E54

Migrating from a Full Function Database to IMS V7 HALDB

Rick Long

Silicon Valley Lab

Anaheim, California October 23 - 27, 2000

RETURN TO INDEX

Introduction

- Rick Long
  - Silicon Valley Lab (remotely)
    - IMS Data Propagator
    - Classic Connect
  - Formerly
    - ITSO IMS Specialist
    - IMS Systems Programmer
    - IMS DBA
    - Application Programmer
  - ricklong@us.ibm.com

Prerequisite

- E53 - IMS V7 High Availability/Large Database Support
- Basic understanding of HALDB structures
- Recommended reading
  - REDBOOK SG24-5751
    - Available at WWW.REDBOOKS.IBM.COM

Objectives

- Describe a DBA's view of HALDBs
- Describe how to define HALDBs
- Describe the migration process from full function
- Describe some programming considerations
- Describe some operation characteristics of HALDBs

Describe HALDBs

- HALDBs are full function databases
  - Fully support logical relationships
    - Bi-directional relationships always implemented as physically paired
    - All logical views supported
  - Support secondary indexes
    - Processing via secondary indexes
    - Processing a partitioned secondary index as a database
  - Technically correct terms are:
    - HALDB full function database
    - Non-HALDB full function database

Describe HALDBs

- Partitioned database
- Partitions have utility independence
  - The ability to run utilities against multiple partitions concurrently
    - Image Copy
    - Recovery
    - Reorganization utilities
- Partitions allocated/authorized independently
  - Image copying a partition while others are allocated

Describe HALDBs

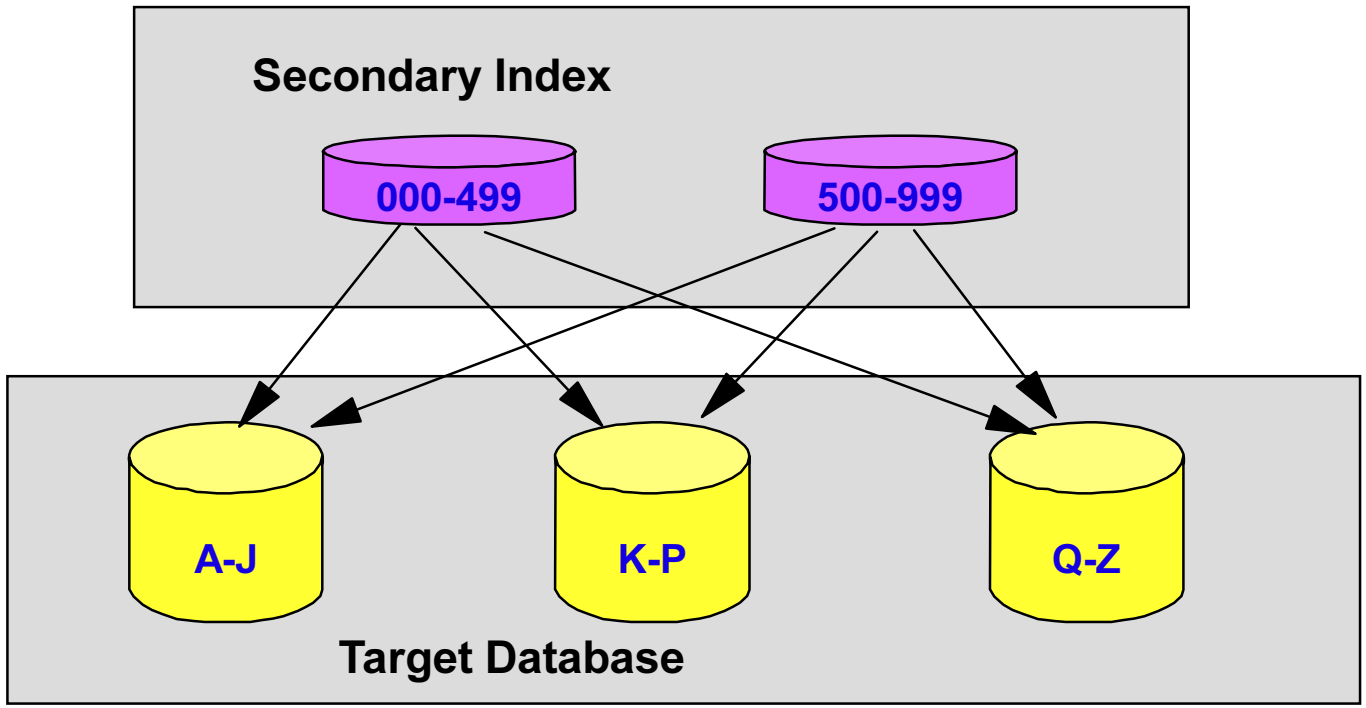
- Partitions do not have program independence
  - IMS does not provide partition information to application programs
    - No feedback on which partition a segment was retrieved from/inserted to
    - No status code issued when crossing partition boundaries
    - IMS authorizes/allocates partition on first access
    - If partition not available a status code is returned

Describe HALDBs

- User could provide program partition independence
  - If the program knows the partition boundaries
  - IMS does not authorize partition until first access
  - Think carefully about recovery issues

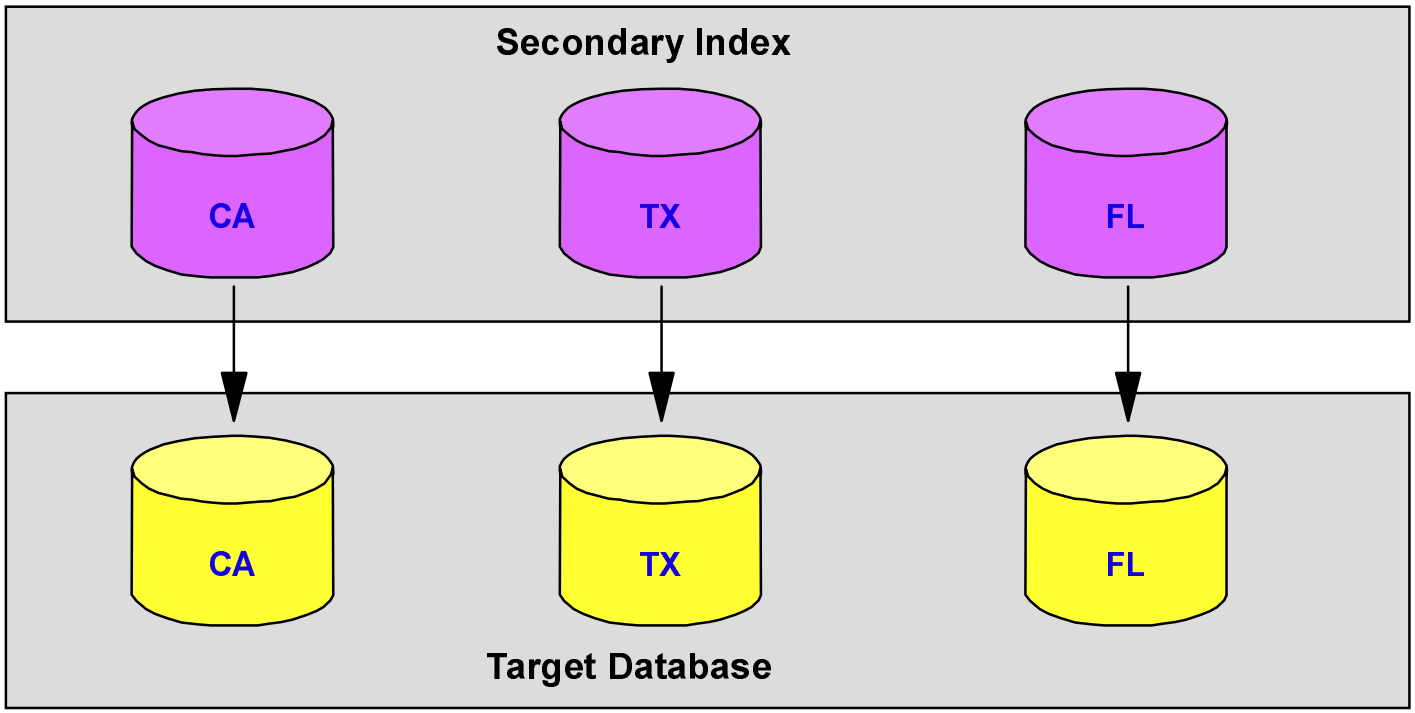
Describe HALDBs

- Secondary Indexes may be partitioned differently
  - Backup/recovery issues
    - Can't take off-line without affecting all partitions
    - Can't use point-in-time recovery without affecting all partitions
  - Application program using secondary index might access all target database partitions



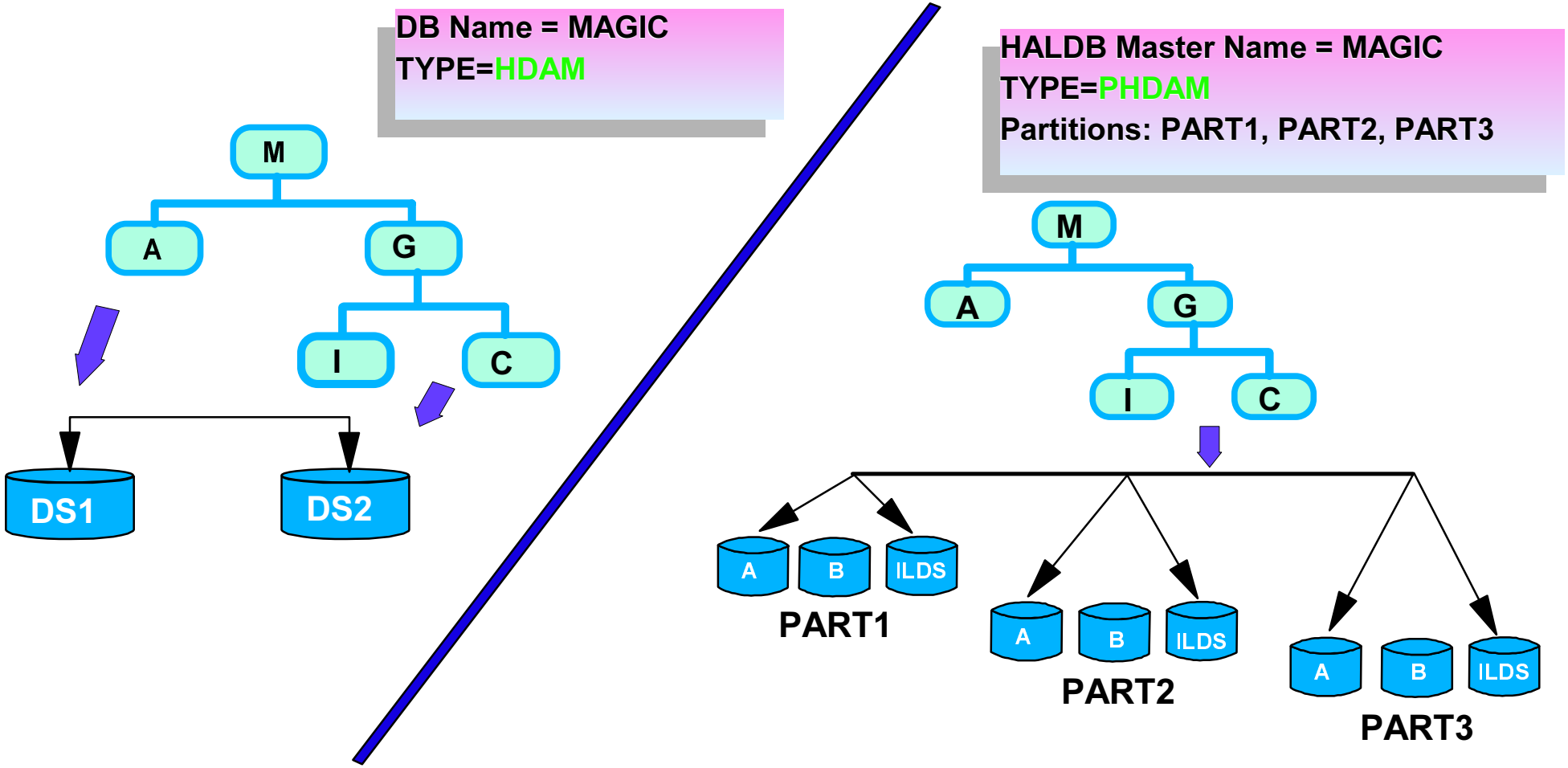
Describe HALDB

- Secondary index may partition like target database
  - Key range values
  - User written partition selection exit could create partition independence
  - Watch out for multiple secondary indexes



DBD Changes

- Partitioning by database record not segment type



DBD Changes

- Remove data set information
- DATASET attributes stored in the RECON
  - Using the HALDB Partition Definition Utility (PDU)
  - Using batch commands (when they become available)
- DSGROUP information can be maintained if still required
  - If data set groups were used to limit physical size of a data set, might not be required now
  - Review requirement for data set groups

DBD Changes

BEFORE

```
DBD NAME=MAGIC,ACCESS=(HIDAM,..)
DATASET DDNAME=1..
SEGMENT NAME=M,...
SEGMENT NAME=A,...
DATASET DDNAME=2
SEGMENT NAME=G,...
SEGMENT NAME=I,...
SEGMENT NAME=C,...
```

AFTER

```
DBD NAME=MAGIC,ACCESS=(PHIDAM..)
~~DATASET ...~~
SEGMENT NAME=M,...
SEGMENT NAME=A,..
~~DATASET ...~~
SEGMENT NAME=G,...DSGROUP=B
SEGMENT NAME=I,...DSGROUP=B
SEGMENT NAME=C,...DSGROUP=B
```

DBD Changes/PDU

- **Partitioning**
  1. **Determine number of partitions required**
     - **Size of data sets (4GB limit)**
     - **Partition independence**
  2. **Determine type of partitioning**
     - **Key range**
     - **Partition selection exit**

DBD Changes/PDU

- Partitioning (continued)
  3. Partition naming standards
     - Meaningful name?
     - Repartitioning later
       - Growth of data sets
       - New data
     - Automatic generation of partitions
       - Lots of partitions?

  4. Database data set name prefix
     - High level qualifiers
     - Test system Vs prod systems

Partition Definition Utility (PDU)

- ISPF front-end to DBRC for HALDB definitions
  - Only way to define HALDBs
  - Batch DBRC command support will be available after GA through service process
- Updates RECON directly
  - RACF access required
  - Timing of changes
  - Some nice functions available
    - List partitions in sequence
    - Copy partition information
    - Change multiple partition information

RECON Information

- HALDB master database information
  - DB record
    - DB=master name
    - TYPE=HALDB
- HALDB partition information
  - DB record
    - DB=partition name
    - TYPE=PART MASTER DB=haldbname
  - DBDS record
    - DBD=partition name
    - DDN=partition + suffix

Migration Options

1. Use current DBD name as the HALDB master name

2. Define new HALDB master name

3. Define new HALDB master/ Use old DBD name as logical

1. Current DBD name = HALDB master

- Constant DBD name
- Lose RECON history when DB is deleted and defined to DBRC
- Backout more complex

2. HALDB master = newname

- Can predefine HALDB
  - Setup can be done early
- Don't lose RECON history
  - Fallback less complex
- New name modifications
  - Modify all PSB's to use new name
  - Modify control cards
  - Modify IMS automation

3. HALDB master=newname oldname=logical DBD

- Don't lose RECON history
  - fallback less complex
- Can predefine HALDBs
- PSBs are not modified

Migration Process

- Example uses current DBD name as HALDB master name
- HIDAM database
- No secondary indexes or logical relationships

Migration Steps

1. **Backup current environment**
   - Image copies of databases
   - RECON (BACKUP.RECON after ICs)
   - DBDLIB/ACBLIB/MDA members

2. **Unload the databases**
   - HD Reorganization Unload Utility
   - Current DBDLIB
   - MIGRATE=YES (control card)

Migration Steps

3. **Delete old objects**
   - RECON DB records
   - MDA members
   - Database data sets (?)

4. **Modify/generate DBD**
   - Could be done early in different DBDLIB
   - Delete dataset groups if not required
   - Remove HIDAM index DBD (not needed any more)

Migration Steps

5. Define HALDB
   - Use ISPF HALDB Partition Definition Utility
   - Number of partitions
   - Key ranges
     - Make sure last partition has highest possible key
     - Watch out for lower case key

6. Define data sets
   - VSAM - DEFINE CLUSTER
   - OSAM - ALLOCATE/DEFINE NON-VSAM
   - 4Gb limit on both OSAM and VSAM
   - ILDS data sets
   - Primary index with increase size (indirect pointers)

Migration Steps

7. **Initialize HALDB database**
   - Prereorganization Utility (DFSURPR0)
   - New DBDLIB member
   - Use DBIL=mastername
   - Remove database DD statements
   - PARTITION INIT NEEDED is set to NO

8. **Load HALDB database**
   - HD Reorganization Reload Utility (DFSURGL0)
   - Remove database DD statements
   - New DBDLIB member

Migration Steps

9. **Image copy database**
   - HALDB partitions only (not HIDAM index)
   - Remove database DD statements
   - Remove HIDAM index references

10. **Update IMS system definitions**
   - Perform ACBGEN
   - Remove DATABASE macro for primary index
   - Remove MDA members

Fallback

1. Restore IMS system definitions
   - ACBLIB
   - MODBLKS
   - MDA members
   - DBDLIB
2. Update RECON
   - Delete HALDB
   - Register non-HALDB
3. Recover databases
   - Use backup RECON to generate JCL
   - Use real RECON for execution JCL
4. Image copy database

Migrating Related DBs

- Same basic steps but consider
  - Unload all related databases before reloading any
    - Logical
    - Secondary indexes
      - Use HD Reorganization Unload Utility
      - APAR available for rebuilding during reload
  - Unload reads logical databases
  - Define data sets
    - Secondary index segment lengths changes
    - Secondary index record size changes
    - Space allocation increase
      - Physically paired segments
      - Indirect pointers

Programming with HALDB

- Segment size increase
  - /SX field is 8 bytes instead of 4 bytes
  - Concatenated key is not present as a result of PTR=SYMB pointers
- Status calls only reflect HALDB master status
  - Programs will get "BA" or U3303 if some partitions are not available
  - IMS authorizes partition at first access to it

Programming with HALDB

- Application program loading of logical child segments is not allowed
  - PROCOPT=L
  - Must be inserted (PROCOPT=A/I)
- Processing options
  - A program with an update PROCOPT may lock segments as a result of resetting indirect pointers which have been altered by a reorganization
    - First program with UPDATE will update altered indirect pointers
    - If concerned about locking/performance
      - Modify read programs to remove update intent
      - Run pointer update BMP/DLI after REORG

Operating with HALDB

- Starting and Stopping HALDB
  - Master must be started to access any partition
  - Partitions can be started or stopped explicitly
  - Once a partition is DBR'd explicitly, it must be STARTED explicitly
    - A /START DB mastername will not start an explicitly DBR'd partition
    - A /START DB ALL will not start an explicitly DBR'd partition

Operating with HALDB

- Displaying HALDB information
  - /DISPLAY DB mastername
  - /DISPLAY DB partitionname
  - /DISPLAY DB ALL

/DISPLAY DB mastername

Stopped HALDB master

```
/DISPLAY DB DFXCSTP
DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED ACC  CONDITIONS
DFXCSTP   PHDAM                            UP   STOPPED
*00228/193925*
```

Started HALDB master

```
/DISPLAY DB DFXCSTP
    DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED ACC  CONDITIONS
   DFXCSTP   PHIDAM                            UP
   CUST1     PART                              UP   NOTOPEN
   CUST2     PART                              UP   NOTOPEN
   CUST3     PART                              UP   NOTOPEN
   CUST4     PART                              UP   NOTOPEN
   CUST5     PART                              UP   NOTOPEN
   CUST6     PART                              UP   NOTOPEN
   *00229/170020*
```

/DISPLAY DB ALL

DFXBALY	PSINDEX	UP	STOPPED
DFXCLAP	DL/I	UP	NOTOPEN
DFXCLAX	DL/I	UP	NOTOPEN
DFXCLAY	DL/I	UP	NOTOPEN
DFXCLY2	DL/I	UP	NOTOPEN
DFXCSTP	PHIDAM	UP	STOPPED
DFXPHNP	DL/I	UP	NOTOPEN
DFXPRDP	PHDAM	UP	STOPPED
DFXSTUP	DL/I	UP	NOTOPEN
DFXSTUY	DL/I	UP	NOTOPEN
DFXZIPY	PSINDEX	UP	STOPPED
DI21PART	DL/I	UP	NOTOPEN
IVPDB1	DL/I	UP	NOTOPEN

RETURN TO INDEX