IBM Education Assistance for z/OS V2R2

Items: zEDC for DFSMSdss and hsm
Storage Tiers
zCDP for DB2 Enhancements
Element/Component: DFSMShsm and DFSMSdss

Material current as of May 2015

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

- **Describe DFSMSdss and DFSMShsm support for zEDC**
- **Describe the enhancements provided for DFSMShsm Space Management.**
  - Classic Migration
    - New Storage Group keyword
  - Storage Tiers
    - Command driven class transitions
  - Data Migration between L0 volumes
    - Command driven movement
- **Describe the enhancements provided for the Continuous Data Protection solution for DB2.** These solutions may also be used for standard DFSMShsm Fast Replication and Dump Processing.
  - Common Dump Queue
  - Dump Enhancements
  - Fast Replication Messaging

Overview

- **Problem Statement / Need Addressed**
  - Host based compression can consume a significant amount of MIPS resources
- **Solution**
  - Enable z/OS V2R1 DFSMSdss and hsm to utilize zEDC
- **Benefit / Value**
  - Reduce MIPS
  - Improve compression ratio
  - Improve throughput

IBM zEnterprise Data Compression (zEDC)

What is it?

zEDC Express is an IO adapter that does high performance industry standard compression

Used by z/OS Operating System components, IBM Middleware and ISV products

Applications can use zEDC via industry standard APIs (zlib and Java)

Each zEDC Express sharable across 15 LPARs, up to 8 devices per CEC.

Raw throughput up to 1 GB/s per zEDC Express Hardware Adapter

What Changes?

It is time to revisit your decisions about compression.

- **Disk Savings**: Many people are already getting value from CMPSC compression and software compression today
- **Performance**: High throughput alternative to existing System Z compression for large or active files.
- **Industry Standard:** Low cost compressed data exchange across all platforms
- **Pervasive:** Standard APIs allow quick adoption by middleware products running on System Z

What is the Value?

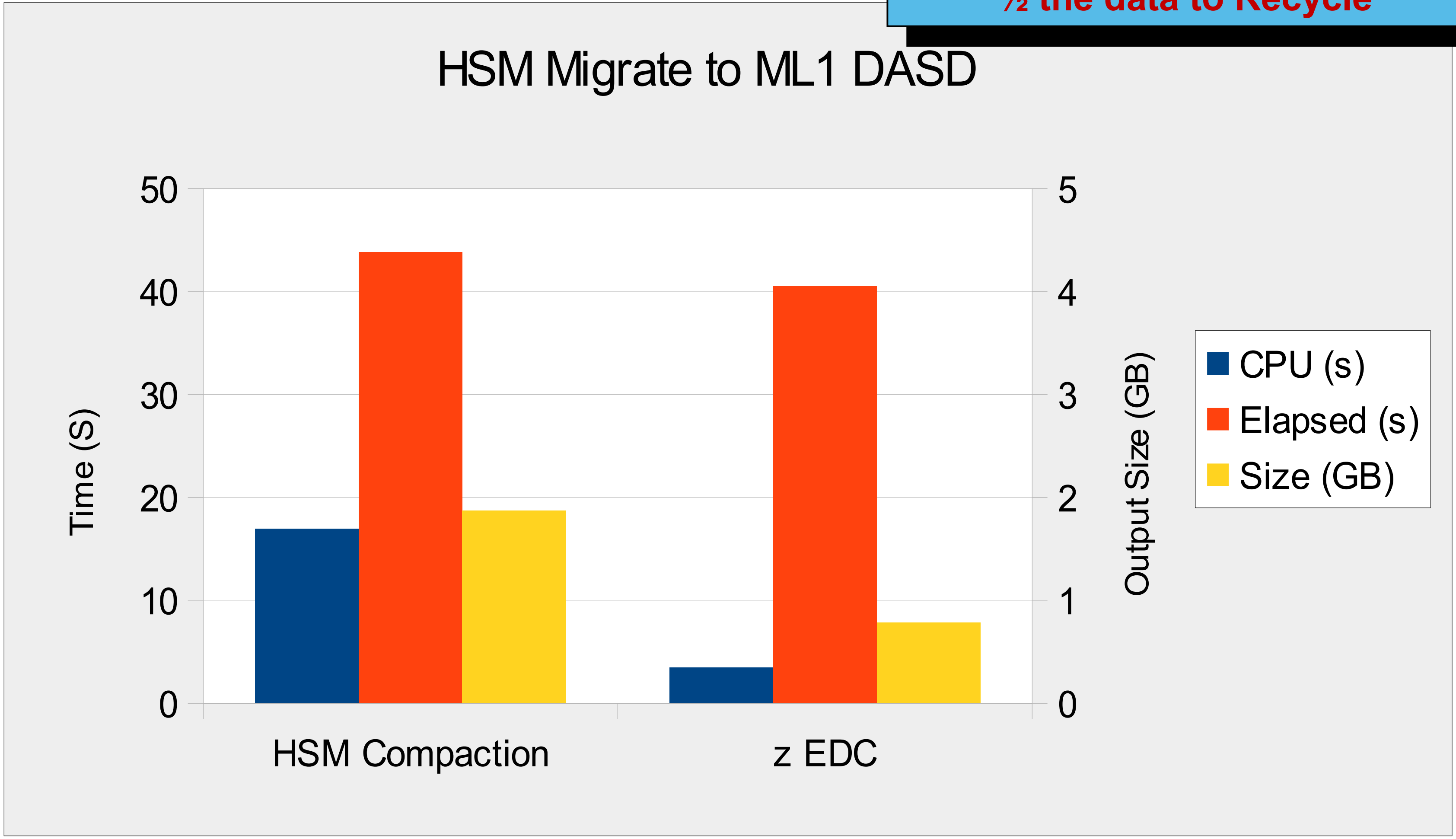
New sources of customer value

- **QSAM/BSAM** compression can save up to 4x disk space and in some cases shorten elapsed time, reducing batch windows.
- **HSM/DSS** Offload host-based compression
- **Business Partner Data Exchange** can have higher throughput with lower CPU cost
- **Managed File Transfer** saves up to 4x link bandwidth, and up to 80% elapsed time.
- **Java for z/OS R7.1** accelerates common compression classes used by applications and middleware
- Improved availability with **SMF** compression.

Overview

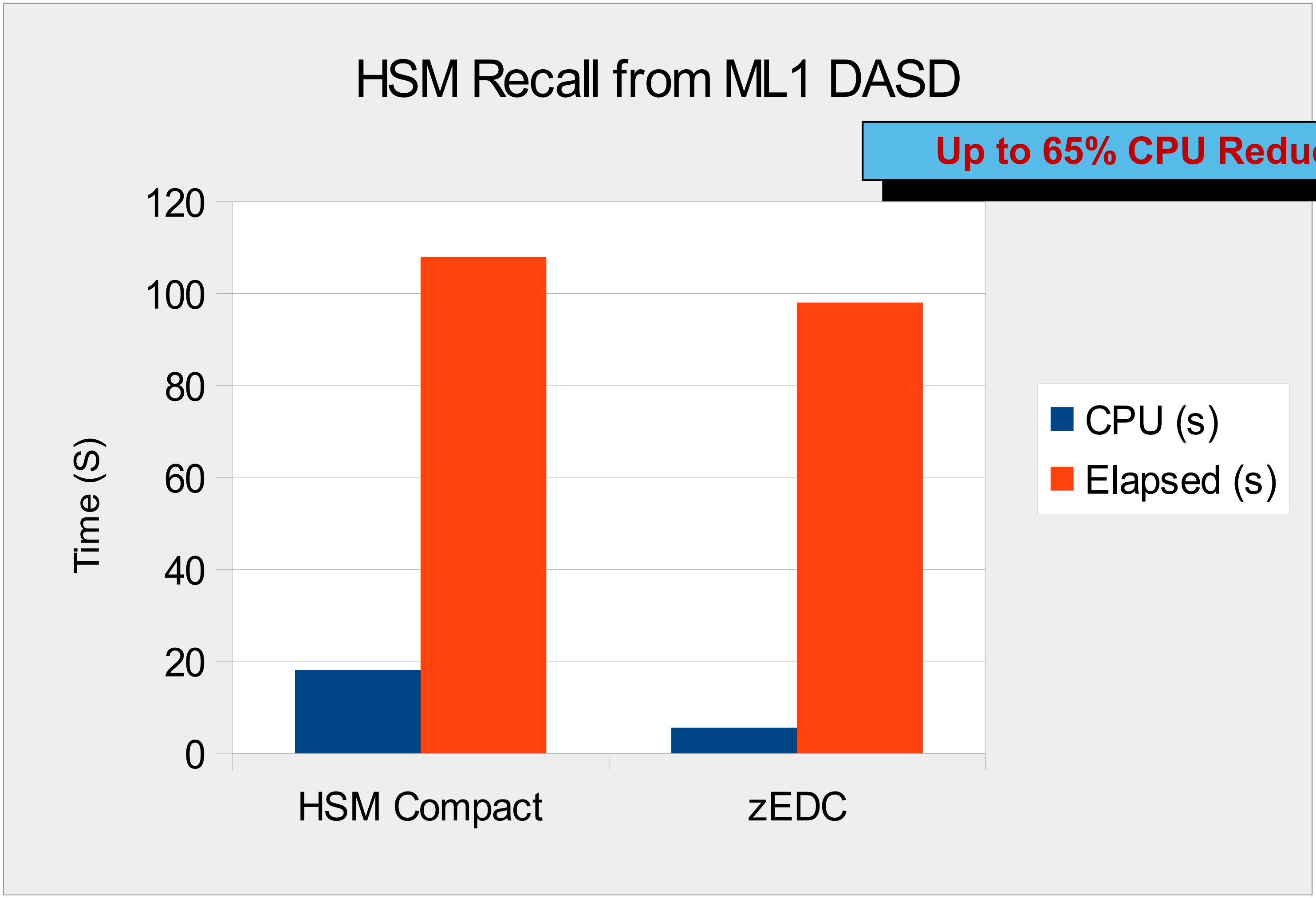
Up to 80% CPU Reduction
Up to 53% Space Reduction

½ the data for SSM
½ the data to Recycle



* **Disclaimer:** ***Based on projections and/or measurements completed in a controlled environment. Results may vary by customer based on individual workload, configuration and software levels.***

Overview



Up to 65% CPU Reduction

* **Disclaimer:** ***Based on projections and/or measurements completed in a controlled environment. Results may vary by customer based on individual workload, configuration and software levels.***

Usage & Invocation: DSS

- New keyword, ZCOMPRESS, provided for the DUMP command
    - Accepted for all FULL, TRACKS, physical and logical DATASET backups to DASD and tape
- Has three sub-parameters
    - REQUIRED – backup must be created using zEDC
        - If zEDC accelerators cannot be used then the backup should fail
    - PREFERRED – the backup should be created using zEDC
        - If zEDC accelerators cannot be used then the backup should continue anyway
    - <u>NONE</u> – backup should not be created using zEDC

Usage & Invocation: DSS

- ZCOMPRESS(PREFERRED) can be specified in conjunction with COMPRESS or HWCOMPRESS

```
DUMP FULL INDY(VOL001) OUTDD(DDUMPOUT)
      ZCOMPRESS(PREF) COMPRESS
```

- DFSMSdss will attempt to use zEDC accelerators but if none are available it will revert to using the algorithm provided by the COMPRESS or HWCOMPRESS keyword
  - A backup is either compressed entirely with zEDC accelerators or entirely with the COMPRESS or HWCOMPRESS algorithms
- The use of zEDC for backups can be restricted using a new facility class profile

STGADMIN.ADR.DUMP.ZCOMPRESS

Usage & Invocation: DSS

- Backups created using zEDC accelerators are automatically decompressed
  - No new keywords for the RESTORE command
- Use either zEDC accelerators or software inflate
- Any user can restore a backup created using zEDC
  - No facility class profile to restrict the use of zEDC during RESTORE

Usage & Invocation: HSM

- DFSMShsm can use zEDC, when DSS is the datamover, for the following functions:
  - Migrate / Recall
  - Backup / Recover
  - Dump / Restore
- PDS data sets are moved entirely by HSM, so are not yet supported
- When specified, DFSMShsm will invoke DSS with the zCOMPRESS(PREFERRED) option

Usage & Invocation: HSM

SETSYS ZCOMPRESS(*options*)

- ALL | NONE
- DASDBACKUP (NO | YES)
- DASDMIGRATE (NO | YES)
- TAPEBACKUP (NO | YES)
- TAPEMIGRATE (NO | YES)

- COMPACTPERCENT works in conjunction with ZCOMPRESS as it does for COMPACT

DEFINE DUMPCLASS(ZCOMPRESS(NO | YES))

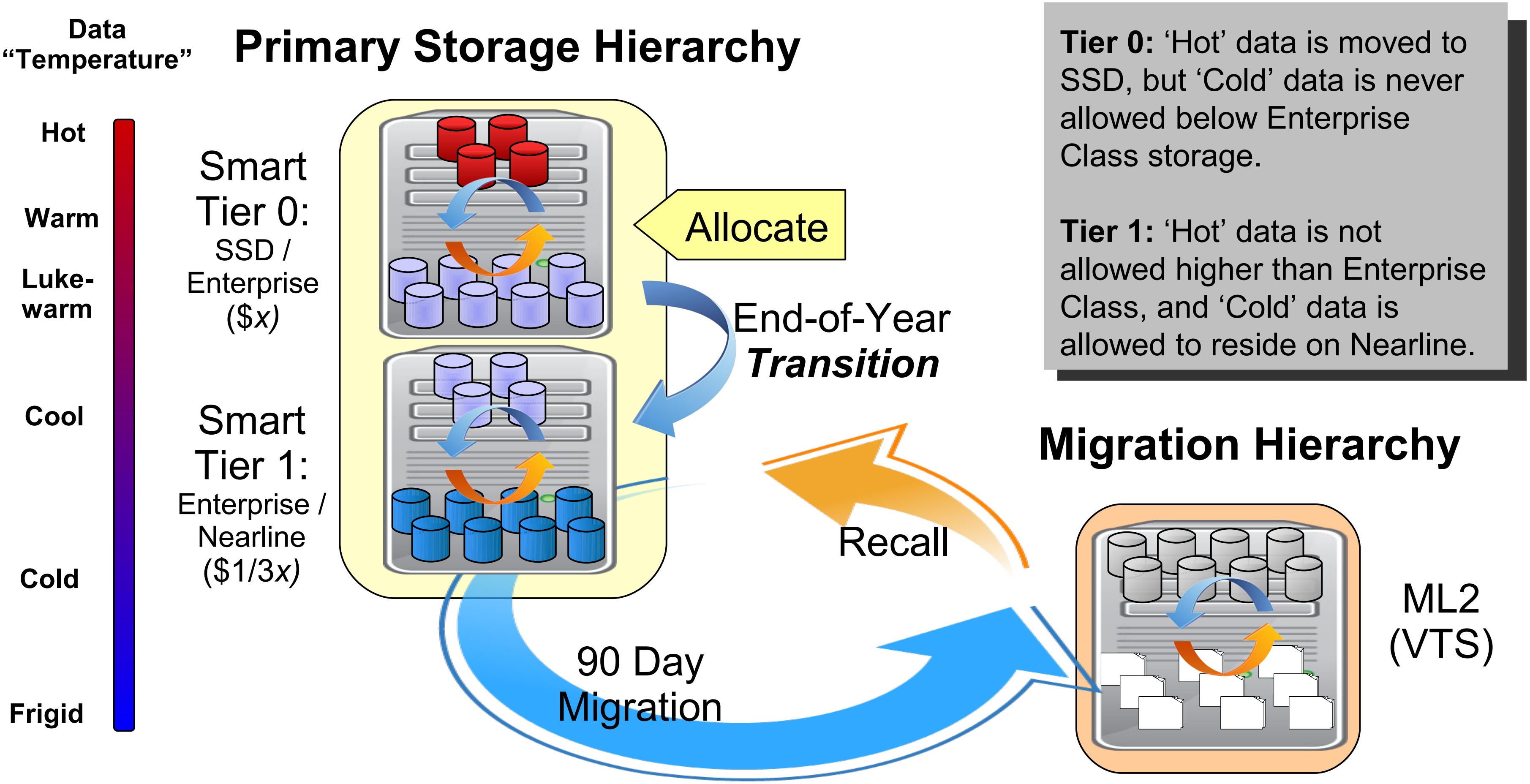
- Valid for BACKVOL and FRBACKUP when DUMP is specified

Overview – 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

Overview – Storage Tiers

This example environment shows data being "transitioned" from 'Smart Tier 0' to 'Smart Tier 1' at the end-of-year, and then becoming eligible for migration after 90 days of inactivity.



Tier 0: 'Hot' data is moved to SSD, but 'Cold' data is never allowed below Enterprise Class storage.

Tier 1: 'Hot' data is not allowed higher than Enterprise Class, and 'Cold' data is allowed to reside on Nearline.

Overview – 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

Overview – 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

Usage & Invocation – 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

Usage & Invocation – 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

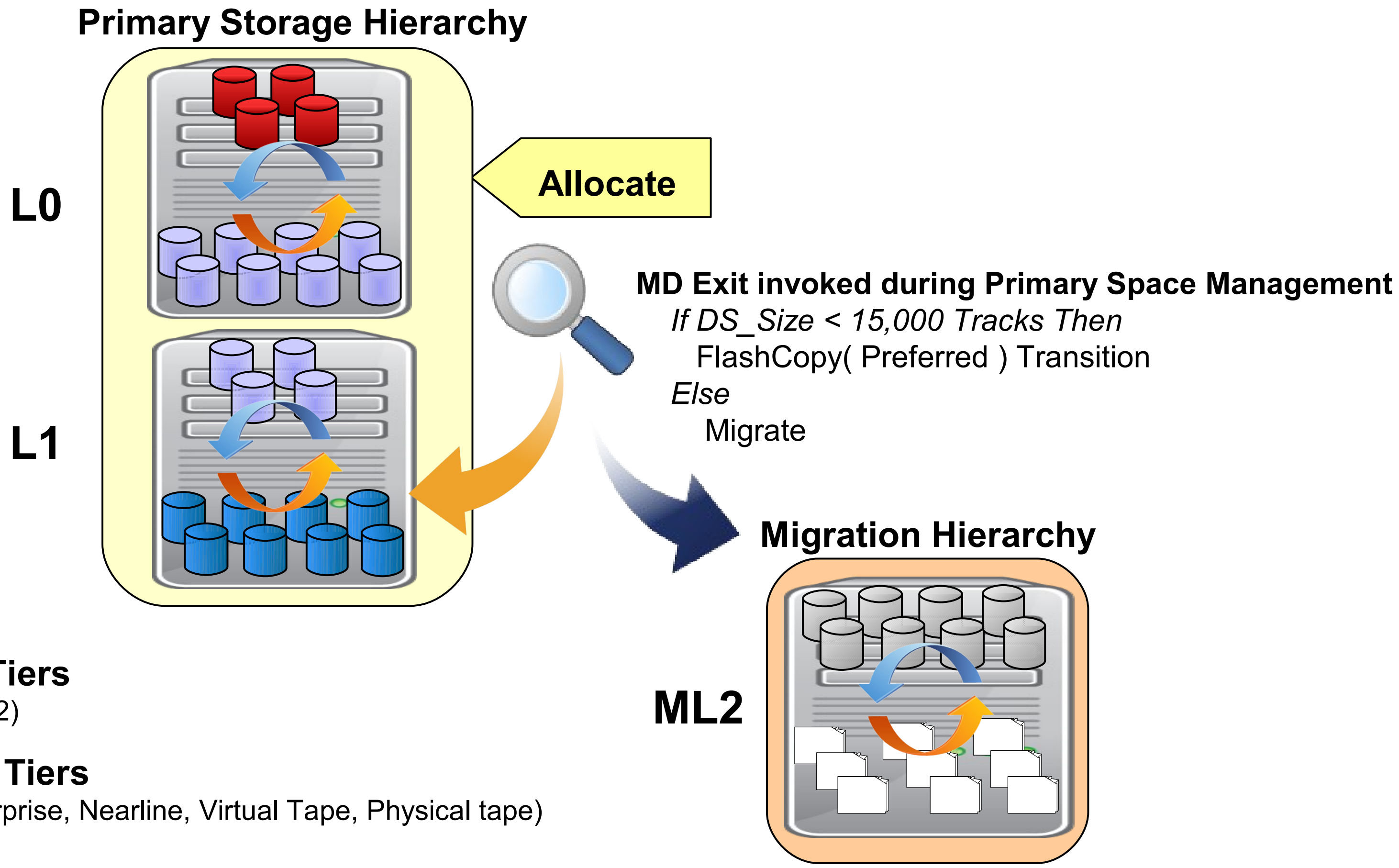
  - The default for MIGRATE DATASET is to perform a migration. The TRANSITION keyword indicates that a transition should be performed

Usage & Invocation – 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

Usage & Invocation – Storage Tiers

MD Exit enhanced to enable RC20 – 40 to be used to override destination. Transition may be converted to Migration. (V2R1 APAR OA46627)



MD Exit invoked during Primary Space Management

```
If DS_Size < 15,000 Tracks Then
  FlashCopy( Preferred ) Transition
Else
  Migrate
```

3 Logical Tiers
(L0, L1, ML2)

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

Usage & Invocation – Storage Tiers

- Enhancement to Backup requirements
  - V2R1: All forms of Fast Replication can only be used if the change indicator is OFF and there is a valid HSM Backup copy
  - V2R2:
    - If Preserve Mirror Required is specified, the change indicator may be ON
    - New Patch to indicate that a nonHSM backup utility is being used. Fast Replication may be used if the change indicator is off and there isn't an HSM backup

```
PATCH .MGCB.+111 BITS(.... ..1.)
```

Usage & Invocation – Data Migration

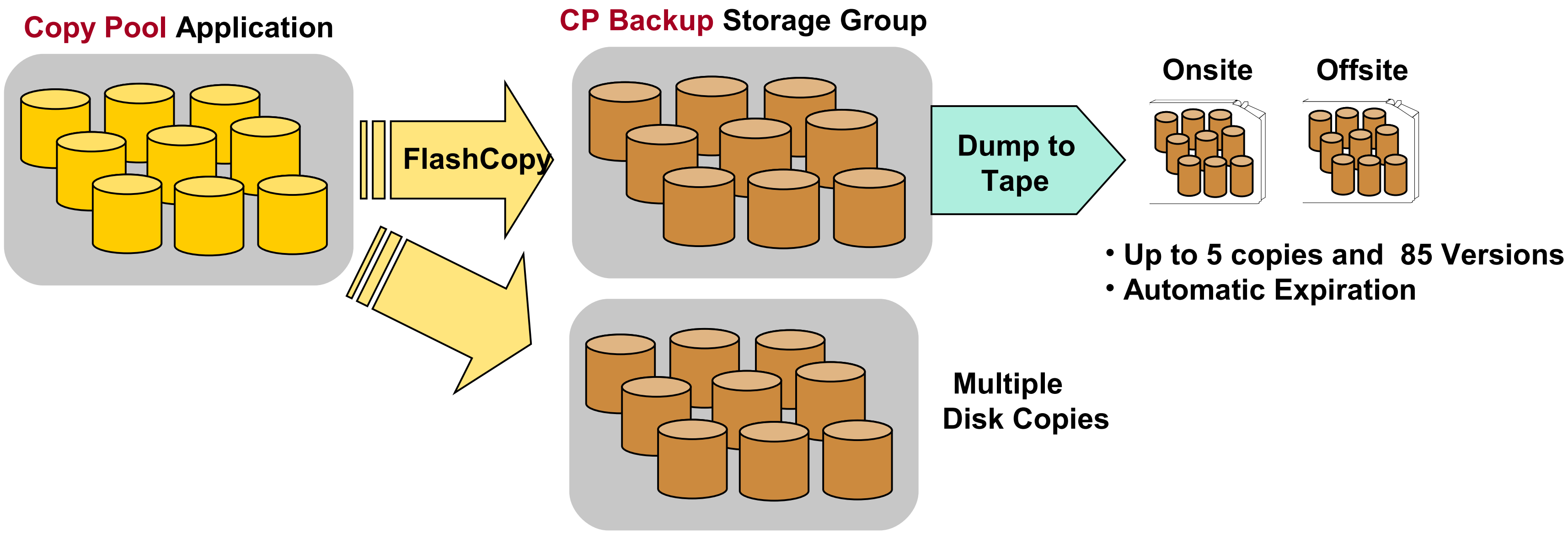
- MOVE keyword added to the MIGRATE DSNAME, VOLUME and STORAGEGROUP commands
  - Every data set will be processed, regardless of management class policy or threshold
  - ACS routines will be invoked to determine the new storage class and/or storage group
  - Management class Serialization Error logic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

Usage & Invocation – 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: The existing volumes are placed 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

Overview – zCDP for DB2

- Continuous Data Protection for DB2 is a joint solution between DFSMS and DB2 (BACKUP SYSTEM utility) in which Point-in-Time backup copies managed by DFSMS can be combined with the DB2 log to recover a system or object to any point in time



- Up to 5 copies and 85 Versions
- Automatic Expiration

Overview – Common Dump Queue

- **Problem**
  - Dump (Command, Auto, Fast Replication) commands must be processed on the same DFSMShsm host that initiated the request
    - The resources of a single DFSMShsm host do not scale to the large copy pool sizes that need to be processed
- **Solution**
  - Distribute dump work across all hosts in a sysplex-wide Common Dump Queue (CDQ)
- **Benefit / Value**
  - Improves overall throughput by distributing the workload across the sysplex instead of concentrating it on a single LPAR

Overview – Dump Enhancements

- **Problem**
  - For Dump Stacking today, the stacking value takes precedence over the number of available tasks

    For example, with STACK(100) if there are 100 volumes and 10 available dump tasks, all 100 volumes will be dumped using a single task to the same tape
  - Volumes from different copy pools are not stacked onto the same tape
  - Updating the expiration date is manually intensive

- **Solution**
  - Provide a minimum stack option to indicate that fully utilizing the available dump tasks is more desirable than optimizing stacking
  - Allow stacking onto the same tape volumes from different copy pools for the same dump class
  - Provide a new command to update the expiration date

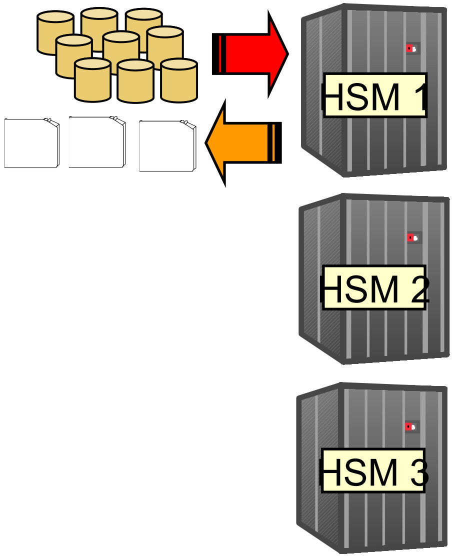
- **Benefit / Value**
  - Optimize Dump throughput and improve usability

Overview – Fast Replication Messaging

- **Problem**
  - Finding messages related to a Fast Replication command is very difficult because they are written to shared logs. You have to find the correct log and then search through all of the messages for all HSM activity
- **Solution**
  - Provide an option to dynamically allocate a unique message data set and write all DFSMShsm and DFSMSdss messages for a specific FRBACKUP or copy pool Automatic Dump to that data set
- **Benefit / Value**
  - Greatly simplify problem determination

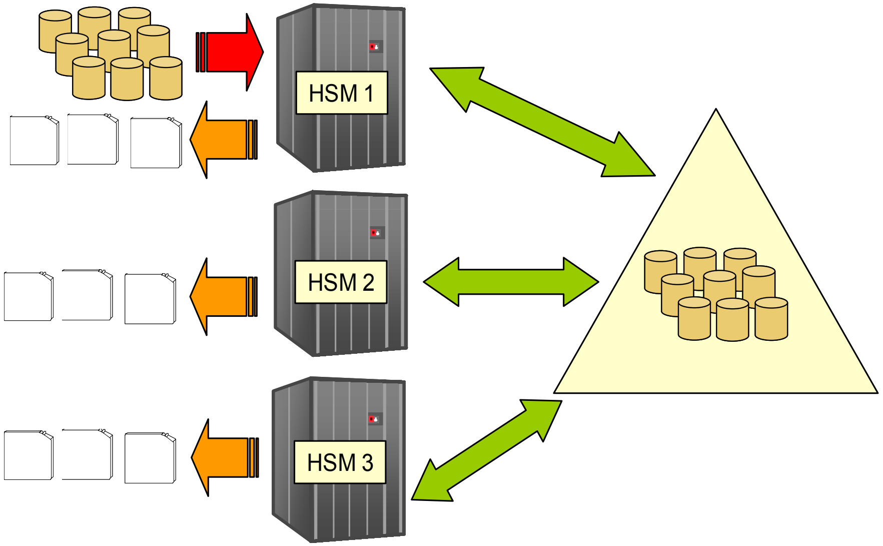
Overview – Common Dump Queue

- Today, all volumes associated with a copy pool must be dumped with a single command processed on a single host
  - Copy pools may be comprised of hundreds of volumes



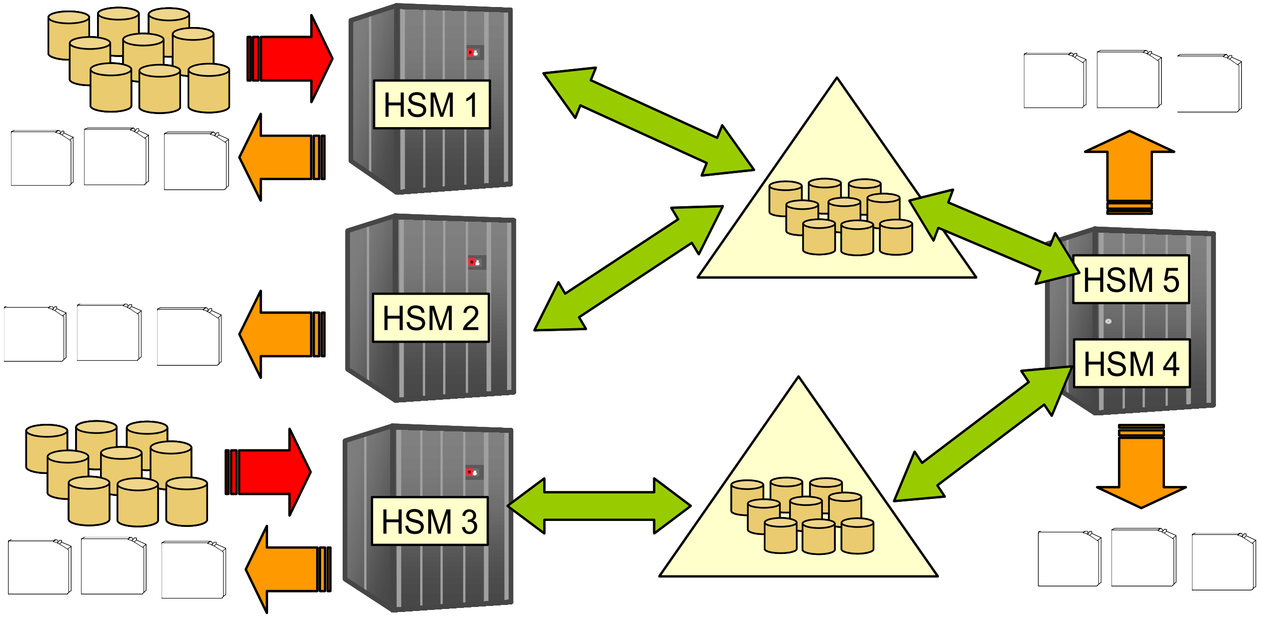
Overview – Common Dump Queue

- Introduces an architecture to distribute workload across host members in an HSMPlex
  - HSMplex-wide queue for distributing work



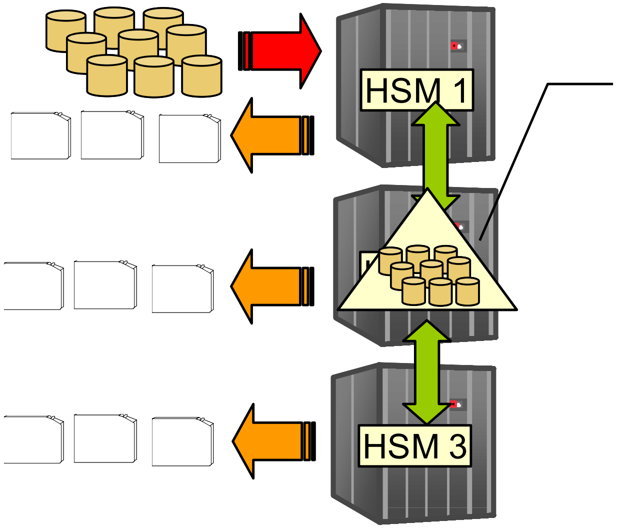
Overview – Common Dump Queue

- Architecture allows for flexible configurations
  - Multiple queues in the same HSMplex, same and different functions
    - Hosts that both receive and process requests from the group
    - Hosts that just process requests
    - Hosts that receive requests, do but not process them
  - No system affinity
  - Supports priority



Overview – Common Dump Queue

- Implementation
  - 'Common Queue' is managed by sending all requests to a 'Master Scheduler' that maintains a singe, common queue
    - Unlike the Common Recall Queue that implements a persistent CF List Structure
    - A host is assigned as the Master Scheduler and is responsible for receiving all requests and distributing them among the available tasks across all of the eligible hosts connected to the common queue
    - Implemented via XCF Group Messaging



HSM 2 is the designated Master Scheduler.

It receives all requests, managing them on a single queue, and distributes the requests among the eligible hosts connected to the common queue, through XCF Messaging.

HSM1 is the submitting and processing host

HSM2 is the MS and processing host

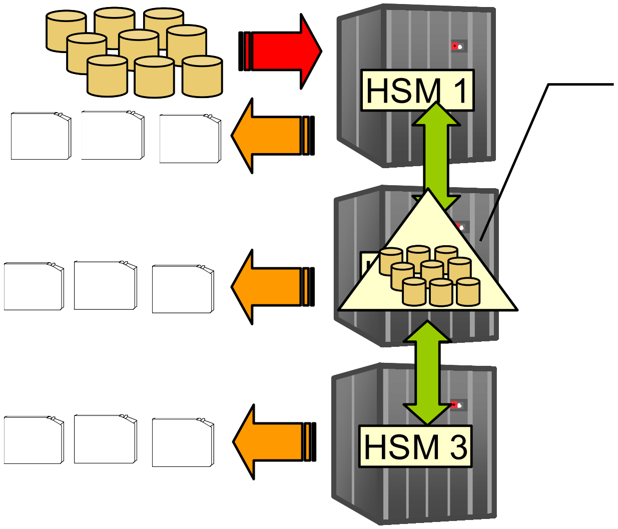
HSM3 is a processing host

Overview – Common Dump Queue

- Terminology and function
  - *Submitting Host*
    - Host that receives requests and using XCF messaging sends it to the *Master Scheduler (MS) host*
    - Processes command complete message from MS and posts user complete
  - *Master Scheduler*
    - Single Master Scheduler – race condition for all candidate hosts
    - The MS can be the submitting host - In this case the command is not sent
    - Accepts requests from *submitting host* members and locally submitted requests and maintains them as a single common queue of all requests for that grouped function
    - Assigns/distributes the requests among the available tasks across all of the eligible hosts (*processing hosts*) connected to the common queue
    - Manages work complete messages received from a *processing host* to support functions like STACKING
    - Sends command complete message back to *submitting host*

Overview – Common Dump Queue

- Terminology and function
  - *Processing Host*
    - Receives the assigned work request from the MS, places it on its local queue to process
    - MAXDUMPTASKS determines the number of concurrent tasks (Max 64)
    - Sends work complete messages back to MS with specific protocols to support functions like STACKING
    - The processing host could be the submitting host and/or the MS host



HSM 2 is the designated Master Scheduler.

It receives all requests, managing them on a single queue, and distributes the requests among the eligible hosts connected to the common queue, through XCF Messaging.

HSM1 is the submitting and processing host

HSM2 is the MS and processing host

HSM3 is a processing host

Usage & Invocation – Common Dump Queue

- SETSYS COMMONQUEUE(DUMP(CONNECT(*basename*)))
  - *basename* is a four character *suffix* for the XCF Group name
    Dump Group name: ARCD*basename*

    Example - **SETSYS CQ(DUMP(CONN(QUE1)))**
    Common Queue Name: **ARCDQUE1**
  - A host may only connect to a single common queue for each function
    - Recall
    - Dump
  - Member Name is **HOST#hsmid** in the XCF group
  - Master Scheduler Candidate (MSC is defaulted to YES)

Usage & Invocation – Common Dump Queue

- **D XCF,GROUP**

```
IXC331I  08.21.32  DISPLAY XCF
   GROUPS(SIZE):   ARCDQUE1(2)
```

- **D XCF,GROUP,ARCDQUE1**

```
IXC332I  08.22.00  DISPLAY XCF
 GROUP ARCDQUE1:    HOST#A             HOST#B
```

- **D XCF,GROUP,ARCDQUE1,ALL**

```
IXC333I  08.22.12  DISPLAY XCF
   INFORMATION FOR GROUP ARCDQUE1
   MEMBER NAME:            SYSTEM:     JOB ID:    STATUS:
   HOST#A                  SYSTEM1     DFHSM      ACTIVE
   HOST#B                  SYSTEM1     DFHSM      ACTIVE

 INFO FOR GROUP ARCDQUE1 MEMBER HOST#A ON SYSTEM SYSTEM1
 FUNCTION: HSM DUMP XCF GROUP
 MEMTOKEN: 01000003 000C0001  ASID: 0038  SYSID: 01000001
```

Usage & Invocation – Common Dump Queue

- SETSYS CQ(DUMP(CONNECT(QUE1))) - Use case
  - Host A and B, not connected, group not defined

```
F H*,SETSYS CQ(DUMP(CON(QUE1)))

Response from Host A
ARC1566I CONNECTION TO THE DUMP GROUP ARCDQUE1 WAS
ARC1566I (CONT.) SUCCESSFUL
ARC1569I HOST=A HAS TAKEN OVER THE MASTER SCHEDULER
ARC1569I (CONT.) RESPONSIBILITIES FOR THE COMMON DUMP QUEUE GROUP
ARC1569I (CONT.) ARCDQUE1

Response from Host B
ARC1566I CONNECTION TO THE DUMP GROUP ARCDQUE1 WAS
ARC1566I (CONT.) SUCCESSFUL
```

- Host A won the race to become the Master Scheduler
- The first connection, Host A, created the CDQ group, ARCDQUE1
- Host B connected to it when it joined the now already existing group

Usage & Invocation – Common Dump Queue

- SETSYS CQ(DUMP(MSC(Y|N))
  - Indicates whether or not a host is a master scheduler (MS) candidate
    - Default is YES
    - First host to CONNECT with a value of YES will become the MS
    - When the MS shuts down, another eligible host becomes the MS
    - NO may be specified on busy or inappropriate hosts
  - Setting remembered for subsequent CONNECT
  - More than one host should be eligible to be the Master Scheduler

- MSC setting can be combined with CONNECT
  - SETSYS CQ(DUMP(CONNECT(*basename*) MSC(Y)))
  - SETSYS CQ(DUMP(CONNECT(*basename*) MSC(N)))

- SETSYS MAXDUMPTASKS(0)
  - Indicates that a host may submit dump requests, but not process them

Usage & Invocation – Common Dump Queue

- SETSYS COMMONQUEUE(DUMP(DISCONNECT))
- SETSYS COMMONQUEUE(DUMP(DISCONNECT FORCE))
  - The current connection to a common queue for this host is removed
    - For a MS host, after all MS assigned work has completed
    - After remote work running on this host has completed
    - When FORCE is not specified, after this hosts submitted work completes in the common queue
      - When FORCE is specified, submitted work from this host will be failed but may still complete in the common queue
- If the host was the master scheduler, this role will be assigned to another candidate master scheduler host after this host disconnects
  - If none available, local processing of work takes over

Usage & Invocation – Common Dump Queue

- SETSYS CQ(DUMP(DISC)) - Use case
  - Host A and B connected to ARCDQUE1 group, Host A is MS
  - Disconnect from Host A

```
F HA,SETSYS CQ(DUMP(DISC))

Response from Host A
ARC1568I HOST=A HAS LOST ITS MASTER SCHEDULER
ARC1568I (CONT.) RESPONSIBILITIES FOR THE COMMON DUMP QUEUE GROUP
ARC1568I (CONT.) ARCDQUE1
ARC1567I DISCONNECTION FROM THE DUMP GROUP ARCDQUE1
ARC1567I (CONT.) WAS SUCCESSFUL

Response from Host B
ARC1569I HOST=B HAS TAKEN OVER THE MASTER SCHEDULER
ARC1569I (CONT.) RESPONSIBILITIES FOR THE COMMON DUMP QUEUE GROUP
ARC1569I (CONT.) ARCDQUE1
```

Usage & Invocation – Common Dump Queue

- SETSYS CQ(MSC(N)) - Use case
  - Host A and B connected to ARCDWEWE group, Host B is MS
  - Transition Host B out of being the MS

```
F HB,SETSYS CQ(DUMP(MSC(N))

Response from Host B
ARC1568I HOST=B HAS LOST ITS MASTER SCHEDULER
ARC1568I (CONT.) RESPONSIBILITIES FOR THE COMMON DUMP QUEUE GROUP
ARC1568I (CONT.) ARCDWEWE

Response from Host A
ARC1569I HOST=A HAS TAKEN OVER THE MASTER SCHEDULER
ARC1569I (CONT.) RESPONSIBILITIES FOR THE COMMON DUMP QUEUE GROUP
ARC1569I (CONT.) ARCDWEWE
```

Usage & Invocation – Common Dump Queue

- Shutdown
  - Prior to shutdown, issue SETSYS CQ(MSC(N))
    - Orderly quiesce of Master Scheduler responsibilities on this host to reassign them to a Master Scheduler candidate
    - If not issued, the Shutdown will Fail queued requests and wait for currently processed requests to complete before shutting down

- Master Scheduler loss of connectivity
  - All hosts finish currently assigned work
  - MS issues a message that it cannot complete work
  - Submitting hosts take back submitted requests
    - If a new MS is assigned, submitting hosts resubmit to new master scheduler
    - Otherwise, they become local requests

Usage & Invocation – Common Dump Queue

- QUERY SETSYS
  - Display the current SETSYS settings on the system that issued the query command
  - The dump group common queue connection settings will be displayed in ARC1500I. Format of the changed message is:

```
ARC1500I PLEXNAME=HSMplex_name,PROMOTE PRIMARYHOST={YES |
NO},PROMOTE SSM={YES | NO},COMMON RECALL QUEUE BASE NAME=
{basename | *****},COMMON RECALL QUEUE TAPEDATASETORDER=
{PRIORITY | FBID |*****},COMMON DUMP QUEUE BASE NAME=
{basename | ****} AND MSC={YES | NO |  ***}
```

Example:

```
ARC1500I PLEXNAME=ARCPLEX0,PROMOTE PRIMARYHOST=NO,
PROMOTE SSM=NO,COMMON RECALL QUEUE BASE NAME=*****,
COMMON RECALL QUEUE TAPEDATASETORDER=*****,
COMMON DUMP QUEUE BASE NAME=QUE1 AND MSC=***
```

This Host requested to connect to CDQ group name, ARCDQUE1. The Master Scheduler Candidate setting was not specified but the default for it allows it to be eligible

Usage & Invocation – Common Dump Queue

- QUERY REQUEST or USER
  - Display requests that originated from the host issuing the query command that match the criteria specified on the REQUEST or USER parameters
  - For a host connected to a CDQ (not the MS), the local tasks and queue will be searched and the master scheduler member's group queue will be searched
    - XCF messaging used to send the command and receive responses to be printed on the QUERY host
  - For a host connected to a CDQ (MS), the local tasks & queue will be searched
  - Existing messages, ARC0161I and ARC0167I will be reported
    - QUERY ACTIVE also gets ARC0161I

Usage & Invocation – Common Dump Queue

- QUERY REQUEST or USER - Use case
  - Host A is the MS, Host B is a member
  - Host B submitted 3 volumes for BACKVOL processing

```
F HB,Q REQUEST
ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=B
ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=A
ARC0101I QUERY REQUEST COMMAND COMPLETED ON HOST=A
ARC0161I DUMPING VOLUME PRIM01 FOR USER **OPER**,
ARC0161I (CONT.) REQUEST 00000037 ON HOST=A
ARC0167I DUMP MWE FOR VOLUME PRIM03 FOR USER **OPER**,
ARC0167I (CONT.) REQUEST 00000037, WAITING TO BE PROCESSED,00000
ARC0167I (CONT.) MWE(S) AHEAD OF THIS ONE
ARC0161I DUMPING VOLUME PRIM02 FOR USER **OPER**,
ARC0161I (CONT.) REQUEST 00000037 ON HOST=B
ARC0101I QUERY REQUEST COMMAND COMPLETED ON HOST=B
```

Query sent to Host A, the MS. Host A reported messages get 'XCF messaged' back to Host B for printing. Host B reported messages are also printed

Usage & Invocation – Common Dump Queue

- QUERY ACTIVE
  - Display the current status on the system that issued the query command
  - The dump group common queue active status will be displayed in new message ARC1560I:

```
ARC1560I COMMON type QUEUE FACTORS: GROUPNAME={groupname | *},
CONNECTION STATUS={CONNECTING | CONNECTED | DISCONNECTING |
UNCONNECTED | QUIESCED  },  MASTERSCHEDULER HOSTID={hostid |
NONE | *}, MASTER SCHEDULER STATUS={DISABLED | CANDIDATE |
ENABLED | *}
```

Example:

```
ARC1560I COMMON DUMP QUEUE FACTORS: GROUPNAME=ARCDQUE1,
ARC1560I (CONT.)  CONNECTION STATUS=CONNECTED, MASTERSCHEDULER
ARC1560I (CONT.) HOSTID=A, MASTER SCHEDULER STATUS=ENABLED
```

Usage & Invocation – Common Dump Queue

- QUERY WAITING
  - Lists the number of requests that originated from the host issuing the query command that are waiting
  - For a host connected to a CDQ (not the MS), the local queue will be searched and the master scheduler member's group queue will be searched
    - XCF messaging used to send and receive responses to be printed on the QUERY host
  - For a host connected to a CDQ (MS), the local queue will be searched
  - Existing messages, ARC0168I and ARC1542I will be reported

```
ARC1542I WAITING MWES ON COMMON QUEUES: COMMON RECALL
QUEUE=type_mwes, COMMON DUMP QUEUE=type_mwes, TOTAL=total_mwes
```

Usage & Invocation – Common Dump Queue

- QUERY COMMONQUEUE
  - Display all the common queue requests queued or active in the group
    - New DUMP option on QUERY CQ
  - XCF messaging used to send and receive responses to be printed on the QUERY host
  - New messages ARC1562I (active) and ARC1563I (waiting) may be issued

```
ARC1563I VOLUME  {volser | volser, SGROUP=sg | volser, COPY POOL
= cpname, }  {DUMPING | FRBACKUP DUMP  | FRBACKUP DUMPONLY  |
RESTORING }   FOR USER  {userid | **AUTO** | * } REQUEST
{request-number [FROM HOST=hostid] | NONE }, WAITING TO BE
PROCESSED ON A COMMON QUEUE, nmwe MWES AHEAD OF THIS ONE,
GROUPNAME=groupname

ARC1562I VOLUME  {volser | volser, SGROUP=sg | volser, COPY POOL
= cpname, }  {DUMPING | FRBACKUP DUMP  | FRBACKUP DUMPONLY  |
RESTORING } , FOR USER  {userid | **AUTO** | * } REQUEST
{request-number [ON HOST=hostid] | NONE }, BEING PROCESSED FROM
A COMMON QUEUE, GROUPNAME=groupname
```

Usage & Invocation – Common Dump Queue

- HOLD DUMP | DUMP(AUTO) | DUMP(FRBACKUP)
  - HOLD continues to prevent dump processing based on its current options
    - Dump tasks not selected to process requests
    - WAIT types failed
  - With CDQ 'any' specified HOLD
    - Prevents a host from being selected to process CDQ work
    - On a MS prevents MS duties from being performed
  - Best reserved for stopping already active dump processing/scheduling
    - To configure a CDQ host that will not be selected for dump processing, it may preferable to use SETSYS MAXDUMPTASKS(0)
      - Host can still submit DUMP requests to CDQ
      - Requests are processed in other available dump hosts in the CDQ
      - If host is the MS, this will not impact MS responsibilities

Usage & Invocation – Common Dump Queue

- If any variant of RELEASE DUMP is issued on a CDQ host:
  - That host will be eligible to be selected for CDQ dump processing if the following conditions are met:
    - No other HOLD for DUMP is in place for the host.
    - SETSYS MAXDUMPTASKS(x) is specified with a value greater than 0
  - If the host is the CDQ MS, it will also be eligible to perform MS duties for the particular dump function that has been released

Usage & Invocation – Common Dump Queue

- CANCEL
  - Cancels queued requests that originated on the cancel host for a request number or userid that matches the CANCEL request parameters
  - Applies to CDQ when the host is connected to a common queue
  - Canceling active requests with the TCBADDRESS must be done on the host that is processing the request
  - XCF messaging used to send the command and receive responses to be printed on the CANCEL host
    - For a MS host, the command is processed locally
    - For a non MS host, the command is processed locally and on the MS
- Existing and unchanged messages, ARC1008I and ARC0931I will be reported

Usage & Invocation – Common Dump Queue

- ALTERPRI
  - Changes the priority of queued requests waiting to be processed in DFSMShsm for a request number or userid matching the alter request
    - Applies to CDQ when the host is connected to a common queue
  - XCF messaging used to send the command and receive responses to be printed on the ALTERPRI host
    - For a MS host, the command is processed locally
    - For a non MS host, the command is processed locally and on the MS
  - Existing messages, ARC0981I and ARC0982I will be reported

```
ARC0980I ALTERPRI {REQUEST | USER | DATASETNAME} COMMAND
STARTING ON HOST=hostid

ARC0981I ALTERPRI {REQUEST | USER | DATASETNAME} COMMAND
COMPLETED ON HOST=hostid,RC=retcode
```

Usage & Invocation – Dump Improvements

- DEFINE DUMPCLASS

  – Added a new MINSTACK, minimum stack, parameter and an MAXSTACK alias for the STACK parameter

```
|___ _______________________ ___ ________________ ___|
    |__ _STACK____ _(maxv)__|   |__MINSTACK(minv)_|
       |_MAXSTACK_|
```

  – Maxv represents the preferred maximum number of dump copies that the system should use to place on a tape volume

  – Minv represents the preferred minimum number of dump copies that the system should use to place on a tape volume

    - Utilizes more dump tasks, but may have less dump copies on tape

  – MAXSTACK value must be equal to or larger than the MINSTACK value

  – If MINSTACK value not specified the MAXSTACK value used for it

  – Example, DEFINE DUMPCLASS(DCLASS1 MAXSTACK(100) MINSTACK(20))

Usage & Invocation – Dump Improvements

- STACK and MINSTACK Use Case
  - 100 volumes to be dumped
  - Dump class with MAXSTACK|STACK(100)
  - 5 dump tasks
  - **Today**
    - 100 volumes are stacked onto 1 tape, utilizing 1 dump task
  - **Enhancement**
    - Specify MINSTACK(20)
    - First 100 volumes dumped
      - 20 volumes will be stacked onto 5 tapes, utilizing all 5 dump tasks

- MINSTACK may also be specified on the BACKVOL command

Usage & Invocation – Dump Improvements

- Multiple Copy pools may now be stacked onto the same tape
  - This will occur when the copy pools are dumped at the same time and use the same dump classes
  - This only occurs with Automatic Dump today

- New UPDTCDS command to update the expiration date for a copy pool dump copy
  - Today, in can take *hundreds* of FIXCDS commands to update the expiration date of a single copy pool dump version

```
UPDTCDS COPYPOOL(name) VERSION(ver)
                    DUMPEXPIRATION(DCLASS(name) NEWDATE(date))
```

  - This new command updates all of the CDS records that need to be updated to reflect the new expiration date.
    - The Fast Replication Dump record
    - The dump record for each volume dumped
  - This command will be extended to other functions in the future

Usage & Invocation – Message Data Set

SETSYS FASTREPLICATION(MESSAGEDATASET(YES|NO HLQ(hlq)))

- Indicates that all Fast Replication messages associated with an FR command or copy pool automatic dump should go to a unique data set
- Message Data Sets must be SMS managed, so SMS and ACS routines must be setup prior to allocation
- DFSMShsm dynamically allocates message data sets with the following characteristics:
  - Physical Sequential
  - FBA
  - LRECL of 121
  - Block size of 1210
- Naming Convention: *msgdsprefix.*FB.C*ccccccc*.i*ppppppp.*D*yyddd*.T*hhmmss.z*
- Collisions

  Time is incremented if a duplicate name already exists

Usage & Invocation – Message Data Set

*msgdsprefix.*FB.C*ccccccc*.i*ppppppp.*D*yyddd*.T*hhmmss.z*

msgdsprefix is the value of the HLQ parameter of the SETSYS command. If the HLQ parameter is not specified, the default prefix of 'HSMMSG' will be used.

ccccccc is up to 7 characters to identify the copy pool that was processed. For a non-DB2 copy pool, this will be the first 7 characters of the copy pool name. For a DB2 copy pool, this will be the first 7 characters of the location qualifier in the DB2 copy pool name.

i is an identifier, indicating whether the message data set contains messages for:

D – DB2 data   L – DB2 log X – non-DB2 copy pool data

ppppppp is up to 7 characters to identify the copy pool that was processed. For a non-DB2 copy pool, this will be the next 7 characters of the copy pool name (the characters following *ccccccc*). For a DB2 copy pool, this will be the next 7 characters of the location qualifier in the DB2 copy pool name.

yyddd is the year and day when the FRBACKUP or copy pool auto dump operation started.

hhmmss is the time in hours, minutes, and seconds when the FRBACKUP or copy pool auto dump operation started.

z indicates the status of the FRBACKUP or copy pool auto dump operation, and will be set to one of the following values:

S – Success, F – Fail, I – In progress

Usage & Invocation – Message Data Set

- Messages indicating parse errors or termination prior to the ARC1801I or ARC1841I start messages are not collected in message data sets.
  - A message data set is allocated after the request is determined eligible for processing.

- Each message is prefixed with the date and time that the message was written to the data set, along with the ID of the host the wrote the message:

  yy/mm/dd hh:mm:ss Hx *message-text*

  - where H is a constant, and *x* is the host id

- Some messages will appear multiple times in a message data set.
  - Each time a message is written to the operator or a log, the message is also logged in the message data set. Duplicates are not removed.

```
14/11/21  10:09:10  H2  ARC1801I FAST REPLICATION BACKUP DUMP IS
STARTING FOR COPY POOL DSN$PAYROLLA123$DB, AT 10:09:10 ON 2014/11/21
14/11/21  10:09:11  H2  ARC1805I THE FOLLOWING 00003 VOLUME(S) WERE
SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL
DSN$PAYROLLA123$DB
14/11/21  10:09:11  H2  ARC1805I (CONT.) SRC001
14/11/21  10:09:11  H2  ARC1805I (CONT.) SRC003
14/11/21  10:09:11  H2  ARC1805I (CONT.) SRC002
14/11/21  10:09:11  H1  ARC1814I FAST REPLICATION BACKUP HAS COMPLETED
SUCCESSFULLY AND DUMP MWES HAVE BEEN QUEUED FOR COPY POOL
DSN$PAYROLLA123$DB, VERSION 3
14/11/21  10:09:11  H1  ARC0622I FULL VOLUME DUMP STARTING ON VOLUME
SRC001(SMS) AT 10:09:11 ON 2014/11/21, SYSTEM 3090, TASK ID=ARCDV001,
TO DUMP CLASS(ES)=   DCLASS1
14/11/21  10:09:11  H1  ARC0640I ARCDV001 - PAGE 0001     5695-DF175
DFSMSDSS V2R02.0 DATA SET SERVICES     2014.325 10:09
14/11/21  10:09:11  H1  ARC0640I ARCDV001 - ADR035I (SCH)-PRIME(06),
INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK
14/11/21  10:09:11  H1  ARC0640I ARCDV001 -  DUMP FULL
OUTDDNAME(SYS00070) 14/11/21  10:09:11  H1  ARC0640I ARCDV001 -
ALLEXCP ALLDATA(*) OPTIMIZE(1) TOLERATE(IOERROR) ...
```

Usage & Invocation – Message Data Set

New Message

ARC1870I MESSAGE DATA SET *msgdsname* CREATED FOR {FAST REPLICATION {BACKUP | BACKUP DUMP | BACKUP DUMPONLY | PREPARE } | AUTO DUMP} OF {COPY POOL *cpname* | VOLUME *volser* | DATA SET *dsname*}

QUERY SETSYS

ARC1823I MAXCOPYPOOL(FRBACKUP TASKS = *backup_tasks*, FRRECOV TASKS = *recover_tasks*,DSS TASKS = *dss_tasks*), FASTREPLICATION (DATASETRECOVERY = {PREFERRED | REQUIRED | NONE } FCRELATION = {EXTENT | FULL} VOLUMEPAIRMESSAGES = {YES | NO} MESSAGEDATASET=({YES | NO} {HLQ=*hlq*}))

Interactions & Dependencies

- **Software Dependencies**
  - None
- **Hardware Dependencies**
  - zEDC Hardware
- **Exploiters**
  - DB2 BACKUP SYSTEM Utility can exploit these new functions with no additional DB2 support. It is transparent to the utility.
  - DB2 plans to reference the new Message Data Sets
- **ZEDC PTFs**
  - **DSS (HSM) Coexistence:** UA74780 (C10), UA74781 (D10)
  - **DSS V2R1 Support:** UA74782
  - **HSM V2R1 Support:** UA74836
  - **IOS:** UA74718

Migration & Coexistence Considerations

- **Migration**
  - None
- **Coexistence**
  - DFSMSdss
    - Will allow a user V1R12 and V1R13 release to restore backups created using zEDC services
    - Software inflate is used
  - DFSMShsm
    - Will allow a user on a V1R12 and V1R13 release to RECALL, RECOVER, RECOVER from DUMP, or FRRECOV from DUMP data sets migrated, backed up or dumped using zEDC Services on V2R1
    - Will leverage the coexistence support provided by DFSMSdss

Presentation Summary

- zEDC can significantly reduce MIPS and improve the compaction ratio for DFSMSdss and hsm
- DFSMS is committed to continual enhancements and improvements to managing the storage hierarchy and the z/OS DB2 Continuous Data Protection solution.

Appendix

- DFSMS Using the New Functions, SC23-6857
- DFSMSdss Storage Administration, SC23-6868
- DFSMShsm Storage Administration, SC23-6871
- DFSMS Installation Exits, SC23-6850
- MVS System Messages Vol 2, SA38-0668
- zEnterprise Data Compression FAQ

https://www-304.ibm.com/support/docview.wss?uid=tss1fq131484&aid=1

- Configuring zEDC on z/OS (a cheat sheet)

https://www-304.ibm.com/connections/blogs/systemz/entry/configuring_zedc_on_zos?lang=en_us