

Scale-out NAS with NFS Referrals and pNFS

Dmitry Yusupov
Nexenta Systems, Inc

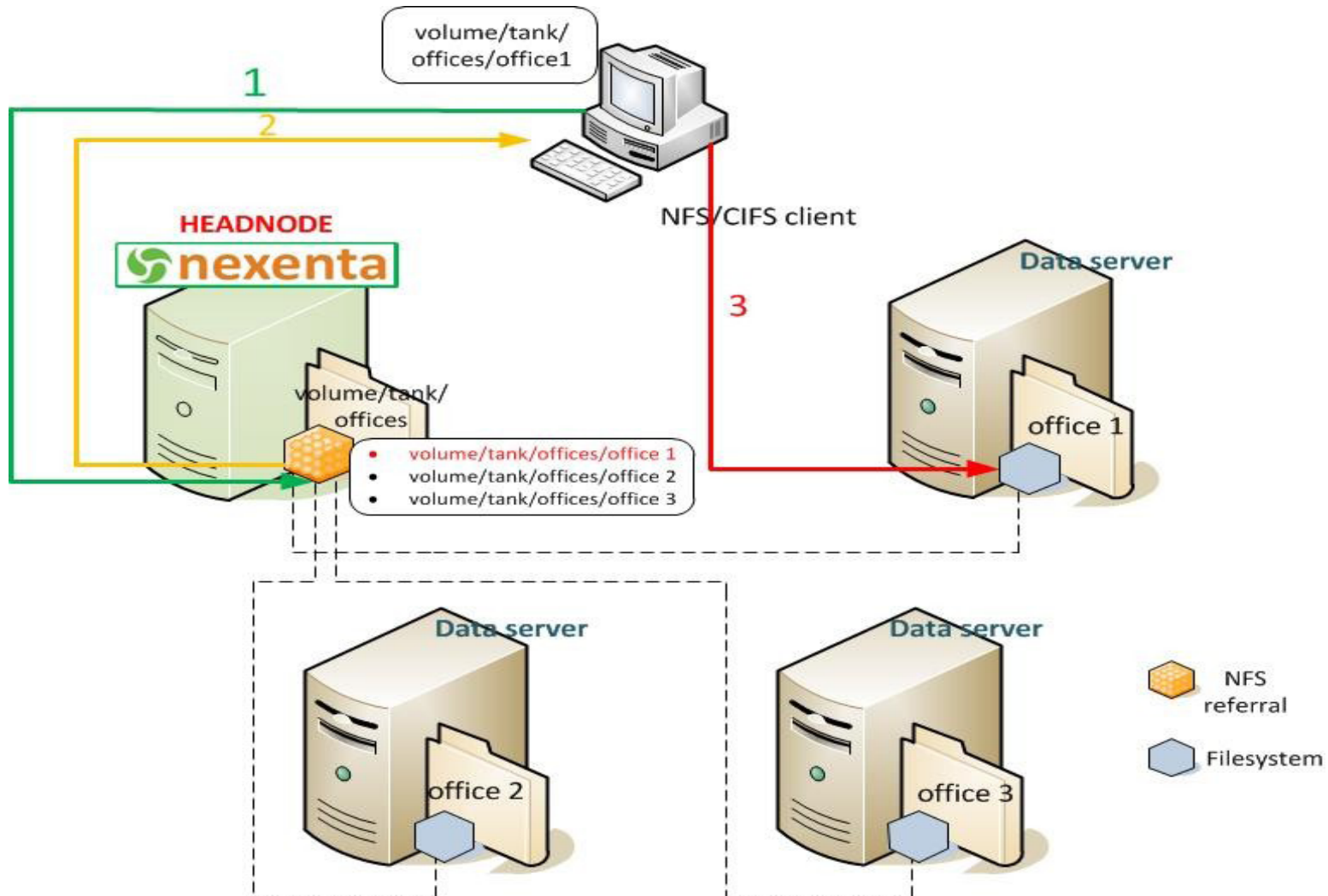
History: Performance upgrades in NFS

- ❑ Invented by Bill Joy at Sun back in 1983
- ❑ NFSv2 base line protocol, 1989
- ❑ NFSv3 added asynchronous I/O
- ❑ NFSv4.0 added delegations
- ❑ **NFSv4.0 adds NFS referrals**
- ❑ NFS over RDMA transport
- ❑ **NFSv4.1 adds parallel data access**

Scalable NAS: Overview

- ❑ Uses industry standard NFSv4.0 referrals
- ❑ Uses industry standard NFSv4.1 parallel architecture
- ❑ Combines NFS referrals with parallel data access
- ❑ Presents total solution as a single name space
- ❑ Focus on NFS file access only
- ❑ Single point of management – Web GUI
- ❑ Enterprise features: live migration, replication, etc

NameSpace Cluster: Overview



NameSpace Cluster: Features

- Supported protocols: NFSv4, **CIFS/DFS**
- Full manageability, single point of management
- Built-in HA capability
- Views: administrative and logical
- Fault Management
- Performance Management and DTrace integration
- Capacity Management
- Integrated Migration and Replication, Live Migration
- Ease of Use

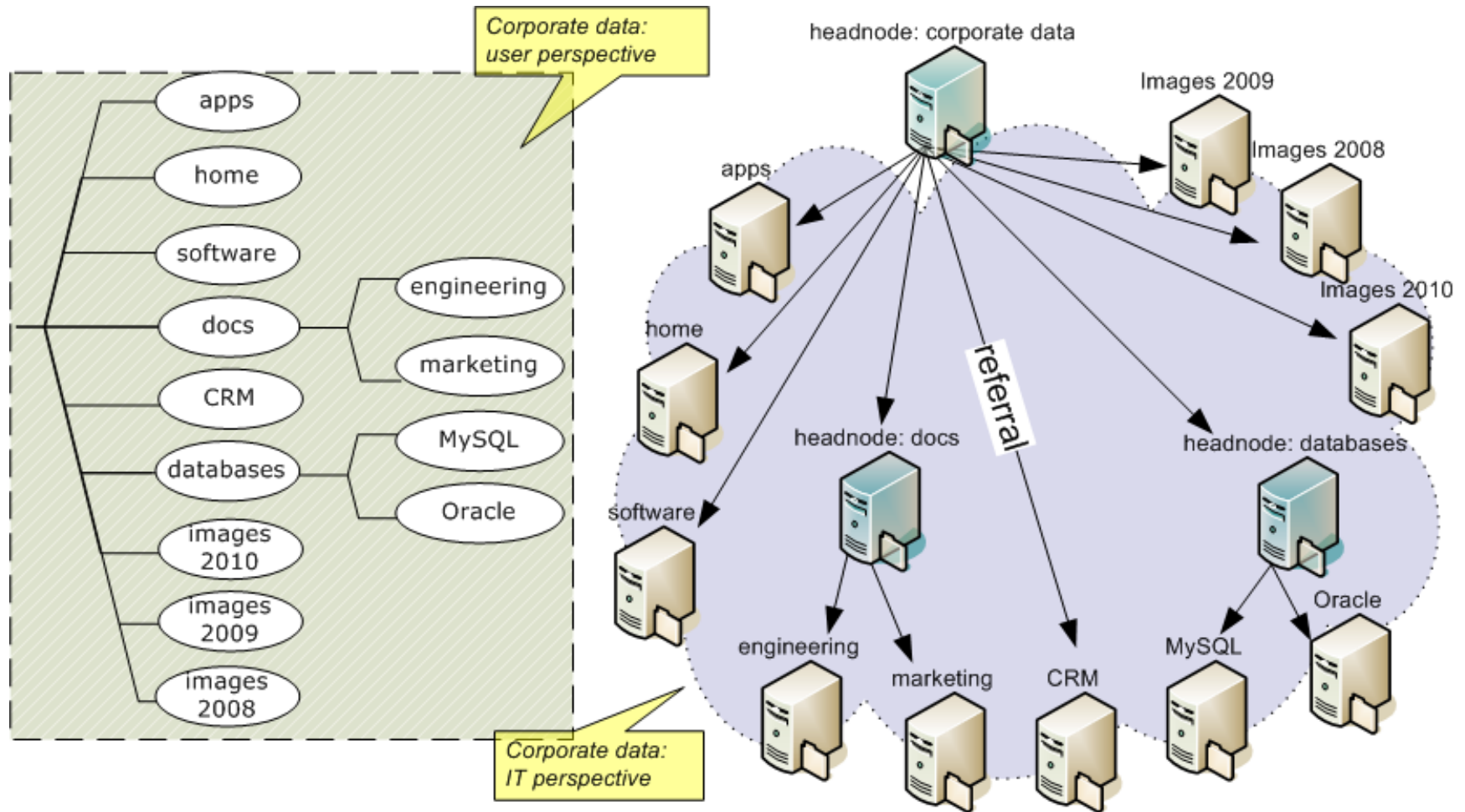
NameSpace Cluster: User Interface

The screenshot displays the NameSpace Cluster User Interface with the following components:

- Navigation Bar:** Status, Settings, Data Management, Analytics, NFS Cluster. Includes Console, View log, Jobs, and nscluster1.
- Namespace Tree (Left):** A hierarchical view of the storage structure, including Corporate data, hd1.office1.company.com, data/corp, docs, databases, and hd2.office1.company.com.
- Capacity usage (Top Right):** A pie chart showing storage utilization:
 - 86.60% (2980.91M) - Free
 - 13.40% (461.09M) - Used
- Namespace Tree (Middle):** A detailed view of the Corporate data namespace, showing sub-directories like marketing, engineering, databases, mysql, and oracle.
- I/O per second (Right):** A line graph showing I/O activity over time, with a peak at 25:17. The Y-axis ranges from 0 to 140.
- adnodes Table (Bottom Right):**

Read	Write	Read latency ma	Write latency ma
0	0	0	0
0	0	0	0
0	0	0	0
- Job Viewer (Bottom):** Run jobs: 0

NameSpace Cluster: Logical View

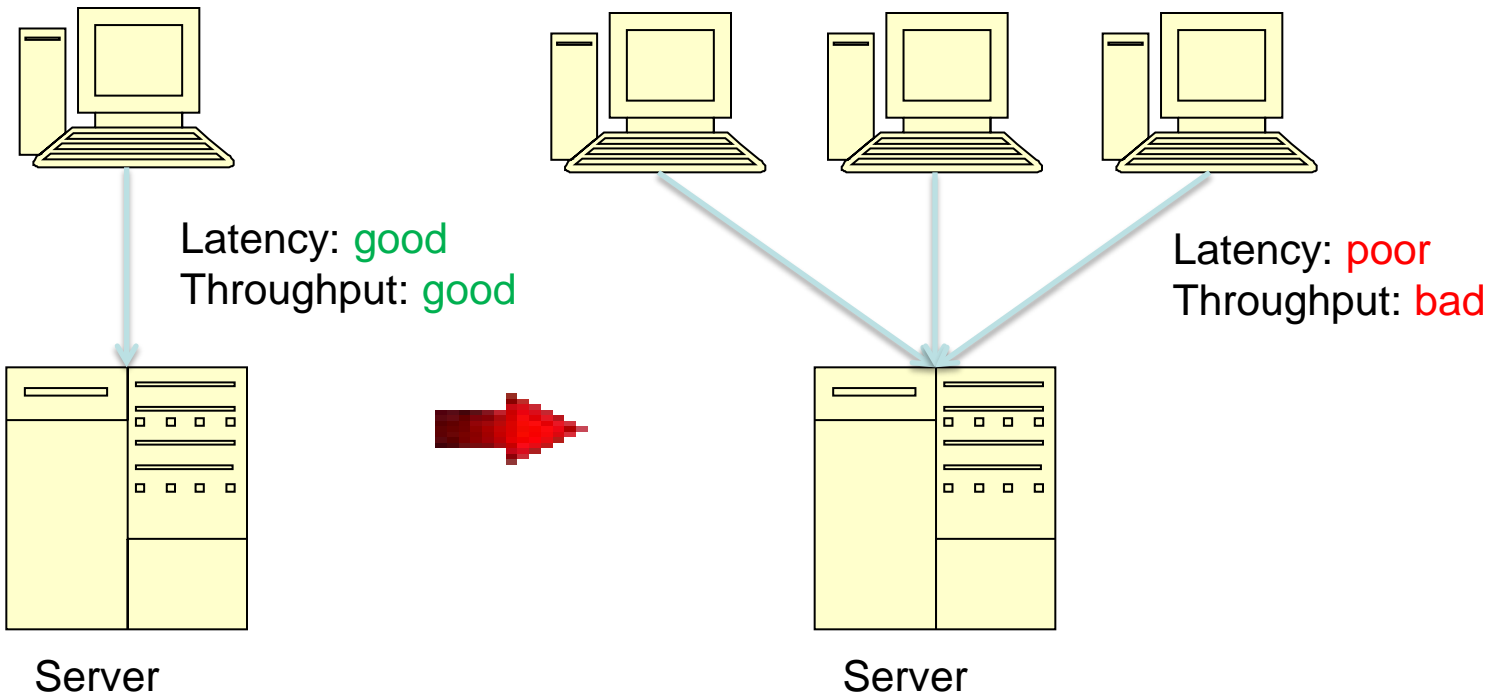


NameSpace Cluster: parallel scale

- ❑ With NFSv4.0 referrals NameSpace Cluster can scale at the level of subdirectories in the shared tree hierarchy
- ❑ NameSpace Cluster can combine NFSv4.1 nodes and enable NFSv4.1 clients to transparently access parallel clustered structure
- ❑ With NFSv4.1 NameSpace Cluster can scale at the level of files

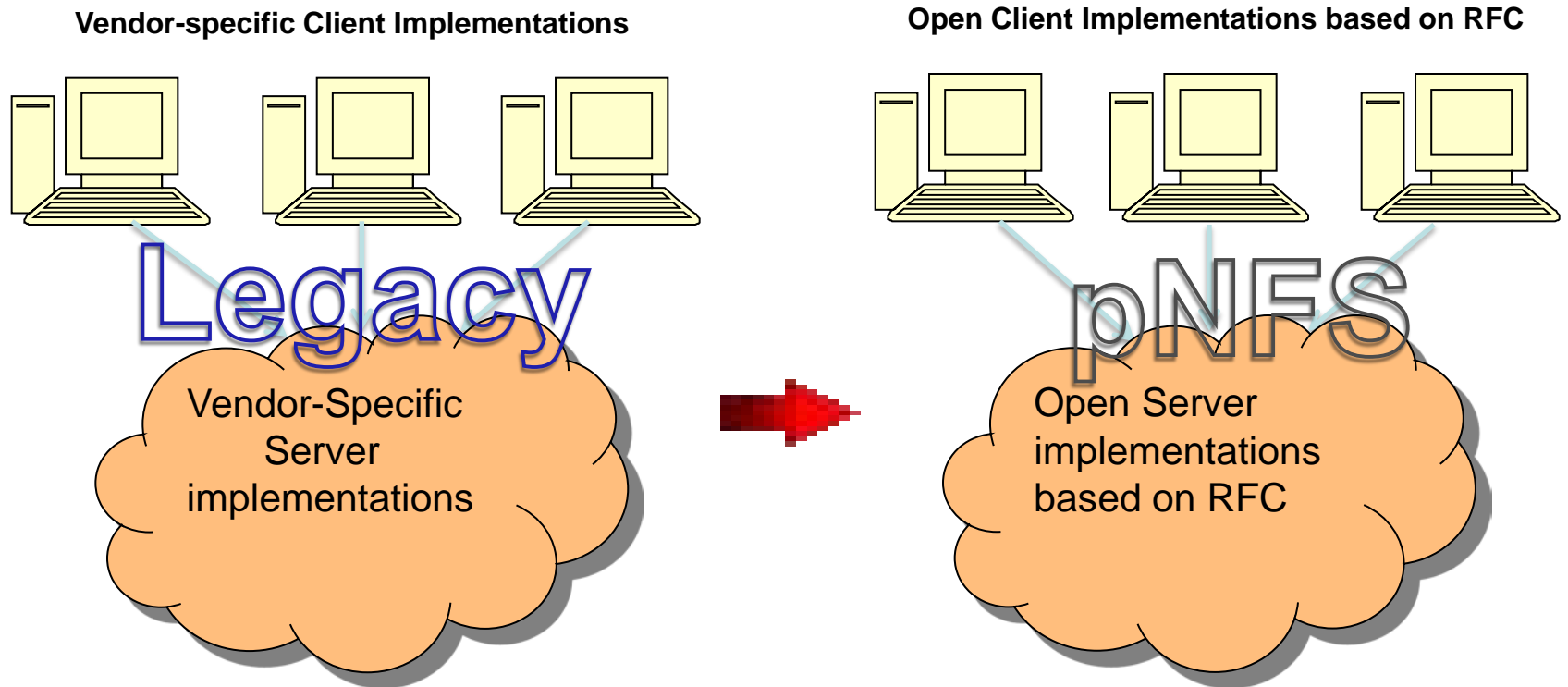
The need for pNFS? – Part I

- Single server bottleneck is a common issue
- Parallel data access difficult to implement



The need for pNFS? – Part 2

- Widespread adoption
- Heterogeneous system support



NFSv4.1 main functionality

- ❑ NFS sessions:
 - ❑ Exactly-Once semantics
 - ❑ Trunking
- ❑ Parallel NFS:
 - ❑ the striping of regular files across several data servers
 - ❑ Types of data servers:
 - ❑ Blocks-based, where the pNFS client accesses data via FC or iSCSI
 - ❑ Object Storage-based, where the pNFS client accesses data via the OSD protocol
 - ❑ File-based, where the pNFS client accesses data via the NFSv4.1 protocol

- ❑ Management problems:
 - ❑ Security:
 - ❑ DS cloud access and authorization
 - ❑ Human error protection
 - ❑ Manageability:
 - ❑ Easy way to manage large number of MDS and DS machines
 - ❑ Layout control
 - ❑ Visibility:
 - ❑ Extended statistics, error counts
 - ❑ Extended diagnostics and snooping support

- ❑ SPE – Simple Policy Engine
 - ❑ a way to determine file layouts at creation
 - ❑ a way to accept or reject DS additions
 - ❑ SPE is implementation specific extension for pNFS
 - ❑ SPE is using XDR/RPC spec as a transport
- ❑ SPE concepts:
 - ❑ a new concept “Network Pools” – **/etc/npools.spe**
 - ❑ a new concept “Policies” – **/etc/policies.spe**

pNFS server: NexentaStor Summary

- ❑ Heavily based on OpenSolaris implementation
- ❑ Interoperability demoed at 06/09 Bakeathon
- ❑ Implementation is based on a standard
- ❑ Policy-driven (SPE) management
- ❑ Utilizing ZFS capabilities
- ❑ Supports pNFS-over-RDMA transport

pNFS server: nfsstat and snoop

- ❑ nfsstat enhancements:
 - ❑ `nfsstat -s`
 - ❑ `nfsstat -c -v 4l`
- ❑ snoop enhancements:
 - ❑ Understands NFSv4.1 semantics
 - ❑ Output can be exported to Wireshark for further analysis

- ❑ pNFS benefits from use of ZFS
 - ❑ Provides a concept of pooled storage
 - ❑ Transactional object system
 - ❑ End to end data integrity
 - ❑ Highly scalable
- ❑ MDS implemented on top of ZFS POSIX layer
- ❑ DS implemented on top of ZFS DMU

□ DTrace provider for NFSv4

- Works on MDS and DS
- Available in Illumos now, NFSv4.1 will extend it with new features
- Example script:

```
#!/usr/sbin/dtrace -s
nfsv4:::op-read-start
{
    self->t0 = timestamp;
}
nfsv4:::op-read-done
/ args[2]->status == 0 /
{
    rr = (READ4res *) args[2];
    printf("read %d bytes in %d microseconds\n",
        rr->data_len, (timestamp-self->t0)/1000);
}
```

- ❑ **Illumos Kernel integration plan:**
 - ❑ December 2011: merge of pNFS server with Illumos Mercurial gate
 - ❑ February 2012: availability of iCore ISO with pNFS support
- ❑ **Linux Kernel status:**
 - ❑ August 2011: v3.0 pNFS clients for file and block access available
- ❑ **Vendor support and enterprise adoption:**
 - ❑ 2012/Q1 time frame: RHEL6 minor revisions may have it integrated
 - ❑ 2012/Q2: NexentaStor 4 minor revisions may have it integrated

NameSpace Cluster: Availability

- ❑ NexentaStor 3.1 released this summer and required
- ❑ NameSpace Cluster plugin for NexentaStor released this August and available for production use

The End

□ Thank You!