RETURN TO INDEX

E53

Vern L Watts

IMS V7 High Availability/Large Database Support

Anaheim, California October 23 - 27, 2000

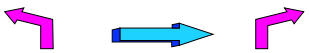
vlwatts@us.ibm.com

HALDB (High Availability Large Database)

- **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

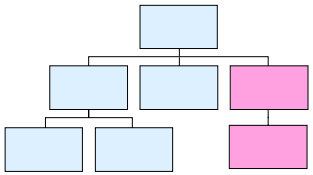


HALDB (High Availability Large Database)

- **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
  - Enhanced usability
    - ISPF utility for partition definitions

Highlights

- **Hierarchic structure is maintained**
  - A database record resides in one partition



- **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

- **New database types**
  - PHDAM - partitioned HDAM
  - PHIDAM - partitioned HIDAM
    - Index is partitioned
  - PSINDEX - partitioned secondary index

Highlights

- **OSAM and VSAM (ESDS and KSDS) are supported**
- **Partition selection is done by key or user exit routine**
- **Logical relationships and secondary indexes are supported**
  - Secondary indexes may be partitioned
- **DBRC is required**
  - Databases must be registered
- **Dynamic allocation uses DBRC information**
  - DFSMDA is not used

Definition Process

- **DBDGEN**
  - Used to define database
    - Hierarchic structure, data set group boundaries, pointer options, logical relationships, secondary indexes,...
- **HALDB Partition Definition Utility**
  - ISPF based
  - Used to define partitions in database
    - Partition selection, space characteristics, randomizers,...

Indirect Pointers

- **HALDB uses both direct and indirect pointers**
  - Direct pointers point to segments
  - Indirect pointers "point" to Indirect List Entries (ILEs) in Indirect List Data Set (ILDS)
  - ILEs have token for key
    - ILEs contain direct pointer to segment
  - ILDS is KSDS associated with a Partition

HALDB Database Structure

- **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

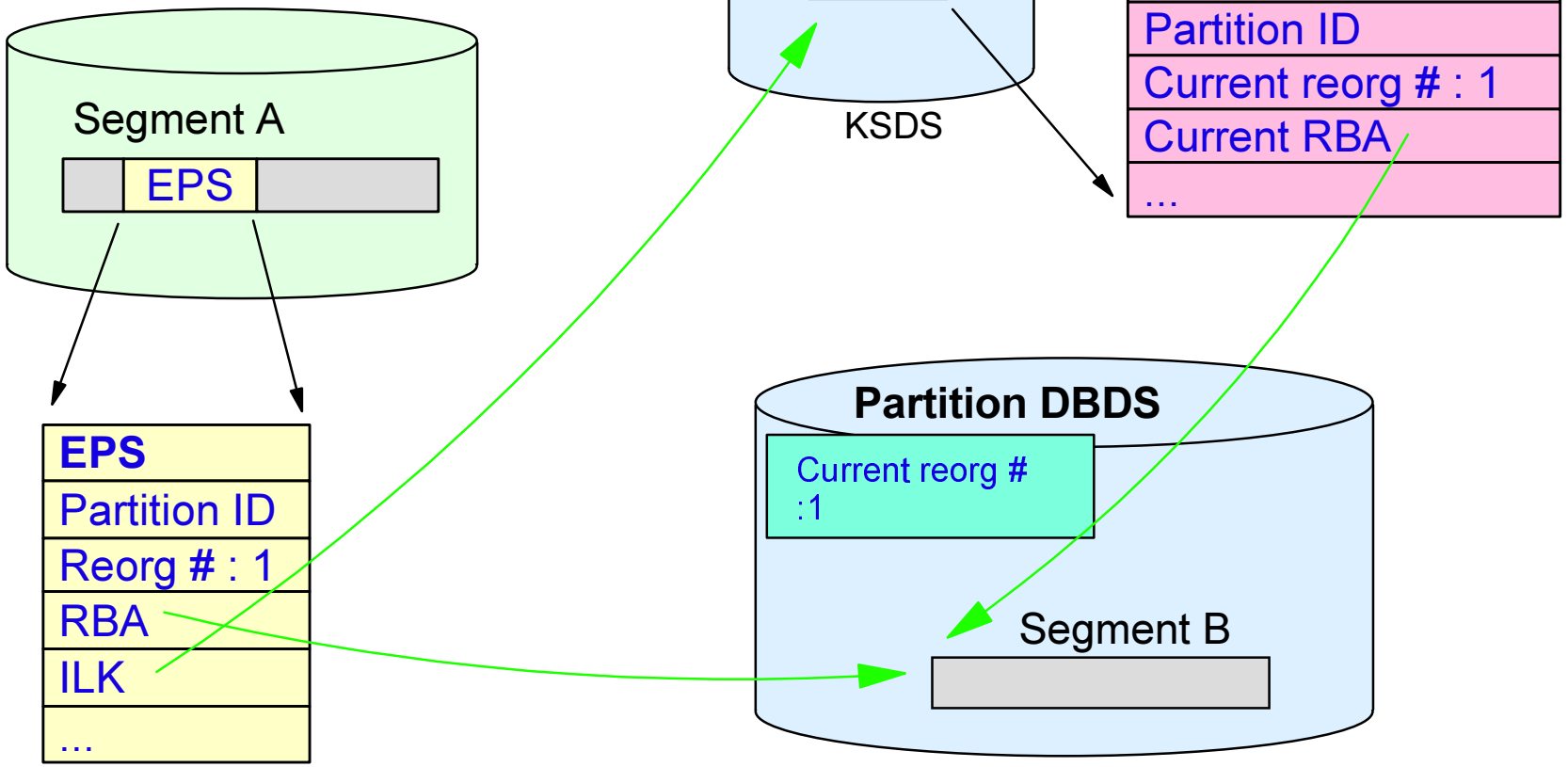
Extended Pointer Set

- **Extended Pointer Set (EPS) is used for logical relationships and secondary indexes**
  - EPS is not updated by reorganizations!
  - 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
  - Direct pointer and reorg number in EPS are updated when ILE is used
- **Self healing pointers!**

Self-Healing Pointers

Using an Extended Pointer Set (EPS)

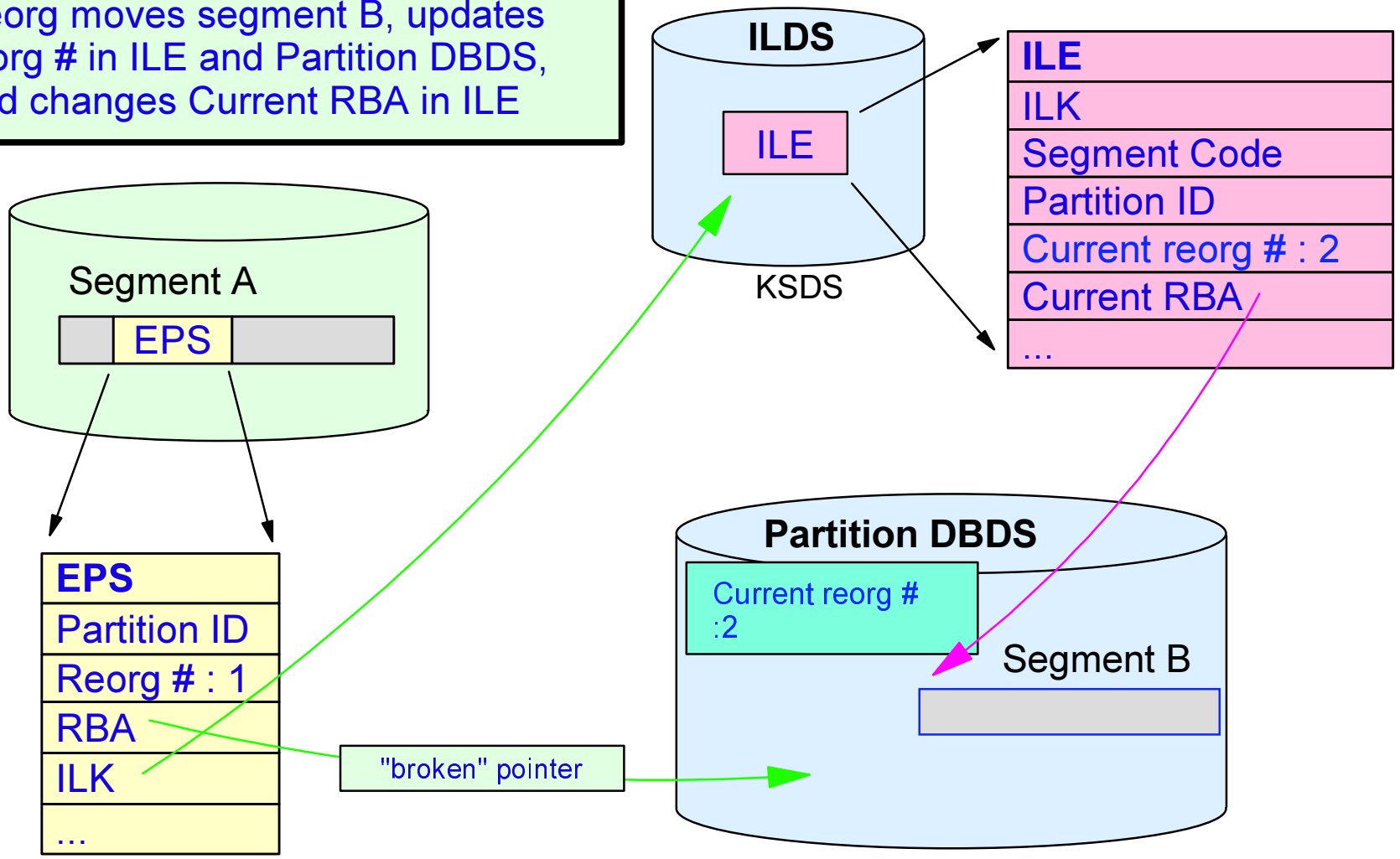
Since reorg # in EPS matches reorg # in Partition DBDS, we use EPS RBA pointer



Self-Healing Pointers

After reorganization of Partition

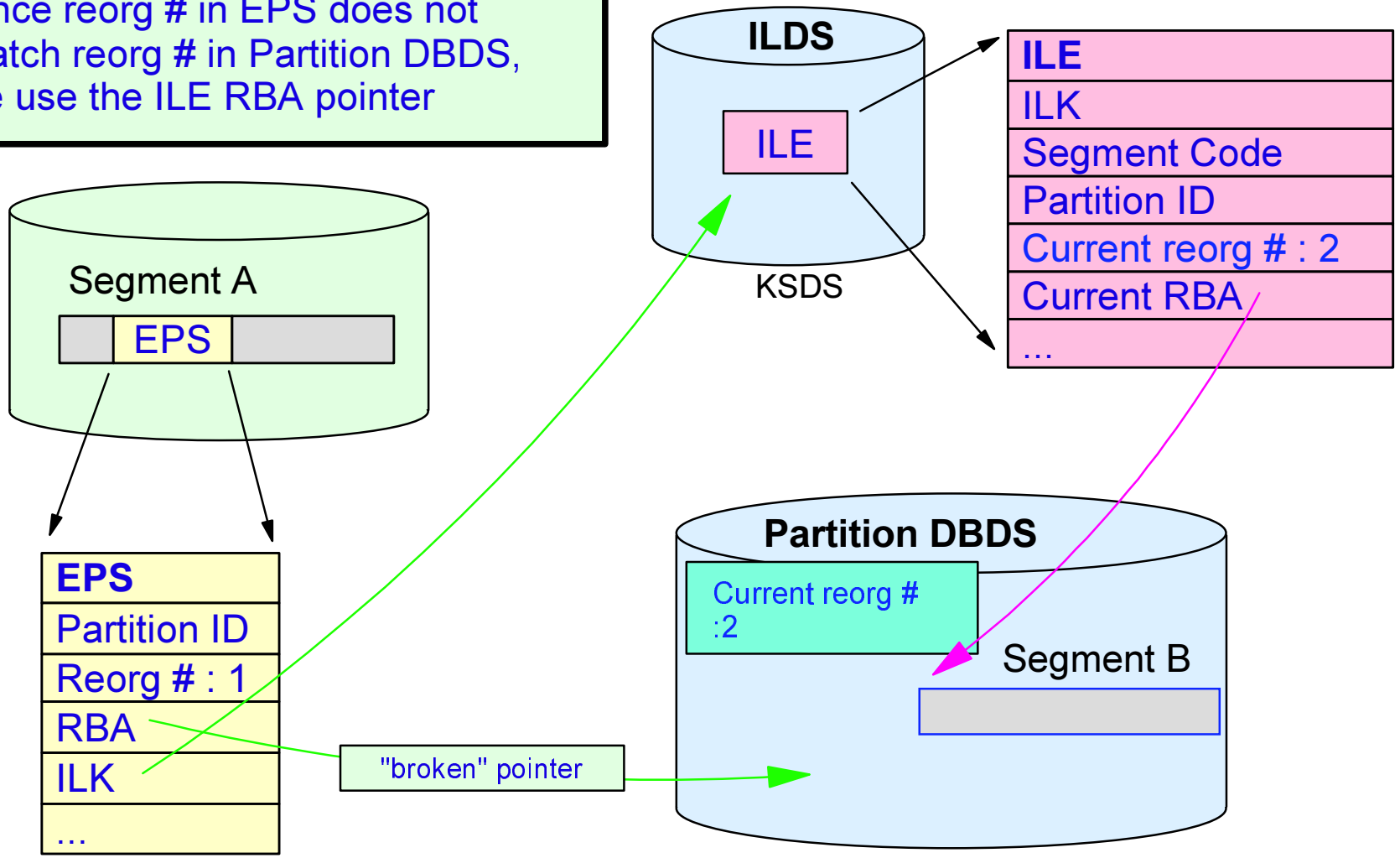
Reorg moves segment B, updates reorg # in ILE and Partition DBDS, and changes Current RBA in ILE



Self-Healing Pointers

Using the EPS after the reorganization

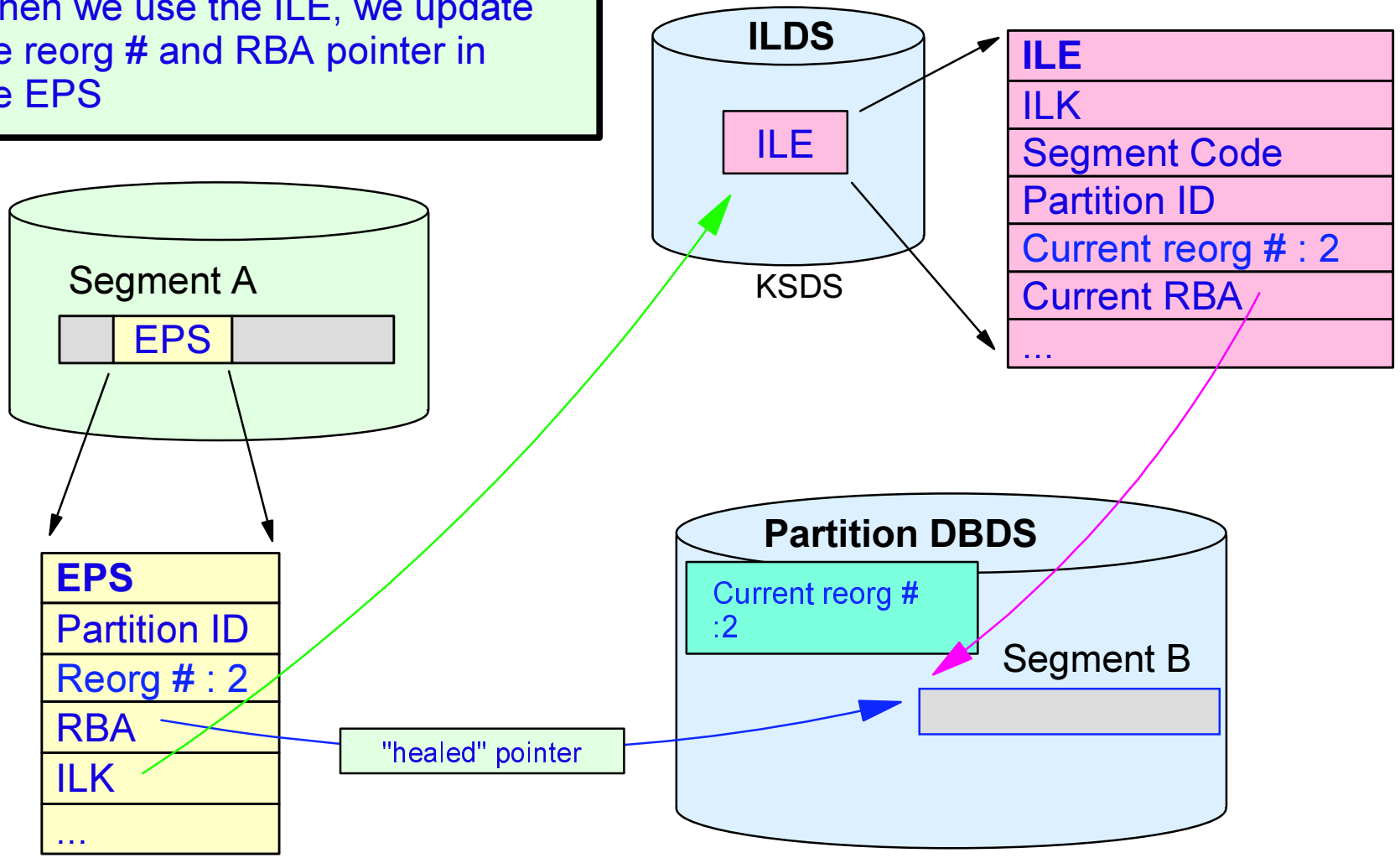
Since reorg # in EPS does not match reorg # in Partition DBDS, we use the ILE RBA pointer



Self-Healing Pointers

"Healing" the EPS

When we use the ILE, we update the reorg # and RBA pointer in the EPS



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

ILDS Data Sets

- **Indirect List Entries (ILEs)**
  - Created or updated by reorg reload
    - Reorgs do not update pointers in segments
  - Not created or updated by non-reload processing
    - This processing updates pointers in segments
  - Initial load does not create ILEs

ILDS Data Sets

- **ILE keys (9 bytes)**
  - ILK (8 bytes)
    - RBA of segment at its creation time (4 bytes)
    - Partition id at creation time (2 bytes)
    - Reorg number at creation time (2 bytes)
  - Segment code (1 byte)

ILDS Data Sets

- **ILE data (50 bytes)**
  - Key (ILK and segment code)
  - Flags
  - Old and new copies of:
    - Partition ID
    - Reorg number
    - Pointer to logical parent or sec. index target
    - Database record lock ID for segment
    - Pointer to paired logical child for physical pairing

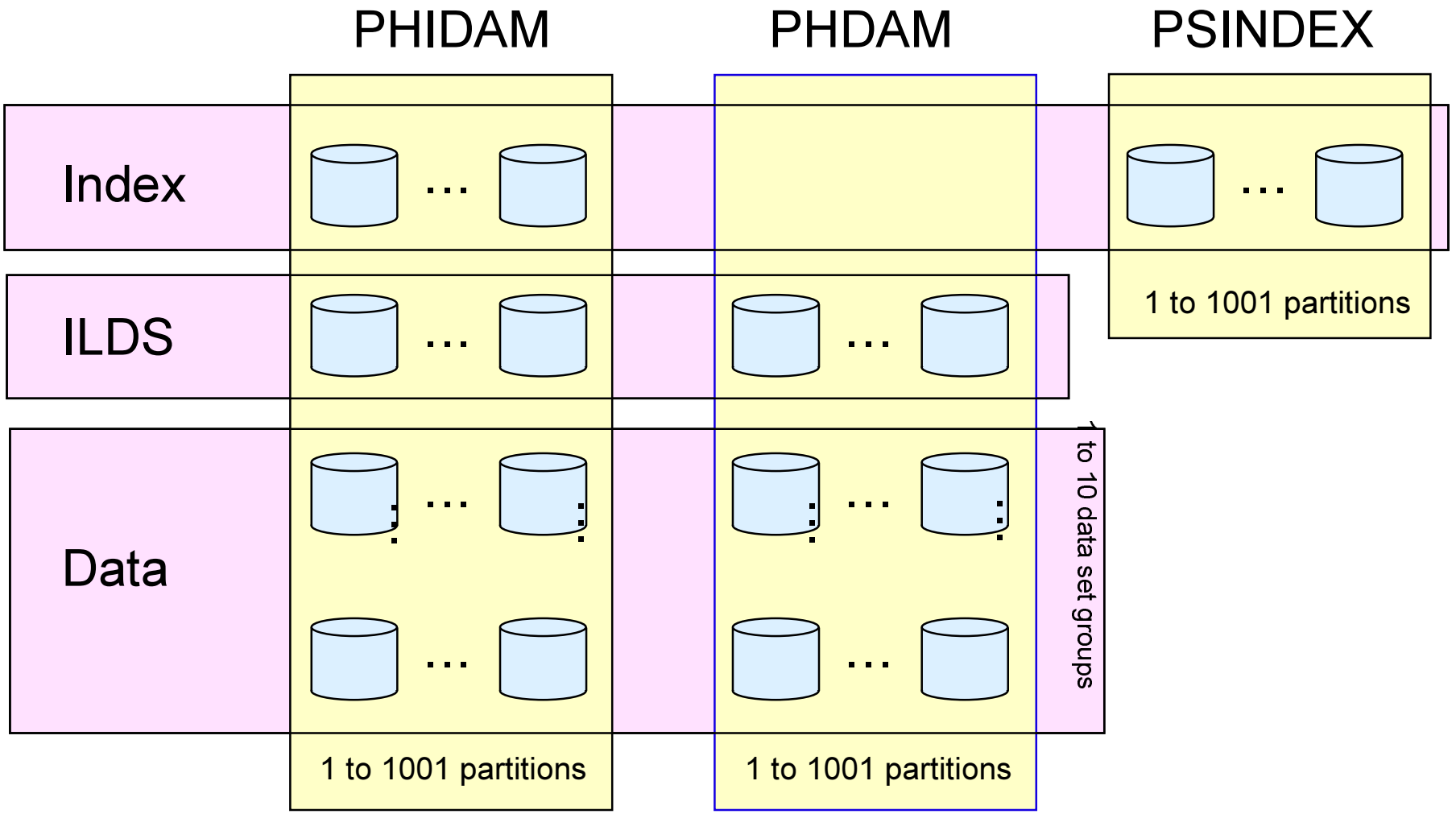
Database Structures

- **PHIDAM prime indexes are not separately defined**
  - Defined as part of the PHIDAM database
    - Applies to DBDGEN and system definition
- **Parent pointers**
  - All 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 Data Sets



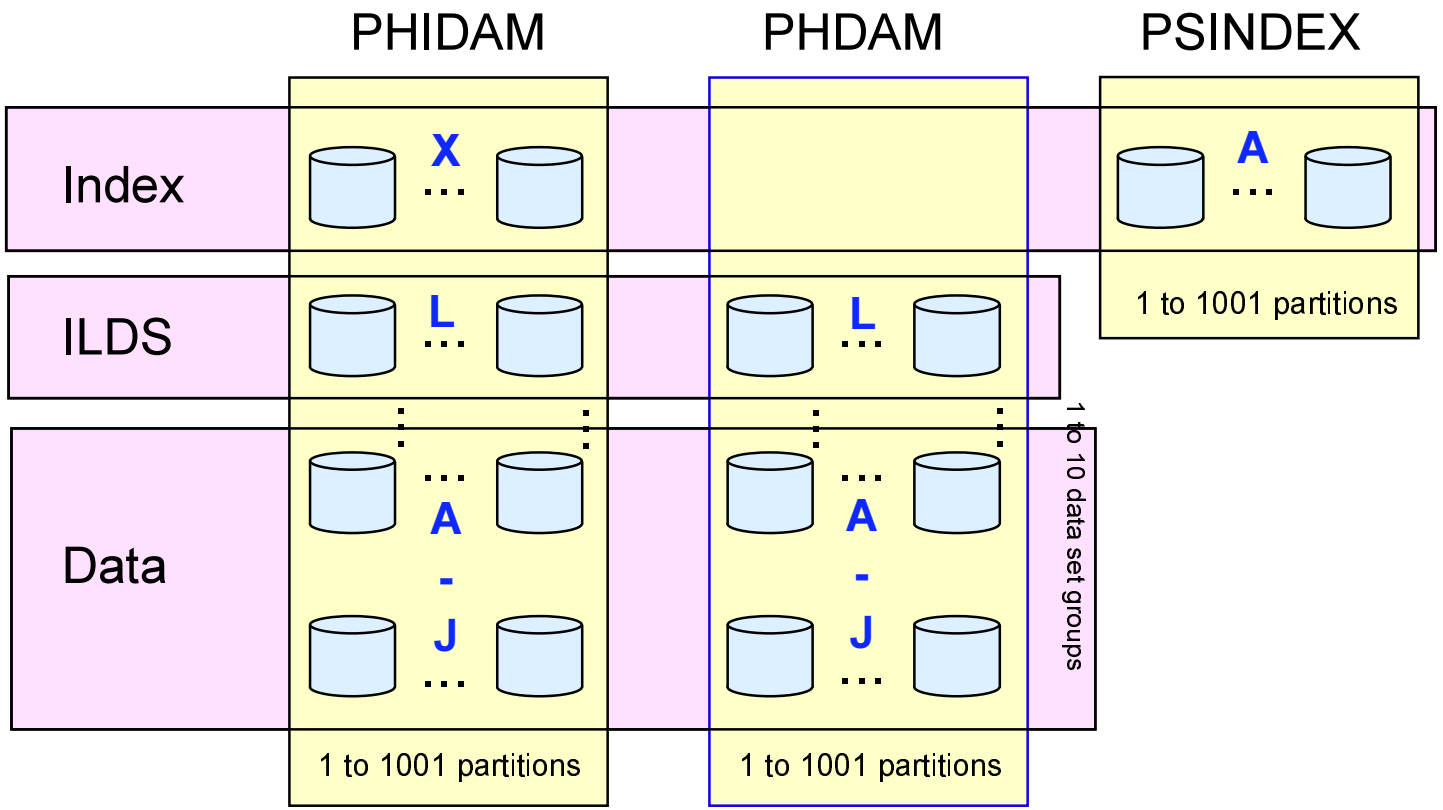
HALDB Database Data Sets

- **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 size limitations**
  - Maximum data set size is 4GB
    - Applies to OSAM and VSAM
- **OSAM block sizes must be even**

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 through J - Data data sets
- A - PSINDEX

Database Data Sets

- **Data set names**
  - Begin with data set name prefix for the partition
    - Up to 37 characters
    - Assigned 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
  - Example:
    - Partition data set name prefix IMP0.DB.INV23.FR
    - Partition ID: 00004
    - Data set name PHIDAM Index: IMP0.DB.INV23.FR.X00004

Database Data Sets

- **DDNAMEs**
  - Partition name is basis for DDNAME
    - Up to 7 characters
    - Assigned in HALDB Partition Definition Utility
  - Letter is used as suffix
    - X for PHIDAM index
    - L for ILDS
    - A for PSINDEX
    - A through J for data
  - Example:
    - Partition name: FRANCE
    - DDNAME for PHIDAM Index: FRANCEX

Partition DDNAMEs and Data Set Names

Partition_name assigned by user in HALDB Partition Definition Utility

DSN_prefix assigned by user in HALDB Partition Definition Utility

PartitionID assigned by IMS in HALDB Partition Definition Utility

Data set	DDNAME	Data Set Name
Data set group 1	Partition_nameA	DSN_prefix.ApartitionID
Data set group 2	Partition_nameB	DSN_prefix.BpartitionID
Data set group 3	Partition_nameC	DSN_prefix.CpartitionID
...	...	...
Data set group 10	Partition_nameJ	DSN_prefix.JpartitionID
ILDS	Partition_nameL	DSN_prefix.LpartitionID
PHIDAM Index	Partition_nameX	DSN_prefix.XpartitionID
Secondary Index	Partition_nameA	DSN_prefix.ApartitionID

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 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
  - Concurrent processing of multiple partitions allowed

HALDB Overview

- **Migration**
  - Uses Prereorg, HD Unload, and HD Reload utilities with new control statements
  - Databases logically related to each other must be migrated together
  - Secondary indexes must be migrated with the databases to which they point
  - Migration Aid Utility
    - Provides statistical information about space requirements, key ranges, suggested partition boundaries,...

HALDB Overview

- **Fallback**
  - Fallback from HALDB to HIDAM, HDAM, and secondary indexes is supported
  - Uses Prereorg, HD Unload, HD Reload, Prefix Resolution, and Prefix Update utilities with new control statements

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

Definition Tasks

- **DBDGEN**
  - Defines the master database
- **HALDB Partition Definition Utility**
  - Defines the partitions
  - Registers the database and partitions in RECONs
- **System Definition**
  - Specifies the database to the online system
- **DFSVSMxx and DFSVSAMP DD**
  - Assigns data sets to buffer pools

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

DL/I Calls with HALDB

- **Cannot initially load logical child segments**
  - LF status code returned if attempted
  - Log. child segments may be inserted by update programs
  - Log. child segments may be reloaded
- **PHIDAM with Partition Selection Exit routine**
  - Root segments are not necessarily in key sequence when crossing partition boundaries
    - Segments are in sequence within a 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

OSAM Sequential Buffering

- **HALDB supports OSAM Sequential Buffering (SB)**
  - Same support as for non-HALDB databases
  - Recommended for batch sequential processing
- **SB monitors by data set**
  - Sequential buffering is turned on/off due to access patterns
  - Sequential buffering may be invoked for some, but not all partitions
    - Well-organized vs. badly-organized partitions

Data Set Groups

- **HALDB supports data set groups (DSGs)**
- **DSGs are not needed to address database size limitations**
  - With non-HALDB databases, DSGs were often used to deal with data set size limitations
- **DSGs remain useful**
  - May improve performance by moving rarely accessed data to another data set
- **Reevaluate need for DSGs when migrating databases to HALDB**

Segment Edit/Compression Exit Routines

- **Segment Edit/Compression Exit Routines**
  - Defined for entire database
    - Defined on DBDGEN SEGM statement
  - All partitions in database use the same exit routines

Restrictions on Number of Data Sets

- **IMS allows up to 10,000 allocated data sets per address space**
  - Limit includes system data sets
  - Could limit number of HALDB data sets
- **With DFSMS 1.3 and 1.4, IMS allows up to 8,183 open full function database data sets**
  - Could limit number of HALDB data sets
  - Restriction is removed in DFSMS 1.5

Data Set Allocation

- **Dynamic allocation uses DBRC information**
  - DFSMDA is not used
- **DD statement allocation may be used**
  - If DD statement is not present, dynamic allocation is used
- **Dynamic allocation is always available for batch**
  - Specification of 'NODYNALLOC' in DFSVSAMP is ignored for HALDB

Data Sharing

- **Data sharing is supported**
  - SHARELVL specified on master database level
    - Authorization is by partition
  - Lock resource names
    - Master database global DMB number used for all partitions
    - Partition IDs added at end of lock resource name

RSR

- **Remote Site Recovery is supported**
  - GSG is specified by database, not partition
  - Tracking level is specified by database, not partition
  - ILDS and prime index data sets are not tracked
    - There are no log records for them
    - They are rebuilt with DFSPREC0 at tracking site

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

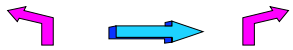
- **Any size database**
  - Faster reorganizations
    - May be done more frequently
  - Partition independence
    - Making only parts of the data unavailable for database maintenance
  - HIDAM to PHIDAM conversion
    - Log reduction for prime index
    - No image copies of prime index

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

RETURN TO INDEX

- **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
  - Enhanced usability
    - ISPF utility for partition definitions