VAXcluster Systems Quorum

The Technical Journal for VAXcluster System Management

In This Issue:

- Volume Shadowing for OpenVMS Version 5.5–2 Performance
- An Introduction to Disaster-Tolerant VAXcluster Systems
- Understanding DECnet–VAX Phase IV Executor Pipeline Quota

Volume 8, Issue 2, November 1992

digital™

VAXcluster Systems Quorum

Quorum is published quarterly by Information Design and Consulting in Marlborough, MA. It contains VAXcluster-specific technical articles and VAXcluster-related articles.

Quorum welcomes comments and suggestions from its readers. Individuals or groups are encouraged to submit articles about their configurations and experiences. The editor reserves the right to edit, condense, and seek further authorization of any contribution. Please send submissions to the editor.

To obtain a *Quorum* subscription, contact Susan Pillsbury, (508) 467-7180 (in the USA).

To request back issues of *Quorum*, contact Susan Pillsbury.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Copyright ©1992 Digital Equipment Corporation

All Rights Reserved.

Editor Susan Pillsbury

Assistant Editor Brenda Tucker

Digital Equipment Corporation
MRO1–3/C8
P.O. Box 1001
Marlborough, MA 01752

The material in this document is for information purposes only. Digital believes the information to be accurate as of its publication date; such information is subject to change without notice. Digital is not responsible for any inadvertent errors. The opinions expressed by non-Digital contributors in this document in no way represent the attitudes or opinions of Digital, its employees, or management.

The performance information in this document is for guidance only. System performance is highly dependent on many factors including system hardware, system and user software, and user application characteristics. Customer applications should be carefully evaluated before performance is measured. Digital does not warrant or represent that a user can or will achieve similar performance expressed in transactions per second (TPS) or normalized cost/performance ($/TPS). No warranty on system performance or cost/performance is expressed or implied in this document.

The following are trademarks of Digital Equipment Corporation: ACMS, BI, CI, DBMS, DECintact, DECmcc, DECnet, DECperformance, DECwindows, DELUA, DEPCA, DEQNA, DEUNA, Digital, HSC, KDM, LAT, MicroVAX, MSCP, MS780–E, MS780–H, OpenVMS, PATHWORKS, RA, Rdb/VMS, RL, RM, RP, RV, RX, SA, SBI, SPM, TA, TK, TU, ULTRIX, VAX, VAX DOCUMENT, VAX Performance Advisor, VAX RMS, VAX Supercomputer, VAX Volume Shadowing, VAXcluster, VAXft, VAXserver, VAXstation, VMS, and the DIGITAL logo.

The following is a third-party trademark: Motif is a registered trademark of Open Software Foundation, licensed by Digital.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

1 Volume Shadowing for OpenVMS Version 5.5–2 Performance 1

2 An Introduction to Disaster-Tolerant VAXcluster Systems 18

3 Understanding DECnet–VAX Phase IV Executor Pipeline Quota . . 25

Additional VAXcluster Information 35

4 R1 and S1 Revision Management Level 37

VAXcluster Customer Configuration Database Questionnaire 59

Correction

The article “DSSI VAXcluster Interconnect” in the May 1992 issue of *VAXcluster Systems Quorum* contained an error in Table 4–2 on page 67. The table states that the KFQSA in/out connector bulkhead supports VAXcluster traffic. This adapter does not support VAXcluster traffic. *Quorum* regrets the error.

1
Volume Shadowing for OpenVMS Version 5.5–2 Performance

Nick Carr
OpenVMS Product Management
Digital Equipment Corporation

Introduction

This article discusses the performance of the Volume Shadowing for OpenVMS product and describes enhancements introduced with OpenVMS Version 5.5–2. It assumes you are familiar with basic Volume Shadowing concepts.[1]

The following subjects are covered:

- Read and write performance of Phase I and Phase II shadow sets
- Compute overhead of Volume Shadowing
- Copy algorithms and enhancements
- Merge algorithms and enhancements

Most of the Phase II algorithms are described, with specific comparisons to the HSC based Phase I implementation. Please refer to the *OpenVMS Version 5.5–2 Release Notes* for additional information on the new copy and merge performance enhancements.

[1] R. G. Davis, "VMS Volume Shadowing Phase II," *VAXcluster Systems Quorum*, Volume 7, Issue 2, November 1991.

Overview of Volume Shadowing

Volume Shadowing for OpenVMS is a Redundant Arrays of Independent Disks (RAID)-1 product with two implementations. Phase I, introduced in 1986, is a storage subsystem (controller)-based implementation. It uses a device driver that works in close conjunction with special-purpose HSC software. The device driver coordinates the functions processed by the HSC controller.

Phase II, introduced in 1990, is a fully host-based implementation. It also uses a device driver, but, unlike Phase I, it processes all the shadowing functions and makes no shadowing-specific demands on the storage subsystem. However, with the introduction of OpenVMS Version 5.5–2, Phase II takes advantage of optional controller-based performance features whenever possible.

There are advantages and disadvantages to both controller-based and host-based RAID implementations. While the Phase I Volume Shadowing implementation was successful, it proved restrictive in terms of configuration flexibility and scalability. These issues were overcome with the Phase II implementation, but the performance of copy and merge operations, while superior to Phase I in certain respects, became an issue in some configurations. Version 5.5–2 enhancements to the copy and merge algorithms, described later in this article, eliminated these issues.

From an industry viewpoint, vendors will continue to offer RAID products that use both implementation styles. Storage vendors will concentrate on subsystem-based RAID implementations, which provide a good level of platform (and operating system) independence and are generally simpler to implement. Platform vendors will concentrate on host-based RAID implementations, which provide storage subsystem independence, provide the ability to achieve higher I/O throughput and availability, and span a wider range of configurations. The Version 5.5–2 enhancements provide a unique mix of the implementation styles, which allow Phase II to benefit from a combination of their best features.

Shadowing Functions

There are four functions that must be performed by any RAID-1 implementation. The most important two, as with any disk I/O subsystem, are to satisfy read and write requests. The other two functions, copy and merge, are required for shadow set maintenance. While RAID-1 enhances data availability, the performance of read and write operations is nevertheless important to product success and should not be worse than a nonshadowed environment.

Copy and merge operations are the cornerstone of achieving data availability. They are transient conditions and should be relatively rare. Ideally, their performance should be unobtrusive to normal read and write operations. Such transient operations are not unique to a RAID-1 environment. All the other defined RAID levels, with the exception of RAID-0, have transient conditions that are required for RAID set

maintenance. RAID-5, for example, has several operations that are required for parity disk support.

Read and write performance is easy to measure, because the goal for these operations is to make them as quick as possible. However, for copy and merge operations, speed is not necessarily the overriding goal. This is because, to achieve speed, read and write performance is impacted while copy and merge operations are in progress. One of the primary design differences between Phase I and Phase II lies in this area. Phase I ensured that copy and merge operations completed quickly, by sacrificing the performance of read and write operations. In Phase II, as a result of numerous customer requests, this situation is reversed; copy and merge operations are designed to be subservient to the availability of good read and write performance. Maintaining data availability and minimizing system resource usage while performing these operations are also important design goals.

Read Operations

Reads are the most common operation in any disk I/O subsystem. In a RAID-1 environment, there are two or more identical drives from which to read. Reads can be issued to all the shadow set members in parallel, resulting in higher total read bandwidth. It may not be possible to achieve this increase if data transfer paths are shared between disks. Shared data transfer paths can be common in storage subsystem-based RAID implementations, but are usually avoided by careful configuration. The effect of shared data transfer paths is often unnoticeable, however, because data transfer is a small fraction of a complete I/O operation. Note that reliance on increased read bandwidth can be self-defeating — if a shadow set member fails, the read bandwidth drops accordingly, and the remaining members may be unable to sustain the required read performance.

Phase I Read Operations

With Phase I, the HSC controller is responsible for deciding which physical disk member should service a read request. This is a simple operation in which the request is queued to the drive that will provide the best response time.

The choice of drive is based on several criteria. First, if possible, requests are spread across different K.SDI modules. If this is insufficient to uniquely identify a drive, the unit with the lowest number of outstanding seeks is selected. If this is also insufficient to identify a drive, the unit with the shortest distance to seek is selected. Once the request is assigned to a physical drive, the standard HSC seek and rotational optimizations may be performed.

Phase II Read Operations

With Phase II, the host software is responsible for determining which physical drive should service a read request. As with Phase I, this is a simple operation — the request is always serviced by a locally connected disk (this includes HSC and Digital Storage Systems Interconnect (DSSI) disks, which are considered local by the host software) in preference to a disk that is MSCP served by a VAX host. If there are two local (or MSCP served) disks to choose from, the request is queued to the one with the shortest I/O queue. If all the queues are the same length, requests are distributed in a round-robin fashion.

A side effect of this approach is that it balances I/O in a VAXcluster environment, even though each VAX has no knowledge of the I/O load on a given disk from other VAXcluster members. If a disk is kept busy by multiple members, I/O from any node takes longer to service, and the node's I/O queue grows accordingly. For the same reason, it is viable to use the HSC Cache option to cache one member of a Phase II shadow set while not caching the others. The cached disk services the majority of I/O requests because of its quick turnaround time (and, thus, shorter I/O queue). However, if all I/O queues are empty, requests are issued in a round-robin fashion to the cached and noncached disks; this is not optimal, but suggests that the I/O load on the disks is such that they do not warrant caching at all.

Write Operations

RAID-1 write operations do not provide the opportunity for performance improvements offered by reads. Writes must be duplicated to every member of the shadow set. An important feature of the OpenVMS implementation is that writes are issued in parallel, so the performance is essentially that of the slowest member. In practice, because all members must be of the same physical type, the disk that has the farthest to seek/rotate becomes the limiting factor. As described in Read Operations, shared data paths may restrict performance slightly.

Phase I Write Operations

In the Phase I implementation, a single MSCP write command is issued to the HSC controller. The controller is responsible for performing the write on all shadow set members. The write data is transferred over the CI interconnect once for each shadow set member.

It proved to be inefficient to implement an algorithm that transferred the data once into the HSC controller's data memory and then transferred it from there to each shadow set member. Data transfer is a quick and simple operation on the CI, while data memory space within the HSC is too valuable to maintain for the relatively long periods of time that separate individual member seek delays.

Phase II Write Operations

In the Phase II implementation, an MSCP write command is issued to each shadow set member's controller. Even when all members are connected to the same HSC, multiple write commands are issued. As with the Phase I implementation, the write data is transferred to each shadow set member individually. The overhead of an additional MSCP command represents approximately 200 bytes on the storage interconnect (100 each for the command and response). The performance of all VAXcluster interconnects is such that this overhead is typically insignificant (the CI bus, for example, has a peak bandwidth of 8.75 megabytes per second per path).

Read and Write Performance

The read and write I/O bandwidth characteristics of any RAID-1 set, whether host-based or controller-based, are predictable — write bandwidth is approximately the same as nonshadowed environments, and read bandwidth is higher than nonshadowed environments. Total I/O throughput gains are heavily dependent on the read versus write mix on the shadow set. Because read activity is usually more common than write activity, some increase in throughput can be expected.

Read and write performance comparisons between Phase I and Phase II show them to be almost identical; test scenarios show that the I/O throughput of the two implementations does not differ by a factor of more than 0.8 to 1.3.

Of course, the goal of any RAID-1 implementation is enhanced data availability. Improved throughput is an additional benefit in many configurations.

Processing Overhead for Shadowing

A common cause of concern with host-based shadowing (Phase II) is the additional computing required by the host to choose which physical disk to read and to issue multiple writes. The point that is frequently overlooked is that the total processing required by the combination of host and subsystem is essentially constant — the work has to be done by one or the other.

Table 1–1 shows the CPU consumption for nonshadowed and Phase II-shadowed read and write operations. The figures are for a 2-member shadow set (RA82 disks) connected to a VAX 8700 (a 6-VUP processor) through an HSC controller. The table shows that an additional 0.14 milliseconds are required for read processing and 0.50 milliseconds for write processing. These times decrease proportionally with faster CPUs. Because the time for a total disk I/O is typically measured in tens of milliseconds, this processing overhead, whether performed by the host or storage subsystem, represents fractions of a percent of the total transfer time. The CPU time for Phase I shadowing is the same as for

nonshadowed operations, however, shadowing processing must be done within the HSC controller.

As an example, the compute overhead on a VAX CPU rated at 12 VUPs performing I/Os on a disk capable of 50 I/Os per second approximates to 0.07 milliseconds (for a read) in 20 milliseconds. This represents 0.35 percent of the overall I/O time.

If both the host and HSC subsystem are lightly loaded, which one performs the shadowing processing is unimportant. Preferably, the faster of the two would be selected (with today's fast VAX processors, this is usually the host). In conditions of heavy load, I/O subsystems tend to become a bottleneck before hosts. This can be especially true in VAXcluster systems, where many CPUs can issue I/O requests to a storage subsystem. So, using host-based processing for this activity is usually beneficial. Peak HSC I/O rates are marginally higher for nonshadowed and Phase II I/O requests than for Phase I I/O requests.

Table 1–1 CPU Time for 2-Member Shadow Set Read and Write (milliseconds)

Transfer Size (blocks)	Unshadowed Read	Phase II Read	Difference	Unshadowed Write	Phase II Write	Difference
4	0.71	0.84	0.13	0.72	1.24	0.52
8	0.76	0.89	0.13	0.76	1.26	0.50
16	0.83	0.97	0.14	0.83	1.34	0.51
32	0.98	1.14	0.15	0.98	1.56	0.53

Another common concern relates to HSC optimization techniques. It is often assumed that better optimizations can be performed if all shadowing processing is performed by the HSC subsystem. This concern is unfounded; both phases achieve similar performance, even though they implement different shadowing optimizations.

Transient Shadowing Conditions

RAID-1 implementations have to contend with two transient conditions that are required for shadow set maintenance — copy and merge. These operations, while simple in concept, are widely misunderstood. The two phases of Volume Shadowing implement these functions differently. Furthermore, with Version 5.2–2, enhanced copy and merge algorithms were introduced for Phase II.

Copy Operations

A copy operation is required when a new member is added to a shadow set or when a shadow set is created. The purpose of a copy operation is to duplicate the data on a source disk to a target disk. At the end of the copy operation, both disks are identical, and the target disk becomes a complete member of the shadow set. A copy operation is initiated by the DCL MOUNT command.

A copy operation is simple in nature; the source disk must be read, and the data must then be written to the target disk. This is typically done in multiblock increments, referred to as a logical block number (LBN) range. There are two complexities with the copy operation — handling user I/O requests while the copy is in progress and dealing with writes to the area that is currently being copied without losing the new write data. Phase I and Phase II Volume Shadowing handle these complexities differently.

Phase I Copy Operations

With Phase I, the copy operation is performed entirely by the HSC subsystem. As a result of a MOUNT command, the HSC is ordered to copy one disk to another. The copy operation continues to completion with no further host involvement. Because the copy operation is internal to the HSC subsystem, there is no consumption of VAX CPU or CI interconnect bandwidth, so the operation is highly efficient. However, the HSC internally generates the necessary copy I/Os at high speed, resulting in a reduction in user I/O bandwidth. When multiple copies are initiated in parallel, the HSC performs them in parallel, resulting in an even greater reduction in user I/O bandwidth. This performance impact is noticed by every node in the VAXcluster system. Therefore, care should be exercised before issuing MOUNT commands that will initiate Phase I copy operations.

To ensure correct synchronization of copy operations with incoming write commands (from VAX CPUs), the HSC subsystem delays the processing of any write that overlaps with the area of disk currently being copied.

Phase II Copy Operations

Prior to Version 5.5–2, the Phase II copy operation was performed by the VAX CPU. With Version 5.5–2, the Phase II copy operation is enhanced so that the HSC subsystem performs the disk-to-disk data movement operations. When performed by the VAX, the copy is termed *unassisted*; when performed by the HSC, the copy is termed *assisted*.

Unassisted Phase II Copy Operations

Phase II Volume Shadowing performs an unassisted copy operation when it is not possible to use the HSC copy assist feature. The most common cause of this situation is when the source and target disks are not on line to the same HSC subsystem (the two disks may be connected to any controller anywhere in the VAXcluster system). An unassisted copy operation consumes a small amount of CPU bandwidth on the node that is performing the copy, but not on other nodes in the VAXcluster. This was measured at approximately 2 to 3 percent of a 5- to 6-VUP CPU per copy operation. It also consumes interconnect bandwidth.

Because the copy operation is fully controlled by a VAX CPU, it is simple to ensure that user I/Os are permitted to proceed at a reasonable rate. On the node performing the copy, user and copy I/Os compete evenly for the available I/O bandwidth. For other nodes in the VAXcluster, user I/Os proceed normally and contend for resources in the HSC with all the other nodes, in the usual fashion. Thus, user I/O performance during copy operations with Phase II shadowing is better than with Phase I. However, the copy operation takes longer as the user I/O load grows.

The copy operation is controlled by the SHADOW_SERVER process. The server chooses an LBN range size based on a multiple of the track size of the disks being copied. This improves copy efficiency by avoiding the need for mid-transfer seek operations. During the copy, the concept of a *copy fence* is created — the fence moves across the disk, logically separating the copied and uncopied areas. With the aid of the lock manager, the copy fence is distributed around the VAXcluster. User reads to the copied side of the fence may be serviced by either disk, while user reads to the uncopied side of the fence may only be serviced by the source disk. The VAXcluster fence enables another node to continue the copy operation, without the need for restarting at the beginning, if the node performing the copy shuts down.

Coordinating the copy operation with VAXcluster write activity to the shadow set is complex. Unlike the HSC with Phase I shadowing, the SHADOW_SERVER has no knowledge of other write activity. Because MSCP disk controllers are free to reorder reads and writes as part of their optimization techniques, care is required to ensure complete data integrity. This prohibits a simple copy algorithm whereby the source disk is read and the target disk is written. A user write issued to both disks in a shadow set by any node in the VAXcluster may be reordered with a write from the copy operation. This can result in the contents of the disks being different. While it is possible to implement a VAXcluster locking strategy to handle this situation, it results in excessive synchronization delays for every write operation to handle the rare case when the SHADOW_ SERVER copy LBN and a user write LBN overlap.

The solution to this problem requires no VAXcluster synchronization and permits good user write performance. All user writes to the uncopied side of the copy fence are serialized so that the source disk is written first, followed by the target disk (writes to the target disk on the uncopied side of the fence are necessary because the copy fence may lag behind the SHADOW_SERVER's copy LBN by several LBN ranges). Writes to the copied side of the copy fence are issued in parallel, as with steady-state operation.

The SHADOW_SERVER uses special-purpose routines within the shadowing device driver to perform the copy operation. These routines implement a 5-stage algorithm. First, the source disk is read, followed by a data compare operation with the target disk. If the data matches (the source and target disks are the same), the SHADOW_SERVER moves on to the next LBN range. If the data does not match, the source data is written to the target disk, and the algorithm returns to the first step. This time the read, followed by the compare, should match. The only time it does not match is when a user write has overlapped the SHADOW_ SERVER's operations, in which case the algorithm once again writes the target disk and returns to the first step. To avoid endless looping at write hotspots, a loop counter is implemented. If a hotspot is identified, a temporary VAXcluster synchronization lock is applied to the disk to allow the copy of the affected LBN range to complete.

This 5-stage algorithm ensures that, at the end of the copy operation, the two disks are identical and that conflicts with other write activity within a copy LBN range are handled correctly, without the need for costly synchronization techniques.

There is one interesting side effect to this algorithm. The time and amount of I/O required to perform the copy operation is heavily dependent on the similarity of the source and target disks. If the disks are very similar, the copy operation consists of reads and compares (two I/Os per LBN range). This is the case if a member is removed from a shadow set for some reason and then remounted into the shadow set. However, if the source and target disks are completely different, five I/Os are required per LBN range (read, compare, write, read, compare). On average, this takes at least 2.5 times longer than a copy of two similar disks.

In summary, unassisted Phase II copy operations are performed by a VAX host CPU, allow good simultaneous user I/O performance, and have elapsed times that are dependent on the similarity of the disks being copied and user I/O rate. As mentioned earlier, unassisted copy operations are not CPU-intensive (the compare operations are performed by the controller), however, they are I/O-intensive and consume interconnect bandwidth.

Assisted Phase II Copy Operations

With the introduction of Version 5.5–2, and HSC software Version 6.5, it is possible, in many cases, to benefit from the best of both Phase I and unassisted Phase II copy features (short copy elapsed times and low user I/O impact).

With assisted Phase II copies, the SHADOW_SERVER process on one VAXcluster node is still responsible for controlling the copy process. The SHADOW_SERVER issues a Disk Copy Data (DCD) MSCP command to the HSC controller for each LBN range. The HSC then performs the disk-to-disk copy. This avoids consumption of interconnect bandwidth. Because the HSC can synchronize copy activity with incoming user write activity, the copy can be implemented within the HSC as a simple read-followed-by-write algorithm. This provides elapsed times similar to Phase I and removes any dependency on the similarity of the data on the source and target disks.

Three features ensure user I/O performance is not excessively hindered by DCD operations. First, the SHADOW_SERVER issues a DCD command to the HSC for each LBN range; it must compete evenly with other user activity for VAX CPU resources to issue these commands. Second, the HSC performs DCD operations at the same priority as user I/O requests. Lastly, the HSC does not perform multiple DCD commands in parallel, rather, it serializes them to preserve I/O bandwidth for user I/O activity.

Assisted Copy Performance

Because the HSC performs multiple DCD commands in series (even though SHADOW_SERVERs may issue them in parallel), the elapsed time for multiple copy operations grows in direct proportion to the number in progress. In other words, four copies started at the same time take approximately four times longer than one copy. In contrast, unassisted copies are performed in parallel with each other and any assisted copies that are in progress.

For configurations where many simultaneous copy operations are performed, it is possible that using a mixture of unassisted and assisted copies will produce shorter total elapsed times than using exclusively assisted copies. The HSC utility SETSHO enables you to specify the number of DCD-assisted copies the HSC will perform at any one time; additional copies are performed unassisted.

The default value for assisted copies is 4. With this setting, VAX CPUs may initiate up to four assisted copy operations per HSC; additional assisted copies are refused by the HSC and this causes the SHADOW_ SERVER to revert to an unassisted copy. Refer to the *OpenVMS Version 5.5–2 Release Notes* for information on how to set the DCD limit.

Remember that unassisted copies consume interconnect bandwidth while assisted copies do not, so elapsed time is not the only parameter to consider.

In summary, assisted Phase II copy operations are controlled by a VAX host and performed by the HSC controller, allow good simultaneous user I/O performance, and have short and consistent elapsed times. They are not dependent on the similarity of the data on disks being copied.

Tables 1–2 through 1–5 show the copy performance of the Volume Shadowing implementations for several types of disk. All figures are given in minutes.

Table 1–2 RA82 Disks

Simultaneous Copies	1	2	3	4	5	6
Assisted Phase II	9	17	25	34	N/A	N/A
Unassisted Identical Phase II	24	28	34	41	N/A	N/A
Unassisted Different Phase II	91	120	157	200	N/A	N/A
Phase I	15	20	21	22	N/A	N/A

Table 1–3 RA90 Disks

Simultaneous Copies	1	2	3	4	5	6
Assisted Phase II	14	28	42	N/A	N/A	N/A
Unassisted Identical Phase II	40	52	67	N/A	N/A	N/A
Phase I	20	32	40	N/A	N/A	N/A

Table 1–4 RA70 Disks

Simultaneous Copies	1	2	3	4	5	6
Assisted Phase II	6	12	18	24	30	36
Unassisted Identical Phase II	17	19	21	23	25	27

Table 1–5 RA92 Disks

Simultaneous Copies	1	2	3	4	5	6
Assisted Phase II	22	37	50	67	N/A	N/A
Unassisted Identical Phase II	54	60	71	82	N/A	N/A

While the tables are not complete (all timing figures were not available at the time this article was written), the major points are clear. Most importantly, note that the timing ratios vary with disk type and copy

algorithm. These figures were achieved using test systems with no user I/O load.

In can be seen that Phase II-assisted copies are generally quicker than Phase I for one or two simultaneous copy operations and slightly slower for three or four copies. When larger numbers of simultaneous copies are required, Phase I tends to saturate the HSC subsystem (depending on the model type), so timings are not easily predictable. Phase II-assisted times grow steadily, reflecting the serialization of DCD commands.

HSC algorithm differences generally cause a Phase II-assisted copy to complete in less time than a Phase I copy. For most disk types, Phase I copies are performed using an LBN range that is set to the size of a single disk track. Each track is read from the source disk and then written to the target disk. Phase II-assisted copies are performed using an LBN range that is the size of approximately 20 tracks. The buffering algorithm permits a spiral read to be performed from the source disk and, at the same time, a spiral write to be performed on the target disk. The overlapping of the read and write operations, combined with larger transfer sizes, allows faster copy operations. Note that most of this efficiency is lost if both the source and target disks are connected to the same HSC requester.

If unassisted copy operations are performed with disks containing dissimilar data, the elapsed times grow significantly. When many simultaneous copies are required for dissimilar disks, it is preferable to use the copy assist feature for more than four copies. You can set the HSC DCD connection limit to a value higher than four to achieve this.

Merge Operations

A merge operation is required when a failure occurs that results in the possibility of incomplete writes. The following sections describe the merge algorithms used by Phase I and Phase II.

What Is a Merge?

Even in a nonshadowing environment, it is possible for system failures to result in write inconsistencies. For example, if a write I/O is issued to a nonshadowed disk and the system fails before the issuer is notified of completion, it is not possible to know the status of the write. It may have completed, or it may not have completed. When the system recovers, the only way to identify the status of the write is to read the affected data. If the old data is still there, the write did not complete; if the new data is there, the write did complete. However, whether the data is old or new cannot be decided by the operating system or the disk subsystem. The old or new decision must be made by the user or application software. Whether recovery is needed (the disk contains old data, so the new data must be written again) is typically determined by people or database and application journaling techniques.

In a shadowed environment, the same problem arises, but with an extra complexity. If a write is issued to a shadow set and the system fails before the issuer is notified of write completion, the status of the write is unknown (as in the nonshadowed example). To identify the status of the write, the affected data must be read when the system recovers. However, unlike the nonshadowed environment, it is not simply a case of a single disk containing either old or new data. With a shadow set, both disks can contain the old data, both can contain the new data, or one disk can contain new data and the other old data. The exact timing of the failure during the original write defines which of these three scenarios results.

Once again, recovery from the old or new data conundrum must be performed by a human user or an application journal file. However, it is essential that the shadowing software always return the same data to the user or application. If one disk contains old data and the other new data, the shadowing software must take steps to ensure that only one is returned to the user (until the data is potentially rewritten by a user recovery). It does not matter which data is returned, because, from the user's perspective, the exact timing of the failure is unknown. Also, as discussed earlier, the shadowing software cannot distinguish old from new.

The solution to this complexity is the merge operation. During the merge, members of the shadow set are physically compared to each other to ensure that they contain the same data. This is done by performing a block-by-block comparison of the disks. During the merge, a *merge fence* is created that moves across the disk and separates the merged and unmerged portions of the disk. As the merge proceeds, any blocks that are identified as different are made the same — either both old or both new. As described earlier, the shadowing software has no knowledge of which data is old or new, so making the disks identical can be considered equivalent to moving the time of the original failure forward or backward by a few milliseconds — enough to ensure that *all* disks performed the write or did not. Looked at another way, making the disks identical is equivalent to converting the original write into a single atomic event — it either completed on all disks or none.

User reads to the already merged side of the fence can be satisfied by any member of the shadow set. User reads to the unmerged side of the fence are also satisfied by any member of the shadow set, but only after the data is compared to all other members of the shadow set. Any differences are caused by one member containing old data while the others contain new (or vice versa). These are corrected immediately, prior to completing the user read request. Further reads of the same data do not require this correction (because the compare operation will no longer fail). By performing dynamic correction of inconsistencies during reads to the unmerged side of the fence, a shadow set member can fail at any point during the merge operation without impacting data availability.

A higher performance alternative would be to satisfy all reads to the unmerged side of the merge fence from a single shadow set member, thereby ensuring that the same data is always returned to the user. However, if that member failed, reads would have to be serviced by the other members, and they might return different data. Dynamic correction avoids this problem.

The critical point is that although disks in a shadow set may not be identical after a failure (because there can be old data and new data differences), they are equally valid. However, shadowing software makes it impossible to determine, from an application viewpoint, that they are different — it always returns the same data. The background merge process ensures that the disks are identical. Because dynamic correction resolves differences detected by reads to the unmerged side of the merge fence, the failure of any shadow set member during the merge process has no impact on data availability.

Phase I Merge Operations

With Phase I shadowing, the HSC subsystem is responsible for performing merges. Merge operations are initiated when an HSC failure occurs and all members of the shadow set fail over to another HSC. The new HSC is not aware of outstanding write operations in the failed HSC, so it must put the shadow set into a merge state. The HSC performs the merge operation and performs dynamic correction, as necessary, for incoming reads.

While it is reasonable to assume that knowledge of outstanding writes exists in the VAX host nodes, which can be reissued to the new HSC, merge processing is still needed. It is possible that a VAX node may fail at the same time that the HSC fails, hence all knowledge of an outstanding write can be lost.

With Phase I shadowing, merge operations are performed at the highest possible speed and, as with copy operations, can impact user I/O performance on every node in the VAXcluster. Unfortunately, HSC failure cannot be predicted, so merge operations can be initiated at the most inopportune times. Furthermore, because all members of a shadow set in a merge state contain completely valid data, there is no requirement for the merge to complete quickly (other than to avoid the overhead of performing dynamic correction on the unmerged side of the merge fence). This behavior is somewhat undesirable and was carefully avoided during the design of the Phase II implementation.

Interestingly, the Phase I merge operation is implemented as a modified form of copy. Instead of performing compare operations, one member is simply copied to the other members. In other respects, conventional merge algorithms are performed.

Phase II Merge Operations

With Phase II, the merge operation is performed entirely by the VAX CPU in versions prior to Version 5.5–2. With the introduction of Version 5.5–2 the merge operation is still performed by the VAX CPU, but can also take advantage of enhanced disk controller software to implement a completely different merge algorithm. When merges are performed using the new algorithm, they are referred to as *assisted merges* or *minimerges*.

A Phase II merge operation is initiated when a VAX node that has a shadow set mounted leaves the VAXcluster without dismounting the set, because it may have completed writes to some, but not all, members of the set. The need for a merge operation is detected by the remaining VAX nodes, and the SHADOW_SERVER process on one of them performs the merge operation. The remaining nodes perform dynamic correction during reads to the unmerged side of the merge fence.

Unassisted Phase II Merge Operations

The unassisted Phase II merge algorithm is a simple read of one member, followed by a compare of the other members. As with a copy, the merge operation uses an LBN range that is chosen for maximum performance. The merge fence location is distributed across the VAXcluster. To ensure minimal impact on user I/O, the SHADOW_SERVER times all merge I/Os and implements a backoff mechanism. If heavy user I/O rates cause merge I/Os to take longer to complete, a backoff delay is inserted between subsequent merge I/Os. A timer-based mechanism is required because the MSCP protocol does not allow priorities to be allocated to I/Os. (All MSCP read and write operations take place at the same priority; this allows controller optimization techniques to be effective.)

The overall effect of the backoff mechanism is to ensure that user I/Os proceed unhindered by merge operations. As described earlier, when a shadow set is in a merge state, data availability and integrity are not compromised in any way. However, because merge operations are initiated by a node failure, there is no way to control when they occur (unlike copy operations, which are only initiated when a user issues a MOUNT command). Performing the merge operation as a background process ensures that failures that occur at inopportune times do not impact user I/O. A side effect of this implementation is that when user I/O loads are high, merges can take extended periods of time to complete. Also, if another node fails before a merge is complete, the merge is abandoned, and a new one initiated. System managers need not be disconcerted by lengthy merge operations — data availability and integrity is fully preserved.

For most user I/O loads, the additional overhead of compare operations with each read on the unmerged side of the merge fence is not noticeable. However, when the read I/O load is very high, the additional burden of compare operations can exceed the total I/O bandwidth of the disks. This causes a drop-off in read throughput. Because the merge backoff algorithm is carefully designed to ensure that the merge process is

unobtrusive (thereby taking longer to complete), the two algorithms can work against each other. Recent changes to the merge backoff algorithm and timers in the SHADOW_SERVER minimized this effect. Moving hot files to the front of the shadow set, ensuring they are merged early, can also minimize it. However, note that this effect only results when the read load on the shadow set is higher than a single member can support, so that the configuration depends on having at least two members present at all times. This reliance on always having more than one member available contradicts the underlying purpose of shadowing — to improve application availability with the ability to continue operations in the event of disk failure.

Assisted Phase II Merge Operations

With the introduction of Version 5.5–2 and new controller software, it is possible to take advantage of the new merge performance assist. When all members of a shadow set support the merge assist, writes to the members are logged in controller memory *write logs*. A write log entry — one for each write — contains the LBN of the write and information regarding which VAX node issued it. If a VAX node fails, thereby triggering merge operations, a remaining node interrogates the controllers' write logs to identify outstanding write operations from the failed node. Once the LBNs of any outstanding writes are known, they may be individually *minimerged*.

This process removes the need for a total read and compare scan of the shadow set members to identify differences. Removing the requirement to perform this scan is the primary benefit of the minimerge feature — it avoids consumption of the I/O resources that are needed to perform the scan (and those required to perform the compare operations when servicing reads to the unmerged side of the merge fence).

Minimerge operations complete in a very short period of time. The exact duration is dependent on the amount of outstanding write activity at the time of the failure, but is usually just a few seconds. Under any circumstances, assisted Phase II merges complete quicker than either Phase I or unassisted Phase II merges, which take between tens of minutes and several hours.

Because controller write log entries are maintained individually for each shadow set member, there is no requirement that all members be connected to the same HSC to take advantage of the minimerge feature. Write logs are also maintained by the KDM70 controller and RF35 and RF73 disks. All future DSA disks and controllers will support write logging. Note that although write logs are contained within the storage controllers, they are interrogated and processed by a VAX host. Controller-based processing of write logs was not implemented, because this requires that all members of a shadow set be connected to the same controller.

The write log function does not require extra MSCP commands to the disk controller, because logging information is embedded within the normal MSCP write command. This avoids additional load on the storage interconnect.

With Version 5.5–2, the minimerge feature cannot be used on a shadowed system disk. This is because the boot driver that is used to write the crash dump file does not maintain write log entries. A minimerge operation would, therefore, miss the crash dump file. As a result, system disks always undergo unassisted merge operations.

Summary

The Volume Shadowing for OpenVMS product provides the ability to achieve very high data availability with excellent performance.

Version 5.5–2 Phase II enhancements have enabled copy operations to be performed by the storage controller whenever possible, resulting in savings in system resources and time, while retaining good user I/O performance. For the most common copy operations — those involving a single copy — assisted Phase II generally outperforms Phase I.

The Phase II write logging enhancements have resulted in industry-leading merge performance, to the point that merge operations are almost invisible during normal system operation.

VAXcluster configurations with HSC subsystems can use either phase of Volume Shadowing. At this time, both are fully supported and, if necessary, may be used at the same time. Prior to Version 5.5–2, the decision of which phase to use was complicated by the advantages and disadvantages present in both implementations. With the availability of the Phase II performance assists, this decision is simpler; it is expected that there will be few, if any, conditions under which Phase I outperforms Phase II.

Experience gained with the two implementations shows that the optimal RAID-1 environment is provided by carefully separating functions between the host and the storage subsystem when possible. Host-based control of all functions provides the foundation for flexibility and scalability; enhanced performance may then be achieved with optional controller-based features. It is probable that implementations of most RAID levels will benefit from a similar division of functions between the host and storage subsystem.

2
An Introduction to Disaster-Tolerant VAXcluster Systems

Roy G. Davis
VAXcluster Systems Engineering
Digital Equipment Corporation

The evolution of VAXcluster technology led to increasing degrees of resource availability. VMS Version 4.0 supported both the dual-pathed CI and the dual porting of disk devices. Thus, a single point of failure for access to a disk device could be avoided when VAXcluster configurations were first implemented. Volume shadowing, introduced in Version 4.4 of VMS, improved information availability by replicating data on multiple disk volumes. Since volume shadowing can also be applied to system disks, it also improved system availability. Version 5.0 introduced the use of multiple interconnects in VAXcluster configurations. It also supported failover of VAXcluster communication from one interconnect to another. Version 5.4 introduced support for a VMS system using multiple CI adapters. Version 5.4–3 introduced support for a VMS system using multiple local area network (LAN) adapters for VAXcluster communication. Thus, there need not be a single point of failure for communication among OpenVMS systems in a VAXcluster.

This same evolution also led to VAXcluster configurations whose nodes are distributed over increasingly larger geographies. The original VAXcluster interconnect limited the maximum supported distance between any two nodes in a VAXcluster to 90 meters. Hence, clustered VMS systems and storage were typically confined to a single computer room. Ethernet permitted nodes to be spread over distances measured in hundreds of meters or, perhaps, a few thousand meters. Thus, Ethernet led to the clustering of nodes throughout a building or multiple buildings in close proximity to each other. With Fiber Distributed Data Interface (FDDI), the geographic distribution of nodes in a VAXcluster configuration can be measured in tens of kilometers. When bridges are used to combine FDDI and Ethernet, even larger VAXcluster configurations can be achieved. Support for clustering over even larger geographies is planned.

Support for FDDI as a VAXcluster interconnect leads to another form of resource availability in the VAXcluster computing environment — disaster tolerance through site redundancy. This concept is based on duplicating critical hardware and software components of a VAXcluster configuration in two distinct and widely separated locations. Even though there is a great distance between these two locations, the OpenVMS systems and storage in both locations function as a single VAXcluster configuration. These systems all satisfy the Rule of Total Connectivity. If disaster strikes one location, the other location continues to provide the critical elements of the computing environment.

In this context, the word *disaster* refers to events such as a fire or flood destroying a computer facility or the loss of electrical power to a computer facility. It does not necessarily include such things as an application software problem corrupting all copies of a database on a shadow set. Also, in this context, each location is called a *site* or *datacenter*.

Certain degrees of redundancy are required in a disaster-tolerant VAXcluster configuration.

- The CPU power and main memory at each site must be sufficient to satisfy the critical computing needs of the application environment. This is typically achieved by duplicating at least a subset of the OpenVMS systems. However, this can be achieved by having unlike CPU types in the two sites, so long as sufficient CPU power and main memory are present at each site to run mission critical applications.
- The system disk for each CPU is located at the same site as the CPU. If a system disk is shadowed (it probably should be for critical systems), all the members of the shadow set are located at the same site as the systems using the system disk. This precludes distributing members of a shadowed system disk between the two sites. It also precludes distributing between the two sites systems using a common system disk.
- For performance reasons, paging and swapping should generally not span sites. In other words, an OpenVMS system should not page or swap to a disk device located in a site other than the one in which the system resides.
- Critical data disks are shadowed. However, members of each such shadow set are distributed between the two sites. If disaster strikes either site, the critical data is still available at the remaining site. If only one member of a critical shadow set is present at the remaining site, a scratch pack can be mounted into the single-member shadow set to increase critical data availability.
- Each site should include at least a limited number of backup devices, such as tape drives and removable disk storage.

Disaster recovery is accomplished through a plan that is specific to both the application environment and the configuration. Typically, this plan is invoked in two situations.

- A disaster results in the loss of one site.

  Normally, quorum votes are divided equally between the two sites. Since neither site has more than half the total number of votes available to the VAXcluster, OpenVMS systems at the remaining site experience a *brief* quorum hang. The word brief is somewhat subjective in this context, because a manual procedure is used at the remaining site to restore quorum for the surviving systems without rebooting them. Critical applications are then resumed. If necessary, resources are reallocated at the remaining site to support resuming critical applications. The membership of critical shadow sets with only one surviving member is increased by adding scratch disks into those shadow sets.

- Communication is lost between the two sites.

  The probability of this event occurring is reduced by using redundant interconnects. For example, critical systems at both sites can each be attached to two FDDI logical rings. Nonetheless, the probability of this event is still greater than zero. Handling this event is somewhat more complex than when disaster strikes one site.

  When communication is lost between two sites for a short time, the configuration can recover without manual intervention. For example, suppose that System Communications Architecture (SCA)[1] virtual circuits and, thus, total connectivity, are lost because of a short burst of noise on the interconnects between the two sites. If the virtual circuits can be reestablished and total connectivity restored within the limits associated with the RECNXINTERVAL parameter, there is no need for a cluster state transition. Mount verification simply completes, and outstanding I/Os are reissued by the class drivers. This is logically equivalent to a short-term transient loss of connectivity among VAXcluster nodes in a single computer room. During the time that connectivity is lost, a quorum hang occurs because neither site has more than half the total number of votes available to the VAXcluster.

  If loss of connectivity between the sites exceeds limits associated with RECNXINTERVAL, manual intervention is required. In a manner analogous to a disaster scenario, established procedures are typically used to take one site off line while restoring quorum at the other. Critical applications are then resumed at the surviving site. If connectivity is even partially reestablished between the sites before

[1] SCA defines the concepts, services, functionality, and procedures that provide for VAXcluster communication on physical transmission media, such as CI, Digital Storage Systems Interconnect (DSSI), Ethernet, and FDDI. A virtual circuit is a reliable logical communication link between two ports on two nodes.

these procedures are applied, some of the OpenVMS systems at either or both sites may crash. Thus, the first of these procedures typically involves disabling use of VAXcluster interconnects between the sites.

Manual intervention can be eliminated in a number of situations by using a *quorum system*. This is a member of the VAXcluster located at a third site and having one vote. It effectively acts as a *tie breaker*, allowing one site to regain quorum and resume activity without manual intervention. OpenVMS systems at the other site are removed from the VAXcluster relative to the systems at the site that regains quorum. The following two examples illustrate using a quorum system:

- Assume that the two sites are in different cities, that the quorum system is somewhere in between, and that the two sites and the quorum system each have separate sources of electrical power. If a power failure strikes one site, the vote from the quorum system combines with the votes at the second site to avoid loss of quorum at the second site. Thus, systems at the second site and the quorum system participate in a cluster state transition. They automatically remove from the VAXcluster those systems at the site that lost power and resume activity.
- Assume that communication between the two sites is provided by one FDDI logical ring. Typically, at least two solid failures in the ring are required for communication to be lost between the sites. If there are two failures, one site remains in contact with the quorum system. That site then has sufficient votes for quorum and performs a cluster state transition to remove the systems at the other site from the VAXcluster. Meanwhile, the other site lacks sufficient votes for quorum and experiences a quorum hang.

  In this case, prior to restoring communication between the two sites, take down the systems that entered a quorum hang. After the failure is corrected, reboot the systems and allow them to join the remainder of the VAXcluster. Add the shadow set members at this site back into their respective shadow sets, and bring them up-to-date by means of shadow set copy operations. It is possible, however, to forget to do this and restore connectivity without rebooting the systems that were removed from VAXcluster membership by the surviving site. If this occurs, the surviving site protects itself by forcing these systems to crash and reboot.

The quorum system is typically an inexpensive system whose only purpose is to contribute a tie breaker vote in situations such as the ones just described. It should have its own local system disk and not be expected to handle critical application processing. However, relatively unimportant reproducible tasks can be off-loaded to it from either site.

Observe that in a 2-site configuration, the quorum system plays a role similar to that of a quorum disk in a single-site configuration involving only two OpenVMS nodes. The quorum system is to each site what the quorum disk is to each of the two OpenVMS nodes in the single-site VAXcluster. In fact, because a quorum disk can be directly accessed by nodes at only one site, it should not be used in a multisite VAXcluster.

Multisite VAXcluster configurations include a degree of system management complexity not usually found in single-site configurations. To address this complexity, Digital provides Operations Management Station (OMS) software as part of its VAXcluster Multi–Datacenter Facility (MDF) package. An OMS system integrates several software components to provide a single management domain for controlling both normal operations and disaster recovery. In so doing, it provides for the consolidation of multiple physically independent datacenters into a single logical datacenter.

One OMS is located at each site in a multisite configuration as shown in Figure 2–1. Each OMS system is a totally self-contained DECnet node. Using LAT protocol, each OMS controls OpenVMS and HSC nodes throughout the VAXcluster by means of console ports on those nodes. Using DECnet protocol and Ethernet-to-FDDI bridges, the two OMS systems communicate with each other over the FDDI logical ring.

Conceptually, an OMS system consists of a *platform* and at least one graphics *display*. The platform contains its own CPU and main memory to execute the OMS software. It also contains at least one disk device to store and retrieve OMS related data and to act as a local system disk. The display consists of a graphics monitor and a keyboard by which a person interacts with the OMS system. Depending on the platform used, an OMS system may include multiple displays. Any node in a VAXcluster managed by OMS can be accessed from any display on each OMS system.

OMS software is currently based on the OpenVMS operating system. However, an OMS is not permitted to be a member of a VAXcluster that it controls. This is because it would be undesirable for activity on an OMS to stall during a state transition or a quorum hang. Technically, an OMS is permitted to be a member of another VAXcluster, but this is discouraged for similar reasons.

During normal operation, one OMS platform acts as the primary OMS platform for the entire configuration, and the other OMS platform acts as the secondary platform for the VAXcluster. OMS software in the primary platform manages OpenVMS and HSC nodes, while the secondary platform is in a *hot standby* mode. However, this distinction is transparent to the user. The two OMS systems interact in a way that presents them to the user as a single logical entity. Any node managed by OMS can be managed from any display attached to either platform.

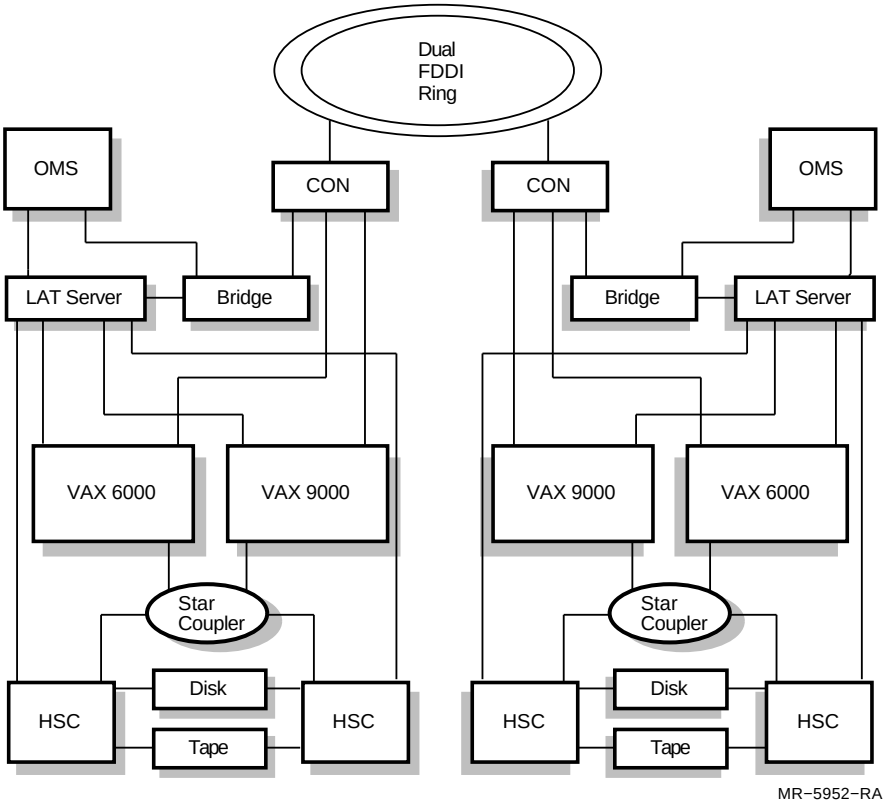

Figure 2–1 Multisite VAXcluster System with OMS

The OMS software was designed so that the secondary platform can automatically assume the role of primary platform if the current primary platform fails. By default, the failed platform assumes the secondary role when it becomes fully functional again.

If communication is lost between the two sites, each platform automatically assumes the role of primary platform for its own site until one site is taken off line.

The OMS software was also designed so that the two platforms can interchange their roles on demand. For example, suppose that a new version of the OMS software is released. The current secondary platform can be upgraded first. Primary platform responsibility can be moved to the upgraded platform, and then the other platform can be upgraded. It is also desirable to interchange roles on demand when it is necessary to perform preventative maintenance on the current primary platform.

Special OMS code was developed to integrate five software entities into the OMS management environment:

- VAXcluster Console System (VCS)
- Data Center Monitor (DCM)

- Configuration Discovery Module (CDM)
- Terminal Server Manager (TSM)
- DECmcc

This management environment provides the following functionality for a multisite VAXcluster configuration:

- The software on each OMS system can provide a logical diagram of the entire VAXcluster configuration. This is sometimes called an *automatic configuration audit*.
- OMS software extends VCS functionality throughout a multisite VAXcluster configuration. OMS manages console functions, such as booting and shutting down nodes at each site. Presently, an OMS system is limited to managing up to 32 nodes (optionally including another OMS system).
- OMS monitoring provides for the automatic reporting of significant anomalous events related to resources such as disk devices and queues. It can also report anomalous events related to applications.
- If a failure or disaster occurs, OMS software supports automatic failover and assumption of primary platform responsibility appropriate to the event. It also supports transferring primary platform responsibility from one OMS system to another on a demand basis.
- Depending on the event and the configuration, an OMS system can be programmed to implement recovery procedures specific to the configuration and application environment.

For further information on disaster-tolerant VAXcluster systems, consult the following sources:

- R. G. Davis, "Introduction to FDDI Concepts," *VAXcluster Systems Quorum*, Volume 7, Issue 3, February 1992, Part Number EA–P1673–32.
- D. Carr, "VAXcluster Multi–Datacenter Facility: Building Disaster-Tolerant VAXcluster Systems," *VAXcluster Systems Quorum*, Volume 7, Issue 3, February 1992, Part Number EA–P1673–32.
- S. Voba and S. Branam, "VAXcluster Multi-Datacenter Facility Operations Management Station," *VAXcluster Systems Quorum*, Volume 7, Issue 4, May 1992, Part Number EA–P1810–32.
- *Digital Technical Journal*, Volume 3, Number 2, Part Number EY–H876E–DP (Bedford: Digital Press, Spring 1991). The articles in this journal are advanced and should probably be read after the first three articles in this list.

3
Understanding DECnet–VAX Phase IV Executor Pipeline Quota

James B. Frazier
Digital Product Services
Digital Equipment Corporation

Introduction

This article addresses an anomalous behavior discovered in local area network (LAN)-based DECnet–VAX Phase IV environments. This behavior was observed in testing, in-house environments, and customer sites.

This anomaly affects all VAX systems using DECnet–VAX Phase IV, especially those VAX systems using PATHWORKS to serve PCs or DECwindows Motif client/server environments. It appears as inconsistent variations in DECnet I/O rates, receiver overruns, and significant amounts of explicit flow control. The user perceives this behavior as poor performance in VAX-to-PC file transfer times and poor DECwindows Motif client/server performance.

The anomaly is caused by misunderstanding DECnet–VAX Phase IV executor pipeline quota usage and buffer management. Ethernet adapter implementation variations cause further complications.

DECnet–VAX Phase IV performance can be optimized by tuning the number of transmitter buffers to mesh with the speed and window size of the Ethernet receiver.

Recommendations

For the majority of LAN-based VAX systems, set the DECnet–VAX executor pipeline quota to 4032, using the following commands:

```
$ MCR NCP
NCP> set exec pipeline quota 4032
NCP> define exec pipeline quota 4032
```

You must carefully evaluate systems with applications such as DTSEND, DFS, SNA, DTF, and RJE or any other application that depends on deep pipelining

Analyze your systems needs, evaluate the network applications in use, and perform some simple experiments.

Avoid overrunning the receiver. This requires knowledge of the relative CPU speeds and Ethernet adapter speed and buffering capabilities.

For VAX systems communicating exclusively with other VAX systems or other fast CPUs/Ethernet adapters, set DECnet–VAX executor pipeline quota to 4032:

$$\frac{4032}{576} = 7\ transmit\ buffers$$

For VAX systems communicating exclusively with slow PCs or slower CPUs and Ethernet adapters, set DECnet–VAX executor pipeline quota to 1728:

$$\frac{1728}{576} = 3\ transmit\ buffers$$

With the emphasis on VAX systems communicating in a wide area network (WAN) environment, especially when there are uplink and downlink latencies caused by satellite links, the optimal setting for DECnet–VAX executor pipeline quota may need to be larger. Cursory testing of file transfer times between two remote nodes indicated that larger values for pipeline quota were more efficient. This must be determined by testing on a site-by-site basis.

DECnet–VAX Phase IV Executor Pipeline Quota Usage

The DECnet–VAX executor pipeline quota determines the maximum transmit window size, that is, the maximum number of packets that are transmitted before asking the receiver for an acknowledgement (implicit flow control). The maximum transmit window size is determined by dividing pipeline quota by the executor buffer size.

Although Ethernet packets can be 1498 bytes and Fiber Distributed Data Interface (FDDI) packets can be 4468 bytes, the default DECnet–VAX executor buffer size is 576. This is independent of the packet size sent on the wire.

Given the default of 576 for exec_buffer_size:

$$max\_transmit\_window\_size = \frac{exec\_pipeline\_quota}{exec\_buffer\_size}$$

$$initial\ transmit\ window\ size = \left(\frac{max\_transmit\_window\_size}{3 * 2}\right) + 1$$

From there, the network services protocol (NSP) flow control algorithms raise and lower transmit window size between 1 and the maximum transmit window size. The minimum is one buffer and a pipeline quota of 576. The maximum is 40 buffers and a pipeline quota of 23,040. Pipeline quota values larger than 23,040 have no effect.

DECnet–VAX Phase IV Buffers

In DECnet–VAX Phase IV there are several layers of buffers for Ethernet communication. This section describes these buffers.

Ethernet Adapter Buffers

The Ethernet adapter has a pool of buffers. This number is determined by the adapter and the driver and cannot be tuned by a system or network manager.

When incoming packets are dropped because of insufficient space in the Ethernet adapter or insufficient speed of the Ethernet adapter, you get "local buffer errors" or "Device overrun errors" messages. In this case, the Ethernet adapter is not fast enough or does not have enough onboard memory. Design limitations of the adapter or an I/O-bound system where the adapter cannot access main system memory fast enough to empty its internal buffers can cause this situation.

When packets are dropped because of insufficient buffering in the driver on the main system, you get a "System buffer unavailable" message. This usually means the CPU is busy at a high interrupt priority level (IPL), so the driver cannot service interrupts and empty its buffers. This condition can be caused or aggravated by the system's total environment, such as memory usage, disk usage, maladjustment of system parameters, runaway applications, CPU speeds, and so on.

Since you cannot increase the Ethernet adapter buffer pool, the only actions you can take in this case is to use a different (faster or more buffer space) Ethernet adapter on the receiving node, tune the OpenVMS system to best advantage, understand the implications of memory and disk loading, and limit the *bursts* of packets other systems transmit to the the overrun system. For DECnet–VAX Phase IV, lowering the executor pipeline quota on transmitting nodes lowers the number of packets the overrun system needs to handle at a given time, making communication smoother and more efficient.

Line Receive Buffers

The next layer of buffers is a pool of buffers held for DECnet–VAX by the Ethernet driver. They specify the length of the line's receive queue. You can tune this pool of buffers by adjusting the setting of the line receive buffer count in the range of 1 to 32.

When a significant user buffer unavailable count occurs, packets are dropped by the Ethernet driver because DECnet is not processing them fast enough.

If the user buffer unavailable circuit counter increments, DECnet is not processing packets fast enough. When the user buffer unavailable line counter increments, it can be any Ethernet application that is not processing packets fast enough.

You can try to limit the user buffer unavailable count by increasing the line receive buffer count or by limiting the *burstiness* of transmission by limiting the executor pipeline quota of transmitting nodes.

Window Size

DECnet–VAX Phase IV provides a 7-buffer (hard-coded) receive window per connection (link). You cannot tune the window size, which specifies the maximum number of frames that may be received before the data is transferred from system buffers to process buffers. The real limit can be lower. When two packets arrive without an associated receive I/O request packet (IRP) queued, a *backpressure* congestion control message is sent back telling the sending system to terminate the logical link. This is the classic case of a high-powered sending system overdriving a small one. Also, if the flow control messages are dropped, data overruns occur. Overruns require retransmissions and incur further delays.

Ethernet Adapter implementation variations

The Ethernet adapter on VAX 4000 series systems is second generation Ethernet chip (SGEC1)-based. The SGEC is much faster than any previous Ethernet adapter. The transceiver turnaround time for the SGEC is 9.6 microseconds, the minumum allowed delay between transmitted packets for Ethernet. Most other Ethernet adapters are based on some version of the local area network controller for Ethernet (LANCE2) chip. The fastest implementation of the LANCE chip is the DEMNA, which has a transceiver minimum interpacket gap of approximately 10.6 microseconds.

The other major difference between the SGEC and the LANCE is that the SGEC defers transmits if it detects carrier before attempting to transmit, but, when it is ready to transmit, it transmits even if carrier is present. The LANCE does the same thing, except, when it is ready to transmit, it defers if carrier is present.

The LANCE cannot turn around its transceiver from receive to transmit within the allotted 9.6 microseconds interpacket gap and is, therefore, required to defer when it detects carrier on the wire. This means that the LANCE generally does not invoke its collision backoff algorithm.

An Ethernet adapter that is ready to transmit after the 9.6-microsecond interpacket gap is supposed to transmit when ready, even if it detects a carrier on the wire. This causes a collision, and the Ethernet adapter invokes its transmit backoff algorithm. The backoff algorithm generates a random number of microseconds that the adapter waits before transmitting again. This is how fair arbitration happens on the LAN.

LANCE-based Ethernet adapters, instead, defer the transmission and retry some fixed number of microseconds later. This can bias the arbitration on a given LAN. Prior to SGEC-based adapters, monitoring a LAN showed an artificially low number of collisions. Monitoring a LAN with a healthy population of SGEC-based adapters showed a higher number of collisions, which seems bad, but is, in fact, desirable.

It is also possible for the opposite to be true — a LAN with a heavily used SGEC and a number of lightly used LANCEs may show an abnormally low number of collisions, because the SGEC locks out the other nodes while it is transmitting. This is typically evident on a network consisting of one SGEC and one LANCE. There should be no collisions. Lowering the executor pipeline quota on transmitting nodes reduces the chance of an SGEC locking out other nodes.

Problem Scenarios

Assume you have a VAX 4000–500 system providing PATHWORKS services to a 286-based PC with an older Ethernet adapter that has only five onboard buffers. On the VAX 4000–500, DECnet–VAX Phase IV executor pipeline quota is set to 10000. This is a popular value, inherited from older DECwindows implementations. This yields 17 transmit buffers:

$$\frac{10000}{576} = 17.XXX$$

The PC user initiates a file copy from the VAX system. The VAX transmits all 17 buffers before asking for an acknowledgement. The PC system is unable to empty the adapter buffers fast enough. Because its Ethernet adapter is much slower than the SGEC, it cannot turn the transceiver around in 9.6 microseconds to send an XOFF message. Twelve of the 17 transmitted packets are dropped and must be resent. On the resend, the same thing happens, and more packets are dropped. The NSP protocol adds a minimum of 4 seconds to each retransmitted packet. NSP has a 3-second delayed ACK timer, so the minimum retransmission has to be greater than the time spent delaying before acknowledging. This situation results in slow file copies.

Another problem scenario assumes you have a VAX 6000–640 as a DECwindows Motif client for a VAXstation 3100 DECwindows Motif server. The VAX 6000–640 system with executor pipeline quota set to 10000 to 20000 sends bursts of packets to the VAXstation 3100 causing excessive flow control and retransmissions. The DECwindows Motif performance is perceived as slow.

Configurations and mixes of systems on a given LAN can lead to a situation where a LAN monitor shows high utilization and modest collision rates and users see poor performance. Actually, most of the traffic is retransmissions of dropped packets and excessive flow control messages.

Test results

In the DECnet tests of OpenVMS systems to small PC systems, optimal file transfer time occurred with pipeline quota settings between 1728 (three buffers) and 3455 (five buffers), providing twice the speed over a setting of 3456 (3456 allows six buffers). The decrease in performance at six buffers was caused by data overruns on the PC side, which led to retransmissions after the retransmission timer expired (the retransmission timer expired after 8 seconds). Variations of the pipeline quota between 1728 (three buffers) and 3455 (five buffers) provided no significant variation in response time.

It was also found that the Digital DEPCA Ethernet adapter had enough buffers (45) to handle any amount of data sent to it without data overrun. With the DEPCA, no significant variation in response time was observed when the pipeline quota was varied between 1728 (three buffers) and 23,040 (40 buffers). With the larger pipeline quota settings, the peak Ethernet utilization increased, but the response times varied only slightly. Therefore, for Ethernet file transfers to a PC, larger settings of the pipeline quota appear to only increase peak Ethernet utilization, while not significantly affecting the time required to transfer a file. This is not a desirable characteristic.

With larger pipeline quota settings, a more bursty behavior was observed. A pipeline quota of three buffers provides enough buffering on the Ethernet to provide near optimal performance. The transmitter requests an acknowledgement and blocks itself after transmission of three packets, which provides implicit flow control. This is much more efficient than XON/XOFF flow control (explicit flow control), as shown in Figures 3–1 and 3–2.

Because of the high speed of Ethernet, the request-response type message required after sending the last packet does not have a significant impact on response time. The optimal setting of transmit pipeline quota for VAX Ethernet communication to PCs is in the range of 1728 to 3455. Larger settings of the transmit pipeline quota may be desirable for asynchronous or synchronous line communication depending on line speed (the transmit pipeline quota setting in DECnet–VAX Phase IV is a per node setting that is used for all transmission lines on that node).

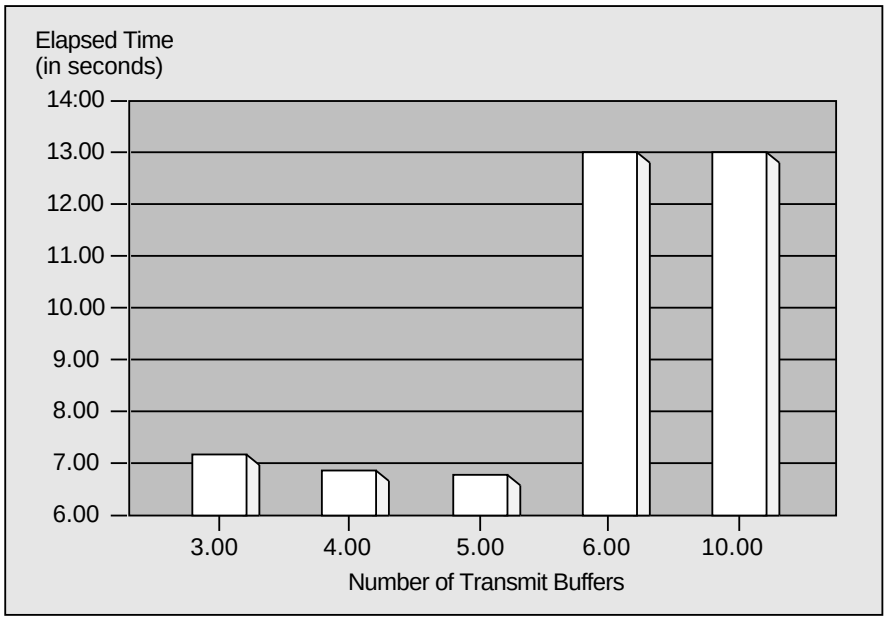

Figure 3–1 DECnet 1-Megabyte File Transfer with a Third-Party Ethernet Board

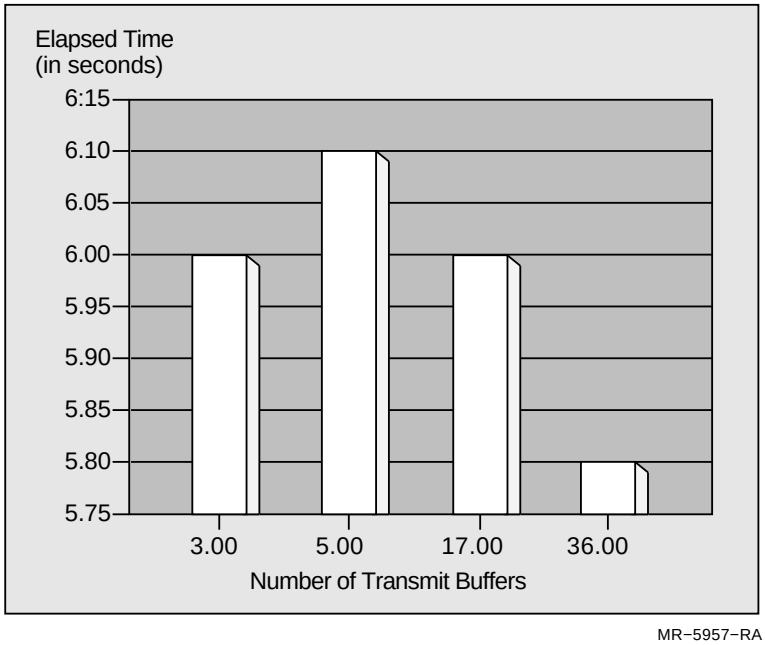

Figure 3–2 DECnet 1-Megabyte File Transfer with a DEPCA Ethernet Board

In a DECwindows Motif client/server environment of large VAX systems to VAX workstations, a pipeline quota setting of 4032 (seven buffers) gave optimal performance. With pipeline quota settings of 10000 (17 buffers) or higher, a more bursty behavior was observed. DECwindows Motif display times were many times slower, and retransmissions and explicit flow control traffic consumed nearly 25 percent of the LAN segment bandwidth.

In the DECnet–VAX tests, once an overrun occurred on a logical link, DECnet–VAX reduced the pipe size used for that connection, so overruns did not recur on that connection. It was observed that each time a logical link was established, it required about 45 seconds for DECnet–VAX to get the pipe size reduced to resume proper communication. DECnet–VAX lowers the pipe size used for a connection dynamically, based on retransmissions and flow control events. If connections are continually established and terminated, the time required to dynamically reduce the pipe size is continually noticed. In the case of receiver overruns, the flow control events are lost, so retransmissions are required. The transmitter gently increases the transmit window size as transmissions occur to the maximum transmit window size, without any problems. When the receiver again signals to terminate or becomes overrun, the transmit window size adjustments recur.

Analysis

Matching transmit buffer pipelines to receive pipeline depth and speed eliminates explicit flow control messages, eliminates dropped packets caused by overruns, reduces bursty network behavior, and improves overall network efficiency and performance.

Setting DECnet–VAX executor pipeline quota in the range of 1728 to 4032 provides near optimal performance for a wide range of configurations, without any performance penalties.

Changing the value of DECnet–VAX executor pipeline quota is dynamic, so that testing can be easily and safely performed on a live system. Logical links that are established do not pick up the changes in executor pipeline quota dynamically. However, changes to executor pipeline quota are picked up by all newly established links. In the problem environments, the DECnet–VAX dynamic transmission pipe control is wasted as links are short lived.

Low collision rates on a busy LAN may be undesirable. A growing population of systems with SGEC-based Ethernet adapters raises the observed collision rate, and this is desirable, since the arbitration is more even.

Some busy LANs or LAN segments are actually wasting considerable bandwidth on retransmissions and excessive explicit flow control traffic. Simply monitoring a LAN/LAN segment and measuring the packet rate, collision rate, and so on, does not give the entire picture. You must use

a systems approach of considering the entire population of nodes and evaluating the capabilities and uses of each type of system.

Other Protocols

Other protocols use the Ethernet adapter through the VCI interface; before VMS Version 5.4–3, they used the FFI interface. This is a side door into the Ethernet adapter driver, bypassing the QIO interface and the system overhead expected with QIOs.

Local area disk/local area system transport (LAD/LAST) has an extensive congestion detection and rate policy, which prevents receiver overruns. The LAT implements flow controls on a per connection basis. Network Interconnect Systems Communications Services (NISCS) uses a maximum of eight transmit buffers on pre-VMS Version 5.4–3 systems. On VMS Version 5.4–3 and later systems, NISCS queries the Ethernet or FDDI adapter for the number of adapter buffers in use. NISCS uses that figure to establish the number of transmit buffers. DECnet–ULTRIX has a default value of 4096 for pipeline quota, which is adequate.

blank page

Additional VAXcluster Information

4
R1 and S1 Revision Management Level

Kathy Thomas
VAXcluster Systems and Support Engineering
Digital Equipment Corporation

Revision Management Level R1 provides support for:

- VAX 9000 Model 110 system
- VAX 9000 Model 3*XX* system
- VAXft 410 system
- VAXft 610 system
- DEMFA adapter
- DEFCN adapter
- RA71, RA72 disk drive
- RF30, RF31, RF31F, RF71, RF72 Integrated Storage Elements (ISEs)
- TF857, TF837 tape subsystems
- HSC60, HSC90 subsystems
- KFQSA adapter
- VMS Version 5.4–3 operating system

Revision Management Level S1 provides support for:

- VAX 6000–6*XX* system
- VAX 4000–500 system
- MicroVAX 3100 Model 30, 40, 80 systems
- RF35, RF73 ISEs
- KMFSA–BA adapter

- DEC Performance Solution (DECps) software
- VMS Version 5.5 operating system

It is recommended that you upgrade your VAXcluster system to the latest revision as soon as practical and over as short a period of time as possible. Although we attempt to maintain VAXcluster system functionality and integrity during upgrades, we cannot guarantee it in all cases.

Table 4–1 summarizes the existing revision levels. Table 4–2 details the applicable versions for individual VAXcluster components within the revision levels, beginning with Level N1.

Revision levels listed in this document are minimum acceptable revisions for products to function reliably in a VAXcluster environment with the current version of the VMS operating system. These revision levels do not reflect the subsequent revisions from Manufacturing or the latest engineering change order (ECO) revision from Engineering, unless these subsequent revisions create a new minimum acceptable revision for VAXcluster systems.

For information about revision levels or changes for components within Table 4–2 that were made available after the *Quorum* print date, contact your Digital Customer Services representative. For further information on Revision Management Levels R1 and S1, contact your Digital Customer Services representative.

Table 4–1 Summary of Revision Management Levels

Revision Level	Feature
K1	SA600 Storage Array, RA70 Disk, HSC Software Version 3.70, VAX 62X0, VAX 88X0, CISCE/CINLE 24-Node Star Coupler, VMS Version 5.0, CIBCA–B VAXBI-to-CI Interface, Local Area VAXcluster Systems
L1	VMS Version 5.1, VAX 6300, DESQA, MicroVAX 3300/3400, MicroVAX 3800/3900, VAXstation 3100, VAXserver 3100, RA90, SA70, SA650, TA90, HSC40
M1	VMS Version 5.2, VAX 6000–400, DEBNI, RV20, RV60, RV64, ESE20
N1	VMS Version 5.4, VAX 9000–200, VAX 6000–500, VAX 4000–300, VAXserver 4000–300, CIXCD, DEMNA, KDM70, TA90E, RA90, RA92
P1	VMS Version 5.4-2, VAX 9000–400, VAXft 310, VAX 4000–200, VAXserver 4000–200, KFMSA, TA91

Table 4–1 (Cont.) Summary of Revision Management Levels

Revision Level	Feature
R1	VMS Version 5.4–3, VAX 9000–110, VAX 9000–300, VAXft 410, VAXft 610, DEMFA, DEFCN, RA71, RA72, RF30, RF31, RF31F, RF71, RF72, TF857, TF837, HSC60, HSC90, KFQSA
S1	VMS Version 5.5, VAX 6000–600, MicroVAX 3100 Model 30, MicroVAX 3100 Model 40, MicroVAX 3100 Model 80, VAX 4000–500, RF35, RF73, KFMSA, DECps

Table 4–2 Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
VAX–11/750 Computer System	60,90 60,98 60,9C	60,90 60,98 60,9C	60,90 60,98 60,9C	60,90 60,98 60,9C
TU58 41 VAX–11/750 Console (BE–FK94?–ME)	A	A	A	A
VAX–11/780 Computer System	8,8B Note 2	8,8B Note 2	8,8B Note 2	8,8B Note 2
RX1 VAX–11/780 Standard Console (AS–T213?–ME)	R	R	R	R

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.) (Notes 2 & 3)	Revision Level N1	P1	R1	S1
RX41 VAX–11/780 Europe RD Console (AS–T215?–DE)	R	R	R	R
RX4 VAX–11/780 Remote Console (AS–T216?–DE)	R	R	R	R
VAX–11/785 Computer System	3,3B Note 2	3,3B Note 2	3,3B Note 2	3,3B Note 2
RX1A VAX–11/785 Console (AS–T793?–ME)	P	P	P	P
VAX 8600 CPU Kernel (KA86-A)	L	L	L	L
VAX 8650 CPU Kernel (KA86-B)	D	D	D	D
VAX 8600/8650 Console with diag MT (BB-FI16?-DE)	T	T	T	T
VAX 8600/VAX 8650 Console RL02 (BC-FI17?-ME)	T	T	T	T
VAX 8600/8650 Console with diag RL02 (BC-FI18?-DE)	T	T	T	T

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.) (Notes 2 & 3)	Revision Level N1	P1	R1	S1
VAX 8200 CPU Kernel (821B) (822B)	B	B	B	B
VAX 8250 CPU Kernel (824B) (825B)	B	B	B	B
VAX 8300 CPU Kernel (831B) (832B)	B	B	B	B
VAX 8350 CPU Kernel (834B) (835B)	B	B	B	B
VAX 8200/VAX 8250 /VAX 8300/VAX 8350 Console Flp (BL-FG81?-ME)	P	S	S	S
VAX 8200/8250/8300/8350 Complete Diag (1600 BPI MT) (BB-FG87?-DE)	V	Y	Y	Y
VAX 8200/8250/8300/8350 Diag Super + Auto (BL-FG79?-ME)	U	W	W	W
VAX 8200/VAX 8250 /VAX 8300/VAX 8350 Util Prog Flp (BL-FG80?-ME)	V	Y	Y	Y

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
VAX 9000–110 System	–	–	D	E
VAX 9000–210 System	C	E	E	E
VAX 9000–3XX System	–	–	D	F
VAX 9000–4XX System	–	E	E	F
VAX 9000–4XX CPU Kernel (KA940)	–	–	–	E
VAX 9000 Console Image TK5O (AQ-PAKH?–ME)	A	B	B	B
VAX 9000 Utility and Microcode (AQ–PAKJ?–ME)	A	B	B	B
VAX 9000 Licensed Diag (AQ–RAKK?–DE)	A	B	B	B
VAX 9000 Field Service SDD (AQ–PBEG?–AE)	–	B	B	B
VAX 9000–110/3XX Microcode and Logic files (AQ–PH5X?–ME)	–	–	A	A
VAX 9000–110/3XX CIS Upgrade (AQ–PHN4?–ME)	–	–	A	A

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.) (Notes 2 & 3)	Revision Level N1	P1	R1	S1
VAX/VAXserver 62XX CPU Kernel (62AMA–Y), (62AMB–Y) (62AMN–Y), (62AMP–Y)	A	A	A	A
VAX 6200/VAX 6300 Complete Diag Set 16MT9 (BB–FK03?–DE)	K	L	–	–
VAX 6200/VAX 6300 Console TK50 (AQ–FJ77?–ME)	K	L	–	–
VAX 6200 EE Prom Patch TK50 (AQ–FJ98?–ME)	F	F	–	–
VAX/VAXserver 63XX CPU Kernel (63AMB–Y), (63AMP–Y)	A	A	A	A
VAX 6300 Complete Diag Set 16MT9 (BB–FK65?–DE)	F	F	–	–
VAX 6300 Console TK50 (AQ–FK60?–ME)	F	F	–	–
VAX 6300 Console Patch TK50 (AQ–FK97?–ME)	D	E	–	–

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.) (Notes 2 & 3)	Revision Level N1	P1	R1	S1
VAX/VAXserver 64XX CPU Kernel (64AMA–Y), (64AMP–Y)	A	A	A	A
VAX 6400 Complete Diag 16MT9 (BB–FK89?–DE)	E	F	–	–
VAX 6400 Console TK50 (AQ–FK87?–ME)	E	F	–	–
VAX 6400 Complete Diag TK50 (AQ–FK88?–DE)	E	F	–	–
VAX 6400 Console Patch (AQ–PBD2?–ME)	B	C	–	–
VAX/VAXserver 65XX CPU Kernel (65*MA–X), (65*PA–X)	A	A	A	A
VAX 6500 Console CD-ROM (AI–PDYQ?–BE)	–	D	–	–
VAX 6500 Complete Diag CDROM (AI–PDZZ?–BE)	–	D	–	–
VAX/VAXserver 6600 CPU Kernel (66***–XE/XJ)	–	–	–	A

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.) (Notes 2 & 3)	Revision Level N1	P1	R1	S1
VAX 6000 Complete Diag CD-ROM (AG–PDWW?–RE)	–	–	F	H
VAX 6000 Console CD-ROM (AG–PDWV?–RE)	–	–	F	H
VAX 6000 Complete Diag TK50 (AQ–PDWX?–DE)	–	D	F	H
VAX 6000 Console TK50 (AQ–PDYP?–ME)	–	D	F	H
VAX 6500 Complete Diag 16MT9 (BB–PDWY?–DE)	–	D	F	H
VAX 8530 CPU Kernel (851BA–Y)	H	H	H	H
VAX 8550 CPU Kernel (855BA–Y)	H	H	H	H
VAX 8700/VAX 8810 CPU Kernel (871BA)	E	E	E	E
VAX 8800/VAX 8820N CPU Kernel (882BA)	F	F	F	F

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
VAX 8500/VAX 8530 /VAX 8550/VAX 8700/VAX 8800 Console Media (BT–ZMAAD–C3)	40	42	42	42
VAX 8500/VAX 8530 /VAX 8550/VAX 8700/VAX 8800 Diag Set (ZM920–C3)	40	42	42	42
VAX 8820/VAX 8830/VAX 8840 CPU Kernel (885BA)	C	D	D	D
VAX 8820/VAX 8830/VAX 8840 Console with Diag TK50 (AQ–FJ79?–DE)	F	F	F	F
VAX 8820/VAX 8830/VAX 8840 Console TK50 (AQ–FJ80?–ME)	F	F	F	F
VAXft 310 CPU Kernel (52AAA–X), (52BAA–X)	–	B	B	B
VAXft 410/610 CPU Kernel (55*AA–X)	–	–	A	A
MicroVAX/VAXstation II CPU Kernel (630QB/630QE /630QY/630QZ)	A	A	A	A

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.) (Notes 2 & 3)	Revision Level N1	P1	R1	S1
MicroVAX/VAXstation /VAXserver 3500/3600 CPU Module (M7620)	A	A	K	K
MicroVAX/VAXstation /VAXserver 3300/3400 CPU Kernel (640QS)	A	A	A	A
MicroVAX/VAXstation /VAXserver 3800/3900 CPU Kernel (655QF/655QS)	A	A	A	A
MicroVAX/VAXstation /VAXserver 3100 CPU Kernel (KA41-A)	A	A	A	A
MicroVAX/VAXstation 3100 Model 30/40 CPU Kernel (KA45)	–	–	–	A
MicroVAX/VAXstation 3100 Model 80 CPU Kernel (KA47)	–	–	–	A
VAX/VAXserver 4200 CPU Module (M7626)	–	A	A	C

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
VAX/VAXserver 4200 Console Firmware	–	–	–	3.7
VAX/VAXserver 4300 CPU Module (L4000)	A	C	C	C
VAX/VAXserver 4300 Console Firmware	–	–	V3.7	V3.7
VAX/VAXserver 4500 CPU Module (L4002)	–	–	–	A
VAX/VAXserver 4500 Console Firmware	–	–	–	V4.1
VAX–11/750 Adapter to CI (CI750)	F,J Note 3	F	F	F
SBI Adapter to CI (CI780)	F,K Note 3	H	H	H
Microcode for CI750, CI780, CIBCI (CI780.BIN)	8.7	8.7	8.7	8.7
VAX BI-to-CI Interface (CIBCA–A)	D	D	D	D
VAX BI-to-CI Adapter for VAX 85XX/8700/8800 (CIBCI)	B,D	B	B	B

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.) (Notes 2 & 3)	Revision Level N1	P1	R1	S1
VAX BI-to-CI Adapter for VAX8200/8300 (CIBCI)	C,E	C	C	C
VAX CIBCA Microcode Update Flp (B–FJ11?–ME)	M	M	M	M
CIBCA–BIN Microcode	7.5	7.5	7.5	7.5
VAX BI-to-CI Interface (CIBCA-B)	A	A	A	A
VAX CIBCA–BA Microcode Update Flp (BL–FK14?–ME)	E	E	E	E
CIBCB–BIN Microcode	5.2	5.2	5.2	5.2
CISCE Star Coupler Expander	A,B Note 4	A,B Note 4	A,B Note 4	A,B Note 4
CINLE–AA, –AB CI7XX Upgrade	A Note 5	A Note 5	A Note 5	A Note 5
CINLE–BA, –BB HSC CI Upgrade	B Note 5	B Note 5	B Note 5	B Note 5
CIXCD–AA (CI-to-XMI interface for VAX 9000)	B	B,D Note 6	D	D

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
CIXCD–AA CIXCD.BIN Microcode	1.04	2.02,2.03 Note 6	2.03	45 Note 7
CIXCD–AA SHOW CLUSTER RP_REVIS	24	42,43 Note 6	43	45
CIXCD–AB (CI-to-XMI Interface for VAX 6000)	B	B	B	B
CIXCD–AB CIXCD.BIN Microcode	1.04	2.02	2.02	45 Note 7
CIXCD–AB SHOW CLUSTER RP_REVIS	24	42	42	45
DEBNA NI Interface	F4,H4	F4,H4	F4,H4	F4,H4
DEBNI NI Interface	C1	C	C	C
DELUA NI Interface	F1	F1	F1	F1
DEUNA NI Interface	E	E	E	E
DELQA NI Interface	D3,E4	D3,E4	D3,E4	D3,E4
DEQNA NI Interface	K,N	K,N	K,N	–
DESVA NI Interface	A	A	A	A

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.) (Notes 2 & 3)	Revision Level N1	P1	R1	S1
DESQA NI Interface	B	B	B	B
DEMNA NI Interface	D	D	D	D
DEMNA Microcode	6.03	6.06	6.06	6.06
DEMFA NI Interface	–	–	A02	A02
DEMFA Microcode	–	–	1.2	1.2
DEFCN–A*/B* DECconcentrator 500	–	–	A	A
DEFCN Microcode	–	–	3.0	3.0
RA60–** — 205 MB DSA Disk Drive	A8,A9	A8,A9	A8,A9	A8,A9
RA60–** Disk Drive Reported HV	1	1	1	1
RA60–** Disk Drive Reported MC	5	5	5	5
RA70–** — 280 MB DSA Disk Drive	K6	K6	L9	L9
RA70–** Disk Drive Reported HV	7	7	7	7

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
RA70–** Disk Drive Reported MC	79	79	83	83
RA71 700 MB DSA Disk Drive	–	–	A	A
RA72 1.0 Gbyte DSA Disk Drive	–	–	A	A
RA80–** — 121 MB DSA Disk Drive	Note 1	Note 1	Note 1	Note 1
RA81–** — 456 MB DSA Disk Drive	Note 1	Note 1	Note 1	Note 1
RA81–** Disk Drive Reported HV	8	8	8	8
RA81–** Disk Drive Reported MC	8	8	8	8
RA82–** — 622 MB DSA Disk Drive	C	C	C	C
RA82–** Disk Drive Reported HV	2	2	2	2
RA82–** Disk Drive Reported MC	49	49	49	49

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
RA90–** — 1216 MB DSA Disk Drive (Long Arm) 70–23899–01	J	S	S	S
RA90–** Disk Drive (Long Arm) Reported HV	17	18,25 Note 8	18,25 Note 8	18,25 Note 8
RA90–** Disk Drive (Long Arm) Reported MC	25	26	26	27
RA90–** — 1216 MB DSA Disk Drive (Short Arm) 70–23899–02	A	B	B	B
RA90–** Disk Drive (Short Arm) Reported HV	49	50	50	50
RA90–** Disk Drive (Short Arm) Reported MC	25	26	26	27
RA92 1506 MB Disk Drive	A	B	B	B
RA92–** Disk Drive Reported HV	81	82	82	82

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
RA92–** Disk Drive Reported MC	25	26	26	27
RF30 150 MB ISE	–	–	H	H
RF31 381 MB ISE	–	–	K	K
RF31F 200 MB ISE	–	–	A	A
RF35 852 MB ISE	–	–	–	B
RF71 400 MB ISE	–	–	F	F
RF72 1000 MB ISE	–	–	E	E
RF73 2.0 Gbyte ISE	–	–	–	A
RF73 Drive Firmware	–	–	–	T329
TA78–** Tape Subsystem	D	D	D	D
TA79–** Tape Subsystem	A	A	A	A
TA81–** Tape Subsystem	E	E	E	E
TA90–** Tape Subsystem	C4	C4	C4	C4
TA90E–** Tape Subsystem	A1	A	A	A

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
TA91–** Tape Subsystem	–	A	A	A
TF837 Magazine Tape Subsystem	–	–	A	A
TF857 Magazine Tape Subsystem	–	–	A	A
KDM70 SI Disk and Tape Controller	A	A	A	A
KDM70 Microcode	2.2	2.4,2.5 Note 10	2.4,2.5 Note 10	3.0
KFMSA–AA DSSI-to-XMI adapter	–	A	A	A
KFMSA–AA Microcode	–	3.14	3.14	3.14
KFMSA–AA SHOW CLUSTER RP_REVIS	–	D26E	D26E	D26E
KFMSA–BA DSSI-to-XMI adapter	–	–	–	B02
KFMSA–BA Microcode	–	–	–	5.6
KFMSA–BA SHOW CLUSTER RP_REVIS	–	–	–	A4A6

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
KFQSA Q-Bus-to-DSSI Adapter Module	–	–	H4,J5 Note 11	H4,J5 Note 11
HSC40–** CI-based Disk and Tape Controller	A1,A2	A1,A2	B	B
HSC50–** CI-based Disk and Tape Controller	H8,H9 Note 4	H8,H9	J	J
HSC70–** CI-based Disk and Tape Controller	D8,D9 Note 4	D8,D9	E	E
HSC60 CI-based Disk and Tape Controller	–	–	A	A
HSC90 CI-based Disk and Tape Controller	–	–	A	A
HSC40/70, HSC60/90 Software (BN–FNAAH–BK)	5.0A	5.0A	6.0	6.0
HSC50 Software (BE–FNWAF–BK, BE–FN04F–BK)	4.0	4.0	4.1	4.1
RV64 Optical Library Juke Box	A	A,B Note 9	A,B Note 9	A,B Note 9
RV60 Optical Drive	A	B	B	B

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

Table 4–2 (Cont.) Component Revision Levels

Description (Part No.)	Revision Level			
(Notes 2 & 3)	**N1**	**P1**	**R1**	**S1**
RV20 Optical Drive	B	C	C	C
ESE20 Electronic Storage Element	Note 1	Note 1	Note 1	Note 1
ESE20 Drive Reported HV	0	0	0	0
ESE20 Drive Reported MC	17	17	17	17
VAX Supercomputer Gateway (825CC–**)	C	C	C	C
VAXcluster Console System 4-node License (QL–V01A9–PD)	1.3	1.3	1.3	1.3
VAX Performance Advisor (QL–VE5A*-**)	2.1	2.1 Note 12	2.1 Note 12	–
DEcperformance Solution (QL–GX2A*–**)	–	–	–	1.0
VMS Operating System	5.4	5.4–2	5.4–3	5.5

Notes

1. No revision level restrictions.
2. Revision Level 8 (for VAX–11/780) or 3 (for VAX–11/785) is acceptable if the memory is not type MS780–E or MS780–H.
3. VAXcluster systems of less than six nodes may use the lower revision indicated. VAXcluster systems of more than five nodes or greater than 4.5 megabytes per second, use the higher revision. A CINLE upgrade is required for node numbers greater than 16.
4. B for 24 to 32 nodes.
5. A CINLE upgrade is required for all L0101-equiped options in a VAXcluster system with nodes numbered greater than 15.
6. Higher revision needed for the VAX 9000 system with XJA revision D05 and higher.
7. CIXCD.BIN has been changed from decimal to hexidecimal.
8. 17 with HDAs above revision M12. 25 with HDAs below revision N12.
9. Revision B for VAX 9000 systems.
10. 2.4 for VAX 6000 systems. 2.5 for VAX 9000 systems and VMS Volume Shadowing Phase II.
11. J5 for KFQSA-SE/SG modified S-box handle assembly for warm swap functionality.
12. 2.1 plus mandatory update package (MUP).

blank page

VAXcluster Customer Configuration Database Questionnaire

VAXcluster Customer Configuration Database (VCCD) Questionnaire

The Digital VAXcluster Group has an online database of VAXcluster customer configuration data. The purpose of this database is to gather a high-quality statistical sampling of installations. The information will help in identifying progress trends, forecasting future product needs, and providing information to serve our customers better.

Since frequent changes occur at a customer site, it is sometimes difficult to capture these changes in a timely manner. For this database to meet its goals, the information must be accurate and current. The best source of this information is you, the customer.

We encourage you to participate in this update process by completing the attached questionnaire. If you have more than one VAXcluster system, please photocopy and complete the form for each VAXcluster system.

Customer Name _____

Division _____

Street _____

City _____ Phone (___) _____

State/Province _____ Zip _____ - _____

Country _____

Name _____ Title _____

MIS Manager _____ Phone (___) _____

Other Contact _____ Phone (___) _____

Title _____

Number of CI-VAXcluster systems at this site: _____ If there is more than one, please copy the remainder of this questionnaire and complete for each VAXcluster system.

Security of Customer Information (check the appropriate box):

☐ SECRET.....Will *not* be disclosed.
☐ RESTRICTED....May be disclosed to a *limited* internal distribution.
☐ NONE.....*Unlimited* internal distribution.

Northeast	(163)	☐	New York/NJ	(1DG)	☐	Mid-Atlantic	(162)	☐
Southern	(1DF)	☐	Southwest	(1WQ)	☐	South Central	(ASL)	☐
Central	(161)	☐	Western	(160)	☐	East Central	(AZ3)	☐
Europe		☐	GIA		☐			

1. System Manager _____ Phone (___) _____
   Department _____
2. Digital Customer Services Representative _____
   Branch Office _____ Phone (___) _____

Check the appropriate box:

Digital Hardware Contract ☐ H
Digital Software Contract ☐ S
Self-Maintenance ☐ Y
Third-Party ☐ T

Check Yes or No:

VCS-VAXcluster Console System	Yes ☐	No ☐
VAXcluster system on Ethernet	Yes ☐	No ☐
VAX Performance Advisor	Yes ☐	No ☐
VAX Volume Shadowing	Yes ☐	No ☐
VAX RMS Journaling	Yes ☐	No ☐

Is there an LAVc connected to this CI-VAXcluster? (Mixed interconnect) Yes ☐ No ☐

If yes, what is the number of satellite nodes? ___

Average active users ___ Peak number of users ___

VAXcluster operating system: VMS ☐ or ULTRIX ☐ Version Number: _____

CPU Type	Serial No.	Memory Size (MB)	Number of CI Adapters	Number of Star Couplers Connected to This CPU

HSC Type	Serial No.	HSC Type	Serial No.

Digital Drives and Tapes: Enter quantity of each type listed below.

RA60				SA482		TA78	
RA70						TA79	
RA80		1.1	GB	SA550-1		TA81	
RA81		2.4	GB	SA550-2		TA90	
RA82		4.2	GB	SA550-3		TA91	
RA90						TE16	
RA92		1.2	GB	SA600-4		TSV05	
RM05		2.4	GB	SA600-3		TU45	
RM80		4.8	GB	SA600-1		TU77	
RP06		9.7	GB	SA600-2		TU78	
RP07						TU79	
RV20		3.5	GB	SA650-1		TU80	
RV64		6.0	GB	SA650-2		TU81	
		9.5	GB	SA650-3		TU90	
		2.8	GB	SA705			
				SA800			
				SA850			
		120	MB	ESE20-1			
		240	MB	ESE20-2			

Non-Digital Storage Devices:

Disk	Tape	Mfr Name	Model	Storage/MB	Qty
☐	☐				
☐	☐				
☐	☐				
☐	☐				

Software Products: Please check all the products used on this VAXcluster system.

☐ VAX DBMS	V01	☐ Remote System Manager	V05
☐ VAX Rdb/VMS (full development)	V02	☐ VAX Distributed File Server	V06
☐ VAX ACMS	V03	☐ VAX Distributed Queueing Server	V07
☐ VAX SPM	V04	☐ DECintact	V08

Applications: Please check *only* the *top five* applications on the VAXcluster system.

Marketing/Sales/Service

- ☐ Marketing Mgmt Support B13
- ☐ Customer Information Service B14
- ☐ Retail Operations & Channels B15
- ☐ Sales Operations & Comm B16
- ☐ Repair Services B17
- ☐ Wholesale Dist Operations B18
- ☐ Other Dist, Mktg, Sales, Svc B19

Finance and Administration

- ☐ Billing/Project Accounting B20
- ☐ General Ledger/Payables/Rcvg B21
- ☐ Legal/Litigation B22
- ☐ General Administration B23
- ☐ Payroll B24
- ☐ Personnel/Policy Administration B25
- ☐ Purchasing/Procurement B26
- ☐ Other Finance & Admin Bus B27

Engineering

- ☐ Arch Engineering & Const B28
- ☐ Process Engineering & Design B29
- ☐ Electrical Engineering B30
- ☐ Engineering Support B31
- ☐ Mapping B32
- ☐ Mechanical Engineering B33
- ☐ Manufacturing Engineering B34
- ☐ Oil Expl/Production/Mining B35
- ☐ Computer Aided Software Eng B36
- ☐ Other Engineering B37

Insurance

- ☐ Agency Systems B50
- ☐ Claims Processing B51
- ☐ Underwriting B52
- ☐ Insurance Policy Admin B53
- ☐ Other Insurance B54

Brokerage

- ☐ Portfolio Management B55
- ☐ Retail Brokerage B56
- ☐ Trading Systems B57

Banking

- ☐ Demand & Time Deposit Acct B58
- ☐ Foreign Exchange B59
- ☐ Funds Transfer B60
- ☐ Cash Management B61
- ☐ Loan Processing B62
- ☐ Other Whsle & Retail Banking B63

Telecommunications

- ☐ Telecom Intelligent Networks B64
- ☐ Telecom Network Management B65
- ☐ Telecom Operational Support B66
- ☐ Computer Integrated Telephone B67
- ☐ Other Telecommunications B68

Research/Lab

- ☐ Lab Information Mgmt B38
- ☐ Scientific Data Analysis B39
- ☐ Data Acquisition & Control B40
- ☐ Signal Processing B41
- ☐ Scientific Image Processing B42
- ☐ Health & Education B43
- ☐ Other Research/Lab B44

Manufacturing

- ☐ Factory/Industrial Automation B45
- ☐ Manufacturing Decision Support B46
- ☐ Mfg Planning & Ctl Sys/MRP II B47
- ☐ Maintenance/Facilities Mgmt B48
- ☐ Other Manufacturing B49

Generic Applications

- ☐ Application Design & Devel B01
- ☐ Economic/Business Analysis B02
- ☐ Electronic Publishing B03
- ☐ Document Imaging B04
- ☐ Word & Document Process B05
- ☐ Data Network Mgmt B06
- ☐ Modeling/Simulation B07
- ☐ Office Automation/Electronic B08
- ☐ Planning/Budget B09
- ☐ Realtime Computing B10
- ☐ Technical Documentation B11
- ☐ Supercomputing B12
- ☐ Other Generic B99

Note: An envelope is provided inside the front cover for you to return your completed questionnaire.

blank page