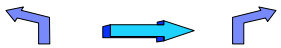
IBM Software Group

IMS12
An Introduction to IMS High Availability Large Databases (HALDB)

Cornelia Hallmen

HALDB (High Availability Large Database)

- Large Database

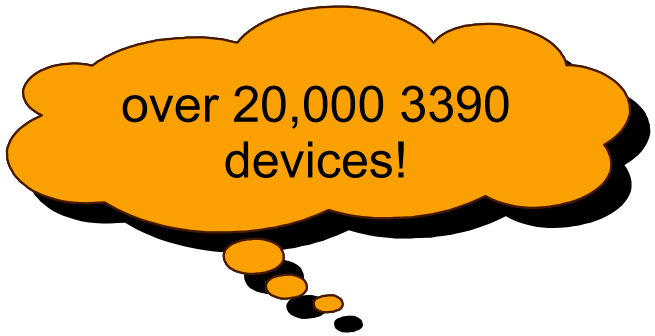


  - Databases are partitioned
    - Up to 1001 partitions per database
    - Partitions have up to 10 data set groups

How Big? - Doing the Math

4 Gig	(dataset size)
x 1001	(partitions)
x 10	(datasets per partition)

~ 40 Tera Bytes	



HALDB (High Availability Large Database)

- High Availability Database

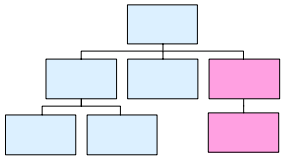
  - Partition independence
    - Allocation, authorization, reorganization, and recovery are by partition
  - Self healing pointers

    - Reorganization of a partition does not require changes to secondary indexes or logically related databases

Maintains and extends the performance and availability characteristics you expect from IMS !!

Highlights

- Database records are grouped into partitions
  - Hierarchic structure is maintained within a partition
  - A database consists of 1 or more partitions



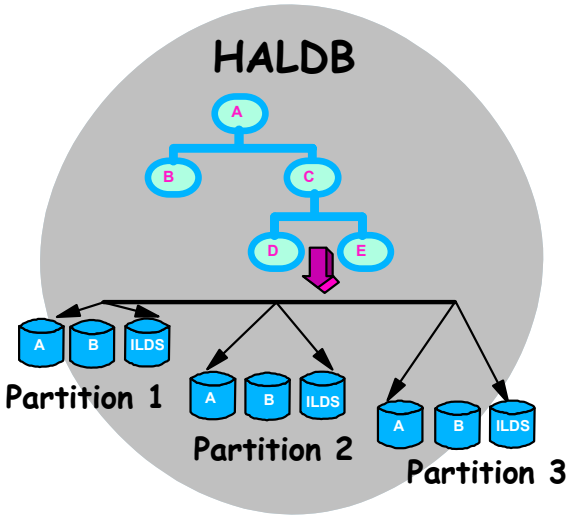
- New database types
  - **PHDAM** - partitioned HDAM
  - **PHIDAM** - partitioned HIDAM
    - Index is partitioned
  - **PSINDEX** - partitioned secondary index
- Partition selection
  - By key range or by user exit routine

Highlights

- OSAM and VSAM (ESDS and KSDS) are supported
- Logical relationships and secondary indexes are supported
  - Secondary indexes may be partitioned
- DBRC is required
  - Databases must be registered
  - Dynamic allocation from DBRC information, not DFSMDA
- Minimal (or no) application changes required
  - Cannot initially load logical child segments
    - New status code for load programs
  - 'Data unavailable' conditions apply to partitions
    - Database may be available, but partition unavailable

HALDB Benefits

- Increased database **capacity and availability**
  - Partitions independence
    - Allocation, authorization, reorganization, and recovery are by partition
    - Batch window is shortened with concurrent processing
  - Partitions, not databases, are removed from system
    - Shortened reorganization process
  - Self healing pointers
    - Reorganization of a partition does not require changes to secondary indexes or logically related databases

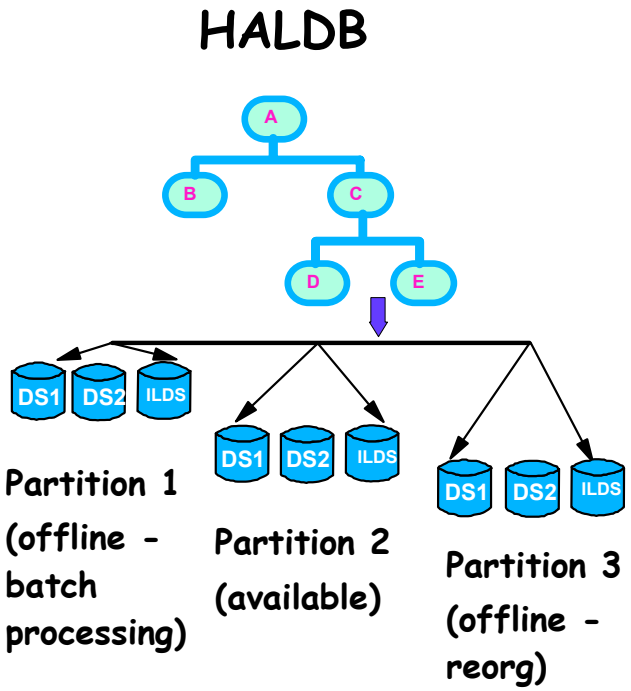


HALDB Benefits

- Improved **manageability**
  - As its size grows, a database becomes difficult to manage
  - Smaller sections of the database are easier to manage
- Enhanced **usability**
  - HALDB removes the steps involved in running the prefix resolution and prefix update utilities
  - ISPF utility for partition definitions

Partition Independence

- Commands
  - Allowed on both databases and partitions
- Availability
  - Partitions are allocated and authorized independently
- Scheduling
  - Based on database availability
    - PCBs and INQY calls report database availability
    - Partition may be unavailable with available database
- Database Utilities
  - Allowed on individual partitions or sets of them
  - Concurrent processing of multiple partitions allowed



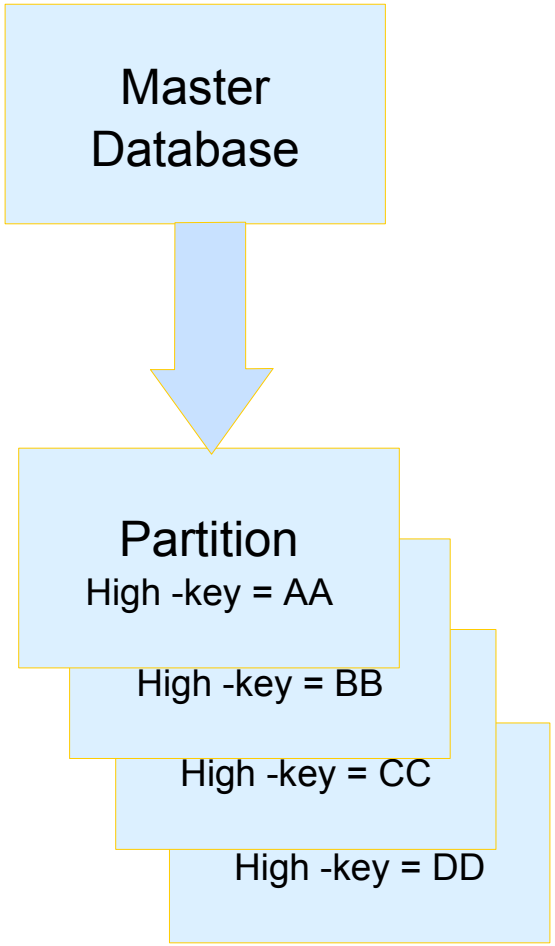
Definition Process

- DBDGEN
  - Used to define database
    - Hierarchic structure, data set groups, pointer options, logical relationships, secondary indexes,...
- HALDB Partition Definition Utility
  - Used to define partitions in database
    - Partition selection, boundaries, space characteristics, randomizers,...
  - ISPF based
  - Stores information in the RECONs
    - Definitions may be done with DBRC commands instead of this utility

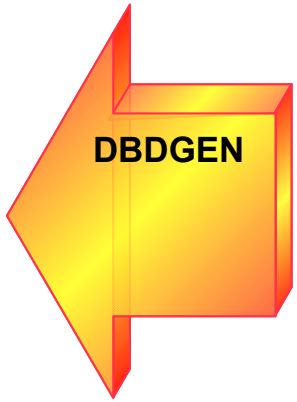
Definition Process

- System Definition
  - Specifies the database to the online system
- DFSVSMxx and DFSVSAMP DD
  - Assigns data sets to buffer pools

Definition tasks



DBD NAME=
~~DATASET ...~~
SEGMENT NAME=
FIELD NAME=



DATASET
attributes are set by PDU,
stored in RECON

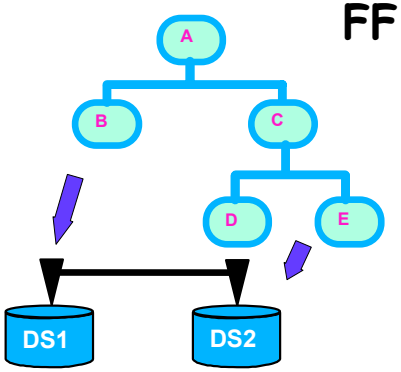
HALDB - Example

- **DL/I With and Without HALDB**

DB Name = MASTER
TYPE=HDAM

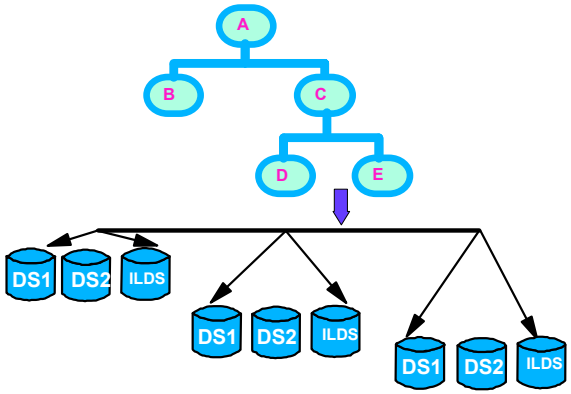
Data Set Groups: DS1
DS2

FF DB



Master DB Name: MASTER
TYPE=PHDAM
Partitions: PART1, PART2, PART3
Data Set Groups: DS1 and DS2 per partition

HALDB



Database Structures

- PHIDAM prime indexes are not separately defined
  - Defined as part of the PHIDAM database
    - Applies to DBDGEN and system definition
- Parent pointers
  - All dependent PHIDAM / PHDAM segments have physical parent pointers
- Symbolic pointers are not used
  - All pointers are direct

Database Structures

- Logical relationships
  - Virtual pairing is not allowed
    - Limited to unidirectional or physically paired
  - Logical child segments cannot be initially loaded
    - Must be added by update
- Secondary indexes must have unique keys
  - /SX or /CK may be used to create uniqueness
    - /SX is increased from 4 to 8 bytes (ILK)

HALDB Database Structure

- HALDBs have a new structure in order to support partitioning and inter-record pointing
  - Each partition in a database has a unique partition ID (PID)
  - A reorganization number is maintained in each partition
    - Incremented by each reorganization reload
  - Each segment in PHDAM or PHIDAM database is assigned a unique token when created
    - Indirect List Entry Key (ILK)
    - 8 bytes stored in segment prefix
    - RBA + partition ID + reorg# at segment creation
    - Key is **permanent**, even if the segment has been moved at reorg!

Reorganizations

- Reorganizations are simplified for logical relationships and secondary indexes
  - Work files are **not used**
  - Prefix Resolution, Scan, and Prefix Update are **not used** to update logical relationship pointers
  - HISAM Unload, HISAM Reload, or tools are **not used** to update secondary index pointers
- A new pointer scheme **is used**!
  - Applies only to logical relationships and secondary indexes
  - HALDB use both direct and indirect pointers

INDIRECT POINTERS

Indirect Pointers

- Indirect pointers are implemented to eliminate the need to update pointers in other database records when a partition is reorganized
  - When a partition is reorganized, it's segment locations can change -- potentially invalidating all inter-record pointers to segments in that partition
  - Segments which can point from one record to another are:
    - Physically paired logical children
    - Logical parents of unidirectional logical relationships
    - Targets of secondary indexes
- An ILDS is KSDS associated with a Partition: one per partition is required!
  - Is used as a repository for the indirect pointers in a partition
  - An entry in an ILDS is called an Indirect List Entry (ILE)
  - It is created for every segment that is involved in inter-record pointing during reload

ILDS - indirect list entry data set

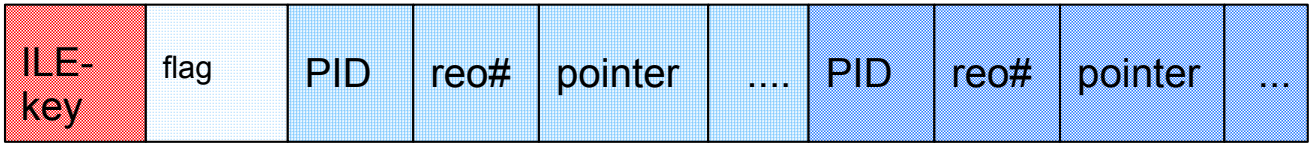
- **ILEs** do **only exist for segments which are targets** of secondary indexes or of logical relationships

- Each ILE consists of 50 bytes and is created or updated **only by reorganization**
- **ILEs** are composed of
  - **ILE-key** (ILK + Segmentcode => 9 bytes) and
  - **ILE data**

    Flag, 20 bytes with Partition ID, Reorg #,accurate pointer to each logical parent or secondary index target, accurate pointer to each physically paired logical child as well as database record lock ID for segment, version#,.... (used at uneven reorg #)

    20 bytes with corresponding information used at even reorg#

ILE



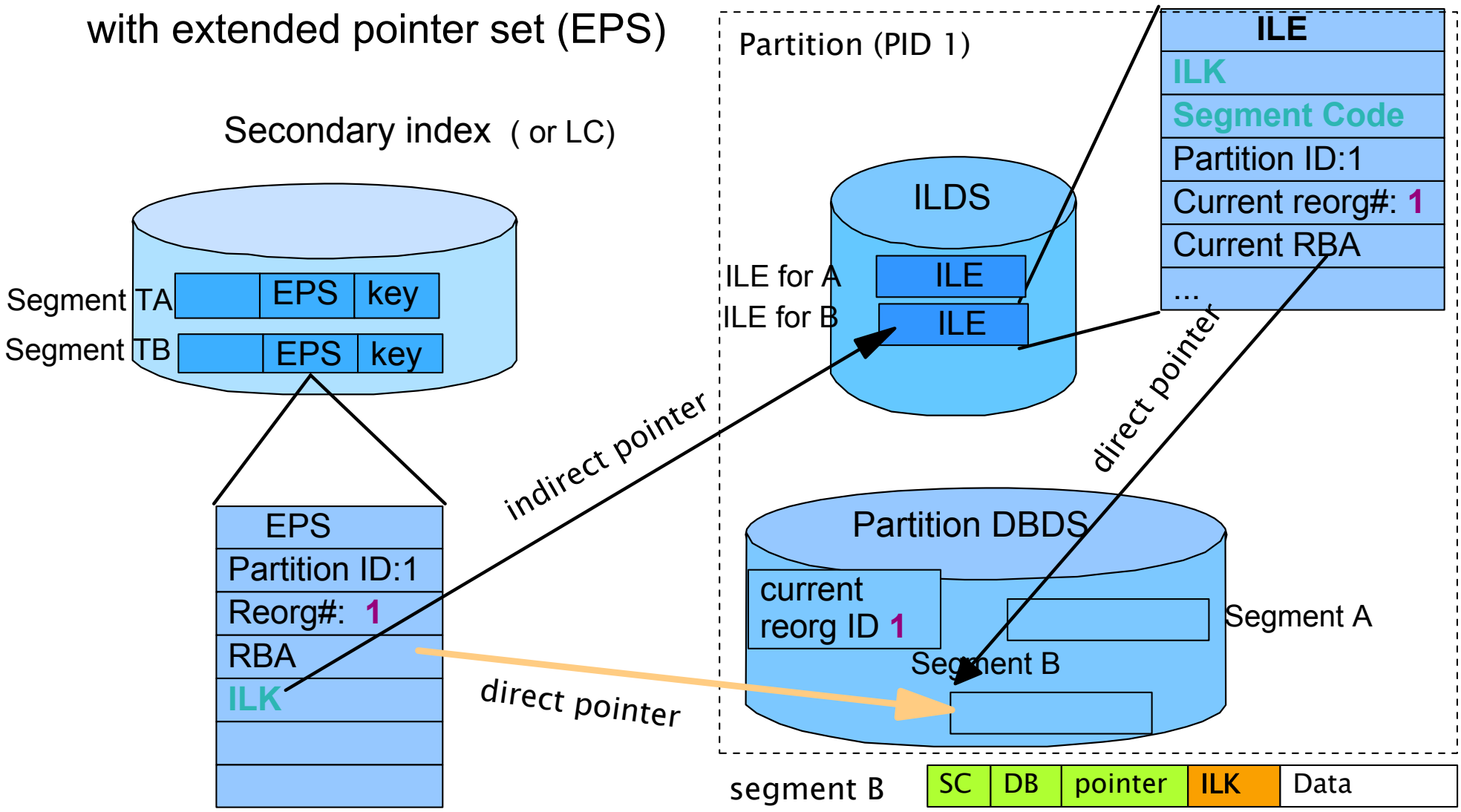
left half **right half of data**

9 bytes 1 byte 20 Bytes 20 Bytes

Extended Pointer Set

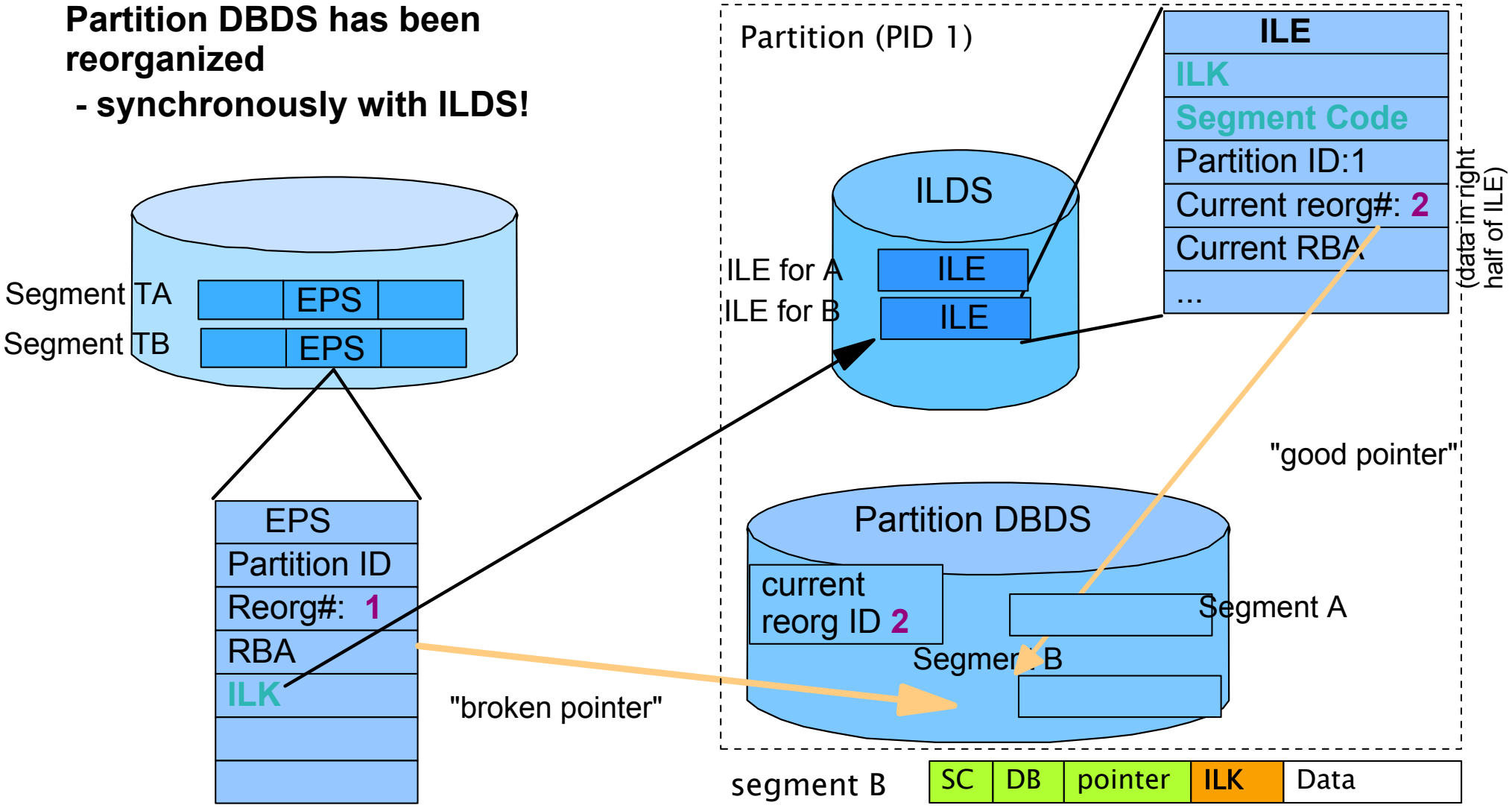
- Extended Pointer Set (EPS) is used for logical relationships and secondary indexes
  - Replaces direct or symbolic pointers used in Non-HALDB databases
  - Key of root is used to determine partition
  - EPS contains direct pointer, reorganization number, target partition ID, and ILK
    - If reorg number is current, direct pointer is used
    - If reorg number is not current, ILK is used to find ILE in ILDS
    - ILE contains pointer to segment
  - EPS is not updated by reorganizations!
  - Direct pointer and reorg number in EPS are updated when ILE is used
- Self healing pointers!

Selfhealing Pointers



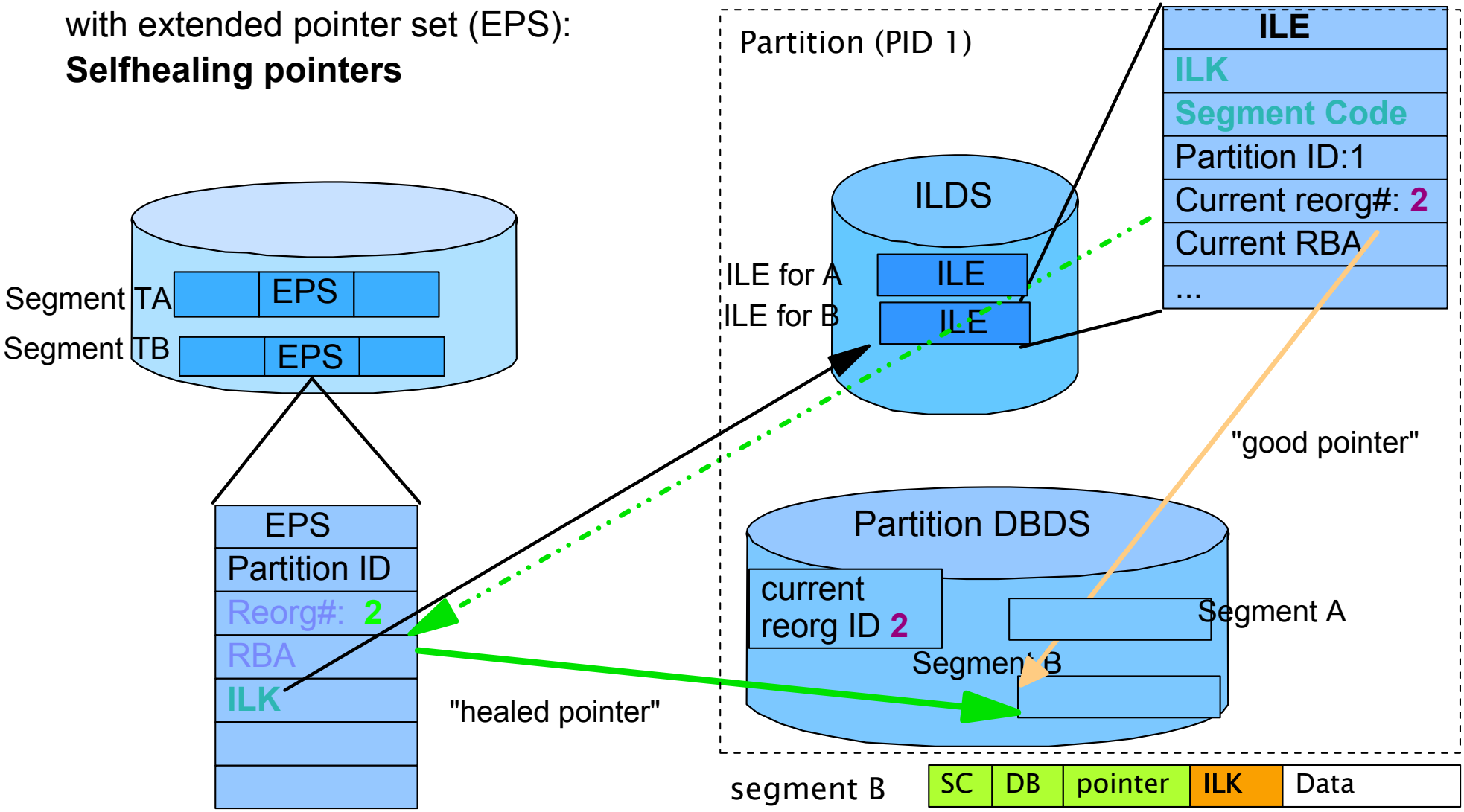
Selfhealing Pointers

Partition DBDS has been reorganized
- synchronously with ILDS!



Selfhealing Pointers

with extended pointer set (EPS):
Selfhealing pointers



Extended Pointer Set (EPS) Adjustments

- When out of date pointer is found it is corrected if:
  - Access intent is update or exclusive
  - PROCOPT is update
- Locking considerations
  - Read programs with update PROCOPTs may hold many locks
    - If block level data sharing is used, block locks are held until sync point

Reorganization Frequencies

- Reorganization frequencies may be changed
- Increased free space may reduce reorganization frequencies
  - HALDB may allow users to increase free space
  - Increased free space may reduce need to reorganize
- Reorganization frequencies may be increased
  - Reorg windows are reduced due to elimination of utility steps and parallel processing
  - Selected partitions may be reorganized independently

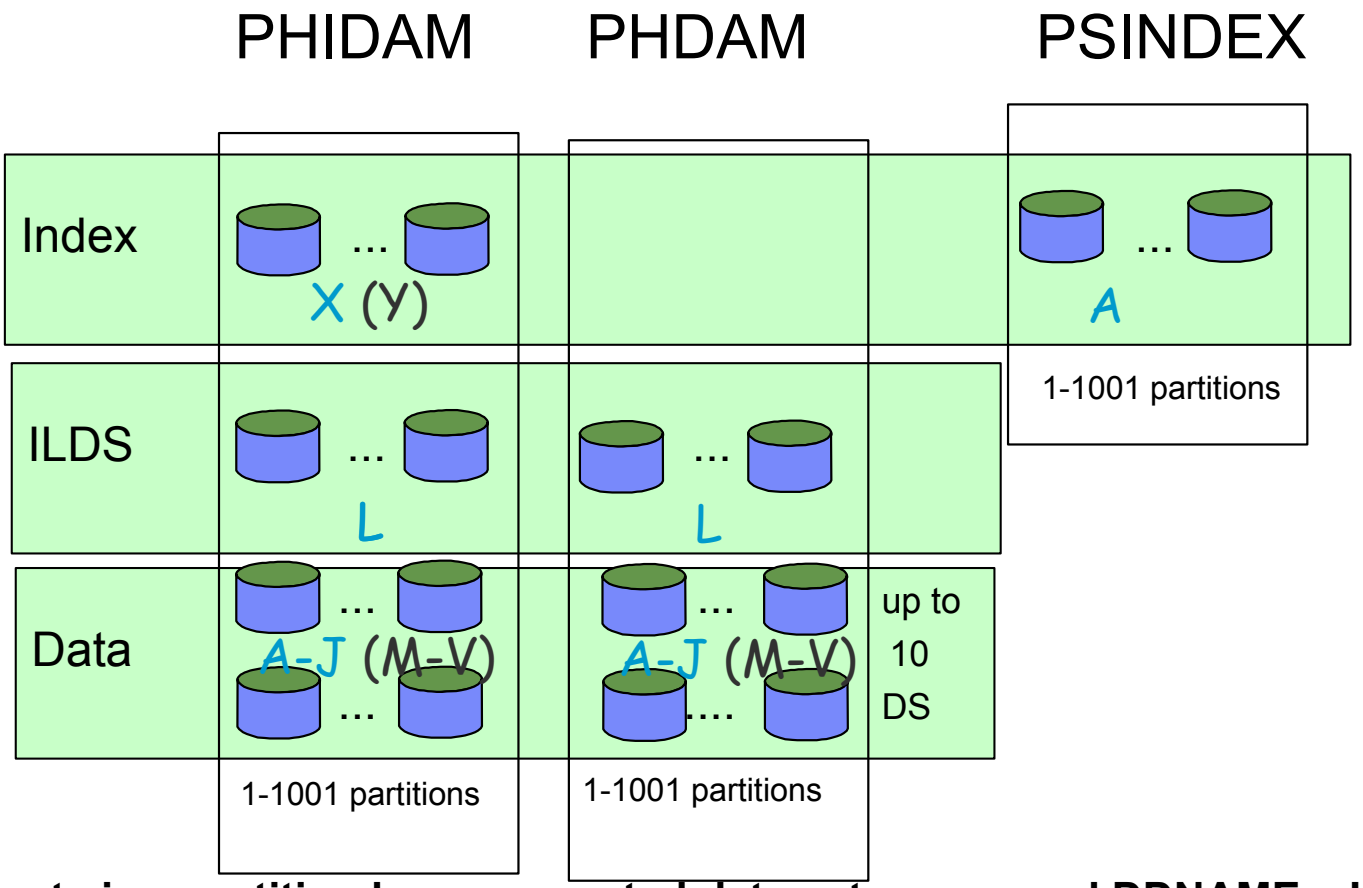
HALDB Database Data Sets

- Each HALDB database has up to 1001 partitions
- PHIDAM has index, ILDS, and up to 10 data set groups per partition
  - 3 to 12 data sets per partition
  - 3 to 12,012 data sets per database
- PHDAM has ILDS and up to 10 data set groups per partition
  - 2 to 11 data sets per partition
  - 2 to 11,011 data sets per database
- PSINDEX has no ILDS or data set groups
  - 1 data set per partition
  - 1 to 1001 data sets per secondary index

Database Data Sets

- Data set names
  - Begin with data set name prefix for the partition
    - Up to 37 characters
    - Assigned by user in HALDB Partition Definition Utility
  - Letter and Partition ID are used as suffix
    - X for PHIDAM index
    - L for ILDS
    - A for PSINDEX
    - A through J for data

HALDB database data sets



The data sets in a partition have generated data set names and DDNAMEs. Letters are used to distinguish them.

- **X - PHIDAM index**
- **L - ILDS**
- **A - PSINDEX**
- **A through J - Data data sets**

Partition DDNAMEs and Data Set Names

Example: PHIDAM with 10 data set groups, FRANCE partition

Partition_name of **FRANCE** (assigned by user in HALDB Partition Definition Utility)

DSN_prefix of **IMP0.DB.INV23.FRANCE** (assigned by user in HALDB Partition Definition Uti

PartitionID of **00004** (assigned by IMS in HALDB Partition Definition Utility)

Data set	DDNAME	Data Set Name
Data set group 1	FRANCE**A**	IMP0.DB.INV23.FRANCE.**A00004**
Data set group 2	FRANCE**B**	IMP0.DB.INV23.FRANCE.**B00004**
Data set group 3	FRANCE**C**	IMP0.DB.INV23.FRANCE.**C00004**
...	...	...
Data set group 10	FRANCE**J**	IMP0.DB.INV23.FRANCE.**J00004**
ILDS	FRANCE**L**	IMP0.DB.INV23.FRANCE.**L00004**
PHIDAM Index	FRANCE**X**	IMP0.DB.INV23.FRANCE.**X00004**

Partition DDNAMEs and Data Set Names

Example: PHIDAM with 10 data set groups, CANADA partition

Partition_name of CANADA

DSN_prefix of IMP0.DB.INV23.CANADA

PartitionID of 00011

Data set	DDNAME	Data Set Name
Data set group 1	CANADAA	IMP0.DB.INV23.CANADA.A00011
Data set group 2	CANADAB	IMP0.DB.INV23.CANADA.B00011
Data set group 3	CANADAC	IMP0.DB.INV23.CANADA.C00011
...	...	...
Data set group 10	CANADAJ	IMP0.DB.INV23.CANADA.J00011
ILDS	CANADAL	IMP0.DB.INV23.CANADA.L00011
PHIDAM Index	CANADAX	IMP0.DB.INV23.CANADA.X00011

Partition Selection

- **Partition selection is based on either:**
  - Key range

    or

  - Partition Selection Exit routine
- **Partition selection determines:**
  - Where root segments are placed
  - Order in which partitions are processed

Partition Selection

- Restricting a PCB to a single partition
  - Batch or BMP
- New DD name, DFSHALDB

```
// DFSHALDB DD *
HALDB PCB=(nnn|dddddddd,ppppppppp)
```

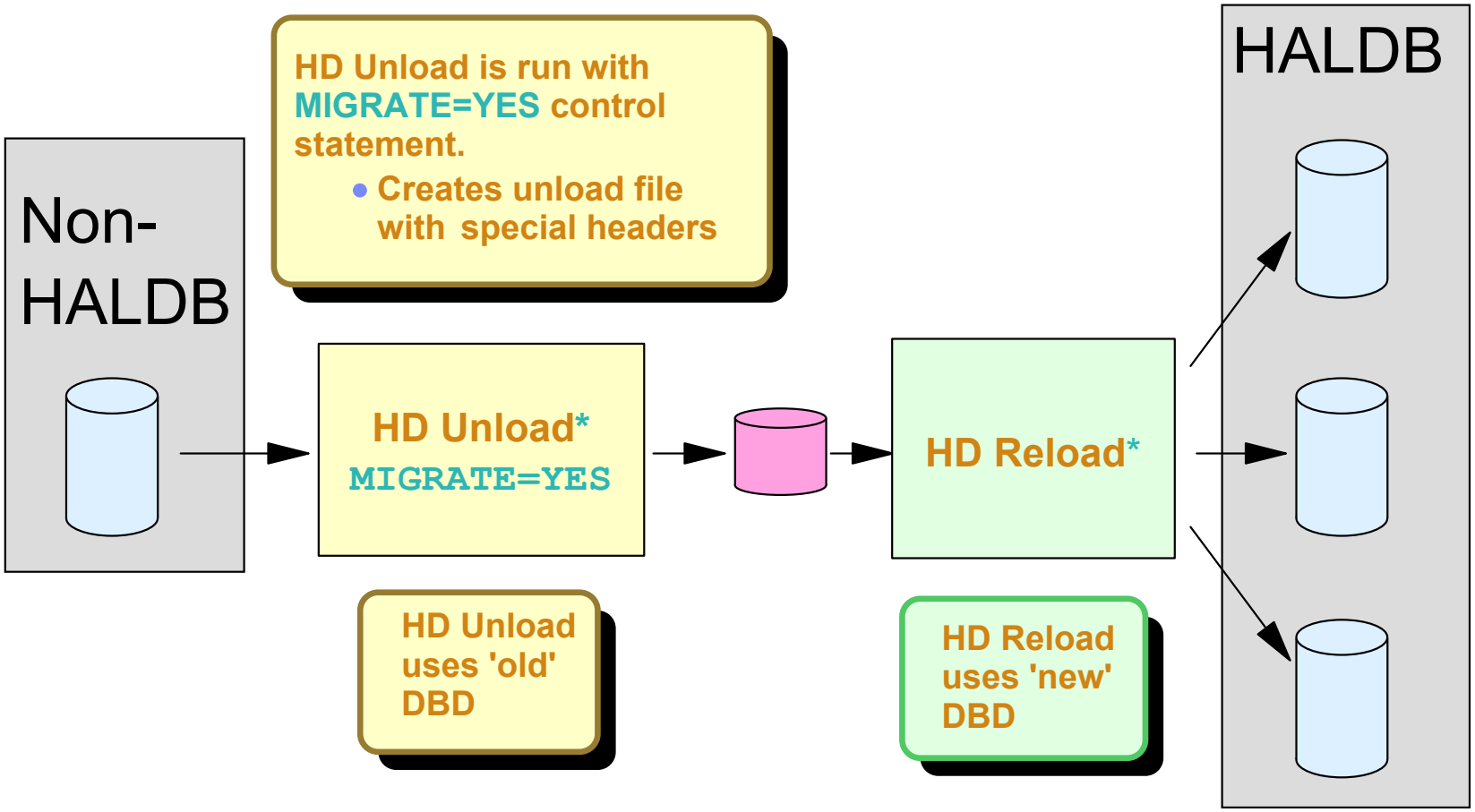
```
nnn       - DBPCB number
dddddddd  - DBPCB label or name
ppppppppp - partition name
```

  - one card per PCB, multiple cards allowed
- SPE - PQ57313/ PQ65496 (IMS V7), PQ58600/PQ65489 (IMS V8)

HALDB Migration

- Migration
  - Uses Prereorg, HD Unload, and HD Reload utilities with new control statements
  - Databases logically related to each other must be migrated together
    - Logical relationships between HALDB and non-HALDB databases are not allowed
  - Secondary indexes must be migrated with the databases to which they point
    - Only HALDB secondary indexes may be used with HALDB databases
  - Migration Aid Utility
    - Provides statistical information about space requirements, key ranges, suggested partition boundaries,...

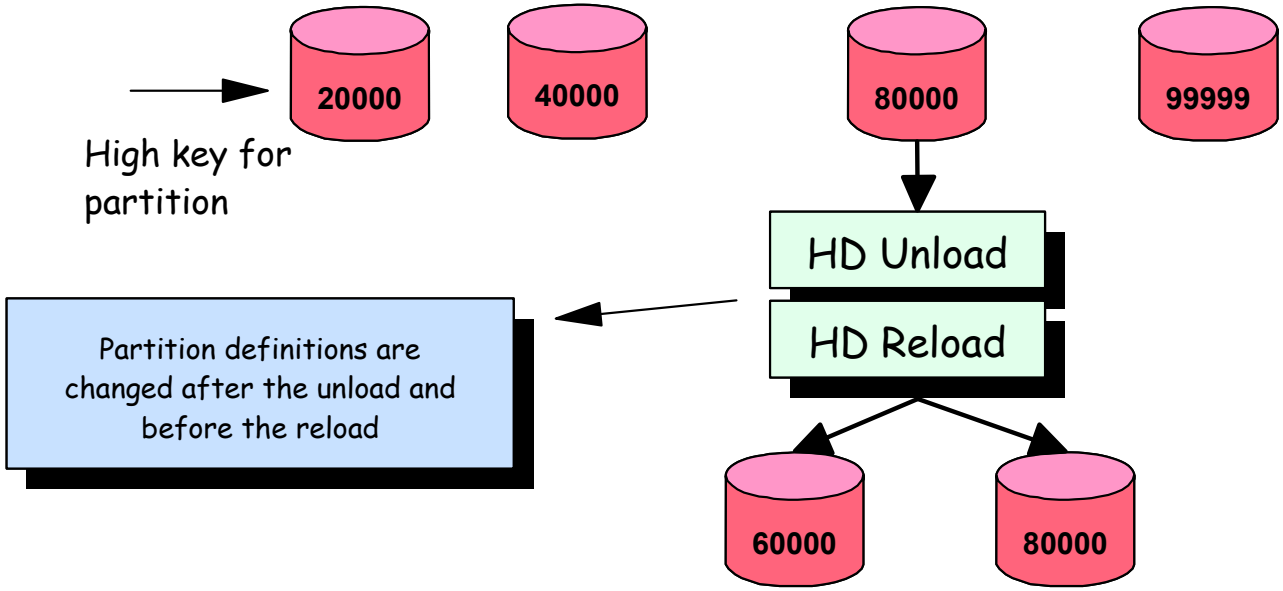
Migration to HALDB



* IBM HP Unload and HP Reload may be used in place of HD Unload / HD Reload

Adding, Deleting, and Changing Partitions

- Partition changes
  - Partitions may be added and deleted
  - Partition boundaries may be changed
- Partition changes are made with HD Unload and HD Reload
  - Only changed partitions are unloaded and reloaded
  - Other partitions remain available during the process !



HALDB Support

- HALDB is supported with:
  - Data sharing
  - Remote Site Recovery (RSR)
  - Extended Recovery Facility (XRF)
  - Online Change
  - OSAM Sequential Buffering
  - IMS Monitor and IMS Performance Analyzer
  - ...

DL/I Calls with HALDB

- Database availability information
  - INIT DBQUERY call and priming of database PCB
    - Report database availability
    - Do not report partition availability
  - Database calls to unavailable partitions
    - 'BA' status code or U3303
    - GN after 'BA' will move to next partition

Logging

- No logging of "after images" for PHIDAM indexes
  - Rebuilt with DFSPREC0 utility
  - "Before images" are not archived
- EPS adjustments are not backed out
- Database change log records include partition name instead of master database name
- No logging for ILDS
  - Only updated by HD Reload utility

HALDB Database Candidates

- Very large databases
  - Approaching 4GB (VSAM) or 8GB (OSAM) limitations
    - To allow for growth
    - To make databases more manageable
  - Previously partitioned databases
    - Using IMS/ESA Partition Support Product (PDB)
    - User partitioning

HALDB Database Candidates

- Medium and large databases
  - Parallel processing to meet time deadlines
    - Application programs
    - Utilities

HALDB Database Candidates

- Any size database
  - Online Reorganization (with IMS V9)
  - Faster reorganizations
    - May be done more frequently
  - Partition independence
    - Making only parts of the data unavailable for database maintenance
  - HIDAM to PHIDAM conversion
    - Log reduction for prime index
    - No image copies of prime index

Advantages of A Single Partition HALDB

- Self-Healing Pointers make reorganized data available sooner
- Primary Index of PHIDAM is automatically defined and partitioned
- Partitioned Database Definitions stored in Recon
  - Certain database changes can be made more non-disruptively
    - Reduced need for online change
- Log Reduction
  - No redo log records for changes to PHIDAM primary index
    - Updates to the ILDS are not logged
- Data sets dynamically allocated without MDA members
- Easy to add partitions as the need arises
  - and take advantage of partition independence and capacity increases
- Online Reorganization with IMS V9

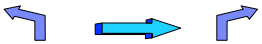
HALDB Summary

- Large Database

  - Databases are partitioned
    - Up to 1001 partitions per database
    - Partitions have up to 10 data set groups

- High Availability Database

  - Partition independence
    - Allocation, authorization, reorganization, and recovery are by partition
  - Self healing pointers



    - Reorganization of partition does not require changes to secondary indexes or logically related databases

HALDB Summary

- Benefits
  - Greater database capacity
    - Without application changes
  - Increased database availability
    - Partitions, not databases, are removed from system
    - Shortened reorganization process
    - Batch window is shortened with concurrent processing
  - Improved manageability
    - Data sets may be smaller