NFS and Commercial Workloads

Darrell Suggs
Stephen Daniel

Overview

- Who We Are and Why We Care
- Our Message, Mission, and Targets
- NFS Performance Escalation – The Customer's View
- NFS Client Performance – What Really Matters
- War Stories From Real Customers
- Why is this happening with NFS in Commercial Workloads
- What we are doing
- What makes an Enterprise NFS Client
- Q&A

NetApp

Overview

- **What is a commercial Workload?**
  - **Large multi-user application**
  - **Ratio of clients to servers is small**
  - **Applications often randomly read and write large files**
- **Examples**
  - **Databases, SAS, large email servers, …**

Who We Are

- **Network Appliance**
  - **Number 1 NAS Storage Vendor – NFS**
- **Darrell Suggs**
  - **Senior Performance Engineer**
  - **Final defense for NFS performance escalations**
- **Steve Daniel**
  - **Technical Director – Database Performance**
  - **Final defense for Database perf escalations**

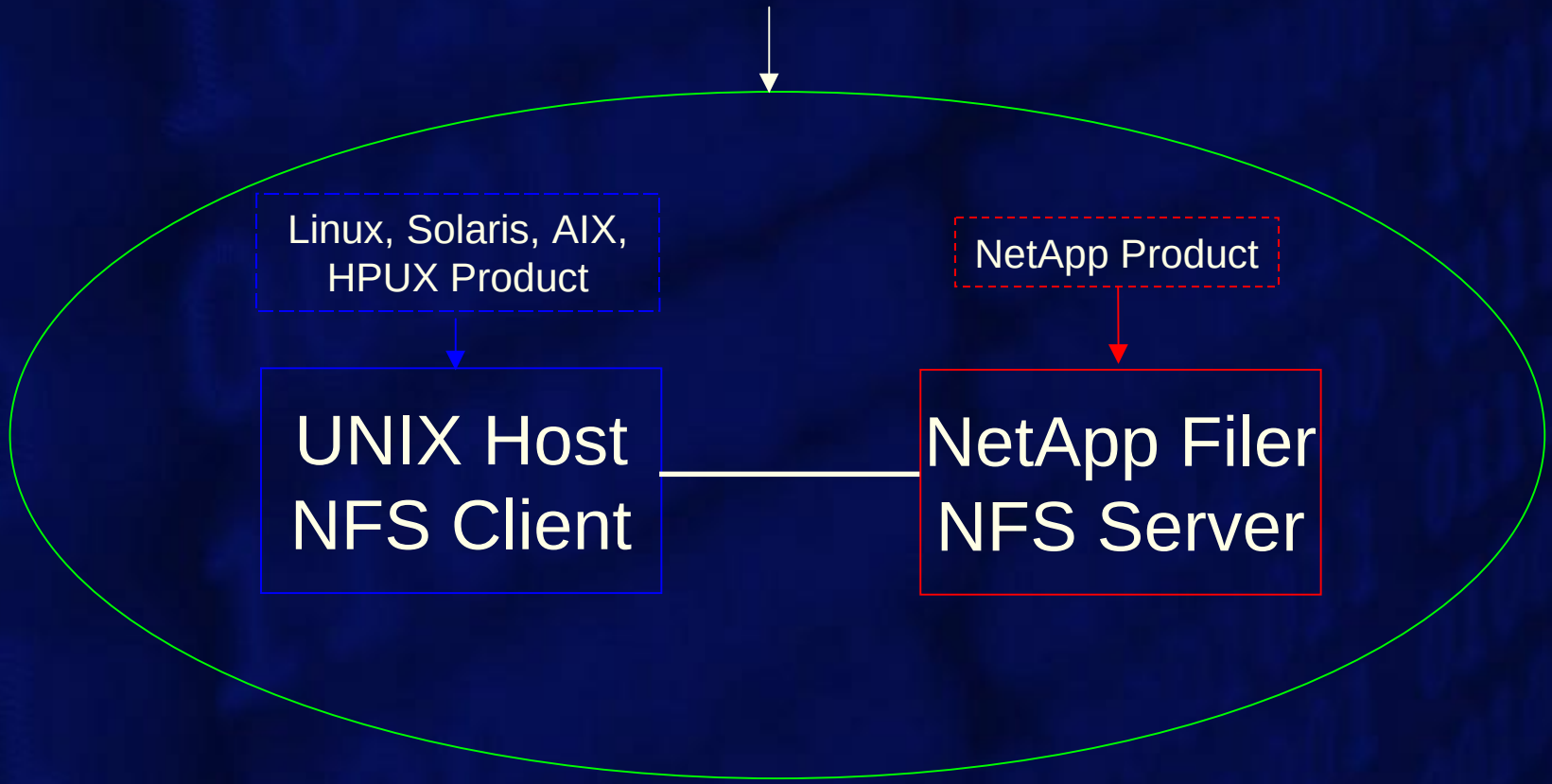
NetApp

Why We Care


Linux, Solaris, AIX,
HPUX Product
UNIX Host
NFS Client
NetApp Product
NetApp Filer
NFS Server

Why We Care

What the Customer Purchases and Deploys
An NFS Solution



Our Message

- **NFS → Delivers real management/cost value**
- **NFS → Core Data Center**
- **NFS → Mission Critical Database Deployments**
- **NFS → Deliver performance of Local FS ???**
- **NFS → Compared directly to Local FS/SAN**

Our Mission

- Support NFS Clients/Vendors
  - We are here to help
- Ensure successful commercial deployments
  - Translate customer problems to actionable plans
- Make NFS as good or better than Local FS
  - This is true under certain circumstances already
- Disseminate NFS performance knowledge
  - Customers, Vendors, Partners, Field, Engineers

NetApp

The Customer's View

- **Typical NFS Performance Escalation**

Performance
-Some metric with Speed X
- TPM or User Latency or Wall clock time

The Customer's View

- **Typical NFS Performance Escalation**

Performance
-Some metric with Speed X
- TPM or User Latency or Wall clock time

Customer Consumes the "NFS Kool-Aid"

Performance
-Speed is now 2 to 10 times WORSE !!!

-Steve and I usually get a phone call…

NFS Client Performance

- **Traditional Wisdom**
  - **NFS is slow due to Host CPU consumption**
  - **Ethernets are slow compared to SANs**
- **Two Key Observations**
  - **Most customers have CPU cycles to spare**
  - **Ethernet is 1 Gbit = 100 MB/s. FC is on 2x**

NFS Client Performance

- **Reality – What really matters**
  - **Caching behavior**
  - **Wire efficiency (application I/O : wire I/O)**
  - **Single mount point parallelism**
  - **Multi-NIC scalability**
  - **Throughput IOPs and MB/s**
  - **Latency (response time)**
  - **Per-IO CPU cost (in relation to Local FS cost)**
  - **Wire speed and Network Performance**

War Stories

- Real situations we've dealt with
- Clients remain Anonymous
  - NFS vendors are our friends
  - Legal issues, yadda, yadda
  - Except for Linux – Fair Game
- So, some examples…

Caching – Weak Cache Consistency

- **Symptom**
  - **Application runs 50x slower on NFS vs Local**
- **Local FS Test**
  - **dd if=/dev/zero of=/local/file bs=1m count=5**
  - **See I/O writes sent to disk**
  - **dd if=/local/file of=/dev/null**
  - **See NO I/O reads sent to disk**
  - **Data was cached in host buffer cache**
- **NFS Test**
  - **dd if=/dev/zero of=/mnt/nfsfile bs=1m count=5**
  - **See I/O writes sent to NFS server**
  - **dd if=/local/file of=/dev/null**
  - **See ALL I/O reads send to disk ?!?**
  - **Data was NOT cached in host buffer cache**

NetApp

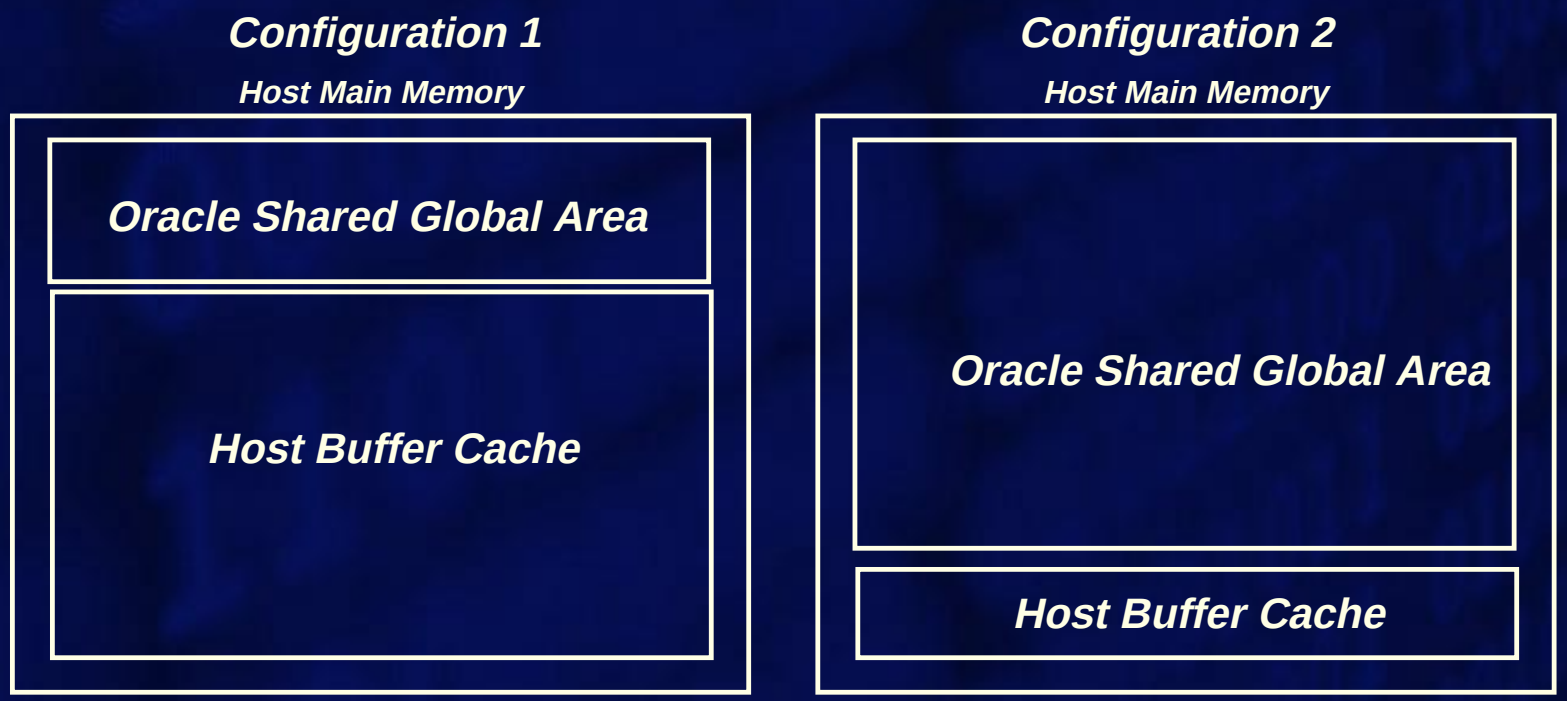
Caching – Weak Cache Consistency

- **Actual Problem**
  - **Threads processing write completions**
  - **Sometimes completed writes out-of-order**
  - **NFS client spoofed by unexpected mtime in post-op attributes**
  - **NFS client cache invalidated because WCC processing believed another client had written the file**
- **Protocol Problem ?**
  - **Out-of-order completions makes WCC very hard**
  - **Requires complex matrix of outstanding requests**
- **Resolution**
  - **Revert to V2 caching semantics (never use mtime)**
- **Customer View**
  - **Application runs 50x faster (all data lived in cache)**

NetApp

Oracle SGA

- **Consider the Oracle SGA paradigm**
  - **Basically an Application I/O Buffer Cache**

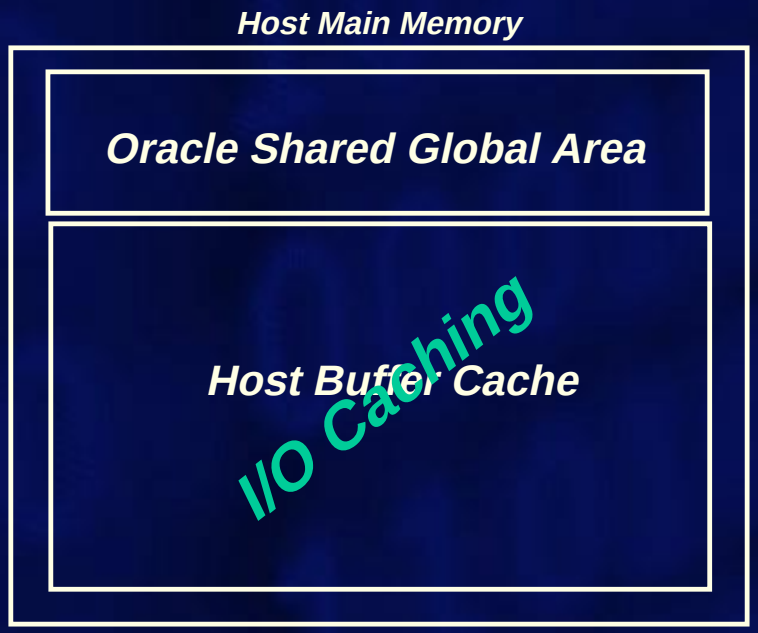


- **Common w/32 bit Arch**
- **Or Multiple DB instances**
- **Common w/64 bit Arch**
- **Or Small Memory Setups**

NetApp

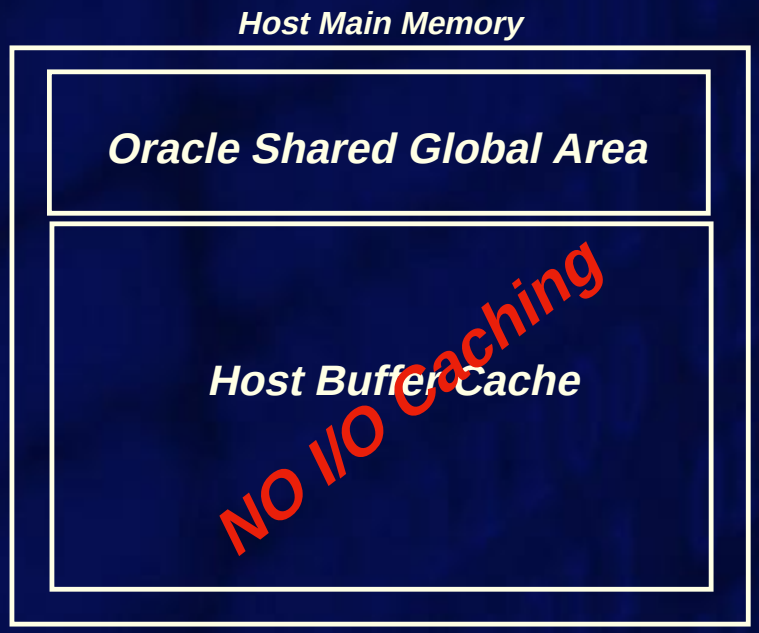
Oracle SGA – The "Cache" Escalation

- **With Local FS**



- **Very Little Physical I/O**
- **Application sees LOW latency**

- **With NFS**



- **Lots of Physical I/O**
- **Application sees HIGH latency**

File Locks

- Commercial applications use different locking techniques
  - No Locking
  - Small internal byte range locking
  - Lock 0 to End of File
  - Lock 0 to Infinity (as large as file may grow)
- NFS Client behavior
  - Each client behaves differently with each type
  - Sometimes caching is disabled, sometimes not
  - Sometimes prefetch is triggered, sometimes not
  - Some clients have options to control behavior, some don't
- DB Setups differ from Traditional Environment
  - Single host connected via 1 or more dedicated links
  - Multiple host locking is NOT a consideration

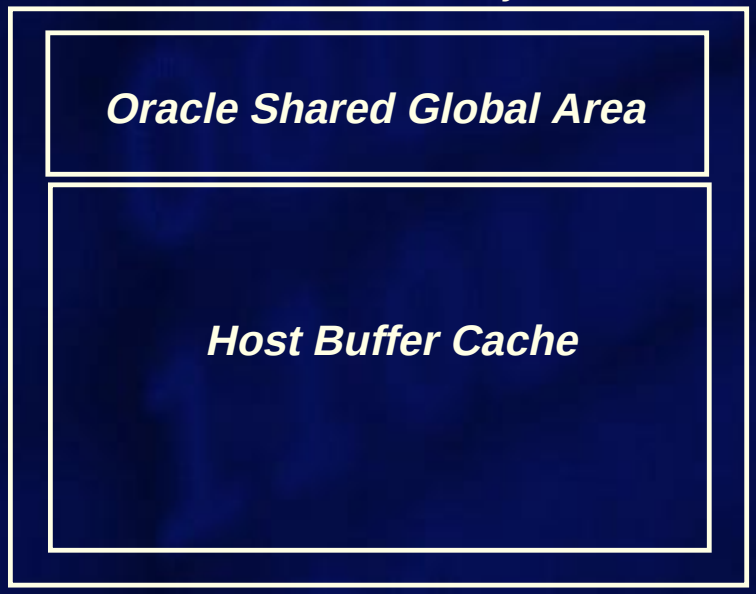
NetApp

File Locks

- **Why does it matter so much?**
  - **Consider the Oracle SGA paradigm again**

Configuration 1

Host Main Memory



- **NOT caching here is deadly**
- **Locks are only relevant locally**

Configuration 2

Host Main Memory

- **Caching here is a waste of resources**
- **Simply want to say "don't bother"**

NetApp

Cache Control Features

- Most of the NFS clients have no “control”
  - Each client should have several “mount” options
    - (1) Turn caching off, period
    - (2) Don’t use locks as a cache invalidation clue
    - (3) Prefetch disabled
- Why are these needed
  - Application needs vary
  - Default NFS behavior usually wrong for DBs
  - System configurations vary

NetApp

Over-Zealous Prefetch

- **Problem as viewed by Customer**
  - **Database on cheesy local disk**
    - **Performance is ok, but need NFS features**
  - **Setup bake-off, Local vs NFS, a DB batch job**
    - **Local results: Runtime X, disks busy**
  - **NFS Results**
    - **Runtime increases to 3X**
- **Why is this?**
    - **NFS server is larger/more expensive**
    - **AND, NFS server resources are SATURATED**
    - **?!? Phone rings…**

Over-Zealous Prefetch

- **Debug by using a simple load generator to emulate DB workload**
- **Workload is 8K transfers, 100% read, random across large file**
- **Consider I/O issued by application vs I/O issued by NFS client**

	Latency	App Ops	NFS 4K ops	NFS 32K ops	4Kops/App Op	32K ops/App op
8K 1 Thread	19.9	9254	21572	0	2.3	0.0
8K 2 Thread	7.9	9314	32388	9855	3.5	1.1
8K 16 Thread	510.6	9906	157690	80019	15.9	8.1

- **NFS Client generating excessive, unneeded prefetch**
- **Resources being consumed needlessly**
- **Client vendor was surprised. Created a patch.**
- **Result: Customer workload faster on NFS than on Local FS**

NetApp

Poor Wire Efficiency – Some Examples

- Some NFS clients artificially limit operation size
  - Limit of 8KB per write on some mount options
- Linux breaks all I/O into page-size chunks
  - If page size < rsize/wsize, I/O requests may be split on the wire
  - If page size > rsize/wsize, operations will be split and serialized
- The Customer View
  - No idea about wire level transfers
  - Only sees that NFS is SLOW compared to Local

RPC Slot Limitation

- **Consider a Linux Setup**
  - **Beefy server, large I/O subsystem, DB workload**
  - **Under heavy I/O load**
    - **Idle Host CPU, Idle NFS server CPU**
    - **Throughput significantly below Wire/NIC capacity**
    - **Customer complains workload takes too long to run**
- **Clues**
  - **Using simple I/O load generator**
  - **Study I/O throughput as concurrency increases**
  - **Result: No increase in throughput past 16 threads**

RPC Slot Limitation

- **Little's Law**
  - **I/O limitation explained by Little's Law**
  - **Latency, concurrency, throughput closely related**
  - **To increase throughput, increase concurrency**
- **Linux NFS Client**
  - **RPC slot table has only 16 slots**
  - **At most 16 outstanding I/O's per mount point, even when there are hundreds of disks behind that mount point**
  - **Artificial Limitation**
- **Customer View**
  - **Linux NFS performance inferior to Local FS**
  - **Must Recompile kernel or wait for fix in future release**

NetApp

Writers Block Readers

- **Symptom**
  - **Throughput on single mount point is poor**
  - **Customer workload extremely slow compared to Local**
  - **No identifiable resource bottleneck**
- **Debug**
  - **Emulate customer workload, study results**
  - **Throughput with only Reads is very high**
  - **Adding a single writer kills throughput**
  - **Discover writers block readers needlessly**
- **Fix**
  - **Vendor simply removed R/W lock when performing direct I/O**

NetApp

Applications Also Have Issues

- Some commercial apps are “two-brained”
  - Use “raw” interface for local storage
  - Use filesystem interface for NFS storage
  - Different code paths have major differences
    - Async I/O
    - Concurrency settings
    - Level of code optimization
- Not an NFS problem, but is a solution inhibitor

Why is this Happening?

- Is NFS a bad solution? Absolutely not!
- NFS began with a specific mission
  - Semi-wide area sharing
  - Home directories and shared data
- Note: problems are NOT with NFS protocol
  - Mostly client implementation issues
- Are the implementations bad? …

Why is this Happening?

- The implementations are NOT bad.
- The Mission has changed!
  - Narrow sharing environment
  - Typically dedicated (often p2p) networks
  - Data sharing → High-speed I/O Interconnect
  - Mission evolved to Mission Critical Workloads
- Actually, NFS has done ok
  - Credit a strong protocol design
  - Credit decent engineering on the implementations

NetApp

Why are things Harder for NFS?

- **What makes Database + NFS different than Local FS?**
  - **For Local Filesystem Caching is simple**
    - **Just do it**
    - **No multi-host coherency issues**
  - **NFS is different**
    - **By default must be concerned about sharing**
    - **Decisions about when to cache/not, prefetch/not**

Why are things Harder for NFS?

- **Database + Filesystem Caching is complex**
  - **Most database deployments are single host (modulo RAC)**
    - **So, cross host coherency not an issue**
    - **However, customers get nervous about relaxing locks**
  - **Databases lock files (many apps don't)**
    - **Causes consternation for caching algorithms**
  - **Databases sometimes manage their own cache (ala Oracle SGA)**
    - **May or may not act in concert with host buffer cache**

What Are We Doing

- **Treading Water**
  - **Working customer escalations**
  - **Developed strong engineering relationships with NFS client vendors**
  - **Uncover bugs, work with vendors, get patches**
  - **Document what we know for customers**
  - **Applying each lesson to other clients/workloads**

What Are We Doing

- **Documenting what we know**
  - **Writing Joint NetApp/Vendor Documentation**
    - **General NFS Performance Tuning**
    - **NFS Database Deployment Guidelines**
  - **E.g.**
    - **“Database NAS Performance: Optimizing Oracle on NFS and Next Generation File Protocols”**
    - **Joint Sun/NetApp Paper by Colaco/Suggs**

What Are We Doing

- **Building a Boat**
  - **Matrix of possible NFS clients is large**
  - **Built a Performance Test environment**
    - **Contains HW from all vendors**
    - **Contains all flavors/releases of each OS (SAN boot)**
  - **Constructing a standard NFS Performance Suite**
    - **Single script bundle**
    - **Runs full spectrum of NFS tests**
    - **Tests all conditions (mount options, caching, etc)**
    - **Can be shared with customers and vendors**
- **So, what to do with all the information…**

What Are We Doing

- **The NFS Scorecard**
  - **Compares each client version/release**
  - **Standard set of results for each client**
  - **Contains**
    - **Actual numbers (e.g. MB/s)**
    - **Check lists (various mount options)**
    - **Behavior grades (caching)**
    - **Each result is rated: Good, bad, ugly**
  - **Can share with Customers and Vendors**
    - **Modulo NDA constraints**

NetApp

What Makes an Enterprise NFS Client

- **We are still figuring this out…**
- **Currently, the Scorecard is divided into categories**
  - **Out-of-box performance**
  - **Mount features**
  - **File system behaviors (caching/locking)**
  - **Wire efficiency (app op : wire op)**
  - **Scaling: concurrency and multi-NIC scaling**
  - **Well-tuned performance**
  - **Suitability for various commercial applications**
  - **Protocol comparisons (UDP vs TCP, V3 vs V4)**

NetApp

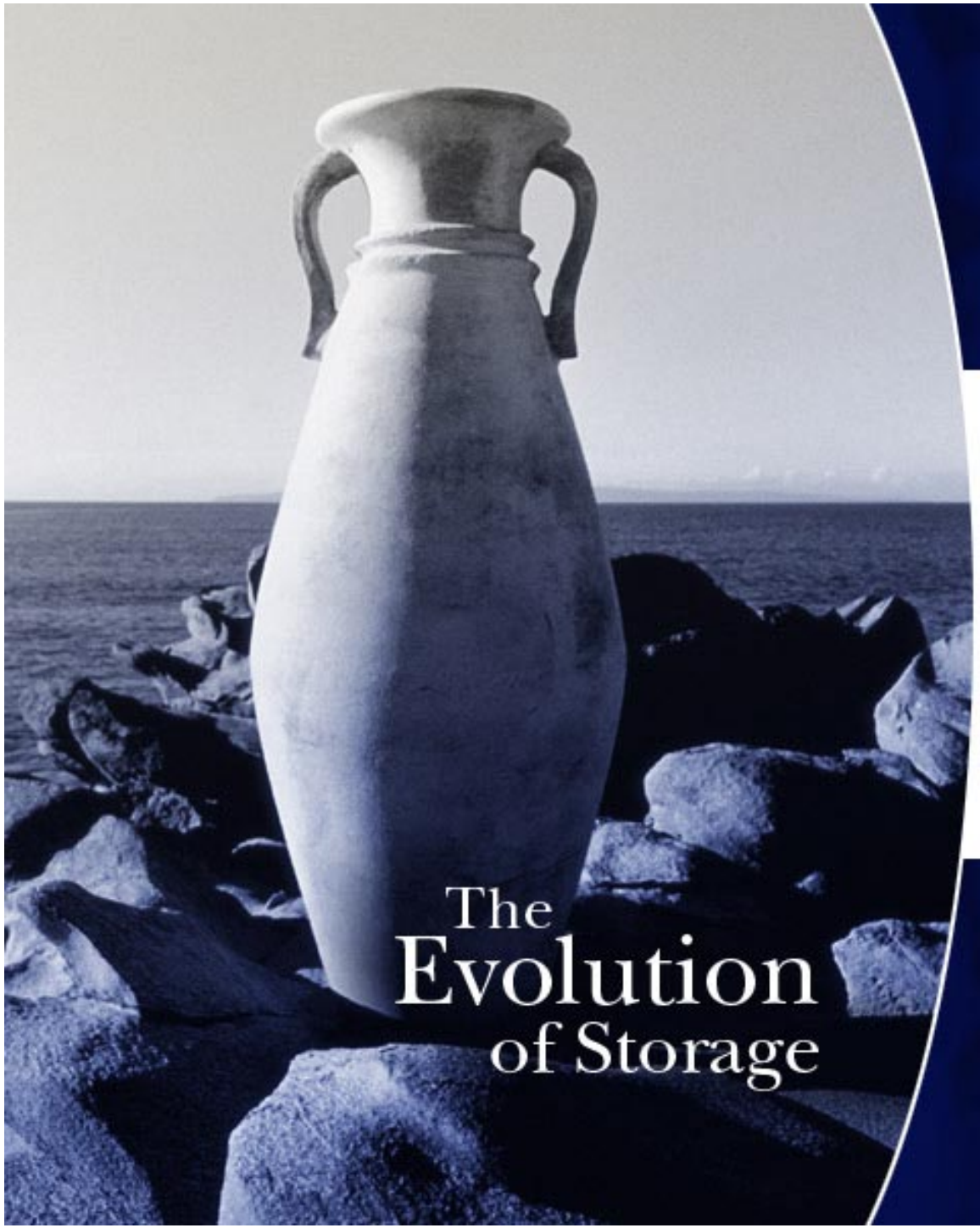
What about NFS evolution

- NFS is evolving
  - v4, RDMA, …
  - But, v3 is today's technology (think "$" now)
- NFS v4
  - Some performance enhancements
    - e.g. Delegations help with caching
  - Reasonable vehicle for general enhancements
- RDMA
  - Addresses higher speed wires
  - Offloads CPU cost and excess memory traffic

NetApp

NFS and Commercial Workloads

Questions and Answers ?



NetApp®