# Volume Shadowing

# Hints / Kinks

# Performance

Manfred Kaser

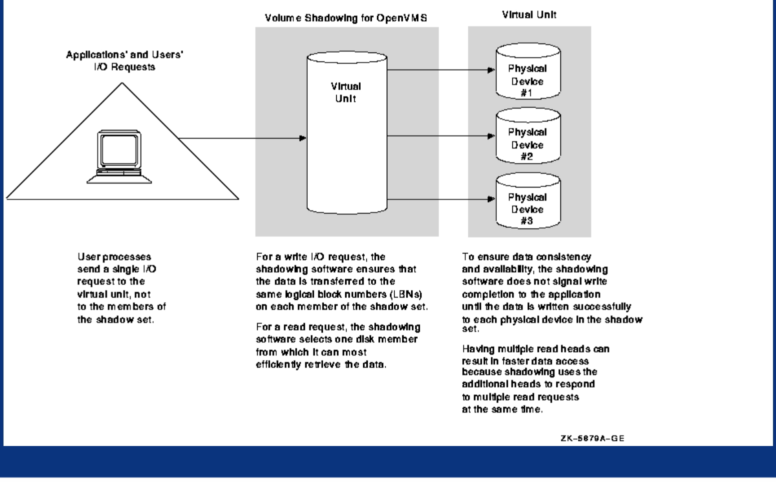
HP Services

Vortrag 1D04

## Overview

- Introduction to OpenVMS Volume Shadowing
- Preparing to use OpenVMS Volume Shadowing
- Creating and Managing Shadow Sets using DCL
- Minicopy for Backing up Data
- MiniMerge FC-based disks
- Shadowing Performance

# Introduction to Volume Shadowing



User processes send a single I/O request to the virtual unit, not to the members of the shadow set.

For a write I/O request, the shadowing software ensures that the data is transferred to the same logical block numbers (LBNs) on each member of the shadow set.

For a read request, the shadowing software selects one disk member from which it can most efficiently retrieve the data.

To ensure data consistency and availability, the shadowing software does not signal write completion to the application until the data is written successfully to each physical device in the shadow set.

Having multiple read heads can result in faster data access because shadowing uses the additional heads to respond to multiple read requests at the same time.

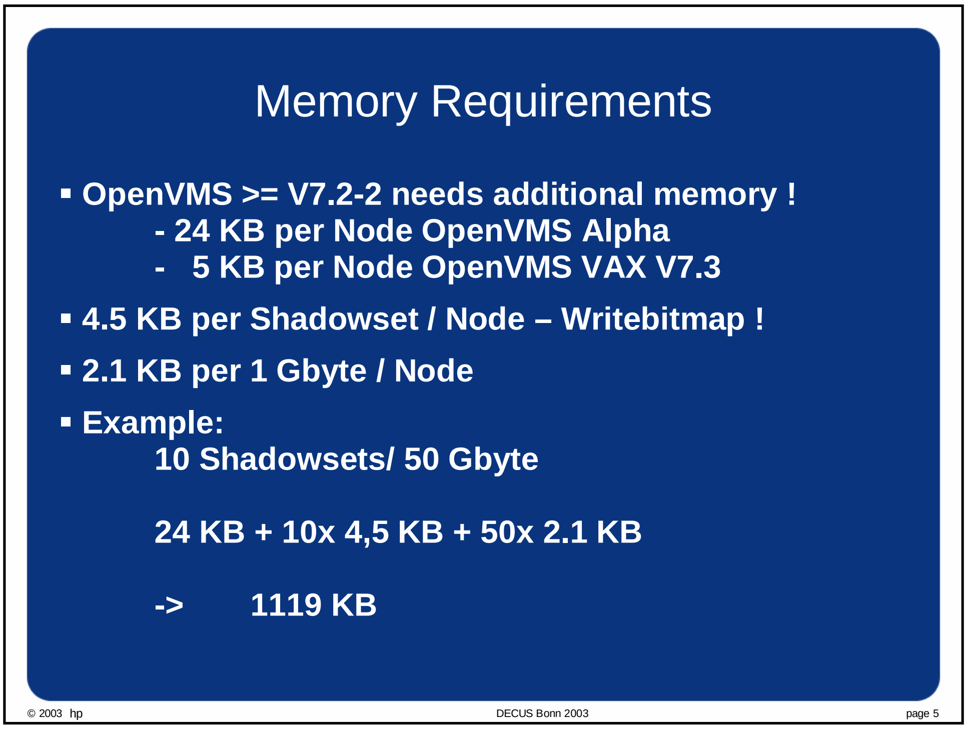
ZK–5879A–GE

# Hardware Environment

- **One CPU**
- **One mass storage controller**
- **One of the following kinds of disk drives:**

  **- Digital Storage Architecture ( DSA )**
  **- Small Computer Systems Interface ( SCSI )**
  **- Fibre Channel**

- **See SPD 27.29.xx for details**

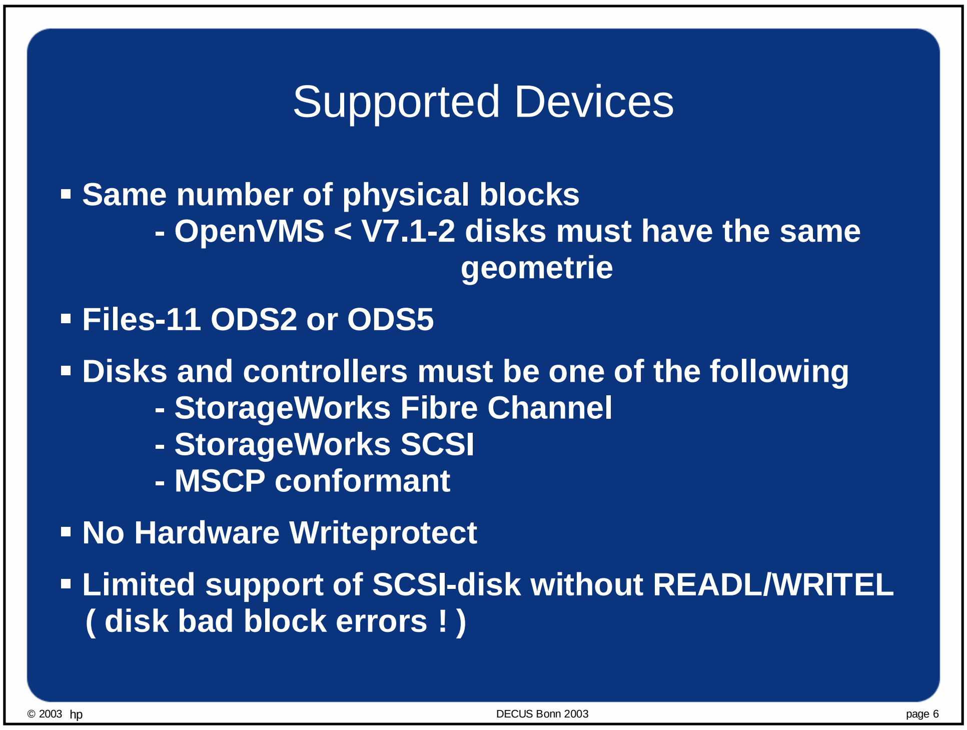


# Memory Requirements

- OpenVMS >= V7.2-2 needs additional memory !
  - 24 KB per Node OpenVMS Alpha
  -   5 KB per Node OpenVMS VAX V7.3
- 4.5 KB per Shadowset / Node – Writebitmap !
- 2.1 KB per 1 Gbyte / Node
- Example:
  10 Shadowsets/ 50 Gbyte

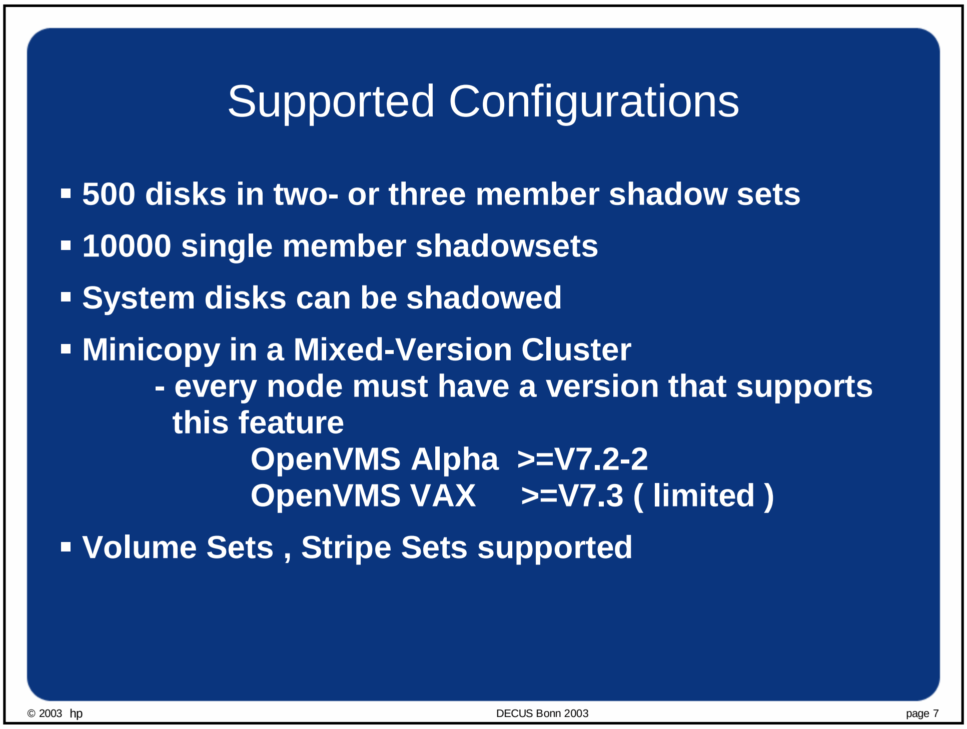
  24 KB + 10x 4,5 KB + 50x 2.1 KB

  ->      1119 KB



# Supported Devices

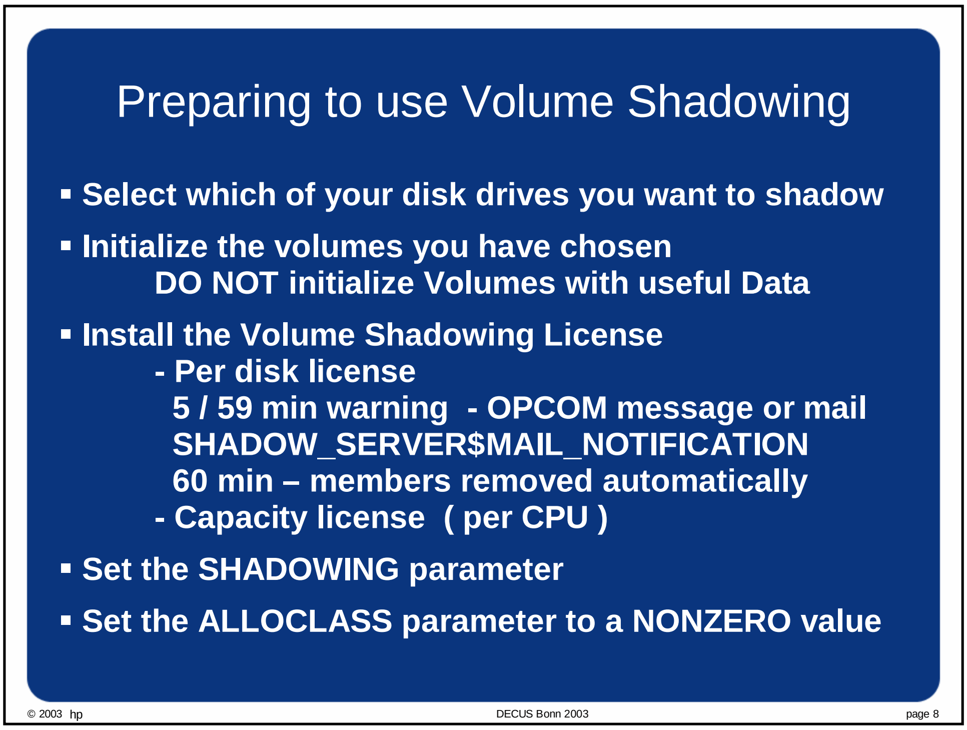
- Same number of physical blocks
  - OpenVMS < V7.1-2 disks must have the same geometrie
- Files-11 ODS2 or ODS5
- Disks and controllers must be one of the following
  - StorageWorks Fibre Channel
  - StorageWorks SCSI
  - MSCP conformant
- No Hardware Writeprotect
- Limited support of SCSI-disk without READL/WRITEL ( disk bad block errors ! )



## Supported Configurations

- 500 disks in two- or three member shadow sets
- 10000 single member shadowsets
- System disks can be shadowed
- Minicopy in a Mixed-Version Cluster
  - every node must have a version that supports this feature
    OpenVMS Alpha >=V7.2-2
    OpenVMS VAX >=V7.3 ( limited )
- Volume Sets , Stripe Sets supported

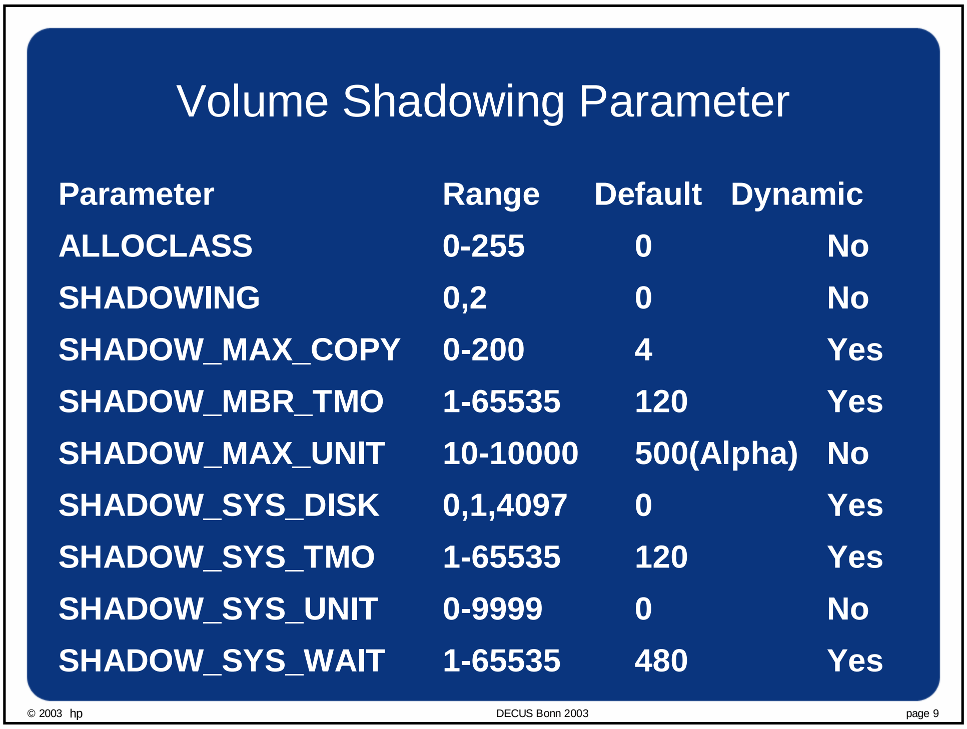
© 2003 hp DECUS Bonn 2003 page 7



## Preparing to use Volume Shadowing

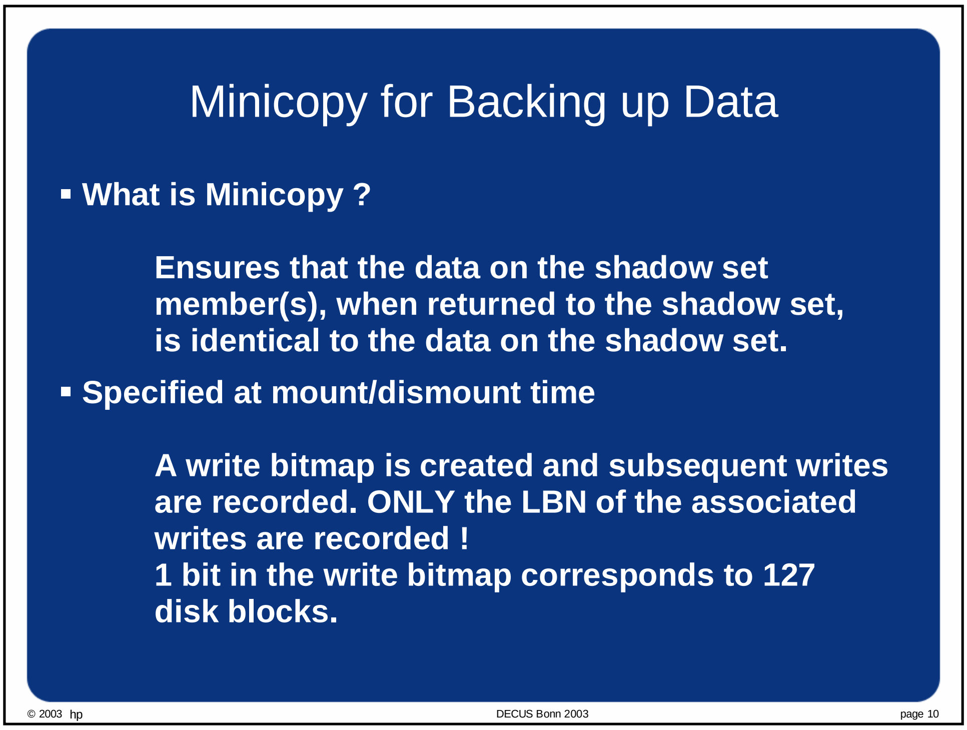
- Select which of your disk drives you want to shadow
- Initialize the volumes you have chosen
  DO NOT initialize Volumes with useful Data
- Install the Volume Shadowing License
  - Per disk license
    5 / 59 min warning - OPCOM message or mail
    SHADOW_SERVER$MAIL_NOTIFICATION
    60 min – members removed automatically
  - Capacity license ( per CPU )
- Set the SHADOWING parameter
- Set the ALLOCLASS parameter to a NONZERO value

© 2003 hp DECUS Bonn 2003 page 8



## Volume Shadowing Parameter

| Parameter | Range | Default | Dynamic |
|---|---|---|---|
| ALLOCLASS | 0-255 | 0 | No |
| SHADOWING | 0,2 | 0 | No |
| SHADOW_MAX_COPY | 0-200 | 4 | Yes |
| SHADOW_MBR_TMO | 1-65535 | 120 | Yes |
| SHADOW_MAX_UNIT | 10-10000 | 500(Alpha) | No |
| SHADOW_SYS_DISK | 0,1,4097 | 0 | Yes |
| SHADOW_SYS_TMO | 1-65535 | 120 | Yes |
| SHADOW_SYS_UNIT | 0-9999 | 0 | No |
| SHADOW_SYS_WAIT | 1-65535 | 480 | Yes |



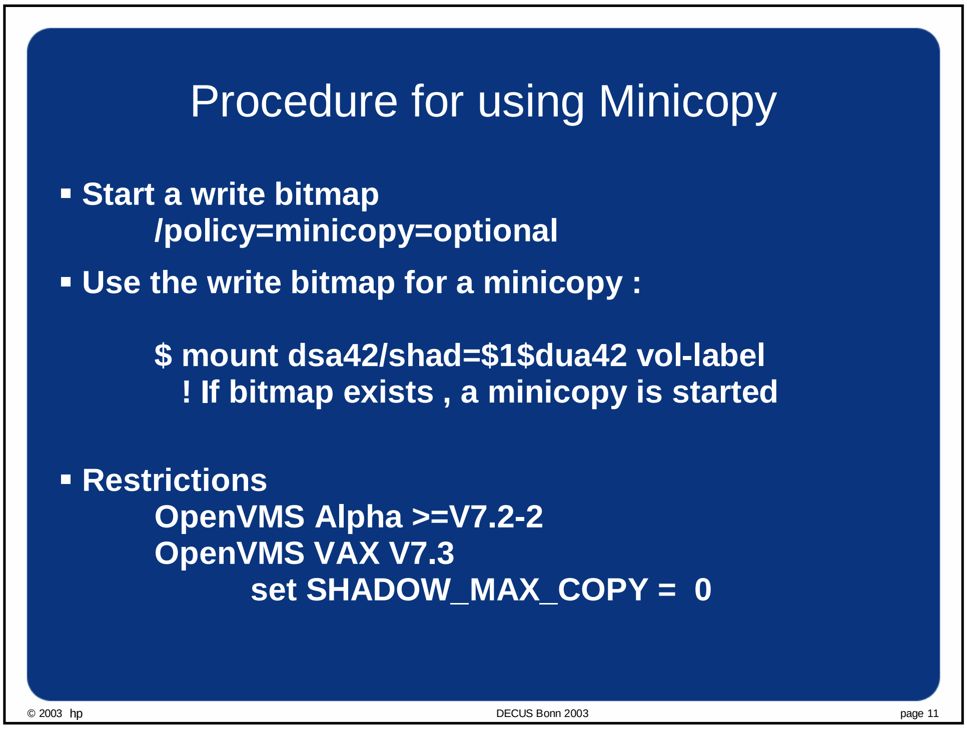
## Minicopy for Backing up Data

- What is Minicopy ?

  Ensures that the data on the shadow set member(s), when returned to the shadow set, is identical to the data on the shadow set.

- Specified at mount/dismount time

  A write bitmap is created and subsequent writes are recorded. ONLY the LBN of the associated writes are recorded !
  1 bit in the write bitmap corresponds to 127 disk blocks.

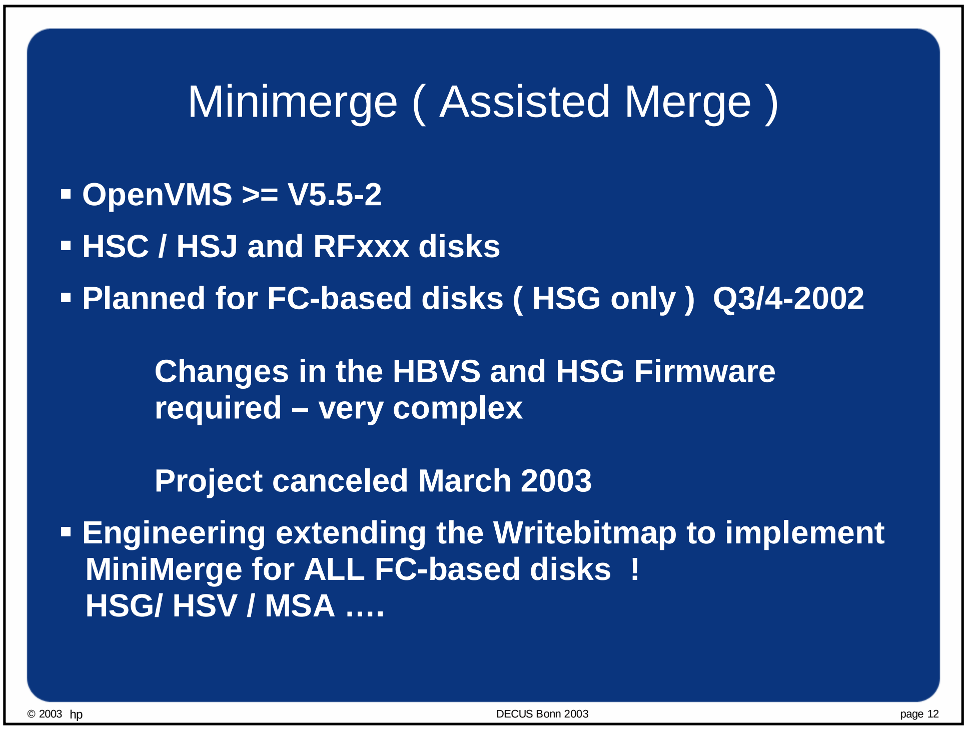


## Procedure for using Minicopy

- Start a write bitmap
  /policy=minicopy=optional
- Use the write bitmap for a minicopy :

```
$ mount dsa42/shad=$1$dua42 vol-label
  ! If bitmap exists , a minicopy is started
```

- Restrictions
  OpenVMS Alpha >=V7.2-2
  OpenVMS VAX V7.3
  set SHADOW_MAX_COPY =  0



## Minimerge ( Assisted Merge )

- OpenVMS >= V5.5-2
- HSC / HSJ and RFxxx disks
- Planned for FC-based disks ( HSG only ) Q3/4-2002

  Changes in the HBVS and HSG Firmware required – very complex

  Project canceled March 2003
- Engineering extending the Writebitmap to implement MiniMerge for ALL FC-based disks !
  HSG/ HSV / MSA ….

# Shadowing Performance

Shadowing is primarily an availability tool, but can often improve performance as well

Some shadowing operations can decrease performance (e.g. merges, minicopy )

# Shadow Copy/Merge Performance: Why Does It Matter?

Shadow copies or merges can generate a high data rate on inter-site links

Excessive shadow-copy time increases Mean Time To Repair (MTTR) after a site failure or outage

Acceptable shadow full-copy times and link costs will be the major factors in selecting inter-site link(s) for multi-site clusters

Shadow merges can have a significant adverse effect on application performance

# Shadowing Between Sites in Multi-Site Clusters

Because:

1) Inter-site latency is typically much greater than intra-site latency, at least if there is any significant distance between sites, and

2) Direct operations are a minimum of 1-3 ms lower in latency than MSCP-served operations, even when the inter-site distance is small,

It is most efficient to direct Read operations to the local disks, not remote disks

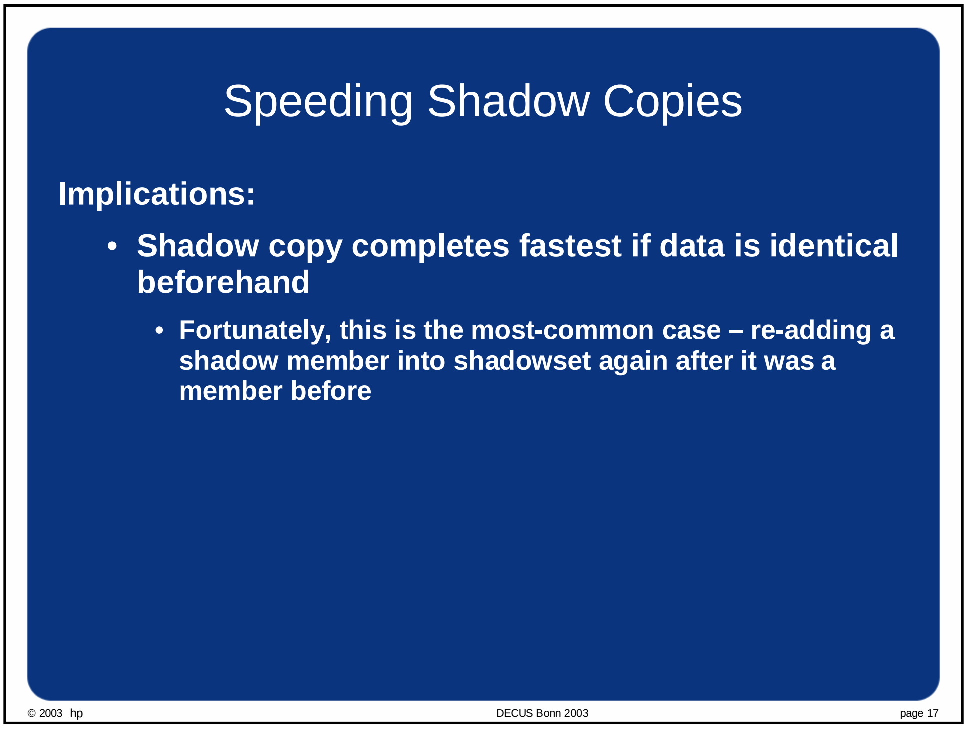
- Write operations have to go to all disks in a shadowset, remote as well as local members
- Usage of the $ set dev /READ_COST and /SITE qualifiers is recommended

# Shadow Copy Algorithm

Host-Based Volume Shadowing full-copy algorithm is non-intuitive:

1. Read from source disk
2. Do Compare operation with target disk
3. If data is different, write to target disk, then go to Step 1.

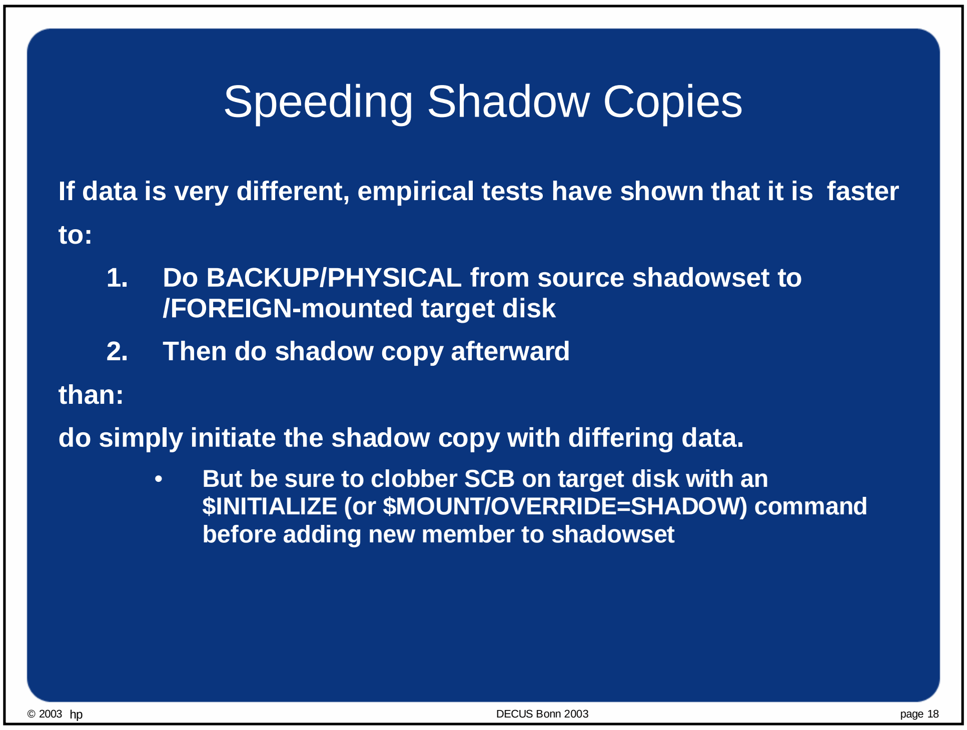
Shadow merges use a similar algorithm.



## Speeding Shadow Copies

Implications:

- Shadow copy completes fastest if data is identical beforehand
  - Fortunately, this is the most-common case – re-adding a shadow member into shadowset again after it was a member before



## Speeding Shadow Copies

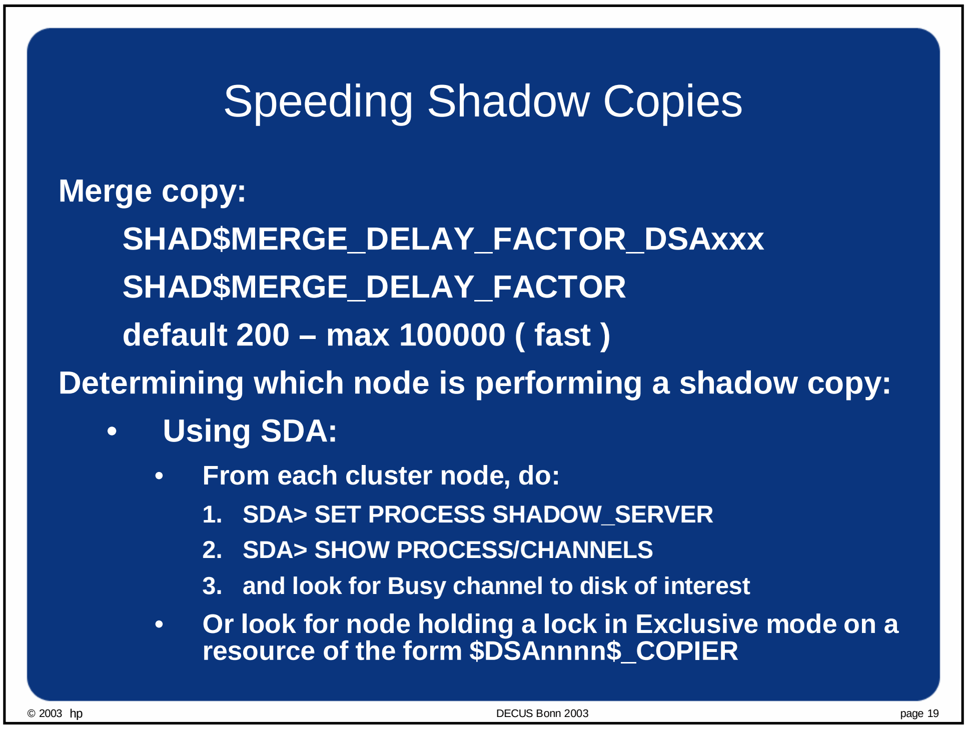
If data is very different, empirical tests have shown that it is faster to:

1. Do BACKUP/PHYSICAL from source shadowset to /FOREIGN-mounted target disk
2. Then do shadow copy afterward

than:

do simply initiate the shadow copy with differing data.

- But be sure to clobber SCB on target disk with an $INITIALIZE (or $MOUNT/OVERRIDE=SHADOW) command before adding new member to shadowset



# Speeding Shadow Copies

Merge copy:

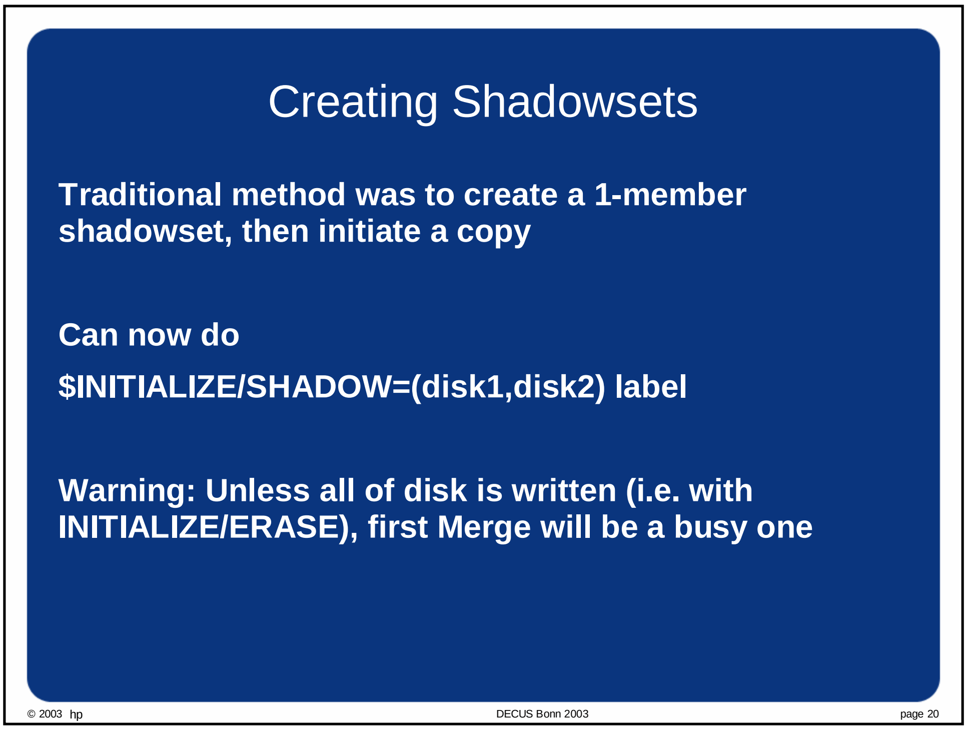
SHAD$MERGE_DELAY_FACTOR_DSAxxx

SHAD$MERGE_DELAY_FACTOR

default 200 – max 100000 ( fast )

Determining which node is performing a shadow copy:

- Using SDA:
  - From each cluster node, do:
    1. SDA> SET PROCESS SHADOW_SERVER
    2. SDA> SHOW PROCESS/CHANNELS
    3. and look for Busy channel to disk of interest
  - Or look for node holding a lock in Exclusive mode on a resource of the form $DSAnnnn$_COPIER



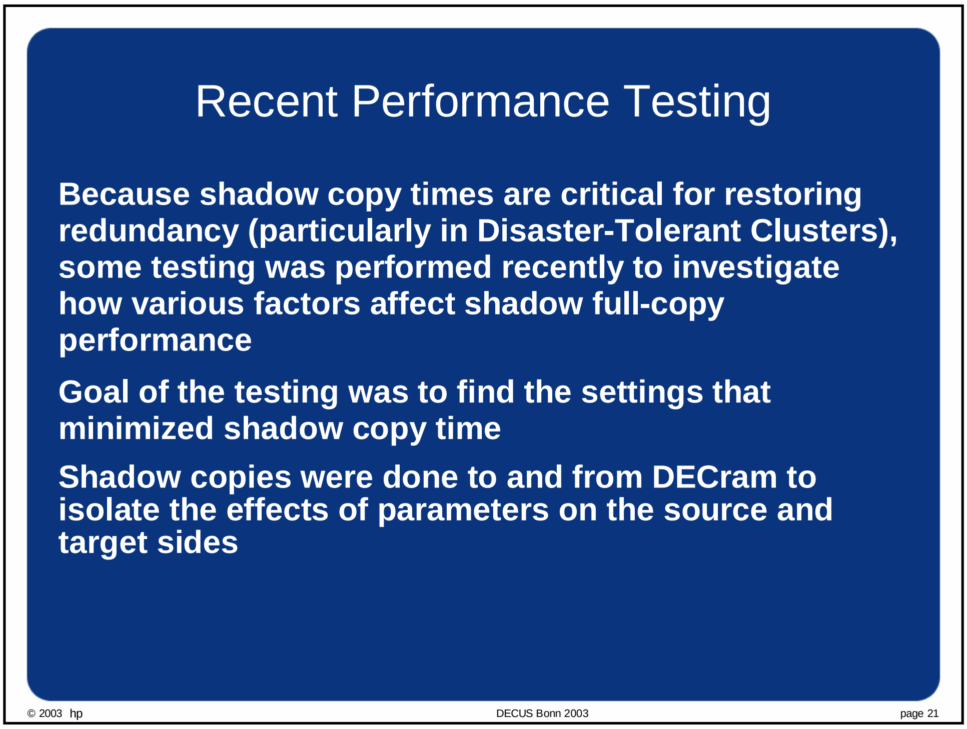
# Creating Shadowsets

Traditional method was to create a 1-member shadowset, then initiate a copy

Can now do

$INITIALIZE/SHADOW=(disk1,disk2) label

Warning: Unless all of disk is written (i.e. with INITIALIZE/ERASE), first Merge will be a busy one

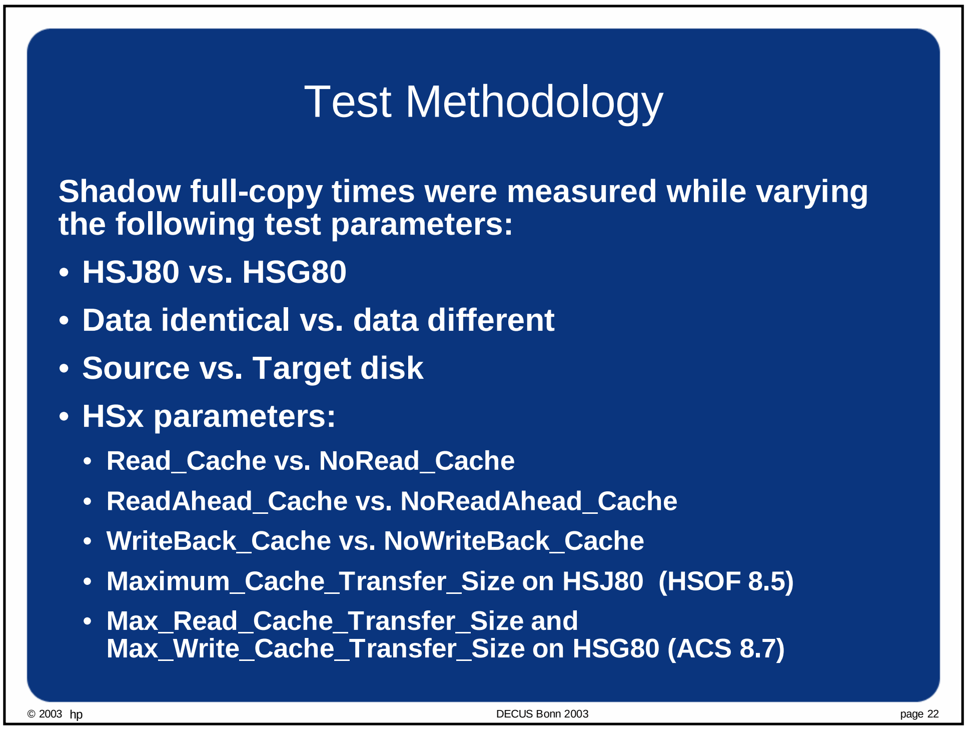


# Recent Performance Testing

Because shadow copy times are critical for restoring redundancy (particularly in Disaster-Tolerant Clusters), some testing was performed recently to investigate how various factors affect shadow full-copy performance

Goal of the testing was to find the settings that minimized shadow copy time

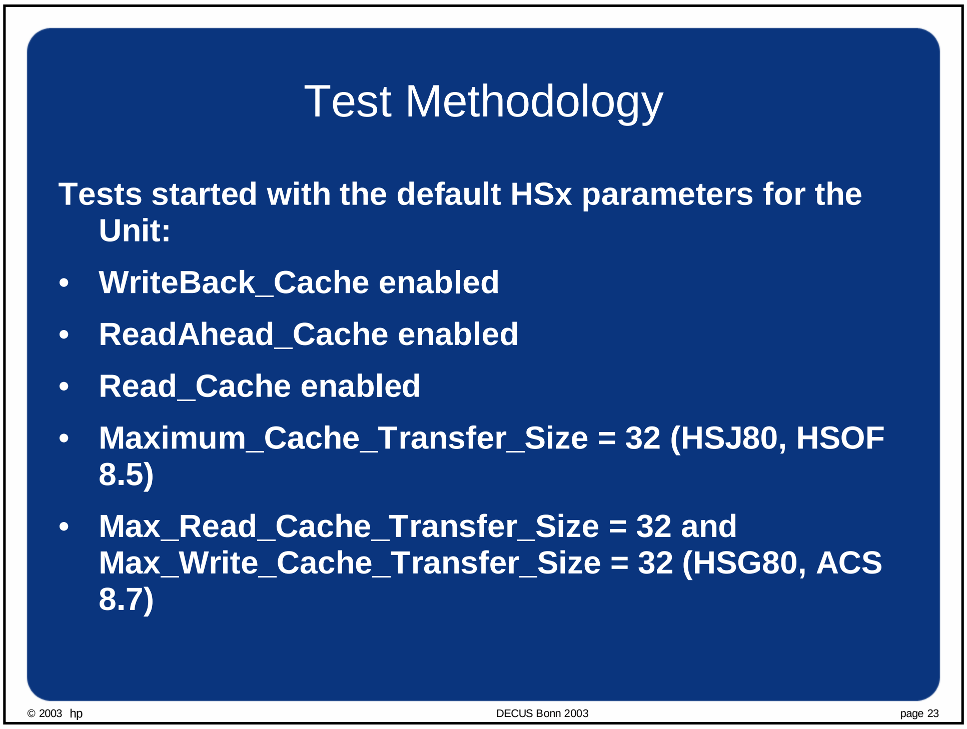
Shadow copies were done to and from DECram to isolate the effects of parameters on the source and target sides



# Test Methodology

Shadow full-copy times were measured while varying the following test parameters:

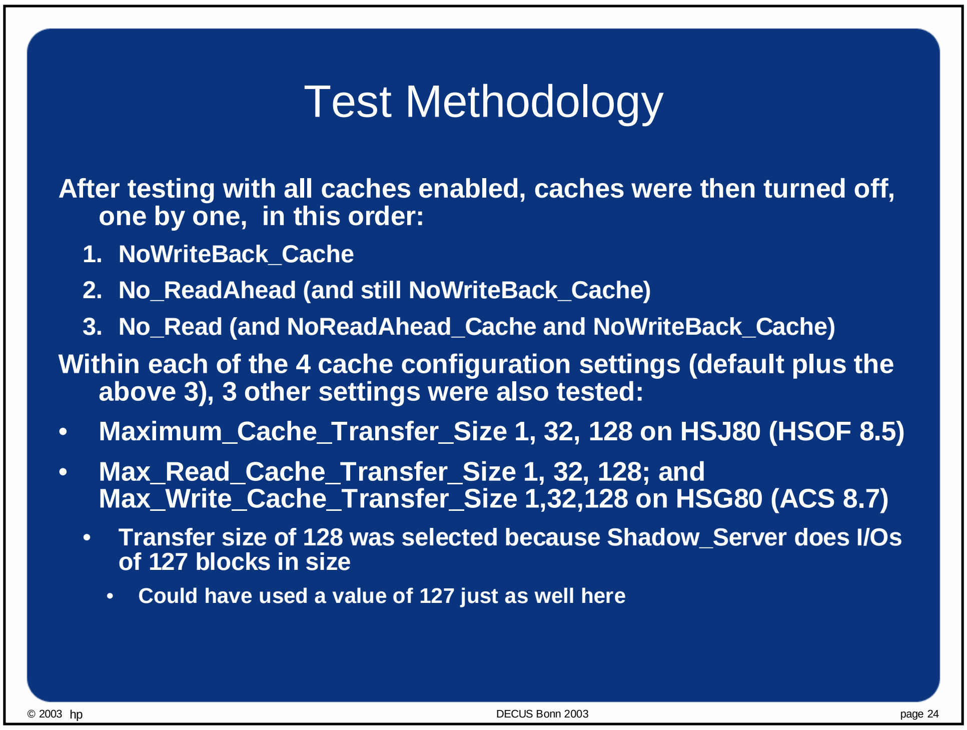
- HSJ80 vs. HSG80
- Data identical vs. data different
- Source vs. Target disk
- HSx parameters:
  - Read_Cache vs. NoRead_Cache
  - ReadAhead_Cache vs. NoReadAhead_Cache
  - WriteBack_Cache vs. NoWriteBack_Cache
  - Maximum_Cache_Transfer_Size on HSJ80 (HSOF 8.5)
  - Max_Read_Cache_Transfer_Size and Max_Write_Cache_Transfer_Size on HSG80 (ACS 8.7)



## Test Methodology

Tests started with the default HSx parameters for the Unit:

- WriteBack_Cache enabled
- ReadAhead_Cache enabled
- Read_Cache enabled
- Maximum_Cache_Transfer_Size = 32 (HSJ80, HSOF 8.5)
- Max_Read_Cache_Transfer_Size = 32 and Max_Write_Cache_Transfer_Size = 32 (HSG80, ACS 8.7)



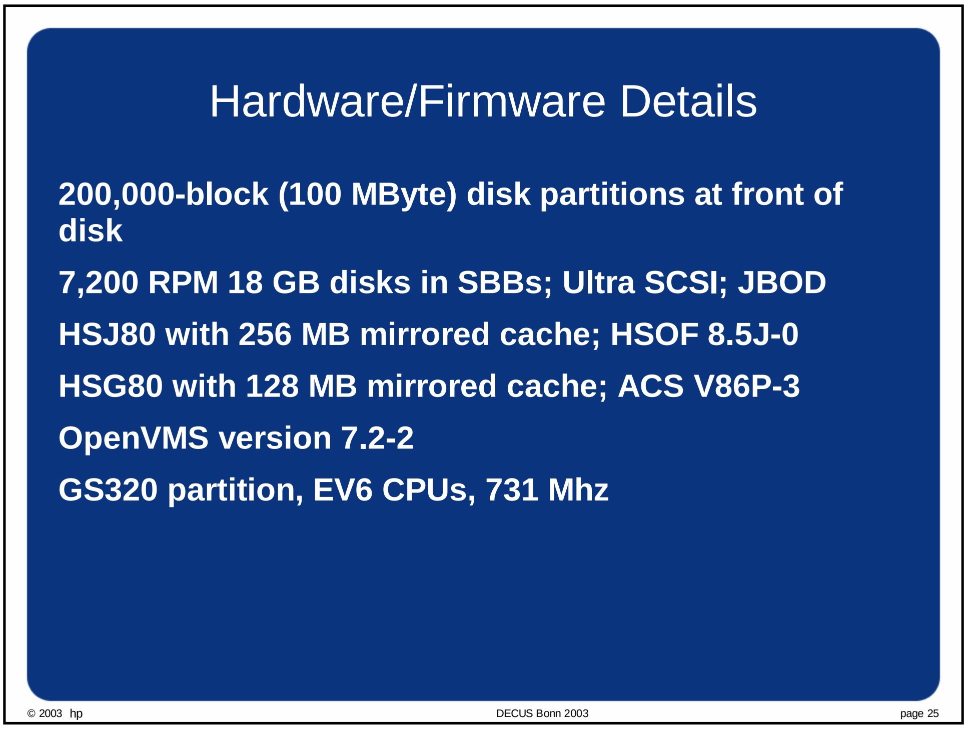
## Test Methodology

After testing with all caches enabled, caches were then turned off, one by one, in this order:

1. NoWriteBack_Cache
2. No_ReadAhead (and still NoWriteBack_Cache)
3. No_Read (and NoReadAhead_Cache and NoWriteBack_Cache)

Within each of the 4 cache configuration settings (default plus the above 3), 3 other settings were also tested:

- Maximum_Cache_Transfer_Size 1, 32, 128 on HSJ80 (HSOF 8.5)
- Max_Read_Cache_Transfer_Size 1, 32, 128; and Max_Write_Cache_Transfer_Size 1,32,128 on HSG80 (ACS 8.7)
  - Transfer size of 128 was selected because Shadow_Server does I/Os of 127 blocks in size
    - Could have used a value of 127 just as well here



## Hardware/Firmware Details

200,000-block (100 MByte) disk partitions at front of disk

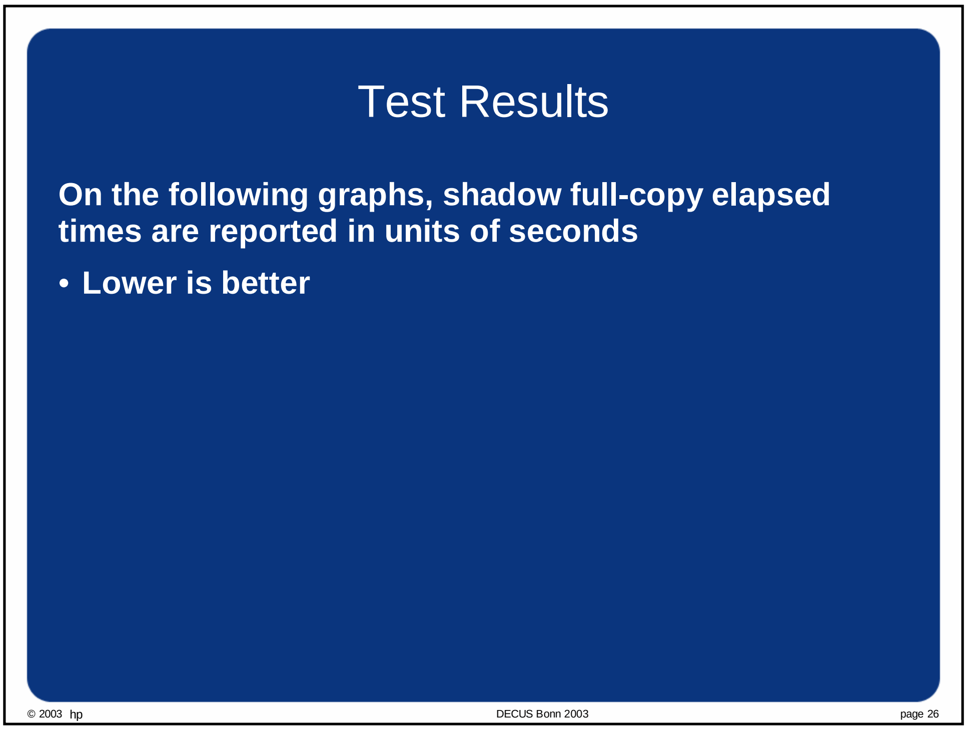
7,200 RPM 18 GB disks in SBBs; Ultra SCSI; JBOD

HSJ80 with 256 MB mirrored cache; HSOF 8.5J-0

HSG80 with 128 MB mirrored cache; ACS V86P-3

OpenVMS version 7.2-2

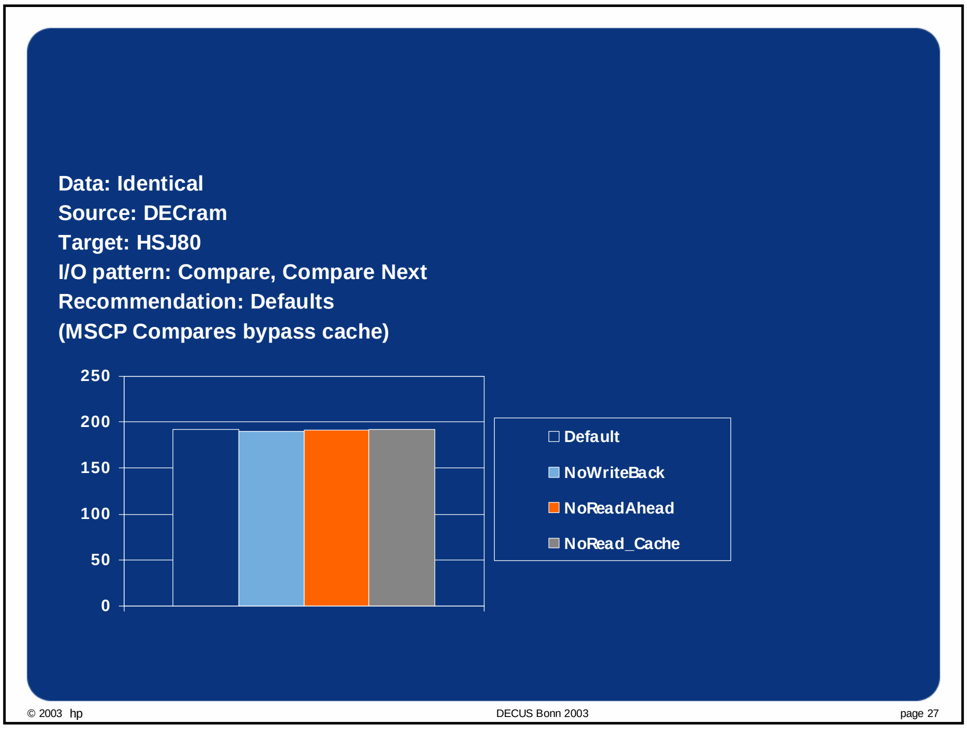
GS320 partition, EV6 CPUs, 731 Mhz



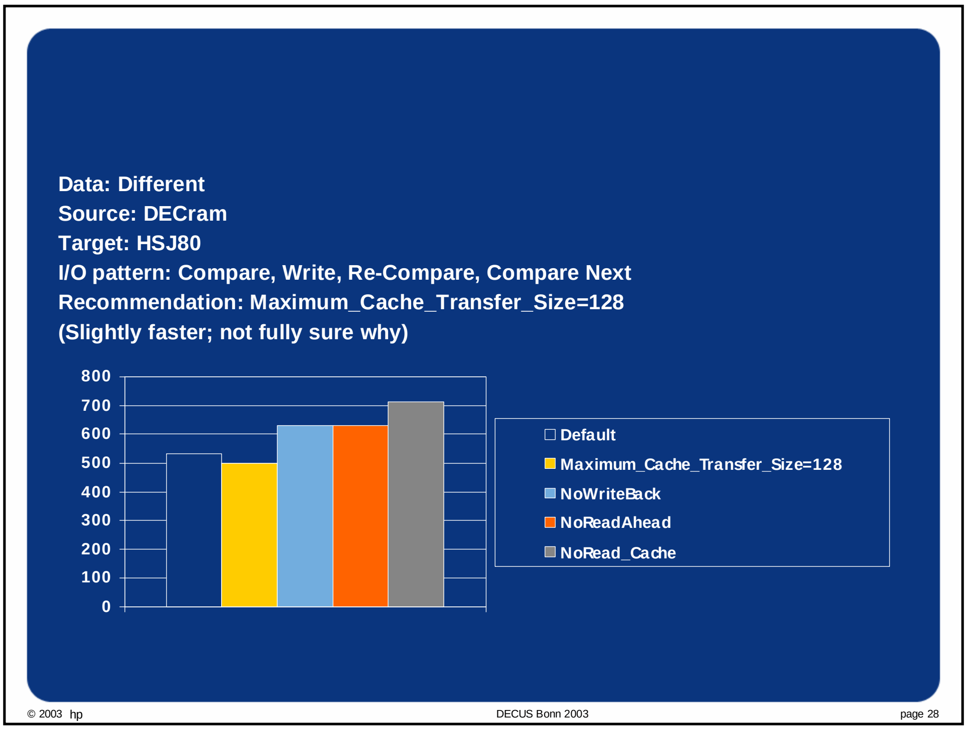
## Test Results

On the following graphs, shadow full-copy elapsed times are reported in units of seconds

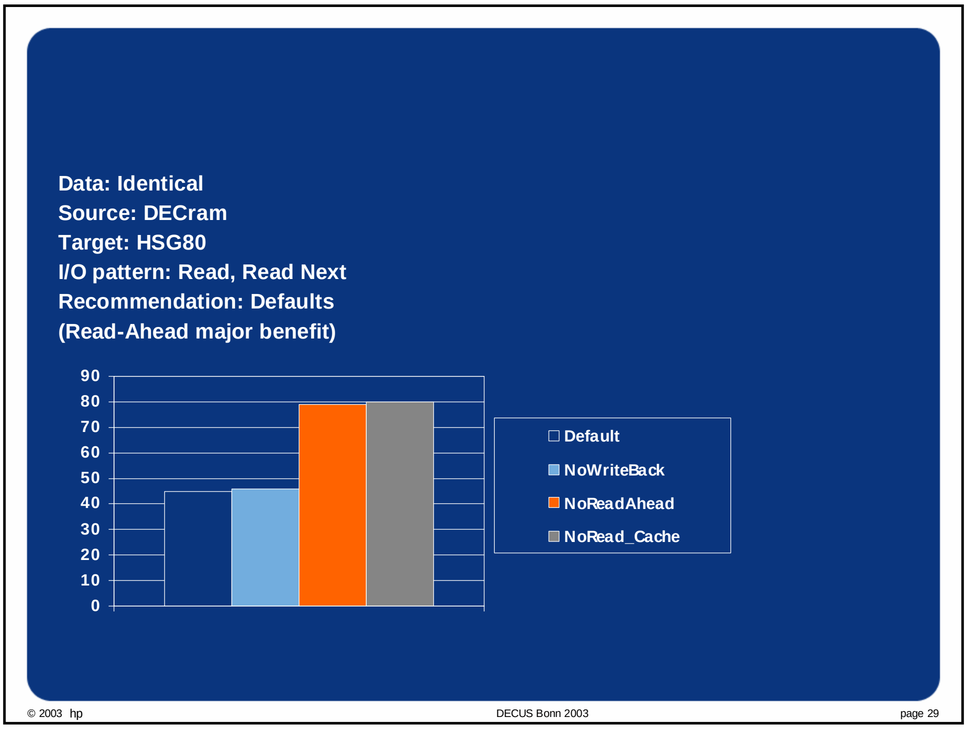
- Lower is better



Data: Identical
Source: DECram
Target: HSJ80
I/O pattern: Compare, Compare Next
Recommendation: Defaults
(MSCP Compares bypass cache)

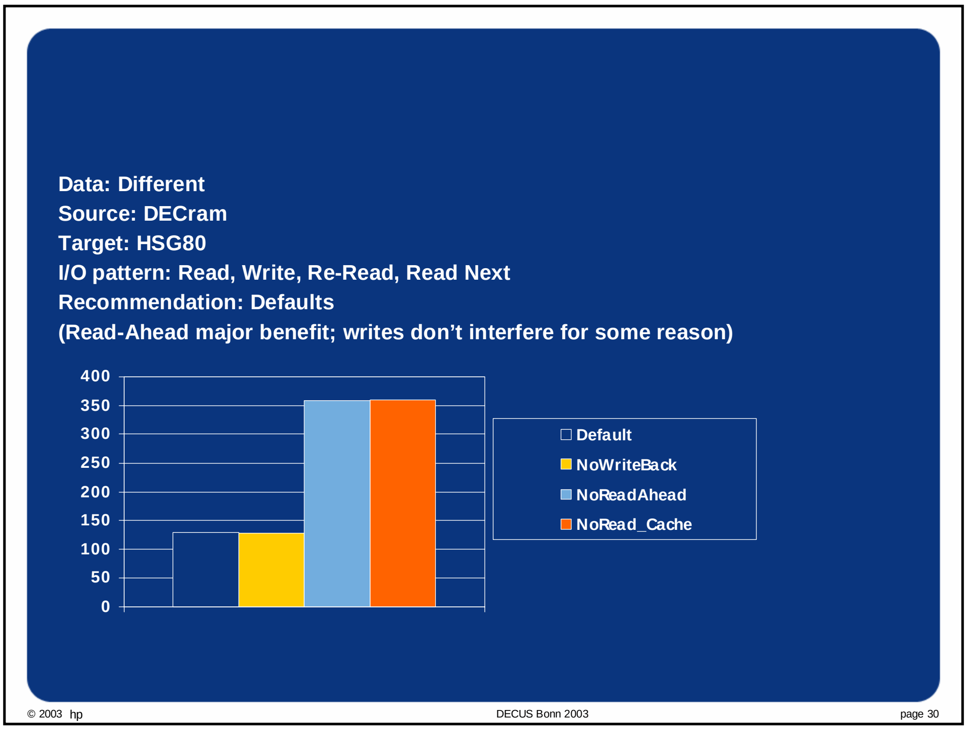


Data: Different
Source: DECram
Target: HSJ80
I/O pattern: Compare, Write, Re-Compare, Compare Next
Recommendation: Maximum_Cache_Transfer_Size=128
(Slightly faster; not fully sure why)



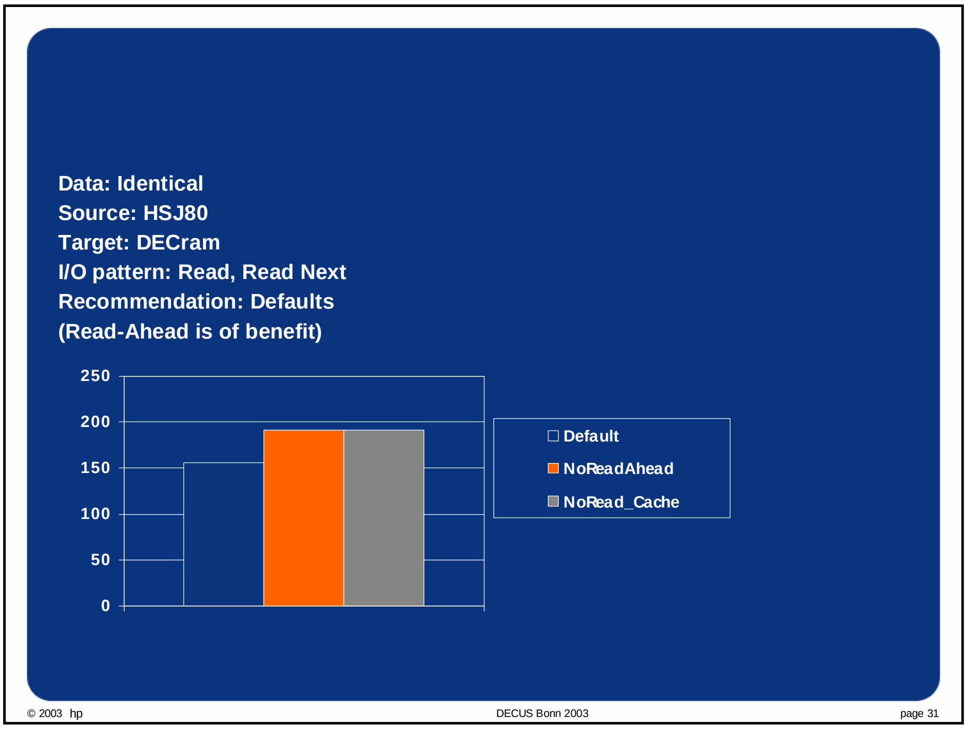
Data: Identical
Source: DECram
Target: HSG80
I/O pattern: Read, Read Next
Recommendation: Defaults
(Read-Ahead major benefit)

© 2003 hp DECUS Bonn 2003 page 29

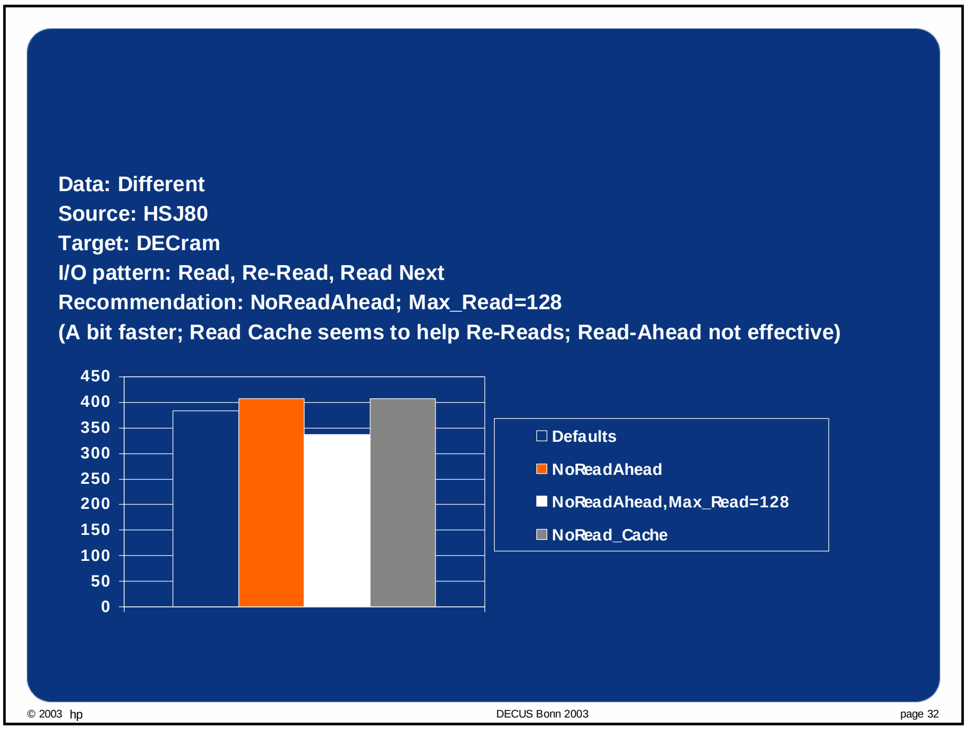


Data: Different
Source: DECram
Target: HSG80
I/O pattern: Read, Write, Re-Read, Read Next
Recommendation: Defaults
(Read-Ahead major benefit; writes don’t interfere for some reason)

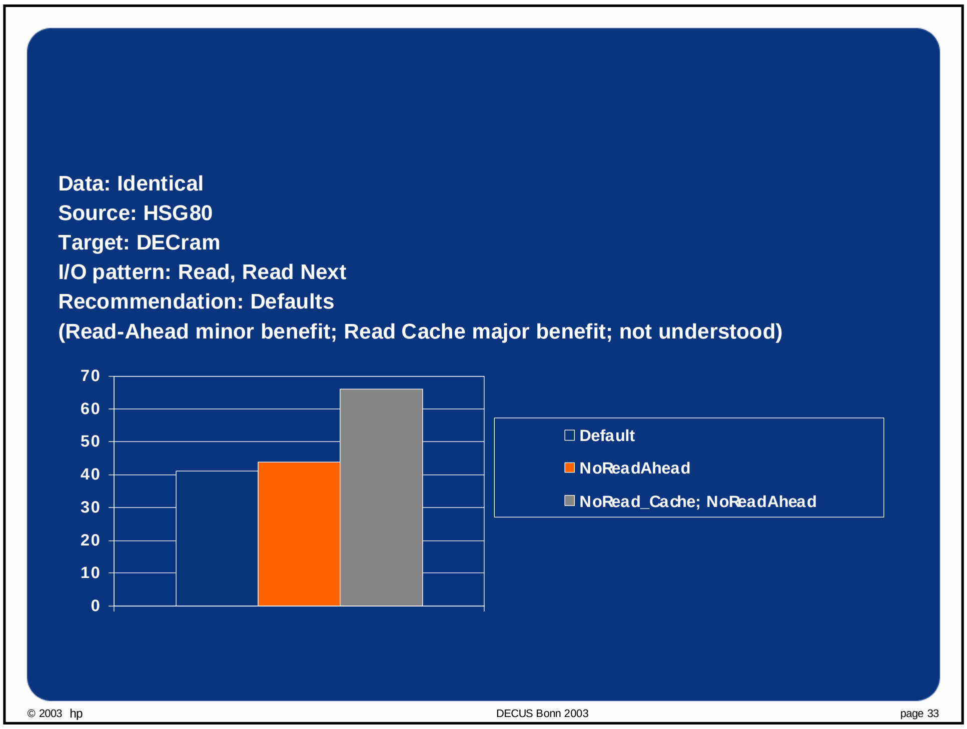
© 2003 hp DECUS Bonn 2003 page 30



Data: Identical
Source: HSJ80
Target: DECram
I/O pattern: Read, Read Next
Recommendation: Defaults
(Read-Ahead is of benefit)

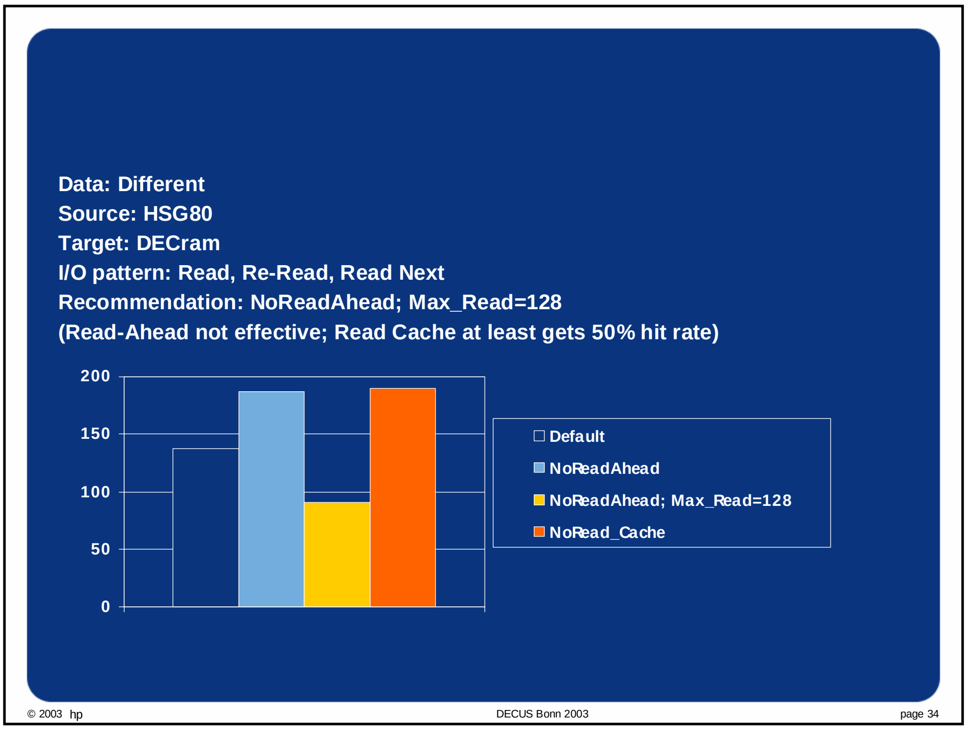


Data: Different
Source: HSJ80
Target: DECram
I/O pattern: Read, Re-Read, Read Next
Recommendation: NoReadAhead; Max_Read=128
(A bit faster; Read Cache seems to help Re-Reads; Read-Ahead not effective)



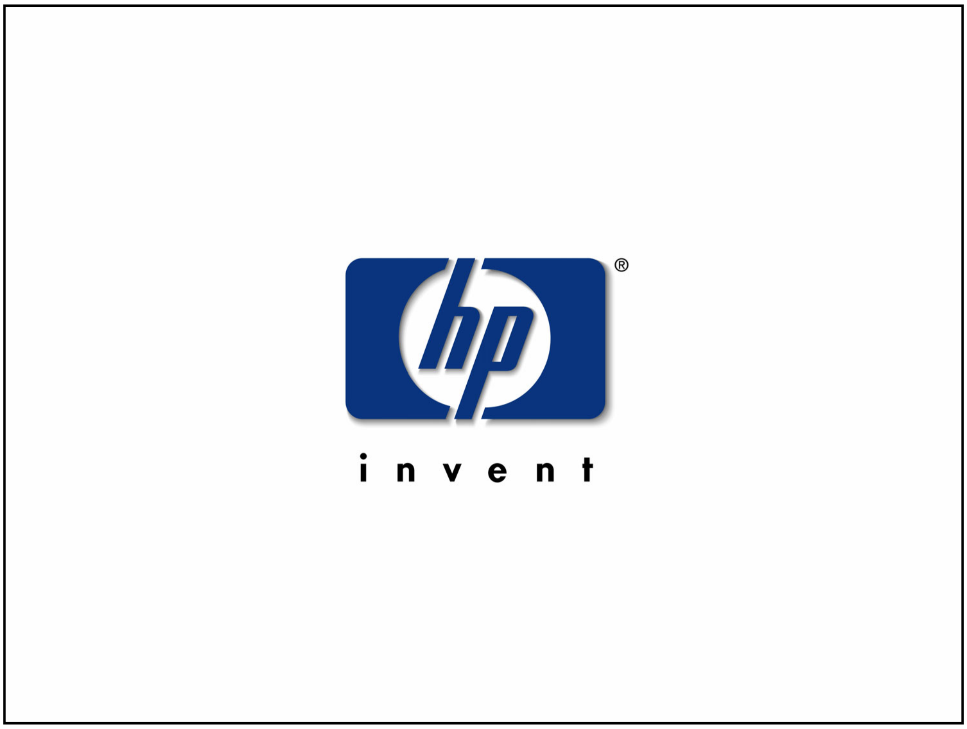
Data: Identical
Source: HSG80
Target: DECram
I/O pattern: Read, Read Next
Recommendation: Defaults
(Read-Ahead minor benefit; Read Cache major benefit; not understood)

© 2003 hp
DECUS Bonn 2003
page 33



Data: Different
Source: HSG80
Target: DECram
I/O pattern: Read, Re-Read, Read Next
Recommendation: NoReadAhead; Max_Read=128
(Read-Ahead not effective; Read Cache at least gets 50% hit rate)

© 2003 hp
DECUS Bonn 2003
page 34



hp®

invent