



**Peter Waterman**

Senior Manager of Technology and Innovation, Managed Hosting  
Blackboard Inc

# Blackboard Managed Hosting<sup>(sm)</sup>

Blackboard Inc. is a world leader in e-Education software - our online learning application, the Blackboard Learning System, is the most widely-adopted course management system among U.S. postsecondary institutions.

Blackboard Managed Hosting is a business unit within Blackboard which provides fully managed hosting services for all Blackboard software suites to a global client base of over 700 institutions. Our service includes management of every layer from the server metal to software administration and internet presence.

Our clients demand cost effective reliability and performance – we require efficiency and agility to provide it.

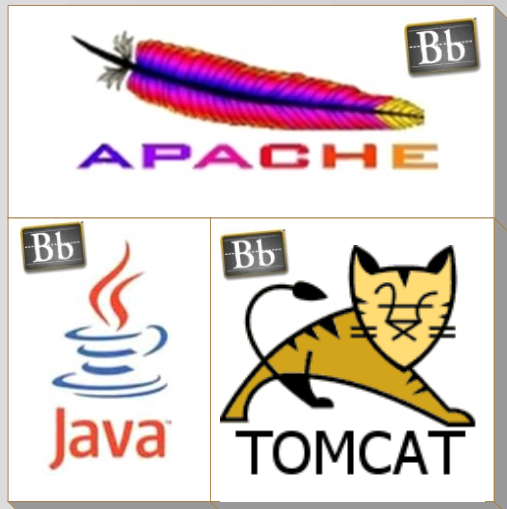
# Blackboard Managed Hosting at a Glance

- **Six datacenter facilities** on three continents
- **1.2+ petabytes** of NetApp enterprise storage
- **2200+ Dell servers** (including 1400+ blades)
- **30+ million users** across all installations
- **990+ Oracle databases**
- **1350+ application servers** providing web content
- **350+ million hits** per day across all installations
- **15+ terabytes** of data transferred daily



# Our Typical Installation (mostly single tenant)

## Hosted Application Stack



**ORACLE®**



**NFS**



**NetApp™**  
Go further, faster

# Rebuilding the Box

In late 2006, Blackboard Managed Hosting recognized that our yearly growth projections were so extreme that we would need to perform a core redesign of our methodology towards the following goals:

- Reduce complexity of system management
- Massively speed up backup and recovery
- Provide near-instant deployment and capacity expansion
- Improve system performance
- Reduce licensing and infrastructure cost using virtualization
- Avoid locking into a single current hypervisor

# Blackboard Advanced Hosting Platform

After extensive testing, a new hosting platform was developed using the following core technology:

- **Linux nfsroot:** moves the OS to NetApp filers and allows migration between physical and virtual (P2V/V2P/V2V)
- **Oracle over NFS:** moves the databases to NetApp filers, increasing performance and simplifying backup/recovery
- **Virtualization (RHEL5 Xen):** allows effective performance upgrades for smaller clients and reduces future infrastructure and licensing costs
- **NetApp Flexible Volumes:** provides near-instant provisioning, recovery, cloning, and backups of NFS data
- **Scalent V/OE:** enterprise software for managing network booting, physical hardware, and virtual machines

# Decisions and Controversy

Why “nfsroot” when talking about “NFS root partitions?”

It just looks cooler...

**NFSroot**

**NFSROOT**

**Nfsroot**

**nfsroot**

... you decide.

# All Hail nfsroot!

nfsroot is one of the simplest, coolest, and most underrated Linux technologies to work with. It opens up many possibilities:

- Move the entire OS to NFS, instead of local disk or SAN
- Add appropriate drivers/kernels to the nfsroot and it can be booted anywhere – even flipping between paravirt Xen kernels, paravirt VMI kernels, and standard kernels!
- Incredibly simple “thin provisioning” – no need to resize the filesystem when growing the nfsroot
- Modify/repair/troubleshoot an nfsroot with the OS offline without needing to boot into rescue mode
- Centralize data analysis, tracking, modification, and deployment of nfsrooted systems
- Backup the nfsroot with zero system overhead

# How nfsroot Works

Assuming you have a prepared nfsroot mount, the following is a simple overview of how nfsroot works:

2. Server (physical or virtual) powers on with Preboot eXecution Environment (PXE) enabled

3. PXE system compares booting server against its internal list of hardware to determine nfsroot location and boot kernel for the appropriate instance

4. PXE system passes a special boot kernel to the Server including the network configuration it should use (or DHCP), the specific kernel to use, as well as the location of the nfsroot

5. Server uses provided kernel to attach to the network and mount the nfsroot

6. Server continues to boot into appropriate run level

# Before You Drink the Kool Aid...

There are some downsides to nfsroot, especially on a large scale:

- Requires a “flat” network or excellent network virtualization
- Requires solid enterprise storage (10gbit ethernet recommended)
- Managing PXE (to mount nfsroot on boot) gets complicated without good software or script-fu
- Linux may completely hang when experiencing network issues (but does recover well)
- Linux quirks include swap files (must be mounted as loop devices over NFS) and /dev/random (no entropy without disks)
- Haven't tested nfsroot with VMware or Xen's live migration (yet!)

• **Network must be secure!** NFS traffic is unencrypted!

# Behind the nfsroot Curtain



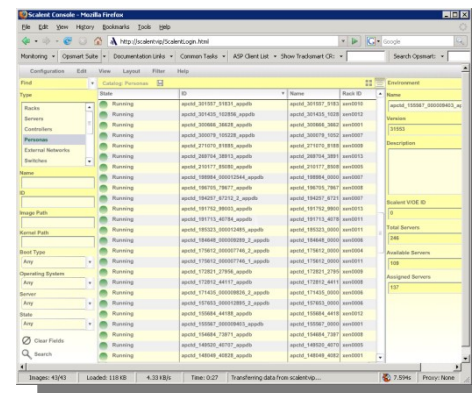
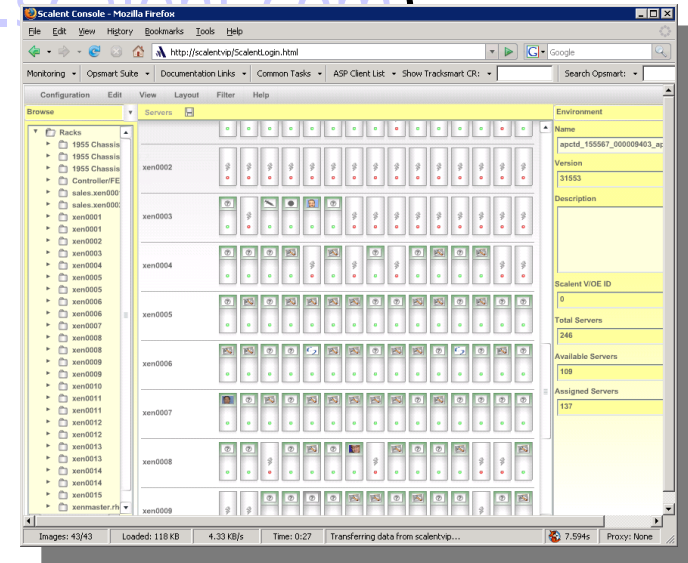
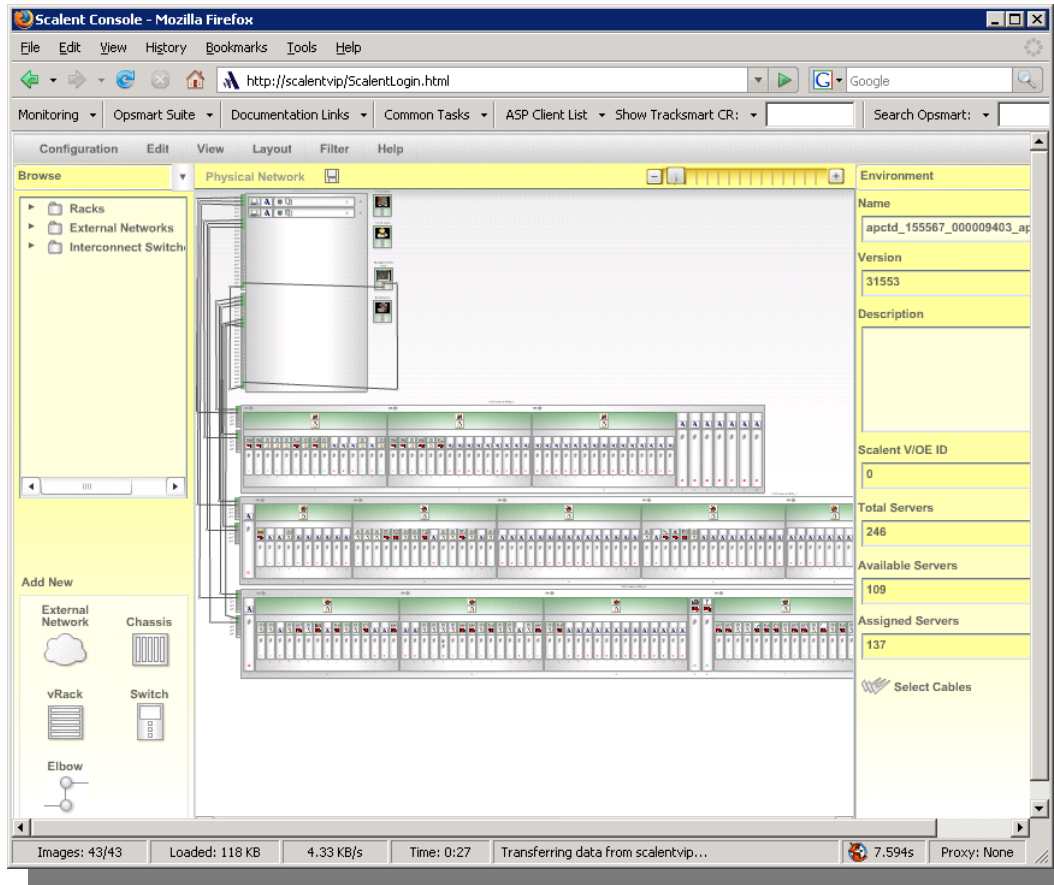
NetApp™  
Go further, faster

At Blackboard Managed Hosting, we now deploy completely diskless blade systems. For our nfsroot and all storage needs (including Oracle over NFS) we use NetApp FAS3070 filer clusters with the following benefits:

- Redundant 10gbit ethernet attached to a dual channel 10gb network distro layer (1gbit access layer)
- SnapShot allows instant on-line point-in-time backups (inconsistent) with no system impact
- FlexClone technology allows instant copy-on-write volume copies for low footprint provisioning, clones and recoveries
- Powerful caching algorithms reduce disk IO (97-99% cache hits in our environments)
- SnapMirror provides site-to-site data replication for business continuity purposes

# Managing nfsroot on the Enterprise Scale

After extensive testing and review, Blackboard Managed Hosting has deployed and recommends the Scalent Virtual Operating Environment (<http://www.scalent.com>)



# Scalent Virtual Operating Enterprise (V/OE) SCALENT™

Scalent V/OE is a large scale datacenter management suite which includes the following features (amongst many others):

- Virtualization management including interaction with hypervisors and virtual machines
- Fully managed failure recovery, status, and transition across hardware (virtual or physical)
- Complete OS level network configuration management (IP addressing, routing, etc.)
- Network virtualization and management, allowing transition of systems across non-flat networks
- Power management via IPMI or Dell DRAC including powering off unused servers and powering on newly activated systems
- iSCSI/SAN management (people use that?!)

# nfsroot is Supported in RHEL5!

There is a lot of great information on doing small scale nfsroot out there, googling “nfsroot howto” is a great start.

Red Hat 5 contains nfsroot support as a technology preview, while RHEL5.1 is fully supported.

The RHEL5.1 release notes have basic instructions on this:

<http://www.redhat.com/docs/manuals/enterprise/RHEL-5-mar>

You can also use normal linux kernel techniques to do nfsroot on earlier versions of RHEL as long as you are willing to roll your own boot kernel with nfsroot support (google will help you do this).

# Small Scale nfsroot – Do It Yourself

In a nutshell, to start you need:

- A dhcp/pxelinux/tftpboot server
- An NFS server with rsync'd copies of a good install (make sure it has any kernel/drivers you might need for different hardware/virt)

To get deeper, you need:

- A mechanism for tracking MAC addresses and assigning nfsroots based on these
- A mechanism for limited rw nfsroot exports to appropriate IP's

# Small Scale nfsroot “Best Practices”

The following guidelines should be followed for managing nfsroot:

- Track the MAC addresses of all physical and virtual machines in your environment along with basic hardware information about these machines
- Ideal kernel for a machine should be tracked with the MAC
- Assign a permanent IP address to each nfsroot, which will be handed out over DHCP
- Export your nfsroots rw to the permanent IP address only
- Dynamically configure the DHCP server to hand out the appropriate IP address and nfsroot information based on the MAC address of the server it should be booted on at this moment

It's actually relatively simple in small scale!

# Super Small Scale nfsroot Example

If we have three different machines (1 physical and 2 virtual) to work with, we might create a table like this:

Description	MAC	Attributes	Kernel
2950 physical	00:01:a1:e2:bc:da	Dual core 16gb ram	2.6.9-42.ELsmp
Esx vm001	00:01:ef:bd:ca:d1	1 cpu 2GB RAM	2.6.9-42.ELsmpVMI
Xen vm001	00:02:de:f2:a9:e8	2 cpu 4GB RAM	2.6.9-42.Xen

To determine where your nfsroot boots and which kernel it uses:

7. Modify your dhcpd.conf to give the permanent IP address to the MAC from the table corresponding to the target

8. Create a pxelinux.cfg file named after the corresponding MAC address

9. Add the MAC-specific Kernel option into the pxelinux.cfg file

# Random Technical Tidbits

How you do a lot of this stuff is up to you. Some of settings we use as a result of our testing are as follows:

**Mount Options for nfsroot:** rw,lock,proto=udp,remount

**Swapfile over NFS via loop:**

4. Create sparse swapfile (/.swapfile)

5. Add to /etc/rc3.d/S99local or similar:

```
losetup /dev/loop0 /.swapfile
```

```
swapon /dev/loop0
```

**Entropy for /dev/random, add to /etc/rc.3d/S99local:**

```
rngd -b -r /dev/urandom
```

# Random Technical Tidbits II

**Readability #1:** Modify /etc/hosts and /etc/fstab to use a single letter for your NFS server's name due to length.

Example:

```
192.168.22.11:/vol/nfsroot01
```

```
a:/vol/nfsroot01
```

**Readability #2:** Alias "df" to "df -P" which forces output on one line. For example:

```
# df /
```

```
Filesystem          1K-blocks      Used Available Use% Mounted on
s:/vol/nfsroot/os01
                    16777216      5008064  11769152  30% /
```

```
# df -P /
```

```
Filesystem          1024-blocks      Used Available Capacity Mounted on
s:/vol/nfsroot/os01 16777216      5008064  11769152      30% /
```

# Quick Recap

- nfsroot allows Linux to boot anywhere anytime without any messy P2V/V2V/etc. transitions
- NetApp filers are pretty neat
- Scalent V/OE is awesome – this presentation only covers about 5% of its featureset, come talk to me to find out more
- Designing and building an environment centered around Linux is good

**A word of warning:** The first time you demo automated failure recovery or clicking a button 20 times to nearly instantly stand up 20 clones of an environment or 2-3 clicks to restore a database from three weeks ago, disbelief is followed by crying :(

# Questions / etc.?

We are always happy to demo this technology and share methods and best practices with other enterprises, and I'm always happy to chat with anyone – drop me a note to set something up: [pete.waterman@blackboard.com](mailto:pete.waterman@blackboard.com)

Scalent can be seen in action at the vendor area at the Red Hat Summit!