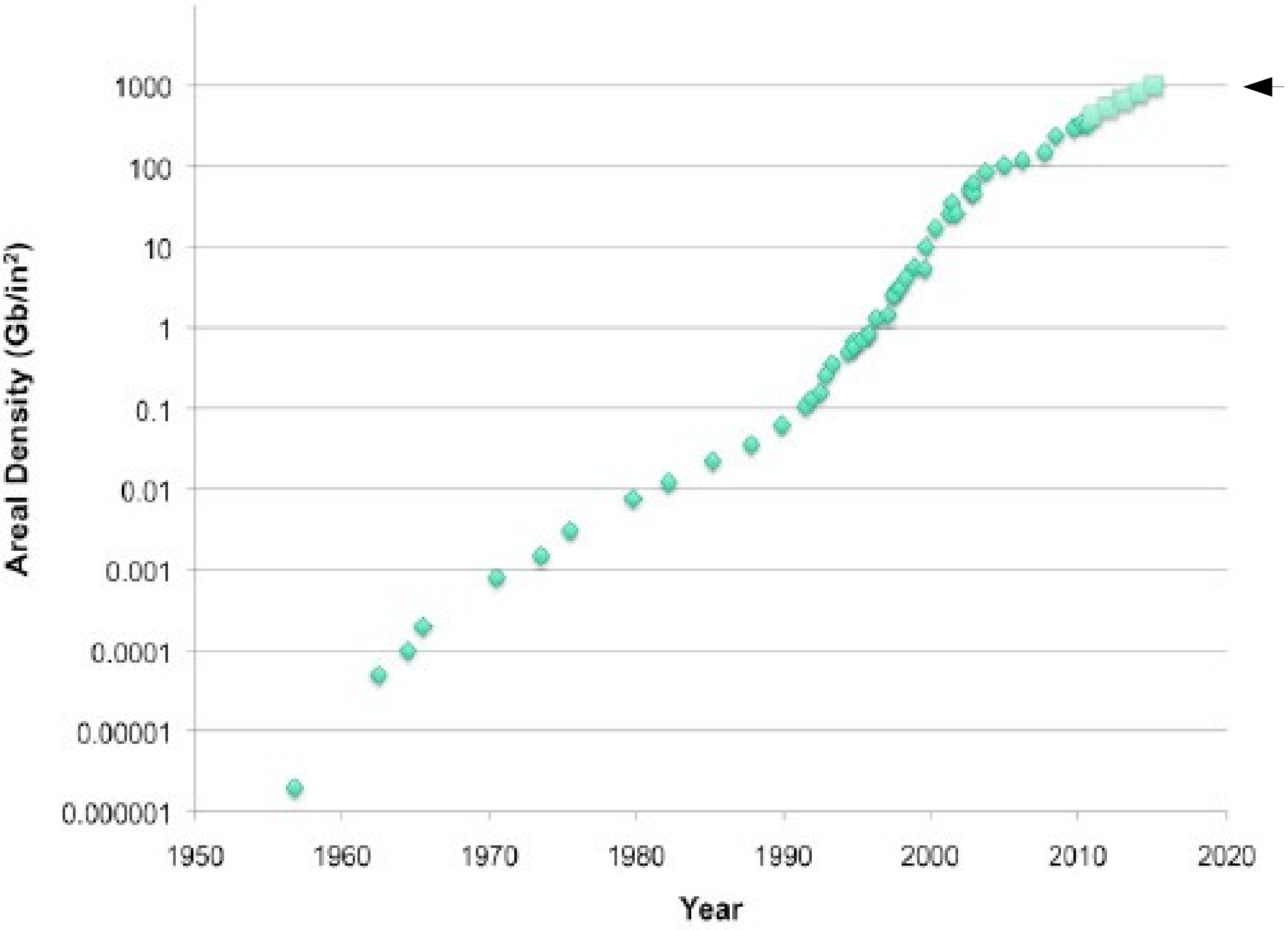
- Lower-cost shared block storage
  - Traditional db, and virtualization workloads
  - May pair nicely with pNFS:



Conclusions

- In the strict big data approach, with no shared storage, the Linux system must perform the specialized functions previously performed by the storage controller.
    - Currently underway:
        - efficient snapshot
        - thin provisioning
        - disk encryption - dm-crypt
        - integration of LVM with md RAID
        - PCIe flash performance optimizations
    - Future:
        - hierarchical storage

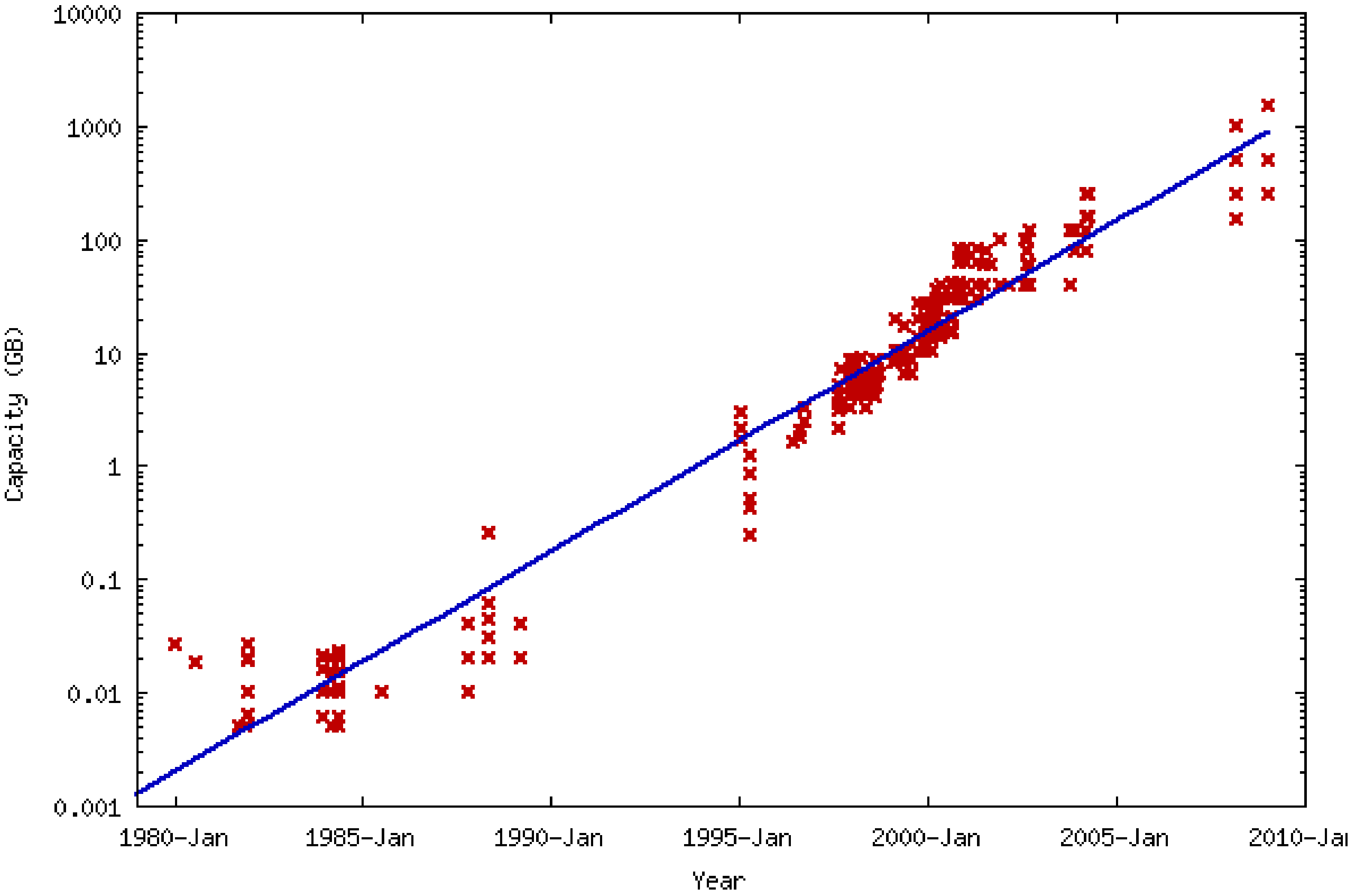
Disk Density



30% CAGR

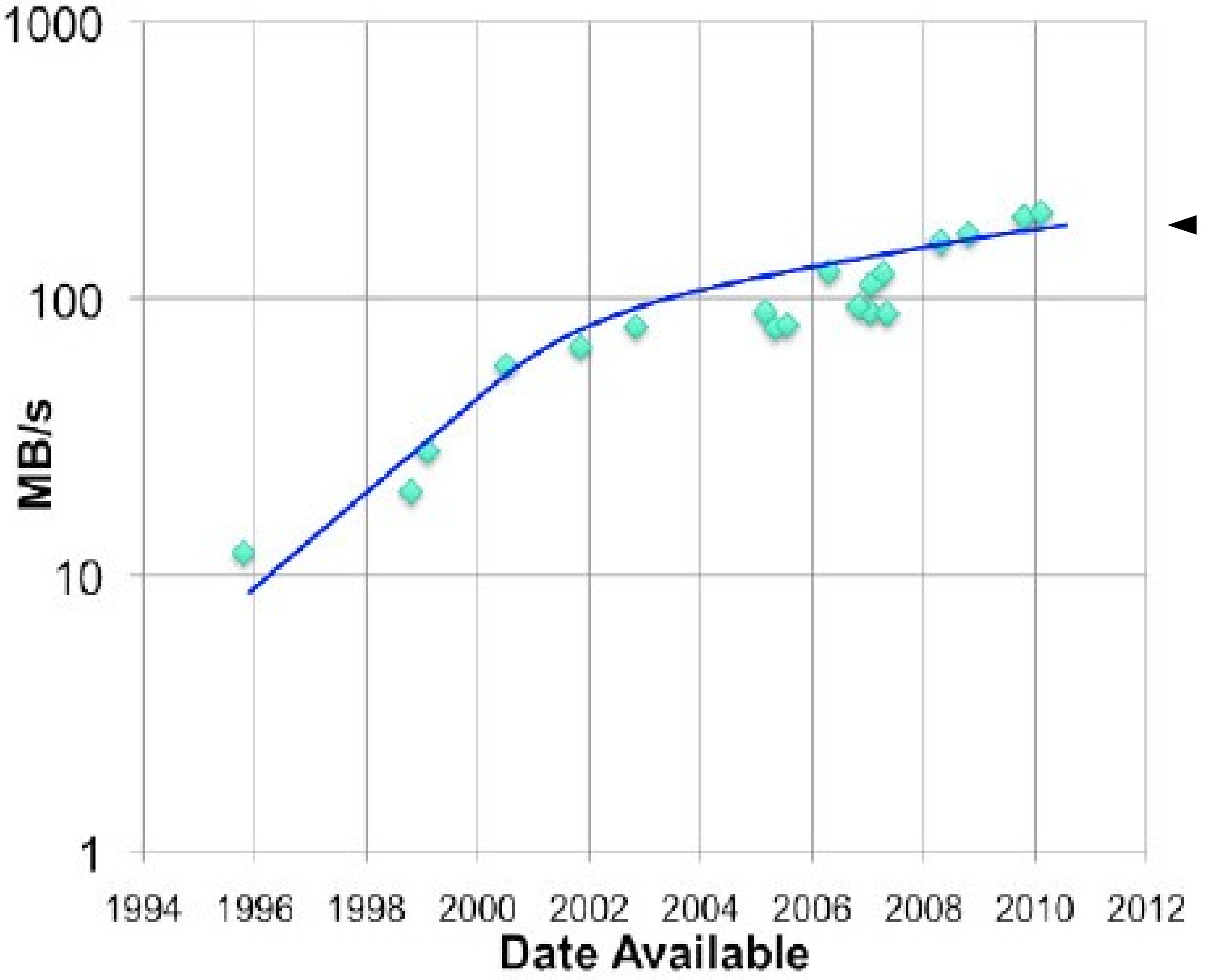
http://www.storagenewsletter.com/news/disk/hdd-technology-trends-ibm

Disk Capacity



http://en.wikipedia.org/wiki/File:Hard_drive_capacity_over_time.png

Disk Max. Sustained Bandwidth

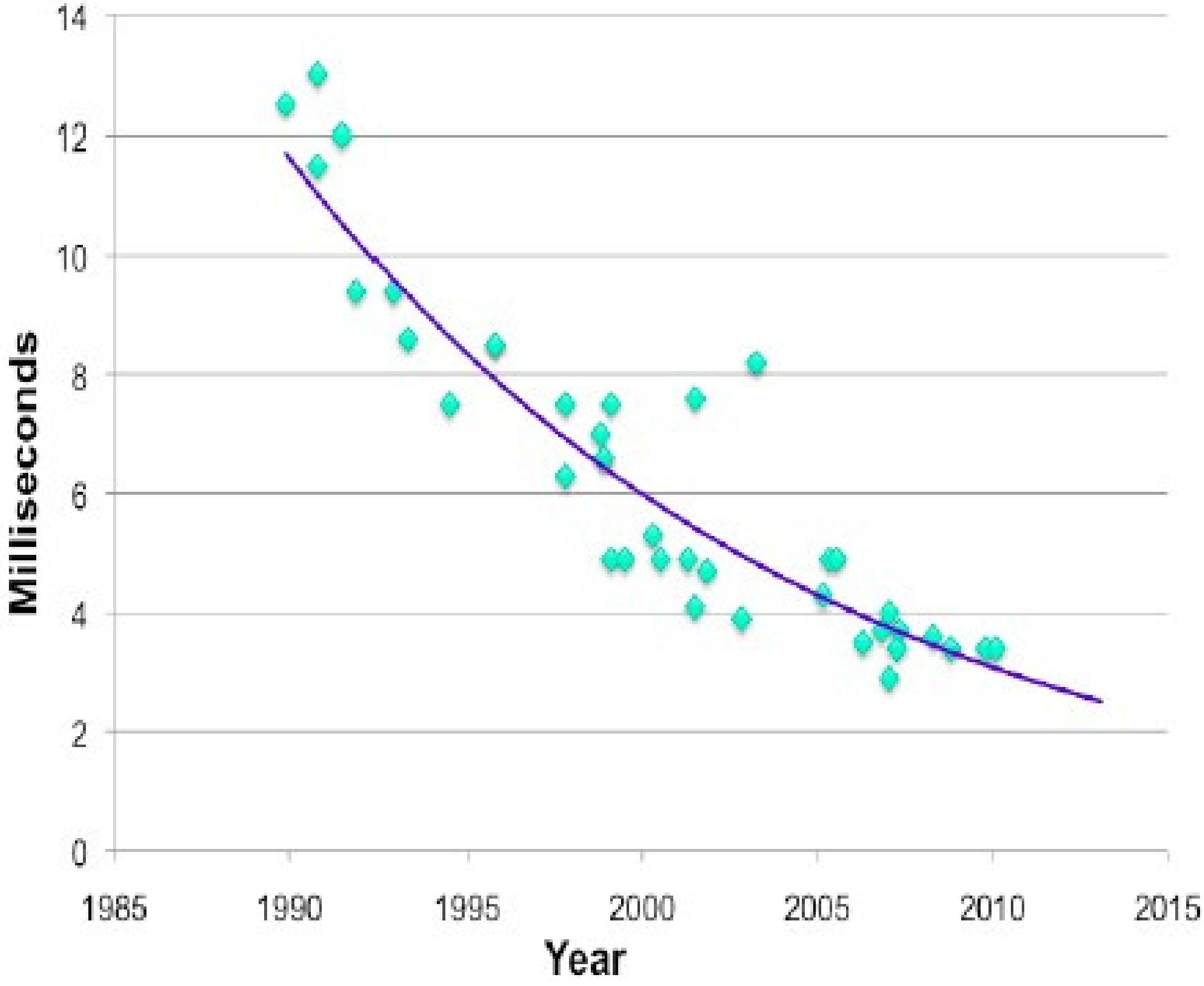


◄ 10-15% CAGR

http://www.storagenewsletter.com/news/disk/hdd-technology-trends-ibm

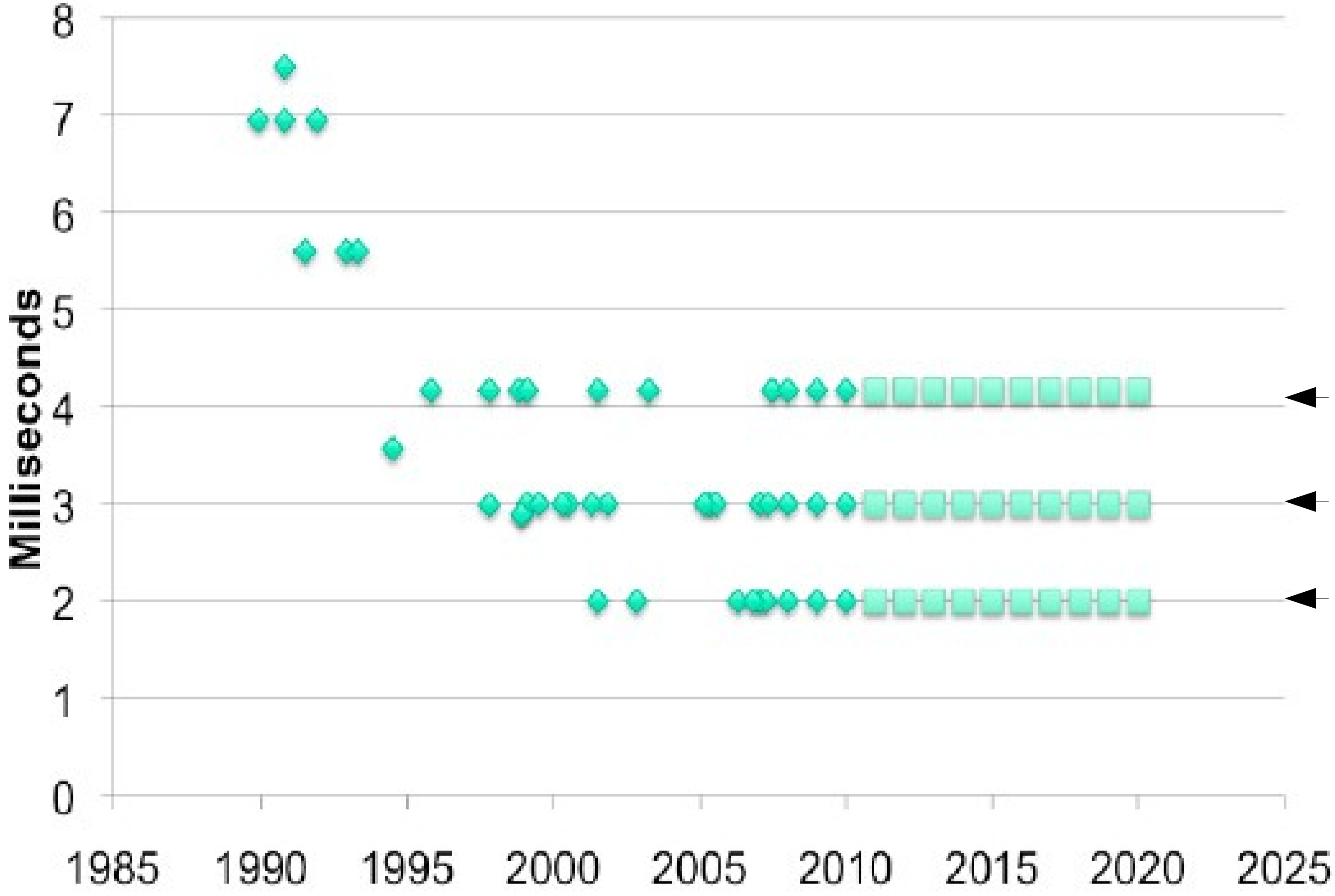
Disk Access Time = seek time + latency

Ave. Seek Time



http://www.storagenewsletter.com/news/disk/hdd-technology-trends-ibm

Disk Latency



7,200 RPM

10,000 RPM

15,000 RPM

http://www.storagenewsletter.com/news/disk/hdd-technology-trends-ibm

Conclusions (cont.)

- Flash has arrived just in time.
- Shared storage (NAS, SAN) will remain prominent, with additional emphasis on cost effective scale-out.
    - Gluster
    - pNFS
    - iSCSI, FCoE
        - Initiator and target
    - More storage boxes => better management is required:
        - libStorageMgmt
            - http://sourceforge.net/projects/libstoragemgmt/
- An opportunity for Linux as a low-cost scale-out storage server.

Thank-you.