

ON THE TECHNICAL SUITABILITY OF HOSTING DATABASES ON NETWORK APPLIANCE™ STORAGE SYSTEMS

Designed for Correctness

Stephen Daniel | Director, Database Platform and Performance Technology

Jeff Kimmel | Technical Director and Architect

January 2, 2006

TECHNICAL REPORT

Network Appliance, a pioneer and industry leader in data storage technology, helps organizations understand and meet complex technical challenges with advanced storage solutions and global data management strategies.

Abstract

Databases guarantee to never lose committed transactions, even when service is interrupted by hardware or software failures. The algorithms used by databases to ensure no loss of data place substantial constraints on the operation of any storage system hosting the database.

This report discusses the requirements imposed by databases and discusses how Network Appliance storage systems meet or exceed these requirements.

1) Executive Summary

Network Appliance storage systems use a variety of technologies to ensure correct database operation and recovery in the face of hardware, software, or environmental failures. The WAFL® storage virtualization system takes responsibility for ensuring data consistency and availability. The WAFL system uses redundant battery-backed non-volatile memory for write-ahead logging of all commitments to storage, ensuring the integrity and atomicity of all updates to storage, including both system meta-data and user data.

2) Requirements

Database systems use a variety of algorithms to ensure the integrity and recoverability of all transactional databases. Decades of algorithm development have produced increasingly complex database algorithms; however, all of these algorithms are based on a fundamental and simple contract between the database software and the storage stack. This contract is expressed as requirements on the durability, atomicity, and serialized nature of reads and writes. Essentially, all storage systems are required to mimic the behavior of a single, idealized disk.

Some of the requirements are hard requirements. These are absolute. Failure to meet the hard requirements risks data corruption and loss of data in the event of routine system outages.

Soft requirements are optional. When a system fails without violating the soft requirements the database will recover normally. If the soft requirements are not met the database may require recovery from back-up media. However, even in the event of media recovery, no committed transactions will be lost.

Durability

Hard Requirements

1. Any data written by a system call that returns without error, or which becomes visible to a read system call, must have first been committed to stable storage. Such data must be retained by the storage system, regardless of any system or environmental failures.

Atomicity

Hard Requirements

1. Any write of any 512-byte unit of data aligned on a 512-byte boundary must be atomic. That is, if the system fails after a write has begun but before it has finished, each aligned 512-byte block within the write must either be completely present on disk or completely absent.
2. Any write that fails after partial completion must be detectable by the database upon a subsequent read. A block that becomes partially written is known as a fractured block.
 - a. Many databases are able to detect any fracture that occurs on a 512-byte boundary.
 - b. Some databases employ weaker algorithms to detect fractured blocks. Typically these algorithms can detect a fracture only if the block header and trailer are inconsistent. For such databases each database block write must either be committed in no more than two atomic fragments, or by a logically ordered sequence of atomic fragments (i.e. committed in either ascending or descending logical order).

Soft Requirements

3. All writes should be atomic with respect to any given database block. In the event of a host or process failure, any database block write which was outstanding should either be completely committed to stable storage or completely absent from stable storage.

* Note: Oracle®, when operated with block-checksums disabled, is an example of a database system that detects fractured blocks by matching the block header and trailer.

Serialization

Hard Requirements

1. All reads must see the effect of all writes completed before the read begins.
2. No write may ever be affected by a write that completed before the second write began.

Operating System Implications

The above requirements apply equally to storage subsystems and to the host operating system (OS) layers used to access the storage. Some of the implications of these rules are not obvious for OS layers which cache application data:

To satisfy the Durability requirement, all write operations must write through any OS cache to stable storage before they are reported as complete or otherwise made visible. Write-back caching behavior is prohibited, and data from failed writes must not appear in an OS cache.

To satisfy the Serialization requirements, any OS cache must be fully coherent with the underlying storage. For instance, each write must invalidate any OS-cached copies of the data to be overwritten, on any and all hosts, prior to commitment. Multiple hosts may access the same storage concurrently under shared-disk clustering, such as that implemented by Oracle RAC and/or ASM.

3) Technical Implementation of Database Storage

Operating Systems

All operating systems supported by Network Appliance storage systems are capable of meeting the operating system requirements listed above. However, for some operating systems and for some storage interconnects, correct operation may require the use of specific options on configuration parameters. These configuration requirements are discussed in best-practices papers that cover the specific combination of operating system, interconnect, and database.

This section provides a brief technical overview of how operating systems meet these requirements. This section is organized by the style of database access in use.

Raw Disks

As a general rule, all UNIX®, Windows®, and Linux® raw disk drivers provide for uncached I/O transfers that are atomic for database page sizes to at least 64 KB.

Raw Volume-Managed Disks

All volume managers provide uncached access to disk. However, if the volume is striped or implemented as a software RAID volume, the volume manager should be set up in such a way that an I/O to a single database page will always flow to disk as a single, atomic operation. Failure to follow this guideline exposes the system to the possibility of fractured blocks.

File Systems

Most databases are deployed on top of a file system. File systems provide great flexibility in managing storage, but because they also provide caching, care must be taken to ensure the correct mount options on file systems that support databases.

All host file systems that are certified by the database vendor may be safely deployed using Network Appliance Fibre Channel and iSCSI block storage. The certification process verifies the full coherency of the host file system cache. All of these file systems support atomic operations at least as large as a database block, and all support synchronous write operations when requested by the database.

Network Appliance supports a number of NFS client implementations for use with databases. These clients provide write atomicity to at least 4 KB, and support synchronous writes when requested by the database. Typically, atomicity is guaranteed only to one virtual memory page,

which may be as small as 4 KB. However, if the NFS client supports a direct I/O mode that completely bypasses the cache, then atomicity is guaranteed to the size specified by the “wsize” mount option, typically 32 KB.

The failure of some NFS clients to assure write atomicity to a full database block means that the soft atomicity requirement is not always met. Some failures of the host system may result in a fractured database block on disk. In practice such failures are rare. When they happen no data is lost, but media recovery of the affected database block may be required.

Network Appliance Storage Systems

This section discusses how Network Appliance storage systems meet the technical requirements of a database storage system.

All read and write operations of 64 KB or less, regardless of whether transported by NFS, Fibre Channel, or iSCSI, are processed by the upper levels of the Data ONTAP® system as atomic operations and passed on to the WAFL layer atomically. Read or write operations larger than 64 KB are broken at boundaries that are aligned 0 modulo 64 KB with respect to the starting offset of the operation. For databases with naturally-aligned, power-of-two block sizes, this ensures that WAFL processing is atomic with respect to all database blocks up to 64 KB in size, even when multiple database blocks are read or written within a single I/O operation.

WAFL provides storage virtualization inside Data ONTAP, and takes responsibility for all caching, storage allocation and retrieval. WAFL fulfills this function regardless of whether the data resides in LUNs or files.

The rest of this section discusses how WAFL processes read and write requests. Normal operation is discussed first, then various hardware and environmental failure modes. Finally, this section discusses the particular issues around the creation of Snapshot™ copies.

Any discussion of robustness in the face of failures must include some WAFL internals. This technical report stands alone, providing sufficient information to understand at a high level the recovery mechanisms inside WAFL. Readers desiring deeper knowledge of WAFL operation should consult the appropriate technical papers from Network Appliance.

Normal Operation

As discussed above, reads and writes are unconditionally atomic to 64 KB. While reads or writes may fail for a number of reasons (out of space, permissions, etc.), the failure is always atomic to 64 KB. All possible error conditions are fully evaluated prior to committing any updates or returning any data to the database.

WAFL manages data using 4 KB blocks. However, this 4 KB block size is not a failure boundary. All host writes up to 64 KB always succeed or fail atomically. All such write operations are either fully present on disk or fully absent. As a corollary, the effect of any write of up to 64 KB is either fully captured or completely missing from all Snapshot copies.

Failure Modes

Before understanding the protections offered by WAFL against various failures, it is important to understand some basics of WAFL operation. In brief, all host writes committed by WAFL are initially logged in non-volatile RAM (NVRAM), then subsequently stored in newly-allocated disk blocks. All writes follow this process, regardless of whether the new data overwrites existing data. From time to time, WAFL declares a consistency point (CP) to checkpoint all updates committed since the last CP. During CP processing, pointers to all of the modified data are adjusted to point to the new data. The modified blocks of pointers are written to free disk blocks. This process continues up the tree of pointers to the root. Simultaneously, the metadata that records free space is modified to show that overwritten data blocks not held in a Snapshot copy are now free. These modifications are also written to free disk blocks.

The last action of consistency point processing is to modify the volume's or aggregate's root block. This modification is done atomically. Because all data on disk is located through this block, this update atomically changes the on-disk structure from an old one which is completely consistent but does not contain any of the new updates to a new one that contains all of the new updates and has freed old, overwritten blocks of data.

Consistency point processing lies at the heart of the WAFL failure recovery design. Within a WAFL volume or aggregate, data on disk is *always* consistent. This consistency guarantee applies not only to WAFL meta-data, but to all user data as well. When WAFL declares a consistency point, all host write operations in progress (or 64 KB fragments of larger writes) have either committed their data to memory, in which case they are completely processed by the CP, or they have not yet begun to modify WAFL buffers, in which case none of their data will be included in the CP. Write operations that complete after the declaration of a CP create modified buffers that will be written to disk only after the current CP finishes.

Because data on disk is always consistent, Network Appliance storage systems will always fail in a way that leaves user data within a WAFL volume consistent. The data remains consistent regardless of the number of simultaneous failures. Data can be lost by the simultaneous failure of enough disks, but data can never be made inconsistent.

WAFL also provides for full recovery of all data modified by writes that have been acknowledged or otherwise made visible, regardless of whether the write has been captured on disk by a consistency point. Whenever WAFL processes any request that will change the state of storage on disk, the operation is logged in a write-ahead log in NVRAM. For write operations, the data is placed in the appropriate WAFL buffers and then logged. If the NVRAM contains insufficient space to log the request, the write operation blocks until NVRAM space can be allocated. Only after the operation is fully logged will WAFL return a successful status back to the host operating system.

In the event of a power failure, unplanned shutdown, or system failure, all of the WAFL volumes' data on disk remains consistent, set by the last CP, and a complete redo log resides in NVRAM. On reboot the log is replayed ensuring that no data is lost.

For storage systems used as primary storage in critical applications, Network Appliance recommends the use of clustered storage system models. In these systems, two storage units work as a unit, and their NVRAM logs are mirrored. In a clustered system, a write will not be acknowledged until the data has been captured in main memory and logged in both the primary system's NVRAM and the cluster partner's NVRAM. Thus in the event of complete failure of a motherboard, NVRAM board, or other complex hardware failure, the cluster partner can ensure no loss of data by taking over the failed partner's disks and replaying its NVRAM log.

WAFL depends on the Data ONTAP RAID subsystem to guarantee no loss of data on failure of a disk drive. However, because of the nature of WAFL consistency point processing, WAFL does *not* depend on atomicity of writes to disk to guarantee the atomicity of user write requests in the event of a failure. If power fails, writes in process to disk may be missing, partially complete, or fully complete; however, these in-process disk writes represent data that is part of a partially completed CP. Upon reboot, all of the on-disk data in the partially completed CP is discarded and then regenerated by replaying the NVRAM log.

Snapshot Considerations

WAFL implements Snapshot copies through the use of a special consistency point. When a Snapshot copy is requested, WAFL initiates a Snapshot CP. Processing of this CP is almost identical to a normal CP. However, during Snapshot copy creation the volume's root block is captured and saved as the root of the Snapshot copy. This block contains the master pointer to all pointers in the Snapshot copy. Future consistency points are prevented from reallocating the blocks used by this Snapshot copy, ensuring that the pointers reachable from the root of the Snapshot copy's tree remain valid.

Because the Snapshot copies are created using the consistency point mechanism, all of the atomicity guarantees provided by normal consistency points are also provided by Snapshot

copies. All write operations up to 64 KB are atomically captured or not captured by the Snapshot copy. Furthermore, because the Snapshot copy-creating consistency point is declared at an instant in time, any write not captured by the Snapshot copy must have been committed after any write that is captured by the CP.

The point-in-time nature of Snapshot copies creates crash-consistent copies of the user data within the volume containing the Snapshot copy. If the Snapshot copy is restored, the restored image looks to the database as an image of a volume that stopped cleanly, as if from a host system crash or a power failure. If an entire database resides within the volume, then a Snapshot copy captures a complete, consistent copy of the database. Although a Snapshot copy of a single-volume database will exclude transactions committed after the Snapshot copy was created, the Snapshot copy is consistent, and the mechanisms used by the database to recover from a system crash will work to recover a Snapshot copy of a database, recovering all transactions committed before the Snapshot copy was created.

4) Conclusion

Network Appliance storage systems exceed the requirements of all database systems for atomicity, durability, and serialization.

The unique Network Appliance storage management system of consistency points and redo logs ensures atomicity and durability of data even under circumstances when writes to disk non-atomically fail.

Network Appliance views meeting and exceeding the technical requirements of databases as part of the requirement to ensure that data is never lost and is maintained in an extremely available fashion.