Building VAXcluster Configurations with Large Numbers of Systems

Keith Parris

VMS Engineering

May 18, 1992; 1st draft

Digital Equipment Corporation
Maynard, Massachusetts

Introduction

In this article, we'll explore the issues involved in configuring a VAXcluster system with a very large number of nodes.

We'll look at:

- What is a "large" VAXcluster configuration?
- Reasons why you might want to build a big cluster
- Alternatives to large clusters: some other solutions which may meet some types of needs better than building a single larger cluster
- Recent changes which make building larger clusters easier
- Potential problem areas and solutions: an in-depth look at areas where you may need to take special care, and solutions or work-arounds to potential problems

What is a "large" VAXcluster configuration?

The VAXcluster Software Product Description (SPD) for version 5.5 contains some specific limits on cluster size:

- 96 nodes total

  The maximum number of CPUs supported in a cluster is currently 96.

- 16 CPUs per star coupler

  The maximum number of CPUs that can be connected to a Star Coupler is 16, regardless of the Star Coupler size.

- 16 FDDI CPUs with DEMFAs

  The maximum number of VAXcluster members that can be directly connected to the FDDI, via the DEC FDDIcontroller 400 (DEMFA), is 16.

Other possible definitions of "large" (other than near, at, or beyond the limits in the 5.5 SPD) might be things like:

- Over "n" VAX Units of Processor speed (VUPs)

  More than some arbitrary amount of overall CPU horsepower, relative to the VAX 11 /780

- 3 or more star couplers

  With I/O requirements so large as to require a number of star couplers

- Wide geographical extent

  Spread over a large geographical area. Fiber Distributed Data Interface (FDDI) technology allows nodes to be separated by a distance of up to 40 kilometers, or about 25 miles, using single-mode optical fiber.

But for the purposes of this article, we'll talking in the area around that number 96, although many of the same principles that we'll talk about would also apply to clusters which are "large" by other definitions.

Some of the same methods we'll talk about can be used to make *smaller* clusters work better as well.

Why build a large VAXcluster configuration?

So, why might you need to build a very large cluster?

When we talked with customesrs who were asking us to raise the 96-node limit, we found there were two main reasons for wanting to build very large clusters:

- Shared access to resources
- Easier system management

Shared access to resources

Here I'm talking about folks using a shared resource where the same user environment is required, like:

- Editing a data file on a workstation, running a job on a vector processor in the cluster or through a VAX-to-Cray gateway to do some calculations, and displaying the data in graphics back on the workstation.
- Something we tend to take for granted, like shared read-write access to common application files, with locking down to the record level.

Easier system management

Managing two clusters can be close to twice as much work as managing one cluster.

Adding a node to an existing cluster requires very little work compared with adding a standalone system; for example, you can simply connect another VMS workstation, boot it into the existing cluster, and add a user in only minutes.

Alternatives to building a large VAXcluster configuration

Not everyone needs a large cluster. Sometimes one of the following choices works out better:

- Distributed File Services (DFS) software

  You can use Distributed File Services software to allow access to files from other systems or clusters.

- DECnet (FAL) file access

  You can use the DECnet File Access Listener.

- X-terminals

  You can use X-Windows terminals instead of workstations.

- Multiple clusters

  You can break the cluster into two or more separate clusters.

Let's look at each of the alternatives in more detail.

Alternative: DFS (Distributed File Services) Software

You could use DFS layered software for access to common disks.

- Pros:
  - Large numbers of client nodes

    DFS can serve a large number of client systems
  - Clients don't count as cluster members

    Nodes accessing files using DFS do not count against the cluster node limit.
  - Good performance

    The performance of DFS is reasonably close to that of MSCP serving, with lower overhead than DECnet (FAL) file service.
- Cons:
  - Limited shared read-write capability

    DFS cannot support multiple clients simultaneously accessing the same file for read/write access.
  - More system management work

    Each of the separate systems or clusters accessing the shared data must be managed separately, increasing system management workload.

Alternative: DECnet FAL access

You might use the DECnet File Access Listener (FAL) to access shared files.

- Pros:
  - Readily available

    DECnet FAL service is available in the DECnet product, which in most clusters is already installed and available on each node.
  - Remote nodes don't count in cluster limit

    Remote nodes accessing files with DECnet FAL do not count against the cluster node limit.
  - Some read/write sharing

    Limited read/write sharing (of indexed files, for example) is supported.
- Cons:
  - Server process count

    Each file opened by each user requires a separate process on the server. The number of these processes can quickly add up.
  - Lower performance

    The relative performance of DECnet file access is a bit lower than MSCP serving within a cluster or access using DFS software.
  - More system management work

All of the separate systems or clusters accessing the data through DECnet must be individually managed, which takes more system management effort than if they were all in a single cluster.

Alternative: X-windows terminals

You might use a few big server nodes, and use X-windows terminals to display the output of applications running on the big nodes. Or, you might set up multi-user satellite nodes, where one user might work at the workstation console, while others share the same system through X-terminals.

- Pros:
  - Low price

    X-terminals typically cost less than a workstation
  - Large numbers of display stations are supported

    X-windows terminals do not count against the cluster node limit.
- Cons:
  - More network bandwidth

    Running an application on an X-terminal typically requires more network bandwidth than running the same application on a workstation.
  - More host resources

    Running an application on X-terminals consumes CPU power and network adapter capacity on a server node; it's possible that the the combination of additional server capacity plus X-terminals may cost more than equivalent capacity on workstations.
  - Performance

    User-perceived performance using an X-terminal may be different than when using a workstation, particularly for graphics-intensive applications.
  - Software support

    Some applications may not be available in an X-windows-compatible version. For example, some applications may have been written for VAX Workstation Software (VWS) and not yet converted to DECwindows.
  - Host is a single point of failure

    The process on a server node (which X calls the client) is a single point of failure. If either the node or the process dies, there is no automatic fail-over to another node.

    With a workstation, if a disk server fails, disk service can automatically and transparently switch over to another disk server, without affecting the application.

Alternative: Multiple, smaller clusters

You could choose to break the configuration into multiple, smaller clusters.

- Pros:
  - Lower node counts

    This helps keep the size of each cluster under the node limit.

- Cons:
  - More system management work

    System management workload is likely to increase in a manner proportional to the number of separate clusters that have to be managed.
  - Duplication of resources

    Separate clusters will mean that it's harder to share hardware resources like disk space, or tape drives for backups.
  - Multiple user environments

    Depending on which cluster a user logs in to, directory structures, commands, logical names, and so forth may be different. Also, the user will need to keep track of and change their passwords on all of the different systems.
  - Sharing common files is difficult

    It is harder to set up centralized files accessible to all users from all the systems, particularly for read-write shared access.

    Users may have to move between clusters to access the data they need, or copy files around between the clusters.

Alternative: Larger cluster

You could choose to build a single larger cluster.

- Pros:
  - Full read/write file sharing

    Full sharing of files for read-write access is supported within a cluster.
  - Lower system management effort

    System management workload is lower than with any of the other options we've looked at.
- Cons:
  - Requires care in configuration

    To build a single larger cluster, you will have to plan carefully. We'll be discussing various items that you'll need to think about.

Factors fostering large VAXcluster configurations

The 96-node limit first appeared in the SPD for version 5.2.

A number of factors have changed since the VMS 5.2 timeframe which make larger VAX-cluster configurations even more practical now than then.

Changes in VAXcluster Systems Since VMS 5.2

- Faster CPUs

  Faster server CPUs are now available, including the 4000-500, 6000-600, and the 9000 series.

- Faster network adapters

  Network interfaces with better performance are now available, such as the DEMNA XMI-to-Ethernet adapter and the second-generation Ethernet chip (SGEC)-based integral Ethernet adapter used in the VAX 4000 series. The DEMFA FDDI adapter provides FDDI throughput for XMI-based systems.

- Multiple network adapters per system

  Adapter bandwidth is even less of an issue, now that multiple network adapters are supported per system.

- FDDI as a VAXcluster interconnect

  Network bandwidth is also less of an problem, now that FDDI, with its 100 megabit-per-second bandwidth, is available as both a network backbone and a VAXcluster interconnect.

- DSSI for 4000 and XMI systems

  Fast Digital Small Systems Interconnect (DSSI) adapters are available for VAX 4000 and XMI-based systems, allowing them to support I/O rates previously attainable only with Hierarchical Storage Controllers (HSCs).

- Multiple CI (and DSSI) adapters per system

  VMS now supports more than one Computer Interconnect (CI) and/or DSSI adapter per system, providing additional bandwidth for disk serving.

- MSCP server load balancing

  MSCP server static load balancing now helps to spread disk serving load evenly across the available disk servers at the time of a disk mount or mount verification operation.

  The next functional release of VMS is slated to include dynamic MSCP server load balancing as well, to adjust server loading dynamically under normal operating conditions as the workload changes.

- Faster disks

  The average seek time on disks keeps getting better, and some now spin at faster rates for lower rotational latency.

- Caching

  Caching is available for the HSC, and some disks now have integral caches.

- Host-based volume shadowing

  Disk shadowing, used to increase disk availability, is no longer limited to configurations with CI & HSC hardware, but is now available across the full range of possible configurations.

- Software improvements

Improvements have been made in the VAXcluster code, reducing the impact of node addition and removal during state transitions, and balancing locking workload during normal operations.

Potential areas of concern

Potential problems/solutions for large VAXcluster Configurations

There are some common pitfalls and problems one can encounter when creating a large node-count cluster. Solutions or work-arounds are available to deal with each of them.

The following list is arranged roughly in descending order of potential severity:

- Boot times
- System parameters
- Network problems
- Hot system files
- Availability of MOP/disk servers and system disk
- Shared resource availability
- Cluster alias
- System disk space
- Limits in MOUNT/CLUSTER and the SYSMAN utility

We'll talk about each area in detail.

By the way, MOP stands for the DECnet Maintenance Operations Protocol. A MOP server responds to down-line load requests from satellite nodes. A system where both MOP service and MSCP disk serving are enabled is what we've traditionally called a boot node.

Boot Times

A large cluster needs to be carefully configured so that there is sufficient capacity to boot the desired number of nodes in a reasonable amount of time.

- System disk throughput

  According to the results of a study which were included in the VMS 5.2 Release Notes, it takes about 4,200 I/O operations to the system disk to boot a satellite node. A typical disk can only handle around 50 I/Os per second. This means each satellite is going to take a minute and a half's worth of attention from the system disk.

  Consider what happens when a hundred nodes all need to boot at once—this could induce an I/O bottleneck which might stretch the reboot time into hours.

- Network bandwidth

  An Ethernet segment can handle only a finite amount of traffic, and individual Ethernet segments and adapters could become saturated during a mass boot.

- Avoiding reboots

  One way to minimize the problem of boot times is just to avoid rebooting.

  — Power protection

Reboots due to power failures may be avoided by protecting server nodes and/or satellites with uninterruptible power supply (UPS) systems.

Some folks have concluded that it's worth paying for a UPS system for each and every workstation. It only adds about 5% to the cost of the workstation, and can easily pay for itself in saved work time.

— Careful configuration

Here's an example:

Consider a typical mixed-interconnect cluster, with:

o a few big nodes as MOP and disk servers, all connected to a star coupler, and all of which have votes, and

o a number of satellites, none of which have any votes

If a big node in the cluster has only a single network adapter, and that network adapter fails, the Rule of Total Connectivity has been broken, and the cluster has to reconfigure itself so that once again all surviving members can talk directly to all other cluster members.

The algorithm to select a subset of the nodes to continue considers votes to outweigh numbers of nodes. This means that the one node without a network connection, which has a vote, wins out over all the satellites, and the satellites will all do a CLUEXIT bugcheck and leave the cluster.

To avoid this problem, you can give one vote to a single node which is attached to the local-area network, but not to the big nodes' CI or DSSI busses. This way, if the only network adapter on one of the big nodes fails, it will be that single node and not *all* of the satellites which will do a CLUEXIT bugcheck and leave the cluster.

Another solution would be to provide redundant network adapters on the server nodes.

I've found that the Guidelines for VAXcluster System Configurations (order number EK-VAXCS-CG) is a useful resource, since it contains a lot of data on the performance, throughput, and capacity of the various pieces of hardware involved in a VAXcluster configuration.

Now let's look at the first two areas involved in boot times in greater detail.

System disk throughput

To achieve enough system disk throughput will typically require some combination of these techniques:

• Off-load work

— Move files off the system disk

Any system files where it is possible (such as SYSUAF, RIGHTSLIST, queue file(s), etc.) should be moved off the system disk.

Page and swap files should be moved off the system disk to less-busy disks.

Put layered products and applications on other disks when possible.

Moving all these files off the system disk and onto a separate disk also typically has the additional benefit of eliminating most of the write activity to the system disk, raising the read/write ratio and thereby maximizing the performance with Volume Shadowing when that is used on the system disk.

— Avoid disk rebuilds at boot time

The VMS file system maintains a cache of pre-allocated file headers and disk blocks. When a disk doesn't get properly dismounted, as happens when a system crashes, this pre-allocated space becomes temporarily unavailable. When the disk is next mounted, the MOUNT utility normally scans the disk to recover that space. This is called a disk rebuild.

User response times can be degraded while a disk rebuild operation is going on, because most I/O activity on that disk is blocked. It helps to avoid doing this during prime time, and you really don't want satellites doing this on every disk every time they reboot.

o ACP_REBLDSYSD = 0 for system disk

System disks are especially critical in this regard. I recommend setting the SYSGEN parameter ACP_REBLDSYSD to 0, at least for all the satellite nodes. This prevents a rebuild operation on the system disk when it gets mounted implicitly by VMS early during the boot process.

o MOUNT/NOREBUILD for all user disks

Startup procedures which mount user disks should include the /NOREBUILD qualifier on MOUNT commands, at least on the satellite nodes.

o SET VOLUME/REBUILD during non-prime time

Once the system is up and running, you can run a batch job or a command procedure to do a SET VOLUME/REBUILD command for each disk drive. Because the SET VOLUME/REBUILD command determines for itself whether or not a rebuild is really needed, the job can simply execute the command for every disk. This job can be run during off hours, preferably on one of the more powerful nodes.

Note: The Customer Support Center recommends that you insert a WAIT command to provide a brief delay between the SET VOLUME commands for successive disks, to allow time for the caches to get flushed and avoid a potential problem in recent VMS versions which could result in things getting hung up.

In large clusters, a lot of disk space may be pre-allocated to caches, and if many CPUs abruptly leave (during a power failure, for example), this space becomes temporarily unavailable. If you tend to run with very full disks, you might not want to disable rebuilds on the big server nodes at boot time.

• Multiple system disk spindles

Using more than one disk spindle for the system disk provides additional actuators to handle the I/O load. This can be done using the following techniques:

— Volume shadowing

To minimize the number of separate system disks which have to be maintained, spindles can be combined by means of Volume Shadowing.

— Multiple system disks

Because there is a limit on the number of spindles which can be combined into a single virtual unit using Volume Shadowing, sometimes you may need to create separate system disks (or shadow sets) for different groups of nodes.

But because system management workload goes up as separate system disks are added, and does so roughly in direct proportion to the number of separate system disks which need to be maintained, you'll want to do everything you can to minimize the number of these while still providing the required level of performance. One technique is to use:

— "Cloned" system disks

When I speak of "cloning" system disks, I mean to create a system disk (or shadow set) with roots on it for all the cluster nodes, and use this as the "master" copy, performing all software upgrades on this system disk. For the "cloned" system disks, you'd make a backup of the "master" to the other disks, then change the volume names so they'll be unique.

If you've moved system files off, great, otherwise you'd need to fix up SYLOGICALS.COM to point to system files on the "master" system disk.

Don't forget to make interim changes to both the "master" and the "clones".

Before an upgrade, be sure to save anything you need that you've changed on the "cloned" disks since the last upgrade (such as MODPARAMS.DAT and AUTOGEN feedback data, accounting files for billing, password history, and anything else you need).

Some other techniques you can use to put off having to add another separate system disk are:

- Caching

  Caching in the HSC or in RF or RZ disks can improve the effective system disk throughput.

  The next functional release of VMS is slated to include file caching capabilities, which we expect to also assist in this area.

- Solid-state disk

  A solid-state disk, such as an ESE20 or ESE50, has lower latencies and can handle a higher request rate than a regular magnetic disk. A solid-state disk can be used as a system disk or at least to hold hot system files. You can SPM, VPA, or DECps to identify hot files during a mass boot.

- RAMdisk

  DECram software provides RAMdisk capabilities for VMS systems. A RAMdisk is an area of main memory within a system which is set aside to store data, but which is accessed as if it were a disk.

  Putting copies of selected hot read-only files into a RAMdisk on MOP servers can help improve boot times.

Network bandwidth

A single Ethernet is unlikely to have sufficient bandwidth to meet the needs of a large cluster. Likewise, a single Ethernet adapter may become a bottleneck, especially for a disk server. A large VAXcluster configuration should be carefully planned in order to provide satisfactory LAN performance.

Sufficient network bandwidth can be provided using some of these techniques:

- Fast network adapters

  Select network adapters with high performance.

- Multiple Ethernet segments

  Divide the network into multiple segments, using bridges to segregate the workload. For example, a bridge could be used to to isolate traffic such that a disk server's network adapter and the satellites it serves are on a private network segment.

- Multiple network adapters per system

  Use multiple network adapters on MOP and disk servers.

- FDDI network backbone

  Use FDDI as a network backbone, with 10/100 bridges to connect to Ethernet segments.

- FDDI adapters

  Use FDDI network adapters in systems for which these are available, to connect them directly to FDDI.

- Fast CPUs for MOP and disk servers

  Provide sufficient MOP and disk server CPU capacity, by selecting a CPU with sufficient horsepower.

  Note that a multiprocessor system provides no more disk serving capacity than the uniprocessor version of the same CPU, because MSCP serving work is done on the interrupt stack on the primary processor.

- Multiple server nodes

  Configuring multiple server nodes allows CPUs to share the disk serving load, and provides better availability than that of a single disk server node.

- Throttling demand

  Various techniques are also available to control or "throttle" the demand generated by rebooting all the satellites at once.

Controlling satellite booting

Here are some ways the satellite boot process may be controlled:

- Use DECbootsync

  A tool such as DECbootsync may be used to control the number of workstations starting up simultaneously.

  This uses the distributed lock manager to control the number of satellites allowed to continue their startup command procedures at once.

Note that this doesn't control the booting of VMS itself, but does control the execution of startup command procedures and the setup work for layered products.

(DECbootsync is available through your local Digital sales office under a program called ASSETS.)

- Disable MOP service on MOP servers

  Boot requests may be temporarily disabled by setting the DECnet Ethernet circuit to a "Service Disabled" state, until the MOP server can complete its own startup operations, shadow copies can complete, and so on.

  Note that this doesn't prevent the satellites from requesting a boot, but it does prevent the MOP server from servicing them.

  Note also that if a satellite is requesting a boot and gets no response, it will make a request less often as time passes, so it may take a little longer than you'd normally expect to get the satellite up again once MOP service is re-enabled.

- Disable MOP service for individual satellites

  Boot requests may also be disabled on a per-node basis by temporarily clearing a node's information from the DECnet database on the MOP server using NCP, then re-setting it to re-enable nodes as desired to control booting.

  Again, this doesn't prevent the satellites from requesting boot service.

- Bring satellites to console prompt on shutdown

  2000, 3100, and 4000 model satellites may be set up so they halt upon restoration of power or execution of a halt instruction, rather than rebooting at once.

  A program can be used to perform a HALT instruction, and thus cause the satellite to enter console mode. This prevents the satellite from immediately trying to reboot, allowing more control over the rebooting process.

  This is also important for those systems which support remote triggering of a boot, because they only allow a remote trigger operation to occur while the system is in console mode.

- Boot satellites using MOP TRIGGER

  3100 model systems and VAXstation 4000 satellites may be set up so that a boot can be initiated remotely using a DECnet TRIGGER operation. For these systems, a command procedure can be created to trigger satellites to boot in a controlled fashion.

  The systems only respond to a trigger while they are in console mode, that is, while they are sitting at the ">>>" prompt.

Controlling MOP service on a MOP server

To disable MOP service during startup of a MOP server:

```
$ MCR NCP DEFINE CIRCUIT circuit SERVICE DISABLED
$ @SYS$MANAGER:STARTNET
$ MCR NCP DEFINE CIRCUIT circuit SERVICE ENABLED
```

In these examples, 'circuit' represents your MOP service circuit name, such as MNA-0 or BNA-0.

Note that at this point service is still disabled in the DECnet volatile database.

I wouldn't advise disabling this permanently (in the DECnet permanent database) because CLUSTER_CONFIG doesn't like it if it finds that MOP service is disabled on a node which is supposed to be a MOP server.

To re-enable MOP service later:

```
$ MCR NCP
NCP> SET CIRCUIT circuit STATE OFF
NCP> SET CIRCUIT circuit SERVICE ENABLED
NCP> SET CIRCUIT circuit STATE ON
```

Do this with a command procedure, so that it is done quickly, and DECnet service to the users will not be disrupted.

Controlling MOP service on a per-node basis

To disable MOP service for a given node:

```
$ MCR NCP
NCP> CLEAR NODE satellite HARDWARE ADDRESS
```

Here, 'satellite' represents the DECnet node name of the satellite node for which MOP service is to be disabled.

To re-enable MOP service for that node later:

```
$ MCR NCP
NCP> SET NODE satellite ALL
```

Bringing satellites to console prompt

A VAXcluster Console System (VCS) connection is the easy way to bring a system to the console prompt.

Manually pushing the HALT button at the workstation is another way. If neither of those is possible, here's how to do it:

To set up a satellite so that it will stop in console mode when a HALT instruction is executed, do this at the console:

```
For 3100s and 4000s:
   >>> SET HALT 3

For 2000s:
   >>> TEST 53
       2 ? >>> 3
```

This means to go into console mode on a HALT (and also upon power-up). The default action setting of 2 means to reboot on a HALT or power-up.

This need only be done once on each system, as the setting is saved in non-volatile random access memory (NVRAM).

When you wish the satellite to halt after a shutdown, you need to set things up in NCP so a reboot will load an image which does a HALT instruction. The READ_ADDR.SYS program, which is normally used to find out the Ethernet address of a satellite node, just happens to do a HALT instruction when it's run, and is readily available, so it will do the job nicely.

To HALT the satellite on the next reboot attempt:

```
$ MCR NCP
NCP> CLEAR NODE satellite LOAD ASSIST PARAMETER
NCP> CLEAR NODE satellite LOAD ASSIST AGENT
NCP> SET NODE satellite LOAD FILE MOM$LOAD:READ_ADDR.SYS
```

Now, shut the satellite down normally, but with an immediate reboot specified.

Plans for the next functional release of VMS include a SHUTDOWN command built into the SYSMAN utility to allow you do this easier and faster.

```
$ MCR SYSMAN
SYSMAN> SET ENVIRONMENT/NODE=satellite
SYSMAN> DO @SYS$UPDATE:AUTOGEN REBOOT
```

On reboot of the satellite, the READ_ADDR.SYS program will be loaded, dutifully print out the Ethernet address, and the system will "halt" in console mode.

Later, when you're ready to allow the satellite to boot normally, you need to fix up the node information in NCP, so that the satellite will load VMS, rather than the READ_ADDR.SYS program, when it is booted.

To allow the satellite to reboot normally:

```
$ MCR NCP
NCP> CLEAR NODE satellite LOAD FILE
NCP> SET NODE satellite ALL
```

Using the DECnet MOP trigger facility to boot satellites

The console firmware in some systems (the 3100 models and VAXstation 4000 systems) allows you to trigger them remotely to boot upon demand.

This capability has to be turned on before it can be used. Use the following commands at the console to turn this on:

```
>>> SET MOP 1
```

This enables the DECnet MOP listener.

```
>>> SET TRIGGER 1
```

This enables remote triggering.

```
>>> SET PSWD
```

This prompts twice for a 16-digit hexadecimal password string, used to validate a remote trigger request.

Again, like the HALT setting, these SET commands need only be done once on each system, since the settings are saved in NVRAM.

The NCP utility is used to initiate a boot of the satellite node:

```
$ MCR NCP
NCP> TRIGGER NODE satellite -
     VIA circuit -
     SERVICE PASSWORD xxxxxxxxxxxxxxxx
```

Here 'circuit' represents your MOP service circuit, and 'xxxxxxxxxxxxxxxx' is the 16-digit hexadecimal number specified when you did the >>> SET PSWD command at the satellite node's console.

The MOP server could be set up to automatically run a command procedure and trigger 5 or 10 satellites at a time to stagger the boot-time workload if desired. This can be done in priority order (i.e. your workstation first, your boss's next, her boss's next, and so on).

Note that when the power-on/halt switch is set as indicated previously, a power failure at the satellite will result in it stopping at the console prompt instead of automatically rebooting when power is restored. For a mass power failure, this is good, because it prevents server overloading, but if someone trips over the power cord for a single satellite, it won't reboot automatically.

A mechanism can be provided to scan for and trigger a reboot of satellites which go down inadvertently. This could simply be a batch job which runs periodically and uses the DCL lexical function F$GETSYI to check each node which should be in the cluster to see if it is in the cluster (using the CLUSTER_MEMBER item code), and issues an NCP TRIGGER command for any satellite which it finds isn't currently a member of the cluster.

System Parameters

As the cluster grows, certain data structures within VMS need to be able to grow to accommodate the larger number of nodes. If this is not possible (due to a shortage of non-paged pool, for example) this may induce intermittent problems which are difficult to diagnose.

It is recommended that AUTOGEN be run periodically (with FEEDBACK) on all nodes so that it can make the appropriate adjustments as the node count increases. Repeating this each time 10 nodes have been added seems to work well.

There are a few parameters which AUTOGEN may not currently handle well in large VAXcluster configurations and which may be of concern. The next maintenance release of VMS after 5.5-1 is slated to include enhancements to AUTOGEN to address these areas of concern.

SCSCONNCNT

The SYSGEN parameter SCSCONNCNT controls the number of Connection Descriptor Table (CDT) entries allocated at boot time. These are used for the various different types of connections that a VAXcluster node makes to other VAXcluster nodes for such tasks as general VAXcluster system coordination and for disk and tape MSCP serving work.

The default value of SCSCONNCNT is 40. An additional 200 entries are allocated in the Connection Description List (CDL) by VMS as a cushion against running out; once the initial CDTs are all used up, up to 200 more can be created. (Prior to VMS version 5.2, there were only 100 extra entries allocated in the CDL.)

The default value typically becomes insufficient when the cluster grows to somewhere around 50 to 70 nodes. A shortage is most likely to occur on VAXcluster nodes which are disk and/or tape servers.

Symptoms of a shortage are:

- nodes which are unable to join the cluster
- nodes which see disks or tapes served from some of the servers, but not from others

The SDA utility may be used to check the number of CDT entries in use on a given VAXcluster node:

```
$ ANALYZE/SYSTEM
VAX/VMS System analyzer

SDA> SHOW CONNECTIONS

    --- CDT Summary Page ---

CDT Address  Local Process  Connection ID  State  Remote Node
-----------  -------------  -------------  -----  -----------

 807C1BB0    SCS$DIRECTORY     CA840000    listen
 807C1D10    MSCP$TAPE         CA840001    listen
 807C1E70    MSCP$DISK         CA840002    listen
 807C1FD0    VMS$VAXcluster    CA840003    listen
 807C2130    SCA$TRANSPORT     CA840004    listen
 807C2290    VMS$DISK_CL_DRVR  CA880005    open    VAX1
 807C23F0    VMS$DISK_CL_DRVR  CAA00006    open    VAX2
 807C2550    VMS$VAXcluster    CA840007    open    VAX1
 807C2810    VMS$VAXcluster    CA860009    open    ALPHA1

Number of free CDT's:  31
```

To determine the actual number of connections in use, count the lines of output, and see if the total is close to or over the value of SCSCONNCNT. Note that the

```
Number of free CDT's:
```

line isn't very helpful in determining exactly how close you are to really running out because it doesn't include any of the extra slots VMS has allocated as a cushion unless they happen to have been used and freed over time.

If the value of SCSCONNCNT is found to be insufficient, add a line such as:

```
MIN_SCSCONNCNT = 300
```

to MODPARAMS.DAT, specifying a value appropriate for your configuration, run AUTOGEN, and reboot.

The number here should be the current actual usage for that node, or slightly higher if you anticipate growth, but don't go hog-wild, because for each unit that you increase this, 352 bytes of memory are allocated at boot time. (Remember that the 200-slot cushion is there in case you need it, and can be used if you're not short of non-paged pool.)

SCSRESPCNT

SCSRESPCNT controls the number of Request Descriptor Table entries, which keep track of requests made to other nodes for which a response has not yet been received. The default value for this may be insufficient once a cluster gets above about 100 nodes.

A shortage of this will have an effect on performance, as requests wait for a free RDT.

To see if SCSRESPCNT is insufficient, check each system to see if requests have had to wait because there were not enough free RDTs. The SDA utility may be used to check this:

```
$ ANALYZE/SYSTEM
SDA> READ SYS$SYSTEM:SCSDEF
SDA> EXAM @SCS$GL_RDT + RDT$L_QRDT_CNT
810610E4:  00000000   "...."
```

If this number is non-zero, and continues to increase during normal operations, you may want to increase the value of SCSRESPCNT.

Each unit that you increase this by only takes 8 bytes of memory, so you can afford to be generous with this one.

SCSBUFFCNT

SCSBUFFCNT controls the number of Buffer Descriptor Table (BDT) entries, which describe data buffers used in block data transfers between nodes.

Again, the effect of a shortage of this would be on performance. We haven't had reports of shortages on this one, but if you have a large cluster, it might not hurt to keep an eye on it, especially on the big server nodes.

To see if SCSBUFFCNT is insufficient, check each system to see if it has had to wait for BDT entries. The SDA utility may be used to check this, or you can use SHOW CLUSTER:

```
$ ANALYZE/SYSTEM
SDA> READ SYS$SYSTEM:SCSDEF
SDA> EXAM @SCS$GL_BDT + CIBDT$L_QBDT_CNT
81092E2C:  00000000   "...."
SDA> EXIT


$ SHOW CLUSTER/CONTINUOUS
Command > ADD BDT_WAITS

View of Cluster from system ID 4321  node: VAX1
+-------------------+---------+-----------+
|      SYSTEMS      | MEMBERS |  COUNTERS |
+--------+----------+---------+-----------+
|  NODE  | SOFTWARE |  STATUS | BDT_WAITS |
+--------+----------+---------+-----------+
| VAX1   | VMS V5.5 | MEMBER  |           |
| HSC1   | HSC V601 |         |         0 |
| VAX2   | VMS V5.5 |         |         0 |
| VAX3   | VMS V5.5 |         |         0 |
+--------+----------+---------+-----------+
```

If you find a count which is non-trivial, and which tends to increase over time under normal operations, consider raising SCSBUFFCNT.

For each unit that you raise this, only 16 bytes of memory are allocated, so overestimating this isn't a big deal either.

Network Problems

Network instability can seriously affect VAXcluster operation.

It is very common for sites to report that they experienced a period of time right after VAXcluster installation in which network defects caused many problems, but that things settled down as network problems were identified and fixed.

Following are some ideas on how to minimize network problems.

- RECNXINTERVAL

  The SYSGEN parameter RECNXINTERVAL specifies the number of seconds the cluster will wait when it loses contact with a node, before it kicks that node out of the cluster.

Many large VAXcluster configurations are operated with the SYSGEN parameter RECNXINTERVAL raised to 40 seconds or so, compared with the default of 20 seconds, at least while network troubleshooting is being done, to allow the cluster to "ride through" most network problems.

Note that raising RECNXINTERVAL can sometimes result in longer perceived application "hangs" when a node leaves the cluster abnormally while holding a lock on a shared resource, since the cluster will wait for RECNXINTERVAL seconds before kicking the departed node out of the cluster and freeing up any locks that it held.

- Network protection

  The LAN really needs to be treated as if it's a part of the cluster. An environment in which a random user can disconnect a thinwire segment and attach a new PC while 20 satellites hang is not a rewarding one.

- Divide and conquer strategy

  Dividing the network up into multiple segments, connected by bridges, and putting barrel connectors at midpoints of Ethernet segments can make troubleshooting a lot easier.

- Careful choice of hardware and configuration

  Be careful in your choice of network equipment. Certain hardware (such as DEQNA Ethernet adapters) is not suitable for use in a large VAXcluster system. If bridges cannot filter and forward packets at full line rates, you may run into trouble.

- LAVC$FAILURE_ANALYSIS

  New capabilities have been added to the VAXcluster code in conjunction with multiple adapter support which can assist in the isolation of network faults. See the VMS Version 5.5 VAXcluster Manual for information about the LAVC$FAILURE_ANALYSIS package, which was introduced with version 5.4-3.

Hot system files

In addition to the system disk throughput issues during a mass boot, which we discussed earlier, attention may need to be paid to the latency of access to particular system files even during steady-state operations, since it can affect the response time for various user activities, such as logging in, starting up an application, or issuing a PRINT command.

With a large number of users, certain system files may be quite busy:

- SYSUAF.DAT, RIGHTSLIST.DAT, etc.
- JBCSYSQUE.DAT (before 5.5)

  Job controller/queue manager performance may be a serious issue for large configurations if you're running VMS version 5.4[-*] or earlier versions. It may be necessary to place the system queue file JBCSYSQUE.DAT on a solid-state disk to achieve satisfactory response times for queue operations. The version 5.5 queue manager has solved these performance problems.

- Application images
- Anything else VPA/SPM/DECps points out

You can use SPM, VPA, or DECps to identify hot files.

Once you identify which files are of concern, several techniques are available to provide the required level of performance in accessing these files:

- Solid-state disks

  Seriously consider putting hot files on a solid-state disk, such as the ESE20 or ESE50.

- DECram for read-only files and images

  Putting hot read-only files (commonly-used executable images, command procedures, etc.) into DECram can help.

- Caching

  All of the standard disk I/O techniques help here, like caching in the HSC and the integral caches in disks, use of RMS global buffers, INSTALLing often-used images, and so on.

- Volume shadowing

  Volume Shadowing is meant to improve availability, but since it provides you with multiple spindles, it can also tend to help performance.

- Disk striping

  You can also use Disk Striping, which wasn't an option for the system disk itself, but can be used for hot files which are located on or have been moved to other disks.

  Note that if you're already using volume shadowing, and then use disk striping on top of that, you may not see as much gain as you'd normally get because you've already spread the I/Os across multiple spindles.

MOP/Disk Server and System Disk Availability

Typically a large number of satellite nodes depend on MOP and disk servers for their operation.

- MOP server failure

  Configuring more than one system as a MOP server allows satellites to boot despite loss of one MOP server node.

- Disk server or controller failure

  It's also good to configure redundant disk servers, such as dual-host DSSI systems, or multiple disk servers on CI, to avoid disruption of satellite disk service if a single disk server becomes unavailable.

- Power disruption

  Putting at least the MOP and disk server nodes on an uninterruptible power supply (UPS) system can help increase availability.

- Disk failure

  Because many satellites typically boot from each system disk, failure of that disk can disrupt operations for a large number of users. Volume shadowing may be used in conjunction with VAXsimPLUS to reduce the chances of an outage due to disk failure.

Shared Resource Availability

To provide a common environment, most clusters use a common authorization file, common queue file, and so on. Sometimes a process will take out a lock on one of these files and then get hung due to some problem such as a resource shortage. Other processes which need to use these common files will then hang waiting for a lock, and other processes may hang waiting for them, and so on. The odds of such a problem occurring increase as the number of nodes goes up.

Tools which can help with shared resource availability are:

- DECamds

  The DEC Availability Manager for Distributed Systems (DECamds) includes capabilities which assist in locating and correcting locking and resource shortage problems.

- DECterm window on system manager's workstation for each major system

  Leaving a session logged in to a privileged account on each of at least the big MOP and disk servers can be helpful during troubleshooting

- DECUS tools

  DECUS members have written tools such as FINDLOCK to assist with this type of problem. These are available through the DECUS library.

Cluster Alias

The DECnet Cluster Alias has the following node limits:

- Phase IV: 64 nodes (router required)

  Support for the DECnet cluster alias is currently limited to 64 nodes, so nodes above this limit cannot participate in the cluster alias. At least one system in the cluster must be a DECnet Full-Function (routing) node.

  As an unsupported work-around, a cluster might be set up to have more than one cluster alias. Each group of 64 nodes or less would be assigned a different cluster alias. There must be a routing node within each cluster alias group.

- Phase V: 144 nodes (no router)

  Plans for DECnet Phase V include support for 144 nodes under a cluster alias, and elimination of the requirement that a cluster member be a routing node.

System Disk Space

Although the minimal essential files for a satellite root take up very little space, such that more than a hundred roots can easily fit on a single system disk, if you tried to have separate dump files for each satellite node, or put page and swap files for all the satellites on the system disk, you'd quickly run out of disk space.

Ways around this include:

- Common dump files

  You can set up common dump files for all the satellites or for groups of satellite nodes. (For debugging purposes, I do recommend separate dump files for each MOP or disk server.)

- Page/swap files off system disk

  Use local disks on satellite nodes to hold page and swap files, instead of putting them on the system disk. Also move page and swap files for MOP and disk servers off the system disk.

Future directions

There are no design limitations inherent in VAXcluster technology which limit its scalability to 96 nodes. VMS Engineering is currently involved in the testing of larger VAXcluster configurations, and clusters of well over 100 nodes have been successfully built and operated.

The intent of this study is to stretch the envelope of cluster configurations in testing, with the aim of making clusters of all sizes work better. Feedback from this testing has already resulted in enhancements to the SYSMAN and MOUNT facilities, as indicated below.

MOUNT/CLUSTER limitations

The MOUNT/CLUSTER command currently has a limit of 96 nodes. If you issue a MOUNT /CLUSTER command when there are more than 96 nodes in the cluster, your node will bugcheck.

Removal of this limitation is slated for the next maintenance release of VMS after 5.5-1.

As a work-around, one can use SYSMAN or batch jobs to do a MOUNT/SYSTEM command on each node.

SYSMAN limitations

The SYSMAN utility currently has a limit of 128 nodes.

This limitation is removed in VMS version 5.5-1.

As a work-around, one can use a batch job on each node or use network tasks (e.g. TELL.COM) instead.

One more operational aspect of SYSMAN to consider is that SYSMAN does a command on each of the nodes sequentially. This can take a very long time to complete if there are a lot of nodes in a cluster.

System managers with large clusters sometimes start multiple SYSMAN sessions at once, with each one handling a subset of the total nodes, to speed up the process. SYSMAN supports the use of logical names to define groups of nodes.

You can also use batch jobs instead, which execute in parallel (by setting up a queue per node).

Conclusions

While building a cluster with a very large number of nodes is not simple, a configuration which provides

- support to a large number of workstations and users, with
- full shared access to common files, yet with

- minimal system management workload

is readily achievable using VAXcluster technology, coupled with some careful planning.

If you have feedback on the issues discussed in this article, or if you have ideas about the directions in which you need VAXcluster technology to be extended, your input is invited. Contact the author in care of this publication, or via Internet at:

keith.parris@zko.mts.dec.com