DXi Log Reading Basics

Page Notes (0) Attachments (2)

Overview

When Service Tickets and GUI status don't provide the information you need to solve SRs, an excellent next step is to review logs. In many cases, logs can be helpful in solving customer problems. Use the information on these pages, as well as the job aids listed under the **Resources** section, to learn which logs are available, how to gather them, the types of information that each log contains, and the types of problems each log is able to help solve.

While there are DXi system logs and storage array logs, this *DXi Log Reading Basics* topic focuses on the DXi system logs.

Main topics include:

Prerequisites

- DXi-Series Core Concepts for Service (2-0443 online self-paced)
- DXi7500 and DXi8500 Installation and Support Part 2 (2-0413 classroom)
- DXi-Series 2.x Hands-on GUI Practice (3-2432 virtual classroom)

Resources

- DXi-Series Status and Log Collection Job Aid
- DXi-Series Log Location and Description Job Aid

Assessment

To get credit for completing this training, you must complete a short assessment (exam) in StorageCare Learning (A-3-2420 Assessment). Instructions on how to access the assessment are presented at the end of this topic.

What's Next?

Getting Started >

Getting Started

Page Notes (0) Attachments (2)

Overview

There are several approaches to troubleshooting using the DXi system logs.

- Start with the **tsunami log** for system services and process information (such as replication, truncation, and space reclamation).
- Look in the **messages log** for Samba, blockpool, or SNFS issues.
- Another approach is to **grep** the DXi diagnostic logs for specific keywords to identify applicable log files that need to be reviewed.

Using DXi system logs to troubleshoot is a skill that you will develop over time as you review more and more logs. Identifying the failure, and knowing where to look in the logs, are key to becoming proficient at reviewing logs.

DXi system logs are constantly updated and stored on the DXi system. You can access and review them from the system, or you can collect a snapshot of the logs at a given point in time.

Note: Some editing programs in Windows may display the CR and LF characters in the file as if they were text when you click on the link above, making the file unreadable. Use a program that supports formatting, such as Notepad++ or WordPad.

How to Collect Logs

Use the DXi-Series Status and Log Collection Job Aid as a reference. It describes various ways to capture DXi diagnostic logs.

Log Location and Contents

Use the DXi-Series Log Location and Description Job Aid to learn about the most-often used log files, their location within a captured diagnostic log file and on the system, and the types of information they contain.

What's Next?

Most-Often Used Logs >

Most-Often Used Logs: Tsunami Log and Messages Log

Tsunami Log Overview

The **tsunami log** includes information about services and processes such as truncation, tape states, and space reclamation. Among the things you can find information about in this log are:

- I/O issues (device driver issues
- Network problems
- ERROR statements related to the component you are having problems with
- Space reclamation (**bpgc,** which stands for blockpool garbage collection) status
- VTL mount and un-mount info (tape mount/unmount)
- TL State information
- **hwmon** (which stands for hardware monitor) status
- **qbfsd** (which stands for Quantum blockpool file system daemon) status

Space Reclamation (bpgc) Status Example

```
INFO - 11/23/10-13:10:27 - bpgc ReconcileThread.cpp(1773) [bpgc]
determineCandidates() - {1082132832} Determining the blockbool deletion candidates
INFO - 11/23/10-13:10:27 - bpgc FileUtil.cpp(201) [bpgc] sortFile() - Sorting
file /data/hurricane/bpgc/blockpooltags
```

Mount/Dismount Example

```
INFO - 11/29/10-16:22:13 - Vmm.ThreadMethods ThreadMethods.cc(320) [vmm]
mountMedium() - Mount of barcode <EA0005> requested from reqId 2033341285
INFO - 11/29/10-16:22:13 - Vmc.CommandGeneric MoveMedium.cc(214)
[vmcVL02CX1029BVA00256] ExecuteCommand() - MoveMedium::ExecuteCommand: requestId
2033
341285: Generic Success
INFO - 11/29/10-16:22:13 - VTDMessenger vtdMessenger.cpp(267)
[vtdVD15CX1029BVA00256] VTD_MountHandler() - Mount: barcode: EA0005;
pathToTape: /Q/part
itions/VL02CX1029BVA00256/Carts/EA0005_CX1029BVA00256_262688; writeProtect: false
INFO - 11/29/10-16:22:13 - Vmm.MediumWorker MediumWorker.cc(80) [vmm] runCmd() -
Locking medium EA0005
INFO - 11/29/10-16:22:13 - Vmm.BaseCmd VTDMountMediumCmd.cc(132) [vmm] doExecute()
- Sending VTD_MountMsg for mount of barcode <EA0005> pathToTape </Q
/partitions/VL02CX1029BVA00256/Carts/EA0005_CX1029BVA00256_262688> to drive
<VD15CX1029BVA00256>
INFO - 11/29/10-16:22:13 - Vmm.DbLoader VmmDbMediumLoader.cc(535) [vmm] update() -
Updated barcode: <EA0005>, state: </Q/partitions/VL02CX1029BVA00256
/Carts/EA0005_CX1029BVA00256_262688 - 385>, vmc <VL02CX1029BVA00256> in the
database for reqId 2033341285
INFO - 11/29/10-16:22:13 - Vmm.MediumWorker MediumWorker.cc(87) [vmm] runCmd() -
Unlocking medium EA0005
INFO - 11/29/10-16:22:15 - Vmm.ThreadMethods ThreadMethods.cc(349) [vmm]
dismountMedium() - Dismount of barcode <EA0005> requested by reqId 2033341345
INFO - 11/29/10-16:22:15 - Vmm.MediumWorker MediumWorker.cc(80) [vmm] runCmd() -
Locking medium EA0005
INFO - 11/29/10-16:22:15 - Vmm.BaseCmd VTDDismountMediumCmd.cc(90) [vmm] doExecute
() - Sending VTD_DismountMsg for dismount of drive <VD15CX1029BVA002
56>
INFO - 11/29/10-16:22:15 - VTDMessenger vtdMessenger.cpp(396)
[vtdVD15CX1029BVA00256] VTD_DismountHandler() - Dismount: EA0005 forceUnload:
false
```

```
INFO - 11/29/10-16:22:15 - Vmm.BaseCmd VMMDismountCmd.cc(68) [vmm] doExecute() -
VTDDismountMediumCmd Drive <VD15CX1029BVA00256> Barcode <EA0005> Phys
ical dismount <0> Virtual force unload <0> Status <0 - Generic Success>
INFO - 11/29/10-16:22:15 - Vmm.DbLoader VmmDbMediumLoader.cc(535) [vmm] update() -
Updated barcode: <EA0005>, state: </Q/partitions/VL02CX1029BVA00256
/Carts/EA0005_CX1029BVA00256_262688 - 384>, vmc <VL02CX1029BVA00256> in the
database for reqId 2033341345
INFO - 11/29/10-16:22:15 - Vmm.BaseCmd VMMDismountCmd.cc(85) [vmm] doExecute() -
Medium.unsetMounted Barcode <EA0005> Status <0 - Generic Success>
INFO - 11/29/10-16:22:15 - Vmm.MediumWorker MediumWorker.cc(87) [vmm] runCmd() -
Unlocking medium EA0005
INFO - 11/29/10-16:22:15 - Vmc.CommandGeneric MoveMedium.cc(101)
[vmcVL02CX1029BVA00256] ExecuteCommand() - MoveMedium::ExecuteCommand: requestId
2033
341345: Generic Success
```

Location

In the system diagnostics file: **scratch\collect\node1-collection\app-info\tsunami.log**
On the system: **/var/log/DXi/tsunami.log** or you can access the symbolic link/shortcut under **/hurricane/tsunami.log**

Messages Log Overview

The **messages log** includes information about the operating system and StorNext filesystem issues. It can be useful for finding information on:

- Core symptoms, memory problems, and system level issues.
- Panics and/or core dump info.
- General boot sequence for systems not bringing up FSM.
- Disk Automatic Event Notifications (AEN) errors. An AEN is a message issued by the 3ware RAID controller when an error is detected or when an action is completed.

Kernel Panic Example

```
# grep KDUMP /var/log/messages

Jul 02 11:43:48 Flames KDUMP: CRIT : KERNEL PANIC/OOPS/CRASH has occurred
Jul 02 11:43:50 Flames KDUMP: INFO : *********begin stack traces ********
Jul 02 11:44:26 Flames KDUMP: PID: 30096 TASK: ffff8107a9d0e140 CPU: 2 COMMAND:
"bash"
```

AEN Example

```
Sep 10 00:07:40 labdxi1 3wcache.sh: Warning: non-optimal cache settings for /c2/u0
Sep 10 00:05:13 labdxi1 kernel: 3w-9xxx: scsi2: AEN: INFO (0x04:0x0029): Verify
started:unit=0.
Sep 10 00:07:41 labdxi1 3wcache.sh: /c2/u0 Read Cache = Intelligent
Sep 10 00:07:42 labdxi1 3wcache.sh: /c2/u0 Write Cache = off
Sep 10 00:07:42 labdxi1 3wcache.sh: Unable to correct cache settings due to BBU
status:
Sep 10 00:07:42 labdxi1 3wcache.sh: /c2/bbu On No Failed OK OK 0 xx-xxx-xxxx
Sep 10 00:07:42 labdxi1 3wcache.sh: Warning: non-optimal cache settings for /c2/u1
Sep 10 00:07:42 labdxi1 3wcache.sh: /c2/u1 Read Cache = Intelligent
Sep 10 00:07:43 labdxi1 3wcache.sh: /c2/u1 Write Cache = off
Sep 10 00:07:43 labdxi1 3wcache.sh: Unable to correct cache settings due to BBU
status:
```

```
Sep 10 00:07:43 labdxi1 3wcache.sh: /c2/bbu On No Failed OK OK 0 xx-xxx-xxxx
Sep 10 00:10:33 labdxi1 3wcache.sh: Warning: non-optimal cache settings for /c2/u0
Sep 10 00:10:34 labdxi1 3wcache.sh: /c2/u0 Read Cache = Intelligent
Sep 10 00:10:35 labdxi1 3wcache.sh: /c2/u0 Write Cache = off
Sep 10 00:10:35 labdxi1 3wcache.sh: Unable to correct cache settings due to BBU
status:
Sep 10 00:10:35 labdxi1 3wcache.sh: /c2/bbu On No Failed OK OK 0 xx-xxx-xxxx
Sep 10 00:10:35 labdxi1 3wcache.sh: Warning: non-optimal cache settings for /c2/u1
Sep 10 00:10:36 labdxi1 3wcache.sh: /c2/u1 Read Cache = Intelligent
Sep 10 00:10:36 labdxi1 3wcache.sh: /c2/u1 Write Cache = off
Sep 10 00:10:36 labdxi1 3wcache.sh: Unable to correct cache settings due to BBU
status:
Sep 10 00:10:36 labdxi1 3wcache.sh: /c2/bbu On No Failed OK OK 0 xx-xxx-xxxx
Sep 10 00:12:33 labdxi1 kernel: 3w-9xxx: scsi1: AEN: WARNING (0x04:0x0023): Sector
repair completed:encl=0, slot=4, LBA=0x1EBAA8A.
```

Location

In System Diagnostics: **scratch\collect\node1-collection\os-info\messages**
On the system: **/var/log/messages**

Potential Gotchas - Spurious Shutdown Errors

One specific way that the **tsunuami log** (and perhaps the also the **messages log)** can be misleading is that during shutdown sequences, you will see errors that are due to various processes being stopped, while others continue. If you are focused in on finding ERROR statements, this could lead you to think you've spotted a problem, when it's actually only a brief anomaly.

What's Next?

The Collect.txt Log >

The Collect.txt Log

Page Notes (0) Attachments (1)

Overview

One of the first things you can do is open the **collect.txt** file and look at the commands used to collect all the various logs, files, and output. This is a good place to start if you can't remember what command gets you the output on the live system that you want.

Purpose

The **collect.txt** file includes high-level information about the system and its components, including the system serial number, host name, and code version. This file is useful for quickly verifying the system's configuration. It also lists the location of the logs being collected.

Open the sample collect.txt file from a DXi6520 system.

Note: Some editing programs in Windows may display the CR and LF characters in the file as if they were text when you click on the link above, making the file unreadable. Use a program that supports formatting, such as Notepad++ or WordPad.

EXAMPLE:

```
###15:52:49### -Version- 'cat //opt/DXi/version_info':
Major=2
Minor=0
Patch=0
LCR=
Port=
Build=40091
Version=2.0.0-40091
Platform=DXi6500
```

Location

In System Diagnostics: **scratch\collect\node1-collection\collect.txt**
On the system: **Not available on the system**

What's Next?

Locating Truncation Issues >

Locating Truncation Issues

Page Notes (2) Attachments (0)

Overview

If you suspect truncation issues, search for **TRUNCATION** in the **tsunami log.**

Note: If you regularly see truncation in an hour or less, the DXi system is too small.

Truncation Information in tsunami.log

From the diagnostics file:
```
grep TRUNCATION scratch/collect/node1-collection/app-info/tsunami.log
```

From the system:
```
grep TRUNCATION /var/log/DXi/tsunami.log
```

EXAMPLE OUTPUT

```
WARN - 05/13/09-11:06:45 - dedupd dedup.cpp(1599) [/opt/DXi/dedupd]
start_trunc_event_handler() - Received SM_START_TRUNCATION_EVENT !!
WARN - 05/13/09-14:29:57 - dedupd dedup.cpp(1615) [/opt/DXi/dedupd]
stop_trunc_event_handler() - Received SM_STOP_TRUNCATION_EVENT !!
```

What does this example output tell you?

This example shows that the customer reached the Truncation level at 11:06:45 and came out of truncation at 14:29:57. Use this information when looking at ingest, blockpool, deduplication, and bpgc performance issues, and whem performing troubleshooting.

Truncation Information in adminLog.hist

From the diagnostics file:
```
grep Truncate
scratch/collect/node1-collection/app-
info/pse_snapshot/usr/adic/SRVCLOG/logs/adminLog.hist
```

From the system:
```
grep Truncate /usr/adic/SRVCLOG/logs/adminLog.hist
```

EXAMPLE OUTPUT

```
May 13 14:29:57 2009 102 1913258518 SpaceManagementDaemon <subject_line>Space
Management: Cleared Immediate Truncate Mode</subject_line><email_body>The system is
not currently truncating original files after de-duplication. The system freespace is
under the truncation threshold.</email_body>
```

What does this example output tell you?

The message output pretty much says it all. The DXi system is no longer truncating original files, and has resumed normal disk space mode.

Right after the above admin alert, you will see:

```
SpaceManagementDaemon <subject_line>Space Management: Resumed Normal Disk Space
Mode</subject_line><email_body>The system is not currently truncating original files
after de-duplication. The system freespace is under the truncation
threshold.</email_body>
```

What's Next?

Locating Replication Issues >

Locating Replication Issues

Page Notes (3) Attachments (0)

Overview

If the customer is having replication issues, do the following:

1. Make sure that replication is set up correctly. This can be verified in the **replication.conf** section of the **collect.txt** log on both the source and target DXi system:

   ### -Replication- 'cat /data/hurricane/replication.conf':
   [Global_Global]
   CvfsMountPoint=/snfs/Q/
   DedupWindowDuration=960
   DedupWindowScheduledTimesForSystem=05:00
   EncryptionEnabled=false
   IsReplicationForSystemEnabled=false
   ProgramReplicationIsPaused=false
   QbfsMountPoint=/Q/
   ReplicationScheduledTimesForSystem=
   SourceHostList=10.212.17.27,10.254.246.27
   UserReplicationIsPaused=false

   [Share_1]
   DedupAge=60
   DedupEnabled=true
   DedupWindowDuration=0
   DedupWindowScheduledTimes=
   NodeId=
   NodeName=AB10_NASTEST
   ReplicationDestinations=10.212.17.27
   ReplicationEnabled=false
   ReplicationRole=3
   ReplicationScheduledTimes=
   RetentionAge=120
   ShareType=0
   TriggerReplication=0
   TriggerReplicationId=

   [Share_2]
   DedupAge=60
   DedupEnabled=true
   DedupWindowDuration=0
   DedupWindowScheduledTimes=
   NodeId=
   NodeName=AB10_NAS002
   ReplicationDestinations=10.212.17.27
   ReplicationEnabled=true
   ReplicationRole=3
   ReplicationScheduledTimes=17:00
   RetentionAge=120
   ShareType=0
   TriggerReplication=0
   TriggerReplicationId=

2. Then make sure that the source and target are communicating.

   Use the **ping** command to check connectivity from source to target, and vice versa, from target to source. Example: **ping 10.212.17.27** (target IP address). Do the same from target to source.

   Use the **telnet** command to telnet from source to target on port 1062

Example: **telnet <target ip> 1062**
Example: **telnet <target ip> 80**

3. Check the **tsunami.log** file and look for error messages.

Examples of Replication Failures

Example 1

ERROR - 03/30/10-22:02:15 - replicationd DestinationThread.cpp(918) [replicationd] continuousReplication() - {1082132832} Continuous data replication activity has failed to target host: 10.212.17.27
Error details: Network error

Example 2

ERROR - 04/10/10-15:51:03 - replicationd DestinationThread.cpp(1749) [replicationd] namespaceReplication() - {1082132832} QLOG_REP_ERROR - Namespace Replication FAILED for task: destinationId=10.212.17.27, launchTime=Sat Apr 10 15:00:00 2010(1270933200), nodeName=AB10_NAS003, nodeId=, nodeType=Share(0), taskMode=Scheduled(3), nodeDirectory: /Q/shares/AB10_NAS003, eventHostId: AB10-DXI004.Celero.ca, eventBundleUID: 0, eventBundleUID: 130, synchronizeId: DETAILS:
WARN - 04/10/10-15:51:03 - replicationd DestinationThread.cpp(1776) [replicationd] namespaceReplication() - {1082132832} Finalizing Source namespace bundle for FAILED Namespace Replication name: AB10_NAS003 destination: 10.212.17.27 error:

Useful Commands When Looking for Replication Issues

The following section provides some commands that can be used to look for replication issues directly from the DXi system:

From the system

ssh to the DXi and from the command prompt type in the following commands:

less /hurricane/tsunami.log
tail -f /hurricane/tsunami.log
grep replicationd /hurricane/tsunami.log | tail -10

EXAMPLE OUTPUT

Example 1

This example provides the actiivty going on in the **tsunami log,** replication activity, and error messages. This command is useful if information from one or more previous months is needed:

INFO - 01/12/11-17:16:39 - replicationd REDaemon.cpp(313) [replicationd] initialize() - Signals blocked
INFO - 01/12/11-17:16:39 - replicationd REDaemon.cpp(528) [replicationd] doInitialize() - rm cmd is : rm -fr /snfs/tmp/replication/namespace/*
INFO - 01/12/11-17:16:39 - replicationd REDaemon.cpp(534) [replicationd] doInitialize() - rm cmd is : rm -fr /Q/partitions/namespace_*
INFO - 01/12/11-17:16:39 - replicationd BfstV2.cpp(67) [replicationd] initialize() - Bfst initialize needed - calling bfst2_initialize
INFO - 01/12/11-17:16:39 - replicationd REDaemon.cpp(623) [replicationd] startAllChildThreads() - Launching Scheduler Thread
INFO - 01/12/11-17:16:39 - replicationd REDaemon.cpp(627) [replicationd] startAllChildThreads() - Launching Command Thread
INFO - 01/12/11-17:16:39 - replicationd REDaemon.cpp(1607) [replicationd] handleProgrammaticPause() - Programmatic Pause: setting paused to false
INFO - 01/12/11-17:16:39 - replicationd ReplicationAPI.cpp(14584) [replicationd] setGlobalReplicationPauseStatus() - System has resumed the replication service1

Press **SHIFT-G** to go to the bottom of the file. You can then scroll up and then down as needed:

INFO - 03/08/11-13:00:19 - bpgc ReconcileThread.cpp(1327) [bpgc] dumpReplicatedNamespaceTags() - {1082132832} Gathering the tags from all replicated namespaces
INFO - 03/08/11-13:00:19 - bpgc NamespaceBundleUtil.cpp(262) [bpgc] untar() - Successfully untared namespace bundle: /snfs/tmp/replication-expansion1082132832_1299589219/target/DXi5-02nh.quantum.com/shares/keithR2/1/bundle.tar
INFO - 03/08/11-13:00:19 - bpgc NamespaceBundleUtil.cpp(262) [bpgc] untar() - Successfully untared namespace bundle: /snfs/tmp/replication-expansion1082132832_1299589219/target/DXi5-02nh.quantum.com/partitions/KP1/1/bundle.tar
INFO - 03/08/11-13:00:19 - bpgc NamespaceBundleUtil.cpp(262) [bpgc] untar() - Successfully untared namespace bundle: /snfs/tmp/replication-expansion1082132832_1299589219/target/DXi5-02nh.quantum.com/partitions/Keiths/3/bundle.tar
INFO - 03/08/11-13:00:19 - bpgc NamespaceBundleUtil.cpp(262) [bpgc] untar() - Successfully untared namespace bundle: /snfs/tmp/replication-expansion1082132832_1299589219/target/DXi5-02nh.quantum.com/partitions/Keiths/2/bundle.tar
INFO - 03/08/11-13:00:19 - bpgc NamespaceBundleUtil.cpp(262) [bpgc] untar() - Successfully untared namespace bundle: /snfs/tmp/replication-expansion1082132832_1299589219/target/DXi5-02nh.quantum.com/partitions/Keiths/1/bundle.tar

Example 2

This command will show you the last and current activity going on in the **tsunami.log.** It is useful to get a quick idea of the current status of the system:

[root@DXi75-nh1 ~]# tail -f /hurricane/tsunami.log
INFO - 03/08/11-19:05:09 - hwmon MonitorArrayStatus.cpp(2260) [hwmon] CheckFailureList() - Failure Type List: RecoveryFailureTypeValue::REC_VOLUME_HOT_SPARE_IN_USE.
INFO - 03/08/11-19:05:09 - hwmon MonitorArrayStatus.cpp(2261) [hwmon] CheckFailureList() - Taking no action.
INFO - 03/08/11-19:07:15 - hwmon MonitorArrayStatus.cpp(2260) [hwmon] CheckFailureList() - Failure Type List: RecoveryFailureTypeValue::REC_VOLUME_HOT_SPARE_IN_USE.
INFO - 03/08/11-19:07:15 - hwmon MonitorArrayStatus.cpp(2261) [hwmon] CheckFailureList() - Taking no action.
INFO - 03/08/11-19:07:15 - hwmon perfmon.cpp(527) [hwmon] PerfServerThread() - PerfServerThread: New connection from client '10.17.21.1' socket 22
INFO - 03/08/11-19:07:15 - hwmon perfmon.cpp(774) [hwmon] ReadClientConnection() - Got get request from client '10.17.21.1' flags 0x131
INFO - 03/08/11-19:07:16 - hwmon perfmon.cpp(527) [hwmon] PerfServerThread() - PerfServerThread: New connection from client '10.17.21.1' socket 22
INFO - 03/08/11-19:07:16 - hwmon perfmon.cpp(774) [hwmon] ReadClientConnection() - Got get request from client '10.17.21.1' flags 0x131
INFO - 03/08/11-19:09:21 - hwmon MonitorArrayStatus.cpp(2260) [hwmon] CheckFailureList() - Failure Type List: RecoveryFailureTypeValue::REC_VOLUME_HOT_SPARE_IN_USE.
INFO - 03/08/11-19:09:21 - hwmon MonitorArrayStatus.cpp(2261) [hwmon] CheckFailureList() - Taking no action.

Example 3

This command is useful to narrow down the search by listing the last 10 lines in the **tsunami.log:**

[root@DXi75-nh1 ~]# grep replicationd /hurricane/tsunami.log |tail -10
INFO - 03/07/11-03:00:02 - replicationd DestinationThread.cpp(1666) [replicationd] namespaceReplication() - {1082132832} QLOG_REP_INFO - Successful namespace replication completed for: Keiths to target: 10.105.13.77
INFO - 03/08/11-03:00:00 - replicationd RESchedulerThread.cpp(650) [replicationd] doProcess() - {1166059872} Adding new namespace replication task [destinationId=10.105.13.77, launchTime=Tue Mar 8 03:00:00 2011(1299553200), nodeName=Keiths, nodeId=VL06CX0743BVA00005, nodeType=Partition(1), taskMode=Scheduled(3), nodeDirectory: /Q/partitions/VL06CX0743BVA00005, eventHostId: DXi75-nh1.labs.northampton.uk, eventBundleUID: 0, eventBundleUID: 21, synchronizeId:]
INFO - 03/08/11-03:00:00 - replicationd NamespaceReplicator.cpp(1115) [replicationd] generateNamespace() - Command /hurricane/metatar -c -f /data/hurricane//snfs/tmp/replication/namespace/Keiths/metadata -b /data/hurricane//snfs/tmp/replication/namespace/Keiths/taglist -s /data/hurricane//snfs/tmp/replication/namespace/Keiths/metastatus -w /data/hurricane//snfs/tmp/replication/namespace/Keiths/waitlist -

```
l /data/hurricane//snfs/tmp/replication/namespace/Keiths/barcodes -t part -
d /Q/partitions/VL06CX0743BVA00005 -n done, exitcode=0
INFO   - 03/08/11-03:00:00 - replicationd NamespaceReplicator.cpp(1129) [replicationd] generateNamespace
() - QLOG_REP_INFO -  Complete replication initiated for Keiths to target: 10.105.13.77
INFO   - 03/08/11-03:00:00 - replicationd RemoteBfstV2.cpp(168) [replicationd] connect() -
[TID=1082132832] RemoteBfst::connect '127.0.0.1'
INFO   - 03/08/11-03:00:00 - replicationd RemoteBfstV2.cpp(281) [replicationd] disconnect() -
[TID=1082132832] In RemoteBfstV2::disconnect()
INFO   - 03/08/11-03:00:00 - replicationd RemoteBfstV2.cpp(168) [replicationd] connect() -
[TID=1082132832] RemoteBfst::connect '127.0.0.1'
INFO   - 03/08/11-03:00:00 - replicationd RemoteBfstV2.cpp(281) [replicationd] disconnect() -
[TID=1082132832] In RemoteBfstV2::disconnect()
INFO   - 03/08/11-03:00:01 - replicationd NamespaceReplicator.cpp(1923) [replicationd] sendDataToTarget()
- Namespace completed for sync id:
INFO   - 03/08/11-03:00:01 - replicationd DestinationThread.cpp(1666) [replicationd] namespaceReplication()
- {1082132832} QLOG_REP_INFO - Successful namespace replication completed for: Keiths to target:
10.105.13.77
[root@DXi75-nh1 ~]#
```

A similar command can be used to list the top 10 lines in the log. To do this, use **head** instead of **tail**":

```
[root@DXi75-nh1 ~]# grep replicationd /hurricane/tsunami.log |head -10
INFO   - 01/12/11-16:39:31 - procmon ProcMonInvoke.cpp(147) [procmon] Start() - Invoked command
'/etc/init.d/replicationd start' (27823)
INFO   - 01/12/11-16:39:31 - procmon ProcMonInvoke.cpp(269) [procmon] Wait() - Command
'/etc/init.d/replicationd start' terminated with status 0 in 0 seconds
INFO   - 01/12/11-16:39:31 - replicationd REDaemon.cpp(313) [replicationd] initialize() - Signals blocked
INFO   - 01/12/11-16:39:32 - replicationd REDaemon.cpp(528) [replicationd] doInitialize() - rm cmd is : rm -
fr /snfs/tmp/replication/namespace/*
INFO   - 01/12/11-16:39:32 - replicationd REDaemon.cpp(534) [replicationd] doInitialize() - rm cmd is : rm -
fr /Q/partitions/namespace_*
INFO   - 01/12/11-16:39:32 - replicationd BfstV2.cpp(67) [replicationd] initialize() - Bfst initialize needed -
calling bfst2_initialize
INFO   - 01/12/11-16:39:32 - replicationd REDaemon.cpp(623) [replicationd] startAllChildThreads() -
Launching Scheduler Thread
INFO   - 01/12/11-16:39:32 - replicationd REDaemon.cpp(627) [replicationd] startAllChildThreads() -
Launching Command Thread
INFO   - 01/12/11-16:39:32 - VpMsg.NameService VpNameService.cc(312) [replicationd] registerName() -
Registered NS_TRIGGERD, host=Qnode1, port=60373, pid=27881, ppid=1.
INFO   - 01/12/11-16:39:32 - procmon ProcMonService.cpp(893) [procmon] StartServices() - Service
'replicationd' is started
[root@DXi75-nh1 ~]#
```

What's Next?

Locating Active Directory Issues >

Locating Active Directory Issues

Page Notes (2) Attachments (0)

Overview

If the customer cannot connect to an Active Directory domain, you need to figure out why. For example, you may need to look at how to check the group policy, administer access to NAS shares, and so on. You should also make sure they are using Active Directory for the right reasons, and if they are not, get them to go back to workgroup.

For answers to these questions and more, review the Active Directory (this link takes you out of the **DXi Log Reading Basics** topic) presentation.

What's Next?

Locating Reconcile Information in Tsunami Log >

Locating Reconcile Information in Tsunami Log

Page Notes (0) Attachments (0)

Overview

The **tsunami log** is a good place to look for reconcile information. **Reconciliation** (stage 2 of space reclamation) is the process of comparing the **blockpooltags** and **referencedtags** files to find candidates for deletion.

Example tsunami.log Output for a Successful Reconcile

```
INFO - 06/24/08-12:59:03 - bpgc bpgc.cpp(165) [bpgc] runDaemon() - Starting the bpgc
daemon
INFO - 06/24/08-12:59:03 - bpgc CleanupStrategyFactory.cpp(128) [bpgc] create() -
Selecting the Reconcile-Only Strategy
INFO - 06/24/08-12:59:03 - bpgc CleanupReconcileStrategy.cpp(127) [bpgc] postRun() -
Sleeping for 43200 seconds
INFO - 06/24/08-12:59:03 - bpgc SchedulerThread.cpp(261) [bpgc] doProcess() -
Reloading GC Schedule
```

Long File Listing of /snfs/common/data/hurricane/bpgc

The **deleteCandidates** files are not included in the **collect.txt**. You can only review the byte size of these files by using the CLI. The location is **/snfs/common/data/hurricane/bpgc.** The **ls -al** command below gives a long file listing of this directory.

Note: If you see *WARN* or *ERROR* during blockpool garbage collection, it means that Reconcile had a problem.

```
# ls -al
total 47232
drwxrwxrwx 2 root root 2429 May 1 10:44 .
drwxr-xr-x 5 root root 28296 May 1 05:44 ..
-rw-r--r-- 1 root root 25791427 May 1 10:44 blockpooltags
-rw-r--r-- 1 root root 0 May 1 10:44 deleteCandidates_Stream0
-rw-r--r-- 1 root root 0 May 1 10:44 deleteCandidates_Stream1
-rw-r--r-- 1 root root 20483629 May 1 10:44 referencedtags
```

Note: In this listing, both **deleteCandidates_Stream0** and **deleteCandidates_Stream1** have a file length of 0 bytes. If the size of either file were greater than 0, Space Reclamation would have Candidates to delete.

Delete Candidates

In DXi software release 1.1 and later, **tsunami.log** provides the percentage completed, the number of streams, and how many candidates need to be deleted. The **tsunami_trace.log** file provides the tag numbers that are being deleted. The Reconcile process runs every 12 hours to find eligible tags to add into the **deleteCandidates** file. This is a part of Space Reclamation function.

What's Next?

Locating General Concerns about Slow Performance >

Locating General Concerns about Slow Performance

Page Notes (0) Attachments (0)

Overview

Start by looking in the **messages log** to see what's been going on with the system. Read more about troubleshooting performance issues in the Performance Troubleshooting Pages (this link takes you out of the **DXi Log Reading Basics** topic).

What's Next?

Locating Results of a Healthcheck and Missing Tags >

Locating Results of a Healthcheck and Missing Tags

Overview

The healthcheck identifies tags that are referenced in the namespace or in replicated namespaces, but that do not exist in the blockpool. This could lead to data loss. When this happens, it is possbile that garbage collection removed the blockpool tag or that namespace replication has failed. If missing tags are found, a RAS ticket is generated, stating how many references to tags are missing.

Note the following:

- A healthcheck can be run manually, or scheduled daily or weekly from the the DXI GUI. The GUI provides a RAS Ticket if the healthcheck fails.
- Running the CLI blockpool command **ispresent** with the tag number provides additional information about the tag.
- If the .**q_frg** data has not been truncated, it's possible to retrieve it with a blockpool **store** command. If it has been truncated, it can be recovered from the target replication system using the blockpool **replicate** command.

Healthcheck Information for Missing Tags within srvcLog.hist

From the diagnostics file:
less scratch/collect/node1-collection/app-info/pse_snapshot/usr/adic/SRVCLOG/logs/srvcLog.hist

From the system:
less /usr/adic/SRVCLOG/logs/srvcLog.hist

EXAMPLE OUTPUT
```
May 26 07:02:12 2009 66 blackjack 66 Healthcheck UNKNOWN 1914691300 51 De-Duplication
Healthcheck found 1 references to tag(s) that are not in the blockpool. Download the
system diagnostics file and contact support.
```

What does this example output tell you?
The message indicates that Healthcheck identified that one tag/blob is missing in the blockpool, as shown in the second line.

Missing Tags Found in Healthcheck - missingTags file

From the diagnostics file:
less scratch/collect/node1-collection/app-info/healthcheck-missingtags

From the system:
less /snfs/healthcheck/missingTags

EXAMPLE OUTPUT
```
CA44F5649B4018C9EEBCB734F1C272D2 sus1-lib1
partitions/VL04CX0910BVA00149/Carts/SU0001_CX0910BVA00149_625753/.q_frg/data/1
```

What does this example output tell you?

Note: The first line gives you the tag number. The second line is the actual file path that the tag belongs to. The **missingTags** file provides details about the missing tag. The tag may belong to a share or partition or to a replication namespace bundle (target).

What's Next?

Recognizing an Unclean Shutdown >

Recognizing an Unclean Shutdown

Overview

An unclean shutdown will cause a blockpool verification to run when a reboot occurs. It is typically fixed automatically on reboot. The following shows the results of an unclean shutdown and clean reboot. If there's an unclean shutdown, the **messages log** will say unclean shutdown in the blockpool startup section. If it's a clean reboot, the unclean shutdown message will not exist in the **messages log**.

Unclean Shutdown

In System Diagnostics: **scratch\collect\node1-collection\os-info\messages**
On the system: **/var/log/messages**

The following output shows an example of an unclean shudown in a **messages log**. The system was unable to shutdown services cleanly. We see an abrupt restart which would indicate that the system was hard reset.

```
Unclean Shut
Mar 15 07:28:48 qtmdxiwalpd01 winbindd[3395]: [2009/03/15 07:28:48, 0]
libads/kerberos.c:ads_kinit_password(228)
Mar 15 07:28:48 qtmdxiwalpd01 winbindd[3395]: kerberos_kinit_password
QTMDXIWALPD01$@CORP.AD.FMCNA.COM failed: Client not found in Kerberos database
Mar 15 07:46:53 qtmdxiwalpd01 kernel: [479081.731083] WARNING: pscsi_checker [scsi15:
id: 0 lun: 169]: application has not been responding over 5 minutes -
check /proc/pscsi/iostat to see the cdbs and TMFs!
Mar 15 07:46:53 qtmdxiwalpd01 kernel: [479081.771245] WARNING: pscsi_checker [scsi15:
id: 0 lun: 172]: application has not been responding over 5 minutes -
check /proc/pscsi/iostat to see the cdbs and TMFs!
Mar 15 08:21:56 qtmdxiwalpd01 syslogd 1.4.1: restart.
Mar 15 08:21:56 qtmdxiwalpd01 syslog: syslogd startup succeeded

Mar 15 08:21:56 qtmdxiwalpd01 kernel: klogd 1.4.1, log source = /proc/kmsg started.
Mar 15 08:21:56 qtmdxiwalpd01 kernel: [ 0.000000] Linux version 2.6.19-2smp
(jali@white-galaxy) (gcc version 3.4.6 20060404 (Red Hat 3.4.6-3)) #1 SMP Tue Dec 2
18:14:01 PST 2008
```

Clean Shutdown

The following output shows an example of a clean shudown. We can see service shutdowns completing successfully and the system "exiting on signal 15" and restarting.

```
Mar 9 14:57:47 qtmdxiwalpd01 ipmi: Stopping ipmi drivers:
Mar 9 14:57:47 qtmdxiwalpd01 ipmi: succeeded
Mar 9 14:57:47 qtmdxiwalpd01 ipmi: ESC[60G
Mar 9 14:57:47 qtmdxiwalpd01 ipmi:
Mar 9 14:57:47 qtmdxiwalpd01 rc: Stopping ipmi: succeeded
Mar 9 14:57:47 qtmdxiwalpd01 irqbalance: irqbalance shutdown succeeded
Mar 9 14:57:47 qtmdxiwalpd01 portmap: portmap shutdown succeeded
Mar 9 14:57:47 qtmdxiwalpd01 slpd: slpd shutdown succeeded
Mar 9 14:57:47 qtmdxiwalpd01 kernel: Kernel logging (proc) stopped.
Mar 9 14:57:47 qtmdxiwalpd01 kernel: Kernel log daemon terminating.
Mar 9 14:57:48 qtmdxiwalpd01 syslog: klogd shutdown succeeded
Mar 9 14:57:48 qtmdxiwalpd01 exiting on signal 15
```

```
Mar 9 14:59:47 qtmdxiwalpd01 syslogd 1.4.1: restart.
Mar 9 14:59:47 qtmdxiwalpd01 syslog: syslogd startup succeeded
Mar 9 14:59:47 qtmdxiwalpd01 kernel: klogd 1.4.1, log source = /proc/kmsg started.
Mar 9 14:59:47 qtmdxiwalpd01 kernel: [ 0.000000] Linux version 2.6.19-2smp
jali@white-galaxy) (gcc version 3.4.6 20060404 (Red Hat 3.4.6-3)) #1 SMP Thu Feb 28
08:14:50 PST 200
```

What's Next?

Determining If an Upgrade Was Successful >

Determining If a Software Upgrade Was Successful

Page Notes (0) Attachments (0)

Overview

The only way to verify that a DXi software upgrade did not complete is to review the **verboseupgradeoutput log**.

Location

*In System Diagnotics:***/scratch/collect/node1-collection/app-info/verboseupgradeoutput**
On the system: **/var/log/DXi/verboseupgradeoutput**

In 1.4.x, look at **/var/log/DXi/upgrades/verboseupgradeoutput.<date>**

EXAMPLE OUTPUT
Here are the last entries in the **verboseupgradeoutput log**:

```
Initiating stop of ADIC DSM component
Unmounting SNFS filesystems
Stopping SNFS Daemons
Stopping SNFS PortMapper
Waiting for FSMs to finish.
SNFS Stop                                                  [  OK  ]
Jan 28 10:34:03: Starting up limited mode GUI
Jan 28 10:34:08: Stopping GUI
Jan 28 10:34:08: Removing local RPMS from /scratch/upgrade
```

For a good upgrade, the last line should be *Removing local RPMS from /scratch/upgrade,* as shown here.

Note: This log file will keep all occurrences of every upgrade preformed on the system. All information is contained in this single file.

What's Next?

Reviewing a Blockpool Verify for Blockpool Corruption >

Reviewing a Blockpool Verify for Blockpool Corruption

Page Notes (0) Attachments (0)

Overview

You can check a blockpool verify in the **messages log**. A blockpool verify will check the integrity of the blockpool structure.

When you list the contents of the **messages log**, you should see the following sections (as worded below):

- Verify blob tree heap
- Index load
- Verify cluster tree
- Verify clusters
- Verify blob tree heap free list
- Verify blob tree data lists
- Verify blocklet lists

You might suspect blockpool corruption in the following cases:

- Replication status shows that replication has failed.
- A healthcheck failed, and blob are missing or corrupted.
- There was a power outage, and the unit does not boot up properly or SNFS is not mounting.
- There was a storage array issue, and SNFS is not mounting.
- There was a filesystem PANIC, and the unit is not responding.
- The unit is in a “hung” state.

You can check the Blockpool Verify status by running:
/hurricane/blockpool/bin/blockpool status +Nlocal@localhost

What's Next?

Locating Serial Numbers >

Locating Serial Numbers

Overview

At times, you may need to find the serial number for a hardware component or for the DXi system itself. You can do this by looking a the **current.config** file.

In the system diagnostics file: **scratch\collect\node1-collection\hw-info\current.config**
On the system: **/opt/DXi/hwdetect/current.config**

What's Next?

When to Escalation to Sustaining >

When to Escalate to Sustaining

Page Notes (0) Attachments (0)

Overview

The information below shows how to find out when DXi logs should be escalated to the Sustaining Engineering group, what types of issues may occur, what information needs to be collected, and the escalation steps.

Note: Every service request escalation requires the capture of DXi logs. It's best to capture these logs immediately after failure, or as soon as possible thereafter.

When to Contact Sustaining Engineering

Before escalating an issue to Sustaining Engineering, make sure that you have completed these steps:

1. Review DXi messages logs, **tsunami.log**, LSI logs, 3ware output, etc., as needed.
2. Verify that DXi Best Practices have been applied.
3. Use available resources:
    - Quest (this search engine searches the next 4 items:
    - Bugzilla for related PTRs
    - CSweb for any related TSBs
    - Oracle SR history
    - KB articles
    - DXi Sustaining TWIKI pages:
        - http://boris.adic.com/twiki/bin/view/DxiSustaining/DxiTipsAndTricks
        - Tips and Notes according to product. Examples:
            - http://boris.adic.com/twiki/bin/view/Galaxy/DebuggingGalaxy
            - http://boris.adic.com/twiki/bin/view/Eclipse/WebHome
            - The Available Information section in the DXi Sustaining Web, “Misc” section: http://boris.adic.com/twiki/bin/view/DxiSustaining/WebHome
    - The ASPS Web site: http://10.20.226.49/ASPS/wp/
4. Consult peers, ask others in ASPS Team, and e-mail Service Engineering Specialist (SES).

It is acceptable practice to contact SES if your question will not take more than 10 minutes. For issues longer than 10 minutes, you will need to search for the information in some of the other sources described in this document, or work through the escalation process. Times could vary, depending on the customer’s temperature and management's input.

Rules of Thumb

Good rules of thumb about when a call to the next level are appropriate are as follows:

- Can the question with any needed context be explained/asked in a few short sentences?
- Does the question have a yes or no answer? If not, is the answer likely to be short?

Questions requiring long log listings or several paragraphs in an e-mail are tip offs that a formal escalation is probably the right way to go.

What Needs to be Sent to Sustaining

It's important to send the Sustaining team information found in the escalation form to the extent it can be determined. Review an example list of those items.

1. Gather the appropriate data, according to the issue, and provide a detailed problem description, including the following:
    - Timestamps of events
    - OS versions
    - Backup application name and version
    - Switch, HBA, or NIC information
    - A description about whether or not the problem is reproducible
    - A description about what has been done to solve the problem
2. Gather additional details:
    - DXi Issues:
        - DXi logs from target and source
        - **dbexport** from target and source (DXi Advanced Reporting Tool)
        - Core files in **/scratch/core**
        - **blockpool_master.log** (replication, deduplication, blockpool issues)
        - Blockpool quick report (replication, deduplication, blockpool issues)
        - LSI logs (array, performance, FC communication issues)
        - **Kdump** (Kernel Panic) **/snfs/kdump**
    - Ecosystem (connectivity, network, I/O performance issues):
        - Backup application error information and logs
        - OS information (UNIX, Windows), server logs
        - SAN/LAN configuration (diagram if possible)
            - Fabric, switch, and FC HBA information, speed, etc
            - Ethernet technology (Gigabit, 100BaseT, etc.), NIC information,
            - CIFS, NFS, Windows Active Directory, Workgroup

What's Next?

DXi Log Reading Summary >

DXi Log Reading Summary

Page Notes (0) Attachments (0)

You can download system logs as well as storage array logs to troubleshoot DXi issues. The system logs include information that is specific to the DXi processes while the storage array logs contain information that is specific to the LSI RAID arrays. This training resource provided basic information about using DXi system logs to troubleshoot issues and when to escalate to Sustaining.

Use the DXi-Series Status and Log Collection Job Aid and the DXi-Series Log Location and Description Job Aid as references.

What's Next?

Assessment >

Assessment

Page Notes (0) Attachments (0)

To get credit for completing this training, you must register for and complete a short assessment (exam) in StorageCare Learning. You can use the information in Qwikipedia and any of the course resources while taking the assessment.

To access and complete the assessment, do the following

1. Go to StorageCare Learning and log in.
2. Search for assessment number **A-3-2420**.
3. Click **Register** to register for the assessment.
4. Click **Submit** to submit your registration request.
5. Click **Start** to take the assessment.
6. Follow the instructions to complete the assessment.