

mow

M

M

X

Miracle OpenWorld 2010

mow

M

M

X

Miracle OpenWorld 2010

# Failure is not an option

## 24x7 VLDB administration

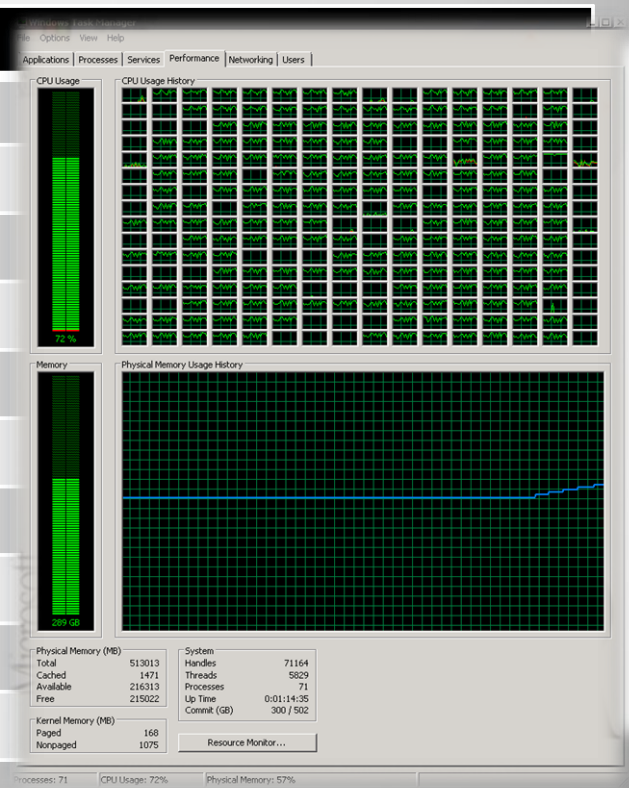
Thomas Grohser =tg=, bwin  
thomas.grohser@bwin.org

tg@grohser.com



# select \* from =tg=

@@Version	Remark
SQL 4.21	First SQL Server ever used (1994)
SQL 6.0	First Log Shipping with failover
SQL 6.5	First SQL Server Cluster (NT4.0 + Wolfpack)
SQL 7.0	2+ billion rows / month in a single Table
SQL 2000	938 days with 100% availability
SQL 2000 IA64	First SQL Server on Itanium IA64
SQL 2005 IA64	First OLTP long distance database mirroring
SQL 2008 IA64	First Replication into mirrored databases
SQL 2008R2 IA64	First 256 CPUs & >500.000 STMT/sec
SQL 11 (Denali)	Can't wait to push the limits even further



## Focus on SQL Server Infrastructure Architecture and Implementation Close Relationship with Microsoft

SQLCAT (SQL Server Customer Advisory Team)

SCAN (SQL Server Customer Advisory Network)

TAP (Technology Adoption Program SQL2008R2 and SQL11)

Close relationship with Hardware Vendors (Focus IA64)



M M X

**Active PASS member  
and PASS Summit Speaker  
SQL Server User Group Austria**



Miracle OpenWorld 2010



**World's biggest publicly listed online gaming platform**

World's **leading provider** of online Sports Betting

One of the largest **Poker networks**

Comprehensive range of **Payment Service Providing**

Integrated gaming portal - **22 languages, 25 core markets**

**Gross gaming revenues 2008 (GGR):**  
EUR **421 million**

More than **20 million registered customers**

**1,500 employees**

bwin builds on the strengths of the web in order to **tie up responsibility and gaming**

**15 million page views** and up to **980,000 users a day**



# Failure is not an option

## Agenda

- The Mission
- The Solution
  - Standardizing
  - Zero data loss
  - High availability
  - Scale up
- The Details
  - SQL Server logins
  - SQL Server jobs
  - Log Shipping
  - Partner databases
  - Replication

24x7 OLTP Database  
Management for VLDB



# Failure is not an option

## The Mission

- VLDB – A database that needs attention it's not size alone
- SLA
  - Zero data loss & 100% transactional consistency on financial transactions
  - 99.99x% availability @ 24 x 7
  - 450.000+ SQL Statements per second
  - Assumed worst case scenario: full datacenter failure with complete data loss within the datacenter
- Budget: unlimited (not kidding)



# Failure is not an option

## The Solution

- Standardize everything
- Work by the book
- Have some clever guys at hand

if the book runs out of pages



M M X

# Failure is not an option

## Standardizing

- Operating System
  - Version, Edition, Service Pack, Patch Level
- File System and Disks



# Failure is not an option

## File System Example

- C:\
- C:\Windows
- C:\Install
- C:\SQL01
- C:\SQL01\BIN
- C:\SQL01\TEMPLOG01
- C:\SQL01\TEMPDATA01
- C:\SQL01\LOG01
- C:\SQL01\LOG02
- C:\SQL01\DATA01
- C:\SQL01\DATA02
- C:\SQL01\DATA03
- C:\SQL01\DATA04



# Failure is not an option

## File System expansion

- C:\
- C:\Windows
- C:\Install
- C:\SQL01
- C:\SQL01\BIN
- C:\SQL01\TEMPLOG01
- C:\SQL01\TEMPDATA01
- C:\SQL01\LOG01
- C:\SQL01\LOG02
- C:\SQL01\DATA01
- C:\SQL01\DATA02
- C:\SQL01\DATA03
- C:\SQL01\DATA04



# Failure is not an option

## File System settings

- Stripesize 64/128/256 kB  
depending on storage
- Partition alignment 64/128 kB  
depending on storage
- Cluster size 64 kB
- 100% write cache
- 0% read cache



# Failure is not an option

## Standardizing

- Operating System
  - Version, Edition, Service Pack, Patch Level
- File System and Disks
- SQL Server
  - Version, Edition, Service Pack, Patch Level
- Network
  - Separate network for data and backup
  - IP Schema
- Documentation, Documentation, ....

# Failure is not an option

## Zero data loss



- Redundant NIC
- Redundant Power Supply
- Data files on SAN  
(RAID 1/0 Multipath/2 Fabrics)
- Transaction log files on RAID 101



M M X

# Failure is not an option

## RAID 101



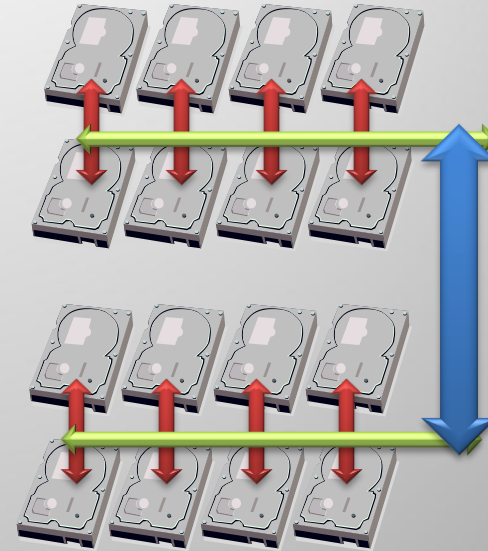
RAID Controller

RAID Controller

HW RAID 1

HW RAID 0

SW RAID 1



# Failure is not an option

## Zero data loss



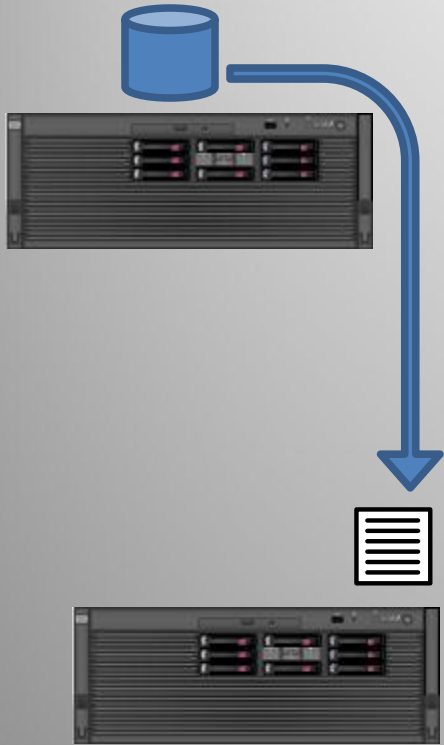
- Redundant NIC
- Redundant Power Supply
- Data files on SAN
  - (RAID 1/0 Multipath/2 Fabrics)
- Transaction log files on RAID 101



Availability: 0,00%  
Data loss: 100,00%

# Failure is not an option

## Zero data loss



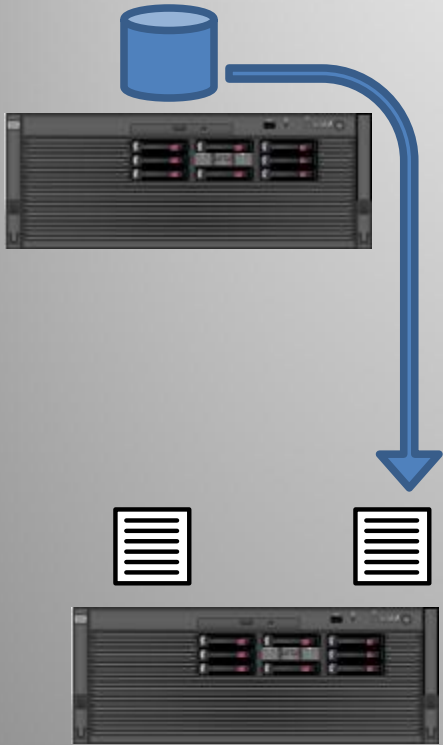
- Full backup every 24 h



Availability: 0,00%  
Data loss: 100,00%

# Failure is not an option

## Zero data loss



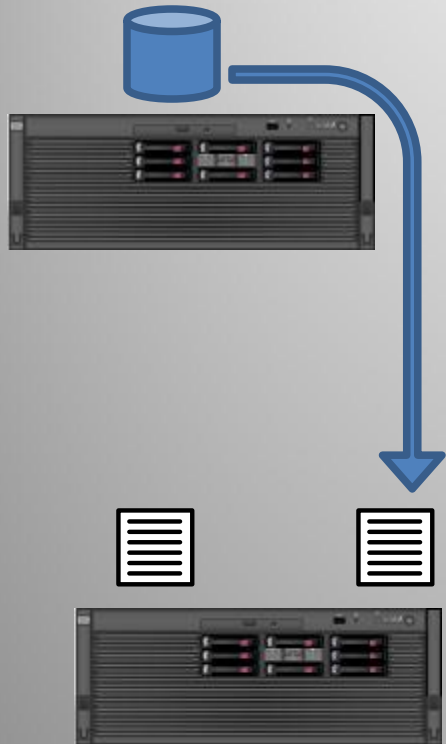
- Full backup every 24 h
- Alternating files



Availability: 0,00%  
Data loss: 100,00%

# Failure is not an option

## Zero data loss



- Full backup every 24 h
- Alternating files
- Transfer of files to tape in second location

**mow**

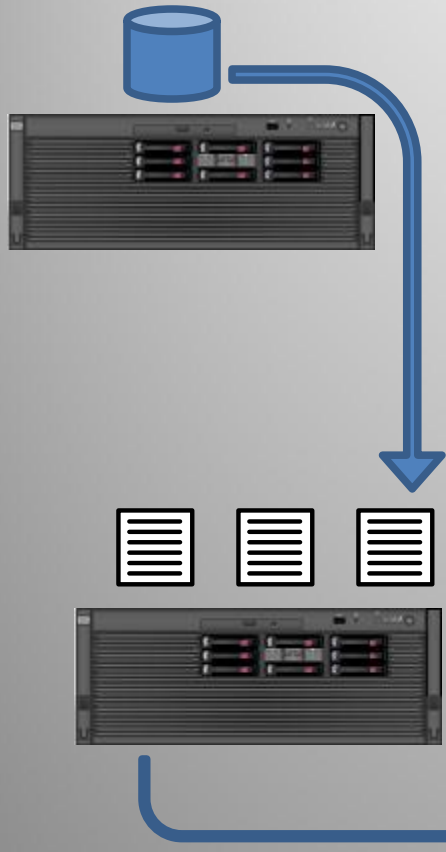
M M X

Availability: 0,00%  
Data loss: 100,00%



# Failure is not an option

## Zero data loss



- Full backup every 24 h
- Alternating files
- Transfer of files to tape in second location
- Verify of backups (daily)

**mow**

M M X

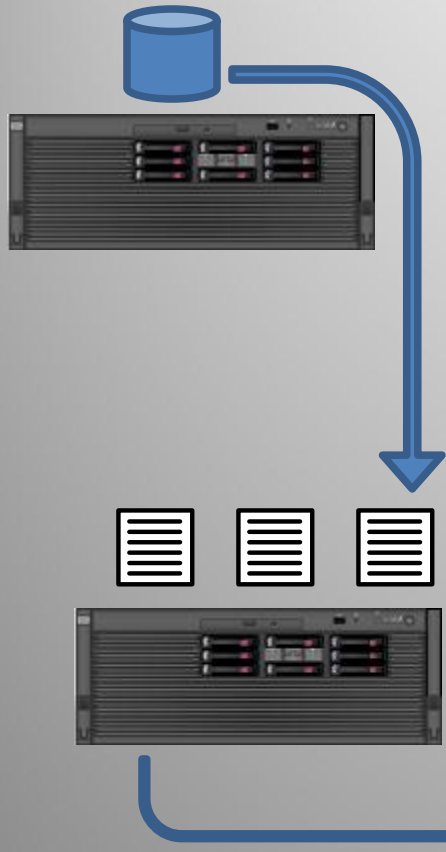
Availability: 98,00%

Data loss: 100,00%



# Failure is not an option

## Zero data loss



- Full backup every 24 h
- Alternating files
- Transfer of files to tape in second location
- Verify of backups (daily)
- Verify of tapes

**mow**

M M X

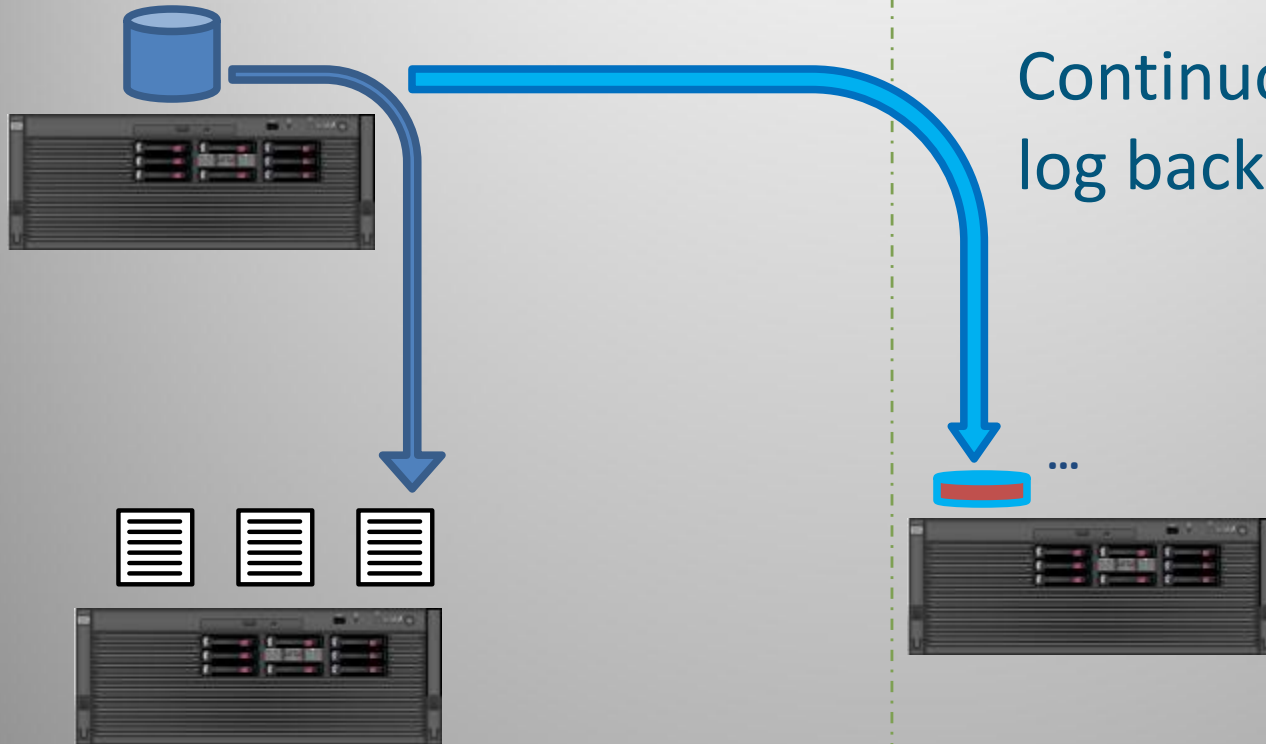
Availability: 98,00%

Data loss: 48 hours ...



# Failure is not an option

## Zero data loss



Continuous transaction log backups

**mow**

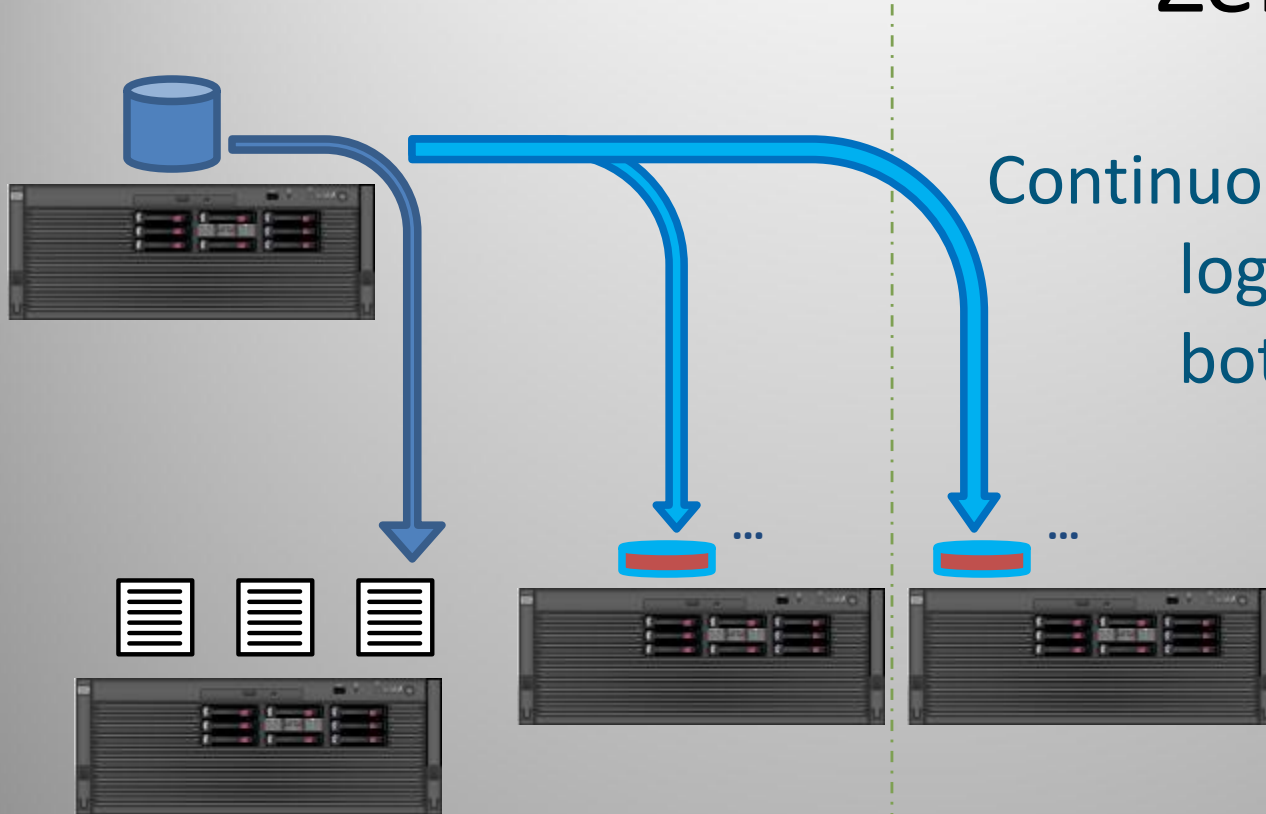
M M X

Availability: 98,00%  
Data loss: 48 hours ...



# Failure is not an option

## Zero data loss



Continuous Transaction  
log backups to  
both locations

**mow**

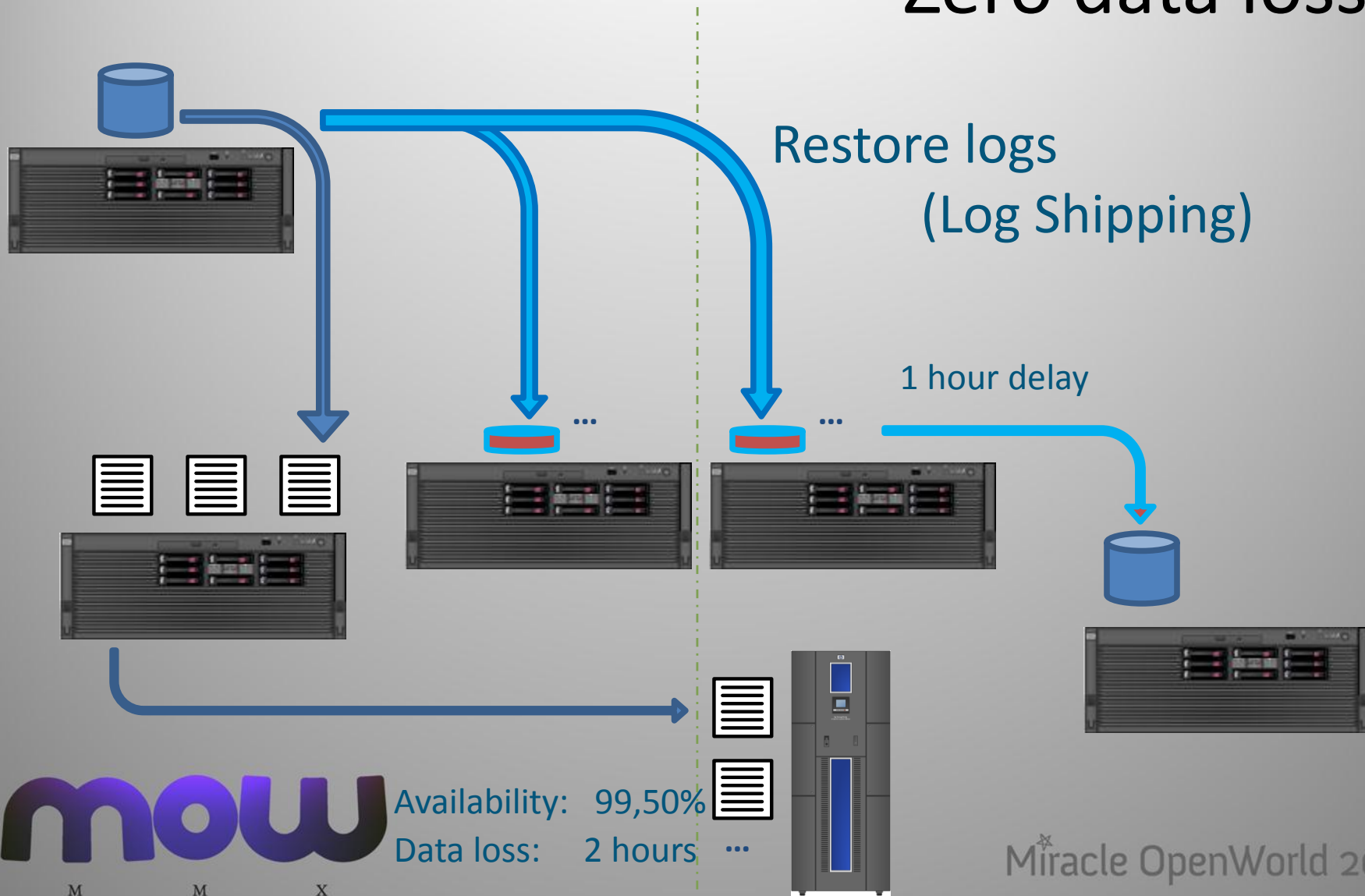
M M X

Availability: 98,00%  
Data loss: 48 hours ...



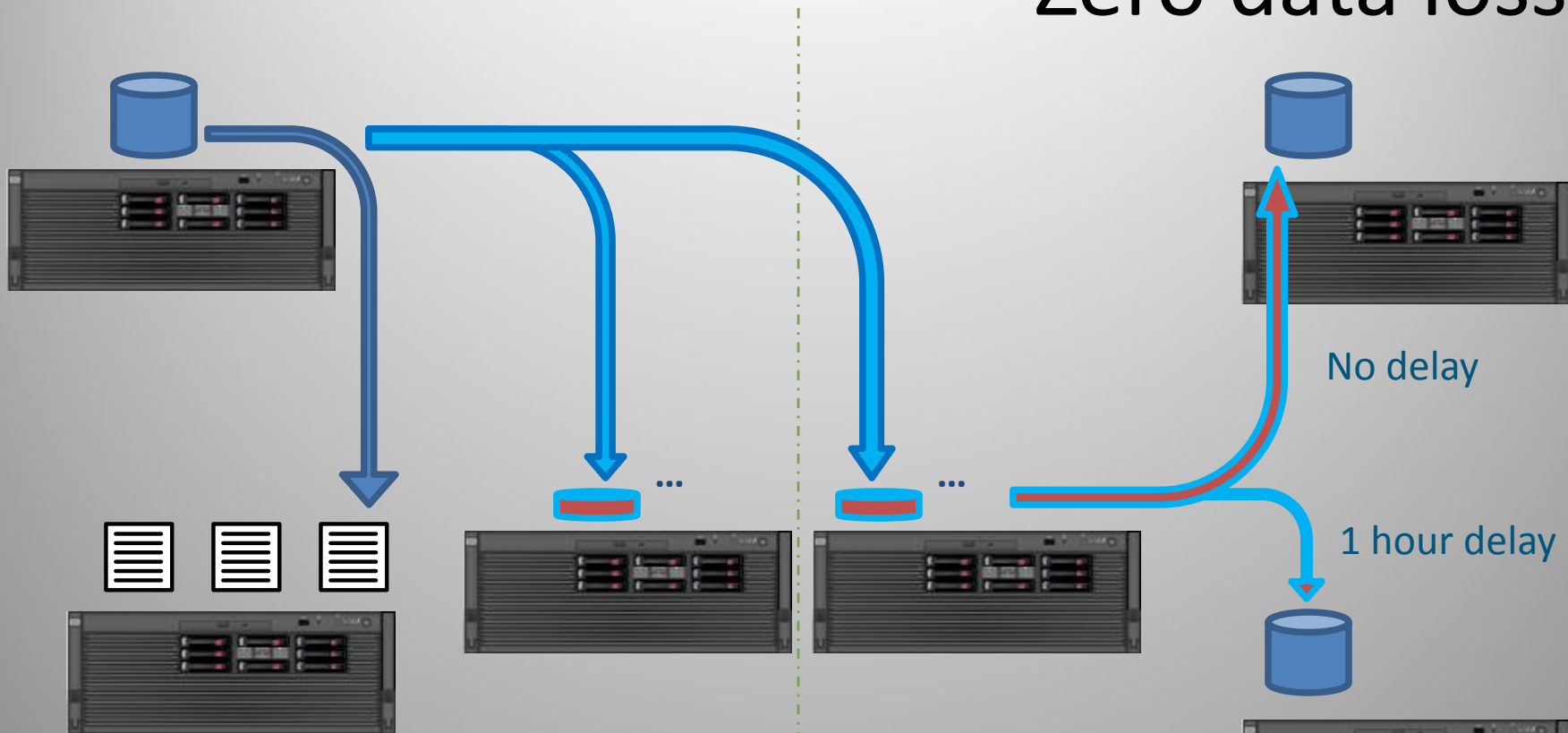
# Failure is not an option

## Zero data loss



# Failure is not an option

## Zero data loss



**mow**

M M X

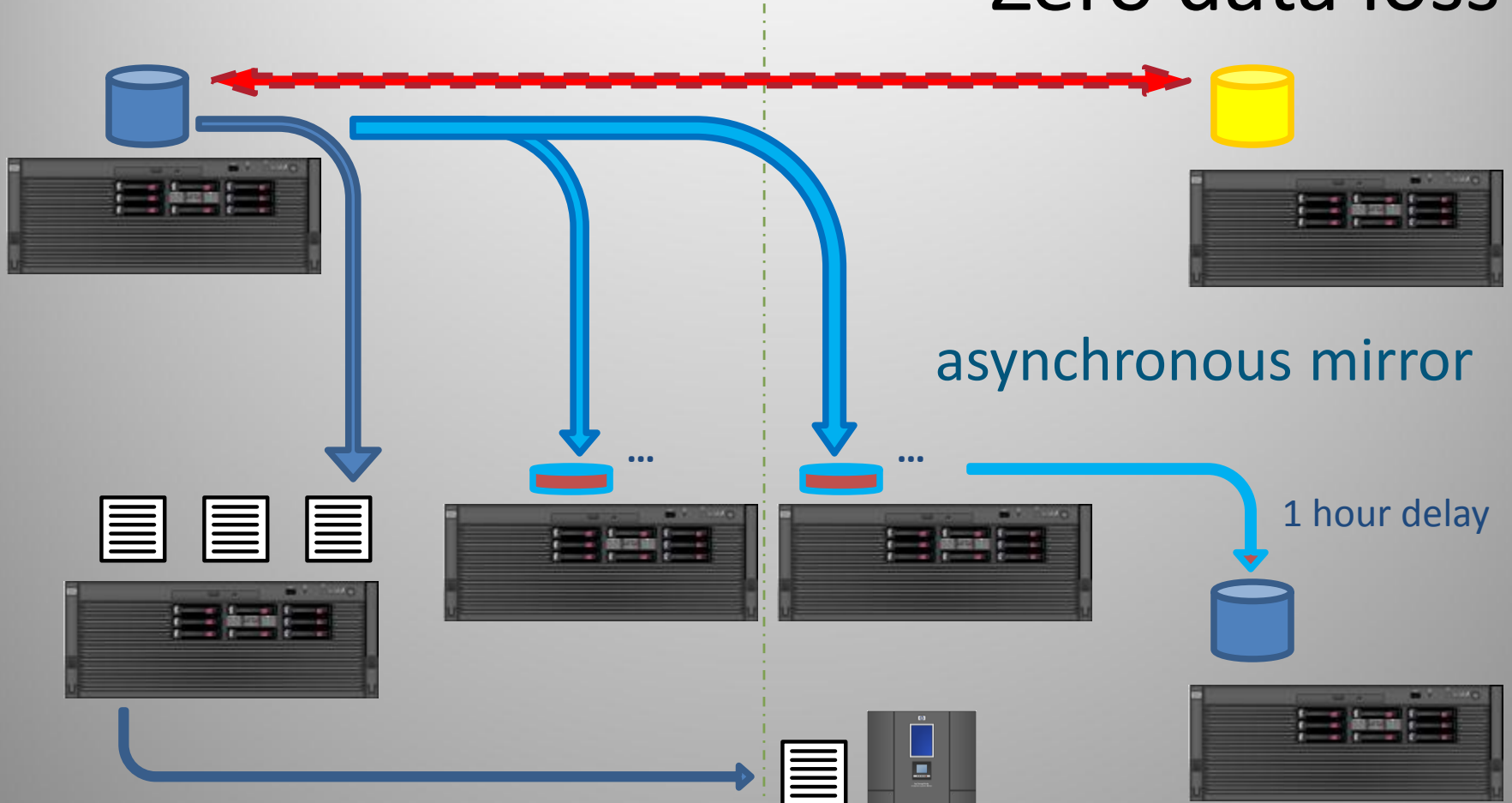
Availability: 99,80%  
Data loss: minutes ...



Miracle OpenWorld 2010

# Failure is not an option

## Zero data loss



asynchronous mirror

1 hour delay

**mow**

Availability: 99,80%

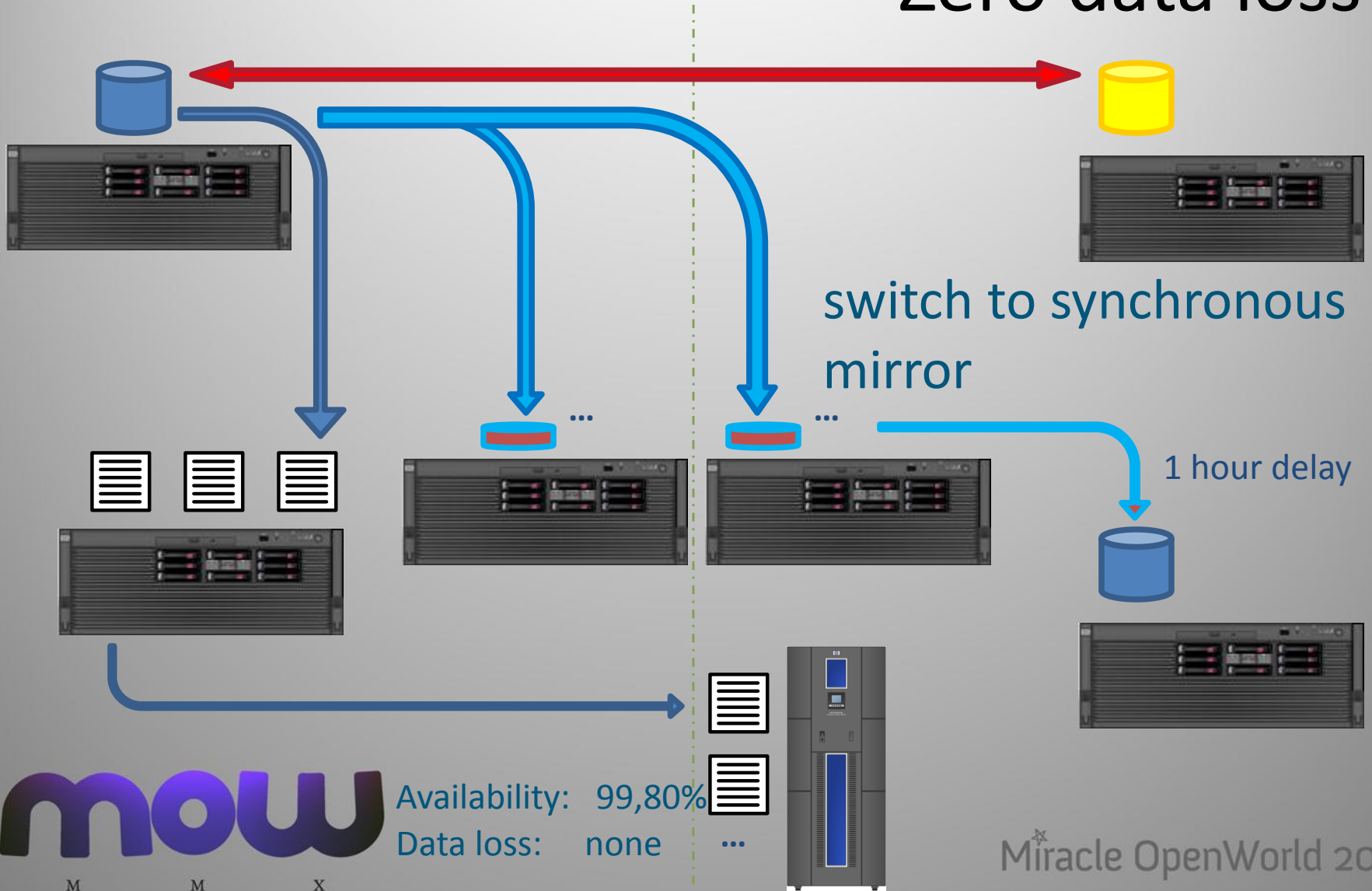
Data loss: seconds ...

Miracle OpenWorld 2010

M M X

# Failure is not an option

## Zero data loss

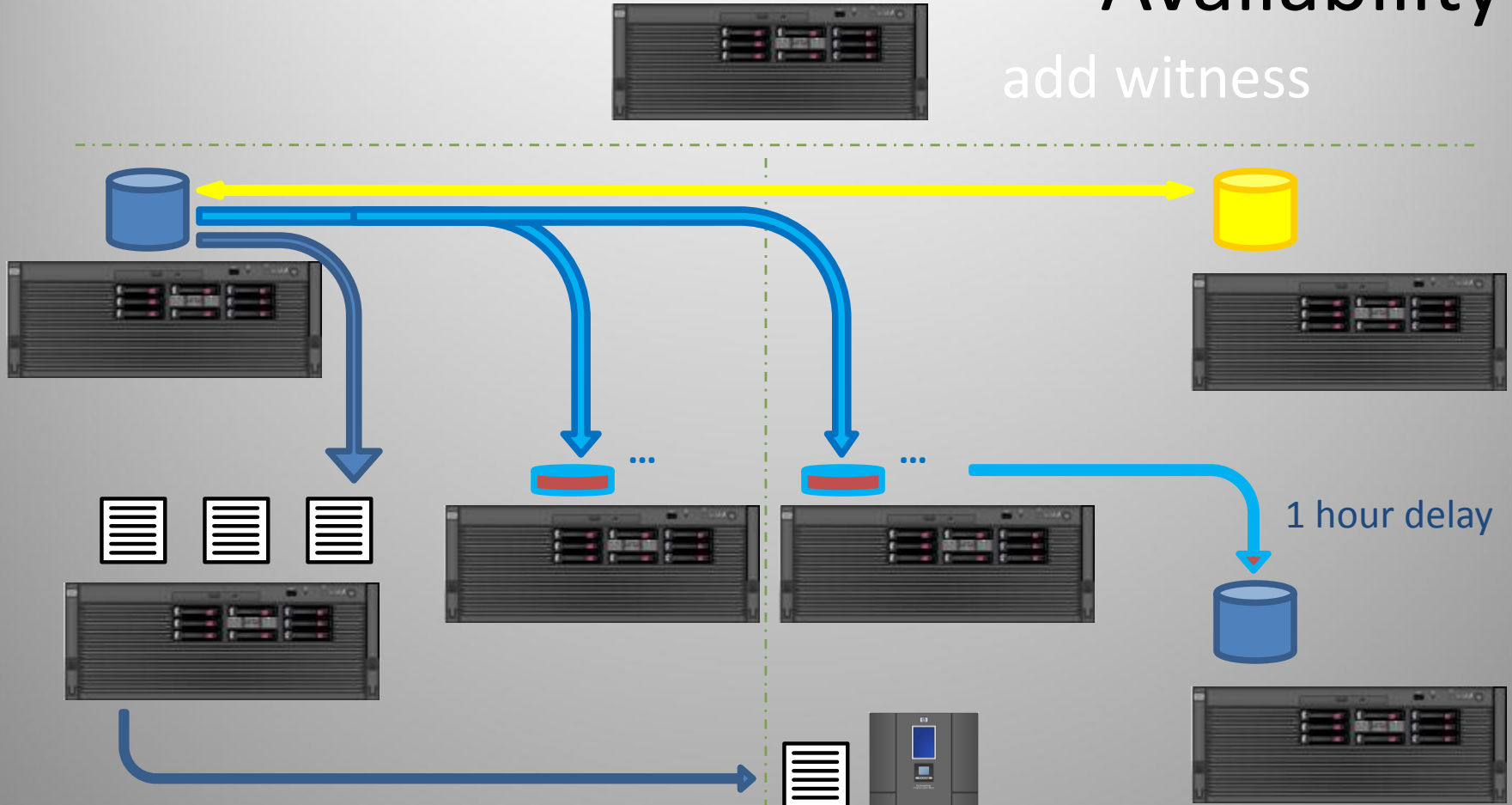




# Failure is not an option

## Availability

add witness



**mow**

Availability: 99,99%  
Data loss: none

Miracle OpenWorld 2010

M M X

# Failure is not an option

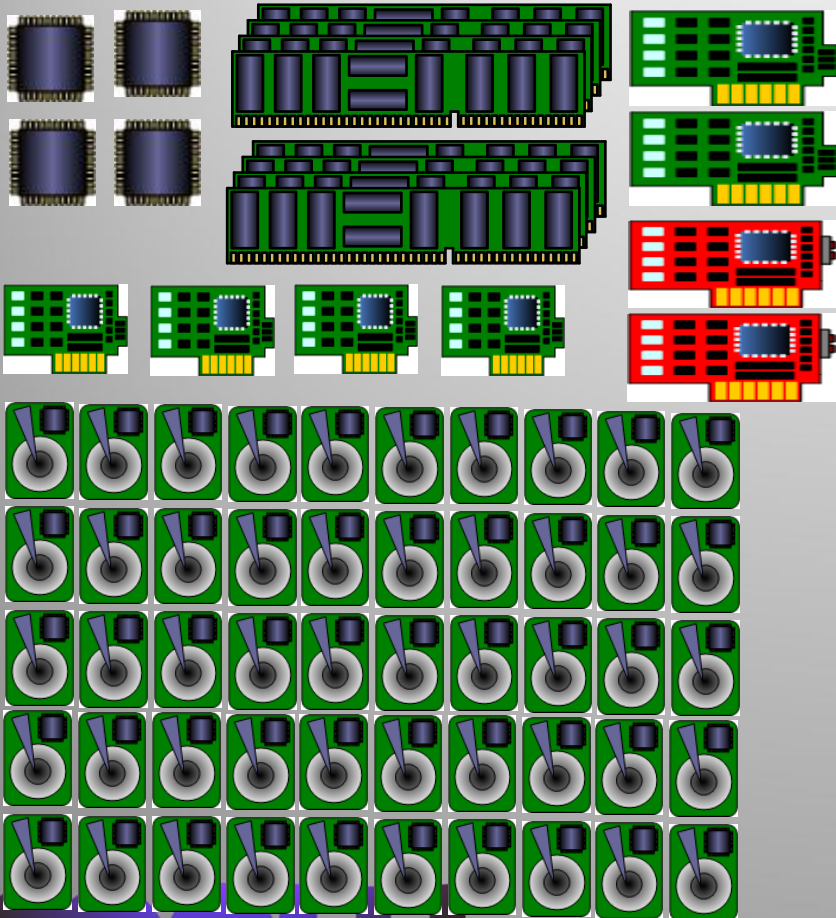
## Scale Up

- Selected CPU IA64 / Itanium 2
- Selected server/memory architecture SMP / NUMA
- SQL Server 2008 Enterprise Edition
- Windows Server 2008 R2 for Itanium-Based Systems



# Failure is not an option

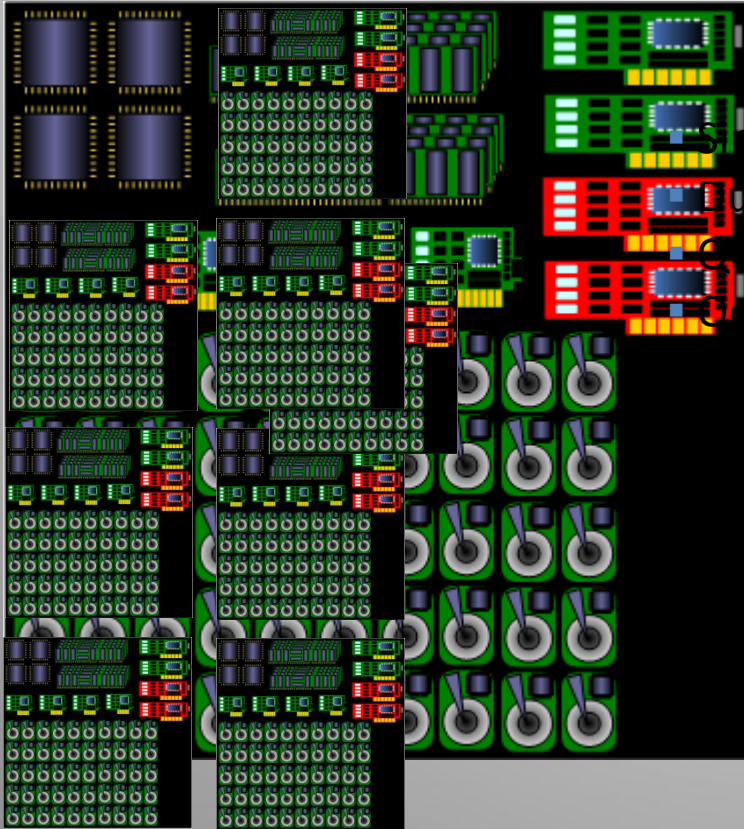
## Scale Up – Single NUMA node



- 4 x Dual Core ITANIUM 2 CPUs  
24 MB cache each
- 64 GB memory
- 4 x dual port 1 Gb/s network card
- 2 x dual port HBA (4Gb/s)
- 2 x P800 RAID controller
- 50 x 72 GB 15kRPM SAS disks
- SAN storage as needed  
n x 512GB (on 64 spindles each)

# Failure is not an option

## Scale Up



	cores	GB	disks	NIC	HBA
Single	8	64	50	8	4
Dual	16	128	100	16	8
Quad	32	256	200	32	16
Octal	64	512	400	64	32

Almost linear scaling



# Failure is not an option

## Scale Up

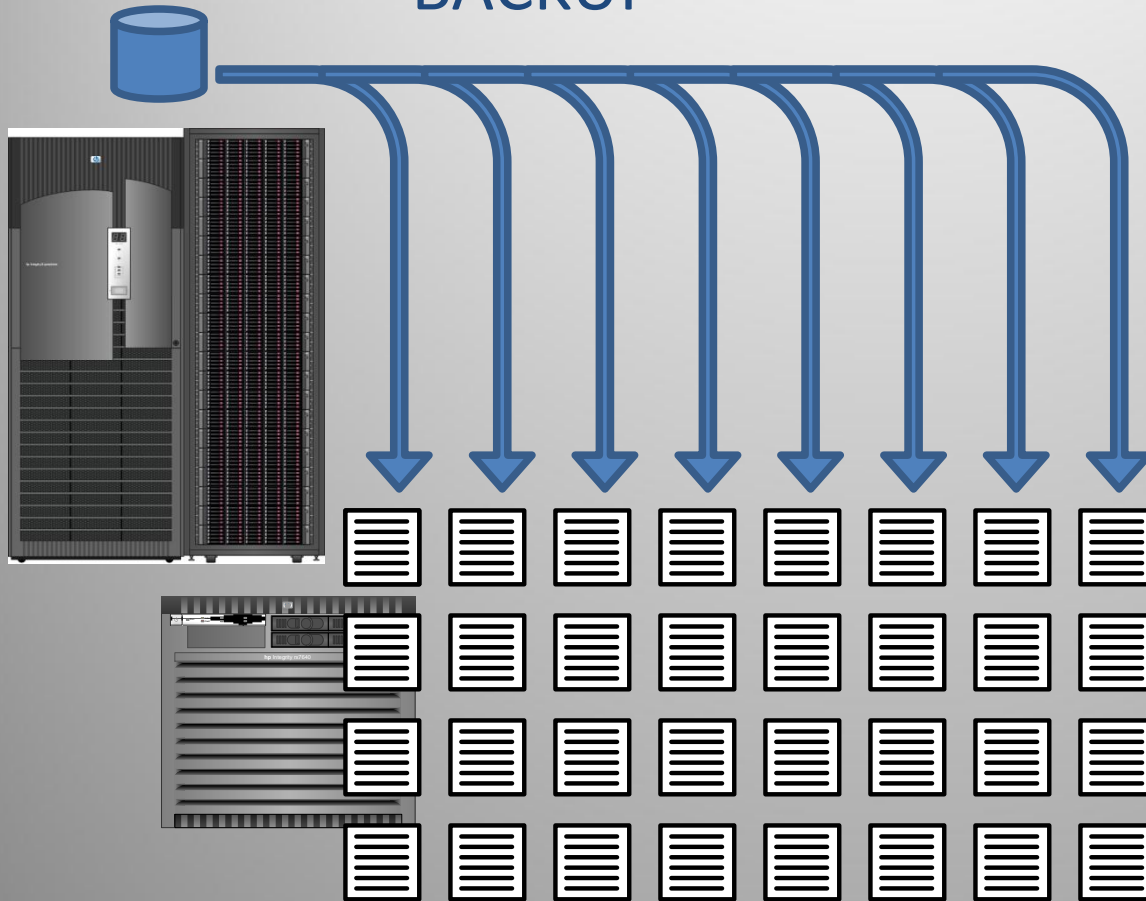
- 1 NUMA Node Server
  - 2 x NUMA node basic configurationplus
  - 2 x P600 (512MB cache)
  - 16 x 72 GB 15kRPM SAS disks
- 2 NUMA Node Server
  - 2 x NUMA node basic configuration
- 4/8/16 NUMA Node Server
  - 4/8/16 x NUMA node basic configurationplus
  - 2 x single port 10 GE network card



# Failure is not an option

## Scale Up

BACKUP



- Use eight parallel one GB/s sec network interface cards (one physical network, eight subnets)

- Use 32 parallel backup files each on a separate set of spindles with aligned partitions

- Transfer four files per network interface card

**mow**

M M X


# Failure is not an option

## Scale Up

SQL Server	IP Address	Network Mask
Network Card 1	192.168.1.2	255.255.255.0
Network Card 2	192.168.2.2	255.255.255.0

File Server	IP Address	Network Mask
Network Card 1	192.168.1.1	255.255.255.0
Network Card 2	192.168.2.1	255.255.255.0



# Failure is not an option

## Scale Up

```
BACKUP DATABASE MyVLDB
TO
    DISK=' \\192.168.1.1\backup\MyVLDB_1
    .bak',
    DISK=' \\192.168.2.1\backup\MyVLDB_2
    .bak`
WITH
    BLOCKSIZE = 8192
```

- Use Jumbo Frames if you can (+100%) with about 9016 bytes frame size

# Failure is not an option

## The Details

- Mirroring (not yet) as simple to manage as clustering
- SQL Server logins
- SQL Server jobs
- Log Shipping
- Replication
- Partner databases



# Failure is not an option

## Miracle Job

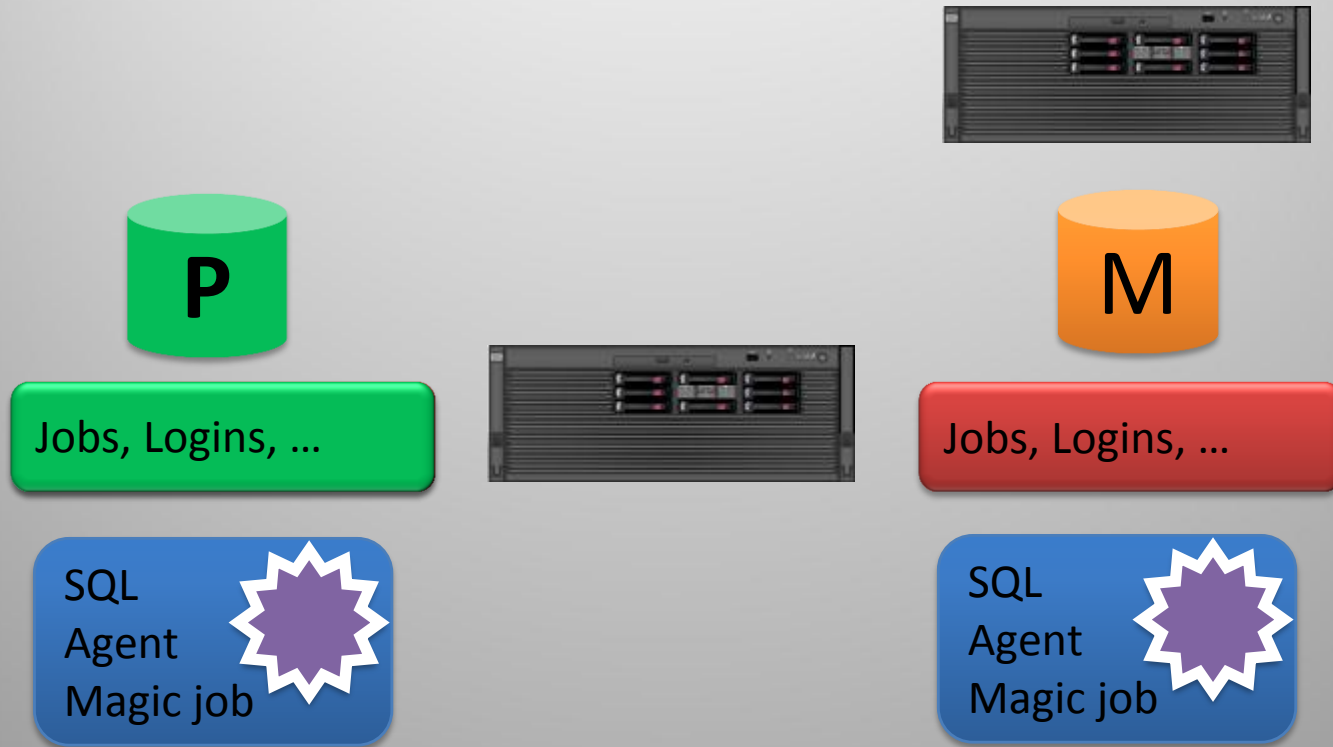
On **both** the principal and the mirror server:

- Create a helper job (e.g.: Manage Mirrors)  
with a schedule to run it once every minute
- Create a helper database (e.g.: admin)  
to store info like the last state of a database



# Failure is not an option

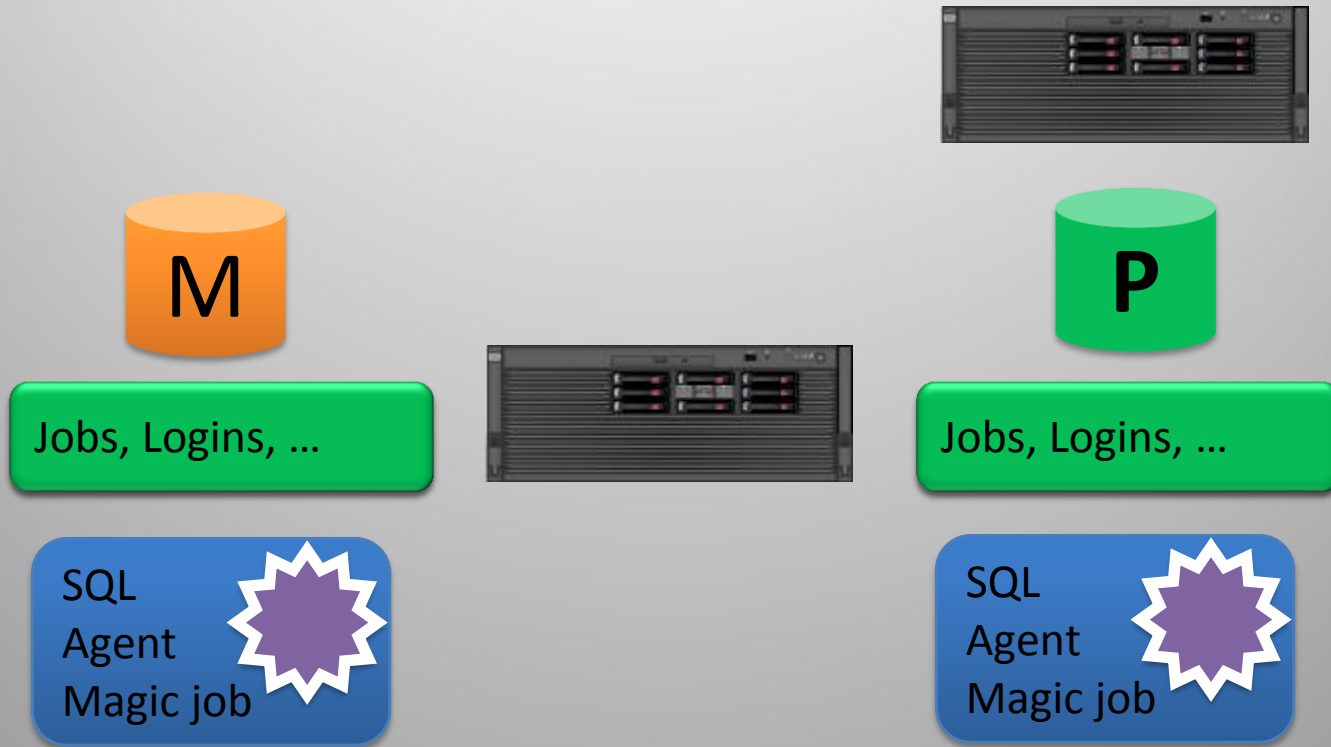
## Miracle Job



M M X

# Failure is not an option

## Miracle Job



M M X

# Failure is not an option

## Miracle Job

```
USE Admin;  
GO  
CREATE SCHEMA FailoverHandler;  
GO  
CREATE TABLE FailoverHandler.DBStatus  
(  
    database_id          int,  
    lastStatus           varchar(16),  
    lastStatusUpdateUTC  datetime  
) ;
```



# Failure is not an option

## Miracle Job

```
CREATE PROC AutoFailoverHandler.CleanupDB @dbID as int
AS BEGIN
    DECLARE @currentStatus AS varchar(16) =
        (SELECT state_desc FROM sys.databases WHERE database_id = @dbID);
    DECLARE @lastStatus AS varchar(16) =
        (SELECT isnull((SELECT lastStatus FROM
Admin.FailoverHandler.DBStatus
        WHERE database_id = @dbID
        AND lastStatusUpdateUTC>dateadd(minute,-5,GetUTCdate())),'N/A'));
    IF (@lastStatus <> @currentStatus)
    BEGIN
        -- Here we place the stuff to update
    END
    UPDATE FailoverHandler.DBStatus
        SET lastStatus = @currentStatus,lastStatusUpdateUTC=GetUTCDate()
        WHERE database_id = @dbID;
END
```



# Failure is not an option

## Miracle Job

- Execute the stored procedure for each mirrored database in a job step in our helper job on each server

```
EXEC AutoFailoverHandler.CleanupDB
    @dbID = db_id('myVLDB')
EXEC AutoFailoverHandler.CleanupDB
    @dbID = db_id('myOtherVLDB')
```



# Failure is not an option

## SQL Server logins

- Windows integrated logins must just be created on the mirror server, they use the Windows SID to map to the Database User.
- For each user / login pair where the login is a SQL Server login map the user with the login using

```
exec sp_change_users_login
```



M M X

# Failure is not an option

## SQL Server logins - Code

```
DECLARE @user AS TABLE (username sysname);
DECLARE @username as sysname;
INSERT INTO @user
    SELECT u.name as username
    FROM sys.sysusers u
        left outer join sys.syslogins l ON (u.sid = l.sid)
    WHERE u.islogin = 1 AND u.isntname <> 1 and u.isntgroup <> 1
        and u.hasdbaccess = 1 AND l.sid is null;
WHILE ((SELECT COUNT(*) FROM @USER) > 0)
BEGIN
    SET @username = (SELECT TOP(1) username from @user);
    EXEC sp_change_users_login
        @Action = 'Auto_Fix', @UserNamePattern = @username;
    DELETE FROM @user where username = @username
END
```



# Failure is not an option

## SQL Server jobs

- Have a first job step that checks if the database is online
- Check in every step
- Enable / Disable with

```
EXECUTE msdb.dbo.sp_update_job  
        @job_id = @jobID,  
        @enabled = 1;
```

- Special care must be taken for jobs that job starts when the server starts and therefore must start with database (used for forever running jobs)



# Failure is not an option

## SQL Server jobs

```
DECLARE @jobID as uniqueidentifier
WHILE 1=1
    BEGIN
        SET @jobID = (SELECT TOP(1) job_id from msdb.dbo.sysjobs
WHERE name like '%#MyDBName' and [enabled] = 0);
        IF (@jobID IS NULL) BREAK;
        EXECUTE msdb.dbo.sp_update_job @job_id=@jobID, @enabled=1;

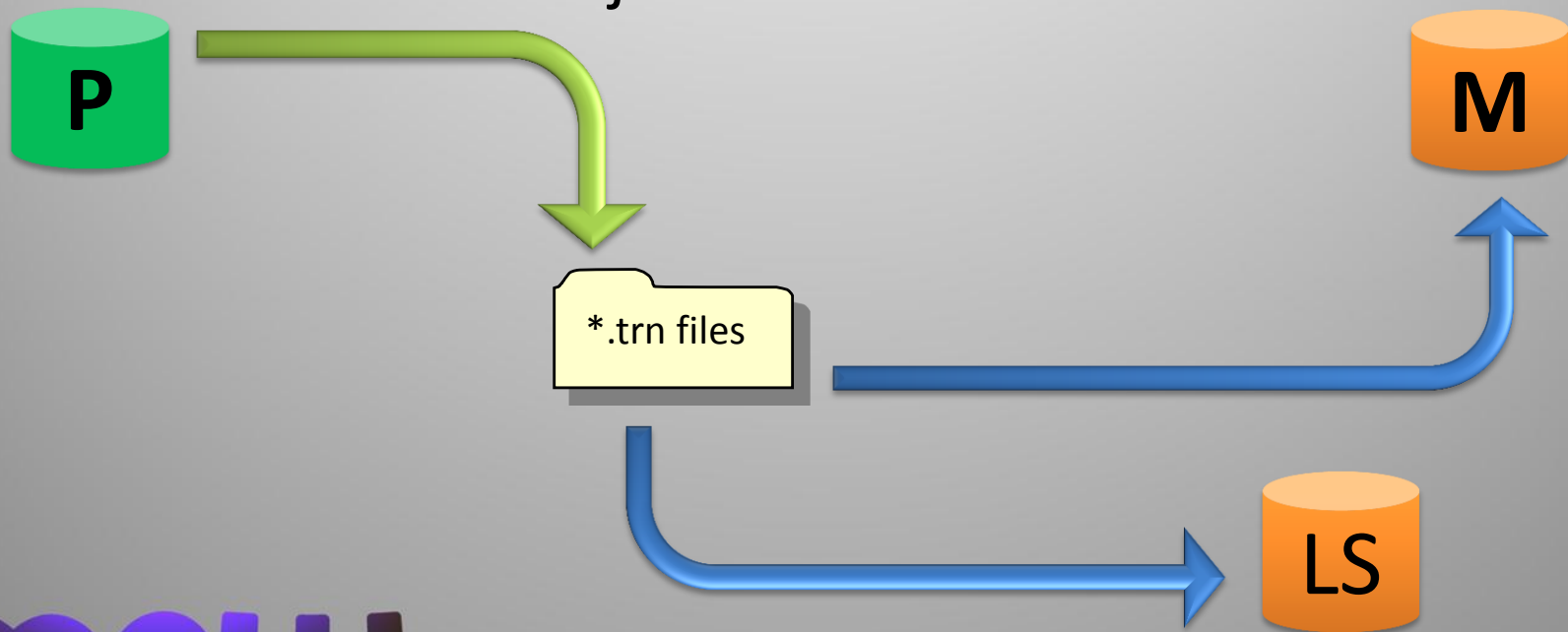
        IF ((SELECT freq_type FROM msdb.dbo.sysjobs j
            inner join msdb.dbo.sysjobschedules js ON (j.job_id=js.Job_id)
            inner join msdb.dbo.sysschedules s ON (js.schedule_id =
                s.schedule_id)
            WHERE j.job_id = @jobID) = 64)
            EXECUTE msdb.dbo.sp_start_job @job_id = @jobID;
    END
```



# Failure is not an option

## Log Shipping

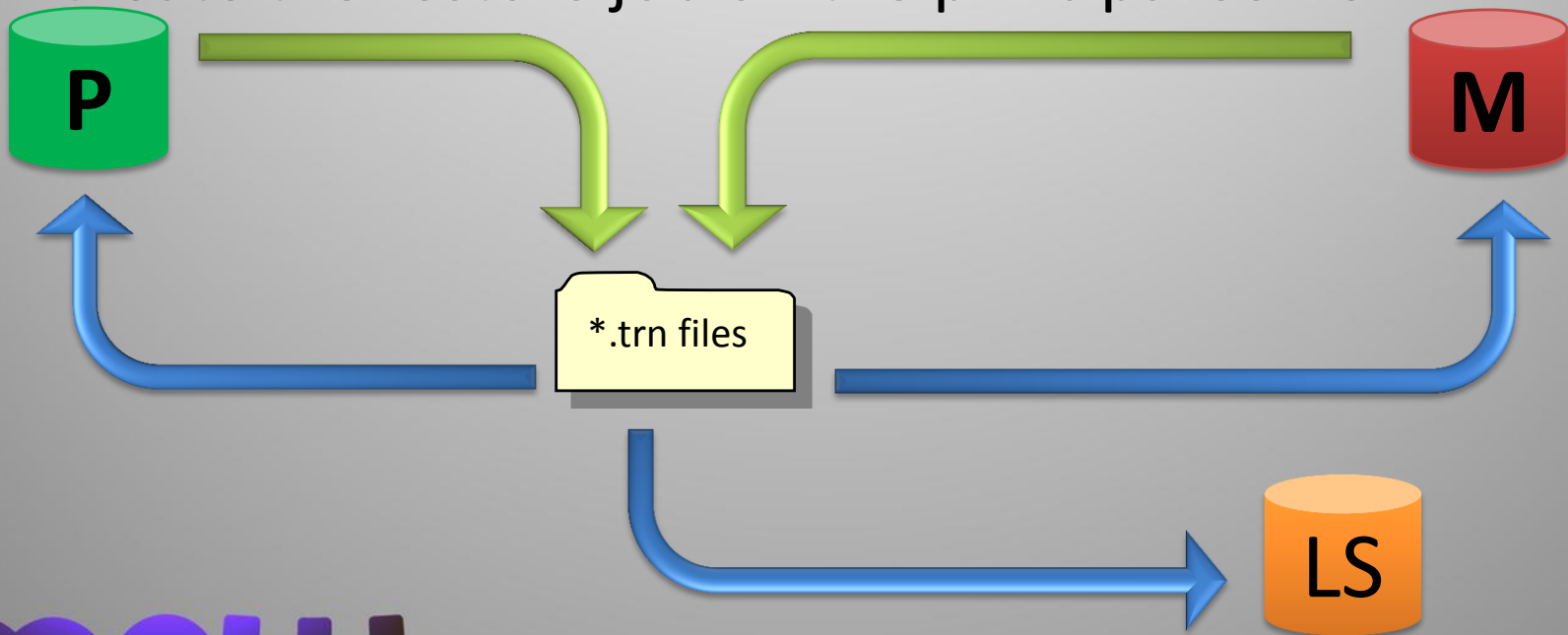
- Create the backup jobs on the principal server
- Create the restore jobs on the log shipping server
- Create the restore jobs on the mirror server



# Failure is not an option

## Log Shipping

- Failover the database
- Create the backup job on the mirror server
- Create the restore job on the principal server



# Failure is not an option

## Partner databases

- Partner databases are a databases that must be online on the same server
- Therefore if one database fails over to the mirror all others must failover too
- No problem if the server fails, because all db's will failover
- Otherwise we must help a little with

```
ALTER DATABASE myOtherDB SET PARTNER  
FAILOVER
```

- The code can be found in the demo scripts



# Failure is not an option

## Replication

- Transactional Replication
  - from a mirrored publisher database
    - (2005 and 2008/2008R2 supported)
    - (distributor must be 2008/2008R2)
  - into a mirrored subscriber database
    - (2005 and 2008/2008R2 possible)
    - (distributor can be 2005 or 2008)



# Failure is not an option

## Replication



M M X

# Failure is not an option

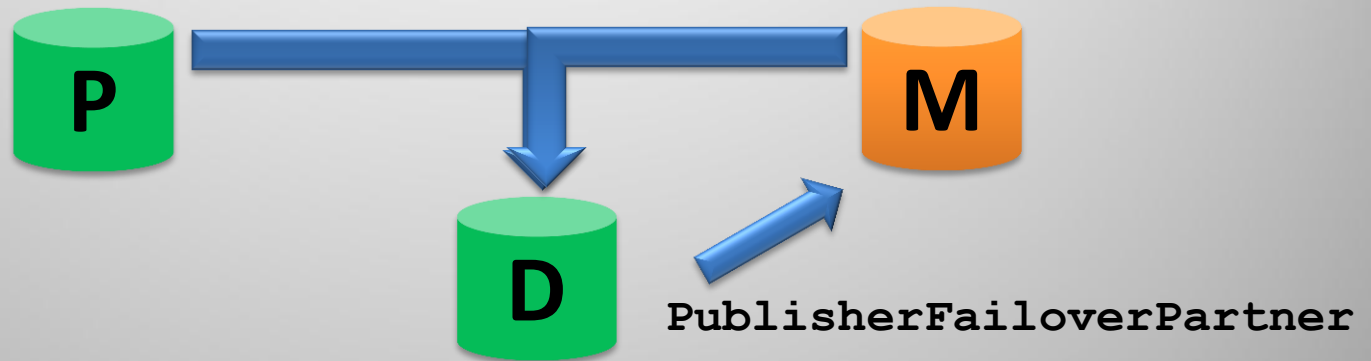
## Replication

- Publishing from a mirrored database  
(2005, 2008 or 2008R2)
  - Publisher : Create the publication as always
  - Distributor (2008): In the Agent Profile you must add a
    - `PublisherFailoverPartner`



# Failure is not an option

## Replication



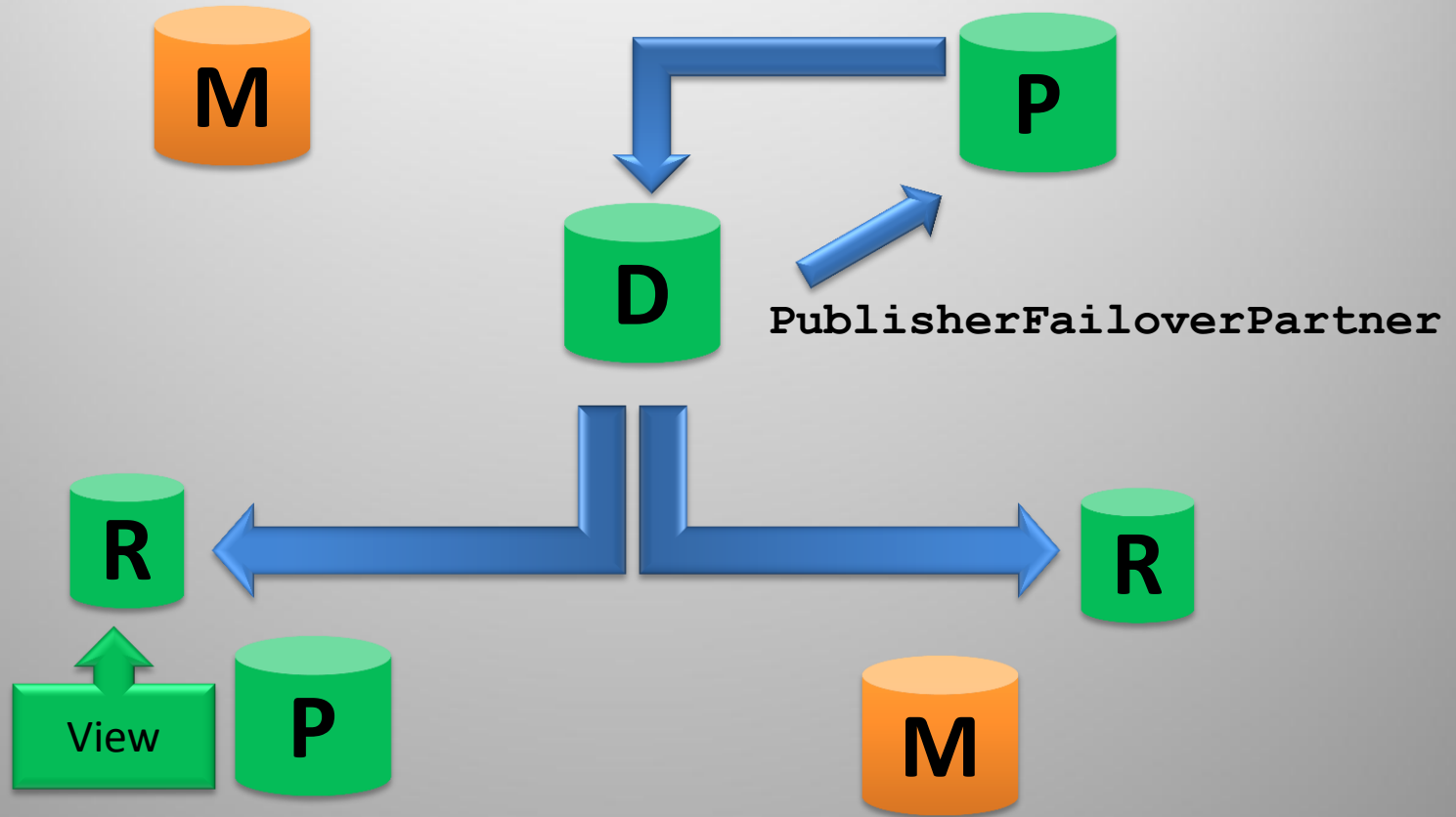
# Failure is not an option

## Replication

- Subscribing into a mirrored database (2005 or 2008)
  - Publisher /Distributor: Create the publication as always
  - Subscriber:
    - Create a helper database on each server (principal and mirror) (same name)
    - Create two subscription one into each of the helper databases
    - On principal database create a view to the replicated data

# Failure is not an option

## Replication



Use **P**;

```
CREATE VIEW dbo.MyTable as
```

```
SELECT * FROM R.dbo.MyTable;
```

**mow**

M M X

# Failure is not an option

## Replication

- Subscriber: Alternative method:
  - Implement reinitialize form LSN
- White paper from Gopal Ashok (Microsoft Corporation) and Paul S. Randal (SQLskills.com)

<http://download.microsoft.com/download/d/9/4/d948f981-926e-40fa-a026-5bfcf076d9b9/ReplicationAndDBM.docx>



# Failure is not an option

## Call to action

- Establish a SLA
- Standardize your environment
- Use your knowledge to build
  - Reliable
  - Highly Available
  - Extreme performing

SQL Server Solutions  
fulfilling the SLA



# Questions?

tg@grohser.com



M M X

Miracle OpenWorld 2010