ORACLE
Cloud Infrastructure

Implementing Remote Replication Using Oracle ZFS Storage Appliance

An Oracle ZFS Storage Appliance White Paper

JUNE 25, 2019

ORACLE

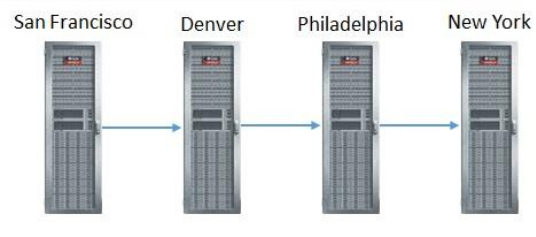
TABLE OF CONTENTS

INTRODUCTION

An increase in storage demand increases the complexity of protecting data. The storage-based remote replication capability of Oracle ZFS Storage Appliance products offers a simple and effective automated solution for businesses that require offsite copies of production data in addition to local backups. By maintaining a replica of the primary data at remote sites, disaster recovery time can be drastically reduced compared to traditional offline backup architectures. Additionally, the replica at the remote sites can be accessed by other business applications, thereby reducing the processing load on the primary system.

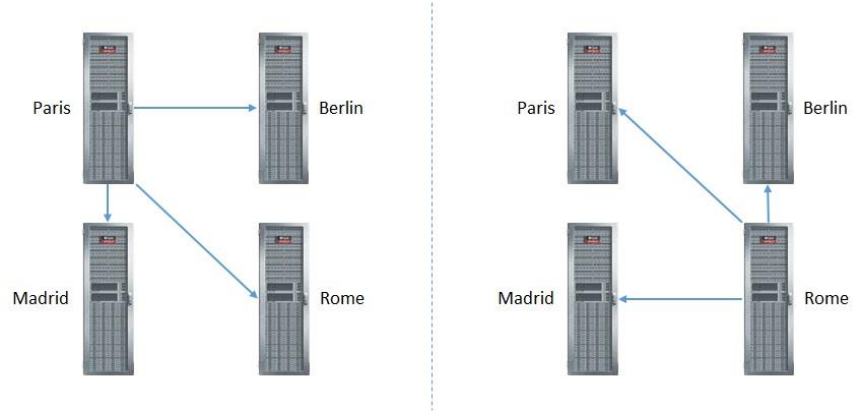
In its simplest form, Oracle ZFS Storage Appliance can replicate projects or shares between two systems within a datacenter or across a geography. This is referred to as "basic replication". Basic replication considers only the data source and one data target for a particular project or share. The foundations of basic replication includes elements such as compression, deduplication, and security of the transferred replica, in addition to restartable data transmissions, and reversing the replication direction to promote the replication target as the source.

"Enhanced replication" builds upon basic replication by eliminating the restrictions of the one-to-one relationship of basic replication for projects. Enhanced replication includes cascaded replication and multi-target reversal. Cascaded replication enables a replica to be propagated along a series of Oracle ZFS Storage Appliance systems, each maintaining an accessible copy of the replica.



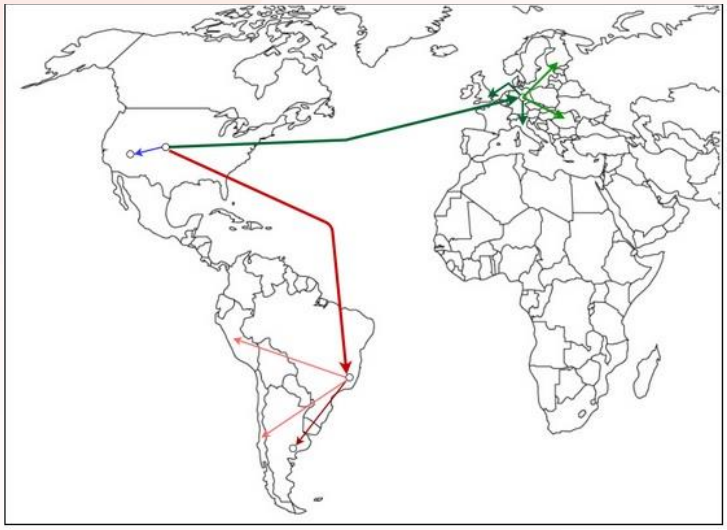
An system outage anywhere in the cascaded series can be circumvented and reestablished to the next system in the series in oder to maintain business continuity.

Multi-target reversal enables more than one immediate Oracle ZFS Storage Appliance system to become the primary source for the data, while maintaining replication connectivity and activities to the other systems.



A replication reversal action by any of the receiving systems will convert that system to the source and will start replicating to the other systems. If the replication reversal was due to a system outage, replication will be reestablished once it is returned to service.

With minimal effort, these replication capabilities can be combined to create complex topologies within datacenters, across sites, and across continents, maintaining fault tolerance and efficient data transfer between all systems.



Lastly, enhanced replication provides the mechanism of encrypting an existing unencrypted replication configuration by directly replicating from the unencrypted replication source to a new encrypted replication target. This allows the unencrypted to encrypted replication conversion to be performed with almost no down time.

This paper addresses considerations in planning and implementing remote replication with Oracle ZFS Storage Appliance. The information particularly addresses system planners seeking to simplify offsite data protection.

This paper presents the following topics:

- Overview of Oracle ZFS Storage Appliance replication features and capabilities
- Replication topology building blocks
- Implementation guidelines for deploying replication
- Application-specific implementation guidelines
- Replication benefits
- Replication considerations

Note that the functions and features of Oracle ZFS Storage Appliance described in this white paper are based on the latest available Oracle ZFS Storage Appliance firmware release. Refer to the software release notes to determine compatibility and limitations.

OVERVIEW OF ORACLE ZFS STORAGE APPLIANCE REPLICATION

Oracle ZFS Storage Appliance products support snapshot-based replication of projects and shares from a source appliance to any number of target appliances or to a different pool in the same appliance. Replication can be executed manually, on a schedule, or continuously for the following use cases:

- **Disaster recovery.** Replication can be used to maintain a replica of an Oracle ZFS Storage Appliance system for disaster recovery. In the event of a disaster that impacts service of the primary appliance (or even an entire data center), administrators activate service at the disaster recovery site, which takes over using the most recently replicated data. When the primary site has been restored, data changed while the disaster recovery site was in service can be migrated back to the primary site and normal service restored. Such scenarios are fully testable before such a disaster occurs.
- **Data distribution.** Replication can be used to distribute data (such as virtual machine images or media) to remote systems across the world in situations where clients of the target appliance would not ordinarily be able to reach the source appliance directly, or such a setup would have prohibitively high latency. One example uses this scheme for local caching to improve latency of read-only data (such as documents).

- **Disk-to-disk backup**. Replication can be used as a backup solution for environments in which tape backups are not feasible. Tape backup might not be feasible, for example, because the available bandwidth is insufficient or because the latency for recovery is too high.
- **Data migration**. Replication can be used to migrate data and configuration between Oracle ZFS Storage Appliance systems, or to move data to a different pool within the same Oracle ZFS Storage Appliance when upgrading hardware or rebalancing storage. The Shadow Migration feature of Oracle ZFS Storage Appliance also can be used for data migration.

Oracle ZFS Storage Appliance replication has the following properties:

- **Snapshot based**. The replication subsystem takes a snapshot as part of each update operation and sends the entire project contents up to the snapshot, in the case of a full update. In the case of an incremental update, only the changes since the last replication snapshot for the same action are sent.
- **Block level**. Each update operation traverses the share at the block level and sends the appropriate share data and metadata to the target.
- **Asynchronous.** Because replication takes snapshots and then sends them, data is necessarily committed to stable storage before replication even begins sending. Continuous replication effectively sends continuous streams of filesystem changes, but it is still asynchronous with respect to NAS and SAN clients.
- **Includes metadata**. The underlying replication stream serializes both user data and Oracle Solaris ZFS metadata, including most properties configured on the Shares screen. These properties can be modified on the target after the first replication update completes, though not all take effect until the replication connection is severed; for example, to allow sharing over NFS to a different set of hosts than on the source. The replication documentation in the “Remote Replication” section in the Oracle ZFS Storage Appliance Administration Guide provides more information.
- **Protocol independent**. Oracle ZFS Storage Appliance supports both file-based (CIFS and NFS) and block-based (FC and iSCSI LUNs) storage volumes. The replication mechanism is protocol independent.
- **Secure**. The replication control protocol used among Oracle ZFS Storage Appliance products is secured with secure socket layer (SSL). Data can optionally be protected with SSL as well. An Oracle ZFS Storage Appliance system can replicate only to/from another Oracle ZFS Storage Appliance system after an initial manual authentication process. The Oracle ZFS Storage Appliance Administration Guide provides more details.

Oracle ZFS Storage Appliance replication also includes the following important features:

- **Replication Auto-Snapshot Management**. Even though the replication subsystem is snapshot based, all user created and system schedule snapshots on the share or project are maintained, and available both on the original share or project and on the replicated share or project. Also, additional independent snapshots can be created on replication targets. All clones created based on snapshots are preserved.
- **Resumable replication**. A replication process that was stopped (due to a network failure or system outage) can be restarted so that the replication process can continue from the point it stopped instead of having to retransmit all previously replicated data from the last snapshot or an initial replication process. For example, if a network failure were to occur after transferring 180GB of a 200GB replication update, only the remaining data in the update will be transferred once the network connection has been reestablished. The update does not need to start over.
- **Efficient Raw Send**. When a local project/share is set up to use compression, data blocks are directly replicated from disk to the target saving the decompression at the source and a compress step at the target. This avoids unnecessary use of CPU and bandwidth resources and reduces the duration of the actual replication process.
- **Adaptive multithreading and dynamic compression level of replication data streams**. The data in replication streams at the source is compressed to make better use of the available network bandwidth between source and target node. This is especially beneficial in situations where distance and network bandwidth are data throughput limiting factors. The compression rate and number of compression threads is dynamically adjusted based upon CPU utilization at the source and available network I/O throughput between the source and target Oracle ZFS Storage Appliance system.
- **Monitoring recovery point objective (RPO) target(s)**. An RPO target can be specified for each replication action. An alert is generated when the replication update exceeds the desired RPO target thresholds, which can inform the system administrator to address potential issues or perform additional system tuning.

Replication Modes

The Oracle ZFS Storage Appliance remote replication capability supports three different replication modes to give administrators flexibility when supporting new deployments and complex legacy applications:

- **On demand**. Replication is manually triggered by the user at any time, either from the web BUI, command line interface, or via REST API script.
- **Scheduled**. Replication is automatically executed according to a predefined schedule. Periodic replication updates can be scheduled to start every 5, 10, 15, 20, or 30 minutes; every 1, 2, 4, 8, or 12 hours; and every day, every week, or every month. More complex update patterns can be created by defining multiple schedules for a single replication action. If a scheduled replication update cannot be started because the previous update has not completed, the scheduled update will automatically be attempted again once that update does complete. Additionally, if a replication update fails for any reason, it will be retried.
- **Continuous**. The replication process is automatically executed continuously. As soon as one replication update is complete, a subsequent update is started. This way, the changes are transmitted as soon as possible.

The replication process is the same for each mode except that the time interval between the replications is different. The replication modes can be changed from one to another at any time to support different and changing business requirements.

For every mode, an RPO target can be specified for each replication action, and a warning alert and error alert thresholds can be specified to monitor the RPO target for each replication action.

The following diagram provides a high-level overview of the replication concept. Oracle ZFS Storage Appliance systems are used as both “source” (original data system) and “target” (any system receiving a replica). A project with two filesystems (NFS and CIFS protocols) and a block-based volume [LUN]) are replicated. At time t1, a full replication happens and subsequently at time t2, only the changes between the time t1 and t2 are replicated.

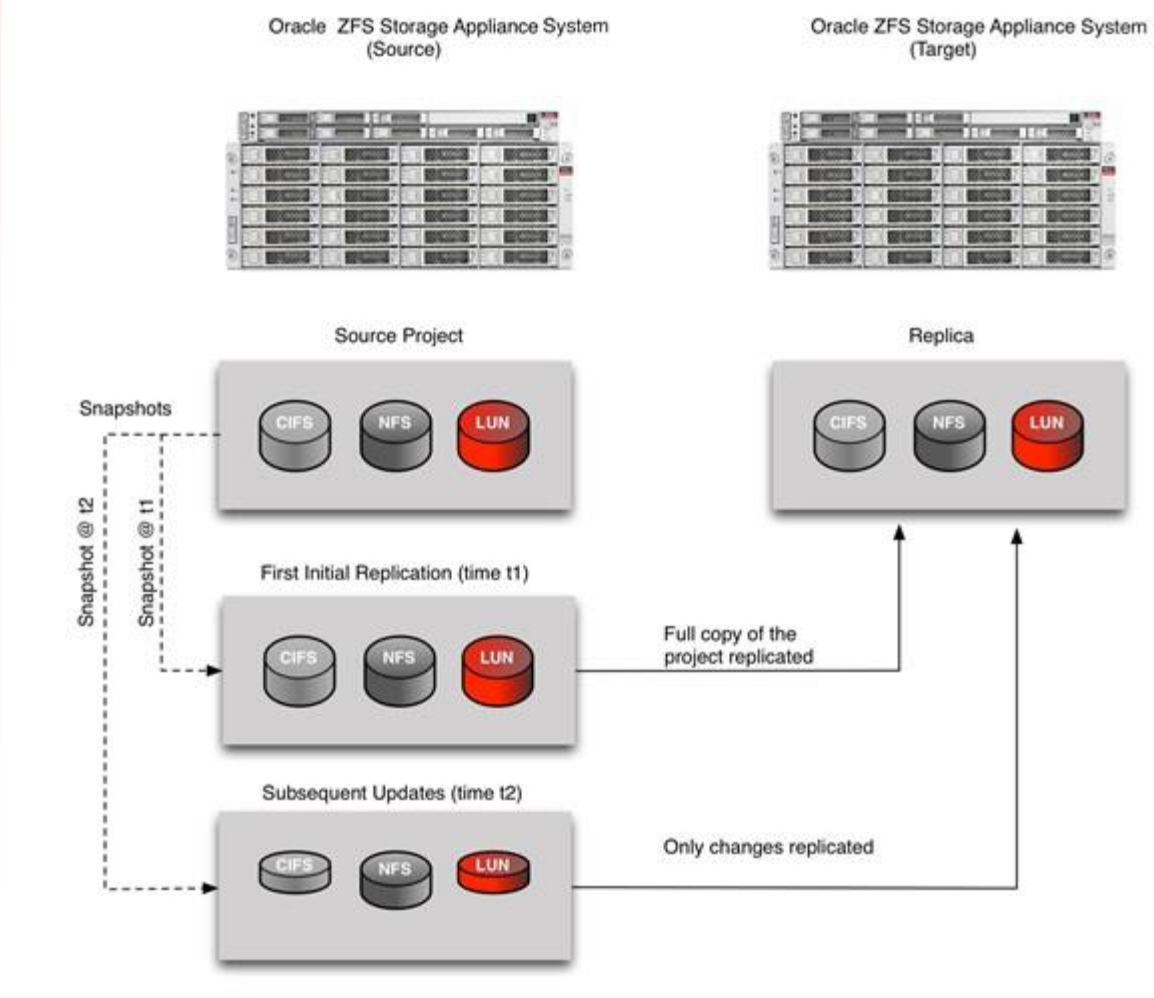


Figure 1. Replication process overview

Using the Replica at a Target Site

The replicated package can be used for failover, test, backup, and reporting purposes after at least one successful transmission to the target site.

The Oracle ZFS Storage Appliance products provide several ways to use the replica target:

- **Read-only access**. Enable shares to be mounted from the client for read-only purposes using the Export option.
- **Clone/export replica**. Create a new writable project from the replica for alternate business processing at the target:
- **Sever link**. Convert a replication package into a writable local project by severing the replication connection.
- **Role reversal (reverse replication)**. Sever the link (see preceding) and establish replication back to the original source; this operation reverses the roles of the source and target sites.

The replica at the target site can be promoted into production operation in the event of a disaster in which the source site is completely lost.

ORACLE ZFS STORAGE APPLIANCE REPLICATION TOPOLOGIES

Several terms are used for this discussion of replication topologies:

- **Project**. Refers to a logical grouping of user defined storage resources.
- **Share**. Refers to the storage resource that is contained in a project. A share can either be a filesystem or a LUN. Each project can contain one or many shares.
- **Package**. Refers to the data bundle sent during replication. It consists of periodic snapshots of the data which are used to keep the data consistent during replication
- **Source**. Refers to the project or share (on a node) containing the active data (read/write) that is being replicated.
- **Target**. Refers to a node receiving, storing (read only), and possibly retransmitting the replication package.
- **Node**. Refers to a generic term to indicate either a source or a target, or both.
- **Action**. Refers to the replication definition between two nodes.
- **Schedule**. Refers to the timing of automatic replication actions. Since actions can be initiated manually or programmatically, an action does not require a schedule.
- **Replication update**. Refers to a cycle of operations starting with an implicit snapshot followed by streaming of the data to the target to completion.

It is important to note that node only refers to one controller (and associated disk pool) of a dual-controller (clustered) Oracle ZFS Storage Appliance system. Replication can be performed between the two controllers of an Oracle ZFS Storage Appliance system.

Basic Replication

Basic replication consists of a single project or share replicated from a source to a target. In the event that the source is unavailable, the target can perform a role-reversal and become the source presenting a new read/write accessible project on the target. If the replicated data was a project, then all the shares within that project will be available. If the replicated data was only a share, then the new project will only present that one share. Once the original source is available again and after it has received the updated package from the current source, the original source can perform a role-reversal, once again becoming the source. The basic replication workflow is shown in Figure 2, using node A as the source, node B as the target, and the dot next to the node letter to indicate the role-reversal.

Figure 2. Basic replication package flow

Basic replication also allows for replication of the same project or share to multiple targets. This is performed by creating multiple actions on the source. Figure 3 depicts node A replicating a project or share to three separate nodes.

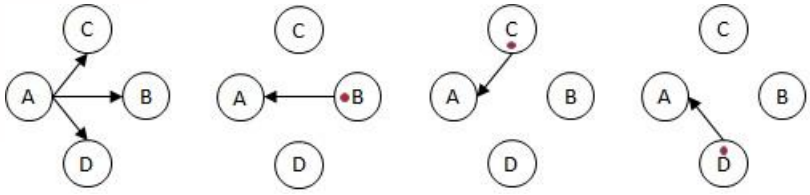


Figure 3. Basic replication of a single project to three separate nodes

This configuration is helpful in replicating the data to various locations, but has limited disaster recovery capability. If a replication reversal is required on any of the other three nodes, then the selected node will perform replication back to node A, but will terminate the relationship with the other two nodes. Once the role-reversal is performed again on node A, the actions to the other two nodes will no longer be valid and would require a full synchronization to become stable again, as shown in the leftmost diagram of Figure 3.

Enhanced Replication: Multi-Target Reverse

Multi-target reverse addresses the issue depicted in Figure 3 by utilizing the concept of "potential source". Potential source is an action attribute that indicates to the node that if it becomes the source after a replication reversal, it would continue all of the replication actions that the original source performed A potential source can be any node that the original project source directly sends replication data, and is selected as replication reverse capable. Figure 4 shows the same initial replication topology as Figure 3.

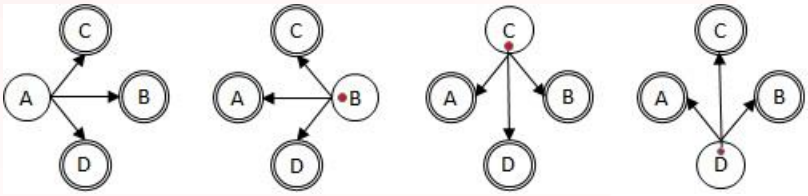


Figure 4. Multi-target reverse replication between potential sources

However, with each node having the potential source attribute selected in its replication action, the replication reverse action can be performed on the package of either node B, C, or D, and still maintain replication to the other three nodes. The nodes depicted with a double walled circle are the potential sources. When node B does a replication reversal, it becomes the project source, and the three remaining nodes are the potential sources. Replication reversals on either node C or D will perform similarly. In any of the cases, when node A performs the replication reversal to become the source again, the configuration returns to the original state shown in the leftmost diagram in Figure 4.

Since replication reversal is a manually initiated task, it is possible to initiate the reversal on multiple nodes at around the same time. If this occurs, a source conflict condition will disrupt the overall operation, since two potential sources will start conversion processes. This condition will be identified and an alert will be generated requiring manual conflict resolution. A source conflict cannot be prevented, but can be easily resolved if encountered.

Not all nodes need to be designated as potential sources. Basic replication can still be configured to targets that will not be able to perform a multi-target reverse operation. These nodes are called dedicated targets. Figure 5 shows a configuration similar to Figure 4, however, node D is designated as a dedicated target.

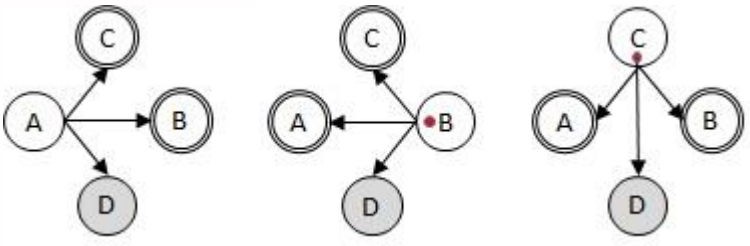


Figure 5. Multi-target reverse replication with potential sources and a dedicated target

Even though node D is a dedicated target, its replication connection is still maintained when replication reversals are performed on nodes B and C. Unlike basic replication, however, as a dedicated target node D cannot perform a replication reversal.

Potential sources utilize more storage space than dedicated targets since they maintain more project snapshots. These additional snapshots are needed to maintain the project consistency between the source and all of the potential sources. Storage capacity should be considered when architecting which nodes should be potential sources and which ones should be a dedicated targets.

Earlier Oracle ZFS Storage Appliance software versions may not support multi-target reverse, but can still be used as dedicated targets. Refer to the version release notes to check compatibility.

It is important to note that the enhanced replication capabilities are only available to project level replication and cannot be configured for share level replication.

Enhanced Replication: Distant Target

Having a potential source replicate back to all other replication targets may not be beneficial in every case. Consider the architecture shown in Figure 6.

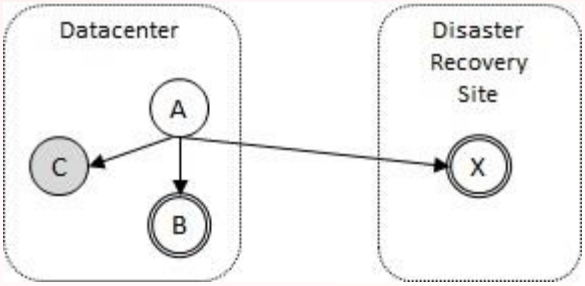


Figure 6. Example replication architecture between a datacenter and a disaster recovery site

In this architecture, node A is the active project replication source, node B is a nearby replication target that is selected as a potential source, node C is a nearby replication dedicated target, and Node X is a replication target in a disaster recovery site that is also selected as a potential source.

In the case of a replication reversal on node B, node B will become the project replication source and continue replicating to all other nodes, as shown in Figure 7.

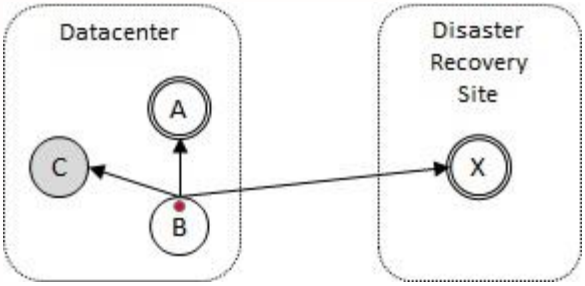


Figure 7. Effects of Node B replication reversal

The network connection between the datacenter and disaster recovery site only carries the network traffic of a single replication action. In the event of a replication reversal on node X, however, will make node X the project source and will then increase the network traffic threefold, as shown in Figure 8.

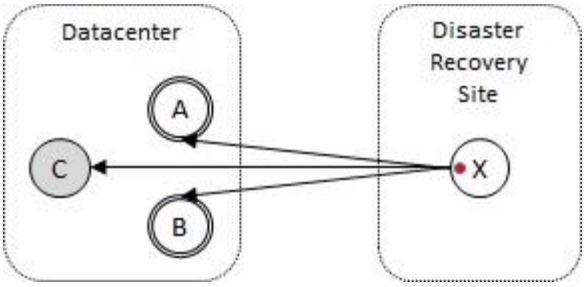


Figure 8. Effects of Node X replication reversal

The increased network traffic would make the recovery of a datacenter node take longer since it would be in contention with the other two nodes. In order to address this issue, a potential source can also be selected to be a distant target. A distant target is a potential source which will only restore direct project replication to other distant sources when a replication reverse is performed. Figure 9 depicts the same replication architecture as provided in Figure 7, however it designates node X as a distant target by using the double-lined arrow connection.

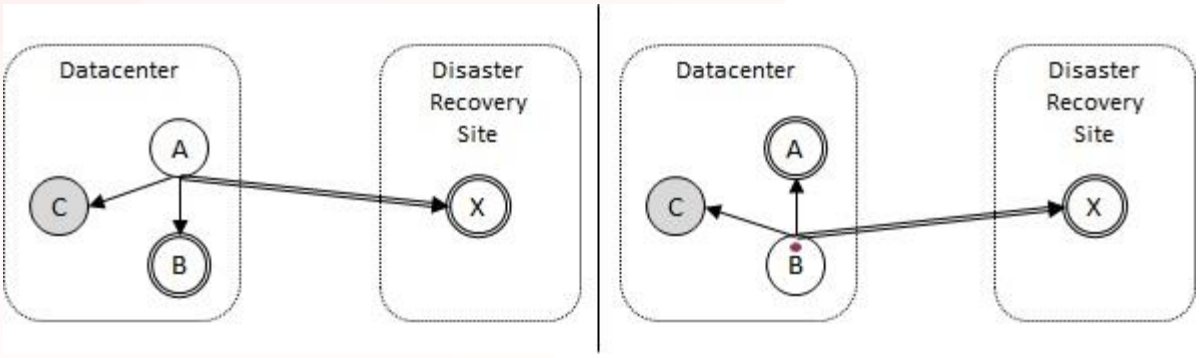


Figure 9. Example replication architecture including distant target configuration

Replication reversals performed on node A or node B exhibit the same behavior as before, continuing replication to the other three nodes. A replication reversal performed on node X, however, will only restore the replication connection to the node that was last replicating to it, and will not create actions to replicate the project to the other nodes, as shown in Figure 10.

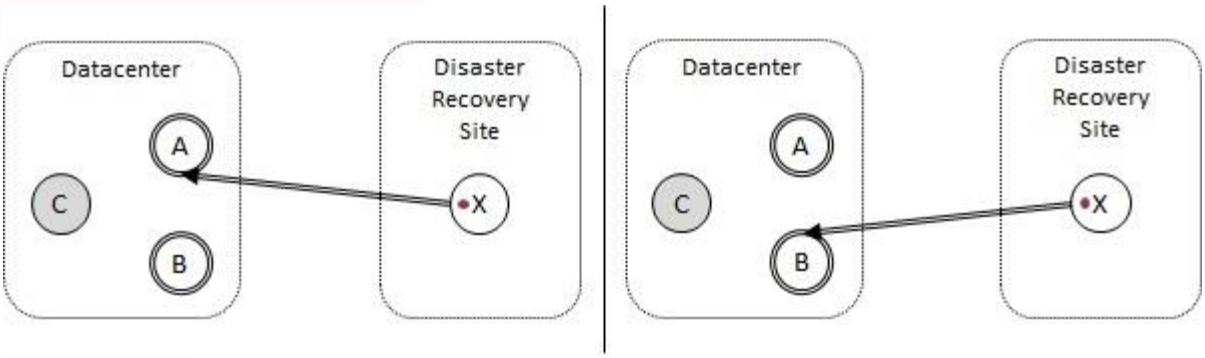


Figure 10. Effects of replication reversal when using distant target configuration

This ensures that the project updates are replicated only once across the network instead of three times. The replication connections in Figure 9 will resume once the node that is receiving the replication updates from node X performs a replication reversal. It may require multiple replication cycles before the replication connections are fully reestablished.

Enhanced Replication: Cascaded Replication

Cascaded replication enables the replication of the package to more than one node in a string of nodes. Figure 11 shows a simple cascaded replication configuration.

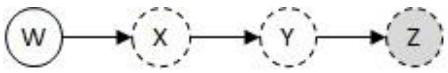


Figure 11. Simple cascaded replication chain

Node W has the defined project and a replication action to node X. Once the replication package is established on node X, a cascading replication action is created on the package on node X to replicate the package to node Y. Once the replication package is established on node Y, a cascading replication action is created on the package on node Y to replicate the package to node Z. Nodes X and Y are shown in dashed circles to indicate that they are performing cascading replication. Node Z is in a shaded, dashed circle to indicate that it is a final (last) target of a cascaded chain. .

Each node along the cascaded chain allows the same operations on the package as targets of basic replication allow. The shares in the package can be exposed for read-only access, or can be snapshot and cloned for temporary ready/write access without affecting the replicating data. The only exception is that a cascaded package does not support replication reverse operations. In the Figure 11 example, only node X can perform a replication reversal since it is directly associated with the source. Nodes Y and Z cannot since they are more than one node removed from the source.

Figure 12 depicts another possible cascaded replication configuration.

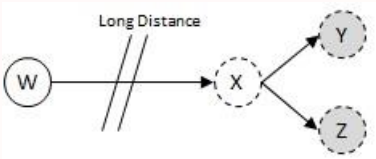


Figure 12. Cascaded chain directed to two final targets

Node W replicates its project to node X, as in Figure 11. In this example, however, two actions are created on the package on node X: one with a cascaded action to node Y and another with a cascaded action to node Z. This configuration can help in controlling costs in network infrastructure and bandwidth. Node W may be in a geographic location distant from the other three nodes, whereas the other three nodes can take advantage of local infrastructure.

As part of its fault tolerant design, cascaded replication also has the ability to bypass a node in the cascaded chain in the event of a node outage. Figure 13 demonstrates the redirection stages to bypass a cascaded replication node.

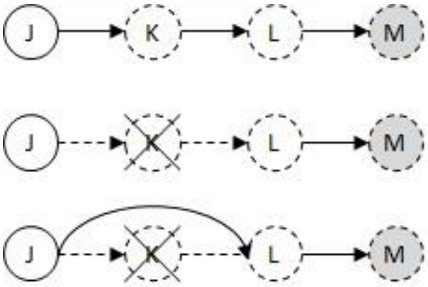


Figure 13. Bypassing a failed node in a cascaded chain.

The top cascaded chain in Figure 13 shows the flow of cascaded package replication during normal operation. The middle cascaded chain demonstrates the outage at node K, which prevents the cascaded package updates to node L. The bottom cascaded chain shows the redirection action which bypasses node K and continues the replication chain to node L, thus resuming package replication to node M. Redirection must be initiated by the storage administrator and does not occur automatically.

Once node K has returned to service, the previous replication action from node J to node K is restored, as shown in the upper diagram of Figure 14.

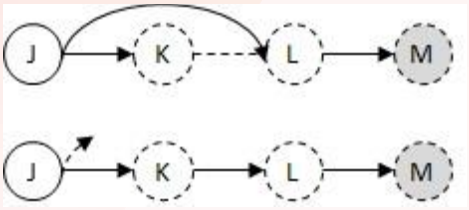


Figure 14. Restoring service to a previously failed node in a cascaded chain

Even though node K is active again, it will not automatically restore the original connections. Another redirect is required where the restore option is selected. The restore will disable the bypass action and fully activate the original cascaded action, as depicted in the bottom cascaded chain in Figure 16. At this point, original service is fully restored.

Combining Enhanced Replication Features

By combining the cascaded replication, multi-target reverse, and distant target features, it is possible to create complex, efficient and fault tolerant replication topologies that traverse global regions. One example is presented in Figure 15.

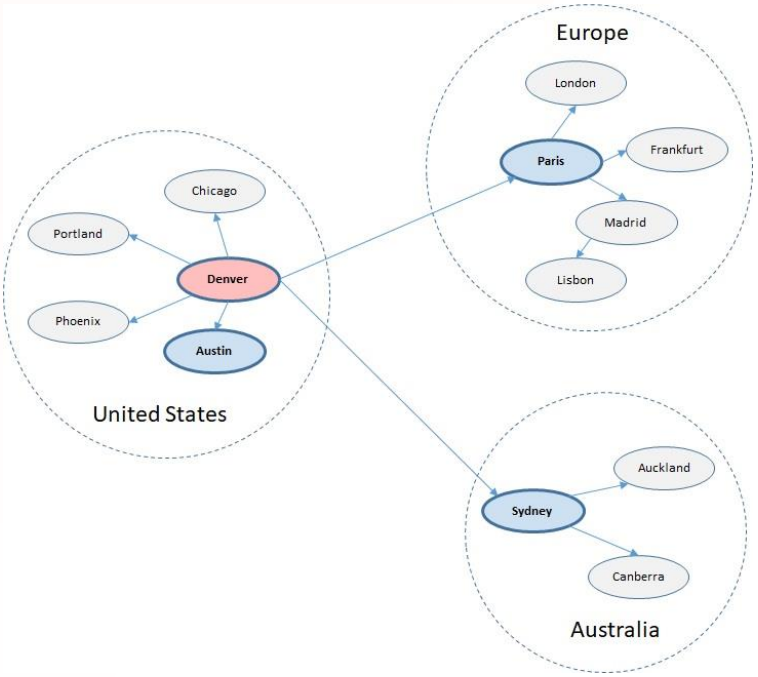


Figure 15. Enhanced replication feature implementation

In the example, Denver has the project source with potential sources located in Austin, Paris, and Sydney. Paris and Sydney cascade the packages to the other direct cities, where Madrid cascades the package even further to Lisbon. Austin acts as a nearby failover site in the case where Denver is unavailable. In that case, Austin will continue to replicate the package overseas and to the United States regional replication targets. The project at the source is read/write/snapshot/clone accessible. All other locations have read/snapshot/clone access to the package.

ENHANCED REPLICATION: DYNAMIC ROLE ACTION SCHEDULES

When using multi-target reverse and cascaded replication together, it is important to understand that there are two types of replication schedules; one is the replication source schedule, which is replicating the currently active project, and the other is the cascaded schedule that further replicates the package to other nodes. Both schedules are part of the same action. When a node becomes the project source, it follows the action of the replication source schedule. Likewise, when a node becomes a package cascading node, it follows the action of the cascaded schedule. These concepts are illustrated with the node design in Figure 16.

Figure 16. Replication action reference architecture

Node Q is the original source of the project. Node Q replicates the project to node P, a dedicated target, and to node R, which is a potential source with distant target enabled. Node R cascades the replication package to node S, a cascaded final target. Nodes Q and R are the only possible source nodes.

When a replication reversal is performed on node R, the roles of each of the nodes change, as shown in Figure 17.

Figure 17. Effects of replication reversal on reference architecture

Node R becomes the project source. Node R replicates the project to node Q, which has become a potential source. It does not replicate the project directly to node P since the relationship between node Q and node R is distant target. Node R replicates the project to node S, since node S has become a dedicated target. Node Q cascades the package to node P, since node P has become a final target.

The actions of nodes Q and R change based on which node is the currently active source. When the node is a source, it replicates the package, but when it is a potential source, it cascades the package. Thus, in order for the replicating or cascading to continue after a replication reversal, nodes Q and R need to have both a replication source schedule and a cascading schedule defined in the same action between node Q to node P, and between node R to node S.

Figure 18 shows the action and schedule breakdown from Figure 16.

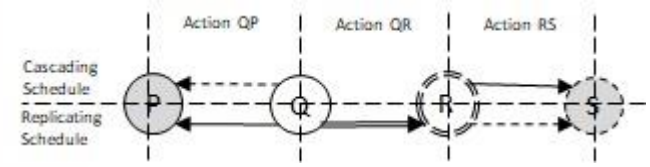


Figure 18. Reference architecture illustrating actions and their schedules

The action between node Q and node P (action QP) has both a replication source schedule and a cascaded schedule. When node Q is the project source, the replication source schedule is active and the cascading schedule is inactive (dashed arrow). The action between node R and node S (action RS) is also comprised of both a replication source schedule and a cascading schedule. When node R is a cascading, potential source, the cascading schedule is active and the replication source schedule is inactive. Since nodes Q and R can only replicate the project between them, the action between node Q and node R (action QR) only has a replication source schedule. When a replication reversal is performed on node R, actions QP and RS switch their active schedules, as shown in Figure 19.

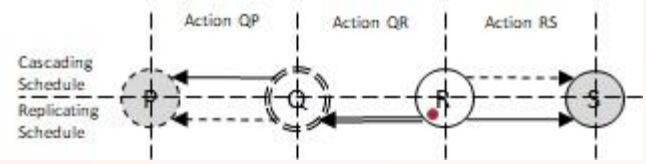


Figure 19. Replication reversal effects on actions and their schedules

Action QP switches to its cascaded schedule and action RS switches to its replication source schedule.

By adding cascaded replication action schedules to the topology provided earlier in Figure 9, the replication reversal events of Node X shown in Figure 10 can continue propagating the replication package to the other nodes, as illustrated in Figure 20.

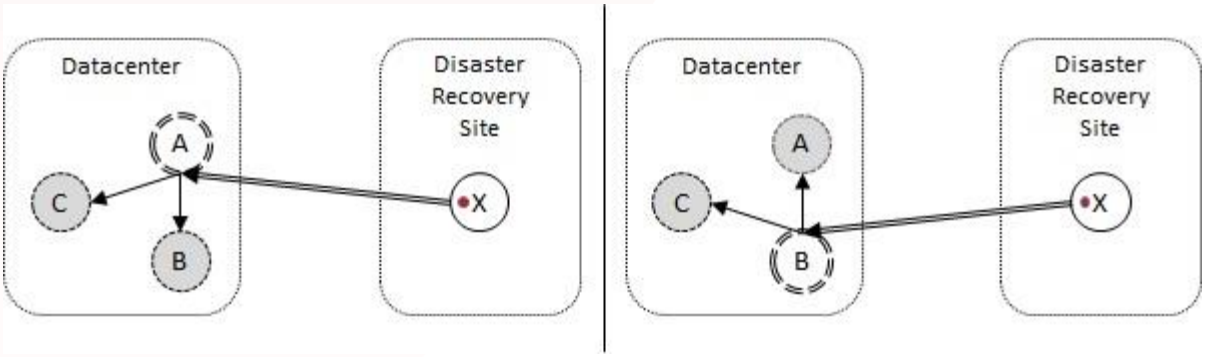


Figure 20. Multi-target reverse example including cascaded replication schedules

In the left diagram, after the replication reversal on node X, node A becomes a cascading potential source and nodes B and C become cascaded final targets. In the right diagram, after the replication reversal on node X, node B becomes a cascading potential source and nodes A and C become cascaded final targets.

Therefore, it is important to consider all of the action transitions between potential sources and pre-configure their appropriate replication schedules. Each source/potential source must account for all replication reversal possibilities and have the necessary schedules in place to maintain continuation of service as the topology changes.

ENHANCED REPLICATION: UNENCRYPTED TO ENCRYPTED DATA REPLICATION

Enhanced replication also includes a deferred update ("Compact File Metadata for Encryption with Enhanced Replication Support") that improves encryption efficiency and adds the ability of replicating an existing unencrypted project to an encrypted pool in order to migrate data from a previously unencrypted environment to a new encrypted environment while maintaining data availability. Previously, basic replication has been able to either replicate unencrypted data to an unencrypted target, or replicate encrypted data to an encrypted target. This enhanced replication feature is provided as a deferred update since the encryption metadata changes can affect existing basic replication of encrypted data configurations. Consult the software release notes to determine how it may affect a configuration and the steps to take before applying the deferred update.

Once the deferred update is applied, the unencrypted to encrypted replication feature can be combined with the other enhanced replication features to convert an existing unencrypted replication environment to an encrypted replication environment. The following sections provide two examples of converting unencrypted replication environments (node A to node B) to new encrypted replication environments (node E to node F). The new encrypted environments may be hosted by the same Oracle ZFS Storage Appliance systems as the unencrypted environments. However, the new encrypted environments must be on encrypted disk pools.

Even though the feature allows unencrypted source to encrypted target replication, it does not allow encrypted source to unencrypted target replication, thus replication reversals are not completed in these configurations.

Unencrypted to Encrypted Replication Conversion Example 1 (Limited Resources)

This replication conversion method should be considered when available replication resources are low (e.g. network bandwidth and/or available CPU), or when the encrypted replication target may be in a different geography. It provides for initial encrypted replication source creation using multi-target reverse groups and seeds the encrypted replication target using cascaded replication.

This example starts with a basic replication configuration where unencrypted project source A is replicated to unencrypted project target B, as shown in Figure 21.

Figure 21. Original unencrypted replication configuration for Example 1

The first step toward achieving encrypted replication is to configure node B as a potential source from node A. This maintains the node A to node B failover capability in a multi-target reversal environment. The next steps are to configure encrypted node E as a potential source with the distant target attribute set on node A, and initiate the replication to node E, as shown in Figure 22.

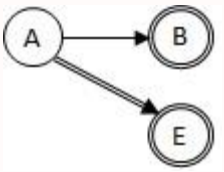


Figure 22. Configuring node B and node E as potential sources for Example 1

The distant target attribute is used with node E in order to maintain the node AB relationship when this overall procedure is completed. In this current configuration, an outage on node A would turn node B into the new source and continues the initial replication seeding to node E.

Once the initial target replication is complete on node E, encrypted node F can be added and configured to node E as a cascaded target, as shown in Figure 23.

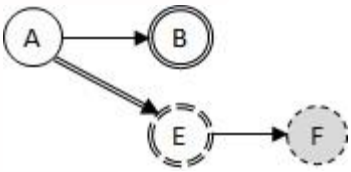


Figure 23. Cascading node E replication to node F for Example 1

Once replication synchronization has completed on node F, a replication reversal can be performed on node E, as shown in Figure 24.

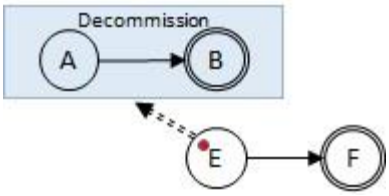


Figure 24. Completed encrypted replication for Example 1

The replication reversal converts node E into the project source. Node F becomes a potential source. Because of the distant target relationship between node E and node A, node E will attempt to perform the reversal with node A, but will fail since an encrypted source cannot replicate to an unencrypted target. The node E to node A relationship will need to be removed manually. Node A will still be configured as a source and will continue replicating to node B, but attempts to replicate to node E will fail. At this point, node E should be configured to all applications as the new source. Nodes A and B can be decommissioned.

Unencrypted to Encrypted Replication Conversion Example 2 (Quickest Conversion)

This replication conversion method should be considered when replication resources are readily available and time is of the essence. It creates the encrypted replication source and target simultaneously using multi-target reverse groups.

As with Example 1, Example 2 starts with a preexisting unencrypted basic replication configuration of source node A replicating to target node B, as shown in Figure 25.

Figure 25. Original unencrypted replication configuration for Example 2

The first step is the same as Example 1 where node B is indicated as a potential source from node A. In Example 2, however, both encrypted nodes E and F are introduced at the same time, each configured as potential sources and with the distant target attribute selected, as shown in Figure 26.

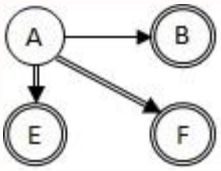


Figure 26. Configuring nodes B, E, and F as potential sources of node A in Example 2

As before, if a failure event is encountered on node A, node B can become the source and continue the replication seeding to nodes E and F. Once initial replication synchronization is complete on both nodes E and F, a replication reversal can be initiated on node E, as shown in Figure 27.

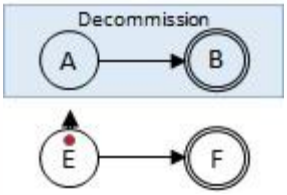


Figure 27. Completed encrypted replication configuration for Example 2

The replication reversal converts node E into the project source. Node F remains a potential source but now has node E as the package source. Because of the distant target relationship between node E and node A, node E will attempt to perform the reversal with node A, but will fail since an encrypted source cannot replicate to an unencrypted target. The node E to node A relationship will need to be removed manually. Node A will still be configured as a source and will continue replicating to node B, but attempts to replicate to nodes E and F will fail. At this point, node E should be configured to all applications as the new source. Nodes A and B can be decommissioned.

MONITORING REPLICATION

Monitoring Recovery Point Objective Settings

For each data set to be replicated, an RPO can be set via its related replication action. The RPO, expressed in time, should be derived from a wider customer disaster recovery or business continuity plan in which RPOs are specified for business processes using the data repositories by the related business applications.

An RPO target for a data set specifies the maximum time a copy data set is allowed to lag behind the source data set at the time of a disaster, that is, loss (of access) to the primary dataset.

The RPO target is specified in the replication action for a specific data set. The RPO target can be monitored by specifying a warning and error threshold level. A related warning/error alert is issued by the Oracle ZFS Storage Appliance system when the actual replication target time lag crosses the set alert threshold. The actual replica time lag can be monitored in real time via the command-line interface (CLI) and RESTful API scripting interface.

Monitoring and Analyzing Replication Actions

After a replication action is created and the associated replication is scheduled, the replication progress can be monitored. Its progress is shown in the BUI under the replication rule, and it shows percentage of the data replicated, the replication throughput, and the estimated remaining time to completion. The same information can also be determined using the CLI.

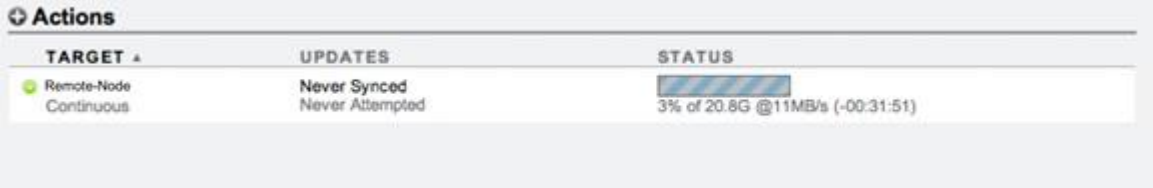


Figure 28. Replication progress monitoring

Replication performance can be investigated in further detail using specific replication analytics statistics available under “Add statistic...” in the analytics worksheet window.

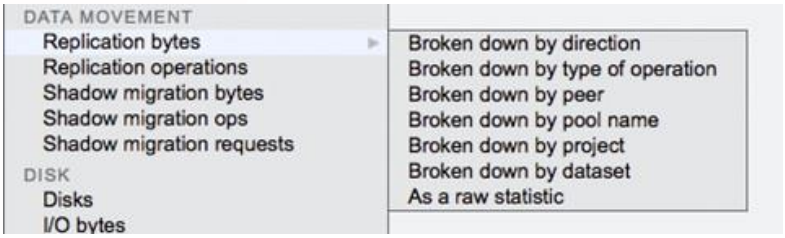


Figure 29. Replication analytics worksheet statistics

Each statistic can be broken down by direction, by type of operation, by peer, by pool name, by project, or by data set.

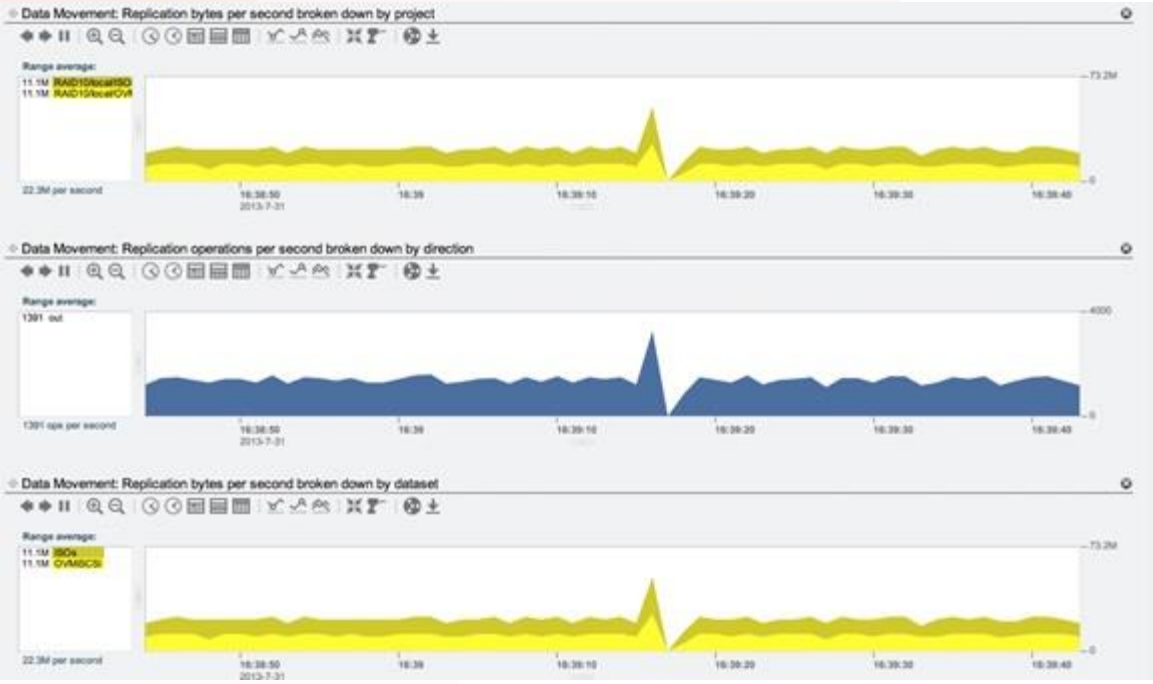


Figure 30. Replication analytics worksheet example

Replication Event Logging and Action Auditing

Replication events are logged in the event log. These events can be monitored via email messages, sending SNMP information, or using the remote syslog mechanism. Replication events also can be configured to execute a specific workflow in Oracle ZFS Storage Appliance.

The Oracle ZFS Storage Appliance system posts alerts for the category remote replication when any of the following replication-related events occur:

- Manual or scheduled replication update starts or finishes successfully (both source and target).
- When any replication update or continuous replication fails, including as a result of explicit cancellation by an administrator (both source and target). The event includes the reason of failure.
- A scheduled replication update is skipped because another update for the same action is still in progress.
- When a continuous replication starts for the first time, fails, stalls, or resumes after a failure.
- A replica time lag exceeds its specified RPO threshold.

The following replication configuration actions are logged in the auditing log so that creation and deletion of all actions and changes to actions can be tracked. For replication actions this includes:

- Creating, modifying, or destroying replication actions
- Adding or removing shares from a replication group
- Creating, modifying, cloning, reversing, severing, or destroying replication packages on the target
- Creating, modifying, or destroying replication targets

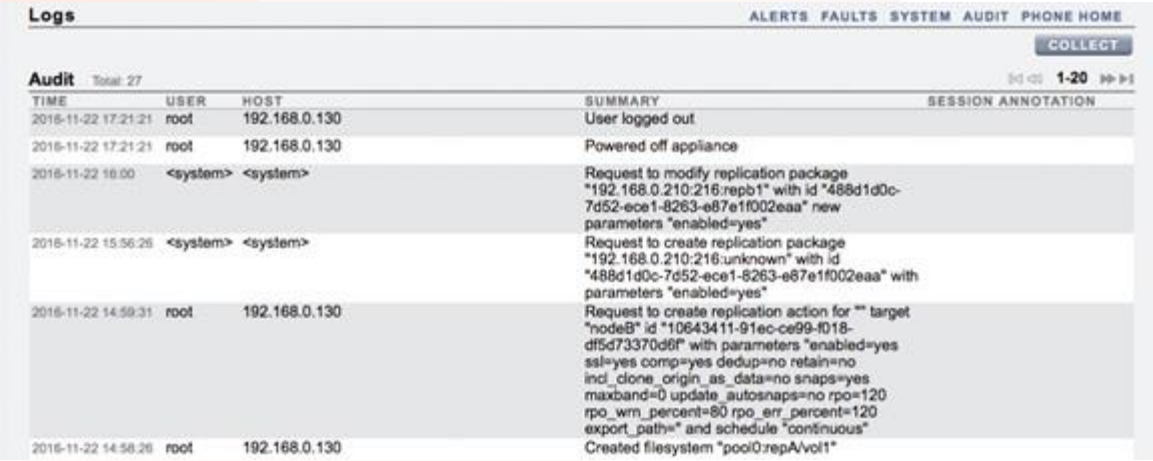


Figure 31. Replication actions audit log messages

GENERAL IMPLEMENTATION GUIDELINES FOR REPLICATION

The following general guidelines and considerations are for administrators who are applying replication capabilities in the Oracle ZFS Storage Appliance environment.

Oracle ZFS Storage Appliance products use asynchronous communication between the source and the targets to ensure that network latency does not slow production operations. This technology cannot guarantee that updates to the source will be present at the target site after a loss of the source site; however, the image of the project at the target site is guaranteed to be write-order consistent as of the time of the most recently completed data transfer.

Configuring a Replication Mode

An administrator of Oracle ZFS Storage Appliance systems can configure or change the replication mode among continuous, scheduled, and on demand at any time. The administrator also can configure different replication modes for the same target when using multiple replication actions.

Choosing the optimal configuration depends on the technical and operational requirements of the specific implementation. With regard to the replication mode, the administrator should consider the following details:

- Recovery point objective (RPO)
- Recovery time objective (RTO)
- Available bandwidth between the source and target sites and related to this, the rate of change (ROC) of data on the data set to be replicated
- Balancing available CPU and memory resources for the various data functions within the Oracle ZFS Storage Appliance system

Scheduled replication mode makes the best use of available resources. As long as the chosen scheduled replication interval is big enough to be able to send the changed data to the secondary site, a stable RPO level can be expected in this mode. With scheduled replication, the Oracle ZFS Storage Appliance source periodically replicates a point-in-time image (snapshot) of the source project to the target. This reduces network traffic while preserving consistent and timely copies of the primary data set. With the option to set the replication interval as low as five minutes, low, predictable RPO values can be maintained without the need for continuous replication.

Continuous replication mode of the project is an appropriate choice for technical and operational requirements that require near-real-time protection of data at the remote site, such as RPO and RTO of less than a few minutes. The achievable RPO target very much depends on the ROC of the data set to be replicated. Updates to the source data set will be sent to the target site as fast as the network permits in this case. The tradeoff to be made is the continuous use of CPU, memory, and network resources when using continuous replication versus scheduled replication mode. Continuous replication mode is not supported with cascaded replication.

On-demand replication mode is designed for applications that need to put data into a specific state before the replication can occur. For example, a replica of a cold or suspended database can be produced every time the database is shut down by integrating a call to trigger an on-demand replication update in the database shutdown or suspend scripts. On-demand replication updates can be triggered from arbitrary locations in the application-processing stack through the RESTful API automated scripting language of the Oracle ZFS Storage Appliance CLI.

Replication Network Setup

When there are enough network ports available on the Oracle ZFS Storage Appliance controller, it makes sense to dedicate a specific port for replication traffic. This is to ensure that replication data to the target IP uses that specific port on the Oracle ZFS Storage Appliance system. An entry in the routing table can be added using a specific /32 static route to the target's IP address over the replication source interface. After the entry in the routing table is added, the replication rule can be set up for the target IP address.

A clustered configuration ensures the replication for the specific network interface is migrated to the other node in the event of a node failure or scheduled maintenance activities. If no static route is defined to the target's IP address, replication traffic may be delivered over a private interface, such as an administrative interface, that becomes unavailable in the event of a takeover within the cluster.

The Oracle ZFS Storage Appliance DTrace Analytics feature can be used to verify that the replication data traffic is using the intended interface on the source Oracle ZFS Storage Appliance system.

Of course, the same setup needs to be repeated on the target Oracle ZFS Storage Appliance systems that will be used for basic replication and enhanced replication features, so when a replication role reversal is executed, the target nodes also continue to use their dedicated replication network ports.

Performing the Initial Seeding of a Target Node

When setting up a replication configuration, the initial replication of the data set can take a long time when the available bandwidth between the locations is limited. To overcome this problem, the initial replication synchronization can be done by locating source and target Oracle ZFS Storage Appliance systems next to each other. Once data is synchronized, the system intended for the remote site can be shipped to its location. Once this Oracle ZFS Storage Appliance system is connected to the network, its new target IP address can be changed in the source Oracle ZFS Storage Appliance system.

A second option is to use the offline replication capability when creating a replication action. With this option, the initial replication data stream is stored on an NFS share as specified under Export Data Path from a local available system. All other replication action properties can be specified as required.

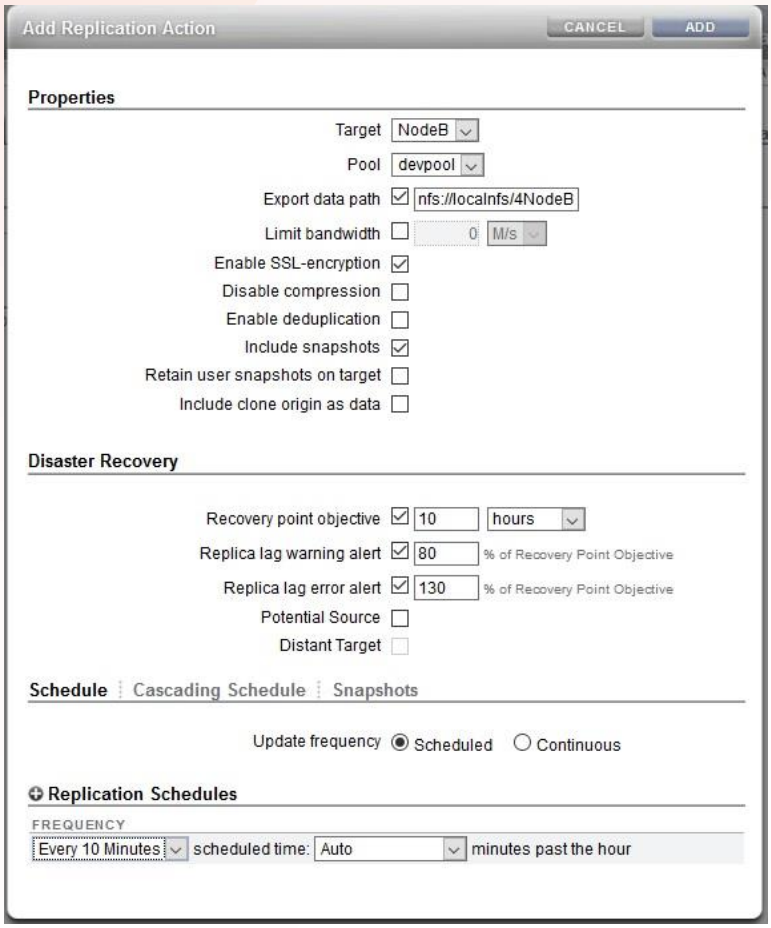


Figure 32. Initial replication seeding by exporting to NFS share

When the export replication action is finished, the system with the NFS share can be moved to the remote site or the replication data can be copied from the NFS share to a transportable media like tape. When setting up a replication action, the initial replication data stream will be stored on a local system's NFS share (visible in the following figure in Import Data Path). As part of the initial replication action, the source node already informs the target node of the replication action setup of the share/project. On the node on the secondary site, the external replication data set then can be imported via NFS from a system that contains the shipped copy of the external replication data set using the "Import update from external media" option.

Figure 33. Import replication set from NFS share

Enhanced Replication Initial Configuration and Modification

It is important to design the enhanced replication topology based on current business needs, fault tolerance, and data communication costs. However, the initial design is not locked in place. The design can be modified as the business needs change. Another potential source or cascaded node can be easily added by creating a new action to that node. Likewise, the topology can shrink by permanently removing a node without affecting the overall operation. All configuration changes are included in the replication packages so that all affected nodes will be aware of the changes within one or two replication updates.

The orchestrated growing and shrinking of the topology is particularly advantageous when performing a node technology refresh at a site. A new Oracle ZFS Storage Appliance system can be introduced into the topology, populated with the replication package, and brought in to active service. The older system can then be removed from the replication topology and taken out of service.

One key element to the overall configuration, though, is ensuring that the remote replication targets are configured correctly under the Services section of the Web BUI. Each node that can potentially redirect its replication role must have all of the possible replication targets defined in advance of the replication reversal or cascaded node bypass operation. If these replication targets are not predefined on each node, then the replication reversal or bypass operation may lead to an unbound condition. The unbound condition essentially indicates “action does not have a target”, and is therefore unusable. If encountered, though, it can be corrected.

Using the Intelligent Replication Compression Feature

Intelligent Replication Compression is enabled by default. When upgrading Oracle ZFS Storage Appliance systems to a firmware level that includes Intelligent Replication Compression, and replication actions are already present, these replication actions will have Intelligent Replication Compression enabled as part of the firmware upgrade process.

The Intelligent Replication Compression feature uses an algorithm to dynamically adjust the level of compression and the number of concurrent replication threads, depending on system load and replication network performance of the Oracle ZFS Storage Appliance system.

In general, there is no need to disable Intelligent Replication Compression for any replication action. It is advised not to use Intelligent Replication Compression for the following exceptional cases, where:

- Source data sets to be used for replication are not compressible at all.
- CPU resources are already heavily utilized by other services in Oracle ZFS Storage Appliance.
- The replication network is extremely fast and dedicated to replication only, so there is no need to save network bandwidth used.
- A WAN accelerator that performs compression is already in use on the replication link.

Be careful not to enable SSL on the Oracle ZFS Storage Appliance system when using a WAN accelerator, as enabling SSL substantially degrades replication throughput in the WAN accelerator.

The benefit to using the compression function of the Oracle ZFS Storage Appliance system is that it will increase the replication throughput as the SSL encryption is performed against compressed data.

A single controller supports up to 120 parallel running replication actions (the sum of all incoming and outgoing replication updates). When a large number of replication actions are active, systems (both at source and at target) should be carefully monitored for CPU resource usage. When CPU utilization passes 90 percent, adding more replication actions that could run in parallel with already existing replication actions—whether they are due to overlapping schedules or the number of continuous replication actions running in parallel—is not recommended. There should be no more than 120 replication actions (threads) running at the same time.

For situations where data can be further compressed by using deduplication, such as VDU-type data images or Oracle Recovery Manager (Oracle RMAN) backup images, the replication deduplication over-the-wire option can be used. However, this option should be used carefully as it requires extra CPU and memory resources to deal with the deduplication process in the Oracle ZFS Storage Appliance system. The efficiency of the deduplication on a specific replication stream can be monitored by looking at the project messages in the BUI Logs > Alerts page that begin with “Finished replicating…”

Figure 34. Deduplication statistics in replication finished alert logs

Project-Level Compared to Share-Level Replication

Basic replication can be performed on both the project level and individual share level. Project-level share replication is the preferred option for a number of reasons:

- **Data consistency**. A project acts as a data consistency group for all shares in the project when they are replicated with a project replication action. This is very important when data transactions are using multiple shares and integrity, between the shares for these transactions, needs to be guaranteed.
- **Limiting the number of snapshots and disk space overhead**. Replication snapshots are always taken on a project level. When replication actions are created on individual shares, more snapshots are likely to be created, resulting in higher capacity consumption on the pool.
- **Reversing replication**. When replication is reversed and shares are replicated individually, the share is placed in its own project, resulting in fragmentation of the original project and loss of the share's property inheritance relationship with its project.

Using the Deduplicated Replication Feature

The Deduplicated Replication feature in the Oracle ZFS Storage Appliance product provides the ability to reduce the amount of data sent over the replication connection to the target node, increasing network efficiency. This feature is different and independent of the appliance on-disk data deduplication feature. On-disk data deduplication is chosen on the project or share level and aims to eliminate duplicated blocks of data stored on disk. When a project or share that has the data deduplication feature enabled, is replicated, its data is deduplicated before being send to the secondary node. This process is independent from the use of the Deduplicated Replication feature.

As with the on-disk data deduplication feature, the Deduplication Replication feature has to be used with caution. This feature requires CPU resources for data preprocessing and memory resources for lookup tables. The amount of these resources needed is also dependent on certain share property settings. Furthermore, the effectiveness of the deduplication is very much dependent on the type of data sets used. The benefits of using Deduplicated Replication has a specific narrow sweet spot use case. The following basic requirements define the sweet spot environment:

- Data should have sufficient duplicates, such as weekly full RMAN images or VDI images.
- The replication connection between the nodes uses a low-bandwidth, high-delay network.
- The checksum property used for the share/project is set to either SHA-256 or SHA-256-MAC and if not, on-disk data deduplication should be set. The Deduplication Replication feature uses crypto-type check summing. When they already have been generated at the time of storing data, they don't need to be recreated. When using on-disk data deduplication, strong type checksums are already provided.
- When encryption is used, a deduplication 'friendly' encryption mode must be set: GCM type encryption never creates repeatable data patterns so deduplication can never be effective here. CCM type encryption should be used and on-disk data deduplication must be switched on.

- Do not use small on-disk block sizes: When small block sizes are used, a larger number of blocks need to be processed, it takes longer to build deduplication tables and tables are much larger with the risk that there is not enough memory available to build the tables needed. Also when there are relatively small updates to the data set, the time it takes to build the tables outweigh the benefits of time gained in the transfer process. It is recommended to use disk block sizes of 1MB for data sets to be used for the Deduplication Replication feature.

If these conditions are not met, the replication transfers are slower and bulkier but will still work.

The effectiveness of the Deduplication Replication feature on data sets can be monitored. The appliance shows a summary of the deduplicated statistics in the event log under alerts.

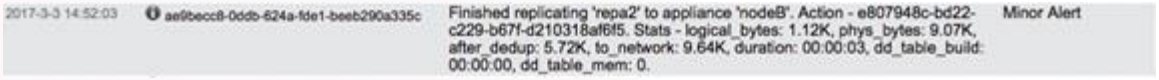


Figure 35. Deduplication replication event reporting example

More detailed information can be obtained via the CLI interface using the ‘stats’ node of the replication action for a share:

```
nodeA:shares repa2/vola2-1 action-000> stats

nodeA:shares repa2/vola2-1 action-000 stats> ls

Properties:

        replica_data_timestamp = Fri Mar 03 2017 15:02:00 GMT+0000 (UTC)

                     last_sync = Fri Mar 03 2017 15:02:03 GMT+0000 (UTC)

                      last_try = Fri Mar 03 2017 15:02:03 GMT+0000 (UTC)

                   last_result = success

            last_logical_bytes = 1.125K

               last_phys_bytes = 9.06640625K

              last_after_dedup = 5.72265625K

               last_to_network = 9.68261719K

                 last_duration = 00:00:03

           last_dd_table_build = 00:00:00

             last_dd_table_mem = 0

                 total_updates = 91

           total_logical_bytes = 606.117188K

              total_phys_bytes = 865.257813K

             total_after_dedup = 511.277344K

              total_to_network = 886.063477K

                total_duration = 00:05:37
```

```
        dd_total_updates = 27
dd_total_logical_bytes = 30.375K
   dd_total_phys_bytes = 244.816406K
  dd_total_after_dedup = 154.226563K
   dd_total_to_network = 260.385742K
     dd_total_duration = 00:01:39
  dd_total_table_build = 00:00:00
    dd_total_table_mem = 0
```

Determine the Replication Snapshot Frequency of Scheduled Replication Actions

To determine the appropriate replication snapshot frequency, both the recovery point objective (RPO) and the rate of change (ROC) of the data repository to be replicated need to be defined. An RPO is one of the key elements in a business continuity plan, specifying the amount of transactions or data loss that can be tolerated by the business in case of a disaster impacting the primary storage repository. The RPO specifies how far back in time a snapshot of the data repository is available at an alternative location. Equally important is knowledge of the ROC of data in the primary data repository. The ROC determines the minimum required transmission speed of the replication link between the primary and secondary site and, to some degree, the replication interval. Both the RPO and ROC determine the frequency of a scheduled replication action for a specific data repository or project in Oracle ZFS Storage Appliance context. For the ROC, it defines the amount of data changed between two replication intervals and the requirement that that data must be transmitted to the secondary location within the subsequent snapshot intervals.

For the RPO, the recommendation is to use a replication frequency of one-half the required RPO.

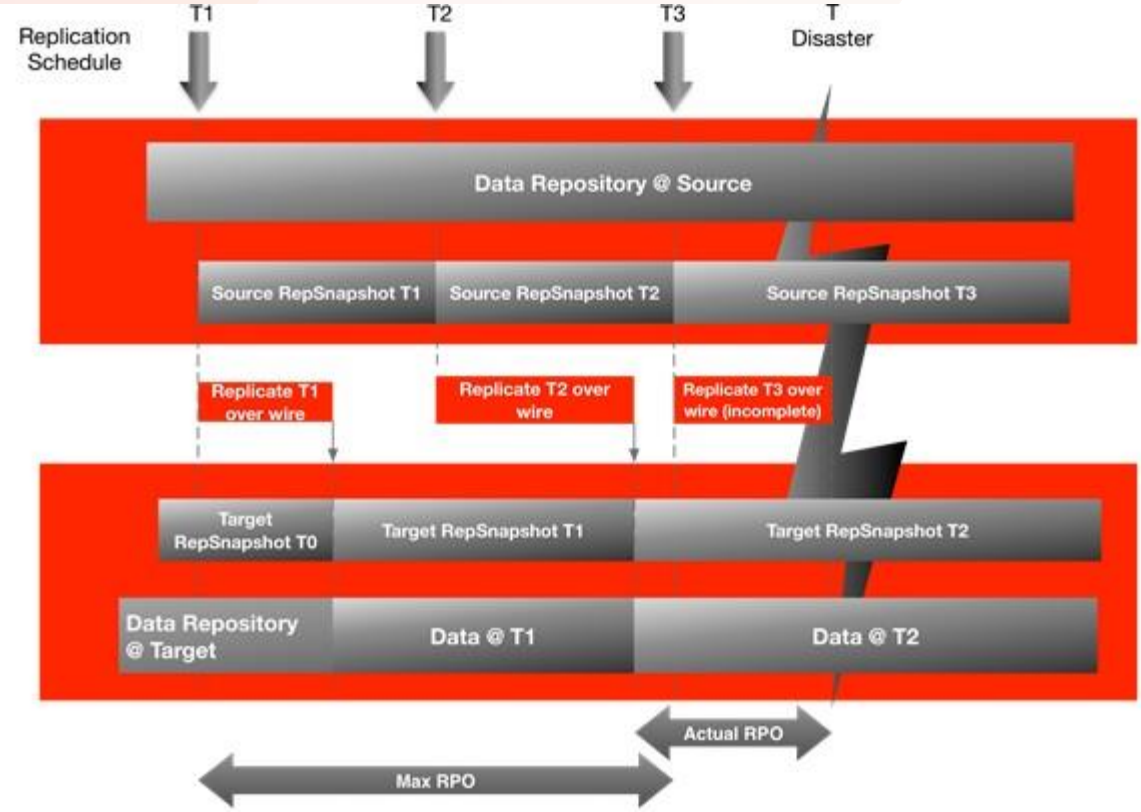


Figure 36. RPO and snapshot frequency relationship

The worst case RPO situation occurs when a replication snapshot contains almost no changed data followed by a snapshot that almost takes the whole time slot between two replication actions. The resulting RPO is twice the snapshot time interval, based on the assumption that the replication of a snapshot will always finish before the next snapshot is taken. When it takes longer for a replication snapshot to be transmitted, the next replication snapshot action is deferred until the in-progress update completes.

When using continuous replication mode, the replication interval is directly determined by the ROC of the related data repository to be replicated. If the ROC is not constant, the replication frequency will not be constant and subsequently the recovery point will vary, too.

When replicating multiple data repositories to a secondary site, the bandwidth requirements of the data connection between the sites are determined by the total ROC of all data repositories.

When replicating compressed data repositories, the actual replicated data will be less than the actual “logical” ROC of the data in the repository. The Oracle ZFS Storage Appliance analytics function can be used to determine the physical ROC on the actual disks as a measure to determine the required bandwidth for replicating data between the primary and secondary site.

Setting Up Required RPO in Replication Action

For each replication action an RPO can be set along with the related warning and error level threshold.

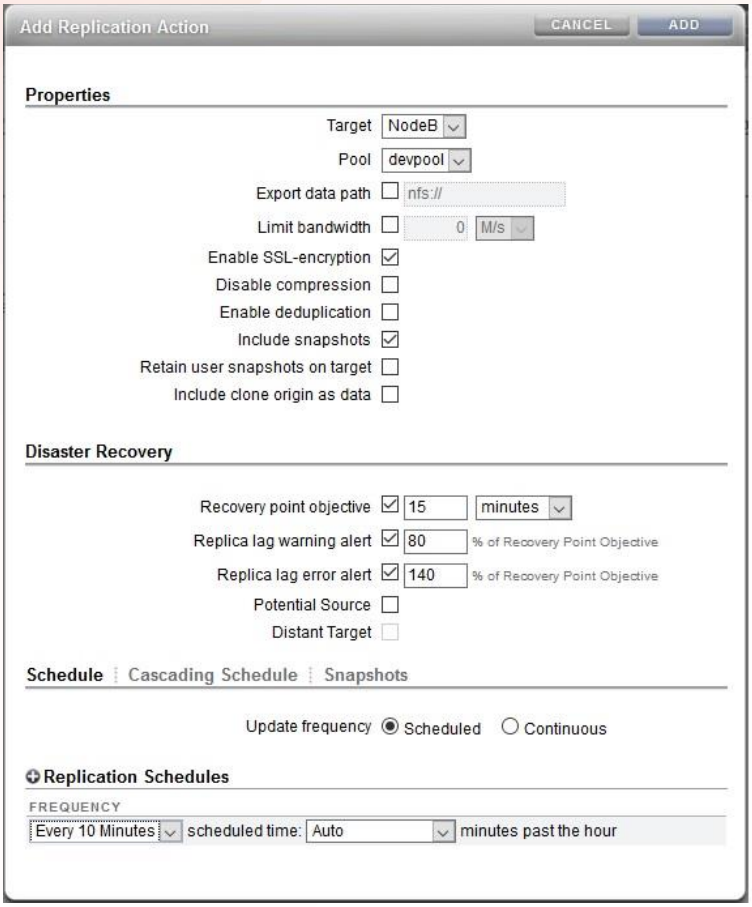


Figure 37. Specifying RPO and RPO monitoring thresholds

Scheduled replication actions are the preferred option. They use fewer resources (such as internal CPU and network processing resources). Another benefit of using scheduled replication is the more predictable RPO level. With the option of setting replication frequency intervals as low as five minutes, low RPO values can be achieved that are close to those of the RPO levels of the continuous replication mode.

How to Configure Replication Alert Events Monitoring

An Oracle ZFS Storage Appliance system contains an alerting system that collects and manages all event and threshold alerts occurring within the system. Oracle ZFS Storage Appliance has a very flexible mechanism for managing the way the system reacts to these alert messages. Threshold alerts can be defined based on analytics metrics like, for instance; capacity percentage threshold used for a specific pool.

For each category of alerts, alert actions can be defined. Alert actions are used to define the type of action to be taken by the system. Alert actions are added via the Configuration > Alerts menu.

Figure 38. Add alert action dialog

The following alert actions are available:

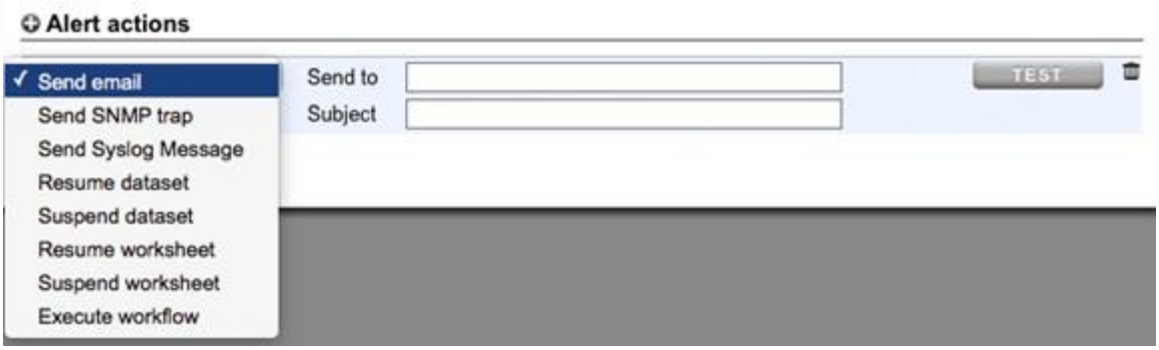


Figure 39. Type of event alert actions

Note that for the SNMP and syslog alert action options, their respective service has to be defined and started via the Configuration > Services section. In a windows environment, the Microsoft Management Console (MMC) can be used to access the event logs. The Oracle ZFS Storage Appliance Administration Guide (also available via the BUI online help) under SMB MMC Integration provides further details.

Three types of event categories are available for the replication function:

- Remote replication, source only
- Remote replication, target only
- Remote replication (which contains both source and target events)

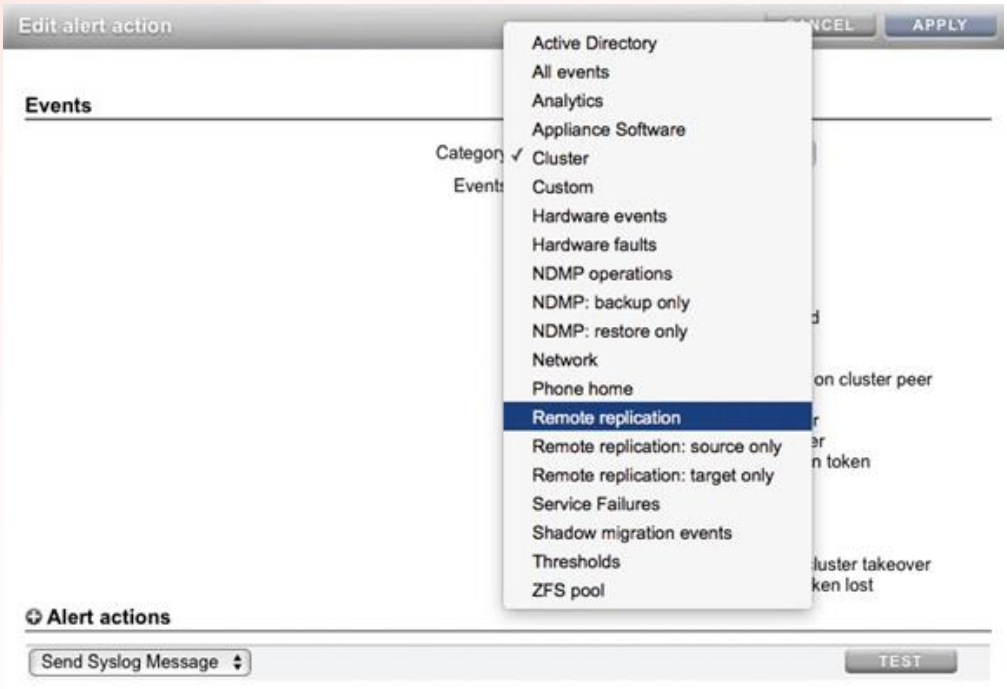


Figure 40. Available event categories via add or edit alert action option

The replication events category contains detailed replication events that can be selected to be included in the alert action messages.

Figure 41. Replication events for alert actions

Replicating Between Nodes with Different Software Release Versions

Replication is compatible between most Oracle ZFS Storage Appliance software releases. Compatibility issues occur when different releases support different ZFS pool features that are incompatible, and those features that are applied to the node by executing the deferred updates function after a software upgrade procedure. Also, issues occur when newer releases contain new replication options and features that may not be supported in combination with earlier releases. More information about compatibility and deferred update features can be found in the Oracle ZFS Storage Appliance Remote Replication Compatibility document (Doc ID 1958039.1) on My Oracle Support https://my.oracle.support .

Replicating Encrypted Shares

The software on both source and target nodes must support the Oracle ZFS Storage Appliance encryption function. Before starting the replication action for an encrypted project or share, the encryption keystore must be set up at both nodes. When using Oracle Key Manager, both nodes must point to the same Oracle Key Manager server and the same wrapping keys must be set up. When using a local keystore, both nodes must use the same wrapping key names. Failure to do so will result in replication failure alerts in the Oracle ZFS Storage Appliance log system available under Maintenance > LOGS in the BUI.

Use at the Target Site

The replica at the target site contains all the information present at the source site as of the most recently completed replication transaction. The replica can be used as a direct substitute for the original source in the event that the source is lost, or clones of the replica can be used for additional business processing, such as backup, test, or analysis.

ACCESSING READ-ONLY COPY

The replicated package can be mounted and can be used as a read-only source from the client. Setting the Export option for the replicated project makes the project and its shares mountable from the client. While this mode is not suitable for database access, the data can be used for backup and other read-only purposes.

ROLE REVERSAL (FOR PRODUCTION FAILOVER)

The client application can be failed over to the target site in the event of a primary site disaster, or during a maintenance operation at the primary site. The target becomes the “production” site in this mode. Only the data that is changed after the failover is replicated back when the primary site is back up again. If the original primary site is set to become the production site again, the role-reversal operation is performed at the new target. It is strongly recommended to disable the replication before initiating this operation. This is the most preferred operation during failover and failback operations for the databases.

If a replication action is not disabled before the reversal process is started, the replication action at the old source will be automatically disabled when it attempts to send the next update to the new source.

TEST, BACKUP, AND REPORTING

To access the replicated package for read/write purposes, a package is cloned. This operation essentially clones the project that is received. Clones of the replication target can be used for test, reporting, and any other type of business processing.

Cloning the replication target efficiently uses storage resources because space allocation for clones is performed only when updates to the primary or cloned data set are performed.

Clones can be recloned and deployed to these different applications in cases where multiple applications need access to copies of the same data set.

Alternatively, many clones from the same replica can be created. Space utilization is further enhanced in this case because the read-optimized flash devices in the L2ARC store a copy of the on-disk image, which is only the original data set plus the updates to the data set.

Multiple test applications can take advantage of the performance of read-optimized flash devices without requiring large amounts of read-optimized flash storage in cases where few updates are performed on the data set. This provides a high-performance, cost-effective solution for using replication resources to perform value-added business functions.

Snapshot Management for Replication

Snapshots can be generated for each project, filesystem, or LUN. These snapshots are instant and offer a frozen read-only view of a project, filesystem, or LUN at the time of the snapshot. Snapshots can be taken by a user (manually) or by a scheduled action. Scheduled snapshots are taken at regular intervals and can include how many snapshots to retain for a specific project, filesystem, or LUN.

Whenever in a replication action the "Include snapshots" property is set, snapshots created via a schedule or manually, will be replicated from the source site to the target site.

Scheduled snapshots of projects, filesystems, or LUNs have an associated retention policy property. This retention policy can be applied to the target site, too, or can be made independent from the source by specifying a different retention policy for the target site in the replication action for the project, filesystem, or LUN. When a replication role reversal takes place, only the snapshots that were originally present on the source site are replicated back and not any that had been deleted at the source site before the replication role reversal took place.

User-created snapshots are always replicated from the source site to the target site. When deleting a snapshot from the source site, the default behavior is that it will also be deleted from the target site. However, there is an option in the replication action to prevent this, so a snapshot at the target will remain there until the user deletes the snapshot. This makes using replication for remote back-up solutions more flexible and gives the user control over the number of snapshots kept at the target independent of the number of snapshots at the source and deletion of snapshots on the source.

Replicating clones from the source site to a target site takes a bit more consideration. Clones are created from a snapshot of a filesystem or a LUN at the source site (this type of cloning is different from cloning a replica at the target). They are thin clones, meaning the data that is not changed is shared between the clone and its originating share or LUN. Any data written to them is stored in the clone instance only. Clones are often used as a quick way to create copies from data repositories for testing or to serve as a seed for a new instance of a fresh installation.

In some cases, clones are put in a different project as the origin and, as such, a different replication action for the clone needs to be created. In such cases, replicating the data in the clone instance itself is not enough for the clone to be useable at the target site. So, the originated filesystem also must be present at the target site.

There are two ways to accomplish this. Either replicate the clone origin first before replicating the project containing the clone, or use the "Include clone origin as data" property in the replication action for the project containing the clone.

When putting clones in a different project as the origin, there is a convenience/space trade-off to be made:

- If the clone origin is replicated before the clone, on the target the relationship between the origin and clone will remain intact and thus the clone will remain a thin clone.
- When using the automatic inclusion option by specifying the "Include clone origin as data" property in the replication action, the origin-clone relationship is lost at the target when the clone is located in a different project as the origin and the clone contains a full copy of the origin of the clone at the source.

Therefore, from a replication perspective, the best practice is to keep clones within the same project as the origin.

APPLICATION SPECIFIC IMPLEMENTATION GUIDELINES

This section augments the general guidelines presented in the previous section. The principles described in this section can be applied to other applications when the behavior of the application is similar to that of the applications described.

Databases

Database replication is accomplished in one of three architectures:

- Full database replication
- Partial database replication
- Logs-only replication

In full database replication, all the files associated with the database are placed in the same project and the project is replicated. This method provides a write-order consistent image of the database as of the last successful replication. Full database replication generally provides the shortest recovery time and simplest process after a failure.

Partial database replication can be set up when not all components of a database are required for disaster recovery (DR) operations. In partial database replication, the database is split into two or more projects: the first project contains all the database files required for DR operations, and the additional projects contain files that are not required for DR operations. For example, temporary data or read-only files can be reconstructed after a failure, so replicating that data is not required.

Logs-only replication is accomplished by replicating only the database's log stream and applying the log stream to a previously shipped copy of the database. This technique makes the most prudent use of network resources; however, it typically leads to longer recovery time and a more complex recovery process.

FULL DATABASE REPLICATION

The simplest implementation method for database replication is full database replication. All files for the database are stored in the same project, and the entire project is replicated from the source to the target in this method. Recovery at the remote site is accomplished through traditional crash recovery methods that are used after a power-fault of the database server, because write-order is preserved throughout the project. The database will start up as though the crash occurred slightly earlier in time in cases where the replication data has not made it to the remote site.

With full database replication, database files can be distributed over multiple shares within a project for optimal deployments. A practical database deployment can include the following shares:

- **Redo logs**: a 128-kilobyte record size share for storing online redo logs and control files
- **Datafiles**: a share configured with a record size that matches the database block size for storing database data files
- **Indexes**: a share configured with a record size that matches the index block size for storing database index files
- **Temp**: a share configured with a record size to match the database sort area page size
- **Recovery**: a share configured with a 128-kilobyte record size and compression to store copies of the redo logs, backup sets of the database, and any other recovery information
- **Unstructured**: a dynamic record size share to store unstructured data associated with the database

Any replication mode (continuous, scheduled, or on demand) can be used with full database replication. If the replication target will be integrated into a backup system, then on-demand replication is required if the database needs to be in a backup state (for example, Oracle hot backup mode) to take a valid copy using a storage-based copy technology.

PARTIAL DATABASE REPLICATION

The administrator can segregate the database into two or more projects to optimize network traffic. The first project contains files that must be replicated, such as the redo stream and production data files, and subsequent projects contain temporary files or read-only files.

Temporary files will need to be recreated, and read-only data will need to be referenced from an alternate location during recovery operations at the remote site. A sample partial database deployment can have two projects: replicated and nonreplicated.

The replicated project can be configured as follows:

- **Redo logs**: a 128-kilobyte record size share for storing online redo logs and control files
- **Datafiles**: a share configured with a record size that matches the database block size for storing database data files
- **Indexes**: a share configured with a record size that matches the index block size for storing database index files

The nonreplicated project can be configured as follows:

- **Temp**: a share configured with a record size to match the database sort area page size
- **Recovery**: a share configured with a 128-kilobyte record size and compression to store copies of the redo logs, backup sets of the database, and any other recovery information
- **Read only**: a share configured to match the block size of the read-only database

Similar to full database replication, any replication mode (continuous, scheduled, or on demand) can be used. If the replication target is to be integrated into a backup system, on-demand replication is required if the database needs to be in a backup state (for example, Oracle hot backup mode) to take a valid copy using a storage-based copy technology.

LOGS-ONLY REPLICATION

Logs-only replication is useful when network bandwidth must be conserved. In this architecture, the log files are shipped using replication and then applied to the database following an outage at the primary site. Logs-only replication can be accomplished for Oracle Database by replicating only the shares that store online redo logs.

The redo log share contains a copy of the online redo log and control file and is maintained with continuous replication. The archived-redo share contains a copy of the online redo log and is maintained with on-demand replication, so the archived log stream

can be periodically applied to the database at the remote site. Finally, the datafiles share contains all the datafiles associated with the database, and this project is not replicated, because the changes to bring the remote copy forward in time can be accomplished by applying the replicated copies of the online and archived redo log streams.

Business Applications and Middleware

Metadata and configuration details, as well as unstructured business application data, can be protected by storing the information on an Oracle ZFS Storage Appliance system and replicating the content to a target system. Different applications' data can be stored on different shares in the same project in cases where the data must be kept consistent with the data of a disparate application. A consistent image of a federated system can be maintained at the remote site in this model. Continuous, scheduled, or on-demand replication can be used, depending on the specific requirements of the implementation.

Consolidating Virtualized Environments

Protecting the virtualized environment infrastructure is critical. The virtual disk images (VMDK) files form the crux of the virtualized environment. The disk images are stored on either iSCSI LUNs or over NFS files. It is recommended to place the related disk images in a single project for replication. Then, a number of projects can be created and replicated.

A VMware VMFS3 filesystem can be created on the iSCSI LUN to host VMDKs, or the iSCSI LUN can be attached directly to a virtual machine using raw device mapping (RDM).

Differing replication schedules can be configured for different types of virtual machines or for subsets of virtual disk types; for example, OS/boot versus production data. For the replication, the recommendation is to configure multiple projects, each to replicate on its own schedule.

Either continuous or scheduled mode of replication can be configured. If the images need to be in a quiesced state, a manual mode using scripting at the virtual machine server that quiesces the image and then initiates the replication is preferred.

Protecting Mail Servers

Mail servers are a critical piece of the business component that need to be protected either by backup or by replicating to a remote site. In the event of a primary site problem, the mail server can then fail over to the target site to avoid downtime. Microsoft Exchange Server uses database and log components. Also, only iSCSI LUNs are supported to host these files.

In order to perform full replication, the logs and the database files are stored in the same project and replicated using any of the modes. Note that Microsoft recommends that, for better protection, logs and database files be separated on different physical drives. However, having a mirrored storage pool in an Oracle ZFS Storage Appliance system alleviates the problem and provides better performance.

FEATURES AND BENEFITS OF REPLICATION

The remote replication capability of Oracle ZFS Storage Appliance systems provides the following features and benefits:

- Replication is supported across Oracle ZFS Storage platforms and across storage profiles.
- No dedicated link is necessary; any network can be used for replication. However, in a more complex environment requiring predictable and guaranteed bandwidth for a replication service, a separated subnet/network to be used for replication is highly recommended.
- By default, the auto-tune compression and adaptive multithreading function is enabled for replication actions, providing more efficient network bandwidth utilization, which is especially important in slower WAN-type replication networks. This function can always be disabled at will if it turns out that data that is replicated cannot be sufficiently compressed to provide any replication performance gain.
- Data deduplication over the wire is an extra option that can be used to further increase network bandwidth utilization efficiency. This option should be used carefully as it requires extra CPU and memory resources from the Oracle ZFS Storage Appliance system.
- SSL encryption of the replication data stream sent over the network is an optional feature for data protection, and it can be disabled when performance is more important.

- Multiple Oracle ZFS Storage Appliance systems can be designated as failover sites (potential sources) to ensure business continuity in the event of almost any disaster.
- A project from a source can be replicated to one or more targets with different schedules and be cascaded along a series of Oracle ZFS Storage Appliance systems to better utilize local and long distance network bandwidth.
- For faster and efficient target site catch-up, only changes are replicated (except during the initial replication).
- When a replication operation is interrupted by a network failure, system outage or operator action, the replication will be automatically resumed from the point of disruption.
- Administrators have the flexibility to change the mode of replication from one mode to another.
- Any of the target site replicas can be used for many purposes, alleviating the performance pressure from the source site.
- The clone from the replica is treated as any other project, including taking snapshots of the shares, which provides more flexibility for QA and test purposes.
- Different retention policies can be specified at source and target for scheduled snapshots of projects/shares being replicated.
- Deleting a manually created snapshot at the source side does not automatically lead to the same snapshot being deleted at the target side. This enables more flexibility for the use of snapshots for creating backups at the target site.
- Efficient single-click reverse replication is provided to reverse the role of the source and target for a project, enabling faster DR.

CONSIDERATIONS FOR REPLICATION USE

Synchronous mode is not supported, so a zero data-loss requirement cannot be met. However, the continuous replication mode can provide an alternate with minimal data loss in the event of a disaster.

The write ordering and write consistency is maintained at the granularity of the replicated component. The write ordering is preserved within the share if the replication is set at the share level. However, the write ordering is not preserved across the shares if more than one share is replicated. The write ordering at the target for all the shares in the project is preserved if the replication happens at the project level. The write ordering is not preserved across the projects. Refer to the administration guide of a particular model of Oracle ZFS Storage Appliance for details.

Business availability and recovery objectives, such as RTO, RPO, and SLA, should be considered in deciding the mode of replication. The rate of change, latency, bandwidth, and number of projects to replicate all influence the decision-making process.

The target sites are not verified for the space requirement when the replication is established. Before initiating the replication, it is necessary to verify that the target sites have enough storage space to receive the replica and maintain snapshot data required to maintain data consistency across the entire replication architecture.

CONCLUSION

This paper describes the fundamentals of Oracle ZFS Storage Appliance software replication capabilities, including the latest enhanced replication features of multi-target reverse and cascaded replication. Complex replication architectures which include local replication, long distance replication, and even multi-continent are possible while using network bandwidth efficiently. Administrators can use this information in tailoring their application of remote replication to their particular storage environments.

REPLICATION TERMINOLOGY, ABBREVIATIONS, AND TERMS INDEX

Clone: A clone action on a filesystem or LUN creates an independent copy of a filesystem or a LUN.

Clone a replica: This action creates a read/write clone of the replica at the target site. The clone is used to promote the most recent received snapshot to a full r/w copy of that replication package.

Cascaded replication: An enhanced replication feature which enables the replication package to traverse along a string of designated Oracle ZFS Storage Appliance system nodes, each maintaining a replica of the package.

Distant Target: Used with multi-target reverse, a potential source that should not have its targets redirected during a replication reversal.

Final Target: Used with cascaded replication, the last Oracle ZFS Storage Appliance node in the cascaded chain of nodes.

Multi-target Reverse: An enhanced replication feature where two or more nodes can be designated as potential sources.

Potential Source: Used with multi-target reverse, an Oracle ZFS Storage Appliance system node that can assume the role of replication source when a replication reversal is performed.

Replication source: This is an Oracle ZFS Storage Appliance system node that is configured to send replication data to another system's node (target).

Replication target: This is an Oracle ZFS Storage Appliance system node that is configured to receive replication data from another system node (source).

Replication action: This action specifies a replicating action from the current node (source) to a target for a specific project/share. The action defines various replication attributes such as target address and pool, security, deduplication, thresholds for recovery point objective (RPO) target monitoring, and so forth. Replication actions can run either continuously or be triggered by user-defined schedules.

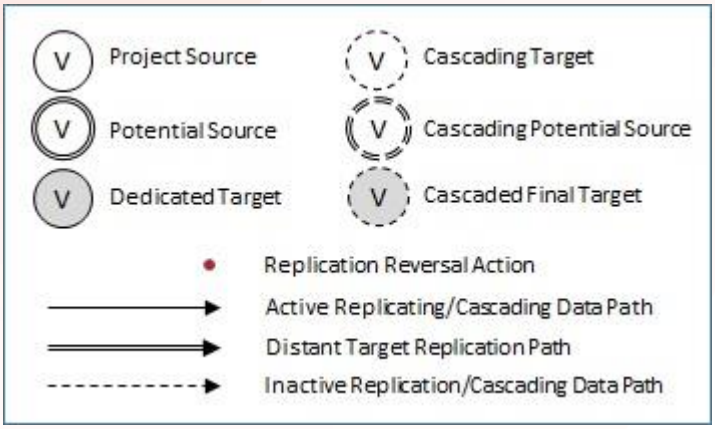
Replication schedule: Each replication action can contain one or more defined schedules. A schedule defines the starting point of each replication action. Schedule definitions are defined by user RPO requirements. RPOs can be monitored by defining RPO level thresholds.

RPO: Recovery point objective (RPO) is the point in time that the restarted business process has to go back to "find" the last consistent copy of data to work with. Basically, RPO is the rollback that will occur as a result of the recovery. To reduce a RPO it is necessary to increase the synchronicity of data replication.

RTO: Recovery time objective (RTO) is the time that is needed to have a business (process) up and running again after a disaster.

ROC: The rate of change (ROC) is the amount of data that is changed over time on the data volumes that are to be replicated to a second location. ROC can be expressed as a peak value or as an average over a period of time, such as a day. The ROC is used in combination with RPO requirements when determining the speed requirements of a replication link.

WOC: When replicating data on multiple volumes the order of updates (in time) to each volume must be kept consistent between the primary and target repository. This is referred to as write order consistency (WOC). When replicating a data repository spread over multiple volumes, all volumes must be kept in the same project in the Oracle ZFS Storage Appliance system, and the data replication must be set up at project level. More information about the WOC concept is available in the "The Art of Data Replication" white paper.


Project Source
Potential Source
Dedicated Target
Cascading Target
Cascading Potential Source
Cascaded Final Target
Replication Reversal Action
Active Replicating/Cascading Data Path
Distant Target Replication Path
Inactive Replication/Cascading Data Path

ORACLE CORPORATION

Worldwide Headquarters
500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries
TELE + 1.650.506.7000 + 1.800.ORACLE1
FAX + 1.650.506.7200
oracle.com

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com. Outside North America, find your local office at oracle.com/contact.

 blogs.oracle.com/oracle  facebook.com/oracle  twitter.com/oracle

Integrated Cloud Applications & Platform Services

White Paper **Implementing Remote Replication Using Oracle ZFS Storage Appliance**
June 2019

Author: Eric Polednik

 Oracle is committed to developing practices and products that help protect the environment