Data Protection Technologies: What comes after RAID?

Vladimir Sapunenko,
INFN-CNAF

HEPiX Spring 2012 Workshop

Arguments to be discussed

- Scaling storage for clouds
- Is RAID dead?
- Erasure coding as RAID replacement
- RAIN – Redundant Array of Independent Nodes
- Some approaches and implementations

Challenges for data storage infrastructure

- Data volume increases
  - More devices
  - higher density
- Risk of data corruption increases over time
- increase of disc size
  - The enterprise SATA/SAS disks have just grown larger, up to 4TB now.
    - Just a few days ago, Hitachi boasted the shipment of the first 4TB HDD, the 7,200 RPM Ultrastar™ 7K4000 Enterprise-Class Hard Drive.
    - And just weeks ago, Seagate advertised their Heat-Assisted Magnetic Recording (HAMR) technology which will bring forth the 6TB hard disk drives in the near future
    - 60TB HDDs not far in the horizon.
  - long recovery time in usual RAID configurations

RAID

- Drive reliability in the 1980's and early 1990's became a non-factor due to RAID.
- However as drive capacities increase, the rebuild times to recalculate parity data and recover from drive failures is becoming so onerous that the probability of losing data is increasing to dangerous levels
  - Use of RAID5 well justified when probability of second failure during reconstruction is low.
- Current state: Time to rebuild one SATA 7.2K RPM disk in single (independent) RAID set at 30MB/s
    - 500 GB – 4 hours
    - 750 GB – 7 hours
    - 2 TB – 17 hours
  - Rebuild time can be much longer in case heavy loaded system
    - Just a week ago we had 2TB disk reconstructed in 4(!) days
- Risk of data loss

Probability of data loss

Disk size, GB	Rebuild rate, MB/s	Number of disks in FS	Data layout	Probability of data loss in 5 years, % (Manufac. spec)	Probability of data loss in 5 years, % (Real measured)
500	10	380	RAID-5	0.27	6.1
500	60	380	RAID-5	0.05	1.5
1000	10	380	RAID-5	0.54	11.9
1000	60	380	RAID-5	0.1	2.1
1000	10	380	RAID-6	0.0001	0.01
2000	10	1200	RAID-5	3.15	52.2
2000	10	1200	RAID-6	0.0014	0.15

Annual Failure rate for 2TB disks :

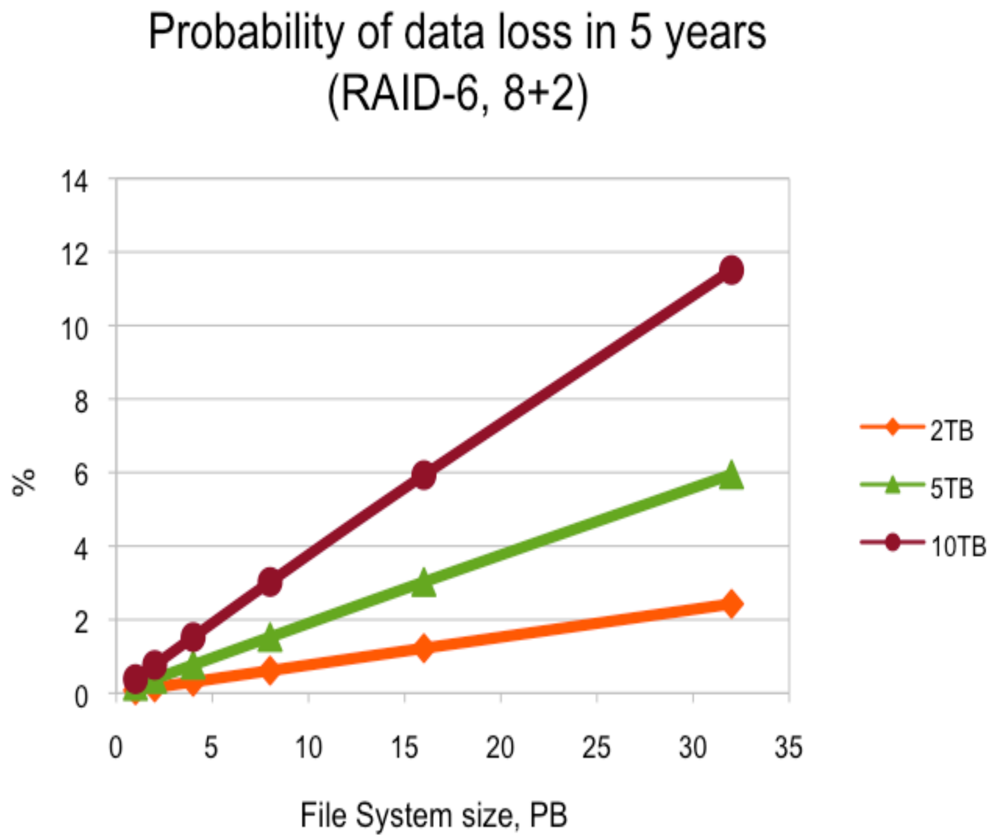
declared 0.73%;
estimate (World) 3.5%;
observed (CNAF) 2.3%

(http://www.zetta.net/docs/Zetta_MTTDL_June10_2009.xls)

RAID5 is dead

- increasing disk drive capacities → longer RAID rebuild times → exposing organizations to data loss
- Would be ~50% of probability of data loss in 5 years on 2PB file system
- Traditional RAID is increasingly less attractive at cloud scale
- RAID-6 provides much better protection

RAID-6: data loss probability



So, what comes after RAID?

- Storage bits happen to fail
  - storage protection systems need the ability to recover data from other storage.
- There are currently two main methods of storage protection:
  - Replication,
  - RAID.
- Erasure coding is method for storage protection
  - provides redundancy by breaking objects up into smaller fragments and storing the fragments in different places.
  - With erasure coding, the original data is recoded using Reed-Solomon polynomial codes into n fragments.
- The term: erasure coding
  - The data can survive the known erasure to up to n-m fragments. It is used in RAID 6, and in RAM bit error correction, in Blu-ray disks, in DSL transmission, and in many other places.

Erasure coding

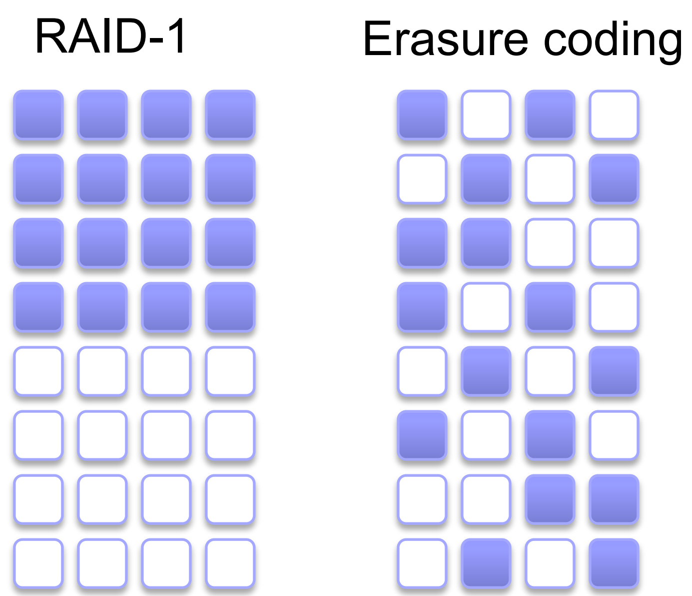
- Some definitions:
  - Number of fragments data divided into: *m*
  - Number of fragments data recoded into: *n (n>m)*
- The key property: stored object in *n* fragments can be reconstructed from any *m* fragments.
  - Encoding rate $r = m/n$ (<1)
  - Storage required $1/r$

Replication and RAID can be described in the erasure coding terms.

- Replication with two additional copies
  $m = 1,\ n=3,\ r = 1/3$
- RAID 5 (4+1)
  $m = 4,\ n = 5,\ r = 4/5$
- RAID 6 (8+2)
  $m = n - 2,\ r = 8/10$

Erasure coding (cont.)

- The usage has been historically constrained by the high processing overhead of computing for the Reed-Solomon algorithms.
- multi-core processor has radically reduced the cost of the processing component of erasure coding-based storage,
- future projections of the number of multi-cores will reduce it even further.



Example: availability

- two copies and 100 machines
  - 5 nodes failures - 99.8% availability
  - 50 nodes failure - 75.3% availability
- 10 fragment (5 data + 5 coded)
  - 5 nodes failure – 100% availability
  - 50 nodes failure - 98.7% availability

Availability of a block can be computed as follows:

P_o probability that a block is available
n total number of fragments
m number of fragments needed for reconstruction
N total number of machines in the world
M number of currently unavailable machines

$$P_o = \sum_{i=0}^{n-m} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}$$

The availability is kept constant, and the cost of the secure highly available cloud system is reduced by an order of magnitude

Some approaches and implementations

Disclaimer

The speaker has not any interest in any company mentioned hereafter.

What they will gain or loose as a consequence of this presentation, is just their business…

Example: Shutterfly.com case

- Web-based consumer publishing service that allows users to store digital images and print customized "photo books" and other stationary using these images.
- storage challenges:
  - Free and unlimited picture storage,
  - Perpetual archiving of these pictures – photos are never deleted,
  - Secure images storage at full resolution.
- 2 years ago, had 19 PB of raw storage
  - created between 7-30 TBs daily
- Near catastrophe accident: failure of a 2PB array
  - Lost access to 172 drives
  - no data was lost
  - needed three days to calculate parity
  - and three weeks to get dual parity back across the entire system

Shutterfly.com case (cont.)

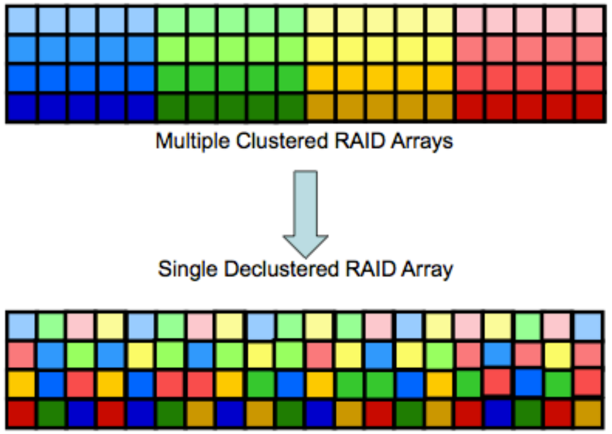
- After an extensive study of requirements and potential solutions, Shutterfly settled on an erasure code-based approach using a RAIN architecture (Redundant Arrays of Inexpensive Nodes)
  - Using commodity-style computing methodology
  - implement an N + M architecture (e.g. 10+6 or 8+5 versus a more limited RAID system of N+1, for example).
- Results:
  - Shutterfly has grown from 19 PBs raw to 30 PBs in the last 18 months.
  - The company is seeing 40% growth in capacity per annum.
  - The increase in image size is tracking at 25% annually.
  - There is no end in sight to these growth rates.
  - Shutterfly experiences typically 1-2 drive failures daily.

Perseus: Advanced software RAID for GPFS

50-disk arrays to 100,000-disk supercomputers

- Software RAID for scalable GPFS (NSD)
- Declustered RAID implementation
  - Spread data strips randomly across all array disks
  - Performance will be minimally affected by rebuilding array
- 2/3-fault tolerant erasure codes
  - "RAID-D2" or "RAID-D3"
  - Software Reed-Solomon
  - Optional 3/4-way mirroring
- End-to-end checksum
- Runs on generic servers with direct-attach disks
- Supersedes traditional external RAID controller
  - Reduces storage subsystem costs by 30 - 60 %

- Improved file system performance
  - With 100k disks, a storage array is always rebuilding
    - 100k disks * 24 / 400 khrs => 6 rebuilds per day
- Improved data integrity
  - RAID-5 is non-starter with 100k disks: MTTDL ~ 9 days!
    - Hard error rate of 1-in-10^{15} bits implies data loss every ~26th rebuild, or once every 26 / 6 = 4 days
  - RAID-D2 (8+2P stripes): MTTDL ~ 100 years
  - RAID-D3 (8+3P stripes): MTTDL ~ 130 million years!



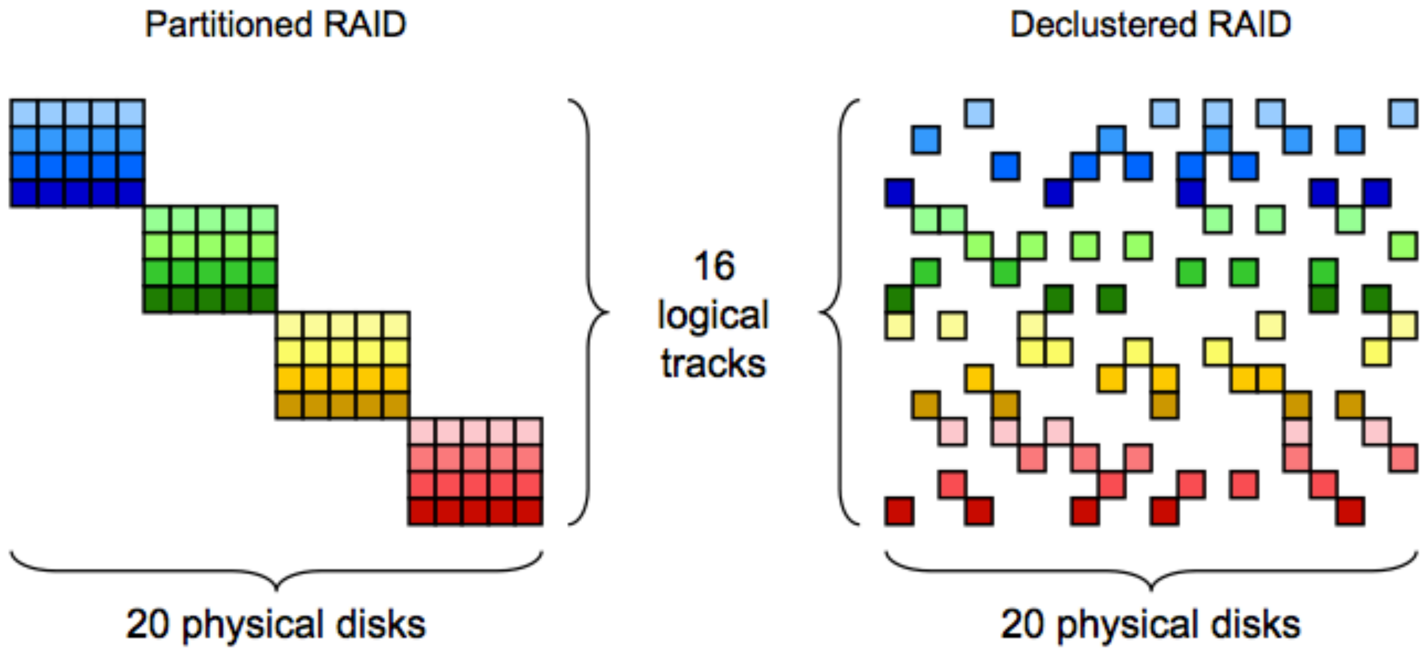
"GPFS™ Native RAID integrates the functionality of an advanced storage controller into the GPFS NSD server. Unlike an external storage controller, where configuration, LUN definition, and maintenance are beyond the control of GPFS, GPFS Native RAID takes ownership of a JBOD array to directly match LUN definition, caching, and disk behavior to GPFS file system requirements."
GPFS V3.4: GPFS Native RAID Administration and Programming Reference

GPFS Software RAID

- Implement software RAID in the GPFS NSD server
- Motivations
  - Better fault-tolerance
  - Reduce the performance impact of rebuilds and slow disks
  - Eliminate costly external RAID controllers and storage fabric
  - Use the processing cycles now being wasted in the storage node
  - Improve performance by file-system-aware caching
- Approach
  - Storage node (NSD server) manages disks as JBOD
  - Use stronger RAID codes as appropriate (e.g. triple parity for data and multi-way mirroring for metadata)
  - Always check parity on read
    - Increases reliability and prevents performance degradation from slow drives
  - Checksum everything!
  - Declustered RAID for better load balancing and non-disruptive rebuild

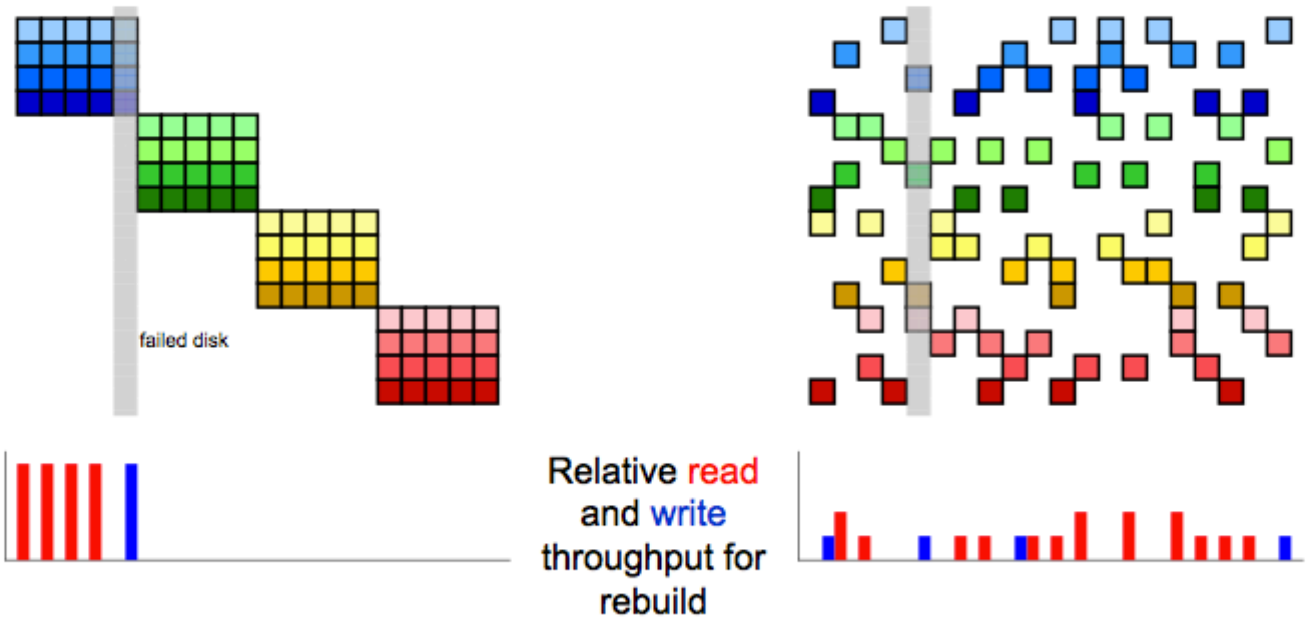
De-clustered RAID

- The logical tracks are spread out across all physical HDDs and the RAID protection layer is applied at the track level, not at the HDD device level. So, when a disk actually fails, the RAID rebuild is applied at the track level. This significant improves the rebuild times of the failed device, and does not affect the performance of the entire RAID volume much.

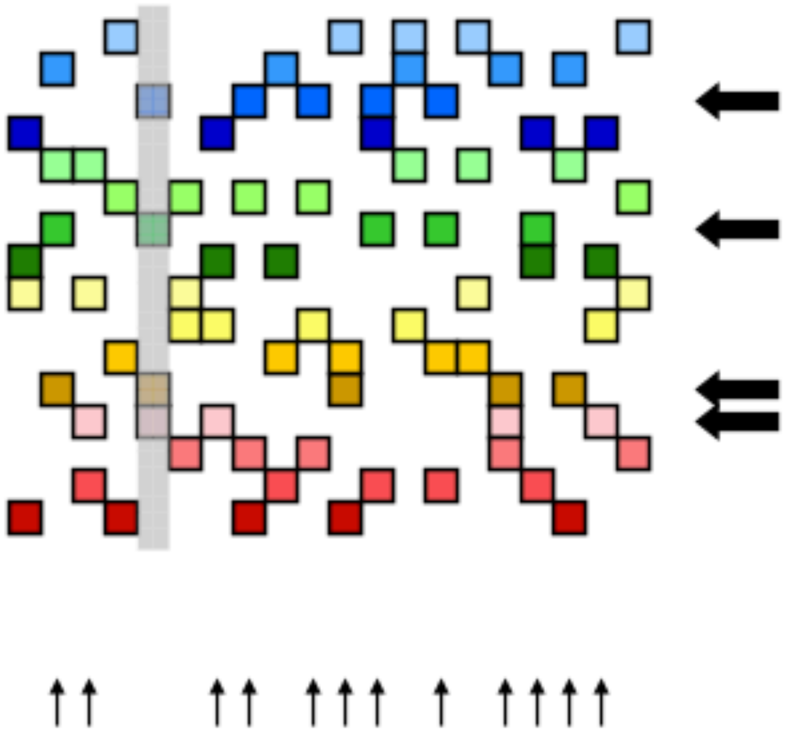


Rebuild Workflow

- When a single disk fails in the conventional array on the left, the redundant disk is read and copied onto the spare disk, which requires a throughput of 4 strip I/O operations.
- When a disk fails in the declustered array, all replica strips of the four impacted tracks are read from the surviving disks and then written to one spare strips, for a throughput of 2 strip I/O operations. The bar chart illustrates disk read and write I/O throughput during the rebuild operations.



Rebuild (2)



Upon the first failure, begin rebuilding the tracks that are affected by the failure (large arrows).

Many disks involved in performing rebuild, so work is balanced, avoiding hot spots.

Isn't it nice?

- Part of standard distribution since v.3.4
- Very well documented (as usual)
- BUT: supported only on AIX
  - GPFS Native RAID is more analogous to the code that runs inside a DS8000 than to a filesystem driver such as historical GPFS.
  - No timeframe for LINUX porting

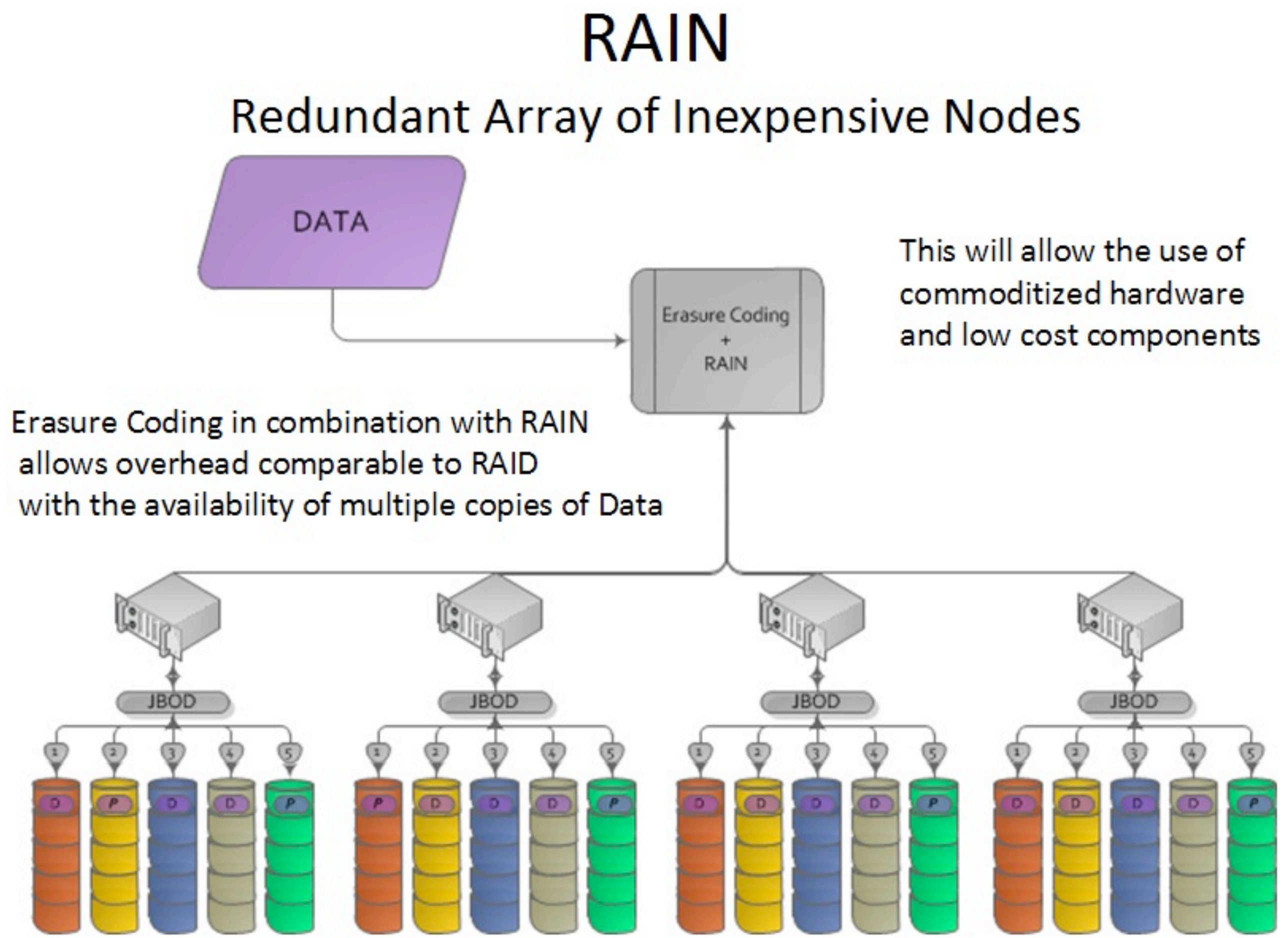
ZFS

- an file system invented by Sun, on which a user can construct mirror, single fault-tolerant or double fault-tolerant soft Raid systems.
- Its double-erasure code is the Reed -Solomon (RS) code.

RAIN: Redundant Array of Inexpensive Nodes.

- AKA:
  - reliable array of independent nodes
  - random array of independent nodes
- is used to increase fault tolerance
- an implementation of RAID across nodes instead of across disk arrays
- can provide fully automated data recovery in a LAN or WAN even if multiple nodes fail
- partitions storage space across multiple nodes in a network

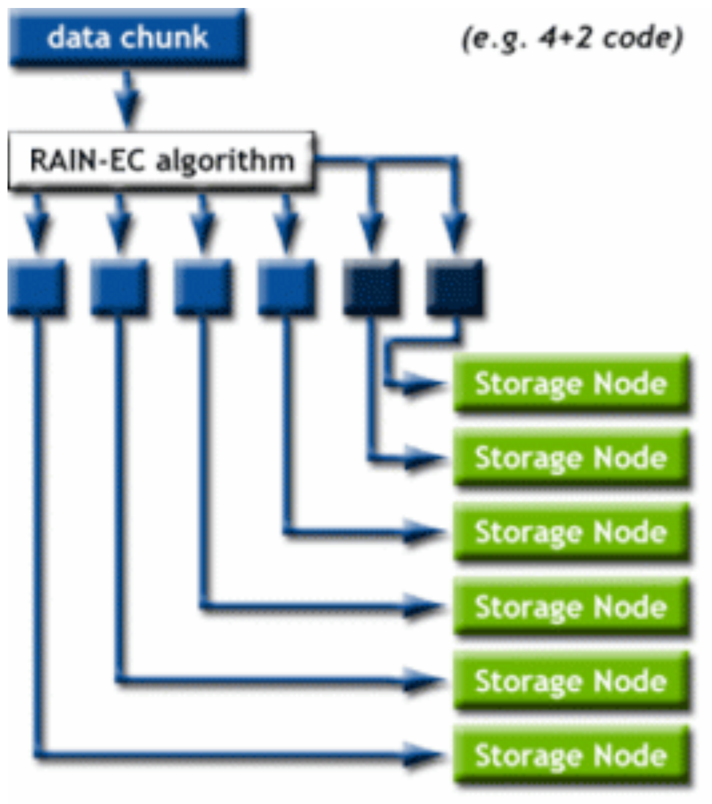
Next step: Erasure coding and RAIN



Source: Justin Stottlemyer, Director Storage Architecture, Shutterfly

Implementation: Permabit

- http://permabit.com/enterprise/technologies/rain-ec/
- RAIN-EC is a standard feature in the Permabit Enterprise Archive. There are no additional licenses, software, or hardware to purchase.
- **Erasure Coding:** Enables Permabit to provide scalable data protection solutions to the multiple-petabyte level and is uniquely employed in Permabit Enterprise Archive.
- Can survive multiple simultaneous failures (multiple drives, power, CPU, memory, etc.) without data loss.



StreamScale

- StreamScale - a startup venture developing innovative digital recording and analysis solutions for the professional CCTV industry.
- Licensing "Big Parity" **C language Erasure Coding library**
  - **Compatible with Linux (GCC), Mac OS X (GCC) and Windows (Intel)**
  - can prevent Silent Data Corruption and can provide up to 127 parity drives (versus 2 for RAID6) without reducing performance.
  - was live tested against ZFS with 3 parity drives and demonstrated a 25X performance advantage.

Web Object Scaler

- DataDirect Networks (DDN) have introduced a WOS 2.0 (Web Object Scaler) object storage system which they claim delivers up to 55 billion small object reads and 25 billion writes/day.
  - This is twice as many as the Amazon S3 system and twenty times the throughput of the DARPA system.
- The components of the DDN system include high density storage appliances, which deliver 2PB per rack and 23PB per cluster. Up to 25 billion objects can be stored in a rack.
- The sustained performance and data protection is achieved by combining the traditional DDN 8+2 hardware enhanced data striping together with de-clustered erasure coding.
- DDN has introduced an asynchronous replication capability that writes a second copy locally before replication to a remote site.

Conclusions

- RAID5 is dead (confirmed)
- RAID6 is still feasible for up to 10PB file systems
- Erasure coding and RAIN looks very promising
  - Lowering costs, increasing availability
  - is being commercialized by some vendors
  - Q: Is there any Open Source projects around?

Some references

- MTTDL calculator: http://www.zetta.net/docs /Zetta_MTTDL_June10_2009.xls
- Reducing the Cost of Secure Cloud Archive Storage by an Order of Magnitude, Wikibon 2011.
- David Floyer. Erasure Coding and Cloud Storage Eternity, Wikibon 2011.
- Theo Schlossnagle. Omniti.com, Thu, 11 Jun 2009. **Concepts of Cloud(ish) Storage**
- Hakim Weatherspoon and John D. Kubiatowicz. Erasure Coding vs. Replication: A Quantitative Comparison.
- GPFS V3.4: GPFS Native RAID Administration and Programming Reference
  SA23-1354-00
- http://permabit.com/enterprise/technologies/rain-ec/
- *DDN Whitepaper*. WOSTM. Breaking the Limits of Cloud Storage. Performance and Scalability – An. Introduction to DataDirect Networks'. *Web Object Scaler*™

Questions?

Comments?