THINXSOLUTIONS

Disk-to-Disk-to-Tape Backup for a Citrix XenServer Cluster

Bacula conference 2012

Robert Müller

25.09.2012

Agenda

(1) Introduction

(2) Use Case

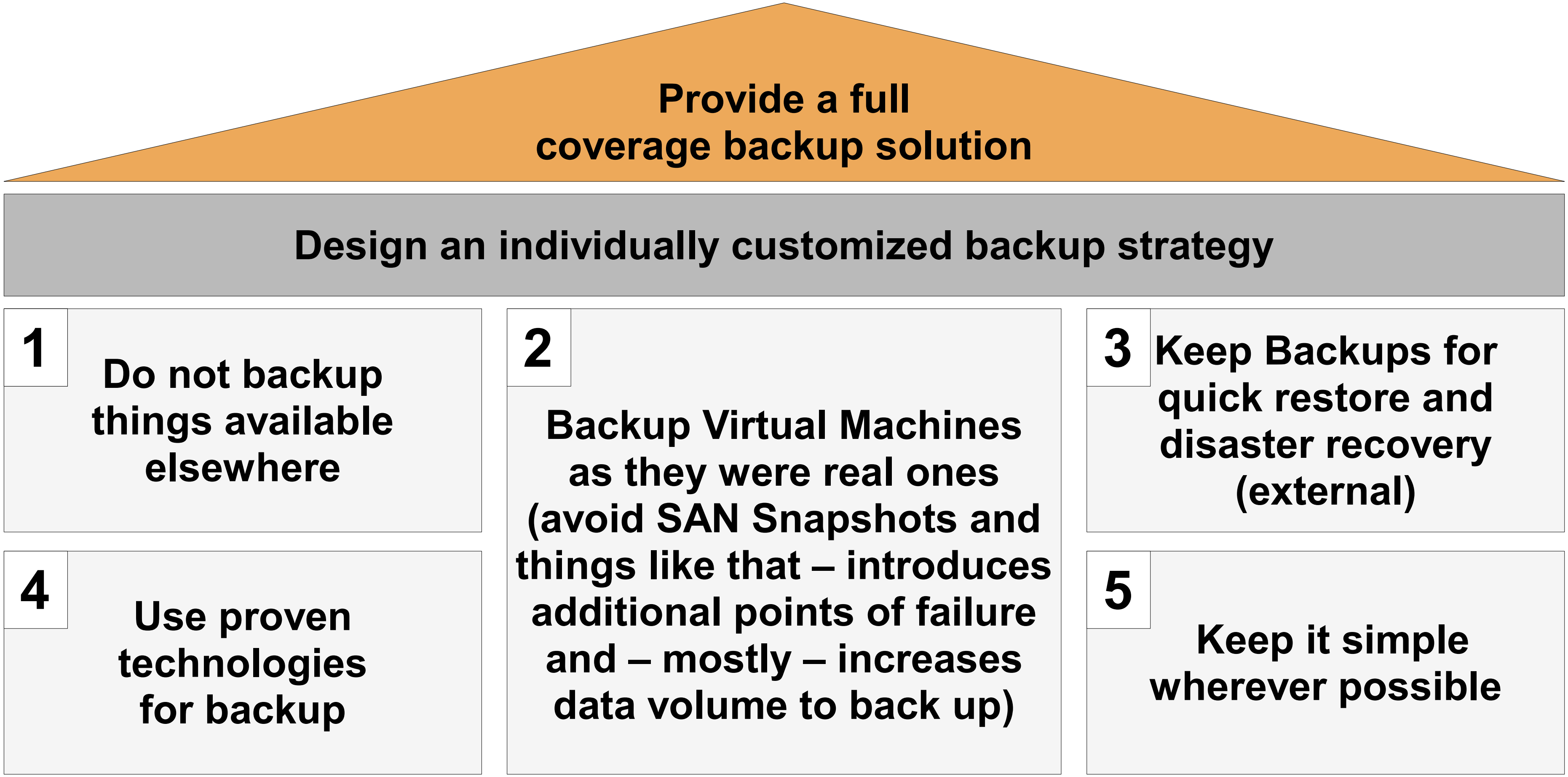
(3) Solution Design

(4) Realisation

(5) Findings

(6) Questions & Answers

Thinxsolutions key backup guidelines



Provide a full coverage backup solution

Design an individually customized backup strategy

1 Do not backup things available elsewhere

2 Backup Virtual Machines as they were real ones (avoid SAN Snapshots and things like that – introduces additional points of failure and – mostly – increases data volume to back up)

3 Keep Backups for quick restore and disaster recovery (external)

4 Use proven technologies for backup

5 Keep it simple wherever possible

What to back up?

Linux Systeme

- List of installed packages
- Configuration folders
- Log files
- Databases
- Application data folders
- /root
- /home
- /opt
- /usr/local
- Other user data folders (if any)

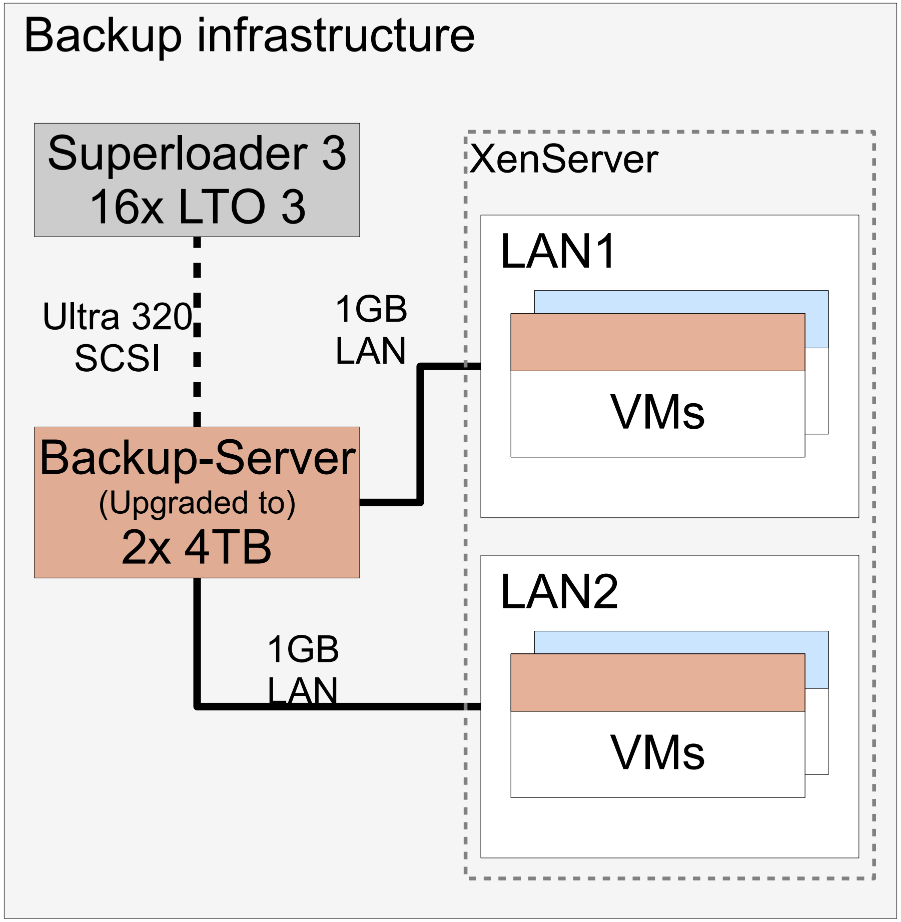
Windows Systeme

- System state
- User Data folders
- Log files
- Variable data (depending on role)

For all Windows systems, there should be an initial image stored on tape and archived externally

Use Case

Infrastruture overview



- One Xen server instance to host VMs of both LANs
- Several Windows systems (Domain controllers, File servers) version 2008
- Several Linux systems (App-, Web-, Collaboration Servers)
- Divided into two LANs because of organizational structure
- Backup has also to be divided by organizations
- Around 2,0TB data volume (in total)
- Servers should be available from 7:00 AM to 01:00 AM

Use Case

Existing backup solution

- Already using Bacula
- Daily incrementals with a full at every Monday
- Alternating even/uneven weekly full backup to always store one set of cartridges elsewhere
- Quarter year backups for archive
- All backups were executed against every VM at night.
- At weekend, 4 full backups have to be done to different targets (weekly and daily cartridges)

Issues throughout operation

- 4 Full backups each weekend (One for the daily backup, one for the external storage, both for each LAN)
- Every backup is processed by the file daemons on the systems, leading to significant additional load on the whole cluster
- Massive performance issues (rate below 25MB/sec), leading to extended backup times => backup runs during work time
- Backup capacity limited by autoloader capacity, which is nearly exceeded due to growing data volume

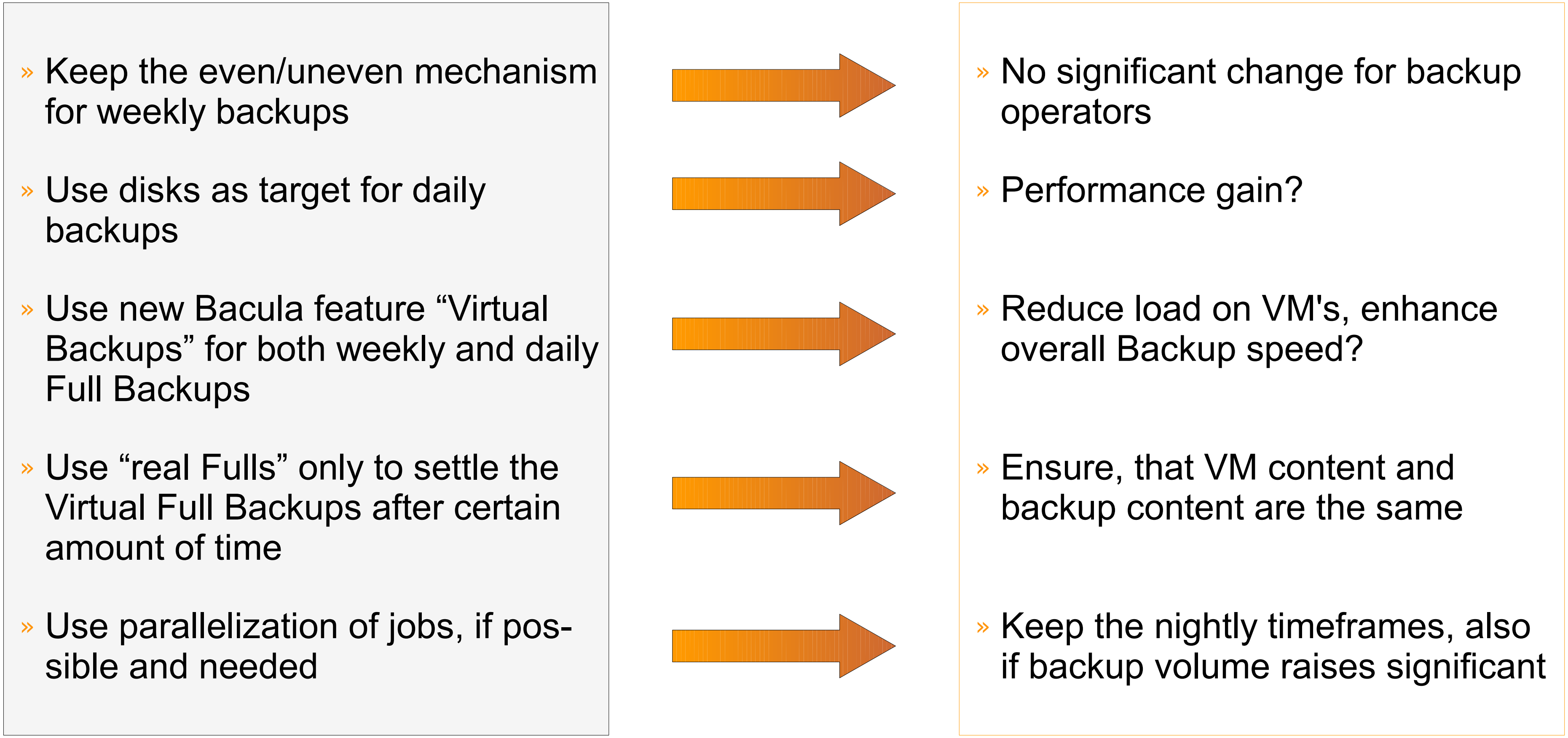
Goals to achieve

- Remain reliable and stable
- Allow external storage of a full backup set
- Keep the backup strategy simple for the ease of tape operations done by the customer
- Greatly enhance overall performance and reduce load to VM's
- Be prepared for raising data volumes
- Avoid significant additional costs (no faster or more HW-components)

So, let's see how this got solved

Solution Design

First ideas...



- Keep the even/uneven mechanism for weekly backups → No significant change for backup operators
- Use disks as target for daily backups → Performance gain?
- Use new Bacula feature “Virtual Backups” for both weekly and daily Full Backups → Reduce load on VM's, enhance overall Backup speed?
- Use “real Fulls” only to settle the Virtual Full Backups after certain amount of time → Ensure, that VM content and backup content are the same
- Use parallelization of jobs, if possible and needed → Keep the nightly timeframes, also if backup volume raises significant

Solution

- Extend the backup server with a large disk array (6TB)
- Backup anything to disk
- Use Virtual Backups when- and wherever possible
- Store weekly backups on tape for external storage, ensure, that **always** one tape set ist out of the office.
- Keep the even/uneven mechanism
- Keep the quarterly backup
- Use „Accurate Backup“
- No parallelization in the first step - as it's not expected to be needed and to keep things simple.

Solution Design

LAN1

		Media: Disk	Media:Tape
even week number	Monday	daily incremental / pool HD-even	virtual full to tape / pool uneven
	Tuesday	daily incremental / pool HD-even	
	Wednesday	daily incremental / pool HD-even	
	Thursday	daily incremental / pool HD-even	
	Friday	daily incremental / pool HD-even	
	Saturday	virtual full / pool HD-uneven	
	Sunday		
uneven week number	Monday	daily incremental / pool HD-uneven	virtual full to tape / pool even
	Tuesday	daily incremental / pool HD-uneven	
	Wednesday	daily incremental / pool HD-uneven	
	Thursday	daily incremental / pool HD-uneven	
	Friday	daily incremental / pool HD-uneven	
	Saturday	virtual full / pool HD-even	
	Sunday		

Hints

- On four uneven weekends close to quarter change, the tape pool has to be exchanged by a quarter pool
- In an uneven week, the tape virtual backups go to an even tape pool and vice versa, because the source backup is from the week before
- Sunday is left out in this picture, because we have a second organization to backup
- The virtual backup towards tape has for sure to be a copy job, but we did not know this at this time.....
- Week 53 (if exists) requires manual interaction

Summary

Backup-Strategy

Daily Backup (Disk)

Purpose: Quick restore of unintentionally deleted data

Pools: HD-even & HD-uneven

Retention: 12 days

Volumes: 20 à 100GB each (per pool)

Weekly Backup (Tape & Disk)

Purpose: External storage of volumes for disaster recovery

Pools: even & uneven (tape)
HD-even & HD-uneven

Retention: 12 days

Volumes: 3 à 400/800GB each (per pool)

Quarterly Backup (Tape)

Purpose: Provide a one year archive for restoration

Pools: quarter

Retention: 350 days

Volumes: 4x3 à 400/800GB each

Fully automatic

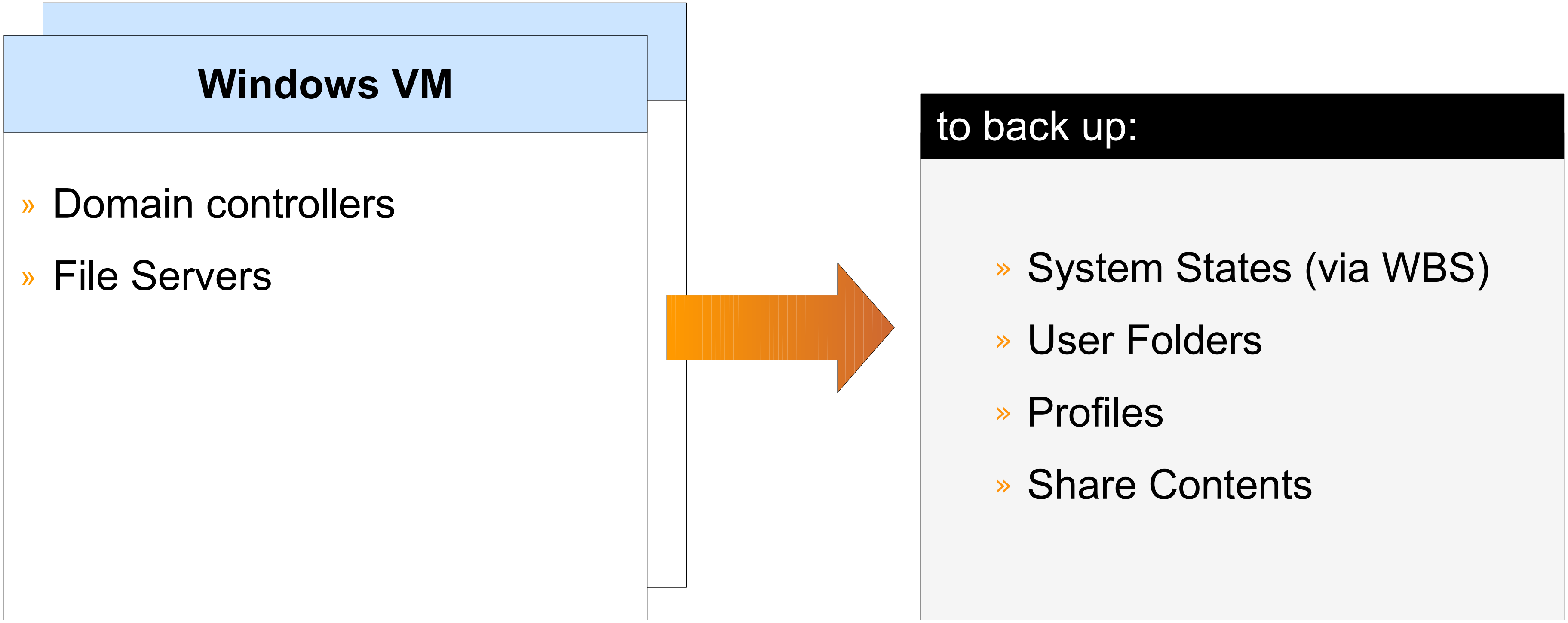
Tape change by operator needed on a regular basis

Known challenges

- Virtual backup on tape does not contain the actual catalog (with the virtual backup itself in it)
- Schedules tend to get complex
- Alternating target pool for virtual backups (damn nextPool statement)
- Operators must be informed which tapes to insert
- Barcode labels are very „human-unreadable“, making operator's life difficult
- System should check if proper tapes are inserted
- Ensure, that virtual fulls really reflect the „real full“

Realisation

Backup content



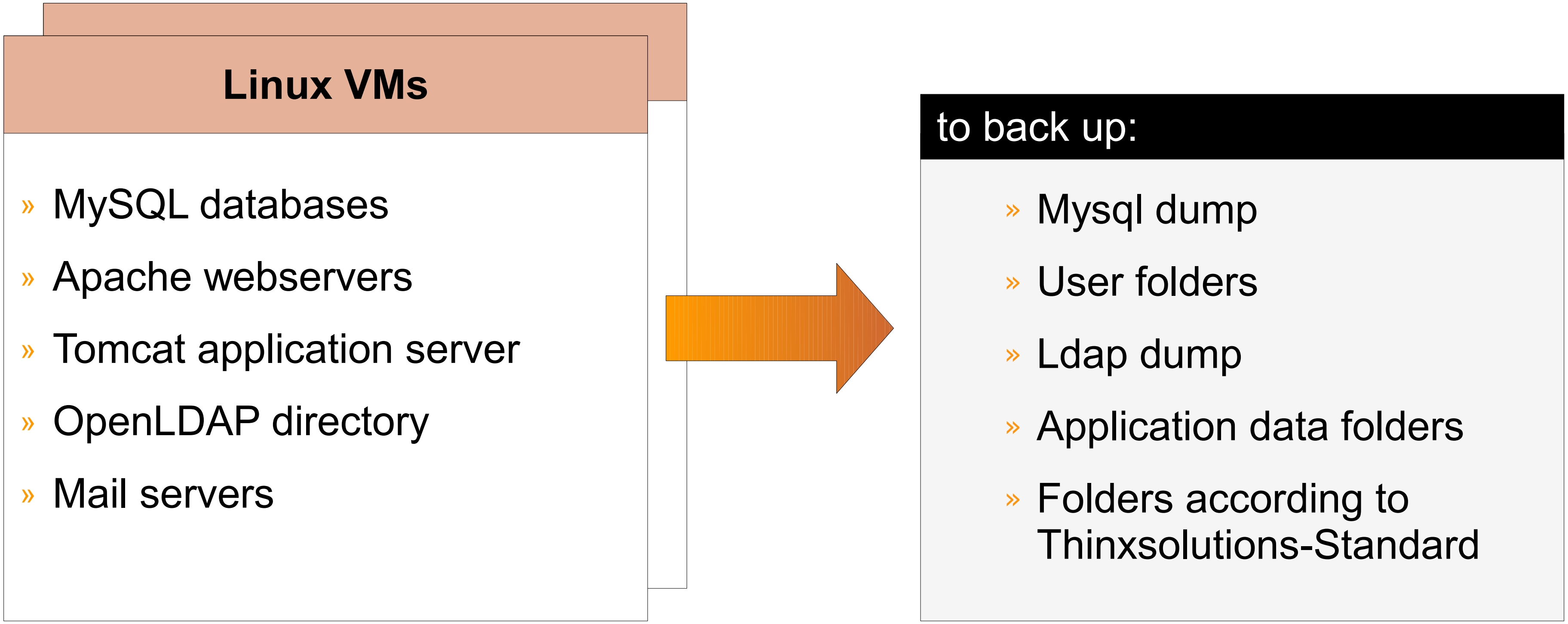
Windows VM

- Domain controllers
- File Servers

to back up:

- System States (via WBS)
- User Folders
- Profiles
- Share Contents

Backup content



Linux VMs

- MySQL databases
- Apache webservers
- Tomcat application server
- OpenLDAP directory
- Mail servers

to back up:

- Mysql dump
- User folders
- Ldap dump
- Application data folders
- Folders according to Thinxsolutions-Standard

Realisation

Schedule: Daily (incremental) to Disk

```
Schedule {
     Name = "BackuptoHDLAN1"

     # Daily Incremental to Disk
       Run = Incremental Pool=HDGeradeLAN1 w00 w02 w04 w06 w08 w10 w12 w14 w16 w18 w20 w22
w24 w26 w28 w30 w32 w34 w36 w38 w40 w42 w44 w46 w48 w50 w52 tue-sat at 1:30

     Run = Incremental Pool=HDUngeradeLAN1 w01 w03 w05 w07 w09 w11 w13 w15 w17 w19 w21 w23
w25 w27 w29 w31 w33 w35 w37 w39 w41 w43 w45 w47 w49 w51 w53 tue-sat at 1:30
```

- Schedule overrides: for backup level and target pool
- Be careful: the backup for Monday's data changes is processed Tuesday morning (service window!)

Schedule: Virtual Full to Disk

```
     # write every 4 weeks a "real" full-backup to disk

       Run = Full Pool=HDGeradeLAN1 w01 w05 w09 w13 w17 w21 w25 w29 w33 w37 w41 w45 w49 w53
sun at 0:30

     # Name = Weekly VirtualFull to Disk
     # write every other week a virtual full to disk
       Run = VirtualFull Pool=HDGeradeLAN1 w00 w02 w04 w06 w08 w10 w12 w14 w16 w18 w20 w22
w24 w26 w28 w30 w32 w34 w36 w38 w40 w42 w44 w46 w48 w50 w52 sun at 1:30

       Run = VirtualFull Pool=HDUngeradeLAN1 w03 w07 w11 w15 w19 w23 w27 w31 w35 w39 w43
w47 w51 sun at 1:30
}
```

- Real Fulls because we are paranoid ;-)
- For virtual full backups the pool statement describes the source volumes, the target pool is determined by the nextpool-directive within the pool configuration.

Job & pool parameters for disk backup

Job configuration

```
JobDefs {
     Name = LAN1
     Accurate = yes
     Max full interval = 9 days
     SpoolData = no
     Max Start Delay = 8h
     Max Run Time = 4h
     Schedule = BackuptoHDLAN1
     Pool = HDGeradeLAN1
}
```

Pool configuration

```
Pool {
     Name = HDGeradeLAN1
     NextPool = HDUngeradeLAN1
     [..]
}

Pool {
     Name = HDUngeradeLAN1
     NextPool = HDGeradeLAN1
     [..]
}
```

➡ **Pure disk backup ist now operational.**

Tape backup

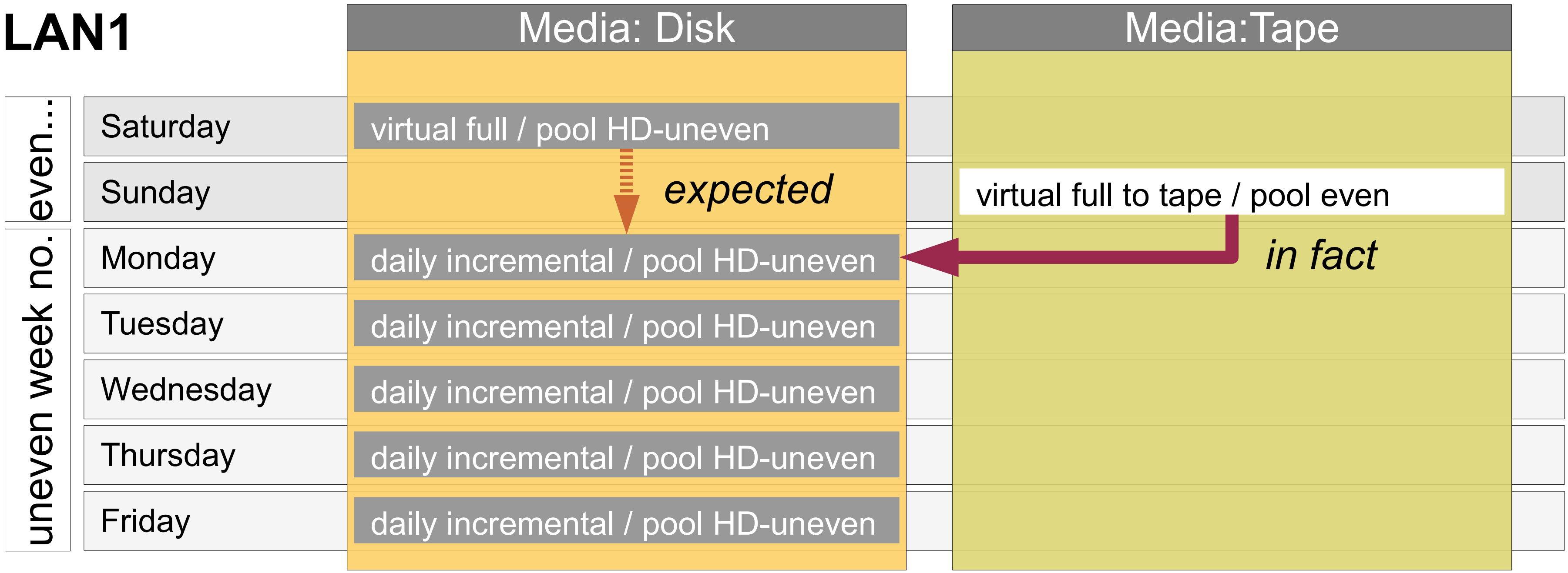
- Virtual Full on disk has been proven to be very fast
- Todo's for Virtual Full to tape:
  - Change Pool config periodically (NextPool-Statement)
  - Add schedule for additional virtual full backup
  - Backup catalog (containing the virtual backup) to tape

➡ **So far, no big deal...**

Unexpected problems

- Virtual Full on tape works as expected in the first week
- In the following week the virtual Full to disk **fails** because Bacula wants to read the tape (which is of course not in the loader)

LAN1



		Media: Disk	Media:Tape
even…	Saturday	virtual full / pool HD-uneven	
	Sunday		virtual full to tape / pool even
uneven week no.	Monday	daily incremental / pool HD-uneven	
	Tuesday	daily incremental / pool HD-uneven	
	Wednesday	daily incremental / pool HD-uneven	
	Thursday	daily incremental / pool HD-uneven	
	Friday	daily incremental / pool HD-uneven	

Realisation

Problem's cause

- Bacula takes the virtual full on tape as base for the daily incrementals, ignoring the pool definition.
- As result, the next virtual full backup demands the tape as data source, which is not available.
- To do the virtual full to tape first breaks the autonomy of disk backups.

Resolution

- Second virtual full is an almost exact copy of the virtual full one day before
- So use a “Bacula copy job“
- No predefined option to select “the last full backup for a client”
- A proper SQL statement is needed to achieve this.

➡ **Resolution with two virtual full backups in different pools not possible.**

Schedule: Copy Jobs

```
Schedule {
     Name = "CopyJobstoTapeLAN1"
     # schedule for copy of virtual full backups to weekly tapes

     Run = Pool=HDGeradeLAN1 w00 w02 w04 w06 w08 w10 w12 w14 w16 w18 w20 w22 w24 w26 w28
w30 w32 w34 w36 w38 w40 w42 w44 w46 w48 w50 w52 mon at 1:30
     # target pool: tape_ungerade

     Run = Pool=HDUngeradeLAN1 w01 w03 w05 w07 w09 w11 w13 w15 w17 w19 w21 w23 w25 w27 w29
w31 w33 w35 w37 w39 w41 w43 w45 w47 w49 w51 w53 mon at 1:30
     # target pool: tape_gerade
}
```

» Target pool is determined also via „NextPool“ statement

SQL: Copy Last Full Backup

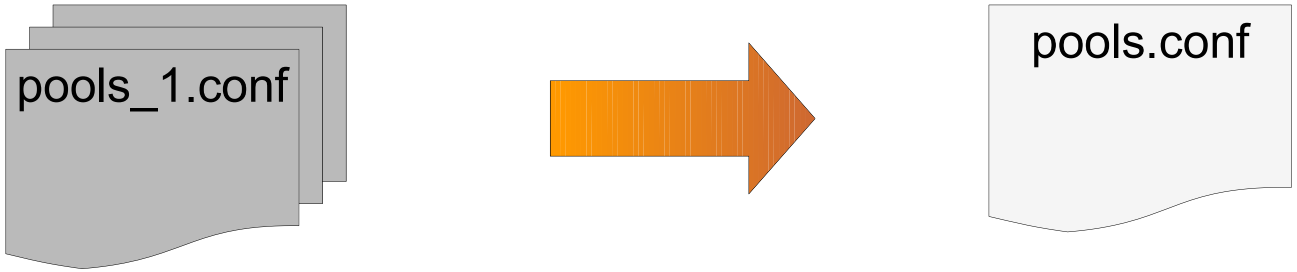
```
## Edit Client name in SQL-statement!!!! ##

Selection Pattern = "SELECT JobId FROM Job,Client WHERE Client.ClientId=Job.ClientId AND
Client.Name=\"Clientname\" AND Job.JobStatus=\"T\" AND Job.Type=\"B\" AND Job.Level=\"F\"
AND TIMESTAMPDIFF(DAY,Job.RealEndTime,NOW()) < 5 ORDER BY RealEndTime DESC LIMIT 1"
```

» Statement returns the last successful Full Backup within the last 4 days, giving operators the possibility to manually run the backup.

Pools.conf

- Keyrole for a working VirtualBackup and CopyJob setup.
- Change of „nextpool“ statement has to be automated

Template solution:



disk
tape
quarter

RunafterJob-Skript
(copy right template
dep. on last backup
level, volumes)

config needed for
next run

Technical Achievements

- Disk backup is running without operator intervention
- Tape backup always duplicates the last full backup (virtual or real) for external storage, which is wanted behaviour
- Periodic change of nextPool-statement is crap, but automated
- Configuration is not that complex as expected

Minor optimizations I

- Barcode labels tend to be difficult for correct tape recognition - in “real” and within bconsole

We use a mapping mechanism with an adapted mtx.changer (Thanks to W. Denk) to support „human readable“ volume names

- Operators should be informed by an E-Mail, which tapes to insert at the end of the week

We use a cron-based script which compares the needed tapes to the ones within the loader and sends the operators a proper mail (also template based)

Minor optimizations II

- System State Backups are done via ClientRunBeforeJob and were collected by normal FD operation

Now WBS Backups are written directly to a Samba share, reducing data transfer volume and time. Negative effect: These Backups have to be backed up additionally onto the tapes.

Summary

- ✔ Capacity problem of loader solved
- ✔ Backup is fast, as only incrementals are run against the FDs (except every 4 weeks)
- ✔ Due to even/uneven mechanism, we have at least 2 weeks retention time (much longer at the moment)
- ✔ Backup to tape is exact copy of disk backup and runs without client interaction (no service availability impact)
- ✔ Parallelization currently not needed, but possible
- ✔ No additional costs, except few large hard disk drives
- ✔ Reduced restore times as volumes are always available
- ✗ Ease of operation in case of failure

Findings

The miracles.....

- If something fails,
  - maintenance is difficult. Deep knowledge of implemented strategy necessary.
  - administrative interaction is definitely needed.
    No „self healing“
- Copy job was observed to be dramatically slower than virtual backup from the same source pool. Reason unclear.
- Backup from VM's is much slower than it (sh)could be and it is on physical machines. Reason after hours of investigation still unclear.
- Parallelization probably only effective when VMs run on different hardware

Questions & Answers

Thank you your for attention!

Any questions left?

rmueller@thinxsolutions.com