IBM GLOBAL SERVICES

Session B70

IMS HALDB Implementation Considerations

Steve Nathan

IMS
Technical Conference

Sept. 27-30, 2004

Orlando, FL

RETURN TO INDEX

Agenda

- **HALDB Databases**
- **HALDB Partition Selection Exit**
- **HALDB Partitions**
- **HALDB Naming Conventions**
- **HALDB and DBRC**
- **HALDB Partition Initialization**
- **HALDB Initial Load**
- **HALDB Database Migration**
- **HALDB Database Buffer Pools**
- **HALDB Partition Processing**
- **Self-Healing Pointers**
- **Increased Database Size with HALDB**
- **HALDB Utilities**
- **HALDB Maintenance**

HALDB Databases

- **HALDB – High Availability Large Data Base**
  - A new type of IMS Full Function (as opposed to Fast Path) partitioned database available in IMS 7.1
  - Up to 1001 partitions per database
    - The architecture supports more
    - IBM could not easily test beyond 1001
  - Up to 10 data set groups per partition
  - Up to 4GB per data set (OSAM or VSAM)
  - Approximately 40 Terabytes of data
    - Over 20,000 3390's
    - Over 6,000 bytes for each person on earth

HALDB Databases

- **HALDB Databases**
  - HALDB supports Partitioned HDAM (PHDAM) and Partitioned HIDAM (PHIDAM) database types
  - HALDB supports OSAM and VSAM access methods
  - HALDB supports Secondary Indexes
    - The Secondary Indexes are also partitioned
  - HALDB supports some Logical Relationships

HALDB Databases

- **The database defined by the DBD and in the IMS SYSGEN is the "Master Database"**
  - The partitions are not defined in the DBDGEN
  - The partitions are not defined in the IMS SYSGEN
  - The partitions are defined in DBRC
    - But they can look like databases
    - /DIS DB partname will work and will show the partition
    - More later

HALDB Databases

- **PHIDAM Primary Index definitions are automatic**
  - There is no entry in the IMS SYSGEN
  - There is no DBD
  - There are no LCHILD statements in the PHIDAM database
  - This means the Primary Index can not be read as a standalone database
    - Some applications did do this
  - There is no entry for the database in DBRC
  - Updates to the PHIDAM Primary Index are not logged
    - The index must be rebuilt if there is an error

HALDB Databases

- **Determining which partition a root segment goes to is determined in one of two ways**

1. A high key can be defined for each partition with the HALDB Partition Definition Utility or batch DBRC definition
2. A Partition Selection Exit (PSE) can determine the partition

– Once a root is in a partition all of its dependent segments go into the same partition

HALDB Databases

- **A “hybrid” database type is also supported**
  - PHIDAM database with a Partition Selection Exit (PSE)
  - The PSE determines the partition for a database record in the same manner as a DEDB randomizer
  - Within the partition the roots are indexed and in key sequence
  - The data can easily be accessed sequentially within a partition
  - You do not have to allow nearly as much free space as with PHDAM

HALDB Databases

- **The application view of a HALDB database is like any other IMS database**
  - The hierarchic structure of an IMS database is not changed
  - Applications access HALDB databases with standard IMS calls
    - Availability status codes in the DBPCB are for the master database at schedule time (NA or NU)
    - A BA status code may be received for an IMS call due to a partition being unavailable

HALDB Databases

- **Secondary indexes for HALDB databases must also be partitioned – PSINDEX**
  - The partitioning is independent of the partitioning of the primary database
    - There can be a different number of partitions
    - One PSINDEX partition can contain records pointing to multiple primary database partitions
      - This could affect a timestamp recovery of a primary partition
      - A Partition Selection Exit on the PSINDEX database could force the partitioning to be the same but it would be very hard to do

HALDB Partition Selection Exit

- **THE HALDB Partition Selection Exit (PSE) does more than just determine the partition for a root**
  - The HALDB Partition Selection Exit is invoked for:
    - Structure Initialization
    - Structure Termination
    - Structure Modification
    - Select First Partition
    - Select Next Partition
    - Select Target Partition
  - Passed 4 pre-chained save areas
  - Passed a 512-byte reentrant work area
  - Has a field which can be valued at Structure Initialization and passed to the other invocations
    - Allows a user table to be built and processed and freed

HALDB Partition Selection Exit

- **In your PSE if you do not like the key you may want to WTO an error message describing what was wrong with the key**
  - If this is not done properly you may abend IMS
  - The PSE may be invoked in cross-memory mode and SVC's (e.g. WTO) are not allowed in cross-memory mode
  - You have to add code to see if you are in cross-memory mode and if so use the Branch Entry WTO
    - This is illustrated in the next two foils

HALDB Partition Selection Exit

```
          LA    R7,WTOMSG              SET ERROR MESSAGE TEXT
          BAL   R10,DOWTO              GO DO WTO

DOWTO     DS    0H
          LA     R5,PECUSER            ADDRESS OF 512-BYTE WORK AREA
          USING WTOD,R5                TELL ASSEMBLER
          MVC    WTOMSGE,WTOMSGL       MOVE MESSAGE MASK
          MVC    WTOTEXT,0(R7)         MOVE MESSAGE TEXT
          EPAR   R0                    PRIMARY ASID
          ESAR   R1                    SECONDARY ASID
          CLR    R0,R1                 PRIMARY = SECONDARY?
          BNE    DOWTOX                   NO - IN XM MODE
          WTO    MF=(E,WTOMSGE)        WTO TRACE MESSAGE
          BR     R10                   RETURN TO CALLER
DOWTOX    DS    0H
          WTO    MF=(E,WTOMSGE),LINKAGE=BRANCH
          BR     R10                   RETURN TO CALLER
          DROP   R5                    TELL ASSEMBLER
          EJECT
```

HALDB Partition Selection Exit

```
WTOMSGL  WTO    `                                                ',    X
               ROUTCDE=(11),DESC=(7),MF=L

WTOD     DSECT
WTOMSGE  DS    0CL58                  REENTRANT WTO MESSAGE
         DS     CL2                   TEXT LENGTH
         DS     CL2                   MCSFLAGS
WTOTEXT  DS     CL50
         DS     CL4                   DESC/ROUT CODES
```

HALDB Partition Selection Exit

- **The main control block passed to the PSE is the Partition Exit Communication Area (DFSPECA)**
  - There was no DSECT for this in the IMS Macro Library
    - It was only documented in the sample PSE exit (DFSPSE00)
  - This also applied to the other two PSE DSECTS
    - Partition Definition Area Prefix (DFSPDA)
    - Partition Definition Area Entry (DFSPDAE)
  - New macro DFSPSEIB now documents these DSECTs
    - PQ88078 (V7) – PQ88679 (V8) – PQ87622 (V9)

HALDB Partitions

- **Each HALDB partition is independent**
  - Allocation
  - Authorization
  - Reorganization
  - Recovery
  - Some commands

HALDB Partitions

- **Each partition has a name**
  - There are many places where you have to refer to the Partition name where you would normally use a real database name
    - IMS commands
    - DBRC commands
    - Utility control cards

HALDB Partitions

- **The documentation is not clear on when the master database name should be used and when the partition name can/should be used**
  - IMS documentation (Dean Meltz) has been extremely responsive in fixing these as they have been found
  - The latest version of *The Complete IMS HALDB Guide* (Redbook SG-24-6945) has a detailed list of DBRC commands and which “database” name to use
    - This book is a “must” for HALDB users
    - Thanks to Jouko Jantti, Rich Lewis and Pat Schroeck for their instant response to questions and suggestions

HALDB Naming Conventions

- **HALDB has special naming conventions**
  - DBD names are 1-8 characters
  - Partition names are 1-7 characters
    - If you have the partition number in the partition name and you can have 1001 partitions you only have 3 characters left for uniqueness
  - DDNAMES are the Partition Name + an IMS assigned suffix
    - xxxxxxxA – xxxxxxxJ for data set groups 1-10 (set 1)
    - xxxxxxxM – xxxxxxxV for data set groups 1-10 (set 2)
    - xxxxxxxX for the primary index (set 1)
    - xxxxxxxY for the primary index (set 2)
    - xxxxxxxL for the ILDS data set

HALDB Naming Conventions

- **HALDB has special naming conventions**
  - Data set names are a user defined (37-byte maximum) prefix plus an IMS assigned 7 byte suffix
    - .Annnnn - .Jnnnnn - Data set groups 1-10 (set 1)
    - .Mnnnnn - .Vnnnnn - Data set groups 1-10 (set 2)
    - .Xnnnnn - Primary Index (set 1)
    - Ynnnnn - Primary Index (set 2)
    - .Lnnnnn – ILDS
  - nnnnn is the partition number
    - 00001 – 01001
  - There can be a different prefix for each partition

HALDB Naming Conventions

- **You have to plan ahead for naming conventions to make Partition Names and DDNAMES and data set names unique**
  - Strongly consider using the DBD name or Partition name (or both) as a middle data set node
  - These naming conventions also force the same high-level node for OSAM and VSAM data sets
    - The Primary Index and the ILDS and the data all have the same prefix
    - Some organizations have standards that require different high-level nodes

HALDB Naming Conventions

- **Database data set allocation is based on the data set names in DBRC**
  - The user prefix defined in DBRC + the 7-byte IMS suffix
  - There are no DFSMDA dynamic allocation members
  - There are no DD cards
    - You can use them
      - If they match DBRC allocation is successful
      - If they do not match DBRC allocation will fail
        - Never use DD cards

HALDB and DBRC

- **HALDB databases MUST be defined to DBRC**
  - ALL HALDB databases
    - Production
    - Training
    - Test
    - Development
  - The database hierarchy (datasets, segments, fields, pointer options, secondary indexes, logical relationships) are defined in the DBD
  - The partitions are defined in DBRC – not in the DBD
    - There are no AREA statements in the DBD

HALDB and DBRC

```
 LIST.DB DBD(DB01MAST)
04.238 08:33:38.2                    LISTING OF RECON                    PAGE 0002
-------------------------------------------------------------------------------
 DB
  DBD=DB01MAST                          DMB#=35     CHANGE#=54     TYPE=HALDB
  SHARE LEVEL=0                 GSGNAME=**NULL**
  PSNAME=DB01PSEX  DBORG=PHIDAM   DSORG=OSAM    CURRENT PARTITION ID=00053
  FLAGS:                            COUNTERS:
    RECOVERABLE           =YES        PARTITIONS               =53
                                      DATA SET GROUP MEMBERS   =1
```

This is the DBRC LIST.DB for the Master Database

There is very little information at this level

HALDB and DBRC

```
--------------------------------------------------------------------------------
04.238 08:33:38.2                    LISTING OF RECON                   PAGE 0003
--------------------------------------------------------------------------------
 DB
  DBD=DB01P01  MASTER DB=DB01MAST                  CHANGE#=54    TYPE=PART
  USID=0000000001  AUTHORIZED USID=0000000000  HARD USID=0000000000
  RECEIVE USID=0000000000  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  DSN PREFIX=HIGHNODE.DB01P01                                PARTITION ID=00001
  PREVIOUS PARTITION=**NULL**    NEXT PARTITION=**NULL**
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J      M-V EXIST=NO
   FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0
   PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=4   )

  PARTITION HIGH KEY/STRING (HEX):
    00000001404040404040404040404040404040404040404040404040404040
```

This is the start of the DBRC LIST.DB for the first partition
All of the information is at this level

HALDB and DBRC

```
OSAM BLOCK SIZE:
    A = 4096
   FLAGS:                                COUNTERS:
    BACKOUT NEEDED          =OFF          RECOVERY NEEDED COUNT    =0
    READ ONLY               =OFF          IMAGE COPY NEEDED COUNT  =0
    PROHIBIT AUTHORIZATION=OFF            AUTHORIZED SUBSYSTEMS    =0
                                          HELD AUTHORIZATION STATE=0
                                          EEQE COUNT               =0
    TRACKING SUSPENDED      =NO           RECEIVE REQUIRED COUNT   =0
    OFR REQUIRED            =NO
    PARTITION INIT NEEDED =NO
    ONLINE REORG ACTIVE     =NO
    PARTITION DISABLED      =NO
```

This is the end of the DBRC LIST.DB for the first partition
All of the information is at this level

HALDB and DBRC

```
--------------------------------------------------------------------------------
04.238 08:33:38.2                    LISTING OF RECON                    PAGE 0004
--------------------------------------------------------------------------------
 DBDS
  DSN=HIGHNODE.DB01P01.A00001                                          TYPE=PART
  DBD=DB01P01   DDN=DB01P01A DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=2      IC AVAIL=0      IC USED=2      DSSN=00000008
  NOREUSE         RECOVPD=0       OTHER DDN=**NULL**
  DEFLTJCL=**NULL**  ICJCL=ICJCL     OICJCL=OICJCL    RECOVJCL=RECOVJCL
  RECVJCL=ICRCVJCL
  FLAGS:                           COUNTERS:
    IC NEEDED       =OFF
    RECOV NEEDED    =OFF
    RECEIVE NEEDED =OFF               EEQE COUNT              =0
```

This is the start of the DBRC LIST.DBDS for the first partition

HALDB and DBRC

```
---------------------------------------------------------------------------
  ALLOC
  ALLOC    =04.229 09:51:50.3              *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 04.229 04:52:21.6
  DEALLOC =04.229 11:56:27.1               DEALLOC LRID =0000000000000000

  IMAGE
  RUN      = 04.225 17:48:15.1             *  RECORD COUNT =2
  STOP     = 00.000 00:00:00.0                BATCH      USID=0000000001
  IC1
  DSN=HIGHNODE.DB01P01.POSTINIT.IC1.G0001V00       FILE SEQ=0001
  UNIT=3380                           VOLS DEF=0001 VOLS USED=0001
                                      VOLSER=SMN471

  REORG
  RUN      = 04.229 11:57:20.4             *    USID = 0000000003
```

This is more of the DBRC LIST.DBDS for the first partition

HALDB and DBRC

```
-------------------------------------------------------------------------------
04.238 08:33:38.2                  LISTING OF RECON                  PAGE 0005
-------------------------------------------------------------------------------
 DBDS
  DSN=HIGHNODE.DB01P01.L00001                                    TYPE=PART
  DBD=DB01P01   DDN=DB01P01L DSID=003 DBORG=INDEX  DSORG=VSAM
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------
04.238 08:33:38.2                  LISTING OF RECON                  PAGE 0006
-------------------------------------------------------------------------------
 DBDS
  DSN=HIGHNODE.DB01P01.X00001                                    TYPE=PART
  DBD=DB01P01   DDN=DB01P01X DSID=005 DBORG=INDEX  DSORG=VSAM
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
```

This is the end of the DBRC LIST.DBDS for the first partition
The ILDS and the Primary Index have DBDS records
There is one ILDS and one Primary Index per Partition

There are no Allocation or Image Copy or Reorg records
for these data sets

HALDB and DBRC

- **In the display of the partition in DBRC you see data set ID's for the data sets**
  - xxxxxxxA – DSID=0001
  - xxxxxxxL – DSID=0003
  - xxxxxxxX – DSID=0005
  - xxxxxxxB – DSID=0006
  - xxxxxxxC – DSID=0007
  - etc.
  - DSID's 0002 and 0004 are used for the index components of the ILDS and Primary Index for I/O error purposes

HALDB and DBRC

- **This will be no fun for your development and test organizations**
  - They will have to learn more about DBRC than they ever wanted to
    - Or your DBA's and SYSPROG's will be busier than they ever wanted to be
  - Some of the commands they may need to become familiar with:
    - CHANGE.DBDS DBD(part) DDN(part) ICOFF
    - CHANGE.DBDS DBD(part) DDN(part) NORECOV
    - CHANGE.DB DBD(part) NOBACK SSID(jobname)
    - CHANGE.DB DBD(part) UNAUTH SSID(jobname)
    - CHANGE.SUBSYS SSID(jobname) STARTRCV
    - CHANGE.SUBSYS SSID(jobname) ENDRECOV
    - DELETE.SUBSYS SSID(jobname)

HALDB and DBRC

- **When HALDB was first announced the only way to define the partitions in DBRC was via the HALDB Partition Definition Utility**
  - This is a time-consuming ISPF application
  - You would have had to go through several hundred ISPF panels to define 50 partitions for each of 8 HALDB's
  - Customers told IBM that HALDB's would not be usable without batch DBRC support for HALDB

HALDB and DBRC

- **There is now batch DBRC support for HALDB**
  - For IMS 7.1 support was added via PQ35893 (UQ49705) for INIT.DB and INIT.PART so you could at least define the partitions with batch DBRC
    - There was also limited CHANGE.DB support
  - IMS 8.1 added support for CHANGE.DB, CHANGE.PART, DELETE.DB and DELETE.PART for HALDB databases and partitions
    - This was propagated to IMS 7.1 with PQ59888 (UQ69393)

HALDB Partition Initialization

- **When the HALDB partitions are first defined to DBRC each partition has a status of PINIT**
  - Partition Initialization is required
  - HALDB database partitions are initialized in one of two ways
    - HALDB Partition Data Set Initialization Utility (DFSUPNT0)
      - Can initialize a list of Master databases
    - IMS Database Prereorganization Utility (DFSURPR0)
      - DBIL=Master DBD name

HALDB Partition Initialization

- **When the Partition Initialization utility is run for a Master database it will initialize all partitions in a PINIT status**
  - This means ALL partitions have to be allocated
  - This may not be what is wanted for development or test or training databases
  - DFSUPNT0 can initialize all partitions in any PINIT state
    - PQ49638 (IMS V7)

HALDB Partition Initialization

- **DFSUPNT0 (V9) can be more selective**
  - It will process a list of partitions
    - It will initialize those partitions
      - Even if the PINIT flag is off
    - But it also initialized any other partitions in the database with the PINIT flag on!!!
      - This was not what customers had requested
      - PQ85497 finally made this utility work correctly
        - > Initialize only those partitions that are listed

HALDB Partition Initialization

- **Prior to IMS V9 there is an alternative HALDB DBRC definition procedure**
  - It is run once per database and has three steps
    - Define the database
      - INIT.DB DBD(xxxxxxxx)
    - Define all of the HALDB partitions
      - INIT.PART
    - Set all of the HALDB partitions to NOPINIT
      - CHANGE.DB DBD(xxxxxxxx) NOPINIT

HALDB Partition Initialization

- **Prior to IMS V9 there is an alternative HALDB initialization procedure**
  - It is run once per partition and has three steps
    - Step 1
      - OSAM delete/define the partition
      - VSAM delete/define the Primary Index
      - VSAM delete/define the ILDS
    - Step 2
      - DBRC utility to set the partition to PINIT
    - Step 3
      - IMS Prereorganization Utility
        - > This will initialize only the PINIT partition

HALDB Partition Initialization

- **This HALDB initialization procedure was prone to error**
  - There were many different scenarios where the jobs failed or running two jobs simultaneously caused errors
    - Identifying the error was not easy
    - Recovering from the error was not easy
  - DFSUPNT0 was enhanced to allow parallel execution
    - PQ75309 (IMS V7)
    - PQ87516 (IMS V8)
    - PQ87517 (IMS V9)

HALDB Partition Initialization

- **After the HALDB partition has been initialized the Image Copy Needed flag is on**
  - The partition MAY have to be Image Copied in its initialized state
  - This is explained in the next foil

HALDB Database Initial Load

- **The manual states that HALDB's are loaded with a PROCOPT=L/LS PCB**
  - PROCOPT=L/LS is NOT required
  - If the partition has been initialized and Image Copied it can be loaded with either PROCOPT=L/LS or PROCOPT=A/I
    - If it has been initialized but not Image Copied PROCOPT=L/LS is required
      - The Image Copy Needed flag is on but is ignored
      - You must Image Copy after the PROCOPT=L/LS job
  - For large amounts of data PROCOPT=L/LS will be much more efficient
  - It is not necessary to load data into all partitions as long as they have been initialized and Image Copied

HALDB Database Initial Load

- **Secondary indexes for HALDB databases are created during the Initial Load of the target database**
  - Work files are not created
  - There is no Prefix Resolution
  - There is no HISAM Unload/Reload
  - Segments are inserted randomly
    - This can greatly affect Initial Load performance!!!

HALDB Database Initial Load

- **Secondary indexes for HALDB databases can be added after the initial load using Secondary Index Utilities**
  - Use the OPTIONS,BLDSNDX=NO statement in the DFSVSAMP member to not build the secondary index entries during the Initial Load job
    - PQ55840
    - – IMS V7
    - PQ56464 – IMS V8

HALDB Database Initial Load

- **APAR PQ73700 documents a permanent restriction for HALDB database initial load**
  - You can not use PARM=DBB for initial load of a HALDB database
    - You will get ABEND0C4 in DFSDDLE0
    - The block builder does not have enough information at ACBGEN time to build the blocks correctly
  - It will be documented that loading HALDB databases in Load Mode (PROCOPT=L/LS) will require PARM=DLI

HALDB Database Initial Load

- **IBM IMS High Availability Large Database (HALDB) Conversion and Maintenance Aid Program Product**
  - This product has a utility aid (Initial Load Assist) that assists applications with (PROCOPT=L/LS) loading HALDB databases

HALDB Database Migration

- **There is help in migrating exist databases to HALDB**
  - The DFSMAID0 utility which is shipped with IMS will scan existing databases and provide statistics and recommendations for HALDB partition boundaries
  - IBM IMS High Availability Large Database (HALDB) Conversion and Maintenance Aid Program Product
    - Provides the utilities and aids needed to analyze, model, and convert existing IMS database structures including DEDB's (for both Versions 7 and 8) to a HALDB format easily and efficiently with minimal manual intervention

HALDB Database Buffer Pools

- **HALDB database data sets are assigned to OSAM and VSAM buffer pools using the DFSVSMxx member just like Full Function databases**
  - If the Master database name is used all partitions are assigned to the same subpool
  - Individual partitions may be assigned to their own subpools

HALDB Partition Processing

- **When HALDB's were first designed there was no way to run a batch or BMP job against just one partition or a list of partitions**
  - This was available for DEDB's using the DFSCTL control cards
  - Customers asked for the same processing capability for HALDB

HALDB Partition Processing

- **The first attempt to fix this was via PQ57313 (UQ65876)**
  - PQ58600 (UQ67984) for IMS 8.1
  - It introduced the DFSHALDB DD card to input the parameters
  - You were able to restrict all calls for a DBPCB to a single HALDB partition
  - Unfortunately you had to specify the PCB as a relative PCB number

HALDB Partition Processing

- **The second attempt to fix this was via PQ65486 (UQ73331)**
  - PQ65489 (UQ73332) for IMS 8.1
  - This allowed the specification of a PCB label instead of the relative PCB number
  - You are still restricted to one HALDB partition
  - Customers would still like to be able to process a list of partitions

Self-Healing Pointers

- **Each partition has 2 numbers associated with it**
  - A unique partition ID (PID)
    - If a partition is deleted in DBRC the partition ID will never be reused
    - You may have to take this into account in your Partition Selection Exit
    - APAR PQ48421 (IMS 7.1) or PQ73858 (IMS 8.1) allows a partition to be “disabled” instead of being deleted
      - Additional maintenance is required
        - PQ78922 (IMS V7)
        - PQ81012 (IMS V8)
        - PQ83112 (IMS V9)
  - A reorganization number
    - Stored in the partition
    - Incremented for each reorganization

Self-Healing Pointers

- **Each segment has a unique ID when it is created**
  - Indirect List Entry Key (ILK)
  - 8 bytes
    - 4-byte RBA
    - 2-byte Partition ID
    - 2-byte Reorganization Number
  - Stored in the segment prefix
  - Never changes for the life of the segment
  - This ID is unique within the entire HALDB database

Self-Healing Pointers

- **Each partition has an Indirect List Data Set (ILDS) associated with it**
  - This is a VSAM KSDS
  - There are no records in the ILDS after Initial Load
    - There are no records until the first reorganization
    - There are never any records if the database has no secondary indexes and no logical relationships

Self-Healing Pointers

- **For each target segment (Secondary Index or Logical Relationship) a record called an Indirect List Entry (ILE) is stored in the ILDS during reorganization reload**
  - The key of the ILE in the ILDS is the ILK
  - This is done without regard for NULLVAL or Secondary Index Maintenance Routines
    - The insert is for the target segment
    - The source segment may not be the target and may be far away in the hierarchy
    - For some database types this may result in MANY unneeded ILDS records

Self-Healing Pointers

- **Secondary Indexes and logical relationships use an Extended Pointer Set (EPS) for finding the target segment**
  - The EPS is 28 bytes
    - Partition Number of the target segment
    - Reorganization Number of the target segment
    - ILK of the target segment
    - RBA Pointer to the target segment
    - RBA pointer to paired logical child
    - Database record lock ID of the target segment
  - This pointer is NOT updated during reorganizations

Self-Healing Pointers

- **When accessing a secondary index target segment (or logical relationship) if the reorganization numbers in the pointer segment and the target database partition are the same then use the RBA pointer to access the segment directly**
  - If the reorganization numbers are not the same use the ILK to find the ILE in the ILDS
    - This has the current RBA pointer
    - The reorganization number and RBA are also updated in the EPS in the buffer
    - If the PCB has update intent and the database access is update or exclusive
      - Also set the “Altered Buffer” flag
    - Self-healing pointers
    - Some customers prefer a utility to mass heal all of the pointers right after reorganization

Self-Healing Pointers

- **If a database has no Secondary Indexes and no Logical Relationships then it NEVER needs to use the ILDS – but IMS still requires it**
  - With APAR PQ61206 (UQ75705) (IMS V7) IMS will not allocate the ILDS in the DLISAS region if it is not needed
    - IMS V8 is PQ72776 (UQ78445)
    - This could result in the savings of 100's of allocated data sets
    - It still needs to be defined to VSAM
    - It will still be allocated by batch jobs
    - Customers have requested that it be totally eliminated if it is not needed

Self-Healing Pointers

- **When using the HD reload utility during database migration (MIGRATE=YES or MIGRATX=YES during HD Unload) be sure to specify the ILDSMULTI keyword**
  - IMS V7 PQ36991
  - IMS V8 PQ54227
  - This will write the ILDS records to a dataspace during reload
    - They are then sorted and inserted using VSAM load mode

Self-Healing Pointers

- **The HD reload also has a NOILDS statement**
  - IMS V7 PQ36991
  - IMS V8 PQ54227
  - This can be used by both migration unload/reload and normal unload/reload
  - No ILDS records are created or updated during HD reload
  - The ILDS must be rebuilt using the DFSPREC0 utility after the HD reload and before the database is used
    - This will scan the source data base partition

Increased Database Size with HALDB

- **Database segment prefixes are larger for HALDB databases**
  - All database segments add the 8-byte ILK
    - 4-byte RBA
    - 2-byte Partition ID
    - 2-byte Reorganization Number
  - All dependent segments will have a 4-byte Physical Parent Pointer

Increased Database Size with HALDB

- **A PSINDEX segment is (much) larger than the same INDEX segment for a non-HALDB secondary index**
  - Contains the 28-byte Extended Pointer Set instead of the 4-byte RBA
  - Contains the target root key
  - The /SX field is 8 bytes instead of 4 bytes
  - If you convert to HALDB from Full Function and you have programs which read the Secondary Index as a standalone database they may have to be changed

Increased Database Size with HALDB

- **A Logical Relationship source segment is (much) larger than the same segment for a non-HALDB database**
  - Contains the 28-byte Extended Pointer Set instead of the 4-byte RBA
  - Always contains the logical parent concatenated key (LPCK)
    - This is optional in non-HALDB databases

HALDB Image Copy

- **Image Copy for HALDB is supported by all current IMS Image Copy products**
  - Image Copy is by partition
    - DD cards must be included in the JCL
      - They are not included in IMS V9 because of HALDB Online Reorg
  - GENJCL.IC DBD(masterdb) generates JCL for all data set groups for all HALDB partitions
  - GENJCL.IC DBD(partname) generates JCL for all data set groups for one HALDB partition
  - GENJCL.IC DBD(partname) DDN(dsname) generates JCL for one data set group for one HALDB partition

HALDB Recovery

- **Recovery for HALDB is supported by all current IMS Recovery products**
  - Recovery is by partition
    - DD cards must be included in the JCL
      - This changed in IMS V9
  - DBRC GENJCL support is the same as for Image Copy
  - PHIDAM Primary Indexes and the ILDS are not recovered (or Image Copied)
    - They are rebuilt using the DFSPREC0 utility
      - They do not have to be deleted and redefined
      - Some doc says the ILDS must be delete/defined but that is incorrect
    - DBRC support for this requires GENJCL.USER

HALDB Reorganization

- **Reorganization for HALDB is supported by some current IMS Full Function Reorganization products**
  - Standard IMS Unload/Reload
  - IBM IMS HP Unload and HP Load
  - BMC
  - Neon
  - CA

HALDB Reorganization

- **Reorganization is by partition**
  - DD cards are not included in the JCL
  - The PHIDAM Primary Index is rebuilt
  - The ILDS is updated
  - Secondary indexes and logically related databases DO NOT have to be reorganized
    - No work data set
    - No Prefix Resolution
    - No HISAM unload/reload
    - No Prefix Update
    - The Self-healing pointers take care of this

HALDB Reorganization

- **HALDB data sets do not have to be deleted and redefined for reorganization**
  - The VSAM REUSE attribute is required for HALDB VSAM databases and is honored by HD reload
    - Delete/define is required for non-HALDB databases
  - OSAM allows reuse for both HALDB and non-HALDB databases

HALDB Reorganization

- **HALDB Secondary Index databases (PSINDEX) are no longer rebuilt during database reorganization but they can become disorganized and need to be reorganized**
  - HALDB Secondary Index databases (PSINDEX) are reorganized using the HD unload/reload utilities
    - DFSURGU0 – HD Unload
    - DFSURGL0 – HD Reload
  - The HISAM unload/reload utilities do not support PSINDEX
    - DFSURUL0 – HISAM Unload
    - DFSURRL0 – HISAM Reload

HALDB Reorganization

- **The IBM IMS High Availability Large Database (HALDB) Conversion and Maintenance Aid Program Product can aid in database reorganizations**
  - The Maintenance Aid in this product supports
    - Splitting partitions
    - DBRC functions for HALDB
    - Creation of backup and recovery JCL
    - Generation of reorganization utilities

HALDB Reorganization

- **Online reorganization for HALDB is available with IMS V9**
  - This is true online reorganization with no database outage
  - Duplicate sets or database data sets are required
    - But reorganization can be done one partition at a time
  - There are other sessions dealing with this topic

HALDB Pointer Checker

- **Pointer Checker/Free Space Analysis is supported by some products**
  - IBM HP Pointer Checker
  - BMC Pointer Checker Plus
  - Neon iCheck

HALDB Maintenance

- **Keeping up to date with maintenance is very important for HALDB**
  - Use the general IMS maintenance recommendation
  - Use Enhanced HOLDDATA and SMP/E REPORT ERRSYSMODS to identify missing HIPER and PE maintenance.
  - Read APAR USERS AFFECTED text to determine if the fix applies to either full function DB or HALDB users
  - Apply and test appropriate maintenance
  - Contact the IMS Support center for questions about specific APARs/PTFs if unclear

Conclusion

- **HALDB's offer support for partitioning Full Function databases for size and availability**
  - As with anything new – there is a learning curve
  - Many customers are in production with HALDB
  - The *Complete IMS HALDB Guide* redbook (SG24-6945) is an indispensable source of information