Ian MacQuarrie
William G. Carlson
Jim O'Connor
Limei Shaw
Shawn Wright

Configuring SDDPCM for High Availability

In this IBM® Redpaper™ publication we discuss new path management attributes that were added to the IBM Subsystem Device Driver Path Control Module (SDDPCM) for IBM AIX®. These new attributes and associated algorithms were developed to address a path failure condition referred to as *temporary/recurring*.

This paper is intended to familiarize you with both the temporary/recurring path failure condition and the new attributes available in SDDPCM version 2.6.3.2 that are designed to address it. Additionally, this paper provides guidance regarding the implementation of these attributes and other SDDPCM tunable attributes for improved availability.

Overview of path failure conditions

Complex storage area networks (SANs) have become prevalent in computer system configurations. SANs enable large numbers of servers to access common storage through an interconnection of switches and cabling.The availability and integrity of this interconnection is critical to the reliable operation of the system, and as such networks are often implemented with redundant routes (paths) between the servers and storage. Such redundant networks, in conjunction with intelligent multipath drivers installed on the servers, allow for failing commands to be recovered by rerouting them down alternate paths in the network, thereby preventing individual path failures from causing system outages.

Traditional failure categories

Traditionally, network path failures are viewed as falling into one of two categories, *permanent* and *temporary*, as explained here:

- Permanent failures

  Perhaps the most well-understood and easiest to manage are the permanent failures that result from a catastrophic failure of a network component. These failures are typically persistent failures, where all commands routed to the failing path or paths will fail. Commands are recovered through retrying the commands down alternate paths. Paths associated with a permanent fault are taken offline.

  Paths taken offline as a result of permanent path failures will remain offline. This is because any subsequent polling that might be performed to test path availability and restore paths to service will also fail due to the permanent nature of the fault.

- Temporary failures

  The second category is path failures that are temporary and transient in nature. These failures can arise from numerous sources including bit flips in electronics due to alpha particle or cosmic rays, electrical noise from intermittent contacts, and microcode defects. These can produce temporary command failures that are recovered through path retry.

  These conditions might or might not cause paths to enter into an offline state. However, because these are temporary conditions, attempts will be made to bring paths removed from service back into service by the path reclamation function.

Both of these traditional failure categories are generally handled well by existing multipath drivers available in the industry today, and in fact rarely result in any adverse effects on the operation of the system.

Emerging failure category

Unfortunately, as the line-speed, IOPS, and complexity of networks have increased over time, a third category of failure has emerged. These path failures are also temporary, like those in the second category, but in addition to being temporary they are also recurring at varying rates. This third category of path failure is called *temporary/recurring*, also sometimes referred to as the “sick but not dead” condition. These failures can arise from marginal components or from components and network routes that are insufficiently sized or oversubscribed for the volume of network traffic present.

Often these failures are provoked by secondary conditions such as an instantaneous increase in network traffic or the convergence of network traffic. These conditions can lead to congested ports and result in frames being dropped or discarded by switches and storage devices, which in turn cause command time-outs to be encountered by the attached hosts. These conditions tend to be more severe when encountered on Inter-Switch Links (ISLs) due to their potential to expand the number of hosts required to perform command time-out recovery.

Multipath drivers have historically not been able to uniquely identify the temporary/recurring failure condition, and therefore they often treat such failures the same as failures that fall into one of the two categories defined earlier, namely permanent or temporary. Without specific detection and path management for this failure category the condition might not be recognized immediately, and thus be allowed to persist for an extended period of time. In many cases the condition persists indefinitely or until manual intervention is performed.

This type of temporary/recurring failure condition drives recursive error recovery by the attached servers, which leads to symptoms ranging anywhere from moderate performance degradation to complete system outage. A common symptom seen are paths cycling between the online and offline states as the multipath driver takes paths offline for command failures that are later returned to service following the successful completion of a periodic path health check command.

New functionality for handling temporary/recurring error conditions

IBM has introduced new path management policies that are targeted at addressing the temporary/recurring failure condition. These policies enable new failure detection and path management algorithms designed to detect and respond to temporary/recurring failure conditions, and thereby improve availability.

New device attribute timeout_policy

SDDPCM 2.6.3.2 introduces a new function that will further improve handling of temporary/recurring failure conditions and reduce the symptoms and associated performance degradation described earlier. This function includes a new device attribute *timeout_policy*, which can be defined to enable new path management options.

> **Enhanced level:** This new function, which incorporates the *timeout_policy* attribute, was initially introduced in SDDPCM 2.6.3.0. However, substantive enhancements were made to the algorithms in 2.6.3.2. Therefore 2.6.3.2 is the advisable level to use when setting the *timeout_policy* attribute to *disable_path*.

The *timeout_policy* device attribute supports the following three options:

- retry_path
- fail_path
- disable_path

The attribute *timeout_policy* influences the path recovery behavior for paths that have been set to FAILED state (with the exception of the last remaining path) due to I/O failures for the *time-out* condition.

The recovery of failed paths will vary, depending on the recovery algorithm specified by the *timeout_policy* attribute:

- *retry_path*: A path will be set to FAILED state if one of the following conditions are met:
  - Error threshold is reached for recoverable I/O errors.
  - Permanent I/O error.

  Paths are recovered immediately upon successful completion of a single healthcheck command.

  The *retry_path* algorithm works in the same way as the SDDPCM versions released prior to 2.6.3.0.
- *fail_path*: A path will be set to FAILED state if one of the following conditions are met:
  - Error threshold is reached for recoverable I/O errors (all errors except for *time-out*).
  - Single command failed due to *time-out*.
  - Permanent I/O error.

  Paths are recovered upon successful completion of two consecutive healthcheck commands.
- *disable_path*: A path failed due to *time-out* will be set to DISABLED (OFFLINE) state if one of the following conditions (thresholds) are met:
  - Three consecutive commands fail due to *time-out.*
  - Three commands fail due to *time-out* within a 15-minute interval.

  The source of failing commands for both of these conditions can be either host I/O or healthcheck I/O. It is the cumulation of command failures from both of these sources that count towards the defined thresholds.

  After a path has been set to DISABLED it will not be returned to service until manually recovered using the **pcmpath** command.

  ```
  pcmpath set device m path n online
  ```

A new **pcmpath** command has been added to allow the user to dynamically change the *timeout_policy* attribute setting.

```
pcmpath set device <n>/<n>  <m>  timeout_policy  <option>
```

For detailed information regarding this **pcmpath** command, refer to the SDDPCM 2.6.3.2 readme file.

Default setting: The default setting for *timeout_policy* is *fail_path.*

A new AIX error log entry has also been added in SDDPCM 2.6.3.2 that will be generated when a path is disabled due to the *disable_path* policy. Example 1 shows a sample of the error log entry that will appear in the **errpt -a** output.

Example 1 AIX error log entry for disable_path

```
LABEL:          SDDPCM_PATH_DISABLE
IDENTIFIER:     6509813E

Date/Time:       Tue Apr 10 16:56:53 MST 2012
Sequence Number: 15954
Machine Id:      00F757EA4C00
Node Id:         arcp8205jmt1p2
Class:           H
Type:            PERM
WPAR:            Global
Resource Name:   hdisk8
Resource Class:  disk
```

```
Resource Type:   2145
Location:        U8205.E6C.1057EAR-V6-C14-T1-W500507680140AD91-L7000000000000

VPD:
        Manufacturer................IBM
        Machine Type and Model......2145
        ROS Level and ID............0000
        Device Specific.(Z0)........0000063268181002
        Device Specific.(Z1)........0200606
        Serial Number...............600507680181853D9000000000000163

Description
PATH HAS DISABLED

Probable Causes
ADAPTER HARDWARE OR CABLE
DASD DEVICE

Failure Causes
EF00

        Recommended Actions
        PERFORM PROBLEM DETERMINATION PROCEDURES
        CHECK PATH

Detail Data
PATH ID
           1
SENSE DATA
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 8000 0011 0000
0006 0000 000F 0000 0972 0000 0000 0000 0000 0000 0000 B002 B6B0 0000 0000 0058
FA8B 0000 0000 005B B9AA 0000 0001 0000 0003
```

Give special consideration to the use of the *disable_path* policy, because disabled paths will not be automatically returned to service. Perform rigorous and thorough monitoring of the path state following events such as concurrent code updates and maintenance to ensure full path redundancy is maintained. Although it might not always be necessary, use the most conservative approach, to avoid conditions where paths can be permanently taken offline during scheduled events such as code updates and maintenance. It is advisable to temporarily change the *timeout_policy* to *fail_path* prior to this activity, and then change it back to *disable_path* at the completion of the maintenance.

If this is not done, it is imperative to check the path state throughout such activity. Test results have shown the exposure to paths disabling during concurrent code updates and maintenance to be minimal. Nevertheless, it is an exposure and therefore needs to be considered and protected against.

Although it is generally advised that clients have automation in place to monitor path status, this becomes critically important when using the *disable_path* policy because DISABLED paths will remain offline until a manual action is performed to return them to service. The underlying condition causing the paths to become disabled, the action that prevents the adverse impact of the recurring intermittent, must be dealt with promptly in the same way a permanent fault must be attended to promptly and the paths reactivated to restore full redundancy to the network. Failure to promptly detect and restore disabled paths can compromise the fault tolerance of the system and expose the client's system to the impact of a secondary fault.

Suggested approaches

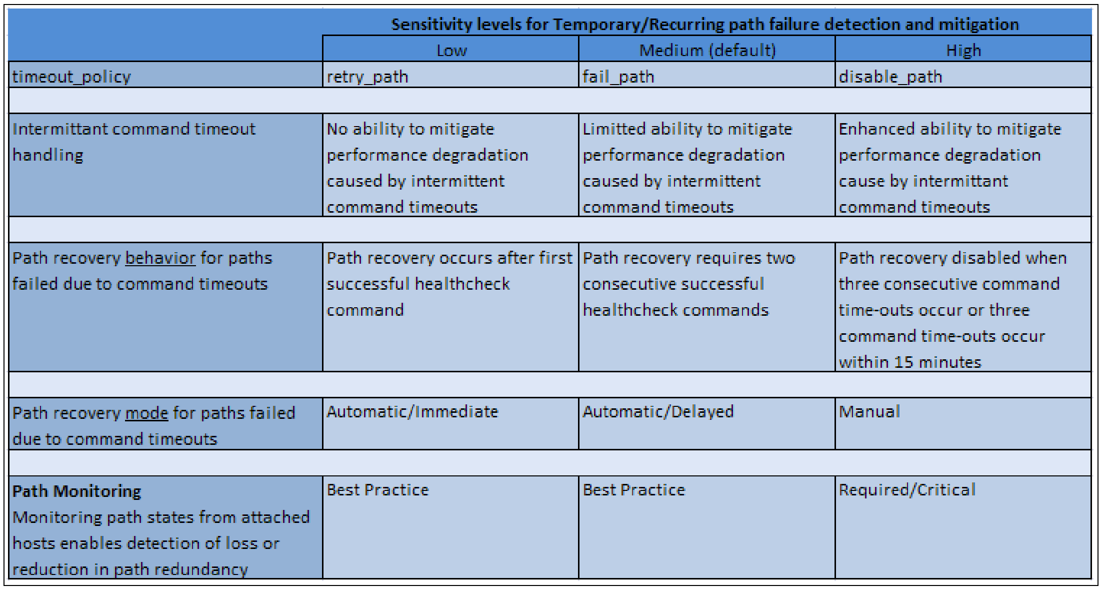
Servers hosting business-critical and response time-sensitive databases and applications should be configured to use *disable_path*.

Attention: Do not use *disable_path* without server side path monitoring in place. Doing so can result in paths being left in the disabled state for an extended period, which can compromise availability.

Although monitoring of path states is a preferred practice regardless of the *timeout_policy* chosen, to ensure that path redundancy is not compromised due to the introduced exposure of having paths remain in an offline state for an extended period, it becomes critically important when using *disable_path*.

When server-side path monitoring is not in place, use the default *fail_path* policy.

Figure 1 on page 6 provides a summary of the *timeout_policy* attributes along with associated behaviors and implementation considerations.



	Sensitivity levels for Temporary/Recurring path failure detection and mitigation		
	Low	Medium (default)	High
timeout_policy	retry_path	fail_path	disable_path
Intermittant command timeout handling	No ability to mitigate performance degradation caused by intermittent command timeouts	Limitted ability to mitigate performance degradation caused by intermittent command timeouts	Enhanced ability to mitigate performance degradation cause by intermittant command timeouts
Path recovery behavior for paths failed due to command timeouts	Path recovery occurs after first successful healthcheck command	Path recovery requires two consecutive successful healthcheck commands	Path recovery disabled when three consecutive command time-outs occur or three command time-outs occur within 15 minutes
Path recovery mode for paths failed due to command timeouts	Automatic/Immediate	Automatic/Delayed	Manual
Path Monitoring Monitoring path states from attached hosts enables detection of loss or reduction in path redundancy	Best Practice	Best Practice	Required/Critical

Figure 1 Summary of timeout_policy attributes

Additional considerations

Keep the following consideration in mind regarding LVM mirroring.

Optimizing for LVM mirroring

When LVM mirroring is in use, it is advantageous to switch to the remaining mirror copy as soon as possible after a failing mirror become unavailable. Doing so will limit the performance degradation associated with the error recovery to the failing mirror copy.

Starting from SDDPCM 2.4.0.3, a new device attribute *retry_timeout* was added for ESS/DS6K/DS8K/SVC devices. This attribute allows users to set the *time-out* value for I/O retry on the last path. The default value of this attribute is 120 seconds, and it is user-changeable with the valid range of 30 to 600 seconds.

For LVM mirrored configurations it is advisable to lower the *retry_timeout* from its default 120 seconds to the minimum 30 seconds.

A new **pcmpath** command has been added to allow the user to dynamically change this attribute setting.

```
pcmpath set device <n1>  [n2] retry_timeout <t>
```

For detailed information regarding this **pcmpath** command, refer to the SDDPCM 2.6.3.2 readme file.

The team who wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Ian MacQuarrie is a Senior Technical Staff Member with the IBM Systems and Technology Group located in San Jose, California. He has 26 years of experience in enterprise storage systems in a variety of test and support roles. He is currently a member of the Systems and Technology Group (STG) Field Assist Team (FAST) supporting clients through critical account engagements, availability assessments, and technical advocacy. Ian’s areas of expertise include storage area networks (SANs), open systems storage solutions, and performance analysis.

William G. Carlson is a Senior Engineer with the IBM Systems and Technology Group located in Poughkeepsie, NY. He has 24 years of experience in enterprise hardware, software development, and test. He is currently a member of the Systems and Technology Group (STG) Integrated Solution Test (IST) and manages testing partnership for several large clients. William’s areas of expertise are computer hardware and software development and test with a focus on open systems with high availability networks (SANs) and storage.

Jim O’Connor is an IBM Distinguished Engineer responsible for STG Advanced System RAS Design and Quality. A recognized expert in crossbrand common hardware design, system quality and system RAS Architecture, he specializes in the analysis and architecture of open and virtualized clustered systems including IBM POWER® Systems, IBM BladeCenter® and IBM PureSystems™. His experience includes hardware, firmware, and system software development, and system test. He is frequently asked to solve complex crossbrand system problems and customer issues, and continues to work closely with customer teams. Jim earned an Engineer’s Degree in Computer Engineering, a Ph.D. for working professionals from Syracuse University (1999), an MS in Computer Engineering (1983), and a BS in Electrical Engineering from Rutgers University (1980).

Limei Shaw is a Senior Software Engineer with the IBM Systems and Technology Group located in San Jose, California. She has 22 years of experience in designing and developing HBA firmware, the AIX HBA device driver, and multipath device drivers for IBM storage systems. Her expertise in AIX device drivers and SCSI/FC storage subsystems plays an important role in developing the products and supporting IBM clients through critical account engagements. Limei’s areas of expertise include the AIX device driver, storage virtualization, and open systems storage solutions.

Shawn Wright is a Senior Software Engineer with the IBM Systems and Technology Group located in Austin, Texas. He has 15 years of experience in integration testing and support from an end-to-end view for various server, SAN, and storage solutions. Shawn is currently the test architect for SAN & Storage within the Integrated Software Systems Test team. His areas of expertise include storage area networks (SANs), AIX IO device drivers, IBM PowerVM®, high availability clustering, and debug analysis.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-4928-00 was created or updated on October 15, 2012.

Send us your comments in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Redpaper™

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
BladeCenter®
IBM®
PowerVM®
POWER®
PureSystems™
Redbooks®
Redpaper™
Redbooks (logo) ®

The following terms are trademarks of other companies:

Other company, product, or service names may be trademarks or service marks of others.