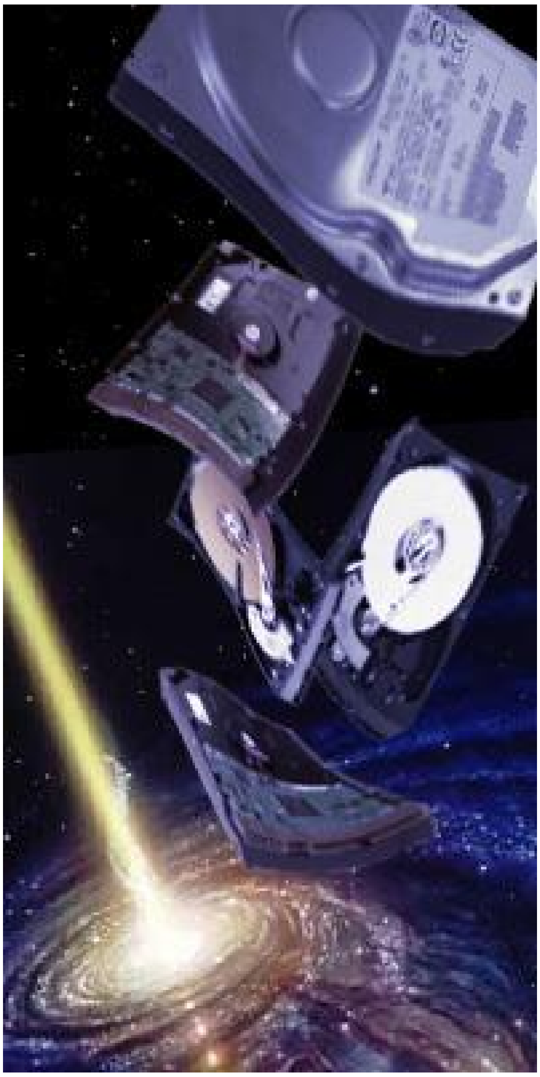
BlackHole



BlackHole is a storage solution. It does data-deduplication, mirroring (sync and async), snapshots, in-line compression and encryption.

Written by Folkert van Heusden.
http://www.vanheusden.com/java/CoffeeSaint/

Table of Contents

What can it do

BlackHole does:

- data deduplication
  - configurable hash
  - configurable block size
- it supports multiple storage backends
  - binairy tree
  - MySQL database
  - Kyoto Cabinet (NoSQL database) – this one gives the best performance
- in-line compression: when using the MySQL or Kyoto Cabinet storage backend, BlackHole can compress the de-duplicate blocks before storing them in the datastore, improving the disk usage even more
- encryption: all data can be stored encrypted with the algorithms supported by the Java JVM
- snapshots (in sparse files)
- async and sync mirrors
- it exports the NBD protocol: “Network Block Device”. Linux has a client build in in the kernel for mapping these to a local block device. It is filesystem agnostic. Tested with ext2/3/4, ntfs, reiserfs, etc. NBD is comparable to iSCSI.
- it also supports IMDisk protocol so that microsoft windows users can mount it as well
- multiple LUNs which share the same datastore

Using it

After configuring the program, it can be invoked as follows:

```
java -jar BlackHole.jar --config myconfig.conf
```

Then connect to it via telnet and configure the luns:

```
folkert@belle:~/$ telnet localhost 8090
Connected to localhost.
Escape character is '^]'.
BlackHole v1.9, (C) 2010-2011 by folkert@vanheusden.com


BH> addnbdlun 0.0.0.0 12341 10 lun1
Sep.04  13:02:12 [LOG_INFO] Starting NBD protocol listener for lun lun1 with number 0 at
0.0.0.0:12341
Created and started new lun with number: 0
```

Enter 'help' to get a list of commands it supports.

IMDisk support:

```
addimdisklun 0.0.0.0 12341 10 lun1
```

and then in windows:

```
c:\windows\system32\imdisk.exe -a -t proxy -o ip -f 192.168.64.100:12341 -m e:
```

Configuration file

Please note that BlackHole needs write access to the configuration file: some changes (e.g. changing the datastore size) cause it to rewrite the configuration file.

The following settings (with example values) define the storage. Block-size and hash-type cannot be changed afterwards.

```
block-size=4096
size=34359738368
hash-type=MD5
```

These settings configure the read- and delayed write cache. The cache-elements setting tells BlackHole how many blocks to cache for read access.

```
cache-elements=262144
cache-delayed-write=true
cache-type=MRU
cache-full-flush-n=16
cache-dirty-blocks=false
dwc-force-flush-threshold=131072
dwc-flush-trigger-threshold=110000
dwc-sort-before-write=true
dwc-flush-interval=2500
dwc-flush-min-n-blocks=1024
```

These settings are storage settings. Storage-type can be 'files', 'kc' and 'sql'. 'kc' (kyoto cabinet) is preferred as it gives the greatest performance. It also supports compression (which files does not). sql-... settings are only for 'SQL'. See below for what schema must be in the database.

```
path=./data2
storage-type=kc
sql-url=jdbc:mysql://localhost:3306/bh
sql-user=bh
sql-password=bh
```

The following settings configure the configuration access methods. Currently only telnet allows you to change settings

```
http-listen-adapter=0.0.0.0
http-listen-port=8089
```

```
telnet-listen-adapter=0.0.0.0

telnet-listen-port=8090
```

When BlackHole reads a block from its datastore, it can verify that it did not change due to e.g. disk-corruption. It does this by verifying the data with the stored hash.

```
verify=false
```

What encryption to use.

```
encryption-cipher=Blowfish/CFB8/NoPadding
encryption-password=...
```

Compression settings. First parameter must be 'zlib', the next (the value) must be 1 upto 9 (=best compression).

```
compression=zlib 3
```

Logging settings.

```
debug=info

logfile=log.txt

logfile-debug=info
```

async/sync iscsi/nbd host port disconnect_block storage_iqn blackhole_iqn
For nbd the iqn-settings must be omitted .
“disconnect_block” indicates if BlackHole should completely halt if the connection with the mirror is broken (until the connection is restored).

```
mirror=async iSCSI 192.168.64.201 3260 true iqn.2001-04.com.example:storage.disk2.sys1.xyz
iqn.2011-04.com.vanheusden
```

Telnet LDAP authentication:

```
ldap-base-dn=ou=users,dc=intranet,dc=vanheusden,dc=com
ldap-url=ldap://172.29.0.1:389
```

MySQL storage backend

This requires a MySQL user capable of select,insert,update and delete.

Also the following tables must be created:

```
CREATE TABLE `bm` (
  `sectornr` int(9) NOT NULL,
  `blocknr` int(9) NOT NULL default '-1',
  PRIMARY KEY  (`sectornr`),
  KEY `bm_bn` (`blocknr`)
);

CREATE TABLE `ds` (
  `blocknr` int(9) NOT NULL auto_increment,
  `hash` varchar(128) NOT NULL,
  `data` blob NOT NULL,
  PRIMARY KEY  (`blocknr`),
  KEY `ds_hash` (`hash`),
  KEY `ds_data` (`data`(16))
);
```

Do's and don'ts

- BlackHole is “a heavy seeker”: due to the nature of data deduplication, data is not lineair laid out on disk. Because of that it is better to put the datastore on an SSD devic
- You'll see that when you run a benchmark on an empty datastore that reading is extremly fast (e.g. 3GB/s). Also writing can give strange results. This again is due to the nature of data deduplication. When benchmarking make sure there's data in the datastore, with +/- 40% deduplicated. Furthermore: when using iozone, use the “-+w x” setting where x is either 40. 40% space gain is regular value for e.g. an office file server.
- When you set “disk-flush” to false, make sure you use a raid adapter with a cache-battery.