AN-45 DSSI emulation in CHARON-VAX/XM/XK/XL for Windows

Author: Vadim Model Date: 25-Feb-2007

Scope

These notes cover the implementation of the virtual DSSI storage subsystem for CHARON-VAX/XM/XK/XL for Windows. For this field test, the DSSI emulation is implemented as additional DLLs to be used with a CHARON-VAX product installation. The use of VAX/VMS as operating system is assumed in these notes.

The DSSI storage subsystem for the VAX 4000 Models 106 and 108 emulators is based on the emulation of SHAC host adapters and the ability to route SCS cluster information between the emulated SHAC host adapters of multiple nodes via separate TCP/IP links.

Limitations

This first implementation was tested with CHARON-VAX/XL product version 1.1.51 and can only be used on a 32 bits Windows host system. The DSSI storage subsystem is functionally emulated, but physically incompatible with DSSI and operates at a much higher throughput than the original hardware. Connection to physical DSSI hardware is neither possible nor planned for future releases.

This version of DSSI emulation for CHARON-VAX/XL for Windows supports up to 3 VAX nodes in a virtual DSSI cluster and handles a maximum cluster size of 8 nodes. A single virtual DSSI network supports up to 256 storage elements.

Field test versions of DSSI emulation should not be used for production. DSSI functionality can be removed from a production system that is temporary used for field test by simply removing the DSSI components from the configuration file.

CHARON-VAX/XL DSSI configuration overview

To use a ***single*** CHARON-VAX/XL system with DSSI emulation, either or both of the two elements must be configured:

- A DSSI storage element (disk or tape).
- A DSSI storage controller. In this field test version an emulated HSD50 storage controller is provided. The emulated HSD50 supports physical host drives, CDRom drives, physical tapes, removable disks, virtual disks and tapes.

To create a ***cluster*** of DSSI interconnected CHARON-VAX/XL systems, the DSSI hardware topology is emulated by establishing TCP/IP channels between the emulated SHAC host adapters of each CHARON-VAX/XL system (The use of TCP/IP for the interconnects makes the cluster in principle routable in a WAN). The emulated HSD50 storage controllers are then connected to every SHAC host adapter in the virtual DSSI network.

Cluster operation requires (virtual) disks that are simultaneously accessible by all CHARON-VAX nodes involved. This can be implemented for instance by using a properly configured iSCSI initiator / target structure or a fiber channel storage back-end. For test purposes Windows remote shares can be used, but these do not offer the reliability required for a VAX cluster. Disks on a multiport SCSI switch are not acceptable, as a SCSI switch does not provides true simultaneous access to multiple nodes.

Note that the emulated DSSI subsystem has many configurable parameters when multiple nodes on a single DSSI bus are to be configured. Incorrect configuration, in particular non-identical specification of emulated HSD50 disks in the DSSI nodes are likely to case data corruption. It is advisable to start any field test with implementing the single node and cluster examples that are provided in these notes.

CHARON-VAX/XL DSSI configuration parameters

To connect an emulated VAX 4000 to a virtual DSSI network, the CHARON-VAX configuration file must load at least one emulated CIXCD host adapter.

Note that VAX/VMS running on an emulated VAX 4000 node enumerates the emulated SHAC host adapters, and assigns them the VMS internal names PAA and PAB. It is recommended for clarity to keep the same naming scheme for emulated SHAC host adapters in the CHARON-VAX configuration file. The two built-in SHAC host adapters are pre-loaded with names PAA and PAB corresponding to VMS names.

The SHAC emulation has the following configuration parameters:

Parameter	**Description**
port[N] N=0…7	An integer value that specifies the TCP/IP port number at which the emulated SHAC host adapter listens for connections from another emulated SHAC host adapter with DSSI address N. Possible values are from 1024 through 32767. Initially not set.
host[N] N=0…7	A string value that specifies the TCP/IP host name (and optionally the TCP/IP port number) to connect to another emulated SHAC host adapter with DSSI address N. The syntax for the string is “host-name[:port-no]”, with possible values for port-no in the range from 1024 through 32767. Initially not set.
scs_node_name[N] N=0…7	A string value that specifies the SCSNODENAME of the emulated storage element. The string is up to 8 characters long. Possible characters are uppercase letters A through Z , figures 0 through 9. Initially set to an arbitrary value that is guaranteed to be unique within the running emulated VAX 4000 node.
scs_system_id[N] N=0…7	An integer value that specifies the SCSSYSTEMID of the emulated storage element. Initially set to an arbitrary value that is guaranteed to be unique within the running emulated VAX 4000 node.
mscp_allocation_class [N] N=0…7	An integer value which specified the ALLOCLASS of the emulated storage element. Possible values are from 0 through 255.

	Initially set to 0.
container[N] N=0…7	A string value that specifies the container of the storage element with DSSI node id N and MSCP unit number N. This storage element might be either a (virtual) disk or tape. In VMS running on an emulated VAX 4000 node, these storage elements appear as DSSI disks (DIA*N*:) or DSSI (TF86) tapes (MIA*N*:). Possible values of the parameter are strings in one of the following forms: 1. “\\.\PhysicalDrive”X, where X is 0, 1, … for accessing non-removable physical disk drives; 2. “\\.\CdRom”X, where X is 0, 1, … for accessing compact disk drives (read-only); 3. “\\.\Tape”X, where X is 0, 1, … for accessing physical tape drives; 4. “\\.\”L, where L is A, B, … for accessing removable disk drives (floppies, some optical media, …); 5. [<path-name>”\”]<file-name>[“.vdisk”] for accessing CHARON disk images, or CHARON virtual disks; 6. [<path-name>”\”]<file-name>[“.dsk”] for accessing CHARON disk images, or CHARON virtual disks; 7. [<path-name>”\”]<file-name>“.vtape” for accessing CHARON tape images, or CHARON virtual tapes; 8. [<path-name>”\”]<file-name>“.mtd” for accessing CHARON tape images, or CHARON virtual tapes; This parameter is initially not set, thus creating NO storage elements on the bus with corresponding DSSI node id.
geometry[N] N=0…7	This string value with a special format specifies the explicit geometry of the disk storage element with DSSI node id N and MSCP unit number N. This parameter is not applicable to tape storage elements. The string format is <X>”/”<Y>[“/”<Z>] where • X is number of sectors on track; • Y is number of tracks on cylinder; • Z (optional) is the number of cylinders on the unit. If omitted, Z is calculated based on X, Y and the total number of sectors on the unit that reflects the size of the disk storage element; If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.

These parameters can be specified in a SET command.

CHARON-VAX application note

Connecting storage controller to a virtual DSSI network

To connect storage controller to virtual DSSI network, the CHARON-VAX configuration file must load at least one emulated HSD50 storage controller. In most cases one emulated HSD50 storage controller per virtual DSSI network is enough.
When loading an instance of emulated HSD50 storage controller the CHARON-VAX configuration file must supply a unique reference name for that instance. While this name is only valid within the configuration file, it is recommended for clarity that the VMS SCSNODENAME is used as instance name.

The line below loads an emulated HSD50 storage controller and assigns it the instance name MYDISKS, and connects it to the primary built-in DSSI controller:

```
load HSD50 MYDISKS dssi_host=PAA
```

The HSD50 emulation has the following configuration parameters (as per CHARON configuration file syntax, all integer values can be entered in Hex, Decimal or Octal form):

Parameter	Description
dssi_host_name	A string value which specifies the instance name of emulated CIXCD host adapter serving the same virtual CI network. If this value is not set, CHARON-VAX will try to locate the host adapter automatically. This automatic lookup works only if the CHARON-VAX configuration has exactly one instance of an emulated CIXCD host adapter.
dssi_node_id	An integer value which specifies the address of an emulated HSD50 storage controller on the virtual DSSI network. Possible values are from 0 through 7 (initially set to 0).
scs_node_name	A string value that specifies the SCSNODENAME of the emulated HSD50 storage controller. The string is up to 8 characters long. Possible characters are uppercase letters A through Z , figures 0 through 9. Initially set to the name of the emulated HSD50 controller.
scs_system_id	An integer value that specifies the SCSSYSTEMID of the emulated HSD50 storage controller. Initially set to an arbitrary value that is guaranteed to be unique within the running emulated VAX 4000 node.
mscp_allocation_class	An integer value which specified the ALLOCLASS of the emulated HSD50 storage controller. Possible values are from 0 through 255 (initially set to 0).
container[N] N=0…255	A string value that specifies the container of the storage element with MSCP unit number N. This storage element might be either a (virtual) disk or tape. In VMS running on an emulated VAX 4000 node, these storage elements appear as HSX00 disks (DUA*N*:) or HST00 tapes (MUA*N*:). Possible values of the parameter are strings in one of the following

	forms: 9. “\\.\PhysicalDrive”X, where X is 0, 1, … for accessing non-removable physical disk drives; 10. “\\.\CdRom”X, where X is 0, 1, … for accessing compact disk drives (read-only); 11. “\\.\Tape”X, where X is 0, 1, … for accessing physical tape drives; 12. “\\.\”L, where L is A, B, … for accessing removable disk drives (floppies, some optical media, …); 13. [<path-name>”\”]<file-name>[“.vdisk”] for accessing CHARON disk images, or CHARON virtual disks; 14. [<path-name>”\”]<file-name>[“.dsk”] for accessing CHARON disk images, or CHARON virtual disks; 15. [<path-name>”\”]<file-name>“.vtape” for accessing CHARON tape images, or CHARON virtual tapes; 16. [<path-name>”\”]<file-name>“.mtd” for accessing CHARON tape images, or CHARON virtual tapes; This parameter is initially not set, thus creating NO storage elements on the controller.
geometry[N] N=0…255	This string value with a special format specifies the explicit geometry of the disk storage element with MSCP unit number N. This parameter should not be used for tape storage elements. The string format is <X>”/”<Y>[“/”<Z>] where • X is number of sectors on track; • Y is number of tracks on cylinder; • Z (optional) is the number of cylinders on the unit. If omitted, Z is calculated based on X, Y and the total number of sectors on the unit that reflects the size of the disk storage element; If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.

The following example configures an emulated HSD50 storage controller with a non-default DSSI network address of 5:

```
load HSD50 YOURDISK dssi_host=PAA dssi_node_id=5
```

Configuration example 1.

One VAX system and 2 disks

The configuration file below creates an emulated VAX 4000 Model 108 node with two DSSI disks.

```
set session hw_model="VAX_4000_Model_108"
set session log="example1.log"
set toy container=" example1.dat"
set rom container=" example1.rom"

load virtual_serial_line OPA0 port=10003 application="opa0.ht"

set PAA container[0]="C:\mydisks\dia0-rz24-vms-v6.2.vdisk"
set PAA container[1]="C:\mydisks\dia1-rz24-vms-v6.2.vdisk"
```

The emulated VAX 4000 can then boot VMS with the following command:

```
>>> BOOT DIA0
```

After logging into VMS, the “SHOW DEVICE” command displays the following:

```
$ sho dev d

Device               Device         Error   Volume        Free  Trans Mnt
 Name                Status         Count    Label       Blocks Count Cnt
20004200$DIA0:       Mounted            0 DSSI01          32022  147    1
20004201$DIA1:       Online             0
$ _
```

Configuration example 2.

One VAX, 2 disks, and a tape drive on a separate HSD50 controllers

The configuration file below emulates a VAX 4000 Model 108 node, one HSD50 storage controller serving two disks and another instance of a HSD50 that serves a tape drive to the VAX over a virtual DSSI:

```
set session hw_model="VAX_4000_Model_108"
set session log="example2.log"
set toy container="example2.dat"
set rom container="example2.rom"

load virtual_serial_line OPA0 port=10003 application="opa0.ht"

load HSD50 MYDISKS dssi_host=PAA dssi_node_id=1
set MYDISKS container[0]="C:\mydisks\dua0-rz24-vms-v6.2.vdisk"
set MYDISKS container[1]="C:\mydisks\dua1-rz24-vms-v6.2.vdisk"

load HSD50 MYTAPE dssi_host=PAA dssi_node_id=2
set MYTAPE container[3]="\\.\Tape0"
```

In this example we emulate two HSD50 instances. Since they are both connected to the same virtual DSSI bus, we must assign them different DSSI node id values.

The emulated VAX 4000 Model 108 can then boot VMS with the following command:

```
>>> BOOT DUA0
```

After logging into VMS, the “SHOW DEVICE” command displays the following:

```
$ sho dev d

Device                 Device           Error    Volume         Free  Trans Mnt
 Name                  Status           Count     Label        Blocks Count Cnt
MYDISKS$DUA0:          Mounted              0  DSSI01           31932   147   1
MYDISKS$DUA1:          Online               0
$ sho dev m

Device                 Device           Error    Volume         Free  Trans Mnt
 Name                  Status           Count     Label        Blocks Count Cnt
MYTAPE$MUA3:           Online               0
$ _
```

Virtual DSSI cluster configurations for the VAX 4000 emulator

The hardware DSSI bus establishes private node-to-node communication channels between any two DSSI nodes that are part of the same DSSI bus. In this context, the DSSI storage controllers are also nodes like VAX computers, but are running special software.

The CHARON DSSI emulation replicates this structure by establishing TCP/IP channels between each pair of emulated SHAC host adapters, and by connecting the emulated HSD50 storage controllers to every SHAC on the same virtual DSSI network.

Consider a pair of emulated VAX 4000 nodes: **VAX_A** and **VAX_B** running on host system **HOST_A** and **HOST_B** respectively.

The two emulated VAX nodes have their own emulated SHAC host adapters interconnected with the virtual DSSI bus. In order to communicate properly, these two adapters must have different DSSI node id’s (DSSI_NODE_ID_A and DSSI_NODE_ID_B), which must be explicitly specified in the corresponding VAX consoles.

Next, every emulated SHAC host adapter participating in the virtual DSSI cluster must be able to communicate to every other emulated SHAC host adapter in the same virtual DSSI cluster. Such communication is performed over TCP/IP links established between every two emulated SHAC host adapters.

A virtual DSSI cluster consisting of N emulated VAX 4000 nodes has $\frac{N \cdot (N-1)}{2}$ such links.

Every end of such link is identified by the pair:
{<TCP/IP-HOST-NAME>,<TCP/IP-PORT-NO>}.

Hence, on HOST_A the CHARON-VAX configuration file describes connection to HOST_B as follows:

```
set PAA port[DSSI_NODE_ID_B]=TCPIP_PORT_A_FOR_B
set PAA host[DSSI_NODE_ID_B]=”HOST_B:TCPIP_PORT_B_FOR_A”
```

And on HOST_B the CHARON-VAX configuration file describes connection to HOST_A as follows:

```
set PAA port[DSSI_NODE_ID_A]=TCPIP_PORT_B_FOR_A
set PAA host[DSSI_NODE_ID_A]=”HOST_A:TCPIP_PORT_A_FOR_B”
```

Next, every emulated SHAC host adapter must also connect to all emulated storage controllers on the same virtual DSSI bus. This is similar to the non-clustered configurations described before, except for the MSCP allocation class:

The VAX Cluster architecture requires every VAX node participating in the cluster be assigned the MSCP allocation class. The same requirement applies to DSSI storage controllers, which are also nodes on the DSSI. Therefore, when configuring an instance of an emulated HSD50 storage controller, the MSCP allocation class for that controller must be supplied through the corresponding parameter. For example, the following line:

```
load HSD50 MYDISKS dssi_host=PAA dssi_node_id=5 mscp_allocation_class=5
```

assigns the MYDISKS emulated storage controller the MSCP allocation class 5.

Configuration example 3.

A dual node VAX cluster with 4 disks

To setup two emulated VAX 4000 Model 108 nodes, we need two host machines, preferably running the same version of Windows. Assume that these host systems have Windows host names CASTOR and POLLUX in the host TCP/IP network.

The following is the CHARON-VAX configuration file for the emulated VAX 4000 Model 108 node named VAX001 running on CASTOR:

```
set session hw_model="VAX_4000_Model_108"
set session log="vax001.log"
set toy container="vax001.dat"
set rom container="vax001.rom"

load virtual_serial_line OPA0 port=10003 application="opa0.ht"

set PAA port[2]=11012 host[2]=”pollux:11021”

load HSD50 OURDISKS dssi_host=PAA dssi_node_id=1
set OURDISKS scs_system_id=3238746238 mscp_allocation_class=1
set OURDISKS container[0]="H:\ourdisks\dua0-rz24-vms-v6.2.vdisk"
set OURDISKS container[1]="H:\ourdisks\dua1-rz24-vms-v6.2.vdisk"
set OURDISKS container[2]="H:\ourdisks\dua2-rz24-vms-v6.2.vdisk"
set OURDISKS container[3]="H:\ourdisks\dua3-rz24-vms-v6.2.vdisk"
```

And the following is the CHARON-VAX configuration file for the emulated VAX 4000 Model 108 node named VAX002 running on POLLUX:

```
set session hw_model="VAX_4000_Model_108"
set session log="vax002.log"
set toy container="vax002.dat"
set rom container="vax002.rom"

load virtual_serial_line OPA0 port=10003 application="opa0.ht"

set PAA port[1]=11021 host[1]=”castor:11012”

load HSD50 OURDISKS dssi_host=PAA dssi_node_id=1
set OURDISKS scs_system_id=3238746238 mscp_allocation_class=1
set OURDISKS container[0]="H:\ourdisks\dua0-rz24-vms-v6.2.vdisk"
set OURDISKS container[1]="H:\ourdisks\dua1-rz24-vms-v6.2.vdisk"
set OURDISKS container[2]="H:\ourdisks\dua2-rz24-vms-v6.2.vdisk"
set OURDISKS container[3]="H:\ourdisks\dua3-rz24-vms-v6.2.vdisk"
```

Note that in the both configuration files, the data related to the emulated HSD50 storage controller OURDISKS must be identical. Not following this rule will very likely cause data corruption on the (virtual) disks. In this respect note the explicit specification of SCS_SYSTEM_ID for the storage controllers.

As mentioned earlier, CHARON-VAX loads an initial (default) value for the SCS_SYSTEM_ID that is guaranteed to be unique within the running emulator. But with two instances of the emulator these default values are likely to be different. Therefore, the SCS_SYSTEM_ID value must be specified explicitly for both emulated HSD50 storage controllers to obtain an identical value.

This example also assumes that drive “H” is a network share, so that all disk images are accessible from both host machines.

Configuration example 4.

A triple node VAX cluster with multiple iSCSI disks

In this example we assume that all three host systems have a iSCSI initiator (for example the Microsoft iSCSI initiator) and are connected to a common iSCSI server. The iSCSI disk server provides 8 virtual disks with R/W access on all hosts. These disks are configured as \\.\PhysicalDrive1 ... \\.\PhysicalDrive8 on each of the host machines.

Since the storage configuration must be identical on all three nodes, it is recommended to describe the storage structure in separate configuration file (to be included in each CHARON configuration file) and store it on a common Windows network share (H:):

```
load HSD50 DISKSET1 dssi_host=PAA dssi_node_id=1
set DISKSET1 scs_system_id=3238746238 mscp_allocation_class=1
set DISKSET1 container[1]="\\.\PhysicalDrive1"
set DISKSET1 container[2]="\\.\PhysicalDrive2"
set DISKSET1 container[3]="\\.\PhysicalDrive3"
set DISKSET1 container[4]="\\.\PhysicalDrive4"

load HSD50 DISKSET2 dssi_host=PAA dssi_node_id=2
set DISKSET2 scs_system_id=1256412654 mscp_allocation_class=2
set DISKSET2 container[5]="\\.\PhysicalDrive5"
set DISKSET2 container[6]="\\.\PhysicalDrive6"
set DISKSET2 container[7]="\\.\PhysicalDrive7"
set DISKSET2 container[8]="\\.\PhysicalDrive8"
```

The following is the CHARON-VAX configuration file for the emulated VAX 4000 Model 108 node named VAX001 running on HOST001:

```
set session hw_model="VAX_4000_Model_108"
set session log="vax001.log"
set toy container="vax001.dat"
set rom container="vax001.rom"

load virtual_serial_line OPA0 port=10003 application="opa0.ht"

set PAA port[2]=11012 host[2]=”host002:11021”
set PAA port[3]=11013 host[3]=”host003:11031”

include h:\ourdisks\disksets.cfg
```

The CHARON-VAX configuration file for the emulated VAX 4000 Model 108 node named VAX002 running on HOST002 is as follows:

```
set session hw_model="VAX_4000_Model_108"
set session log="vax002.log"
set toy container="vax002.dat"
set rom container="vax002.rom"

load virtual_serial_line OPA0 port=10003 application="opa0.ht"

set PAA port[1]=11021 host[1]=”host001:11012”
set PAA port[3]=11023 host[3]=”host003:11032”

include h:\ourdisks\disksets.cfg
```

And the following is the CHARON-VAX configuration file for the emulated VAX 4000 Model 108 node named VAX003 running on HOST003:

```
set session hw_model="VAX_4000_Model_108"
set session log="vax003.log"
set toy container="vax003.dat"
set rom container="vax003.rom"

load virtual_serial_line OPA0 port=10003 application="opa0.ht"

set PAA port[1]=11031 host[1]=”host001:11013”
set PAA port[2]=11032 host[2]=”host002:11023”

include h:\ourdisks\disksets.cfg
```

For further information:

Software Resources International S.A.
P.O. Box 156
CH-1228 Plan-les-Ouates
Switzerland

WWW.SOFTRESINT.COM
Email: info@softresint.com
Tel: +41 22 794 1070