Using Nexenta as a cloud storage

dr. Matjaž Pančur, UL FRI

dr. Mojca Ciglarič, UL FRI
Agenda

- Standardization and commoditization of storage HW, “open storage”
- What is NexentaStor and who is Nexenta
- Use cases:
  - High end system: VMworld 2011
  - Low end system: cloud storage on UL FRI
The move to standardization

- Hardware components have become more reliable
- More features moved into software
  - RAID
  - Replication
- Some bespoke features remaining in silicon
  - 3PAR dedicated ASIC
  - Hitachi VSP virtual processors
- Reduced Cost
  - Cheaper components
  - No custom design
  - Reusable by generation
- Higher Margins
It’s all about SW

- Today, storage arrays look like servers
  - Common components
  - Generic physical layer
- Independence from hardware allows:
  - Reduced cost
  - Design hardware to meet requirements
    - Quicker to market with new hardware
  - More scalability
  - Quicker/Easier upgrade path
  - Deliver new features without hardware upgrade

Source: Nexenta European User Conference, 2011
It’s all about SW

- Many vendors have produced VSAs (Virtual Storage Appliances)
  - Lefthand/HP, Gluster, Falconstor, Openfiler, OPEN-E, StorMagic, NexentaStor, Sun Amber Road, ...
- Most of these run exactly the same codebase as the physical storage device
- As long as reliability & availability are met, then the hardware is no longer significant

Source: Nexenta European User Conference, 2011
Storage Virtualization & Hardware Independence

- VSAs show closely coupled hardware/software is no longer required
- Software can be developed and released independently
  - Feature release not dependent on hardware
- Hardware can be designed to meet performance, availability & throughput, leveraging server hardware development
  - Branches with smaller hardware
  - Core data centres with bigger arrays
  - Both using same features/functionality

Source: Nexenta European User Conference, 2011
Proprietary Disk System

- Storage Software
- Proprietary Head Unit
- Controller Hardware
- Disks

Open Storage Server System

- NexentaStor
- Commodity Hardware
- Industry Controllers
- Disks

10X Storage cost
Must buy vendor units
Must buy vendor controller
Must buy vendor disk at 5x markup

Lower cost
Better selection
Market price disks

Source: http://www.slideshare.net/hnwilcox/open-stack-ucb-virt-theory
Nexenta and NexentaStor
What is NexentaStor?

Software-based, unified storage appliance

Leading OpenStorage solution
• Runs on standard hardware

Key features:
• End to end data integrity
• Detect and correct data corruption
• Unlimited file size & snaps
• Synchronous and asynchronous replication

Superior storage for virtualized environments

Nexenta Systems is a privately-held company
Based in Mountain View, California
Founded in 2005
http://www.nexenta.com
What is NexentaStor

**NexentaStor**
- Hardware independent
- NAS/SAN/iSCSI/FC
- CDP via ZFS snapshots
- CDP via block sync
- Advanced graphics
- Event based API

**Optional modules**
- + VM management + WORM + Windows ‘Deloreon’ + HA Cluster

**Enterprise Edition**
- + search + synch replication
- + ease of use + remote management

**NexentaOS:**
- Loves multiple cores
- Boot level ZFS
- >1 million downloads

**Solaris kernel**
- multi-core + clustering

**Debian / Ubuntu**
- #1 community + packaging

**ZFS**
- checksums
- not volumes
- 128 bit

**ZFS:** File system
- universal: SAN/NAS/iSCSI/FC
- performance: variable block size + prefetch
- Software RAID that identifies and **corrects** data corruption
NexentaStor

ZFS
- 128-bit checksums
- Hybrid storage pools
- Thin provisioning
- In-line compression
- In-line and in-flight de-duplication
- In-line virus scan

Storage Services and Mgt
- Infinite snapshots
- Asynchronous & synchronous replication
- HA Cluster
- Windows backup
- VM management
- WORM
Competitively priced

- NexentaStor runs on commodity x86 servers
  - Gives customer more control over hardware component choices
- Customers own **perpetual** licenses
  - Hardware refresh can proceed without any additional payment to Nexenta
  - Refresh of legacy storage is often more expensive than the initial purchase
- Reduce effective price through storage efficiency:
  - instantaneous snapshots
  - compression
  - de-duplication
  - thin provisioning
  - hybrid storage pools
  - reservations
  - quotas (including user and group quotas)
Flexibility and Scalability

- Flexible
  - Unified storage appliance
    - NAS + SAN
  - Supports key protocols
    - CIFS, NFS, iSCSI, WebDAV
  - APIs and Web GUI to easily reconfigure

- Designed to scale
  - Multi-core support
  - SSD support
  - “No limits”
  - “just add hardware – and it accelerates”
  - Increased chance of silent data corruption as you scale.
    - NexentaStor can detect and correct the silent corruption.
Elastic

- Thin provisioning
- Ability to easily or automatically grow (but not shrink!) volumes
Easy to Manage

- Web GUI
- Command-line shell
  - Auto-complete and help facility
- REST APIs
  - Also D-BUS APIs with Perl, PHP, and C bindings
- Scheduled storage services
  - Replication, snapshots, scrubs
Ease of management at scale

NexentaStor's NamespaceCluster
NFS Referrals
Optimized for virtual machines

- Unifies management of storage for VMware, Citrix Xen and Hyper-V
- View VM storage usage for storage perspective
- Quiesce VMs when taking snapshots
- De-duplication
Deploying Storage as a VM

Provides isolation for multi-tenancy

Performance benefits for some use cases
## ZFS – extraordinary scalability

<table>
<thead>
<tr>
<th>Description</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of data volumes on a system</td>
<td>$2^{64}$</td>
</tr>
<tr>
<td>Maximum size of a data volume</td>
<td>$2^{78}$ bytes</td>
</tr>
<tr>
<td>Number of file systems in a data volume</td>
<td>$2^{64}$</td>
</tr>
<tr>
<td>Maximum size of a file system</td>
<td>$2^{64}$ bytes</td>
</tr>
<tr>
<td>Number of devices in a data volume</td>
<td>$2^{64}$</td>
</tr>
<tr>
<td>Number of files in a directory</td>
<td>$2^{56}$</td>
</tr>
<tr>
<td>Maximum file size</td>
<td>$2^{64}$ bytes</td>
</tr>
<tr>
<td>Number of attributes of a file</td>
<td>$2^{48}$</td>
</tr>
<tr>
<td>Maximum size of any attribute</td>
<td>$2^{64}$ bytes</td>
</tr>
<tr>
<td>Number of snapshots of a file system</td>
<td>$2^{64}$</td>
</tr>
</tbody>
</table>

*Unlimited snapshots with integrated search*
Hybrid storage pools

<table>
<thead>
<tr>
<th></th>
<th>Write optimized device (SSD)</th>
<th>Main Pool</th>
<th>Level 2 ARC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Replacement Cache (ARC)</td>
<td>separate intent log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Write optimized device (SSD)</td>
<td>HDD</td>
<td>Read optimized device (SSD)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Size (GBytes)</th>
<th>1 - 10 GByte</th>
<th>large</th>
<th>big</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>write iops/$</td>
<td>size/$</td>
<td>size/$</td>
</tr>
<tr>
<td>Use</td>
<td>sync writes</td>
<td>persistent storage</td>
<td>read cache</td>
</tr>
<tr>
<td>Performance</td>
<td>low-latency writes</td>
<td>secondary optimization</td>
<td>low-latency reads</td>
</tr>
<tr>
<td>Need more speed?</td>
<td>stripe</td>
<td>more, faster devices</td>
<td>stripe</td>
</tr>
</tbody>
</table>
Storage architecture

Minimal
- SSD
- HDD

Good
- SSD
- HDD (mirror, raidz)

Better
- SSD (mirror)
- HDD (raidz2, raidz3)

Best
- SSD (mirror, stripe)
- SSD (mirror, stripe)
- SSD (stripe)
Storage architecture contd.
Network architecture
Replication for backup

- Traditional Backup
  - NDMP
- Auto-Tier
  - rsync
- Auto-Sync
  - ZFS send/receive
- Auto-CDP
  - AVS (SNDR)
- Mirror
- Application Level Replication

Recovery Point Objective

- Days
- Hours
- Seconds

System I/O Performance

- Slower
- Faster
RAID-Z – comparison Conceptually to standard RAID

- RAID-Z has 3 redundancy levels:
  - RAID-Z1 – Single parity
    - Withstand loss of 1 drive per zDev
    - Minimum of 3 drives
  - RAID-Z2 – Double parity
    - Withstand loss of 2 drives per zDev
    - Minimum of 5 drives
  - RAID-Z3 – Triple parity
    - Withstand loss of 3 drives per zDev
    - Minimum of 8 drives
  - Recommended to keep the number of disks per RAID-Z group to no more than 9

Source: http://www.slideshare.net/OpenStorageSummit/oss-kevin-halgren-washburn-univ-10488421
Zmirror

- Zmirror – conceptually similar to standard mirroring.
- Can have multiple mirror copies of data, no practical limit (3way mirror, 4way mirror, ...)
  - E.g. Data+Mirror+Mirror+Mirror+Mirror...
  - Beyond 3-way mirror, data integrity improvements are insignificant
- Mirrors maintain block-level checksums and copies of metadata. Like RAID-Z, Zmirrors are self-correcting and self-healing (ZFS).
- Resilvering is only done against active data, speeding recovery

Source: http://www.slideshare.net/OpenStorageSummit/oss-kevin-halgren-washburn-univ-10488421
CERN study

- Write and verify 1 GB data file
  - Write 1 MB, sleep 1s, etc.. repeat until 1 GB
  - Read 1 MB, verify, sleep 1s, ...

- On 3000 servers with HW RAID card

- After 3 weeks:
  - **152** cases of silent data corruption

- HW RAID only detected “noisy” data errors

- Need end-to-end verification to catch silent data corruption

Source: J Bonwick, B. Moore, Sun: ZFS: the last word if file systems
Self-Healing Data in ZFS

1. Application issues a read. ZFS mirror tries the first disk. Checksum reveals that the block is corrupt on disk.

2. ZFS tries the second disk. Checksum indicates that the block is good.

3. ZFS returns known good data to the application and repairs the damaged block.

Source: J Bonwick, B. Moore, Sun: ZFS: the last word if file systems
Performance

- Performance ZFS software RAID roughly equivalent in performance to traditional hardware RAID solutions

- RAIDZ performance in software is comparable to dedicated hardware RAID controller performance

- RAIDZ will have slower IOPS than RAID5/6 in very large arrays, there are maximum disks per vDev recommendations for RAIDZ levels because of this

- As with conventional RAID, Zmirror provides better performance I/O and throughput than RAIDZ with parity

Source: http://www.slideshare.net/OpenStorageSummit/oss-kevin-halgren-washburn-univ-10488421
Use case: VMworld 2011
WMworld 2011

- Approached February 2011 with a design for a commodity white box solution to compete with Vblock and vPod.
- Designed to run High Density Generic Cloud Workloads as a trial within VMware
- Cost per TB was primary design consideration but good IOP’s performance was important
- Architected around Commodity X86 Hardware with Nexenta’s technology partners
- Nexenta HA design and scale out for future with NFS referalls

Source: Nexenta
VMWaaS Super Rack

Super Rack Overview
- Commodity x86 Hardware
- 2 x 48U Racks
- 120 Compute Servers
- 8 Management Servers
- 1,440 Compute Cores
- 11.25 TB Compute Memory
- 486TB Usable Storage
- 10GB Ethernet Network

Individual Blade configuration:
- Dual Intel® Xeon
- 2 X5650 6-core processor 5600 series 95W processor wattage @ 2.66GHz
- Intel® X5500 chipset
- Netlist: 96GB 1066mhz DDR3 ECC Registered Memory
- Intel® 82576 Dual-Port Gigabit Ethernet LAN
- Integrated Matrox G200eW graphics
- Integrated IPMI 2.0 with remote KVM and virtual media

VM per Core Consolidation Ratios
Dependent on VM workload:
- 4 VMs per Core = 5,700 VMs
- 6 VMs per Core = 8,640 VMs
- 8 VMs per Core = 11,520 VMs
- 10 VMs per Core = 14,400 VMs
- 12 VMs per Core = 17,280 VMs

Arista Networks 7204
- Designed for real-world deployments of 40Gb and 100GbE
- Data Center Bridging - lossless Ethernet transport
- Energy Efficiency
- Less than 10 watts per wirespeed 10GbE port
- Gold Climate Saver power supplies
- Front-to-rear cooling and HVAC
- 192 wirespeed 1/10Gb Ethernet port options
- Up to 5.76Bpps L2 and L3 switching and routing
- 2.3GB packet buffer per linecard module
- 4.5usec port to port latency

System Capabilities and Architecture
- Up to 10 Terabit lossless switch fabric
- 64GB/s bandwidth per linecard Slot (1.25 Terabit full-duplex)
- Low-latency, lossless, VOQ fabric
- Data Center Class airflow and resiliency

Nexentastor Enterprise Storage
- NexentaStor provides enterprise class unified storage capabilities via a software solution that ends vendor lock-in while delivering superior storage management functionality with a particular focus on virtualized environments.

Nexentastor Capacity
- 360 Drives
- 720TB Raw Capacity
- 480TB Usable Storage

Source: Nexenta
Details...

- Super Rack, inclusive of NexentaStor license: $325,000 list
- Super Rack, w/ NexentaStor, powered over 50% of labs

Source: Nexenta
Nexenta Storage Module at VMworld 2011

- Nexenta Layout
  - multiple pools (active/active cluster)
  - 2x data pools

- Per Pool
  - vdev layout is 6x RAID Z2
  - 30x vdevs per pool
  - 2x SLOG devices (STEC ZeusRAM mirrored)

- Totals (Rack)
  - 360 drives, 720TB raw, 480TB usable

Source: Nexenta
Hands-on-Lab Workload

Some highlights

- Labs:
  - 26x distinct labs
  - 4x to 25x VMs per lab
  - max VM size was 26GB
  - create, deploy, destroy per login (on demand)
  - Lab choice random (student choice), no pre-pop possible!

- nested VMs workload, highly latency sensitive

Takeaways:

- 148,103 VM’s created during Vegas show
- 1x VM created every 1.215 seconds

Source: Nexenta
Nexenta Statistics

- Ran 4 of 8 VMware verticals in Vegas
- 10.3 billion NFS IOPs served
  - 7.9 billion in Vegas, 2.4 billion in Copenhagen
- 3 billion NFS IOPs from one head in Vegas
- Peak controller load
  - 154,000x 4K NFS ops/sec at **sub 1 ms latency**
  - 38,590x 16K NFS ops/sec on a single controller

Source: Nexenta
Nexenta Statistics cont.

- Highest Bandwidth (single head, 16K average I/O)
  - 1 305 MB/sec total
  - 928MB/sec read
  - 376MB/sec write
  - ... less than 2ms latency throughout above!

Source: Nexenta
Nexenta operational issues

- DRAM failure
  - DRAM failure in one head triggered HA failover
  - partner head ran the workload of “both” for 6hrs until evening maintenance window
  - NS called out failed DIMM serial number
  - DIMM replaced head back in service 12mins

- High Availability RSF-1 HA plugin worked flawlessly
  - VMware saw no loss of service
  - monitoring informed NOC before they attributed issue
  - head over provisioning in design meant solution didn’t glitch with extra workload

Source: Nexenta
Use case - University of Ljubljana: Nexenta as a private cloud storage

... or how to build a good cloud storage for Virtual Computing Laboratory with minimal budget...
Building a private cloud at UL FRI

- Need for VCL (Virtual Computing Laboratory)
  - LaaS – Laboratory as a Service
- We already used 11 standard x86 Intel quad core servers with local 7200 rpm SATA drives
  - At the start of semester we deployed all needed VMs for students up-front (700 VMs)
  - No flexibility, “fixed” configuration of VMs

- Identity management project for students of UL and FRI
- Bologna reform: student mobility, old and new programs overlap
- Wanted to support other courses
- We needed to support off-campus users (voluntary after school activities on high schools – popularization of computer science)
Platform choice for our cloud solution

- Private cloud
  - In the future: we already have proof of concept provisioning modules for hybrid/public cloud integration...
- Apache VCL – Virtual Computing Laboratory
  - NCSU has a production deployment for almost 10 years
  - It was ideal mix of features for “pedagogical” usage
  - Good vmWare support
  - Open Source – we can do customization
    - Support for complex network topologies needed
Our VCL

- Questions
- Excercise
- Grade
- Assesment
- Solution
- Response
- LMS
- Teacher
Our VCL
Requirements for data storage HW

- Low budget
  - SATA disks, standard HW, standard controllers, ...
- Must survive spikes in usage (students tend to work on assignments 3 hours before deadline on Sunday ...;))
- At least 330 concurrent users
- This was our stress test – automatically provision 330 VMs
- Must survive parallel booting of 75 x 3 VMs over NFS storage
- Did I mention low budget?
Choosing SW

- Requirements:
  - Thin provisioning, overprovisioning, flexibility, easy administration and configuration, snapshots, quotas, ...
    - Typical enterprise requirements
  - NFS, iSCSI, CIFS, WebDAV...
  - Infiniband (native or IPoIB)
  - Low price, preferably open source

- We looked at:
  - Linuxes (and Linux based appliances)
  - BSD
  - Oracle(former Sun) Solaris/OpenSolaris
  - Nexenta (Opensolaris core + Debian userland system)
Choosing SW - Nexenta

- We liked ZFS features, but we lacked Solaris administration proficiency
- Nexenta offered best mix of required features:
  - ZFS
  - Hybrid storage pools – automatic SSD integration in storage pools as read and/or write cache
  - Web GUI, easy to use CLI, Open API (no deep knowledge of Solaris needed)
  - Good NFS implementation
  - Tuned and optimized for data storage
  - Low price, low maintenance costs
    - Free Nexenta Community Edition (up to 16 TB w/o enterprise plugins)!
Performance tests

- We used two common benchmarks (installed onto Nexenta appliance – local benchmarks, no network involved)
  - bonnie++ 1.4 (r8)
  - Iozone 1.1 (r1)
- 3 configurations evaluated (ordinary low cost 0,5 TB SATA 7200RPM!):
  - 3 x 7 disk Raid-Z2, SSD ZIL
  - 3 x 7 disk Raid-Z, SSD ZIL
  - 11 x 2 disk mirror, SSD ZIL
- Nexenta configuration:
  - Sys_zfs_nocacheflush = Yes
  - Sys_zil_disable = No
- Low cost HW:
  - SuperMicro server, 16 GB DDR3 ECC, 24x3.5” SATA/SAS disk enclosure, Intel Xeon X3440 2.5 GHz, Intel X25-E SSD
  - 24 x 0,5 TB WD 7200 HDD
- Infiniband 10G SDR for networking
### bonnie++

- **3 x 7 disk Raid-Z2, SSD ZIL, 32 GB file**

<table>
<thead>
<tr>
<th>WRITE</th>
<th>CPU</th>
<th>RE-WRITE</th>
<th>CPU</th>
<th>READ</th>
<th>CPU</th>
<th>RND-SEEKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>261MB/s</td>
<td>39%</td>
<td>133MB/s</td>
<td>20%</td>
<td>324MB/s</td>
<td>28%</td>
<td>254/sec</td>
</tr>
<tr>
<td>273MB/s</td>
<td>44%</td>
<td>137MB/s</td>
<td>21%</td>
<td>333MB/s</td>
<td>28%</td>
<td>248/sec</td>
</tr>
<tr>
<td>534MB/s</td>
<td>41%</td>
<td>270MB/s</td>
<td>20%</td>
<td>657MB/s</td>
<td>28%</td>
<td>251/sec</td>
</tr>
</tbody>
</table>

- **3 x 7 disk Raid-Z, SSD ZIL, 32 GB file**

<table>
<thead>
<tr>
<th>WRITE</th>
<th>CPU</th>
<th>RE-WRITE</th>
<th>CPU</th>
<th>READ</th>
<th>CPU</th>
<th>RND-SEEKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>305MB/s</td>
<td>40%</td>
<td>153MB/s</td>
<td>23%</td>
<td>337MB/s</td>
<td>27%</td>
<td>234/sec</td>
</tr>
<tr>
<td>317MB/s</td>
<td>41%</td>
<td>154MB/s</td>
<td>23%</td>
<td>335MB/s</td>
<td>28%</td>
<td>230/sec</td>
</tr>
<tr>
<td>623MB/s</td>
<td>40%</td>
<td>307MB/s</td>
<td>23%</td>
<td>673MB/s</td>
<td>27%</td>
<td>232/sec</td>
</tr>
</tbody>
</table>

- **11 x 2 disk mirrors, SSD ZIL, 32 GB file**

<table>
<thead>
<tr>
<th>WRITE</th>
<th>CPU</th>
<th>RE-WRITE</th>
<th>CPU</th>
<th>READ</th>
<th>CPU</th>
<th>RND-SEEKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>231MB/s</td>
<td>29%</td>
<td>140MB/s</td>
<td>22%</td>
<td>447MB/s</td>
<td>38%</td>
<td>576/sec</td>
</tr>
<tr>
<td>223MB/s</td>
<td>29%</td>
<td>136MB/s</td>
<td>22%</td>
<td>443MB/s</td>
<td>37%</td>
<td>569/sec</td>
</tr>
<tr>
<td>454MB/s</td>
<td>29%</td>
<td>276MB/s</td>
<td>22%</td>
<td>891MB/s</td>
<td>37%</td>
<td>573/sec</td>
</tr>
</tbody>
</table>
iozone

pool: generating 32750MB files, using 32 blocks
Version $Revision: 3.308 $
Compiled for 32 bit mode.
Build: Solaris10gcc
Include fsync in write timing
Include close in write timing
Record Size 32 KB
Command line used: iozone -ec -r 32 -s 32750m -l 2 -i 0 -i 1 -i 8
Output is in Kbytes/sec
Time Resolution = 0.000001 seconds.
Processor cache size set to 1024 Kbytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.
Min process = 2
Max process = 2
Throughput test with 2 processes
Each process writes a 33536000 Kbyte file in 32 Kbyte records
...
* Output truncated for clarity
<table>
<thead>
<tr>
<th>Workload</th>
<th>3 x 7 disk Raid-Z2, SSD ZIL (KB/s)</th>
<th>3 x 7 disk Raid-Z, SSD ZIL (KB/s)</th>
<th>11 x 2 disk mirror, SSD ZIL (KB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 initial writers</td>
<td>480,300.72</td>
<td>602,779.72</td>
<td>432,561.00</td>
</tr>
<tr>
<td>Avg throughput per process</td>
<td>240,150.36</td>
<td>301,389.86</td>
<td>216,280.50</td>
</tr>
<tr>
<td>2 rewriters</td>
<td>158,897.89</td>
<td>176,138.99</td>
<td>94,610.09</td>
</tr>
<tr>
<td>Avg throughput per process</td>
<td>79,448.95</td>
<td>88,069.50</td>
<td>47,305.04</td>
</tr>
<tr>
<td>2 readers</td>
<td>667,758.81</td>
<td>711,554.84</td>
<td>924,458.06</td>
</tr>
<tr>
<td>Avg throughput per process</td>
<td>333,879.41</td>
<td>355,777.42</td>
<td>462,229.03</td>
</tr>
<tr>
<td>2 re-readers</td>
<td>674,348.28</td>
<td>720,884.53</td>
<td>930,028.28</td>
</tr>
<tr>
<td>Avg throughput per process</td>
<td>337,174.14</td>
<td>360,442.27</td>
<td>465,014.14</td>
</tr>
<tr>
<td>2 mixed workload</td>
<td>6,353.50</td>
<td>6,352.45</td>
<td>8,627.94</td>
</tr>
<tr>
<td>Avg throughput per process</td>
<td>3,176.75</td>
<td>3,176.22</td>
<td>4,313.97</td>
</tr>
</tbody>
</table>
Performance inside a virtual machine

- Windows 7 on vmWare ESXi 4, Crystal Disk Mark 3.0, Infiniband IPoIB 10 GBs, VMDK via NFS on Nexenta
UL FRI case study summary

- We implemented 8x2 mirror configuration (8TB license)
  - 4 x hot spare
  - thin provisioning and linked clones enabled us to work with pretty small disk space requirements
  - VMs favor small random R/W
- Real workload experiences
  - It works quite good, with peak loads and all
  - Up until now we successfully completed almost 15,000 reservations!
Infiniband support

- We’ve been running our production cloud on NFS over IPoIB for more than 2 years without bigger issues
  - But IPoIB takes heavy toll on performance – we eagerly wait for native IB support...
- However, IB not officially supported
  - Official support announced for NexentaStor 4.0 due in 2Q2012
  - Nexenta’s technicians resolved our tickets (configuration issues) nevertheless ;)
- IB **much cheaper** than 10G Ethernet
  - especially older 10G SDR and 20 G DDR chipsets from Mellanox...
  - Our IB cards were cheaper than Intel 2x1GB Ethernet...

We think that IB has a potential as a 10G Ethernet replacement in certain situations
Cheap SATA disks: “Desktop” or “RAID edition”?  

- Based on our experiences with standard disk shelves with RAID5 configurations, we bought “RAID edition” disks  
- Not needed! We could save ½ of our budget for disks (and bought a better SSD).  
- With Nexenta (and ZFS) you can use cheaper “desktop” models
Discussion and questions...
Thank you!

Additional info:
matjaz.pancur (at) fri.uni-lj.si

Prices and other commercial stuff for Nexenta: CHS d.o.o. - [www.chs.si](http://www.chs.si) - Nexenta’s partner
prodaja@chs.si