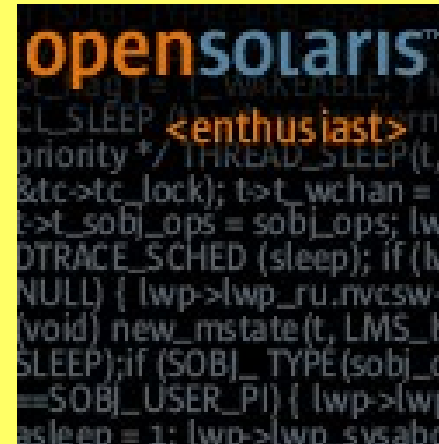


Software or Hardware RAID?

Robert Miłkowski
System Group Manager
Wirtualna Polska



Why Do We Need RAID?

- ◆ Performance
 - ◆ More IOPS and/or throughput
- ◆ Scalability
 - ◆ More capacity
- ◆ **Reliability**
 - ◆ Disk drives fail
 - ◆ Disk controllers fail
 - ◆ Data corruption

RAID – it's simple

- ◆ No, it's not
- ◆ RAID systems are complex
 - ◆ RAID-0, RAID-1, RAID-5, RAID-10, RAID-6, ...
 - ◆ Nonvolatile memory caches
 - ◆ Storage controllers
 - ◆ Built-in block checksumming
 - ◆ Virtualized storage allocation
 - ◆ MAID, PAR RAID, ...

RAID Implementations

- ◆ Software RAID
 - ◆ SVM, VxVM, LVM, Raidtools, ZFS, ...
- ◆ Hardware RAID
 - ◆ PCI RAID Controllers
 - ◆ Storage Arrays
 - ◆ EMC, Hitachi, NetApp, IBM, Sun, ...
- ◆ Software RAID on top of hardware RAID
- ◆ “Application” RAID

HW RAID is Better

Really?

What does HW RAID mean?

- ◆ Software running on a dedicated HW
 - ◆ Controlled and tested environment
 - ◆ Out of date HW comparing to modern CPUs
- ◆ Arrays are often x86 servers now
 - ◆ EMC Clariion CX3-40 is a 2x 2.8GHz, 4GB RAM Intel server – **there's no magic**
- ◆ **Non-volatile memory cache**

Nonvolatile Memory Cache

- ◆ Speeds up write performance
 - ◆ No performance gain for large sequential writes
 - ◆ No performance gains for a large stream of IOs
 - ◆ Greatly helps RAID-5 write performance
 - ◆ Clever software can solve that problem (RAID-Z)
- ◆ Array memory cache is **expensive**
 - ◆ Servers usually have much more RAM anyway
 - ◆ Easier to add memory to servers than to an array

PCI RAID Management

- ◆ Complex monitoring and statistics
 - ◆ *cat /proc/SCSI/*? dellmgr? RaidMan.sh? megarc? raidctl? a:\dptmgr? BIOS? ...?*
 - ◆ Each model behaves differently – error prone
- ◆ Can't easily move data between ctrls
- ◆ Feature set is basically not expandable
- ◆ Impossible to buy same card in a future

Array RAID Management

- ◆ Every vendor provides different tools
- ◆ Every vendor uses different terminology
- ◆ Different arrays with different feature set
- ◆ Can't easily move data between arrays
- ◆ Software updates are costly
- ◆ Good centralized management for several platforms (Solaris, Linux, AIX, Windows)

SW RAID Management

- ◆ You can standardize on one solution
 - ◆ ZFS or SVM on all Solaris servers (x86, SPARC)
 - ◆ Raidtools on Linux
 - ◆ VxVM on Solaris, Linux, Windows, ... - expensive
- ◆ Easy monitoring, management, statistics and troubleshooting – same everywhere
- ◆ Easy to get the latest features/fixes – free
- ◆ Easy to move data between servers

HW Mirrored Boot Disks

- ◆ Most x86 server boards provide RAID-1
 - ◆ Allows to easily boot from a mirror
 - ◆ Utilizes **software driver** in a system
 - ◆ Different vendors, different drivers, different tools
- ◆ PCI RAID Cards for boot disks
 - ◆ No real-world advantage from its cache
 - ◆ Driver needed
 - ◆ Different tools for different adapters

SW Mirrored Boot Disks

- ♦ Minor complications with booting on x86
- ♦ Works on every server you can boot
- ♦ Easier to manage
 - ♦ The same tools on all servers (large datacenters)
 - ♦ Learn once use everywhere
 - ♦ Less error prone
- ♦ Practically the same performance as HW
- ♦ **Ability to always use latest technology**

Performance

- ◆ HW RAID is always faster
 - ◆ **NO, IT IS NOT**
 - ◆ More latency
 - ◆ Low-end RAID has often less power than modern CPUs
- ◆ “understanding” data
- ◆ XOR
- ◆ File servers
- ◆ Real workloads

IOPS

- ◆ SW RAID needs to issue more IOPS
 - ◆ The same number of IOPS are issued to disks
 - ◆ More IOPS **only** between server and ctrl/array
 - ◆ **It's not a problem in 99.99% of environments as the limiting factor usually are disks themselves not a server**
- ◆ **With lot of IOPS ctrls/arrays introduce additional latency**
- ◆ The same applies to throughput

IOPS - filebench/varmail

- ◆ SE3510, 12x 73GB 15K, **HW RAID-10**, ZFS
 - ◆ IO Summary: 503112 ops 8320.2 ops/s, (1280/1280 r/w) 41.0mb/s, 296us cpu/op, **5.9ms latency**
- ◆ SE3510 **JBOD**, 12x 73GB 15K, **ZFS RAID-10**
 - ◆ IO Summary: 558331 ops 9244.1 ops/s, (1422/1422 r/w) 45.2mb/s, 312us cpu/op, **5.2ms latency**
- ◆ <http://milek.blogspot.com/2006/08/hw-raid-vs-zfs-software-raid.html>

IOPS – RAID-5 vs. RAID-Z

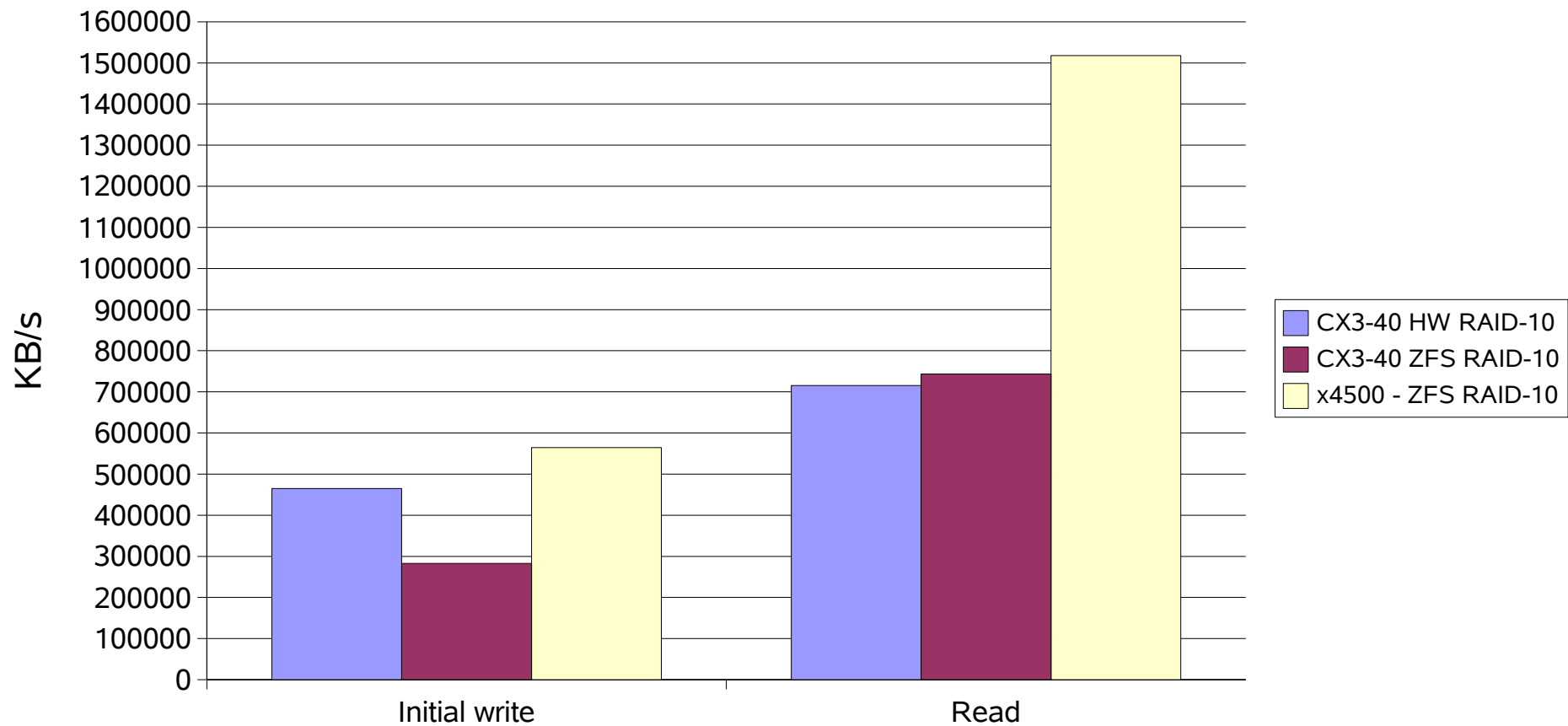
- ◆ SE3510, 6x 73GB 15K, **HW RAID-5**, ZFS
 - ◆ IO Summary: 444386 ops 7341.7 ops/s, (1129/1130 r/w) 36.1mb/s, 297us cpu/op, **6.6ms latency**
- ◆ SE3510 **JBOD**, 6x 73GB 15K, **ZFS RAID-Z**
 - ◆ IO Summary: 457767 ops 7567.8 ops/s, (1164/1165 r/w) 36.9mb/s, 340us cpu/op, **6.4ms latency**
- ◆ <http://milek.blogspot.com/2006/08/hw-raid-vs-zfs-software-raid-part-ii.html>

Throughput

- ◆ Read/Write throughput approx. the same
- ◆ SW RAID-10 needs 2x **write** throughput
 - ◆ Modern servers (even x86) can easily put >1GB/s
 - ◆ HBAs and/or disks are often the limiting factor
 - ◆ How much can you put thru a network?
 - ◆ Most environments are IOPS bound not throughput
 - ◆ **Benchmarks are just benchmarks** – how much throughput your application needs?
 - ◆

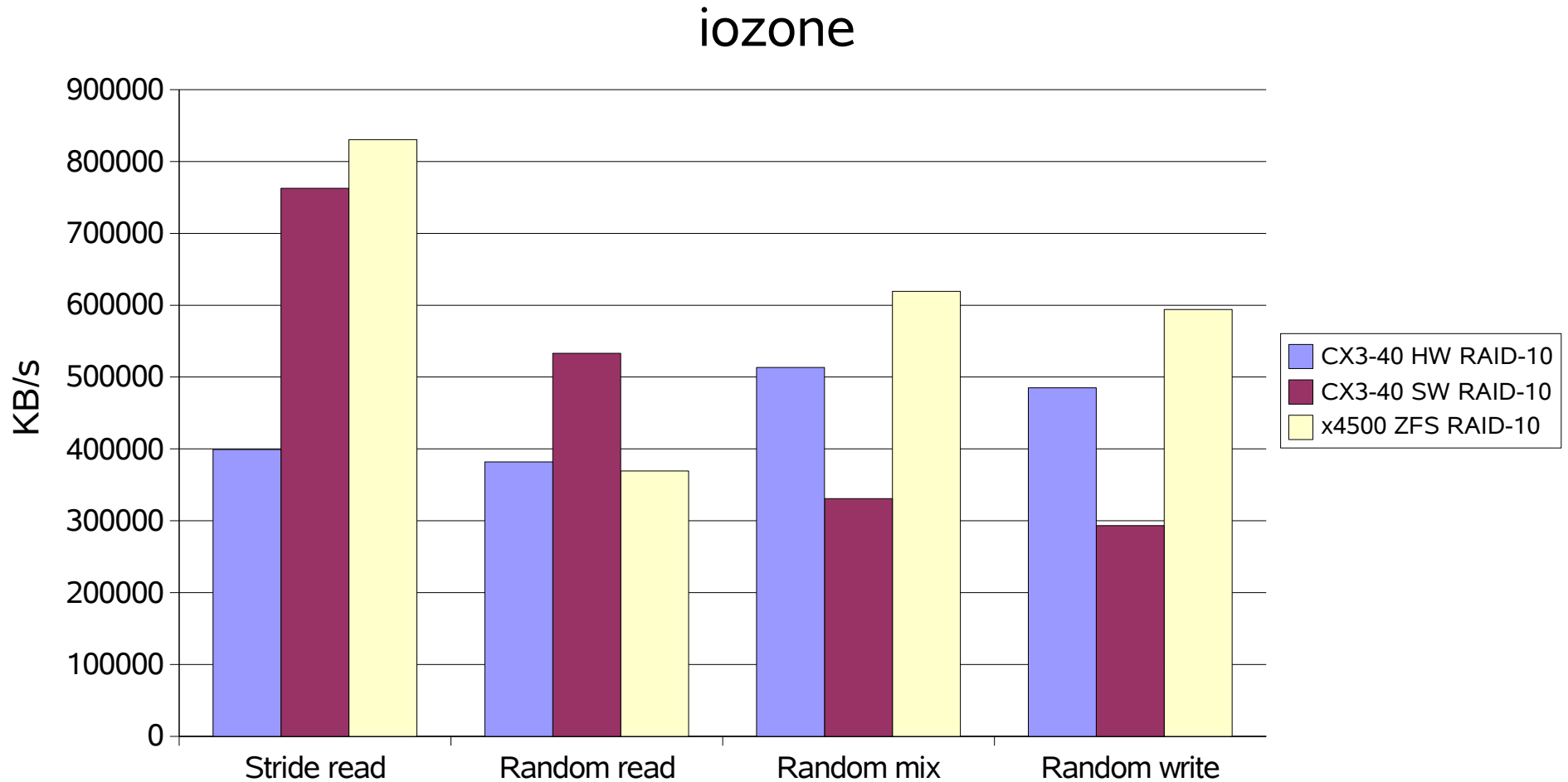
Throughput

iozone



<http://milek.blogspot.com/2007/04/hw-raid-vs-zfs-software-raid-part-iii.html>

Throughput – Random IO

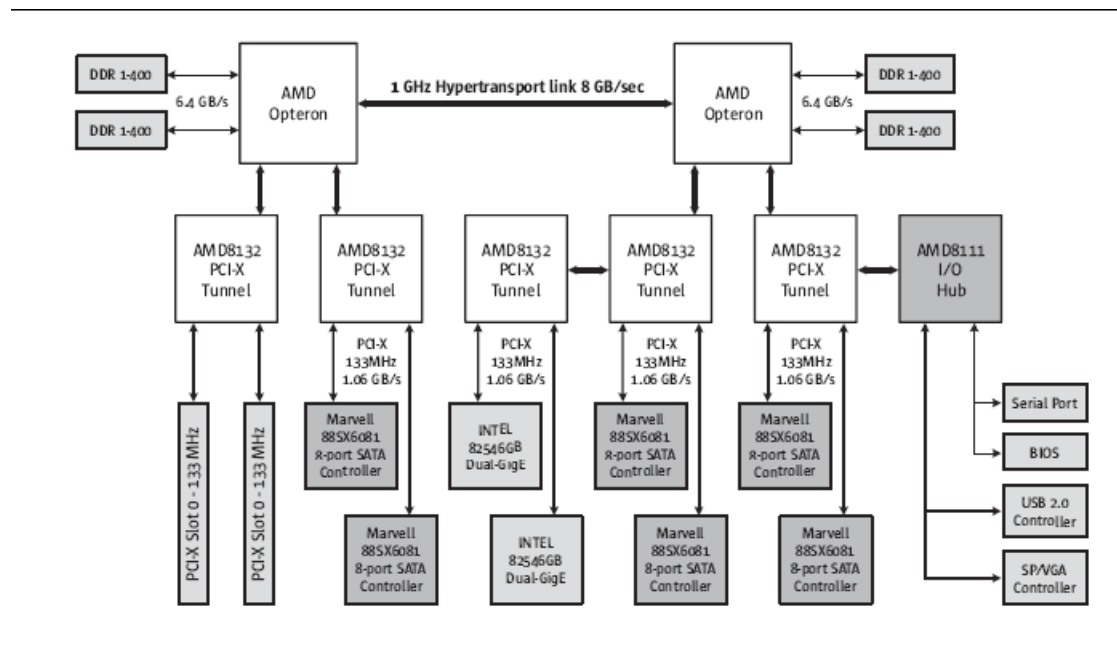


<http://milek.blogspot.com/2007/04/hw-raid-vs-zfs-software-raid-part-iii.html>

Thumper - x4500

- ◆ Storage box or server?
 - ◆ 4U, 48 disks, 4 cores, 4 GbE, 32GB RAM
 - ◆ 24TB RAW storage
 - ◆ 10x Thumpers in a rack = **240TB RAW**
 - ◆ 48TB RAW soon (480TB RAW in a rack!)
 - ◆ NFS, CIFS, iSCSI, local storage
 - ◆ **No RAID controllers**
 - ◆ 2GB/s real throughput to disks
 - ◆ Low cost

X4500 - details



<http://www.sun.com/servers/x64/x4500/arch-wp.pdf>

Data Corruption

- ◆ Data can be corrupted in many places
 - ◆ Disk drive, array, SAN, HBA, driver, ...
- ◆ Why haven't we noticed it before?
 - ◆ Well, we did – we've just got used to it
 - ◆ It's a matter of scale
 - ◆ Large storage capacities means more corrupted data
 - ◆ Disks are getting bigger and bigger
 - ◆ Nothing unusual managing hundreds TBs of data or more

Data Corruption - Example

- ◆ Document id: MIGR-60072 (2006-03-23)
 - ◆ Model affected: IBM ServeRAID-7k

*“There is a very rare occasion where data could be **corrupted** while using the ServeRAID 7k controller option on IBM eServer xSeries 236 or 346 system. This issue has only been found to occur when both the ServeRAID 7k controller is used and the system has 8GB of RAM or greater. Additionally, this issue may arise not only when a hard drive is attached to the SCSI bus, but also when any other device (e.g. tape or drive) is attached to the SCSI bus. The system may have **corrupted data without any notification to its user**. Also, system stability issues might be observed, like system hangs or crashes.”*

Data Corruption - Example

- ◆ Document id: MIGR-60771 (**2006-03-23**)
 - ◆ Model affected: IBM ServeRAID 8i

*“A system equipped with a ServeRAID 8i option has a substantial chance of experiencing **corrupted data** if using a RAID 5EE array and a drive failure occurs. The corruption can occur after a compaction/expansion cycle due to such a drive failure.”*

Data Corruption - Example

- ◆ Document id: MIGR-55696
 - ◆ Model: IBM TotalStorage DS4100 (FAStT100)
 - ◆ Look for “corruption” in a changelog
 - “Data corruption after large VolGrp with 2 luns”
 - “Data corruption occurred after spindown of GHS drive during copy back followed shortly by reset of the controller”
 - “Data corruption reported during Failover and DSS”
 - “Data corruption while running I/O to flashcopy on **fastt600** where the LBA in the head and tail did match as expected”

Data Corruption - Example

- ◆ Sun Alert ID: 102815 (22-Feb-2007)
 - ◆ Models affected: Sun SE 3310, 3320, 3510, 3511

*“The above raid arrays (single or double controller) with "Write-Back Caching" enabled on Raid 5 LUNs (or other raid level LUNs and an array disk administration action occurs), **can return stale data** when the I/O contains writes and reads in a very specific pattern.”*

Sun Alert ID: 102815 Workaround

“Use ZFS to detect (and correct if configured) the Data Integrity Events.”

“If not using a filesystem make sure your application has checksums and identity information embedded in its disk data so it can detect Data Integrity Events.”

End-to-end Data Integrity

- ◆ Checksum is checked in a server memory
 - ◆ Entire IO path is checked
 - ◆ Detection and correction of
 - ◆ Writes of physically and logically corrupt blocks
 - ◆ Writes of blocks to incorrect locations
 - ◆ Partially written blocks
 - ◆ Phantom writes
 - ◆ other
- ◆ There are commercial and free solutions

Oracle HARD

- ◆ Hardware Assisted Resilient Data Initiative
 - ◆ EMC, HP, Hitachi, NetApp, Sun, ...
- ◆ Data Integrity Initiative (DII)
 - ◆ Oracle, Emulex, LSI, Seagate
 - ◆ Expected in 2008
- ◆ Software and hardware support needed
 - ◆ Works only with specific applications and array
 - ◆ **Expensive** to implement

DB Validator – Lightning 9900V

<http://www.internetnews.com/bus-news/article.php/1449641>

“While data corruption in storage networks is rare, it can be costly if the problem goes undetected for some time. The recovery process can be slow and costly, depending on when the database was last backed-up.”

“[...] data corruption that occurs outside of the database is very difficult to detect and can be very expensive and time consuming to fix after the fact. Corruption can occur in any or all of the layers before writing data into storage; for example, while passing through the operating system, channel adapter, or network. In these cases, since the output data is written without error into the storage, the companies say the database cannot detect the corrupted data until it tries to read the data, at which time, a read error occurs and the system stops.”

Google FS

- ♦ 3-way mirror between servers by default
- ♦ 32-bit checksum for each 64KB block
 - ♦ Checked on each read
 - ♦ Self healing
 - ♦ Background scrubbing
- ♦ Not POSIX-compliant
- ♦ Proprietary, not publicly available
- ♦ <http://labs.google.com/papers/gfs.html>

Google FS - Checksums

- ◆ <http://labs.google.com/papers/gfs.html>

*“Some of our biggest problems were disk and Linux related. Many of our disks claimed to the Linux driver that they supported a range of IDE protocol versions but in fact responded reliably only to the more recent ones. Since the protocol versions are very similar, these drives mostly worked, but occasionally the mismatches would cause the drive and the kernel to disagree about the drive’s state. **This would corrupt data silently** due to problems in the kernel. This problem motivated our **use of checksums to detect data corruption,**”*

“Additionally, we use checksumming to detect data corruption at the disk or IDE subsystem level, which becomes all too common given the number of disks in the system.”

Enterprise vs. Commodity Disks

- ◆ UER for SATA disk is 1 in 10^{14} bits read
 - ◆ 1 UER in ~12TB read
 - ◆ Rebuilding a 500GB disk in a RAID-5 group of 5 disks means ~20% probability of UE
 - ◆ 140GB SAS disk has <1% probability of UE
 - ◆ Risk can be reduced by
 - ◆ Intelligent rebuild (only used blocks)
 - ◆ Background scrubbing
 - ◆ Use of RAID-1 or RAID-6 instead of RAID-5

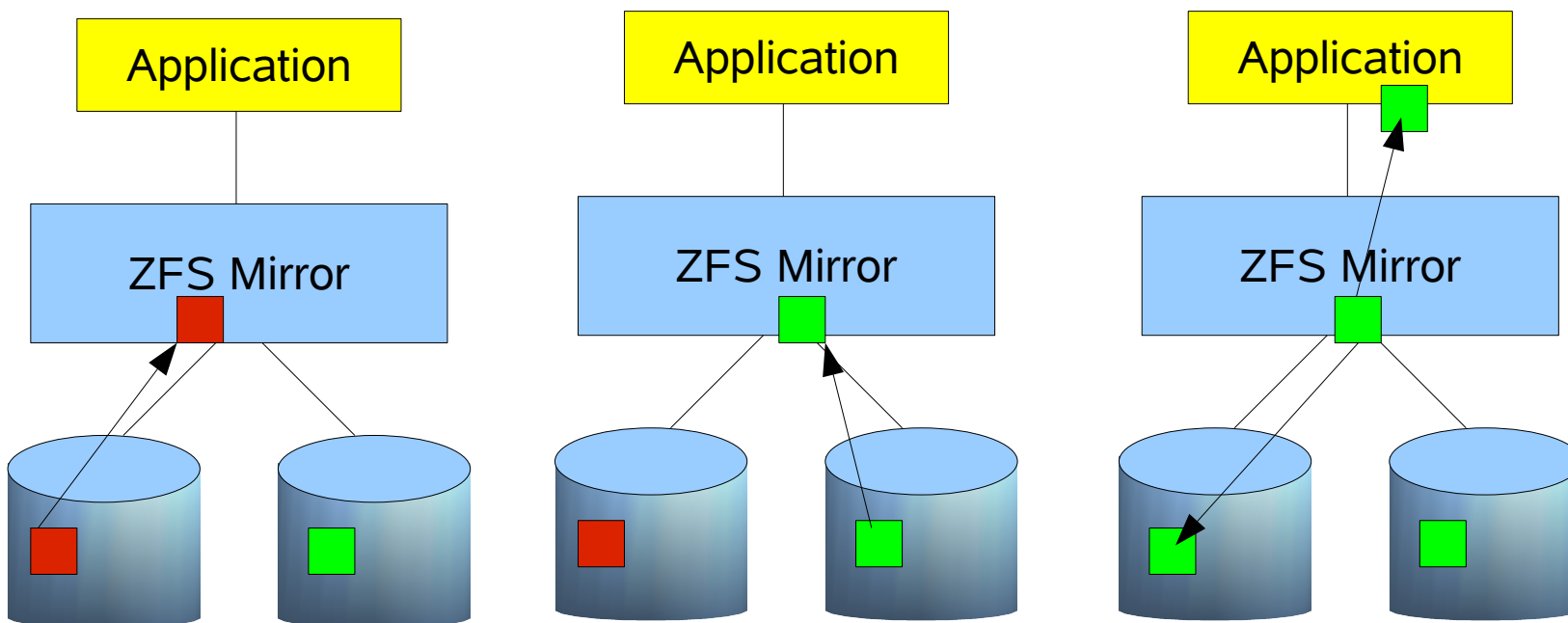
◆ <http://www.usenix.org/publications/login/2006-06/openpdfs/chan.pdf>

◆ http://download.microsoft.com/download/9/8/f/98f3fe47-dfc3-4e74-92a3-088782200fe7/TWST05005_WinHEC05.ppt

ZFS

- ◆ End-to-end data integrity
 - ◆ Great for cheap hardware
- ◆ Easy to manage
- ◆ Pooled storage
- ◆ Faster disk re-synchronization
- ◆ Open sourced, free
- ◆ Works with every application

ZFS Self Healing



Data Corruption – My Experience

- ♦ I've observed data corruption on IBM, EMC, Sun, Dothill arrays
 - ♦ **In all cases only when SATA disks were used**
 - ♦ 24+ hours waiting for fsck – later migrated to ZFS
- ♦ I've observed data corruption due to failing SCSI controller in a server
- ♦ So far one block corruption detected and corrected on my workstation (thanks ZFS)

Data Corruption - Conclusion

- ♦ Classic RAID does not protect you from data corruption
- ♦ Arrays themselves do not protect you from data corruption
- ♦ Technologies like Oracle's HARD are expensive and work only with selected hardware and software
- ♦ ZFS is the solution

Conclusions

- ◆ HW RAID is more **expensive**
- ◆ HW RAID is **more complex** to manage
 - ◆ Except for centralization
- ◆ HW RAID is **not necessarily faster**
- ◆ HW RAID itself offers **less reliability**

So why people bother with it?

Old habits die hard

Next Generation RAID?

- ◆ How to turn unreliable parts into reliable and scalable solution?
 - ◆ Haystack
 - ◆ http://martingreen.typepad.com/forward_looking_statement/2006/06/more_on_haystac.html
 - ◆ Honeycomb
 - ◆ Free & open sourced
 - ◆ <http://www.opensolaris.org/os/project/honeycomb/>
 - ◆ GoogleFS
 - ◆ ZFS
 - ◆ Free & open sourced

Cheap And Reliable NAS

- ◆ 2x x86 server
- ◆ Cheap JBOD
- ◆ Free software
 - ◆ Sun Cluster
 - ◆ Solaris 10 (IPMP, MPxIO)
 - ◆ ZFS

Personal files

- ◆ Where do you store your pictures?
 - ◆ CD-RW – low capacity, slow, not reliable
 - ◆ DVD-RW – medium capacity, slow, not reliable
 - ◆ Disk drive – large capacity, fast, not reliable
- ◆ How do you share your data with friends?
 - ◆ External disks? Memory sticks? e-mail?
 - ◆ Network disk?

My Personal Data Archive

- ◆ Old Celeron 500MHz PC
 - ◆ 2x 250GB mirrored ATA disks (different vendors)
 - ◆ 256MB RAM
 - ◆ Solaris 10 + **ZFS**
 - ◆ Samba
 - ◆ Local network only (direct link to notebook) + DHCP
 - ◆ No keyboard, monitor, mouse – just power on/off button (clean shutdown with ACPI)
- ◆ **MUCH more reliable than CDs, DVDs, ...**

E-disk

- ◆ WebDAV
 - ◆ Easy to use, built-in in most OS'es
- ◆ Personal files
 - ◆ Remote backup
 - ◆ Music, pictures, videos
- ◆ Corporate files
 - ◆ Encryption, secure download/upload
- ◆ What is a cost?

E-disk – Examples

- ◆ Amazon S3 (Simple Storage Service)
 - ◆ 150\$ for 100GB/year
 - ◆ Additional bandwidth costs
- ◆ Joyent BingoDisk
 - ◆ 199\$ for 100GB/year
 - ◆ Free bandwidth
- ◆ Carbonite
 - ◆ 50\$ per year for unlimited storage

Q&A

...