Oracle Data Warehousing SAS UNIX AlphaServer 8000 TruCluster

DIGITAL HiTest Notes

Part Number: EK–HDWUB–HN. A01

April 1997

Revision/Update Information:	This is a new manual.
Version:	Product Version 1.0

Digital Equipment Corporation
Maynard, Massachusetts

April 1997

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

This software is proprietary to and embodies the confidential technology of Digital Equipment Corporation and other parties. Possession, use, or copying of this software and media is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

While DIGITAL believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. DIGITAL is not responsible for any inadvertent errors. DIGITAL conducts its business in a manner that conserves the environment and protects the safety and health of its employees, customers, and the community.

The following are trademarks of Digital Equipment Corporation: AlphaServer, AlphaStation, DEC, DEC C++, DEC COBOL, DEC Fortran, DIGITAL, HSZ, OpenVMS, StorageWorks, ThinWire, TruCluster, and the DIGITAL logo.

Oracle, SQL*Net, and SQL*Plus are registered trademarks of Oracle Corporation

Oracle7, Oracle7 Enterprise Backup Utility, PL/SQL, and Oracle 64-bit Option are trademarks of Oracle Corporation

DB2, DRDA, ES/9000, and VTAM are registered trademarks of IBM Corporation

UNIX and DIGITAL UNIX are registered trademarks in the United States and other countries, licensed exclusively through X/Open Company Ltd.

SAS/SHARE*NET is a trademark of SAS Institute, Inc.

SAS, SAS/AF, SAS/ASSIST, SAS/CALC, SAS/CONNECT, SAS/EIS, SAS/ENGLISH, SAS/ETS, SAS/FSP, SAS/GRAPH, SAS/IML, SAS/INSIGHT, SAS/LAB, SAS/OR, SAS/QC, SAS/SHARE, SAS/SHARE*NET, SAS/SPECTRAVIEW, SAS/STAT, and SAS/TOOLKIT are registered trademarks of SAS Institute, Inc.

All other trademarks and registered trademarks are the property of their respective holders.

04/18/97 es3.2.0

Table of Contents

1 Introduction

2 Configuration Data

3 DIGITAL HiTest System Installation and Setup

4 Interoperability Tests and Results

5 System Limits and Performance Data

6 Problems and Resolution

A Detailed Hardware Configuration

Figures

Tables

Preface

This document provides an overview of DIGITAL HiTest Systems, and detailed technical information about inter-operability test results for the *Oracle Data Warehousing with SAS HiTest AppSet* on the *DIGITAL UNIX TruCluster ASE 8000 5/440 HiTest Foundation*.

Audience

Primary users of this document are DIGITAL and Partners sales representatives and customers. Secondary audiences include technical support personnel, product managers, and the personnel responsible for installing, setting up, and operating a DIGITAL HiTest System.

Road Map

This document contains the following sections:

1. Introduction — Provides a brief summary of the benefits of DIGITAL HiTest Suites and an overview of the suite covered in this document.

2. Configuration Data — Gives tables of configuration data about the hardware and software components that define the suite, and special configuration rules if any.

3. HiTest System Installation and Setup — Presents information useful when installing and tuning a DIGITAL HiTest System configured from this DIGITAL HiTest Suite.

4. Interoperability Tests and Results — Describes how tests were set up (including database organization), what data and programs were placed on what disks, and how the tests were run.

5. System Limits and Performance Data — Summarizes any system limitations or performance data that were identified during testing.

6. Problems and Solutions — Discuses any problems and solutions that were discovered during testing.

Appendix A Detailed Hardware Configuration — Contains a more detailed treatment of the hardware and software components listed in the Configuration Data section.

Feedback and Ordering Information

What our readers think of this or any other DIGITAL documentation is important to us. If you have any comments, no matter how great or small, we'd appreciate hearing from you. Send your comments to: *reader-comments@digital.com*.

Copies of this and other DIGITAL documents can be ordered by calling 1-800-DIGITAL. Please reference the document title and part number (EK-HDWUB-HN. A01) in your correspondence about this manual.

Publications

- *DIGITAL UNIX (formerly DEC OSF/1), Installation Guide*
  (July 1995, Digital Equipment Corp., Order No: AA-PS2DF-TE)
- *DIGITAL UNIX, Release Notes & Installation Instructions for Version 4.0A*
  (September 1996, Digital Equipment Corp., Order No: AA-ROKJA-TE)
- *POLYCENTER NetWorker Save and Restore Database Module for Oracle Administrator's Guide* (Digital Equipment Corp., Order No: AA-QV9BA-TE)
- *NetWorker Administrator's Guide, UNIX Version*
  (Digital Equipment Corp., Order No: AA-QH5CA-TE)
- *DEC SNA Peer Server Installation and Configuration Manual*
  (Digital Equipment Corp., AA-Q1P8C-TE)
- *DEC SNA APPC/LU6.2 Programming Interfaces for DIGITAL UNIX*
  (Digital Equipment Corp., AA-QQW3A-TE)
- *Oracle7 Server for DIGITAL UNIX Installation and Configuration Guide Release 7.2.3*
  (December 1995, Oracle Corp., Part No. A42515-1)
- *Oracle7 Server for DIGITAL UNIX 7.3.2.3*, (Part # A49350-1).
- *Oracle7 Server Administrator's Guide Release 7.2*
  (April 1995, Oracle Corp., Part No. A20322-2)
- *Oracle7 Server Tuning, Release 7.3* (June 1996, Oracle Corp., Part No. A32537-1).
- *Oracle Transparent Gateway for IBM DRDA for DIGITAL UNIX Installation and User's Guide Release 3.0.17* (1996, Oracle Corp., Part No. A42542-1)
- *Oracle7 Enterprise Backup Utility™ Administrator's Guide Release 2.0.12*
  (1996, Oracle Corp.)
- *Oracle7 Enterprise Backup Utility™ Installation Guide for DIGITAL UNIX, Release 2.0.12* (January 1996, Oracle Corp., Part No. A45307-1)
- *Installation Instructions for the SAS System under UNIX Environments, Release 6.11, TS040*, (Cary, NC: SAS Institute Inc., 1996).
- *DB2 Installation Guide* (IBM Corp., SC26-3456)
- *DB2 Messages and Codes* (IBM Corp., SC26-3268)
- *DB2 Reference for Remote DRDA Requesters and Servers* (IBM Corp., SC26-3282)
- *DB2 Utility Guide and Reference* (IBM Corp., GC26-3395)
- *VTAM Resource Definition Reference* (IBM Corp., SC31-6412)

Web Sites

- DIGITAL Software Partner Engineering, Database Technology Center Web site at `http://www-spe.pa.dec.com/`
- Digital Equipment Corporation Web site at `http://www.digital.com/`
- Oracle Corporation Web site at `http://www.oracle.com`
- Oracle Corporation Platforms Web site at `http://www.oracle.com/platforms/`
- SAS Institute Web site at `http://www.sas.com/`

1 Introduction

The DIGITAL HiTest Suite and Its Advantages

DIGITAL HiTest Suites are guidelines for configuring a set of pre-qualified computer systems. A HiTest Suite often contains all the hardware and software needed for a complete customer solution. DIGITAL HiTest Suites can be used as a basis for configuring systems that satisfy a wide set of customer requirements. Typically, suites target specific markets such as Data Warehousing or Enterprise Applications.

DIGITAL Product Management and Engineering select the components and design the configurations in each HiTest Suite to ensure high system reliability, application performance, and upgradability. A suite's hardware and software components have been successfully tested for interoperability.

A HiTest Suite specifies allowed ranges of hardware and software components, as well as each component's part number, description, and revision information. These specifications are listed in the *DIGITAL HiTest Template*.

The components in a HiTest Suite are organized into two groups, *the DIGITAL HiTest Foundation* and the *DIGITAL HiTest AppSet*. The HiTest Foundation includes the hardware, operating system, middleware, and database software. The HiTest AppSet includes the software specific to one class of customer solutions.

Configuring a DIGITAL HiTest Suite is easy. Simply select components from the HiTest Template to configure a DIGITAL HiTest System. Any system configured as specified in the DIGITAL HiTest Template can be called a DIGITAL HiTest System.

The HiTest Suite is documented in the *DIGITAL HiTest Notes*. The HiTest Notes list the HiTest Foundation and HiTest AppSet components. HiTest Notes also describe the testing of the suite and include configuration details, installation instructions, tuning parameters, problems encountered and their solutions, and system diagrams.

Some components listed in the HiTest Foundation or AppSet may be optional. If the minimum quantity is zero (0), then the component is optional. If the minimum quantity is one or more, then you must order at least the minimum quantity.

The maximum quantities represent the largest group of components that were tested for interoperability with all the other components in the suite. Although it may be possible to place more than the specified maximum quantity of a component on a DIGITAL system, extensive interoperability testing was not done at that level and such a system would not be considered a DIGITAL HiTest System.

You can select any combination of components with quantities ranging from the minimum to the maximum specified. Occasionally, special configuration rules give further guidance or restrict configurations. These rules appear in the Configuration Data section of the HiTest Notes.

A customer can include the suite specified hardware and software they need and then layer on additional software. Other types of hardware, called *add-on hardware*, can also be added to a DIGITAL HiTest System. The add-on hardware is specified in the Configuration Data section of the HiTest Notes, and in the HiTest Systems Web Pages, available through the following URLs:

`http://cosmo.tay.dec.com`	(Intranet)
`http://www.partner.digital.com:9003`	(Internet)

Even though the customer may install application software that is not specified in the suite, the customer and DIGITAL still experience the advantages of knowing that all of the suite base hardware and software interoperates correctly. Of course, the full benefit of configuring a system from a HiTest Suite is obtained when the system includes only specified HiTest Foundation and AppSet components.

Overview of the DIGITAL HiTest Suite for UNIX Oracle Data Warehousing with SAS

The *Oracle Data Warehousing with SAS HiTest AppSet* includes the following software components:

- DIGITAL UNIX, Version 4.0A
- Oracle7 Server, Version 7.3.2.3.0
- The SAS System, Version 6.11

This DIGITAL HiTest Suite capitalizes on Digital products’ 64-bit architecture, very large memory and large database capabilities to provide unequalled performance for high-end data warehousing solutions.

DIGITAL HiTest offers a range of one to three 8200 or 8400 systems. Each system can include from one to four dual processors.

2
Configuration Data

The hardware and software specified in this suite describes a large Oracle with SAS data warehousing solution. The hardware configuration consists of a three-node AlphaServer 8000 TruCluster, each node including eight CPU's and 8 GB of memory, with over one terabyte of shared storage.

Configuration data includes the hardware and software components as tested. Special configuration considerations, and firmware levels tested, are covered in Chapter 3, *System Installation and Setup*.

Hardware and Software Components

Table 2-1 identifies the hardware and software components that can be configured using the DIGITAL HiTest Suite for UNIX Oracle Data Warehousing with SAS on an AlphaServer 8000 5/440 TruCluster Available Server Environment.

Two base systems are specified. The memory capacity of the suite is eight GB per node. A wide selection of options can be added to create the desired configuration from the DIGITAL HiTest Suite.

Table 2-1: DIGITAL HiTest Template

Oracle Data Warehousing with SAS *HiTest AppSet*
DIGITAL UNIX TruCluster ASE 8000 5/440 *HiTest Foundation*

For documentation and updates: http://cosmo.tay.dec.com *and* http://www.partner.digital.com:9003
For a hard copy of this suite's HiTest Notes, order EK-HDWUB-HN.

Line Item	**Description**	**Part Number**	**Tested Range**	
			Min	**Max**
	AppSet Software			
1	**SAS INSTITUTE LAYERED PRODUCTS** **The SAS System, Release 6.11** (See below for detail components)		1	1
	Foundation Hardware			
2	*HiTest recommends ordering 1 to 3 identical systems:* **AS8200 5/440 Expanded Base Server** **AS8400 5/440 Expanded Base Server** *Hardware includes*: • Two 5/440 MHz CPUs; 4-MB cache • I/O module with four I/O channels (KFTHA-AA) • 2 GB of memory • RZ28D-VW 2 GB 3.5" SCSI disk • One BN21K-XX PCI FWD SCSI cable • One DWZZB-VW SCSI signal converter • 600 MB CD-ROM drive • One KZPSA-BB, PCI FWD SCSI controller • KZPAA-AA, PCI FNSE SCSI controller, for CD-ROM connection only • BN21H-02 2-meter SCSI cable • Shielded console cable *8200 Only Hardware includes*: • BA656 Internal Storage Drawer & BA356-JB SCSI-2 16-bit wide StorageWorks shelf. • One DWLPB-CA PCI Shelf Mount Box. • Universal single phase power. 48 VDC power supply does not include power cord. • DE435-AA or DE500-XA 10/100 Mbit DIGITAL Etherworks 32-bit Network Interface Card *8400 Only Hardware includes*: • BA660-AB StorageWorks Plug-in-unit. • One DWLPB-AA, PCI 12 slot Plug-in-unit (8400) • Two H7263-AC or H7263-AD non-BBU capable 48 VDC power regulators. Three-Phase power subsystem includes power cord. • DE500-XA 10/100 Mbit DIGITAL Etherworks 32-bit Network Interface Card *Software includes*: • Factory Installed Software • DIGITAL UNIX Operating System base license • DIGITAL NAS Base Server 200 software	**DA-282FF-B9** **DA-292FF-BA**	1	3
	In general the minimum configuration is equivalent to one third of the following products required for the maximum system configuration:			
3	**Additional dual 5/440 CPU module**	**756P2-AX**	2	9
4	**PCI to Memory Channel Controller**	**CCMAA-BA**	0	6

Oracle Data Warehousing with SAS *HiTest AppSet*

DIGITAL UNIX TruCluster ASE 8000 5/440 *HiTest Foundation*

For documentation and updates: http://cosmo.tay.dec.com *and* http://www.partner.digital.com:9003
For a hard copy of this suite's HiTest Notes, order EK-HDWUB-HN.

Line Item	Description	Part Number	Tested Range	
			Min	Max
5	**Additional 2147 MB Memory Module**	**MS7CC-FA**	1	9
6	**Additional PCI Plug-in Unit with One PCI Box for System Cabinet**	**DWLPB-AA**	1	3
7	**PCI Expansion Box Mounted in DWLPB-AA**	**DWLPB-BA**	2	6
8	**CD & Tape SCSI Cables**	**BN21H-02**	1	3
9	**PCI to FWD SCSI host adapter** (16 per node)	**KZPSA-BB**	16	48
10	**FWD to FWSE or FNSE SCSI converter**	**DWZZB-VW**	4	8
11	**PCI NI Adapter**	**DE435-AA**	1	3
12	**PCI FDDI Adapter**	**DEFPA-AB**	1	1
13	**FDDI Cable**	**BN34D-10**	1	1
14	**3, 5, 10 or 20m SCSI Cables** (Number of cables required depends on the specific configuration)	**BN21K-****	18	43
15	**MODEL 450-C4, StorageWorks 464 GB Enterprise Storage Array** *Hardware includes*: • SW822-GA 60 HZ Cabinet • HSZ52-AJ controller pairs (3) • RZ29B-VA 4.3 GB wide disks with snap in carriers (84) • BN21K-20 host SCSI-2 cables (3) • StorageWorks Command Console V1.0 • media & documentation • MCS installation services • SI RAID initialization & configuration service	**SW8T2-AA**	4	4
16	**Memory Channel Hub**	**CCMHA-AA**	0	2
17	**Memory Channel Link Cable**	**BC12N-10**	0	6
18	**VT510 Console Terminal**	**VT510-****	1	1
19	**Keyboard**	**LK411-****	1	1
20	**52 Cart DLT Automated Tape Library with 4 DLT TZ88 Tape Drives**	**TL812-BA**	1	2

Oracle Data Warehousing with SAS *HiTest AppSet*

DIGITAL UNIX TruCluster ASE 8000 5/440 *HiTest Foundation*

For documentation and updates: http://cosmo.tay.dec.com *and* http://www.partner.digital.com:9003
For a hard copy of this suite's HiTest Notes, order EK-HDWUB-HN.

Line Item	Description	Part Number	Tested Range	
			Min	**Max**
	AppSet Software Detail			
	SAS INSTITUTE LAYERED PRODUCTS *For SAS Institute products contact: 919-677-8000* **The SAS System, Release 6.11** *License to include:* • Base SAS Software: DIGITAL UNIX Alpha SAS • SAS/ACC-ORACLE: DIGITAL UNIX Alpha SAS/ACCESS Interface to ORACLE • SAS/AF • SAS/ASSIST • SAS/CALC • SAS/CONNECT • SAS/EIS • SAS/ENGLISH • SAS/ETS • SAS/FSP • SAS/GRAPH • SAS/IML • SAS/INSIGHT • SAS/LAB • SAS/OR • SAS/QC • SAS/SHARE • SAS/SHARE*NET • SAS/SPECTRAVIEW • SAS/STAT • SAS/TOOLKIT	Included with item 1.	1	1
	Foundation Software Detail			
	DIGITAL UNIX, Version 4.0A	Included with Item 1	1	1
	Asterisks (*) in part numbers below denote variant fields. For additional information, refer to the appropriate price book. **DIGITAL CD-ROM Software Library for DIGITAL UNIX**	**QA-054A*-***		
	DIGITAL UNIX Associated Products CD-ROM	**QA-MT4AA-H8**		
21	**TruCluster Available Server, Version 1.4**	**Q*-05SA*-***		1
22	**POLYCENTER NetWorker Save and Restore (NSR) for DIGITAL UNIX, Version 4.2A** *(Available only as part of the DIGITAL UNIX Layered Product Subscription service on CD-ROM)* *License includes right to use*: • **SCSI CAM Layered Components, Version 3.1A** *(Distributed on UNIX CD-ROM: QA-054A*-**)*		1	1
23	**POLYCENTER NetWorker Save and Restore Database Module for Oracle (DMO), Version 1.0**	**Q*-3P4A*-***	1	1

Oracle Data Warehousing with SAS *HiTest AppSet*
DIGITAL UNIX TruCluster ASE 8000 5/440 *HiTest Foundation*

For documentation and updates: http://cosmo.tay.dec.com *and* http://www.partner.digital.com:9003
For a hard copy of this suite's HiTest Notes, order EK-HDWUB-HN.

Line Item	Description	Part Number	Tested Range	
			Min	Max
	Products distributed on DIGITAL Software Library for DIGITAL UNIX on CD-ROM: *QA-054A*-***			
24	**DEC C++ for DIGITAL UNIX Systems, Version 5.1**	**QL-MTRA*-****	1	1
25	**DEC COBOL for DIGITAL UNIX Systems, Version 2.4**	**QL-2BZA*-****	1	1
26	**DEC SNA Peer Server, Version 1.3**	**QL-2D7A*-****	1	1
27	**DIGITAL Fortran for DIGITAL UNIX Alpha Systems, runtime library, Version 3.6A**	**QL-MV2A*-****	1	1
28	**DIGITAL SNA APPC/LU6.2 Programming Interface for DIGITAL UNIX, Version 3.0**	**QL-4URA*-****	1	1
29	**Micro Focus COBOL for DIGITAL UNIX, Version 4.0**	**QL-213A*-****	1	1
30	**SNA APPC/LU6.2 Runtime for OSF/1, Version 3.0**	**QL-4UQA*-****	1	1
	For Oracle products contact: 1-800-ORACLE1 (800-672-2531) or e-mail: infodec@us.oracle.com **Oracle7 Server for DIGITAL UNIX Components:**			
31	**Oracle7 Server (RDBMS), Release 7.3.2.3.0**		1	1
32	**Oracle Server Manager, Release 2.3.2.0.0**		1	1
33	**Oracle7 Parallel Server Option, Release 7.3.2.3.0**		1	1
34	**SQL*Plus, Release 3.3.2.0.0**		1	1
35	**Oracle WebServer, Release 1.0.2.0.0**		1	1
36	**Oracle7 Enterprise Backup Utility (OEBU), Release 2.0.12.4**		1	1
37	**Oracle7 Transparent Gateway for IBM DRDA for DIGITAL UNIX, Release 3.0.17**		1	1
38	*For CA products call 800-225-5224* **COMPUTER ASSOCIATES POLYCENTER Pathdoctor, Version 1.0**		1	1
39	*For IBM products call 800-426-2255* **IBM MVS/ESA (Mod 2) Version 5, Rel. 2**	.	1	1
40	**IBM ACF/VTAM Version. 4, Rel. 3**		1	1
41	**IBM ACF/NCP Version 7, Rel. 3**		1	1
42	**IBM DB2 Version 4, Rel. 1**		1	1

For further details on the hardware configuration, see Appendix A.

Special Configuration Rules

For special configuration considerations encountered in testing, see Chapter 3, DIGITAL HiTest System Installation and Setup.

3 DIGITAL HiTest System Installation and Setup

DIGITAL, Oracle, and the SAS Institute assembled and stress tested combinations of the hardware, operating systems, database and application/system management software in a one TB database environment. Testing included selected software products in a typical data warehousing/decision support solution.

The *Oracle Data Warehousing with SAS HiTest AppSet on the DIGITAL UNIX TruCluster ASE 8000 5/440 HiTest Foundation* is based on released versions of DIGITAL UNIX, Oracle7, and the SAS System. Through ongoing testing DIGITAL, Oracle, and the SAS Institute will assure customers that future hardware and software releases continue to work together as an integrated product suite.

The focus of this *DIGITAL HiTest Suite* is on a complex Consumer Packaged Goods (CPG) Data Warehousing/Decision Support environment. The assurance testing conducted on the base software layer components (that is Database, System Management, Networking, Database Gateways to IBM, and so forth) has broad application across diverse database environments. For example, what is learned through the Enterprise Solution Initiative is directly relevant to other product integration efforts.

This one terabyte database environment consisted of a 419.4 GB Oracle7 database, a 450 GB SAS database, and 168 GB of RAID5 storage sets for loading data.

The 419.4 GB Oracle7 database was configured based on the star-schema CPG demo database, and queries currently available from the Oracle Data Warehouse Consumer Products group. We modified the build of the CPG database to generate a larger database, but neither the star-schema design nor the queries from the CPG demo were changed.

The intent of large database stress testing is to ensure that the most likely needs of a combined DIGITAL/Oracle customer have been tested, and to ensure that similar performance can be achieved, and assured to work, at other large customer sites with similar configurations.

Introduction: DIGITAL AlphaServer 8000 TruCluster HiTest System

The maximum hardware configuration of the HiTest DIGITAL UNIX TruCluster ASE 8000 Oracle Data Warehousing environment includes three AlphaServer systems in a TruCluster Available Server configuration with Memory Channel, RAID-5 disk subsystem, and tape backup hardware, as well as network links for a remotely located IBM ES9000-9121, and associated network hardware. The AlphaServer systems are each configured with four dual 5/440 MHz CPUs, and eight GB of RAM. Total mass storage available is 1,858 GB, in a RAID-5 configuration in four StorageWorks cabinets.

AlphaServer 8000 TruCluster HiTest System Configuration Data

CPU System Hardware

The CPU system hardware consists of a cluster of three AlphaServer 8400 system cabinets, each containing four Alpha dual-microprocessor 21164/440 MHz CPUs. This chapter describes the basic system hardware components required to replicate the test bed. CPU system hardware components used included:

Data Warehousing Test Environment

The ESAC test environment includes the base AlphaServer 8400 TurboLaser system with StorageWorks, connected via T1 link to a remote DB2 database on an IBM ES9000-9121 mainframe, through an Alpha 2100 front-end.For complete detailed drawings and specifications of the DIGITAL UNIX TruCluster ASE 8000 5/440 HiTest Foundation, see Appendix A “Detailed Hardware Configuration”.

DIGITAL HiTest System Installation and Setup

On our AlphaServer 8000 system, we installed and configured DIGITAL and Oracle software in the following sequence. Except as noted below, we used the recommended configuration and settings, according to the vendor documentation cited.

1. Configure DIGITAL StorageWorks (Page 3–2)
2. Install and Set Up Operating System and Network (Page 3–4)
3. Install the *DIGITAL HiTest Suite*:
   DIGITAL Layered Products (Page 3–5)
   Oracle Layered Products (Page 3–5)
   SAS Institute Layered Products (Page 3–6)
4. Install and Set Up Gateway Software
   to Remote DB2 Database (Page 3–7)
5. Set VLM64 Parameters (Page 3–14)
6. Create Consumer Packaged Goods Database (Page 3–18)

Configuring DIGITAL StorageWorks

We configured our HSZ40 StorageWorks Array Controllers using the StorageWorks HSZ40 Array Controller Utility for DIGITAL UNIX, according to the *StorageWorks Array Controllers: HS Family of Array Controllers User’s Guide*
(order number: EK-HSFAM-UG1).

Configuring the RAID-5 Stripe Sets

We configured our storage in RAID-5 stripe sets. Each of our stripe-sets consists of four RZ29Bs, at 4.3 GB per device, for a total of 17.2 GB of disk space with approximately 12.9 GB exported usable space per RAID set.

Where *n* is the number of drives:

$$\frac{(n\text{-}1)}{n} \quad * \quad (\ n * \text{size of each device}) \quad = \text{ disk space per RAID set}$$

$$\frac{(4\text{-}1)}{4} \quad * \quad (\ 4 * 4.3\text{ GB}) \quad = \text{ 12.9 GB usable space}$$

Calculating Maximum Possible Chunk Size

The maximum chunk size possible for a RAID-5 set is determined by the number of members in the RAID set. Beginning with the maximum chunk size setting for our HSZ controllers (2048 disk blocks of 512 bytes each), and based on our 32 Kbytes db_block_size, the maximum possible chunk size for our configuration is calculated using the formula below.

Table 3-1: Calculation for Maximum Possible Chunk Size

Variable	Calculation	Comment
db_block_size	= 32 Kbytes	= 32,768 bytes.
max chunk size	$= \frac{2048}{(n-1)}$	where *n* = # of RAIDset disks.
	$= \frac{2048}{(4-1)}$ = 682.7	Maximum HSZ chunk size, divided by RAID-5 stripe sets of 4 disks each less 1.
max chunk bytes	= disk bytes per block * max chunk size	
	= 512 * 682 = 349,184 bytes	Disk block size (512 bytes), times our maximum chunk size.
whole db blocks/chunk	$= \frac{\text{chunk bytes}}{\text{db\_block\_size}}$	
	$= \frac{349{,}184}{32{,}768}$ = 10.65 = 10	Our maximum chunk size possible in whole db blocks per chunk (rounded down).

We chose a chunk size of 128 Kbytes (4 db_blocks at 32 Kbytes per block) based on the maximum possible I/O size of Oracle during a sequential scan. Our chunk size of 128 Kbytes, or 4 db blocks/chunk, is within the maximum possible calculated above.

Setting Chunk Size on HSZ Console

For best performance, the chunk size for the Oracle7 database raw devices should be computed and set accordingly. Based on the maximum chunk size calculated above, we used the fixed chunk size of 256 disk blocks (128 Kbytes) for our configuration. Based on our 32 Kbytes db_block_size, and four db_blocks per chunk, calculated using the formula below.

$$\frac{\text{(db\_block\_size) * (\# of db blocks per chunk)}}{512}$$

Table 3-2: Calculation for Determining Chunk Size

Variable	Value	Comment
db_block_size	= 32,768 Bytes	32 Kbytes
# of whole db blocks per chunk	= 4	
total db block bytes/chunk	= whole db blocks/chunk * db_block_size	
	= 4 * 32,768 = 131,072 Bytes	 128 Kbytes
disk blocks/chunk	$= \frac{\text{total db block bytes/chunk}}{\text{bytes/block}}$	
	$= \frac{131,072}{512}$	
chunk size	= 256 disk blocks	setting for chunk size in terms of blocks

Enabling Write-Back Cache

We recommend the HSZ Write-back Cache be enabled. The Write-back Cache allocates cache memory to both read and write operations, permitting processing to continue without waiting for I/O completions. Using the Write-Back Cache performance feature we saw improved write operation performance, and reduced database/datafile creation times. The Write-back Cache option provides significantly improved I/O throughput.

Operating System and Network

We installed and configured DIGITAL UNIX Version 4.0A according to the *DIGITAL UNIX Release Notes and Installation Instructions for Version 4.0A* (order number: AA-ROKJA-TE, September 1996), except as noted below.

Selecting Deferred Swap Mode (Lazy Swap)

When installing DIGITAL UNIX, the default swap mode is Immediate Swap. We recommend using Deferred Swap Mode to reduce the amount of swap space required, and to reduce the system overhead required for mapping the address space during the creation of processes:

```
mv /sbin/swapdefault /sbin/swapdefault.old
```

Configuring Additional Swap Space

When installing DIGITAL UNIX, the primary swap space defaults to partition b, at the required size of *exactly* 128 MB. If the primary swap space is set at any other size, crash dumps will not be generated properly. A secondary swap space may also be assigned during installation, to enable complete crash dump when using eight GB of memory. We assigned a secondary swap space of 8026 MB. After installation, we added a tertiary swap space of 6020 MB, as detailed in Table 3-15. Additional swap space is assigned by including entries in `/etc/fstab`. An example of such an entry is:

```
/dev/rz0b swap1 ufs sw 0 2
```

Configuring swap space greater than 2 GB is discussed in detail in *DIGITAL UNIX Release Notes*, for DIGITAL UNIX Version 3.2C, (order number: AA-PS2BF-TE, July 1995).

Applications

Installing DIGITAL Layered Products

The full list of DIGITAL layered products in the *Oracle Data Warehousing with SAS HiTest AppSet*, along with the versions tested, appears in the table in chapter 2.

POLYCENTER system management products installed include NetWorker Save and Restore (NSR) for DIGITAL UNIX, NSR Database Module for Oracle (DMO), and Pathdoctor.

NSR must be installed before DMO; otherwise there is no preferred sequence for installation of these products. In order to have the proper privileges, the user who will be performing restores using the *Oracle7 Enterprise Backup Utility* must be included in the Operator Group set in `/etc/passwd`. We followed the installation instructions provided with each of the products.

Installing Oracle Layered Products

Installing and Setting up the Oracle RDBMS

We installed Oracle7 Server, Version 7.3.2.3, including the Parallel Query Option, PL/SQL, SQL*Plus, and SQL*Net, using Oracle Installer from the Oracle7 Server CD. We subsequently applied the *Oracle7 Server Release Update for DIGITAL UNIX*, *Release 7.3.2.3, the DIGITAL UNIX V4.0 Compatibility Patch,* and the *Large File* and *Shared Memory* patches to the *Oracle7 Enterprise Backup Utility* (OEBU), as detailed below.

Oracle Patches

Oracle7 Server Release Update for DIGITAL UNIX, Release *7.3.2.3: Patch* 7.3.2.3

We installed the Version 7.3.2.3 mandatory Release Update (patch 7.3.2.3), using Oracle Installer from the Release 7.3.2.3 CD, bringing our Oracle Version 7.3.2.2 VLM platform to Version 7.3.2.3, according to the *Oracle7 Server Release Update for DIGITAL UNIX*, *Release 7.3.2.3* (November 1996, Oracle Corp., Part No. A49350-1).

Oracle7 Server / Digital UNIX V4.0 compatibility: Patch # 424307

We installed Oracle Patch # 424307 to update the Oracle executable and Motif Server manager to be compatible with our DIGITAL UNIX 4.0 system. Oracle Patch # 424307 is required to ensure compatibility of any Oracle7 Server Version 7.3.x product with DIGITAL UNIX 4.0 and 4.0A systems.

When installing the patch, the `$ORACLE_HOME/bin` directory and the Oracle program cannot be shared by earlier versions of DIGITAL UNIX, as with NFS sharing, for example. The patch was applied by setting our current directory to the directory where the patch was read and running the **`patch.sh`** script.

Large File Patch to OEBU: Patch # 407801

We installed the Large File Patch to OEBU: Patch # 407801 (file brccfg.o, archived in libbrc.a, & file brctj.o archived in librct.a.) to allow raw data files greater than 4 GB in size. This feature is included in subsequent releases of OEBU and is a patch for Oracle bug # 407801.

Shared Memory Patch to OEBU: Patch # 407427

We installed the Shared Memory Patch to OEBU: Patch # 407427 (file sbrs.o, archived in libsbr.a.) to permit OEBU to allocate multiple shared memory segments. This feature is included in subsequent releases of OEBU, and is a patch for Oracle bug # 407427.

Installing SAS Institute Layered Products

Installing and Setting up the SAS System, Release 6.11

We installed the SAS System, Release 6.11, using the SAS System Installation Manager, from the SAS System distribution media, according to *Installation Instructions for the SAS System under UNIX Environments, Release 6.11, TS040,* (Cary, NC:SAS Institute Inc., 1996).

Making SAS Available to Users by Linking

We used the link method to make the SAS System available to our users, as described on Page 25 of the *Installation Instructions for the SAS System*.
We defined a logical for SAS as follows:

```
ln -s /var/sas611/sas /usr/bin/sas
```

NOTE

When issuing the link command, you must be root user.

Editing the SAS Config file

We edited the SAS configuration file, **`config.sas611,`** to provide sufficient memory for our large SAS dataset size (225 GB), as below.

Table 3-3: SAS configuration file Settings

Setting	Comment
-memsize 4096m	`-memsize` limits the amount of memory allocated by the SAS System. Default is 32 MB.
-sortsize 4048m	`-sortsize` limits the amount of memory allocated during sorting operations. Default is 16 MB.
-work /sas2	`-work` specifies where the SAS work library is created. This is a temporary work library: SAS data sets created there are deleted when the system terminates. Default is `/usr/tmp`.

SAS System Patches

We subsequently applied the two patches detailed below.

SAS Note V6-SYS.SYS-C443: Running SAS 6.11 on DIGITAL UNIX 4.0

We installed the SAS Patch associated with SAS Note V6-SYS.SYS-C443, to ensure error-free operation of the SAS system with DIGITAL UNIX 4.0 and higher. The module provides a new SAS executable compatible with DIGITAL UNIX 4.0 and higher systems. SAS Institute advises that without this new binary, SAS users may experience incorrect numeric results or other odd errors.

We obtained the patch from SAS technical support, and installed it according to documentation provided by them. Installing this patch makes the system Release 6.11, TS041.

SAS Note V6-SYS.SYS-C398: Oracle7 7.3.2 compatibility

We installed the SAS Patch associated with SAS Note V6-SYS.SYS-C398, to ensure compatibility of SAS System, Release 6.11 with Oracle7 Server Version 7.3.2. Changes in Oracle library names in Oracle7 Server, Version 7.3.2 cause errors to be reported when building SAS/ACCESS for Oracle Version 7.3.2. The corrected script **`!sasroot/install/zdbiora.ins`** resolves the problem by updating SAS/ACCESS for Oracle with the corrected library names.

We obtained the patch from SAS technical support, and installed it according to documentation provided by them.

Gateway Software to Remote DB2 Database

Overview

Table 3-4 lists the requirements for use of the *Oracle7 Transparent Gateway for IBM DRDA on DIGITAL UNIX*.

Table 3-4: Transparent Gateway for IBM DRDA Software Requirements

Area	Requirement
IBM mainframe resources	DB2 database
	VTAM communications
DIGITAL/SNA Gateway	Peer Server
DEC SNA APPC/LU62 Software	LU6.2 Server and LU6.2 Client
Oracle7 Components	Oracle7 Server
	Oracle7 Transparent Gateway for IBM DRDA
	SQL*Net
	Oracle Protocol Adapter for TCP/IP
	Oracle Listener

The Oracle7 database gateway product, *Oracle7 Transparent Gateway for IBM DRDA for DIGITAL UNIX*, can be installed either on the AlphaServer 8000 System, or on a remote AlphaServer. In order to test the Oracle7 software components as both an integrated suite on a single server, and as an application server component querying a remote database component, we tested two configurations. The two configurations we tested are detailed in this section.

All AlphaServer components may be installed on the same system. When components are spread across multiple systems, the *Transparent Gateway for IBM DRDA* and the *DEC APPC/LU6.2 Programming Interface* client must be located on the same system.

The steps we followed to install and configure each of these components are described below.

Remote Database Configurations

We prepared and tested two configurations of links from our Oracle7 database on the AlphaServer 8000 System to a remote DB2 database on an IBM mainframe.

In our first configuration, we located the Oracle7 Transparent Gateway and the LU6.2 server and client components on an AlphaServer 2100 at our remote site, and linked to the Oracle7 database via TCP over a T1 link. In our second configuration, we located these components on the AlphaServer 8000 System, on the same node as our Oracle7 database. The logical view of the software stack for each configuration is shown in the following two tables.

Table 3-5: Remote Database Configuration One

HiTest AlphaServer (depot1)				
SQL Query Application Program				
Oracle7 Database Server		**Remote AlphaServer 2100 (esacn1)**		
SQL*Net				
	TCP/IP over T1	Oracle Listener		
		Transparent Gateway for DRDA		
		LU6.2 Programming Interface Client		
		LU6.2 Server		
		SNA Peer Server	**IBM Token-ring**	**IBM 9121 mainframe**
				ACF/NCP
				VTAM
				DB2

Table 3-6: Remote Database Configuration Two

HiTest AlphaServer (depot1)				
SQL Query Application Program				
Oracle7 database Server				
Transparent Gateway for DRDA				
LU6.2 Programming Interface Client				
LU6.2 Server		**Remote AlphaServer 2100 (esacn1)**		
Oracle Listener				
	TCP/IP over T1	SNA Peer Server	IBM Token-ring	**IBM 9121 mainframe**
				ACF/NCP
				VTAM
				DB2

In configuration two, TCP is used between the two AlphaServer systems. If the SNA Peer Server and other Alpha Server based components are located on a single system, either TCP or IPC may be used.

Mainframe Software Installation and Configuration

After installing DB2's Distributed Data Facility, we used the IBM mainframe utility *DSNJU003* to update the DB2 bootstrap dataset to include the VTAM APPLICATION ID (a.k.a. ACB name) ‘DB24MVS’. We then created and activated a VTAM Application Major Node for DB24MVS.

We configured the Transparent Gateway according to the *Oracle7 Transparent Gateway for IBM DRDA for DIGITAL UNIX Installation and Users Guide,* Chapter 5, Configuring the DRDA Server. The steps we followed were:

Step	Action	Comment
1.	Configure VTAM for LU6.2 communications with workstation.	As our IBM mainframe was already configured for token ring operation, we simply created and activated a VTAM switched major node to describe the newly installed Peer Server.
2.	Define IBM mainframe UserID and Password for use by Oracle.	We chose a userid and password of ORACLE1/ORACLE1.
3.	Define IBM mainframe UserID and Password for recovery.	We chose a userid and password of ORACLE1/ORACLE1.
4.	Determine the location name of DB2	Using the DB2 SQL statement "select current server from any_table"
5.	Configure DB2's communications database.	We believe that this step could have been skipped because all of our testing was from Oracle7 to DB2. The information provided in this step is only used for DB2 to Oracle7 access.

VTAM APPLICATION MAJOR NODE

The VTAM Application Major Node describes DB2 to VTAM. The Application Major Node we used is shown below:

```
***********************************************
* 2/15/96  LU FOR DDF FOR MVSESA
***********************************************
VBUILD TYPE=APPL
DB24MVS  APPL  APPC=YES,
         ATNLOSS=ALL,
         AUTH=(ACQ),
         AUTOSES=1,
         DMINWNL=25,
         DMINWNR=25,
         DSESLIM=50,
         MODETAB=MODESNA,
         PARSESS=YES,
         PRTCT=ORACLE,
         SECACPT=ALREADYV,
         SRBEXIT=YES,
         SYNCLVL=SYNCPT,
         VPACING=8
```

VTAM SWITCHED MAJOR NODE

The VTAM Switched Major Node describes the SNA Peer Server and the DIGITAL UNIX node that the Peer Server runs on to VTAM. The Switched Major Node we used is shown below.

```
***********************************************
* SWESACN1 - DEFINITIONS FOR PU ON 2100 ESACN1
* 1 GOES THROUGH 3172 TRN - (PUESACNY)
***********************************************
SWESACN1 VBUILD TYPE=SWNET,MAXGRP=5,MAXNO=50
PUESACNY  PU MAXPATH=5,MAXDATA=17300,ADDR=13,PUTYPE=2,
                DISCNT=NO,
                IDBLK=DEC,
                IDNUM=80150,
                MODETAB=MODESNA,DLOGMOD=S3278M2,
                ISTATUS=ACTIVE,MAXOUT=127,
                SSCPFM=USSSCS,USSTAB=USSR2,
                       VPACING=63,
                DYNLU=YES,CONNTYPE=LEN
PESACN1Y PATH GRPNM=G1B43,DIALNO=00040000C91AB466,USE=YES,PID=2
   *
   IESACN01 LU    LOCADDR=0,ISTATUS=ACTIVE
```

Peer Server Installation

We installed the DIGITAL SNA Peer Server according to the instructions in *DEC SNA Peer Server Installation and Configuration Manual* (Part No. AA-Q1P8C-TE).

We configured the DIGITAL SNA Peer Server to:

1. Communicate with one IBM mainframe over a token ring.
2. Support SNA sessions between one local independent LU (ILU) and one IBM mainframe application program (DB2).

We used all configuration default values, except as shown in the attached configuration files.

As part of the installation process, a Peer Server startup procedure is added to /sbin/rc3.d directory. As a result, the Peer Server starts automatically when the node is booted.

For a detailed configuration file for a basic Peer Server with One Token Ring Adapter, and for Peer Server parameters we added to support Oracle7 Transparent Gateway for DRDA and IBM DB2, see Page 4–32 for *Remote DB2 Database Files and Scripts.*

DIGITAL LU6.2 Installation and Configuration

We installed DIGITAL SNA APPC/LU6.2 on nodes ESACN1 and DEPOT1. The configuration files and discussion refer to the installation on DEPOT1.

DIGITAL LU6.2 was installed according to the instructions in *DEC SNA APPC/LU6.2 Programming Interface for DIGITAL UNIX* (Part No. AA-QQW3A-TE)

DIGITAL SNA APPC/LU6.2 Server Configuration

Modify `/etc/services` to contain two additional lines:

```
LU62_SRV_SES    9001/tcp   # port for DEC/SNA GATEWAYs
LU62_SRV_CLI    9002/tcp   # port for Client Applications
```

The first part of the service names, ‘LU62_SRV’, corresponds to the default name that is declared when the LU6.2 Server process is started. The value shown in the default value can be an arbitrary number chosen by the system administrator. If the LU6.2 client is run on a separate system from the LU6.2 server, the two lines added to the /etc/services file must be the same.

The last part of the service names, "_SES and _CLI", refer to the services provided or used by the LU6.2 server. The "_SES" value declares the port used to listen for requests from the Peer Server. This value must match the value specified in the "SNA access server object IESACN01 port = value" statement of the Peer Server configuration file.

Develop a Startup Script for the LU 6.2 Server

The LU62_SERVER does not have a built in, or default location for its startup and configuration information. For our tests, we started the LU6.2 server manually, using the following UNIX shell script:

```
#!/usr/bin/ksh
/sbin/init.d/lu62_server start      # Start LU62 Server Process
sleep 5                             # Give it time to start up
./startitup <db22                   # Configure LU62 Server
```

The script 'startitup' is a slightly modified version of a script supplied as part of the LU6.2 product. The file /usr/lib/SNA/lu62_script was modified by changing the line CT=8 to CT=10. This changes the default upper limit on the number of concurrent sessions supported.

For the specific file db22 invoked in the shell script, see the end of this chapter for *Remote DB2 Database Files and Scripts.*

Establish SNA Session Between LU6.2 Server and IBM System

After running the script shown in the previous section, at least one SNA session should be established between the LU6.2 server and the IBM mainframe. The tool /usr/lib/SNA/lu62_config can be used to display and modify the current LU6.2 server configuration.

DIGITAL SNA APPC/LU6.2 Client Configuration

The Oracle DRDA component uses the CPIC interface to the DIGITAL SNA APPC/LU62 software. The CPIC interface gets the majority of its environmental values from a "side" or "configuration" file. In our tests, we utilized the default "side" file located at /etc/cpic.conf. This file is created during the DIGITAL SNA APPC/LU62 product installation.

For a copy of the edited file see the end of this chapter for *Remote DB2 Database Files and Scripts.*

NOTE

When upgrading our LU6.2 software components from the earlier version, we made a duplicate of the file `/etc/cpic.conf`. During the upgrade process, this file is deleted and re-created. After the new version of the software was installed, we replaced the newly created file with our original; if we had not, the Oracle LU6.2 client would not have been able to re-establish connections to the LU6.2 server, the SNA Peer Server, and the IBM mainframe.

Oracle7 Transparent Gateway Installation and Configuration

We installed the Oracle7 Transparent Gateway for DRDA according to *Oracle7 Transparent Gateway for IBM DRDA for DIGITAL UNIX Installation and User's Guide* except as noted below.

Because our version of the installed Oracle7 server is a later version than 7.0.16, we installed the DRDA product in a different directory from the Oracle7 server's ORACLE_HOME. Our DRDA home directory was /var/ORA_TG4DRDA_HOME.

Based on our experience installing the Transparent Gateway, we make the following recommendations:

1. During installation, be sure to choose "RE-LINK ALL APPLICATIONS" as covered in the manual. This is not the default value.
2. If you plan to use SQL*NET over TCP between the Oracle7 server and the DRDA components, install all SQL*NET, and TCP components in a single pass with DRDA. DRDA may not function properly otherwise
3. At completion of the Oracle installation procedure, skip running 'root.sh', as directed by the installation guide. This avoids potential conflicts with defining environmental variables that were defined by the Oracle7 Server version of root.sh.
4. Where file trees are mentioned in the manual, we found it important to take care to determine whether the file indicated was the one in the TG4DRDA file tree, or in the Oracle7 server file tree.

Configuration Files for Oracle7 Transparent Gateway for IBM DRDA (TG4DRDA)

In directory `/etc, f`ile `oratab`: Add one line for each instance of *Transparent Gateway for IBM DRDA* running on this node:

```
DB22:/var/ORA_DTDA_HOME:n:g4drdrv
DB2I:/var/ORA_DTDA_HOME:n:g4drdrv
```

In directory `/var/ORA_DRDA_HOME/tg4drda/admin`, the following files were changed:

- `initDB22.gtwboot` – We copied this file from the example file (`initDB2.gtwboot`) supplied by Oracle. Changes we made were:

```
LOG_DESTINATION=  (a different file name for log file)
CPIC_SYM_FILE=   (changed to the DIGITAL UNIX CPIC default /etc/cpic.conf)
```

NOTE

This file is used to define environmental variables before starting a process that uses the DIGITAL APPC/LU6.2 Client software. As such, it is the place to add the environmental variables to produce SNA level traces. To do this, add definitions for environmental variables SNALOG_MASK and SNALOG_FILE. See the manual, *DEC SNA APPC/LU6.2 Programming Interfaces for DIGITAL UNIX*, for details on setting environmental variables.

- `initDB2I.`gtwboot – Same considerations apply as for file initDB22.gtwboot.
- initDB22.ora – We copied this file from the example file supplied by Oracle, INITDB2.ORA.

Changes we made to the Transparent Gateway configuration file are detailed in Table 3-7.

Table 3-7: Transparent Gateway Configuration File Settings

Parameter	From:	To:
GATEWAY_SID	DB2	DB22
DB_NAME	DRD1	DB22
DRDA_CONNECT_PARM	DB2V23LU	DB22
DRDA_REMOTE_DB_NAME	DB2V2R3	LKG
DRDA_RECOVERY_USERID	ORADRDA	ORACLE1
DRDA_RECOVERY_PASSWORD	ORADRDA	ORACLE1

GATEWAY_SID should be embedded in the file name as well. This is also the value specified in the Oracle7 database link definition.

DRDA_CONNECT_PARM is the entry name used in the CPIC side file (/etc/cpic.conf).

DRDA_REMOTE_DB_NAME must match the DRDA location name specified in the DB2 data base. See the *Oracle7 Transparent Gateway for IBM DRDA for DIGITAL UNIX Installation and User's Guide*, chapter 7, section "MVS DB2 DRDA Server" for a procedure to determine a DB2 database's location.

DTDA_RECOVERY_USERID and **_PASSWORD** are IBM mainframe userid and passwords that are authorized to access DB2 data.

File **initDB2I.ora**: Same considerations apply as for **initDB22.ora**.

In directory **/var/ORA_DRDA_HOME/network/admin**

File **listener.ora**: We copied this file from a sample file provided by Oracle. It directs the Oracle Listener process to accept requests via IPC and/or TCP for database ids of DB22 and DB2I:

```
LISTENER =
(ADDRESS_LIST =
      (ADDRESS =
            (PROTOCOL=IPC)
            (KEY= net.world)
      )
      (ADDRESS =
            (PROTOCOL=IPC)
            (KEY= net)
      )
      (ADDRESS =
            (PROTOCOL = TCP)
            (Host = depot1)
            (Port = 2510)
      )
 )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = 16
trace_directory_listener=/var/tmp
SID_LIST_LISTENER =
      ( SID_LIST =
            (SID_DESC =
                  (SID_NAME = DB22)
                  (ORACLE_HOME = /var/ORACLE_TG4DRDA_HOME)
                  (PROGRAM=g4drdrv)
            )
            (SID_DESC =
                  (SID_NAME = DB2I)
                  (ORACLE_HOME = /var/ORACLE_TG4DRDA_HOME)
                  (PROGRAM=g4drdrv)
            )
      )
```

After editing all configuration files, run Oracle utility *g4rutl* as described in the *Oracle Transparent Gateway for IBM DRDA for DIGITAL UNIX Installation and User's Guide*, Chapter 7.

Oracle7 Server Configuration changes

We altered the $ORACLE_HOME/network/admin/tnsnames.ora file (in Oracle7 server) by adding one entry for each database link:

For the *Transparent Gateway For IBM DRDA* instance called DB22 reached via TCP:

```
DB22.world=(
   DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=depot1)(PORT=2510))
   (CONNECT_DATA=(SID=DB22)(GLOBAL_NAME=DB22.world))
   )
```

For the *Transparent Gateway for IBM DRDA* instance called DB2I reached via IPC:

```
DB2I.world=(DESCRIPTION=(ADDRESS=
(PROTOCOL=IPC)(HOST=depot1)(KEY=net.world))
(CONNECT_DATA=(SID=DB2I)(GLOBAL_NAME=DB2I.world)))
```

NOTE

The above example is shown on multiple lines for readability. In the file on disk, all of the information for each entry is run together on one line.

We created database links in the Oracle7 database using these `svrmgr` commands:

```
create public database link db22
connect to ORACLE1 identified by ORACLE1 using 'DB22';

create public database link db2i
connect to ORACLE1 identified by ORACLE1 using 'DB2i';
commit;
```

Overall Startup sequence

The overall startup sequence we followed was:

1. Verify that DB2, Oracle7, and the Peer server are operational.
2. Start (or restart) LU6.2 server.
3. Start Oracle7 Listener (Be sure to set ORACLE_HOME to TG4DRDA directory -- otherwise, the correct Oracle Listener configuration file will not be used).

At this point, it should be possible to access IBM DB2 tables from Oracle applications. For instance, an SQL statement like:

```
select * from cpg.sales_fact@db22;
```

should retrieve all rows of data from table `cpf.sales_fact` on the IBM mainframe DB2 system.

Setting VLM64 Parameters

UNIX VLM kernel parameters (/etc/sysconfigtab)

The parameter settings we use for the UNIX VLM kernel are based on a configuration with eight GB of memory. The settings we used are based on parameters for a VLM database on DIGITAL UNIX, suggested in Oracle7 Server for DIGITAL UNIX 7.3.2.3, “Chapter 3: The 64 Bit Feature” (Part # A49350-1).

We minimized the Unified Buffer Cache (parameters ubc-minpercent and ubc-maxpercent) to allow a larger Oracle buffer cache.

Depending on specific customer requirements for balancing a system for using both Oracle and an application such as the SAS System, ubc-minpercent, ubc-maxpercent, and gh-chunks may be set differently, to make more physical memory available to the application.

Table 3-8: UNIX VLM Kernel Parameters

Parameter	Value
vm parameters:	
vm-maxvas	8589934592
vm-mapentries	400
ubc-minpercent	1
ubc-maxpercent	2
gh-chunks	520 during index creation; modified to 1540 after index creation, which is appropriate for 6 GB SGA
new-wire-method	1
rt parameters:	
aio-max-num	512
aio-max-percent	2
proc:	
max-proc-per-user	1024
max-threads-per-user	1024
max-per-proc-data-size	8589934592
max-per-proc-address-space	8589934592
ipc parameters:	
shm-max	2139095040 *** 8 MB aligned address
shm-seg	32
msg-max	8192
msg-mnb	16384
msg-mni	1024
msg-tql	4096
sem-aem	16384
sem-mni	200
num-of-sems	200
sem-msl	200
sem-opm	200
sem-ume	200
sem-vmx	32767
ssm-threshold	0

Setting Oracle 64-bit Option Parameters

For the *Oracle Data Warehousing with SAS HiTest AppSet on the DIGITAL UNIX TruCluster ASE 8000 5/440 HiTest Foundation*, we optimized our Oracle7 server and database for decision support with the following parameter settings.

Table 3-9: Oracle 64-bit Option Parameters

Parameter	Value	Comment
db_block_size	32768	DSS and data warehouse applications benefit from a Big Oracle Block (BOB).
_init_sql_file	(?/dbs/sql.bsq.32k)	Contains storage parameters specifically for 32K block size. Although we set this parameter, it is no longer required as of Oracle7 Version 7.3.
vlm_sga_base_address	0x400000000	Use for SGA in excess of 3 GB.
db_block_buffers	192617	Approximately 6 GB (6019M) of buffer cache based on the db_block_size of 32768. This number should be maximized to provide the highest cache hit ratio without adversely affecting the memory requirements of other Oracle and system processes.
async_write	1	Enables the DIGITAL UNIX asynchronous I/O feature which allows the DB writer to perform multiple writes to multiple disks simultaneously without waiting for previous writes to finish.
sort_area_size	209715200	Memory area allocated to each process/thread performing sorts. This parameter should be maximized during intensive sort operations, such as index creation on very large tables. However, this is highly sensitive to the parallel degree of the table(s) being sorted, as each thread will allocate this amount or space.
sort_direct_writes	auto	Setting this parameter will allocate memory in addition to the sort area to perform sort writes directly to disk, bypassing the buffer cache. Setting this parameter to auto disables the sort_write_buffer_size parameter.
sort_write_buffer_size	131072	The correct default value for this parameter is documented in the 7.3.2 README file. When sort_direct_writes is set to true, this parameter must be set to a value between 32768 and 131072, that is a multiple of 32768.
cache_size_threshold	180000	This parameter controls the buffer cache space reserved for table scans. Tables with this or fewer Oracle blocks will be cached in the SGA during full table scans. This parameter is set sufficiently large enough to accommodate the smaller dimension tables in the Consumer Packaged Goods database.
ccf_io_size	524288	Determines the number of bytes per write when creating a contiguous file.
db_file_multiblock_ read_count	4	Due to an Oracle limit of 128 Kb per I/O, this parameter was set to 4, which is the maximum based on the formula max_io_size/db_block_size (128Kb/32Kb).
parallel_max_servers	72	Setting this parameter greater than 1 allows the Oracle RDBMS to take advantage of the parallel query option, for tables where the parallel degree has been set. This parameter should minimally be set to the largest degree specified on any table. Index creation on a table with a parallel degree set will also use this option; however 2 threads will be created for each degree, therefore this parameter should be set to twice the degree for full benefit on index creation.
shared_pool_size	18000000	This parameter affects the performance of the library cache (shared SQL and PL/SQL areas) and the dictionary cache. This parameter may be reduced if the cache hit ratio is not adversely affected. Otherwise, if cache misses are prevalent, then more memory should be allocated and the open_cursors parameter may need to be increased.

Parameter	Value	Comment
log_checkpoint_interval	99999999	Set to sufficiently large number to disable time based checkpoints forcing checkpoints only to occur on log switches.
gc_db_locks	191021	This is the total number of Parallel Cache Management (PCM) locks covering the datablocks in the SGA's of a Parallel Server.
gc_rollback_segments	41	The maximum number of rollback segments system wide. This parameter is set to the total number of rollback segments acquired by all instances in a parallel server.

Enabling Process Limits for Oracle DBA User Account

We edited the Oracle DBA User Account Process Settings to enable use of larger parameter process values that are required for the Oracle 64-bit Option, according to *Oracle7 Server for DIGITAL* UNIX *7.3.2.3*, "Chapter 3: The 64 Bit Feature" (Part # A49350-1). An Oracle DBA user account, using the UNIX C Shell, should have these settings in its `.login` file.

`.login` file commands:

```
limit datasize unlimited
limit stacksize unlimited
limit memoryuse unlimited
limit addresspace unlimited
```

Oracle Parallel Server Considerations

The implementation of a shared database utilizing the Oracle Parallel Server on a DIGITAL UNIX TruCluster requires some special considerations to enable the proper and efficient use. This section is designed to make the reader aware of these topics and provide an overview of how they were utilized.

Distributed Raw Device (DRD)

The shared storage on which the common database resides is provided by the Distributed Raw Device functionality of DIGITAL UNIX TruClusters. This is a client/server implementation and allows for the management of devices across the cluster. At any given point of time each DRD service is provided to the other cluster members by one of the members. The ASEMGR facility is provided to assist in the assignment of physical storage (device/partition level) to DRD entity and node location of a DRD services.

Oracle Initialization Parameters

Implementation of a Oracle Parallel Service requires some parameters be identical across all instances and this is desired behavior for some other parameters. Use of both a common and instance specific parameter file allows the flexibility of constant and variable parameters across instances. The common initialization parameter file is included in the instance specific parameter file, through use of the oracle IFILE parameter. In general the majority of our initialization parameters were consistent across the instances, but a few such as ROLLBACK_SEGMENTS, THREAD and INSTANCE_NUMBER were uniquely defined in the instance specific parameter file.

Redo Logs

Each instance must have a unique set of redo logs, but they must be accessible by the other nodes in the Parallel Server to maintain integrity during an instance or node failure. One of the remaining instances will detect the failure and apply the failed instances redo log to the database to provide consistency across the shared database. Hence the placement of the redo logs for all three instances on DRD's of the shared storage. Each instance is assigned a thread number and each redo log group is enabled for a specific thread. There is a slight performance benefit in using local DRD's when configuring for a given instance, so each node is the server for DRD’s used for the instance's redo logs. The commands for adding the redo threads for two additional instances is as follows:

```
alter database add logfile thread 2
    '/oracle/links/LOGA_2' size 1000M reuse,
    '/oracle/links/LOGB_2' size 1000M reuse,
    '/oracle/links/LOGC_2' size 1000M reuse;
alter database enable public thread 2;
alter database add logfile thread 3
    '/oracle/links/LOGA_3' size 1000M reuse,
    '/oracle/links/LOGB_3' size 1000M reuse,
    '/oracle/links/LOGC_3' size 1000M reuse;
alter database enable public thread 3;
```

Rollback segments

The creation of one tablespace per instance (node) is not a requirement, but was used here to allow each instance private rollbacks segments that would produce only local DRD access. This slightly more optimal I/O path is likely to produce benefits in an update intensive environment. One tablespace for each instance is created to hold rollback segments, using a local DRD. Then rollbacks segments are created in the tablespace and assigned to the instance using Oracle initialization parameter ROLLBACK_SEGMENTS, which should be in the instance specific oracle parameter file.

Control Files

The control files need to be accessed by all instances in the Oracle Parallel Server Environment. To meet this requirement, and to ensure availability of at least one control file in the event of physical disk failure, three control files were distributed across three separate DRD's, which are served by different cluster nodes and utilize three unique storage controllers.

Data Warehousing Database

The Consumer Packaged Goods Database

The Oracle Consumer Packaged Goods demo database was chosen as representative of a typical workload for a data warehouse with a star-schema design. We modified the build of the 419.4 GB Oracle7 database component from the standard Oracle CPG database, by doubling the number of sales channels represented and by building the SALES_FACT datafile on a daily rather than a weekly basis, to generate a larger database. The star schema design was not affected by our changes, and we did not change the queries from the CPG demo.

Our DIGITAL UNIX 419.4 GB Consumer Packaged Goods database contains data representative of six years of sales information for a variety of products across geographical areas of the US, including sales volume, channels, and other characteristics.

Database Creation and Build

Our database consists of distributed raw devices, most of which are 419.4 MB in size. Each volume is a 4 disk RAID-5 stripe set, with dual HSZ40 Array Controllers. This solution optimizes for both performance and availability.

The large SALES_FACT table consists of 12 RAID-5 datafiles. The index for the SALES_FACT table consists of 10 datafiles. 13 volumes of TEMP space was required to build the SALES_FACT_INDEX.

Performance considerations guided our placement of data files. Our data files are primary pathed across 6 HSZ40 Array Controllers, while the index files are primary pathed across a separate set of 6 HSZ40’s. TEMP file space is evenly dispersed across all 12 HSZ40 Array Controllers. This configuration maximizes I/O access paths, while minimizing I/O contention for data load and data retrieval.

Both the FACTS and FACTSINDEX tablespaces use the storage parameters listed below, with the exception of the initial extent of the FACTS tablespace. The FACTS tablespace contains the SALES_FACT table, which is populated using `sqlldr`; as the initial extent of a tablespace is not used by `sqlldr`, we set the initial extent to 64K to limit this unusable space.

Table 3-10: Database Tablespace Storage Parameters

Storage parameters	FACTS	FACTSINDEX	TEMP
datafile size	12,269 MB	12,269 MB	12,269 MB
initial_extent	64 K	1000 MB	200 MB
next_extent	1000 MB	1000 MB	200 MB
max_extent	2041	2041	2041
pctincrease	0	0	0
pctfree	0	0	NA

We divided three 12.0 GB RAID-5 stripe sets into 7 smaller equal partitions of 1,785,490 Kbytes each to store the SYSTEM (190 MB), DIMENSION (10 MB), and DIMINDEX (10 MB) tablespaces, and the redo logs (1000 MB / log).

The database build, including allocation of the small DIMENSION and DIMINDEX tablespaces, took 10 minutes. The creation of the large tablespaces (FACTS, FACTSINDEX, and TEMP) allocating the initial datafile of 12269 MB was done serially. In Oracle7, the `CREATE TABLESPACE` command is a serial function due to data dictionary lock requirements. Each tablespace is created serially with only one data file. Once created, the `ALTER TABLESPACE` command can be issued in parallel to allocate additional data files to each tablespace.

We performed some optimization in allocating the 12,269 MB datafiles.

The following parameter values are key factors in maximizing I/O throughput, and should be set accordingly:

Parameter	Setting
RAID chunk size	131,072 bytes (128 Kbytes or 256 disk blocks)
ccf_io_size	524,288 bytes (512 Kbytes or 1024 disk blocks)

See Table 3-15, *File Systems and Database Storage Map*.

We set the RAID chunk size at 131,072 , which accommodates four 32 KB database blocks. For a detailed description of how we calculated chunk size, see Page 3–3. For ccf_io_size, the Oracle parameter specifying the number of bytes per write when creating a contiguous file, we found the optimal setting to be a value equivalent to the number of disk drive spindles times chunksize. Using an equivalent size allows for the largest possible asynchronous write, without incurring additional overhead of the I/O spanning drives on the stripe-set.

At these optimal settings we allocated a 12,269 MB datafile in approximately 55 minutes, at the rate of 13.1 GB/Hour. Additional testing with two allocations on the same HSZ provided a 52.7% increase in I/O throughput, to 20 GB/Hour. In building the three large tablespaces (FACTS, FACTSINDEX, and TEMP) multiple threads on all three cluster nodes were used, with all I/O localized to the owner of the DRD device, to create a total of 31 datafiles.

During the initial phases of extending these three tablespaces, three separate scripts, each executing on distinct cluster nodes, allowed the simultaneous execution of 23 threads, to build the first 23 of 31 datafiles. Eight of the 23 threads performed a second step, to build the remaining eight datafiles. The total time to execute the `ALTER TABLESPACE` commands, allocating 372 GB, was approximately 2 hours and 30 minutes. The overall average I/O rate for this creation was 148.8 GB/Hour, with a peak rate in excess of 166 GB/Hour during the initial phase of the `alter`, with 23 threads active.

See Table 3-15, *File Systems and Database Storage Map*.

Database Load, SQL Loader

We found `sqlldr` to be the most efficient mechanism for loading the large SALES_FACT table, which consisted of 2,766,836,356 rows. The SALES_FACT table represents 2 years of sales data derived from the dimension tables MARKET, PRODUCT, CHANNEL, and DAILY_PERIOD. Although this table could have been loaded using `INSERT` with `SELECT` commands from the dimension tables, due to the sheer volume of data, this was not a viable option; on the basis of small scale testing, we estimated it would have taken several weeks to load our data using the `INSERT` with `SELECT` method. Instead, we developed a set of scripts and a C program to extract dimension data, and generate the SALES_FACT row data in `sqlldr` format.

We partitioned data generation by month, where one month's data is generated and formatted in a flat file, which in turn is used as input to `sqlldr`. We required approximately 165 GB of storage to generate 2 years (24 months) of data; each month consists of a flat file averaging 5.2 GB in size. We created three RAID-5 UFS volumes for this purpose.

Planning sufficient storage for a dedicated load volume is necessary in data warehouse environments, where data from production systems is regularly loaded in bulk. An inadequate storage strategy may result in bottle-necks occurring, slowing the data load rate, and limiting the use of multiple threads. Our storage strategy required the flexibility to allow loading from each of 3 separate cluster members, and to permit a high read-throughput rate. We accomplished this flexibility by implementing three separate 56 GB file systems on separate HSZ40 Array Controllers. These three datafiles could have been mounted on any single node.

We tested several approaches to loading our data by varying the number of concurrent streams. In all cases, each stream targeted 1 datafile, loading 1 month’s data of approximately 118 million rows from a unique 5.2 GB flat file. We loaded each datafile with a total of 2 months worth of data, in two steps, with two streams each loading half the datafile, giving us 12 datafiles of 12,269 MB each. On average these datafiles are 7% free, for a total of 133 GB of data with 2,766,836,356 rows.

Load testing was conducted prior to upgrading all cluster nodes to 440Mhz CPU’s, on a mixed environment including a single 300Mhz system and two 350Mhz machines. This provided additional opportunities for observing and capturing `sqlldr` characteristics.

Our results indicate a slight benefit when performing load operations to files that are local Distributed Raw Devices (DRD’s). Our configuration of 8 CPU’s per node results in CPU utilization of 12% for each SQL loader thread; this CPU resource configuration constrains the number of load streams on a single node. We found that we obtained optimal load performance using a single load stream per CPU on the system. The table below shows the slight (2.7%) performance gain we observed when loading to local DRD’s.

Table 3-11: Remote and Local Load from the Same Node

Local DRD (depot1)	Rows	Minutes	Rows/Minute
FACTS1	118882768	02:29:35.60 (150 min.)	792551
FACTS2	107377984	02:17:03.48 (137 min.)	783780
Remote DRD (depot2)			
FACTS3	118882768	02:35:21.69 (155 min.)	766985
FACTS4	115047840	02:30:32.83 (150 min.)	766985

Optimal loading can occur when simultaneously loading all twelve fact files, but would require more CPU power than is available from a single cluster node. The multiple UFS file systems are used to prevent an input bottleneck to the load streams and allow loading from multiple cluster nodes. The three file systems holding the load data were mounted on separate nodes of the cluster and used to allow local DRD loading for all data.

The results of the first load confirm that load time is tightly coupled to the speed of the processor performing the load operation. The first half of the fact table load was evenly distributed across the three system with all DRD access local to the node. As expected the longest load times were experienced by the slowest CPU (depot1). A comparison of the load times is provided below:

Rows	Elapsed Time	Rate	% Increase
depot1 (300MHZ)			
118882768	149 minutes	797870 rows/min	
115047840	N/A		
107377984	131 minutes	819679 rows/min	
depot2 (350 MHz)			
118882768	130 minutes	914482 rows/min	14.6
115047840	129 minutes	891843 rows/min	
107377984	N/A		
depot3 (350 MHz)			
118882768	129 minutes	921571 rows/min	15.5
115047840	124 minutes	927805 rows/min	
107377984	N/A		

The increase in processing of rows per second is almost linear with the increase of processor speed (16.6%) between nodes 300 MHz and 350 MHz systems. As a result the second half of the load was changed to perform the load from only depot2 and depot3 (350MHz machines) and was run with 8 processes on depot2 and 4 processors on depot3. This means that half of the database files being loaded from Depot2 would be remote DRD services from Depot1. This produced the anticipated results of a much narrow band of load times.

Load Part II

Rows	Elapsed Time	Rate	% Increase
depot1 (300 MHz) local			
118882768	130 minutes	914482 rows/min	14.6
115047840	129 minutes	891843 rows/min	
107377984	N/A		
depot2 (remote DRD)			
118882768	130 minutes	914482 rows/min	14.6
115047840	129 minutes	891843 rows/min	
107377984	N/A		
depot3 (350 MHz)			
118882768	129 minutes	921571 rows/min	15.5
115047840	124 minutes	927805 rows/min	
107377984	N/A		

Calculating Required Index Space

We found that it's critical to properly calculate initial index and temporary space required, to allow creation of a concatenated index SF_KEY on the large SALES_FACT table. To decrease unused data block space, the Oracle7 parameter PCTFREE is set to 0 when building the CPG database. Using a small PCTFREE setting is desirable in data warehouses due to the read-only and bulk-update nature of the environment.

For a reasonable estimate of the space requirements use the following procedure:

1. Obtain, or if necessary estimate, the total number of rows in the SALES_FACT table:

   ```
   sales_fact = 2766836356 rows
   ```

2. Calculate the index block header size, using the formula:

block header	**`= fixed header + variable header`**
	`= 113 bytes + initrans*24bytes`
	`= 113 bytes + 2*24bytes=48bytes`
block header	`= 161 bytes`

Note

We used the INITRANS default value for indexes of 2 for all our indexes.

3. Calculate available data space per data block.

available data space	**`= DB block size - block header`**
	`=    32768 bytes  - 161 bytes`
	`=    32607 bytes`

4. Use the sum of the length of each column in the index, which is the column bytes per index entry. The four columns used in this index are product_id, chan_id, market_id, day, which are declared as 3 char(6) and a date field.

entry column/s size	= **sum (length of 4 columns)**
	= 6+6+6+7
	= 25 bytes

5. Calculate the bytes per entry

bytes/entry	= **entry column/s size + entry header + rowid + F + V**
entry header	= 2 bytes
rowid	= 6 bytes
F	= **fixed length bytes per entry (# of columns <127bytes)**
	= 5 (4 specified columns plus 1 for rowid)
V	= **variable length byte and is 0 for all calculations**
average entry	= 25 + 2 + 6 + 5
	= 38 bytes

6. Calculate total index blocks required

index blocks	= 1.05*((#rows*avg entry) -- ((data bytes per block/avg entry)*avg entry))
index blocks	= 1.05 * ((2766836356 * 38)/((32607/38) * 38))
	= 3385677 db blocks
index bytes	= index blocks * db block size
	= 3385677 * 32768
	= 1.11E+11 bytes
index MB’s	= index bytes / (1024 * 1024)
	= 1.11E +11 bytes/(1024 * 1024)
	= 105802.4 MB’s

The multiplier 1.05 is used to account for the index branch nodes and the use of parallel index create will most likely result in files that are not 100% utilized. The utilization of any extent by a parallel query server is a function of the number of rows apportioned to it by the query manager. The parallelism and extent sizes used to enhance performance will adversely affect the efficiency of the storage utilization for the index. To account for this behavior a 10% safety margin is built into the final calculation of index space.

Final index bytes	= index MB * 1.1
	= 105802.4 MB *1.1
	= 116382.6 MB

The extent size will determine the maximum integer number of extents in a single datafile, as space is always allocated in full extents. This provides the following given are settings:

Max. number of extents per datafile	= datafile size/extent size
	= 12269Mb/1048576000 = 12

Thus the total number of files required to hold the SF_KEY index is found by the following:

index files	= final index bytes -- (extent size * number of extents per file)
	= 353108Mb/(1048576000*12)
	= 9.69 (always rounded up)
	= 10

Temporary space requirements should be estimated at a minimum of 10% above the required index space. This results in 12 files being required for creation of this index. Due to the time required to build the database, and the amount of unusable space, a failure to allocate enough index space can be costly, therefore we added one additional file to increase our margin for error, resulting in temporary space of 13 files, or 167.7 GB.

For a complete discussion of calculating space for indexes, see *Oracle7 Server Administrator's Guide, Release 7.2* (April 1995, Oracle Corp., Part No. A20322-2), Managing Schema Objects.

Creating the Index

Parallel index creation, provided by the Oracle Parallel Query option, is an important database management function. We tested the Oracle Parallel Query option as part of the *Oracle Data Warehousing with SAS HiTest Suite*. We used parallel index creation during creation of the index on the large SALES_FACT table.

We created the concatenated SF_KEY index with a parallel degree of 16, which requires the parallel_max_servers parameter be set to a minimum of twice this value. This is necessary because the degree of parallelism on index creation is implemented by two cooperating query servers. The parallel degree for index creation was enabled by altering the default degree on the SALES_FACT table. The `svrmgr` commands to perform this are:

```
svrmgr> CONNECT cpg/cpg
svrmgr> alter TABLE SALES_FACT parallel (degree 16);
```

During *index create*, we significantly reduced the System Global Area (SGA) size as the SORT_DIRECT_WRITES parameter allows the SGA to be bypassed when performing index creation. By decreasing the SGA size, we provided additional memory for sorting by each parallel query server, reducing the elapsed time required to create the index.

We made these changes to the standard database parameters to facilitate index creation.

Table 3-12: Index Parameters

Parameter	Value
DB_BLOCK_BUFFERS	42000
SORT_AREA_SIZE	209715200
SORT_WRITE_BUFFER_SIZE	131072

During *index create*, we reduced the UNIX kernel parameter gh_chunks to 520 and rebooted the system. Doing this allows the physical memory normally set aside for shared memory (SGA) to be accessible to the general system memory pool and support the sort_area_size memory requirements for each Parallel query server.

We performed index creation for two years of SALES_FACT data, corresponding to 2.8 billion rows. The resulting elapsed time for index creation appears in the following table.

Index creation	elapsed time	rows	rows/sec
2 yrs	793 min.	2,766,836,356	58,151 rows/sec

The actual *create index* statement we used was:

```
SQL> create unique index CPG.SF_KEY on CPG.SALES_FACT(PRODUCT_ID,
CHANNEL_ID, MARKET_ID, DAY) tablespace FACTSINDEX unrecoverable;
```

Bitmap Indexes

Oracle7 Server, Version 7.3.2.3.0 provides powerful new functionality, in the form of bitmap indexes. The use of bitmap indexes has the potential to significantly reduce index storage space requirements, and improve performance on queries that involve low cardinality columns. Although we did not use bitmap indexes in our query testing, we tested the creation of this index type by implementing four indexes on the SALES_FACT columns, used in the concatenated SF_KEY index. To completely replace the b-tree index on the SALES_FACT table, four bitmap indexes must be created, as only one column is allowed in a bit map index. The creation times and resulting storage space for each index are below.

Table 3-13: Creation Times for Bitmap Indexes

index_name	Disk Storage
sf_key_pid	607,911,936 bytes (0.57 GB)
sf_key_cid	605,257,728 bytes (0.56 GB)
sf_key_mid	11,059,331,072 bytes (10.3 GB)
sf_key_day	628,129,792 bytes (0.58 GB)

Below are the default values for the parameters most likely to have a significant impact on the performance during the creation of bitmap indexes. We used these values in the creation of all of the bitmap indexes.

Table 3-14: Bitmap Parameter Settings

bitmap_merge_area_size	= 1048756
create_bitmap_area_size	= 8388608

The parameter values above are used by each of the query servers in a parallel index create and, as with binary tree indexes, parallel creation of bitmap indexes is implemented using cooperating query servers. A full discussion of the use of bitmap indexes can be found in the *Oracle7 Server Tuning, Release 7.3* (June 1996, Oracle Corp., Part No. A32537-1).

Table 3-15: File Systems And Database Storage Map: Three Node TruCluster

SCSI BUS #0, depot1		
HSZ40 controller:		
/dev/rz0a:	system disk	
/dev/rz0b	swap1	
/dev/rz0g:	/usr	
/dev/rz2c:	/var	
/dev/rz3c:	swap2	
/dev/rz4c:	swap3	
/dev/rzb0c:	/oracle_7223	
/dev/rzb2c:	/oracle_732	
/dev/rzb3c:	/oracle_733	
/dev/rzb4c:	/oracle_free	
SCSI BUS #1, depot2		
HSZ40 controller:		
/dev/rz8a	/	
/dev/rz8g	/usr	
/dev/rz9c	/var	
/dev/rz8b	swap1	
/dev/rz10c	swap2	
/dev/rz11c	swap3	
SCSI BUS #2, depot3		
HSZ40 controller:		
/dev/rz16a	/	
/dev/rz16g	/usr	
/dev/rz17c	/var	
/dev/rz16b	swap1	
/dev/rz18c	swap2	
/dev/rz19c	swap3	
SCSI BUS #4		
HSZ40 controller:		
/dev/rz32c:		SAS1
/dev/rzb32c:		SAS1
/dev/rzc32c:		SAS1
/dev/rz33c		SAS2
/dev/rzb33c		SAS2
/dev/rzc33c		SAS2
SCSI BUS #5, depot1		
HSZ40 controller:		
/dev/rz40c	/data 1	
HSZ40 controller:	drd_rz42 on mcdepot1	
/dev/rz42b:	/dev/rdrd/drd50	LOGA_1
/dev/rz42c:	/dev/rdrd/drd51	LOGB_1
/dev/rz42d:	/dev/rdrd/drd52	LOGC_1
/dev/rz42e:	/dev/rdrd/drd53	CTL_1
/dev/rz42f:	/dev/rdrd/drd54	RBS_1
/dev/rz42g:	/dev/rdrd/drd55	SYSTEM

SCSI BUS #6, depot2		
HSZ40 controller:		
/dev/rz48c:	/data2	
HSZ40 controller:	drd_rz50 on mcdepot2	
/dev/rz50a:	/dev/rdrd/drd57	DIMENSIONS
/dev/rz50b:	/dev/rdrd/drd58	LOGA_2
/dev/rz50c:	/dev/rdrd/drd59	LOGB_2
/dev/rz50d:	/dev/rdrd/drd60	LOGC_2
/dev/rz50e:	/dev/rdrd/drd61	CTL_2
/dev/rz50f:	/dev/rdrd/drd62	RBS_2
SCSI BUS #7, depot3		
HSZ40 controller:		
/dev/rz56c:	/data3	
HSZ40 controller:	drd_rz72 on mcdepot3	
/dev/rz58a:	/dev/rdrd/drd65	DIMENINDEX
/dev/rz58b:	/dev/rdrd/drd66	LOGA_3
/dev/rz58c:	/dev/rdrd/drd67	LOGB_3
/dev/rz58d:	/dev/rdrd/drd68	LOGC_3
/dev/rz58e:	/dev/rdrd/drd69	CTL_3
/dev/rz58f:	/dev/rdrd/drd70	RBS_3
SCSI BUS #8		
HSZ40 controller:		
/dev/rz64c:		SAS1
/dev/rzb64c:		SAS1
/dev/rzc64c:		SAS1
/dev/rz65c		SAS2
/dev/rzb65c		SAS2
/dev/rzc65c		SAS2
SCSI BUS #9	**drd_rz72 on mcdepot1**	
HSZ40 controller:		
/dev/rrz72c:	/dev/rdrd/drd1	FACTS1
/dev/rrz73c:	/dev/rdrd/drd3	FACTS2
/dev/rrzb72c:	/dev/rdrd/drd2	TEMP1
/dev/rrzb73c:	/dev/rdrd/drd4	
HSZ40 controller:		
/dev/rrz74c:	/dev/rdrd/drd5	FACTSINDEX1
/dev/rrz75c:	/dev/rdrd/drd7	FACTSINDEX2
/dev/rrzb74c:	/dev/rdrd/drd6	TEMP2
/dev/rrzb75c:	/dev/rdrd/drd8	
SCSI BUS #10	**drd_rz80 on mcdepot2**	
HSZ40 controller:		
/dev/rrz80c:	/dev/rdrd/drd9	FACTS3
/dev/rrz81c:	/dev/rdrd/drd11	FACTS4
/dev/rrzb80c:	/dev/rdrd/drd10	TEMP3
/dev/rrzb81c:	/dev/rdrd/drd12	
HSZ40 controller:		
/dev/rrz82c:	/dev/rdrd/drd13	FACTSINDEX3
/dev/rrz83c:	/dev/rdrd/drd15	FACTSINDEX4
/dev/rrzb82c:	/dev/rdrd/drd14	TEMP4
/dev/rrzb83c:	/dev/rdrd/drd16	

SCSI BUS #11		
HSZ40 controller:		
/dev/rz88c:		SAS1
/dev/rzb88c:		SAS1
/dev/rzc88c		SAS1
/dev/rz89c		SAS2
/dev/rzb89c		SAS2
/dev/rzc89c		SAS2
SCSI BUS #12	**drd_rz96 on mcdepot3**	
HSZ40 controller:		
/dev/rrz96c:	/dev/rdrd/drd17	FACTS5
/dev/rrz97c:	/dev/rdrd/drd19	FACTS6
/dev/rrzb96c:	/dev/rdrd/drd18	TEMP5
/dev/rrzb97c:	/dev/rdrd/drd20	
HSZ40 controller:		
/dev/rrz98c:	/dev/rdrd/drd21	FACTSINDEX5
/dev/rrz99c:	/dev/rdrd/drd23	FACTSINDEX6
/dev/rrzb98c:	/dev/rdrd/drd22	TEMP6
/dev/rrzb99c:	/dev/rdrd/drd24	
SCSI BUS #14	**drd_rz112 on mcdepot1**	
HSZ40 controller:		
/dev/rrz112c:	/dev/rdrd/drd25	FACTS7
/dev/rrz113c:	/dev/rdrd/drd27	FACTS8
/dev/rrzb112c:	/dev/rdrd/drd26	TEMP7
/dev/rrzb113c:	/dev/rdrd/drd28	
HSZ40 controller:		
/dev/rrz114c:	/dev/rdrd/drd29	FACTSINDEX7
/dev/rrz115c:	/dev/rdrd/drd31	FACTSINDEX8
/dev/rrzb114c:	/dev/rdrd/drd30	TEMP8
/dev/rrzb115c:	/dev/rdrd/drd32	
SCSI BUS #15	**drd_rz120 on mcdepot2**	
HSZ40 controller:		
/dev/rrz120c:	/dev/rdrd/drd33	FACTS9
/dev/rrz121c:	/dev/rdrd/drd35	FACTS10
/dev/rrzb120c:	/dev/rdrd/drd34	TEMP9
/dev/rrzb121c:	/dev/rdrd/drd36	
HSZ40 controller:		
/dev/rrz122c:	/dev/rdrd/drd37	FACTSINDEX9
/dev/rrz123c:	/dev/rdrd/drd39	
/dev/rrzb122c:	/dev/rdrd/drd38	TEMP10
/dev/rrzb123c:	/dev/rdrd/drd40	
SCSI BUS #16	**drd_rz128 on mcdepot3**	
HSZ40 controller:		
/dev/rrz128c:	/dev/rdrd/drd41	FACTS11
/dev/rrz129c:	/dev/rdrd/drd43	FACTS12
/dev/rrzb128c:	/dev/rdrd/drd42	TEMP11
/dev/rrzb129c:	/dev/rdrd/drd44	
HSZ40 controller:		
/dev/rrz130c:	/dev/rdrd/drd45	FACTSINDEX10
/dev/rrz131c:	/dev/rdrd/drd47	
/dev/rrzb130c:	/dev/rdrd/drd46	TEMP12
/dev/rrzb131c:	/dev/rdrd/drd48	TEMP13

4

Interoperability Tests and Results

The DIGITAL HiTest System for Oracle Data Warehousing and SAS on DIGITAL UNIX AS8000 TruCluster, was tested with a database built using the CPG Database Demo scripts provided by Oracle Corporation.

Our testing of the SAS System, release 6.11 and Oracle7 Server Release 7.3.2.3.0 on DIGITAL UNIX Version 4.0A continues the testing effort of the Enterprise Systems Assurance Center. Prior combinations of products tested included testing of Oracle7 Server Release 7.3.2.2 on DIGITAL UNIX Version 3.2G, and testing of Oracle7 Server Release 7.2.2.4 on DIGITAL UNIX Version 3.2C.

The Consumer Packaged Goods Database is fully described below.

This section describes the:

- Configuration of the tested system
- Tools and scripts used to perform the tests
- Test process
- Test Results

Configuration of the Tested System

Table 4-1: Tested CPU Hardware

AlphaServer TruCluster Node 1 (depot1)		
Qty	**Part Number**	**Description**
1		AlphaServer 5/440: Three phase, Dual-CPU, with DIGITAL UNIX
System includes:		
4	KN7CE-AB	Dual 21164/440 MHz CPU SMP module
4	MS7CC-FA	2147 MB memory module
1	KFTHA-AA	System I/O module with four I/O channels
20	KZPSA-BB	PCI-based Fast Wide Differential (FWD) SCSI-2 Adapter
1	KZPAA-AA	PCI-based one port high-performance Fast SCSI-2 controller for CD-ROM connection
2	CCMAA-BA	PCI to Memory Channel Controller
Storage		
4	BA655-AB	StorageWorks Plug-in unit
1	RRDCD-CA	600 MB 5.25” CD-ROM

AlphaServer TruCluster Node 1 (depot1)		
Qty	**Part Number**	**Description**
4	DWZZB-VW	Fast Wide Differential Single-ended SCSI Converter
I/O Expansion Bus		
2	DWLPA-AA	PCI plug-in unit with one PCI box for system cabinet
2	DWLPA-BA	PCI expansion box mounted in DWLPA-AA
Networks and Communications		
1	DE435-AA	DIGITAL Etherworks 32-bit High Performance Network Interface Card
1	DEFPA-AA	PCI FDDI Adapter

AlphaServer TruCluster Node 2 (depot2)		
Qty	**Part Number**	**Description**
1		AlphaServer 5/440: Three phase, Dual-CPU, with DIGITAL UNIX
System includes:		
4	KN7CE-AB	Dual 21164/440 MHz CPU SMP module
4	MS7CC-FA	2147 MB memory module
1	KFTHA-AA	System I/O module with four I/O channels
15	KZPSA-BB	PCI-based Fast Wide Differential (FWD) SCSI-2 Adapter
1	KZPAA-AA	PCI-based one port high-performance Fast SCSI-2 controller for CD-ROM connection
2	CCMAA-BA	PCI to Memory Channel Controller
Storage		
2	BA655-AB	StorageWorks Plug-in unit
1	RRDCD-CA	600 MB 5.25” CD-ROM
2	DWZZB-VW	Fast Wide Differential Single-ended SCSI Converter
1	RZ28D-VW	2.1 GB 16-bit 7200 RPM SCSI-2 disk drive
I/O Expansion Bus		
2	DWLPA-AA	PCI plug-in unit with one PCI box for system cabinet
2	DWLPA-BA	PCI expansion box mounted in DWLPA-AA
Networks and Communications		
1	DE435-AA	DIGITAL Etherworks 32-bit High Performance Network Interface Card

AlphaServer TruCluster Node 3 (depot3)		
Qty	**Part Number**	**Description**
1		AlphaServer 5/440: Three phase, Dual-CPU, with DIGITAL UNIX
System includes:		
4	KN7CE-AB	Dual 21164/440 MHz CPU SMP module
4	MS7CC-FA	2147 MB memory module
1	KFTHA-AA	System I/O module with four I/O channels
11	KZPSA-BB	PCI-based Fast Wide Differential (FWD) SCSI-2 Adapter
1	KZPAA-AA	PCI-based one port high-performance Fast SCSI-2

AlphaServer TruCluster Node 3 (depot3)		
Qty	**Part Number**	**Description**
		controller for CD-ROM connection
2	CCMAA-BA	PCI to Memory Channel Controller
Storage		
2	BA655-AB	StorageWorks Plug-in unit
1	RRDCD-CA	600 MB 5.25” CD-ROM
2	DWZZB-VW	Fast Wide Differential Single-ended SCSI Converter
1	RZ28D-VW	2.1 GB 16-bit 7200 RPM SCSI-2 disk drive
I/O Expansion Bus		
2	DWLPA-AA	PCI plug-in unit with one PCI box for system cabinet
2	DWLPA-BA	PCI expansion box mounted in DWLPA-AA
Networks and Communications		
1	DE435-AA	DIGITAL Etherworks 32-bit High Performance Network Interface Card

Shared Networks and Communications		
Qty	**Part Number**	**Description**
2	CCMHA-AA	Memory Channel Hub with 4 Line Cards
Storage		
4	DWZZA-VA	SCSI Bus Converter 16-bit differential to 8-bit single-ended
1	H7263-AB	Power supply
Cables		
3	BN21H-02	SCSI-2 Single-ended cable
4	BN21K-02	SCSI Cable
10	BN21K-05	Cables
20	BN21K-10	Cables
1	BN34D-10	FDDI Cable
Console		
1	AlphaServer 2100	Console
		Monitor

StorageWorks Cabinets: Four Cabinets

Qty	Part Number	Description
1	SW800-AA	StorageWorks Data Center Cabinet
18	BA350-JA	Storage Shelf
3	BA350-MA	Storage Controller Shelf
42	BA35X-HF	Power Supply
108	RZ29B-VA	4.3 GB 8-bit 7200 RPM SCSI-2 disk drive
6	HSZ40-BF	StorageWorks Array Controller, 42 SCSI-2 Disk/SSD Support; 32-MB Read Cache, RAID-0, Base Firmware and license, RAID license, Disk Mirroring license, Write-back Cache option

Tape Backup Hardware

Qty	Part Number	Description
2	TL810-BA	52 Cart DLT Automated Tape Library with 4 TZ87N Drives

Remote Hardware

	Qty	Part Number	Description
IBM		ES9121-480	Mainframe Computer
		3745-31A	Communication Controller
		3746-900	Communication Controller Expansion
		3172-003	Interconnect Communication Controller with Token Ring adapter
DIGITAL		AlphaServer 2100	Console

Table 4-2: Operating System and Network Software

Product	Version	Comments
DIGITAL		
AlphaServer 8400 TruCluster		
DIGITAL UNIX	V4.0A	(rev. 464)
2100 Console		
DIGITAL UNIX	V4.0B	(rev. 564)
Remote 2100 Console		
DIGITAL UNIX	V3.2G	(rev. 62)

Data Warehousing Test Environment

The *Oracle Data Warehousing with SAS* test environment includes the base AlphaServer 8400 TurboLaser TruCluster system with StorageWorks, connected via T1 link to a remote DB2 database on an IBM ES9000-9121 mainframe, through an Alpha 2100 front-end.For complete detailed drawings and specifications of the *DIGITAL UNIX TruCluster ASE 8000 5/440 HiTest Foundation* hardware configuration, see Appendix A “Detailed Hardware Configuration”.

Oracle Data Warehousing with SAS Tested Software

Specific software components of the Oracle Data Warehousing and SAS test suite and the versions tested are detailked in Table 4-3.

Table 4-3: Tested Software

Product	Version	Comments
DIGITAL Layered Products		
TruCluster		
TruCluster Available Server	1.4	
System Management		
POLYCENTER (DIGITAL)		
NetWorker Save and Restore (NSR) for DIGITAL UNIX	4.2A	

Product	Version	Comments
NetWorker Save and Restore Database Module for Oracle (DMO)	1.0	
SCSI CAM Media Changer	3.1A	
POLYCENTER (CA)		
Pathdoctor	1.0	
NETWORKING		
SNA APPC/LU6.2 Runtime for OSF/1	1.0	
COMPILERS, UTILITIES		
DEC C++ for DIGITAL UNIX Systems	5.1	
DEC COBOL for DIGITAL UNIX Systems	2.3	
DIGITAL Fortran for DIGITAL UNIX Alpha Systems, runtime library	4.0	
Micro Focus COBOL for DIGITAL UNIX	4.0	
STORAGE		
StorageWorks HSZ40 Array Controller Utility for DIGITAL UNIX	1.1	
ORACLE LAYERED PRODUCTS		
ORACLE7 SERVER for DIGITAL UNIX		
Oracle7 Server (RDBMS)	7.3.2.3.0	
Oracle Server Manager	2.3.2.0.0	
Oracle7Parallel Server Option	7.3.2.3.0	
SQL*Plus	3.3.2.0.0	
Oracle WebServer	1.0.2.0.0	
Oracle7 Enterprise Backup Utility (OEBU)	2.0.12.4	
Oracle7 Transparent Gateway for IBM DRDA for DIGITAL UNIX	Rel. 3.0.17	
Oracle products included in the above or otherwise not separately licensed included:		
Oracle Server Manager (Motif)	2.3.2.0.0	
ORACLE Common Libraries & Utilities	7.3.2.3.0	
Parallel Query Option	7.3.2.3.0	
PL/SQL (V2)	2.3.2.3.0	
Remote Operations	1.3.2.0.0	
Toolkit 2.1 Base	2.1.4.14.1	
Toolkit 2.1 Extension	2.1.4.14.1	
SLAX Parser	7.3.2.1.0	
Oracle Trace	4.0.0	
Precomp	7.3.2.1.0	
Pro*C	2.2.2.0.0	
Multimedia APIs	2.0.5.4.0	
Oracle Help	2.1.1.0.0	
Oracle7 Server / DIGITAL UNIX V4.0 compatibility patch	Patch # 424307	
SQL*Net (V2)	2.3.2.1.0	
Oracle Protocol Adapter for TCP/IP	2.3.2.1.0	
Shared Memory Patch to OEBU:	Patch #407427	

Product	Version	Comments
Large File Patch to OEBU:	Patch #407801	
SAS Institute Layered Products		
The SAS System, Our license included: • Base SAS Software: DIGITAL UNIX Alpha SAS • SAS/ACC-ORACLE: DIGITAL UNIX Alpha SAS/ACCESS Interface to ORACLE • SAS/AF • SAS/ASSIST • SAS/CALC • SAS/CONNECT • SAS/EIS • SAS/ENGLISH • SAS/ETS • SAS/FSP • SAS/GRAPH • SAS/IML • SAS/INSIGHT • SAS/LAB • SAS/OR • SAS/QC • SAS/SHARE • SAS/SHARE*NET • SAS/SPECTRAVIEW • SAS/STAT • SAS/TOOLKIT	Release 6.11	
GATEWAY and REMOTE SYSTEMS SOFTWARE		
IBM		
MVS/ESA	Ver. 5, Rel. 2, Mod. 2	
ACF/VTAM	Ver. 4, Rel. 3	
ACF/NCP	Ver. 7, Rel. 3	
DB2	Ver. 4, Rel. 1	
DIGITAL		
DEC SNA Peer Server	1.3	ECO 2
DIGITAL SNA APPC/LU6.2 Programming Interface for DIGITAL UNIX including LU6.2 Client and Server	3.0	ECO 4

Component Revision Levels

Hardware Component	Hardware	Firmware (depot1/ depot2/ depot3)	Software
StorageWorks HSZ40 Firmware		Version 2.7Z	
SCSI CAM Medium Changer Driver			Version 3.1A
RZ29B-VA		0016	
TL810-BA, DLT Tape Library		V1.10	
DECchip 21040-AA		23/24/24	
PALcode		V1.21	

Hardware Component	Hardware	Firmware (depot1/ depot2/ depot3)	Software
KFTHA, Type 2020,	D02/D03/D03		
DEC PCI MC, Type 181011, Rev. 000B			
KZPSA, Type 81011		A10	
DEC PCI FDDI, Type F1011			

Hardware Component	Hardware	Firmware
8400 console		4.1-6
SROM console		3.1
5/440 MHz CPU UNIX (756P2-AX)	E02	
2 GB memory module (MS7CC-FA)	B01	
2 GB 3.5 in. disk (RZ28D-VW)	B03	
System I/O module (KFTHA-AA)		D03
StorageWorks Plug-in unit (BA660-AB)		A01
PCI plug-in unit (DWLPB-AA)		A02
PCI plug-in unit (DWLPB-BA)		A02

Test Tools and Scripts

The Consumer Packaged Goods Database

Overview

We constructed our databases to demonstrate and test the performance capabilities of the *Oracle Data Warehousing with SAS HiTest AppSet on the DIGITAL UNIX TruCluster ASE 8000 5/440 HiTest Foundation*, in a Data Warehousing and Decision Support application.

The database was built using the Consumer Packaged Goods Database Demo scripts provided by Oracle Corporation. The Consumer Packaged Goods Database represents typical marketing and sales data for a consumer products manufacturing firm. The data includes six years of sales data in a data warehouse, star schema, optimized for decision support.

Functional Testing

Database Data Queries

Functional verification of the ability to perform query operations was demonstrated utilizing five SQL join scripts, a full table scan, a database link join, and local table create with a database link. These queries exercised functionality of the RDBMS server, SQL*Plus, Parallel Query, and Transparent Gateway components.

The scripts were designed to emulate typical decision support questions about the historical activity of a product sales environment. In most cases the result of these types of queries were to generate sales trends.

Typical of the kind of question asked was:

"What percentage of product share in dollars and units did a particular product have in a particular area as compared to competitive products in the same area?"

The queries were designed to search the database in varying ways to exercise the database. All queries returned the results grouped by month.

Characterization of the Queries

We used five queries in our testing:

Query 1

Query 1 asks *"What was the product share of a specific brand of cereal as compared to other cereals in the same product category, in a particular state in a particular type of store. The information was grouped by month to show market trends"*.

The business question asked is:

"How did 20 oz. Wheat Flakes do in 1995 as compared to all types of wheat flakes in supermarkets in the state of Connecticut?"

Query 2

Query 2 compares the sales of a specific product, in a particular outlet in a region, against the sales of the same product through all channel outlets. The information is grouped by month to show market trends.

The business question asked is:

"What percentage of sales of 15 oz. Wheat Flakes were made in the Safeway stores in NY and PA as compared to all outlets in the NY and PA areas?"

Query 3

Query 3 compares the market share of a product in a particular type of store, in a particular market location, to sales of all types of outlets in the region. The information was grouped by month to show market trends.

The business question asked is:

“How are 10 oz. Wheat Flakes doing in convenience stores in Bridgeport Connecticut as compared to the entire northeast region?”

Query 4

Query 4 compares the market share of a particular product, in a particular type of store, in a particular market location to all sales of competitive products in the same market location. The information is grouped by month to show market trends.

The business question asked is:

“What was the market share of 20 oz. Wheat Flakes in Connecticut supermarkets in 1995?”

Query 5

Query 5 compares the product share of a given product, combining several areas, to total sales across the same areas.

The business question asked is:

“What was the market share of 20 oz. Wheat Flakes across 10 test market areas?”

Test Process

Building the Consumer Product Database

The database, constructed in Oracle7, consists of six tables including: the SALES_FACTS table, which consists of the bulk of the database, and four dimension tables: PRODUCT, MARKET, CHANNEL, DAILY_PERIOD, and MONTHLY_PERIOD.

The Oracle7 Consumer Product Database consists of five tablespaces:

Table 4-4: 419.4 GB Oracle7 database

Tablespace	Contains	Size	Free Space %
FACTS	SALES_FACTS Table 12 data files (12,269 MB each)	147,228 MB 2.8 B rows (2,766,836,356)	11 %
FACTSINDEX	SF_KEY Index on SALES_FACTs Table 10 data files (12,269 MB each)	122,690 MB	12 %
DIMENSION	1 Data File Dimension Tables: CHANNEL DAILY_PERIOD MARKET PRODUCT	10 MB 41 rows 2189 rows 1002 rows 522 rows	92 %
DIMINDEX	1 Data File Index for Dimension Tables	10 MB	88 %
TEMPFILE	134 data files (12,269 MB each) required for SF_KEY index build	159,497 MB	

Test Results for Local Database Queries

Our initial tests on the AlphaServer 8000 system verified the ability to process queries, in which all query data resides on the local three-node TruCluster, in a single Oracle7database. We executed the set of five SQL join scripts, as described above, and a `SELECT` operation to implement a full table scan. Whereas our data loading was done in a mixed environment of 300 and 350 MHz CPUs in the TruCluster, Index Creation and all testing was performed after upgrading the system to be composed entirely of 440 MHz CPUs.

We subsequently conducted functional testing with the SAS System database, running scripts to test forecasting and statistical modeling functions.

Oracle Database Tests

Test One - comparison of STAR hint usage

One process executing queries 1-5 sequentially.

The first test performed the five SQL joins sequentially, with a 6 GB SGA size. This test was performed with a cold cache.

The performance of a query is dependent upon the optimizer choosing an efficient query plan. In general analyzing the tables and indexes will enable the optimizer to produce effective plans, even for star queries. However, it sometimes becomes necessary to assist the optimizer in choosing an improved query plan. Oracle7 accomplishes this through *hints*; the queries used to test our star schema are one instance where the optimizer required this help. The table below demonstrates the benefit of utilizing the *star hint*. All queries in the following sections were executed with the *star hint*.

NOTE

The SAS System permits passing hints to Oracle in the SQL pass-through, by adding preserve_comments to the connect string, and adding comments to the SQL. See SAS documentation for further discussion of preserve comments.

Performance Chart for Test One: One Process Executing 1-5 Sequentially - Cold Cache

6 GB SGA	(a) Without STAR hint	(b) With STAR hint
Query 1	12.4 min.	7.6 min.
Query 2	7.0	4.3
Query 3	3.0	3.0
Query 4	22.0	12.3
Query 5	823.9	10.8
Total	868.3 min. (14.5 hr)	38.0 min. (0.6 hr)

Test Two

The second test again performed the five SQL joins sequentially, with three different SGA sizes. The SGA sizes used were approximately 2 GB, 5 GB, and 6 GB. We varied the SGA size by setting the **`DB_BLOCK_BUFFERS`** parameter in the **`initCPGx.ora`** file, where x represents the node. Each test was performed with both a warm and a cold cache.

We used the following parameter values for each SGA:

2 GB SGA (DB_BLOCK_BUFFERS = 60000)	
Total System Global Area	2060985064 bytes
Fixed Size	52424 bytes
Variable Size	94688800 bytes
Database Buffers	1966080000 bytes
Redo Buffers	163840 bytes

5 GB SGA (DB_BLOCK_BUFFERS = 166000)	
Total System Global Area	5566470024 bytes
Fixed Size	52424 bytes
Variable Size	126765760 bytes
Database Buffers	5439488000 bytes
Redo Buffers	163840 bytes

6 GB SGA (DB_BLOCK_BUFFERS = 191000)	
Total System Global Area	6393239872 bytes
Fixed Size	52424 bytes
Variable Size	134335608 bytes
Database Buffers	6258688000 bytes
Redo Buffers	163840 bytes

The second test scenario consists of an Oracle7 Parallel Server (OPS) environment of 3 nodes, each with a different SGA size. A single process, on each of the 3 nodes, executes the same 5 queries sequentially. This test was performed with both cold and warm caches. Observations during the running of these queries indicated clearly a considerable performance benefit using VLM, primarily due to the number of index blocks that could be cached. With a warm SGA, using the 6 GB SGA shows a performance improvement of almost 75% over the 2 GB SGA.

Performance Chart for Test Two: One Process Executing Queries 1-5 Sequentially - Cold and Warm Cache

	Cold Cache			**Warm Cache**		
	2 GB SGA (2.C.1)	**5 GB SGA (2.C.2)**	**6 GB SGA (2.C.3)**	**2 GB SGA (2.W.1)**	**5 GB SGA (2.W.2)**	**6 GB SGA (2.W.3)**
Query 1	7.95 min.	7.95 min.	7.83	0.55	0.79	0.56
Query 2	4.53	4.53	4.45	3.96	1.10	0.80
Query 3	3.18	3.11	2.98	2.99	1.16	0.82
Query 4	15.73	13.24	12.61	14.62	2.16	1.60
Query 5	11.15	10.93	11.09	10.78	6.75	5.02
Total	42.54 min.	39.76 min.	38.96 min.	32.90 min	11.96 min.	8.80 min.

Test Three

Our third series of tests demonstrates the scalability of the OPS environment over a single node. In the OPS environment, as described in Test Two, 2 users on each node (6 users total) executed all 5 queries sequentially.

We compared our results in Test Three for 2 and 6 users using a 6 GB SGA, with our results in Test Two for a single user using a 6 GB SGA.

First, we compared the 6 GB SGA results with that of a single node with 2 users executing the same queries. The comparable elapsed times indicate that any overhead incurred in the OPS environment is negligible. Second, we compared the 6 GB SGA OPS results with 6 users executing queries on a single node. The OPS environment indicates a performance improvement of 40%.

The table below illustrates the shortest and longest elapsed time for each query for each test iteration.

Test Three Results

	OPS						Single node			
	2 GB		5 GB		6 GB		6 GB		6 GB	
	2 Users (3.O.1)		2 Users (3.O.2)		2 Users (3.O.3)		2 Users (3.S.1)		6 Users (3.S.2)	
	min	**max**	**min**	**max**	**min**	**max**	**min**	**max**	**min**	**max**
query1	0.58 min.	0.58 min.	0.56 min.	0.58 min.	0.61 min.	0.62 min.	0.61 min.	0.63 min.	0.83 min.	1.15 min.
query2	3.98	4.09	0.76	0.77	0.76	0.77	0.77	0.79	1.07	1.32
query3	3.23	3.23	0.85	0.85	0.88	0.89	0.83	0.90	1.25	1.64
query4	14.79	14.79	1.73	1.75	1.67	1.71	1.71	1.75	2.48	2.89
query5	12.09	12.09	5.26	5.29	5.29	5.41	5.45	5.63	7.99	9.00
total elapsed time	34.78 min.		9.24 min.		9.40 min.		9.70 min.		16.00 min.	

Test Four

To further demonstrate the scalability of the OPS environment, we doubled the workload described above. Our test of a single node with 12 users executing the same queries shows a 50% performance degradation. A total of 12 users executing queries in the OPS environment, with 4 users on each node, only suffers a 30% performance degradation. This represents a 50% performance improvement over what we observed executing the same volume of queries concurrently on a single node.

Test Four Results

	OPS						**Single node**			
	2 GB		**5 GB**		**6 GB**		**6 GB**		**6 GB**	
	4 Users (4.O.1)		**4 Users (4.O.2)**		**4 Users (4.O.3)**		**4 Users (4.S.1)**		**12 Users (4.S.2)**	
	min	**max**	**min**	**max**	**min**	**max**	**min**	**max**	**min**	**max**
query1	0.62 min.	0.64 min.	0.66 min.	0.83 min.	0.70 min.	0.88 min.	0.69 min.	0.73 min.	1.58 min.	1.71 min.
query2	3.83	4.14	0.99	1.10	0.95	0.97	0.94	1.10	2.01	2.11
query3	3.29	3.29	1.07	1.19	1.08	1.16	0.97	1.03	2.19	2.35
query4	15.27	15.27	2.18	2.32	2.05	2.27	1.98	2.01	4.44	4.58
query5	12.28	12.28	6.52	7.22	6.61	6.97	6.22	6.58	13.42	13.89
total elapsed time	35.62 min		12.66		12.25 min		11.45 min		24.64 min	

Test Five

The fifth series of tests demonstrates how the workload shown above can be paralleled and the total elapsed time further reduced. For each user above, 5 processes are used to run the 5 queries in parallel. In the previous scenarios all users were fetching the same data simultaneously. In Test Five, we varied the contents of the cache by having different queries being run concurrently by each user.

Here we tested a workload equivalent to that of the 6 users above (with 2 users per node). The total elapsed time is given by the elapsed time of the longest running query. Although, not as dramatic as the 2nd test, the benefit of the large cache shows a 50% improvement.

This test scenario again shows the significant performance benefit of the OPS environment and further demonstrates scalability of clusters. This scenario increases the number of processes in tests three and four by five. The first table indicates an OPS performance improvement of 46% over a single node. This model of distributing work (by a factor or 5) offers a 6% improvement over the serial model in test 3. The second table indicates an OPS performance improvement of 58% over a single node, an 8% improvement over the serial model in test 4. The performance improvements of OPS/cluster over a single node is shown to increase as the workload increases.

Test Five Results

	OPS						Single node			
	2 GB		5 GB		6 GB		6 GB		6 GB	
	10 processes (5.O.1)		10 processes (5.O.2)		10 processes (5.O.3)		10 processes (5.S.1)		30 processes (5.S.2)	
	min	**max**	**min**	**max**	**min**	**max**	**min**	**max**	**min**	**max**
query1	5.90 min.	6.00 min.	1.28 min.	1.45 min.	1.42 min.	1.47 min.	1.27 min.	1.34 min.	3.21 min.	3.35 min.
query2	5.15	5.34	1.61	2.11	1.82	1.85	1.66	1.69	4.31	4.50
query3	3.39	3.49	2.30	2.34	1.95	1.97	1.80	1.81	4.45	4.89
query4	13.32	13.42	3.32	3.90	3.07	3.09	2.67	3.06	6.89	6.97
query5	12.13	12.23	7.26	7.37	6.76	6.85	6.22	6.67	12.19	12.62
Total elapsed time	13.42 min		7.37 min		6.85 min		6.67 min		12.62 min	

	OPS						Single node			
	2 GB		5 GB		6 GB		6 GB		6 GB	
	20 users		20 users		20 users		20 users		60 users	
	min	**max**	**min**	**max**	**min**	**max**	**min**	**max**	**min**	**max**
query1	3.71 min.	6.67 min.	1.67 min.	2.65 min.	0.85 min.	2.43 min.	2.46 min.	2.81 min.	5.91 min.	6.14 min.
query2	4.36	5.71	2.76	3.64	1.32	3.17	3.14	3.25	7.92	8.10
query3	3.28	4.03	2.84	3.47	1.41	3.35	3.37	3.54	7.86	9.13
query4	13.27	13.81	4.32	5.67	2.62	5.07	5.04	5.27	12.31	12.61
query5	11.23	14.72	7.52	9.94	7.13	9.64	9.16	9.96	21.92	22.93
Total elapsed time	14.72		9.94		9.64		9.96		22.93	

Test Six

To exercise the Parallel Query Option, we used the following query to obtain the number of rows in the SALES_FACT table:

```
select count(*) from sales_fact;
```

We used three unique degrees of parallelism (12, 16 and 24) for this query. The high degree of parallelism, and read-ahead functionality, helped to offset an Oracle limitation of 128K bytes per I/O. Given the 32K block size we used, this limitation translated to 4 data blocks per I/O, regardless of the **`db_file_multiblock_read_count`** setting.

We chose the parallel degree of 12, to provide one thread performing I/O to each of the 12 FACTS datafiles. We observed some improvement in performance by increasing the degree to 16 (2 threads per CPU), however adding additional threads beyond 16 did little to improve overall throughput.

Test Six Results: Parallel Query: Count # of rows from sales_fact: 2,766,836,356 rows (142 GB)

Degree	**Time**
degree 12	107.5 min = 1.8 hr
degree 16	89.1 min = 1.5 hr
degree 24	89.2 min = 1.5 hr

Low-end Configuration - Single Node System Test

As a representative test of performance on a single node, we ran the series of five queries with the Oracle database opened in exclusive mode, using a 2 GB SGA, and with all DRD disk services relocated to the local node.

We tested for one, two, and four users, concurrently executing all 5 queries sequentially. Additionally, we ran a test to exercise the Parallel Query Option, performing a full table scan with a parallel degree of 12. All tests were conducted with a warm cache, 2 GB SGA.

Single Node System Test Results

2 GB SGA, warm cache					
	1 User	**2 Users**		**4 Users**	
		minimum	**maximum**	**minimum**	**maximum**
query1	0.5min.	0.6 min.	0.7 min.	0.8 min.	0.9 min.
query2	3.6	3.7	3.8	3.7	4.1
query3	2.8	3.0	3.0	3.5	3.5
query4	13.6	14.1	14.1	14.6	14.6
query5	10.2	11.6	11.6	13.6	13.6
total elapsed time	30.7 min	33.2		36.7	
Degree	**Time**				
degree 12	54 min = 0.9 hr				

SAS System Database Tests

Functional testing of SAS included both interactive and batch modes of script execution, where interactive mode utilized the X-window display manager. Scripts retrieved data from the existing Oracle CPG Data Warehouse via SAS/Access.

The scripts of the SAS test suite fall into two categories:

Forecasting Tests

We executed scripts to create both views and datasets for forecasting, at a daily, weekly, and monthly level and viewed the graphed results.

Test One

Daily forecast for a single market, single channel, single product, and 2 years of daily data to produce a 30 day ahead forecast on a daily level. The view of the graphed results contained the last 30 days and the 30 days ahead forecast.

Test Two

Monthly forecast for a single market, single channel, single product, and 2 years of data aggregated to monthly levels to produce a 12 month ahead forecast. The view of the graphed results contained all data points and forecast months:

Test Three

Weekly forecast for a single market, single channel, single product and 2 years of data aggregated to week ending levels to produce a 12 week ahead forecast. The view of the graphed results contained data for the last 12 weeks and the forecast 12 weeks.

Statistical Modeling Tests

We executed scripts to create both views and datasets for statistical modeling to detect differences between stores, channels groups, products, etc.

Test One

Build a statistical model of how channel and monthly factors effect daily sales using daily level data for a single market, single product and two years of data. The interaction between channel and month is also examined.

Test Two

Build two statistical models looking at channel effects and channel and month effects with no interactions using two years of data aggregated to the monthly channel for a single market and a single product.

Test Three

Build a repeated measure analysis to examine channel group differences over time using one years data aggregated to the monthly channel level for a single product and a single market.

Test Results for Queries via Gateway to Remote DB2 Database

The Oracle7 Transparent Gateway for IBM DRDA provides the capability to access remote, non Oracle data sources on legacy or non DIGITAL platforms. We tested the functionality of connecting to a remote database through the Transparent Gateway by executing a modified version of the first SQL join query, Test One. Test One was modified to join the small dimension tables on the remote DB2 system with the large SALES_FACT table on the local Alpha 8400 server. We successfully demonstrated three remote database access topologies using this query. The three scenarios we tested all utilized the Oracle Transparent Gateway, DEC SNA APPC/LU6.2 Server, and DEC SNA Peer Server components, but varied by the connection protocol to the Transparent Gateway and where the software resided. Scenario DB2 utilized our Configuration One, where we connected the Oracle7 database to the Oracle7

Transparent Gateway over TCP via a T1 link. Scenarios DB22 and DB2I utilized our Configuration Two, where the Oracle7 database and the Oracle7 Transparent Gateway both resided on the HiTest AlphaServer system, but were connected over TCP and IPC respectively. The location of the Transparent Gateway, the LU6.2 Server, and the SNA Peer Server, and the types of connections between them are summarized in the following table.

Table 4-5: Connections for Remote DB2 Testing

Scenario	**Location**			**Connections**		
	Transparent Gateway	LU6.2 Server	SNA Peer Server	Oracle to Transparent Gateway	Transparent Gateway to LU6.2 Server	LU6.2 Server to SNA Peer Server
DB2	Remote	Remote	Remote	TCP (via T1 link)	TCP (on same node)	TCP (on same node)
DB22	Local	Local	Local	TCP (on same node)	TCP (on same node)	TCP (via T1 link)
DB2I	Local	Local	Local	IPC (on same node)	TCP (on same node)	TCP (via T1 link)

The elapsed time to process the query showed no discernible difference in performance between the three topologies. The functionality of the Transparent Gateway and associated connection products was demonstrated in the three scenarios. Testing was extended to include table migration from the remote DB2 database to the database on the local server. A SQL script to perform table creation on the local database, with data selected from the remote database was executed for each database link. The corresponding SQL statements for the three database links (DB2, DB22, and DB2I) are:

DB2 `CREATE AS SELECT` query:

```
create table cpg.channel_tmp_db2
tablespace tempfile
as select * from cpg.channel@db2;
```

DB22 `CREATE AS SELECT` query:

```
create table cpg.channel_tmp_db22
tablespace tempfile
as select * from cpg.channel@db22;
```

DB2I `CREATE AS SELECT` query:

```
create table cpg.channel_tmp_db2i
tablespace tempfile
as select * from cpg.channel@db2i;
```

The table was successfully created, with the 21 rows from the remote DB2 database, for all three database links in under 3 seconds.

Test Results for Operational Testing

Operational Tests

The primary goal of our operational testing was to verify that the complete software configuration was capable of recovery in circumstances that occur in typical production environments, such as after performing typical system maintenance functions, and after failures caused by such things as power outages or component failures. In addition, database restores and typical maintenance of a production installation was performed with minimal service outage.

Operational tests performed included:

Test	Description	Purpose	Comments
Power Outage	Full power failure during stress test	Evaluate recovery time and database integrity	Beware of battery backup time of the HSZ caches. The write back time should be set to no more that 10 seconds
DB Tape Backup and Restore	Use available backup procedures to back up and restore the database	Determine if procedures work as advertised	See section on Backup
System Disk Backup	Perform disk backup procedures while exercising DB	Determine if typical disk maintenance procedures affect application operation	See section on Backup
Network Failure	Disconnect network link to Mainframe	Evaluate effect of result on loss of remote DB	Update data is lost unless committed
DB restore from Mainframe	Load data from DB2 database into ORACLE database	Determine if DB load procedures work as expected	Updates from DB2 to ORACLE on UNIX worked as expected. We found that using SQL Loader from flat files is more efficient than direct loading.

POLYCENTER Console Manager

POLYCENTER Console Manager, (PCM) centralizes communications and control of large, distributed, heterogeneous computing environments. From a single point of control, PCM enables system managers to perform routine system management tasks on multiple systems enterprise-wide:

- Shutdown of systems
- Rebooting systems
- Running standalone diagnostics
- Installing layered products
- Configuring HSZs
- Monitoring system and HSZ controller status

We used PCM to manage our HSZ configurations and capture failure messages, alerting us to the need for repair or replacement of faulty disk drives. Alarm conditions can be reported to a single workstation using PCM; HSZ40 consoles are connected to terminal servers, which are reported as ‘nodes’ to the PCM, simplifying the management of our large configuration of over 700 disk drives.

Backup and Restore Testing and Results

Setting up for Backup and Restore

Setting up Automated Tape Libraries (Jukeboxes) for Backup

For each jukebox we executed MAKEDEV.MC mcxx, where xx is the value of (SCSI bus ID * 8) + y and y is the value of the LUN. This command produces /dev/mcxx files.

To use the jukeboxes with *POLYCENTER NetWorker Save and Restore* (NSR) you first need to configure them by executing the NetWorker jb_config command for each jukebox you will be using. This utility creates and stores the necessary resource data that NetWorker requires to manipulate the jukebox. The jb_config command will ask you for the following jukebox information; repeat with appropriate values for each jukebox:

Jukebox Device Type:	**TL810**
jukebox name:	juke_1
jukebox device path:	/dev/mc48
jukebox tape drive path:	/dev/nrmt1h
jukebox tape drive type:	TZ87
jukebox tape drive path:	/dev/nrmt2h
jukebox tape drive type:	TZ87
jukebox tape drive path:	/dev/nrmt3h
jukebox tape drive type:	TZ87
jukebox tape drive path:	/dev/nrmt4h
jukebox tape drive type:	TZ87
jukebox bar code reader enabled:	Y
jukebox volume labels to match bar code labels:	Y

Label all tapes using the following command for each jukebox, 48 tapes per jukebox:

```
nsrjb -L -j juke_1 -v -Y noconfirm -f /dev/nrmt1h -S 1-48
```

Setting up Backup Software

Installation of software components for Backup and Restore is included in Chapter 3.

Configuring POLYCENTER NetWorker Save and Restore Including NSR Database Module for Oracle

The *NetWorker Save and Restore Database Module for Oracle* (DMO) installation procedure places files in these default locations:

Location	**File**
/usr/bin	nsrobkcon
/usr/bin	obkasm
/usr/bin	dmo_ize
/usr/shlib	libobk.so

We used the NetWorker Administrator's utility, *nwadmin* to set up *NSR*, according to the *POLYCENTER NetWorker Save and Restore Database Module for Oracle Administrator's Guide (Order No. AA-QV9BA-TE).*

Setting Up NetWorker Directives

NetWorker Directives for Oracle are set from the *Directives* window of the *Customize* pull-down menu in the *NetWorker Administrator* window.

To set Oracle directives we created the following directive named Oracle:

```
<</>>
forget
ignore
+obkasm:*
```

The Oracle directive will be associated with the client performing the backup/restore. obkasm is a DMO specific program that specifies how a set of files is to be backed up and recovered; this directive instructs the client to run obkasm on all files specified in the backup/restore.

The Oracle directive's instructions tell the process to:

`<</>>`	Start at root
`forget`	Forget previous directives
`ignore`	Ignore directives in the filesystem (.nsr files)
`+obkasm:*`	Run obkasm on all files under root (the + indicates all subdirectories)

Setting Up a NetWorker Group

We set up a single NetWorker Group for Oracle from the *Groups* window of the *NetWorker Administrator Customize* pull-down menu, according to the DMO *Administrator's Guide*. We setup one Group called Oracle, to correspond with the environment variable NSR_GROUP used by the client processes using OEBU.

Setting Up a NetWorker Client

NetWorker Clients for Oracle7 database backup are set up from the *Clients* window of the *Client Setup* pull-down menu.

In our configuration, *NetWorker Save and Restore* and the database resided on a single system, therefore we defined it using *NetWorker Administrator*. We specified "depot1" as our client system name (this is where our database resides); "Oracle" as the directive; and "Oracle" as the group. We used the default for all other settings.

Setting Up Backup Schedules

Backup schedules for *NetWorker Save and Restore* can be set up at this point in the configuration sequence. We did not set up or perform scheduled backups.

Setting Up NetWorker Volume Pools

As we used the default volume pool, no other setup was required for volume pools.

Setting Up NetWorker Save and Restore Server

We set up the *NetWorker Save and Restore* Server, to correspond with our settings in *Oracle Enterprise Backup Utility (OEBU)* for parallelism of 16, according to the *NetWorker Administrator's Guide, UNIX Version* (Digital Equipment Corp., Order No: AA-QH5CA-TE).

NSR Server Settings Used:	
Parallelism	16
Active Devices	8
Sessions per device	2

Setting Up the Oracle7 Enterprise Backup Utility

Before installing the *Oracle7 Enterprise Backup Utility* (OEBU), it is necessary to create a database to store the backup catalog. You will be asked to supply the connect string for the owner of the backup catalog during installation of the OEBU executables.

If you do not supply this information now, then you must later include the proper connection information in $OBK_HOME/admin/catalog.obk.

We used obk/obk@BKCT as our connect string, where BKCT is the Oracle SID of the backup catalog database. This connection uses SQL*Net; therefore, appropriate entries must be made in tnsnames.ora and listener.ora.

Following are the entries in tnsnames.ora and listener.ora for the catalog database, BKCT and the target database, CPG. As they both reside locally, we used the IPC communication protocol.

$ORACLE_HOME/network/admin/tnsnames.ora

```
BKCT.world=
  (DESCRIPTION=
        (ADDRESS=
           (PROTOCOL=IPC)
           (HOST=depot1)
           (KEY=BKCT)
        )
        (CONNECT_DATA=
           (SID=BKCT)
           (GLOBAL_NAME=BKCT.world)

        )
  )

CPG.world=
  (DESCRIPTION=
        (ADDRESS=
           (PROTOCOL=IPC)
           (HOST=depot1)
           (KEY=CPG)
        )
        (CONNECT_DATA=
           (SID=CPG)
           (GLOBAL_NAME=CPG.world)
        )
  )


[Blistener.ora


LISTENER =

  (ADDRESS_LIST =
        (ADDRESS =
           (PROTOCOL = IPC)
           (key = BKCT)
        )
        (ADDRESS =
           (PROTOCOL = IPC)
           (key = CPG)
        )
  )

STARTUP_WAIT_TIME_LISTENER = 0

CONNECT_TIMEOUT_LISTENER = 10

TRACE_LEVEL_LISTENER = 16

trace_directory_listener=/tmp

SID_LIST_LISTENER =
  ( SID_LIST =
    (SID_DESC =
       (SID_NAME = BKCT)
       (ORACLE_HOME = /oracle/app/oracle/product/7.3.2)
       (PROGRAM=oracle)
     )
    (SID_DESC =
       (SID_NAME = CPG)
       (ORACLE_HOME = /oracle/app/oracle/product/7.3.2)
       (PROGRAM=oracle)
     )
  )
```

Before creating the backup catalog start the SQL*Net listener:

```
lsnrctl start
```

Once the backup catalog database is created the OEBU executables can be installed using the Oracle Installer from the OEBU distribution media. This should be done after NSR and DMO are installed as it has to link against them.

During installation select "Legato NetWorker" as the 3rd party media management software vendor and entered /usr/shlib as the directory path for the Networker API library. Enter the connect string for the catalog database as described above.

Choose to re-link all executables.

Use the backup catalog `Oracle_SID BCKT`, which can be installed from the Oracle7 distribution media.

The Large File Patch to OEBU: Patch # 407801, and the Shared Memory Patch to OEBU: Patch # 407427 should be installed at this time.

For a full discussion of installing OEBU, see *Oracle7 Enterprise Backup Utility™ Installation Guide for DIGITAL UNIX, Release 2.0.12* (January 1996, Oracle Corp., Part No. A45307-1).

Functional Test of Backup and Restore

In order to execute the scripts, the UNIX user executing them must be in the UNIX operator group. The functional tests we performed for backup involved registration of the database in the backup catalog, and full database backups, conducted both off and on line. We performed restore of the database using backups taken off line and on line. No partial backups were tested.

Data restore was verified by dropping the large SALES_FACT table (8.4 million rows) after a backup was taken and canceling recovery just prior to the drop transaction. In all cases, the restore completed successfully.

Prior to the backup operations, we reduced the TEMPFILE tablespace to one datafile and removed the FACTSINDEX index tablespace entirely, resulting in the backup of approximately 430 GB of data. NSR_DEBUG flag was enabled; though the performance impact of this is unknown.

OEBU parallelism and NetWorker parallelism were both set to 10, with active devices set at 5 and sessions per devices to 2.

Scripts

Building the Oracle Database and Creating the Index

build_salem.sh

```
#!/bin/sh
# set -x
SYSTEM_TS=/oracle/links/SYSTEM
LOGA_1=/oracle/links/LOGA
LOGB_1=/oracle/links/LOGB
LOGC_1=/oracle/links/LOGC
LOGD_1=/oracle/links/LOGD
DIMENSION_TS=/oracle/links/DIMENSIONS
DIMINDEX_TS=/oracle/links/DIMENINDEX
FACTS_TS=/oracle/links/FACTS1
FACTS_INDEX=/oracle/links/FACTSINDEX1


echo
echo " ******* >>>>>>>>>>>   CREATE the DATABASE STAGE1"


svrmgrl <<!
CONNECT internal;
spool create.log
shutdown abort
startup pfile=/oracle/dbs/virginCPG.ora nomount
create database CPG controlfile reuse
datafile '$SYSTEM_TS' size 190 M reuse
logfile group 1 ('$LOGA_1')
                 size 1000 M reuse,
        group 2 ('$LOGB_1')
              size 1000 M reuse,
        group 3 ('$LOGC_1')
              size 1000 M reuse,
        group 4 ('$LOGD_1')
                 size 1000 M reuse
      MAXDATAFILES 1022
;
@/oracle/rdbms/admin/catalog.sql
@/oracle/rdbms/admin/catproc.sql
CREATE rollback segment s1  storage (initial 100k minextents 2 next 10k);
CREATE rollback segment s2  storage (initial 100k minextents 2 next 10k);
CREATE rollback segment s3  storage (initial 100k minextents 2 next 10k);
CREATE rollback segment s4  storage (initial 100k minextents 2 next 10k);
CREATE rollback segment s5  storage (initial 100k minextents 2 next 10k);
CREATE rollback segment s6  storage (initial 100k minextents 2 next 10k);
CREATE rollback segment s7  storage (initial 100k minextents 2 next 10k);
CREATE rollback segment s8  storage (initial 100k minextents 2 next 10k);
CREATE rollback segment s9  storage (initial 100k minextents 2 next 10k);
CREATE rollback segment s10  storage (initial 100k minextents 2 next 10k);


SHUTDOWN;
DISCONNECT;
EXIT
!
echo "******* >>>>>>>>>> Done Creating Tables"
date
```

```
echo "******* >>>>>>>>>> Creating the TableSpaces "

 svrmgrl <<!
   SET echo on;
   CONNECT internal;
   spool tablespace.log
   STARTUP pfile=/oracle/dbs/buildCPG.ora open CPG

   DROP TABLESPACE DIMENSION including contents;
   CREATE TABLESPACE DIMENSION datafile '$DIMENSION_TS' size 10 M reuse
   DEFAULT storage (initial 128K next 128K MAXEXTENTS 121 pctincrease 0);

   DROP TABLESPACE DIMINDEX including contents;
   CREATE TABLESPACE DIMINDEX datafile '$DIMINDEX_TS' size 10 M reuse
   DEFAULT storage (initial 32k next 32k pctincrease 0);
EXIT
!

echo "******* >>>>>>>>>> Create cpg account "

svrmgrl <<!
   SET echo on;
   CONNECT system/manager;
   spool account.log
   GRANT CONNECT,RESOURCE,UNLIMITED TABLESPACE TO cpg IDENTIFIED BY cpg;
EXIT
!

echo "******* >>>>>>>>>> Done Creating TableSpaces"
date

#
# Create tables
#

echo "******* >>>>>>>>>> Create cpg Tables "

svrmgrl <<!
   SET echo on;
   spool tables.log
   CONNECT cpg/cpg

   DROP TABLE MARKET;
   CREATE TABLE MARKET
   (
      MARKET_ID         CHAR(6),
      MARKET        VARCHAR2(30),
      DISTRICT          VARCHAR2(30),
      REGION        VARCHAR2(30),
      COUNTRY             VARCHAR2(30),
      SEQUENCE            NUMBER(6),
      MARKET_LEVEL  NUMBER(1),
      SEED1         NUMBER,
      SEED2         NUMBER,
      SEED3         NUMBER
   )
       tablespace DIMENSION
       PCTFREE 0 storage(PCTINCREASE 0);

   DROP table PRODUCT;
   CREATE TABLE PRODUCT
   (
      PRODUCT_ID        CHAR(6),
      PRODUCT             VARCHAR2(40),
      BRAND         VARCHAR2(40),
      SUBCATEGORY       VARCHAR2(30),
      PRODUCT_LEVEL     NUMBER(1),
      SEQUENCE            NUMBER(7),
      UPC           CHAR(12),
      MANUFACTURER  VARCHAR2(30),
      BASE_SIZE           NUMBER,
      CASE_PACK           NUMBER,
      MULTI_PACK    NUMBER,
      PACK_TYPE           VARCHAR2(10),
      UNIT_SIZE_DESCRIPTION VARCHAR2(30),
      SIZE_GROUP        VARCHAR2(10),
      CEREAL_TYPE   VARCHAR2(10),
      GRAIN         VARCHAR2(20),
      KEY_CHARACTERISCTIC  VARCHAR2(30),
      SEED1         NUMBER,
      SEED2         NUMBER,
      SEED3         NUMBER,
      SEED4             NUMBER
   )
       tablespace DIMENSION
       PCTFREE 0 storage(PCTINCREASE 0);

   DROP TABLE DAILY_PERIOD;
   CREATE TABLE DAILY_PERIOD
   (
      DAY           DATE,
      WEEK_ENDING   DATE,
      WEEK_SEQUENCE     NUMBER(2),
      MONTH         DATE,
      MONTH_NAME    CHAR(10),
      MONTH_SEQUENCE    NUMBER(2),
      QUARTER             CHAR(7),
      QUARTER_SEQUENCE  NUMBER(1),
      YEAR          NUMBER(4),
      SEED1         NUMBER,
      SEED2         NUMBER,
      SEED3         NUMBER
   )
       tablespace DIMENSION
       PCTFREE 0 storage(PCTINCREASE 0);
   DROP TABLE MONTHLY_PERIOD;
   CREATE TABLE MONTHLY_PERIOD
   (
      MONTH         DATE,
      MONTH_NAME    CHAR(10),
      MONTH_SEQUENCE    NUMBER(2),
      QUARTER             CHAR(7),
      QUARTER_SEQUENCE  NUMBER(1),
      YEAR          NUMBER(4),
      SEED1         NUMBER,
      SEED2         NUMBER,
      SEED3         NUMBER
   )
```

```
       tablespace DIMENSION
       PCTFREE 0 storage(PCTINCREASE 0);
   DROP TABLE CHANNEL;
   CREATE TABLE CHANNEL
   (
      CHANNEL_ID          CHAR(6),
      CHANNEL             VARCHAR2(30),
      CHANNEL_GROUP VARCHAR2(30),
      SEQUENCE            NUMBER(5),
      CHANNEL_LEVEL NUMBER(1),
      SEED1         NUMBER,
      SEED2         NUMBER,
      SEED3         NUMBER
   )
       tablespace DIMENSION
       PCTFREE 0 storage(PCTINCREASE 0);
EXIT;
!
echo "******* >>>>>>>>>> Done Creating Tables"
date
```

create_facts.sh

```
DROP TABLESPACE FACTS including contents;
   CREATE TABLESPACE FACTS datafile '$FACTS_TS' size 12269 M reuse
   DEFAULT storage (initial 64K next 1000M MAXEXTENTS 2041 PCTINCREASE 0);

   DROP TABLESPACE FACTSINDEX including contents;
   CREATE TABLESPACE FACTSINDEX datafile '$FACTS_INDEX' size 12269 M reuse
   DEFAULT storage (initial 1000M next 1000M MAXEXTENTS 2041 PCTINCREASE 0);
CONNECT cpg/cpg
DROP TABLE SALES_FACT;
   CREATE TABLE SALES_FACT
   (
      PRODUCT_ID    CHAR(6),
      MARKET_ID           CHAR(6),
      CHANNEL_ID    CHAR(6),
      DAY           DATE,
      AVG_RETAIL_PRICE    NUMBER,
      UNIT_SALES    NUMBER,
      POUND_BASIS_UNIT_SALES NUMBER,
      DOLLAR_SALES  NUMBER,
      PCT_STORES_SELLING NUMBER
   )
       tablespace FACTS
       PCTFREE 0 storage(PCTINCREASE 0);

create_temp.sh
creat_temp.sh

#!/bin/sh
# set -x
TEMP_TS1=/oracle/links/TEMP1
TEMP_TS2=/oracle/links/TEMP2
svrmgrl <<!
   SET echo on;
   CONNECT internal;
```

```
DROP TABLESPACE TEMPFILE including contents;
   CREATE TABLESPACE TEMPFILE datafile '$TEMP_TS1' size 12269 M reuse
   DEFAULT storage (initial 200M next 200M MAXEXTENTS 1041 PCTINCREASE 0);
   ALTER TABLESPACE TEMPFILE ADD datafile '$TEMP_TS2' size 12269 M;

alter user cpg temporary tablespace TEMPFILE;
EXIT;
!
```

increment_factsindexn.sh.

Below is an example of the increment_factsindex scripts used.

```
#!/bin/sh
# set -x
FACTS_INDEX1=/oracle/links/FACTSINDEX4
FACTS_INDEX2=/oracle/links/FACTSINDEX5
FACTS_INDEX3=/oracle/links/FACTSINDEX6
FACTS_INDEX4=/oracle/links/FACTSINDEX7
svrmgrl <<!
   SET echo on;
   CONNECT internal;

ALTER  TABLESPACE FACTSINDEX ADD datafile '$FACTS_INDEX1' size 12269 M;
ALTER  TABLESPACE FACTSINDEX ADD datafile '$FACTS_INDEX2' size 12269 M;
ALTER  TABLESPACE FACTSINDEX ADD datafile '$FACTS_INDEX3' size 12269 M;
ALTER  TABLESPACE FACTSINDEX ADD datafile '$FACTS_INDEX4' size 12269 M;

EXIT;
!
```

Oracle Database Test Queries

Below are Timed Queries used for “Consumer Packaged Goods” (CPG) Tests

Queries are (c) 1996 InfoDynamics LLC.

Query One

```
/*  1.  Star -- Product Share of Brand   */

select
SYSDATE,'All Wheat Flakes' Product, AL2.MONTH,
sum(AL5.UNIT_SALES) UNITS, sum(AL5.DOLLAR_SALES) DOLLARS, count(*),
DISTRICT, CHANNEL_GROUP CHNL
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND (district='Connecticut'
AND CHANNEL_GROUP in('Supermarket')
AND BRAND in ('Quellogs Wheat Flakes')
AND YEAR=1995)
group by DISTRICT, CHANNEL_GROUP,
'All Wheat Flakes', AL2.MONTH, SYSDATE
UNION
select
SYSDATE, '20 Oz Wheat Flakes' Product, AL2.MONTH,
sum(AL5.UNIT_SALES) UNITS, sum(AL5.DOLLAR_SALES) DOLLARS, count(*),
DISTRICT, CHANNEL_GROUP CHNL
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND (district ='Connecticut'
AND CHANNEL_GROUP in('Supermarket')
AND PRODUCT='QLGS WHT FLK 20 OZ'
AND YEAR=1995)
group by DISTRICT, CHANNEL_GROUP,
'20 Oz Wheat Flakes', AL2.MONTH, SYSDATE ;
```

Query Two

```
/*  2. Star -- Channel share of all channels  */

select
SYSDATE,  'All Channels' CHNL,  AL2.MONTH,
sum(AL5.UNIT_SALES) Units, sum(AL5.DOLLAR_SALES) Dollars, count(*),
'NY + PA' DISTRICT, PRODUCT
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND DISTRICT in ('New York', 'Pennsylvania')
AND CHANNEL_GROUP in ('Supermarket','Convenience',
'Warehouse','Drug','Discount')
AND PRODUCT= 'QLGS WHT FLK 15 OZ'
AND YEAR=1995
group by SYSDATE, 'NY + PA', 'All Channels',
PRODUCT , AL2.MONTH
UNION
select
SYSDATE, CHANNEL CHNL,  AL2.MONTH,
sum(AL5.UNIT_SALES) Units, sum(AL5.DOLLAR_SALES) Dollars, count(*),
'NY + PA' DISTRICT , PRODUCT
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND DISTRICT in ('New York' , 'Pennsylvania')
AND CHANNEL='Safeway'
AND PRODUCT='QLGS WHT FLK 15 OZ'
AND YEAR=1995
group by SYSDATE, 'NY + PA', CHANNEL,
PRODUCT, AL2.MONTH;
```

Query Three

```
/*  3.  Star Market share of Region  */
select
SYSDATE, 'Northeast Total' MARKET,  AL2.MONTH,
sum(AL5.UNIT_SALES) Units, sum(AL5.DOLLAR_SALES) Dollars, count(*),
CHANNEL_GROUP, PRODUCT
```

```
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND (REGION='Northeast'
AND CHANNEL_GROUP in ('Convenience')
AND PRODUCT= 'QLGS WHT FLK 10 OZ'
AND YEAR=1995)
group by SYSDATE, 'Northeast Total', CHANNEL_GROUP,
PRODUCT , AL2.MONTH
UNION
select
SYSDATE, MARKET, AL2.MONTH,
sum(AL5.UNIT_SALES) Units, sum(AL5.DOLLAR_SALES) Dollars, count(*),
CHANNEL_GROUP, PRODUCT
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND (MARKET='Bridgeport'
AND CHANNEL_GROUP in ('Convenience')
AND PRODUCT='QLGS WHT FLK 10 OZ'
AND YEAR=1995)
group by SYSDATE, MARKET, CHANNEL_GROUP,
PRODUCT, AL2.MONTH;
```

Query Four

```
/*  4.  Star -- Product share of SubCategory -all competitive prods */

select
SYSDATE, 'All Wheat Products' Product, AL2.MONTH,
sum(AL5.UNIT_SALES) Units, sum(AL5.DOLLAR_SALES) Dollars, count(*),
DISTRICT, CHANNEL_GROUP CHNL
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND (district='Connecticut'
AND CHANNEL_GROUP in('Supermarket')
AND BRAND IN ('Quellogs Wheat Flakes', 'Boast Weeties', 'Boast Oatey Rounds',
'Quellogs Wheaten Rye')
AND YEAR=1995)
group by SYSDATE, DISTRICT, CHANNEL_GROUP,
'All Wheat Products', AL2.MONTH
UNION
select
SYSDATE, '20 Oz Wheat Flakes' Product, AL2.MONTH,
sum(AL5.UNIT_SALES) Units, sum(AL5.DOLLAR_SALES) Dollars, count(*),
DISTRICT, CHANNEL_GROUP CHNL
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND (district ='Connecticut'
AND CHANNEL_GROUP in('Supermarket')
AND PRODUCT='QLGS WHT FLK 20 OZ'
AND YEAR=1995)
group by SYSDATE, DISTRICT, CHANNEL_GROUP,
'20 Oz Wheat Flakes', AL2.MONTH;
```

Query Five

```
/*  5.  Star -- Product share of brand in 10 test markets aggregated */

select
SYSDATE, 'All Wheat Flakes' Product, AL2.MONTH,
sum(AL5.UNIT_SALES) Units, sum(AL5.DOLLAR_SALES) Dollars, count(*),
CHANNEL_GROUP CHNL, '10-States'
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND (district in ('Connecticut', 'Delaware','Maine','Pennsylvania','New York',
'Oregon', 'Alaska', 'CA North', 'CA South','Washington')
AND CHANNEL_GROUP in('Supermarket')
AND BRAND IN ('Quellogs Wheat Flakes')
AND YEAR=1995)
group by
SYSDATE, 'All Wheat Flakes', AL2.MONTH, CHANNEL_GROUP, '10-States'
UNION
select
SYSDATE, '20 Oz Wheat Flakes' Product, AL2.MONTH,
sum(AL5.UNIT_SALES) Units, sum(AL5.DOLLAR_SALES) Dollars, count(*),
CHANNEL_GROUP CHNL, '10-States'
FROM PRODUCT AL4, SALES_FACT AL5,
CHANNEL AL1, DAILY_PERIOD AL2, MARKET AL3
WHERE (AL5.PRODUCT_ID=AL4.PRODUCT_ID
AND AL5.MARKET_ID=AL3.MARKET_ID
AND AL5.CHANNEL_ID=AL1.CHANNEL_ID
AND AL5.DAY=AL2.DAY)
AND (district in ('Connecticut', 'Delaware','Maine','Pennsylvania','New York',
'Oregon', 'Alaska', 'CA North', 'CA South','Washington')
AND CHANNEL_GROUP in('Supermarket')
AND PRODUCT='QLGS WHT FLK 20 OZ'
AND YEAR=1995)
group by
SYSDATE, '20 Oz Wheat Flakes', AL2.MONTH ,CHANNEL_GROUP, '10-States';
```

SAS Institute Database Files and Scripts

Example of Forecast Script - `forecast_month_dataset.sas`

```
libname foo '/sas1';
goptions device=xcolor;
proc sql;
connect to oracle(user=cpg orapw=cpg);
create table forcastm as select * from connection to oracle(
select
 market,product,sum(dollar_sales) as dollar,month,channel,
sum(unit_sales) as unit
from market,product,
daily_period,channel,sales_fact where market.market_id=sales_fact.market_id and
product.product_id=sales_fact.product_id and
channel.channel_id=sales_fact.channel_id and
 daily_period.day=sales_fact.day and market.market='Albany' and
channel.channel='Wal Mart' and
product.product='NUGN CTY WHT FLK 20 OZ' and
daily_period.month>'31-DEC-93' and month <'01-JAN-96'
group by market, product, month, channel);
quit;

proc forecast data=forcastm out=month outfull lead=12
              interval=dtmonth;
var dollar  unit;
id month ;
run;

data month; set month;
    date=datepart(month) ;  format date date7.;
run;
proc gplot data=month;
 plot dollar*date =_type_  /
 haxis='01jan94'd to '01dec96'd by month  href='01jan96'd;
 symbol1 i=none v=star h=1;
 symbol2 i=spline v=circle;
 symbol3 i=spline l=3;
 symbol4 i=spline l=3;
 legend  value=(  font=swissx) label=(font=swissx);
run;
quit;
```

Example of Statistical Modeling Script - channel_diff_month.sas:

```
libname foo '/sas1';
proc sql;
connect to oracle(user=cpg orapw=cpg);
create table channeld as select * from connection to oracle(
select
 market,product,avg(dollar_sales) as dollar,month,channel,
avg(unit_sales) as unit,count(daily_period.day) as weight
from market,product,
daily_period,channel,sales_fact where market.market_id=sales_fact.market_id and
product.product_id=sales_fact.product_id and
channel.channel_id=sales_fact.channel_id and
 daily_period.day=sales_fact.day and market.market='Wales 000' and
product.product='NUGN CTY WHT FLK 20 OZ' and
daily_period.month>'31-DEC-93' and month <'01-JAN-96'
group by market, product, month, channel
order by channel, month);
quit;

data month; set channeld;
    date=datepart(month) ;  format date date7.;
 if substr(channel,1,5)='Total' then delete;
run;
proc glm data=month; class channel;
model dollar= channel;
means channel/bon duncan;
run;quit;
```

```
proc glm data=month; class month channel;
model dollar= channel month;
means channel month/bon duncan;
run;quit;
```

Remote DB2 Database Files and Scripts

File db22 used in LU6.2 server startup script

```
#
# Describe the environment:
#      LU62_SRV     - name of LU62 Server (there could be more
#                         than 1 LU62 server on a node)
#      depot1 - the LU62 server executes on this node
#      esacn1 - the Digital/SNA gateway runs on this node
#      TCP TCP      - Client to LU62 server uses TCP
#                   - LU62 server to Digital/SNA gateway uses TCP
#
IDS             LU62_SRV        depot1 esacn1 TCP TCP
#
# Describe the Local LU:
#      USDEC101     - the 'network name' (in our example, all
#                     SNA resources are contained within a single
#                     SNA network called USDEC101.
#      IESACN01     - the 'lu name' is the name of the control
#                     unit that exists in Digital's network.  This
#                     parameter MUST match the Peer Server
#                     definitions 'create SNA lu services lu'
#
LOCAL           USDEC101        IESACN01
#
# Describe the Remote LU: (in this case, IBM's DB2 data base)
#      USDEC101        - the 'network name' (in our example, all
#                         SNA resources are contained within a single
#                   SNA network called USDEC101.
#      DB2MVS - this name is the IBM mainframe LU name. It
#                         MUST match the Peer Server definitions
#                         'create SNA lu services partner lu'
#
REMOTE          USDEC101        DB24MVS
#
# Initialize the SNASVCMG mode for LUs IESACN01 and DB24MVS
#
INT_SVC
#
# Initialize the #INTER mode for LUs IESACN01 and DB24MVS
#
INT_MODE        #INTER  2       1
#
# Define mode XL6216K for LUs IESACN01 and DB24MVS
#
MODE            XL6216K         8   1024 16384 16384 1024 16384 16384
#
# Initialize the XL6216K mode for LUs IESACN01 and DB24MVS
#
INT_MODE        XL6216K 2       1
EXIT
```

Basic Peer Server Configuration File -- One Token Ring Adapter

```
!
!  CONFIGURE PEER SERVER with
! -1 intranode transmission group (for loop back testing)
! -1 external transmission group to one IBM mainframe
! -1 one token ring adapter for use by the external transmission group
! -1 enable all resources
!
! NOTE: See the next configuration file for additional
! configuration information.
!
create node 0 SNA access server
!
create node 0 llc2
create node 0 llc2 sap SNA-0
set node 0 llc2 sap SNA-0 -
        lan station = Token Ring station TRN-0, -
        local lsap address = 04
create node 0 llc2 sap SNA-0 link LINK-0
set node 0 llc2 sap SNA-0 link LINK-0 -
        acknowledge timer = 1000, -
        holdback timer = 5, -
        local receive window size = 127, -
        maximum data size = 17200, -
        remote lsap address = 04, -
        remote mac address = 02-00-B4-37-7B-C3, -
        retry maximum = 10
!
! NOTE: The remote mac address value is the IBM MAINFRAME ADDRESS.
!      The value declared in the Peer Server Configuration file
!     if DIFFERENT from the value described in IBM mainframe
!     documentation -- each byte has been "flipped".  For instance
!     the fourth byte of IBM address, x'37', is shown on the IBM
!     mainframe as x'EC'.
!
create token ring
create token ring station TRN-0 communication port tra0
set token ring station TRN-0 ring speed 16
create node 0 SNA lu services
set node 0 SNA lu services default transmission group =
create node 0 SNA cp services -
        network id = USDEC101, -
        cp name = ESACN1, -
        maximum active TGs = 10
create node 0 SNA cp services transmission group Intranode -
        intranode = Yes
create node 0 SNA cp services transmission group PUESACNX -
        data link = LLC2 sap SNA-0 link link-0, -
        intranode = No, -
        dependent lu support = Yes, -
        connection type = Permanent, -
        node id = %XDEC80150
!
! NOTE: The node id value must match the IBM mainframe values
! IDBLK and IDNUM in the switched major node definition
!
! Enable all entities
!
```

```
enable node 0 llc2 sap sna-0
enable node 0 llc2 sap sna-0 link *
enable token ring station TRN-0
enable node 0 SNA lu services lu *
enable node 0 SNA cp services transmission group *
enable node 0 SNA access server
```

Peer Server Parameters Added to Support Oracle7 Transparent Gateway for DRDA and IBM DB2

```
!
! DEFINE and ENABLE AN INDEPENDENT LU
!
create SNA lu     services lu IESACN01
set    SNA lu     services lu IESACN01 dependent lu address = 0
set    SNA lu     services lu IESACN01 capability = Both
set    SNA lu     services lu IESACN01 maximum active sessions = 0
set    SNA lu     services LU IESACN01 object IESACN01
enable SNA lu     services lu IESACN01
!
! DEFINE AN ACCESS SERVER OBJECT TO ROUTE SESSION REQUEST to DEPOT1/9001
!
create SNA access server   object     IESACN01
set    SNA access server   object     IESACN01 Internet Node = depot1
set    SNA access server   object     IESACN01 port         =  9001
!
! NOTE: The internet node MUST correspond to the node
! where the Digital SNA APPC/LU6.2 Server is to be run.
! The port must match the /etc/services '_SRV_SESS' entry
! for that server on that node.
!
! DEFINE IBM MAINFRAME DB2 system to PEER SERVER
!
create SNA lu services partner lu DB24MVS
set SNA lu services partner lu DB24MVS destination name   = DB24MVS
set SNA lu services partner lu DB24MVS destination network = USDEC101
set SNA lu services partner lu DB24MVS transmission group = { PUESACNX }
!
!     NOTE: The destination name must match the value in the IBM MAINFRAME
!           application major node
!
!        The transmission group show the list of transmission group names
!            (in this case just one) can be used to reach DB24MVS.
!
!           The destination network name must match the value used by VTAM
!          during its startup processing
!
```

DIGITAL SNA APPC/LU6.2 Client Configuration File

Copy of the edited default configuration file located at /etc/cpic.conf, created during the DIGITAL SNA APPC/LU6.2 product installation:

```
#
# The DEFAULT entry provides default values for contacting the
# SNA LU62 SERVER:
#     server_node         - node where LU62 server is running
#     server_name         - name of LU62 server on node (there
#                               could be more than one server
#                               running on a node)
#      server_transport    - Protocol to use to reach LU62_server
#                          (TCP means use the port number
#                          specified by the _CLI  entry in the
#                          /etc/services file)
#
DEFAULT {
             server_node=depot1,
             server_name=LU62_SRV,
             server_transport=TCP
}

### DB2 DRDA server named DB22
#
# The DB22 entry provides information in addition to the DEFAULT
# entry:
#      mode_name           - use this mode name to communicate
#                            When a session is not available,
#                            try to establish one unless alread
#                            at the session limit
#      local_lu_name       - application program should use this
#                            value as the "local" lu
#      partner_lu_name     - the is the name of the distant
#                            application.  In this case, it means
#                            DB2 on the IBM mainframe
#      tp_name             - the name of the transaction to run
#                            on the DB2 mainframe.  The name is
#                            specified in hexidecimal.  This is
#                            is the name required by IBM's DB2
#                            to support DRDA functions.
DB22 {
             mode_name = XL6216K,
             local_lu_name = USDEC101.IESACN01,
             partner_lu_name = USDEC101.DB24MVS,
             tp_name = \07f6c4c2
}
## DB2 DRDA server named DB2I
#
# This entry is identical to the one just above, except for entry
# name.  We chose to have unique entries that corresponded to the
# Oracle 7 data base "links".
#
DB2I {
             mode_name = XL6216K,
             local_lu_name = USDEC101.IESACN01,
             partner_lu_name = USDEC101.DB24MVS,
             tp_name = \07f6c4c2
}
```

5

System Limits and Performance Data

It was not within the scope of our testing to specifically determine system limitations or provide comprehensive performance characterization. Performance data that is available, will be found in the context of the testing we performed that is documented in Chapter 4, *Interoperability Tests and Results*.

6 Problems and Resolution

Executive Summary

The DIGITAL HiTest Testing Center, located in Salem N.H., is chartered to integrate and test DIGITAL software and hardware with selected third party applications and middleware such as from the Oracle Corporation and the SAS Institute.

The integrated *Oracle Data Warehousing with SAS HiTest AppSet on the DIGITAL UNIX TruCluster ASE 8000 5/440 HiTest Foundation* targets the Data Warehousing market, and was sponsored and funded by the DIGITAL HiTest program office. The Data Warehousing environment tested consisted of an 8X8 TurboLaser (eight processors, with eight MB of memory), and one TB of storage, loaded with a very large version of the Oracle Consumer Packaged Goods Database. Over 20 DIGITAL software products were integrated with products of the Oracle Corporation and The SAS Institute, including *Oracle7 Server*, *Oracle7 Transparent Gateway for IBM DRDA* and *the SAS System*. DIGITAL products included Storage, System Management, Networking, and Compiler products.

Below is a summary of problems found. In all we encountered 27 problems of various kinds during this systems integration effort. Of the 27 problems and cautions documented, 17 were previously encountered and documented in the documents: *Digital VLM64 Suite for Oracle Data Warehousing* (June, 1996), or *HiTest Notes for Oracle Data Warehousing DIGITAL UNIX AlphaServer 8000 Family* (October, 1996). Additional detail and the context for these problems can be found in the appropriate chapter of this document.

Summary of Findings

We have summarized our findings according to the aspect of the project where problems were encountered: Scaling the Database, the Oracle7 Database, Backup, the SAS System, Storage, Networking, System Management, the DIGITAL UNIX Operating System, and the TruCluster Available Server Environment.

Each category includes the primary areas we found: bugs, installation, or configuration problems. Each problem encountered could have been a significant disruption, if encountered in the course of installation in a customer environment. We believe that without benefit of the testing conducted by the HiTest Program, some of these problems might not have surfaced until a comparable system was installed in the field.

Details of Findings

Issues Related to Scaling Database to One Terabyte

Two (2) problems related to scaling the Oracle database were encountered during testing.

	Problem	Resolution
1.	Our initial estimate of time required for data load using the `INSERT` statement was greater than 60 days.	Developed tool to generate load using SQL Loader (`sqlldr`) input.
2.	Additional storage space is required on the system for loading data, beyond that required for the database components themselves.	Temporarily reducing database index and temp space can free up space for data loading.

Oracle7 Database Issues

Six (6) problems related to the Oracle7 database and the Oracle7 Enterprise Backup Utility were encountered during the course of testing. All problems were resolved by working directly with Oracle.

	Problem	Resolution
3.	Failure to create index under DIGITAL UNIX V. 4.0A.	Installed Oracle7 Server / Digital UNIX V4.0 compatibility patch # 424307.
4.	Our testing of our five queries with a 6 GB SGA and cold cache, initially took 14 hours.	Utilizing *star hint* when running the Optimizer, reduced the elapsed time of Query 5 to just under eleven minutes. (See Page 4–10)
5.	There is an undocumented limitation on multi-block read count; due to a maximum of 128 Kb per I/O, the maximum setting is 4 according to the formula: maximum db_file_ multiblock_read_count =max_io_size/db_block_size (128Kb/32Kb)	As we used 32K data blocks (db_blocks) we set this parameter to 4, which is the maximum based on the formula. (See Page 3–15)

Oracle7 Enterprise Backup Utility (OEBU):

	Problem	Resolution
6.	OEBU V2.0.12.4 has a 4 GB limitation on data file size.	Oracle logged bug # 407801 and provided a patch. (See Page 3–5.)
7.	OEBU V2.0.12.4 requires a single shared memory segment and won't allocate a 2nd segment as required.	Oracle logged bug # 407427 and provided a patch. (See Page 3–5.)

8.	Ambiguous question asked during installation of backup catalog, when performed after installation of OEBU executables.	Answer "YES" to the question on re-installing OEBU software. Use alternative method of installing catalog database. (See Page 4–21.)

SAS System Issues

Two (2) problems related to the SAS System were encountered during the course of testing. All problems were resolved by working directly with the SAS Institute.

	Problem	**Resolution**
9.	SAS System found to be incompatible with DIGITAL UNIX 4.0	Installed mandatory SAS Patch: SAS Note V6-SYS.SYS-C443, to ensure error-free operation of the SAS system with DIGITAL UNIX 4.0 and higher. (See Page 3–6.)
10.	SAS Oracle/ACCESS compatibility with Oracle V7.3.2.x	Changes in Oracle library names in Oracle7 Server, Version 7.3.2 cause errors to be reported when building SAS/ACCESS for Oracle Version 7.3.2. Installed SAS Patch: SAS Note V6-SYS.SYS-C398, to ensure compatibility with Oracle7 Server Version 7.3.2. (See Page 3–6.)

DIGITAL AlphaServer Hardware

We experienced two (2) problems related to early life failures of KN7CE-AB dual CPU modules during the course of testing.

	Problem	**Resolution**
11.	AS8400 DIGITAL UNIX panic after a machine check on CPU2 could not boot successfully.	Removed CPU.
12.	AS8400 hangs. No response from console and no communication with other cluster members.	We removed the CPU in Slot 3. Self test LED was off. After removing the CPU, the system became stable.

DIGITAL Storage

We experienced four (4) problems related to Storage in the course of testing. All problems were worked directly with The DIGITAL Storage group.

Storage Arrays

	Problem	**Resolution**
13.	Unless max cache threshold size is set properly, caching won't be performed with large database blocks.	Set the HSZ write-back cache high enough to allow the 32K db_block_size, to be cached. (See Page 3–4.)
14.	Database build and query failures caused by HSZ40 battery failure.	New HSZ40 firmware, version 3.0z-2 will minimize potential for problems due to battery failure, by shutting down the affected controller, and failing over to the second of the dual controllers.

15. Removal of KZPSA Controllers causes renumbering of devices, requiring re-mapping.	Carefully plan and execute system re-configuration. Take care to plan and correctly map existing device locations to new locations. This can be accomplished by taking a snapshot of the configuration using show config.
16. HSZ cache not properly flushed before planned power shutdown.	Take care to perform full HSZ shutdown procedure as documented in StorageWorks Array Controllers, HS Family of Array Controllers, User's Guide

Networking

We experienced six (6) problems related to Networking products during the course of testing. Problems were encountered in the course of setting up a T1 link to our remote DB2 database on an IBM Mainframe and testing Oracle/DIGITAL SNA and DB2 gateway access.

Testing

Problem	**Resolution**
17. During installation of Oracle7 Transparent Gateway for IBM DRDA for DIGITAL UNIX, applications should be re-linked automatically.	During installation, be sure to specify RE-LINK ALL APPLICATIONS. This is NOT the default value. (See Page 3–11.)
18. We could not get the gateway to function at first. To use SQL*NET over TCP between Oracle7 server and the DRDA components, SQL*NET and TCP components must be installed in the same pass as Transparent Gateway for IBM DRDA.	On recommendation from Oracle, we de-installed all components, and re-installed them in a single pass.
19. At completion of the Oracle7 Transparent Gateway for IBM DRDA for DIGITAL UNIX installation procedure, the directions say to run root.sh. We did this, but were worried that it might conflict with whatever had been done by 'root.sh' when the Oracle7 server was installed.	On recommendation from Oracle, skip running 'root.sh' on installation of Oracle7 Transparent Gateway for IBM DRDA for DIGITAL UNIX.
20. Some documents are ambiguous in their reference to file trees.	Where file trees are mentioned in the manual, we found it important to take care to determine whether the file indicated was the one in the TG4DRDA file tree, or in the Oracle7 server file tree.

21. SQL*Plus returns identical error:
`“Error ORA-9100: target system returned the following message sqlcode=-204”`
for two different exception conditions:
`- access a table that contains no rows`
and
`- access a non-existent table.`
	Known problem: -204 error code should be translated into an Oracle message.

22. Ambiguous error information returned: `“Error ORA-01460: unimplemented or unreasonable conversion requested”` did not reference row or column in DB2 involved.	Known problem, under investigation.

Systems Management

We experienced two (2) problems, related to System Management products, during the course of testing.

POLYCENTER

Networker *Save and Restore*, NSR Database Module for Oracle:

Problem	Resolution
23. OEBU group must be set to match a valid NSR group.	Group must be set up to correspond with the environment variable NSR_GROUP used by the client processes using OEBU. See Page 4–19 for proper setup procedure.
24. Operator permission problem: on running restore, we found we did not have appropriate privileges.	During setup of Oracle Account on DIGITAL UNIX, add the Oracle7 Enterprise Backup Utility user to Operator group. The user who will be performing restores using OEBU must be included in the Operator Group set in /etc/passwd. (See Page 3–5.)

UNIX Operating System

We experienced one (1) problem, related to the DIGITAL UNIX operating system, during the course of testing and a number of operating system patches were added to eliminate known problems. See Page 3–14 for details.

Problem	Resolution
25. SSM and use of gh-chunks are mutually exclusive.	Disable default setting of SSM memory: in the `/etc/sysconfigtab` file (ipc section) to: ssm-threshold=0

TruCluster Available Server Environment

We experienced two (2) problems, related to the DIGITAL TruCluster Available Server Environment during the course of testing.

	Problem	Resolution
26.	Encountered file protections on all Distributed Raw Disk (DRD) files on all nodes.	Set Protections to allow database access on each cluster node; all files associated with the Oracle database in /dev/rdrd/ must be owned by Oracle, and in group DBA.
27.	`svrmgrl` fails to run on node being NFS served the Oracle Software.	Coordinate group membership for the oracle account on each node so the oracle user has group access to the NFS shared files

See the relevant chapter of this document for further operational and characterization details.

A

Detailed Hardware Configuration

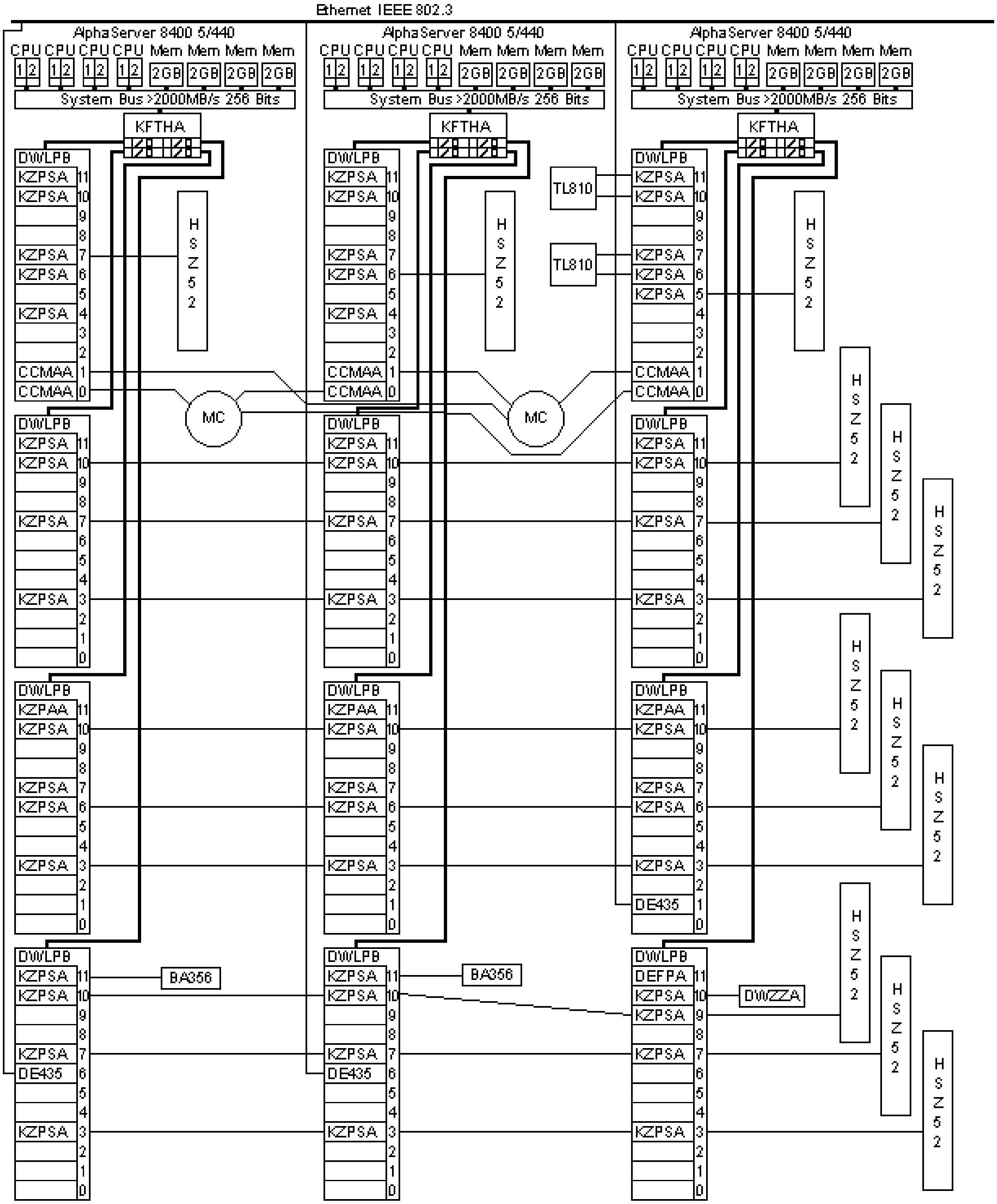
Appendix A contains a detailed configuration document for the one terabyte, DIGITAL AlphaServer 8400 TruCluster HiTest System:

- System Overview
- VLM AlphaServer 8400 Three Node TruCluster configuration, including
  - Turbo Laser System Bus
  - PCI backplanes
- Cabling Configuration

VLM AlphaServer 8400 Three Node TruCluster System Overview

Figure A-1 shows an overview of the HiTest Suite.

Figure A-1: Overview of the AlphaServer 8400 TruCluster HiTest System



AlphaServer 8400 (DEPOT1)

AlphaServer 8400 (DEPOT1) Turbo Laser System Bus (TLSB) Usage

Figure A-2: (DEPOT1) System Cabinet TLSB Centerplane Module Layout

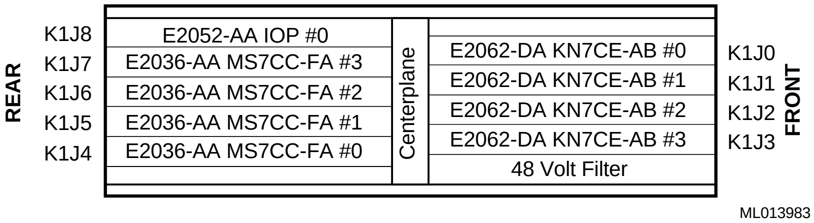


Table A-1: (DEPOT1) TLSB Centerplane Layout

SLOT	MODULE	DEVICE	DESCRIPTION
K1J0	E2062-DA #0	KN7CE-AB	TurboLaser Dual CPU Module
K1J1	E2062-DA #1	KN7CE-AB	TurboLaser Dual CPU Module
K1J2	E2062-DA #2	KN7CE-AB	TurboLaser Dual CPU Module
K1J3	E2062-DA #3	KN7CE-AB	TurboLaser Dual CPU Module
K1J4	E2036-AA #0	MS7CC-FA	2.1GB Memory Module
K1J5	E2036-AA #1	MS7CC-FA	2.1GB Memory Module
K1J6	E2036-AA #2	MS7CC-FA	2.1GB Memory Module
K1J7	E2036-AA #3	MS7CC-FA	2.1GB Memory Module
K1J8	E2052-AA #0	KFTHA-AA	System I/O Module

AlphaServer 8400 (DEPOT1) PCI #0 Device Identification

Figure A-3: (DEPOT1) System Cabinet PCI #0 Module Layout

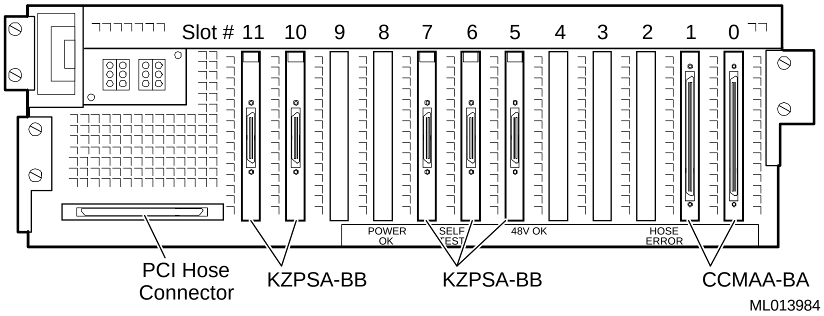


Table A-2: (DEPOT1) PCI #0 Devices

SLOT	MODULE/DEVICE	DESCRIPTION
K0J0	CCMAA-BA #0	PCI to MEMORY CHANNEL Host Bus Adapter
K0J1	CCMAA-BA #1	PCI to MEMORY CHANNEL Host Bus Adapter
K0J2	Available	
K0J3	Available	
K0J4	Available	
K0J5	KZPSA-BB #0	PCI to SCSI (FWD) Host Bus Adapter
K0J6	KZPSA-BB #1	PCI to SCSI (FWD) Host Bus Adapter
K0J7	KZPSA-BB #2	PCI to SCSI (FWD) Host Bus Adapter
K0J8	Available	
K0J9	Available	
K0J10	KZPSA-BB #3	PCI to SCSI (FWD) Host Bus Adapter
K0J11	KZPSA-BB #4	PCI to SCSI (FWD) Host Bus Adapter

AlphaServer 8400 (DEPOT1) PCI #1 Device Identification

Figure A-4: (DEPOT 1) System Cabinet PCI #1 Module Layout

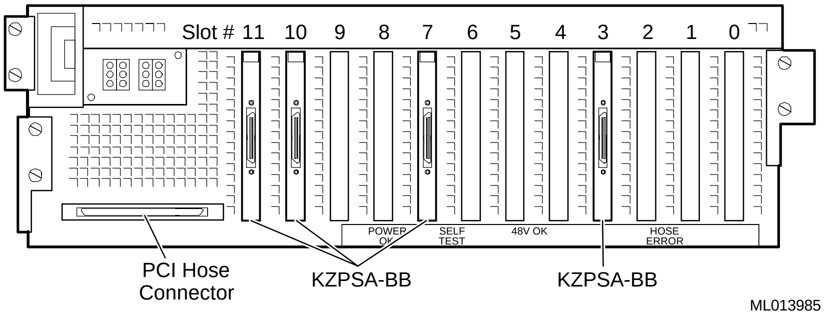


Table A-3: (DEPOT1) PCI #1 Devices

SLOT	MODULE/DEVICE	DESCRIPTION
K1J0	Available	
K1J1	Available	
K1J2	Available	
K1J3	KZPSA-BB #5	PCI to SCSI (FWD) Host Bus Adapter
K1J4	Available	
K1J5	Available	
K1J6	Available	
K1J7	KZPSA-BB #6	PCI to SCSI (FWD) Host Bus Adapter
K1J8	Available	
K1J9	Available	
K1J10	KZPSA-BB #7	PCI to SCSI (FWD) Host Bus Adapter
K1J11	KZPSA-BB #8 (Unused)	PCI to SCSI (FWD) Host Bus Adapter

AlphaServer 8400 (DEPOT1) PCI #2 Device Identification

Figure A-5: (DEPOT1) System Cabinet PCI #2 Module Layout

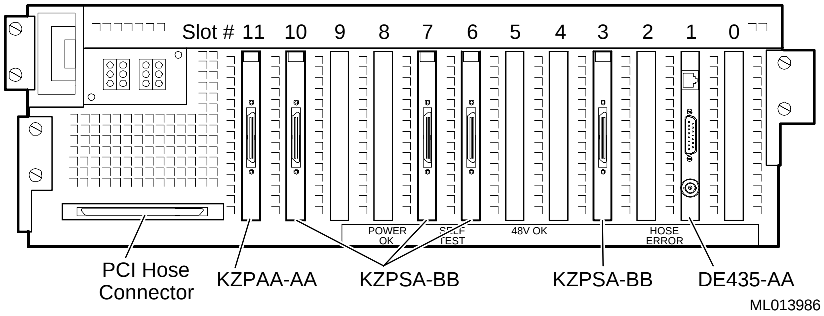


Table A-4: (DEPOT1) PCI #2

SLOT	MODULE/DEVICE	DESCRIPTION
K2J0	Available	
K2J1	DE435-AA #0	PCI to Ethernet Interface Adapter
K2J2	Available	
K2J3	KZPSA-BB #9	PCI to SCSI (FWD) Host Bus Adapter
K2J4	Available	
K2J5	Available	
K2J6	KZPSA-BB #10	PCI to SCSI (FWD) Host Bus Adapter
K2J7	KZPSA-BB #11 (Unused)	PCI to SCSI (FWD) Host Bus Adapter
K2J8	Available	
K2J9	Available	
K2J10	KZPSA-BB #12	PCI to SCSI (FWD) Host Bus Adapter
K2J11	KZPAA-AA #0	PCI to SCSI (FNSE) Host Bus Adapter

AlphaServer 8400 (DEPOT1) PCI #3 Device Identification

Figure A-6: (DEPOT1) System Cabinet PCI #3 Module Layout

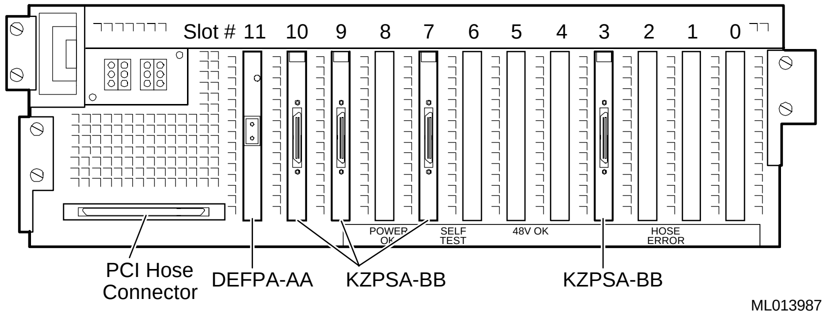


Table A-5: (DEPOT1) PCI #3

SLOT	MODULE/DEVICE	DESCRIPTION
K3J0	Available	a
K3J1	Available	a
K3J2	Available	a
K3J3	KZPSA-BB #13	PCI to SCSI (FWD) Host Bus Adapter
K3J4	Available	a
K3J5	Available	a
K3J6	Available	a
K3J7	KZPSA-BB #14	PCI to SCSI (FWD) Host Bus Adapter
K3J8	Available	a
K3J9	KZPSA-BB #15	PCI to SCSI (FWD) Host Bus Adapter
K3J10	KZPSA-BB #16	PCI to SCSI (FWD) Host Bus Adapter
K3J11	DEFPA-AA #0	PCI to FDDI Host Bus Adapter

AlphaServer 8400 (DEPOT2)

AlphaServer 8400 (DEPOT2) Turbo Laser System Bus (TLSB) Usage

Figure A-7: (DEPOT2) System Cabinet TLSB Centerplane Module Layout

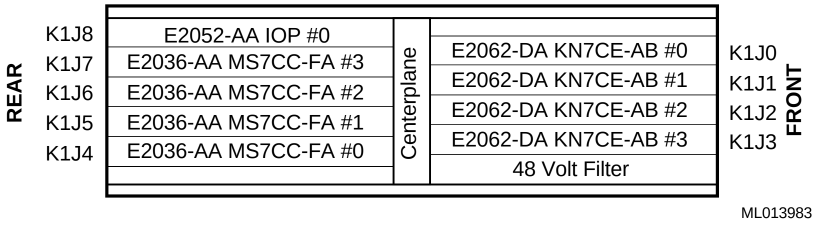


Table A-6: (DEPOT2) TLSB Centerplane Layout

SLOT	MODULE	DEVICE	DESCRIPTION
K1J0	E2062-DA #0	KN7CE-AB	TurboLaser Dual CPU Module
K1J1	E2062-DA #1	KN7CE-AB	TurboLaser Dual CPU Module
K1J2	E2062-DA #2	KN7CE-AB	TurboLaser Dual CPU Module
K1J3	E2062-DA #3	KN7CE-AB	TurboLaser Dual CPU Module
K1J4	E2036-AA #0	MS7CC-FA	2.1GB Memory Module
K1J5	E2036-AA #1	MS7CC-FA	2.1GB Memory Module
K1J6	E2036-AA #2	MS7CC-FA	2.1GB Memory Module
K1J7	E2036-AA #3	MS7CC-FA	2.1GB Memory Module
K1J8	E2052-AA #0	KFTHA-AA	System I/O Module

AlphaServer 8400 (DEPOT2) PCI #0 Device Identification

Figure A-8: (DEPOT2) System Cabinet PCI #0 Module Layout

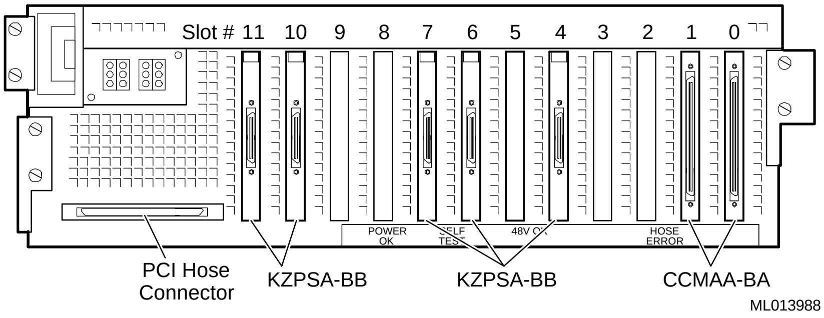


Table A-7: (DEPOT2) PCI #0

SLOT	MODULE/DEVICE	DESCRIPTION
K0J0	CCMAA-BA #0	PCI to MEMORY CHANNEL Host Bus Adapter
K0J1	CCMAA-BA #1	PCI to MEMORY CHANNEL Host Bus Adapter
K0J2	Available	
K0J3	Available	
K0J4	KZPSA-BB #0 (Unused)	PCI to SCSI (FWD) Host Bus Adapter
K0J5	Available	
K0J6	KZPSA-BB #1	PCI to SCSI (FWD) Host Bus Adapter
K0J7	KZPSA-BB #2 (Unused)	PCI to SCSI (FWD) Host Bus Adapter
K0J8	Available	
K0J9	Available	
K0J10	KZPSA-BB #3 (Unused)	PCI to SCSI (FWD) Host Bus Adapter
K0J11	KZPSA-BB #4	PCI to SCSI (FWD) Host Bus Adapter

AlphaServer 8400 (DEPOT2) PCI #1 Device Identification

Figure A-9: (DEPOT2) System Cabinet PCI #1 Module Layout

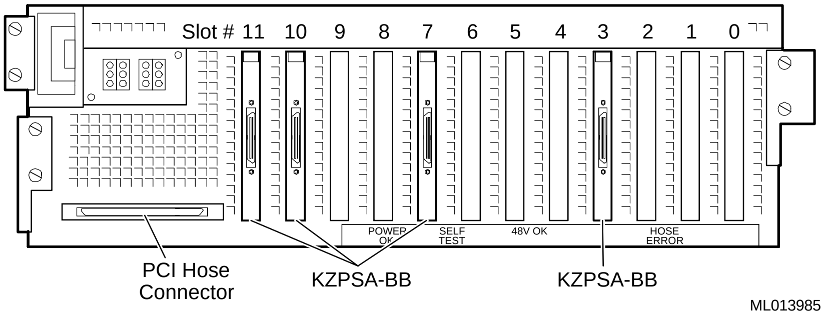


Table A-8: (DEPOT2) PCI #1 Devices

SLOT	MODULE/DEVICE	DESCRIPTION
K1J0	Available	
K1J1	Available	
K1J2	Available	
K1J3	KZPSA-BB #5	PCI to SCSI (FWD) Host Bus Adapter
K1J4	Available	
K1J5	Available	
K1J6	Available	
K1J7	KZPSA-BB #6	PCI to SCSI (FWD) Host Bus Adapter
K1J8	Available	
K1J9	Available	
K1J10	KZPSA-BB #7	PCI to SCSI (FWD) Host Bus Adapter
K1J11	KZPSA-BB #8	PCI to SCSI (FWD) Host Bus Adapter

AlphaServer 8400 (DEPOT2) PCI #2 Device Identification

Figure A-10: (DEPOT2) System Cabinet PCI #2 Module Layout

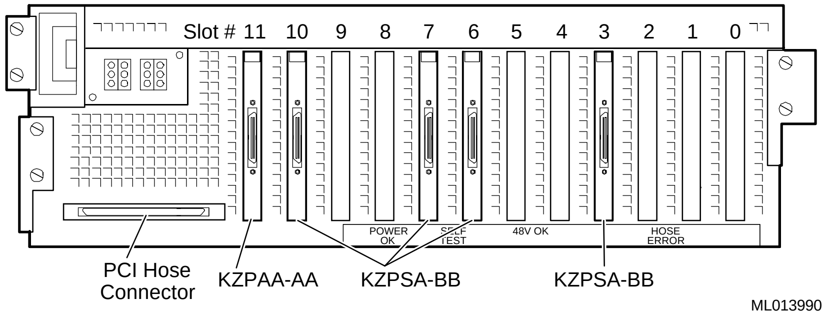


Table A-9: (DEPOT2) PCI #2

SLOT	MODULE/DEVICE	DESCRIPTION
K2J0	Available	
K2J1	Available	
K2J2	Available	
K2J3	KZPSA-BB #9	PCI to SCSI (FWD) Host Bus Adapter
K2J4	Available	
K2J5	Available	
K2J6	KZPSA-BB #10	PCI to SCSI (FWD) Host Bus Adapter
K2J7	KZPSA-BB #11	PCI to SCSI (FWD) Host Bus Adapter
K2J8	Available	
K2J9	Available	
K2J10	KZPSA-BB #12	PCI to SCSI (FWD) Host Bus Adapter
K2J11	KZPAA-AA #0	PCI to SCSI (FNSE) Host Bus Adapter

AlphaServer 8400 (DEPOT2) PCI #3 Device Identification

Figure A-11: (DEPOT2) System Cabinet PCI #3 Module Layout

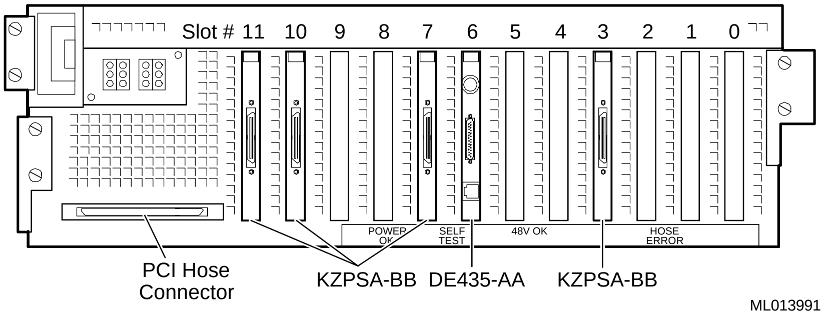


Table A-10: (DEPOT2) PCI #3 Devices

SLOT	MODULE/DEVICE	DESCRIPTION
K3J0	Available	
K3J1	Available	
K3J2	Available	
K3J3	KZPSA-BB #13	PCI to SCSI (FWD) Host Bus Adapter
K3J4	Available	
K3J5	Available	
K3J6	DE435-AA #0	PCI to Ethernet Interface Adapter
K3J7	KZPSA-BB #14	PCI to SCSI (FWD) Host Bus Adapter
K3J8	Available	
K3J9	Available	
K3J10	KZPSA-BB #15	PCI to SCSI (FWD) Host Bus Adapter
K3J11	KZPSA-BB #16	PCI to SCSI (FWD) Host Bus Adapter

AlphaServer 8400 (DEPOT3)

AlphaServer 8400 (DEPOT3) Turbo Laser System Bus (TLSB) Usage

Figure A-12: (DEPOT3) System Cabinet TLSB Centerplane Module Layout

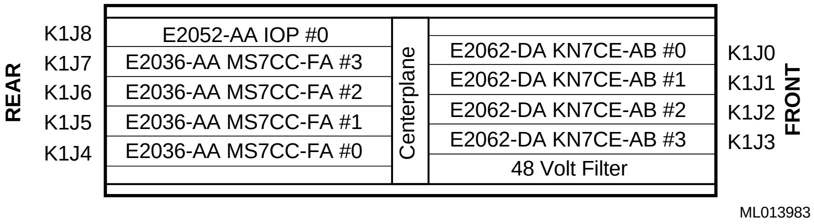


Table A-11: (DEPOT3) TLSB Centerplane Layout

SLOT	MODULE	DEVICE	DESCRIPTION
K1J0	E2062-DA #0	KN7CE-AB	TurboLaser Dual CPU Module
K1J1	E2062-DA #1	KN7CE-AB	TurboLaser Dual CPU Module
K1J2	E2062-DA #2	KN7CE-AB	TurboLaser Dual CPU Module
K1J3	E2062-DA #3	KN7CE-AB	TurboLaser Dual CPU Module
K1J4	E2036-AA #0	MS7CC-FA	2.1GB Memory Module
K1J5	E2036-AA #1	MS7CC-FA	2.1GB Memory Module
K1J6	E2036-AA #2	MS7CC-FA	2.1GB Memory Module
K1J7	E2036-AA #3	MS7CC-FA	2.1GB Memory Module
K1J8	E2052-AA	KFTHA-AA	System I/O Module

AlphaServer 8400 (DEPOT3) PCI #0 Device Identification

Figure A-13: (DEPOT3) System Cabinet PCI #0 Module Layout

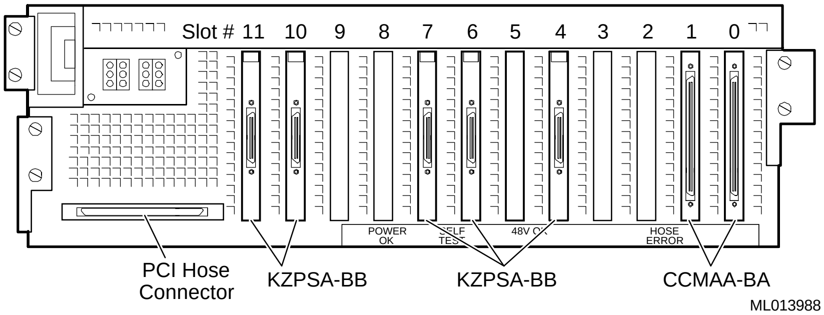


Table A-12: (DEPOT3) PCI #0

SLOT	MODULE/DEVICE	DESCRIPTION
K0J0	CCMAA-BA #0	PCI to MEMORY CHANNEL Host Bus Adapter
K0J1	CCMAA-BA #1	PCI to MEMORY CHANNEL Host Bus Adapter
K0J2	Available	
K0J3	Available	
K0J4	KZPSA-BB #0 (Unused)	PCI to SCSI (FWD) Host Bus Adapter
K0J5	Available	
K0J6	KZPSA-BB #1 (Unused)	PCI to SCSI (FWD) Host Bus Adapter
K0J7	KZPSA-BB #2	PCI to SCSI (FWD) Host Bus Adapter
K0J8	Available	
K0J9	Available	
K0J10	KZPSA-BB #3 (Unused)	PCI to SCSI (FWD) Host Bus Adapter
K0J11	KZPSA-BB #4 (Unused)	PCI to SCSI (FWD) Host Bus Adapter

AlphaServer 8400 (DEPOT3) PCI #1 Device Identification

Figure A-14: (DEPOT3) System Cabinet PCI #1 Module Layout

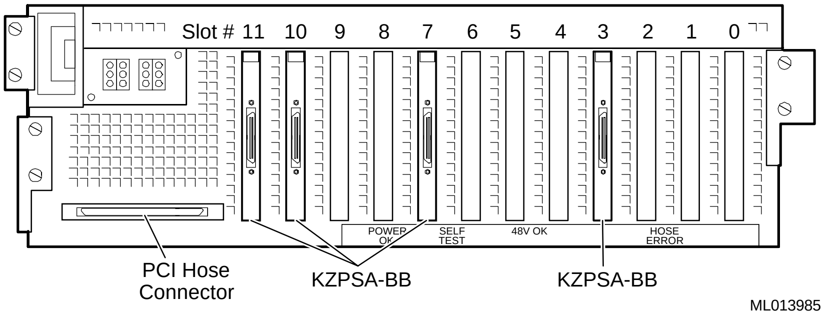


Table A-13: (DEPOT3) PCI #1 Devices

SLOT	MODULE/DEVICE	DESCRIPTION
K1J0	Available	
K1J1	Available	
K1J2	Available	
K1J3	KZPSA-BB #5	PCI to SCSI (FWD) Host Bus Adapter
K1J4	Available	
K1J5	Available	
K1J6	Available	
K1J7	KZPSA-BB #6	PCI to SCSI (FWD) Host Bus Adapter
K1J8	Available	
K1J9	Available	
K1J10	KZPSA-BB #7	PCI to SCSI (FWD) Host Bus Adapter
K1J11	KZPSA-BB #8 (Unused)	PCI to SCSI (FWD) Host Bus Adapter

AlphaServer 8400 (DEPOT3) PCI #2 Device Identification

Figure A-15: (DEPOT3) System Cabinet PCI #2 Module Layout

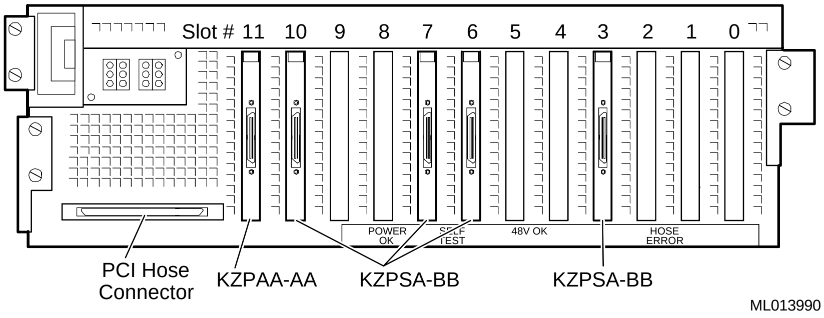


Table A-14: (DEPOT3) PCI #2

SLOT	MODULE/DEVICE	DESCRIPTION
K2J0	Available	
K2J1	Available	
K2J2	Available	
K2J3	KZPSA-BB #9	PCI to SCSI (FWD) Host Bus Adapter
K2J4	Available	
K2J5	Available	
K2J6	KZPSA-BB #10	CI to SCSI (FWD) Host Bus Adapter
K2J7	KZPSA-BB #11 (Unused)	PCI to SCSI (FWD) Host Bus Adapter
K2J8	Available	
K2J9	Available	
K2J10	KZPSA-BB #12	PCI to SCSI (FWD) Host Bus Adapter
K2J11	KZPAA-AA #0	PCI to SCSI (FNSE) Host Bus Adapter

AlphaServer 8400 (DEPOT3) PCI #3 Device Identification

Figure A-16: (DEPOT3) System Cabinet PCI #3 Module Layout

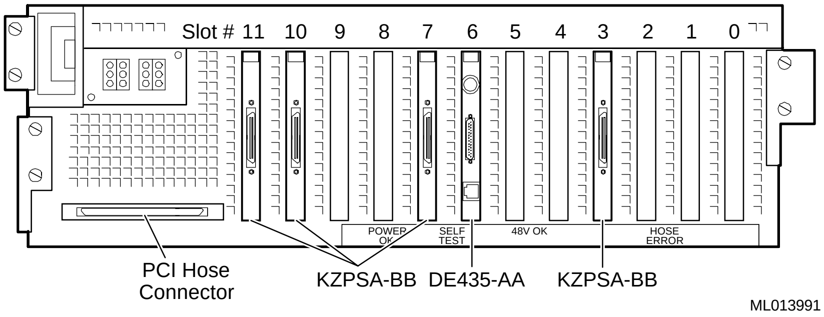


Table A-15: (DEPOT3) PCI #3

SLOT	MODULE/DEVICE	DESCRIPTION
K3J0	Available	
K3J1	Available	
K3J2	Available	
K3J3	KZPSA-BB #13	PCI to SCSI (FWD) Host Bus Adapter
K3J4	Available	
K3J5	Available	
K3J6	DE435-AA #0	PCI to Ethernet Interface Adapter
K3J7	KZPSA-BB #14	PCI to SCSI (FWD) Host Bus Adapter
K3J8	Available	
K3J9	Available	
K3J10	KZPSA-BB #15	PCI to SCSI (FWD) Host Bus Adapter
K3J11	KZPSA-BB #16	PCI to SCSI (FWD) Host Bus Adapter

Memory Channel

Memory Channel Hub

Table A-16: Memory Channel Slot Utilization

SLOT	MODULE/DEVICE	DESCRIPTION
S0	CCMLA-AA	MEMORY CHANNEL Line Card
S1	Available	
S2	CCMLA-AA	MEMORY CHANNEL Line Card
S3	Available	
S4	CCMLA-AA	MEMORY CHANNEL Line Card
S5	Available	
S6	CCMLA-AA	MEMORY CHANNEL Line Card
S7	Available	

Memory Channel Connections

Figure A-17: Memory Channel Connections to DEPOT1, DEPOT2 & DEPOT3

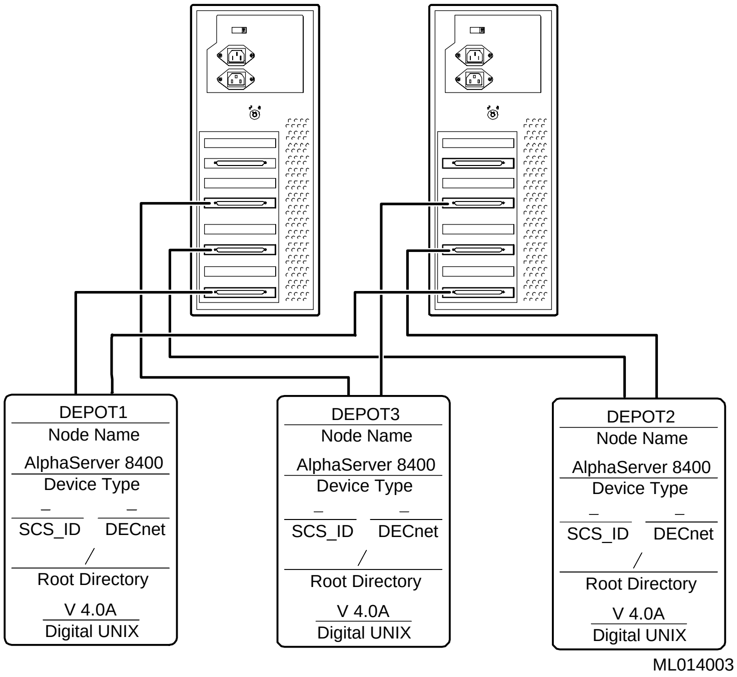


Table A-17: Memory Channel Connections

Source	Destination	Part Number	Length
CCMHA #1/CCMLA #1	DEPOT1/CCMAA #0	BC12N-10	10 ft.
CCMHA #1/CCMLA #2	DEPOT2/CCMAA #0	BC12N-10	10 ft.
CCMHA #1/CCMLA #3	DEPOT3/CCMAA #0	BC12N-10	10 ft.
CCMHA #2/CCMLA #1	DEPOT1/CCMAA #1	BC12N-10	10 ft.
CCMHA #2/CCMLA #2	DEPOT2/CCMAA #1	BC12N-10	10 ft.
CCMHA #2/CCMLA #3	DEPOT3/CCMAA #1	BC12N-10	10 ft.

Cable Layouts

LAN Network Interconnect

Figure A-18: ThinWire Backbone Connections

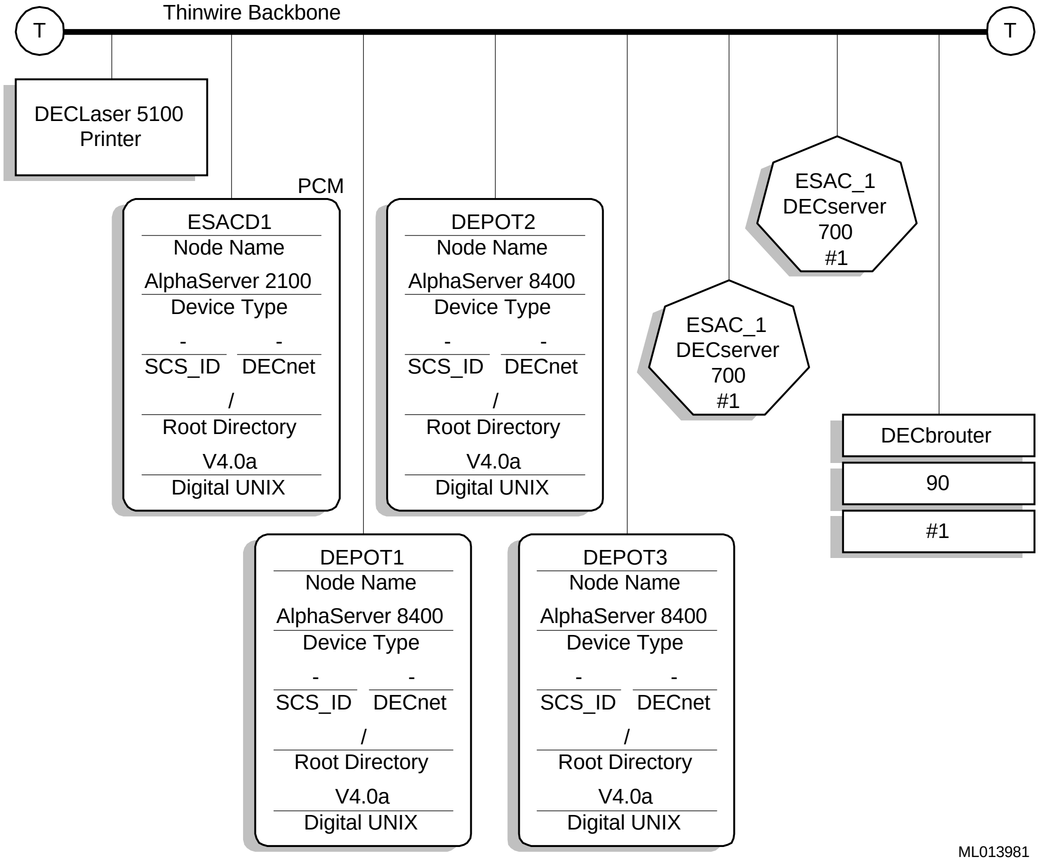


Table A-18: Thinwire Backbone Connections

Source	Destination	Node Name	PART #	LENGTH
DEClaser 5100	AlphaServer 2100	ESACD1	BC16M-XX	XX ft.
AlphaServer 2100	AlphaServer 8400#1	DEPOT1	BC16M-XX	XX ft.
AlphaServer 8400#1	AlphaServer 8400#2	DEPOT2	BC16M-XX	XX ft.
AlphaServer 8400#2	AlphaServer 8400#3	DEPOT3	BC16M-XX	XX ft.
AlphaServer 8400#3	DECserver 700 #1	ESAC_1	BC16M-XX	XX ft.
DECserver 700 #1	DECserver 700 #2	ESAC_2	BC16M-XX	XX ft.
DECserver 700 #2	DECbrouter 90 #1		BC16M-XX	XX ft.

WAN Network Interconnect

Figure A-19: WAN/Remote Connections

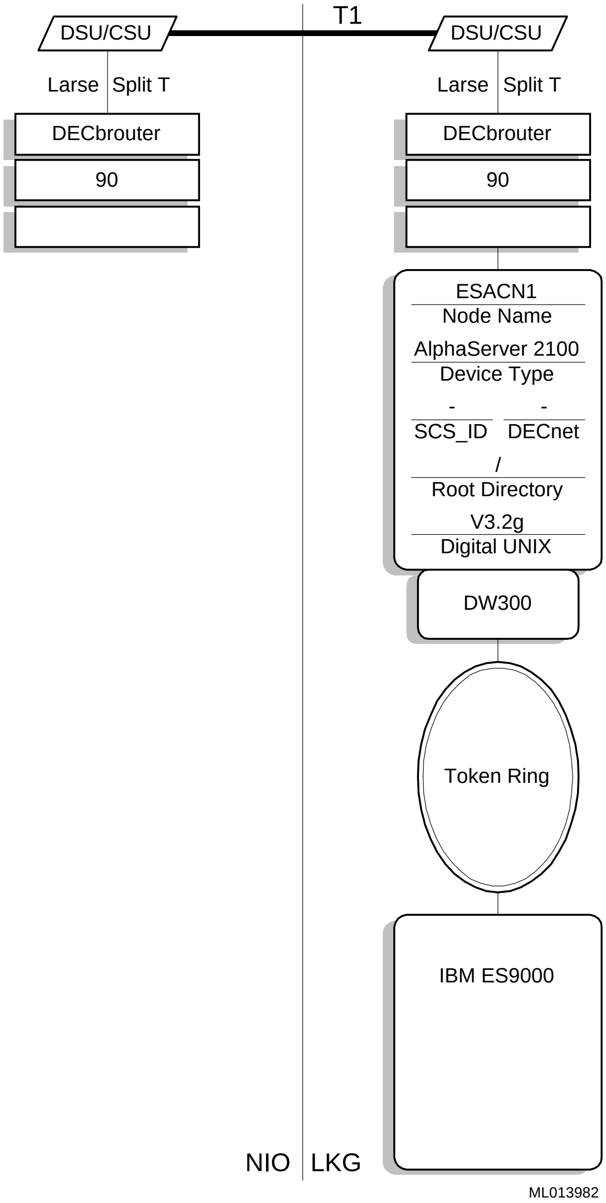


Table A-19: WAN/Remote Connections

Source	Destination	PART #	LENGTH
DECbrouter 90 #1 (NIO)	Larse DSC/CSU (NIO)	BC12G-08	8 ft.
Larse DSC/CSU (NIO)	Larse DSC/CSU (LKG)	T1	XX ft.
Larse DSC/CSU (LKG)	DECbrouter 90 #2 (LKG)	BC12G-08	8 ft.
DECbrouter 90 #2 (LKG)	AlphaServer 2100	BC16M-XX	XX ft.
AlphaServer 2100 DW300 #1	IBM ES9000	Token	XX ft.

DSRVW #1 (ESAC_1) Connections

Table A-20: DSRVW #1 (ESAC_1) Connections

	Source	Destination	Node Name	Part Number #	LENGTH
Port 1	TTY08	HSZ40 #5	HSZ_RZ48	BC16E-XX	XX ft
Port 2	TTY02	DEPOT #1	DEPOT #1	BC16E-XX	XX ft
Port 3	TTY03	HSZ40 #11	HSZ_RZ120	BC16E-XX	XX ft
Port 4	TTY04	HSZ40 #3	HSZ_RZ16	BC16E-XX	XX ft
Port 5	TTY05	HSZ40 #2	HSZ_RZ8	BC16E-XX	XX ft
Port 6	Available				
Port 7	TTY07	HSZ40 #1	HSZ_RZ0	BC16E-XX	XX ft
Port 8	Available				
Port 9	TTY0K	DEPOT #2	DEPOT #2	BC16E-XX	XX ft
Port 10	Available				
Port 11	Available				
Port 12	Available				
Port 13	Available				
Port 14	TTY0P	DEPOT #3	DEPOT #3	BC16E-XX	XX ft
Port 15	Available				
Port 16	Available				

DSRVW #2 (ESAC_2) Connections

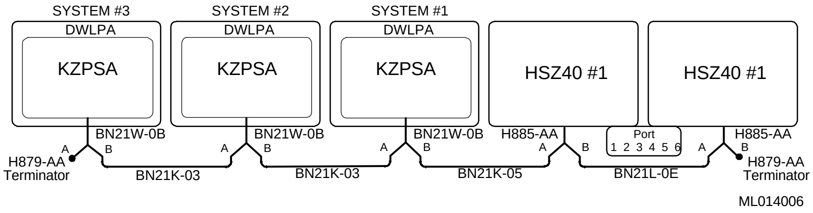
Table A-21: DSRVW #2 (ESAC_2) Connections

	Source	Destinatio n	Node Name	Part Number	Length
Port 1	Available				
Port 2	TTY09	HSZ40 #4	HSZ_RZ40	BC16E-XX	XX ft
Port 3	TTY0A	HSZ40 #6	HSZ_RZ56	BC16E-XX	XX ft
Port 4	Available				
Port 5	Available				
Port 6	TTY0D	HSZ40 #12	HSZ_RZ128	BC16E-XX	XX ft
Port 7	TTY0E	HSZ40 #10	HSZ_RZ112	BC16E-XX	XX ft
Port 8	TTY0F	HSZ40 #9	HSZ_RZ96	BC16E-XX	XX ft
Port 9	TTY0G	HSZ40 #7	HSZ_RZ72	BC16E-XX	XX ft
Port 10	TTY0H	HSZ40 #8	HSZ_RZ80	BC16E-XX	XX ft
Port 11	Available				
Port 12	Available				
Port 13	Available				
Port 14	Available				
Port 15	Available				
Port 16	Available				

SCSI Disk/Tape Interconnect

Cluster KZPSA to HSZ Connection Example

Figure A-20: Cluster KZPSA to HSZ Connection Example



8400 (DEPOT1) to HSZ40 SCSI Connections

Figure A-21: 8400 (DEPOT1) to HSZ40 SCSI Device Cable Connections

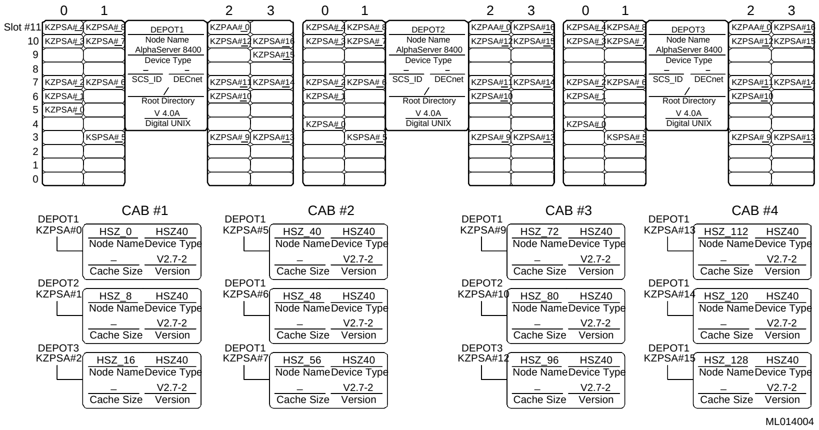
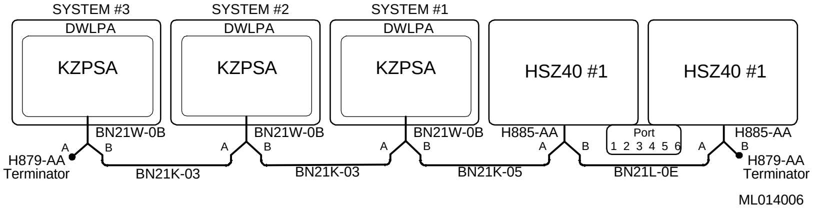


Table A-22: (DEPOT1) KZPSA to HSZ40 Connections

Source	Destination	Node Name	Part Number	Length
8400 #1 KZPSA#0	HSZ40 #0	HSZ_RZ0	BN21K-05	5 m
8400 #1 KZPSA#5	HSZ40 #3	HSZ_RZ40	BN21W-0B/BN21K-05	.15/5 m
8400 #1 KZPSA#6	HSZ40 #4	HSZ_RZ48	BN21W-0B/BN21K-05	.15/5 m
8400 #1 KZPSA#7	HSZ40 #5	HSZ_RZ56	BN21W-0B/BN21K-05	.15/5 m
8400 #1 KZPSA#9	HSZ40 #6	HSZ_RZ72	BN21W-0B/BN21K-05	.15/5 m
8400 #1 KZPSA#10	HSZ40 #7	HSZ_RZ80	BN21W-0B/BN21K-05	.15/5 m
8400 #1 KZPSA#12	HSZ40 #8	HSZ_RZ96	BN21W-0B/BN21K-05	.15/5 m
8400 #1 KZPSA#13	HSZ40 #9	HSZ_RZ112	BN21W-0B/BN21K-05	.15/5 m
8400 #1 KZPSA#14	HSZ40 #10	HSZ_RZ120	BN21W-0B/BN21K-05	.15/5 m
8400 #1 KZPSA#15	HSZ40 #11	HSZ_RZ128	BN21W-0B/BN21K-05	.15/5 m

Cluster KZPSA to HSZ Connection Example

Figure A-22: Cluster KZPSA to HSZ Connection Example



8400 (DEPOT2) KZPSA Connections

Figure A-23: 8400 (DEPOT2) KZPSA Cable Connections

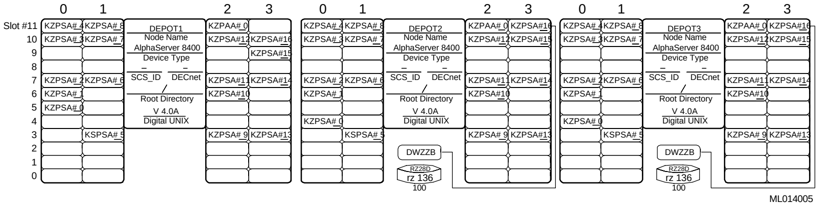


Table A-23: (DEPOT2) KZPSA Connections

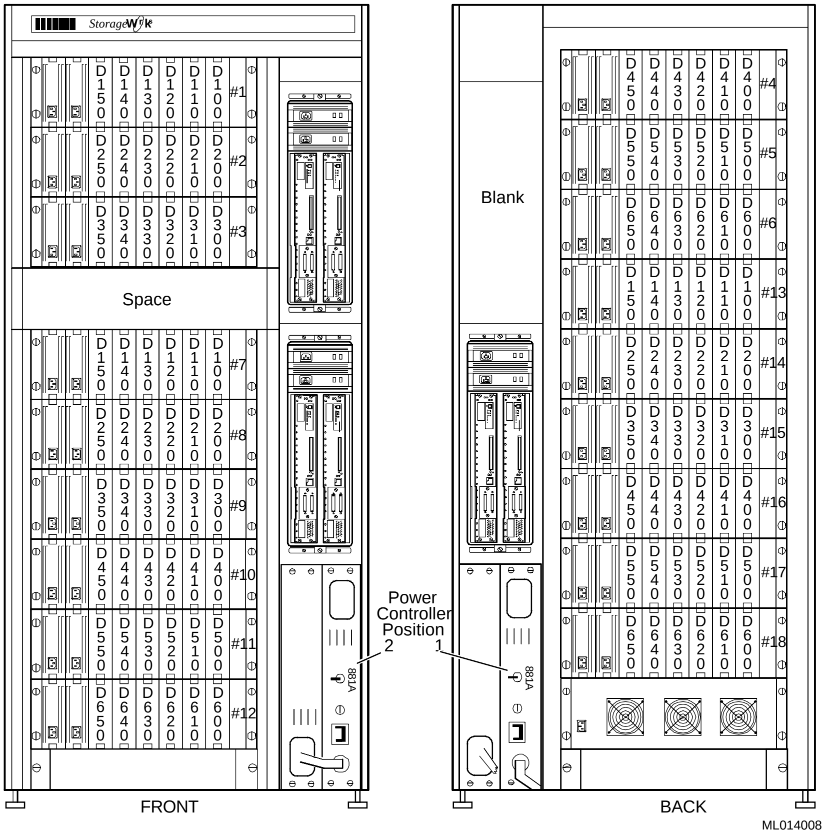
Source	Destination	Node Name	Part Number	Length
8400 #2 KZPSA#1	HSZ40 #1	HSZ_RZ8	BN21K-05	5 m
8400 #2 KZPSA#4	HSZ40	HSZ_RZ32	BN21K-10	10 m
8400 #2 KZPSA#5	8400 #1 KZPSA#5	DEPOT1	BN21W-0B/BN21K-03/ BN21W-0B	.15/3/.15 m
8400 #2 KZPSA#6	8400 #1 KZPSA#6	DEPOT1	BN21W-0B/BN21K-03/ BN21W-0B	.15/3/.15 m
8400 #2 KZPSA#7	8400 #1 KZPSA#7	DEPOT1	BN21W-0B/BN21K-03/ BN21W-0B	.15/3/.15 m
8400 #2 KZPSA#8	HSZ40	HSZ_RZ64	BN21K-10	10 m
8400 #2 KZPSA#9	8400 #1 KZPSA#9	DEPOT1	BN21W-0B/BN21K-03/ BN21W-0B	.15/3/.15 m
8400 #2 KZPSA#10	8400 #1 KZPSA#10	DEPOT1	BN21W-0B/BN21K-03/ BN21W-0B	.15/3/.15 m
8400 #2 KZPSA#11	HSZ40	HSZ_RZ88	BN21K-10	10 m
8400 #2 KZPSA#12	8400 #1 KZPSA#12	DEPOT1	BN21W-0B/BN21K-03/ BN21W-0B	.15/3/.15 m
8400 #2 KZPSA#13	8400 #1 KZPSA#13	DEPOT1	BN21W-0B/BN21K-03/ BN21W-0B	.15/3/.15 m
8400 #2 KZPSA#14	8400 #1 KZPSA#14	DEPOT1	BN21W-0B/BN21K-03/ BN21W-0B	.15/3/.15 m
8400 #2 KZPSA#15	8400 #1 KZPSA#15	DEPOT1	BN21W-0B/BN21K-03/ BN21W-0B	.15/3/.15 m
8400 #2 KZPSA#16	BA356-LB #1	DEPOT2	BN21K-03	3 m

Table A-24: (DEPOT3) KZPSA Connections

Source	Destination	Node Name	Part Number	Length
8400 #3 KZPSA#2	HSZ40 #2	HSZ_RZ16	BN21K-05	5 m
8400 #3 KZPSA#5	8400 #2 KZPSA#5	DEPOT2	BN21W-0B/BN21K-03 /BN21W-0B	.15/3/.15 m
8400 #3 KZPSA#6	8400 #2 KZPSA#6	DEPOT2	BN21W-0B/BN21K-03 /BN21W-0B	.15/3/.15 m
8400 #3 KZPSA#7	8400 #2 KZPSA#7	DEPOT2	BN21W-0B/BN21K-03 /BN21W-0B	.15/3/.15 m
8400 #3 KZPSA#9	8400 #2 KZPSA#9	DEPOT2	BN21W-0B/BN21K-03 /BN21W-0B	.15/3/.15 m
8400 #3 KZPSA#10	8400 #2 KZPSA#10	DEPOT2	BN21W-0B/BN21K-03 /BN21W-0B	.15/3/.15 m
8400 #3 KZPSA#12	8400 #2 KZPSA#12	DEPOT2	BN21W-0B/BN21K-03 /BN21W-0B	.15/3/.15 m
8400 #3 KZPSA#13	8400 #2 KZPSA#13	DEPOT2	BN21W-0B/BN21K-03 /BN21W-0B	.15/3/.15 m
8400 #3 KZPSA#14	8400 #2 KZPSA#14	DEPOT2	BN21W-0B/BN21K-03 /BN21W-0B	.15/3/.15 m
8400 #3 KZPSA#15	8400 #2 KZPSA#15	DEPOT2	BN21W-0B/BN21K-03 /BN21W-0B	.15/3/.15 m
8400 #3 KZPSA#16	BA356-LB #1	DEPOT3	BN21K-03	3 m

VLM64 Storage Cabinet SCSI Disks Unit Address Identification

Figure A-24: SW800 Cabinet #1 Unit Address Identification


StorageWorks
D150 D140 D130 D120 D110 D100 #1
D250 D240 D230 D220 D210 D200 #2
D350 D340 D330 D320 D310 D300 #3
Space
D150 D140 D130 D120 D110 D100 #7
D250 D240 D230 D220 D210 D200 #8
D350 D340 D330 D320 D310 D300 #9
D450 D440 D430 D420 D410 D400 #10
D550 D540 D530 D520 D510 D500 #11
D650 D640 D630 D620 D610 D600 #12
881A
FRONT
Power
Controller
Position
2 1
Blank
D450 D440 D430 D420 D410 D400 #4
D550 D540 D530 D520 D510 D500 #5
D650 D640 D630 D620 D610 D600 #6
D150 D140 D130 D120 D110 D100 #13
D250 D240 D230 D220 D210 D200 #14
D350 D340 D330 D320 D310 D300 #15
D450 D440 D430 D420 D410 D400 #16
D550 D540 D530 D520 D510 D500 #17
D650 D640 D630 D620 D610 D600 #18
881A
BACK
ML014008

HSZ40 SCSI Cable Tables

The following table defines the point to point **SCSI** cable connections to be used between the HSZ40s' port and the BA350 Modular Storage shelves. All SCSI Ports that are not used are labeled **"Available",** and all SCSI ports without modules are labeled **"Empty".** All SW800 cabinets have the same connections

Table A-25: HSZ40 #1 (Top Front) Connections

Source	Destination	Part Number	Length
Port 1	BA356 #1 (Front)	BN21H-01	1 m
Port 2	BA356 #2 (Front)	BN21H-01	1 m
Port 3	BA356 #3 (Front)	BN21H-01	1 m
Port 4	BA356 #4 (Rear)	BN21H-02	2 m
Port 5	BA356 #5 (Rear)	BN21H-02	2 m
Port 6	BA356 #6 (Rear)	BN21H-02	2 m

Table A-26: HSZ40 #2 (Bottom Front) Connections

Source	Destination	Part Number	Length
Port 1	BA356 #7 (Front)	BN21H-01	1 m
Port 2	BA356 #8 (Front)	BN21H-01	1 m
Port 3	BA356 #9 (Front)	BN21H-01	1 m
Port 4	BA356 #10 (Front)	BN21H-01	1 m
Port 5	BA356 #11 (Front)	BN21H-01	1 m
Port 6	BA356 #12 (Front)	BN21H-01	1 m

Table A-27: HSZ40 #3 (Back) Connections

Source	Destination	Part Number	Length
Port 1	BA356 #13 (Rear)	BN21H-02	2 m
Port 2	BA356 #14 (Rear)	BN21H-02	2 m
Port 3	BA356 #15 (Rear)	BN21H-02	2 m
Port 4	BA356 #16 (Rear)	BN21H-02	2 m
Port 5	BA356 #17 (Rear)	BN21H-02	2 m
Port 6	BA356 #18 (Rear)	BN21H-02	2 m

Note

All HSZ40 are cascaded via BN21L-0B cables and H885 Tri-link connectors, giving each cabinet a total of six (6) HSZ40's.

System to Tape SCSI Connections

Figure A-25: System to Tape SCSI Device Cable Connections

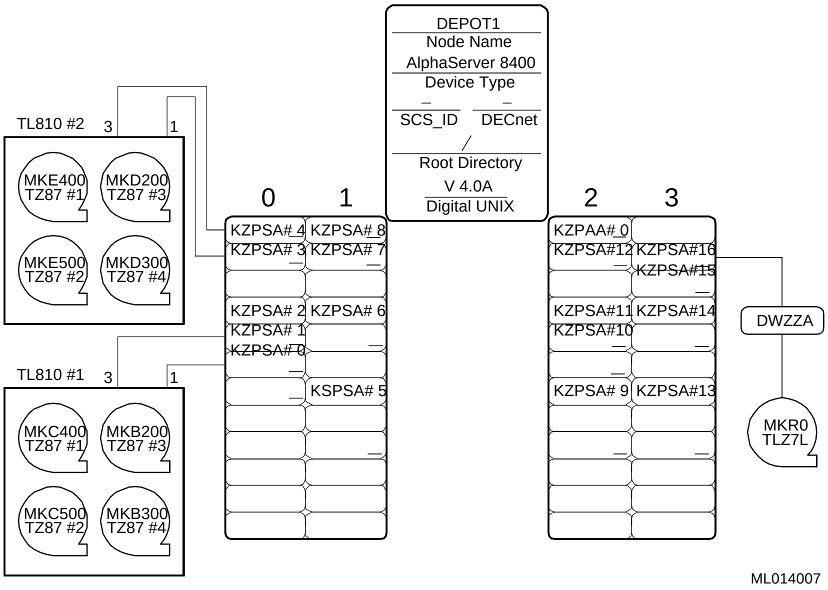


Table A-28: KZPSA to TL810 Connections

Source	Destination	Part Number	Length
8400 #1 KZPSA#1	TL810 #1 Port #1	BN21K-10	10 m
8400 #1 KZPSA#2	TL810 #1 Port #3	BN21K-10	10 m
8400 #1 KZPSA#3	TL810 #2 Port #1	BN21K-10	10 m
8400 #1 KZPSA#4	TL810 #2 Port #3	BN21K-10	10 m

Table A-29: KZPSA to TLZ07 Connections

Source	Destination	Part Number	Length
8400 #1 KZPSA#16	DWZZA #1	BN21K-05	05 m
DWZZA#1	TLZ7L#1	BC19J-1E	1.5 ft.