Open Source Data Protection

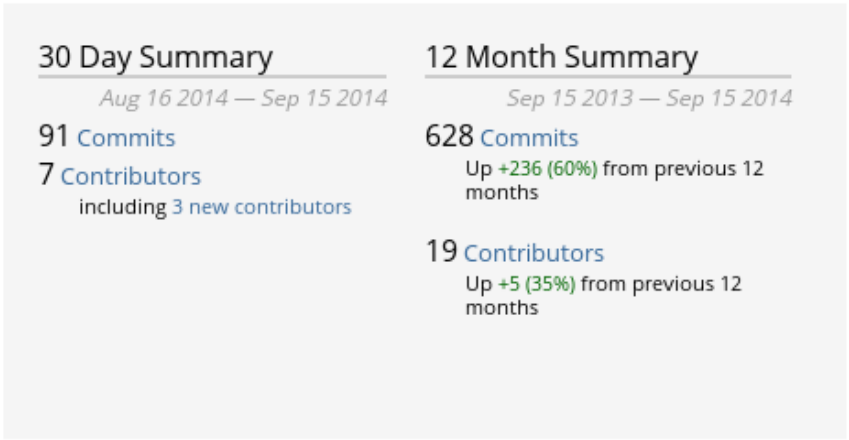
What have we done in the last year?

- OpenHub (former Ohloh) analyses our code and has some interesting numbers.
- see http://openhub.net/p/bareos

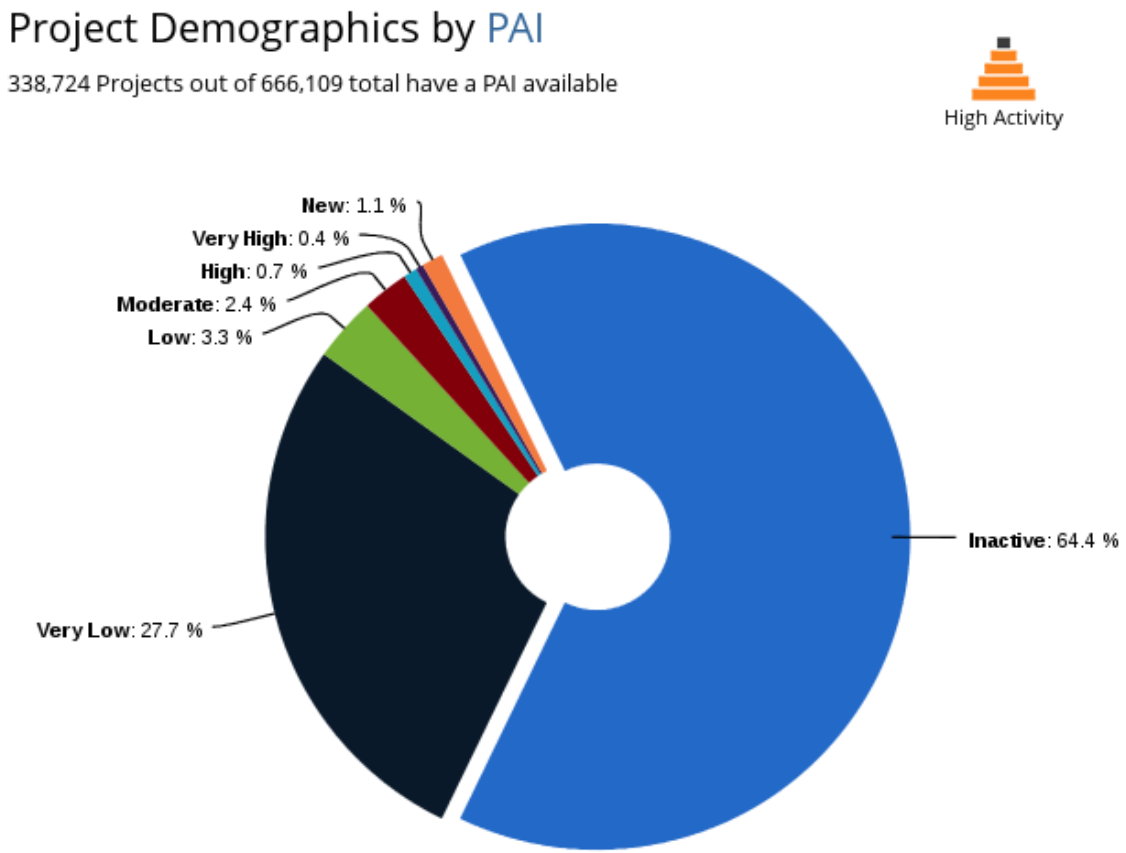
Project Activity

Project Activity Index (PAI) is **High Activity**

Activity



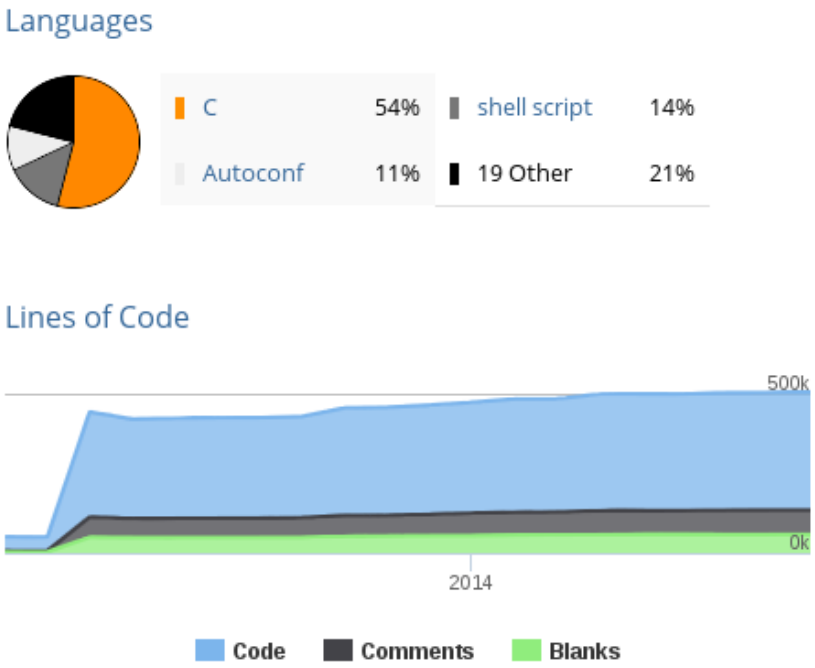
Activity relation



- More active than 98% of Open Hub Projects

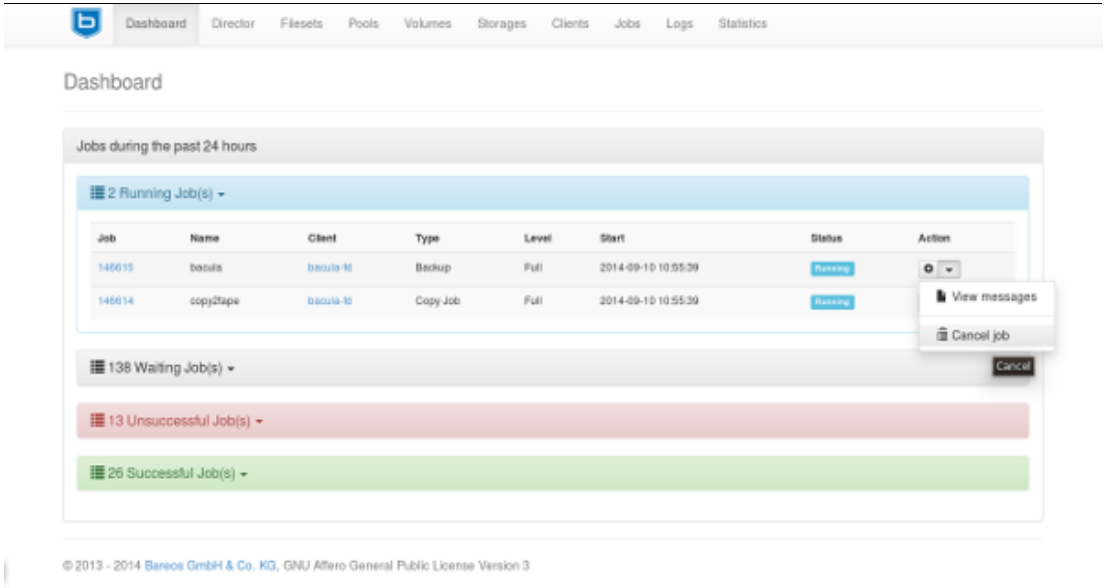
Bareos 14.2

31.879 Lines added since 13.2



Bareos-WebUI

- Webfrontend for Bareos



Supported platforms added

- Univention Corporate Server
- RHEL 7
- CentOS 7
- Ubuntu 14.04
- openSUSE 13.1
- customer request:
  - RHEL4 (FD)
  - Ubuntu 8.04 (FD)
  - SLE_10 (FD)
  - HP-UX (FD)

General enhancements

- Traymonitor now shows exclamation icon on error
- Added Python plugin api
  - Filedeamon
  - Storage Daemon
  - Director
- Class Abstraction for FD Python Plugin

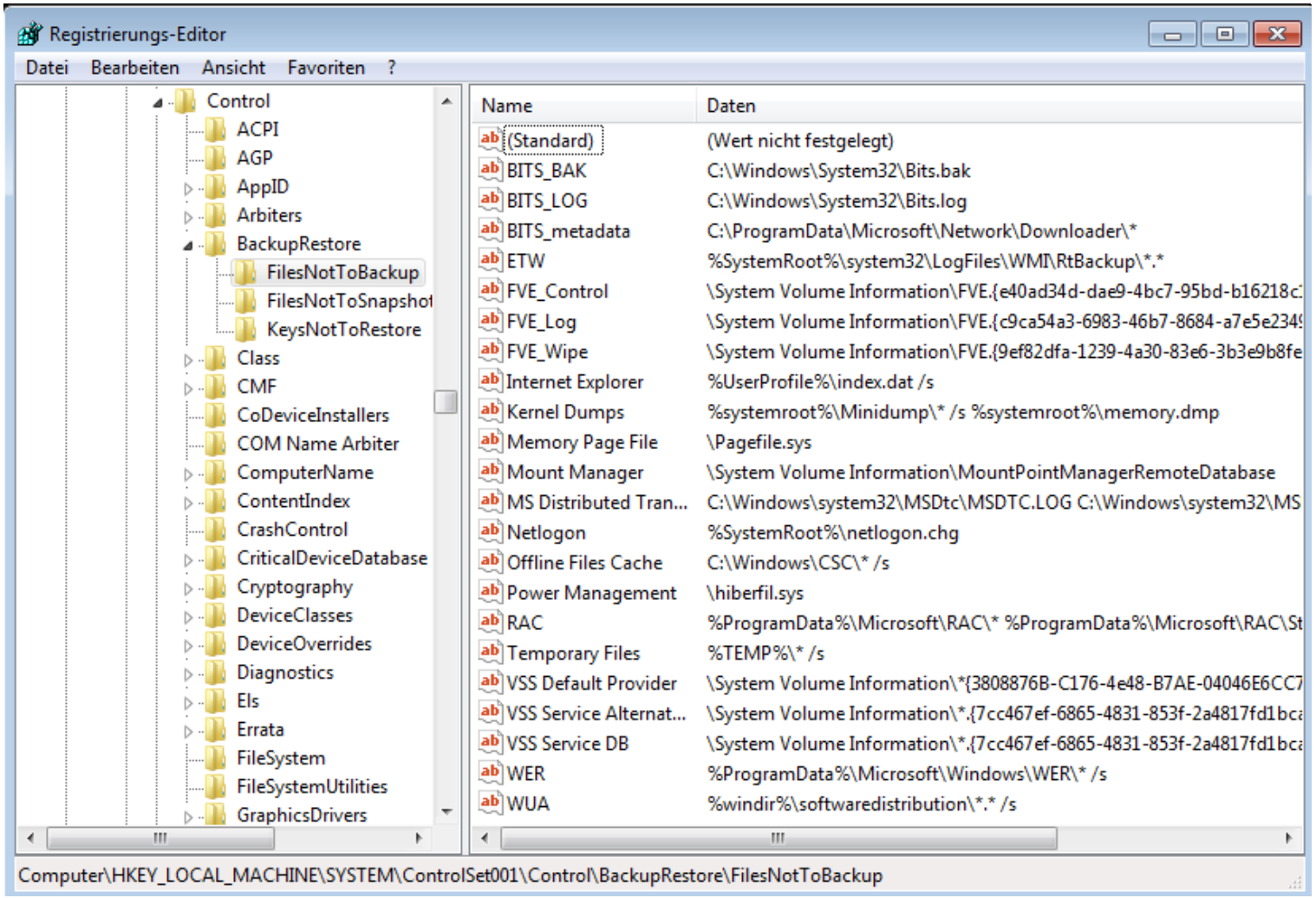
Python plugin api:


```
/src/plugins/filed/python-fd.c
```

- C-api plugin calls are translated to python calls and back
  - access to:
    - native python objects
      - dictionaries, lists
      - python modules
- Plugin functionality is implemented in python
- Python api can create job and debug messages
- examples in: https://github.com/bareos/bareos-contrib

Windows enhancements

- ACLs on windows config files
- changetime now detected by „GetFileInformationByHandleEx“ call
  - ACL changes were not detected
- Support for Windows Deduplicated Filesystem
- Windows Encrypting Filesystem (EFS) support
- FilesNotToBackup Registry Key support



*

Standard Key that contains Files/Dirs not to Backup with wildcards

FilesNotToBackup Registry Key

- Content of FilesNotToBackup Registry is automatically excluded from Backup
- Win7 testVM already more than 500MB
- joblog shows how many wildcards have been created:

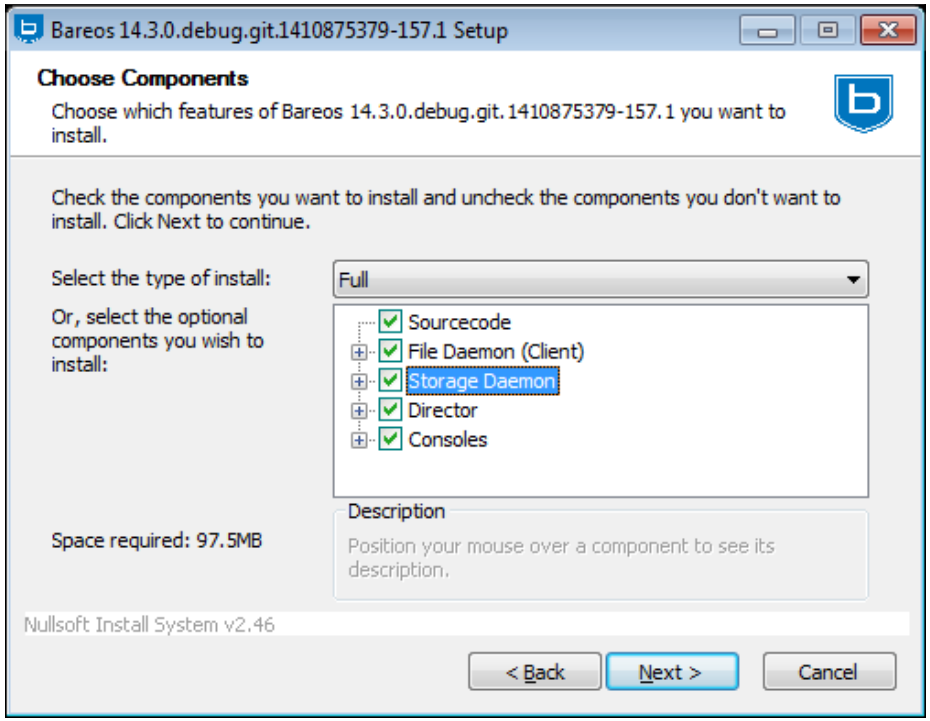
```
-win-fd JobId 232: Created 28 wildcard excludes from FilesNotToBackup Registry key
-win-fd JobId 232: Generate VSS snapshots. Driver="Win64 VSS", Drive(s)="C" VMP(s)=0
```

All daemons ported to windows

- Storage Daemon
  - Filestorage only
  - tools ported: bextract, bls
    - disaster recovery on windows incl. ACLs
- Director
  - postgresql server backend only
- Filedaemon was always available
- Installer supports setup of database and configures catalogbackup

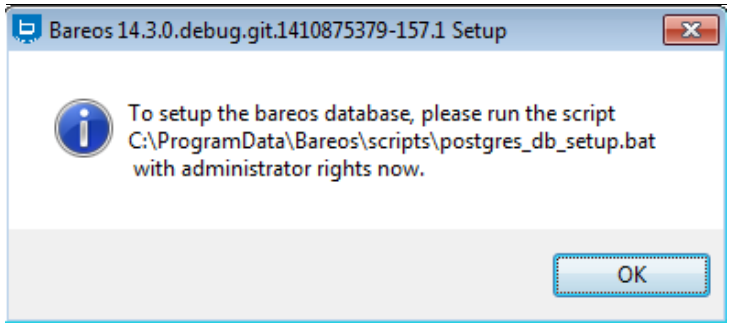
Windows Installation

- FD, SD and Director can be selected
- Firewall is opened as needed
- Debug Package also installs sourcecode



Windows Installation

- Needs installation of Postgresql first
  - script does setup of database



Windows Daemons

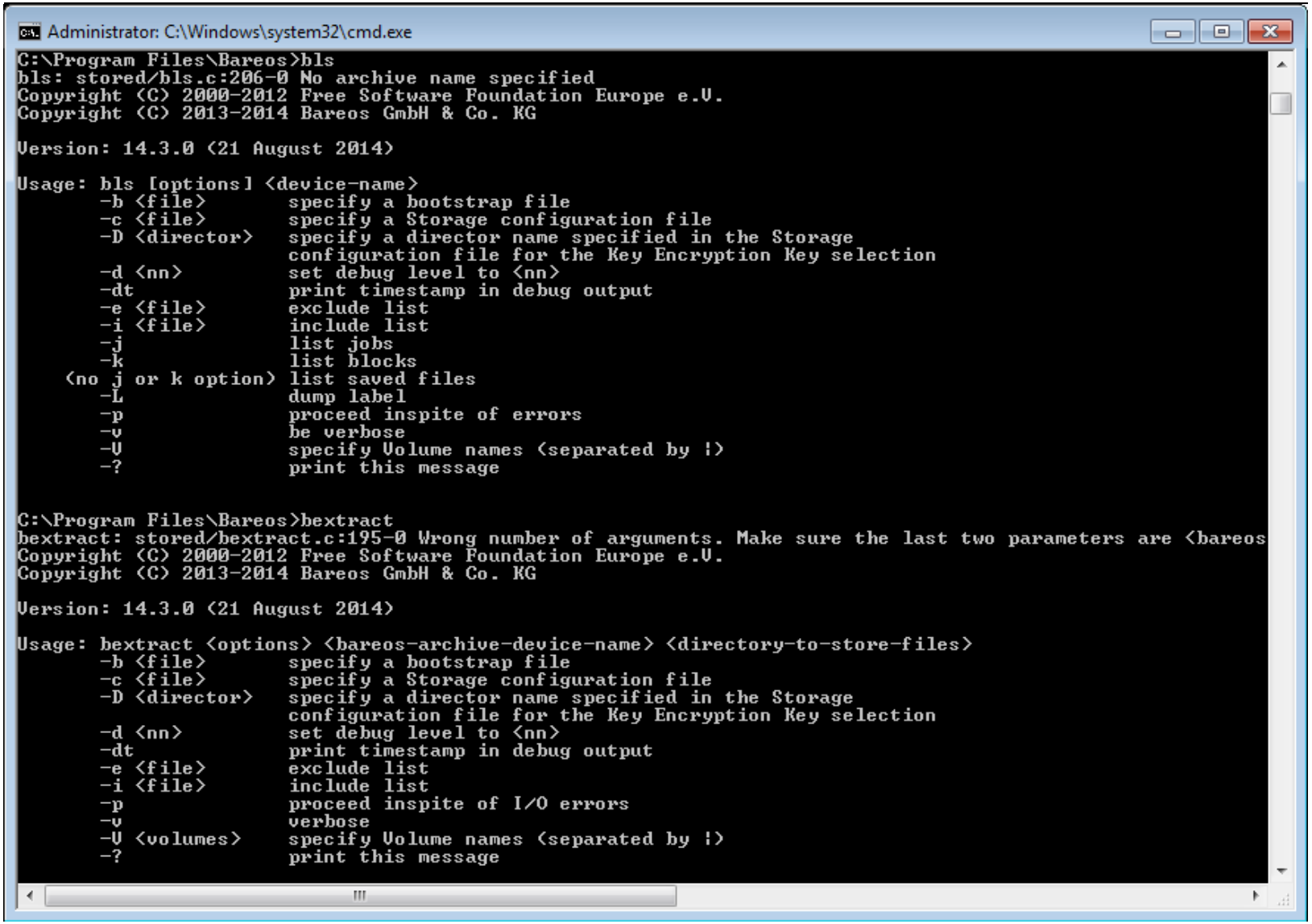
- Director, Storage Daemon and Local Filestorage on Windows

```
linuxtag-win-dir Version: 14.3.0 (21 August 2014) Linux Cross-compile Win64
Daemon started 17-Sep-14 12:17. Jobs: run=0, running=0 mode=0
 Heap: heap=0 smbytes=85,951 max_bytes=90,112 bufs=323 max_bufs=324
```

```
linuxtag-win-sd Version: 14.3.0 (21 August 2014) Linux Cross-compile Win64
Daemon started 17-Sep-14 12:14. Jobs: run=0, running=0.
Microsoft Windows 7 Ultimate Edition Service Pack 1 (build 7601), 64-bit
 Heap: heap=0 smbytes=40,167 max_bytes=93,133 bufs=87 max_bufs=88
 Sizes: boffset_t=8 size_t=8 int32_t=4 int64_t=8 mode=0 bwlimit=0kB/s
```

```
Device "FileStorage" (C:/bareos-storage) is not open.
```

SD Tools


Administrator: C:\Windows\system32\cmd.exe
C:\Program Files\Bareos>bls
bls: stored/bls.c:206-0 No archive name specified
Copyright (C) 2000-2012 Free Software Foundation Europe e.V.
Copyright (C) 2013-2014 Bareos GmbH & Co. KG
Version: 14.3.0 (21 August 2014)
Usage: bls [options] <device-name>
-b <file> specify a bootstrap file
-c <file> specify a Storage configuration file
-D <director> specify a director name specified in the Storage
configuration file for the Key Encryption Key selection
-d <nn> set debug level to <nn>
-dt print timestamp in debug output
-e <file> exclude list
-i <file> include list
-j list jobs
-k list blocks
(no j or k option) list saved files
-L dump label
-p proceed inspite of errors
-v be verbose
-V specify Volume names (separated by |)
-? print this message
C:\Program Files\Bareos>bextract
bextract: stored/bextract.c:195-0 Wrong number of arguments. Make sure the last two parameters are <bareos
Copyright (C) 2000-2012 Free Software Foundation Europe e.V.
Copyright (C) 2013-2014 Bareos GmbH & Co. KG
Version: 14.3.0 (21 August 2014)
Usage: bextract <options> <bareos-archive-device-name> <directory-to-store-files>
-b <file> specify a bootstrap file
-c <file> specify a Storage configuration file
-D <director> specify a director name specified in the Storage
configuration file for the Key Encryption Key selection
-d <nn> set debug level to <nn>
-dt print timestamp in debug output
-e <file> exclude list
-i <file> include list
-p proceed inspite of I/O errors
-v verbose
-V <volumes> specify Volume names (separated by |)
-? print this message

Storage Daemon Enhancements

- autoinflation/deflation plugin:
  - can compress and decompress streams on-the-fly
  - option to do compression on the sd instead fd
- Support for tapealerts via sd plugin
  - use the same device,
  - store alerts in database

SD Enhancements

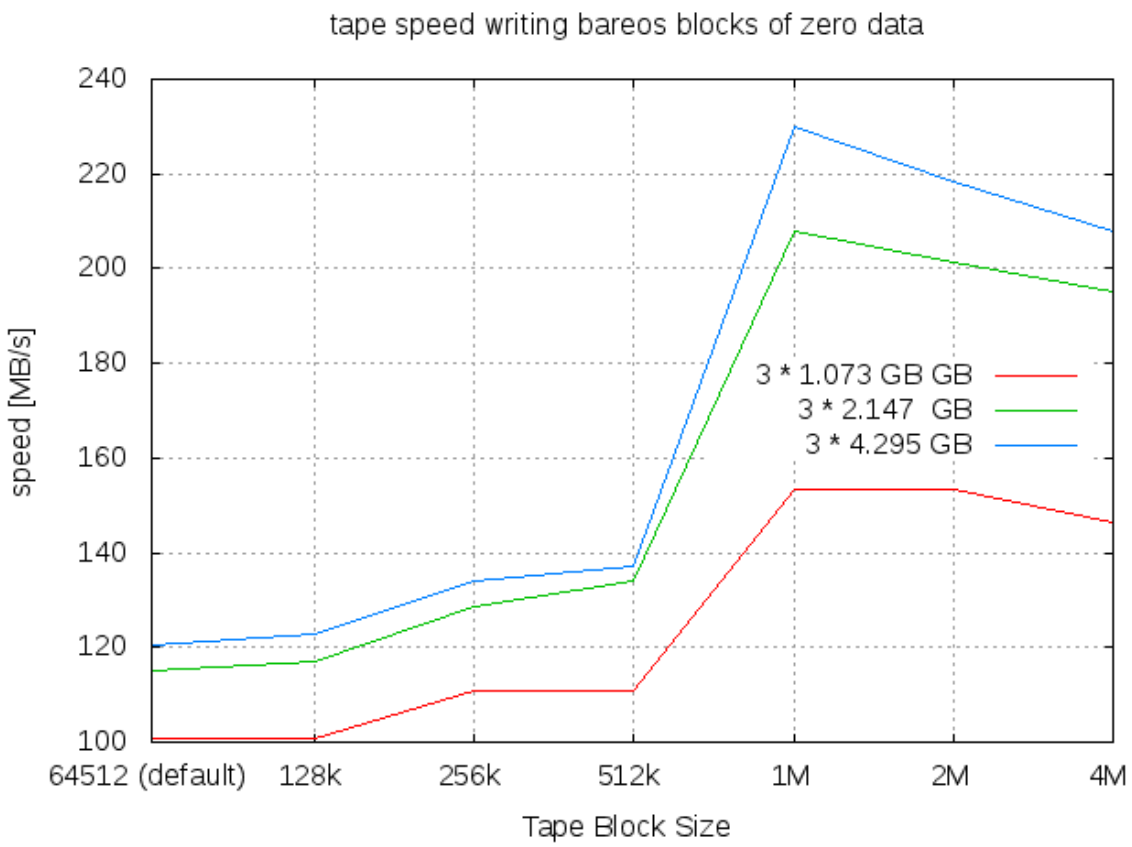
- Collect Storage Daemon statistics
  - Jobstatistics
  - Device Statistics
    - both are stored in Database
- dynamic loading of storage backends
  - shared code between the different storage programs in a new shared library

Cloud Storage Backends

- Gluster FS
- Ceph/Rados
- Object Storage (S3/Swift)
  - Talk: Backup to and of the Cloud (Marco van Wieringen)

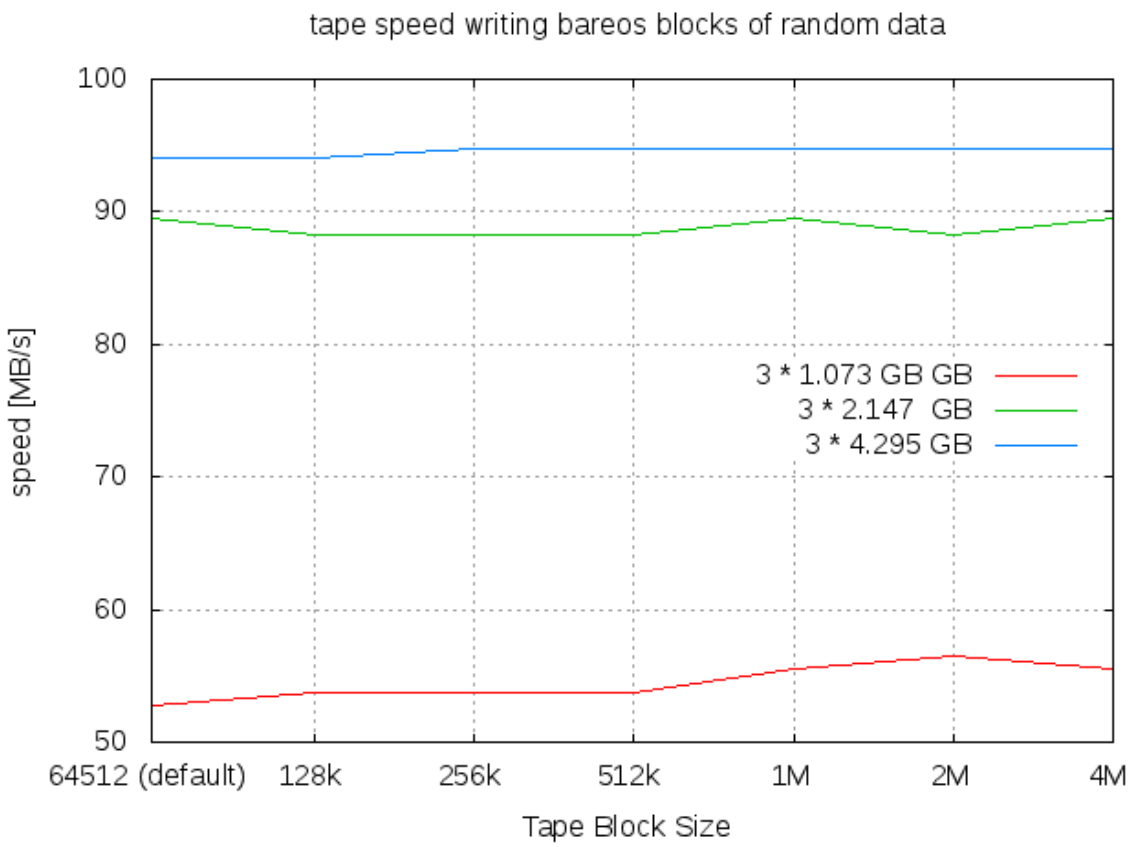
Blocksize and tape write speed

- Blocksizes impact on speed with compressible data:



Blocksize and tape write speed

- Blocksizes do not hurt on uncompressable data:



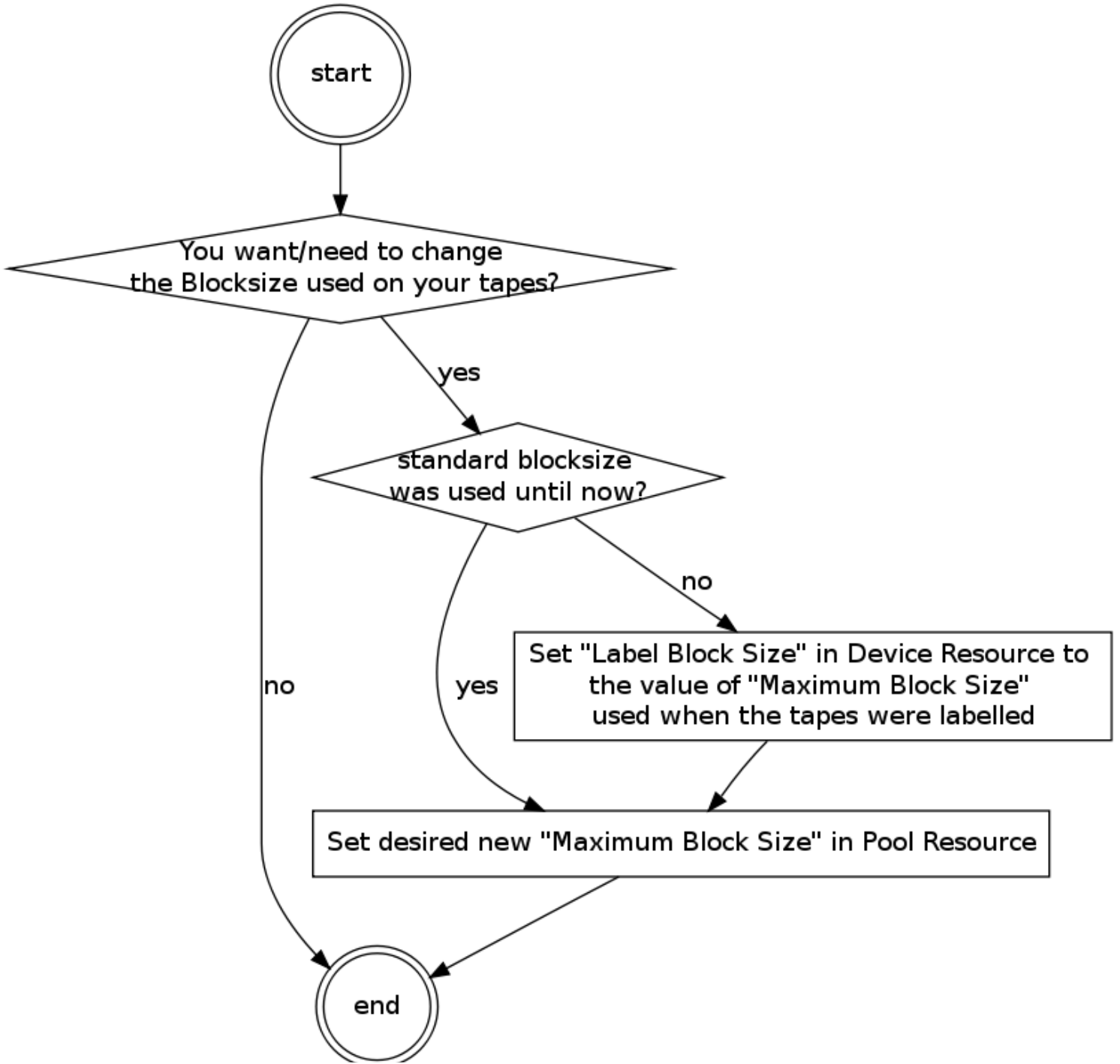
Blocksize setting in Pool

- **You want to use bigger blocksizes!**
- Problem: Changing Blocksize in Device makes old Backups *unreadable*
- Tape Label is always written in given blocksize

Blocksize setting in Pool

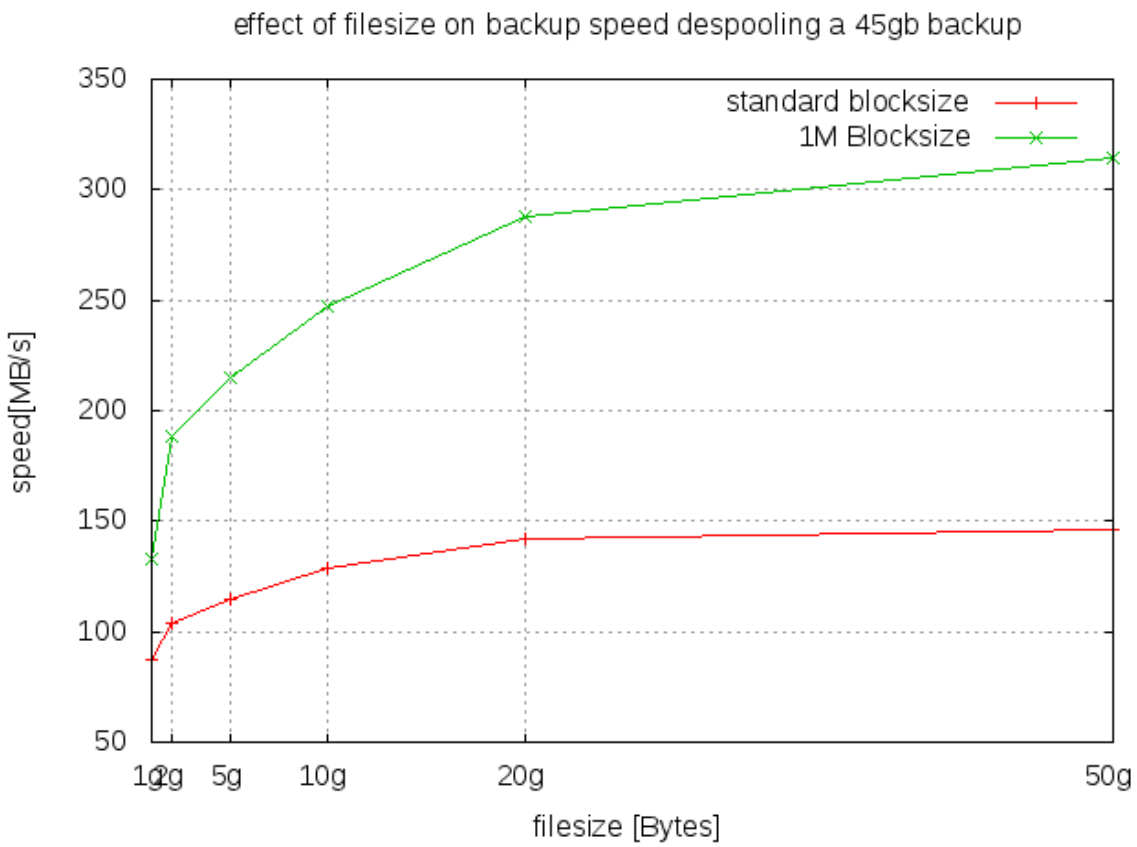
- Solution:
  - Blocksize is property of pool
  - Labels are always written with default blocksize
  - Data blocks are written with configured blocksize

```
Pool {
  Name = Pool-1M
  Pool Type = Backup
  Recycle = yes
  AutoPrune = yes
  Volume Retention = 3 months
  RecyclePool = Scratch
  Maximum Block Size = 1048576 # 1M
}
```


start
You want/need to change
the Blocksize used on your tapes?
yes
no
standard blocksize
was used until now?
yes
no
Set "Label Block Size" in Device Resource to
the value of "Maximum Block Size"
used when the tapes were labelled
Set desired new "Maximum Block Size" in Pool Resource
end

More info in tape tuning whitepaper

- File Size also important



Director Enhancements

- Client and Schedule can be disabled
- Autochanger-sensitive commands
  - e.g. status slots only can be run on autochanger
- Exclude Dir Containing allows multiple entries

Director Enhancements

- Copy/Migration Jobs:
  - Client and Fileset not needed anymore
    - Jobs show original values for
      - client
      - level
      - fileset
    - Disabled checking client concurrency

What comes next

- Windows Volume Shadow Service Support
  - VSS Writers are available for many applications
  - Standard API that supports
    - Full, Incremental and Differential Backups
    - Partial files
  - When implemented, any Windows Application that supports VSS is automatically supported.
  - Status: Full Backup and Restore run without errors.

What comes next

- VMware Vstorage API support
  - Allows backup of vmware virtual machines
    - supports Changed Block Tracking
    - only used/changed blocks are backed up/restored
  - Status: Backup and Restore work in lab environment

Thank you

Questions?