SAS Oracle
DIGITAL UNIX AlphaServer 4100

DIGITAL HiTest Notes

Part Number: EK-HSSUA-HN. B01

January 1998

Revision/Update Information: This is a revised manual.

Digital Equipment Corporation
Maynard, Massachusetts

January 1998

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.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from DIGITAL or an authorized sublicensor.

Table of Contents

1 Introduction

2 Configuration Data

3 System Installation and Setup

4 Interoperability Tests and Results

5 System Limits and Characterization Data

6 Problems and Resolutions

A Detailed Hardware Configuration

B Test Scripts

Figures

Tables

Preface

This document provides an overview of DIGITAL HiTest Suites and detailed technical information about interoperability test results for the SAS Oracle DIGITAL UNIX AlphaServer 4100 HiTest Suite.

Audience

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

Road Map

This document contains the following chapters:

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 – Includes tables of configuration data about the hardware and software components that define the Template, and special configuration rules if any.

3. System Installation and Setup – Presents useful information for installing and setting up this DIGITAL HiTest Suite.

4. Interoperability Tests and Results – Describes how the 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 Characterization Data – Summarizes any system limitations or characterization data that were identified during testing.

6. Problems and Resolutions – Discusses any problems and resolutions that were discovered during testing.

Appendix A: Detailed Hardware Configuration – Contains more detailed information about the hardware and software components listed in the Configuration Data chapter.

Appendix B: Test Scripts – Contains detailed test scripts used to build Oracle databases, perform test queries, and execute database tests.

Feedback and Ordering Information

What our readers think of this or any other DIGITAL documentation is important to us. If you have any comments, we would appreciate hearing from you. Send your comments to: *reader-comments@digital.com*.

Please reference the document title and part number (EK-HSSUA-HN. B01) in your correspondence about this document.

Copies of this and other DIGITAL documents can be ordered by calling 1-800-DIGITAL.

1 Introduction

Digital HiTest Suite and Its Advantages

DIGITAL HiTest Suites are guidelines for configuring a set of prequalified 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 .

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 foundation can be used as a base on which any customer-desired applications can be installed. The HiTest AppSet includes the software specific to one class of customer solutions.

Configuring a DIGITAL HiTest Suite is straightforward. 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 chapter 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 chapter 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/cgi-bin/comet`	(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-based 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 This DIGITAL HiTest Suite

The SAS Oracle DIGITAL UNIX AlphaServer 4100 HiTest Suite includes the following software components:

- SAS System software
- Oracle7 Database Software
- DIGITAL UNIX Operating system software

The Suite will meet the needs of customers who want to implement Data Warehousing solutions using databases ranging in size from 24 GB to 140 GB for loading data files, summarizing detail data, building data marts, and creating multi-dimensional summary objects.

2
Configuration Data

This chapter describes the tested DIGITAL HiTest Configuration Suite including the hardware, software, and firmware components, and their revision levels. Special configuration rules are explained if required.

Hardware and Software Components

Table 2-1 identifies the range of hardware and software components that can be configured using the SAS Oracle DIGITAL UNIX AlphaServer 4100 HiTest Suite.

Table 2-2 lists the System Management Station hardware and software.

Table 2-3 lists the revision levels of the components.

The HiTest Template (Table 2-1) consists of three categories:

- **AppSet Software** – Includes software specific to one class of customer solutions, in this case the Data Warehousing market.
- **Foundation Hardware** – Includes the base system, storage, and other hardware options.
- **Foundation Software** – Includes the operating system, middleware, and database software.

When ordering an item from a HiTest Template, select a quantity that is within the minimum/maximum range for the item. If the minimum quantity is zero (0), then the component is optional. If the minimum quantity is one or more, then order at least the minimum quantity, but not more than the maximum quantity. The maximum quantity represents the greatest number of components that were tested for interoperability with all the other components in the Suite.

Storage can be implemented through one of two storage options:

Option 1 - External StorageWorks cabinet with RAID controllers (Scales to 140 GB)
Option 2 - Internal UltraSCSI RAID (Scales to 14 disks)

For more details on the HiTest Suite hardware configurations tested, see Appendix A.

Table 2-1: SAS Oracle DIGITAL UNIX AlphaServer 4x00 HiTest Template

SAS Oracle *HiTest AppSet* **DIGITAL UNIX AlphaServer 4100 HiTest Foundation**				
For documentation and updates: http://cosmo.tay.dec.com and http://www.partner.digital.com:9003/cgi-bin/comet For hardcopy of this Suite’s HiTest Notes, order EK-HSSUA-HN.				
Line Item	**Description**	**Part Number**	**HiTest Range**	
			Min	**Max**
AppSet Software				
1	**The SAS System, Release 6.12** For SAS Institute products, call 919-677-8000. See Line Item 35 for detailed license requirements.	**SAS Institute**	1	1
Foundation Hardware				
2	*Select one system drawer:* **AlphaServer 4100 5/466, 1 GB, DIGITAL UNIX** **AlphaServer 4100 5/466, 2 GB, DIGITAL UNIX** *Hardware includes:* • 5/466 MHz CPU with 4 MB cache • 1 GB Memory option • PB2GA-JB; S3 TRIO64 1 MB Graphics adapter • DE500-AA 10/100 Mbit Fast Ethernet adapter • KZPDA-AA; FWSE SCSI adapter and SCSI cable • Integral FNSE (internal only SCSI bus) • Integral CD-ROM drive • Integral 1.44 MB Diskette drive • 450 Watt auto-sensing power supply **Note:** Americas and Asia Pacific orders include power cords, keyboard, and mouse as follows: • LK47W-A2 (U.S./English) keyboard • Three-button mouse For European orders, select country-specific keyboard and power cords. *Software Includes:* • DIGITAL UNIX Operating System (included with Foundation Hardware)	 **DA-51JAC-FB** **DA-51JAB-GB**	1	1
3	*Select the enclosure:* **Pedestal enclosure** *Hardware includes:* • One 4.3 GB Hard Disk Drive • One BA36R-SB StorageWorks shelf; the pedestal supports up to two more.	 **BA30P-AB**	 1	 1
4	**466 MHz CPU DIGITAL UNIX SMP UPG** **Note:** Up to three additional CPUs may be added to the 4100 System.	**KN304-DB**	0	3
5	**450 Watt Power Supply** **Note:** The power supply included with the system drawer supports two CPUs, memory, and I/O adapters. Add a second supply to support four CPUs or for redundancy (N+1) with a two CPU system. Add a third supply for redundancy (N+1) with a four CPU system (AlphaServer 4100).	**H7291-AA**	0	2

SAS Oracle *HiTest AppSet*
DIGITAL UNIX AlphaServer 4100 HiTest Foundation

For documentation and updates: http://cosmo.tay.dec.com and
http://www.partner.digital.com:9003/cgi-bin/comet
For hardcopy of this Suite's HiTest Notes, order EK-HSSUA-HN.

Line Item	Description	Part Number	HiTest Range	
			Min	Max
6	**1 GB Memory Option** **2 GB Memory Option** **Note:** AlphaServer 4100 system drawer supports three additional memory options. The maximum memory for this HiTest Suite is 4 GB. Add 1 GB of memory for each CPU added.	**MS330-FA** **MS330-GA**	0	See Note
7	**PCI-FDDI Adapter SAS, MMF**	**DEFPA-AB**	1	1
8	**20 meter SC-to-SC dual fiber optic cable**	**BN34B-20**	1	1
9	**7 meter 10BaseT Twisted Pair Ethernet cable**	**BN25G-07**	1	
Storage Option 1: External StorageWorks Cabinet with RAID Controllers (Scales to 108 Disks) **(Hardware Items 10, 11, 12, and 13, and Software Items 26 and 27)**				
10	**PCI one-port FWD SCSI Controller**	**KZPSA-BB**	3	3
11	**20 meter 16-bit SCSI cable (external)** **Note:** Order one for each KZPSA-BB ordered. Cable used to interconnect KZPSA-BB SCSI controller with an HSZ52-AJ StorageWorks Raid Array controller.	**BN21K-20**	3	3
Note: Customer should order **"**Configure to Order**"** and list SW822 part number (Item 12) first, followed with Optional StorageWorks Devices (Controllers and Disk Drives, Items 13 and 17) to be included; otherwise, all hardware will arrive in separate boxes.				
12	**StorageWorks Storage Array** *Hardware includes:* • SW800-GA Cabinet Assembly with Power Controller • SW8XP-BA (Backup) Power Controller • 3 BA350-MB Disk Controller Shelves • 3 BA35X-HF (Backup) Power Supplies for Disk Controller Shelves • 18 BA350-JA Disk Shelves • 18 BA35X-HA (Backup) Power Supplies for Disk Shelves	**SW822-GA**	1	1
13	**StorageWorks RAID Array Dual SCSI Controller** *Each HSZ52-AJ Controller Option includes:* • 2 HSZ50 256 MB SCSI Controllers • An External Cache Battery Backup Power Supply which is installed in place of a (Backup) power supply in the associated disk storage shelves • 2 2-meter cables to interconnect Controllers with external cache batteries • 2 H885-AA SCSI Tri-link Connectors, H879-AA SCSI Terminator, and a BN21L-0B cable to interconnect the 2 SCSI Controllers	**HSZ52-AJ**	3	3
Storage Option 2: Internal RAID (Scales to 14 Disks) **(Hardware Items 14, 15, 16, and 17)**				
14	**StorageWorks UltraSCSI Shelf** **Note:** Testing performed using UltraSCSI RAID controller, external StorageWorks BA356-SB enclosures with RZ29B-VW disk drives. This configuration uses less than full UltraSCSI speed. Special configuration rules explain setup of the UltraSCSI configuration to run at the slower speed.	**BA36R-SD**	2	2

SAS Oracle *HiTest AppSet*

DIGITAL UNIX AlphaServer 4100 HiTest Foundation

For documentation and updates: http://cosmo.tay.dec.com and
http://www.partner.digital.com:9003/cgi-bin/comet
For hardcopy of this Suite's HiTest Notes, order EK-HSSUA-HN.

Line Item	Description	Part Number	HiTest Range	
			Min	Max
15	**PCI three-port RAID controller with 4 MB cache**	**KZPAC-CA**	1	1
16	**1.5 meter VHDC male to VHDC male UltraSCSI cable** **Note:** BN31S-1E cables were used during test to interconnect the KZPAC-CA ports with each BA356-SB StorageWorks enclosure.	**BN37A-1E**	2	2
17	**4.3 GB 7200 RPM UltraSCSI Disk** **Note:** This part number replaces RZ29B-VW, which was used for testing this HiTest Suite.	**DS-RZ1CB-VW**	14	108
	Foundation Software			
18	**DIGITAL UNIX, Version 4.0B**	**Included with item 2**	1	1
19	**ServerWORKS**	**Included with item 2**	1	1
20	**DIGITAL UNIX Alpha Software product library CD-ROM**	**QA-054AA-H8**	1	1
21	**DIGITAL UNIX Associated Products CD-ROM**	**QA-MT4AA-H8**	1	1
22	**DEC C++ for DIGITAL UNIX Systems, V5.5**	**QL-MTRAQ-AA**	0	1
23	**DEC COBOL for DIGITAL UNIX Systems, V2.4**	**QL-2BZAQ-AA**	0	1
24	**Micro Focus COBOL, V4.0A for DIGITAL UNIX**	**QP-5S9AM-3B**	0	1
25	**Developers' Toolkit for DIGITAL UNIX, V4.0C**	**QL-MT5AQ-AA**	0	1
26	**HSZ50 Array Controller Operating Software (HSOF), V5.1 Platform Software Kit** *Includes:* StorageWorks Solutions Command Console SW V1.1 HSZ50 Array Controller Operating Software HSOF V5.1 **Note:** For use with Storage Option 1.	**QB-5JCAB-SA**	0	1
27	**HSZ50 Array Controller Operating Software (HSOF), V5.1** **Note:** For use with Storage Option 1. A separate copy of the HSOF Software must be installed in each HSZ50 controller in the storage subsystem. Order additional HSOF Software modules as required.	**QB-5JCAB-SB**	0	5
28	**Oracle7 Server for DIGITAL UNIX, V7.3.3** *Including:* Oracle7 Server (RBMS) Oracle Common Libraries and Utilities Oracle Parallel Query Oracle Trace Oracle contacts: Phone: 1-800-ORACLE1, E-mail: infodec@us.oracle.com URL `http://www.oracle.com`	**Oracle**	1	1
29	**Oracle Server Manager, V2.3.3**	**Oracle**	1	1
30	**Oracle PL/SQL, V2.3.3**	**Oracle**	1	1
31	**Oracle SQL*Net , V2.3.3**	**Oracle**	1	1
32	**Oracle SQL*Plus, V3.3.3**	**Oracle**	1	1

SAS Oracle *HiTest AppSet*
DIGITAL UNIX AlphaServer 4100 HiTest Foundation

For documentation and updates: http://cosmo.tay.dec.com and
http://www.partner.digital.com:9003/cgi-bin/comet
For hardcopy of this Suite's HiTest Notes, order EK-HSSUA-HN.

Line Item	Description	Part Number	HiTest Range	
			Min	Max
33	**Oracle Protocol Adapter for TCP/IP (V2), V2.3.3**	**Oracle**	1	1
34	**Oracle Remote Operations Software, V1.3.3**	**Oracle**	1	1
	AppSet Software Detail			
35	**SAS Institute Layered Products** **The SAS System, Release 6.12** *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	**SAS Institute**	1	1

Table 2-2: System Management Station Template

SAS Oracle *HiTest AppSet*
System Management Station

Line Item	Description	Part Number	HiTest Range	
			Min	Max
System Management Station Hardware				
Note: This HiTest Suite supports the use of a system management station. When the management station option is included, this HiTest Template identifies the items required. When system management is to be provided through other means, this option may be omitted without invalidating the HiTest Suite.				
1	**DIGITAL PWS Personal Workstation for Windows NT** *Hardware includes:* 200 MHz CPU with 256 Kbyte cache, 64 MB memory, Matrox Millenium 3D graphics, 10BaseT/10Base2 Ethernet, 2.0 GB UW disk, EIDE CD-ROM, 1.44 MB floppy, PS/2 style keyboard, two-button PS/2 compatible mouse) **Note:** A functionally equivalent X86 system may be substituted without invalidating this HiTest Template.	**SN-B3KAP-EL**	1	1
2	**21” Monitor** **Note:** The 17” (SN-VRCX7-WA) or 15” (SN-VRCX5-WA) monitor may be substituted if desired.	**SN-VRCX1-WA**	1	1
System Management Station Software				
3	**Windows NT V4.0**	**Included with item 1**	1	1
4	**Windows NT Service Pack 3**	**Microsoft**	1	1
5	**Exceed V5.1.3**	**Hummingbird**	1	1
6	**ServerWORKS Manager V2.2A (QB-4QYAA-SA)**	**Included with the AlphaServer**	1	1
7	**StorageWorks Solution Command Console V1.1 (QB-5GZAA-SA)**	**Included with StorageWorks Platform Kit**	1	1
8	**BMC PATROLWATCH for ServerWORKS**	**Included with the AlphaServer**	0	1
Note: Line Items 9 through 12 are used with the System Management Station and are installed on the target AlphaServer system.				
9	**Base UNIX systems management tools**	**Included with** **DIGITAL UNIX**	0	1
10	**BMC PATROL Agent**	**Included with the AlphaServer**	0	1
11	**BMC Operating System Knowledge Module**	**QB-5KLAA-WC**	0	1
12	**BMC Knowledge Module Middleware/Messaging Internet/Intranet**	**QB-5KWAA-WC**	0	1

Table 2-3: Component Revision Levels

Hardware Component	Hardware	Firmware	Software
SRM Console	–	V1.21-18	V4.9-1
SCSI host adapter (KZPDA-AA)	0002	–	–
FWD SCSI controller (KZPSA-BB)	0000	A10	–
Three Port UltraSCSI RAID Controller (KZPAC-CA)	–	V2.60	–
RAID Array controller (HSZ52-AJ)	A01	HSOF V5.1Z-0	–
4.3 GB disks (RZ29B-VW)	–	0016	–
466 MHz CPU (KN304-DB)	A03	–	–
Memory (MS330-FA)	A03	–	–
Fast Ethernet adapter (DE500-AA)	C01	–	–
PCI-FDDI Adapter (DEFPA-AB)	B02	V3.1	–
StorageWorks Shelf Power Supply (DS-BA35X-HH)	B01 or Greater	–	–
Software Component	**Version**	**Patch Level**	
DIGITAL UNIX	4.0B	(BL7)DUV40BAS00005-19970926	

Special Configuration Rules

The minimum configuration was tested using an UltraSCSI controller and fast SCSI disk drives, the data transfer rate between the controller and disk drives was not tested at UltraSCSI speeds. Therefore, when configuring the KZPAC-CA Array Controller, set the data transfer rate at 5 MHz.

3

System Installation and Setup

This chapter presents information that is useful when installing and setting up a DIGITAL HiTest System configured from this DIGITAL HiTest Suite. System preparation includes installation of the hardware, operating system, and applications.

Hardware Installation (Maximum Configuration)

Install and interconnect the hardware as shown in Appendix A.

Disk Storage Configuration

Configure the HSZ50 StorageWorks Array Controllers using the StorageWorks HSZ50 Array Controller Utility for DIGITAL UNIX, according to the *StorageWorks HSZ50 Array Controller, HSOF V5.1 Configuration Manual* (EK-HSZ50-CG. C01*)* and the associated *CLI Reference manual* (EK-HSCLI-RM. B01).

Configuring the RAID 5 Stripesets

Configure the storage in RAID 5 stripesets. For this test configuration, each stripeset consisted of four disks, 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 RAIDset.

Where *n* is the number of drives:

(n - 1) / n *   (n * size of each device) = disk space per RAIDset

(4 - 1) / 4 *   (4 * 4.3 GB) = 12.9 GB usable space

Calculating Maximum Possible Chunk Size

The maximum chunk size possible for a RAID 5 set is determined using the number members in the RAIDset. Beginning with the maximum chunk size setting for HSZ controllers (2048 disk blocks of 512 bytes each), and based a db_block_size of 32 Kbytes, the maximum possible chunk size for our configuration was calculated as outlined in Table 3-1.

Calculate maximum possible chunk size using Table 3-1 as an example.

Table 3-1: Calculation for Maximum possible Chunk Size

Variable	Calculation	Comment
db_block_size	= 32 Kbytes	= 32,768 bytes.
max chunk size	$= \frac{2048}{(n\text{-}1)}$ $= \frac{2048}{(4\text{-}1)}$ = 682.7	*n* = # of RAIDset disks. Maximum HSZ chunk size, divided by RAID 5 stripesets 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	Maximum chunk size possible in whole db blocks per chunk (rounded down).

Setting Chunk Size on HSZ Console

For best performance, the chunk size for the Oracle7 database raw devices should be computed and set accordingly. Compute optimal chunk size based on db_block_size, and db_blocks per chunk using the following formula:

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

The optimal chunk size will be within the maximum possible chunk size calculated in Table 3-1.

A chunk size of 128 Kbytes (four db_blocks at 32 Kbytes per block) was chosen for this test configuration based on the maximum possible I/O size of Oracle during a sequential scan. The chunk size of 128 Kbytes, or four db blocks/chunk, is within the maximum possible calculated in Table 3-1.

Enabling Write-Back Cache

Enabling HSZ Write-Back Cache is recommended. The Write-Back Cache allocates cache memory to both read and write operations, permitting processing to continue without waiting for I/O completions. With Write-Back Cache enabled, write operation performance was improved, and database creation times reduced. Enabling Write-Back Cache provides significant improvement in I/O throughput.

Hardware Installation (Minimum Configuration)

Install and interconnect the hardware as shown in Appendix A.

Disk Storage Configuration

Disk storage is configured using the RAID Configuration Utility (RCU) to setup the RAID Array Controller and configuring the disk storage.

Setting up the RAID Array Controller

After installing the KZPAC-CA controller and prior to configuring the RAID Array, perform the controller setup procedure to load parameters into the controller firmware. These parameters help define the controller interface with the disk drives in the StorageWorks storage pedestals or internal shelves: enables battery backup (if installed), the method of caching, and the rate at which data are transferred between the controller and disk drives.

1. Configure the KZPAC-CA RAID Array Controller using the RAID Configuration Utility (RCU) according to the *RAID Array 230/Plus Subsystem RAID Configuration Utility User's Guide* (AA-R07GA-TE).
2. If a battery backup option is installed, enable Battery Backup setting.
3. Set the Maximum Transfer Rate on each SCSI Bus to 5 MHz.
4. Use default settings for rebuild rate and stripe size.

Configuring Disk Storage

Configure the array of disks attached to each SCSI bus attached to the KZPAC Array Controller. For this test configuration, each stripeset consisted of seven disks, at 4.3 GB per device, for a total of 30.1 GB of disk space with approximately 25.8 GB exported usable space per RAIDset.

Use the RCU according to the *RAID Array 230/Plus Subsystem RAID Configuration Utility User's Guide* (AA-R07GA-TE) and configure the disk storage as follows:

1. Select RAID 5 stripesets.
2. Select write-back cache setting if a battery backup option is installed. If a battery backup option is not installed, select write-through caching policy.

Operating System Installation

Install the DIGITAL UNIX 4.0B operating system with all kernel options according to the *DIGITAL UNIX Installation Guide for Version 4.0B* (AA-TLGA-TE) and *DIGITAL UNIX Version 4.0B Release Notes* (AA-QTLMB-TE), except where noted.

Install DIGITAL UNIX V4.0B Patch Kit 5 (BL7) (DUV40BAS00005-19970926).

Swap Space

When installing DIGITAL UNIX, use the 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
```

UNIX Kernel Parameters (/etc/sysconfigtab)

The parameter settings used for the UNIX kernel were based on a configuration with 1 GB (Minimum Configuration) and 4 GB (Maximum Configuration) of memory. Recommended parameter settings for these memory configurations are specified in Table 3-2.

Maximize 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.

Applications

Install the Oracle and SAS applications as discussed in the following paragraphs.

Oracle

Install the Oracle7 Server, Version 7.3.3; including the Parallel Query Option, PL/SQL, SQL*Plus, and SQL*Net, using Oracle Installer from the Oracle7 Server CD. Use standard defaults.

Oracle 64-bit Option Parameters

Set the Oracle7 Server parameters to optimize decision support using the parameter settings specified in Table 3-3.

Enabling Process Limits for Oracle DBA User Account

Edit the Oracle DBA User Account process settings to enable use of larger parameter process values that are required for the Oracle 64-bit Option. For more information, see O*racle7 Server for DIGITAL UNIX 7.3.3*.

Using the UNIX C Shell, ensure that the following settings are in the `.login` file of the Oracle DBA user account:

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

Table 3-2: UNIX Kernel Parameters

Parameter	Minimum Configuration Value	Maximum Configuration Value
vm parameters:		
vm-maxvas	1073741824	4292967296
ubc-minpercent	1	1
ubc-maxpercent	2	2
gh_chunks	Not Set	740
rt parameters:		
aio-max-num	1024	1024
aio-max-percent	2	2
proc:		
max-proc-per-user	1024	1024
max-threads-per-user	1024	1024
max-per-proc-data-size	1073741824	42924967296
max-per-proc-address-space	1073741824	42924967296
ipc parameters:		
shm-max	1073741824	2139095040
shm-seg	32	32
msg-max	8192	8192
msg-mnb	16384	16384
msg-mni	1024	1024
msg-tql	4096	4096
sem-aem	16384	16384
sem-mni	200	256
num-of-sems	200	200
sem-msl	Not Set	200
sem-opm	200	200
sem-ume	200	200
sem-vmx	32767	32767
ssm-threshold	8	0

Table 3-3: Oracle 64-bit Option Parameters

Parameter	Minimum Configuration Value	Maximum Configuration Value	Comment
db_block_size	32768	32768	DSS and data warehouse applications benefit from a Big Oracle Block (BOB).
db_block_buffers	10000	90000	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	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	104857600	Memory area allocated to each process/thread performing sorts. This parameter should be maximized during intensive sort operations, such as index creation on large tables. This is highly sensitive to the parallel degree of the tables being sorted, as each thread will allocate this space.
sort_direct_writes	true	true	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	131072	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	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 large enough to accommodate the smaller dimension tables in the Consumer Packaged Goods database.
ccf_io_size	524288	524288	Determines the number of bytes per write when creating a contiguous file.
db_file_multiblock_ read_count	4	4	Due to an Oracle limit of 128 KB per I/O, this parameter was set to 4: the maximum based on the formula max_io_size/db_block_size (128 KB/32 KB).
parallel_max_servers	72	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	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. If cache misses are prevalent, the open_cursors parameter may need to be increased.
Log_checkpoint_interval	99999999	99999999	Set sufficiently large number to disable time-based checkpoints forcing checkpoints only to occur on log switches.

SAS Products

Install the SAS System, Release 6.12 using the SAS System Installation Manager from the SAS System distribution media, according to *Installation Instructions for the SAS System under UNIX Environments, Release 6.12, TS040*, (Cary, NC:SAS Institute Inc., 1997).

Making SAS Available to Users by Linking

Use the link method to make the SAS System available to users, as described in the *Installation Instructions for the SAS System*.

Define a link for SAS as follows:

```
ln -s /var/sas612/sas /usr/bin/sas
```

Note

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

Editing the SAS Config File

Edit the SAS configuration file, `config.sas612,` to provide sufficient memory for the large SAS dataset size as indicated in Table 3-4.

Table 3-4: SAS Configuration File Settings

Setting	Comment
-memsize 256m	-`memsize` limits amount of memory allocated by the SAS System. Default is 32 MB.
-sortsize 256m	-`sortsize` limits 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`.

Database Creation and Storage Considerations

Creating a database which ensures proper and efficient use of Oracle7 requires special considerations. This section describes these considerations and provides an overview of how they were used.

When creating the file system and a database, configure the file system and database storage based on the characteristics of the database to be installed. Allocate disk space to optimize performance. Allocate storage to minimize Input/Output (I/O) contention and maximize I/O throughput.

The file system and database storage map for the maximum and minimum configurations of this HiTest suite are shown and discussed in Chapter 4. The processes used to create the maximum and minimum configuration databases for this HiTest suite are discussed in Chapter 4.

Raw Disk Handling Considerations

Oracle supports the use of file system and raw devices. Implement the database to take advantage of DIGITAL UNIX asynchronous I/O capability. The amount of overall disk space required depends on the volume of data in the database, the database processes during peak load, as well as the expected growth and expansion over time. When using an Oracle application, spread Oracle database files over as many disks as possible.

Consider the following when setting up the raw disk:

- Use RAID controllers. This Suite had the disks set up as RAID 5 sets at the hardware level using the RAID array controllers. RAID 5 provides disk striping with distributed parity, but provides only the capacity of *n*-1 devices, where *n* = Number of RAIDset members. For example, four 4.3 GB drives have a 12.9 GB RAID 5 capacity.
- When possible, spread system disk, database tables, indexes, and Oracle redo logs across physically separate raw disks.
- Plan to place data to raw disks on different SCSI buses to reduce I/O contention.
- Use a symbol link to assign meaningful names to all the raw disks to improve manageability.

An example of how the raw disk was setup and partitioned is shown in Chapter 4.

Temporary Tables

Temporary tables are used for sorting. Consider the following when creating temporary tables:

- Create this tablespace on its own disk.
- Increase sort_area_size to support large sort operations.

Tablespace Configuration Considerations

Consider the following when configuring tablespaces:

- For better performance, place the tablespaces on separate disks.
- Try to put the tablespaces for an index on a different disk from the one that holds the index’s table.

4

Interoperability Tests and Results

This chapter describes how the tests were set up (including database organization), what data and programs were placed on what disks, and how the tests were run.

This chapter describes:

- Test Environment
- Test Tools
- Test Configuration
- Test Process and Results

Overview of Results

Interoperability testing was performed successfully on the SAS Oracle DIGITAL UNIX AlphaServer 4x00 HiTest Suite

Oracle database tests were executed on the AlphaServer 4x00 system to verify the ability to process queries, in which all query data resides on the local node in a single Oracle7 database. All tests were conducted in both minimum and maximum configurations with both cold and warm cache. Single-user tests used cold cache and multiple user tests were executed with warm cache. All users were local terminal users.

Subsequently, SAS system database tests were conducted with the SAS system running scripts to test forecasting and statistical modeling functions.

All tests demonstrated typical   queries and verified interoperability of the components.

Test Environment

Figure 4-1 shows the maximum configuration SAS Oracle DIGITAL UNIX AlphaServer 4x00 HiTest System Test Environment.

Figure 4-2 shows the minimum configuration SAS Oracle DIGITAL UNIX AlphaServer 4x00 HiTest System Test Environment.

Detailed drawings showing the test configurations of the DIGITAL UNIX AlphaServer 4x00 foundation hardware are shown in Appendix A.

Figure 4-1: Maximum Configuration Test Environment

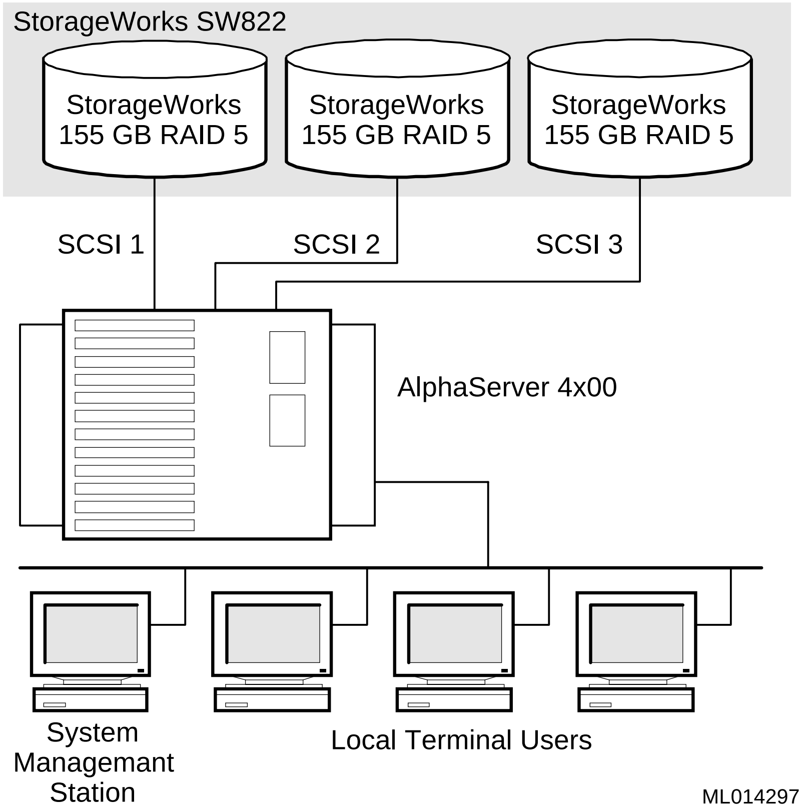
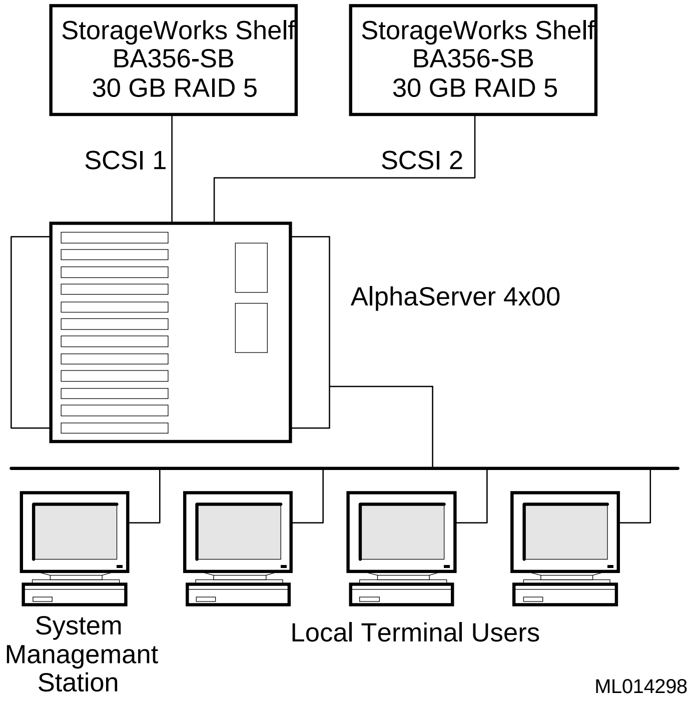

StorageWorks SW822
StorageWorks 155 GB RAID 5
StorageWorks 155 GB RAID 5
StorageWorks 155 GB RAID 5
SCSI 1
SCSI 2
SCSI 3
AlphaServer 4x00
System Managemant Station
Local Terminal Users
ML014297

Figure 4-2: Minimum Configuration Test Environment


StorageWorks Shelf BA356-SB 30 GB RAID 5
StorageWorks Shelf BA356-SB 30 GB RAID 5
SCSI 1
SCSI 2
AlphaServer 4x00
System Managemant Station
Local Terminal Users
ML014298

Test Tools

This section describes the test tools used for interoperability testing.

Consumer Packaged Goods Database

The Consumer Packaged Goods (CPG) database containing data representative of sales information for a variety of products across geographical areas of the US, including sales volume, channels, and other characteristics were used as the test database. The minimum configuration database used consists of 164061480 rows of sales data. The maximum configuration database consists of 928643603 rows of sales data

The databases were 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 was optimized for decision support.

Test Scripts

Test scripts were used to demonstrate and test the performance capabilities of the SAS Oracle DIGITAL UNIX AlphaServer 4x00 HiTest suite, in a Data Warehousing and Decision Support application. The scripts were designed to emulate typical decision support questions about the historical activity of a product sales environment.

Test Configuration

The minimum configuration and maximum configuration test loads were generated using the SQL Loader. The database consists of distributed raw devices. In the maximum configuration tested, each volume is a four disk RAID 5 stripeset on one of three HSZ50 Array Controller pairs. In the minimum configuration tested, each volume is a seven disk RAID 5 stripeset on a KZPAC-CA, a three port PCI-based RAID Array Controller.

Maximum Configuration

The maximum configuration database is made up of four RAID 5 data files comprising the large SALES_FACT table. The index for the SALES_FACT table consists of four data files. Four volumes of TEMP space was required to build the SALES_FACT_INDEX. The database used to test the maximum configuration consists of 928643603 rows of sales data.

Minimum Configuration

The minimum configuration database is made up of two RAID 5 data files comprising the large SALES_FACT table. The index for the SALES_FACT table consists of two data files. Two volumes of TEMP space was required to build the SALES_FACT_INDEX. The database used to test the minimum configuration consists of 164061480 rows of sales data.

File System and Database Storage Map (Maximum Configuration Database)

The file system and database storage map for the maximum configuration tested is provided in Table 4-1. Performance considerations guided the placement of the data files. As shown in the file system and database storage map, the database consists of approximately 140 GB of storage. Each volume is a four disk RAID 5 stripeset, on one of three HSZ50 Array Controller pairs.

As shown in Table 4-1, the data files are primary pathed across three HSZ50 Array Controller pairs, while the index files are primary pathed across a separate set of three HSZ50 pairs. TEMP file space is evenly dispersed across all HSZ50 Array Controller pairs. This configuration maximizes I/O access paths, while minimizing I/O contention for data load and data retrieval.

One 12.9 GB RAID 5 stripeset was subdivided to seven smaller equal partitions of 1,785,490 KB 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.

File System and Database Storage Map (Minimum Configuration Database)

The file system and database storage map for the minimum configuration tested is provided in Table 4-2. As shown in the file system and database storage map, the database consists of approximately 24 GB of storage. Each volume is a seven disk RAID 5 stripeset on one of two KZPAC-CA Array Controller ports. Performance considerations guided the placement of the data files. Tables and indexes for the tables were placed on separate ports to minimize I/O contention.

Test Process and Results

Interoperability of the SAS Oracle DIGITAL UNIX AlphaServer 4x00 HiTest suite was tested by creating a database, loading data to the database using the SQL loader, and creating the indexes required to access the stored data. Then, test queries were made to the data using Oracle and SAS applications to demonstrate typical queries.

Database Creation

The database was created by partitioning the disk space, creating tablespaces, altering the tablespaces to expand the database, creating the tables, loading the data, and creating indexes.

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.

The resultant Oracle7 Consumer Product Database consists of five tablespaces.

Table 4-1: File System/Database Storage Map, Maximum Configuration

SCSI BUS #1	
HSZ50 Controller	
/dev/rrzb8c:	FACTS1
/dev/rrz8c:	FACTS4
/dev/rrzb9c:	/sas1
/dev/rrz9c:	/sas2
HSZ50 Controller	
/dev/rrzb10c:	TEMP1
/dev/rrz10c:	/sas2
/dev/rrzb11c:	FACTS INDEX3
/dev/rrz11c:	/sas1
SCSI BUS #2	
HSZ50 Controller	
/dev/rrz16a:	system
/dev/rrz16b:	swap1
/dev/rrz16g:	usr
/dev/rrzb16c:	ORACLE
/dev/rrzc16c:	FACTS2
/dev/rrz17c:	var
/dev/rrzb17a:	DIMENSIONS
/dev/rrzb17b:	loga_1
/dev/rrzb17c:	logb_1
/dev/rrzb17d:	logc_1
/dev/rrzb17e:	DIMENINDEX
/dev/rrzb17f:	SYSTEM
HSZ50 Controller	
/dev/rrz18c:	swap2
/dev/rrzb18c:	/data1
/dev/rrz19c:	FACTSINDEX2
/dev/rrzb19c:	TEMP4
SCSI BUS #3	
HSZ50 Controller	
/dev/rrzb24c:	FACTS3
/dev/rrz24c:	TEMP2
/dev/rrzb25c:	/sas1
/dev/rrz25c:	/sas2
HSZ50 Controller	
/dev/rrzb26c:	FACTSINDEX1
/dev/rrz26c:	TEMP3
/dev/rrzb27c:	FACTS INDEX4
/dev/rrz27c:	/data2

Table 4-2: File System Database Storage Map, Minimum Configuration

SCSI BUS #1	
KZPAC Controller (Port 1)	
/dev/re0a	loga_1
/dev/re0b	system
/dev/re0c	DIMENSIONS
/dev/re0d	TEMP1
/dev/re0e	FACTS1
/dev/re0f	/data1
/dev/re0g	FACTS2
/dev/re0h	/sas1
SCSI BUS #2	
KZPAC Controller (Port 2)	
/dev/re1a	logb_1
/dev/re1c	DIMENINDEX
/dev/re1d	TEMP2
/dev/re1e	FACTS_INDEX1
/dev/re1f	/data2
/dev/re1g	FACTS_INDEX2
/dev/re1h	/sas2

Tablespace Configurations

Both the FACTS and FACTSINDEX tablespaces use the storage parameters shown in Table 4-3. The FACTS tablespace contains the SALES_FACT table, which is populated using `sqlldr`. Since the initial extent of a tablespace is not used by `sqlldr`, the initial extent was set to 64K to limit this unusable space. As shown in Table 4-3, the only difference between the tablespace configurations between the maximum and minimum configurations is the datafile size.

Table 4-3: Database Tablespace Storage Parameters

Storage Parameters	FACTS	FACTSINDEX	TEMP
datafile size (Max. Configuration)	12,269 MB	12,269 MB	12,269 MB
(Min. Configuration)	4,670 MB	4,670 MB	4,670 MB
initial_extent	64 KB	1,000 MB	200 MB
next_extent	1,000 MB	1,000 MB	200 MB
max_extent	2,041 MB	2,041 MB	2,041 MB
pctincrease	0	0	0
pctfree	0	0	N/A
Note: 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.			

Tablespace Creation

The creation of the large tablespaces (FACTS, FACTSINDEX, and TEMP) allocating the initial datafile of 12269 MB for the maximum configuration database and 4670 MB for the minimum configuration database 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.

The following Oracle parameter is a key factor in maximizing I/O throughput, and should be set accordingly:

Parameter	Setting
`ccf_io_size`	524,288 bytes (512 Kbytes or 1024 disk blocks)

For ccf_io_size, the Oracle parameter specifying the number of bytes per write when creating a contiguous file, the optimal setting was found 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 stripeset.

As shown in Table 4-4, the resultant database consists of five tablespaces. The free space associated with each tablespace is shown to indicate the extent to which the various tablespaces are used.

Table 4-4: Oracle7 Database, Maximum Configuration

Tablespace	Contains	Size	Free Space %
FACTS	SALES_FACTS Table 4 Data Files (12,269 MB each)	49,076 MB 0.9 B rows (928,643,603)	11 %
FACTSINDEX	SF_KEY Index on SALES_FACTs Table 4 Data Files (12,269 MB each)	49,076 MB	18 %
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	4 Data Files (12,269 MB each) required for SF_KEY index build	49,076 MB	N/A

Database Load, SQL Loader (Maximum Configuration)

Data was loaded to the database using the SQL Loader. As shown in Table 4-5, three different load scenarios were used to load the facts data. In each method, one months data in a flat file, approximately 5.2 GB in size, is used as input to the sqlldr. Each data file has two months data:

- One flat file was loaded through sqlldr to the FACTS1 data file. The number of rows loaded and the overall load time are listed in Table 4-5. System monitoring showed that only 86% of the CPU was being used during this load operation.
- The remaining three flat files in the FACTS 1 and FACTS 2 data files of the minimum configuration database were loaded in parallel. The result was an increase in CPU utilization to 96% and an overall reduction in total load time of approximately 30 minutes.
- When loading the additional data files required to build the maximum configuration database, the additional four flat files were loaded in parallel into the FACTS3 and FACTS4 database files. The number of rows loaded, overall load times, and load rates at which the databases were created for all three methods are listed in Table 4-5.

Table 4-5: Facts Database Load Rates, Maximum Configuration

Data Files	Rows	Time	Rows/Minute	Load Scenario
FACTS1	118882768	02:08:17	928772	Serial Creation, 1 Stream
	107377984	05:25:43	992176	Parallel Creation, 3 Streams
FACTS2	118882768	05:44:57		
	115047840	05:29:59		
FACTS3	118882768	01:58:23	1007481	Parallel Creation, 4 Streams
	115047840	01:48:44	1055484	
FACTS4	118882768	01:46:24	1121535	
	115047840	01:46:20	1090932	

Database Load, SQL Loader (Minimum Configuration)

For the minimum configuration database, two data files were loaded to FACTS1 and FACTS2 in parallel resulting in 100% CPU utilization. The number of rows loaded and the load times are specified in Table 4-6.

Table 4-6: Facts Database Load Rates, Minimum Configuration

Data Files	Rows	Time	Rows/Minute	Load Scenario
FACTS1	82033536	02:28:45	554281	Parallel Creation (2 Streams)
FACTS2	82027944	02:24:06	569639	

Creating the Index

Creating the index is accomplished by allocating temporary storage to allow the index to be built and then using the Oracle Parallel Query option to build the index. During index creation, it is important to allocate sufficient space for temporary storage for the index to build successfully. Due to the time required to build an index on a data warehouse, it is important to calculate these parameters correctly.

Calculating Required Index Space

Temporary space requirements should be estimated at a minimum of 10% above the required index space.

Calculate initial index and temporary space required, to allow creation of a concatenated index SF_KEY on the large SALES_FACT table.

Calculate the space requirements using the following procedure:

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

   ```
   sales_fact = 928643603 rows (Maximum Configuration)
   ```

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

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

3. Calculate available data space per data block as follows:

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

4. Calculate the entry column size by summing the length of the columns in the index. The four columns used in this index are product_id, chan_id, market_id, and day, which are declared as 3 char (6) and a date field (7).

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

5. Calculate the bytes per entry as follows:

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

6. Calculate total index blocks required as follows: (Maximum configuration example)

index blocks	1.05 * ((#rows * bytes per entry) = -- ((Available data space/bytes per entry)*byte per entry)) = 1.05 * ((928643603 * 38) / ((32607/38) * 38)) = 1130764 index blocks **Note:** 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.
Index bytes	= index blocks * db block size = 1130764 * 32768 = 37052879760 bytes
index MB	= index bytes / (1024 * 1024) = 35336.375 = 35510.9 MB

7. The parallelism and extent sizes used to enhance performance will adversely affect efficiency of the storage utilization for the index. To account for this behavior a 10% safety margin is built into the calculation of final index space as follows:

Final index space (MB)	= index MB * 1.1 = 35510.9 MB * 1.1 = 39061.9 MB

8. Determine (whole number) the maximum number of extents in a single datafile. For this database, maximum number of extents is calculated as follows:

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

9. Determine total number of files required to hold the SF_KEY index as follows:

index files	final index space (bytes) = -- (extent size * number of extents per file) = 39061.9MB/(1048576000*12) = 3.18 (always rounded up) = 4

Index Creation

Parallel index creation, provided by the Oracle Parallel Query option, is an important database management function. Parallel index creation was used during creation of the index on the large SALES_FACT table.

The concatenated SF_KEY index was created with a parallel degree of two for the minimum configuration tested and a parallel degree of six for the maximum configuration tested, which requires that the parallel_max_servers parameter be set to a minimum of twice the parallel degree 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
```

For Minimum Configuration Tested:

```
svrmgr> alter TABLE SALES_FACT parallel (degree 2);
```

For Maximum Configuration Tested:

```
svrmgr> alter TABLE SALES_FACT parallel (degree 6);
```

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

The changes specified in Table 4-7 were made to the standard database parameters to facilitate index creation.

Table 4-7: Index Parameters

Parameter	Minimum Configuration Value	Maximum Configuration Value
db_block_buffers	2000	2000
sort_area_size	83886080	536870912
sort_write_buffer_size	131072	131072

Index creation was performed for 164061480 rows of SALES_FACT data to support the minimum tested configuration and 928643603 rows of SALES_FACT data to support the maximum configuration tested. The resulting elapsed time for index creation appears in Table 4-8.

Table 4-8: Index Creation Time

Index Creation (Rows of Data)	Elapsed Time	Rows/sec	Parallel Degree
Minimum Configuration			
164061480	03:41:30	12345	2
Maximum Configuration			
928643603	09:12:00	28039	6

The actual *create index* statement used was:

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

Oracle Database Queries

Functional verification of the ability to perform query operations was demonstrated using five SQL join scripts and a full table scan. These queries exercised functionality of the RDBMS server, SQL*Plus, and Parallel Query. The scripts for each of the queries are detailed in Appendix B.

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 queries would be used to generate sales trends.

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

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?"

Oracle Database Tests

Three Oracle Database tests were performed to test interoperability. The tests and test results are discussed in the following paragraphs.

Test One - Process Executing Queries 1– 5 Sequentially

The first test performed five sequential queries (SQL joins). The performance of a query is dependent upon the optimizer choosing an efficient query plan. In general, analyzing the tables and indexes enable the optimizer to produce effective plans. The results of test one are listed in Table 4-9 and Table 4-10; times recorded for multiple users are average times.

Table 4-9: Performance Results for Test One, Maximum Configuration

Maximum Configuration (928643603 Rows)			
	1 USER	**2 USERS**	**4 USERS**
	Cold Cache	**Warm Cache**	
Query 1	234 sec	24 sec	38 sec
Query 2	143 sec	31 sec	53 sec
Query 3	72 sec	33 sec	56 sec
Query 4	429 sec	68 sec	112 sec
Query 5	344 sec	205 sec	335 sec
Total	1222 sec	361 sec	594 sec

Table 4-10: Performance Results for Test One, Minimum Configuration

Minimum Configuration (164061480 Rows)			
	1 USER	**2 USERS**	**4 USERS**
	Cold Cache	**Warm Cache**	
Query 1	49 sec	28 sec	58 sec
Query 2	39 sec	39 sec	80 sec
Query 3	27 sec	39 sec	80 sec
Query 4	103 sec	82 sec	133 sec
Query 5	148 sec	242 sec	502 sec
Total	366 sec	430 sec	853 sec

Test Two - Executing Queries 1– 5 in Parallel

The second test executed all five queries in parallel. Parallel queries more stressfully exercise the database versus sequential queries performed in test one.

Performance results for the parallel query tests for the minimum and maximum configurations are listed in Table 4-11 and Table 4-12; times recorded for multiple users are average times.

Table 4-11: Performance Results for Test Two, Maximum Configuration

Maximum Configuration (928643603 Rows)			
	1 USER (5 Processes)	2 USERS (10 Processes)	4 USERS (20 Processes)
	Cold Cache	Warm Cache	
Query 1	230 sec	91 sec	154 sec
Query 2	198 sec	114 sec	214 sec
Query 3	90 sec	119 sec	224 sec
Query 4	489 sec	192 sec	325 sec
Query 5	404 sec	322 sec	560 sec
Total	404 sec	322 sec	560 sec

Table 4-12: Performance Results for Test Two, Minimum Configuration

Minimum Configuration (164061480 Rows)		
	1 USER (5 processes)	2 USERS (10 Processes)
	Cold Cache	Warm Cache
Query 1	66 sec	142 sec
Query 2	98 sec	187 sec
Query 3	79 sec	186 sec
Query 4	173 sec	275 sec
Query 5	238 sec	442 sec
Total	238 sec	442 sec

Test Three - Testing Parallel Queries Using Degrees of Parallelism

The degree of parallelism used in parallel query operations effects level of performance during parallel query operations. The impact to overall performance was tested using several different degrees of parallelism in both the minimum and maximum test configurations.

Tests were performed using three different degrees of parallelism: 2, 4, and 6 during tests of the minimum configuration, and 4, 8, and 12 during tests of the maximum configuration. The degree of parallelism is set using the following commands:

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

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 used, this limitation translated to 4 data blocks per I/O, regardless of the following setting:

```
db_file_multiblock_read_count
```

The Parallel Query Option was tested using the following query to obtain the number of rows in the SALES_FACT table:

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

Table 4-13 shows test three results and the effects of the degree of parallelism selected.

Table 4-13: Test Three Results, Count # of Rows from SALES_FACT

Minimum Configuration (164061480 Rows)	
Degree	**Time**
degree 2	11.3 min
degree 4	14.3 min
degree 6	14.8 min
Maximum Configuration (928643603 Rows)	
Degree	**Time**
degree 4	35.0 min
degree 8	39.1 min
degree 12	41.6 min

SAS System Database Tests

Functional testing of SAS included interactive script execution, where the interactive mode used the X-window display manager. Scripts retrieved data from the existing Oracle Consumer Packaged Goods Data Warehouse via SAS/Access. Forecasting and statistical modeling test scripts were executed.

Forecasting Tests

Scripts were executed to create datasets for forecasting at a daily, weekly, and monthly level results.

Test One

Daily forecast for a single market, single channel, single product, and eight months of daily data to produce a 30 day ahead forecast on a daily level.

Test Two

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

Statistical Modeling Tests

Scripts were executed to create datasets for statistical modeling to detect differences between stores, channels groups, products, and so on.

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 eight months of data. Examine the interaction between channel and month.

Test Two

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

5

System Limits and Characterization Data

This chapter describes any system limits that may have been determined as a result of the testing, along with information about the system characterization during testing. Area covered includes parallel database load characteristics.

Three different methods were used to load a flat file through sqlldr to a FACTS data file.

The three methods used were:

- Serial creation – 1 Stream
- Parallel Creation – 3 Streams
- Parallel Creation – 4 Streams

Tests using the three different methods indicate that use of parallel load significantly reduces database load times. Details regarding the database load characteristics are discussed in Chapter 4.

6
Problems and Resolutions

This chapter describes any problems that may have been encountered during the testing that have not been fixed and eliminated. The resolution provides the system manager or user with a fix or workaround for the problem.

The following problems were identified during testing:

DIGITAL UNIX Operating System

Setting Address Space Unlimited

Problem Cannot set "Limit Address Space Unlimited" in C-Shell on the 4 GB memory configuration.

Workaround Set limit-h address space 4117776K, then issue the "limit address space unlimited" command.

Resolution Edit the /sys/conf/nodename file and change the value for maxdsize to the memory size and do a kernel rebuild.

Application Software

Oracle7 Database Issues

Problem If the Oracle account is changed for any reason, a possible problem accessing the "control" file might occur. If the Oracle account is recreated and the group IDs do not match the original group IDs, the "control" file cannot be accessed even after the protection is reset on the control file to match the current group IDs.

Resolution Select one of the following:

1. Edit the password file to restore the previous default IDs associated with the Oracle account.
2. Save the previous group IDs of the Oracle account and reinstitute them in the new Oracle account.

Problem 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)

Resolution If using 32K data blocks (db_blocks), set this parameter to 4, which is the maximum based on the formula.

A

Detailed Hardware Configuration

Detailed Hardware Configuration This appendix describes the minimum and maximum hardware configuration for the following:

- System Diagram
- Configuration Cabling
- AlphaServer 4100 Configurations, including:
  - System Motherboard
  - PCI Backplane
- StorageWorks Cabinet Configuration

System Diagrams

Figure A-1 and Figure A-2 show overviews of the maximum and minimum HiTest Systems configured from the HiTest Template.

Figure A-1: System Diagram, Maximum Configuration

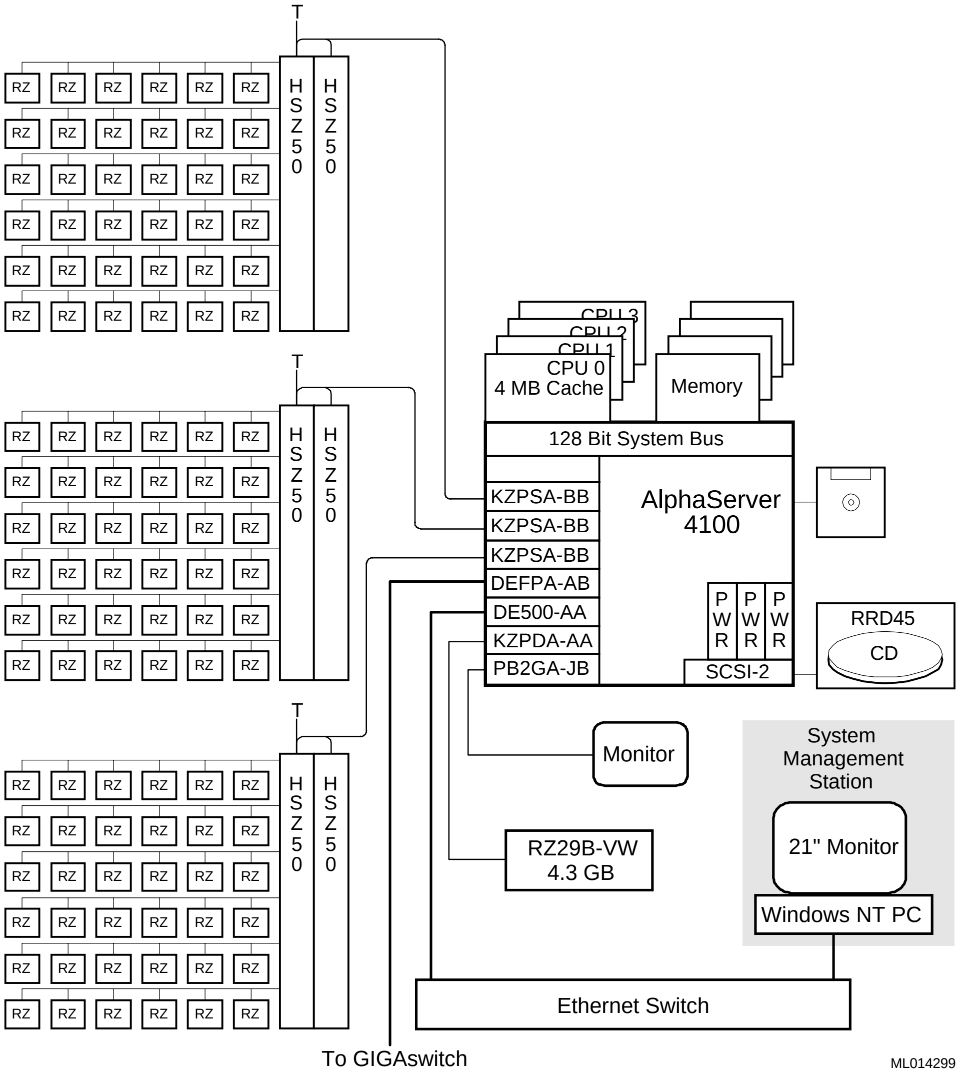
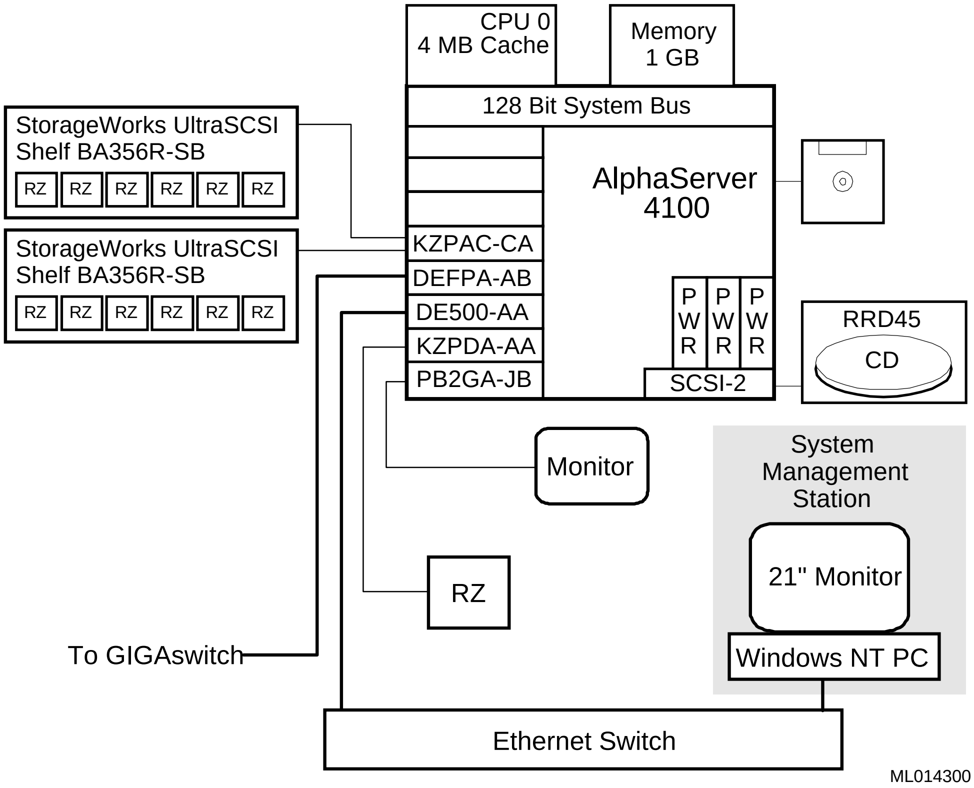


Figure A-2: System Diagram, Minimum Configuration



Configuration Cabling

Table A-1 and Table A-2 list the cabling required to interconnect the SCSI components of the maximum and minimum configurations, respectively.

Table A-1: Maximum Configuration SCSI Interconnect Cabling

Part Number	Qty	Description	From	To
BN21K-10	3	SCSI bus cables	KZPSA-BB in AlphaServer System enclosure	H885-AA on HSZ50s in StorageWorks SW800 Cabinet
Note: The following cables, connectors, and terminators are supplied as part of the HSZ52-AJ option and are used to interconnect the HSZ controller pairs and terminate the SCSI bus.				
H885-AA	6	Tri-link connectors		HSZ50 Controller
BN21L-0B	3	Shared SCSI bus cable	H885-AA on HSZ50	H885-AA on HSZ50
H879-AA	3	SCSI bus terminators		H885-AA on HSZ50

Table A-2: Minimum Configuration SCSI Interconnect Cabling

Part Number	Qty	Description	From	To
BN37A-1E	2	UltraSCSI bus cables	KZPAC-CA ports in AlphaServer System drawer	StorageWorks Shelf Kits BA356-SB Personality Module

AlphaServer 4100 Configurations

Figure A-3 and Table A-3 show the AlphaServer 4000 and AlphaServer 4100 system motherboard and describe the minimum and maximum hardware configurations used in this HiTest Template.

Figure A-3: AlphaServer 4000/4100 Motherboard Layout

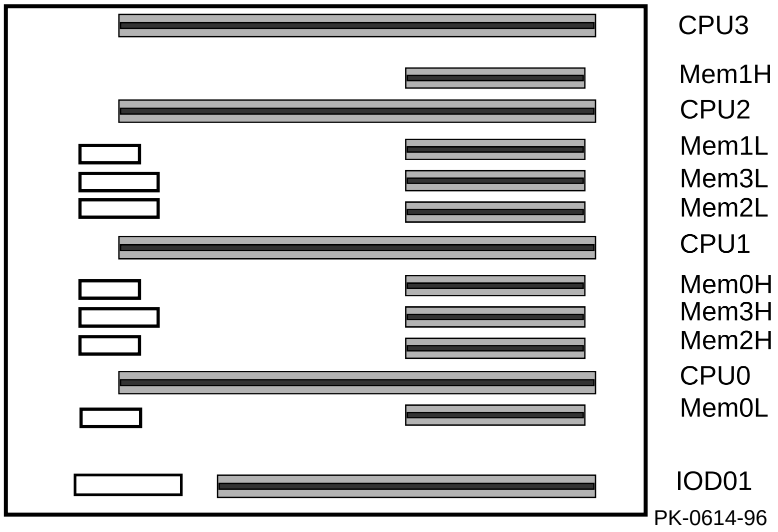


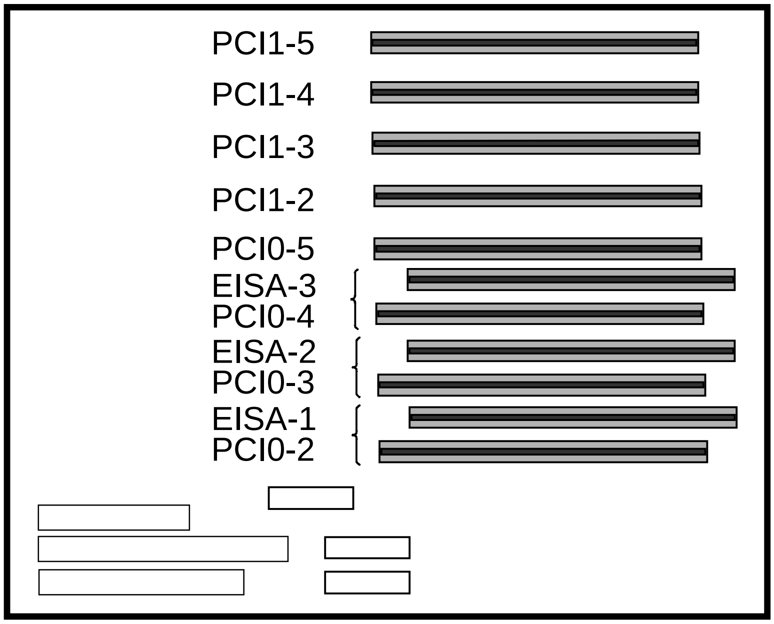
Table A-3: AlphaServer 4000/4100 Usage, (Min and Max Configurations)

Slot	Minimum Configuration Options	Maximum Configuration Options	Description
CPU3		KN304-DB	466 MHz CPU 4 MB cache
Mem1H		MS330-FA	Memory pair 1 (2 of 2)
CPU2		KN304-DB	466 MHz CPU 4 MB cache
Mem1L		MS330-FA	Memory pair 1 (1 of 2)
Mem3L		MS330-FA	Memory pair 3 (1 of 2)
Mem2L		MS330-FA	Memory pair 2 (1 of 2)
CPU1		KN304-DB	466 MHz CPU 4 MB cache
Mem0H	MS330-FA	MS330-FA	Memory pair 0 (2 of 2)
Mem3H		MS330-FA	Memory pair 3 (2 of 2)
Mem2H		MS330-FA	Memory pair 2 (2 of 2)
CPU0	KN304-DB	KN304-DB	466 MHz CPU 4 MB cache
Mem0L	MS330-FA	MS330-FA	Memory pair 0 (1 of 2)
IOD01	Bridge	Bridge	System Bus to PCI bus bridge module

AlphaServer 4100 PCI Slot Usage

Figure A-4 and Table A-4 show the PCI slot usage for the minimum and maximum configurations of this HiTest Template.

Figure A-4: AlphaServer 4100 PCI Slot Usage



ML013980

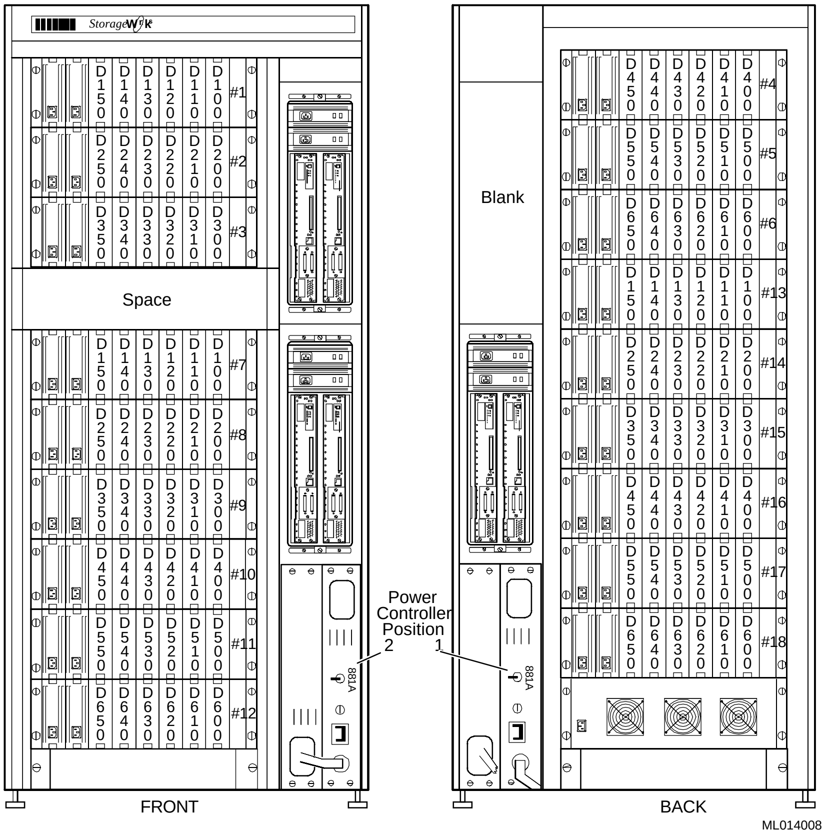
Table A-4: PCI Slot Usage (Minimum and Maximum Configurations)

Slots	Minimum Configuration Options	Maximum Configuration Options	Description
PCI1-5	DEFPA-AB		PCI-FDDI Adapter
PCI1-4		KZPSA-BB	FWD SCSI controller
PCI1-3		KZPSA-BB	FWD SCSI controller
PCI1-2		KZPSA-BB	FWD SCSI controller
PCI1-1	KZPAC-CA	KZPDA-AA	3 Port RAID controller FWD SCSI controller
PCI0-5	DE500-AA	DE500-AA	Ethernet controller
EISA-3/ PCI0-4	KZPDA-AA	DE500-AA	FWD SCSI controller Ethernet controller
EISA-2/ PCI0-3			
EISA-1/ PCI0-2			
PCI0-1	PB2GA-JB	PB2GA-JB	S3 TRIO64 1 MB Graphics Adapter

StorageWorks Cabinet SCSI Disks Unit Address Identification

Figure A-5 shows front and back views of the SW822 StorageWorks cabinet.

Figure A-5: SW822 Cabinet 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
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
FRONT
BACK
ML014008

HSZ50 SCSI Cabling

Table A-5, Table A-6, and Table A-7 define the point-to-point SCSI cable connections to be used between the HSZ50 ports and BA350 Modular Storage shelves of the SW822 cabinet.

Table A-5: HSZ50 #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-6: HSZ50 #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-7: HSZ50 #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

Each HSZ50 pair is cascaded using a BN21L-0B cable and two H885 Tri-link connectors. Additionally, an H879-AA terminator attaches to one of the Tri-link connectors to terminate the SCSI bus. Two H885 Tri-link connectors, a BN21L-0B cable and H879-AA terminator are supplied as part of each HSZ52-AJ option

B
Test Scripts

Test scripts were used to build the Oracle Database (tables, tablespace, and indexes), perform the Oracle database test queries, and execute the SAS System database tests.
This appendix provides the actual scripts used in testing.

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
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,
       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

```
#!/bin/sh
# set -x
echo “*** start create_facts.sh ***” >> bld_facts.log
date >> bld_facts.log
FACTS_TS=/oracle_home/links/FACTS1

svrmgr1 <<!
       SET echo on;
       CONNECT internal;


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

```
#!/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
 svrmgrl <<!
   SET echo on;
   CONNECT internal;

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

EXIT;
!
```

Oracle Database Test Queries

This section shows Timed Queries used for “Consumer Packaged Goods” (CPG) Tests.

Queries are (c) 1996 InfoDynamics LLC.

Query One

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

select /*+ STAR */
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 /*+ STAR */
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 /*+ STAR */
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 /*+ STAR */
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 /*+ STAR */
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;
```