Technical Report

# Virtualized Oracle Database Performance on NetApp AFF A-Series and C-Series

Joe Carter, Jeffrey Steiner, NetApp

May 2023 | TR-4971

## Important

Consult the Interoperability Matrix Tool (IMT) to determine whether the environment, configurations, and versions specified in this report support your environment.

TABLE OF CONTENTS

# Introduction

NetApp® ONTAP® is a powerful data-management platform with native capabilities that include inline compression, nondisruptive hardware upgrades, and the ability to import a LUN from a foreign storage array. Up to 24 nodes can be clustered together, simultaneously serving data through Network File System (NFS), Server Message Block (SMB), iSCSI, Fibre Channel (FC), and Nonvolatile Memory Express (NVMe) protocols. In addition, NetApp Snapshot technology is the basis for creating tens of thousands of online backups and fully operational database clones.

In addition to the rich feature set of ONTAP, there are a wide variety of user requirements, including database size, performance requirements, and data protection needs. Known deployments of NetApp storage include everything from a virtualized environment of approximately 6,000 databases running under VMware ESX to a single-instance data warehouse currently 996TB in size and growing.

The focus of this document is virtualized database performance using NetApp AFF storage systems, including both the A-Series and C-Series, and it covers both maximums and the practical difference between the two AFF options.

For more details, see the following additional resources:

- TR-4591: Database Data Protection
- TR-4592: Oracle on MetroCluster
- TR-4534: Migration of Oracle Databases to NetApp Storage Systems

# ONTAP AFF platforms

ONTAP software is the foundation for advanced data protection and management. However, ONTAP doesn’t only refer to software. There are multiple ONTAP hardware platforms to choose from that rely on a variety of storage technologies including flash media, spinning drives, and virtualized storage. Nearly all databases deployed today are hosted on solid-state storage, and the trend is accelerating.

NetApp offers two AFF platforms, the A-Series and the C-Series. Both are all-flash, solid-state storage solutions, but the A-Series targets ultra latency-sensitive workloads whereas the C-Series is aimed at solutions for which optimizing costs and capacity is a higher priority. The difference is almost entirely the media. TLC flash media has emerged as the enterprise performance market leader in solid-state drive technology, whereas QLC costs significantly less.

A-Series and C-Series systems are based on the same controllers, allowing you to right-size the solution for your workloads in terms of CPU, host and network connectivity, and RAM. You can also mix A-Series and C-Series in the same cluster, allowing you to create a tiered architecture. Finally, both the A-Series and C-Series use the same NVRAM and WAFL® technology in which write I/O is committed to mirrored NVRAM and written to media in full RAID stripes, yielding write latencies measured in microseconds.

# Database storage performance

The most important requirement for choosing a platform for your database is understanding your actual needs. Many customers made the transition from spinning media to 100% solid-state storage the moment an all-flash solution became affordable, but not all of them experienced a clear benefit. Some databases were never limited by latency in the first place; they were limited by bandwidth, for which solid-state and spinning media perform about the same. In other cases, databases weren't limited by storage performance at all, but were limited by the database query logic or database server CPU resources.

The following sections explain some of the considerations in choosing an AFF storage platform in more detail.

## Read latency - storage

Before the arrival of affordable all-flash storage, storage latency was regarded as the #1 problem with database performance, as was usually the actual case. There were two reasons for this. First, reading a database block from spinning media required around 8 to 10ms. If a database needed to serially read 1,000,000 individual blocks, read latency of 10ms each added up to a lot of time.

The second reason spinning media latency was a challenge was a result of the maximum number of I/O operations a single drive could service at a time. It was typically around 120 operations per second. Attempting to push more than 120 IOPS resulted in soaring latencies. The only solution at the time was to either add more drives into the storage solution or to use increasingly larger controllers with additional cache in hopes of avoiding drive I/O entirely.

NetApp AFF storage addresses both of these latency limitations:

- A-Series controllers with TLC media can deliver read latencies approaching 100µs.
- C-Series controllers with QLC media can deliver read latencies around 2ms.
- Both A-Series and C-Series drives support dramatically more IOPS than spinning media. The result isn't just better latency, but consistent, predictable, and better latency.

Whether you will notice a difference between A-Series and C-Series performance depends on the type of workload. While many database tasks require billions of individual reads performed serially, many other database tasks are driven by user activity. For example, if you're waiting for a report that summarizes hundreds of thousands of individual bank transactions performed during the day, you'll probably see a benefit from the improved read latency of the A-Series controllers. If a database is hosting an online order entry system, it probably isn't latency sensitive. End-users aren't going to notice a few extra milliseconds delay between the time they click the **Submit** button and the time they see the words **Order accepted!**

## Read latency - cache

The amount of RAM in a controller was often a critical consideration when sizing a storage solution based on spinning disks due to the limited IOPS and higher latency of spinning media I/O. Good performance depended on RAM offsetting the comparatively poor performance of spinning media.

With most all-flash storage solutions, the amount of RAM in a controller is rarely important because the service time of a read I/O from storage as compared to RAM is comparable. In this context, “service time” means the elapsed time from the moment the database issues an I/O operation to the moment the response to that I/O is received. The actual time required to read a block from RAM is obviously much lower than the time required to read a block from even an NVMe drive, but once the time required by the host, protocol, and network layers is included, cache reads and drive reads show comparable latency.

This changes a little with C-Series due to the higher latency of the QLC media. Read I/O that can be serviced by cache on A-Series and C-Series can be serviced with latencies approaching 100µs, but the latencies on drive reads will be noticeably higher with QLC.

Whether you see a difference between A-Series and C-Series sometimes depends on the chosen controller model. Databases often have very small working sets. For example, NetApp knows of one customer who has a nearly 250TB database, but only around 1TB of the database is active. The result would be nearly equivalent performance on an A800 and a C800 because the working set would reside in RAM on the controller.

This is an important consideration when choosing between A-Series and C-Series - how big is the working set size of the database? It's extremely difficult to directly quantify it, but sometimes it can be estimated. For example, a call center database with 5 years of data that is 50TB in size nevertheless might have a working set of only 100GB because most of the activity is from recent customer contacts. Customers rarely call in to inquire about bills that are more than a few months old, and, if they do, some additional latency in retrieving the required data is unlikely to cause problems.

## Write latency

Nothing damages database performance more than high write latency. Every time a change is committed by a database, one or more writes to the transaction logs must be completed and acknowledged by the storage system.

Fortunately, databases are almost never limited by write latency. The primary reason is that NetApp arrays, like most modern storage systems, do not commit writes directly to backend media. With ONTAP, inbound writes are journaled into mirrored NVRAM and then acknowledged to the host. The update to the drives happens much later in the write process. This is why ONTAP can deliver write latency measured in microseconds, even with spinning drive storage systems.

In addition, ONTAP WAFL technology avoids the RAID parity problem that affects writes with many competing storage systems. Without WAFL, you would never want to use a RAID-4/5/6 implementation because of parity. Every database write would require multiple read I/Os from storage in order to recalculate parity. This was sometimes called the RAID penalty because completing a write required additional read I/O. ONTAP does not have this limitation because of WAFL. Inbound write I/O is journaled into NVRAM and then organized into full RAID stripes that can be written as a single unit. Reads are not required to complete the write.

Both the A-Series and C-Series use ONTAP with WAFL with the same NVRAM write technology, which means both deliver the same ultra-low latency write I/O. Much of the legacy database documentation and guidelines surrounding the use of RAID, media, and other aspects of database storage sizing is not applicable to NetApp storage because ONTAP is immune to the problems such recommendations are trying to solve.

## Bandwidth

Some databases are bandwidth hungry. This is common among databases referred to as data warehouses or with tasks such as batch reports. The actual I/O pattern depends on many factors, but they include a lot of large-block sequential I/O operations. Despite the fact such workloads are very I/O-intensive, the actual media type used rarely makes any difference. The reason is large block sequential I/O operations are inherently very efficient operations. ONTAP storage systems can detect a sequential I/O operation in progress and proactively start reading from drives and assembling required data before a host has even issued the request. Data is also read more efficiently in larger blocks.

The result is these types of workloads should run more or less identically on A-Series and C-Series because these workloads are not latency-sensitive. Databases that are known to be limited by sequential I/O performance are usually having problems because of either host configuration errors or limitations in the network itself where ports have reached their maximum line speed.

## CPU

Nearly all database performance problems reported to the NetApp Support Center are actually a result of database-level operations. In most cases, the root cause is CPU computation work on the database server itself. If, for example, 85% of the database time is spent with DB CPU work, there is rarely value in trying to optimize storage performance because improvements are unlikely to be noticeable. Improving performance requires optimizing the SQL query itself to make it more efficient. In other cases, performance is limited by database contention; queries are delayed because they are waiting for other operations to release locks on data.

## Oracle buffer cache

Avoiding I/O altogether is preferable to optimizing I/O. The cost of RAM constantly decreases, but many DBAs have not taken the easy opportunity to increase the size of the Oracle buffer cache, which would reduce the amount of storage I/O being performed. An A-Series controller is fast, but, even with NVMe drives, the best latency achievable is around 100µs. In many cases, a small investment in server RAM (or using overlooked RAM that already exists on the database server) can turn those 100µs I/O operations into local memory reads with latencies measured in nanoseconds.

Increased database caching also reduces the IOPS demand on the storage system. A small investment in server RAM may allow you to use a less expensive storage solution.

## AFF performance: A-Series and C-Series

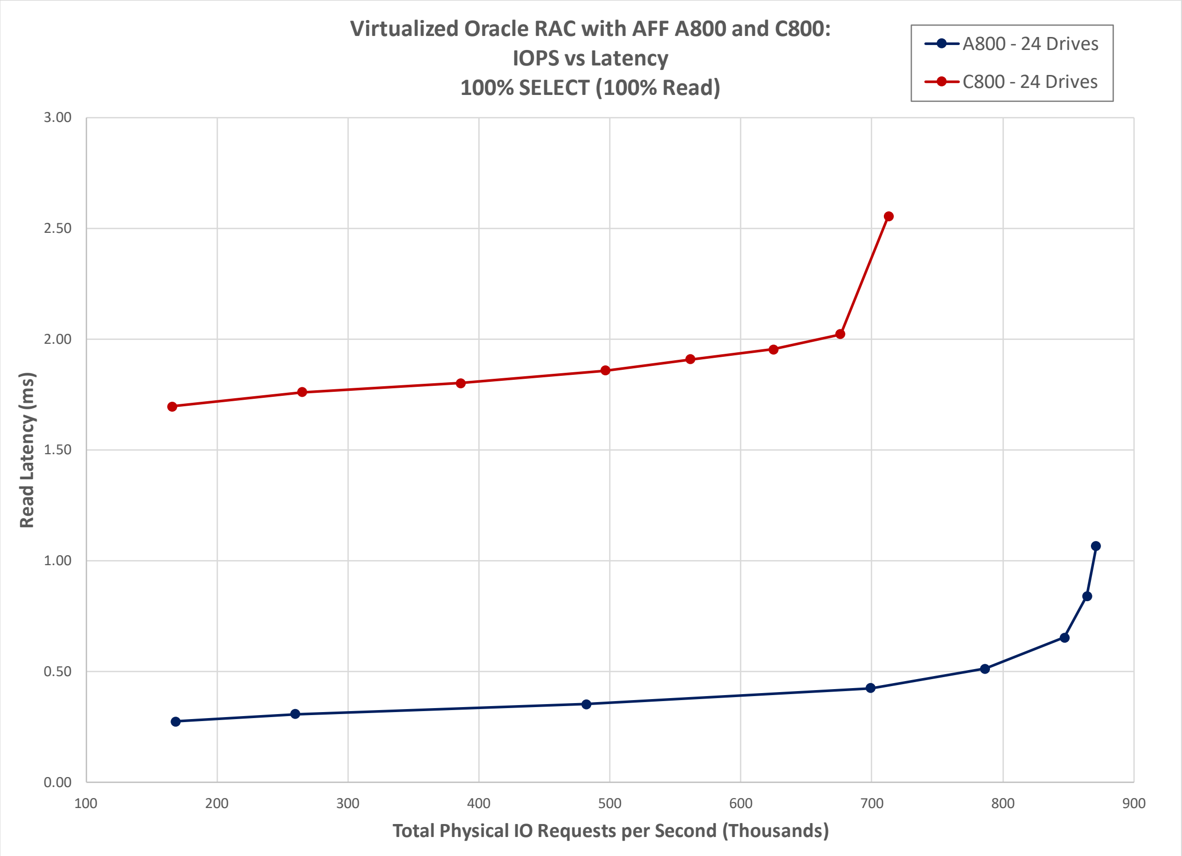
The following graphs show the performance capabilities of the AFF platforms in a number of configurations. The tests used a four-node Oracle 19c RAC cluster to generate the workload, and tests were performed on representative two-node A800 HA pairs and a two-node C800 HA pairs.

**Notes:**

- We chose the FCP SAN protocol for these tests. Although other protocols are supported by various hypervisors, including NVMe/FCP, FCP is still the most popular.
- Similar tests with NFS, iSCSI, and NVMe/TCP show a latency increase of around 100µs. This increase should be almost undetectable on C-Series controllers because of the generally higher latencies delivered by these systems. While a 100µs latency increase is detectable on A-Series, it is still unlikely to be noticeable to users because 250µs is still fast enough that storage is unlikely to be a performance limitation. The IOPS maximums with these protocols are approximately 30% lower. The reason is the overhead of the TCP/IP protocol on the storage system and the connected hosts. Even with this reduction, 600K IOPS is still far more than most database footprints need, and the cost and manageability benefits of NFS or the use of an IP network often outweigh any hypothetical impact to real-world performance.

## 100% read

Figure 1) A-Series and C-Series 100% reads.



The highlights of this chart are as follows:

### Latencies

The A-Series latencies are consistently around 250µs, and only reach 500µs as the controller nears saturation. This is a result of the faster TLC media. Most databases requirements from customers are based on a latency cutoff of 1ms, and the A800 meets this requirement over the entire performance curve. This is the reason that NetApp A-Series controllers usually erase storage as a database performance bottleneck. Databases become limited almost entirely by the query logic and CPU processing on the database server itself.

The latencies shown for the C-Series are higher due to the use of QLC media but are still far better than latencies for legacy spinning drive systems. Many advanced databases require the performance capabilities of an A-Series controller, but not all of them. Many of the largest databases in use today are performing the same functions they were performing 15 years ago before the era of all-flash storage.

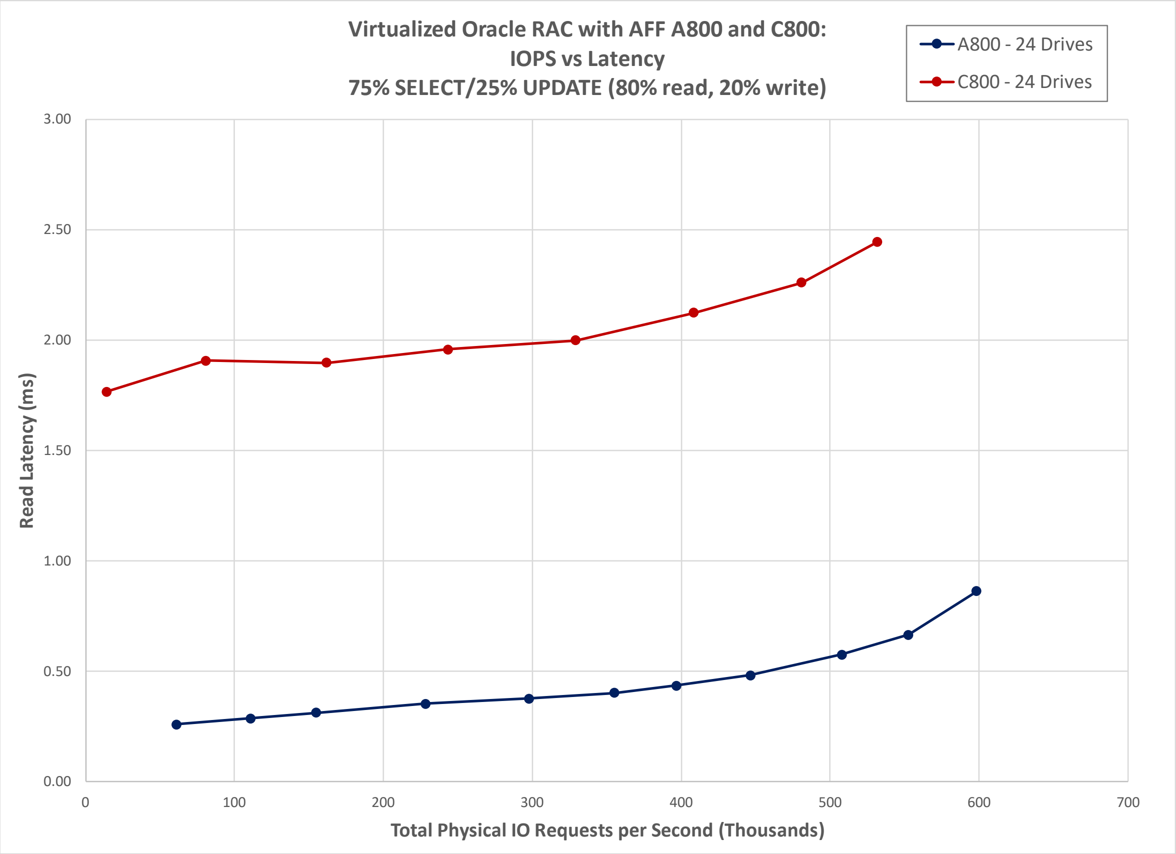
### IOPS

The graphs show the performance capabilities up to the point where the configuration saturates.

C-Series configurations reach a saturation point more gradually because the performance limit is mostly the drives. Latency increases as the drives reach their limits. In contrast, A-Series configurations are usually limited by the controller CPU. The drives themselves usually offer more performance potential than the controllers can extract. The result is much more consistent performance right up to the point the controller CPUs reach 100% capacity.

## Read-write

Figure 2) 80% read, 20% write.



The graph above shows a 75% SELECT test, which results in about an 80% read ratio. The reason is a test involving 25% update operations creates a read for each block updated, which slightly increases the read percentage.

### Latencies

The latencies are a little higher than the 100% read test, which is to be expected as the write I/O is producing additional load on the controller. Both the A-Series and C-Series are equally affected by the write I/O because the write path is the same. The storage OS is still ONTAP, which means the same write path is used when changes are committed to NVRAM, staged in RAM, and then written out to the drives.

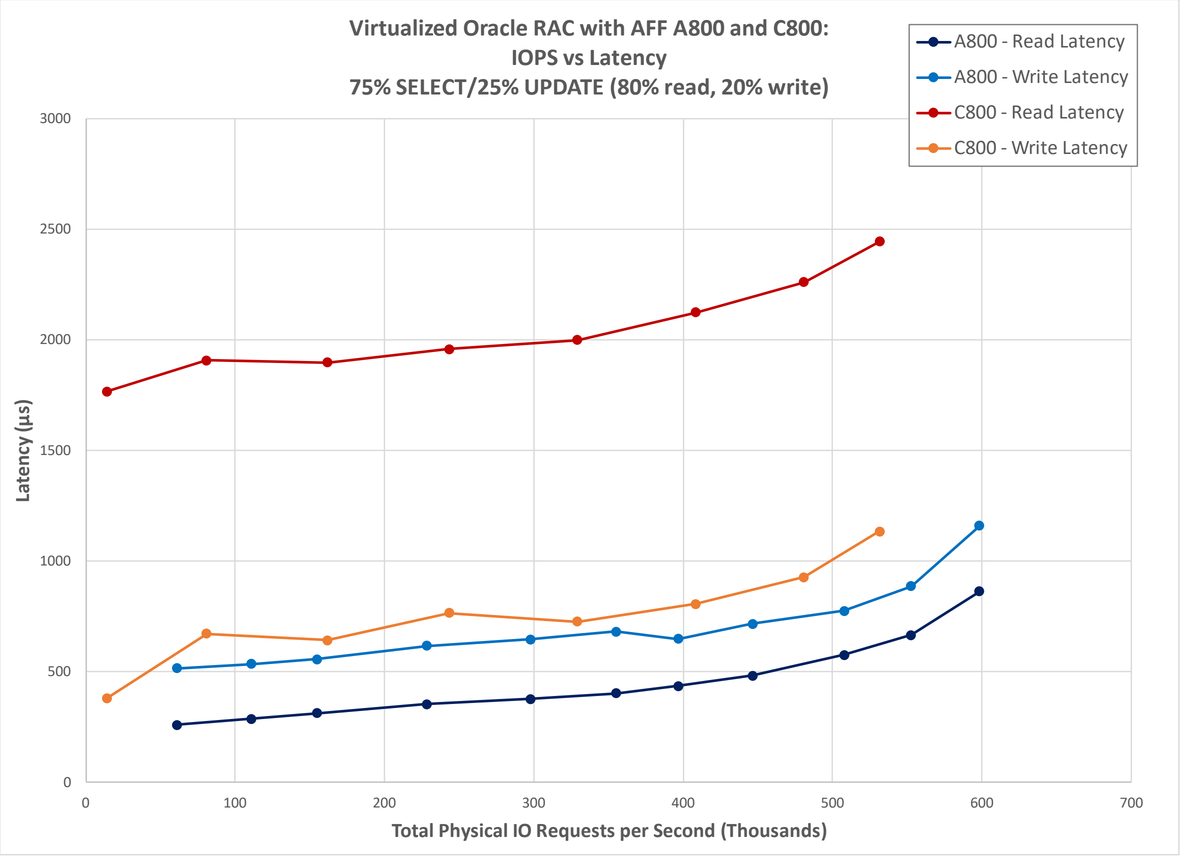
### IOPS

The IOPS maximums are similarly affected on both platforms as a result of the increased write activity. Write I/O requires more CPU work to process, and drives are slightly slower with a mix of read and write I/O than with pure read I/O.

## Write latency

There are many ways to measure write latency at the database level. Block sizes, parallelism, and the system calls used to perform the write vary. NetApp generally focuses on `log file parallel write` latency. This is the storage I/O component of a database redo logging operation, which is usually the most latency-sensitive aspect of database performance. DBAs sometimes focus on `log file sync` latency, but this is not entirely a storage operation. It will trigger a `log file parallel write` event, but it also includes other database-level operations that can delay the eventual storage I/O component of a redo log commit.

**Figure 3) A-Series and C-Series write latency comparison.**



As explained above, the write path with A-Series and C-Series is essentially the same. Whether you choose A-Series or C-Series, the write operation is complete from the host point of view once the I/O is journaled into NVRAM. There is a slight increase in C-Series write latency as overall I/O loads increase, which is a result of the drives becoming generally busy. Overall, write performance is the same on each platform.

# A-Series and C-Series compared

The most important question with any database sizing effort is “What do you need?”

For example, there is no reason to pay for IOPS capabilities or latency benefits if the workload does not rely on them. A POC that is based on nothing more than which configuration shows the highest IOPS potential on a given test isn't useful unless you know the limiting factor for a given business need is genuinely raw IOPS. In addition, a pure IOPS test ignores additional critical factors such as latency.

Not all workloads can be numerically quantified. Sometimes the only option is to take some time to understand the business need. For example, who is using the database? Are they just making updates, or are they running intensive reports that require millions of individual I/O?

When working with an Oracle database, the gold standard is the AWR report. It only takes a moment to run `awrrpt.sql` and generate a detailed performance breakdown of a particular hour in which I/O was known to be high, users were complaining, or critical processes were running. This tells you what level of IOPS you really need, what the current latencies look like, and whether storage is even a performance problem in the first place.

Once you have an idea what your needs are, you can select the right controller.

## Latency-sensitive workloads

The A-Series controllers are unquestionably the best option where read latency is critical. Both the A-Series and C-Series deliver comparable write latencies, ensuring critical processes such as redo logging are always running at peak performance. Likewise, reads of hot data that are already cached in RAM yield the same ultra-low response times. The difference between A-Series and C-Series becomes relevant when read I/O operations need to be serviced by a drive.

The difference between a 150us read operation on A-Series and a 2ms read operation on C-Series might seem substantial. However, 2ms is a substantial improvement over the latency on spinning disk solutions that are still used today by many mission-critical databases that require many terabytes of storage and high throughput and yet accept 8-12ms of latency on random reads.

### Buffer Caching

As described previously, avoiding I/O altogether is preferable to optimizing I/O. Increasing RAM allocated to the database buffer cache effectively reduces latency by changing physical reads from storage into logical reads from cache.

Counterintuitively, increasing the size of the buffer cache often results in an increase in average storage latency, but this is a result of converting the cache hits within the storage system into cache hits within the database. As the low-latency storage cache hits disappear, only the I/O operations that required drive access remain. This skews the average latency higher, but performance has nevertheless improved because there are fewer storage I/O operations occurring. Less time is spent by the database on storage I/O.

If a C-Series controller does not provide sufficiently low latency for a particular database, simply increasing the RAM allocated to the database buffer cache improves effective latency and may be preferable to re-hosting the database on an A-Series system.

## IOPS

The required IOPS level is mostly controlled by the storage controllers. The NetApp account teams and partners have sizing tools that can help you select the right controller. The drive count with AFF systems does not affect the maximum IOPS as much as it did with spinning media, but there is some effect. As a rule-of-thumb, AFF systems should have a minimum of 24 drives per HA pair (12 per controller).

## Durability

A second difference (in addition to the latency characteristics) between QLC drives and other flash technology is the wear capacity. The manufacturer specifications for many QLC drives include a reduced overwrite capability as compared to TLC drives, but this has minimal effect on ONTAP storage systems. First, ONTAP RAID protects data against media failures and includes both double and triple parity drive options. Furthermore, ONTAP WAFL technology distributes inbound write data to free blocks across multiple drives. This minimizes overwrites of the individual cells within the drive, which maximizes the drive's useful life. Finally, NetApp support agreements that cover drive failures also include drive replacement for SSDs that have exhausted their write cycles.

### Tiering

The choice between A-Series and C-Series is not an either/or decision. For example, you could build a four-node cluster containing two A800 controllers for latency-sensitive databases and two C800 controllers for others. You can easily and nondisruptively migrate databases between tiers as needs change. You could add C250 controllers for cold data, such as Oracle database backups.

### Compression

Temperature-sensitive storage efficiency (TSSE) is available on the A-Series and always enabled on C-Series. This feature improves storage efficiency by detecting inactive data and recompressing it with a larger compression block. This further reduces storage requirements. In practice, there is usually enough routine Oracle activity, including full table scans, backups, reindexing, upgrades, and other activities that touch most or all blocks in a database, to prevent TSSE from taking effect in the first place.

If TSSE with a 14-day cooling policy was enabled for a particular set of database volumes, and the database was shut down entirely for 30 days, the underlying blocks would be 100% recompressed by TSSE. If the database was then restarted and subject to heavy random read and write I/O, performance would be affected. Performance would eventually return to normal pre-TSSE levels.

This does not necessarily mean that the increased latency would be problematic, especially on a C-Series system where drive latency is already higher than A-Series drive latency. Furthermore, the increased savings on cold data might offset any such increase.

### Spinning and hybrid array upgrades

If we performed similar tests on a legacy spinning-disk solution, the average latency would start at around 8ms because that is the time required for an electromagnetic drive head to seek to the right position and transfer a block. The C-Series delivers a nearly 4X improvement.

The saturation point problem also drives migrations from spinning disk solutions toward C-Series. Many workloads are not latency-sensitive, but they still require high IOPS levels. A typical high-performance database in the days of spinning disk solutions would involve as many as 1000 drives because each drive could only service around 120 IOPS. 1000 drives would yield around 120K IOPS.

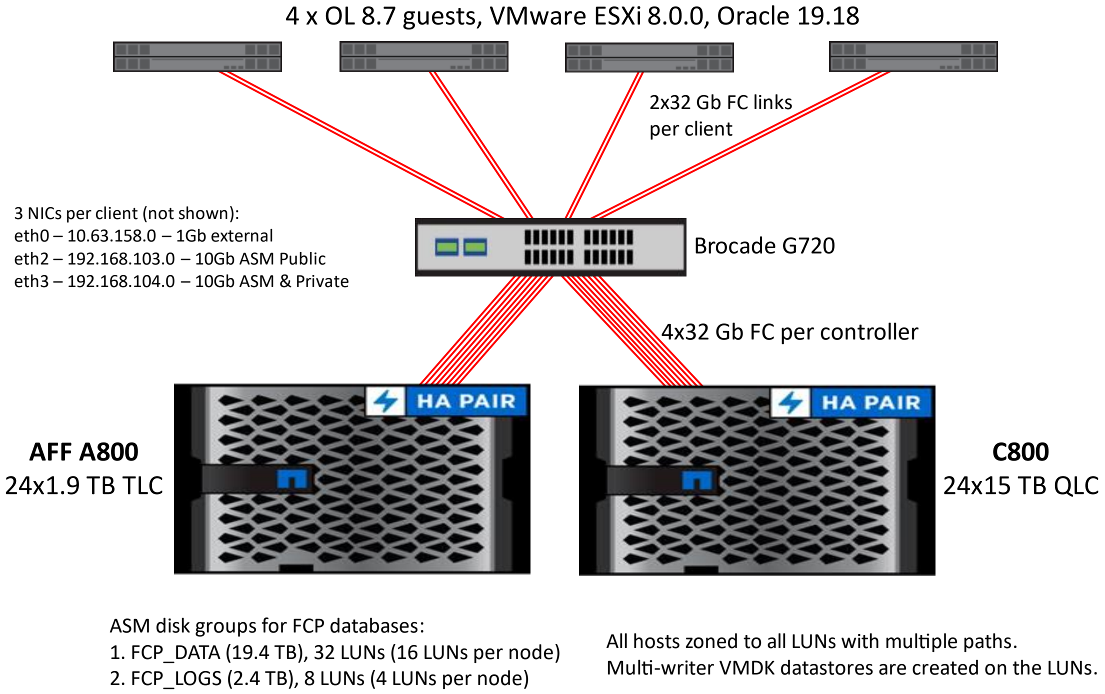
This is no longer viable. Disregarding the decreased reliability, increased power consumption, and increased heat output of spinning drives, such solutions are impossible to build because the size of drives in the dwindling spinning disk markets are enormous. Storage solutions would reach their maximum capacity in terms of bytes with a small handful of drives. The IOPS limitation would be severe. A solution with only 50 spinning drives would only support around 6000 IOPS.

C-Series solves this problem without the expense of the TLC media in the A-Series controllers; you get vastly more IOPS potential and increased reliability in a much smaller footprint with much lower power and cooling requirements. That alone would be a significant benefit, but you also get a 4X (or better) improvement in latency.

## Test configuration

The tests were performed as follows:

Figure 4) Oracle Database and AFF configuration.



## Database servers

The virtual database servers were configured as a four-node cluster running on Oracle Linux 8.7, with each virtual machine hosted on a separate ESXi server.

## Oracle version

Oracle 19.18 RAC Grid Infrastructure and Oracle 19.18 Database were used for these tests.

## Network

The FC network was configured with two 32Gb FC connections for each RAC node and four 32Gb FC connections for each storage controller. As controllers are normally deployed in HA pair, we used a total of eight FC connections for the A-Series system and eight FC connections for the C-Series controller. Testing was performed on either the A-Series or C-Series systems (not simultaneously), so that each test used 128Gb of bandwidth from the RAC cluster to the target storage system.

The IP network used 10Gb NICs, but the workload generator requires minimal RAC node-to-node communication, so IP networking does not affect the results.

## Oracle ASM configuration

The storage was configured based on NetApp best practices for Oracle on FCP SAN with a database intended to consume the total performance capabilities of a single storage system.

The datafile diskgroup was built on 32 VMDKs, and the log diskgroup was built on eight VMDKs.

Each individual VMDK was hosted on a dedicated VMFS datastore, as is required for parallelization. Although individual VMFS datastores can span multiple LUNs, the result is concatenated storage, not striped storage. Placing multiple VMDKs on a single VMFS datastore does not guarantee even loading across all available LUNs. Therefore, multiple VMFS datastores were built with one VMDK each to guarantee that the I/O performed against the ASM diskgroup utilized all LUN devices evenly.

A single LUN can support as much as 200K database IOPS. Using an ASM configuration with 16 datafile LUNs per controller allows a single database to drive the controller to nearly 100% maximum capacity. There is little benefit to increasing the ASM disk group beyond this level, and it would complicate management by unnecessarily increasing the number of storage objects on the system.

The number of LUNs used for log IO, including control files, archive logs, and redo logs, is less important because of the limited parallelism requirements. Four LUNs per controller should be more than sufficient.

Each LUN was placed in a dedicated volume. Normally the LUNs of a single ASM diskgroup would be co-located in a single volume to simplify management, but separating LUNs across volumes slightly improves write latency as a result of improved parallelism at the ONTAP layer.

## Oracle SLOB

SLOB (Silly Little Oracle Benchmark) is the premier tool for I/O benchmarking with an Oracle database. Other tools such as HammerDB or Swingbench are either complicated to set up or have dependencies on the server hardware, Oracle version, or Oracle configuration. SLOB is the ideal tool to drive a database to perform storage I/O, and the results are a product of the complete storage path, meaning the storage-related parameters on the database, the network characteristics and limits, and of course the storage IOPS and response time capabilities.

All tests were run with a very small Oracle buffer cache, which generates minimal cache hits on server RAM and maximizes storage I/O on the array.

Finally, the SLOB tables were populated using `OBFUSCATE_COLUMNS=TRUE`. This setting is critical to obtaining accurate results with modern storage systems such as ONTAP that include efficiency features such as compression and deduplication. Without this setting, the data created by SLOB is extremely and unrealistically compressible and deduplicatable, which can lead to excessive caching on the storage system. The goal of these tests is to stress the complete code path between the Oracle database and the drives, not to test cached I/O performance.

## Working sets

The working set is the portion of a dataset that is frequently accessed. Most Oracle databases have random I/O concentrated in a small portion of the total database size. Many POCs and Oracle performance results reported by competitors involve small working sets for which nearly all the I/O was serviced directly from RAM and not the backend drives. This skews the results, and we wanted to avoid that skewing in these tests by ensuring most of the I/O involved actual drive reads.

We also did not vary the number of SLOB users during the tests. Understanding why requires understanding the difference between a SLOB user and a SLOB thread. When you create tables for SLOB testing, you must specify the number of users (also referred to as schemas). If you create a SLOB database that is 1TB in size with 16 users, the result is effectively 16 partitions of that 1TB space, each of which is 64GB in size. Tests involving only a single user or schema would exercise only 64GB of space.

As a result, testing that increases the overall IOPS on the system by increasing the number of users or schemas also expands the working set size. The tests with fewer users or schemas have a much higher proportion of cache hits on the storage system, and thus the latency looks better than it would have in a more realistic configuration. All databases benefit from cache hits, but few customers buy a system as large as an A800/C800 for a database that fits wholly in cache.

A more realistic approach is to always use the all the SLOB schemas but vary the threads per schema. This is supported by SLOB.

Therefore, we created a balanced configuration. We built 10TB tables for use by SLOB, which is about 10X the size of available RAM on the storage controller. We set the SLOB SCALE parameter to 655360M and used 16 schemas. All tests involved all 16 schemas to ensure that the entire 10TB database was being accessed.

### Curve generation

The performance graphs were created by performing 20-minute tests at increasing loads until the saturation point was reached. Each test was run three times to validate that the results were consistent.

### Data selection

Results were reported in terms of IOPS and latency. A SLOB thread is only a load injector, and the IOPS produced by a defined number of SLOB threads vary based on the RAC configuration, Oracle version, and other factors. There is no way to extrapolate SLOB users to something useful in the real world. What does matter, however, is the resulting IOPS and associated latencies.

The X-axis of the graph shows the total datafile reads and writes. Testing involving more than 0% writes would also have redo logging activity, which can generate a substantial amount of I/O but cannot easily be expressed as IOPS because the block sizes and the asynchronous or synchronous nature of the I/O both vary greatly. Therefore, we focused on the most demanding type of I/O: datafile random read IOPS and datafile random overwrites.

The Y-axis in the initial graphs show the random read latency, which is usually the most important number because random read latency is the primary performance limit for a database, assuming storage I/O is a source of performance limits for a given database. We take this number from the Oracle `db file sequential read` statistic. Despite its name, this is not sequential I/O. It is a random I/O operation against an indexed sequence of blocks, or sometimes defined as a sequence of block reads.

We also provided random block overwrite latency on some graphs. This was taken from the Oracle `db file parallel write` statistic. This type of I/O is almost never a limiting factor for databases because it is almost always background operations. Changes to the database are committed to the redo logs, and the datafiles are updated at a later time. We only provided the random write latency to demonstrate that the write behavior is largely the same on the A-Series and C-Series because the write path is the same. Writes are committed to the NVRAM, not the backend disk.

In addition, the storage write latency is better than this number suggests. This type of I/O is usually performed by the database as asynchronous batches of I/O. The individual I/O operations for each block would show lower latency if that level of detail could be measured from the database level.

## Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp A-Series
  https://www.netapp.com/data-storage/aff-a-series/
- NetApp C-Series
  https://www.netapp.com/data-storage/aff-c-series/
- NetApp Product Documentation
  https://docs.netapp.com

## Version History

| Version | Date | Document Version History |
|---|---|---|
| Version 1.0 | May 2023 | Initial release |

| Version | Date | Document Version History |
|---|---|---|
| Version 1.0.1 | May 2023 | Minor typographical fixes |

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer’s installation in accordance with published specifications.

**Copyright information**

Copyright © 2023 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications. LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data—Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

**Trademark information**

NETAPP, the NETAPP logo, and the marks listed at https://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.