StarWind Virtual SAN® Architecture

2025

StarWind Documents

Trademarks

Changes

The material in this document is for information only and is subject to change without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, StarWind Software assumes no liability resulting from errors or omissions in this document, or from the use of the information contained herein. StarWind Software reserves the right to make changes in the product design without reservation and without notification to its users.

Technical Support and Services

If you have questions about installing or using this software, check this and other documents first - you will find answers to most of your questions on the Technical Papers webpage or in StarWind Forum. If you need further assistance, please contact us .

About StarWind

StarWind is a pioneer in virtualization and a company that participated in the development of this technology from its earliest days. Now the company is among the leading vendors of software and hardware hyper-converged solutions. The company’s core product is the years-proven StarWind Virtual SAN, which allows SMB and ROBO to benefit from cost-efficient hyperconverged IT infrastructure. Having earned a reputation of reliability, StarWind created a hardware product line and is actively tapping into hyperconverged and storage appliances market. In 2016, Gartner named StarWind “Cool Vendor for Compute Platforms” following the success and popularity of StarWind HyperConverged Appliance. StarWind partners with world-known companies: Microsoft, VMware, Veeam, Intel, Dell, Mellanox, Citrix, Western Digital, etc.

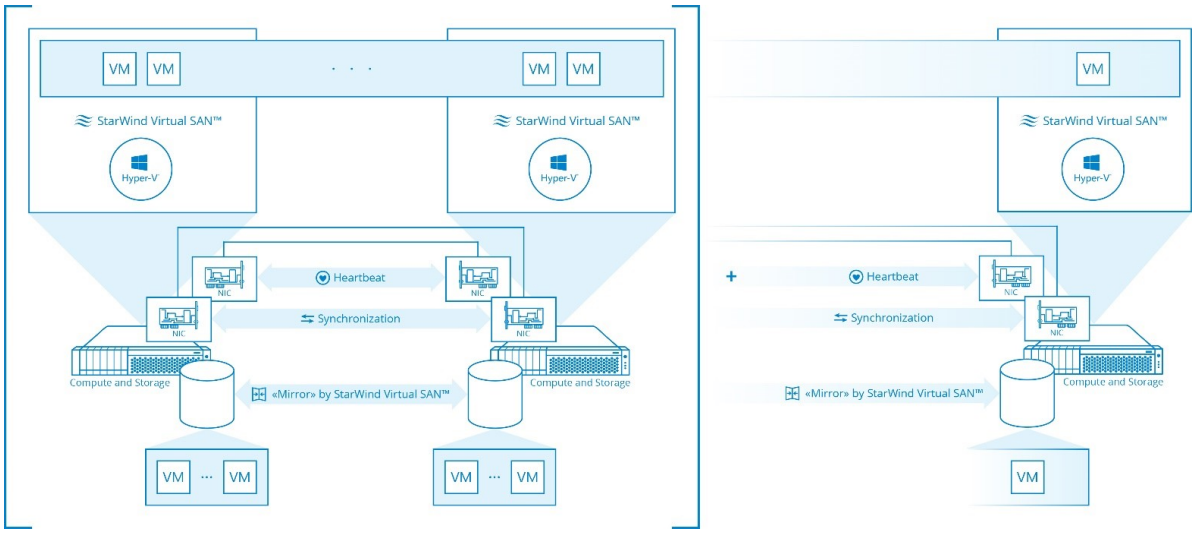
Copyright ©2009-2018 StarWind Software Inc.

1. Hyper-Converged Setup.

Both Hypervisor And Virtual Shared Storage Run On Same Physical Machine.

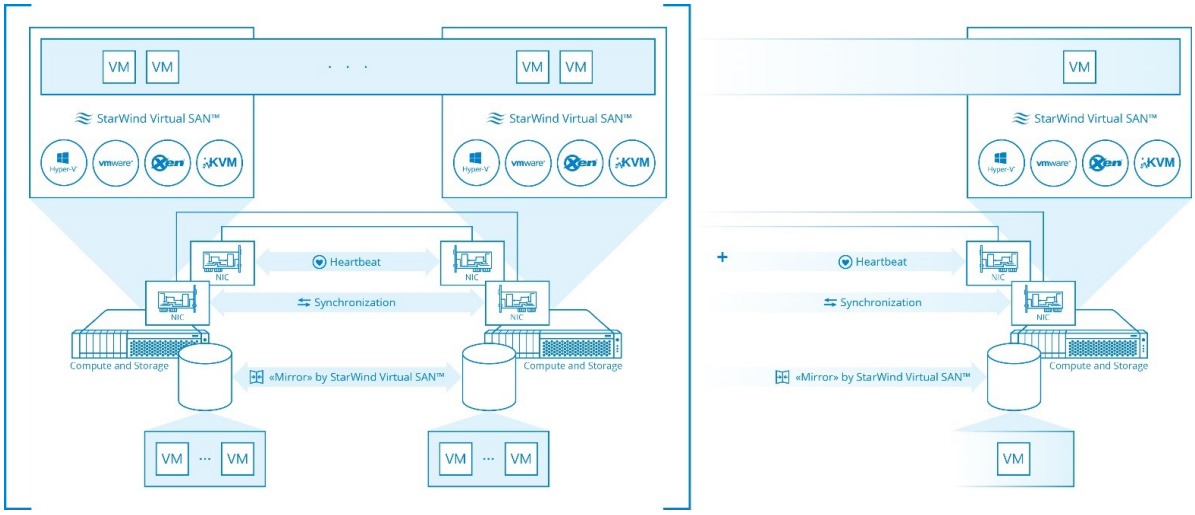
Now we have to separate Hyper-V here from all other hypervisors (VMware, Xen, and KVM) because StarWind is a native Windows application (OK, set of kernel-mode drivers, user-land services and just ordinary apps + some PowerShell scripts for automation, of course), and because of that StarWind does run inside hypervisor kernel when running Hyper-V, and it does run inside guest VM when running under all other hypervisors mentioned (non-Hyper-V). Makes A LOT of difference! I'm not going to turn this into any sort of "holy war" for "kernel VS VM virtual storage" or give away any FUD I'll just stick with pure traceable facts.

a. **Hyper-V.** StarWind runs inside the "parent partition" (not to be confused with a partition on the disk, it's a primary VM where actual Hyper-V kernel runs. Sidetrack: Microsoft was always clumsy in picking up right names for what they do (LOL) and exposes iSCSI connection to Hyper-V. But it's not a regular iSCSI like you would see with some other implementations (if you manage to find a proper Windows implementation) in loopback. Pretty much like with SMB Direct, where connection starts life as TCP and then goes RDMA, we do not route iSCSI traffic over TCP in loopback. Neither have we used most of storage drivers below us for disk I/O nor built-in system RAM cache from Cache and Memory Manager. With us iSCSI connection only starts life as iSCSI and "looks" like iSCSI to Microsoft iSCSI initiator. But because of special "past path" drivers and some design mods we applied on the service side we don't route all data over TCP and most storage drivers. Like I've already mentioned - they are simply bypassed! So, basically, in a loopback configuration we do DMA (Direct Memory Access) and zero-copy from and to L1 cache (dynamic RAM for us)



- This is pretty unique and I don't know anybody who's doing that with Hyper-V. Further step would be moving all components into the kernel mode but it's not going to help with stability: BSOD will take whole system offline and not just the storage VM like in case of some other designs. VMware Virtual SAN does have very similar design except they do all-kernel, and because they have control over the hypervisor they don't use system-wide standard APIs for storage and basically "inject" storage right into the VMs. Very smart approach if done properly! If I were Microsoft I'd do it that way J. EMC ScaleIO is also native to Windows, but they keep a whole "cake" by their own, so where we do network-wide iSCSI, they simulate virtual Fibre Channel just on the machine where their driver runs. Virtual FC driver "talks" to their user-land services using proprietary protocol. **Tiny remark:** we can run inside a guest Hyper-V VM as well and expose iSCSI or even SMB 3.0 connections to Hyper-V (Microsoft doesn't support a kludge like feeding SMB3 in loopback, but since it's a VM now there's no more issue with running SMB 3.0 in loopback). We just prefer not to do that as the performance is not that great as well as stability. . - This is exactly what Nutanix uses when running on Hyper-V: SMB 3.0 in loopback fed from inside a guest VM. We do have pre-built VM appliances for Hyper-V, but those are only for easy and fast demo deployments. Performance is not expected to be evaluated this way!

b. **VMware, Xen and KVM.** Here we are not much different from other guys, and you can do either iSCSI (like HP VSA does, for example, another truly mature product to consider BTW) or NFS (again Nutanix). We do prefer iSCSI here since in our particular case we layer NFS services on top of iSCSI so iSCSI is both faster and easier to configure (NFS configuration part is simply skipped). I repeat – faster only because we designed the system this way! It's not fair to say NFS is slower than iSCSI or anything like that. Properly done native NFS and iSCSI implementations should be nearly the same in terms of performance (NFS is still preferred because of more efficient and better scalable file locks rather than LUN locks).



- HP VSA, Nutanix, ScaleIO, and others who combine some optional in-kernel components and in-guest-VM storage services are working this way. Native implementations like VMware Virtual SAN are definitely superior. However as we target SMB space here where thousands of VMs are rare on a single LUN, we consider our approach "good enough" and mature because we're doing this for many-many years. (OR we've been doing this…)

2. Compute And Storage Separated:

Hypervisor And Virtual Shared Storage Run On Separated Physical Machines.

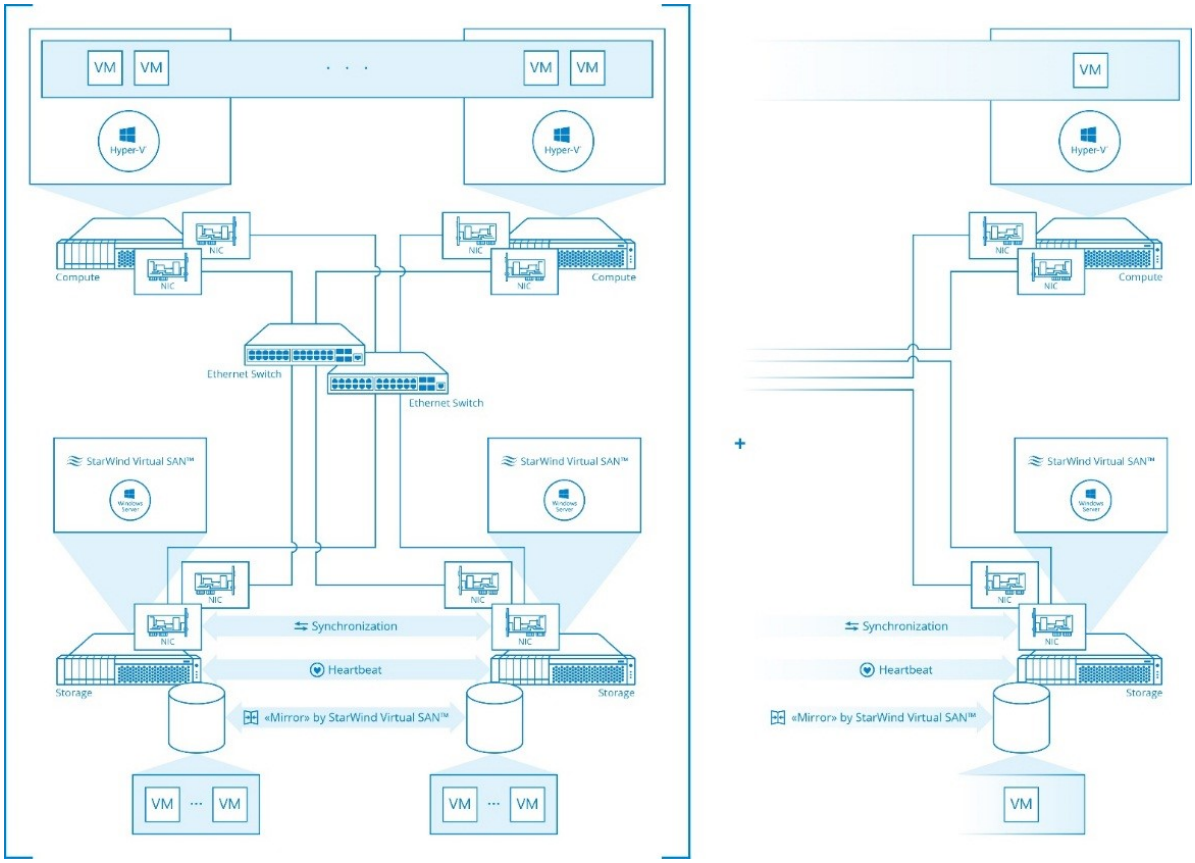
Again, we need to separate Hyper-V from other hypervisors (VMware, Xen and KVM) here. StarWind runs on Windows (surprise!), and this fact opens us a lot of opportunities, closed to other guys.

a. **Hyper-V.** So, when doing this scenario, we basically create a Scale-Out File Server with the help of StarWind on a set of different servers. StarWind "powers and fuels" private SoFS cluster CSVs (Cluster Shared Volumes), which in turn power the SMB 3.0 shares, where guest VM VHD / VHDX files are stored. Any Hyper-V client external to the SoFS cluster uses SMB 3.0 protocol to "talk" to this Scale-Out File Server. In a nutshell: Microsoft SMB 3.0 client from Hyper-V host "talks" to Microsoft SMB 3.0 server host. This allows us to have both:

   - Much better performance as all the 'SMB 3.0'-related features like SMB Direct

(RDMA), SMB Encryption, persistent handles, and SMB Multichannel are fully functional (this is very different from all major third-party SMB 3.0 server implementations where, e.g., RDMA is completely missing).

- We have 100% compatibility as Microsoft “talks” to Microsoft, StarWind is completely hidden behind the scenes, where it does internal storage replication housekeeping and distributed RAM write cache.



- We’re kind of special here. Other guys are mostly Linux open-source based, so they are typically stuck with Samba and what it can (or cannot) do. Very few do actually contribute to independent SMB 3.0 servers. And, as Microsoft does not support combining Hyper-V (virtualization) and Scale-Out File Server roles on the same hardware, it’s physically no way other guest-VM based storage virtualization solutions can “imitate” what we do. They do need virtualization and SoFS are bare metal only. The only other implementation capable of doing “kind of” what we do is EMC ScaleIO, but they require a minimum of 3 and reasonably 6-7 nodes to aggregate to some reasonable bandwidth.
Tiny remark: we can absolutely do iSCSI here as well. However, in context of Hyper-V iSCSI, compared to SMB 3.0, is the second choice because:

- Unlike VMware, who has properly implemented cluster-aware file system called VMFS, Microsoft is still using CSVFS, which is a combination of block access for data on top of NTFS (or ReFS) and SMB redirector access for metadata and some

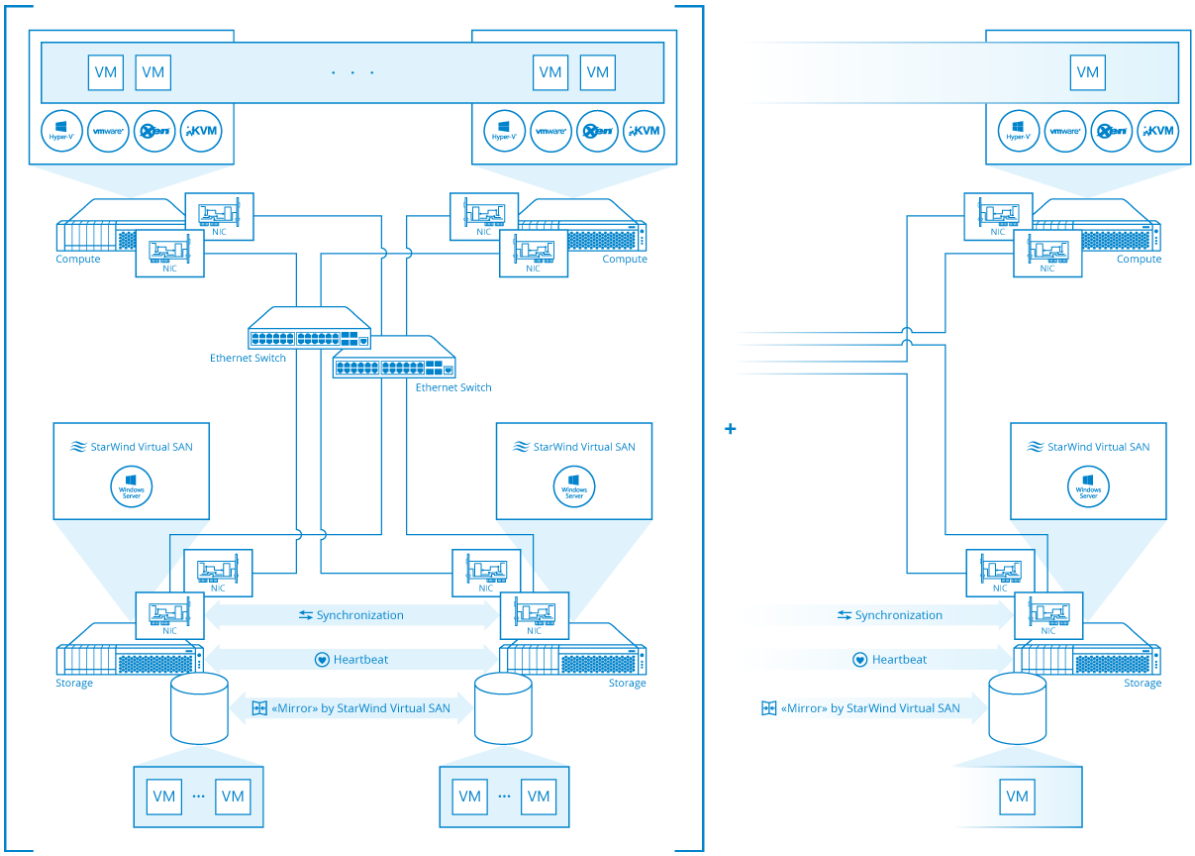
redirected mode when access is required, but ownership cannot be granted in a reasonable time term. This makes Microsoft much less effective for placing VMs right on CSV where LUN locks are used. Makes sense to have file as part of SMB 3.0

- With proper equipment SMB 3.0 will do RDMA, and iSCSI needs iSER or SRP, both not implemented by Microsoft iSCSI initiator (even worse, it looks like Microsoft somehow convinced Mellanox to take away SRP support from their Mellanox drivers, so while there was SRP for Windows Server 2008 R2, we don't have it for Windows Server 2012 R2 and Windows Server 2016).
- Microsoft considers iSCSI "old school" (guess because their own iSCSI implementation does not even do HA and caching out of box) and puts all bets on SMB 3.0. Questionable bet as iSCSI is very common and SMB 3.0 is "Microsoft only". Makes Microsoft own storage activities pretty much isolated for anybody outside Microsoft hypervisors.

Small benefit: as we don't use Windows built-in clustering services, it's even possible to build a StarWind HA iSCSI cluster using StarWind on desktop Windows editions J. And, we'll also do ODX for iSCSI when setup is done this way. As a result, more CPU cycles and network bandwidth off-loaded from your hypervisor hosts.

b. **VMware, Xen and KVM.** Here iSCSI is a preferred way to go. All these hypervisors run proper clustered file systems, so LUN locks are OK. Also, we'll do VAAI for VMware to give CPU and network some air on the hypervisor hosts (like ODX helps Hyper-V).

- This is considered a production setup, and we do have it on HCLs for all referenced hypervisors, including Hyper-V, and except for KVM, where we didn't certify our storage just yet.



Remark: Technically, we can do NFS here, and that's exactly what we do in our free version targeting VMware, Xen and KVM. But because we spawn not the best TBH Microsoft (licensed from University of Michigan) NFS server, and we layer this NFS services on top of iSCSI, neither performance nor features are matching direct iSCSI configuration done with just StarWind. However, performance and features are "good enough", and NFS is a bit more difficult to configure in a short run, but easier to use in a long run, that's why we see this one as an excellent free version offering, targeting mostly labs and other similar use cases (virtualization and lightly used NFS file servers).

Contacts

US Headquarters	EMEA and APAC
+1 617 829 44 95	+44 2037 691 857 (United Kingdom)
+1 617 507 58 45	+49 800 100 68 26 (Germany)
+1 866 790 26 46	+34 629 03 07 17 (Spain and Portugal)
	+33 788 60 30 06 (France)

Customer Support Portal: https://www.starwind.com/support

Support Forum: https://www.starwind.com/forums

Sales: sales@starwind.com

General Information: info@starwind.com

StarWind

StarWind Software, Inc. 100 Cummings Center Suite 224-C Beverly MA 01915, USA
www.starwind.com