



How to create a SCSI multihost VAX cluster

building a Mixed Interconnect cluster using the CHARON-VAX emulator
and StorCase InfoStation shared SCSI storage

1. Background and purpose

VAX clusters have appeared using a number of transports: CI, NI and DSSI. By the time SCSI storage became mainstream, the VAX platform had been replaced by Alpha in most situations. Alpha OpenVMS clusters using shared SCSI storage have been introduced into the OpenVMS operating system as far back as 1995, but were never widely used. For low end two or three node VAX clusters, DSSI was the prominent storage and clustering technology.

With the advent of CHARON-VAX as a production class VAX replacement technology, and the availability of modern shared SCSI storage solutions by StorCase, this type of VAX cluster is again a viable solution for a certain class of VAX application systems. This is the case where migration to Alpha OpenVMS, Unix or Windows applications is not technically possible or not economically feasible. The solution allows for replacing the existing VAX hardware with modern mainstream hardware, while allowing all existing applications to run unmodified. The remainder of this document is intended to be read by a technical audience with an understanding of SCSI technology, OpenVMS clustering, CHARON-VAX software and Windows in general.

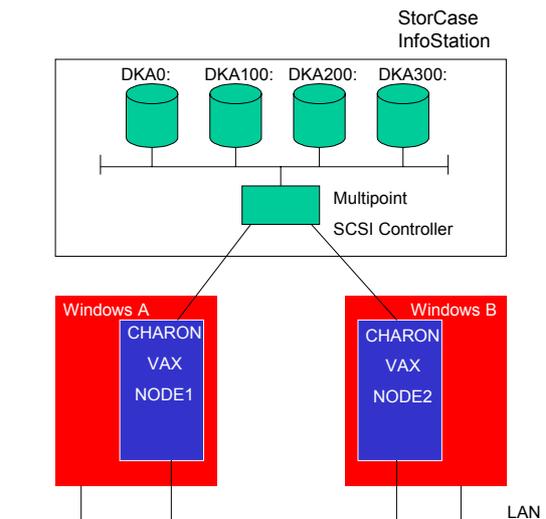
2. Description of the components used

The VAX cluster solution is built using two general purpose Windows / Intel based server systems, as prescribed by the CHARON-VAX emulator. Each system has a single 2.6 GHZ Intel Pentium IV processor w/ hyperthreading enabled, 1 GB of internal memory, two network interfaces and a local disk that holds the Windows operating system and the CHARON-VAX application. Each system also contains a Ultra160 SCSI-controller. The network interfaces are interconnected using a hub or switch.

The shared storage is contained in a Storcase InfoStation II 4-bay desktop pedestal. The InfoStation shared I/O module design enables sharing of the same SCSI bus by two or more host systems. In our setup, this box holds 4 70GB drives and is connected to each host systems SCSI controller. The same storage solution is also available in 9 bay and 14 bay rack-mount systems. The 9 bay system has been tested as well and is considered equivalent for our purposes.

The resulting cluster delivers approx. 100 VUPs.

Always power up the SCSI pedestal first. This allows the system to autoconfigure and makes the drives available to the systems as soon as they boot up.



3. Preparing your Windows installation

Install W2000 SP4 or WXP SP1 on both systems. Install the CHARON-VAX version suitable for clustering as per the installation instructions. Optionally install a terminal emulator for use with CHARON-VAX. We used Reflection from WRQ Systems. The built in Hyperterm on Windows will work as well, although keyboard mapping of the emulated VT100 terminal is not optimal.

Make sure that the network interfaces are properly configured. One interface is used for the Windows TCP/IP stack, and the other interface is used for the OpenVMS network facilities. The OpenVMS interface is not configured for TCP/IP in Windows, but holds only the CHARON-VAX NDIS5 packet driver. It is practical to rename the Windows network connections to "OpenVMS Network" and "Windows Network" on each system.

The shared SCSI storage is usually automatically recognized by Windows. The general idea is to use this to make a record of the SCSI configuration details when the system first boots, and then make sure that Windows does not use the SCSI drives / devices at all after that.

The easiest way to do this is to look up the SCSI port number, bus number and target ID of all shared drives in the Windows registry (hive key: HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\Scsi). After doing this, disable the disk devices in device manager. They will appear in device manager from now on crossed out with a red cross.

Set up this way, no Windows driver will interfere with the OpenVMS I/O for these drives. CHARON-VAX does not have its own disk drivers, nor uses Windows drivers in this case (it would use Windows drivers if it uses emulated disk container files). It maps hardware. In this case the VAX/VMS drivers directly talk to the SCSI hardware, the only role of CHARON-VAX being the SCSI address translation from the emulated SCSI ID for the VAX to the physical SCSI ID of the actual SCSI drive.

Finally, integrate the Windows systems into your computing environment in the usual way.

4. Preparing the CHARON-VAX installation

Create the CHARON-VAX configuration file as per the installation handbook. For the shared SCSI devices, the goal is to make the device names as seen from the VAX boot prompt identical on both systems. They will have identical device names in OpenVMS as well.

For each shared drive, load the device driver in the emulator, using the string `CONTAINER="\\.\SCSIx:y:z"` to specify the drive, where x is the SCSI port number from the Windows registry, y is the SCSI bus number and z is the SCSI target ID.

If the shared SCSI drives are the only devices used by OpenVMS, the two CHARON-VAX configuration files are exactly identical.

To test this setup, start both instances of CHARON-VAX, one on each system, and issue the command "SHOW SCSI" at the VAX boot prompt. Make sure that the responses to this command are identical, or, if they are different, that you can explain the differences. Note that each VAX sees only the SCSI disk devices, not the other VAX as a SCSI host system.

**BE ABSOLUTELY SURE NOT TO BOOT BOTH VAXES AT THE SAME TIME IN THIS STATE.
THE DISKS ARE SHARED BUT NO VAX CLUSTER IS PRESENTLY CONFIGURED TO COORDINATE THE I/O.**

5. Booting the first VAX

Boot your VAX of choice. It is safest to stop the CHARON-VAX emulator on the other system until the VAX cluster is prepared properly in a later stage.

At this point, it may be necessary to load the OpenVMS operating system from your OpenVMS distribution CD, and install OpenVMS in the normal way. Proceed to configure OpenVMS and any network components (DECnet, LAT, TCPIP) you require. Load your licenses, do not forget the VAX cluster license and licenses for any layered software you require.

Make sure you see the shared drives in OpenVMS as DKA0:, DKA100:, DKA200: etc. as resulting from your CHARON-VAX configuration file parameters.

Finally, tell this first VAX that it is to be part of a cluster with a single shared system disk. This involves running CLUSTER_CONFIG, editing MODPARAMS.DAT, running AUTOGEN and does not differ any way from regular VAX cluster setup. For our two-node cluster, we use a quorum disk, that has to be visible from both systems during boot. Choose one of the shared SCSI devices as a quorum disk, and give it one vote. Each of the OpenVMS systems is to have one vote as well. You may reconfigure and reboot from this first system disk until it satisfies your requirements of a valid cluster system disk. You will be booting from system root SYS0 usually.

6. Creating the cluster disk

Now it is time to add the second cluster node. Usually this is done via the "Add Node" option of CLUSTER_CONFIG.COM, but THIS OPTION IS NOT VALID HERE. CLUSTER_CONFIG needs to think you are building a DSSI cluster, but its internal logic cannot determine this from the hardware environment it sees, and it gracefully exits.

So, proceed to do a manual add of the second node. You can use the "Create Root" option of CLUSTER_CONFIG to create a second system disk root, usually SYS1. Then copy MODPARAMS.DAT from [SYS0.SYSEXEXE] to [SYS1.SYSEXEXE].

Modify the newly copied MODPARAMS.DAT to alter the SCSSNODE and SCSSYSTEMID parameters. Do not alter the ALLOCLASS parameter. The allocation class is to be the same on both systems. This MODPARAMS.DAT will of course not be used the first time you boot your second VAX into the new root, but it is so easy to forget that it's best to do it now.

Finally, shut down the VAX and stop the CHARON-VAX emulator. It is now time to boot the second VAX for the first time.

7. Booting the second VAX and forming the cluster

Start CHARON-VAX on the second system and allow the system to proceed until the boot prompt >>> is displayed. Use SHOW SCSI again to verify that your shared drives are all visible.

Use SET BOOT to record the boot device (if not the default DKA0:) into the CHARON-VAX boot rom, but more important, use SET BFLG 10000000 to specify that this system should regularly boot into system root SYS1.

For the first boot, use a conversational boot (R5:10000001) to halt at the SYSBOOT> prompt. Use SYSBOOT to alter the SCSNODE and SCSSYSTEMID parameters as specified above in the new MODPARAMS.DAT, and CONTINUE.

Now boot the VAX (the other VAX is still not running), and verify that OpenVMS starts up in root SYS1. Finish setting up the system, at least running AUTOGEN once, until you are satisfied that it is identical in functionality to the system that booted from root SYS0 before.

Only now is the time to boot the other VAX and form a cluster. If all goes well, the cluster will form and you will have a two node, quorum disk enhanced cluster. All disk IO will be handled by each systems SCSI controller and the shared disks, and all distributed lock manager traffic will use the OpenVMS Ethernet device. (By the way, that Ethernet device may run at 100Mbit/s on recent versions of CHARON-VAX)

Appendix A of the HP OpenVMS Cluster Configuration Guide, which is available at http://h71000.www7.hp.com/doc/72final/6318/6318pro_contents.html contains configuration guidelines for shared SCSI clusters on Alpha OpenVMS. We strongly urge you stick to those rules when forming a SCSI VAX cluster as well.

8. Verifying your cluster setup

So, how is this cluster different from a traditional SCSI cluster with no shared SCSI disks? These have been around for a while, and used the 10 Mbit/s VAX Ethernet interface for both disk IO and distributed lock manager traffic. Each OpenVMS host served its own SCSI disks to the cluster. This cluster configuration had its limitations, both in performance and in functionality.

“Our” cluster is different. All disks are shared on the SCSI bus, and disk I/O is handled by the local SCSI controller on each system. This is best seen by issuing a `$ SHOW DEVICE 1DKA100:` (for example) on each of the two OpenVMS systems. So, we are looking at the same disk from the viewpoint of each of the two OpenVMS systems. From host NODE1, this disk appears to be served by NODE1. The same command issued from NODE2, shows us this same physical disk is served by NODE2. This is the determining characteristic of a properly formed shared SCSI-cluster. It is very much like a DSSI-cluster, but with a different storage technology.

For comparison, in a traditional NI cluster with local SCSI disks, the systems have a different allocation class, and each node serves its own disk to the cluster. So, the `$SHOW DEVICE` for a given disk always shows it as served By the same node, whether the command is issued on NODE1 or on NODE2.

That's all, now add your applications and off you go...

VX Company, October 2003, V1.3
Wilm Boerhout, e-mail wboerhout@vxcompany.com

Related links:

<http://www.vxcompany.com/>
<http://www.vaxemulator.com/>
<http://www.emulatorsinternational.com/>
<http://www.storcase.com/>

CHARON-VAX, OpenVMS, StorCase, VAX, Windows are registered trademarks.