

# The Top 3 Challenges of Adopting AlwaysOn And How to Overcome Them

EXECUTIVE SUMMARY

So many organizations are eager to embrace Microsoft SQL 2012/2014 for obvious reasons. You want take advantage of new functionality and use it to modernize your database infrastructure. Opportunity is knocking in the form of AlwaysOn failover, synchronous replication appropriate for any application and the option to read from secondary replicas. Aren't you going to answer?

Let's explore three common challenges behind embracing that new set of availability features in AlwaysOn, and get a sneak peek at the what's waiting on the other side of your next upgrade.

## 1. New load balancing features

Forget secondary servers waiting for their 15 minutes of fame. Now they can process read traffic which means you're doing more with the same hardware, and faster, too. But to get there, expect to modify applications. Simply connecting to a secondary server won't cut it. And you don't get a say in how that sharing works, either. SQL Server 2012/2014 doesn't support intelligent load balancing. It simply shifts traffic from one busy server to another server at random. That second one might be less crowded. Or not. It might be closest. Or not. So how can you take advantage of the new features without ripping everything apart first?

## 2. Stay ahead of the lag

AlwaysOn replicates data 24/7, but that doesn't mean your primary servers and secondary servers match up 100% of the time. Lags are part of the package and your organization must endure the sometimes painful change management process of understanding the consequences and agreeing on how big a lag is acceptable.

## 3. Failproof failovers

With failovers, failure isn't an option. Consistent, reliable architecture is critical, and with a distributed environment, you're eager to take advantage of SQL 2012's ability to scale beyond a single datacenter. But 2012 limits automated failover to servers sitting in the same datacenter. And because of the way VNN (Virtual Network Name / Active Group Listener) is built, it carries the same limitation.

## Meeting and Exceeding the Challenge

There's nothing small about Microsoft. The company's plan to upgrade to AlwaysOn in SQL Server 2014 presented a set of challenges as large and diverse as the organization itself. With end-of-life server hardware also in play, the company wanted to virtualize as part of the upgrade, too.

They presented ScaleArc this set of requirements: Virtualize the environment, migrate to AlwaysOn (no architecture changes allowed), overcome AlwaysOn's known limitations and orchestrate a failover plan with no downtime allowing for faster patching and testing.

ScaleArc's approach to database load balancing software allowed Microsoft to covert to SQL 2012 with flawless results. With the help of ScaleArc, Microsoft's most popular community platform now enjoys zero downtime for maintenance and patching, a read/write load balancing split with geo-awareness and eight times more traffic capacity perfectly balanced across all servers. No architecture changes. No app errors. No problem. Did we mention the project went live months ahead of schedule?

To learn more about these three challenges and solutions for embracing the new set of availability features in AlwaysOn, [download the whitepaper in its entirety here »](#)