



Thomas Edlinger, AVL List  
Manager Web Based Applications

→ Mehr Wissen. Vorsprung sichern!

▶ Microsoft SharePoint Konferenz, 9.–10. Juni 2010 in Wien

# Planung von hochverfügbaren Installationen für SharePoint 2010

[thomas.grohser@bwin.org](mailto:thomas.grohser@bwin.org)

HATCHSET

SOLVION  
INFORMATION MANAGEMENT

d.develop


dccs

AvePoint®  
Unleashing the Power of SharePoint™

SIEMENS  
Siemens IT Solutions and Services

www.Sharx.com  
Make more out of SharePoint



Because it's everybody's  business



Thomas Edlinger, AVL List  
Manager Web Based Applications

→ Mehr Wissen. Vorsprung sichern!

▶ Microsoft SharePoint Konferenz, 9.–10. Juni 2010 in Wien

# Planning High Available Infrastructures for SharePoint 2010

**tg@grohser.com**

HATCHSET

SOLVION  
INFORMATION MANAGEMENT

d.develop


dccs

AvePoint®  
Unleashing the Power of SharePoint™

SIEMENS  
Siemens IT Solutions and Services

www.Sharx.com  
Make more out of SharePoint



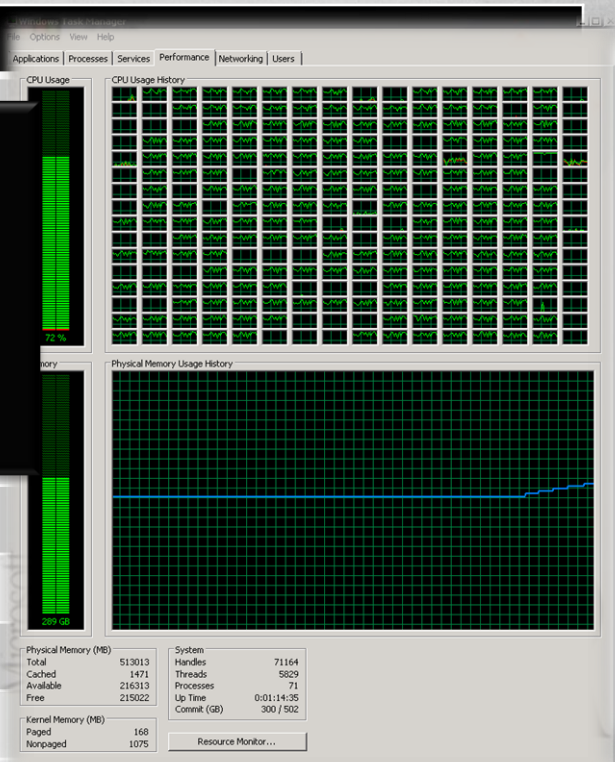
Because it's everybody's  business

select \* from =tg=

@@Version	Remark
SQL 4.21	First SQL Server
SQL 6.0	First SQL Server
SQL 6.5	First SQL Server
SQL 7.0	2
SQL 2000	9
SQL 2000 IA64	S
SQL 2005 IA64,x64	Long distance mirroring & > 400.000 STMT/s
SQL 2008 IA64,x64	Replication into mirrored databases
SQL 2008R2 IA64, x64	256 CPUs & >500.000 STMT/s
SQL 11 (Denali)	Can't wait to push the limits even further

**SharePoint Systems**

- Installed
- Operated
- Backed up and restored



**Focus on SQL Server Infrastructure Architecture and Implementation**  
**Close Relationship with Microsoft and Hardware Vendors**

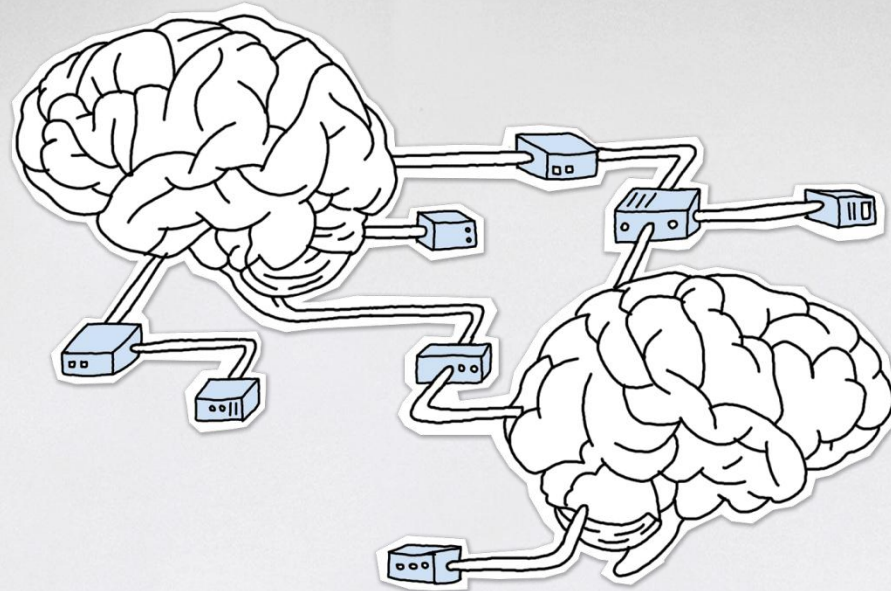
- SQLCAT (SQL Server Customer Advisory Team)
- SCAN (SQL Server Customer Advisory Network)
- TAP (Technology Adoption Program SQL2008R2 and SQL11)

**Active PASS member and PASS Summit Speaker**

Senior Database Engineer at **bwin** Interactive Entertainment AG for over 6 years designing and implementing one of the most challenging SQL Server and SharePoint infrastructures



- What is “High Availability”
- The Layers of SharePoint
  - Frontend
  - Backend
- Infrastructure Options
- Restore and Backup
- Q&A



## Getting everyone on the same Page: What is "High Availability"

# What is "High Availability"?

- The "available" means that "something" is able to be used as expected
  - E.g. the back-end database behind a SharePoint web server is able to service requests
- The "high" means that the "something" is available whenever it's needed
  - E.g. the server hosting the back-end database has been designed to be resilient to disaster and to meet maximum down-time and data-loss requirements
- Basically, users are always able to do what they need to be able to do
- But what is the "something" mentioned above?

# What is "High Availability"?

- The "something" will vary by situation
- Examples:
  - A data-center
    - Protecting against site loss
  - A server (Web or Database)
  - A database or group of databases
  - A file/file group in a database
    - Protecting against hardware or software failure
  - A table
  - Data in a table
    - Protecting against user/application error (e.g. accidental deletion or badly configured maintenance)

# How Can You Achieve High-Availability?

- First answer from many people is “get a failover cluster” – wrong!
- Takes a very layered approach
- Requires the *right* solution for the problem
- Needs to be constantly re-evaluated, re-examined, updated – managed/monitored
- Should just work – even through a fault
  - If your system isn't designed to protect against the fault that occurs – what's the point?

# What High-Availability Is NOT...

- NOT a single, pre-packaged solution from any vendor – hardware, software or application
- NOT technology without business knowledge and risk assessment
- NOT a business decision isolated from:
  - The cost of downtime
  - The cost of data loss
  - The cost of design and implementation
  - The cost of management and monitoring

# Aim of Requirements Gathering

- Goal: an ordered list of what needs to be protected and to what level
- Example:
  1. Database X must be available as close to 24x7 as possible, and no data loss can be tolerated
  2. The group of databases A, B, C must be available from 9am to 5pm weekdays, no data loss can be tolerated, and they must fail over together
  3. At least 3 of the 5 Webservers running the SharePoint web farm must be running
- Used for:
  - Input into the design
  - Validating the design before its implemented
  - Testing the design before it goes into production
  - Input into the operations guide and testing plan

# #1 Requirement: SLAs

- Do you know what SLAs are?
  - Service Level Agreements
- Do you know why they're important?
  - They're a contract to be met
- Do you know what your SLAs are?
  - \*YOU\* are responsible for meeting them
- Do you think you can meet them?
  - Have you taken them into account when designing?
- Once the system is operational, do you know you can meet them?
  - Have you tested the SLAs on the live system?
- The answer to all of these should be YES!

# When is a System Unavailable

- Application not Available
  - Server failure
  - Network failure
  - Software failure
    - Bug, Crash
    - Configuration failure
  - Infrastructure failure
    - Power failure
    - Environmental failure
- Data is not Available
  - Missing
  - Incomplete
  - wrong

# Most Important SLAs

- Maximum allowable data-loss per database or application
  - Known in the industry as RPO – Recovery Point Objective
- Maximum allowable down-time per database or application
  - Known in the industry as RTO – Recovery Time Objective
- Feed into choices of:
  - Database recovery model
  - Backup strategy (types, interval, storage media)
  - Redundancy strategy (technology, configuration)

# Granularity of Protection

- Granularity of protection helps dictate:
  - Technology to use
  - Selection backup strategy
  - Edition of Software to purchase
- What are you protecting against?
  - Site failure
  - Server failure
  - Database loss
  - Data loss
    - Isolated hardware failure (e.g. page checksum failure)
    - Human error (e.g. dropped table)

# Assess the Risk of Unavailability

- Assess your potential risks for downtime
  - Complete SharePoint System is unavailable
  - Site Collection is unavailable
  - Site is unavailable
  - Some Data in a Site is unavailable
- What do you think is going to be the greatest impact on your availability?
- The strategy should cope with the *highest impact* disasters first

## Top Questions I like to Ask

- What are you trying to protect?
- What is the SLA for maximum acceptable downtime?
- What is the SLA for maximum acceptable data-loss?
- What's the application eco-system?
  - i.e. what dependencies are there?
- Should failover be automatic?

## Consider Limitations

- There's no point going through the entire design process only to find that its flawed because of a pre-existing limitation
  - E.g. the design calls for a redundant data centre but the budget is only \$10k
- Limitations need to be known up front and are just as important as requirements

# Non-SharePoint Limitations (1)

- Budget
  - \$10k won't buy another data centre...
- Space
  - Only so many servers you can fit in a closet
- Power
  - Only so many servers you can run off a 20A circuit
- Air Conditioning
  - Too many servers in the closet and they'll overheat
- Manpower
  - Who's going to implement, monitor, and maintain the system?

## Non-SharePoint Limitations (2)

- Time to implement
  - A good design and solid implementation won't happen overnight
- Single solution provider for hardware or software
  - E.g. does your hardware vendor provide a SAN replication solution?
  - E.g. does your software vendor provide backup compression?
- Politics
  - E.g. who needs to authorize a failover?
  - E.g. does the same team manage Windows, SharePoint and SQL?
- Most of the limitations boil down to **BUDGET**

# Compromise

- Limitations may stop requirements being met
- If they do, several solutions:
  - Push back on limitations and/or requirements
  - Prioritize requirements and meet in order
- Crucial that management are aware of which requirements cannot be met and why
  - Not a good idea to only complain for the first time when a disaster strikes

# When to Pick Technologies?

- Note I said “technologies”, not “a technology”
  - Most solutions involve multiple technologies
- Once the final requirements are known, THEN and only then is the time to start evaluating technologies
  - Choosing technologies and then having requirements change just wastes time
- Make sure to understand the technologies before picking them
  - Success will not come from picking a technology and then trying to make it do something it’s not designed to
    - E.g. picking log shipping for a system that requires zero data loss
    - E.g. picking database mirroring for a system that requires multiple databases to failover

# Evaluating Technologies

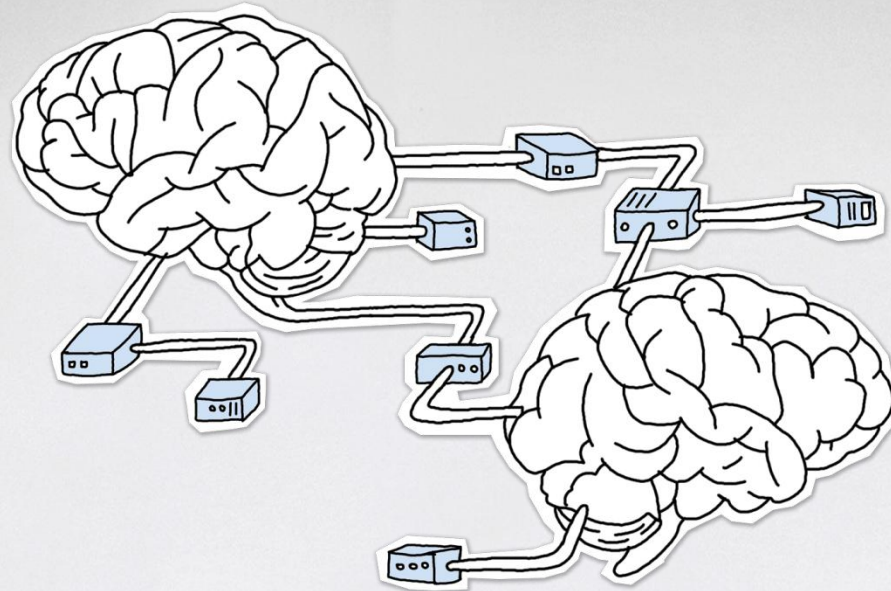
- For each technology, consider:
  - Cost
  - Complexity
    - Implementation, configuration, manageability
  - Impact on performance
  - Data loss exposure
  - Downtime potential
  - Feature compatibility
  - Does it allow you to meet some or all of your requirements?

# Planning the Backup Strategy

- Designing a backup strategy is integral to designing a highly-available system
- Even with the most sophisticated redundancy, recovery from total loss of all data centers can only be done using backups
  - And only then if they're stored somewhere else!
- The whole requirements gathering process feeds into the backup strategy too
  - And the fact that storage space and management are required for backups then feeds back into the requirements

# What Backups Do You Need?

- Backups exist solely to allow restores
- The question is really - *what kind of restores do you need to do?*
- Depends on:
  - What you're protecting
  - Data loss SLA
  - Downtime SLA
  - Whether you end up including log shipping in the HA design



## The Layers of SharePoint

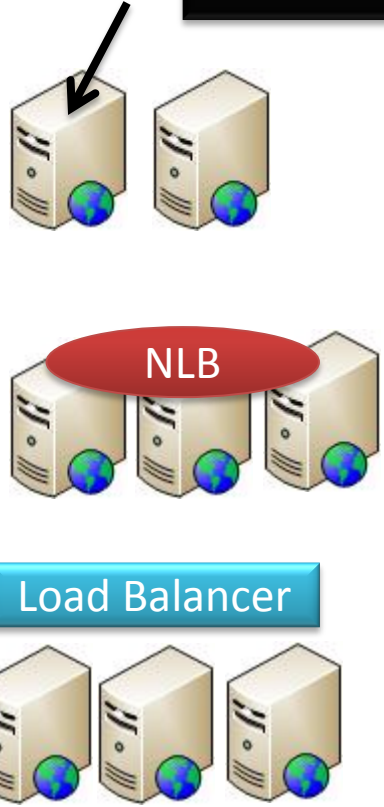
- \* ) Frontend
- \* ) Backend

# Frontend

- Webserver(s)
- Easy to scale and make high available
  - Set of equally configured servers
  - Load balancing between them
    - Hardware load balancer (recommended)
    - Windows Network load balancing
    - DNS round robin
    - Manual switch from one server to the other (only feasible when one server can handle the load)
    - Global load balancing across datacenters

# Infrastructure Options

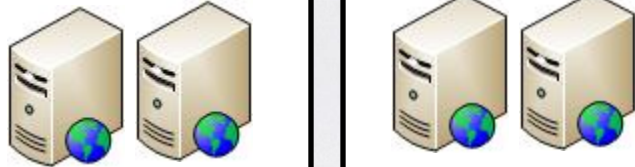
**Datacenter A**



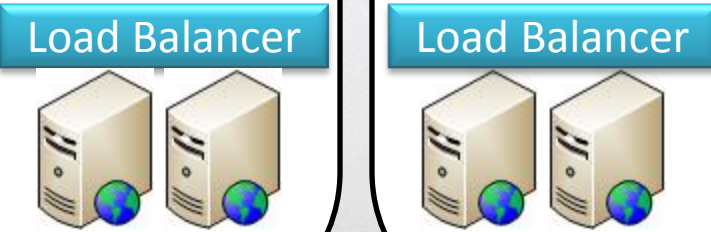
**Datacenter B**



**Load Balancer**



**Load Balancer**



**Load Balancer**

# Backend

- Application Servers
  - All backed roles on one server
  - Each role on a separate server
  - Each (some) roles on multiple servers
    - Same Load balancing options as with frontend

# Backend

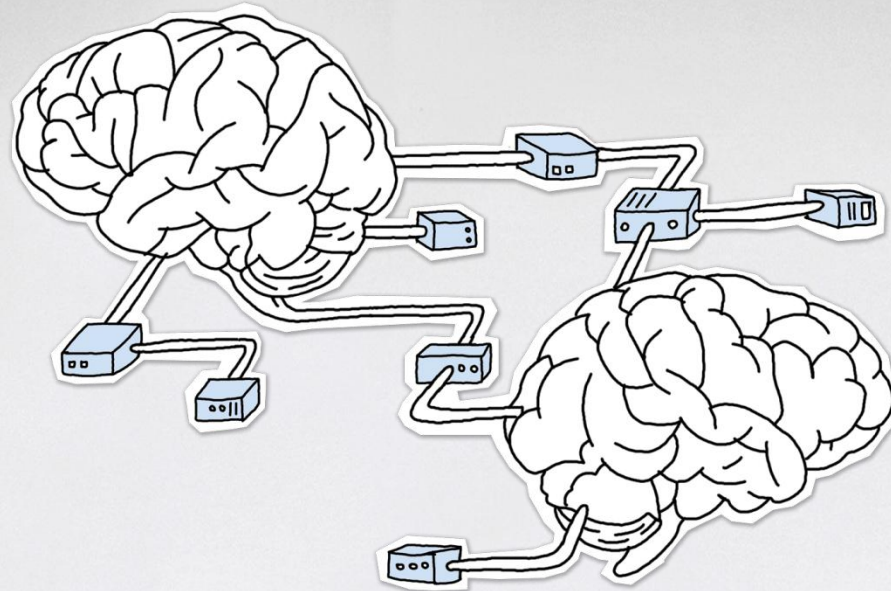
- Database Servers
  - Single Server
  - Failover clustering
    - Protection against server hardware and OS failure
    - Expensive
  - Geo stretched Failover clustering
    - Protection against server hardware and storage failure
    - Very Expensive
  - Log Shipping
    - Several Minutes delay, no automatic failover
  - Mirroring
    - Protection against server hardware and storage failure
    - Harder to maintain

# Database scaling

- Scale up
  - Careful backups getting harder and harder
  - One 4 Way server way more expensive than two 2 Way servers
- Scale out
  - Easy when you can split SharePoint into multiple site collections
  - You can place each site collection in a different database on a different servers

# Putting all on one Server

- Option for very small environments
- Option for Development and Test
- Not scalable



## Infrastructure Options

# Infrastructure Options

- How much data
- How fast to access
- Is an outage an option?
  - How long?
- Can I lose data?
  - How much data can I lose?
- Is degraded performance acceptable?
  - How much can we go slower?

# Infrastructure Options

Datacenter A

Datacenter B

Load Balancing

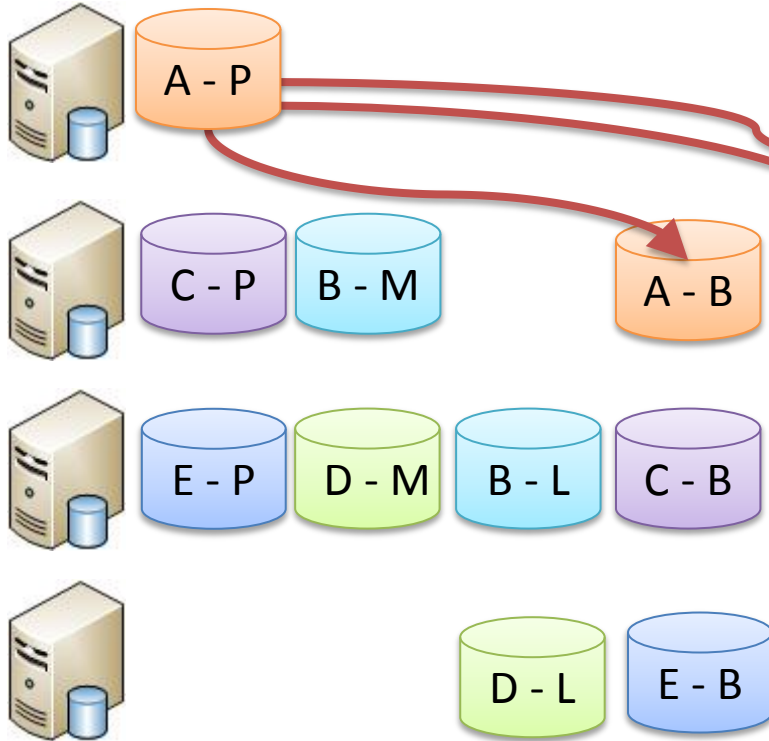


Load Balancing

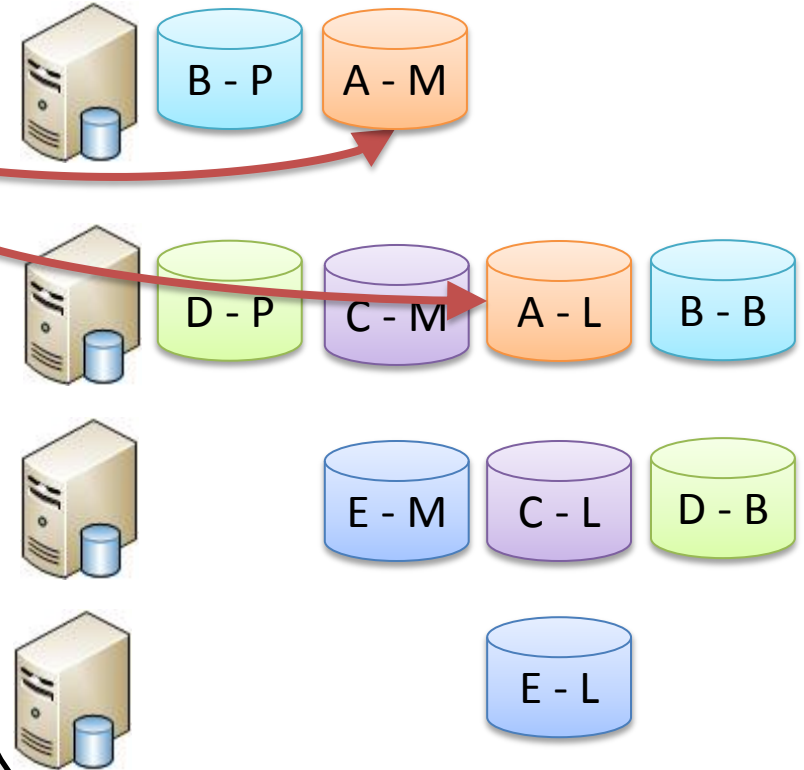


# Active - Active Mirroring

Datacenter A

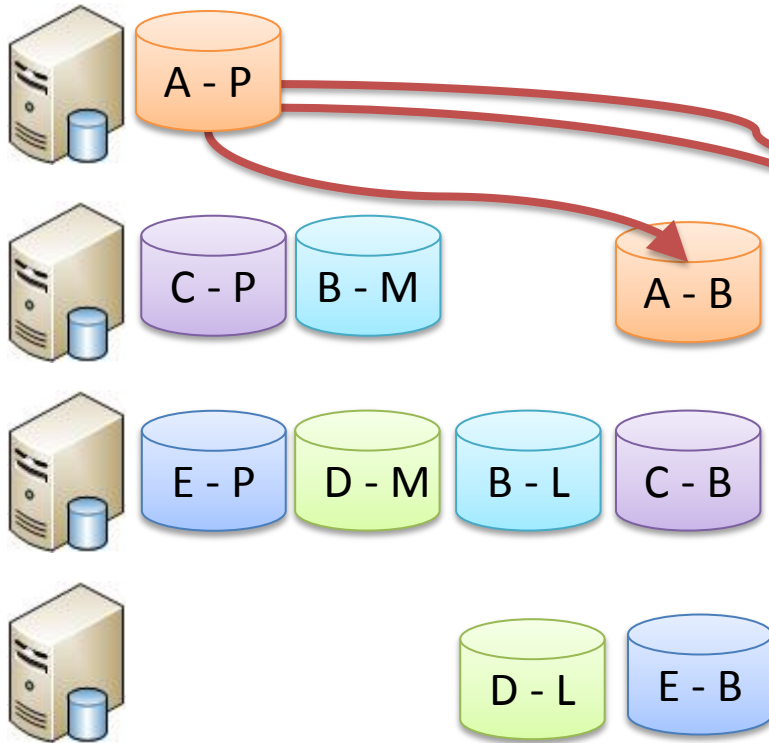


Datacenter B

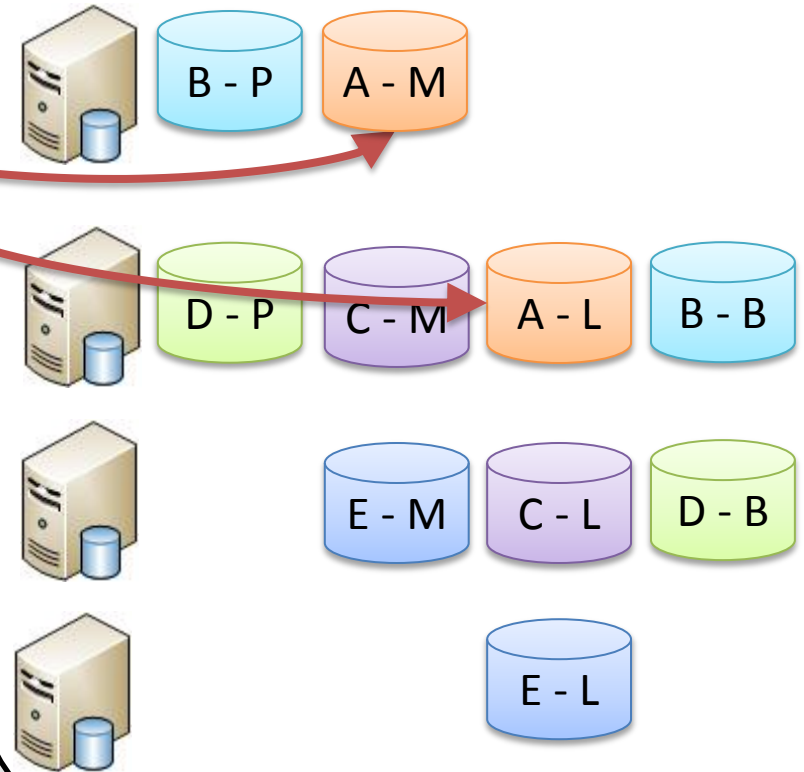


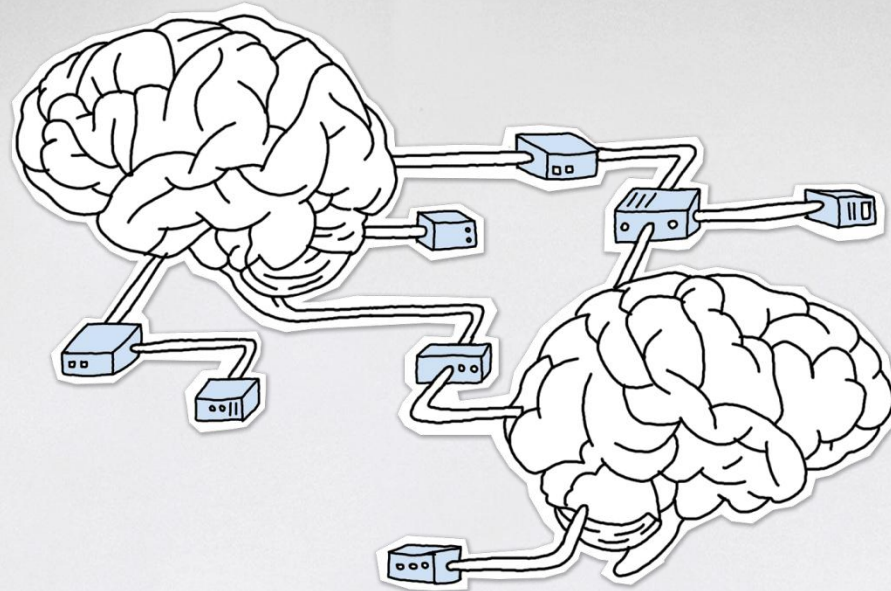
# Active - Active Mirroring

**Datacenter A**



**Datacenter B**





## Restore and Backup

# Backup

- You must backup
  - Databases
  - Webservers
- in a somehow synchronized way
  - Simple: On one Server with a snapshot
  - Easy: In time of inactivity
  - Hard: Webserver Snapshots and “log mark” in database on a scale out solution

# How much can I backup/restore

- Limits

- Network (calculated with  $\frac{1}{2}$  of best case = reality)

- 1GE 100MB/sec = 6GB/min = 360GB/hour
- 10GE 800MB/sec = 48GB/min = 2.8TB/hour
- Compression helps

- Disk

- SATA per 7kRPM Disk 200GB/h
- SAS per 15kRPM Disk 300GB/h
- SAN per 4Gbit HBA: 480MB/sec = 1,7 TB/hour
  - Check with storage people if volume can keep up
- Careful RAID level / controller might have influence
- Compression helps only on destination server not the actual database server

# Single Server Backup

- All components installed on one Server
  - Implies a relatively small Share Point
- Use a backup tool that uses VSS to create a snapshot and then backups the snapshot
- System Center Data Protection Manager
- Almost all commercial backup tools

# Multi Server Backup

- Easy case: there is a backup window
  - Backup all servers in the time of inactivity at almost the same time (the closer the better)
  - If you can do the backup only sequential start with the webserver
  - The larger the amount of data gets the more unlikely it is that you can use this technique
  - Handle the case when one of the backups fail

# Multi Server Backup

- Hard case: 24x7 operation
  - When you backup the webservers with a VSS snapshot, execute a script that inserts a log mark in all databases
  - BEGIN TRANSACTION
    - 'SharepointBackup09062010' WITH MARK
  - UPDATE dummytable SET dummyscol = 1;
  - COMMIT TRANSACTION;

# Restore

- Restore all databases up to the logmark
- RESTORE LOG ...

WITH

STOPATMARK = 'SharepointBackup09062010'

# Single item restore

- Need extra hardware
  - Database Server with space to full restore site collection
- Automatic
  - Extra tool (not out of the box)
    - Microsoft: SCDPM (Data Protection Manager)
    - 3<sup>rd</sup> Party Backup Solution
- Manual

# Review

- What does “high-availability” mean?
- Requirements
  - What are you trying to achieve?
  - What obstacles are in the way?
  - How do you successfully compromise?
- Planning a backup/restore strategy

# FRAGEN ?



Wir sehen uns wieder:



Im Februar 2011 in München  
[www.SharePointKonferenz.de](http://www.SharePointKonferenz.de)



ppedv Trainings & Events  
[www.ppedv.at](http://www.ppedv.at)

# Hat Ihnen mein Vortrag gefallen?



## Ich freue mich auf Ihr Feedback!

VIELEN DANK!