

SQLskills Immersion Event

IE3: High Availability & Disaster Recovery

Module 1: Designing a Strategy

Paul S. Randal
Paul@SQLskills.com



Overview

- HA vs. DR
- Requirements
- Limitations
- DR planning and testing



2

© SQLskills, All rights reserved.
<http://www.SQLskills.com>



Definition of 'High Availability'

- **'Availability' means that 'something' is able to be used as expected**
 - E.g. the back-end database behind a web-store is able to service new sales transactions
- **'High availability' means that the 'something' is protected by various technologies to prevent it becoming unavailable**
 - E.g. the back-end database is protected with database mirroring so continues to be available if a disaster strikes
- **Basically, users/applications are always able to do what they need to be able to do**
- **But what is the 'something' mentioned above?**

What is the 'Something'?

- **The 'something' will vary by situation and so will the protecting technologies**
- **E.g. a table**
 - Could be protected by replication, or a solution that protects the whole database
- **E.g. a group of databases**
 - Could be protected by an Availability Group in SQL 2012
- **E.g. a server**
 - Could be protected by failover-clustering
- **E.g. a data-centre**
 - Could be protected using SAN-replication

What High Availability is NOT...

- **HA is NOT a single, pre-packaged solution from any vendor**
 - Hardware, software or application
- **HA is NOT a technology choice isolated from:**
 - Business knowledge and risk assessment
 - The cost of downtime
 - The cost of data loss
- **HA is NOT a business decision isolated from:**
 - The cost/complexity of design and implementation
 - The cost/complexity of management and monitoring

HA is Also NOT DR

- **HA is how you protect something against becoming unavailable**
- **DR is how you make something available again after it has become unavailable**
- **Example:**
 - Implementing database mirroring is adding an HA technology to protect a database, but performing a manual failover is performing DR using the HA technology after a failure
- **Many people consider HA and DR together because they're so interlinked**

Disaster Recovery Planning

- **Designing a disaster-recovery 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**
 - Even if redundancy is available when disaster strikes, failover may not be automatic or the first resort
- **The requirements gathering process feeds into the backup/restore strategy too**
 - And the fact that storage space and management are required for backups then feeds back into the requirements
- **What restores you need to be able to do depends on:**
 - What needs to be brought online first
 - Data loss SLA
 - Downtime SLA

How To Plan an HA/DR Strategy

- **First answer from many people is “get a failover cluster” or the latest new whiz-bang feature: wrong!**
- **It takes a very layered approach**
 - Requirements gathering
 - Requirements ordering
 - Limitations gathering
 - Trade-off/compromise
 - Technology evaluation and choice
- **It requires the right solution for the problem**
- **You can’t pick a technology without evaluating requirements**
 - Don’t just choose incumbent technologies
- **It needs to be constantly re-evaluated, re-examined, updated**
- **It should just work, even through a fault**
 - If your system isn’t designed to protect against the fault that occurs then what’s the point?

Aim of Requirements Gathering

- **Goal:** an ordered list of what needs to be protected and to what level
- **Examples:**
 - Database X must be available as close to 24x7 as possible, and no data loss can be tolerated
 - 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
- **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 and also they're basic design input
- **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!**

Downtime SLA

- **Maximum allowable downtime or RTO**
 - Recovery Time Objective
- **Commonly discussed in terms of 'number of nines'**
 - 5-nines = 99.999% uptime
 - Slightly over 5 minutes downtime per year
 - 4-nines = 99.99% uptime
 - Almost 52.5 minutes downtime per year
 - 3-nines = 99.9% uptime
 - Almost 8.75 hours downtime per year
- **Must consider how downtime is defined for you**
 - Is it 24x7 or, maybe just 9am-5pm weekdays
- **Achieving 5-nines of 24x7 operation is very hard**
- **Management may ask for zero downtime!**
 - Not technically possible
- **You have to know how technologies introduce downtime**
 - E.g. failover time with database mirroring

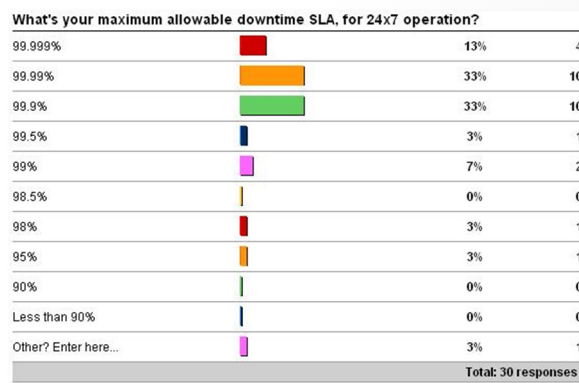
Data-Loss SLA

- **Maximum allowable data loss or RPO**
 - Recovery Point Objective
- **Must consider how data loss is defined for you**
 - Number of transactions
 - Work done in a period of time
- **May be different for different tables or databases**
- **Zero data loss is much more easily achievable than 5-nines uptime**
- **Press the business owners for proper answer**
- **Management may ask for zero data loss too!**
 - This IS achievable, but is it absolutely necessary?

Defining SLAs

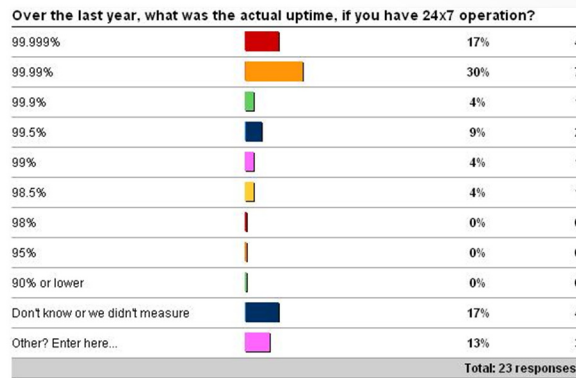
- Although management will want zero downtime and zero data loss, this is usually unreasonable
- All interested parties need to be involved in defining SLAs otherwise agreement may not be reached
- SLAs may need to be revisited and tweaked after:
 - The design limitations have been identified
 - The design is complete
 - The implementation and testing phases are complete
- SLAs should also be periodically reassessed to account for:
 - Changing business requirements
 - Increases/decreases in workload
 - Changes in application behavior

SLA Survey: 24x7 Target



- Source: <http://www.sqlskills.com/BLOGS/PAUL/post/Importance-of-defining-and-measuring-SLAs.aspx> (<http://bit.ly/18qRia>)

SLA Survey: 24x7 Actual



- Source: <http://www.sqlskills.com/BLOGS/PAUL/post/Importance-of-defining-and-measuring-SLAs.aspx> (<http://bit.ly/18qRia>)

Tracking SLAs

- Not much point having SLAs if you don't track progress against them
- Monitor all defined SLAs and then periodically reassess them and/or HA strategy
 - If they're not being met, decrease them or change HA strategy
 - If they're being met, consider pushing them higher?
- Don't fudge the numbers, you're just kidding yourself!

Assess the Risk of Unavailability

- **Assess your potential risks for downtime**
 - Site is unavailable
 - System is unavailable
 - Database is completely offline/unavailable
 - Database is partially offline/unavailable
 - Table is unavailable
 - Data/row(s) are unavailable
- **What is going to be the greatest impact on the business?**
- **The strategy should cope with the highest impact disasters first**
 - Site loss might not be highest impact if critical data is available in another location

Site Failure Considerations

- **Problems by region**



- **Where is power cheap?**

Top Requirements Questions

- **What are you trying to protect?**
 - A list ranked by risk/cost
- **What are the downtime SLAs?**
- **What are the data-loss SLAs?**
- **What are the performance SLAs?**
 - Some technologies can impact performance in a big way
 - E.g. synchronous database mirroring
- **What is the application ecosystem?**
 - E.g. what dependencies are there?
- **Should failover be automatic?**

What Has to Come Online First?

- **The order of recovery of parts of the data/database dictates physical layout and sometimes backup type**
- **E.g. the whole database**
 - Sounds simple, but not really
 - Consider size, tran-log generation rate, regulatory compliance
- **E.g. a single, critical table in a very large database**
 - Can you do single-table restore from native SQL backups? No
 - Requires the table to be isolated in a filegroup, or 3rd-party software, plus log backups
- **E.g. the most recent month's data in a very large table**
 - Requires partitioning, multiple-filegroups, and log backups

Consider Limitations

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

Non-Technical Limitations (1)

- **Power**
 - Consider power requirements for all parts of the system
 - Servers, drives, networking, HVAC
 - Do you need UPSs? Backup generators?
- **Space**
 - Only so many servers you can fit in a closet
 - Consider servers, I/O subsystem, networking, people, HVAC, PSUs
- **Air Conditioning**
 - Too many servers in the closet and they'll overheat
 - More HVAC means more power and space required too
- **Manpower**
 - Who's going to implement, monitor, and maintain the system?
 - Consider costs of failover site – vendor or company operated?
 - One person can't recover multiple systems simultaneously without the possibility of mistakes

Non-Technical Limitations (2)

- **Time to implement**
 - A good design and solid implementation won't happen overnight
 - After a disaster, HA usually becomes the hot topic
- **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. which teams need to be involved?
- **Most of the limitations boil down to BUDGET**
 - You can't implement a new data-centre with US\$10k

Technical Limitations (1)

- **What is the transaction log generation rate of your workload?**
 - Impacts database mirroring, log shipping, replication, log file size management, Availability Groups
- **How large are the average transactions?**
 - Impacts all failover technologies, log file size management
 - Smaller transactions affected more on sync mirroring and AGs
- **What is the maximum recovery time?**
- **What recovery model are you running?**
 - Database mirroring requires FULL, impacts index maintenance
- **What's the network bandwidth and latency to the second site?**
 - Impacts ability to implement any cross-site technology
- **Can you alter the application at all?**
 - Impacts ability to failover gracefully
- **What features do you need?**
 - FILESTREAM can heavily impact backup size
 - FILESTREAM not available with database mirroring

Technical Limitations (2)

- **Are you limited in terms of:**
 - CPU architecture
 - 32-bit is limited by worker threads which impacts database mirroring
 - Disk space
 - Impacts ability to implement partitioning, backup strategy, RAID 10
 - I/O subsystem
 - Low throughput can hamper all multi-instance HA technologies
 - Memory
 - Impacts ability to have multi-instance clustering
 - SQL Server Edition
 - Standard Edition can't use asynchronous database mirroring
- **How long does it take the physical server to boot?**
- **Does the hardware in the other data-centers match that in the primary data-center?**
 - If not, workload performance after failover may be lower

Compromise

- **Limitations may stop all requirements being met**
- **If they do, several solutions:**
 - Push back on limitations and/or requirements
 - Accept limitations, prioritize requirements, and design to meet them in order
- **Crucial that management are made aware of which requirements cannot be met and why**
 - Unacceptable for you to only complain for the first time when a disaster strikes

A Good Disaster Recovery Plan...

- Should be written by the most senior staff members who have seen the most failures
- Should be tested by the most junior staff members who may be on duty late at night
 - It won't be the most senior people on call on a holiday...
- Should be comprehensive
 - 'Restore the database from backups' isn't good enough
 - What if something goes wrong?
- Should consider human factors in a widespread disaster
- Should be tested regularly and updated

More DR Planning Considerations

- Consider glitches at each step:
 - What if the server is physically damaged?
 - Performing a bare-metal install of back-end and mid-tier servers...
 - What if the SAN is physically damaged?
 - How many USB drives can you find?
 - What if there's no power at the data centre?
 - What if there's no data centre?
 - Where are the off-site backups stored?
 - What if the backups are corrupt?
 - What if no people are available?
- What insurance do you have to buy new equipment?
- How do people work with no data center? Phones?

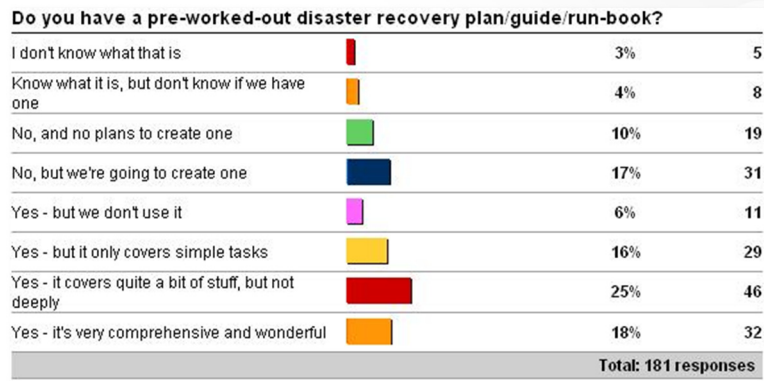
DR Planning: People Issues

- Who gets notified first of failures?
- Who is responsible (i.e. in charge of decision process) at each phase of recovery?
- Who's the 'sponsor' that can resolve disputes about progress method? (e.g. CIO)
- Who needs to be kept informed? (e.g. CIO, customers, users)
- Who has to authorize a failover?
- How do things get escalated if progress is not being made?
- Who's in overall command of the DR effort?
- Contact info for everyone who may become involved
- Which other teams need to be involved for success?
 - Server team, SAN team, network team, 3rd-party companies
- How to test the application is working after DR is performed

Examples

- (Reminders for me to tell stories)
- Christchurch earthquake building demolitions
- Nisqually earthquake coffee company
- Figure out if there are any volunteer emergency personnel in your organization
 - They may be able to get into a dangerous building to retrieve hard-drives
 - But they may be unavailable to help because they're busy responding...

DR Planning Survey (1)



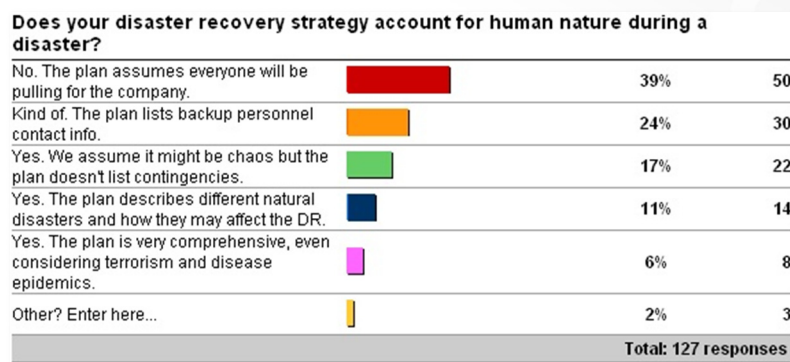
- Source: <http://www.sqlskills.com/BLOGS/PAUL/post/Importance-of-having-a-good-disaster-recovery-plan.aspx> (<http://bit.ly/rfnGD>)



31

© SQLskills. All rights reserved.
<http://www.SQLskills.com>

DR Planning Survey (2)



- Source: <http://www.sqlskills.com/BLOGS/PAUL/post/Human-nature-is-a-significant-hurdle-to-successful-disaster-recovery.aspx> (<http://bit.ly/RbcFP6>)



32

© SQLskills. All rights reserved.
<http://www.SQLskills.com>

Testing

- **Test the solution before going into production with various disasters**
 - Pull out a drive
 - Drop a table
 - Unplug a network cable
 - And everything else you've specified as a requirement
- **Try doing a bare metal install or a full restore from backups**
- **What if you can't meet your requirements?**
 - Push back or tweak the strategy as appropriate
 - Make sure management knows what's possible and what's not BEFORE going into production
- **Perform regular real-life disaster testing IN production**
 - No other way to test it for real... but easier said than done

DR Testing Survey (1)

Have you ever tested your disaster recovery plan, and if so, what happened?			
Don't know if we have a disaster recovery plan		6%	8
We definitely don't have a disaster recovery plan		12%	16
Never tested it, no idea if it works		17%	22
Tested it, we crashed and burned		1%	1
Tested it, we'd forgotten a few things		17%	23
Tested it, couldn't meet down-time SLA (RTO)		2%	2
Tested it, couldn't meet data-loss SLA (RPO)		0%	0
Tested it, couldn't meet either SLA		0%	0
Tested it once, everything went to plan		14%	19
Test it regularly, everything goes to plan		22%	29
Other? Enter here...		9%	12
Total: 132 responses			

- Source: <http://www.sqlskills.com/BLOGS/PAUL/post/Importance-of-testing-your-disaster-recovery-plan.aspx> (<http://bit.ly/BLAtT>)

DR Testing Survey (2)

When did your company last test the disaster recovery strategy?

Ha - we don't even take backups!		1%	1
We don't have a DR strategy		20%	32
We have a DR strategy but never test it		19%	30
Our management does not allow us to test it		3%	5
We last tested it more than 2 years ago		3%	4
We last tested it 1-2 years ago		8%	12
We test at least once per year		20%	32
We test at least every 6 months		8%	12
We test at least every 3 months		10%	16
We are legally mandated to test every X months		3%	4
Other? Enter here...		6%	9
Total: 157 responses			

- Source: <http://www.sqlskills.com/BLOGS/PAUL/post/Human-nature-is-a-significant-hurdle-to-successful-disaster-recovery.aspx> (<http://bit.ly/RbcFP6>)



35

© SQLskills. All rights reserved.
<http://www.sqlskills.com>

Database Layout Considerations

- Physical layout of the database can have a HUGE effect on the availability of the data
- Comes down to a few simple questions:
 - Do you split the database into multiple files/filegroups for performance?
 - Depends on I/O subsystem and possibility of contention
 - Remember from IE1 and IE2 classes:
 - Multiple data files can help
 - Multiple log files gives no performance gain
 - Tempdb is a special case
 - Do you split the database into multiple filegroups for availability?
 - What RAID level to use for availability?



36

© SQLskills. All rights reserved.
<http://www.sqlskills.com>

Choosing an Appropriate RAID Level

- **Consider:**
 - Read/write vs. read-mostly workload that the application puts on the database (e.g. OLTP vs. DW)
 - Performance characteristics of the RAID levels
 - E.g. RAID 5 requires an extra parity calculation and write for each data write, but that overhead may be masked by caching
 - Redundancy capabilities of the RAID levels
 - Resources available (usually budget)
- **For example (generalizations):**
 - RAID 5 is the least costly way to get some redundancy
 - Transaction logs usually go on RAID 1 or 10
- **What about SSDs?**
 - Don't just use RAID 0, even for tempdb

Essential Database Maintenance

- **When I teach database maintenance I say:**
 - 'you usually can't put a database into production and walk away'
- **A database must be maintained so it remains available:**
 - Index/statistics maintenance
 - File management
 - Page protection
 - Consistency checking
 - Monitoring
 - Backups

When to Pick Technologies?

- **Note I said 'technologies', not 'a technology'**
 - Most solutions involve multiple technologies
- **Once the final, post-compromise requirements are known, THEN and only then is the time to start evaluating technologies**
 - Choosing technologies then evaluating requirements and having to then change technology choice wastes time
- **Make sure to understand 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 only some or all of your requirements?
- **If, after evaluating technologies you can't meet all requirements, you'll need to reduce them**

Review

- HA vs. DR
- Requirements
- Limitations
- DR planning and testing

General Resources

- Whitepaper: High Availability with SQL Server 2008
 - <http://msdn.microsoft.com/en-us/library/ee523927.aspx>
- Whitepaper: Proven SQL Server Architectures for High Availability and Disaster Recovery
 - <http://tinyurl.com/2852t6w>
- Book: Pro SQL Server 2005 High Availability

Questions?

