Database Layout with Data ONTAP™ 7G

Radhakrishnan Manga | September 2005 | TR 3411

Abstract

With the release of Data ONTAP 7G, Network Appliance introduced a powerful storage virtualization structure called flexible volume or FlexVol™. The FlexVol technology changes the way database can be laid out, thereby making effective use of the available disk space and bandwidth. This paper describes how the database can be laid out to make best use of the disk resources and how to reduce unwanted space overhead when Network Appliance™ Snapshot™ technology is used for faster database backup and recovery. It also describes various ways to handle the database reliability requirements with FlexVol. FlexClone™ is the logical extension of FlexVol; it enables the DBA and storage administrators to clone databases.

Table of Contents

1 Executive Summary

This technical report describes the newest Network Appliance storage technology and shows how it can be used to deploy databases more effectively. With the ever-increasing demand from worldwide enterprises for maximizing storage utilization while minimizing the storage management costs, the storage virtualization technology from Network Appliance provides a perfect answer. This document describes how the storage administrators and database administrators (DBAs) can effectively provision storage for database applications and worry less about their day-to-day management.

2 Introduction

Flexible volumes are a new feature introduced in the Data ONTAP 7G platform that allows storage administrators and DBAs to do flexible provisioning of available storage resources. Prior to Data ONTAP 7G, it was a standard practice to overallocate storage for various applications, thereby reducing effective storage utilization. This problem becomes more prevalent as larger hard drives become more common. Generally, the performance requirements of an application drive the minimum requirement for spindle count, commonly wasting the space unused on those spindles. With platforms prior to Data ONTAP 7G, traditional volumes are inextricably tied to the attributes and constraints of physical spindles. Release 7G introduces a new virtualization layer called **aggregate,** and the FlexVol volumes are carved out of these aggregates.

The introduction of this new storage virtualization technology by Network Appliance changes the way the storage administrators and DBAs can lay out their databases. The main goal of this technical report is to answer frequently asked questions like

- How many aggregates?
- How many flexible volumes?
- How many flexible volumes per database?
- Is it necessary to separate transaction logs from datafiles?
- What are the performance gains and risks of doing so?
- Can OLTP and DSS workloads be combined on a single aggregate?

3 Intended Audience

This technical report is targeted to storage administrators and database administrators. Though most of the discussions in this report appear generic to all databases, most of the examples discussed will be specific to Oracle® databases.

4 Prerequisites and Assumptions

The following assumptions are made about the readers of this document:

- The reader has a minimal knowledge of Network Appliance platforms and products.
- The reader has a general knowledge of database administration and database backup and recovery.

5 What Are Flexible Volumes?

FlexVol virtualizes volumes in Network Appliance storage systems by abstracting them from the underlying physical media. The size of the physical disk no longer determines how big a volume we can create; a FlexVol volume can be practically any size. A FlexVol volume can be dynamically grown and shrunk depending on the application needs without disrupting the application.

The two major advantages that flexible volumes provide are

- Improved space utilization
- Automatic load shifting

5.1 Improved Space Utilization

It is a common practice to partition the application at the volume boundary for various application management reasons. Prior to Data ONTAP 7G, the volumes were carved out of physical spindles, which did not provide optimal space utilization. With Release 7G, the applications can make more effective use of space and still get the advantages of having it partitioned at the volume boundary. In most of the customer deployments, storage utilization almost doubles.

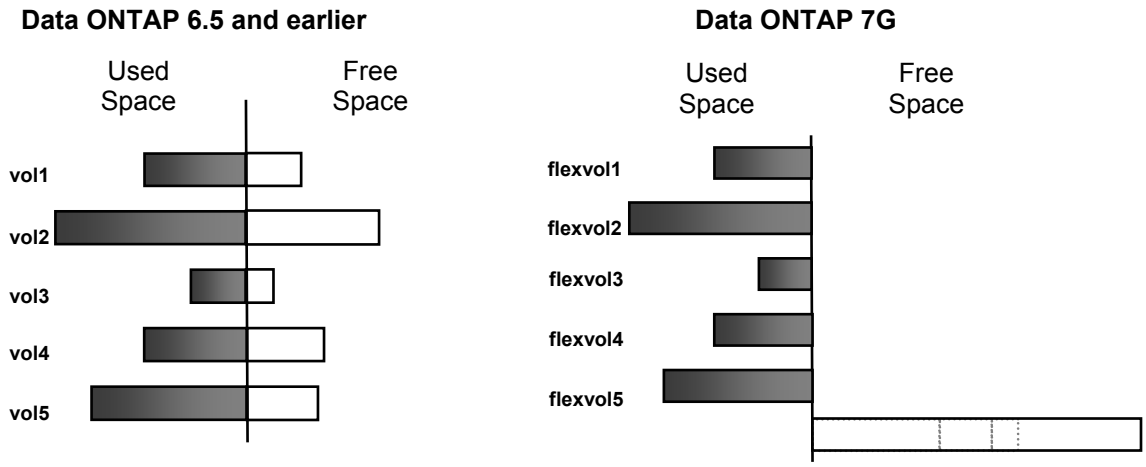

Figure 1) Improved space utilization with FlexVol.

5.2 Automatic Load Shifting

One of the factors that determine application performance is the number of physical spindles that support it. Earlier, when the volumes were carved out of physical spindles, application performance was determined by the number of spindles available in the volume. In a typical production environment, it is very common that not all applications require peak performance at the same time. It would be ideal to share the spindles across all applications while still ensuring space requirements for each application. FlexVol addresses this challenge by allowing automatic load balancing.

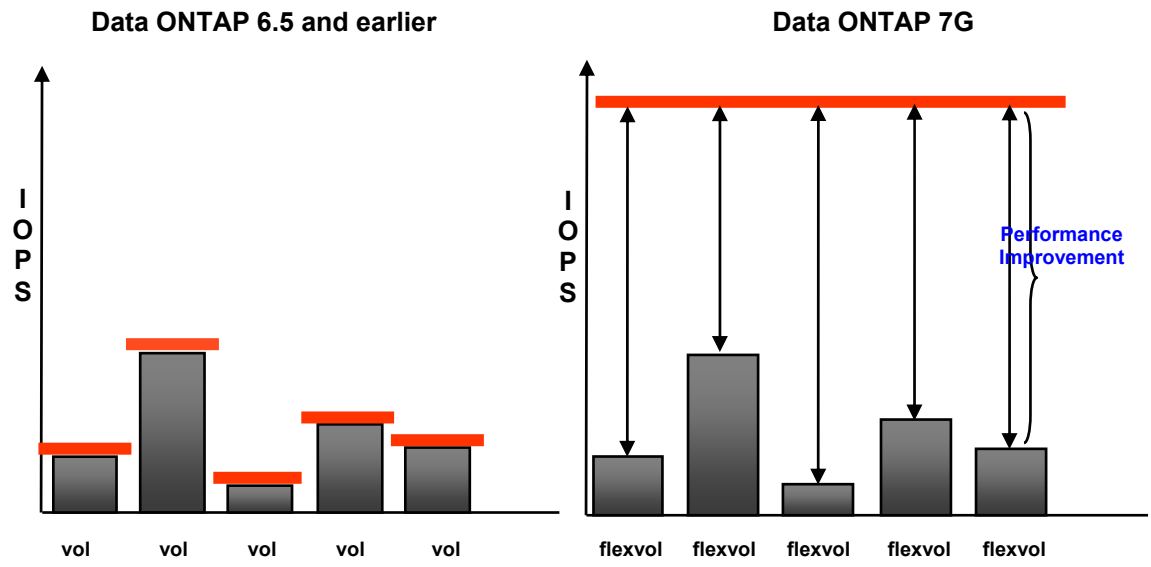

Figure 2) Automatic load shifting with FlexVol.

More details about the Network Appliance storage virtualization technology can be obtained from the following technical reports:

NetApp Virtualization technology – TR-3400
A Thorough Introduction to FlexClone Volumes – TR-3347
The Ideal Platform for Database Applications - TR-3373

6 Number of Aggregates

Aggregates are the new storage entities defined in Data ONTAP 7G. Aggregates are built out of a number of physical spindles. Each aggregate comprises a number of RAID groups and each RAID group is built out of number of physical spindles, including one or two disks for parity depending on the parity model selected. A FlexVol volume is a virtual entity, built above an aggregate. The storage bandwidth of all the spindles in an aggregate are available to all the FlexVol volumes built on top of the hosting aggregate. The greater number of spindles in an aggregate, the better the performance of all the FlexVol volumes utilizing the aggregate.

The common question any storage administrator or DBA may have is "How many aggregates?" The right number of aggregates depends on several factors: the type of storage system, performance requirements of the application, reliability requirements of the application data, recovery time objective (RTO) and recovery point objective (RPO) of the application. Several tests were conducted to determine the performance impact of having one large aggregate or multiple aggregates. Here is the list of experiments:

- Separating I/O within a database (does this mean just the logs and data or multiple data?)
- Mixing OLTP and DSS workloads on a single aggregate

6.1 Separating IO within a Database

A database system has two major kinds of I/O depending on the class of the application running on top of it:

- Sequential I/O
- Random I/O

Similarly, any database system also has different file I/O activities which can broadly be classified as shown in Table 1.

Type of File	Type of I/O	Comments
Data file activity	Depends on the workload type OLTP–Random I/O DSS–Sequential I/O	This is the I/O activity on the files where the real table data and index data reside.
Transaction log activity	Sequential I/O	This is the I/O activity on the transaction log files. The DBMS needs these files for transactional consistency. The DBMS engine ensures that writes reach the physical device before proceeding further.
Temporary file activity	Sequential I/O	This I/O activity takes place on the scratch pad area where the database engine builds dynamic tables and stores interim query results. These files are also heavily used for sorting and aggregation activities.

Table 1) Classification of I/O activities.

An OLTP workload representative of the real world was chosen for this testing effort. The database size was 1024GB. The database was created on the Network Appliance FAS980 system with 32 spindles in two different configurations as described below:

1. Separating the sequential and random I/O onto separate aggregates:

   - All the datafiles contributing to random I/O were put on the first aggregate of 24 disks.
   - All the files that contribute to sequential I/O, like transaction log, archive log, and temporary datafiles, were put on a second aggregate of 8 disks.

2. Combining all the files on a single aggregate of 32 disks

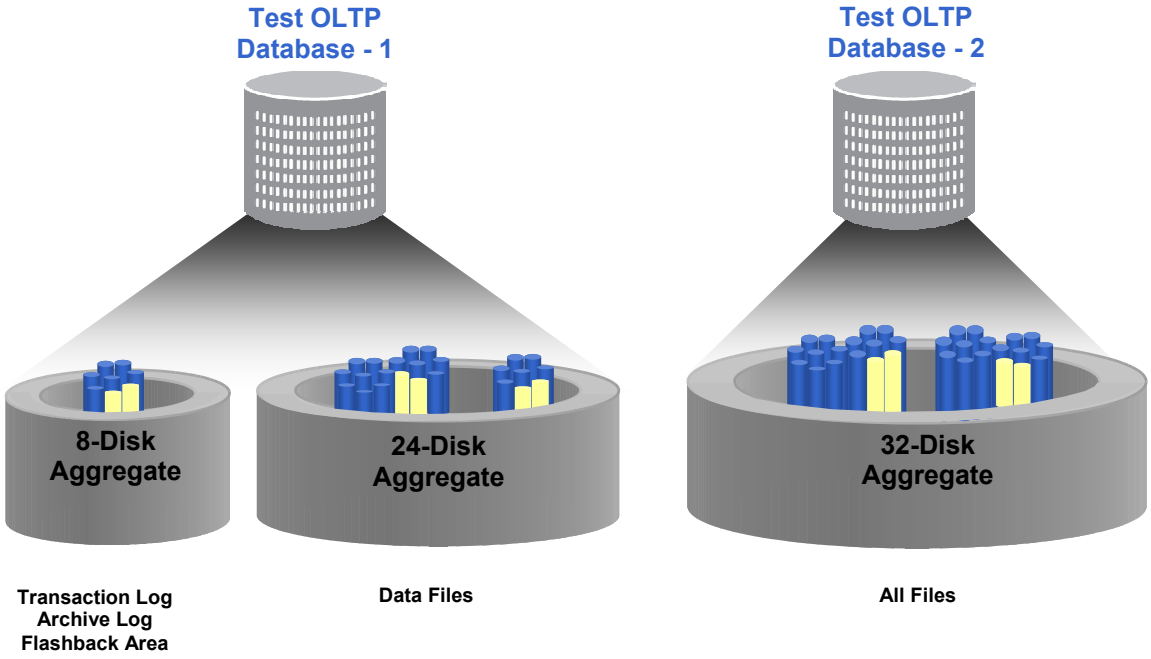

Figure 3) Database layout for test—"Separating I/O within database."

The single aggregate test showed better performance compared to the two aggregate one. The benefit seems to be more in the case of cold cache run where almost all of the data was directly read from disks. The normalized transactions / sec results are as shown below. Results clearly illustrated a marked performance improvement of 10% to 17% for a heavy OLTP workload.

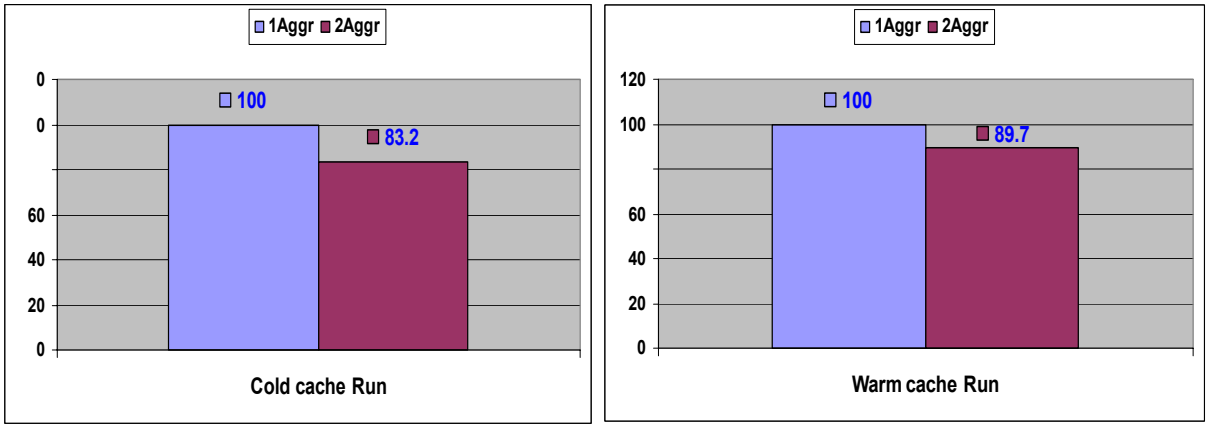

Figure 4) Results of test—"Separating I/O within database."

This test clearly proves that huge OLTP workloads perform better with one large aggregate compared to two aggregates with different types of I/O activity. It also proves that storage administrators and DBAs don't have to worry about separating different types of I/O as conventional wisdom dictated in legacy storage environments.

6.2 Mixing OLTP and DSS Workloads on a Single Aggregate

This section focuses on the performance impact of putting multiple workloads on a single aggregate. The overall workload for this test is a mixture of two database workloads: an OLTP workload and a DSS workload. These two workloads were run concurrently against the Network Appliance storage subsystem in two different configurations. In the first test both the OLTP and DSS workloads were running concurrently on two separate aggregates of 32 disks each. In the second test, both the OLTP and DSS workloads were running on a single dedicated 64 disk aggregate. The OLTP workload is identical to one used in the previous tests, and the database size was 1024GB. The DSS workload was running a very complex query against a 600GB data warehouse, which took about 30 minutes to finish. The volume layout for this test is as shown below.

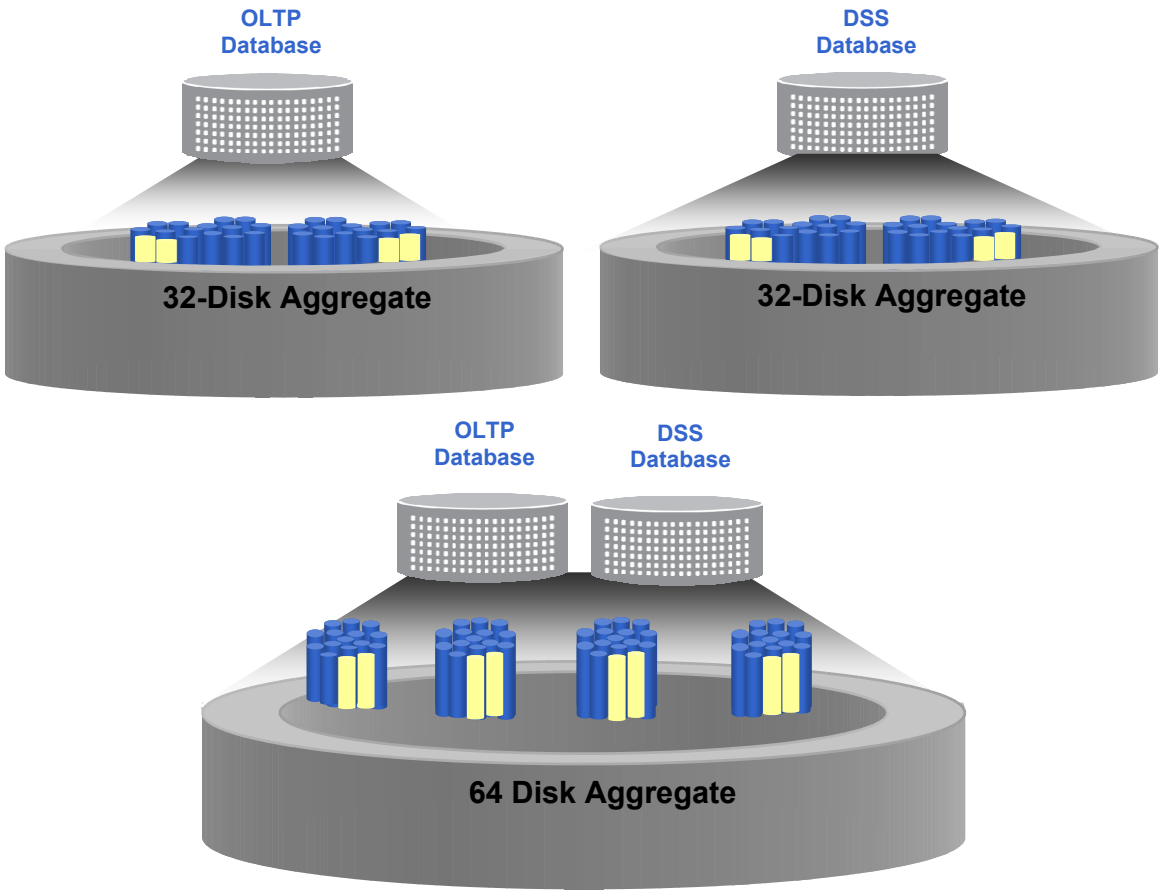

Figure 5) Database layout for test—"Mixing OLTP and DSS workloads on a single aggregate."

The performance results indicate that running both the workloads on a 64-disk aggregate is better than running them on two different aggregates. There was an 8% improvement in the heavy OLTP workload performance. Here are the normalized results.

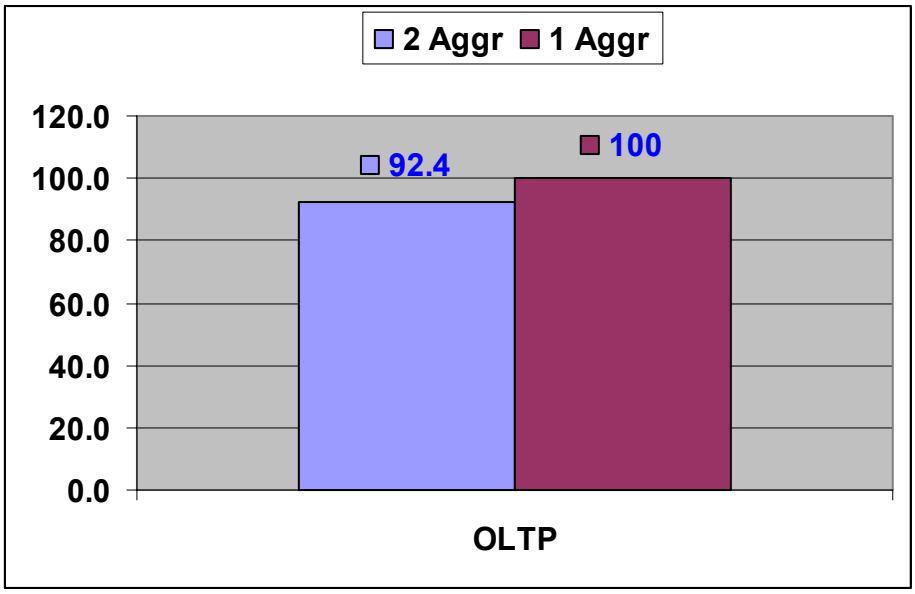

Figure 6) Results for test—"Mixing OLTP and DSS workloads on a single aggregate."

We did not see any change in the DSS query runtime because the system was already working at full network bandwidth even when the DSS workload was running on it own 32-spindle aggregate.

The above two tests indicate that creating a single aggregate with same number of spindles delivers better performance then creating multiple aggregates for different workloads or type of I/O.

6.3 Exceptions to Single Aggregate

There are certain scenarios where the single aggregate may not work:

Disks of multiple sizes, speeds and type involved. Network Appliance storage systems support drives of different speeds (10K/15K RPM), size and type (ATA/FC). It is not recommended to create an aggregate across disk of different speeds, sizes, and types.

Storage requirement of more than 16TB. The biggest aggregate that can be created with Data ONTAP 7G is 16TB. Applications involving storage above 16TB need to create more than one aggregate.

Data reliability requirements. The customer may have reliability requirements that can drive the choice for multiple aggregates. Section 8 will discuss various scenarios and their merits and demerits as far as reliability is concerned.

Special software requirements. The customer may use certain software features that work only at aggregate level. If the customer uses any of these software features, they may need to create more than one aggregate. Software features that can warrant the use of multiple aggregates, presuming an operator wants to use them on just a portion of data are

- SyncMirror®
- MetroCluster

7 Database Layout

A database instance is simply a process or collection of processes that work on a set of datafiles to store and retrieve information. Storage administrators and DBAs can maximize performance and utilization of their storage subsystem by laying out these files in an optimal way. The efficient layout of files can significantly improve application lifecycle management and can improve the backup/recovery window for the databases.

7.1 Types of Database Files

Databases deal with different types of files for storing data, providing the transaction consistency, storing the configuration information and storing intermediate results. The type of I/O on these files is different. The typical files found in any database installation are

- Database binaries
- Database configuration files
- Data files
- Temporary database files
- Transaction log files
- Archive log files
- Cluster-related files (only some databases)

7.1.1 Database Binaries

These are the database executables, shared libraries, etc. and will change only during the patching process. A separate FlexVol volume should be created for storing these files. Network Appliance Snapshot and FlexClone technologies can be used to improve the patching process.

7.1.2 Database Configuration Files

These files store the metadata about the database state and configuration. They are referred to as **control files** in Oracle database. It is recommended that a separate FlexVol volume be created for storing these files. Some databases support multiple (multiplexed) copies of these files; it is recommended that the copies of these files be stored along with on-line transaction logs. During the hot backup creation with Oracle, the order of creating Snapshot copies on database volumes is very important to create a valid backup set. The Snapshot copy of the control files should be created at the end.

7.1.3 Data Files

These files store the actual data of the database. It is recommended that one or more FlexVol volumes be created for storing these files. These files need periodic backup depending on the RTO and RPO requirements of the database application.

7.1.4 Temporary Database Files

These are the files used as scratch pad area by the database engine. The contents of this file are transient in nature and do not need backup and recovery. Intermediate results from the database queries, database sorting, and aggregations are stored in these files. The I/O activity on these files is sequential in nature with huge average I/O transfer size. These files should be placed on a separate FlexVol volume and Snapshot should be disabled on this volume to reduce unnecessary space overhead.

7.1.5 Transaction Log Files

These files store the database transaction information and are critical to database consistency and recoverability. Some of the databases support multiplexed transaction log files to protect against the loss of a file. A separate FlexVol volume should be created to store these files. It is recommended that multiplexed files be stored on the FlexVol volumes consisting of the database configuration files.

7.1.6 Archive Log Files

These files store the database transaction information and are very important for the database recoverability. In the typical production environments, these files are often stored for longer times. The database can be rolled forward from a valid backup state using these files. A separate FlexVol volume should be created to store these files.

7.1.7 Cluster related files

These are the configuration and status files of the clusterware. Only few databases have these files. It is recommended to place them on a separate FlexVol volume because of some special mount requirements they have in the case of NFS. It is recommended to use the vendor-supplied utilities to back up and restore these files rather than using Snapshot copies.

7.2 Number of Flexible Volumes

7.2.1 Oracle Layout Example

The number of flexible volumes is determined by how many types of files from the above list exist in the given database setup. Figure 7 shows what an Oracle flexible volume layout will look like:

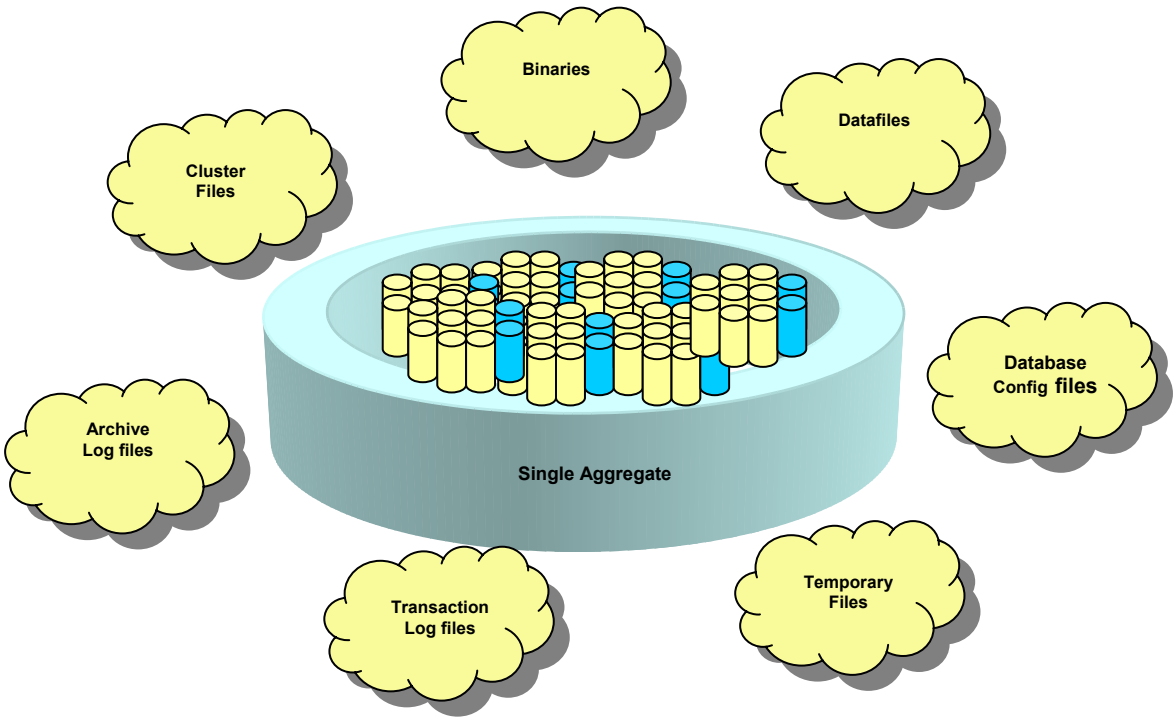

Figure 7) Oracle flexible volume layout.

This layout significantly improves the management of database backup and recovery. The Network Appliance Snapshot technology can be efficiently used to create a minimal required backup set and reduce unnecessary overhead.

7.2.2 Dealing with a Multiple Database Scenario

Frequently a single NetApp storage system hosts more than one database. The layout described above can be easily used as long as there are fewer databases. In Data ONTAP 7G the maximum number of flexible volumes that can be created is 200. This value is only 100 in the CFO scenario (When a node fails the surviving node has to handle all 200 flexible volumes). The only way we can solve this problem is to use the qtrees underneath the flexible volume. Table 2 describes the volume layout options with multiple databases.

Database file type	FlexVol / Qtree	Reason
Datafiles	Separate FlexVol volume for each database	Each database may have a different backup schedule. It is easier to manage Snapshot and SnapMirror® operations if they reside on a separate FlexVol volume by themselves.
Database config files	Qtree under a single flexible volume.	These files are typically backed up using the database supplied utilities.
Transaction log files	Qtree under a single flexible volume	These files are not part of the database backup. It is common to use SnapMirror on these files for DR purposes. The QSM functionality can be used to replicate these files.
Archive log files	Qtree under a single flexible volume	QSM can be used to replicate these files for DR purposes.
Temporary files	Qtree under a single flexible volume	These files are not part of a database backup process and never need to be replicated.
Clustered related files	Qtree under a single flexible volume	These files are typically backed up using the database supplied utilities.
Database binaries	Depending on user needs	These files are updated when the application patches are applied. Depending on the number of flexible volumes available, the customer can decide whether to use a separate flexible volume for each database or create the qtree for each database under a single flexible volume. The disadvantage of using qtrees is that the SnapRestore® operations have to be performed at the file level.

Table 2) Volume layout options with multiple databases.

The production environments may contain different classes of applications that require various levels of availability. The databases that require highest availability use the SnapMirror Sync replication for the transaction and archive logs, which ensures minimal or zero data loss in the event of disaster. The best approach in these scenarios is to separate the databases into two classes:

- Databases that require SnapMirror Sync for logs
- Databases that do not require SnapMirror Sync for logs

Create a separate flexible volume for each class and use SnapMirror Sync at the volume level on the flexible volume that contains the logs for the databases that require SnapMirror Sync replication.

Figure 8 shows a sample layout in a 20-database scenario.

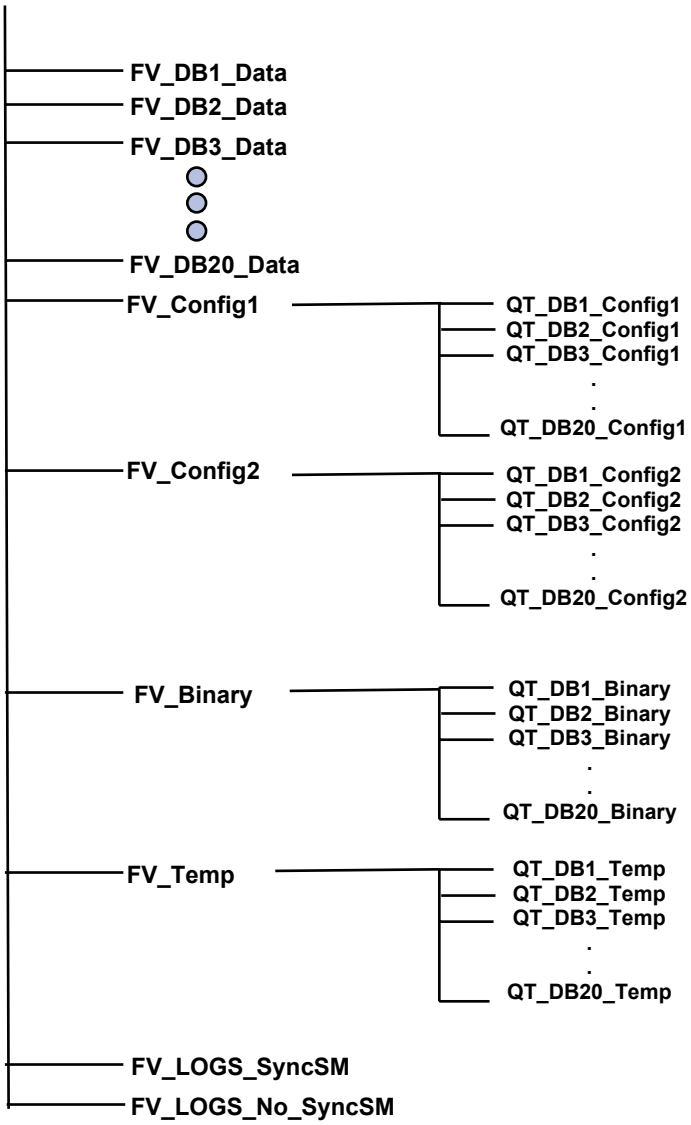

Figure 8) A sample layout in a 20-database scenario.

7.3 File Layout with Failover Cluster

Database implementations involving the Network Appliance clustered failover configuration should lay out database files on both the nodes of a cluster to get maximum performance out of all the spindles available. A Network Appliance storage cluster does not do the automated I/O load balancing; the clustered failover feature is strictly for HA purposes. The ideal way to split the workload is to split the datafiles into temporary files that account for most of the I/O for both nodes and place transaction log files on one node and archive log files on the other.

Node 1: Part of datafiles, part of temporary files and transaction log files
Node 2: Part of datafiles, part of temporary files and archive log files

This layout will ensure that the I/O bandwidth is distributed during the automatic archive process.

8 Reliability Concerns

Storage reliability is a major concern for any database deployment. It is a common practice to allocate a set of disks for database files and another set of disks for transaction log files. The following are the advantages and disadvantages of laying out data in this fashion:

- **Advantages**
  - **Reliable database recovery**
- **Disadvantages**
  - **Inefficient space utilization**
  - **Inefficient spindle utilization**

With the latest improvements in disk reliability there is less chance of double disk failure within a single RAID group. RAID-DP™ also significantly improves reliability, which is far higher than that of RAID5 and significantly closer to that of RAID1. Detailed descriptions of Network Appliance RAID-DP technology can be obtained from the technical report TR-3298. An entire aggregate will fail only if three disks of a single RAID group contained within that aggregate fail.

It is always recommended to follow the layout recommendations with single aggregate. In extreme cases where there is a need for separating datafiles from transactions logs, it is advised to follow the recommendations listed below.

8.1 Separate Aggregate for Logs

The number of spindles required to support the I/O associated with transaction and archive logs is small because of the sequential nature of the I/O. An aggregate with fewer spindles can be created and all the flexible volumes associated with the transaction and archive logs should be created on it. The spindle sizing is beyond the scope of this technical report, but it should be noted that the log aggregate created should support at least five times the peak transaction log rate. In Oracle, for example:

- Oracle LGWR writes to online-redo log until it gets full.
- Once the online-redo log becomes full, Oracle LGWR switches to a second online-redo log.
- Oracle ARCH process also launches an archival operation concurrently on the first online-redo log and onto an archive file.

Assuming the there is a redo rate of **X** MBps, the total I/O involved as soon as a switch happens can be calculated with formula shown below:

X MBps write on to the new online-redo log
+
Total I/O on log aggregate = X MBps read from the old online-redo log
+
X MBps write on to the archive log file

The log aggregate should be able to support at least three times the redo rate. It is better to have the rate slightly higher than this because it is recommended to finish the archiving before the next switch happens and to support some unexpected spikes in the redo rate.

8.2 Multiplexed Redo Logs

Some databases support multiplexed transaction logs. It is recommended to create two aggregates according to the recommendations in the previous section and have the primary logs on the same volume as data files and the multiplexed set of logs on the log aggregate. This will reduce the bandwidth requirement on the log aggregate as the archive process always reads from the primary logs and creates the archive logs. In the case of clustered failover, the multiplexed set can also be created on the second node, but this may introduce extra latency as the logger process does not treat a write as successful until it successfully writes to all the multiplexed locations.

9 Summary

Data ONTAP 7G provides answers to common DBA and storage administrator questions and concerns such as

- How many spindles for data volume?
- How many spindles for log volume?
- What are my usage levels of my volumes?
- Can I mix OLTP and DSS on the same volume?
- Can I mix redo log of two databases on a single volume?

The experimental data proves that creating fewer aggregates and using the FlexVol feature of Network Appliance Data ONTAP 7G significantly improves overall spindle utilization. It is highly recommended to create as fewer aggregates as possible and let Data ONTAP handle the complexity.