Replication Monitoring

Introduction

NexentaStor (www.nexenta.com) is a software-based unified storage appliance. In addition to providing file and block access to storage, NexentaStor includes advanced replication capabilities. Multiple replication services can be enabled, and they can be individually scheduled. NexentaStor offers two types of replication services: AutoSync and AutoTier. AutoSync was fully developed by Nexenta, and AutoTier is built on top of the industry-standard rsync capability.

End users would like to be able to monitor the status and performance of their replication services. AutoSync has some monitoring capabilities, but AutoTier does not.

This document proposes some improved monitoring of the AutoTier service.

Related Documentation

A free trial of NexentaStor is available at http://www.nexenta.com/products/downloads/nexentastor.

The Nexenta User Guide is available at http://info.nexenta.com/rs/nexenta/images/NexentaStor-4-0-2-User-Guide.pdf.

The PerfMon User Guide is available at:

Desired Functionality

Functionality can be divided into three categories:

1. Track statistics that show how well the replication service is performing
2. Alarm if the replication service is not performing optimally
3. Detect if a remote replication service has failed, and restart the replication service if necessary.

Development Notes

The expectation is that software development will be done in C or C++ and will run on a Solaris-based operating system.

Statistics

The following statistics are the highest priority to collect:

- filesTransferred

- filesExamined
- bytesSent
- totalTime

These statistics are available when logging is turned on for the rsync job. The replication monitor will periodically empty the log file (mv file somewhere else), calculate the statistics and write the resulting statistics to the performance database.

The following network statistics should be provided for each network interface:

- BytesOut/sec
- OutErrors/sec

The third-party PerfMon monitoring package will provide the network statistics. The replication monitor will subscribe for network statistics, periodically retrieve them, and write them to the performance database.

Writing to the performance database

A new library routine will be needed similar to the following:

```
updatePerfDB (string filename, service, statArray *stats) {

   if (!fileExists(filename)) {

      createPerfDB (filename,service);

   }

   Foreach array entry

       System (“perfdb.exe update filename time:stat1:stat2”).

   }

}
```

Alarm

The monitor should detect the following alarm conditions:

1. Replication service is not running
2. Replication service did not start at the scheduled time
3. Replication service has not run in a time greater than interval X

Shell commands such as “nmc –c ‘show auto-tier :svcA –v`” can be used to get detailed status information about the replication service.

Alarms should be written to the system log file.

It should also be possible to query the monitor for status. The output should be in the format suitable for Nagios. Here are some sample responses to queries:

> OK Service: auto-tier-svcA running since 03/04/2014 05:34:00
>
> CRITICAL Service: auto-tier-svcA not online
>
> WARNING Service: auto-tier-svcA did not start at its last scheduled start time. Last started 03/02/2014 02:11:33.
>
> CRITICAL Service: auto-tier-svcA not running. Last started 03/02/2014 02:11:33.

When written to the system log, the status would map to “err” or “warn” and be shown as the severity in the system log.

Restart Replication Service

NexentaStor may run on an HA Cluster. If a system fails, the other cluster system will take over the services. However, if AutoTier is being used, and the replication source system fails, the failure is not detected. The service needs to be restarted manually.

A new program repHB (replication heartbeat) should monitor the remote system (via ping etc.) and detect that the system is no longer available. This can be further validated by validating that the HA cluster VIP has moved to the secondary system. Once this is verified, the AutoTier services previously running should be restarted on the alternate node.

Additional Technical Notes

Administering the AutoTier service

NexentaStor provides a CLI called NMC to administer the storage system. The following example commands can be used to administer a replication service:

> setup auto-tier :bradtest-000 disable
>
> setup auto-tier :bradtest-000 enable
>
> setup auto-tier :bradtest-000 destroy -y

To invoke these commands from the Solaris shell instead of NMC, you can wrap the command in “nmc –c”.

This script is an example of creating a replication service from the shell:

nmc -c "setup auto-tier create -y -u bradtest -i minute -p 10 -r 2 -s rsync+ssh:

//nza311/volumes/ost/test/* -d dd/rp -I myha -a -R no"

Rsync supports additional options (-o options). We need to enable additional options for statistics:

"--stats --logfile=<fileName>" will write statistics to a file

Reading the statistics log file

Rsync writes to the log file specified when the replication service was started. Sample output is shown in the Appendix “Sample rsync Log Output”.

When reading the file, the repMonitor should first move the file to a new location. Rsync can then continue writing while we process the statistics. The monitor should then calculate the statistics and write to a performance database.

But in some cases the replication job may be in the middle of running. In that case the monitor has removed some data from the log, but can’t fully process it. This data should be buffered so that it can be used the next time the monitor reads the log. Also, this buffer should be persisted to a file in case the monitor fails. The API getConfigRoot() can be used to locate a shared directory so that the file can be recovered if the system fails.

Summary of Inputs and Outputs

Summary of commands: start, checkServiceStatus

The replication monitor should automatically start monitoring logs when it starts. Logs will be in a known directory such as /var/log/replication/stats/svcname (default) or specified with logDir=X option on command line.

The monitor will periodically (interval specified in seconds on command line with interval=X) run and retrieve logs for each service and calculate statistics for each completed job. A job is completed if it has the final log entry indicating the bytes sent. Statistics are written with the updatePerfDB() call.

As part of its periodic execution, the monitor should also check for alarm condition and write to the system log file as needed.

The monitor should also be prepared to be invoked with a checkServiceStatus call. This will check for the alarm conditions noted above.

Appendix: Sample rsync Log Output

2014/05/06 05:47:39 [21966] building file list

2014/05/06 05:47:39 [21966] cd++++++++++ mnt/

2014/05/06 05:47:39 [21966] sent 32 bytes received 16 bytes total size 0

2014/05/06 05:47:39 [21969] building file list

2014/05/06 05:47:39 [21969] >f+++++++++ genunix

2014/05/06 05:47:40 [21969] sent 2900585 bytes received 32 bytes total size 2900148

2014/05/06 05:47:50 [22027] building file list

2014/05/06 05:47:50 [22027] sent 29 bytes received 13 bytes total size 0

2014/05/06 05:47:50 [22030] building file list

2014/05/06 05:47:50 [22030] sent 42 bytes received 13 bytes total size 2900148

2014/05/06 05:49:12 [22342] building file list

2014/05/06 05:49:12 [22342] Number of files: 1

2014/05/06 05:49:12 [22342] Number of files transferred: 0

2014/05/06 05:49:12 [22342] Total file size: 0 bytes

2014/05/06 05:49:12 [22342] Total transferred file size: 0 bytes

2014/05/06 05:49:12 [22342] Literal data: 0 bytes

2014/05/06 05:49:12 [22342] Matched data: 0 bytes

2014/05/06 05:49:12 [22342] File list size: 22

2014/05/06 05:49:12 [22342] File list generation time: 0.001 seconds

2014/05/06 05:49:12 [22342] File list transfer time: 0.000 seconds

2014/05/06 05:49:12 [22342] Total bytes sent: 29

2014/05/06 05:49:12 [22342] Total bytes received: 12

2014/05/06 05:49:12 [22342] sent 29 bytes received 12 bytes 82.00 bytes/sec

2014/05/06 05:49:12 [22342] total size is 0 speedup is 0.00

2014/05/06 05:49:12 [22342] sent 29 bytes received 13 bytes total size 0

2014/05/06 05:49:12 [22345] building file list

2014/05/06 05:49:12 [22345] Number of files: 2

2014/05/06 05:49:12 [22345] Number of files transferred: 0

2014/05/06 05:49:12 [22345] Total file size: 2900148 bytes

2014/05/06 05:49:12 [22345] Total transferred file size: 0 bytes

2014/05/06 05:49:12 [22345] Literal data: 0 bytes

2014/05/06 05:49:12 [22345] Matched data: 0 bytes

2014/05/06 05:49:12 [22345] File list size: 35

2014/05/06 05:49:12 [22345] File list generation time: 0.001 seconds

2014/05/06 05:49:12 [22345] File list transfer time: 0.000 seconds

2014/05/06 05:49:12 [22345] Total bytes sent: 42

2014/05/06 05:49:12 [22345] Total bytes received: 12

2014/05/06 05:49:12 [22345] sent 42 bytes received 12 bytes 108.00 bytes/sec

2014/05/06 05:49:12 [22345] total size is 2900148 speedup is 53706.44

2014/05/06 05:49:12 [22345] sent 42 bytes received 13 bytes total size 2900148

2014/05/06 05:49:46 [22459] building file list

2014/05/06 05:49:46 [22459] .d..t..... mnt/

2014/05/06 05:49:46 [22459] Number of files: 1

2014/05/06 05:49:46 [22459] Number of files transferred: 0

2014/05/06 05:49:46 [22459] Total file size: 0 bytes

2014/05/06 05:49:46 [22459] Total transferred file size: 0 bytes

2014/05/06 05:49:46 [22459] Literal data: 0 bytes

2014/05/06 05:49:46 [22459] Matched data: 0 bytes

2014/05/06 05:49:46 [22459] File list size: 22

2014/05/06 05:49:46 [22459] File list generation time: 0.001 seconds

2014/05/06 05:49:46 [22459] File list transfer time: 0.000 seconds

2014/05/06 05:49:46 [22459] Total bytes sent: 32

2014/05/06 05:49:46 [22459] Total bytes received: 15

2014/05/06 05:49:46 [22459] sent 32 bytes received 15 bytes 94.00 bytes/sec

2014/05/06 05:49:46 [22459] total size is 0 speedup is 0.00

2014/05/06 05:49:46 [22459] sent 32 bytes received 16 bytes total size 0

2014/05/06 05:49:46 [22463] building file list

2014/05/06 05:49:47 [22463] >f+++++++++ big

2014/05/06 05:49:47 [22463] Number of files: 2

2014/05/06 05:49:47 [22463] Number of files transferred: 1

2014/05/06 05:49:47 [22463] Total file size: 2900148 bytes

2014/05/06 05:49:47 [22463] Total transferred file size: 2900148 bytes

2014/05/06 05:49:47 [22463] Literal data: 2900148 bytes

2014/05/06 05:49:47 [22463] Matched data: 0 bytes

2014/05/06 05:49:47 [22463] File list size: 35

2014/05/06 05:49:47 [22463] File list generation time: 0.001 seconds

2014/05/06 05:49:47 [22463] File list transfer time: 0.000 seconds

2014/05/06 05:49:47 [22463] Total bytes sent: 2900585

2014/05/06 05:49:47 [22463] Total bytes received: 31

2014/05/06 05:49:47 [22463] sent 2900585 bytes received 31 bytes 1933744.00 bytes/sec

2014/05/06 05:49:47 [22463] total size is 2900148 speedup is 1.00

2014/05/06 05:49:47 [22463] sent 2900585 bytes received 32 bytes total size 2900148