# Transition to Transitions
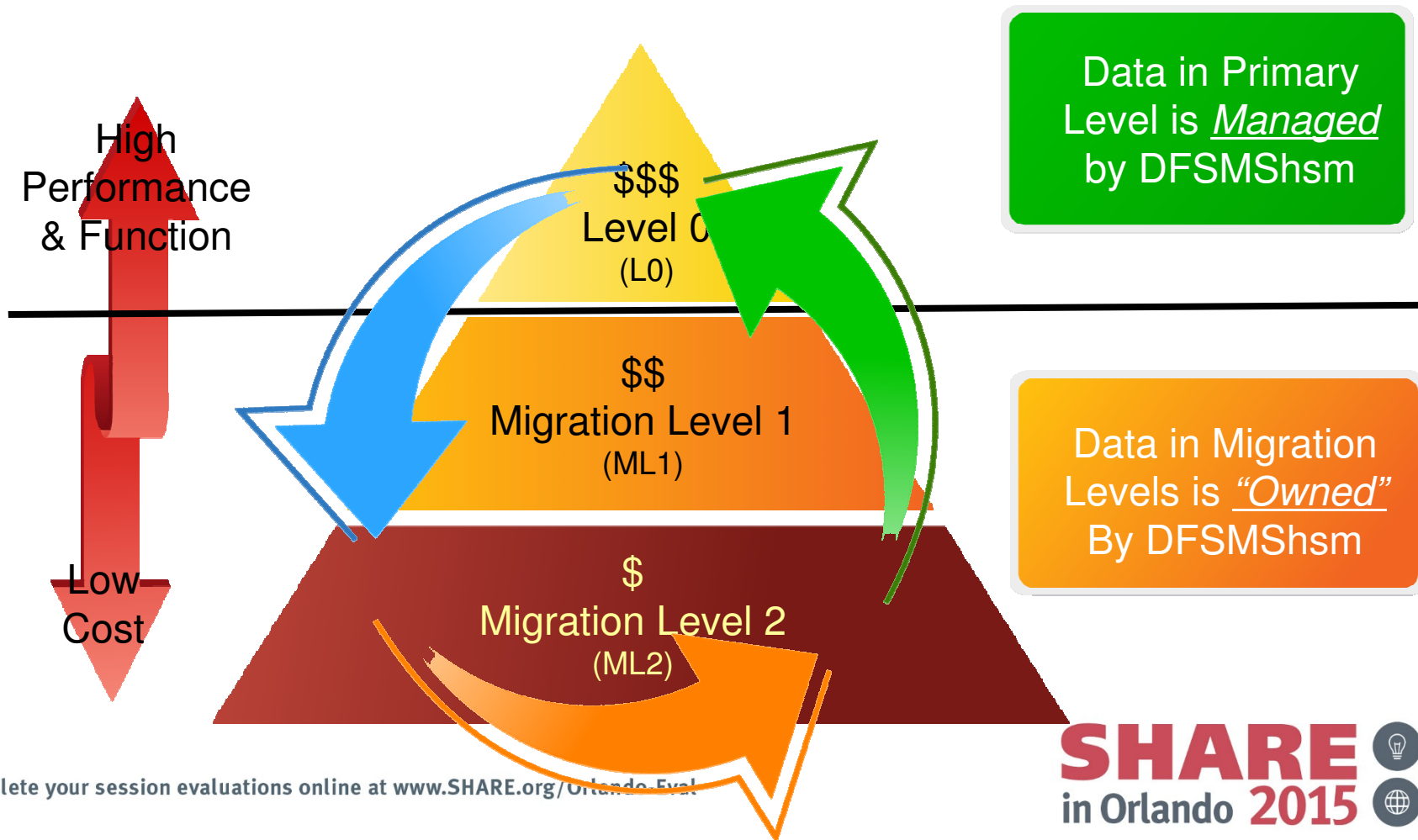## Session 17804

Glenn Wilcock
*DFSMShsm Architect*
*IBM*

# Agenda

- Overview
- Storage Hierarchy
- Management Class
- ACS Routines
- Processing
- Use Case
- V2R2 Enhancement
- Looking Forward…

# Background – DFSMS Today

**The *classic* DFSMS storage hierarchy (37 years old)**



High Performance & Function

Low Cost

$$$ Level 0 (L0)

$$ Migration Level 1 (ML1)

$ Migration Level 2 (ML2)

Data in Primary Level is *Managed* by DFSMShsm

Data in Migration Levels is *"Owned"* By DFSMShsm

# DFSMS Storage Tiers z/OS V2R1

**Automated, policy-based space management that moves data from tier to tier within the Primary (Level 0) Hierarchy**
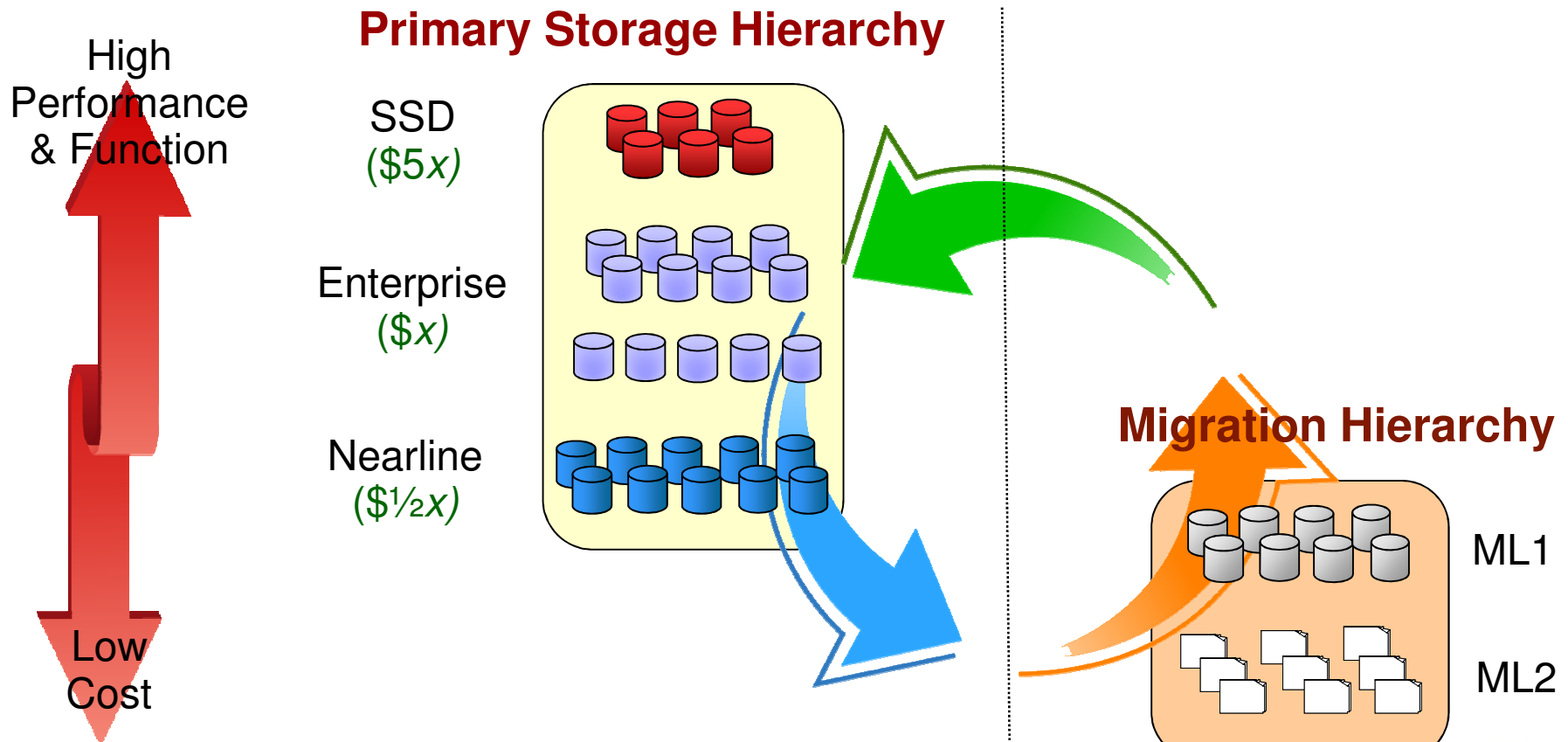
- ✓ Automated movement provided via the existing DFSMShsm Space Management function
    - Movement is referred to as a 'Class Transition'
    - Data remains in its original format and can be immediately accessed after the movement is complete
- ✓ Policies implemented via existing Class Transition policies and updated Management Class policies
- ✓ Enhanced support for DB2, CICS and zFS data
    - Open data sets are temporarily closed to enable movement

# Overview

- The function of *DFSMShsm Space Management* processing is to use *policy-based automation* to ensure that volumes within the Primary Storage Hierarchy have enough *free space* for new data and to ensure that data is stored at the *lowest acceptable tier in the Storage Hierarchy*
  - This is accomplished through
    - Data set expiration
    - Migration of unreferenced data to the Migration Hierarchy
    - "Class Transitions" within the Primary Hierarchy

- "Class Transition" processing is integrated into the existing DFSMShsm *Space Management* functions
  - Primary Space Management
  - On-Demand Migration
    - *New function introduced in V1R13. Performs space management on a volume as soon as it goes over its high threshold. Replacement for on-the-hour Interval Migration processing.*
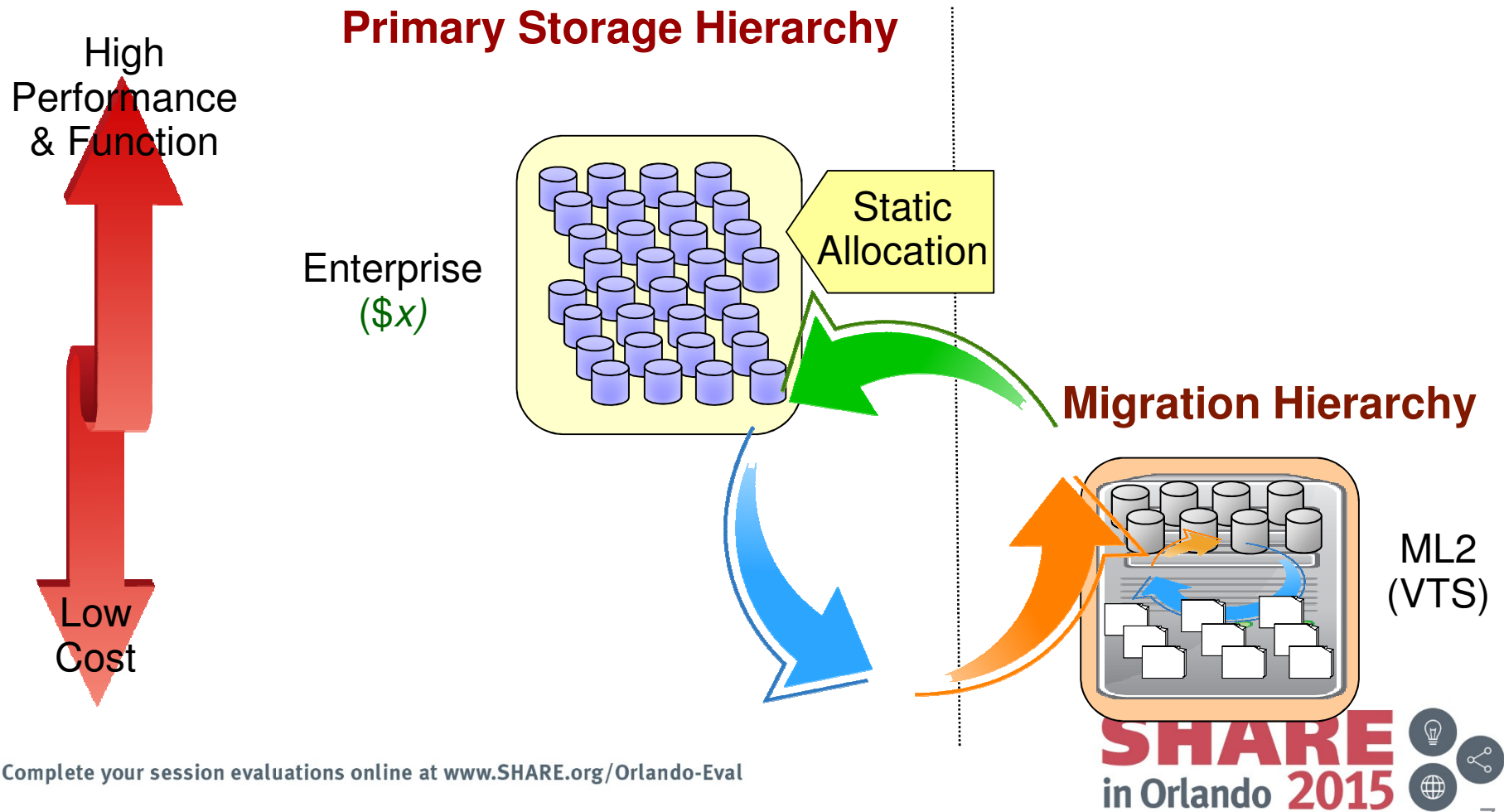  - Interval Migration

# Overview

The *classic* DFSMS storage hierarchy is modified to represent *distinct* Primary and Migration Hierarchies...

**Primary Storage Hierarchy**

High Performance & Function

Low Cost

SSD ($5*x*)

Enterprise ($*x*)

Nearline ($½*x*)

**Migration Hierarchy**

ML1

ML2

# Overview

*Transition from a classic environment …*

**Primary Storage Hierarchy**

High
Performance
& Function

Low
Cost

Enterprise
*($x)*

Static
Allocation

**Migration Hierarchy**

ML2
(VTS)

# Overview

*… to a tiered environment*

**Data "Temperature"**

**Primary Storage Hierarchy**

Hot

Warm

Luke-warm

Cool

Cold

Frigid

Smart Tier 0:
*SSD / Enterprise*
*($x)*

Smart Tier 1:
*Enterprise / Nearline*
*($1/3x)*

Allocate

*Transition*
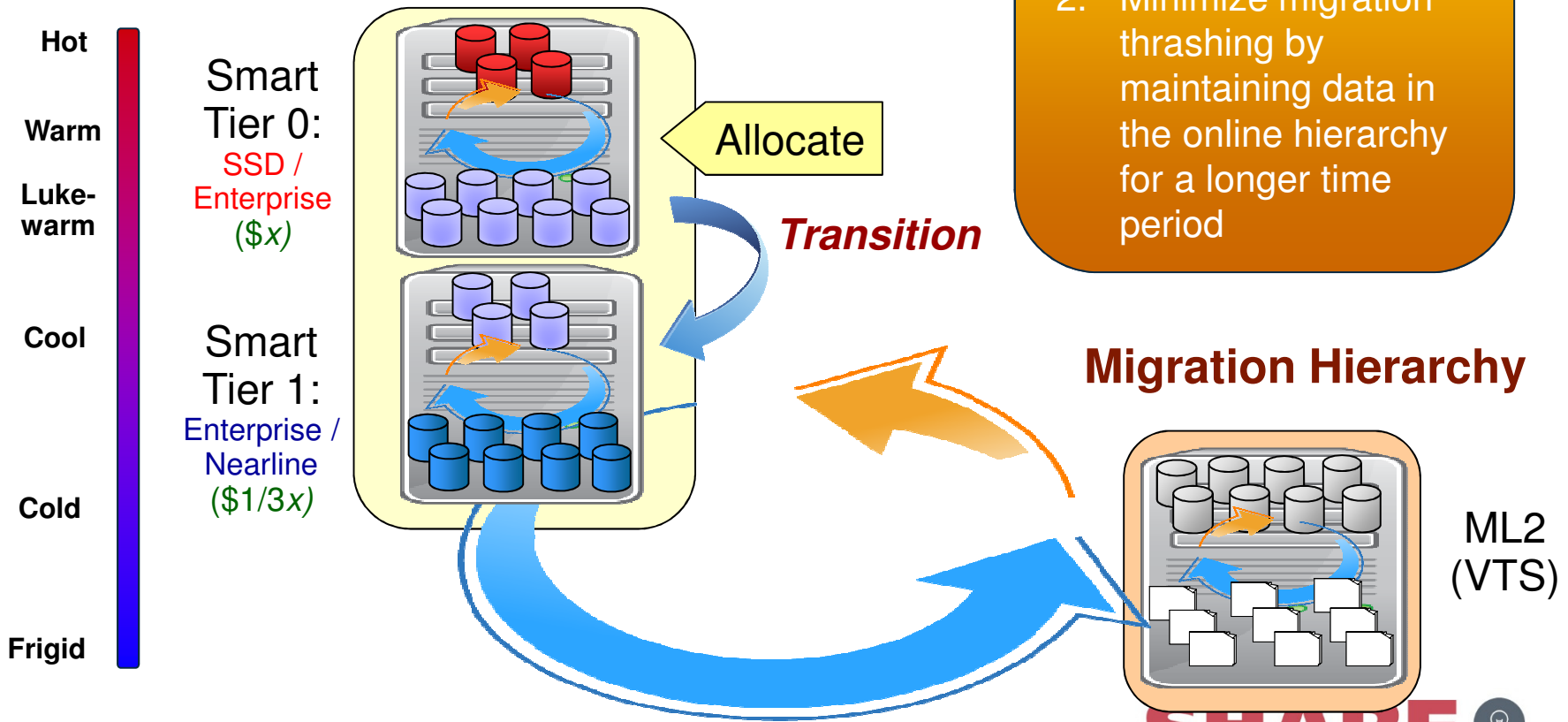
**Goals:**
1. Minimize the cost of storing data that is never migrated
2. Minimize migration thrashing by maintaining data in the online hierarchy for a longer time period

**Migration Hierarchy**

ML2 (VTS)

# Usage Notes

- **There is a distinction between using Migrate/Recall and Class Transitions**
  - When a data set is recalled, it will be returned to the class of storage as directed by the ACS routines, which would typically be higher than where a data set would reside after a transition
  - When a data set transitions to a lower class of storage, it will remain there until it is transitioned again or until it migrates

- *In order for FlashCopy to be used for a transition*, the movement must be within the same storage controller. This may be difficult to achieve.
  - ✓ FlashCopy will be discussed in detail later in the presentation

# Overview

As a data set ages and goes through cycles of activity and inactivity, it can go through many migration and recall iterations.
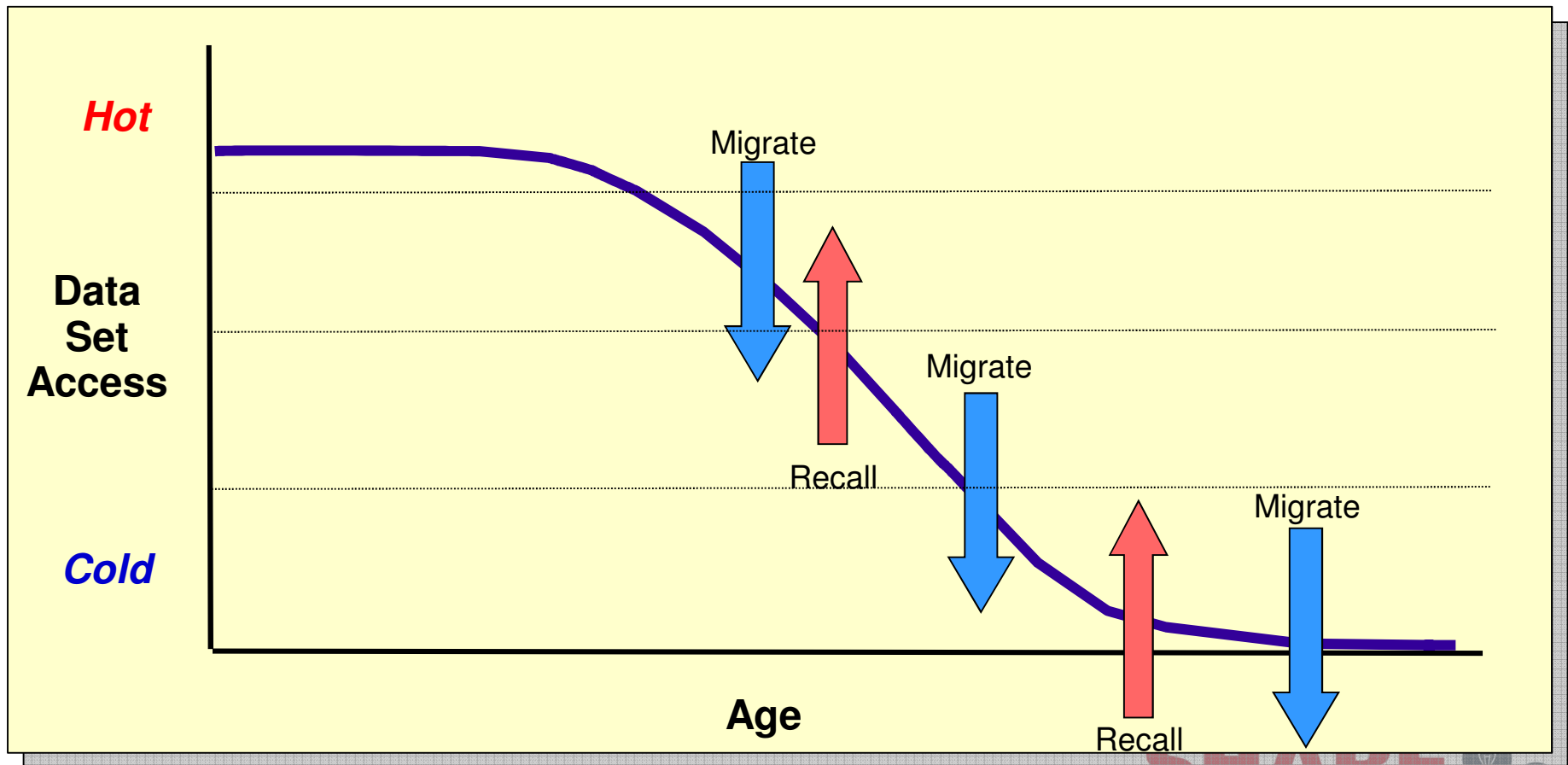
Complete your session evaluations online at www.SHARE.org/Orlando-Eval
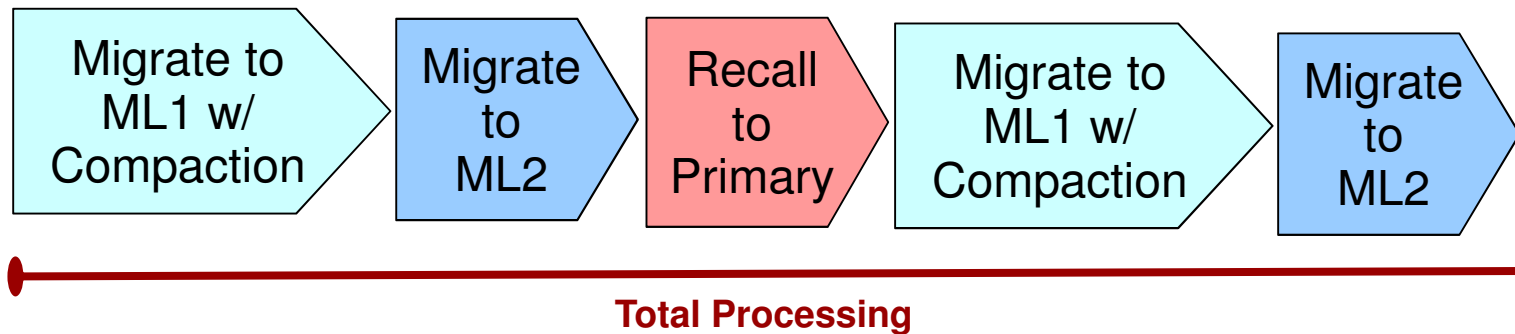
# Overview

The migration / recall iterations can be replaced with a single class transition and potentially single migration.
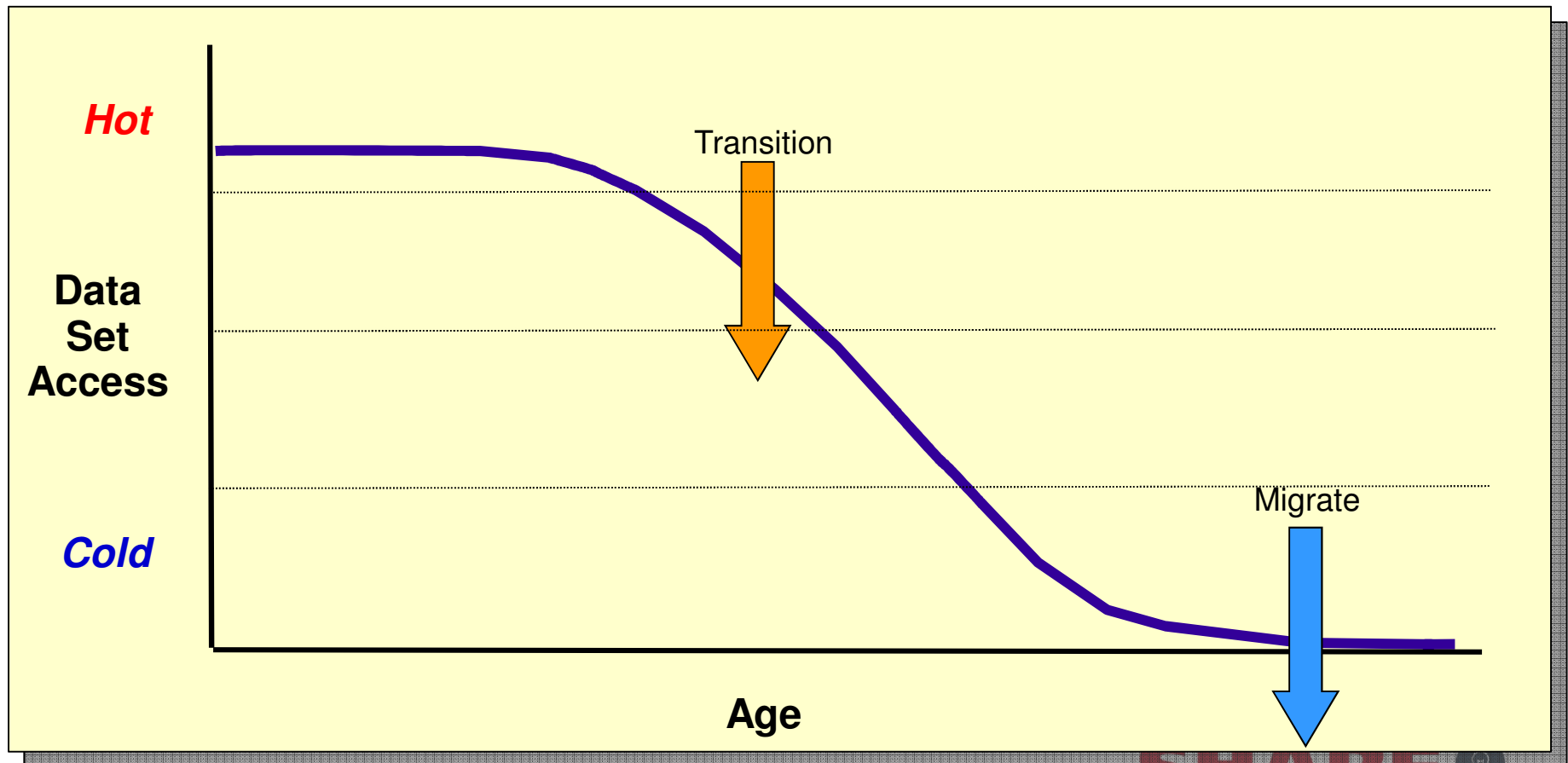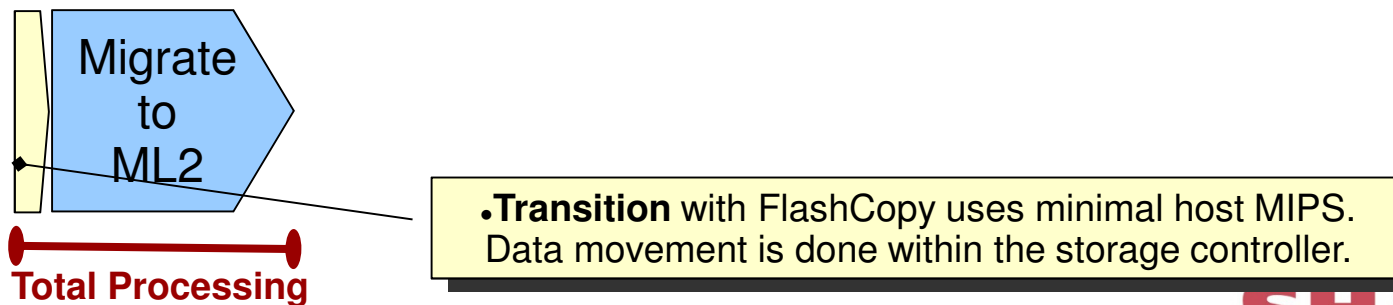
# Overview

## Contrast the Space Management processing required when ...

- A data set is migrated to ML1, then ML2, recalled and remigrated just once

```
Migrate to    →   Migrate   →   Recall   →   Migrate to   →   Migrate
ML1 w/            to            to           ML1 w/            to
Compaction        ML2           Primary      Compaction        ML2
```

**Total Processing**

- A data set is transitioned using FlashCopy and then migrated directly to ML2

```
Migrate
to
ML2
```

**Total Processing**

> • **Transition** with FlashCopy uses minimal host MIPS. Data movement is done within the storage controller.

# Use Cases

✓ **Use Case 1:** DB2 objects that are always open

  • When certain DB2 objects reach 6 months of age, they will be transitioned from Tier 0 to Tier 1 storage

  • Note: This is applicate to DB2 objects that are organized in a chronological technique

✓ **Use Case 2:** Minimize space management thrashing

  • After Tier 0 data sets are unreferenced for 45 days, they are transitioned to Tier 1 storage

  • After Tier 1 data sets are unreferenced for 366 days, they are migrated to ML2

# Storage Hierarchy

## Virtual Tape with no ML1

- Offload compression to tape control unit
- Single data movement for HSM (Eliminates ML1->ML2)
- Tape Controller manages multiple copies
- With optional tape back end, you still get the advantages of tape
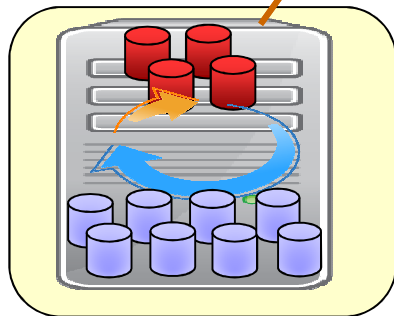
**Migration Hierarchy**
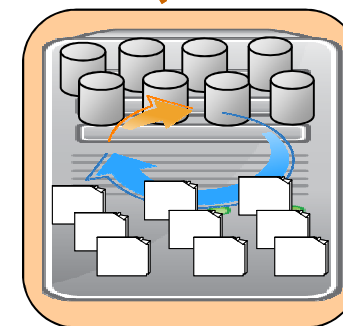
ML2 (VTS)

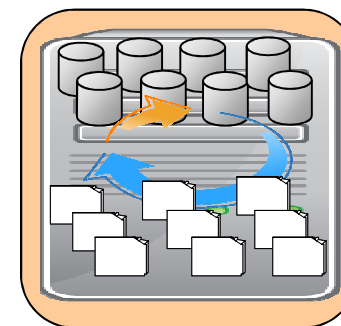# Storage Hierarchy

## Primary Storage Hierarchy

Smart
Tier 0:
SSD /
Enterprise
($x)

### Control Unit Tiering

- Disk control unit transparently moves data to the most appropriated tier based on its relative heat map
- Hottest data is placed on the fastest tier to maximize i/o

## Migration Hierarchy

ML2
(VTS)

# Storage Hierarchy

- Storage controller uses heat map to transparently move data to the most appropriate tier…

**Data Legend**

| | |
|---|---|
| 🟥 | Hot |
| 🟧 | Warm |
| 🟦 | Cool |
| 🟦 | Cold |

**Disk Legend**

| | |
|---|---|
| 🟥 | SSD |
| 🟦 | Nearline |

**Pool** ↔

**Pool** ↔

## VOLA

**Physical extents comprising logical volume**
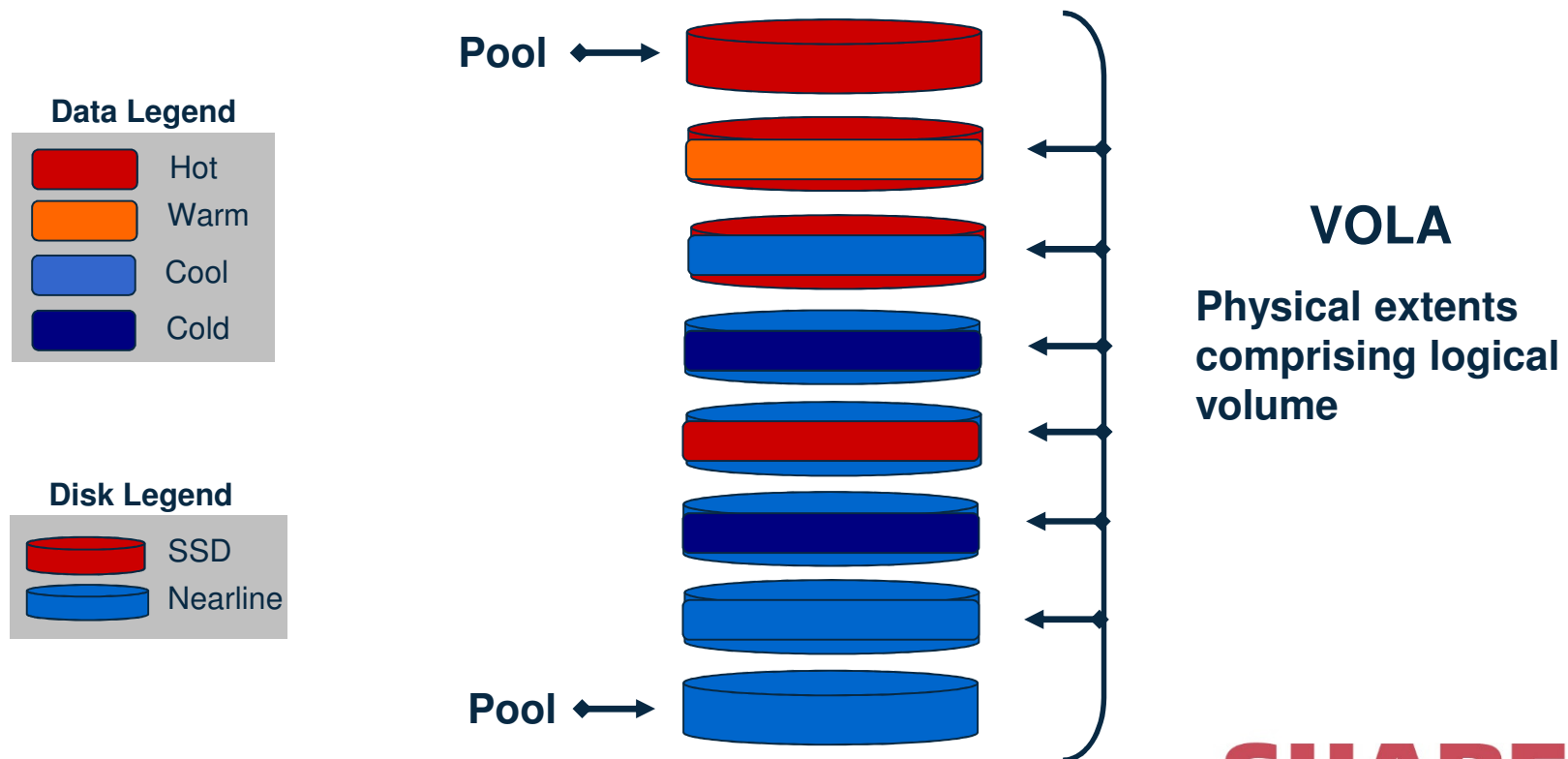
Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

# Storage Hierarchy

- For example…



> 3 x IOPS Improvement
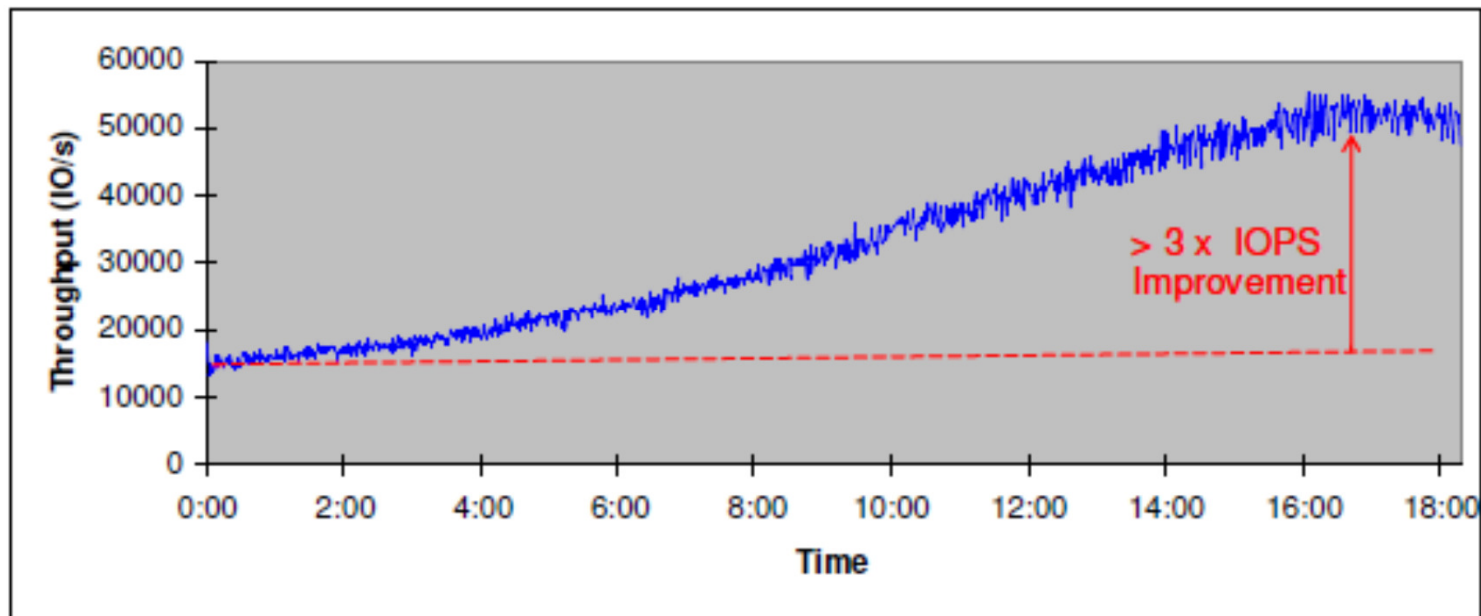
**Storage performance council, IBM DS8700, 2010**:  http://www.storageperformance.org/results/benchmark_results_spc1#a00092

# Storage Hierarchy

## Primary Storage Hierarchy

**Smart Tier 0:**
SSD / Enterprise
($x$)
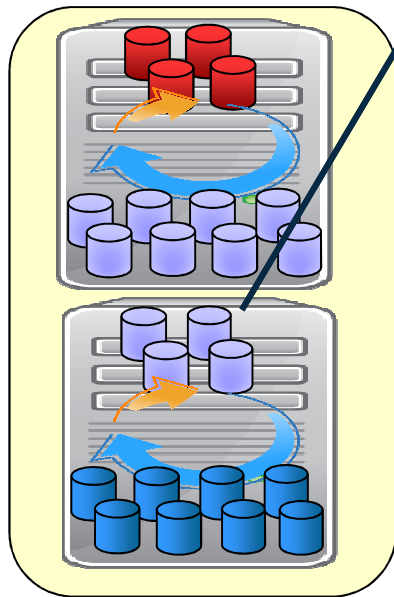
**Smart Tier 1:**
Enterprise / Nearline
($1/3x$)

## Second Tier

- Introduction of a lower cost secondary tier that is comprised mostly of nearline drives, but also has enterprise drives for the more active data

## Migration Hierarchy

ML2 (VTS)

# Storage Hierarchy

## Primary Storage Hierarchy

L0

L1

Allocate

*Transition*
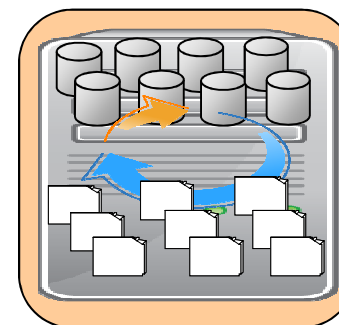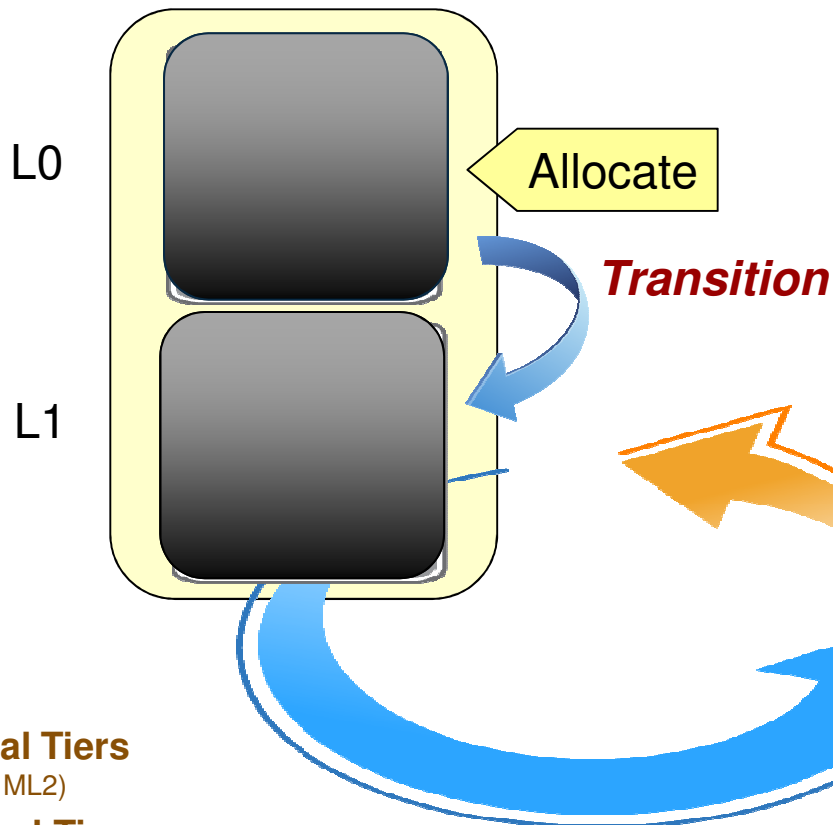
- Control unit tiering is transparent to HSM
- Consider these as 'Black Boxes' to HSM

## Migration Hierarchy
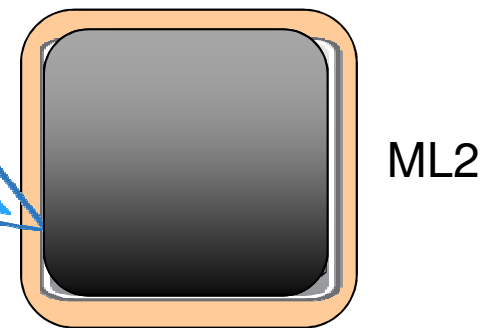
ML2

**3 Logical Tiers**
(L0, L1, ML2)

**5 Physical Tiers**
(SSD, Enterprise, Nearline, Virtual Tape, Physical tape)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

19

# Storage Hierarchy

- **Defining Tiers to DFSMS**
  - No change - Storage groups are collections of like-type devices
    - Control Unit boundaries are transparent
    - Mixed devices within a storage group will result in mixed results
      - SSD volumes can be 'preferred' via storage class

**Storage Group Name**

SSD / Enterprise

SGTIER0

Initial Allocation

ACS Routines

Enterprise / Nearline

SGTIER1
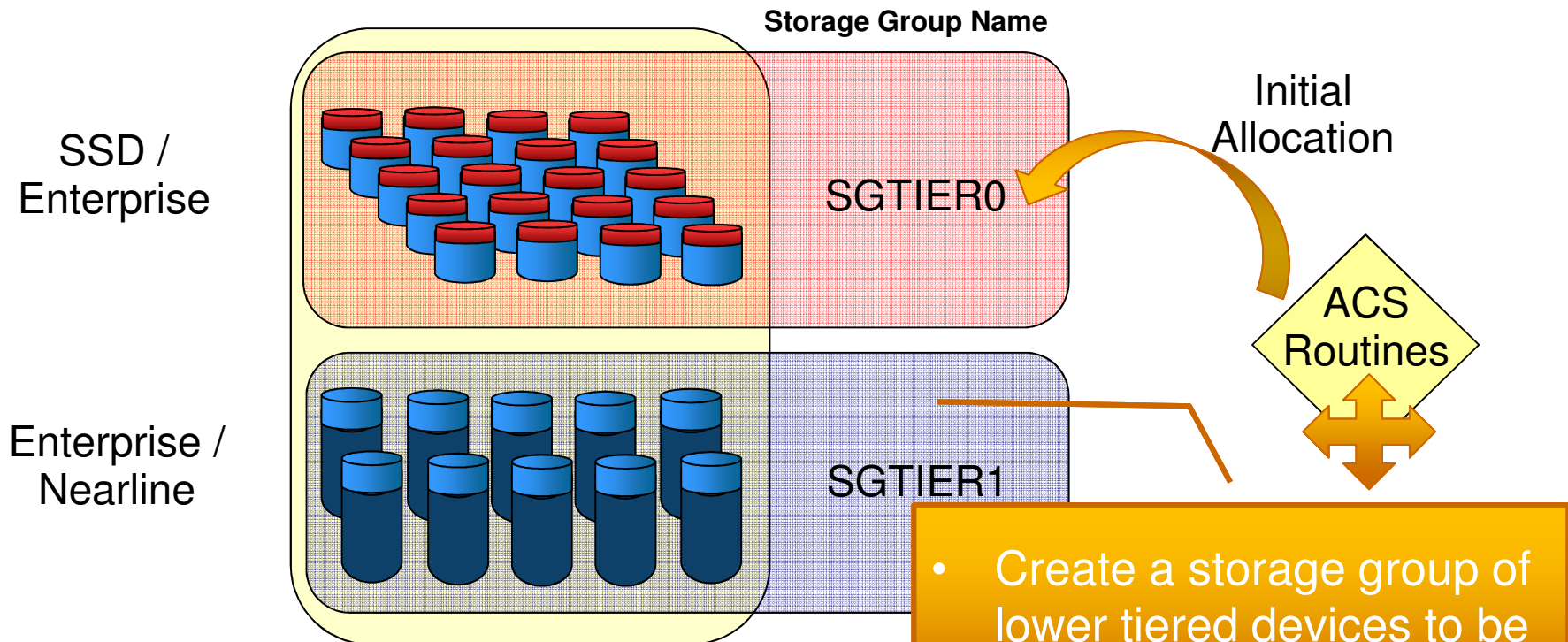
- Create a storage group of lower tiered devices to be the target of transitions

# Storage Hierarchy

## Primary Storage Hierarchy

**Smart Tier 0:**
SSD /
Enterprise
($x$)

**Smart Tier 1:**
Enterprise /
Nearline
($1/3x$)

Allocate
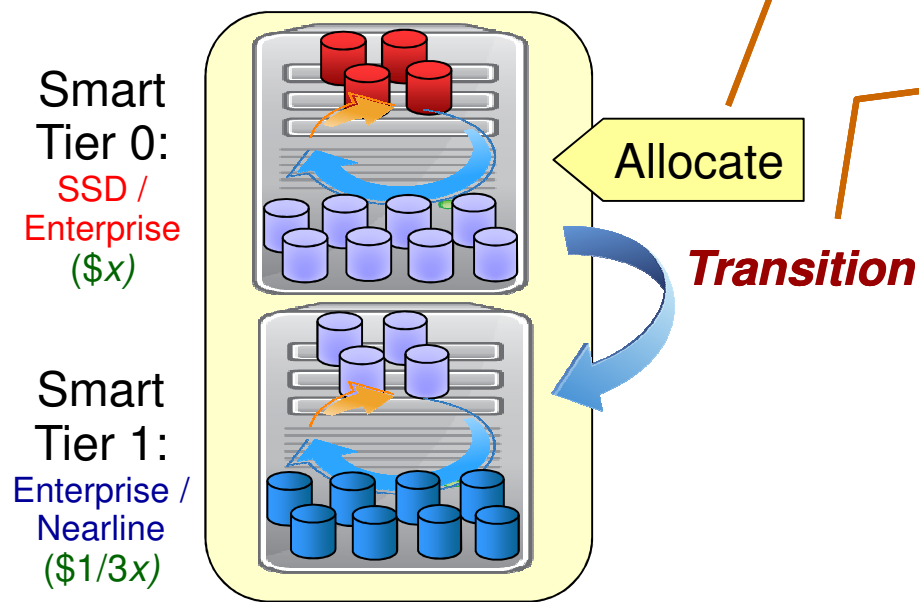
*Transition*

**Allocation**
- Data is allocated to Tier 0

**Transition**
- Based on management class policy, data will transition from Tier 0 to Tier 1

## Migration Hierarchy

ML2
(VTS)

**3 Logical Tiers**
(L0, L1, ML2)

**5 Physical Tiers**
(SSD, Enterprise, Nearline, Virtual Tape, Physical tape)

# Storage Hierarchy

**Primary Storage Hierarchy**

Smart
Tier 0:
*SSD /*
*Enterprise*
*($x)*

Smart
Tier 1:
*Enterprise /*
*Nearline*
*($1/3x)*

Allocate

*Transition*

**Migration**
- Inactive data is migrated from Tier 1 *(Some data is always active and will not migrate)*

**Migration Hierarchy**

ML2
(VTS)
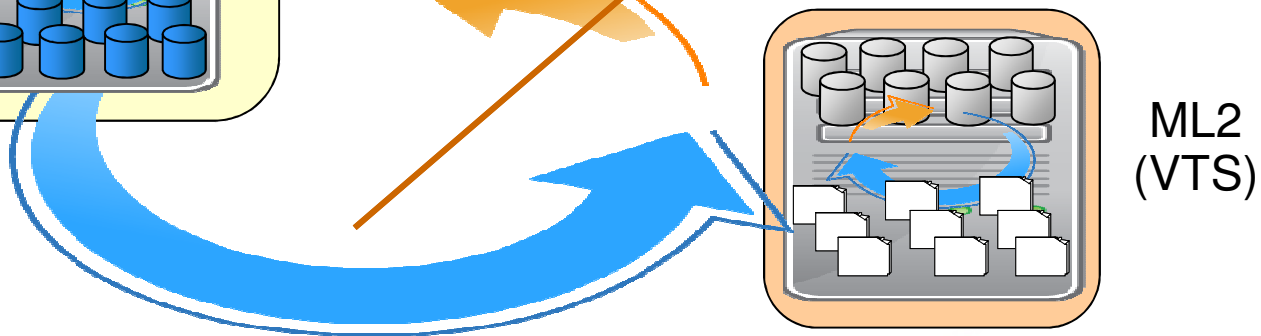
# Management Class

- The SMS Management Class provides the Class Transition policies:

    - **Class Transition Criteria:** If and when a data set should be transitioned

    - **Serialization Error Exit:** Indicates what type of special processing should occur if the data set cannot be serialized

    - **Transition Copy Technique:** Which copy technique should be used to move the data set

# Management Class

- The Class Transition Criteria specifies if and when a data set should be transitioned.

    - **Default:** Class transitions are *not* performed
    - **Time since Creation:** Data set is eligible for a transition on or after this time.
        - This is a *subjective* setting. It indicates that regardless of the usage of the data set, it should be transitioned.
    - **Time since Last Use:** Data set is eligible for a transition on or after this time.
        - This is an *objective* setting. It indicates that a data set should not be transitioned until it has not been referenced for a certain period of time.
    - **Periodic:** Data set is eligible for a transition on a specific date.
        - This is a *subjective* setting. It indicates that regardless of the usage of the data set, it should be transitioned.

- Only one criteria may be specified.

# Management Class

# Management Class



**Use Case 1: DB2**
- Set 'Time Since Creation' to 6 Months
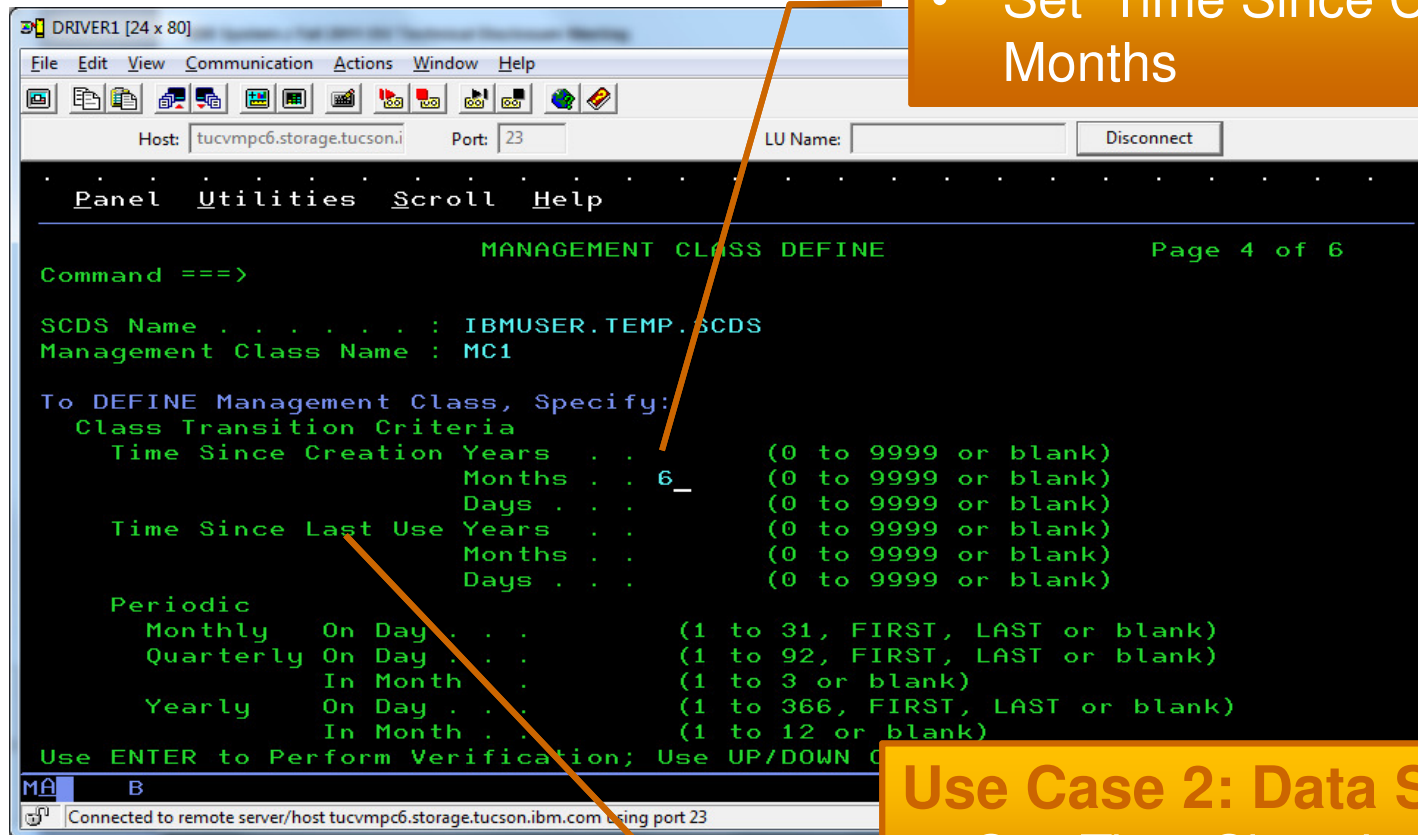
```
DRIVER1 [24 x 80]
File  Edit  View  Communication  Actions  Window  Help

Host: tucvmpc6.storage.tucson.i    Port: 23        LU Name:            Disconnect

   Panel   Utilities   Scroll   Help

                        MANAGEMENT CLASS DEFINE              Page 4 of 6
Command ===>

SCDS Name . . . . . . . : IBMUSER.TEMP.SCDS
Management Class Name : MC1

To DEFINE Management Class, Specify:
   Class Transition Criteria
     Time Since Creation Years  . .        (0 to 9999 or blank)
                         Months . .  6_    (0 to 9999 or blank)
                         Days . . .        (0 to 9999 or blank)
     Time Since Last Use Years  . .        (0 to 9999 or blank)
                         Months . .        (0 to 9999 or blank)
                         Days . . .        (0 to 9999 or blank)

     Periodic
       Monthly   On Day . . .       (1 to 31, FIRST, LAST or blank)
       Quarterly On Day . . .       (1 to 92, FIRST, LAST or blank)
                 In Month  .        (1 to 3 or blank)
       Yearly    On Day . . .       (1 to 366, FIRST, LAST or blank)
                 In Month . .       (1 to 12 or blank)
Use ENTER to Perform Verification; Use UP/DOWN
MA    B

Connected to remote server/host tucvmpc6.storage.tucson.ibm.com using port 23
```

**Use Case 2: Data Sets**
- Set 'Time Since Last Use' to 45 Days

in Orlando 2015

# Management Class

- The Serialization Error Exit indicates what type of special processing should occur if the data set cannot be exclusively serialized
  - Since **'Time since Creation'** and **'Periodic'** may specify that a transition occur during a period of time when a data set is being accessed, this setting specifies what to do if a data set cannot be exclusively serialized for data movement
  - For database data, it may be expected that the data is always open and special processing must be done to transition the data at any time

- Since it is expected that data sets may be open, the default is to not issue an error message if a data set cannot be exclusively serialized, it is just skipped *(similar to migration processing)*
  - To issue a DFSMShsm message for this condition, issue PATCH .MGCB.+EF BITS( . . .1 . . . .)
  - An FSR record is created to enable reporting for this condition

# Management Class

- Serialization Error Exit settings
  - **NONE:** (Default) If a data set cannot be exclusively serialized, the data set is not transitioned
  - **DB2:** Data assigned to this management class are DB2 objects
    - Invoke DB2 to close and unallocate the object. If this is successful, the object is serialized and moved and DB2 is invoked to reopen the object
  - **CICS:** Data assigned to this management class are CICS objects
    - Invoke CICS to take the object offline. If this is successful, the object is serialized and moved and CICS is invoked to reopen the object
  - **zFS:** Data assigned to this management class are zFS data sets
    - Invoke zFS to unmount the data set. If this is successful, the data set is serialized and moved and zFS is invoked to remount the data set
  - **EXIT:** User exit is invoked to preprocess and post-process the data set
    - Enables users / ISVs to provide an exit that will be invoked before and after transitioning an allocated data set. The data set is only transitioned if serialization is obtained after the first invocation of the exit
    - Documented in *DFSMS Installation Exits  (SC26-7396)*

# Management Class

✓ Use Case 1: DB2 objects that are always open
- Set management class Serialization Error Exit to 'DB2'

✓ Use Case 2: Standard Data Sets
- Use default value of 'NONE'

# Active Data

SETSYS CLASSTRANSITIONS(EVENTDRIVENMIGRATION(Y|N
      SERIALIZATIONEXIT(Y|N)))

- This SETSYS indicates if Event Driven Migration (On-Demand Migration and Interval Migration) should process class transitions
  - The default value is 'Y'es
- If 'Y'es is specified (or defaulted), then SERIALIZATIONEXIT indicates if Serialization Error Exit processing should be performed
  - The default is 'N'o
  - This means that any open data sets will be skipped during ODM and IM if exclusive access cannot be obtained for data movement

- Aliases:
  SETSYS CLTR(EDM(Y|N SERL(Y|N)))

# Management Class

- The Transition Copy Technique indicates what type of copy technique should be used to move the data set
  - **Standard:** (Default) Use standard I/O
  - **Fast Replication Preferred:** Prefer Fast Replication. If it cannot be used, then use standard I/O.
  - **Fast Replication Required:** Require Fast Replication. If it cannot be used, fail the data movement.
    - Requires the target volume to be in the same storage controller.
  - **Preserve Mirror Preferred:** Prefer Preserve Mirror. This indicates that a Metro Mirror primary volume is allowed to become a FlashCopy target volume. If Preserve Mirror cannot be used, FlashCopy or standard I/O may be used.
  - **Preserve Mirror Required:** Require Preserve Mirror. The transition is only performed if the Metro Mirror primary target volume will not go duplex pending. This parameter has no affect if the target volume is not a Metro Mirror primary volume.

# Management Class

- **Transition Copy Technique**

    – If a copy technique *other than* 'Standard' is specified, then *a valid backup copy must exist and the DS Change Indicator OFF* before the data set is transitioned

    - This is required because DFSMShsm receives control immediately after the FlashCopy relationship is established.
    - DFSMShsm is not notified if there is a physical error within the storage controller during the background physical copy
    - While very unlikely, this ensures that if any physical error occurs while transitioning the data, that a backup copy is available to recover the data set
        » This is critical because the data set is Deleted after the logical completion

# Management Class

- ## V2R1 D-Type APAR OA46914

  - Patch to indicate that a product other than DFSMShsm is used to create backup copies. (HSM will only verify that the Data Set Change Indicator is OFF before allowing a transition).
    - PATCH .MGCB.+111 BITS(......1.)

  - Patch to enable a transition with Preserve Mirror Required even when the DSCI is ON
    - PATCH .MGCB.+111 BITS(.......1)
    - Enables Preserve Mirror to be used for DB2, CICS and zFS objects that are always open
    - *This will be the default in V2R2*

# ACS Routines

- Once DFSMShsm determines that a data set has met the Class Transition criteria specified by the Management Class, it invokes the ACS routines to determine what the transition should be

  - ACS Routines are invoked with new ACS environment (&ACSENVIR) of SPMGCLTR, for 'space management class transition'
  - The following routines are invoked *(in this order)*
    - Storage Class
    - Management Class
    - Storage Group
  - Any or all can be *transitioned*

- ✓ If the classes and storage group returned match the existing classes and storage group, *then no transition occurs*
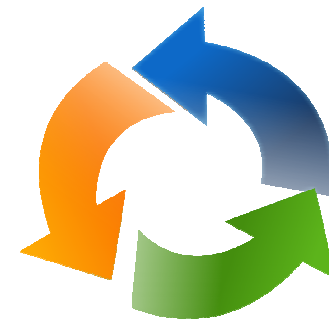
# ACS Routines

- **Storage Class**

    - Storage Class indicates the *'preferred'* class of storage to which the data set should be allocated

    - ★ If storage class changes, but storage group remains the same, *and a device matching the new storage class attributes cannot be selected*, the data set is *not* moved

```
IF &ACSENVIR = 'SPMGCLTR' THEN
    Logic…
        SET &STORCLAS = 'STANDARD'
        OTHERWISE  SET &STORCLAS = &STORCLAS
    END
ELSE ...
```

# ACS Routines

- ### Management Class

  - When a new management class is assigned, DFSMShsm will begin using the newly assigned policies to manage the data set

  - If only the management class changes, then the data set is altered to assign it to the new management class and *no data movement is performed*

  - ### *Examples*

    - Upon creation, a data set is assigned to a management class for which the data set is only eligible to migrate to ML1 (not ML2) and 2 backup copies are maintained

    - After 120 days from creation, the data set is transitioned to a different management class for which the data set is eligible to migrate to ML2 and only 1 backup copy is maintained.

# ACS Routines

- Use Case 1: DB2

```
IF &ACSENVIR = 'SPMGCLTR' THEN
/*  SPACE MANAGEMENT CLASS TRANSITION  */

    Logic…
        SET &MGMTCLAS = 'DB2AGED'
    OTHERWISE  SET &MGMTCLAS = &MGMTCLAS
      END
ELSE
    ...
```
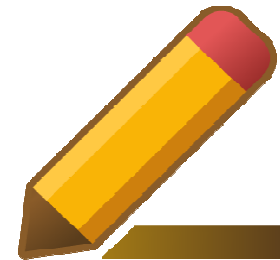
# Management Class

- Use Case 2: Standard Data Sets
  - Update the management class Migration criteria for PRIMARY DAYS NON-USAGE from the existing value to 366 Days

# ACS Routines

- ## Storage Group
    - From 1 to 15 storage groups may be returned
        - It is the administrator's responsibility to ensure that a different storage group name provides a meaningful transition

```
IF &ACSENVIR = 'SPMGCLTR' THEN
    Logic…
        SET &STORGRP = 'SGTIER1'
    ELSE…
```

# Storage Group

- The new Storage Group Processing Priority specifies the relative order in which storage groups should be processed during Primary Space Management

  - *Class Transition processing will move data from one storage group to another*

  - In order to help ensure that the 'receiving' storage groups have enough space for the data sets that will be moved to them, a new storage group Processing Priority is provided
    - These storage groups should be assigned a higher priority

  - Storage Groups will be processed in the order of their priority
    - A higher value means a higher priority
    - Valid values are 1-100
    - Default value is 50

| 80 | 50 | 50 | 20 |

# Class Transition Processing

- When DFSMShsm determines that a data set should be moved for a Class Transition, DFSMSdss is invoked to perform a Logical COPY with DELETE

  - ★ DFSMSdss is the full data mover
    - Unlike migrate/recall and backup/recover where DSS is only the half data mover
  - DFSMSdss handles Copy Technique and Exit processing
    - After the movement, the data set retains all existing attributes and can be immediately accessed

- The catalog is updated as a part of the movement

  - ★ *No* new DFSMShsm control data set records created for transitions
    - New FSR record type created for reporting purposes
      - FSRTYPE = 24
      - New DFSMS Report Generator *sample report* provided
  - DSR and VSR records are updated for DFSMShsm REPORT

# Class Transition Processing

- REPORT FUNCTION(TRANSITION) SYSOUT(A)
  - 'TRANSITION' is a new option

```
--DFSMSHSM STATISTICS REPORT ------- AT 08:05:07 ON 2011/09/13 FOR SYSTEM=2094
DAILY STATISTICS REPORT FOR 11/09/13
STARTUPS=000, SHUTDOWNS=000, ABENDS=000, WORK ELEMENTS PROCESSED=000005, BKUP VO
DATA SET MIGRATIONS BY VOLUME REQUEST= 0000000, DATA SET MIGRATIONS BY DATA SET
EXTENT REDUCTIONS= 0000000 RECALL MOUNTS AVOIDED= 00000 RECOVER MOUNTS AVOIDED=
DATA SET CLASS TRANSITION = 00000085 REQUESTED, 00000002 FAILED
                        NUMBER  ------READ--------  -----WRITTEN------  ------REQUES
   HSM FUNCTION        DATASETS TRK/BLK    BYTES     TRK/BLK    BYTES    SYSTEM USER
CLASS TRANSITION
 PRIMARY - PRIMARY 0000083  00053161 000632776K 00053161 000632776K 000085 00000
```

- New Report Generator Sample Report
  - ARCGS011 Statistics for Class Transitions

# Class Transition Processing

- A new field was created in the NVR/VVR called 'Last Successful Class Transition Date' (LSCTD)
  - When a data set is successfully transitioned, the LSCTD is...
    - set to zero when the management class was changed
      -or-
    - set to the current date if the management class was not changed

  - LSCTD is used by DFSMShsm to know when a data set has already been successfully transitioned
  - DFSMShsm will attempt to transition a data set if it has met the transition criteria *AND* the LSCTD is zero
  - Exception for PERIODIC transitions. Data set will transition if the last transition was before the specified period

# Class Transition Processing

- 'Last Successful Class Transition Date' (LSCTD) shown in LISTCAT output:
  *(when field is nonzero)*

```
NONVSAM ------- STORTIER.M01.S01.D01.N01.PSFB
   IN-CAT --- STRTRFVT.USERCAT
   HISTORY
     DATASET-OWNER-----(NULL)    CREATION--------2012.001
     RELEASE----------------2    EXPIRATION------0000.000
     ACCOUNT-INFO--------------------------------(NULL)
   SMSDATA
     STORAGECLASS ---SCLASS22    MANAGEMENTCLASS-MCLASS01
     DATACLASS ------DCLASS01    LBACKUP ---2012.001.0701
     LAST TRANSITION-2012.013
   VOLUMES
     VOLSER------------LSMS12    DEVTYPE------X'3010200F'
   ASSOCIATIONS--------(NULL)
   ATTRIBUTES
```

- Field is also available via DCOLLECT

# Class Transition Processing

- Since Class Transitions are a part of the existing space management function...

  - The same tasking level that controls data set migrations controls data set transitions
    - If you expect more work during an existing space management window, then you may consider increasing your tasking level

  - The same HOLD, RELEASE, CANCEL, etc commands for space management control Class Transition processing also

  - ARC0734I (standard space management message) issued for class transitions
    - Insert updated to show 'ACTION=CLASS-TR'
    - Indicates From volume, To volume, RC, RSC, etc

# Class Transition Processing

- RECALL Processing

  - When a data set is Recalled, DFSMShsm will determine if a data set missed one or more transitions while the data set was migrated

    - If a transition was missed, then DFSMShsm will invoke the ACS routines for each transition that was missed, to determine the appropriate management class, storage class and storage group to which the data set should be recalled

  - If a migrated data set was transitioned before migration, the DFSMShsm will invoke the ACS routines with SPMGCLTR to ensure that the recall is to the correct device

# Use Case

*Monthly customer statements are allocated on standard enterprise disk. After 45 days, the statements should remain online, but should be transitioned to Nearline disk in order to minimize the cost of storing the statements online.*

## Implementation Steps

- Ensure one or more storage groups comprised of Nearline disk are setup/available to receive the transitioned statements
- Update the storage group ACS Routines with the appropriate class transition logic.
  - For example, when the ACS Environment is a class transition, if the management class is 'ONLNSTMT', then assign to a storage group comprised of Nearline disks.
- Update the management class to which these statements are assigned to have a class transition occur 45 days after creation
- During Space Management, DFSMShsm will transition eligible statements from enterprise disk to Nearline
- Be Careful, don't transition too much data at once!

# Use Case

**Primary Storage Hierarchy**

**L0**

Allocate

**L1**

**Migration Hierarchy**

**ML2**
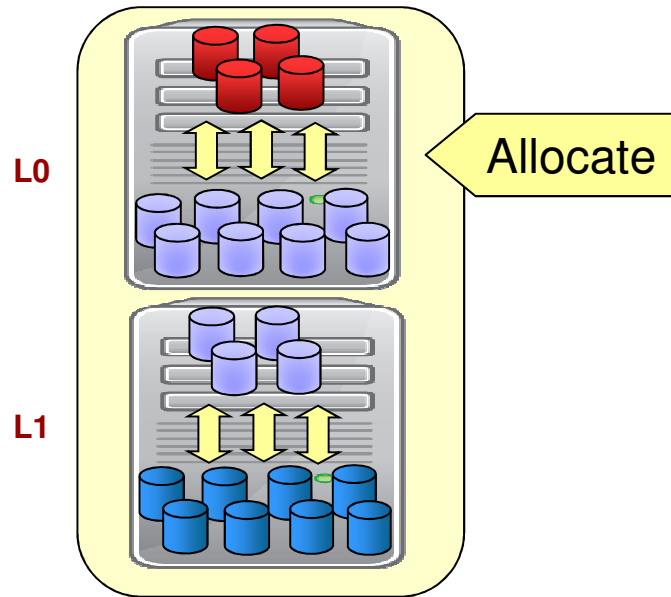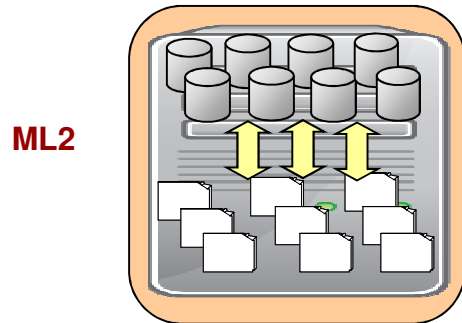
## State 1 – Allocation

**Expected access**
Random reads and writes
with periods of high activity
**Storage Group (SGLEV0)**
Storage Tiered devices
High – SSD
Low - Enterprise
**Management Class (ONLNSTMT)**
No Migration
*Data will be active*
*Even if not, no migration*
Two Backup copies
*Data is actively changing*
Transition after 45 days
No Expiration

# Use Case

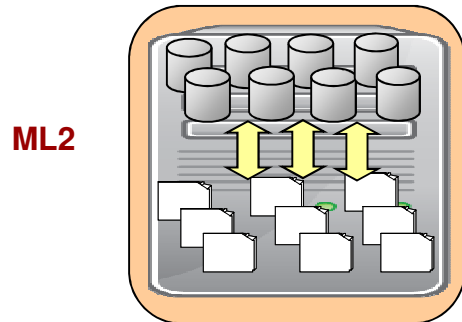**Primary Storage Hierarchy**

**L0**

**L1**

**Migration Hierarchy**

**ML2**
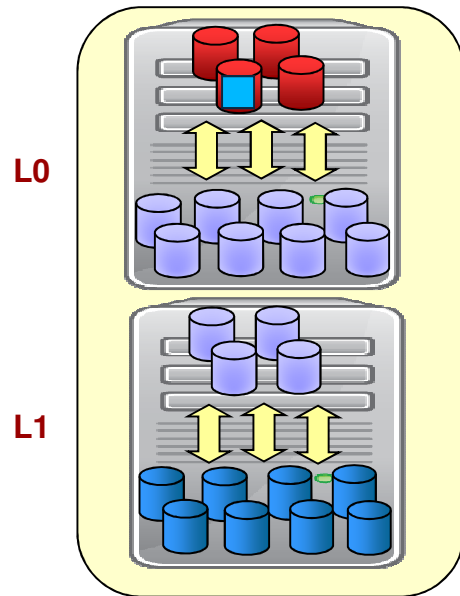
## State 2 – First 45 Days

Storage Controller moves the data between SSD and Enterprise based on the data's heat map

DFSMShsm manages based on ONLNSTMT
Two Backup Copies

# Use Case

**Primary Storage Hierarchy**

**L0**

**L1**

45 Day
***Transition***

**Migration Hierarchy**

**ML2**
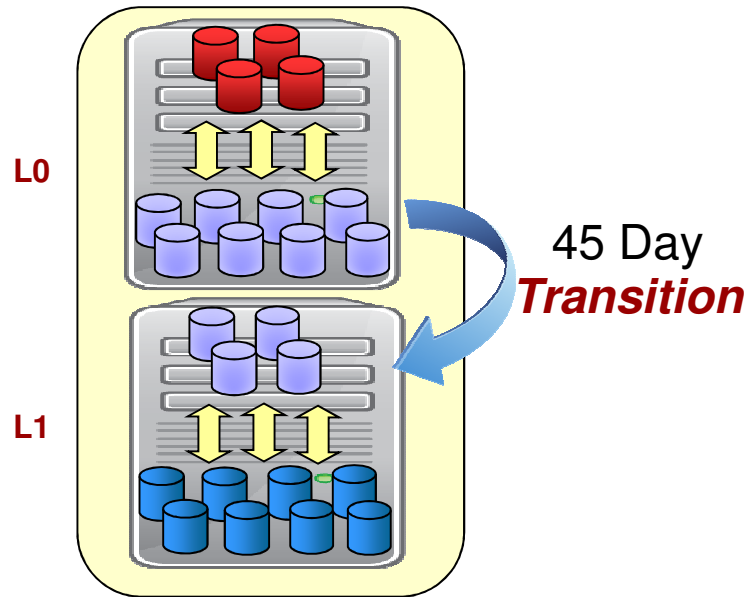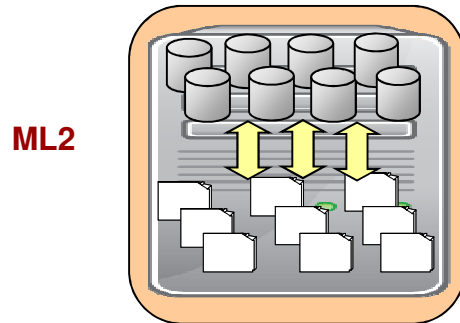
## State 3 – 45 Day Transition

**Expected access**
After 45 days, the data is expected to be Historical, instead of Active. No expected writes and mostly *sequential reads*.

**Storage Group (SGLEV1)**
Storage Tiered devices
High – Enterprise
Low - Nearline

**Management Class (HISTSTMT)**
Migration
Migrate after 21 days of inactivity
Single Backup copy
*Data is not changing*
No further transitions
Expire after 7 years

# Use Case

**Primary Storage Hierarchy**

**L0**

**L1**

**Migration Hierarchy**

**ML2**
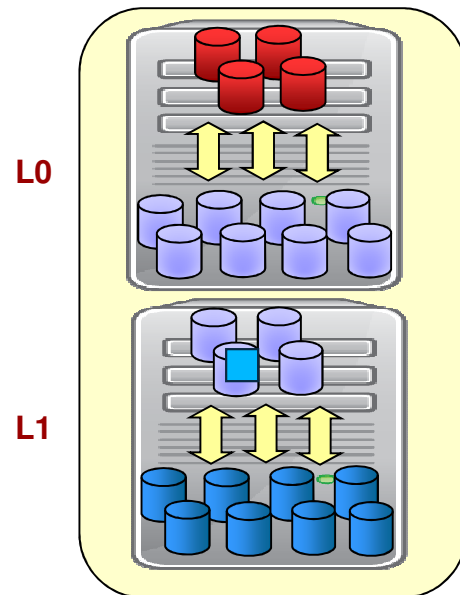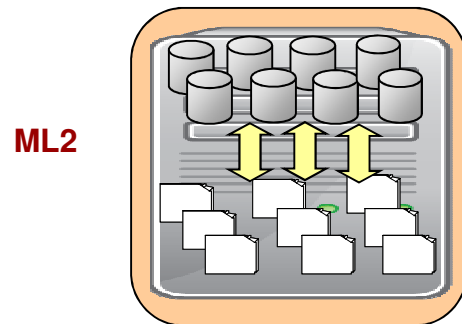
## State 4 – The Golden Years

**Storage Controller moves the data between Enterprise and Nearline based on the data's heat map**

**DFSMShsm manages based on HISTSTMT**
Single Backup Copy
Migrate at 21 days inactive
*Storage Controller compresses the data and moves it from disk to tape*
Expire after 7 years

# V2R1 D-Type APAR OA46627

## MD Exit enabled to override transition with a migration
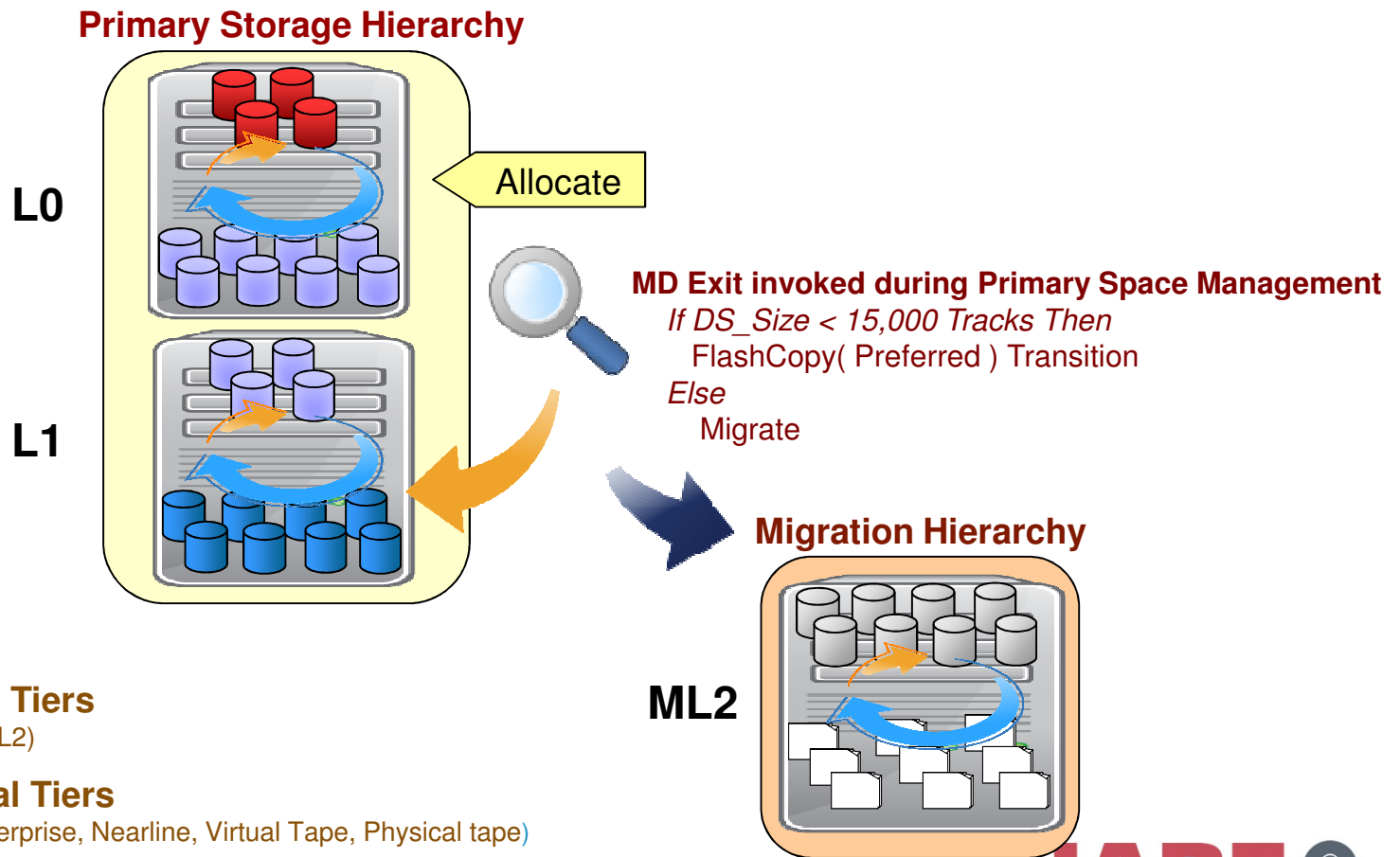
**Primary Storage Hierarchy**

**L0**

Allocate

**L1**

**MD Exit invoked during Primary Space Management**
*If DS_Size < 15,000 Tracks Then*
FlashCopy( Preferred ) Transition
*Else*
Migrate

**Migration Hierarchy**

**ML2**

**3 Logical Tiers**
(L0, L1, ML2)

**5 Physical Tiers**
(SSD, Enterprise, Nearline, Virtual Tape, Physical tape)

# z/OS V2R2 – Classic Migration

- **Problem**
  - Command initiated volume migration is single threaded. This severely limits the throughput.

- **Solution**
  - New STORAGEGROUP keyword to initiate migration for all volumes within a storage group

- **Benefit / Value**
  - Significantly improve the throughput for command initiated space management and improve usability

# z/OS V2R2 – Classic Migration

- New STORAGEGROUP keyword on the MIGRATE command
  - MIGRATE STORAGEGROUP(sgname, sgname, …) …

- Up to 30 storage group may be specified
- Primary Space Management processing will be performed for the storage groups, as opposed to the standard volume-by-volume processing performed by MIGRATE VOLUME

- **Use Case**
  - Clients create DB2 image copies to a specific storage group and then invoke DFSMShsm MIGRATE to move the copies to tape
  - This new function enables a single command to process all of the volumes in the storage group in parallel

# z/OS V2R2 – Storage Tiers

- **Problem**
  - The DFSMS Class Transition function is limited to automatic space management processing

- **Solution**
  - Enable Class Transitions to be initiated with the various Migration commands

- **Benefit / Value**
  - Improve usability

# z/OS V2R2 – Storage Tiers

- The various Migrate commands are enhanced to support class transitions at the data set, volume and storage group level
  - The default behavior is to perform both migration and transition processing for VOLUME and STORAGEGROUP operations
    - **BOTH** – default, both migrations and transitions are performed
    - **MIGRATIONONLY** – a data set is only processed if it is eligible for migration
    - **TRANSITIONONLY** – a data set is only processed it is eligible for a class transition
  - If a data set is eligible for both migration and transition processing, then it will be migrated

  - The default for MIGRATE DATASET is to perform a migration. The **TRANSITION** keyword indicates that a transition should be performed

# z/OS V2R2 – Storage Tiers

- *Why would I specify MIGRATE DSNAME(...) TRANSITION as opposed to just using DSS COPY w/ Delete?*

  - Transition processing invokes the management class Serialization Error Logic
    - Serialization Error logic during transition processing indicates what to do if the data set is Open at the time of processing
    - DB2, CICS and zFS data can be temporarily closed, transitioned and then re-Opened
  - Transition processing also invokes the ACS Routines with the 'SPMGCLTR' environment variable

# z/OS V2R2 – Data Migration

- **Problem**
  - Moving data to newly defined disk volumes within a storage environment can be manually intensive and cumbersome

- **Solution**
  - Enable the DFSMS Class Transition function to move data laterally to the same tier of storage, in addition to the ability of moving data to different tiers storage

- **Benefit / Value**
  - Introduce new functionality to simplify the task of migrating data to newly defined disk volumes

# z/OS V2R2 – Data Migration

- **MOVE** keyword added to the MIGRATE DSNAME, VOLUME and  STORAGEGROUP commands

    - Every data set will be processed, regardless of management class policy or threshold
    - ACS routines will be invoked to determine the new storage class and/or storage group
    - Management class Serialization Error logic and Class Transition movement techniques will be used
    - Only SMS data is supported

# z/OS V2R2 – Data Migration

- Use Case
  - Move DB2 data from existing smaller volumes to the new larger, newly defined EAVs
    - Step 1: Management Class Serialization Error logic indicates that the data is **DB2**
    - Step 2: Place current volumes into **a DISNEW** or **DISALL** state
    - Step 3: MIGRATE VOLUME(vol1, vol2, …) MOVE
      - DFSMShsm will process every data set on every volume
      - If the DB2 object is open, DB2 will be invoked to close the object, Fast Replication can be used for the data movement, and then the DB2 object reopened
      - Since the EAVs have the most free space, they will be selected for the movement

# Single command to move data:

**MIGRATE VOLUME(VOL1, VOL2, VOL3) MOVE**

- With Preserve Mirror, FlashCopy can be used. Movement complete in minutes. Minimal Downtime at the object level!
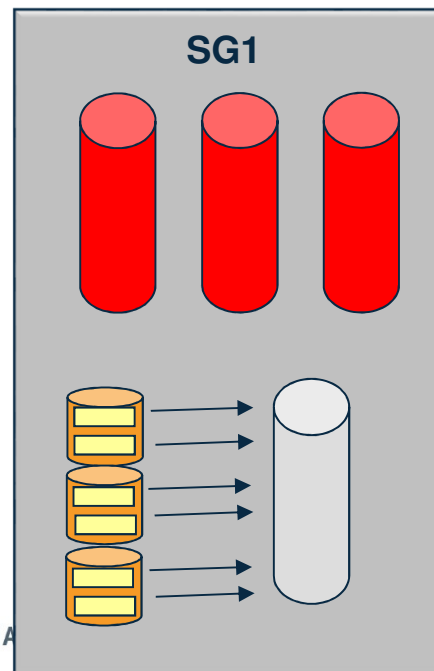
# Looking Forward...

*Interlock DFSMS and Control Unit Tiering to provide automated, policy-based transitions of open data at the data set level*

|  | DFSMS Tiering | Controller Tiering |
|---|---|---|
| **Movement Boundary** | Data Set Level | Physical Extent Level |
| **Scope** | Sysplex (across controllers) | Intra-controller |
| **Level of Management** | Data Policy based | Extent Temperature based |
| **Access** | Closed Data Only | Open and Closed Data |
| **Impact** | Data must be quiesced | Transparent |
| **Cost** | Host based MIPS | No host based MIPS |

# Transition to Transitions
## Session 17804

*Glenn Wilcock*
*DFSMShsm Architect*
*IBM*

# Trademarks and Disclaimers

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

| | | | |
|---|---|---|---|
| CICS* | FICON* | Lotus* | System Storage |
| DB2* | FlashCopy* | MQSeries* | Tivoli* |
| DFSMS | GDDM* | Multiprise* | TotalStorage* |
| DFSMSdfp | GDPS* | OMEGAMON* | Virtualization Engine |
| DFSMSdss | geoManager* | OS/390* | VisualAge* |
| DFSMShsm | HiperSockets | Parallel Sysplex* | VM/ESA* |
| DFSMSrmm | HyperSwap | PR/SM | VSE/ESA |
| DFSORT | IBM* | QMF | VTAM* |
| DFSMS | IBM logo* | RACF* | WebSphere* |
| DS4000 | ImagePlus* | Rational* | z/Architecture* |
| DS6000 | IMS | RMF | z/OS* |
| DS8000 | Intelligent Miner | System i | z/VM* |
| Enterprise Storage Server* | Language Environment* | System z | z/VSE |
| ESCON* | | System z9 | zSeries* |
| | | | zSeries Entry License Charge |

The following are trademarks or registered trademarks of other companies:

Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries or both

Microsoft, Windows,Windows NT and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

NOTES:

Any performance data contained in this document was determined in a controlled environment.  Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration.  Some measurements quoted in this document may have been made on development-level systems.  There is no guarantee these measurements will be the same on generally-available systems.  Users of this document should verify the applicable data for their specific environment.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Information is provided "AS IS" without warranty of any kind.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

# Trademarks and Disclaimers (continued)

NOTES:

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices are suggested US list prices and are subject to change without notice. Starting price may not include a hard drive, operating system or other features. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM makes no representation or warranty regarding third-party products or services including those designated as ServerProven, ClusterProven or BladeCenter Interoperability Program products. Support for these third-party (non-IBM) products is provided by non-IBM Manufacturers.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval