



BSnapDiff for ZFS Plugin

Bacula Systems Documentation

Contents

1 Overview	3
1.1 Features Summary	3
2 Installation	3
2.1 Packages	3
3 Configuration	3
3.1 Accurate mode in Job Resource	3
3.2 Enable snapshots in Fileset Resource	4
3.3 File Daemon Configuration	4
4 Snapshot Management	4
5 ZDB Scan Phase	5
6 Estimating Performance	5
6.1 Ubuntu 22 with 32MB RAM	5
6.2 FreeBSD 32MB RAM	5
6.3 Results	6
7 Limitations	6
7.1 Fileset Directive: ExcludeDirContaining	6
7.2 Fileset Directive: HonorNoDump flag	6
8 Handling ZFS Missing Parent Dinodes	6

Contents

Note

You can download this article as a [PDF](#)

- *Overview*
- *Installation*
- *Configuration*
- *Snapshot Management*
- *ZDB Scan Phase*
- *Estimating Performance*
- *Limitations*
- *Handling ZFS Missing Parent Dinodes*

1 Overview

1.1 Features Summary

Bacula Enterprise Edition now supports BSnapshot technology for the ZFS file system on Linux and FreeBSD backup clients. BSnapshot is a technology that compares two snapshots of a file system to quickly and accurately determine the changes required for backup. For the ZFS file system, the native `zfs diff` command is integrated into bsnapshot. The File Daemon uses the results to properly back up changes to the file system without the need to scan the entire file system. When few changes have occurred since the last backup, BSnapshot can report differences very quickly, regardless of the total number of files in the file system.

The feature requires the BSnapshot for ZFS File Daemon Plugin to be installed. When the plugin detects that it is backing up a ZFS file system, it automatically enables the BSnapshot feature. All backups automatically create and use ZFS snapshots. During the initial Full backup, additional checks are performed to ensure data consistency and to guarantee that subsequent backups correctly detect all changes.

2 Installation

In order to use the BSnapshot plugin, bsnapshot and the BSnapshot for ZFS Plugin must be installed. No additional configuration is required. Bsnapshot will call the BSnapshot for ZFS Plugin if it is available.

2.1 Packages

Packages of the **BSnapshot** Plugin are available for supported platforms. Please contact Bacula Systems Support team to obtain them.

Download the **BSnapshot** Plugin package to your server where the Bacula File Daemon is installed, then install it using your system's package manager.

Debian/Ubuntu

```
dpkg -i bacula-enterprise-bsnapshot-plugin*.deb
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the **BSnapshot** Plugin.

RHEL

```
rpm -ivh bacula-enterprise-bsnapshot-plugin*.rpm
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the **BSnapshot** plugin.

3 Configuration

3.1 Accurate mode in Job Resource

In order to manage hard links properly, the `Accurate = yes` must be defined in the Job resource of the Director for jobs that include ZFS file systems.

```
Job {  
  Name = "ZfsJob"
```

(continues on next page)

(continued from previous page)

```
Client = storage-fd
Fileset = ZfsFileSet
...
Accurate = yes
}
```

If Accurate mode is not enabled, or is explicitly disabled by setting `Accurate = no`, backups will not benefit from the BSnapDiff Plugin. In this case, the backup log will include the following message:

```
bsnapdiff is available but accurate mode is not enabled, running regular
->filesystem backup
```

3.2 Enable snapshots in Fileset Resource

The BSnapDiff Plugin also requires snapshots to be enabled in the target file systems. Set `EnableSnapshot = yes` in the Fileset resource(s) where BSnapDiff is used.

```
Fileset {
  Name = ZfsFileSet
  EnableSnapshot = yes
  Include {
    Options {
      Signature = MD5
    }
  }
  File = /path/to/backup
}
```

If snapshots are not enabled, or are explicitly disabled by setting `EnableSnapshot = no`, backups will not use ZFS snapshots and will not take advantage of the BSnapDiff plugin. No log message will be generated, since the BSnapDiff plugin is never invoked when snapshots are disabled.

3.3 File Daemon Configuration

On the File Daemon host, the **BSnapDiff** Plugin is enabled by default. It can be disabled in the `/opt/bacula/etc/bsnapshot.conf` file with:

```
bsnapdiff=no
```

Once the plugin is enabled, snapshots are enabled, and Accurate mode is enabled, the backup will report a log message:

```
Snapshot incremental accelerator BSnapDiff for ZFS enabled
```

4 Snapshot Management

ZFS snapshot management is handled automatically. Snapshot history is stored in `/opt/bacula/etc/bsnapdiff.dat`. Snapshots are retained until they are no longer needed. If a user manually removes any snapshots currently required by the system, a standard, non-accelerated incremental backup will be performed by scanning the file system. Once snapshot history is re-established, the plugin will resume performing incremental backups more efficiently.

5 ZDB Scan Phase

When BSnapDiff is first enabled for backing up a ZFS file system, a ZDB Scan is activated and a regular Full backup is performed. The log file will report:

```
bsnapdiff is available, running a regular full filesystem backup
```

If the file system was backed up by Bacula prior to installing the plugin, or if any issues are detected with the snapshot history, a regular Incremental backup is performed while the ZDB Scan phase is activated. The log file will report:

```
bsnapdiff is available, running a regular incremental filesystem backup
```

During this step, the live file system is checked for files that are associated with an invalid parent **dinode** - a known issue in ZFS. The plugin checks for, repairs, and reports these errors during this backup. The log file will report messages similar to:

```
zdb_fix_missing_dinodes backup relinked 9 files
```

See also *Handling ZFS Missing Parent dinodes*.

Once a backup is completed using the ZDB Scan phase, it is no longer required. Incremental Backups with BSnapDiff enabled will continue to check for orphaned files, but with significantly reduced overhead.

6 Estimating Performance

The tables below include timing in seconds for a series of backups with the `Accurate = yes` directive in the Job resource and `EnableSnapshot = yes` directive in the Fileset resource. The client ZFS file system in the examples below contains 12.5 million small files.

6.1 Ubuntu 22 with 32MB RAM

Total	Accurate Setup	ZDB Scan	Scan	Backup Type
1177	0	0	1103	Full, BSnapDiff disabled
633	168	0	465	Incremental, BSnapDiff disabled
915	165	172	578	Incremental, BSnapDiff enabled
218	165	0	53	Incremental, BSnapDiff enabled

6.2 FreeBSD 32MB RAM

Total	Accurate Setup	ZDB Scan	Scan	Backup Type
1231	0	0	1150	Full, BSnapDiff disabled
684	106	0	577	Incremental, BSnapDiff disabled
1148	104	393	651	Incremental, BSnapDiff enabled
139	104	0	34	Incremental, BSnapDiff enabled

6.3 Results

The first two backups in this example are regular snapshot backups with the BSnapDiff Plugin installed but disabled. During the third backup, the plugin and the ZDB Setup phase are both enabled. The final Incremental is taken with BSnapDiff active and shows the improved backup performance.

An Incremental backup with BSnapDiff enabled runs is 3x to 5x faster than an Incremental backup with the feature disabled. This performance gain is primarily achieved by eliminating the overhead of reading unchanged directories and performing `stat` operations on a large number of unchanged files. Reducing unnecessary metadata lookups also helps keep file system caches cleaner, improving response times for other users of the file system.

These tests were performed on virtual machines running on SSD storage. Each system functioned as an all-in-one Bacula test environment, with the Director, Storage Daemon and File Daemon running locally. The ZFS file system was created as a flat file with `/bin/dd`. In a large production environment, a single ZFS server may host multiple, large file systems and millions or even billions of files. In such cases, performance improvements can be even more significant. For example, at one customer site, a regular Incremental backup that previously required 14 hours was reduced to just one hour with BSnapDiff enabled.

7 Limitations

7.1 Fileset Directive: ExcludeDirContaining

The BSnapDiff Plugin generates a list of file and directory changes in a random order. Sometimes, a file change may appear before one or more of its parent directories. As a result, the `ExcludeDirContaining` directive cannot be used with the plugin. If this directive is used in a Fileset that includes one or more ZFS volumes with the BSnapDiff Plugin enabled, backups will fail with the following log message:

```
bsnapdiff: Fileset option ExcludeDirContaining is not currently supported
```

7.2 Fileset Directive: HonorNoDump flag

As stated above, the BSnapDiff Plugin generates a list of file and directory changes in a random order. Because a file may be listed before its parent directory, the `HonorNoDump` directive cannot be used with the plugin. If a user sets this directive to `yes` in a Fileset that includes one or more ZFS volumes with the BSnapDiff Plugin enabled, backups will fail with the following log message:

```
bsnapdiff: Fileset option HonorNodumpFlags=yes is not currently supported
```

8 Handling ZFS Missing Parent Dinodes

When hard links are created across multiple directories, removing the last hard link created can result in missing parent dinodes for the remaining links. Any subsequent modifications to those files will not be detected by `zfs diff`. The BSnapDiff Plugin for ZFS identifies and corrects these errors.

During any backup operation, Bacula may detect files that are missing their internal parent directory status. When found, the Bacula File Daemon repairs the parent directory status by temporarily creating and then removing a hard link to the file on the live file system (not the snapshot). Although this behavior is unusual for a backup process, it is required to ensure that future changes to the file are properly recognized by `zfs diff`.

A summary of repaired files, if any, is reported in the backup log:

```
zdb_fix_missing_dinodes bsnapdiff relinked 16 files
```

This behavior is specific to the ZFS file system.