Education

Running Database Applications On NAS: How and Why?

Stephen Daniel, NetApp

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

  NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Abstract

- Running Database Applications On NAS: How and Why?
  - The use of NFS as a storage interconnect protocol for serious enterprise-class databases began around 1995. The number of businesses running production databases over NAS protocols has grown steadily since then. This talk will discuss the benefits and risks of using NFS as a database storage protocol, provide some historical examples of the subtle bugs that NFS clients have had that caused problems for databases, and show some performance measurements comparing NFS-based database performance to database performance using more traditional interconnects such as Fibre-Channel and iSCSI. The talk will conclude with some future directions for this technology.

Database on NAS, in the beginning…

- Brief History
  - 1995: First NAS appliance purchased for DB use
  - 2000: First major ERP-outsourcing shop built on NAS
  - Today: Approx 10% of fabric connected storage for databases uses NAS
- In the beginning there were may who said that databases on NAS was not supportable
  - and, given the state of NAS at that time, they were basically right.

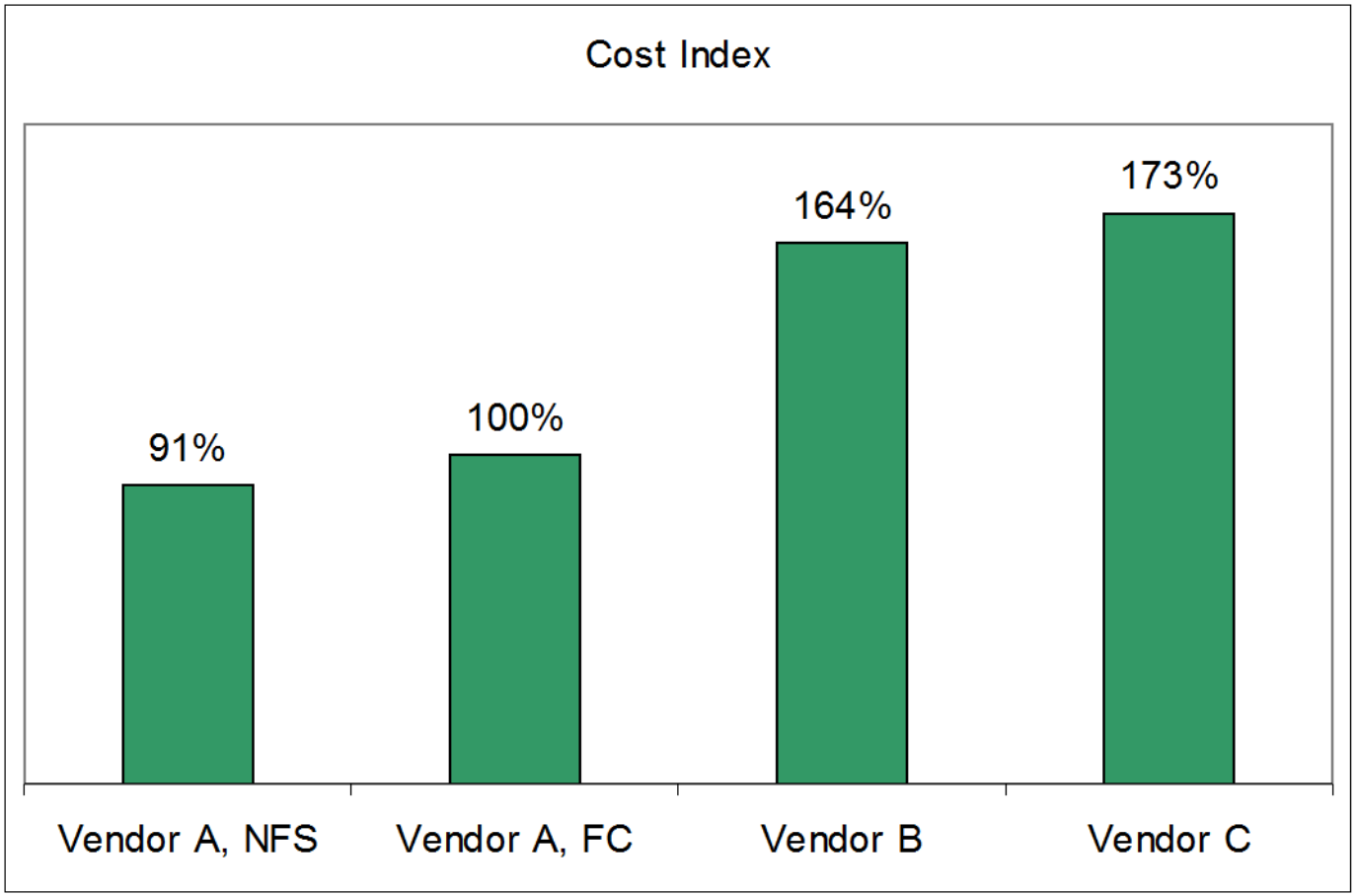
Database on NAS, why?

- 1980's wisdom: run databases on raw disks
- 1990's wisdom: run databases on files
  - Performance of raw not worth the hassle
- Early implementations of DB on NAS were seeking file system features only found in NAS servers
  - Snapshots (point-in-time, consistent logical copies)
  - Inexpensive asynchronous replication

Database on NAS, why?

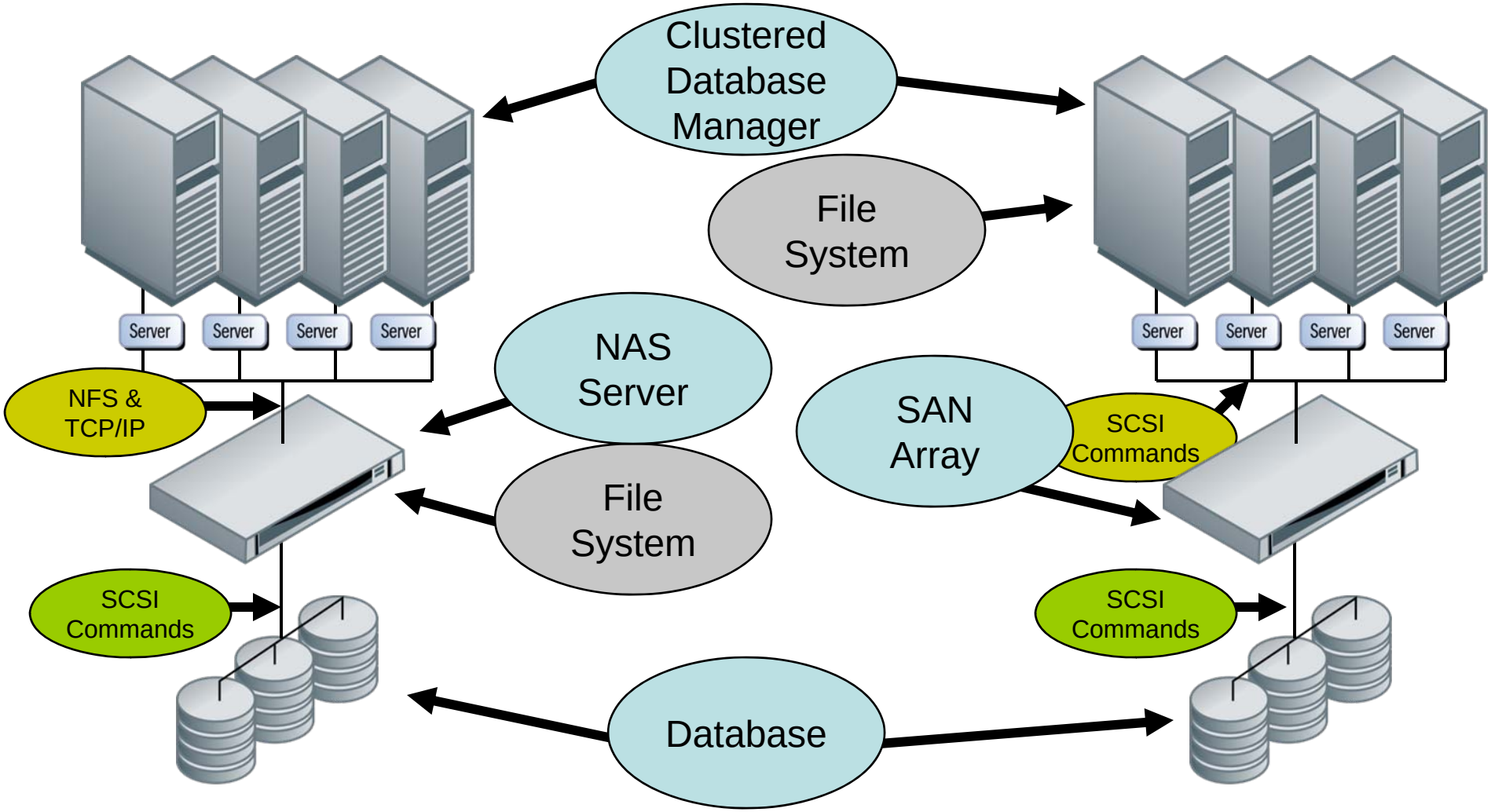
- Top 5 Reasons to run Database on NAS today
  - Easy to manage Ethernet infrastructure
  - Ease of provisioning
  - Very flexible, manageable, binding of object names to storage locations
  - Some NAS protocols are very fault tolerant
  - NAS servers continue to offer advanced features that are missing or harder to use in SAN configurations
- Summary: Lower cost

Cost Advantages



Note: Cost index is the relative total-cost-of-ownership of storage, as measured by an independent market research organization, looking at 4 different, comparable solutions.
See http://media.netapp.com/documents/ar1038.pdf for more details

Block Diagrams: SAN and NAS



Functional Issues

- Write-through caches (you'd think O_SYNC would suffice)
  - VM subsystem locking issues will cause modern Linux kernels to occasionally fail to write through even in the presence of O_SYNC
  - Old UNIX and Linux kernels had issues with confused handling of post-operation attributes that sometimes prevented write-through
- With current kernels
  - Current UNIX systems: No known write-through bugs
  - Linux: must use non-caching mode

Functional Issues

- Non-atomic writes
  - Writes must be atomic to the size of a database block
  - Failure to maintain atomicity causes database recovery problems, possibly including loss of data
  - Be wary of any implementation where the kernel's page size is smaller than the database block size
- With current kernels
  - Current UNIX systems: No known atomicity bugs
  - Linux: must use non-caching mode

Functional Issues

- Delayed duplicates
  - At least one major database repeatedly writes the same block with updated information
  - Failure to detect delayed duplicates causes corruption
  - Failures were found when client-reconnect happened on a new port and the server used source-port as part of the touple that detects duplicates

Functional Issues

- Non-issue: locking
  - Concern:
    - NFS file locking implementation are not suitable for high-performance distributed lock management
    - Neither high performance nor robust
  - However,
    - No current enterprise-class database uses file-locking system calls to lock pages or rows in the database
    - All such databases use the same locking code on all databases (SAN, NAS, DAS)
  - Historical note: Cobol ISAM implementations are a different story…

Functional Issues

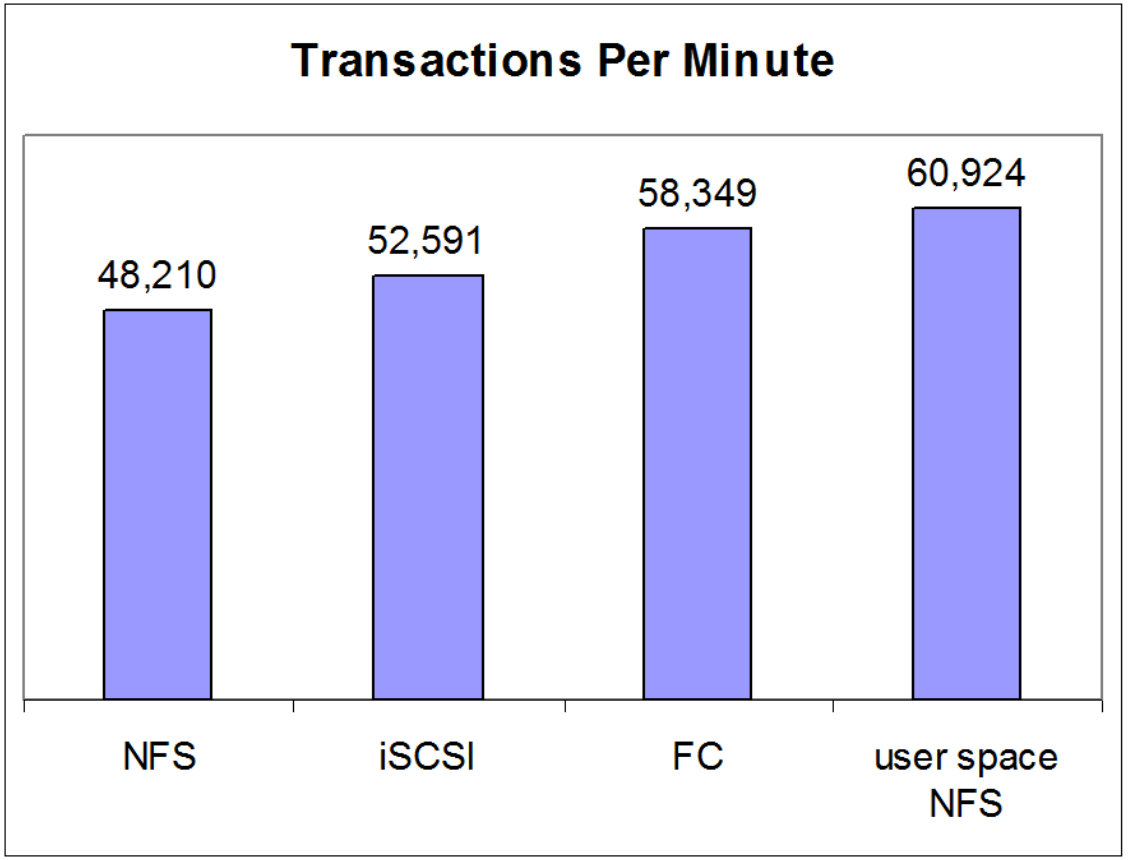
- Protocols that don't work for DB on NAS:
  - CIFS (SMB 1.0) – Windows' file sharing protocol
    - Stateful sessions are not handled well
    - Other issues are plentiful, and never fully debugged
    - Beginning to see some DB on NFS on Windows
    - However, Microsoft's SMB 2.0 is much more robust, so this may change in the future
  - NFS/UDP (as opposed to NFS/TCP):
    - Insufficient sequence number space in UDP
    - Corruption due to delayed duplicate fragments has been observed in production

Summary of functional requirements

- Caches must be write-through
- Shared-disk databases require no read cache and no attribute cache
- Writes must be atomic
- Robust delayed-duplicate detection

Performance – OLTP

- Comparing host-limited OLTP performance, user-space NFS vs other protocols
- FC & iSCSI are raw, no Linux FS
- Data from a modern Linux kernel



- Note: User space NFS is faster because the implementation is optimized for scalable, non-caching use.
  - Pre-publication data. Contact author for more details.

Future Directions

- NFS v4
  - Caching
  - pNFS
  - Other extensions
- Security

Check out SNIA Tutorial:

pNFS, Parallel Storage for Grid and Enterprise Computing

Q&A / Feedback

- Please send any questions or comments on this presentation to SNIA: trackapplications@snia.org

Many thanks to the following individuals for their contributions to this tutorial.

- SNIA Education Committee

John Elliott
Dan Morgan
Saad Jafri