Lessons learned from backing up a large, not supported filesystem

Bjørn Nachtwey
Gesellschaft für wissenschaftliche
Datenverarbeitung Göttingen (GWDG)

outline

- a few words on GWDG, ISP & storage @ GWDG
- the challenge: backing up our fileservices
- the basic approach
- strange behavior or *problems observed*
- an extra challenge ☺
- conclusions & lessons learned

a few words on GWDG

GWDG is

- Computing Center for
  - Max-Planck-Society
  - University of Göttingen (incl. University Clinic)
- hoster of several services for other universities
  - Lower Saxony (e.g. HLRN-IV)
  - Germany (e.g. academic cloud)
- chair of practical informatics (University of Göttingen)

our TSM environment

- 30 ISP server instances
    - 4x dedicated LibraryManager + 1 instance managing it‘s own library
- ~ 2500 clients (mostly file backup)
- 4 tape libraries (IBM TS3584, TS4500, 2x Quantum i6000)
    - ~ 10,500 slots (LTO6, LTO7)
- 18 hardware servers running up to 3 ISP instances

- Still running ISP 7.1.7
- Replication
    - between two local servers, will replace COPY pools in future
    - WAN replication with an Max-Planck-Institute
- no container pools (by now), just DISK, FILE + TAPE
- TSMManager, Operations Center as a PoC

GWDG‘s file storage environment

- some NetApp filers (incl. Metro clusters): ~ 1,500 TB
- 1 Windows cluster for user profiles: ~ 250 TB
- 1 x ISS/GPFS as part of HLRN-IV: ~ 350 TB
- 2 x BeeGFS for local HPC: ~ 440 TB

- 64 global StorNext filesystems (SNFS): ~ 19,000 TB
  - sizes from 5 TB -- 1.5 PB
  - some filesystems include HSM policies
  - most are file shares for user $HOME and workgroup shares
  - Linux, Mac and Windows clients accessing via CIFS, NFS and proprietary SNFS protocol

the challenge

- some StorNext-Filesystems are large (> 100 TB -- 1.3 PB)
- contain to totally ~ 2 billion files
  - some filesystems have > 100 million files
- StorNext-FS does not provide changed-files lists
  - filetree walk scanning for changed files takes time
  - especially if done via CIFS or NFS

So the challenge is how to scan each filesystem within one day!

the approach

Not talking about speeding up the backup using

- built-in approaches such as
  - `-incrbydate`
- parallelizing the backup using multiple threads
  - ➢ see General-Storage Keynote @ ISP 2015
    http://tsm2015.uni-koeln.de/10191.html#c1967
    see General-Storage Keynote yesterday:
    https://isp2019.rrz.uni-koeln.de/31901.html#c101829
  - ➢ check open-source perl script at
    https://gitlab.gwdg.de/bnachtw/dsmci

the approach

StorNext filesystems delegate file meta data operations to a database based meta data server (MDS)!

- ➢ using this as an advantage :
  - ➢ set up an ISP client as a member of the StorNext cluster
    - direct access to MDS
  - ➢ Install ISP-Server on the same physical server
    - use SHMEM communication or pipe lineing instead of TCPIP
      - no limits due to window / frame size, retransmission, etc.

conclusions (so far)

- quick filetreewalk
  - Up to 3.5 mio files in 19‘ => **>10.5 mio. files scanned / h**
- good throughput / bandwidth up to 600 MB/s
  - while parallel streams are running, with concurrent IO
  - the SAN bandwidth seems to be the limit ;-)

- further remarks
  - one ISP server and 60 clients can be run on a 2-socket server ☺
    - be aware of the RAM consumption of server and clients!
  - SNFS also works fine for staging (> 2 TB/h @ 2 migrations)

strange behavior – problems observed

- ✓ approach works fine for linux

- but running on windows
  all data is backed up each spring and autumn
  (doing a full dump)
  - ➢ seems to be related to „daylight saving“ time change

solution by IBM & Quantum

- initially both companies insist their software is working fine, never heard about such problems from other customers
- some months later they remembered, there were some customers also complaining about this behavior
- at the end they found an explanation:

„everything works as designed!“

so what's the problem?

TSM client uses Microsoft APIs **`FindFirstFile/FindNextFile`** *to get fileattributes.*
The last write date-time (also known as modification time) returned as FILETIME structure.
The FILETIME structure is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).

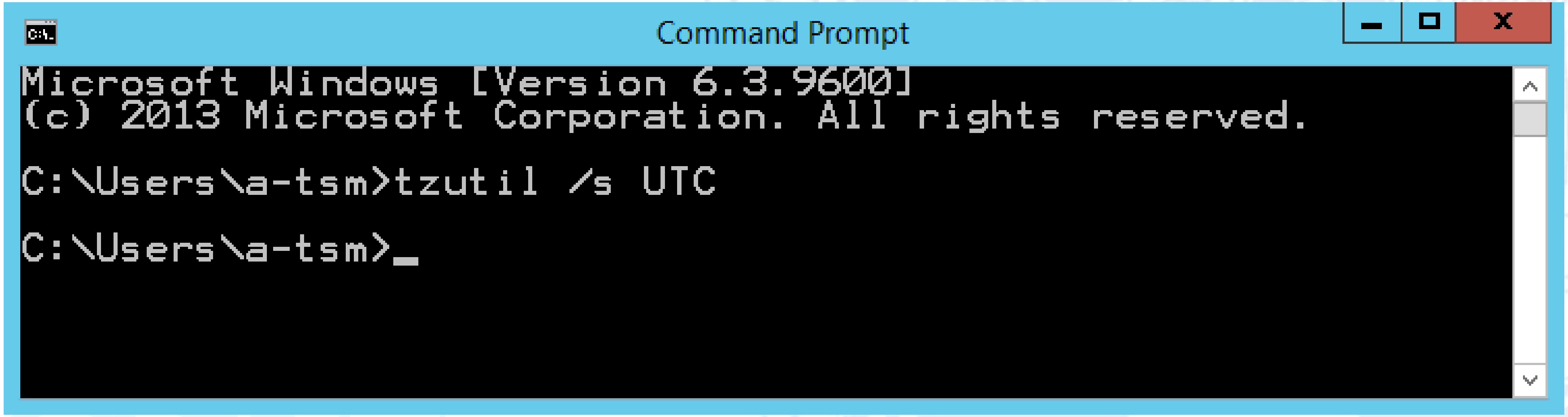
On non-NTFS file systems tsm client converts FILETIME time to MS-DOS time using the **`FileTimeToLocalFileTime`** *function and the* **`FileTimeToDosDateTime`** *functions*
`FileTimeToLocalFileTime` *uses the current settings for the time zone and daylight-saving time.*

Therefore, if it is daylight saving time, this function will take daylight saving time into account.

- **on non-NTFS file systems the tsm client uses a MS-DOS last modification date/time function considering daylight-saving**
- **as the SNFS timestamp does not change,**
  this conversation gives a different offset
  for summer / winter time!

guess how easily it could be solved!

- just run the ISP client without daylight-saving, e.g. set timezone to UTC!



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\a-tsm>tzutil /s UTC

C:\Users\a-tsm>_
```

- question @ IBM & Quantum:
  why did this answer take more than two years to be given?

an extra Challenge

- one of the SNFS/ISP servers also provides CIFS shares beeing member of a DFS domain
  - timezone cannot be set to UTC ☹

- solution
  - buy an additional server, but it in the SNFS cluster, but not in DFS
  - Set up a „embedded“ VM using UTC time on the same server host
    - grant access to „local“ StorNext filesystems for the VM
    - backup these filesshare inside the VM using the loopback device between host and VM
    - ✓ less than 5% loss of throughput

conclusion / lessons learned

- putting the client inside a cluster filesystem speeds up the backup
- an ISP Server and many clients can reside on one hardware box, but have an eye on the RAM consumption

- Non-supported filesystems may produce unexpected side effects
- if more than one vendor is envolved, make both support teams communicate directly with each other!

Questions?

do you miss the VIRTUALMOUNTPOINT approach for windows?

- unfortunately, time is already over
- You should have attended my BOF Session on tricks ☹

Bjørn Nachtwey
mailto: bjoern.nachtwey@gwdg.de

VIRTUALMountpoints for Windows

- a VIRTUALMOUNTPoint option is only available for Linux, Mac and Unices
- The only way to get dedicated folders backed up: Exclude all other folders (e.g. by EXCLUDE.DIR), but
  - a lot of effort (even if RegEx is possible)
  - error prone
  - does not allow to use RESOURCEUTILIZATION for multiple DOMAINs

VIRTUALMountpoints for Windows

Workaround

1. Create a (hidden) share for all folders you want to backup

```
net share sharename=folderpath /grant:username,permissions
```

- sharename: You can assign name to the share you are going to create
- username: Login id of the user whom you want to share the folder with
- permission: Read, Change or Full

VIRTUALMountpoints for Windows

2. Backup the folder(s) using the share names

example lines for „dsm.opt“:

```
Domain „\\127.0.0.1\share1$“
Domain „\\127.0.0.1\share2$“
Domain „\\127.0.0.1\share3$“
```

VIRTUALMountpoints for Windows

3. Further Benefits

- you can define differing access rights for the share
  - you can use backup privileges instead of access rights
  - access for local administrators will be sufficient, no domain nor enterprise admin rights needed
- moving the data to different volumes does not change the path for TSM/ISP (e.g. when rebalancing due to performance issues)
  - no new full dumps needed