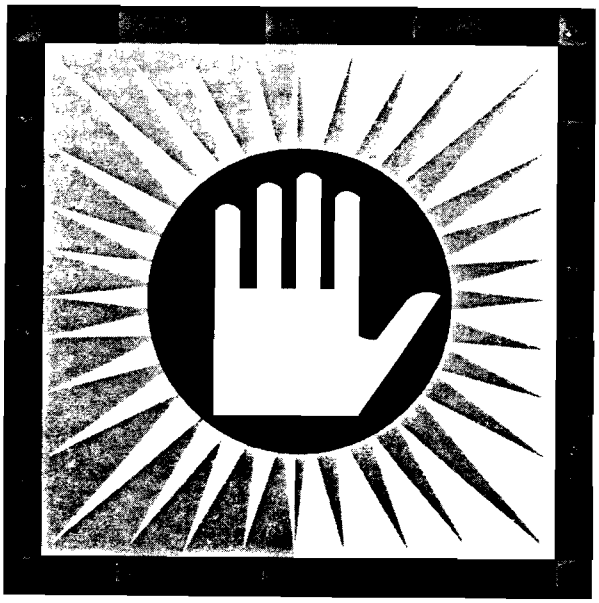
19th Annual HP User Conference and Expo

Interex '93

San Francisco, Moscone Center ■ September 19–23,1993



Tutorial Workbook **T510**

Managing MPE/3000 Systems AVAILABILITY

Rakesh Patel
Hewlett-Packard Co.

Sponsored by Interex, The International Association of Hewlett-Packard Computer Users

Table of Contents

1. Definition and Goal 1
   1.1 Definition of System Availability 1
   1.2 System Availability Goal 1
   1.3 System Availability Model Structure 2

2. Disk as a Factor Of System Availability 3
   2.1 System Disk Failure = 0 Total System Availability 3
   2.2 Localizing the Impact of Disk Failure 3
   2.3 Design Consideration of the System Volume Set 5
   2.4 Partitioning of User Data With User Volumes 5
      2.4.1 Main Advantages of User Volume Sets 6
      2.4.2 Consideration for Planning 6
   2.5 Volume Management Basics 7
   2.6 Designing a User Volume Set 9
      2.6.1 Calculating the Minimum Number of Disks Required 9
      2.6.2 Planning for Data Growth 10
      2.6.3 Arranging Disks in a Volume Set 10
   2.7 Creating User Volumes 12
   2.8 Setting Up Accounting Structures 12

3. Disk Arrays For System Availability 14
   3.1 Recommended Steps for Adding Disk Mechanisms 15
   3.2 What Happens When a Disk Mechanism Fails? 15
   3.3 Replacing a Failed Mechanism 16

4. Disk Mirroring 17
   4.1 Creating a Mirrored Volume Set 19
   4.2 Creating Accounting Structures on Mirrored Volume Set 19
      4.2.1 Switching 20
      4.2.2 Resuming Normal Mirroring on a Mirror-Suspended Volume 20
      4.2.3 Disk-related Failure During Normal Mirroring 21
      4.2.4 Points to Keep in Mind During Use of Mirrored Disks 21

5. SPU Switchover 23
   5.1 Planning for switchover setup 23
   5.2 On Detecting the Home's Death 24
   5.3 Synchronizing Directory Structures 25
   5.4 Network and Terminal Switchover 25
      5.4.1 Hardware Configuration 27
      5.4.2 Network Configuration 27

6. Backup For System Availability 29
   6.1 Dedicated Backup 29
      6.1.1 Selecting a Backup Device Based on its Speed 30
      6.1.2 Selecting the Capacity of Backup Devices 30
      6.1.3 Determining the Number of Backup Devices 30
      6.1.4 Determining the Amount of Data to be Backed Up 31
      6.1.5 Considering the Time Allotted for Backup 31
   6.2 Online Backup - With Full System Availability 32

6.2.1 Using TurboStore With Online Facility 32
6.2.2 Using TurboStore With Split Mirrored Disks 32

7. AutoRestart For System Availability 34
7.1 Mini-Dump 34
7.2 Preparing Your System 36
7.3 Determining your Dump-to-Disk Needs 36
7.4 Estimating the Size of the Largest Dump File 37
7.4.1 Using the Estimation Table 37
7.4.2 Using DTODSIZ utility 37
7.5 Determining Mini-Dump Needs 37
7.6 Creating the AutoRestart account structure 38
7.7 Configuring and Managing AutoRestart for System Availability 38

8. Threshold Manager 39
8.1 What are Jumpoff and Fallback Events? 41
8.2 Multiple Activations of JOB_SESSION_CONTROL 41
8.3 How are Logons Possible Despite Activation of JOB_SESSION_CONTROL? 42
8.4 Using Threshold Manager 42

9. SharePlex/iX - System Availability On Disaster 43

10. HP OpenView System Manager 45

A. NEWGROUP - A sample UDC 47

List of Figures

Figure 1-1. A pyramidal representation of customers' needs for system availability. 2

Figure 2-1. An example configuration of disks and data on an HP 3000. 4

Figure 2-2. Volume states, volume operations, and state transitions. 8

Figure 2-3. Partitioning application data using user volume sets. 9

Figure 2-4a. Single card/link arrangement of disks for a user volume set. 11

Figure 2-4b. Multiple card/link arrangement of disks for a user volume set. 11

Figure 4-1. A simple one card/link arrangement of disks for a mirrored volume set. 17

Figure 4-2. A multiple card/link arrangement of disks in a mirrored volume set. 18

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

1. Definition and Goal

1.1 Definition of System Availability

There are many definitions of system availability. Each person, based on his or her use of computer systems, may define it differently. An enduser will define it as the state in which a computer system continues to provide the application services he or she needs, regardless of the limitations the system hardware or software may impose on the internal operation of the system. To an IT manager, it is the attribute of the computer system that keeps his or her business running despite any failures and limitations encountered during its use. To a system programmer, it is the ability of the operating system to detect and overcome discrepancies in the software and the hardware, either automatically or through timely intervention of the operator. For the purpose of the present discussion, we define system availability in terms of applications, which define the business function for which the system is setup.

In the following sections we present our view of what constitutes system availability and the ways it is achieved in varying degrees. We hope that by the end of this paper the reader will be able to both distinguish varying degrees of system availability and change the degree of system availability when required.

1.2 System Availability Goal

As we mentioned above, applications that "run" the business define the availability goals for a system. Applications can have a varying degree of influence on system availability based on their relative importance to the business. For example, two applications A and B may denote system availability for a system. Say that A has 60% influence on the system availability compared to B, with 40%. The degree to which the system availability is affected is determined by the applications that are nonoperational. For this example, if A becomes nonoperational while B is still unaffected, the system availability is deteriorated by 60%. However, if both A and B become nonoperational at the same time, the system availability is zero. The overriding goal can be summarized as follows:

1. No application should be nonoperational (system availability is 100%).
2. If nonoperational (because the above is not feasible within the constraints of the IT budget), then an application with more influence on the business should become nonoperational less often than applications of lesser influence.
3. The cause for failure of one application should not, if possible, become the cause of failure for another application. In other words, multiple applications should never fail because of the same cause of failure.

1.3 System Availability Model Structure

The degree of system availability necessary on any HP3000 depends on the business needs of its users. Achieving the necessary degree of system availability may involve adding additional software and hardware products to complement the level of system availability already supported by the operating system. Significant gains may be achieved by changes in the operations environment as well.

Figure 1-1 illustrates varying degrees of system availability in the form of a pyramid. At the base of the pyramid is the basic degree of system availability which is intrinsic to MPE/iX. A computer is merely a device in which the data evolves from one form to another and from one state to another. At the lowest degree of system availability, this data, in whatever form or state, must always be valid. The heart of MPE/iX is designed to provide this guarantee. This degree of system availability is often perceived as the most basic need for commercial computing.

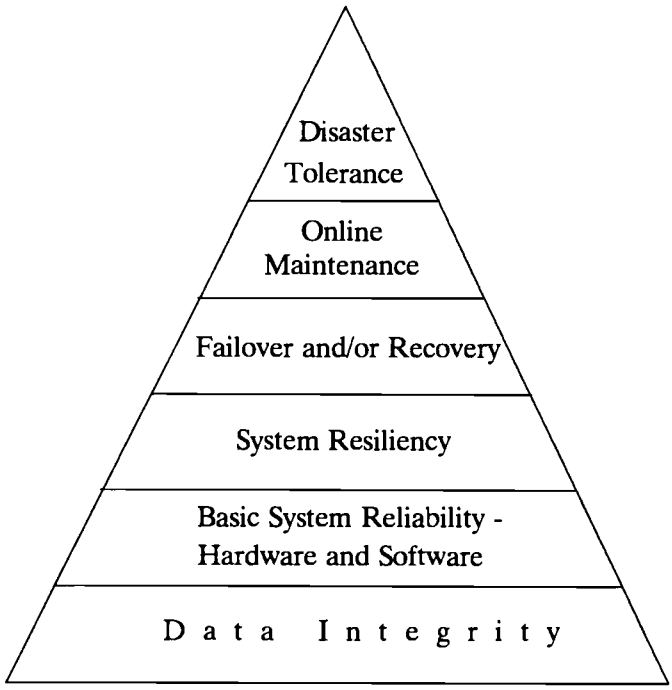


Figure 1-1. A pyramidal representation of customers' needs for system availability.

Like most modern computer systems, the HP3000 is a complicated interconnection of many hardware devices, such as central processors, primary memory, disk and tape drives, terminals and line printers. To provide a minimum degree of system availability on the HP3000, these devices are manufactured and tested for high reliability.

2. Disk as a Factor Of System Availability

Of all devices on a system, disk can be considered as the most important asset. It is the one on which all the permanent business data resides. Furthermore, it is the most widely used peripheral component, being accessed by online users, batch processes, user applications, and system processes. Its failure therefore has dramatic effects on the way the system will function.

To illustrate these points, let's look at Figure 2-1 as a snapshot of a typical system configuration highlighting a disk "farm". The disks are connected to the system through interface cards and cables. A disk is considered failed if the disk itself, the card or the cable fails. A disk will also fail if one of its components such as the controller, the head assembly, or the media fails.

2.1 System Disk Failure = 0 Total System Availability

From the point of view of system availability, let us understand the consequence of the failure of a disk on an HP3000. Generally, when a disk fails, the data it contains is not available and consequently system functions, applications, and users depending on the data cannot operate normally. The degree of system availability affected is based on the dependencies on the unavailable data. If the disk contains the operating system and its related data (i.e., part of the 'MPEXL_SYSTEM_VOLUME_SET'), the failure of the disk will cause the failure of the entire system, thus affecting total system availability. Almost always, the only way out of this situation is to replace the failed disk with another one, recreate the volume set to which the failed disk volume belongs, and reload all of the data back onto the volume set. In the case of a system disk failure, it is necessary to reinstall the operating system (thus recreating the MPEXL_SYSTEM_VOLUME_SET) and then reload the remaining data that resides on the set.

The degree to which the system availability is affected because of a disk failure also depends upon the time it takes to reload all of the data onto a volume set. For a system disk failure, the total reloading time is the sum of the reloading time of the operating system and its related data and the reloading time of all of the user data on the system volume set. As soon as a particular set of data is reloaded, anyone depending on that data can begin to work. For example, once the operating system and its related data is completely reloaded, the operating system can run. Similarly, an application is operational after its data is completely reloaded and the necessary recovery procedures are applied to it. As you can see, the system availability can increase as overall reloading progresses. However, the affected system availability is not totally restored until all the affected data is reloaded.

2.2 Localizing the Impact of Disk Failure

As mentioned above, the degree to which the system availability is affected because of a disk failure depends on the importance of the corresponding data to the system and to the business. If a disk contains any system data, then its failure means zero system availability. If the system volume set contains all of the user data, it is likely that the set is large and the system data is spread across all system volumes. As volume set size increases, so does the likelihood of disk failure.

Application Disk Failure = $100 - \Delta x$ **System Availability**

However, if a disk contains only application data, then its failure does not affect the system, only the application. Therefore, the remaining system availability is greater than zero but less than 100%. The actual value depends on the percentage of the total system availability that the application influences. In the above equation, that share is represented as Δx.

An argument in favor of homing all data on the system volume set relates to performance. Since the volume set is large and the data contained on it is evenly spread across the volumes, the I/O access to the data will also occur equally across the volumes. This decreases the bottleneck on a particular disk and therefore improves the performance. However, this gain in performance can be offset by the system logging process. A single logfile resides on the system master that insures system volume data recovery and integrity. The sheer number of log transactions (I/Os) can bottleneck the path to the system master and consequently degrade performance.

Separating system data and user data by migrating user data onto multiple user volume sets can reduce the risk of zero availability. This is often referred to as partitioning. With partitioning of user data, the reloading effort involved in the case of a disk failure is cut down significantly.

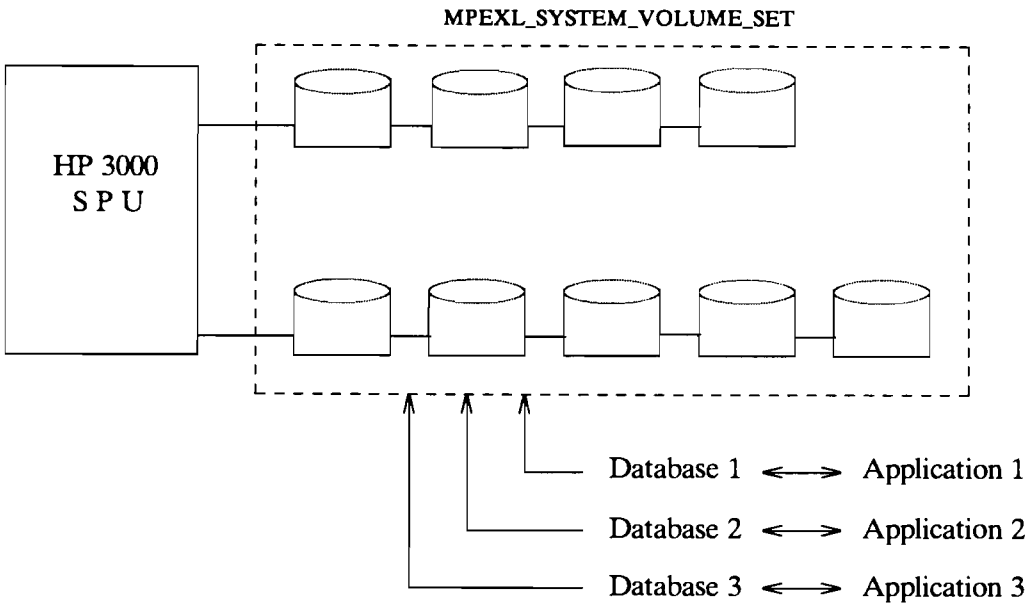


Figure 2-1. An example configuration of disks and data on an HP 3000.

Although the operating system is up and running as soon as the reloading of the operating system and its related data is complete, the system is not available for business until the applications and their data are reloaded. Therefore, the worst case scenario for system availability can happen when all the user data resides on the system volume set. In order to avoid this situation, the user data must be separated from the system data by offloading it onto user volume sets.

2.3 Design Consideration of the System Volume Set

When moving application data to user volume sets, careful consideration should be given to designing the system volume set. One of the key design aspects is the number of disks to configure on the system volume set. To decide, one must understand the total permanent space required for loading the operating system, related permanent files and disk structures. Additionally, the total transient space required for running the system to its intended capacity must be estimated.

Following are some of the data points that will help in evaluating the disk space needs for the system volume set:

- The maximum number of jobs and sessions that will run simultaneously on your system.
- The maximum transient disk space consumed by each job or session when it is active on the system. This may be dependent on the operating system release. For MPE/iX Release 4.0, each job or session will require approximately 2.5MB of transient disk space.
- The permanent disk space required for the operating system, its related files, and the collective subsystems. This is dependent on the operating system release. For MPE/iX Release 4.0, this will require approximately 2.5GB of permanent space.
- The estimated total disk space required for spool files.
- The Transaction Management log file on the master volume consumes 100MB of disk space.
- The additional disk space required for "load fluctuations" and the future growth.

2.4 Partitioning of User Data With User Volumes

User data is everything but system code and system data. It includes flat files and databases. Ideally, from the point of view of system availability, a user data partition should only consist of one application per user volume set. Resource constraints might dictate a cost-effective solution (albeit, at the expense of system availability) and place more than one application on a user volume set. In no case, though, should an application's data be put on more than one volume set.

2.4.1 Main Advantages of User Volume Sets

In this section, the main advantages of partitioning user data are summarized.

- When a disk failure occurs, only the applications residing on the volume set are inaccessible. Therefore, the system availability is only partially affected.
- A disk failure on a user volume set requires only the data of that volume set to be reloaded. Recovery time is shorter and availability of other applications is not affected.
- If all application data is contained on user volume sets, only an install is required if a volume in 'MPE_SYSTEM_VOLUME_SET' fails. No application data has to be reloaded.
- Only limited system availability is affected when backing up a user volume set because only one user volume set is backed up at a time.
- If the system volume set doesn't contain application data, backup of the system volume set is unnecessary and can be avoided. This saves time and costs relating to bookkeeping and storage media.
- If the system volume set doesn't contain application data, the set will consist of fewer volumes. Therefore, dump sizes and associated processing times are reduced.

2.4.2 Consideration for Planning

Before the task of partitioning is undertaken, it is essential to properly plan for the restructuring. Proper planning includes consideration of attributes normally unrelated to system availability, such as I/O performance and data growth rate and size. Neglect of these considerations may lead to bad I/O performance from the partition. Correction of this may require another partitioning, thus affecting data and application availability.

Therefore, it is worth considering the following points when planning for the partitioning of user data.

- A disk failure brings down all applications that depend on the data on the failed disk.
- A disk failure will warrant recreating the associated volume set and reloading the data on that volume set.
- Reloading of data onto a user volume set always means the corresponding data is not available to the applications and to the users.
- A disk failure on a user volume set that contains data of only one application has minimal impact on system availability. Because only one application is affected, reloading time is minimized.
- Performance will improve as the number of volumes in a volume set increases.
- On each user volume set approximately 100 MB of disk space is reserved exclusively for system use. System structures such as the directory, the Volume Set Information Table, and transaction management logfiles reside in this area.

2.5 Volume Management Basics

This section provides a review of the implementation of user volume sets, including user volume set creation and volume set accounting structure creation.

Following are some basics of volume management.

- A volume set is a collection of volumes, each of which must have a unique name within the set.
- A volume is a single disk represented on the system as a logical device.
- A volume class is a grouping of one or more volumes in a set.
- Each volume set on the system must have a unique name.
- Each volume within a set must have a unique name. Two volumes, each belonging to different sets can have the same name. Volume names, when qualified with their set names will be unique across the system.
- The first volume in a set is automatically created and added into the set. It is called the master volume. Subsequent volumes added into the set are called members.
- A volume set is said to be "mounted" and "open" only after the master volume has been mounted and is open.
- A volume set is completely mounted and open only when all volumes belonging to it are mounted and open. All data may not be available from an incompletely mounted volume set.
- A volume set can be explicitly disabled from file system access in which case the set is mounted but not open. Data is not accessible on a closed set.
- A closed volume set can be explicitly opened for file system access. The set becomes mounted and open, thus enabling data accessibility.
- Each volume member incurs negligible disk space overhead due to system data structures.
- Volume sets provide for easy and flexible expandability. The demand for additional disk space on the volume set can be easily met by adding another volume into the set.
- Volutil is the system utility used to manage volumes, except for opening and closing a volume set. The command interpreter commands VSOPEN and VSCLOSE accomplish these functions, respectively.

Figure 2-2 illustrates the operations and resulting volume state transitions involved in volume management.

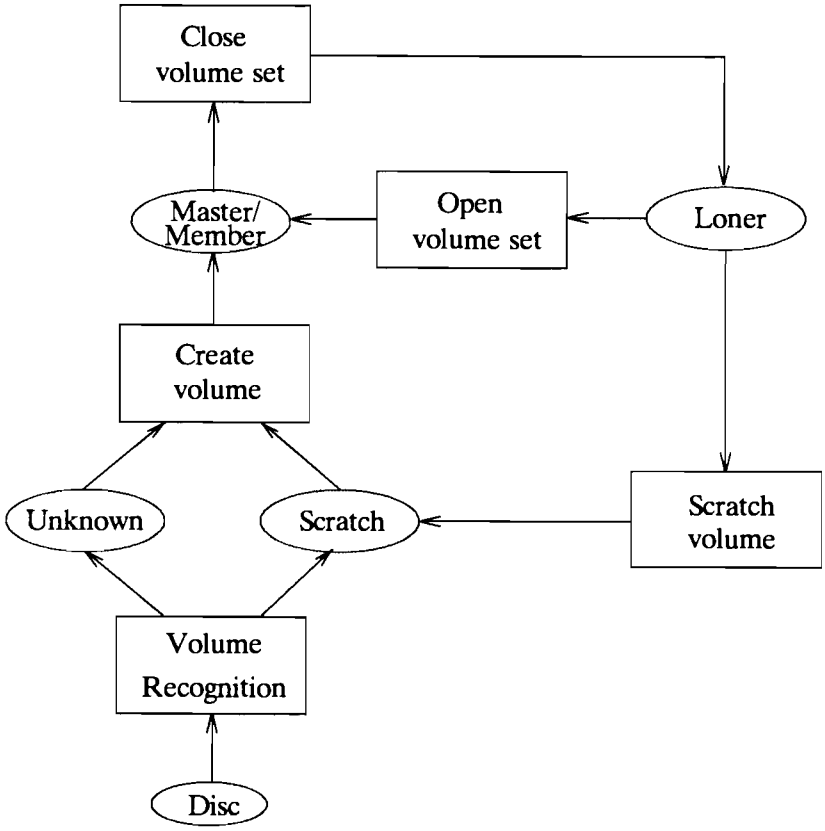


Figure 2-2. Volume states, volume operations, and state transitions.

Let us revisit Figure 2-1 to illustrate a hypothetical data organization. The disk farm consists of only the system volume set (MPEXL_SYSTEM_VOLUME_SET). The set, besides containing the system data, also contains the user data in the form of three important databases numbered 1 through 3. Each database is accessed by a separate application, correspondingly numbered 1 through 3.

Let us review the effect of a disk failure on the system availability for the above data organization. With a single disk failure, some system data is lost, causing the system to halt with system availability reduced to zero. To recover from this situation, the following steps are performed:

1. Replace the failed disk with a new disk. (System availability is 0%)
2. Reset the system and install the operating system from the archived System Load Tape (SLT). (System availability is 0%)
3. Reboot the system. (System availability is 0%)
4. Initialize (add) the remaining volumes in the system volume set. (System availability is 0%)
5. Reinstall all products and subsystems. (System availability is 0%)

6. Reload the application data. Recover the data after it's reloaded. Then, initiate each application. (System availability is from 0% to 100%.)

To improve system availability, let's reorganize the data by setting up user volumes. The size of the system volume set will consequently be reduced. The new data organization is illustrated in Figure 2-3. Here, all disks connected to I/O card 1 comprise the system volume set and those connected to I/O card 2 comprise APPLICATION_1_VOL_SET, which houses Database 1. Disks connected to the I/O cards 2 and 3 are configured into APPLICATION_2_VOL_SET and APPLICATION_3_VOL_SET, respectively. The former set is intended exclusively for Database 2, and the latter one for Database 3. With this configuration, the failure of a disk in one of the user volume sets only affects the availability of the data from that set. That is, only the corresponding application that depends on that database will not function. All other applications will continue to function as before. Note, though, the failure of any of the disks included in the system volume set will still bring the system down.

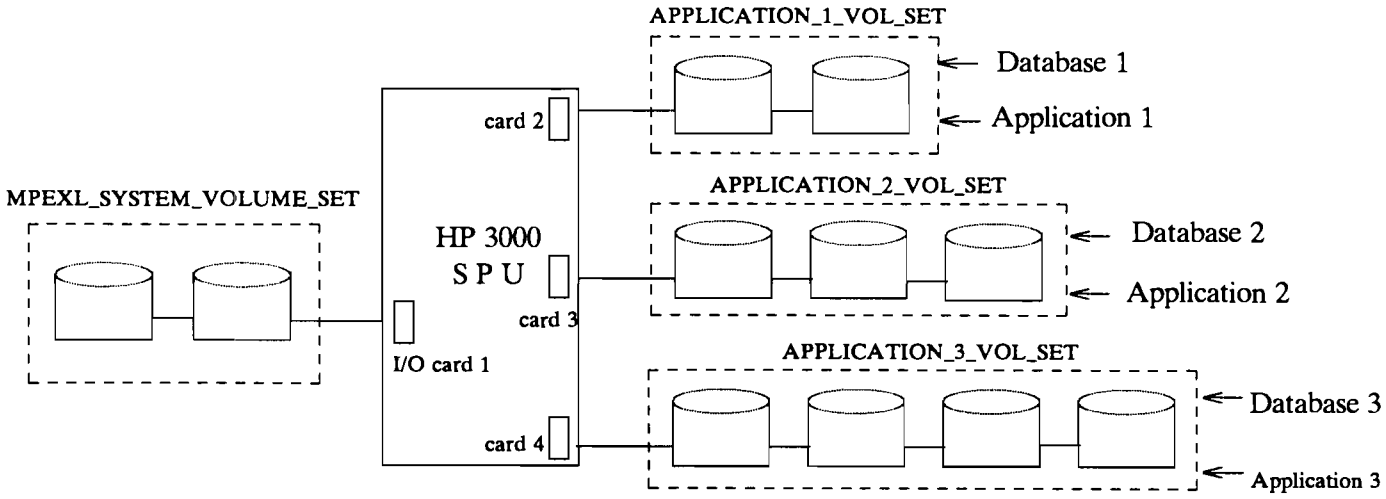


Figure 2-3. Partitioning application data using user volume sets.

2.6 Designing a User Volume Set

Following are the key points that must be kept in mind when designing a user volume set:

- Number of applications residing on the volume set.
- The disk storage requirements of the application data.
- Number of disks required in the volume set.

2.6.1 Calculating the Minimum Number of Disks Required

Obviously, the minimum number of disks required is that which provides the disk space to accommodate the application databases and the system structures. However, because of performance implications, consider using more disks than are minimally required for data storage. In assessing the optimum number of disk drives to configure, consider the following points:

- More disks allow more I/Os to occur concurrently. Therefore, consider low capacity disks.
- Since each disk is a volume, more volumes must be created if using low capacity disks versus higher capacity disks. This will cause more impact to system availability when initially setting up the volume set.
- The average I/O service rate per disk type.
- The average I/O request rate for the candidate data per disk.

The average I/O service rate per disk type may be obtained from the specification of the disk. However, in order to obtain the average I/O service rate for all of the candidate data collectively, performance tools must be used. Some extrapolation may be needed, depending on whether there is data other than the candidate data residing on the system volume set that is required for the performance analysis.

For disks of different types $T_1, T_2, \ldots, T_n$, with capacities $C_1, C_2, \ldots, C_n$, and the number of disks of each type $N_1, N_2, \ldots, N_n$, the following must hold true:

$$\sum_{i=1}^{n} N_i C_i > \textit{Total Data}$$

and

$$\sum_{i=1}^{n} N_i S_i > \textit{Total I/O Request Rate}$$

where S_i is the I/O service rate for the disk type T_i.

2.6.2 Planning for Data Growth

If the database growth rate is believed to be high, prudent planning during setup of the volume set can greatly improve application availability and performance. If database growth rate is high, you may soon need to add a disk or two to respond to the additional disk space demand. Instead of waiting until the application becomes impacted, it may be prudent to accommodate future data growth by increasing the spare capacity of the volume set. This is achieved by adding one or more additional disks into the set. Note that by including additional disks, before the storage space is absolutely necessary, the data on the volume set will be more spread out, resulting in better performance.

2.6.3 Arranging Disks in a Volume Set

A one-to-one correlation between a volume set and an application ensures that the failure of disk-related components only cause one application to be impacted. To guarantee this, all disks in a chain must belong to only one volume set. If any of the disk-related components fails (e.g. I/O cards or cables), only that chain and therefore, volume set, is affected. Conversely, if the disks are scattered around, connected to various cards and their associated cables, failure of any one of the I/O cards or cables will affect the availability of the application on the volume set.

Figure 2-4 illustrates both disk arrangements and the related availability risks.

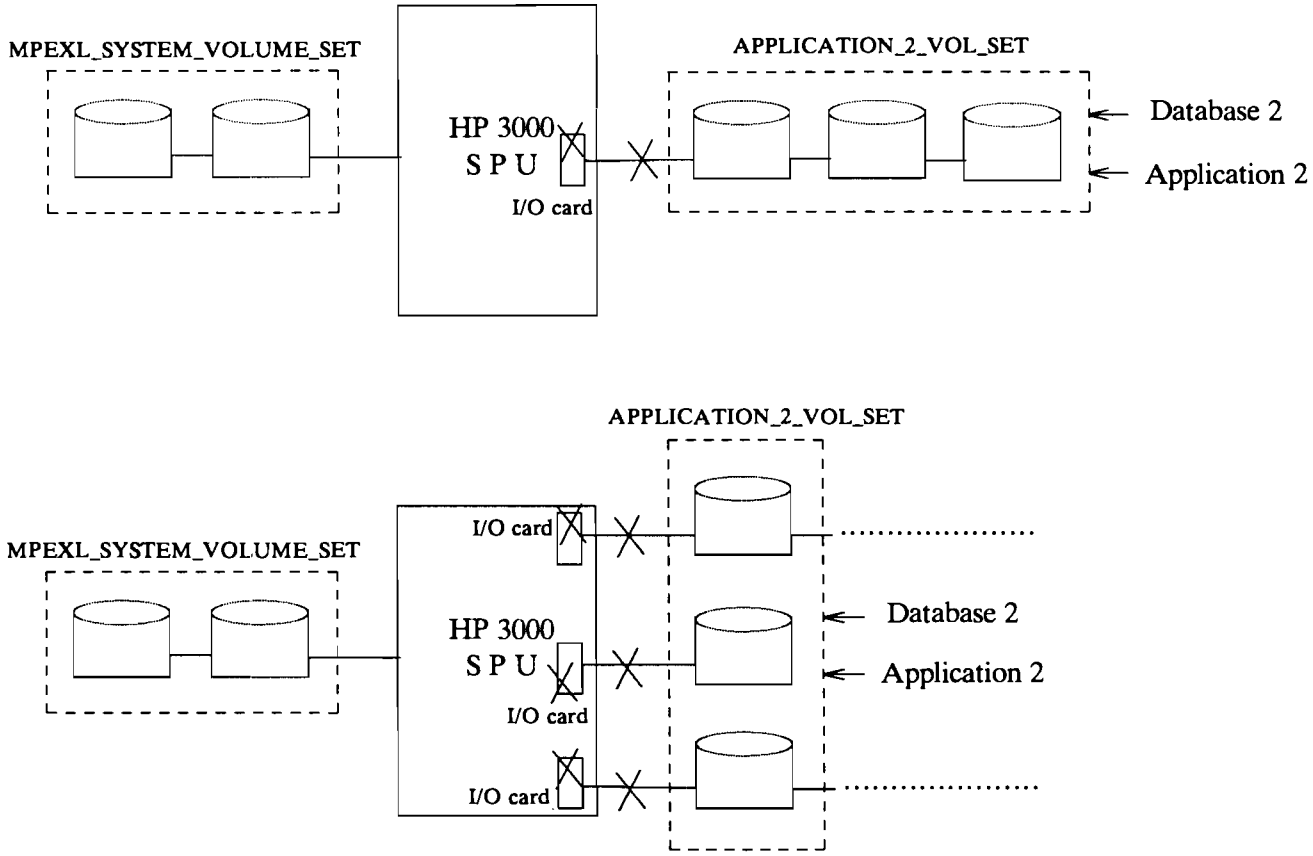


Figure 2-4a. Single card/link arrangement of disks for a user volume set.
Figure 2-4b. Multiple card/link arrangement of disks for a user volume set.

The creation of user volumes is relatively straight forward, but is only a small part of the entire process of implementing user volumes. Often times, the process of setting up user volume sets is not well understood by the operations staff, and therefore deters a data center from incorporating them into its environment. The difficult task is often creating accounting structures. Both tasks, creating volumes and creating accounting structures for user volume sets are described in the next section.

2.7 Creating User Volumes

To illustrate the process of creating a user volume set, let's create a user volume set called APPLICATION_1_VOL_SET (shown in Figure 2-3). It consists of four volumes, one of which is a master. The volumes are named APP1_MASTER, APP1_MEMBER1, APP1_MEMBER2 and APP1_MEMBER3. As mentioned earlier, Volutil is the system utility which must be used to create the set master, APP1_MASTER and to add member volumes.

As shown in Figure 2-2, only disk volumes mounted in "scratch" and "unknown" states can be used in creating a new volume in a set. Also, as shown, only "loner" disk volumes can be converted into "scratch" volumes.

To create the volume set and add its members, invoke Volutil.

MPE/iX: run volutil.pub.sys

Create APPLICATION_1_VOL_SET with the master volume on logical device number 40 as follows:

Volutil: newset APPLICATION_1_VOL_SET APP1_MASTER 40

Add all three members to the volume set in three separate commands:

Volutil: newvol APPLICATION_1_VOL_SET:APP1_MEMBER1 41

Volutil: newvol APPLICATION_1_VOL_SET:APP1_MEMBER2 42

Volutil: newvol APPLICATION_1_VOL_SET:APP1_MEMBER3 43

The above simple set of commands completes the creation of our volume set, APPLICATION_1_VOL_SET.

2.8 Setting Up Accounting Structures

Before any data can be stored on a newly created volume set, appropriate MPE accounting structures must be created on the volume set. Briefly, an MPE accounting structure consists of accounts, each of which has one or more groups, which in turn hold one or more files. Each account must have a "home" volume set, where the account "root" is located. Similarly, each group must have a "home" volume set. All accounting structures needed for an application must be created on the volume set. To ensure all accounts and groups are correctly created on the user volume set, we recommend the use of User Defined Commands (UDCs). By using UDCs, the account manager needs only to enter minimal information because the remaining information is managed via the UDC. A sample UDC is provided in Appendix A.

- Each volume set contains a root directory which is the map for locating all data contained on the volume set. The root directory on the system volume set is distinguished as the "system directory".

- The system directory contains entries for each account on the system.
- Each root directory contains entries for each account on the volume set.
- Each account entry contains group entries for each group within the account.
- A group entry in the system directory must indicate whether a group entry for the same account on the user volume set must be used instead.
- An account must be created on the user volume set before any groups can be created within the account on the set. An account entry is created within the root directory and must also be created in the system directory.
- A group must be created on the user volume set before any files can be created within the groups in the account on the set. A group entry is created within the account entry in the root directory and must also be created in the system directory.
- Accounting structures must explicitly be specified to reside on a user volume set. Otherwise, by default, they will be created on the system volume set.

For illustration, let us assume that you have an account named APP1_ACC and groups within it named ACC_GRP1, ACC_GRP2, and ACC_GRP3, which must all be created on the volume set, APPLICATION_1_VOL_SET.

Without using UDCs one has to do the following in order to create the above account and its respective groups:

1. Logon to the system as manager.sys
2. Enter the following command to create the account:

   MPE/iX: newacct APP1_ACC, <user>
   MPE/iX: newacct APP1_ACC, <user>; onvs = APPLICATION_1_VOL_SET

3. Next enter the following commands to create the group ACC_GRP1:

   MPE/iX: newgroup ACC_GRP1.APP1_ACC; homevs = APPLICATION_1_VOL_SET
   MPE/iX: newgroup ACC_GRP1.APP1_ACC; onvs = APPLICATION_1_VOL_SET

4. Repeat step 3 for the remaining groups.

Notice each creation requires a pair of commands which are similar except for the keyword specified in the second parameter used with the volume set name. The first command line simply adds an entry for the account into the system directory. The second command (with the keyword 'onvs') creates an entry for the account in the root directory on the master volume of APPLICATION_1_VOL_SET.

Similarly, the first line for the group creation adds an entry for ACC_GRP1 into the system directory. The entry also marks that the identical group entry located on APPLICATION_1_VOL_SET must be used to actually locate all files in the group. The second command creates the identical group entry ACC_GRP1 in the account APP1_ACC in the root directory of the user volume set APPLICATION_1_VOL_SET.

3. Disk Arrays For System Availability

In the previous discussion we showed how a disk failure affects system availability to varying degrees with and without data partitions. No matter how data is partitioned, a disk failure always has some affect on system availability. It is crucial to insure minimum disk failures. To accomplish this objective, we recommend using disks arrays with high availability features, commonly referred to as RAID (Redundant Arrays of Inexpensive Disks).

A disk array is a disk storage subsystem based on the RAID concept of multiple disk mechanisms under the command of one controller. A disk array offers several features that differentiate it from a traditional multi-controller disk storage system. Disk array models have different feature sets. The feature that distinguishes a disk array as a high availability array is the parity data disk. In this paper, unless otherwise stated, any discussion regarding disk arrays refers to this high availability model.

HP's high availability disk array has the capability to tolerate an outright failure of a single disk mechanism within the device without interrupting normal host processing or experiencing data loss. Bad sector errors are corrected on-the-fly with an autosparing process, while a parity disk enables recovery of all data.

Consider the following when planning for the implementation of disk arrays to achieve optimum benefit in terms of system availability:

- Only disk arrays containing a parity disk provide data protection in case of bad sectors or mechanism failure.
- Disk arrays can only sustain and recover from one disk mechanism failure at a time.
- Each disk mechanism in the disk array is the same size.
- Disk arrays with five disk mechanisms can be initially configured with either three or five disk mechanisms.
- One disk mechanism is automatically configured for parity data use only.
- The total disk storage space is calculated by multiplying the disk space of a disk mechanism by the number of disk mechanisms used for user data (i.e. total capacity minus parity mechanism capacity).
- Disk arrays provide significantly higher storage capacity than regular disks. Therefore, the number of disk arrays required will be lower than that of regular disks. This may raise performance issues regarding the use of disk arrays. Systems with heavy disk I/O will have better performance if many small capacity arrays are configured rather than a few large capacity arrays.

Adding two more disk mechanisms to an array already configured with three disks requires understanding of the following points:

- The controller cannot operate with the additional disk mechanisms when they are initially added because of the constraints imposed by the data striping technique. First, the data must be rearranged on the expanded set of disk mechanisms.
- The controller has to be reset and then reconfigured to use four disk mechanisms for data storage. In doing so, all data must be reloaded onto the four mechanisms.
- Planned downtime of the volume set or the system (if a system volume set) is required to recreate the volume set and reload the data onto it.

3.1 Recommended Steps for Adding Disk Mechanisms

Adding additional disk mechanisms requires careful planning and execution. The following steps are recommended to ensure successful implementation:

Step 1: Identify which disk arrays in a volume set are operating with three disk mechanisms (including the parity disk). Decide which arrays should be expanded to achieve the desired increase in storage capacity.

Step 2: Take a full backup of the data on the volume set.

Step 3: If a user volume set, close the set. If a system volume set, shutdown the system.

Step 4: If a user volume set, scratch all loner volumes in the set in preparation for recreating the volume set and reloading the data onto it.

Step 5: Add disk mechanisms to the identified disk arrays.

Step 6: Reset the controller on the affected disk arrays and reconfigure to use all five disk mechanisms (four for data storage and one for parity). Refer to the disk array reference manual or contact the CE for the exact instructions to follow.

Step 7: Power on the disk arrays if they are powered off.

Step 8: Recreate the volume set. An install must be performed first if the system volume set is being reconfigured.

Step 9: Reestablish the accounting information on the volume set.

Step 10: Restore the data onto the volume set.

3.2 What Happens When a Disk Mechanism Fails?

The array controller will disable a mechanism from use when a serious hardware malfunction is detected. However, if a second mechanism fails, the entire array becomes unavailable. If one disk mechanism fails, the controller can recreate the lost data on-the-fly by using data on the surviving data mechanisms and the parity mechanism. If the parity mechanism fails, no actual data is lost and only the generation of parity data will stop.

The operator is alerted of a failed mechanism through a console message that repeats until a pending reply is answered. Every minute, the following message will appear on the console:

DISK ARRAY HAS DISABLED A MECHANISM IN LDEV # . NOW IN DATA RECOVERY MODE. NO DATA LOST OR CORRUPTED. OPERATION MAY CONTINUE. PLACE A SERVICE CALL SOON.

The console request message is:

ACKNOWLEDGE DISABLED MECHANISM IN DISK ARRAY IN LDEV # (Y/N)?

The sole reason for using disk arrays is to increase system availability through continued data availability. If there is a malfunction in the disk array, one must be aware of the urgency of replacing the failed mechanism. As mentioned, failure of a mechanism opens a window of vulnerability wherein the failure of another mechanism will cause the failure of the entire disk array.

3.3 Replacing a Failed Mechanism

The disk array supports "hot" replacement of a failed mechanism with another mechanism. As soon as a defective-free mechanism is inserted, the operator is reminded, with a console message, to rebuild the contents of the new mechanism. The message will be repeated every minute until the pending reply is responded to.

At this point, a REBUILD command must be issued using the disk diagnostic FLEXDIAG to recreate the new mechanism's content. While the rebuilding is in progress, the array will continue to service reads and writes with no performance degradation. However, the greater the read and write traffic on the array, the longer it will take to complete the rebuild process. With no read and write array activity, it will take approximately 30 minutes. With heavy activity, the rebuild process will take closer to two hours to complete.

4. Disk Mirroring

Disk arrays do not satisfy the need for the benefit of mirrored disks (Mirrored Disk/iX product). While disk arrays can tolerate a single mechanism failure, they are not completely fault tolerant. If any one of the following disk array components fails, the entire array will be unavailable.

- The disk array controller board.
- The cable connection to the disk array.
- The I/O card to which the disk array is connected.

If the required degree of system availability demands that certain data must always be available, despite probable disk failure, then one must turn to mirrored disks. Simply stated, mirrored disks mirror data on two physically different disks mounted on two different LDEVs so that one copy of data is available if one of the other mirrored pair fails.

Figure 4-1 shows one simple physical configuration of mirrored disks. The mirrored disks provide two copies of identical data, and there is only one link to these copies of data (disks). In this configuration, even if one disk fails, a copy of the data on the failed disk will be available.

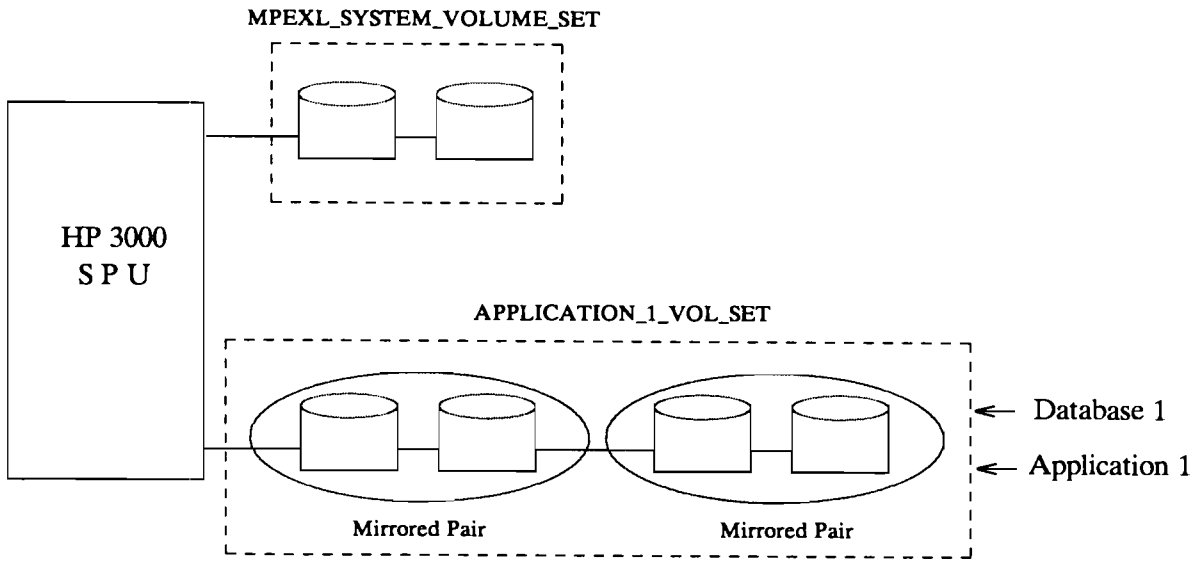


Figure 4-1. A simple one card/link arrangement of disks for a mirrored volume set.

Figure 4-2 illustrates a modified configuration of mirrored disks. Recall, in Figure 4-1 there is only one link to the disks and its failure will cut off access to both copies. In Figure 7, the same disks are rearranged so that each disk has a separate link to the system via a separate I/O card and cable. If one of the links fails, access to the other copy of data is not affected.

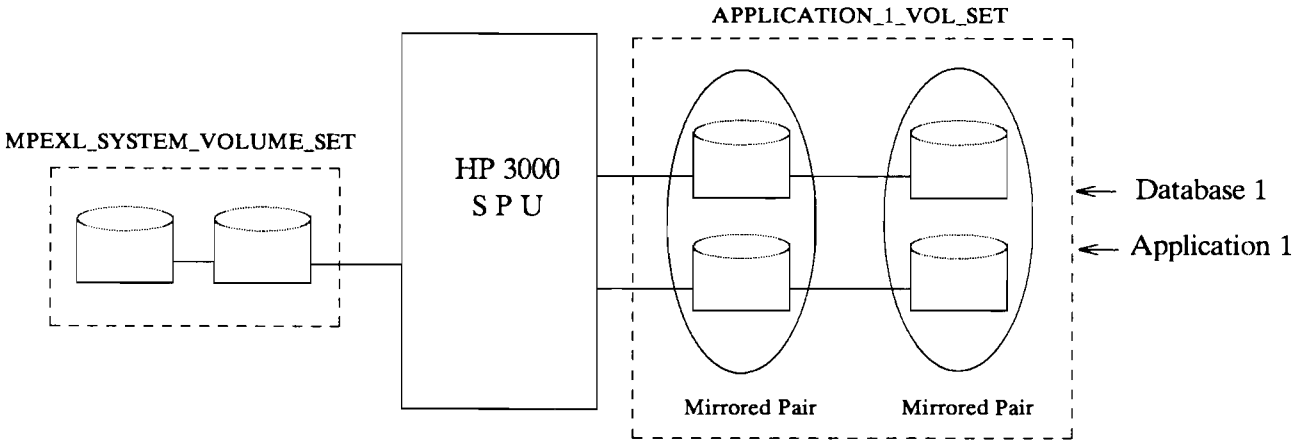


Figure 4-2. A multiple card/link arrangement of disks in a mirrored volume set.

Following are the key points to consider when planning for mirrored disks:

- All volumes in the set must be mirrored. A set cannot contain both mirrored pairs and regular volumes.
- Only user volume sets can be mirrored.
- A user volume set has to be created as a mirrored set. That is, an existing regular user volume set cannot be converted into a mirrored set without recreating it as a mirrored set.
- Disks in a mirrored pair must be of the same type. This ensures that the two disks have the same capacity.
- Twice as many disks are required in a mirrored volume set than in a regular volume set.
- To mirror data currently contained on a regular user volume set, one must do the following:
    - Add additional disks to provide for mirroring as appropriate.
    - Take a full backup of the data.
    - Close the volume set.
    - Scratch all volumes in the set.
    - Using Volutil, recreate the same volume set as a mirrored set.
    - Recreate all accounting structures (as you would do for a regular volume set). The process was previously described in Chapter 2.

- Reload all the data from the backup onto the set.

- To mirror user data currently contained on the system volume set, the data must be offloaded onto a newly created set mirrored user volume set.

4.1 Creating a Mirrored Volume Set

The process of creating a mirrored volume set is similar to that of creating a regular volume set. Note, though, that the VOLUTIL command requires two LDEV parameters, instead of one.

To create our example volume set, APPLICATION_1_VOL_SET, the following Volutil commands are issued:

Mirvutil: newmirrset APPLICATION_1_VOL_SET APP1_MASTER (40, 50)

Add all three members of the previous example to the volume set with three separate commands:

Mirvutil: newmirrvol APPLICATION_1_VOL_SET:APP1_MEMBER1 (41, 51)

Mirvutil: newmirrvol APPLICATION_1_VOL_SET:APP1_MEMBER2 (42, 52)

Mirvutil: newmirrvol APPLICATION_1_VOL_SET:APP1_MEMBER3 (43, 53)

Note the additional term "mirr" in the command names and the presentation of LDEVs as a pair.

4.2 Creating Accounting Structures on Mirrored Volume Set

The process of planning and creating the accounting structures on a mirrored volume set is identical to that of a regular volume set. For a description of the process, please refer to Chapter 2.

Points worth remembering are:

- A newly created mirrored master and mirrored volume undergoes "repairing". Repairing is the process in which all data, whether valid or garbage, is copied from one disk to another in an effort to make the two disks "look" identical.
- During the repairing process, one of the disks is identified as the "source" and the other as the "destination", referring to the direction of copying.
- During the repairing process, only data from the source is available.
- During the repairing process, the failure of the master disk is catastrophic and therefore no data is available from the mirrored pair. The situation is similar to a disk failure on a regular volume set.

- On every reboot, except after a shutdown, mirrored disks undergo the repairing process to create the identical data images.
- No repairing occurs for a mirrored volume when opening (VSOPEN) a volume set, unless the volume was being repaired at the time of closing (VSCLOSE) the set.
- During the mounting of disks, which is a serial process, the first disk of a mirrored pair mounts in a "pending" state. The pending state is an intermediate state in which the data on the volume is inaccessible. Once the partner disk begins mounting, the data on the pair becomes available to users.

4.2.1 Switching Pending"

As previously mentioned, the first disk in a mirrored pair always mounts in an intermediate state called "pending". The system will alert the console operator about the situation by displaying the following console message as soon as the disk is put into a pending state. The message repeats itself every 30 seconds.

?09:09/99/MIRRORED PARTNER MISSING FOR LDEV# 40

To acknowledge the previous message, another console message will be displayed. Replying to this message will discontinue the repetitive display of the former message.

?09:09/22/ACKNOWLEDGE MIRRORED PARTNER MISSING FOR LDEV#40(Y/N)?

The disk remains in the pending state until its partner disk starts mounting, at which time mirroring begins.

However, what happens when the partner disk does not mount because it's defective or it is taken offline? For system availability reasons, you may want to use the partner disk that is available, albeit with mirroring disabled. This state of a mirrored disk is referred to as the "(mirror-) suspend" state. To minimize the affect to system availability, a 'suspendmirrvol' command should be issued on the pending disk as quickly as possible.

In order to move the disk from its intermediate pending state to the suspend state, one must issue a Volutil command as follows:

Mirvutil: suspendmirrvol APPLICATION_1_VOL_SET:APP1_MASTER 40

Here, the master volume is assumed to have its mirrored pair on LDEV 40 in pending state, a situation in which a volume set has not yet mounted.

4.2.2 Resuming Normal Mirroring on a Mirror-Suspended Volume

When a mirrored disk is operating mirror-suspended, mirroring can be resumed after one of the following situations occurs:

- The partner disk becomes available. (Note, however, when it mounts up it will be put into loner state.)
- The original partner disk is replaced by another scratch volume.

In the first situation, the original partner disk has to be scratched using Volutil. The following command is used to replace a mirrored disk. Before issuing the command, the disk must be in the scratch state.

Mirvutil: replacemirrvol APPLICATION_1_VOL_SET:APP1_MASTER 60

Note that "replacemirrvol" always causes the repair process to occur. The disk that was in suspend state will be the source and the scratch disk will be the destination.

4.2.3 Disk-related Failure During Normal Mirroring

If one of the disk-related failures highlighted earlier occurs during normal mirroring, the defective mirrored disk is disabled by the system. Consequently, the surviving disk is put in "nonmirror" state in which mirroring is disabled. The surviving disk is accessible by users. As soon as the disk moves to the non-mirrored state, the operator is informed through a console message that repeats every 30 seconds. The message appears as follows:

?04:04/44/MIRRORED VOLUME DISABLED ON LDEV# 50

To suppress this message from repeating, the operator must reply to the following message:

?04:04/22/ACKNOWLEDGE MIRRORED PARTNER DISABLED ON LDEV# 50(Y/N)?

4.2.4 Points to Keep in Mind During Use of Mirrored Disks

- A disabled mirrored disk simply means the system encountered a problem during I/O access to the disk. It may be because of one of the disk-related hardware failures mentioned previously, or because of a prolonged powerdown of the disk. The disk itself may not be defective.
- Once disabled, a mirrored disk remains disabled for the current mount only. In the subsequent mounting, an attempt will be made to reunite it with its surviving partner.
- A disabled disk that is not defective can be reunited with the surviving disk by issuing the "replacemirrvol" command in Volutil.
- A surviving disk is immediately put in a nonmirror state after its partner is disabled.
- A surviving disk will "look" for its partner disk during subsequent system boots in an attempt to reunite in mirroring. Since it knows that its partner disk is disabled it will initiate repairing as part of reuniting.
- The suspend state and the nonmirror state indicate different actions which lead to the disabling of the mirroring process. A user puts a pending disk into the suspend state by issuing the "suspendmirrvol" command in Volutil. The system puts a normally

mirroring disk into the nonmirror state automatically when it detects "noaccess" to its partner disk.

- A mirrored disk remains in pending state as long as its partner disk is missing or until a suspendmirrvol is issued.
- Data from the pending mirrored disk is not available for access.
- During the repair process, the destination disk is still considered to be disabled, and the source still maintains that its partner is "bad" for all practical purposes.
- A repairing on a volume is automatically aborted when the corresponding volume set is closed or the system is shut down. No status regarding the repair is recorded, which means the next repairing will start from the beginning.

To improve system availability with mirrored disks, we recommend the following guidelines.

- In addition to the recommendations made in Chapter 2 regarding user volume sets, ensure no two disks on the same chain comprise a mirrored pair. In other words, make sure that only one disk of each mirrored pair is chained on a card.
- Respond immediately to a pending mirrored disk if it's known that the partner disk will not be mounted and the data on the pending disk has system availability consequences.
- Respond quickly to resume normal mirroring on suspended and nonmirrored volumes.
- Postpone shutting down the system or closing the mirrored volume set if one or more mirrored volumes are undergoing repairing.

5. SPU Switchover

Even in the ideal configuration of mirrored disks, system availability is still vulnerable to hardware and software failures. You can further improve data availability of important user volume sets by configuring them to two systems in what is called the SPU switchover environment (SPU Switchover/iX product).

The SPU switchover environment can be summarized as follows:

- Two systems have physical access to the same user volume set.
- Only one system is given the capability to mount the volume set during normal circumstances. That system is said to be the "home" for the volume set.
- The other system is given the capability to mount the set only when a user issues the switchover command. This system is called the "alternate" for the volume set.
- Each system has a unique name for the purpose of identification.
- The process of switching a volume set from its home to the alternate is called switchover. Switchover is accomplished when an operator issues the
- The process of switching a volume set from its alternate to its home is called switchback, and is only accomplished when an operator issues the
- During switchover and switchback, a volume set always undergoes a recovery process similar to that which happens during the mounting of a volume set at bootup.
- The alternate system will enter into a continuous polling of the home in order to detect its "death". This process is referred to as "heartbeat".

5.1 Planning for switchover setup

- Decide what applications on a system are important in terms of availability.
- Are these applications and their data already on the user volume sets? If not, you must first move them onto one or more user volume sets. Use mirrored disks to further improve system availability.
- Identify which user volume sets house the required applications and their data.
- Identify what other system will run the above applications in the event of the failure of the original system.
- Determine names for both the home and the alternate systems.
- Take a full backup of each system.
- Identify disks for the required applications and their data. This will include new disks which may be added in order to move application data from one user or system volume set to one or more user volume sets in an effort to partition the data for system

availability.

- Draw a map of the physical connections of the disks. Use recommendations in the sections on data partitioning and mirrored disks, if applicable.
- Update SYSGEN, if necessary, on both systems to reflect the new physical paths to the disks (shown in the map).
- Bring both systems down.
- Physically connect the disks per the map. Cross connect the same disks (most probably, chains) to the alternate system.
- If modifying the system volume set on the home system, perform an install.
- Bootup the home system.
- Create member volumes for the system volume set if an install was done prior to bootup.
- Follow the steps outlined in the section on data partitioning if new user volume sets are to be created to house the application data.
- Boot up the alternate system.
- Configure both systems for switchover as described in the User's Manual on SPU Switchover/iX.
- Create duplicate environments on the alternate system so that the applications can run after switchover, including duplicate accounting structures. Details regarding this follow in another section.

5.2 On Detecting the Home's Death

As soon as the home's death is detected, the alternate will display a message on the console to alert the operator. This message, which repeats every 30 seconds until the operator replies, is as follows:

?09:09/99/SYSTEM <name> NOT RESPONDING. PLEASE CHECK <name>.

The operator may decide at this point to switchover the volume set. However, the operator must complete the following before issuing the switchover command:

- Check if the home is indeed dead.
- If so, inform all the users on the home about the situation.
- Inform all users on the alternate who will be affected because of switchover to prepare for the switchover per the planned switchover plans.
- Quiesce any applications on the alternate system per the switchover plans.
- Execute the switchover command.

After execution of the switchover, the following must be done:

- If a mirrored volume in the switchset was operating in suspend state on the home, it will mount on the alternate system in the pending state. Issue the "suspendmirrvol"

command in Volutil against the pending mirrored disk.

- If a mirrored volume in the switchset is operating in the nonmirror state on the home and its partner disk will be missing during switchover, issue the "suspendmirrvol" command in Volutil against this pending disk.
- Initiate applications on the newly switched volume set.
- Inform users of the appropriate applications to now logon to on the alternate system to access their applications.

5.3 Synchronizing Directory Structures

As mentioned earlier, requisite to running applications on the alternate system after the switchover are duplicate accounting structures on the system volume set of the alternate system. This duplication can be done when setting up the switchover environment, however, there is a major drawback. Directory structures on the home system may change during the normal operation of the system. Therefore, at the time of switchover, the directory structures on the alternate system may be inconsistent with the home system. To ensure this doesn't occur, the directory structures on the alternate system should be modified at the time they are modified on the home system.

A utility useful in efficiently creating the accounting structures on the alternate system is called BULDACCT. BULDACCT builds a job stream corresponding to the directory structures on the home system that must be duplicated on the alternate system. The job stream is created on the appropriate home volume sets. BULDACCT can be executed whenever accounting structure modification takes place. The job can also be scheduled to execute regularly to ensure all modifications to the accounting structures are duplicated. After a switchover is performed, the BULDACCT job is streamed to recreate the directory structures on the alternate system.

5.4 Network and Terminal Switchover

SPU Switchover does not automatically switch terminals and networks from one system when a switchover occurs. The type of terminal switchover required depends on the terminal protocol of your system. Network switchover is performed through the OpenView DTC Manager.

Both network and terminal switchover involves planning and execution of multiple tasks which are often cumbersome. This topic is covered in detail in the User Manual for the SPU Switchover product.

Following are the key points related to the network and terminal switchover:

- Special hardware configurations are required for network switchover. Refer to the section 'Hardware Configuration', below, for additional information.
- The nailed terminals used by the applications on the home system are required to be preconfigured with the same ldev number on the alternate system.

- Two configuration files are required on the alternate system. One file contains the standard configuration which is used during the normal operation prior to switchover. Let's call this file 'NMCONSTD.PUB.SYS'. The other file contains the configuration which is used, instead of the standard configuration, after switchover. Let's call this file See 'Network Configuration' below for further details.
- The effects of the network switchover on terminal I/O and PAD access are:
    - The alternate system takes the identity of the home system. Since there can't be multiple node names nor multiple node addresses on the same system, the alternate system will lose its network identity. The second configuration file mentioned contains this information.
    - Once the alternate system takes the identity of the home system, all connections to it from remote nodes are only possible if they use the home system's name and IP address.
    - The LAN and X.25 network services must be stopped and restarted with the "backup" configuration. Therefore, all previous network services (NS) users lose their connections.
- The alternate system must contain the same network directory (NSDIR.NET.SYS) as the home system.
- The home and alternate systems must both reside on the OpenView DTC Manager (OVDTCMGR) Network Map.
- Each DTC being used for TIO/PAD traffic on the home and alternate systems requires two configurations:
    - A standard DTC configuration with terminals and PAD connections configured for either switching or connecting to a host (home or alternate) system as desired. Let's call this file 'DTC01.STD'.
    - A switchover DTC configuration where all nonswitched (directly-connected) terminal and PAD connections are configured to connect to the alternate system. Let's call this file 'DTC01.BAC'.
- The working configuration is a third file which is used to download the DTC. Either of the above two configurations must be copied to this file before the download is done. It must have the suffix 'DTC'; for the above example, the file will be called 'DTC01.DTC'.
- The effects of the network switchover on the alternate system's terminal I/O and PAD access are:
    - All DTC and PAD connections are lost. Before the DTC01.BAC configuration is downloaded, the DTC is reset. This clears all DTC connections.
    - The DTCs continue to recognize the TIO and PAD connections as being on the alternate system.
- While in switchover mode, the transport layer recognizes the alternate system as the home system. TIO recognizes the alternate system as the alternate system.
- If the alternate system is rebooted while the switchover configuration NMCONBAC.PUB.SYS is active (because it was copied to NMCONFIG.PUB.SYS), TIO will recognize the alternate system as the home system.

- During switchover, the following steps, related to the network switchover, must be taken.
    - Stop both LAN and X.25 on the alternate system.
    - Completely stop the OpenView DTC Manager.
    - Copy the configuration file DTC01.BAC to the working configuration file DTC01.DTC.
    - Restart the OpenView DTC Manager.
    - Copy the configuration file NMCONBAC.PUB.SYS to the working file NMCONFIG.PUB.SYS.
    - Restart both LAN and X.25.

5.4.1 Hardware Configuration

The following hardware configurations are required for network switchover:

- The NS LAN must be separate from the TIO/X.25 LAN.
- All DTC SNP cards used for an X.25 LAN network on the home system must be dedicated to the home system or the home and alternate systems. No other systems can use these SNPs.
- Both the home and alternate systems must use the OpenView DTC Manager.

5.4.2 Network Configuration

The standard configuration for the alternate system (NMCONSTD.PUB.SYS) must specify the following terminal I/O configuration qualifications.

- The TIO/X.25 Link configuration (e.g. LNK.DTSLINK) is identical to the home system.
- The TIO configuration (DTS subtree) is identical to the home system. (Additional terminals and printers can be configured, but the file must contain the same profiles and nailed port configuration as the home system.)

The switchover configuration for the alternate system (NMCONBAC.PUB.SYS) must match that of the home system as follows.

- For the LAN Network:
    - The nodename
    - The LAN IP address
    - The 802 address (this will override the chip 802 address)
- For the X.25 Network, the X.25 NI configuration (for example, the NETXPORT.NI.X25NET subtree)

- For Terminal I/O:
    - The TIO/X.25 Link configuration
    - The TIO configuration (DTS subtree)

6. Backup For System Availability

Backup on the HP 3000 affects system availability in varying degrees.

- Dedicated backup processes require that no users and applications can use the files being backed up. Here, the system availability may be near zero.
- Selective sets of users and applications can run on the system while the data corresponding to another set of users and applications is being backed up. This is considered as a dedicated, but data-specific backup. Here, the system availability is neither zero nor 100%.
- Online backup processes allow all users and applications to run on the system concurrently with the backing up of their data. This is possible with both a full backup and a data-specific backup. The system availability in this case is 100%, except during the quiesce period (it is the time during which all related files are closed prior to initiating the online backup).

6.1 Dedicated Backup

The key points of the dedicated backup related to system availability are:

- The data being backed up must be in a consistent state. To achieve data consistency, the users and applications which depend on the data cannot access the data during the backup.
- For a dedicated system backup, the entire system is unavailable to all users and applications on the system.
- For a dedicated backup of a selective data on the system, only users and applications which depend on that data are restricted from access.
- Multiple dedicated backups can occur concurrently (each backing up a different set of data), thus speeding up the overall backup.
- A dedicated backup can use multiple backup devices in parallel (and hence using multiple storesets), thus speeding up the backup process.

Factors influencing the design of a dedicated backup, within the context of system availability, are as follows:

- The speed of the backup device.
- The capacity of the backup device.
- The number of backup devices used.
- The amount of data to be backed up.

- The amount of time allotted for backup.

6.1.1 Selecting a Backup Device Based on its Speed

The speed of a backup device allows us to evaluate how long it will take to backup a fixed amount of data. Generally, the destination of backups are tapes, and the tape device is typically one of the slowest devices on a system. However, there are different classes and types of tape devices available, with varying degrees of speed and capacity. Also, there are backup devices other than tape devices which are superior in terms of speed and capacity. Since system availability is always affected during a dedicated backup, the goal on selecting a backup device must be based on speed. In most cases, data transfer is the performance inhibitor for a dedicated backup. However, using multiple backup devices in parallel will increase the data transfer rate for a dedicated backup.

Another important factor to consider when evaluating the speed of a device is data compression. Some devices support hardware (data) compression, while there are software products which provide data compression on certain devices. With data compression, one must evaluate the device speed in terms of the time it will take to backup a fixed amount of original data.

6.1.2 Selecting the Capacity of Backup Devices

Once a device is selected based on its speed, the next consideration is how many tapes will be required to completely backup the data. Fewer tapes will provide higher system availability because operator intervention will be required less frequently. In this respect, higher capacity tapes are preferable to lower capacity tapes.

6.1.3 Determining the Number of Backup Devices

Once a backup device is selected based on its speed and capacity, any of the following operational strategies can be used.

1. Use only one backup device and mount the appropriate number of media on the same device in order to complete the backup.
2. Use as many backup devices as necessary in parallel, with their sum total of capacity equal to or greater than the total amount of data. In this case, either of the following strategies can be used.
    - One dedicated backup process handling all the above devices in parallel. Since the sum total of the capacities of all devices is less than the total amount of data, one or more devices will need additional media to be mounted.
    - As many separate dedicated backup processes as there are devices, each handling a separate device and separate unique fileset. Here, too, one or more devices will need additional media to be mounted in order to complete the backup of the corresponding data.

3. Use a combination of items 1 and 2, such that the number of devices is greater than one but less than the number derived in item 2.

Note that when using multiple devices for backing up multiple storesets in parallel within a single dedicated backup, all such devices must be of identical type. Also, the interleaving, which is optional, can be used to speed up the backup process when the data to be backed up resides in two or more disk volumes.

6.1.4 Determining the Amount of Data to be Backed Up

The total amount of data is the critical factor in deciding how many devices of a particular capacity and speed should be used in a particular kind of backup environment. For example, let's consider choosing a device whose capacity is x bytes. Let the total amount of data to be backed up be $4x+y$ bytes, where $y < x$. Let's say you choose one dedicated backup process using TurboStore/iX to completely backup the data. From the system availability point of view, you decide to use unattended backup which would not require media to be replaced once the backup has started. To accomplish this, we proceed as follows:

- Round off the total amount of data to an integral multiple of the device capacity - for the above case, 5x.
- Divide 5x by x to obtain the number of devices to use - 5.
- Connect 5 like devices and configure them, as well as configure unconnected devices for future growth in backup data.

6.1.5 Considering the Time Allotted for Backup

The time available for backup is another key factor in determining the backup environment design. When considering the available time, there are some questions which must be answered:

- Are backups performed at only one time during a weekly period? If so, can the system availability be zero for the entire time, or can only a few applications be down for backup at a time?
- If the system availability can be zero, then you may use one or more dedicated backup processes with the appropriate number of devices to finish the backup in the allotted time.
- If the system availability cannot be zero at any time (i.e., one or more applications must be running at all times), then one or more application-specific dedicated backup processes can be designed to be used with the appropriate number of devices for each process.
- Is one backup scheduled per day? If so, can the system availability be zero during this time, or can only a few applications be down for backup at a time?
- If the system availability cannot be zero at any time (i.e., one or more applications must be running at all times), then should application backup be scheduled on a daily basis. This will also apply in the case where system availability can be zero during the

allotted time, but the allotted time itself will be insufficient to finish backup of all applications.

- Is the allotted time a combination of multiple backups per day? If so, what how long is alloted for each backup?

6.2 Online Backup - With Full System Availability

Online backup is the only way to backup data in a 24 X 7 business operation, where system availability must always be nearly 100%. There are two ways in which the online backup can be done.

1. Using TurboStore/iX with the online backup facility.
2. Using TurboStore/iX with split mirrored volume sets.

6.2.1 Using TurboStore With Online Facility

TurboStore with Online Backup supports backup on both user volume sets and the system volume set. The key points to remember when considering TurboStore for online backup are:

- All files in the data set must be closed (quiesced) before initiating the backup process.
- Quiescing all applications in preparation for backing up the data on the system will affect total system availability.
- Backup of application data can be serialized. Serialization only requires one application at a time to be quiesced and unavailable for backup. Therefore, system availability is only partially affected.
- Online backup can use multiple devices in parallel with interleaving. This is identical to what was discussed above under 'Dedicated Backup'. All design considerations for a dedicated backup environment discussed previously apply here.

6.2.2 Using TurboStore With Split Mirrored Disks

"Split volumes" is a concept that only applies to mirrored disks. A mirrored volume set can be split into two equal halves, each half being the exact replica of the other half. The CI command which is used to close any volume set, i.e., VSCLOSE, is invoked with a 'SPLIT' option in order to accomplish this. As soon as the split is done, one half is automatically identified as the "user" half, and the other half as the "backup" half. Each half is in itself a complete volume set. The user half of the volume set can be opened for data access from the system, while the backup half of the volume set can be simultaneously opened for backup using TurboStore.

While this feature was incorporated into TurboStore prior to the availability of the online feature, split volume backup is no longer recommended for the following reasons.

- Split volume backup only works with mirrored disks. On a system with both mirrored and regular volume sets, different methods of backup must be planned and executed, consuming extra efforts and resulting in inconsistent backup methods.
- Split volume backup only works when all volumes in the set are mounted and no repairs are happening at the time of splitting the set.
- Mirroring remains disabled during the backup period (because the set has been split), leaving the volume set at risk of disk failure without the protection the mirrored disk product provides.

7. AutoRestart For System Availability

A software related system abort brings the system availability to zero. The total system availability affected depends on how long the system availability remains at zero. That is, how much time it takes to bring the system up and restart the applications. The longer it takes, the higher the total system availability is affected.

The time spent between the system abort occurrence and the initiation of applications is summarized as follows.

1. Time passes before the operator notices the system abort.
2. The operator logs the system abort information.
3. The operator takes some time to decide if a dump should be taken.
4. If a dump is deemed appropriate, the operator soft resets the system and takes a dump.
5. The system is rebooted.

In order to increase the total system availability in the event of a software related system failure, the time it takes to complete these activities must be decreased. Foremost, it would save time if the intervention of the operator could be avoided. To do so would mean that all the activities would have to be automatically initiated without operator intervention. AutoRestart/iX provides the capability to setup an operatorless environment, per the terms and conditions predefined by the operator, to manage the system and application recovery following a software related abort.

AutoRestart has the following features:

- **Dump-to-disk.** The Dump-to-disk feature enables MPE/iX to write system dump information directly to a preallocated disk file. This feature is an addition to the dump-to-tape feature available on all MPE/iX-based systems. However, it is much faster than dump-to-tape.
- **Restart.** With special enhancements to the system abort feature available on all MPE/iX-based systems, AutoRestart allows an automatic nondestructive reboot (memory contents are preserved) following a system abort.
- **Autoboot.** AutoRestart provides a FORMAT utility that enables you to create a specially formatted autoboot format file that contains a customized sequence of ISL startup commands.
- **Mini-dump.** A mini-dump is a summary of system failure information written to an ASCII disk file. It is independent of the full dump.

7.1 Mini-Dump

Mini-dump writes a summary of the system failure information to an ASCII disk file using the SAT utility. It is configured using the INITMD command of the BLDDUMP utility.

Mini-dump can be configured to take a particular type of dump or no dump at all, depending on the system abort that occurs. A scenario illustrating the use of this feature follows. In an environment with many systems with similar configurations and operating system versions, duplicate problems can occur. With this feature, once a full dump for a recurring problem is taken, the mini-dump can be configured to automatically avoid taking another full-dump for the same problem. However, in most cases, a problem may only appear once which may not warrant diagnosing it. For this case, the mini-dump can be configured initially, as the default.

After a problem occurs, you may reconfigure the mini-dump to take a full dump in case the particular problem reoccurs. Mini-dump can be configured as the default for all other problems. After the problem occurs a second time and a full dump is taken per the configuration mentioned above, you may now reconfigure the mini-dump to skip taking a full dump for this specific problem.

Following is a summary of the events and actions for the above scenario:

- Before any system abort has occurred.....
  - Configure the mini-dump without full-dump as the default.
- A problem (system abort) occurs.....
  - The system takes a mini-dump for all problems (default configuration).
  - You reconfigure the mini-dump to take a full dump for the above problem only.
- The same problem (system abort) occurs again.....
  - The system takes a mini-dump as well as a full dump.
  - You reconfigure the mini-dump to skip taking a full dump for the above problem.
- The same problem (system abort) occurs still again.....
  - The system takes a mini-dump and skips taking a full dump.

In order to accomplish the goal of system availability using AutoRestart, one must configure AutoRestart to:

- Use the mini-dump feature to save the summary of system abort information. The time spent to log this information manually is saved.
- Use the dump-to-disk feature so that the time spent taking a dump is drastically cut (when compared to dump-to-tape feature).
- Anticipate certain system aborts and decide whether or not to take a dump. If an unnecessary dump is skipped, significant time is saved.
- Reboot the system.

System availability can significantly improve if the requirements and limitations of Autorestart are understood and appropriately planned for.

- AutoRestart is an integrated environment in which the modified versions of the existing utilities, DUMP, SAT and START, execute based on files and configuration that are created using the new utility BLDDUMP. The autoboot feature is used by AutoRestart to automatically execute these utilities when taking a full-dump, a mini-dump, and when restarting the system.
- AutoRestart dumps system state information primarily to disk and alternately to tape. The former is often referred to as the dump-to-disk feature of AutoRestart.
- AutoRestart's dump-to-disk facility dumps information to preallocated disk files only.
- Up to ten disk files can be preallocated for dump-to-disk files.
- AutoRestart dumps to tape only if no more preallocated files are available for the dump, and only if it is configured to use tape (on the alternate boot path) as an alternative dump device.
- Only one mini-dump file can be configured. Every time a mini-dump is taken, that mini-dump file is overwritten.
- An autoboot file must be created containing ISL instructions for taking a full dump, mini-dump and the start command with the required "-R" option.
- A mini-dump is taken if the autoboot file includes the instruction "SAT SATINIT".
- A full dump is taken if the autoboot file includes the instruction "dump", or if the mini-dump script file specifies that a full dump should be taken for a particular system failure.
- If the "-R" option is specified in the autoboot file, the system will reboot automatically after the appropriate dump completes.

7.2 Preparing Your System

System preparation for AutoRestart consists of the following steps:

- Determine your dump-to-disk needs.
- Determine your mini-dump needs.
- Add one or more disks to your system's configuration.
- Physically install the disks.
- Create the AutoRestart account structure.

7.3 Determining your Dump-to-Disk Needs

Give careful consideration to managing your system's dump files by determining the following dump-to-disk requirements:

1. Amount of disk space allocated for dumps.
2. Volume set on which the disk space will be available.

3. Will the above volume set be newly created?
4. If so, the number of disks minimally needed to provide the above amount of disk space.
5. The estimated size of the largest dump file.
6. The number of preallocated disk files (to determine, divide the total disk space allocated for dumps (from item 1) by the above estimated size (from item 5).

7.4 Estimating the Size of the Largest Dump File

There are two ways to estimate the size of the largest dump file.

- Use the Estimation Table. This method can be used even if AutoRestart is not installed.
- Use the DTODSIZ utility located in HP36375.TELESUP. This utility is only available in the above specified location after AutoRestart is installed on the system.

7.4.1 Using the Estimation Table

- Determine the maximum number of active jobs and sessions on the system at any time.
- Determine the main memory size on the system.
- Use Table 2-1 in the User's Guide for AutoRestart/iX to find the dump size indicated in the row and column corresponding to the above values.

7.4.2 Using DTODSIZ utility

- Wait until the system activity is at a peak (when the maximum number of jobs and sessions are active.)
- Run DTODSIZ(.HP36375.TELESUP).
- Schedule a DTODSIZ run as often as required during different peak periods of system activity to get the most accurate estimate.

7.5 Determining Mini-Dump Needs

A mini-dump is copied to an ASCII file. This file can be either the default file, called MINIDUMP, or can be one you create using the BUILD command. In either case, the file must be of appropriate size to hold the mini-dump information. The default size is 500, 80-byte records, which is sufficient in most cases, providing the default values for mini-dump aren't changed.

7.6 Creating the AutoRestart account structure

AutoRestart requires two groups, HP36375 and DISKDUMP, located in the TELESUP account. If you create a new volume set for the dump-to-disk files, you will need to create the TELESUP account with the above groups on the volume set. If the HP36375 group was already created on the system volume set during the install of AutoRestart, then the following must be done.

- Store all files in the group onto a tape.
- Purge the group from the system volume set.
- Create the group on your user volume set that is set reserved for dump files.
- Restore all files from the tape into the group. The files will now reside on the user volume set.

7.7 Configuring and Managing AutoRestart for System Availability

- Use a separate volume set for dump-to-disk files. Mixing system files and dump files on the same disk can result in simultaneous system failure and dump facility failure.
- Create more than one dump-to-disk file. While one dump-file is being analyzed, the dump process can use another dump-file to take a new dump.
- Reset a dump-file as soon as analysis is finished. Resetting the dump-file allows the dump process to reuse a dump-file for the next dump.
- If only one dump-file is created and configured, then assign and configure the tape device on the alternate boot path as an alternative dump device. Always keep a tape loaded on the drive.
- Leave all dump files unprotected so the system can always successfully overwrite the dump-file. Therefore, the system does not have to resort to using the tape device, which takes longer, affecting system availability. Because dump-files are automatically overwritten, special attention is required to ensure dumps are analyzed or stored for later analysis before they are written over.
- Configure AutoRestart so that the dump facility terminates and AutoRestart proceeds with the rest of the ISL commands in the autoboot file in case the dump is unsuccessful. Following are the reasons for unsuccessful dump-taking:
    - If all dump files are configured as protected from overwriting and the tape device is not configured as the alternative dump device, the dump facility finds no dump-files available for the next dump. In this case, all dump information is lost.
    - The dump is larger than the dump-file. In this case, only part of the dump is lost.
- The mini-dump script file contains the instruction to bypass the full dump of a particular system abort.

8. Threshold Manager

There are many internal resources that the MPE/iX operating system depends on for its fundamental operations. When some of these resources are depleted, the operating system cannot make any progress, which causes it to fail. In an effort to alert the operator before these resources are completely depleted, Threshold Manager allows the system manager to specify thresholds on the usage of these resources and alerts the operator when these thresholds are reached. It does this in two ways.

- Threshold Manager can provide notification whenever a monitored resource exceeds user defined thresholds.
- Threshold Manager can turn off job and session logons when a monitored resource exceeds user defined critical thresholds.

Threshold Manager provides the following features:

- A single resource can have a maximum of two thresholds.
- Threshold Manager sends notification of fallback and jumpoff events through messages to the system console.
- A predefined control, JOB_SESSION_CONTROL, when configured for a threshold, disallows initiation of new jobs and sessions, upon jumpoff event and allows it again upon fallback event.
- Another predefined control, NULL_CONTROL, as the name implies, takes no action upon jumpoff and fallback events.
- Notification can be turned on or off, both at the global and at the threshold level.
- Configuration of all or selected thresholds can be displayed.
- A threshold can be modified or deleted.
- Threshold Manager itself can be enabled or disabled.
- Threshold Manager operates in cycles which repeat every 120 seconds by default. The cycle repeat time can be changed.
- OpenView Console supports notification of fallback and jumpoff events.

A list of resources monitored by Threshold Manager is:

1. Access_Rights
2. CM_Code_Segments
3. CM_Data_Segments
4. CM_Physical_Code_Segments

5. CM_Ports_Completion
6. CM_Ports
7. File_Descriptors
8. File_Extents
9. I/O_Notifications
10. I/O_Requests
11. Locality_List_Entries
12. Memory_Information_Blocks
13. Shared_Global_Space
14. Sys_Vol_Set_Permanent_Space
15. Sys_Vol_Set_Transient_Space
16. Timer_Req_List
17. Timers
18. Virtual_Space_Cache
19. Virtual_Space_Objects
20. Virtual_Storage

As you may have noted from the above list, the resource names are of operating system jargon. Therefore, only people familiar with the general principles of the theory and design of modern operating systems can comprehend what these resources mean to the system operation.

Understanding some of the key points of Threshold Manager can help in planning the setup of Threshold Manager for your system.

- Threshold Manager is shipped with a default configuration of thresholds for the above list of resources.
- Threshold Manager is disabled at installation and remains so until it is turned on explicitly.
- Upon installation, the Threshold Manager configuration contains default values. Therefore, if you turn on Threshold Manager right after the installation, it operates based on the default configuration.
- If no thresholds in the default configuration are configured with JOB_SESSION_CONTROL control, instant jumpoff will not be in effect. Therefore, only notification to the console will be in effect.
- The display of configuration includes the current usage levels of all resources. This data can be helpful for planning the first custom configuration.
- You can always reset the current configuration to the default configuration in order to operate with default values, or to start all over again for custom configuration.

8.1 What are Jumpoff and Fallback Events?

For the sake of explanation, let's assume that your custom configuration has the resource Sys_Vol_Set_Transient_Space configured with two thresholds as follows:

No.	Level, %	Control Name	Notification
1	75	NULL_CONTROL	ON
2	90	JOB_SESSION_CONTROL	ON

Also, let's assume that the system has just been booted and Threshold Manager is now enabled. Now, as more and more jobs and sessions logon onto the system, usage of transient space gradually increases. At some point in time, let's say the usage of transient space reaches 76%. When the Threshold Manager notices (which it attempts every 120 seconds by default) that the usage has "crossed" the 75% threshold, it takes action on behalf of this threshold and based on the other configuration data for the threshold. As the above configuration shows, with notification enabled (i.e., ON), a message on the console notifies the operator about this event, which is called jumpoff. Since the control is NULL_CONTROL, essentially, no other action takes place.

Say the usage continues to climb until it reaches 91%. Notice that a second threshold is set at 90%, and configured with JOB_SESSION_CONTROL control and with notification ON. Threshold Manager reacts to this jumpoff event through notification of another message pertaining to the same resource but identifying a different threshold. Additionally, the Threshold Manager activates the JOB_SESSION_CONTROL control, which essentially "shuts off" the job-session subsystem, preventing further logons.

Now, let's assume that half of the current jobs and sessions begin to logoff. With this, the usage of transient space begins to fall, and as soon as Threshold Manager notices that the usage is at 88% or below, it notifies the event of falling usage that has crossed comfortably below the 90% threshold. It refers to this event as fallback. Threshold Manager also deactivates the JOB_SESSION_CONTROL control which essentially "opens up" the job-session subsystem, allowing further logons.

8.2 Multiple Activations of JOB_SESSION_CONTROL

Multiple thresholds, mostly on different resources, may have JOB_SESSION_CONTROL configured as the control. The first threshold to experience the jumpoff event will cause the shutting down of the job-session subsystem. All subsequent activations of the same control have essentially no effect on the job-session subsystem, except that the subsystem keeps count of the activations. Conversely, for the fallback events, the last threshold to experience the fallback event, counting down the number of activations to zero, will cause the opening up of the job-session subsystem.

8.3 How are Logons Possible Despite Activation of JOB_SESSION_CONTROL?

Once the JOB_SESSION_CONTROL control is activated due to a jumpoff event, no more logons are possible. However, there is an exception. A user with the capability to logon with 'hipri' can still logon and use the system. The operator cannot selectively allow users to logon. However, the operator can forcefully deactivate the control by temporarily modifying all thresholds, whose JOB_SESSION_CONTROL controls are activated to 100% levels. Threshold Manager will instantly react to this modification by deactivating JOB_SESSION_CONTROL for each of the above thresholds. The last deactivation essentially opens up the job-session subsystem for all.

8.4 Using Threshold Manager

The main objective of Threshold Manager is to provide early warning for resource depletion by giving notifications when resource usage reaches configured thresholds. Every system environment is different. Some require higher usage of resources than others. In a given environment, not all resources are equally important because not all resources' usage may deplete, causing system abort. One of the ways in which to select important resources is based on the history of problems (system aborts) on a particular system. If this history reveals a problem related to depletion of one of the Threshold Manager resources, then that resource becomes an important resource to be closely monitored with the help of Threshold Manager.

There are no hard and fast guidelines in setting thresholds for any resources, let alone important resources as identified above. If a resource is causing system abort due to its depletion, you may want to set thresholds with notification turned on. This way warnings can be considered early on in time to avoid a system abort.

Also, there are no hard and fast guidelines for taking actions when Threshold Manager provides early warning in the form of notifications when the usage of a resource reaches a configured threshold. However, it is obvious that you may want to take those actions which may focus on reducing the system load affecting the usage of that resource. One of the actions which Threshold Manager will perform automatically, if JOB_SESSION_CONTROL is configured for a threshold, is not allow further jobs and sessions to logon to the system. However, you may take additional actions such as identifying the jobs and sessions which are not crucial to the business, and "killing" them. Another action which may be considered is to quiesce one or more applications for a short time until it is safe to put them back in full operation.

9. SharePlex/iX - System Availability On Disaster

With SPU Switchover/iX, a single copy of application data, which is primarily accessed by an application on the home system, can be switched to the alternate system. The application can run on the alternate system and access the same data. This is possible because both systems are in close proximity within the same data center.

When unexpected disasters occur, from a local disaster like vandalism to a widespread disaster, such as a major earthquake, both of the above systems and the disks containing the application data might be destroyed, leaving the business in a crippled state for a very long time. Many businesses see the need to maintain a redundant copy of their application data on another system, far away from the system on which the primary copy resides and is accessed by the application. The redundant copy should be dynamically synchronized with the primary copy, so that when a disaster destroys the primary system, the application will run on the secondary system where it will access the redundant copy of the data. SharePlex/iX is a sophisticated product incorporating the above solution, and also including other features which are often required to design an effective disaster recovery plan.

The key points of SharePlex are:

- SharePlex clusters together two or more geographically dispersed HP 3000 systems and provides a single-system view to their users, programmers, operators and administrators.
- SharePlex connects multiple computer systems together in a cluster through Wide and Local-area networks.
- SharePlex automatically replicates databases, programs, flat files, etc., in an effort to replicate the entire application environment elsewhere. In the event of a disaster, the replication of the applications and resources can be used.
- SharePlex allows the geographic separation of the system where the application is running, and the system where the application's data actually resides. Since all application resources, including programs, are replicated, if one of them is lost the other is used.
- Files, databases, etc. ,distributed in the cluster, are accessed through a central directory, so that applications are unaware of where their data resides. This is essential when one source of application data is destroyed and the other copy must now be "linked" to the application.
- Similarly, the other computing resources essential to running the business, such as printers, are also distributed in the cluster. These are managed centrally, so that if one printer, for example, is destroyed, the printing job is spooled to one of the other printers in the cluster.
- Just as an application can run on one system and its data can reside on another system, users of an application can logon to the local system and connect to the application as if it's running on the local system. In actuality it's running on a remote system. This is

how SharePlex presents to the users a unified view of the cluster.

- SharePlex allows central management of a cluster with HP OpenView System Manager. Many system functions related to recovery from the disaster are automated, so that system availability is maintained at the highest level possible.

10. HP OpenView System Manager

HP OpenView System Manager is a PC-based Windows environment for integrated management of one or more system consoles of HP 3000 systems networked together on a LAN. Its salient features are:

- An easy to use graphical user interface.
- Allows creation of icons corresponding to various subsystems and products. Events associated with these subsystems or products that require operator's attention cause these icons to light up and change colors.
- Event Message Review facility, a function that provides a list of messages which have been sent to your workstation and "logged" or accumulated against various icons. You can view them in chronological order or severity order. You can also see messages for a single icon or several icons at one time. In addition to viewing, this function allows you to print, annotate (add your own text) and remove messages.
- Provides Management Session facility, which is a terminal emulator that enables you to initiate an interactive session on your HP 3000. Once you have evaluated messages for a highlighted icon, you may wish to logon to your HP 3000 to respond to the message using a Management Session. This session is generally used to manage the system during normal day-to-day operations. A Management Session can only access the HP 3000 when the HP 3000 is up and running.
- Provides Control Console, which accesses the HP 3000 via a physical connection to the console port. For this reason, you can use the Control Console to perform tasks such as starting the system, running system diagnostics and other functions commonly performed when the HP 3000 is down. This is the difference between a Management Session and a Control Console.
- Provides the Event Message Log, which is a database that resides on the HP 3000. All error messages that are sent to your workstation are written to this database. You may access this database to generate reports regarding error messages logged to various icons on your System Map. This enables you to evaluate historical data and may help identify trends.
- Provides the Automated Response facility which enables you to automate the commands that would be issued in response to selected console messages. Once automated, the commands are invoked whenever necessary with no human intervention.

We have seen throughout this paper that even though we design our system for system availability, we still cannot avoid failure of some system resources which are critical to maintaining system availability. In some cases, the system availability is due to using two redundant hardware resources. The failure of one resource does not affect system availability, but does put the system, which depends minimally one of these resources, vulnerable to the failure of the second resource. This results in degradation of system availability. A classic example is mirrored disks. In such cases, immediate actions are sought from the operator in

order to fix the problem and put system back to the normal state with respect to the system availability.

In other cases, any failure in hardware or software can cause system availability to be affected. These events are even more critical since they require fixing the problem in order to restore the system availability to its original level. The longer it takes to alert the operator, the longer the system availability suffers.

To draw the operator's attention to all matters relating to system availability, do the following:

- Use OpenView Console to collect all messages from various subsystems which are related to system availability.
- Use a separate symbol or "icon" to represent the state of a particular subsystem which is important with respect to system availability. Since the messages are coded to highlight the icons with color, the operator will be better equipped to understand the relative severity of various events by looking at the System Map. System Map is a window collecting all icons corresponding to various subsystems of a particular MPE/iX system.
- For automatic response to events related to system availability, use Automatic Response Programming on HP 3000 and add appropriate decisions into the rule file EMGOAUTO.

A. NEWGROUP - A sample UDC

```
NEWGROUP  gname
anyparm PARMS=**

comment **************        Example  UDC        ************************
comment *   This UDC intercepts the NEWGROUP command and strips it of any    *
comment *   ONVS and HOMEVS keywords. If the user is logged onto the SYS     *
comment *   or TELESUP accounts then no stripping is done.                   *
comment *                                                                    *
comment *   Users logged onto FINANCE are 'homed' to DB1 user volume.        *
comment *   Users logged onto AUDIT are 'homed' to DB2 user volume.          *
comment *   All other users are 'homed' to a user volume named MISC_UV.      *
comment *                                                                    *
comment ********************************** Contributed by Walt McCullough **

comment     Enter the acct names for no checking in the var OKACCOUNTS
setvar NWGRP_OKACCOUNTS ',SYS,TELESUP,'
setvar NWGRP_POSACCT pos(',!hpaccount,',NWGRP_OKACCOUNTS)
setvar NWGRP_COMOUT  ''
setvar NWGRP_MUSTVS 'MISC_UV'

IF (hpaccount='FINANCE') THEN
  SETVAR NWGRP_MUSTVS 'DB1'
ELSE
  IF (hpaccount='AUDIT') THEN
  SETVAR NWGRP_MUSTVS 'DB2'
  ENDIF
ENDIF

IF ('!parms'='**') THEN
  setvar NWGRP_COMIN  ''
ELSE
  setvar NWGRP_COMIN  ups('!parms')
ENDIF


WHILE len(NWGRP_COMIN)>0
  WHILE (str(NWGRP_COMIN,1,1)=' ')
    setvar NWGRP_COMIN rht(NWGRP_COMIN,len(NWGRP_COMIN)-1)
  ENDWHILE

  SETVAR NWGRP_END    pos(';',NWGRP_COMIN)-1
  IF (NWGRP_END<=0) THEN
    SETVAR NWGRP_END  len(NWGRP_COMIN)
  ENDIF
```

```
  IF (pos('ONVS',NWGRP_COMIN)<>1 and pos('HOMEVS',NWGRP_COMIN)<>1)&
  OR (NWGRP_POSACCT>0) THEN
    setvar NWGRP_COMOUT  '!NWGRP_COMOUT;'+str(NWGRP_COMIN,1,NWGRP_END)
  ELSE
    ECHO ONVS OR HOMEVS IS NOT VALID FOR THIS ACCOUNT.
  ENDIF

  IF NWGRP_END=len(NWGRP_COMIN) THEN
    setvar NWGRP_COMIN  ''
  ELSE
    setvar NWGRP_COMIN str(NWGRP_COMIN,NWGRP_END+2,len(NWGRP_COMIN)-NWGRP_END+1
  ENDIF
ENDWHILE

IF (!NWGRP_POSACCT>0) THEN
  NEWGROUP !GNAME !NWGRP_COMOUT
ELSE
  CONTINUE
  NEWGROUP !GNAME !NWGRP_COMOUT;HOMEVS=!NWGRP_MUSTVS
  NEWGROUP !GNAME !NWGRP_COMOUT;ONVS=!NWGRP_MUSTVS
ENDIF
*******
```

Tutorial

MANAGING
MPE/3000 SYSTEM AVAILABILITY

Rakesh Patel
Hewlett-Packard Company

HEWLETT PACKARD

Managing MPE System Availability

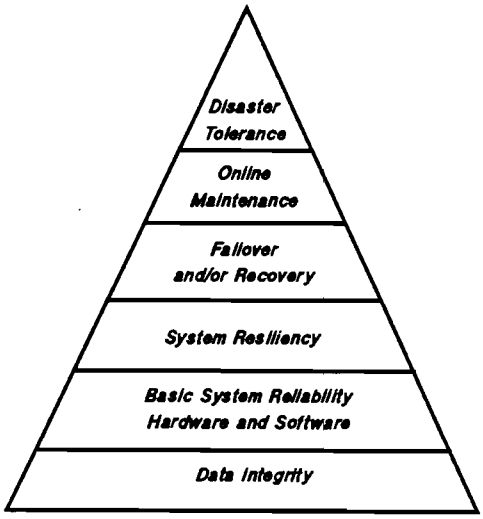
Defining Availability

- **Just system being up is not enough**

- **Measured in terms of end users ability to perform business functions**

System Availability

Hierarchy of Needs



- Protection of business from disasters
- More critical as system dependence increases

- Very important for 24x7 operation
- No downtime for planned maintenance

- Failover through component redundancy
- Fast recovery/minimum impact

- Withstand serious exception conditions without failing the system

- Foundation for commercial computing

- Foundation for commercial computing
- Foundation for system availability

HEWLETT PACKARD

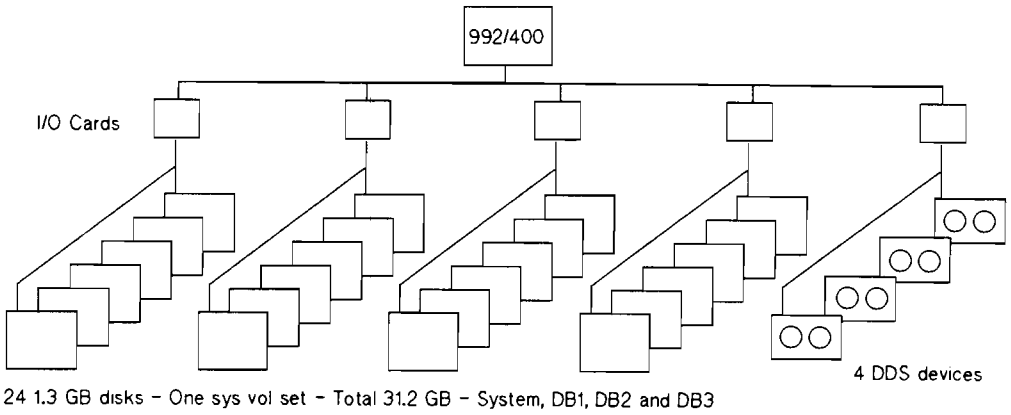
Managing MPE System Availability

Costs

- **Hardware**
- **Software**
- **Operations**
- **Planning**
- **Costs are like insurance policy**
  - Decide how much coverage you need
  - Cost of downtime vs. premiums paid

Non-High Availability Configuration

Example



24 1.3 GB disks - One sys vol set - Total 31.2 GB - System, DB1, DB2 and DB3

3 data bases, Total 20 GB
(4, 6, and 10 GB respectively)

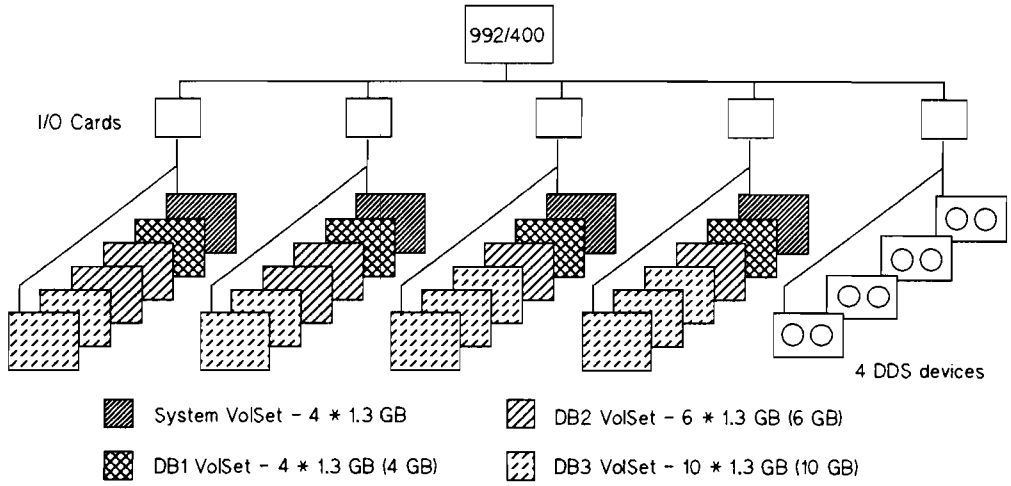
250 Users
~4 hours backup w/2:1 compression

Disk Failure Scenario

- Entire system unavailable on any disk failure
- Must always install (1 hour)
- Always restore all application data (4 hours)
- Application recovery on every application

Using User Volume Sets

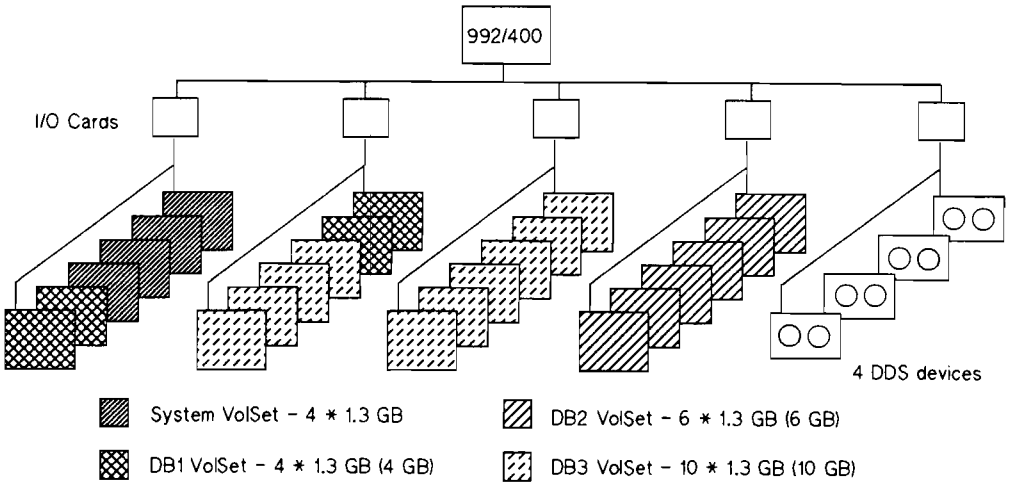
Example



HEWLETT PACKARD

Using User Volume Sets

Example



HEWLETT PACKARD

User Volume Set Advantages

- **Lose only one set and its applications on disk failure**
- **Reload only that subset of system data**
  - shorter restore time
  - other data available during restore
- **Only do install if failed disk in sys vol set**
  - don't have to reload user data

User Volume Set Advantages

- **Backup by volume set to decrease time files are unavailable**
- **Can avoid backing up system vol set, static data**
  - saves time
  - saves bookeeping and storage costs
- **Faster dump time**

User Vol Set Considerations

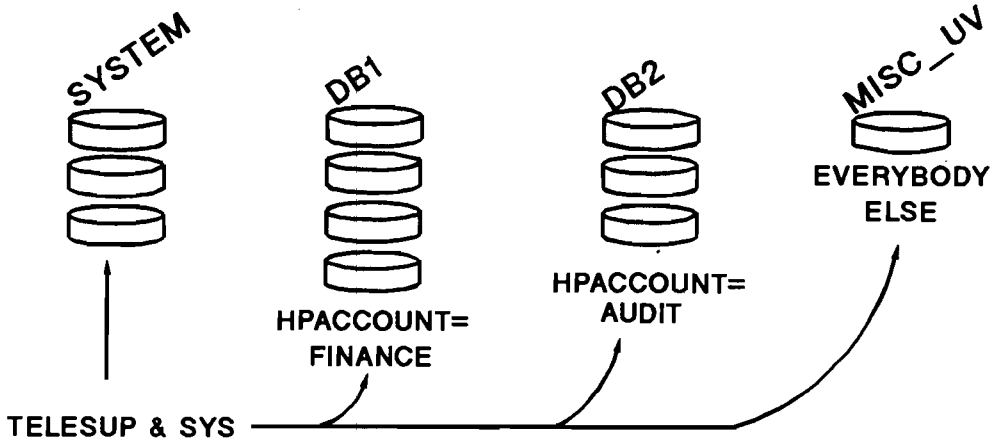
- **Costs associated with UVs**
  - additional hardware
  - possible operations changes
- **System Volume Set - how large should it be?**
  - spool file space
  - transient space
  - spindles for performance

NEWGROUP UDC

```
NEWGROUP  gname
anyparm PARMS=**

comment ***************          Example   UDC          ************************
comment *   This UDC intercepts the NEWGROUP command and strips it of any     *
comment *   ONVS and HOMEVS keywords. If the user is logged onto the SYS      *
comment *   or TELESUP accounts then no stripping is done.                    *
comment *                                                                     *
comment *   Users logged onto FINANCE are 'homed' to DB1 user volume.         *
comment *   Users logged onto AUDIT are 'homed' to DB2 user volume.           *
comment *   All other users are 'homed' to a user volume named MISC_UV.       *
comment *                                                                     *
comment ************************************  Contributed by Walt McCullough  **

comment     Enter the acct names for no checking in the var NWGRP_OKACCOUNTS
setvar NWGRP_OKACCOUNTS ',SYS,TELESUP,'
setvar NWGRP_POSACCT pos(',!hpaccount,',NWGRP_OKACCOUNTS)
setvar NWGRP_COMOUT  ''
setvar NWGRP_MUSTVS 'MISC_UV'

IF (hpaccount='FINANCE') THEN
  SETVAR NWGRP_MUSTVS 'DB1'
ELSE
  IF (hpaccount='AUDIT') THEN
  SETVAR NWGRP_MUSTVS 'DB2'
  ENDIF
ENDIF

IF ('!parms'='**') THEN
  setvar NWGRP_COMIN  ''
ELSE
  setvar NWGRP_COMIN  ups('!parms')
ENDIF


WHILE len(NWGRP_COMIN)>0
  WHILE (str(NWGRP_COMIN,1,1)=' ')
    setvar NWGRP_COMIN rht(NWGRP_COMIN,len(NWGRP_COMIN)-1)
  ENDWHILE

  SETVAR NWGRP_END    pos(';',NWGRP_COMIN)-1
  IF (NWGRP_END<=0) THEN
    SETVAR NWGRP_END  len(NWGRP_COMIN)
  ENDIF

  IF (pos('ONVS',NWGRP_COMIN)<>1 and pos('HOMEVS',NWGRP_COMIN)<>1)&
  OR (NWGRP_POSACCT>0) THEN
    setvar NWGRP_COMOUT  '!NWGRP_COMOUT;'+str(NWGRP_COMIN,1,NWGRP_END)
  ELSE
    ECHO ONVS OR HOMEVS IS NOT VALID FOR THIS ACCOUNT.
  ENDIF

  IF NWGRP_END=len(NWGRP_COMIN) THEN
    setvar NWGRP_COMIN  ''
  ELSE
    setvar NWGRP_COMIN str(NWGRP_COMIN,NWGRP_END+2,len(NWGRP_COMIN)-NWGRP_END+1)
  ENDIF
ENDWHILE

IF (!NWGRP_POSACCT>0) THEN
  NEWGROUP !GNAME !NWGRP_COMOUT
ELSE
  CONTINUE
  NEWGROUP !GNAME !NWGRP_COMOUT;HOMEVS=!NWGRP_MUSTVS
  NEWGROUP !GNAME !NWGRP_COMOUT;ONVS=!NWGRP_MUSTVS
ENDIF
*******
```

NEWGROUP UDC Example

User Volume Setup



HEWLETT PACKARD

User Vol Set Considerations (Cont.)

- **Size of data sets**
  - which to combine
  - which to separate (mirroring of critical data)
- **Performance**
  - application performance on fewer spindles
  - match applications disk I/O rate

User Vol Set Considerations (Cont.)

- **Size of existing disks**
  - mixing small (existing) and large capacity disks
- **Capacity Management**
  - free space mgt. less straightforward
  - multiple pools of free space
- **Applications cannot span volume sets**

Disk Failure with User Vols

- **Member of DB2 volume set**
  - DB2 application unavailable
  - may bring system down
  - restore only DB2 data
  - other applications available during DB2 restore and application recovery

Disk Failure with User Vols (Cont.)

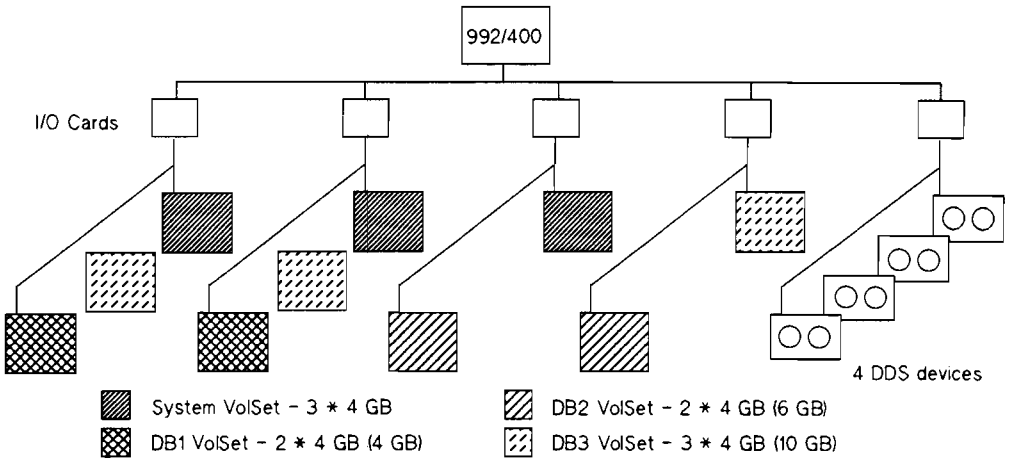
- **Member of system volume set**
  - entire system unavailable
  - must install
  - no reload of user data

Disk Arrays

- **Protection against mechanism and media failure**
- **Single points of failure**
  - disk controller
  - I/O card
  - cables
- **Larger capacity has performance implications**

Using Disk Arrays

Example

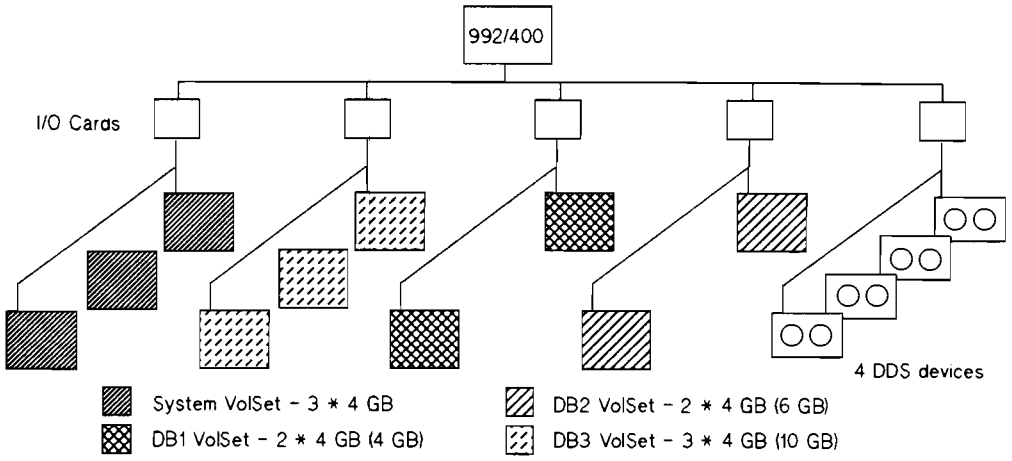


May need more disks for performance reasons

HEWLETT PACKARD

Using Disk Arrays

Example



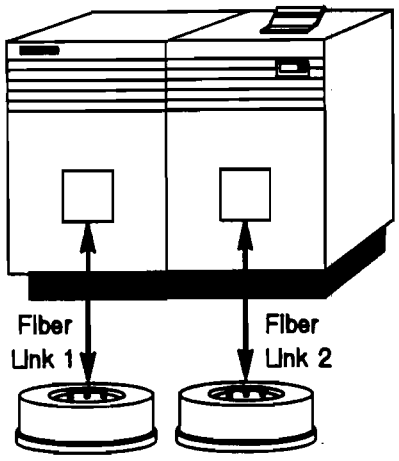
May need more disks for performance reasons

Disk Array Failure Scenario

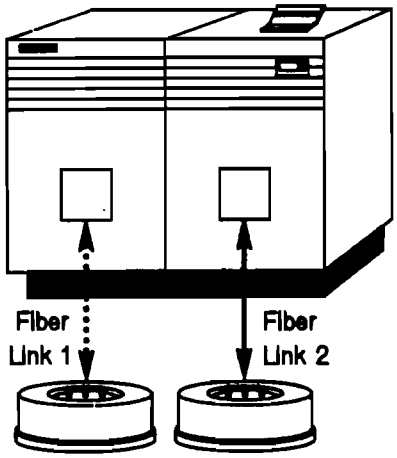
- **Mechanism or media failure – no impact**

- **Controller, card, or cable failure**
  - same as regular disk drive
  - may need to reload data

Mirrored Disk/XL



Data duplicated on mirrored disks



Disk failure

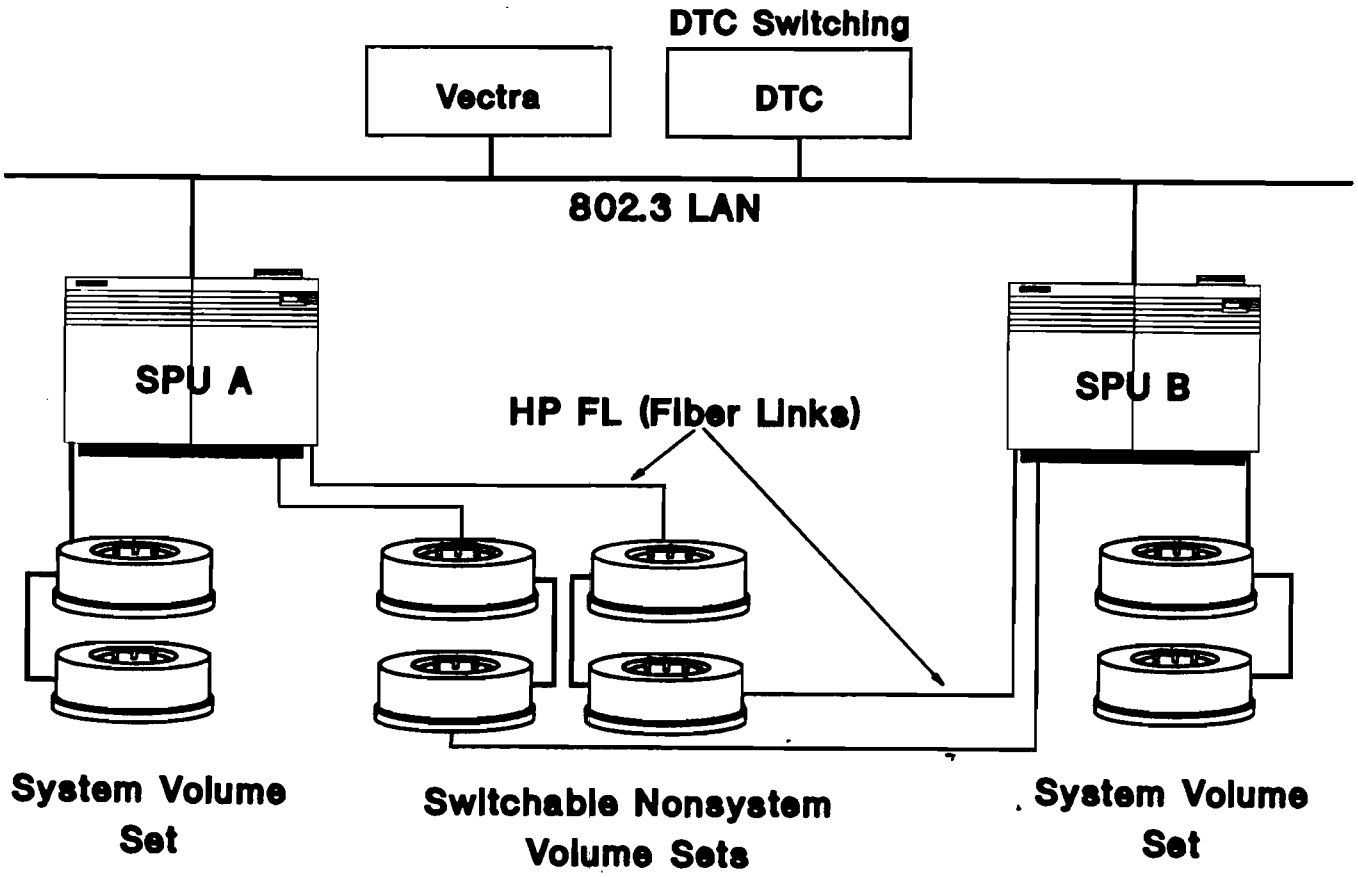
Access to second disk

Normal operating mode

- Transparent to users/applications
- Minimal overhead on disk writes
- Higher performance on disk reads
- Simple control and operation

When disk fails

- Transparent switch on failure
- Online replacement of disk



HEWLETT PACKARD

Backup and Recovery

- **Minimize direct impact of backup process**
- **Minimize time to recover when necessary**
- **Should consider them together**
  - how do backup processes affect recovery time

Dedicated Backup Considerations

- **Window of time available**
- **Speed and capacity of backup devices**
  - store time and restore time
- **Interleaving with multiple devices**
  - number of devices needed

Backup/Recovery Alternative Chart

	DDS			7980SX			7980S			Optical			1/2" Cartridge		
1 Device	2	*0.7*	**0.5**				0.14	*2.5*	**1.9**	20	*4.0*	**3.0**	2	*9.3*	**7.0**
DC 2:1	4	*1.3*	**1.0**	0.28	*5.0*	**3.8**	0.28	*5.4*	**4.1**	40	*7.0*	**5.3**	4	*13.0*	**9.8**
DC 3.5:1	7	*2.2*	**1.7**	0.5	*7.9*	**5.9**	0.5	*9.4*		70	*12.0*		7	*13.7*	
2 Device	4	*1.3*	**1.0**				0.28	*5.0*	**3.8**	40	*7.0*	**5.3**	4	*12.0*	**9.0**
DC 2:1	8	*2.6*	**2.0**	0.56	*10*	**7.5**	0.56	*10.8*	**8.1**	80	*12.3*	**9.2**	8	*18.0*	**13.5**
DC 3.5:1	14	*4.4*	**3.3**	1	*15.8*	**11.9**	1	*18.8*		140	*21.1*		14	*25.0*	
3 Device	6	*2.0*	**1.5**				0.42	*7.5*	**5.6**						
DC 2:1	12	*3.9*	**2.9**	0.84	*15*	**11.3**	0.84	*16.2*	**12.2**						
DC 3.5:1	21	*6.6*	**5.0**	1.5	*23.7*	**17.8**	1.5	*28.2*							
4 Device	8	*2.6*	**2.0**												
DC 2:1	8	*5.2*	**3.9**												
DC 3.5:1	28	*8.8*	**6.6**												

Red Unattended Capacity in GB

Blue *Dedicated Speed in GB/Hour*

Green **On-Line Speen In GB/Hour**

Backup Performance may vary depending on H/W platform and I/O configuration

HEWLETT PACKARD

Dedicated Backup Considerations (Cont.)

- **Backup by volume set**

  - file unavailable for less time, more control

  - can avoid backing up system vol set
    save time and money

  - easier bookkeeping and recovery

Dedicated Backup Considerations

Reducing Application Downtime

- **Business need: Want no application down for more than 2 hours**
- **Total backup time is 4 hours**
- **Backup by volume set**
  - total system backup time is about the same
  - No application down for more than 2 hours

On-line Backup

- **Mirrored disk split volume backup**
  - not recommended due to exposure time
- **TurboStore with on-line backup**
  - quiesce period
  - can be shorter and more controlled with vol sets
- **True (no quiesce) on-line backup for Allbase**

Recovery

- **Speed of restore devices**
- **Interleaving**
- **Backup by volume set**
  - shorter restore time
  - small impact on other applications
- **Applications recovery**
  - dynamic roll back recovery

Reducing MTTR for Software Failures

Autorestart/iX

- **Eliminate operator intervention and wait time**
- **Faster dump -- to disk**
- **Configurable mini-dump option**
  - less than 5 minutes dump time

Using Autorestart/iX

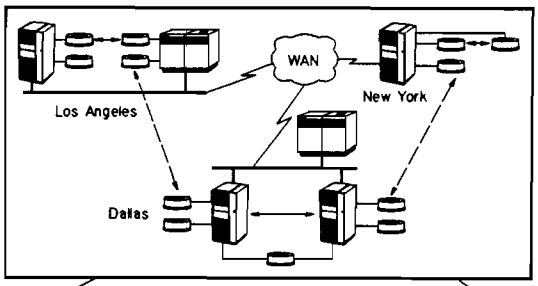
- **Dedicated volume set for dumps (not req.)**

- **Customized mini-dump script for system aborts**

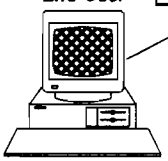
  - Don't take full dump for known problem

  - Take full dump only on second occurrance

SharePlex/iX
Local and Wide Area Cluster



End User



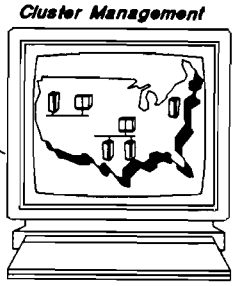
Single System Image
Transparent Access To:

- Files, Databases
- Programs, Applications
- System Resources

Cluster Benefits

- Disaster tolerance
- No single point of failure
- Shared system resources
- Investment protection
- Standard network hardware/software

Cluster Management



- Central Monitoring and Control
- Lights Out Remote Operations
- Automated Response & Recovery

HEWLETT PACKARD

NETBASE: Key SharePlex/iX Enabling Technology

Software to loosely couple multiple systems

Application Environment Replication (Shadowing)

- Databases, programs, flat files, etc. automatically replicated over local and wide area networks → Fast recovery and access to business critical data upon system or data center outage

Common File System (Network File Access)

- Files, databases, etc. distributed in the cluster are accessed through central directory → Shared data without special user or application awareness

Master Print/Spooling Management (NB Spool)

- Central printing environment and spooler management → Simple and practical cluster-wide shared printing/spooling

Shared Program Execution (AutoRPM)

- Automatic/simple remote process management interface → Programs reside anywhere in the cluster, appearing local to the user

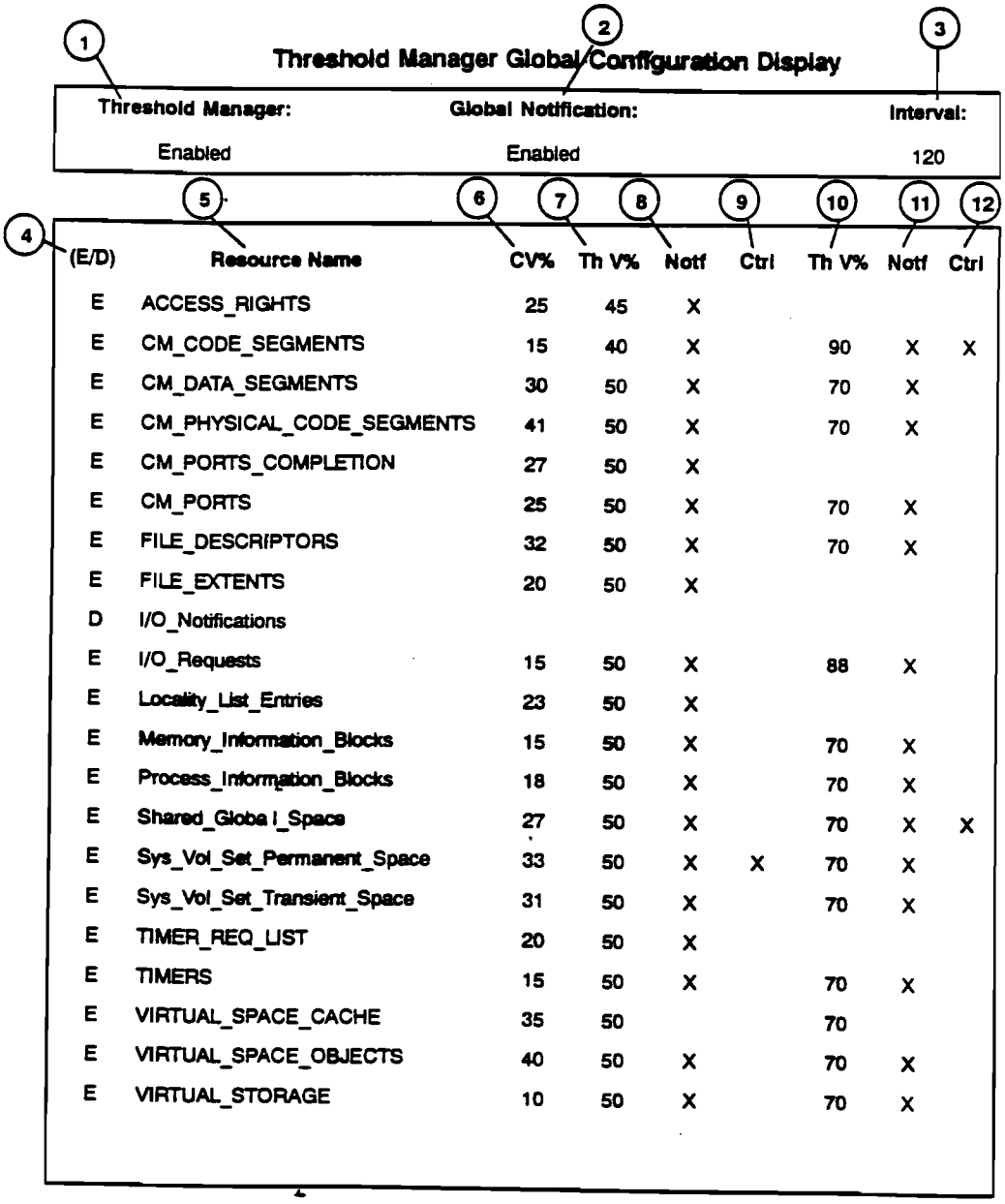
HEWLETT PACKARD

Commercial Systems Division
SPLEX-04.GAL BD;RO 1/93

HP3000 HA : Software Resiliency

Threshold Manager

- **Help prevent system aborts due to depletion of critical system resources**
- **Monitors usage levels of resources**
- **User configurable thresholds placed on resources**
- **Provides notification or prohibits logons when thresholds are crossed**


1
2
3
Threshold Manager Global Configuration Display
Threshold Manager:
Enabled
Global Notification:
Enabled
Interval:
120
4
5
6
7
8
9
10
11
12
(E/D)
Resource Name
CV%
Th V%
Notf
Ctrl
Th V%
Notf
Ctrl
E ACCESS_RIGHTS 25 45 X
E CM_CODE_SEGMENTS 15 40 X 90 X X
E CM_DATA_SEGMENTS 30 50 X 70 X
E CM_PHYSICAL_CODE_SEGMENTS 41 50 X 70 X
E CM_PORTS_COMPLETION 27 50 X
E CM_PORTS 25 50 X 70 X
E FILE_DESCRIPTORS 32 50 X 70 X
E FILE_EXTENTS 20 50 X
D I/O_Notifications
E I/O_Requests 15 50 X 88 X
E Locality_List_Entries 23 50 X
E Memory_Information_Blocks 15 50 X 70 X
E Process_Information_Blocks 18 50 X 70 X
E Shared_Global_Space 27 50 X 70 X X
E Sys_Vol_Set_Permanent_Space 33 50 X X 70 X
E Sys_Vol_Set_Transient_Space 31 50 X 70 X
E TIMER_REQ_LIST 20 50 X
E TIMERS 15 50 X 70 X
E VIRTUAL_SPACE_CACHE 35 50 70
E VIRTUAL_SPACE_OBJECTS 40 50 X 70 X
E VIRTUAL_STORAGE 10 50 X 70 X

HP3000 High Availability

On Line Configuration

Ability to add/delete devices and their config. parameters without a system reboot

- **Terminals**
- **Printers**
- **Disks**
- **Device Classes**

HP3000 HA : Software Resiliency

Subsystem Dump

- **On-line capture of process state and data**
- **Analyzed with existing tools, DAT and Macros**
- **Allows quick analysis of problem**

HP3000 HA : Software Resiliency

File System Resiliency

- **On-line quarantine of individual file**
- **File operations fail, system stays up**
- **On-line recovery of file -- Rename and Copy**
- **Uses Subsys Dump to save diagnostic info**